Date: December 2022

OMG Systems Modeling Language $^{\text{TM}}$ (SysML®) Annex C: SysML v1 to SysML v2 Transformation

Version 2.0
Release 2022-11

Submitted in response to Systems Modeling Language (SysML®) v2 RFP (ad/2017-12-02) by:

88Solutions Corporation	Lockheed Martin Corporation
Dassault Systèmes	MITRE
GfSE e.V.	Model Driven Solutions, Inc.
IBM	PTC
INCOSE	Simula Research Laboratory AS
Intercax LLC	Thematix Partners

Table of Contents

List of Tables

C Annex: SysML v1 to SysML v2 Transformation

(Informative)

C.1 General

C.1.1 Overview

This annex describes a transformation that specifies a semantic translation from SysML v1 [SysMLv1] to SysML v2 in a precise way. (In this annex, "SysML v1" refers to SysML v1.7, the last version of SysML prior to v2.0, and "SysML v2" refers to SysML as defined in this specification.)

The main intent is to provide the rules on which automated conversions of SysML v1 models to the SysML v2 standard can be developed. In addition, this annex can be considered an educational document that provides useful information for people who would like to compare using SysML v2 and using SysML v1.

More sophisticated applications of this transformation can also be envisaged. For instance, a SysML v1 conformant tool could use this transformation to implement a limited subset of the SysML v2 API that will provided "SysMLv2-like" read-only access to its SysMLv1 models for external applications.

Release Note. The transformation specification currently only covers a restricted scope, which will be extended in the final submission.

C.1.2 Mapping Approach

The SysML v1 to v2 transformation is specified by directional mappings between UML metaclasses and stereotypes that are part of the SysML v1 specification and the set of the metaclasses included in KerML and the SysMLv2 libraries.

Each mapping is a directed relationship that reifies a semantic link between a concept belonging to the SysMLv1 scope on the source side and one concept belonging to the SysMLv2 scope on the target side. As a set, the mappings specify a formal transformation that describes how the information encoded by the SysMLv1 concepts can be reliably represented using constructs of SysMLv2 metaclasses instances.

In this approach, a mapping is represented by a UML class that has a pair of associations. One provides the "from" end that designates the source SysML v1 concept while the other provides the "to" end that designates the target SysML v2 metaclass.

In addition to those associations, a mapping class provides a set of operations defining how the attribute values of the target metaclass instance have to be computed based on attribute values reachable from the source object. The computation algorithm is provided by the body condition of those operations and expressed using OCL code.

Note that the values assigned to attributes of the target object shall be instances of the target (i.e., SysMLv2) metamodel, coming themselves from transformations of SysMLv1 objects to SysMLv2 objects. The getMapped static operation is provided for this purpose. It returns a (possibly null) value, based on the type of the target metaclass.

Each mapping specification enables the transformation of any object that has the type specified by the “from” role to an object of the type specified by the “to” role, as long as it is not overloaded by a more specific mapping definition. In other words, assume a mapping is specified as the class "A" (i.e., that has A typing its "from" property), then it applies to any instance of a class B if B is a subclass of A and if there is no specialization of that mapping class specified for B (i.e., that has B typing its "from" property).

It is possible to restrict the applicability of a mapping specification to a specific subset of objects. This is achieved by the "filter" static operation that is evaluated against each candidate object. Only objects for which this "filter" operation returns “true” shall be translated according to the specifications of that mapping class. By default, the filter operation always returns “true”.

Some mapping classes have one or more qualifiers for their "to" attribute. In such a case, each of those qualifiers reflect the specific attribute of the source type (i.e. the type of the "from" attribute) that has the same name and the same type. For those specific mappings, it is expected to get one instance of the target class (as specified by the type of the "to" attribute") for each combination of value of those attributes per instance of object of the source type, assuming they pass the applicability filter as described above.

C.2 Mappings

C.2.1 Overview

C.2.2 Mapping Helper and Library

C.2.2.1 Helper

Description

The Helper class contains operations that are used by multiple mapping classes. The specification is in the bodyCondition.

Operations

- actionOwnedRelationship (in src : Element) : Relationship [0..*]
  Reusable mapping rule for owned relationships of a SysMLv1::Action mapping.
  bodyCondition:

```
result =
let actionInputPin: Set(UML::Element) = src.ownedElement->select(e | e.oclIsTypeOf(UML::Actio
let triggers: Set(UML::Element) = src.ownedElement->select(e | e.oclIsKindOf(UML::Trigger)) i
let toElementFMS: Set(UML::Element) = src.ownedElement->select(e | e.oclIsKindOf(UML::Pin)) i
let toElementOMS: Set(UML::Element) = (((src.ownedElement - toElementFMS) - actionInputPin) -
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
```

- activityOwnedRelationship (in src : Element) : Relationship [0..*]
  Reusable mapping rule for owned relationships of a SysMLv1::Activity mapping.
  bodyCondition:

```
result =
let initialNodes : Set(UML::Element) = src.ownedElement->select(e | e.oclIsKindOf(UML::Initia
let finalNodes : Set(UML::Element) = src.ownedElement->select(e | e.oclIsKindOf(UML::FinalNod
let elementsFMS : Set(UML::Element) = ((src.ownedElement->select(e | e.oclIsKindOf(UML::Contr
let parameters: Set(UML::Parameter) = src.ownedElement->select(e | e.oclIsKindOf(UML::Paramet
let ignoreParameterNodes: Set(UML::ActivityParameterNode) = src.ownedElement->select(e | e.oc
let ignoreActivityPartition: Set(UML::ActivityPartition) = src.ownedElement->select(e | e.ocl
let ignoreInterruptibleActivityRegion: Set(UML::InterruptibleActivityRegion) = src.ownedEleme
let ownedClassifier: Sequence(UML::Classifier) = src.ownedElement->select(e | e.oclIsKindOf(U
let variables: Sequence(UML::Variable) = src.ownedElement->select(e | e.oclIsKindOf(UML::Vari
let parameterSets: Set(UML::ParameterSet) = src.ownedElement->select(e | e.oclIsKindOf(UML::P
let elementsOMS: Set(UML::Element) = (((((((((src.ownedElement-initialNodes)-finalNodes)-el
let memberships : Sequence(UML::Element) =
elementsOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(initialNodes->collect(e | InitialNodeMembership_Mapping.getMapped(e)))
```

```
->union(finalNodes->collect(e | ActivityFinalNodeMembership_Mapping.getMapped(e)))
->union(elementsFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(variables->collect(e | VariableMembership_Mapping.getMapped(e)))
->union(parameterSets->collect(e | ParameterSetMembership_Mapping.getMapped(e)))
->union(ownedClassifier->collect(e | ElementOwningMembership_Mapping.getMapped(e))) in
if src.classifierBehavior.oclIsUndefined() then memberships else memberships->append(Classifi
```

- createUUID () : String [1]
  Creates a UUID. The specification is implementation-specific and therefore cannot provided here.
- getAppliedStereotypes (in element : Element) : Stereotype [0..*]
  Returns the list of applied stereotypes. The specification is implementation-specific and therefore cannot provided here.
- getEnumerationType (in t : Enumeration) : EnumerationDefinition [1]
  Maps a given SysMLv1::Enumeration to the appropriate SysMLv2::EnumerationDefinition.
  bodyCondition:

```
result =
if t.name = 'VerdictKind' then SYSML2::EnumerationDefinition.allInstances()->any(e | e.qualif
SYSML2::EnumerationDefinition.allInstances()->any(e | e.qualifiedName = 'SysMLv1Library::Enum
```

- getID (in src : Element) : String [1]
  Returns the identifier of a SysMLv1::Element. The specification is implementation-specific and therefore cannot provided here.
- getKerMLFeatureDirectionKind (in v : EnumerationLiteral) : FeatureDirectionKind [1]
  Maps a given SysMLv1 feature direction enumeration literal to a SysMLv2::FeatureDirectionKind enumeration literal.
  bodyCondition:

```
result = if v.enumeration.qualifiedName = 'SysML::Ports&Flows::FeatureDirectionKind' or
                v.enumeration.qualifiedName = 'SysML::Ports&Flows::FeatureDirection' then
    if v = SysML::FeatureDirectionKind::provided then
        KerML::FeatureDirectionKind::_'out'
    else if (v = SysML::FeatureDirectionKind::required) then
        KerML::FeatureDirectionKind::_'in'
    else if (v = SysML::FeatureDirectionKind::providedRequired) then
        KerML::FeatureDirectionKind::inout
    else
        invalid
    endif endif endif
else
    invalid
endif
```

- getKerMLParameterDirectionKind (in v : ParameterDirectionKind) : FeatureDirectionKind [1]
  Maps a given SysMLv1 parameter direction enumeration literal to a SysMLv2::FeatureDirectionKind enumeration literal.
  bodyCondition:

```
result = if v = UML::ParameterDirectionKind::_'in' then
    KerML::FeatureDirectionKind::_'in'
else if (v = UML::ParameterDirectionKind::return) then
    KerML::FeatureDirectionKind::out
else if (v = UML::ParameterDirectionKind::out) then
    KerML::FeatureDirectionKind::out
else if (v = UML::ParameterDirectionKind::inout) then
    KerML::FeatureDirectionKind::inout
else
```

```
    invalid
endif endif endif endif
```

- getKerMLVisibilityKind (in v : VisibilityKind) : VisibilityKind [1]
  Maps a given SysMLv1::VisibilityKind enumeration literal to a SysMLv2::VisibilityKind enumeration literal.
  bodyCondition:

```
result = if (v = UML::VisibilityKind::public) then
    KerML::VisibilityKind::public
else if (v = UML::VisibilityKind::protected) then
    KerML::VisibilityKind::protected
else if (v = UML::VisibilityKind::private) then
    KerML::VisibilityKind::private
else if (v = UML::VisibilityKind::package) then
    KerML::VisibilityKind::public
else
    invalid
endif endif endif endif
```

- getMetadataByName (in mdName : String) : AttributeDefinition [1]
  Returns the metadata attribute definition element for a given metadata name.
  bodyCondition:

```
result = SYSML2::AttributeDefiniton.allInstances()->any(e | e.name = mdName)
```

- getRequirementStereotype (in element : NamedElement) : Stereotype [0..1]
  Returns the requirement stereotype for a given element.
  bodyCondition:

```
result = let stereotypes: Set(UML::Stereotype) = Helper.getAppliedStereotypes(element) in
stereotypes->any(s | s.general->collect(g | g.qualifiedName)->includes('SysML::Requirements::
```

- getScalarValueType (in t : DataType) : DataType [1]
  Maps a given SysMLv1 primitive type to a SysMLv2 scalar value type.
  bodyCondition:

```
result =
if t.name = 'UnlimitedNatural' then
SYSML2::DataType.allInstances()->any(e | e.qualifiedName = 'ScalarValues::Natural')
else
SYSML2::DataType.allInstances()->any(e | e.qualifiedName = 'ScalarValues::' + t.name)
endif
```

- getScalarValueTypeByName (in ptName : String) : DataType [1]
  Maps a given SysMLv1 primitive type name string to a SysMLv2 scalar value type.
  bodyCondition:

```
result = SYSML2::DataType.allInstances()->any(e | e.qualifiedName = 'ScalarValues::' + ptName
```

- getSysMLv2EnumerationDefinition (in v1Enumeration : Enumeration) : EnumerationDefinition [1]
  Maps a given SysMLv1::Enumeration to the appropriate SysMLv2::EnumerationDefinition.
  bodyCondition:

```
result = if v1Enumeration = UML::ParameterDirectionKind then
    KerML::FeatureDirectionKind
else if v1Enumeration.qualifiedName = 'SysML::Libraries::ControlValues::ControlValueKind' the
    SYSML2::EnumerationDefinition.allInstances()->any(e | e.qualifiedName = 'SysMLv1Library::
```

```
else
    invalid
endif endif
```

- getTagValue (in element : Element, in stereotypeName : String, in tagValueName : String) [1]
  Returns the value of a stereotype property. The specification is implementation-specific and therefore cannot provided here.
- getTagValue2 (in element : Element, in stereotype : Stereotype, in tagValueName : String) [1]
- getTagValueAsElement (in element : Element, in stereotypeName : String, in tagValueName : String) : Element [1]
  Returns the value of a stereotype property. The specification is implementation-specific and therefore cannot provided here.
- getTagValueAsElementColl (in element : Element, in stereotypeName : String, in tagValueName : String) : Element [0..*]
  Returns the value of a stereotype property as a collection. The specification is implementation-specific and therefore cannot provided here.
- getTagValueAsString (in element : Element, in stereotypeName : String, in tagValueName : String) : String [1]
  Returns the value of a stereotype property as a string. The specification is implementation-specific and therefore cannot provided here.
- getTagValueAsStringColl (in element : Element, in stereotypeName : String, in tagValueName : String) : String [0..*]
  Returns the value of a stereotype property as a string collection. The specification is implementation-specific and therefore cannot provided here.
- getV1V2Lib_PartUsage (in name : String) [0..1]
- globalNamespace () : Namespace [1]

  bodyCondition:

  ```
  result = KerML::Package.allInstances()->any(p | p.owningNamespace->isEmpty())
  ```

- hasStereotypeApplied (in element : Element, in stereotypeName : String) : Boolean [1]
  Returns true if the given stereotype is applied to the element. The specification is implementation-specific and therefore cannot provided here.
- isConnectionDef (in association : Association) : Boolean [1]
  Checks if a SysMLv1::Association is mapped to a SysMLv2::ConnectionDefinition.
  bodyCondition:

  ```
  result =
  -- Case 1: composite association with multiplicity 1..1 on owner side
  let case1: Boolean = association.memberEnd->exists(e | not e.isComposite and e.lower=1) and
  association.memberEnd->exists(e | e.isComposite) in

  -- Case 2: association is not composite and there is no owned end with multiplicity 0..*
  let case2: Boolean = not association.memberEnd->exists(e | e.isComposite) and
  not association.ownedEnd->exists(e | e.lower = 0 and e.upper = -1) in

  association.oclIsTypeOf(UML::AssociationClass) or
  case1 or
  case2
  ```

- isRequirement (in element : Element) : Boolean [1]
  Checks whether the stereotype AbstractRequirement is applied to the given element.
  bodyCondition:

```
result = let stereotypes: Set(UML::Stereotype) = Helper.getAppliedStereotypes(element) in
stereotypes->exists(s | s.general->collect(g | g.qualifiedName)->includes('SysML::Requirement
```

- mappedValueSpecification (in valueSpec : ValueSpecification) : Expression [1]

  bodyCondition:

```
result =
if valueSpec.oclIsKindOf(UML::LiteralString) then
    LiteralString_Mapping.getMapped(valueSpec)

else if valueSpec.oclIsKindOf(UML::LiteralBoolean) then
    LiteralBoolean_Mapping.getMapped(valueSpec)

else if valueSpec.oclIsKindOf(UML::LiteralInteger) then
    LiteralInteger_Mapping.getMapped(valueSpec)

else if valueSpec.oclIsKindOf(UML::LiteralUnlimitedNatural) then
    if valueSpec.value = -1 then
        LiteralUnlimitedToUnbounded_Mapping.getMapped(valueSpec)
    else
        LiteralUnlimitedToInteger_Mapping.getMapped(valueSpec)
    endif

else if valueSpec.oclIsKindOf(UML::LiteralReal) then
    LiteralReal_Mapping.getMapped(valueSpec)

else if valueSpec.oclIsKindOf(UML::LiteralNull) then
    LiteralNull_Mapping.getMapped(valueSpec)

else
    invalid

endif endif endif endif endif endif
```

- packageOwnedRelationship (in src : Element) : Relationship [0..*]
  Reusable mapping rule for owned relationships of a SysMLv1::Package mapping.
  bodyCondition:

```
result =
let elementGroups: Set(UML::Comment) = src.ownedElement->select(e | Helper.hasStereotypeAppli
let useCaseAssociations : Set(UML::Association) = src.ownedType->select(e | e.oclIsKindOf(UML
let unmappedAssociations : Set(UML::Association) = src.ownedType->select(e | e.oclIsKindOf(UM

let relationships: Set(SysMLv2::Relationship) = (src.ownedComment-elementGroups)->collect(e |
->union(((src.ownedType-useCaseAssociations)- unmappedAssociations)->collect(e | ElementOwnin
->union(elementGroups->collect(e | ElementGroupMembership_Mapping.getMapped(e)))
->union(src.ownedElement->select(e | e.oclIsKindOf(UML::Dependency) or e.oclIsKindOf(UML::Pac
or (e.oclIsKindOf(UML::InstanceSpecification) and e.oclAsType(UML::InstanceSpecification).cla
->collect(e | ElementOwningMembership_Mapping.getMapped(e))) in

if src.URI.oclIsUndefined() or src.URI = '' then
    relationships
else
    relationships->including(PackageURIMetadataMembership_Mapping.getMapped(src))
endif
```

- stateOwnedRelationship (in src : Element) : Relationship [0..*]
  Reusable mapping rule for owned relationships of a SysMLv1::State mapping.
  bodyCondition:

```
result =
let initialState : Set(UML::Element) = from.ownedElement->select(e | e.oclIsKindOf(UML::Pseud
let toElementOMS : Set(UML::Element) = from.ownedElement - initialState in
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(initialState->collect(e | InitialStateMembership_Mapping.getMapped(e)))
```

C.2.2.2 SysML v1 Library

The SysML v1 library is a SysML v2 model library with metadata definitions for annotating some model elements resulting from a transformation from a SysML v1 model using the SysML v1 to SysML v2 transformation.

```
package SysMLv1Library {

        doc /*
         * The SysMLv1Library defines metadata for SysML elements which cannot mapped to a SysML v2
         */

    metadata def ActivityEdgeData {
            doc /* Metadata definition for UML::ActivityEdge::weight property */
            attribute weight : ScalarValues::Natural;
    }

    metadata def AssociationData {
            doc /* Metadata definition for UML::StructuredClassifiers::Association::isDerived proper
        attribute isDerived : ScalarValues::Boolean;
    }

    metadata def BlockData {
            doc /* Metadata definition for SysML::Blocks::Block::isEncapsulated property */
            attribute isEncapsulated : ScalarValues::Boolean;
    }

    metadata def ElementGroupData {
            doc /* Metadata definition for the criterion of a SysML::ModelElements::ElementGroup */
        attribute criterion : ScalarValues::String;
    }

    metadata def ModelData :> PackageData {
            doc /* Metadata definition for the UML::Model::viewpoint property */
            attribute 'viewpoint' : ScalarValues::String;
    }

    metadata def PackageData {
            doc /* Metadata definition for the UML::Package::URI property */
            attribute URI : ScalarValues::String;
    }

    metadata def ParameterSetData {
        doc /* Metadata definition to tag parameter that the mapping source of the parameter was a U
        attribute isParameterSet : ScalarValue::Boolean;
        }

    metadata def PortData {
            doc /* Metadata definition to tag a SysML v2 port that the mapping source of the port wa
            attribute isFullPort : ScalarValues::Boolean;
    }

    metadata def ViewpointData {
            doc /* Metadata definition for SysML::ModelElements::Viewpoint properties */
            attribute concerns [0..*] : ScalarValues::String;
            attribute languages [0..*] : ScalarValues::String;
            attribute purpose : ScalarValues::String;
            attribute presentations [0..*] : ScalarValues::String;
    }
}
```

C.2.3 Generic Mappings

C.2.3.1 Overview

Generic mappings are partial definitions of transformation rules that are intended to factorize reusable algorithms for making the global specification more compact and easier to read and maintain. Basically, they provide a default value for all the non-derived attributes of their target metaclass wherever possible, or declare an abstract operation for them otherwise. All of them have "UML::Element" defined as their source type. The operations provided by the generic mappings can be redefined by their specialization, as appropriate according to the source type specified by the redefinition of their "from" attribute.

All of those generic mappings are abstract.

C.2.3.2 Generic Mappings To KerML

C.2.3.2.1 GenericToAnnotatingElement_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *AnnotatingElement*.

General Mappings

GenericToElement_Mapping

Mapping Source

Mapping Target

AnnotatingElement

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- AnnotatingElement::annotation () : Annotation [0..*]

```
Set{}
```

C.2.3.2.2 GenericToAnnotation_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *Annotation*.

General Mappings

GenericToRelationship_Mapping

Mapping Source

Mapping Target

Annotation

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Annotation::owningAnnotatedElement () : Element [0..1]

  ```
  null
  ```

- Annotation::annotatingElement () : AnnotatingElement [1]
  abstract rule
- Annotation::annotatedElement () : Element [1]
  abstract rule

C.2.3.2.3 GenericToAssociation_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *Association*.

General Mappings

GenericToRelationship_Mapping
GenericToClassifier_Mapping

Mapping Source

Mapping Target

Association

Owned Mappings

(none)

C.2.3.2.4 GenericToBehavior_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *Behavior*.

General Mappings

GenericToClassifier_Mapping

Mapping Source

Mapping Target

Behavior

Owned Mappings

(none)

C.2.3.2.5 GenericToClassifier_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *Classifier*.

General Mappings

GenericToType_Mapping

Mapping Source

Mapping Target

Classifier

Owned Mappings

(none)

C.2.3.2.6 GenericToComment_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *Comment*.

General Mappings

GenericToAnnotatingElement_Mapping

Mapping Source

Mapping Target

Comment

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Comment::locale () : String [1]

  ```
  null
  ```

- Comment::body () : String [1]
  abstract rule

C.2.3.2.7 GenericToConjugation_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *Conjugation*.

General Mappings

GenericToRelationship_Mapping

Mapping Source

Mapping Target

Conjugation

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Conjugation::conjugatedType () : Type [1]
  abstract rule
- Conjugation::originalType () : Type [1]
  abstract rule

C.2.3.2.8 GenericToConnector_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *Connector*.

General Mappings

GenericToFeature_Mapping
GenericToRelationship_Mapping

Mapping Source

Mapping Target

Connector

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Connector::isDirected () : Boolean [1]

  ```
  false
  ```

C.2.3.2.9 GenericToDocumentation_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *Documentation*.

General Mappings

GenericToComment_Mapping

Mapping Source

Mapping Target

Documentation

Owned Mappings

(none)

C.2.3.2.10 GenericToElement_Mapping

Description

This is the general abstract class to be used as an ancestor for any class mapping specification.

General Mappings

Mapping

Mapping Source

Mapping Target

Element

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Element::aliasId () : String [0..*]

  ```
  Set{}
  ```

- Element::name () : String [0..1]

  ```
  null
  ```

- Element::shortName () : String [0..1]

  ```
  null
  ```

- Element::elementId () : String [1]

  ```
  Helper.createUUID()
  ```

- Element::ownedRelationship () : Relationship [0..*]

  ```
  Set{}
  ```

C.2.3.2.11 GenericToEndFeatureMembership_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *EndFeatureMembership*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Mapping Target

EndFeatureMembership

Owned Mappings

(none)

C.2.3.2.12 GenericToExpression_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *Expression*.

General Mappings

GenericToStep_Mapping

Mapping Source

Mapping Target

Expression

Owned Mappings

(none)

C.2.3.2.13 GenericToFeature_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *Feature*.

General Mappings

GenericToType_Mapping

Mapping Source

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isPortion () : Boolean [1]

```
false
```

- Feature::isEnd () : Boolean [1]

```
false
```

- Feature::isReadOnly () : Boolean [1]

```
false
```

- Feature::direction () : FeatureDirectionKind [0..1]

```
null
```

- Feature::isDerived () : Boolean [1]

```
false
```

- Feature::isUnique () : Boolean [1]

```
true
```

- Feature::isComposite () : Boolean [1]

```
false
```

- Feature::isOrdered () : Boolean [1]

```
false
```

C.2.3.2.14 GenericToFeatureChaining_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *FeatureChaining*.

General Mappings

GenericToRelationship_Mapping

Mapping Source

Mapping Target

FeatureChaining

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureChaining::chainingFeature () : Feature [1]
  abstract rule

C.2.3.2.15 GenericToFeatureMembership_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *FeatureMembership*.

General Mappings

GenericToOwningMembership_Mapping
GenericToTypeFeaturing_Mapping

Mapping Source

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]
  abstract rule
- FeatureMembership::ownedRelatedElement () : Element [0..*]

```
Set{self.ownedMemberFeature()}
```

C.2.3.2.16 GenericToFeatureReferenceExpression_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *FeatureReferenceExpression*.

General Mappings

GenericToExpression_Mapping

Mapping Source

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

C.2.3.2.17 GenericToFeatureTyping_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *FeatureTyping*.

General Mappings

GenericToSpecialization_Mapping

Mapping Source

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]
  abstract rule
- FeatureTyping::typedFeature () : Feature [1]
  abstract rule

C.2.3.2.18 GenericToFeatureValue_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *FeatureValue*.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::isDefault () : Boolean [1]

  ```
  false
  ```

- FeatureValue::isInitial () : Boolean [1]

  ```
  false
  ```

- FeatureValue::featureWithValue () : Feature [1]
  abstract rule
- FeatureValue::value () : Expression [1]
  abstract rule
- FeatureValue::ownedRelatedElement () : Element [0..*]

  ```
  Set{self.value()}
  ```

C.2.3.2.19 GenericToFunction_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *Function*.

General Mappings

GenericToBehavior_Mapping

Mapping Source

Mapping Target

Function

Owned Mappings

(none)

C.2.3.2.20 GenericToImport_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *Import*.

General Mappings

GenericToRelationship_Mapping

Mapping Source

Mapping Target

Import

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Import::importedNamespace () : Namespace [1]
  abstract rule
- Import::importedMemberName () : String [0..1]

  ```
  null
  ```

- Import::isRecursive () : Boolean [1]

  ```
  false
  ```

- Import::visibility () : VisibilityKind [1]

  ```
  KerML::VisibilityKind::public
  ```

- Import::isImportAll () : Boolean [1]

  ```
  false
  ```

C.2.3.2.21 GenericToInvocationExpression_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *InvocationExpression*.

General Mappings

GenericToExpression_Mapping

Mapping Source

Mapping Target

InvocationExpression

Owned Mappings

(none)

C.2.3.2.22 GenericToInteraction_Mapping

C.2.3.2.23 GenericToItemFlow_Mapping

C.2.3.2.24 GenericToMembership_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *Membership*.

General Mappings

GenericToRelationship_Mapping

Mapping Source

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Membership::membershipOwningNamespace () : Element [0..*]
  abstract rule
- Membership::memberShortName () : String [0..1]

  ```
  null
  ```

- Membership::memberName () : String [0..1]

  ```
  null
  ```

- Membership::memberElement () : Element [1]
  abstract rule
- Membership::visibility () : VisibilityKind [1]

  ```
  KerML::VisibilityKind::public
  ```

C.2.3.2.25 GenericToNamespace_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *Namespace*.

General Mappings

GenericToElement_Mapping

Mapping Source

Mapping Target

Namespace

Owned Mappings

(none)

C.2.3.2.26 GenericToOwningMembership_Mapping

C.2.3.2.27 GenericToPackage_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *Package*.

General Mappings

GenericToNamespace_Mapping

Mapping Source

Mapping Target

Package

Owned Mappings

(none)

C.2.3.2.28 GenericToParameterMembership_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *ParameterMembership*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ParameterMembership::ownedMemberParameter () : Feature [1]

  ```
  null
  ```

- ParameterMembership::ownedRelatedElement () : Element [0..*]

  ```
  Set{self.ownedMemberParameter()}
  ```

C.2.3.2.29 GenericToPredicate_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *Predicate*.

General Mappings

GenericToFunction_Mapping

Mapping Source

Mapping Target

Predicate

Owned Mappings

(none)

C.2.3.2.30 GenericToRedefinition_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *Redefinition*.

General Mappings

GenericToSubsetting_Mapping

Mapping Source

Mapping Target

Redefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Redefinition::redefinedFeature () : Feature [1]
  abstract rule
- Redefinition::redefiningFeature () : Feature [1]
  abstract rule

C.2.3.2.31 GenericToRelationship_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *Relationship*.

General Mappings

GenericToElement_Mapping

Mapping Source

Mapping Target

Relationship

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Relationship::ownedRelatedElement () : Element [0..*]

  ```
  Set{}
  ```

- Relationship::source () : Element [0..*]

  ```
  Set{}
  ```

- Relationship::target () : Element [0..*]

  ```
  Set{}
  ```

C.2.3.2.32 GenericToReturnParameterMembership_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *ReturnParameterMembership*.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

Mapping Target

ReturnParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReturnParameterMembership::isComposite (in src : Element) : Boolean [1]

  returns "true" if the element provided as the actual parameter value can have a mapping to an instance of the type specified by the "to" attribute (i.e. can be used as a value for the "from" attribute)

  ```
  false
  ```

C.2.3.2.33 GenericToSpecialization_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *Specialization*.

General Mappings

GenericToRelationship_Mapping

Mapping Source

Mapping Target

Specialization

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Specialization::specific () : Type [1]
  abstract rule

- Specialization::general () : Type [1]
  abstract rule

C.2.3.2.34 GenericToStep_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *Step*.

General Mappings

GenericToFeature_Mapping

Mapping Source

Mapping Target

Step

Owned Mappings

(none)

C.2.3.2.35 GenericToSubclassification_Mapping

C.2.3.2.36 GenericToSubsetting_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *Subsetting*.

General Mappings

GenericToSpecialization_Mapping

Mapping Source

Mapping Target

Subsetting

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Subsetting::ownedRelatedElement () : Element [0..*]

```
Set{}
```

- Subsetting::subsettingFeature () : Feature [1]

```
from
```

- Subsetting::subsettedFeature () : Feature [1]
  abstract rule

C.2.3.2.37 GenericToSuccession_Mapping

C.2.3.2.38 GenericToSuccessionItemFlow_Mapping

C.2.3.2.39 GenericToTextualRepresentation_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *TextualRepresentation*.

General Mappings

GenericToAnnotatingElement_Mapping

Mapping Source

Mapping Target

TextualRepresentation

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- TextualRepresentation::language () : String [1]
  abstract rule
- TextualRepresentation::body () : String [1]
  abstract rule

C.2.3.2.40 GenericToType_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *Type*.

General Mappings

GenericToNamespace_Mapping

Mapping Source

Mapping Target

Type

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Type::isSufficient () : Boolean [1]

  ```
  false
  ```

- Type::isAbstract () : Boolean [1]

  ```
  false
  ```

C.2.3.2.41 GenericToTypeFeaturing_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *TypeFeaturing*.

General Mappings

GenericToRelationship_Mapping

Mapping Source

Mapping Target

TypeFeaturing

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- TypeFeaturing::featureOfType () : Feature [1]
  abstract rule
- TypeFeaturing::featuringType () : Type [1]
  abstract rule

C.2.3.3 Generic Mappings FromTo KerML

C.2.3.3.1 CommonMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *TypedElement* mapping.

General Mappings

GenericToMembership_Mapping

Mapping Source

TypedElement

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Membership::memberElement () : Element [1]

  ```
  from
  ```

C.2.3.3.2 CommonParameterReferenceUsageInMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Element* mapping.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

Element

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ParameterMembership::ownedMemberParameter () : Feature [1]

```
if not from.oclIsKindOf(UML::TypedElement) then CommonParameterReferenceUsageIn_Mapping.getMa
else if from.oclAsType(UML::TypedElement).type.oclIsUndefined() then CommonParameterReference
else CommonParameterReferenceUsageInUntyped_Mapping.getMapped(from) endif endif
```

C.2.3.3.3 CommonParameterReferenceUsageIn_Mapping

Description

*** not specified yet ***

General Mappings

CommonParameterReferenceUsageInUntyped_Mapping

Mapping Source

Element

Mapping Target

ReferenceUsage

Owned Mappings

- commonParameterReferenceUsageInFeatureTyping : CommonParameterReferenceUsageInFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
 if from.oclIsKindOf(UML::TypedElement) then Set{commonParameterReferenceUsageInFeatureTyping
```

C.2.3.3.4 CommonParameterReferenceUsageInUntyped_Mapping

Description

*** not specified yet ***

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Element

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::direction () : FeatureDirectionKind [0..1]

```
KerML::FeatureDirectionKind::_'in'
```

C.2.3.3.5 CommonReferenceUsageInFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *TypedElement* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

TypedElement

Mapping Target

FeatureTyping

Owned Mappings

- commonReferenceUsageIn : CommonReferenceUsageIn_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

```
commonReferenceUsageIn.to
```

- FeatureTyping::type () : Type [1]

```
if from.type.oclIsKindOf(UML::PrimitiveType) then
    Helper.getScalarValueType(from.type)
else
    from.type
endif
```

C.2.3.3.6 CommonReferenceUsageInUntyped_Mapping

Description

*** not specified yet ***

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

TypedElement

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::name () : String [0..1]

```
from.name
```

- ReferenceUsage::direction () : FeatureDirectionKind [0..1]

```
KerML::FeatureDirectionKind::_'in'
```

C.2.3.3.7 CommonReturnParameterFeature_Mapping

Description

*** not specified yet ***

General Mappings

CommonReturnParameterFeatureUntyped_Mapping

Mapping Source

Element

Mapping Target

Feature

Owned Mappings

- commonReturnParameterFeatureTyping : CommonReturnParameterFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::ownedRelationship () : Relationship [0..*]

```
if from.oclIsKindOf(UML::Property) then Set{commonReturnParameterFeatureTyping.to} else Set{
```

C.2.3.3.8 CommonReturnParameterFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *Element* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Element

Mapping Target

FeatureTyping

Owned Mappings

- commonReturnParameterFeature : CommonReturnParameterFeature_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

```
 commonReturnParameterFeature.to
```

- FeatureTyping::type () : Type [1]

```
 if from.oclIsKindOf(UML::Property)
then
if from.oclAsType(UML::TypedElement).type.oclIsKindOf(UML::PrimitiveType) then
    Helper.getScalarValueType(from.oclAsType(UML::TypedElement).type)
else
    from.oclAsType(UML::TypedElement).type
endif
else OclUndefined endif
```

C.2.3.3.9 CommonReturnParameterFeatureUntyped_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeature_Mapping

Mapping Source

Element

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::direction () : FeatureDirectionKind [0..1]

```
 KerML::FeatureDirectionKind::_'out'
```

C.2.3.3.10 CommonReturnParameterFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *Element* mapping.

General Mappings

GenericToReturnParameterMembership_Mapping

Mapping Source

Element

Mapping Target

ReturnParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReturnParameterMembership::ownedMemberParameter () : Feature [1]

```
if not from.oclIsKindOf(UML::TypedElement) then CommonReturnParameterFeatureUntyped_Mapping.g
else if from.oclAsType(UML::TypedElement).type.oclIsUndefined() then CommonReturnParameterFea
else CommonReturnParameterFeatureUntyped_Mapping.getMapped(from) endif endif
```

C.2.3.3.11 CommonReturnParameterReferenceUsageMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Element* mapping.

General Mappings

GenericToReturnParameterMembership_Mapping

Mapping Source

Element

Mapping Target

ReturnParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReturnParameterMembership::ownedMemberParameter () : Feature [0..1]

```
if not from.oclIsKindOf(UML::TypedElement) then CommonReturnParameterReferenceUsageUntyped_Ma
else if from.oclAsType(UML::TypedElement).type.oclIsUndefined() then CommonReturnParameterRef
else CommonReturnParameterReferenceUsageUntyped_Mapping.getMapped(from) endif endif
```

C.2.3.3.12 CommonReturnParameterReferenceUsage_Mapping

Description

Creates a reference usage for the *Element* mapping.

General Mappings

CommonReturnParameterReferenceUsageUntyped_Mapping

Mapping Source

Element

Mapping Target

ReferenceUsage

Owned Mappings

- commonReturnParameterReferenceUsageFeatureTyping : CommonReturnParameterReferenceUsageFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
 if from.oclIsKindOf(UML::TypedElement) then Set{commonReturnParameterReferenceUsageFeatureTy
```

C.2.3.3.13 CommonParameterReferenceUsageInFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *Element* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Element

Mapping Target

FeatureTyping

Owned Mappings

- commonParameterReferenceUsageIn : CommonParameterReferenceUsageIn_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

```
 commonParameterReferenceUsageIn.to
```

- FeatureTyping::type () : Type [1]

```
 if from.oclIsKindOf(UML::TypedElement)
then
if from.oclAsType(UML::TypedElement).type.oclIsKindOf(UML::PrimitiveType) then
    Helper.getScalarValueType(from.oclAsType(UML::TypedElement).type)
else
    from.oclAsType(UML::TypedElement).type
endif
else OclUndefined endif
```

C.2.3.3.14 CommonReturnParameterReferenceUsageFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *Element* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Element

Mapping Target

FeatureTyping

Owned Mappings

- commonReturnParameterReferenceUsage : CommonReturnParameterReferenceUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

```
 commonReturnParameterReferenceUsage.to
```

- FeatureTyping::type () : Type [1]

```
 if from.oclIsKindOf(UML::TypedElement)
then
if from.oclAsType(UML::TypedElement).type.oclIsKindOf(UML::PrimitiveType) then
    Helper.getScalarValueType(from.oclAsType(UML::TypedElement).type)
else
    from.oclAsType(UML::TypedElement).type
endif
else OclUndefined endif
```

C.2.3.3.15 CommonReturnParameterReferenceUsageUntyped_Mapping

Description

*** not specified yet ***

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Element

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::direction () : FeatureDirectionKind [0..1]

```
KerML::FeatureDirectionKind::_'out'
```

C.2.3.3.16 EmptyReturnParameterFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *Element* mapping.

General Mappings

GenericToReturnParameterMembership_Mapping

Mapping Source

Element

Mapping Target

ReturnParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReturnParameterMembership::ownedMemberParameter () : Feature [1]

```
CommonReturnParameterFeatureUntyped_Mapping.getMapped(from)
```

C.2.3.4 Generic Mappings to Systems

C.2.3.4.1 GenericToActionUsage_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *ActionUsage*.

General Mappings

GenericToUsage_Mapping
GenericToStep_Mapping

Mapping Source

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ActionUsage::isComposite () : Boolean [1]

```
true
```

C.2.3.4.2 GenericToActorMembership_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *ActorMembership*.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

Mapping Target

ActorMembership

Owned Mappings

(none)

C.2.3.4.3 GenericToAssignmentActionUsage_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *AssignmentActionUsage*.

General Mappings

GenericToActionUsage_Mapping

Mapping Source

Mapping Target

AssignmentActionUsage

Owned Mappings

(none)

C.2.3.4.4 GenericToConnectionUsage_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *ConnectionUsage*.

General Mappings

GenericToPartUsage_Mapping

Mapping Source

Mapping Target

ConnectionUsage

Owned Mappings

(none)

C.2.3.4.5 GenericToConjugatedPortDefinition_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *ConjugatedPortDefinition*.

General Mappings

GenericToPortDefinition_Mapping

Mapping Source

Mapping Target

ConjugatedPortDefinition

Owned Mappings

(none)

C.2.3.4.6 GenericToConjugatedPortTyping_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *ConjugatedPortTyping*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Mapping Target

ConjugatedPortTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ConjugatedPortTyping::conjugatedPortDefinition () : ConjugatedPortDefinition [1]
  abstract rule
- ConjugatedPortTyping::portDefinition () : PortDefinition [1]
  abstract rule

C.2.3.4.7 GenericToConstraintDefinition_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *ConstraintDefinition*.

General Mappings

GenericToDefinition_Mapping

Mapping Source

Mapping Target

ConstraintDefinition

Owned Mappings

(none)

C.2.3.4.8 GenericToDefinition_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *Definition*.

General Mappings

GenericToClassifier_Mapping

Mapping Source

Mapping Target

Definition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Definition::isVariation () : Boolean [1]

  ```
  false
  ```

C.2.3.4.9 GenericToEventOccurerenceUsage_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *EventOccurrenceUsage*.

General Mappings

GenericToOccurrenceUsage_Mapping

Mapping Source

Mapping Target

EventOccurrenceUsage

Owned Mappings

(none)

C.2.3.4.10 GenericToItemDefinition_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *ItemDefinition*.

General Mappings

GenericToDefinition_Mapping

Mapping Source

Mapping Target

ItemDefinition

Owned Mappings

(none)

C.2.3.4.11 GenericToMetadataUsage_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *MetadataUsage*.

General Mappings

GenericToUsage_Mapping

Mapping Source

Mapping Target

MetadataUsage

Owned Mappings

(none)

C.2.3.4.12 GenericToObjectiveMembership_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *ObjectiveMembership*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Mapping Target

ObjectiveMembership

Owned Mappings

(none)

C.2.3.4.13 GenericToOccurenceDefinition_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *OccurrenceDefinition*.

General Mappings

GenericToDefinition_Mapping

Mapping Source

Mapping Target

OccurrenceDefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OccurrenceDefinition::isIndividual () : Boolean [1]

```
false
```

C.2.3.4.14 GenericToOccurrenceUsage_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *OccurrenceUsage*.

General Mappings

GenericToUsage_Mapping

Mapping Source

Mapping Target

OccurrenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OccurrenceUsage::portionKind () : PortionKind [1]

```
OclUndefined
```

- OccurrenceUsage::isIndividual () : Boolean [1]

```
false
```

C.2.3.4.15 GenericToPartUsage_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *PartUsage*.

General Mappings

GenericToUsage_Mapping

Mapping Source

Mapping Target

PartUsage

Owned Mappings

(none)

C.2.3.4.16 GenericToPortConjugation_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *PortConjugation*.

General Mappings

GenericToConjugation_Mapping

Mapping Source

Mapping Target

PortConjugation

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- PortConjugation::originalPortDefinition () : PortDefinition [1]
  abstract rule

C.2.3.4.17 GenericToPortDefinition_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *PortDefinition*.

General Mappings

GenericToDefinition_Mapping

Mapping Source

Mapping Target

PortDefinition

Owned Mappings

(none)

C.2.3.4.18 GenericToReferenceUsage_Mapping

Description

Provides the basic features to map to a ReferenceUsage element.

General Mappings

GenericToUsage_Mapping

Mapping Source

Mapping Target

ReferenceUsage

Owned Mappings

(none)

C.2.3.4.19 GenericToRequirementUsage_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *RequirementUsage*.

General Mappings

GenericToUsage_Mapping

Mapping Source

Mapping Target

RequirementUsage

Owned Mappings

(none)

C.2.3.4.20 GenericToStateUsage_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *StateUsage*.

General Mappings

GenericToActionUsage_Mapping

Mapping Source

Mapping Target

StateUsage

Owned Mappings

(none)

C.2.3.4.21 GenericToSubjectMembership_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *SubjectMembership*.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

Mapping Target

SubjectMembership

Owned Mappings

(none)

C.2.3.4.22 GenericToUsage_Mapping

Description

Generic mapping class for mappingsto the SysML v2 element *Usage*.

General Mappings

GenericToFeature_Mapping

Mapping Source

Mapping Target

Usage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Usage::isVariation () : Boolean [1]

```
false
```

C.2.4 SysML v1.7

C.2.4.1 Overview

C.2.4.2 Activities

C.2.4.2.1 Overview

Table 7. List of all Overview Mapping Specfications

SysML v1 Concept	SysML v2 Concept	Mapping Class
Continuous		*** not specified yet ***
ControlOperator		*** not specified yet ***
Discrete		*** not specified yet ***
NoBuffer		*** not specified yet ***
Optional		*** not specified yet ***
Overwrite		*** not specified yet ***
Probability		*** not specified yet ***
Rate		*** not specified yet ***

C.2.4.2.2 Mapping Specifications

C.2.4.3 Allocations

C.2.4.3.1 Overview

Table 8. List of all Overview Mapping Specfications

SysML v1 Concept	SysML v2 Concept	Mapping Class
Allocate	AllocationUsage	_AllocationUsage_Mapping
AllocateActivityPartition		*** not specified yet ***

C.2.4.3.2 Mapping Specifications

C.2.4.3.2.1 AllocationDefinition_Mapping

Description

*** not specified yet ***

General Mappings

Abstraction_Mapping

Mapping Source

Dependency

Mapping Target

AllocationDefinition

Owned Mappings

- allocationDefinitionFromFeatureMembership : AllocationDefinitionFromFeatureMembership_Mapping
- allocationDefinitionToFeatureMembership : AllocationDefinitionToFeatureMembership_Mapping

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
Helper.hasStereotypeApplied(from, 'SysML::Allocations::Allocate') and from.client->select(t | t.ocl
```

Mapping rules

The following lists the mapping rules for the target element properties.

- AllocationDefinition::ownedRelationship () : Relationship [0..*]

  ```
  Set{allocationDefinitionFromFeatureMembership.to, allocationDefinitionToFeatureMembership.to
  ```

C.2.4.3.2.2 AllocationDefinitionToFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *Dependency* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Dependency

Mapping Target

FeatureMembership

Owned Mappings

- allocationDefinitionToReferenceUsage : AllocationDefinitionToReferenceUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::memberName () : String [0..1]

```
'allocatedTo'
```

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
allocationDefinitionToReferenceUsage.to
```

C.2.4.3.2.3 AllocationDefinitionFromFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *Dependency* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Dependency

Mapping Target

FeatureMembership

Owned Mappings

- allocationDefinitionFromReferenceUsage : AllocationDefinitionFromReferenceUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
allocationDefinitionFromReferenceUsage.to
```

- FeatureMembership::memberName () : String [0..1]

```
'allocatedFrom'
```

C.2.4.3.2.4 AllocationDefinitionFromFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *Dependency* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Dependency

Mapping Target

FeatureTyping

Owned Mappings

- allocationDefinitionFromReferenceUsage : AllocationDefinitionFromReferenceUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

```
allocationDefinitionFromReferenceUsage.to
```

- FeatureTyping::type () : Type [1]

```
from.source.get(0)
```

C.2.4.3.2.5 AllocationDefinitionFromReferenceUsage_Mapping

Description

Creates a reference usage for the *Dependency* mapping.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Dependency

Mapping Target

ReferenceUsage

Owned Mappings

- allocationDefinitionFromFeatureTyping : AllocationDefinitionFromFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::isEnd () : Boolean [1]

```
true
```

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
Set{allocationDefinitionFromFeatureTyping.to}
```

C.2.4.3.2.6 AllocationDefinitionToFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *Dependency* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Dependency

Mapping Target

FeatureTyping

Owned Mappings

- allocationDefinitionToReferenceUsage : AllocationDefinitionToReferenceUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

```
allocationDefinitionToReferenceUsage.to
```

- FeatureTyping::type () : Type [1]

```
from.target.get(0)
```

C.2.4.3.2.7 AllocationDefinitionToReferenceUsage_Mapping

Description

Creates a reference usage for the *Dependency* mapping.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Dependency

Mapping Target

ReferenceUsage

Owned Mappings

- allocationDefinitionToFeatureTyping : AllocationDefinitionToFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::isEnd () : Boolean [1]

```
true
```

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
Set{allocationDefinitionToFeatureTyping.to}
```

C.2.4.4 Blocks

C.2.4.4.1 Overview

Table 9. List of all Overview Mapping Specfications

SysML v1 Concept	SysML v2 Concept	Mapping Class
AdjunctProperty		*** not specified yet ***
BindingConnector	BindingConnectorAsUsage	BindingConnector_Mapping
Block	PartDefinition PartDefinition	EncapsulatedBlock_Mapping Block_Mapping
BoundReference		*** not specified yet ***
ClassifierBehaviorProperty		*** not specified yet ***
ConnectorProperty		*** not specified yet ***
DirectedRelationshipPropertyPath		*** not specified yet ***
DistributedProperty		*** not specified yet ***
ElementPropertyPath		*** not specified yet ***
EndPathMultiplicity		*** not specified yet ***
NestedConnectorEnd		*** not specified yet ***

SysML v1 Concept	SysML v2 Concept	Mapping Class
ParticipantProperty		*** not specified yet ***
PropertySpecificType		*** not specified yet ***
ValueType		*** not specified yet ***

C.2.4.4.2 SysML v1 Blocks elements not mapped

Table 10. List of SysML v1 elements not mapped of this section

SysML v1 Concept	Rationale
AdjunctProperty	The concept of adjunct properties is not needed in SysML v2, where the principal of the adjunct property can be used directly in the appropriate place.
ConnectorProperty	The connector property is a special case of an adjunct property and is not mapped, just like the adjunct property.

C.2.4.4.3 Mapping Specifications

C.2.4.4.3.1 AssociationBlock_Mapping

Description

*** not specified yet ***

General Mappings

AssociationClass_Mapping

Mapping Source

AssociationClass

Mapping Target

ConnectionDefinition

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
Helper.hasStereotypeApplied(from, 'SysML::Blocks::Block')
```

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

C.2.4.4.3.2 BindingConnector_Mapping

Description

*** not specified yet ***

General Mappings

Connector_Mapping

Mapping Source

Connector

Mapping Target

BindingConnectorAsUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
Helper.hasStereotypeApplied(from, 'SysML::Blocks::BindingConnector')
```

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

C.2.4.4.3.3 Block_Mapping

Description

A SysML::Block is mapped to a SysMLv2::PartDefinition.

General Mappings

Class_Mapping

Mapping Source

Class

Mapping Target

PartDefinition

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
 not from.oclIsTypeOf(UML::AssociationClass) and Helper.hasStereotypeApplied(src, 'SysML::Blocks::Bl
    and not Helper.hasStereotypeApplied(src, 'SysML::ConstraintBlocks::ConstraintBlock')
    and not Helper.hasStereotypeApplied(src, 'SysML::Ports&Flows::InterfaceBlock')
```

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

C.2.4.4.3.4 Part_Mapping

Description

A property with composite aggregation which is typed by a block is mapped to a SysMLv2::PartUsage.

General Mappings

Property_Mapping

Mapping Source

Property

Mapping Target

PartUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
if src.oclIsKindOf(UML::Property) then
    let p: UML::Property = src.oclAsType(UML::Property) in
    not p.type.oclIsUndefined() and
    Helper.hasStereotypeApplied(p.type, 'SysML::Blocks::Block') and
    (p.association.oclIsUndefined() or p.association.ownedEnd->excludes(p)) and
    p.aggregation = UML::AggregationKind::composite
else
    false
endif
```

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

C.2.4.4.3.5 EncapsulatedBlock_Mapping

Description

A SysML::Block with *isEncapsulated=true* is mapped to a PartDefinition, and, additionally, gets a metadata feature defined by the SysML v1 library which represents the SysML v1 isEncapsulated property.

General Mappings

Block_Mapping

Mapping Source

Class

Mapping Target

PartDefinition

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
 not from.oclIsTypeOf(UML::AssociationClass) and Helper.hasStereotypeApplied(src, 'SysML::Blocks::Bl
    and not Helper.hasStereotypeApplied(src, 'SysML::ConstraintBlocks::ConstraintBlock')
    and not Helper.hasStereotypeApplied(src, 'SysML::Ports&Flows::InterfaceBlock')
    and Helper.getTagValue(src, 'SysML::Blocks::Block', 'isEncapsulated')
```

Mapping rules

The following lists the mapping rules for the target element properties.

- PartDefinition::ownedRelationship () : Relationship [0..*]

```
let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | e.oclIsKindOf(UML::Proper
let redefinedAttributes: Set(UML::Element) = from.ownedElement->select(e | from.oclIsKindOf(U
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.oclIsKindOf(
let toElementOMS: Set(UML::Element) = (((from.ownedElement - toElementFMS) - redefinedAttribu
let relationships: Sequence(UML::Element) =
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(redefinedAttributes->collect(e | AttributeRedefinedMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))
->including(EncapsulatedBlockMetadataMembership_Mapping.getMapped(from)) in
if from.classifierBehavior.oclIsUndefined() then relationships else relationships->append(Cla
```

C.2.4.4.3.6 EncapsulatedBlockMetadataMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Class* mapping.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Class

Mapping Target

OwningMembership

Owned Mappings

- encapsulatedBlockMetadata : EncapsulatedBlockMetadata_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

```
encapsulatedBlockMetadata.to
```

C.2.4.4.3.7 EncapsulatedBlockMetadata_Mapping

Description

*** not specified yet ***

General Mappings

GenericToMetadataUsage_Mapping

Mapping Source

Class

Mapping Target

MetadataUsage

Owned Mappings

- encapsulatedBlockMetadataFeatureMembership : EncapsulatedBlockMetadataFeatureMembership_Mapping
- encapsulatedBlockMetadataFeatureTyping : EncapsulatedBlockMetadataFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- MetadataUsage::ownedRelationship () : Relationship [0..*]

```
Set{encapsulatedBlockMetadataFeatureMembership.to, encapsulatedBlockMetadataFeatureTyping.to
```

C.2.4.4.3.8 EncapsulatedBlockMetadataFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *Class* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Class

Mapping Target

FeatureMembership

Owned Mappings

- encapsulatedBlockMetadataReferenceUsage : EncapsulatedBlockMetadataReferenceUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [0..1]

```
encapsulatedBlockMetadataReferenceUsage.to
```

C.2.4.4.3.9 EncapsulatedBlockMetadataFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *Class* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Class

Mapping Target

FeatureTyping

Owned Mappings

- encapsulatedBlockMetadata : EncapsulatedBlockMetadata_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
SYSML2::MetadataDefinition.allInstances()->any(m | m.qualifiedName = 'SysMLv1Library::BlockD
```

- FeatureTyping::typedFeature () : Feature [1]

```
encapsulatedBlockMetadata.to
```

C.2.4.4.3.10 EncapsulatedBlockMetadataReferenceUsage_Mapping

Description

Creates a reference usage for the *Class* mapping.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Class

Mapping Target

ReferenceUsage

Owned Mappings

- encapsulatedBlockMetadataFeatureValue : EncapsulatedBlockMetadataFeatureValue_Mapping
- encapsulatedBlockMetadataRedefinition : EncapsulatedBlockMetadataRedefinition_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
Set{encapsulatedBlockMetadataRedefinition.to, encapsulatedBlockMetadataFeatureValue.to}
```

C.2.4.4.3.11 EncapsulatedBlockMetadataFeatureValue_Mapping

Description

Creates a feature value relationship for the mapping class *Class*

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Class

Mapping Target

FeatureValue

Owned Mappings

- *literalBooleanTrue : LiteralBooleanTrue_Mapping*

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

```
literalBooleanTrue.to
```

C.2.4.4.3.12 EncapsulatedBlockMetadataRedefinition_Mapping

Description

Creates a redefinition relationship for the *redefiningFeature()* and the *redefinedFeature()* for the *Class* mapping.

General Mappings

GenericToRedefinition_Mapping

Mapping Source

Class

Mapping Target

Redefinition

Owned Mappings

- encapsulatedBlockMetadataReferenceUsage : EncapsulatedBlockMetadataReferenceUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Redefinition::redefiningFeature () : Feature [1]

```
encapsulatedBlockMetadataReferenceUsage.to
```

- Redefinition::redefinedFeature () : Feature [1]

```
SYSML2::AttributeUsage.allInstances()->any(m | m.qualifiedName = 'SysMLv1Library::BlockData:
```

C.2.4.5 Libraries

C.2.4.5.1 Requirements

C.2.4.5.1.1 VerdictKind

Description

The VerdictKind is an enumeration that contains the values fail, inconclusive, pass, and error indicating how this test case execution has performed.

A pass indicates that the test case is successful and that the system under test has behaved according to what should be expected. A fail on the other hand shows that the system under test is not behaving according to the specification. An inconclusive means that the test execution cannot determine whether the system under test performs well or not. An error tells that the test system itself and not the system under test fails.

The VerdictKind is derived from the Verdict element from the UTP specification v1.2.

Literals

- error
- fail
- inconclusive
- pass

C.2.4.5.2 UnitAndQuantityKind

C.2.4.6 Model Elements

C.2.4.6.1 Overview

Table 11. List of all Overview Mapping Specfications

SysML v1 Concept	SysML v2 Concept	Mapping Class
Conform		*** not specified yet ***
ElementGroup	Package	ElementGroup_Mapping
Expose		*** not specified yet ***
Problem	Comment	ProblemRationale_Mapping
Rationale	Comment	ProblemRationale_Mapping
Stakeholder	PartDefinition	Stakeholder_Mapping
View		*** not specified yet ***
Viewpoint		*** not specified yet ***

C.2.4.6.2 Mapping Specifications

C.2.4.6.2.1 ProblemRationaleMetadataUsage_Mapping

Description

*** not specified yet ***

General Mappings

GenericToMetadataUsage_Mapping

Mapping Source

Comment

Mapping Target

MetadataUsage

Owned Mappings

- problemRationaleMetadataFeatureTyping : ProblemRationaleMetadataFeatureTyping_Mapping
- unnamed1 : Boolean

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- MetadataUsage::ownedRelationship () : Relationship [0..*]

```
Set{problemRationaleMetadataFeatureTyping.to, ProblemRationaleMetadataFeatureMembership_Mapp
```

C.2.4.6.2.2 CommentToConcern_Mapping

Description

*** not specified yet ***

General Mappings

Comment_Mapping

Mapping Source

Comment

Mapping Target

ConcernDefinition

Owned Mappings

- commentToConcernReturnParameterMembership : CommentToConcernReturnParameterMembership_Mapping

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
(not Helper.hasStereotypeApplied(from, 'SysML::ModelElements::ElementGroup')) and UML::Classifier.al
```

Mapping rules

The following lists the mapping rules for the target element properties.

- ConcernDefinition::ownedRelationship () : Relationship [0..*]

```
    let toStakeholderMS : Set(UML::Classifier) = UML::Classifier.allInstances()->select(s | Helpe
    toStakeholderMS->collect(e | StakeholderMembership_Mapping.getMapped(e))->append(commentToCor
```

C.2.4.6.2.3 CommentToConcernComment_Mapping

Description

*** not specified yet ***

General Mappings

GenericToAnnotatingElement_Mapping

Mapping Source

Comment

Mapping Target

Comment

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Comment::body () : String [1]

```
    UML::Classifier.allInstances()->select(s | Helper.hasStereotypeApplied(s, 'SysML::ModelElemer
```

C.2.4.6.2.4 CommentToConcernDocumentation_Mapping

Description

*** not specified yet ***

General Mappings

GenericToAnnotation_Mapping

Mapping Source

Comment

Mapping Target

Annotation

Owned Mappings

- commentToConcernComment : CommentToConcernComment_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Annotation::ownedRelatedElement () : Element [0..*]

```
Set{commentToConcernComment.to}
```

C.2.4.6.2.5 CommenttToConcernReturnParameter_Mapping

Description

*** not specified yet ***

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Comment

Mapping Target

ReferenceUsage

Owned Mappings

(none)

C.2.4.6.2.6 CommentToConcernReturnParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Comment* mapping.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

Comment

Mapping Target

ReturnParameterMembership

Owned Mappings

- commentToConcernDocumentation : CommentToConcernDocumentation_Mapping
- commentToConcernReturnParameter : CommenttToConcernReturnParameter_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReturnParameterMembership::ownedRelatedElement () : Element [0..*]

```
 let member: KerML::Element = self.ownedMemberParameter() in
if member.oclIsUndefined() then
    Set{commentToConcernDocumentation.to}
else
    Set{self.ownedMemberParameter(), commentToConcernDocumentation.to}
endif
```

- ReturnParameterMembership::ownedMemberParameter () : Feature [0..1]

```
 commentToConcernReturnParameter.to
```

C.2.4.6.2.7 ProblemRationaleMetadataFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *Comment* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Comment

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [0..1]

```
ProblemRationaleMetadataReferenceUsage_Mapping.getMapped(from)
```

C.2.4.6.2.8 ProblemRationaleMetadataFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *Comment* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Comment

Mapping Target

FeatureTyping

Owned Mappings

- problemRationaleMetadataUsage : ProblemRationaleMetadataUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

```
problemRationaleMetadataUsage.to
```

- FeatureTyping::type () : Type [1]

```
if Helper.hasStereotypeApplied(from, 'SysML::ModelElements::Problem') then
  SYSML2::MetadataDefinition.allInstances()->any(m | m.qualifiedName = 'ModelingMetadata::Iss
else if Helper.hasStereotypeApplied(from, 'SysML::ModelElements::Rationale') then
  SYSML2::MetadataDefinition.allInstances()->any(m | m.qualifiedName = 'ModelingMetadata::Rat
else OclUndefined endif endif
```

C.2.4.6.2.9 ProblemRationaleMetadataReferenceUsage_Mapping

Description

Creates a reference usage for the *Comment* mapping.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Comment

Mapping Target

ReferenceUsage

Owned Mappings

- problemRationaleMetadataRedefinition : ProblemRationaleMetadataRedefinition_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
Set{problemRationaleMetadataRedefinition.to, ProblemRationaleMetadataFeatureValue_Mapping.ge
```

C.2.4.6.2.10 ProblemRationaleMetadataFeatureValue_Mapping

Description

Creates a feature value relationship for the mapping class *Comment*

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Comment

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

```
ProblemRationaleMetadataFeatureValueString_Mapping.getMapped(from)
```

C.2.4.6.2.11 ProblemRationaleMetadataMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Comment* mapping.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Comment

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

```
ProblemRationaleMetadataUsage_Mapping.getMapped(from)
```

C.2.4.6.2.12 ElementGroup_Mapping

Description

*** not specified yet ***

General Mappings

GenericToPackage_Mapping

Mapping Source

Comment

Mapping Target

Package

Owned Mappings

- elementGroupMetadaMembership : ElementGroupMetadaMembership_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Package::ownedRelationship () : Relationship [0..*]

```
ElementOwnership_Mapping.getMappedColl(from.ownedElement)->including(elementGroupMetadaMembe
```

C.2.4.6.2.13 ElementGroupCriterion_Mapping

Description

*** not specified yet ***

General Mappings

GenericToExpression_Mapping

Mapping Source

Comment

Mapping Target

LiteralString

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- LiteralString::value () : String [1]

```
Helper.getTagValueAsString(from,'SysML::ModelElements::ElementGroup', 'criterion')
```

C.2.4.6.2.14 ElementGroupMetadaMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Comment* mapping.

General Mappings

GenericToMembership_Mapping

Mapping Source

Comment

Mapping Target

Membership

Owned Mappings

- elementGroupMetadataUsage : ElementGroupMetadataUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Membership::ownedMemberElement () : Element [0..1]

  ```
  elementGroupMetadataUsage.to
  ```

- Membership::memberName () : String [0..1]

  ```
  'ElementGroupData'
  ```

- Membership::memberElement () : Element [1]

  ```
  self.ownedMemberElement()
  ```

C.2.4.6.2.15 ElementGroupMetadataFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *Comment* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Comment

Mapping Target

FeatureMembership

Owned Mappings

- elementGroupMetadataReferenceUsage : ElementGroupMetadataReferenceUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [0..1]

```
elementGroupMetadataReferenceUsage.to
```

- FeatureMembership::memberFeature () : Feature [1]

```
self.ownedMemberFeature()
```

C.2.4.6.2.16 ElementGroupMetadataFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *Comment* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Comment

Mapping Target

FeatureTyping

Owned Mappings

- elementGroupMetadataUsage : ElementGroupMetadataUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
 let m : SYSML2::Membership = SYSML2::AttributeDefinition.allInstances()->collect(dt | dt.own
if (m.oclIsUndefined()) then
        OclUndefined
else
        m.memberElement
endif
```

- FeatureTyping::typedFeature () : Feature [1]

```
elementGroupMetadataUsage.to
```

C.2.4.6.2.17 ElementGroupMetadataFeatureValue_Mapping

Description

Creates a feature value relationship for the mapping class *Comment*

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Comment

Mapping Target

FeatureValue

Owned Mappings

- *elementGroupCriterion : ElementGroupCriterion_Mapping*

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

```
elementGroupCriterion.to
```

C.2.4.6.2.18 ElementGroupMetadataRedefinition_Mapping

Description

Creates a redefinition relationship for the *redefiningFeature()* and the *redefinedFeature()* for the *Comment* mapping.

General Mappings

GenericToRedefinition_Mapping

Mapping Source

Comment

Mapping Target

Redefinition

Owned Mappings

- elementGroupMetadataReferenceUsage : ElementGroupMetadataReferenceUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Redefinition::redefinedFeature () : Feature [1]

```
 let m : SYSML2::Membership = SYSML2::AttributeUsage.allInstances()->collect(dt | dt.owningRe
if (m.oclIsUndefined()) then
        OclUndefined
else
    m.memberElement
endif
```

- Redefinition::redefiningFeature () : Feature [1]

```
 elementGroupMetadataReferenceUsage.to
```

C.2.4.6.2.19 ElementGroupMetadataReferenceUsage_Mapping

Description

Creates a reference usage for the *Comment* mapping.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Comment

Mapping Target

ReferenceUsage

Owned Mappings

- elementGroupMetadataFeatureValue : ElementGroupMetadataFeatureValue_Mapping
- elementGroupMetadataRedefinition : ElementGroupMetadataRedefinition_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
Set{elementGroupMetadataRedefinition.to, elementGroupMetadataFeatureValue.to}
```

C.2.4.6.2.20 ElementGroupMetadataUsage_Mapping

Description

*** not specified yet ***

General Mappings

GenericToMetadataUsage_Mapping

Mapping Source

Comment

Mapping Target

MetadataUsage

Owned Mappings

- elementGroupMetadataFeatureMembership : ElementGroupMetadataFeatureMembership_Mapping
- elementGroupMetadataFeatureTyping : ElementGroupMetadataFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- MetadataUsage::ownedRelationship () : Relationship [0..*]

```
Set{elementGroupMetadataFeatureTyping.to, elementGroupMetadataFeatureMembership.to}
```

C.2.4.6.2.21 ElementGroupMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Element* mapping.

General Mappings

ElementOwningMembership_Mapping

Mapping Source

Element

Mapping Target

OwningMembership

Owned Mappings

- : Comment
- elementGroup : ElementGroup_Mapping

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
Helper.hasStereotypeApplied(from, 'SysML::ModelElements::ElementGroup')
```

Mapping rules

The following lists the mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [0..*]

```
 self.memberElement()
```

- OwningMembership::memberElement () : Element [1]

```
 elementGroup.to
```

- OwningMembership::ownedRelatedElement () : Element [0..*]

```
 let member: KerML::Element = self.ownedMemberElement() in
if member.oclIsUndefined() then
    Set{}
else
    Set{self.ownedMemberElement()}
endif
```

- OwningMembership::memberName () : String [0..1]

```
 Helper.getTagValueAsString(from, 'SysML::ModelElements::ElementGroup', 'name')
```

C.2.4.6.2.22 ProblemRationale_Mapping

Description

The mapping class combines the mapping of SysMLv1::Problem and SysMLv1::Rationale. The SysMLv1::Problem is mapped to the library element ModelingMetadata::Issue and the SysMLv1::Rationale is mapped to ModelingMetadata::Rationale. The expected SysML v2 textual syntax of the mapping is as follows.

```
metadata ModelingMetadata::Issue {text = "This is a problem statement";}

metadata ModelingMetadata::Rationale {text = "This is a rationale statement";}
```

General Mappings

Comment_Mapping

Mapping Source

Comment

Mapping Target

Comment

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
(not Helper.hasStereotypeApplied(from, 'SysML::ModelElements::ElementGroup')) and (Helper.hasStereot
```

Mapping rules

The following lists the mapping rules for the target element properties.

- Comment::ownedRelationship () : Relationship [0..*]

  ```
  self.annotation()->append(ProblemRationaleMetadataMembership_Mapping.getMapped(from))
  ```

C.2.4.6.2.23 ProblemRationaleMetadataRedefinition_Mapping

Description

Creates a redefinition relationship for the *redefiningFeature()* and the *redefinedFeature()* for the *Comment* mapping.

General Mappings

GenericToRedefinition_Mapping

Mapping Source

Comment

Mapping Target

Redefinition

Owned Mappings

- problemRationaleMetadataReferenceUsage : ProblemRationaleMetadataReferenceUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Redefinition::redefinedFeature () : Feature [1]

```
if Helper.hasStereotypeApplied(from, 'SysML::ModelElements::Problem') then
  SYSML2::AttributeUsage.allInstances()->any(m | m.qualifiedName = 'ModelingMetadata::Issue::
else if Helper.hasStereotypeApplied(from, 'SysML::ModelElements::Rationale') then
  SYSML2::AttributeUsage.allInstances()->any(m | m.qualifiedName = 'ModelingMetadata::Rationa
else
  OclUndefined
endif
endif
```

- Redefinition::redefiningFeature () : Feature [1]

```
problemRationaleMetadataReferenceUsage.to
```

C.2.4.6.2.24 ProblemRationaleMetadataFeatureValueString_Mapping

Description

*** not specified yet ***

General Mappings

GenericToExpression_Mapping

Mapping Source

Comment

Mapping Target

LiteralString

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- LiteralString::value () : String [1]

```
from.body
```

C.2.4.6.2.25 Stakeholder_Mapping

Description

*** not specified yet ***

General Mappings

Class_Mapping

Mapping Source

Class

Mapping Target

PartDefinition

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
Helper.hasStereotypeApplied(from, 'SysML::ModelElements::Stakeholder')
```

Mapping rules

The following lists the mapping rules for the target element properties.

- PartDefinition::ownedRelationship () : Relationship [0..*]

  ```
  let toClassifierMS: Sequence(UML::Element) = src.ownedElement->select(e | e.oclIsKindOf(UML::
  let excludeOwnedConcerns: Sequence(UML::Element) = src.ownedElement->select(e | e.oclIsKindOf
  let toConcernMS: Sequence(UML::Element) = Helper.getTagValue(src, 'SysML::ModelElements::Stak
  let toFeatureMS: Sequence(UML::Element) = src.ownedElement->select(e | e.oclIsKindOf(UML::Pro
  let toElementOMS: Set(UML::Element) = (((src.ownedElement - toFeatureMS) - excludeOwnedConcer
  let relationships: Sequence(UML::Element) =
  toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
  ->union(toFeatureMS->collect(e | PropertyMembership_Mapping.getMapped(e)))
  ->union(toClassifierMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))) in
  if from.classifierBehavior.oclIsUndefined() then relationships else relationships->append(Cla
  ```

C.2.4.6.2.26 StakeholderMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Classifier* mapping.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

Classifier

Mapping Target

StakeholderMembership

Owned Mappings

- stakeholderPartUsage : StakeholderPartUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- StakeholderMembership::memberName () : String [0..1]

  ```
  from.name
  ```

- StakeholderMembership::ownedMemberParameter () : Feature [0..1]

  ```
  StakeholderPartUsage_Mapping.getMapped(from)
  ```

C.2.4.6.2.27 StakeholderPartUsage_Mapping

Description

*** not specified yet ***

General Mappings

GenericToPartUsage_Mapping

Mapping Source

Classifier

Mapping Target

PartUsage

Owned Mappings

(none)

C.2.4.6.2.28 Viewpoint_Mapping

Description

*** not specified yet ***

General Mappings

Class_Mapping

Mapping Source

Class

Mapping Target

ViewpointDefinition

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
Helper.hasStereotypeApplied(from, 'SysML::ModelElements::Viewpoint')
```

Mapping rules

The following lists the mapping rules for the target element properties.

- ViewpointDefinition::ownedRelationship () : Relationship [0..*]

```
let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | e.oclIsKindOf(UML::Proper
let redefinedAttributes: Set(UML::Element) = from.ownedElement->select(e | from.oclIsKindOf(U
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.oclIsKindOf(
let toElementOMS: Set(UML::Element) = (((from.ownedElement - toElementFMS) - redefinedAttribu
let relationships: Sequence(UML::Element) =
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(redefinedAttributes->collect(e | AttributeRedefinedMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))
->including(CommonReturnParameterReferenceUsageMembership_Mapping.getMapped(from))
->including(ViewpointSubjectMembership_Mapping.getMapped(from))
->including(ViewpointPurposeMetadataMembership_Mapping.getMapped(from)) in
if from.classifierBehavior.oclIsUndefined() then relationships else relationships->append(Cla
```

C.2.4.6.2.29 ViewpointPurposeMetadata_Mapping

Description

*** not specified yet ***

General Mappings

GenericToMetadataUsage_Mapping

Mapping Source

Class

Mapping Target

MetadataUsage

Owned Mappings

- viewpointPurposeMetadataFeatureTyping : ViewpointPurposeMetadataFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- MetadataUsage::ownedRelationship () : Relationship [0..*]

  ```
  Set{viewpointPurposeMetadataFeatureTyping.to}
  ```

C.2.4.6.2.30 ViewpointPurposeMetadataFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *Class* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Class

Mapping Target

FeatureTyping

Owned Mappings

- viewpointPurposeMetadata : ViewpointPurposeMetadata_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

  ```
  viewpointPurposeMetadata.to
  ```

- FeatureTyping::type () : Type [1]

  ```
  SYSML2::MetadataDefinition.allInstances()->any(m | m.qualifiedName = 'SysMLv1Library::Viewpo
  ```

C.2.4.6.2.31 ViewpointPurposeMetadataMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Class* mapping.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Class

Mapping Target

OwningMembership

Owned Mappings

- viewpointPurposeMetadata : ViewpointPurposeMetadata_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

```
viewpointPurposeMetadata.to
```

C.2.4.6.2.32 ViewpointSubject_Mapping

Description

*** not specified yet ***

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Class

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::direction () : FeatureDirectionKind [0..1]

```
KerML::FeatureDirectionKind::_'in'
```

C.2.4.6.2.33 ViewpointSubjectMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Class* mapping.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

Class

Mapping Target

SubjectMembership

Owned Mappings

- viewpointSubject : ViewpointSubject_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- SubjectMembership::ownedMemberParameter () : Feature [0..1]

```
viewpointSubject.to
```

C.2.4.7 PortsAndFlows

C.2.4.7.1 Overview

Table 12. List of all Overview Mapping Specfications

SysML v1 Concept	SysML v2 Concept	Mapping Class
AcceptChangeStructuralFeatureEventAction		*** not specified yet ***
AddFlowPropertyValueOnNestedPortAction		*** not specified yet ***
ChangeStructuralFeatureEvent		*** not specified yet ***
DirectedFeature		*** not specified yet ***
FlowProperty		*** not specified yet ***
FullPort	PartUsage	FullPort_Mapping

SysML v1 Concept	SysML v2 Concept	Mapping Class
InterfaceBlock	PortDefinition	InterfaceBlock_Mapping
InvocationOnNestedPortAction		*** not specified yet ***
ItemFlow	FlowConnectionUsage	ItemFlow_Mapping
ProxyPort		*** not specified yet ***
TriggerOnNestedPort		*** not specified yet ***
~InterfaceBlock		*** not specified yet ***

C.2.4.7.2 Mapping Specifications

C.2.4.7.2.1 AcceptChangeStructuralFeatureEventAction_Mapping

Description

*** not specified yet ***

General Mappings

AcceptEventAction_Mapping

Mapping Source

AcceptEventAction

Mapping Target

AcceptActionUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
 Helper.hasStereotypeApplied(src, 'SysML::Ports&Flows::AcceptChangeStructuralFeatureEventAction')
```

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

C.2.4.7.2.2 FullPort_Mapping

Description

A SysMLv1::FullPort is mapped to a part usage in SysML v2.

General Mappings

Port_Mapping
CommonFullPort_Mapping

Mapping Source

Port

Mapping Target

PartUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
(not from.type.oclIsUndefined()) and Helper.hasStereotypeApplied(src, 'SysML::Ports&Flows::FullPort'
```

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

C.2.4.7.2.3 FullPortMetadata_Mapping

Description

*** not specified yet ***

General Mappings

GenericToMetadataUsage_Mapping

Mapping Source

Port

Mapping Target

MetadataUsage

Owned Mappings

- fullPortMetadataFeatureTyping : FullPortMetadataFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- MetadataUsage::ownedRelationship () : Relationship [0..*]

```
Set{fullPortMetadataFeatureTyping.to, FullPortMetadataFeatureMembership_Mapping.getMapped(fr
```

C.2.4.7.2.4 FullPortMetadataFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *Port* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Port

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

  ```
  FullPortMetadataReferenceUsage_Mapping.getMapped(from)
  ```

C.2.4.7.2.5 FullPortMetadataFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *Port* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Port

Mapping Target

FeatureTyping

Owned Mappings

- fullPortMetadata : FullPortMetadata_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

  ```
  fullPortMetadata.to
  ```

- FeatureTyping::type () : Type [1]

  ```
  SYSML2::MetadataDefinition.allInstances()->any(m | m.qualifiedName = 'SysMLv1Library::PortDa
  ```

C.2.4.7.2.6 FullPortMetadataOwningMembership_Mapping

Description

Creates a owning membership relationship for *ownedMemberElement()* for the *Port* mapping.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Port

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

  ```
  FullPortMetadata_Mapping.getMapped(from)
  ```

C.2.4.7.2.7 FullPortMetadataReferenceUsage_Mapping

Description

Creates a reference usage for the *Port* mapping.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Port

Mapping Target

ReferenceUsage

Owned Mappings

- fullPortMetadataReferenceUsageRedefinition : FullPortMetadataReferenceUsageRedefinition_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
Set{fullPortMetadataReferenceUsageRedefinition.to, FullPortMetadataReferenceUsageFeatureValu
```

C.2.4.7.2.8 FullPortMetadataReferenceUsageFeatureValue_Mapping

Description

Creates a feature value relationship for the mapping class *Port*

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Port

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

```
LiteralBooleanTrue_Mapping.getMapped(from)
```

C.2.4.7.2.9 FullPortMetadataReferenceUsageRedefinition_Mapping

Description

Creates a redefinition relationship for the *redefiningFeature()* and the *redefinedFeature()* for the *Port* mapping.

General Mappings

GenericToRedefinition_Mapping

Mapping Source

Port

Mapping Target

Redefinition

Owned Mappings

- fullPortMetadataReferenceUsage : FullPortMetadataReferenceUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Redefinition::redefinedFeature () : Feature [1]

```
SYSML2::AttributeUsage.allInstances()->any(m | m.qualifiedName = 'SysMLv1Library::PortData::
```

- Redefinition::redefiningFeature () : Feature [1]

```
fullPortMetadataReferenceUsage.to
```

C.2.4.7.2.10 FullPortUntyped_Mapping

Description

*** not specified yet ***

General Mappings

PortUntyped_Mapping
CommonFullPort_Mapping

Mapping Source

Port

Mapping Target

PartUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
from.type.oclIsUndefined() and Helper.hasStereotypeApplied(src, 'SysML::Ports&Flows::FullPort')
```

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

C.2.4.7.2.11 InterfaceBlock_Mapping

Description

*** not specified yet ***

General Mappings

Block_Mapping

Mapping Source

Class

Mapping Target

PortDefinition

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
 Helper.hasStereotypeApplied(from, 'SysML::Ports&Flows::InterfaceBlock')
```

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

C.2.4.7.2.12 ItemFlow_Mapping

Description

*** not specified yet ***

General Mappings

InformationFlow_Mapping

Mapping Source

InformationFlow

Mapping Target

FlowConnectionUsage

Owned Mappings

- itemFlowFeatureMembership : ItemFlowFeatureMembership_Mapping
- itemFlowSourceEndFeatureMembership : ItemFlowSourceEndFeatureMembership_Mapping
- itemFlowTargetEndFeatureMembership : ItemFlowTargetEndFeatureMembership_Mapping

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
Helper.hasStereotypeApplied(from, 'SysML::Ports&Flows::ItemFlow')
```

Mapping rules

The following lists the mapping rules for the target element properties.

- FlowConnectionUsage::target () : Element [0..*]

  ```
  NamedElementMain_Mapping.getMappedColl(from.informationTarget)
  ```

- FlowConnectionUsage::ownedRelationship () : Relationship [0..*]

  ```
  Set{itemFlowFeatureMembership.to, itemFlowSourceEndFeatureMembership.to, itemFlowTargetEndFe
  ```

- FlowConnectionUsage::source () : Element [0..*]

  ```
  NamedElementMain_Mapping.getMappedColl(from.informationSource)
  ```

C.2.4.7.2.13 ItemFlowFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *InformationFlow* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

InformationFlow

Mapping Target

FeatureMembership

Owned Mappings

- itemFlowItemFeature : ItemFlowItemFeature_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
itemFlowItemFeature.to
```

C.2.4.7.2.14 ItemFlowItemFeature_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeature_Mapping

Mapping Source

InformationFlow

Mapping Target

ItemFeature

Owned Mappings

- itemFlowItemFeatureTyping : ItemFlowItemFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ItemFeature::ownedRelationship () : Relationship [0..*]

```
Set{itemFlowItemFeatureTyping.to}
```

C.2.4.7.2.15 ItemFlowItemFeatureTyping_Mapping

Description

Currently, only one conveyed item is supported

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

InformationFlow

Mapping Target

FeatureTyping

Owned Mappings

- itemFlowItemFeature : ItemFlowItemFeature_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
if from.conveyed->size() > 0 then
Classifier_Mapping.getMapped(from.conveyed.get(0))
else OclUndefined
endif
```

- FeatureTyping::typedFeature () : Feature [1]

```
 itemFlowItemFeature.to
```

C.2.4.7.2.16 ItemFlowSourceEndFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *InformationFlow* mapping.

General Mappings

GenericToEndFeatureMembership_Mapping

Mapping Source

InformationFlow

Mapping Target

FeatureMembership

Owned Mappings

- itemFlowSourceFeature : ItemFlowSourceFeature_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

  ```
  itemFlowSourceFeature.to
  ```

C.2.4.7.2.17 ItemFlowSourceFeature_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeature_Mapping

Mapping Source

InformationFlow

Mapping Target

ItemFlowEnd

Owned Mappings

- itemFlowSourceFeatureSubsetting : ItemFlowSourceFeatureSubsetting_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ItemFlowEnd::isEnd () : Boolean [1]

  ```
  true
  ```

- ItemFlowEnd::ownedRelationship () : Relationship [0..*]

  ```
  Set{itemFlowSourceFeatureSubsetting.to}
  ```

C.2.4.7.2.18 ItemFlowSourceFeatureSubsetting_Mapping

Description

Creates a subsetting relationship for the *subsettingFeature()* and the *subsettedFeature()* for the *InformationFlow* mapping.

General Mappings

GenericToSubsetting_Mapping

Mapping Source

InformationFlow

Mapping Target

Subsetting

Owned Mappings

- itemFlowSourceFeature : ItemFlowSourceFeature_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Subsetting::subsettedFeature () : Feature [1]

```
from.source.get(0)
```

- Subsetting::subsettingFeature () : Feature [1]

```
itemFlowSourceFeature.to
```

C.2.4.7.2.19 ItemFlowTargetEndFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *InformationFlow* mapping.

General Mappings

GenericToEndFeatureMembership_Mapping

Mapping Source

InformationFlow

Mapping Target

FeatureMembership

Owned Mappings

- itemFlowTargetFeature : ItemFlowTargetFeature_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

  ```
  itemFlowTargetFeature.to
  ```

C.2.4.7.2.20 ItemFlowTargetFeature_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeature_Mapping

Mapping Source

InformationFlow

Mapping Target

ItemFlowEnd

Owned Mappings

- itemFlowTargetFeatureSubsetting : ItemFlowTargetFeatureSubsetting_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ItemFlowEnd::ownedRelationship () : Relationship [0..*]

  ```
  Set{itemFlowTargetFeatureSubsetting.to}
  ```

- ItemFlowEnd::isEnd () : Boolean [1]

  ```
  true
  ```

C.2.4.7.2.21 ItemFlowTargetFeatureSubsetting_Mapping

Description

Creates a subsetting relationship for the *subsettingFeature()* and the *subsettedFeature()* for the *InformationFlow* mapping.

General Mappings

GenericToSubsetting_Mapping

Mapping Source

InformationFlow

Mapping Target

Subsetting

Owned Mappings

- itemFlowTargetFeature : ItemFlowTargetFeature_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Subsetting::subsettingFeature () : Feature [1]

  ```
  itemFlowTargetFeature.to
  ```

- Subsetting::subsettedFeature () : Feature [1]

  ```
  from.target.get(0)
  ```

C.2.4.7.2.22 OperationDirectedFeature_Mapping

Description

*** not specified yet ***

General Mappings

Operation_Mapping

Mapping Source

Operation

Mapping Target

PerformActionUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
Helper.hasStereotypeApplied(src, 'SysML::Ports&Flows::DirectedFeature')
```

Mapping rules

The following lists the mapping rules for the target element properties.

- PerformActionUsage::direction () : FeatureDirectionKind [0..1]

```
Helper.getKerMLFeatureDirectionKind(Helper.getTagValueAsElement(from,'SysML::Ports&Flows::Di
```

C.2.4.7.2.23 CommonFullPort_Mapping

Description

*** not specified yet ***

General Mappings

PropertyCommon_Mapping

Mapping Source

Port

Mapping Target

PartUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- PartUsage::ownedRelationship () : Relationship [0..*]

```
let typings: Set(KerML::FeatureTyping) = if from.type.oclIsUndefined() then
    Set{}
else
    Set{StructuralFeatureToFeatureTyping_Mapping.getMapped(from)}
endif in
let subsettings: Set(KerML::Subsetting) = from.subsettedProperty
    ->collect(p | PropertySubsetting_Mapping.getMapped(from, p))->asSet() in
let defaultValue: Set(KerML::OwningMembership) = if from.defaultValue.oclIsUndefined() then
    Set{}
else
    Set{DefaultValue_Mapping.getMapped(from)}
endif in
typings->union(subsettings)->union(defaultValue)
->including(MultiplicityMembership_Mapping.getMapped(from))->asSet()
->including(FullPortMetadataOwningMembership_Mapping.getMapped(from))
```

C.2.4.8 Requirements

C.2.4.8.1 Overview

Table 13. List of all Overview Mapping Specfications

SysML v1 Concept	SysML v2 Concept	Mapping Class
AbstractRequirement		*** not specified yet ***
Copy		*** not specified yet ***
DeriveReqt		DeriveReqt_Mapping
Refine		Refine_Mapping
Requirement	RequirementDefinition	Requirement_Mapping
Satisfy	SatisfyRequirementUsage	Satisfy_Mapping
TestCase	VerificationCaseDefinition	TestCaseActivity_Mapping
Trace	Dependency	Trace_Mapping
Verify	RequirementVerificationMembership	Verify_Mapping

C.2.4.8.2 SysML v1 Requirements elements not mapped

Table 14. List of SysML v1 elements not mapped of this section

SysML v1 Concept	Rationale
Copy	The copy relationship is not covered by SysML v2.

C.2.4.8.3 Mapping Specifications

C.2.4.8.3.1 Requirement_Mapping

Description

A SysML::Requirement is mapped to a SysMLv2::RequirementDefinition.

General Mappings

GenericToDefinition_Mapping
NamedElementMain_Mapping

Mapping Source

NamedElement

Mapping Target

RequirementDefinition

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
Helper.isRequirement(src)
```

Mapping rules

The following lists the mapping rules for the target element properties.

- RequirementDefinition::reqId () : String [1]

  ```
   let stereotype: UML::Stereotype = Helper.getRequirementStereotype(from) in
  Helper.getTagValue2(from, stereotype, 'id').oclAsType(String)
  ```

- RequirementDefinition::ownedRelationship () : Relationship [0..*]

  ```
   ElementOwnership_Mapping.getMappedColl(from.ownedElement)
  ->including(CommonReturnParameterReferenceUsageMembership_Mapping.getMapped(from))
  ->including(RequirementDocumentationMembership_Mapping.getMapped(from))
  -->including(RequirementSubjectMembership_Mapping.getMapped(from))
  ```

C.2.4.8.3.2 DeriveReqt_Mapping

Description

*** not specified yet ***

General Mappings

Abstraction_Mapping

Mapping Source

Abstraction

Mapping Target

Dependency

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
Helper.hasStereotypeApplied(src, 'SysML::Requirements::DeriveReqt')
```

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

C.2.4.8.3.3 Refine_Mapping

Description

*** not specified yet ***

General Mappings

Abstraction_Mapping

Mapping Source

Abstraction

Mapping Target

Dependency

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
Helper.hasStereotypeApplied(from, 'SysML::Requirements::Refine')
```

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

C.2.4.8.3.4 RequirementDocumentation_Mapping

Description

The mapping class creates a Comment contained in a Requirement which contains the SysMLv1::AbstractRequirement::text property.

General Mappings

GenericToDocumentation_Mapping

Mapping Source

NamedElement

Mapping Target

Documentation

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Documentation::body () : String [1]

```
        Helper.getTagValueAsString(from,'SysML::Requirements::AbstratRequirement', 'text')
```

C.2.4.8.3.5 RequirementDocumentationMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *NamedElement* mapping.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

NamedElement

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

```
RequirementDocumentation_Mapping.getMapped(from)
```

C.2.4.8.3.6 RequirementSubjectMembership_Mapping

Description

The subject is not used, because it is not a SysML v1 concept, but must be created for a SysML v2 requirement.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

NamedElement

Mapping Target

SubjectMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- SubjectMembership::ownedMemberParameter () : Feature [0..1]

```
Helper.getV1V2Lib_PartUsage('something')
```

C.2.4.8.3.7 Satisfy_Mapping

Description

*** not specified yet ***

General Mappings

GenericToOccurrenceUsage_Mapping
Abstraction_Mapping

Mapping Source

Abstraction

Mapping Target

SatisfyRequirementUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
let satisfy: UML::Abstraction = src.oclAsType(UML::Abstraction) in
    if satisfy.oclIsUndefined() then
        false
    else
        Helper.hasStereotypeApplied(satisfy, 'SysML::Requirements::Satisfy') and
        satisfy.client->exists(c | not c.oclIsKindOf(UML::Classifier))
    endif
```

Mapping rules

The following lists the mapping rules for the target element properties.

- SatisfyRequirementUsage::ownedRelationship () : Relationship [0..*]

```
Set{SatisfyFeatureTyping_Mapping.getMapped(from)}
```

C.2.4.8.3.8 TestCaseActivity_Mapping

Description

*** not specified yet ***

General Mappings

ActivityAsDefinition_Mapping

Mapping Source

Activity

Mapping Target

VerificationCaseDefinition

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
Helper.hasStereotypeApplied(from, 'SysML::Requirements::TestCase')
```

Mapping rules

The following lists the mapping rules for the target element properties.

- VerificationCaseDefinition::ownedRelationship () : Relationship [0..*]

```
let relationships : Set(KerML::Relationship) = Helper.activityOwnedRelationship(from) in
let verdictParameter : Set(UML::Parameter) = from.ownedElement->select(e | e.oclIsKindOf(UML:
let parameters : Set(UML::Paramter) = ((from.ownedElement->select(e | e.oclIsKindOf(UML::Para
let verifyRelationships : Set(UML::Abstraction) = from.clientDependency->select( v | Helper.h
relationships
->union(parameters->collect(p | ParameterMembership_Mapping.getMapped(p)))
->union(verdictParameter->collect(vp | TestCaseActivityReturnParameterMembership_Mapping.getM
->including(CaseSubjectMembership_Mapping.getMapped(from))
->including(CaseObjectiveMembership_Mapping.getMapped(from))
-->union(verifyRelationships->collect(v | Verify_Mapping.getMapped(v)))
```

C.2.4.8.3.9 TestCaseActivityReturnParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Parameter* mapping.

General Mappings

ParameterMembership_Mapping

Mapping Source

Parameter

Mapping Target

ReturnParameterMembership

Owned Mappings

(none)

C.2.4.8.3.10 TestCaseVerifyObjectiveMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Abstraction* mapping.

General Mappings

CaseObjectiveMembership_Mapping

Mapping Source

Abstraction

Mapping Target

ObjectiveMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ObjectiveMembership::ownedMemberFeature () : Feature [1]

```
TestCaseVerifyObjectiveRequirementUsage_Mapping.getMapped(from)
```

C.2.4.8.3.11 TestCaseVerifyObjectiveRequirementUsage_Mapping

Description

*** not specified yet ***

General Mappings

CaseObjectiveRequirementUsage_Mapping

Mapping Source

Abstraction

Mapping Target

RequirementUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- RequirementUsage::ownedRelationship () : Relationship [0..*]

```
Set{Verify_Mapping.getMapped(from)}
```

C.2.4.8.3.12 TestCaseVerifyRequirementUsageReferenceSubsetting_Mapping

Description

Creates a subsetting relationship for the *subsettingFeature()* and the *subsettedFeature()* for the *Abstraction* mapping.

General Mappings

GenericToSubsetting_Mapping

Mapping Source

Abstraction

Mapping Target

ReferenceSubsetting

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceSubsetting::referencedFeature () : Feature [1]

```
from.supplier->get(0)
```

C.2.4.8.3.13 TestCaseVerifyRequirementUsage_Mapping

Description

*** not specified yet ***

General Mappings

GenericToUsage_Mapping

Mapping Source

Abstraction

Mapping Target

RequirementUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- RequirementUsage::ownedRelationship () : Relationship [0..*]

```
Set{TestCaseVerifyRequirementUsageReferenceSubsetting_Mapping.getMapped(from), CaseSubjectMe
```

C.2.4.8.3.14 Trace_Mapping

Description

*** not specified yet ***

General Mappings

Abstraction_Mapping

Mapping Source

Abstraction

Mapping Target

Dependency

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
Helper.hasStereotypeApplied(from, 'SysML::Requirements::Trace')
```

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

C.2.4.8.3.15 Verify_Mapping

Description

*** not specified yet ***

General Mappings

GenericToRelationship_Mapping

Mapping Source

Abstraction

Mapping Target

RequirementVerificationMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- RequirementVerificationMembership::ownedRelatedElement () : Element [0..*]

  ```
  Set{TestCaseVerifyRequirementUsage_Mapping.getMapped(from)}
  ```

C.2.5 UML4SysML

C.2.5.1 Overview

C.2.5.2 Actions

C.2.5.2.1 Overview

Table 15. List of all Overview Mapping Specfications

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
AcceptCallAction	AcceptActionUsage	AcceptCallAction_Mapping	

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
AcceptEventAction	AcceptActionUsage AcceptActionUsage AcceptActionUsage	AcceptCallAction_Mapping AcceptChangeStructuralFeatureEventAction_Mapping AcceptEventAction_Mapping	Helper.hasStereotypeApplied(src, 'SysML::Ports&Flows::AcceptChangeStructur
ActionInputPin			
AddStructuralFeatureValueAction	ActionUsage	AddStructuralFeatureValueAction_Mapping	
AddVariableValueAction	ActionUsage	AddVariableValueAction_Mapping	
BroadcastSignalAction	ActionUsage	BroadcastSignalAction_Mapping	
CallBehaviorAction	ActionUsage	CallBehaviorAction_Mapping	
CallOperationAction	ActionUsage	CallOperationAction_Mapping	
Clause			
ClearAssociationAction	ActionUsage	ClearAssociationAction_Mapping	
ClearStructuralFeatureAction	ActionUsage	ClearStructuralFeatureAction_Mapping	
ClearVariableAction	ActionUsage	ClearVariableAction_Mapping	
ConditionalNode			
CreateLinkAction	ActionUsage	CreateLinkAction_Mapping	
CreateLinkObjectAction			
CreateObjectAction	ActionUsage	CreateObjectAction_Mapping	
DestroyLinkAction	ActionUsage	DestroyLinkAction_Mapping	
DestroyObjectAction	ActionUsage	DestroyObjectAction_Mapping	
ExpansionRegion			
InputPin			
LinkEndCreationData			
LinkEndData			
LinkEndDestructionData			
LoopNode	ActionUsage	LoopNode_Mapping	
OpaqueAction	ActionUsage	OpaqueAction_Mapping	

OutputPin	ReferenceUsage
	ReferenceUsage
	ReferenceUsage
	ReferenceUsage
	ReferenceUsage
		ReadIsClassifiedObjectActionOutputPin_Mapping
		CreateObjectPin_Mapping
		ValueSpecificationActionOutputPin_Mapping
		ReadExtentActionOutputPin_Mapping
		ReadSelfActionOutputPin_Mapping
			from.owner.oclIsTypeOf(UML::ReadIsClassif
			from.owner.oclIsTypeOf(UML::CreateObjectA
			from.owner.oclIsKindOf(UML::ValueSpecific
			from.owner.oclIsTypeOf(UML::ReadExtentAc
			from.owner.oclIsKindOf(UML::ReadSelfActic

RaiseExceptionAction	ActionUsage	RaiseExceptionAction_Mapping	
ReadExtentAction	ActionUsage	ReadExtentAction_Mapping	
ReadIsClassifiedObjectAction	ActionUsage	ReadIsClassifiedObjectAction_Mapping	
ReadLinkAction	ActionUsage	ReadLinkAction_Mapping	

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
ReadLinkObjectEndAction	ActionUsage	ReadLinkObjectEndAction_Mapping	
ReadSelfAction	ActionUsage	ReadSelfAction_Mapping	
ReadStructuralFeatureAction	ActionUsage	ReadStructuralFeatureAction_Mapping	
ReadVariableAction	ActionUsage	ReadVariableAction_Mapping	
ReclassifyObjectAction	ActionUsage	ReclassifyObjectAction_Mapping	
ReduceAction	ActionUsage	ReduceAction_Mapping	
RemoveStructuralFeatureValueAction	ActionUsage	RemoveStructuralFeatureValueAction_Mapping	
RemoveVariableValueAction	ActionUsage	RemoveVariableValueAction_Mapping	
ReplyAction	ActionUsage	ReplyAction_Mapping	
SendObjectAction	ActionUsage	SendObjectAction_Mapping	
SendSignalAction	ActionUsage ActionUsage	SendObjectAction_Mapping SendSignalAction_Mapping	
SequenceNode	ActionUsage	SequenceNode_Mapping	
StartClassifierBehaviorAction	ActionUsage	StartClassifierBehaviorAction_Mapping	
StartObjectBehaviorAction	ActionUsage	StartObjectBehaviorAction_Mapping	
StructuredActivityNode	ActionUsage ActionUsage ActionUsage	LoopNode_Mapping StructuredActivityNode_Mapping SequenceNode_Mapping	
TestIdentityAction	CalculationUsage	TestIdentityAction_Mapping	
UnmarshallAction	ActionUsage	UnmarshallAction_Mapping	
ValuePin	ReferenceUsage ReferenceUsage	ValuePinUntyped_Mapping ValuePin_Mapping	from.type.oclIsUndefined() not from.type.oclIsUndefined()
ValueSpecificationAction	ActionUsage	ValueSpecificationAction_Mapping	

C.2.5.2.2 SysML v1 Activities elements not mapped

Table 16. List of SysML v1 elements not mapped of this section

SysML v1 Concept	Rationale
AcceptCallAction	Since the CallEvent is not supported by SysML v2, the AcceptCallAction is also not covered. It is mapped to an empty action usage to keep the connections within the activity respectively action definition.
ActionInputPin	The UML4SysML::ActionInputPin concept is not covered by SysML v2. The model element is mapped as a input or output pin, but without the special action input pin semantics.

SysML v1 Concept	Rationale
ReclassifyObjectAction	The SysMLv1::ReclassifyObjectAction is not supported by SysML v2. It is mapped to an empty action usage to keep the connections within the activity respectively action definition.
ReplyAction	The UML4SysML::ReplyAction is only used with UML4SysML::AcceptCallAction. Since we have no mapping of AcceptCallAction to SysML v2, there is also no mapping for ReplyAction. However, it is mapped to an empty action usage to keep the connections within the activity respectively action definition.
StartClassifierBehaviorAction	The SysMLv1::StartClassifierBehaviorAction is not supported by SysML v2. It is mapped to an empty action usage to keep the connections within the activity respectively action definition.
StartObjectBehaviorAction	The SysMLv1::StartObjectBehaviorAction is not supported by SysML v2. It is mapped to an empty action usage to keep the connections within the activity respectively action definition.

C.2.5.2.3 Mapping Specifications

C.2.5.2.3.1 Accept Event Actions

C.2.5.2.3.1.1 AcceptCallAction_Mapping

Description

Since the CallEvent is not supported by SysML v2, the AcceptCallAction is also not covered. It is mapped to an empty action usage to keep the connections within the activity respectively action definition.

General Mappings

AcceptEventAction_Mapping

Mapping Source

AcceptCallAction

Mapping Target

AcceptActionUsage

Owned Mappings

(none)

C.2.5.2.3.1.2 AcceptEventAction_Mapping

Description

The UML4SysML::AcceptEventAction is mapped to a AcceptActionUsage element. If the trigger is a signal, it is mapped to an accept parameter typed by the signal. SysMLv2 does not support more than one trigger. Therefore only the first specified trigger of the action is transformed. All further triggers are ignored. The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action acceptEventAction1 accept : SysMLv1Signal;
```

General Mappings

CommonAction_Mapping

Mapping Source

AcceptEventAction

Mapping Target

AcceptActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- AcceptActionUsage::ownedRelationship () : Relationship [0..*]

  ```
  Helper.actionOwnedRelationship(from)->including(AcceptEventActionParameterMembership_Mapping
  ```

C.2.5.2.3.1.3 AcceptEventActionParameter_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeature_Mapping

Mapping Source

AcceptEventAction

Mapping Target

ReferenceUsage

Owned Mappings

- acceptEventActionParameterFeatureTyping : AcceptEventActionParameterFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::direction () : FeatureDirectionKind [0..1]

  ```
  KerML::FeatureDirectionKind::_'in'
  ```

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

  ```
  Set{acceptEventActionParameterFeatureTyping.to}
  ```

C.2.5.2.3.1.4 AcceptEventActionParameterFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *AcceptEventAction* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

AcceptEventAction

Mapping Target

FeatureTyping

Owned Mappings

- acceptEventActionParameter : AcceptEventActionParameter_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

  ```
  acceptEventActionParameter
  ```

- FeatureTyping::type () : Type [1]

  ```
  let event : UML::Event = from.trigger.get(0).event in
  if event.oclIsTypeOf(UML::SignalEvent) then event.oclAsType(UML::SignalEvent).signal else Ocl
  ```

C.2.5.2.3.1.5 AcceptEventActionParameterMembership_Mapping

Description

The mapping class creates the parameter membership relationship for the element that can be received by the accept action. The source of the element is the trigger of the UML::AcceptEventAction. Currently, more than one trigger is not supported by the transformation.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

AcceptEventAction

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ParameterMembership::ownedMemberParameter () : Feature [1]

  ```
  AcceptEventActionParameter_Mapping.getMapped(from)
  ```

C.2.5.2.3.1.6 ReplyAction_Mapping

Description

The UML4SysML::ReplyAction is only used with UML4SysML::AcceptCallAction. Since we have no mapping of AcceptCallAction to SysML v2, there is also no mapping for ReplyAction. However, it is mapped to an empty action usage to keep the connections within the activity respectively action definition.

General Mappings

CommonAction_Mapping

Mapping Source

ReplyAction

Mapping Target

ActionUsage

Owned Mappings

(none)

C.2.5.2.3.1.7 UnmarshallAction_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping

Mapping Source

UnmarshallAction

Mapping Target

ActionUsage

Owned Mappings

(none)

C.2.5.2.3.2 Actions

C.2.5.2.3.2.1 CommonAction_Mapping

Description

Base mapping class for model elements of kind UML4SysML::Action. The target element is a SysMLv2::ActionUsage.

General Mappings

GenericToActionUsage_Mapping
NamedElementMain_Mapping

Mapping Source

Action

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ActionUsage::isComposite () : Boolean [1]

```
true
```

- ActionUsage::ownedRelationship () : Relationship [0..*]

```
Helper.actionOwnedRelationship(from)
```

C.2.5.2.3.2.2 OpaqueAction_Mapping

Description

The UML4SysML::OpaqueAction is mapped to a SysMLv2::ActionUsage with a textual representation. The following shows an example of the expected SysMLv2 textual syntax of a UML4SysML::OpaqueAction.

```
action thisIsAOpaqueAction {
  in x : ScalarValues::Integer;
  in y : ScalarValues::Integer;
  out result : ScalarValues::Boolean;

  language "OCL"
  /*
   * x = y + 1;
  */
}
```

General Mappings

CommonAction_Mapping

Mapping Source

OpaqueAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ActionUsage::ownedRelationship () : Relationship [0..*]

```
if from.body->size() > 0 then
```

```
        Helper.actionOwnedRelationship(from)->append(OpaqueActionBodyMembership_Mapping.getMapped(fro
        else
        Helper.actionOwnedRelationship(from)
        endif
```

C.2.5.2.3.2.3 OpaqueActionBody_Mapping

Description

The mapping class maps the language and the body properties from the UML4SysML::OpaqueAction to a SysMLv2::TextualRepresentation. Currently, multiple languages and bodies are not supported yet.

General Mappings

GenericToAnnotatingElement_Mapping

Mapping Source

OpaqueAction

Mapping Target

TextualRepresentation

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- TextualRepresentation::body () : String [1]

```
if from.body.notEmpty() then from.body.first() else OclUndefined endif
```

- TextualRepresentation::language () : String [1]

```
if from.language.notEmpty() then from.language.first() else OclUndefined endif
```

C.2.5.2.3.2.4 OpaqueActionBodyMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *OpaqueAction* mapping.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

OpaqueAction

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

  ```
  OpaqueActionBody_Mapping.getMapped(from)
  ```

C.2.5.2.3.2.5 Pin_Mapping

Description

Base mapping class for model elements of kind UML4SysML::Pin with a type. The target element is a SysMLv2::ReferenceUsage.

General Mappings

UntypedPin_Mapping

Mapping Source

Pin

Mapping Target

ReferenceUsage

Owned Mappings

- pinFeatureTyping : PinFeatureTyping_Mapping

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
not from.type.oclIsUndefined()
```

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
        Set{pinFeatureTyping.to, MultiplicityMembership_Mapping.getMapped(from)}
```

C.2.5.2.3.2.6 PinFeatureTyping_Mapping

Description

Creates the feature typing for the UML4SysML::Pin target ReferenceUsage.

General Mappings

TypedElementToFeatureTyping_Mapping

Mapping Source

Pin

Mapping Target

FeatureTyping

Owned Mappings

- pin : Pin_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

```
        pin.to
```

C.2.5.2.3.2.7 UntypedPin_Mapping

Description

Base mapping class for model elements of kind UML4SysML::Pin without a type. The target element is a SysMLv2::ReferenceUsage.

General Mappings

GenericToReferenceUsage_Mapping
NamedElementMain_Mapping

Mapping Source

Pin

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
from.type.oclIsUndefined()
```

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

  ```
  ElementOwnership_Mapping.getMappedColl(from.ownedComment)->including(MultiplicityMembership_
  ```

- ReferenceUsage::direction () : FeatureDirectionKind [0..1]

  ```
  if src.oclIsTypeOf(UML::InputPin) then KerML::FeatureDirectionKind::_'in'
  else if src.oclIsTypeOf(UML::OutputPin) then KerML::FeatureDirectionKind::_'out'
  else OclUndefined endif endif
  ```

C.2.5.2.3.2.8 ValuePin_Mapping

Description

Mapping of UML4SysML::ValuePin with a specified type.

General Mappings

Pin_Mapping

Mapping Source

ValuePin

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
        Set{pinFeatureTyping.to, ValuePinFeatureValue_Mapping.getMapped(from), MultiplicityMembershi
```

C.2.5.2.3.2.9 ValuePinFeatureValue_Mapping

Description

Creates a feature value relationship for the mapping class *ValuePin*

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

ValuePin

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

```
        if from.value.oclIsKindOf(UML::LiteralSpecification) then Helper.mappedValueSpecification(fr
```

C.2.5.2.3.2.10 ValuePinUntyped_Mapping

Description

Mapping of UML4SysML::ValuePin without a specified type.

General Mappings

UntypedPin_Mapping

Mapping Source

ValuePin

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
Set{ValuePinFeatureValue_Mapping.getMapped(from), MultiplicityMembership_Mapping.getMapped(f
```

C.2.5.2.3.3 Invocation Actions

C.2.5.2.3.3.1 BroadcastSignalAction_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping

Mapping Source

BroadcastSignalAction

Mapping Target

ActionUsage

Owned Mappings

(none)

C.2.5.2.3.3.2 CallBehaviorAction_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping

Mapping Source

CallBehaviorAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ActionUsage::ownedRelationship () : Relationship [0..*]

```
Helper.actionOwnedRelationship(from)->append(CallBehaviorFeatureTyping_Mapping.getMapped(fro
```

C.2.5.2.3.3.3 CallBehaviorFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *CallBehaviorAction* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

CallBehaviorAction

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
from.behavior
```

- FeatureTyping::typedFeature () : Feature [1]

```
from
```

C.2.5.2.3.3.4 CallOperationAction_Mapping

Description

A UML4SysML::CallOperationAction is mapped to a SysMLv2::ActionUsage which calls the operation. The expected SysML v2 textual syntax is as follows.

```
action thisIsACallOperationAction {
  in paramIn;
  in target : ThisIsABlock;
  out paramReturn = target.thisIsAnOperation;
}
```

General Mappings

CommonAction_Mapping

Mapping Source

CallOperationAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ActionUsage::ownedRelationship () : Relationship [0..*]

  ```
  Helper.actionOwnedRelationship(from)->including(CallOperationPerformActionFeatureMembership_
  ```

C.2.5.2.3.3.5 CallOperationOutputPin_Mapping

Description

*** not specified yet ***

General Mappings

Pin_Mapping

Mapping Source

OutputPin

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
from.owner.oclIsTypeOf(UML::CallOperationAction)
```

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

  ```
  Set{CallOperationOutputPinFeatureMembership_Mapping.getMapped(from), pinFeatureTyping.to, Mu
  ```

C.2.5.2.3.3.6 CallOperationOutputPinFeature_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeature_Mapping

Mapping Source

OutputPin

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::ownedRelationship () : Relationship [0..*]

  ```
  Set{CallOperationOutputPinFeatureFeatureValue_Mapping.getMapped(from), CallOperationOutputPi
  ```

- Feature::direction () : FeatureDirectionKind [0..1]

  ```
  KerML::FeatureDirectionKind::_'in'
  ```

C.2.5.2.3.3.7 CallOperationOutputPinFeatureChainExpression_Mapping

Description

*** not specified yet ***

General Mappings

GenericToInvocationExpression_Mapping

Mapping Source

OutputPin

Mapping Target

FeatureChainExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureChainExpression::ownedRelationship () : Relationship [0..*]

```
Set{CallOperationOutputPinParameterMembership_Mapping.getMapped(from), CallOperationOutputPi
```

C.2.5.2.3.3.8 CallOperationOutputPinFeatureChainExpressionMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *OutputPin* mapping.

General Mappings

GenericToMembership_Mapping

Mapping Source

OutputPin

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Membership::memberElement () : Element [1]

```
from.owner.oclAsType(UML::CallOperationAction).operation
```

C.2.5.2.3.3.9 CallOperationOutputPinFeatureFeature_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeature_Mapping

Mapping Source

OutputPin

Mapping Target

Feature

Owned Mappings

(none)

C.2.5.2.3.3.10 CallOperationOutputPinFeatureFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *OutputPin* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

OutputPin

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
CallOperationOutputPinFeatureFeature_Mapping.getMapped(from)
```

C.2.5.2.3.3.11 CallOperationOutputPinFeatureFeatureValue_Mapping

Description

Creates a feature value relationship for the mapping class *OutputPin*

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

OutputPin

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

```
CallOperationOutputPinFeatureReferenceExpression_Mapping.getMapped(from)
```

C.2.5.2.3.3.12 CallOperationOutputPinFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *OutputPin* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

OutputPin

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
CallOperationOutputPinReferenceUsage_Mapping.getMapped(from)
```

C.2.5.2.3.3.13 CallOperationOutputPinFeatureReferenceExpression_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeatureReferenceExpression_Mapping

Mapping Source

OutputPin

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureReferenceExpression::ownedRelationship () : Relationship [0..*]

```
Set{CallOperationOutputPinFeatureReferenceExpressionMembership_Mapping.getMapped(from), Empt
```

C.2.5.2.3.3.14 CallOperationOutputPinFeatureReferenceExpressionMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *OutputPin* mapping.

General Mappings

GenericToMembership_Mapping

Mapping Source

OutputPin

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Membership::memberElement () : Element [1]

  ```
  from.owner.oclAsType(UML::CallOperationAction).target
  ```

C.2.5.2.3.3.15 CallOperationOutputPinParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *OutputPin* mapping.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

OutputPin

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ParameterMembership::ownedMemberParameter () : Feature [1]

  ```
  CallOperationOutputPinFeature_Mapping.getMapped(from)
  ```

- ParameterMembership::visibility () : VisibilityKind [1]

  ```
  KerML::VisibilityKind::private
  ```

C.2.5.2.3.3.16 CallOperationOutputPinReferenceUsage_Mapping

Description

Creates a reference usage for the *OutputPin* mapping.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

OutputPin

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

  ```
  Set{CallOperationOutputPinReferenceUsageFeatureValue_Mapping.getMapped(from)}
  ```

C.2.5.2.3.3.17 CallOperationOutputPinReferenceUsageFeatureValue_Mapping

Description

Creates a feature value relationship for the mapping class *OutputPin*

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

OutputPin

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

  ```
  CallOperationOutputPinFeatureChainExpression_Mapping.getMapped(from)
  ```

C.2.5.2.3.3.18 CallOperationPerformAction_Mapping

Description

*** not specified yet ***

General Mappings

GenericToActionUsage_Mapping

Mapping Source

CallOperationAction

Mapping Target

PerformActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- PerformActionUsage::ownedRelationship () : Relationship [0..*]

```
Set{CallOperationPerformActionReferenceSubsetting_Mapping.getMapped(from)}
```

C.2.5.2.3.3.19 CallOperationPerformActionFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *CallOperationAction* mapping.

General Mappings

GenericToEndFeatureMembership_Mapping

Mapping Source

CallOperationAction

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
CallOperationPerformAction_Mapping.getMapped(from)
```

C.2.5.2.3.3.20 CallOperationPerformActionReferenceSubsetting_Mapping

Description

Creates a subsetting relationship for the *subsettingFeature()* and the *subsettedFeature()* for the *CallOperationAction* mapping.

General Mappings

GenericToReferenceSubsetting_Mapping

Mapping Source

CallOperationAction

Mapping Target

ReferenceSubsetting

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceSubsetting::ownedRelatedElement () : Element [0..*]

```
Set{CallOperationPerformActionReferenceSubsettingFeature_Mapping.getMapped(from)}
```

C.2.5.2.3.3.21 CallOperationPerformActionReferenceSubsettingFeature_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeature_Mapping

Mapping Source

CallOperationAction

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::ownedRelationship () : Relationship [0..*]

```
Set{CallOperationPerformActionReferenceSubsettingFeatureChainingTarget_Mapping.getMapped(fro
```

C.2.5.2.3.3.22 CallOperationPerformActionReferenceSubsettingFeatureChainingOperation_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeatureChaining_Mapping

Mapping Source

CallOperationAction

Mapping Target

FeatureChaining

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureChaining::chainingFeature () : Feature [1]

```
from.operation
```

C.2.5.2.3.3.23 CallOperationPerformActionReferenceSubsettingFeatureChainingTarget_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeatureChaining_Mapping

Mapping Source

CallOperationAction

Mapping Target

FeatureChaining

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureChaining::chainingFeature () : Feature [1]

```
from.target
```

C.2.5.2.3.3.24 SendSignalAction_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping

Mapping Source

SendSignalAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ActionUsage::ownedRelationship () : Relationship [0..*]

```
Helper.actionOwnedRelationship(from)->including(SendActionFeatureMembership_Mapping.getMappe
```

C.2.5.2.3.3.25 SendObjectAction_Mapping

Description

*** not specified yet ***

General Mappings

SendSignalAction_Mapping

Mapping Source

SendObjectAction

Mapping Target

ActionUsage

Owned Mappings

(none)

C.2.5.2.3.3.26 SendActionFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *InvocationAction* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

InvocationAction

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
SendActionSendActionUsage_Mapping.getMapped(from)
```

C.2.5.2.3.3.27 SendActionParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *InvocationAction* mapping.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

InvocationAction

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ParameterMembership::ownedMemberParameter () : Feature [1]

  ```
  SendActionReferenceUsage_Mapping.getMapped(from)
  ```

C.2.5.2.3.3.28 SendActionReferenceUsage_Mapping

Description

Creates a reference usage for the *InvocationAction* mapping.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

InvocationAction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::direction () : FeatureDirectionKind [0..1]

  ```
  KerML::FeatureDirectionKind::_'in'
  ```

C.2.5.2.3.3.29 SendActionItemParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *InvocationAction* mapping.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

InvocationAction

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ParameterMembership::ownedMemberParameter () : Feature [1]

```
SendActionItemReferenceUsage_Mapping.getMapped(from)
```

C.2.5.2.3.3.30 SendActionItemReferenceUsage_Mapping

Description

Creates a reference usage for the *InvocationAction* mapping.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

InvocationAction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

  ```
  Set{SendActionItemReferenceUsageFeatureValue_Mapping.getMapped(from)}
  ```

- ReferenceUsage::direction () : FeatureDirectionKind [0..1]

  ```
  KerML::FeatureDirectionKind::_'in'
  ```

C.2.5.2.3.3.31 SendActionItemReferenceUsageFeatureValue_Mapping

Description

Creates a feature value relationship for the mapping class *InvocationAction*

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

InvocationAction

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

  ```
  SendActionItemReferenceUsageFeatureValueValue_Mapping.getMapped(from)
  ```

C.2.5.2.3.3.32 SendActionItemReferenceUsageFeatureValueTyping_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

InvocationAction

Mapping Target

FeatureTyping

Owned Mappings

- sendActionItemReferenceUsageFeatureValueValue : SendActionItemReferenceUsageFeatureValueValue_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

```
 sendActionItemReferenceUsageFeatureValueValue.to
```

- FeatureTyping::type () : Type [1]

```
if from.oclIsTypeOf(UML::SendSignalAction) then from.signal
else if from.oclIsTypeOf(UML::SendObjectAction) then from.request else OclUndefined endif enc
```

C.2.5.2.3.3.33 SendActionItemReferenceUsageFeatureValueValue_Mapping

Description

*** not specified yet ***

General Mappings

GenericToInvocationExpression_Mapping

Mapping Source

InvocationAction

Mapping Target

InvocationExpression

Owned Mappings

- sendActionItemReferenceUsageFeatureValueTyping : SendActionItemReferenceUsageFeatureValueTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- InvocationExpression::ownedRelationship () : Relationship [0..*]

```
Set{sendActionItemReferenceUsageFeatureValueTyping.to, EmptyReturnParameterFeatureMembership
```

C.2.5.2.3.3.34 SendActionTargetParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *InvocationAction* mapping.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

InvocationAction

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ParameterMembership::ownedMemberParameter () : Feature [1]

```
SendActionTargetReferenceUsage_Mapping.getMapped(from)
```

C.2.5.2.3.3.35 SendActionTargetReferenceUsage_Mapping

Description

Creates a reference usage for the *InvocationAction* mapping.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

InvocationAction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::direction () : FeatureDirectionKind [0..1]

  ```
  KerML::FeatureDirectionKind::_'in'
  ```

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

  ```
  Set{SendActionTargetReferenceUsageFeatureValue_Mapping.getMapped(from)}
  ```

C.2.5.2.3.3.36 SendActionTargetReferenceUsageFeatureValue_Mapping

Description

Creates a feature value relationship for the mapping class *InvocationAction*

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

InvocationAction

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

  ```
  SendActionTargetReferenceUsageFeatureValueExpression_Mapping.getMapped(from)
  ```

C.2.5.2.3.3.37 SendActionTargetReferenceUsageFeatureValueMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *InvocationAction* mapping.

General Mappings

GenericToMembership_Mapping

Mapping Source

InvocationAction

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Membership::memberElement () : Element [1]

  ```
  from.target
  ```

C.2.5.2.3.3.38 SendActionTargetReferenceUsageFeatureValueExpression_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeatureReferenceExpression_Mapping

Mapping Source

InvocationAction

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureReferenceExpression::ownedRelationship () : Relationship [0..*]

```
Set{SendActionTargetReferenceUsageFeatureValueMembership_Mapping.getMapped(from), EmptyRetur
```

C.2.5.2.3.3.39 SendActionSendActionUsage_Mapping

Description

*** not specified yet ***

General Mappings

GenericToActionUsage_Mapping

Mapping Source

InvocationAction

Mapping Target

SendActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- SendActionUsage::ownedRelationship () : Relationship [0..*]

```
Set{SendActionItemParameterMembership_Mapping.getMapped(from), SendActionParameterMembership
```

C.2.5.2.3.3.40 StartClassifierBehaviorAction_Mapping

Description

The SysMLv1::StartClassifierBehaviorAction is not supported by SysML v2. It is mapped to an empty action usage to keep the connections within the activity respectively action definition.

General Mappings

CommonAction_Mapping

Mapping Source

StartClassifierBehaviorAction

Mapping Target

ActionUsage

Owned Mappings

(none)

C.2.5.2.3.3.41 StartObjectBehaviorAction_Mapping

Description

The SysMLv1::StartObjectBehaviorAction is not supported by SysML v2. It is mapped to an empty action usage to keep the connections within the activity respectively action definition.

General Mappings

CommonAction_Mapping

Mapping Source

StartObjectBehaviorAction

Mapping Target

ActionUsage

Owned Mappings

(none)

C.2.5.2.3.4 Link Actions

C.2.5.2.3.4.1 ClearAssociationAction_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping

Mapping Source

ClearAssociationAction

Mapping Target

ActionUsage

Owned Mappings

(none)

C.2.5.2.3.4.2 CreateLinkAction_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping

Mapping Source

CreateLinkAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ActionUsage::ownedRelationship () : Relationship [0..*]

  ```
  let linkEndCreationData : Set(UML::Element) = src.ownedElement->select(e | e.oclIsTypeOf(UML:
  let actionInputPin: Set(UML::Element) = src.ownedElement->select(e | e.oclIsTypeOf(UML::Actic
  let triggers: Set(UML::Element) = src.ownedElement->select(e | e.oclIsKindOf(UML::Trigger)) i
  let toElementFMS: Set(UML::Element) = src.ownedElement->select(e | e.oclIsKindOf(UML::Pin)) i
  let toElementOMS: Set(UML::Element) = (((src.ownedElement - toElementFMS) - actionInputPin)
  toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
  ->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
  ```

C.2.5.2.3.4.3 DestroyLinkAction_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping

Mapping Source

DestroyLinkAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ActionUsage::ownedRelationship () : Relationship [0..*]

```
let actionInputPin: Set(UML::Element) = src.ownedElement->select(e | e.oclIsTypeOf(UML::Actic
let triggers: Set(UML::Element) = src.ownedElement->select(e | e.oclIsKindOf(UML::Trigger)) i
let linkData: Set(UML::Element) = src.ownedElement->select( e | e.oclIsKindOf(UML::LinkEndDat
let toElementFMS: Set(UML::Element) = src.ownedElement->select(e | e.oclIsKindOf(UML::Pin)) i
let toElementOMS: Set(UML::Element) = (((src.ownedElement - toElementFMS) - actionInputPin)
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
```

C.2.5.2.3.4.4 ReadLinkAction_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping

Mapping Source

ReadLinkAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ActionUsage::ownedRelationship () : Relationship [0..*]

```
let actionInputPin: Set(UML::Element) = src.ownedElement->select(e | e.oclIsTypeOf(UML::Actic
let triggers: Set(UML::Element) = src.ownedElement->select(e | e.oclIsKindOf(UML::Trigger)) i
let linkData: Set(UML::Element) = src.ownedElement->select( e | e.oclIsKindOf(UML::LinkEndDat
let toElementFMS: Set(UML::Element) = src.ownedElement->select(e | e.oclIsKindOf(UML::Pin)) i
let toElementOMS: Set(UML::Element) = (((src.ownedElement - toElementFMS) - actionInputPin)
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
```

C.2.5.2.3.4.5 ReadLinkObjectEndAction_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping

Mapping Source

ReadLinkObjectEndAction

Mapping Target

ActionUsage

Owned Mappings

(none)

C.2.5.2.3.4.6 ReadLinkObjectEndQualifierAction_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping

Mapping Source

ReadLinkObjectEndQualifierAction

Mapping Target

ActionUsage

Owned Mappings

(none)

C.2.5.2.3.5 Object Actions

C.2.5.2.3.5.1 CommonFeatureReferenceExpression_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeatureReferenceExpression_Mapping

Mapping Source

TypedElement

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureReferenceExpression::ownedRelationship () : Relationship [0..*]

```
Set{CommonMembership_Mapping.getMapped(from), CommonReturnParameterFeatureMembership_Mapping
```

C.2.5.2.3.5.2 CommonReferenceUsageIn_Mapping

Description

*** not specified yet ***

General Mappings

CommonReferenceUsageInUntyped_Mapping

Mapping Source

TypedElement

Mapping Target

ReferenceUsage

Owned Mappings

- commonReferenceUsageInFeatureTyping : CommonReferenceUsageInFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
Set{commonReferenceUsageInFeatureTyping.to}
```

C.2.5.2.3.5.3 CommonReferenceUsageInFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *TypedElement* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

TypedElement

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
if from.type.oclIsUndefined() then CommonReferenceUsageInUntyped_Mapping.getMapped(from) els
```

C.2.5.2.3.5.4 CreateObjectAction_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping

Mapping Source

CreateObjectAction

Mapping Target

ActionUsage

Owned Mappings

(none)

C.2.5.2.3.5.5 CreateObjectInvocationExpessionFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *CreateObjectAction* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

CreateObjectAction

Mapping Target

FeatureTyping

Owned Mappings

- createObjectInvocationExpression : CreateObjectInvocationExpression_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

  ```
  createObjectInvocationExpression.to
  ```

- FeatureTyping::type () : Type [1]

  ```
  from.classifier
  ```

C.2.5.2.3.5.6 CreateObjectInvocationExpression_Mapping

Description

*** not specified yet ***

General Mappings

GenericToInvocationExpression_Mapping

Mapping Source

CreateObjectAction

Mapping Target

InvocationExpression

Owned Mappings

- createObjectInvocationExpessionFeatureTyping : CreateObjectInvocationExpessionFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- InvocationExpression::ownedRelationship () : Relationship [0..*]

```
Set{createObjectInvocationExpessionFeatureTyping.to, CommonReturnParameterFeatureMembership_
```

C.2.5.2.3.5.7 CreateObjectPin_Mapping

Description

*** not specified yet ***

General Mappings

Pin_Mapping

Mapping Source

OutputPin

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
from.owner.oclIsTypeOf(UML::CreateObjectAction)
```

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
Set{pinFeatureTyping.to, CreateObjectPinFeatureValue_Mapping.getMapped(from)}
```

C.2.5.2.3.5.8 CreateObjectPinFeatureValue_Mapping

Description

Creates a feature value relationship for the mapping class *OutputPin*

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

OutputPin

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

```
CreateObjectInvocationExpression_Mapping.getMapped(from.owner)
```

C.2.5.2.3.5.9 DestroyObjectAction_Mapping

Description

The UML4SysML::DestroyObjectAction is conceptually mapped to the SysML v2 library function OccurrenceFunctions::destroy. The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action destroyObjectAction1 {
  in target : Block1;
  action : OccurrenceFunctions::destroy {in occ = target;}
}
part def Block1;
```

General Mappings

CommonAction_Mapping

Mapping Source

DestroyObjectAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ActionUsage::ownedRelationship () : Relationship [0..*]

```
Helper.actionOwnedRelationship(from)->including(DestroyObjectActionDestroyFeatureMembership_
```

C.2.5.2.3.5.10 EqualOperatorExpressionOperand_Mapping

Description

*** not specified yet ***

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

TypedElement

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ParameterMembership::ownedMemberParameter () : Feature [1]

```
EqualOperatorExpressionFeature_Mapping.getMapped(from)
```

- ParameterMembership::visibility () : VisibilityKind [1]

```
        KerML::VisibilityKind::private
```

C.2.5.2.3.5.11 ReadIsClassifiedObjectAction_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping

Mapping Source

ReadIsClassifiedObjectAction

Mapping Target

ActionUsage

Owned Mappings

(none)

C.2.5.2.3.5.12 ReadIsClassifiedObjectActionFeatureValue_Mapping

Description

Creates a feature value relationship for the mapping class *ReadIsClassifiedObjectAction*

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

ReadIsClassifiedObjectAction

Mapping Target

FeatureValue

Owned Mappings

- *readIsClassifiedObjectActionFeatureValueOperatorExpression : ReadIsClassifiedObjectActionFeatureValueOperatorExpression_Mapping*

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

```
readIsClassifiedObjectActionFeatureValueOperatorExpression.to
```

C.2.5.2.3.5.13 ReadIsClassifiedObjectActionFeatureValueOperatorExpression_Mapping

Description

*** not specified yet ***

General Mappings

GenericToExpression_Mapping

Mapping Source

ReadIsClassifiedObjectAction

Mapping Target

OperatorExpression

Owned Mappings

- readIsClassifiedObjectActionFeatureValueOperatorExpressionParameterMembership : ReadIsClassifiedObjectActionFeatureValueOperatorExpressionParameterMembership_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OperatorExpression::operator () : String [1]

  ```
  if from.isDirect then 'istype' else 'hastype' endif
  ```

- OperatorExpression::ownedRelationship () : Relationship [0..*]

  ```
  Set{readIsClassifiedObjectActionFeatureValueOperatorExpressionParameterMembership.to}
  ```

C.2.5.2.3.5.14 ReadIsClassifiedObjectActionFeatureValueOperatorExpressionFeature_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeature_Mapping

Mapping Source

ReadIsClassifiedObjectAction

Mapping Target

Feature

Owned Mappings

- readIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValue : ReadIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValue_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::direction () : FeatureDirectionKind [0..1]

```
KerML::FeatureDirectionKind::_'in'
```

- Feature::ownedRelationship () : Relationship [0..*]

```
Set{readIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValue.to}
```

C.2.5.2.3.5.15 ReadIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValue_Mapping

Description

Creates a feature value relationship for the mapping class *ReadIsClassifiedObjectAction*

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

ReadIsClassifiedObjectAction

Mapping Target

FeatureValue

Owned Mappings

- *readIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValueExpression : ReadIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValueExpression_Mapping*

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

```
readIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValueExpression.to
```

C.2.5.2.3.5.16 ReadIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValueExpression_Mapping

Description

*** not specified yet ***

General Mappings

GenericToExpression_Mapping

Mapping Source

ReadIsClassifiedObjectAction

Mapping Target

FeatureReferenceExpression

Owned Mappings

- readIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValueExpressionMembership : ReadIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValueExpressionMembership_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureReferenceExpression::ownedRelationship () : Relationship [0..*]

```
Set{readIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValueExpressionMembersh
```

C.2.5.2.3.5.17 ReadIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValueExpressionMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *ReadIsClassifiedObjectAction* mapping.

General Mappings

GenericToMembership_Mapping

Mapping Source

ReadIsClassifiedObjectAction

Mapping Target

Membership

Owned Mappings

(none)

C.2.5.2.3.5.18 ReadIsClassifiedObjectActionFeatureValueOperatorExpressionParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *ReadIsClassifiedObjectAction* mapping.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

ReadIsClassifiedObjectAction

Mapping Target

ParameterMembership

Owned Mappings

- readIsClassifiedObjectActionFeatureValueOperatorExpressionFeature : ReadIsClassifiedObjectActionFeatureValueOperatorExpressionFeature_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ParameterMembership::ownedMemberParameter () : Feature [1]

```
readIsClassifiedObjectActionFeatureValueOperatorExpressionFeature.to
```

- ParameterMembership::visibility () : VisibilityKind [1]

```
KerML::VisibilityKind::private
```

C.2.5.2.3.5.19 ReadIsClassifiedObjectActionOutputPin_Mapping

Description

*** not specified yet ***

General Mappings

Pin_Mapping

Mapping Source

OutputPin

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
from.owner.oclIsTypeOf(UML::ReadIsClassifiedObjectAction)
```

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

  ```
  Set{pinFeatureTyping.to, ReadIsClassifiedObjectActionFeatureValue_Mapping.getMapped(from.owr
  ```

C.2.5.2.3.5.20 ReadExtentAction_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping

Mapping Source

ReadExtentAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ActionUsage::ownedRelationship () : Relationship [0..*]

  ```
  Helper.actionOwnedRelationship(from)
  ```

C.2.5.2.3.5.21 ReadExtentActionFeatureValue_Mapping

Description

Creates a feature value relationship for the mapping class *OutputPin*

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

OutputPin

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

  ```
  ReadExtentActionFeatureValueOperatorExpression_Mapping.getMapped(from)
  ```

C.2.5.2.3.5.22 ReadExtentActionFeatureValueOperatorExpression_Mapping

Description

*** not specified yet ***

General Mappings

GenericToExpression_Mapping

Mapping Source

OutputPin

Mapping Target

OperatorExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OperatorExpression::ownedRelationship () : Relationship [0..*]

  ```
  Set{ReadExtentActionFeatureValueOperatorExpressionMembership_Mapping.getMapped(from), Commor
  ```

- OperatorExpression::operator () : String [1]

  ```
  'all'
  ```

C.2.5.2.3.5.23 ReadExtentActionFeatureValueOperatorExpressionFeature_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeature_Mapping

Mapping Source

OutputPin

Mapping Target

Feature

Owned Mappings

- readExtentActionFeatureValueOperatorExpressionFeatureTyping : ReadExtentActionFeatureValueOperatorExpressionFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::ownedRelationship () : Relationship [0..*]

  ```
  Set{readExtentActionFeatureValueOperatorExpressionFeatureTyping.to}
  ```

C.2.5.2.3.5.24 ReadExtentActionFeatureValueOperatorExpressionFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *OutputPin* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

OutputPin

Mapping Target

FeatureTyping

Owned Mappings

- readExtentActionFeatureValueOperatorExpressionFeature : ReadExtentActionFeatureValueOperatorExpressionFeature_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

```
readExtentActionFeatureValueOperatorExpressionFeature.to
```

- FeatureTyping::type () : Type [1]

```
from.owner.classifier
```

C.2.5.2.3.5.25 ReadExtentActionFeatureValueOperatorExpressionMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *OutputPin* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

OutputPin

Mapping Target

FeatureMembership

Owned Mappings

- readExtentActionFeatureValueOperatorExpressionFeature : ReadExtentActionFeatureValueOperatorExpressionFeature_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
readExtentActionFeatureValueOperatorExpressionFeature
```

C.2.5.2.3.5.26 ReadExtentActionOutputPin_Mapping

Description

*** not specified yet ***

General Mappings

Pin_Mapping

Mapping Source

OutputPin

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
from.owner.oclIsTypeOf(UML::ReadExtentAction)
```

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
Set{pinFeatureTyping.to, ReadExtentActionFeatureValue_Mapping.getMapped(from), MultiplicityM
```

C.2.5.2.3.5.27 ReadSelfAction_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping

Mapping Source

ReadSelfAction

Mapping Target

ActionUsage

Owned Mappings

(none)

C.2.5.2.3.5.28 ReadSelfActionFeatureValue_Mapping

Description

Creates a feature value relationship for the mapping class *OutputPin*

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

OutputPin

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

  ```
  ReadSelfActionFeatureValueFeatureReferenceExpression_Mapping.getMapped(from)
  ```

C.2.5.2.3.5.29 ReadSelfActionFeatureValueFeatureReferenceExpression_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeatureReferenceExpression_Mapping

Mapping Source

OutputPin

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureReferenceExpression::ownedRelationship () : Relationship [0..*]

```
 Set{ReadSelfActionFeatureValueFeatureReferenceExpressionMembership_Mapping.getMapped(from),
CommonReturnParameterFeatureMembership_Mapping.getMapped(from)}
```

C.2.5.2.3.5.30 ReadSelfActionFeatureValueFeatureReferenceExpressionMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *OutputPin* mapping.

General Mappings

GenericToMembership_Mapping

Mapping Source

OutputPin

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Membership::memberElement () : Element [1]

```
SYSML2::Feature.allInstances()->any(e | e.qualifiedName = 'Occurrences::Occurrence::this')
```

C.2.5.2.3.5.31 ReadSelfActionOutputPin_Mapping

Description

*** not specified yet ***

General Mappings

Pin_Mapping

Mapping Source

OutputPin

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
from.owner.oclIsKindOf(UML::ReadSelfAction)
```

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::isUnique () : Boolean [1]

  ```
  false
  ```

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

  ```
  Set{pinFeatureTyping.to, ReadSelfActionFeatureValue_Mapping.getMapped(from), MultiplicityMem
  ```

- ReferenceUsage::isAbstract () : Boolean [1]

  ```
  true
  ```

C.2.5.2.3.5.32 ReclassifyObjectAction_Mapping

Description

The SysMLv1::ReclassifyObjectAction is not supported by SysML v2. It is mapped to an empty action usage to keep the connections within the activity respectively action definition.

General Mappings

CommonAction_Mapping

Mapping Source

ReclassifyObjectAction

Mapping Target

ActionUsage

Owned Mappings

(none)

C.2.5.2.3.5.33 TestIdentityAction_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping

Mapping Source

TestIdentityAction

Mapping Target

CalculationUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- CalculationUsage::ownedRelationship () : Relationship [0..*]

```
 Helper.actionOwnedRelationship(from)
->including(TestIdentityActionResultExpressionMembership_Mapping.getMapped(from))
```

C.2.5.2.3.5.34 TestIdentityActionOperator_Mapping

Description

*** not specified yet ***

General Mappings

GenericToExpression_Mapping

Mapping Source

TestIdentityAction

Mapping Target

OperatorExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OperatorExpression::operator () : String [1]

  ```
  '=='
  ```

- OperatorExpression::ownedRelationship () : Relationship [0..*]

  ```
  Set{EqualOperatorExpressionOperand_Mapping.getMapped(from.first), EqualOperatorExpressionOpe
  ```

C.2.5.2.3.5.35 EqualOperatorExpressionFeature_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeature_Mapping

Mapping Source

TypedElement

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::ownedRelationship () : Relationship [0..*]

  ```
  Set{EqualOperatorExpressionFeatureValue_Mapping.getMapped(from)}
  ```

C.2.5.2.3.5.36 TestIdentityActionResultExpressionMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *TestIdentityAction* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

TestIdentityAction

Mapping Target

ResultExpressionMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ResultExpressionMembership::ownedMemberFeature () : Feature [0..1]

  ```
  TestIdentityActionOperator_Mapping.getMapped(from)
  ```

C.2.5.2.3.5.37 ValueSpecificationAction_Mapping

Description

The ValueSpecificationAction::value element is removed from the set of owned elements in the ownedRelationship() operation. It is considered as the return value of the mapping target of the output pin. The expected SysML v2 textual notation of a SysMLv1::ValueSpecificationAction is as follows:

```
action thisIsAValueSpecificationAction {
  out result : ScalarValues::Integer = 42;
}

action thisIsAnotherValueSpecificationAction {
  out result = thisIsAnOpaqueExpression.result;
  calc thisIsAnOpaqueExpression {
    language "Math"
    /*
```

```
      * 42 + 23
     */
   }
}
```

General Mappings

CommonAction_Mapping

Mapping Source

ValueSpecificationAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ActionUsage::ownedRelationship () : Relationship [0..*]

```
let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | e.oclIsKindOf(UML::Pin))
let toElementOMS: Set(UML::Element) = (from.ownedElement - toElementFMS) - Set{from.value} ir
toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e))
->union(toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e)))
```

C.2.5.2.3.5.38 ValueSpecificationActionOutputPin_Mapping

Description

*** not specified yet ***

General Mappings

Pin_Mapping

Mapping Source

OutputPin

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
from.owner.oclIsKindOf(UML::ValueSpecificationAction)
```

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
 Set{pinFeatureTyping.to, ValueSpecificationActionOutputPinFeatureValue_Mapping.getMapped(fro
```

C.2.5.2.3.5.39 ValueSpecificationActionOutputPinFeatureValue_Mapping

Description

Creates a feature value relationship for the mapping class *OutputPin*

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

OutputPin

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

```
 if from.owner.value.oclIsTypeOf(UML::OpaqueExpression) then
     OpaqueExpressionAsValue_Mapping.getMapped(from.owner.value)
else
     Helper.mappedValueSpecification(from.owner.value)
endif
```

C.2.5.2.3.5.40 DestroyObjectActionDestroyActionUsage_Mapping

Description

*** not specified yet ***

General Mappings

GenericToActionUsage_Mapping

Mapping Source

DestroyObjectAction

Mapping Target

ActionUsage

Owned Mappings

- destroyObjectActionDestroyActionUsageFeatureTyping : DestroyObjectActionDestroyActionUsageFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ActionUsage::ownedRelationship () : Relationship [0..*]

```
Set{destroyObjectActionDestroyActionUsageFeatureTyping.to, DestroyObjectActionDestroyActionU
```

C.2.5.2.3.5.41 DestroyObjectActionDestroyActionUsageFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *DestroyObjectAction* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

DestroyObjectAction

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
DestroyObjectActionDestroyActionUsageReferenceUsage_Mapping.getMapped(from)
```

C.2.5.2.3.5.42 DestroyObjectActionDestroyActionUsageFeatureReferenceExpression_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeatureReferenceExpression_Mapping

Mapping Source

DestroyObjectAction

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureReferenceExpression::ownedRelationship () : Relationship [0..*]

```
Set{DestroyObjectActionDestroyActionUsageFeatureReferenceExpressionMembership_Mapping.getMap
```

C.2.5.2.3.5.43 DestroyObjectActionDestroyActionUsageFeatureReferenceExpressionMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *DestroyObjectAction* mapping.

General Mappings

GenericToMembership_Mapping

Mapping Source

DestroyObjectAction

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Membership::memberElement () : Element [1]

```
from.target
```

C.2.5.2.3.5.44 DestroyObjectActionDestroyActionUsageFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *DestroyObjectAction* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

DestroyObjectAction

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
SysMLv2::Function.allInstances()->any(e | e.qualifiedName =  'OccurrenceFunctions::destroy')
```

- FeatureTyping::typedFeature () : Feature [1]

```
destroyObjectActionDestroyActionUsage.to
```

C.2.5.2.3.5.45 DestroyObjectActionDestroyActionUsageFeatureValue_Mapping

Description

Creates a feature value relationship for the mapping class *DestroyObjectAction*

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

DestroyObjectAction

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

  ```
  DestroyObjectActionDestroyActionUsageFeatureReferenceExpression_Mapping.getMapped(from)
  ```

C.2.5.2.3.5.46 DestroyObjectActionDestroyActionUsageReferenceUsage_Mapping

Description

Creates a reference usage for the *DestroyObjectAction* mapping.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

DestroyObjectAction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
Set{DestroyObjectActionDestroyActionUsageFeatureValue_Mapping.getMapped(from)}
```

C.2.5.2.3.5.47 DestroyObjectActionDestroyFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *DestroyObjectAction* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

DestroyObjectAction

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
DestroyObjectActionDestroyActionUsage_Mapping.getMapped(from)
```

C.2.5.2.3.6 Other Actions

C.2.5.2.3.6.1 RaiseExceptionAction_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping

Mapping Source

RaiseExceptionAction

Mapping Target

ActionUsage

Owned Mappings

(none)

C.2.5.2.3.6.2 ReduceAction_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping

Mapping Source

ReduceAction

Mapping Target

ActionUsage

Owned Mappings

(none)

C.2.5.2.3.7 Structural Feature Actions

C.2.5.2.3.7.1 AddStructuralFeatureValueAction_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping

Mapping Source

AddStructuralFeatureValueAction

Mapping Target

ActionUsage

Owned Mappings

- addStructuralFeatureValueActionAssignActionMembership : AddStructuralFeatureValueActionAssignmentActionMembership_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ActionUsage::ownedRelationship () : Relationship [0..*]

```
 Helper.actionOwnedRelationship(from)
->including(addStructuralFeatureValueActionAssignActionMembership.to)
```

C.2.5.2.3.7.2 AddStructuralFeatureValueActionAssignmentAction_Mapping

Description

*** not specified yet ***

General Mappings

GenericToAssignmentActionUsage_Mapping

Mapping Source

AddStructuralFeatureValueAction

Mapping Target

AssignmentActionUsage

Owned Mappings

(none)

C.2.5.2.3.7.3 AddStructuralFeatureValueActionAssignmentActionMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *AddStructuralFeatureValueAction* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

AddStructuralFeatureValueAction

Mapping Target

FeatureMembership

Owned Mappings

- addStructuralFeatureValueActionAssignmentAction : AddStructuralFeatureValueActionAssignmentAction_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
addStructuralFeatureValueActionAssignmentAction.to
```

- FeatureMembership::memberFeature () : Feature [1]

```
self.ownedMemberFeature()
```

C.2.5.2.3.7.4 ClearStructuralFeatureAction_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping

Mapping Source

ClearStructuralFeatureAction

Mapping Target

ActionUsage

Owned Mappings

(none)

C.2.5.2.3.7.5 ReadStructuralFeatureAction_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping

Mapping Source

ReadStructuralFeatureAction

Mapping Target

ActionUsage

Owned Mappings

(none)

C.2.5.2.3.7.6 RemoveStructuralFeatureValueAction_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping

Mapping Source

RemoveStructuralFeatureValueAction

Mapping Target

ActionUsage

Owned Mappings

(none)

C.2.5.2.3.8 Structured Actions

C.2.5.2.3.8.1 LoopNode_Mapping

Description

*** not specified yet ***

General Mappings

StructuredActivityNode_Mapping

Mapping Source

LoopNode

Mapping Target

ActionUsage

Owned Mappings

(none)

C.2.5.2.3.8.2 SequenceNode_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping
StructuredActivityNode_Mapping

Mapping Source

SequenceNode

Mapping Target

ActionUsage

Owned Mappings

(none)

C.2.5.2.3.8.3 StructuredActivityNode_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping

Mapping Source

StructuredActivityNode

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ActionUsage::ownedRelationship () : Relationship [0..*]

```
let initialNodes : Set(UML::Element) = src.ownedElement->select(e | e.oclIsKindOf(UML::Initia
let finalNodes : Set(UML::Element) = src.ownedElement->select(e | e.oclIsKindOf(UML::FinalNod
let objectFlowsWithGuard : Set(UML::ObjectFlow) = src.ownedElement->select(e | e.oclIsKindOf(
let objectFlows : Set(UML::ObjectFlow) = src.ownedElement->select(e | e.oclIsKindOf(UML::Obje
let ignoreInterruptibleActivityRegion: Set(UML::InterruptibleActivityRegion) = src.ownedEleme
let elementsFMS : Set(UML::Element) = ((src.ownedElement->select(e | e.oclIsKindOf(UML::Contr
let elementsOMS: Set(UML::Element) = ((((((src.ownedElement-initialNodes)-finalNodes)-objectF
elementsOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(elementsFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(initialNodes->collect(e | InitialNodeMembership_Mapping.getMapped(e)))
->union(finalNodes->collect(e | ActivityFinalNodeMembership_Mapping.getMapped(e)))
->union(objectFlowsWithGuard->collect(e | ObjectFlowGuardFeatureMembership_Mapping.getMapped(
->union(objectFlows->collect(e | ObjectFlowFeatureMembership_Mapping.getMapped(e)))
```

C.2.5.2.3.9 Variable Actions

C.2.5.2.3.9.1 AddVariableValueAction_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping

Mapping Source

AddVariableValueAction

Mapping Target

ActionUsage

Owned Mappings

- addVariableValueActionFeatureTyping : AddVariableValueActionFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ActionUsage::ownedRelationship () : Relationship [0..*]

```
Helper.actionOwnedRelationship(from)->including(addVariableValueActionFeatureTyping.to)
```

C.2.5.2.3.9.2 AddVariableValueActionFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *AddVariableValueAction* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

AddVariableValueAction

Mapping Target

FeatureTyping

Owned Mappings

- addVariableValueAction : AddVariableValueAction_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

  ```
  SYSML2::ActionUsage.allInstances()->any(m | m.qualifiedName = 'Actions::AssignmentAction')
  ```

- FeatureTyping::typedFeature () : Feature [1]

  ```
  addVariableValueAction.to
  ```

C.2.5.2.3.9.3 ClearVariableAction_Mapping

Description

The expected SysML v2 textual notation of a SysMLv1::ClearVariableAction is as follows

```
action thisIsAClearVariableAction {
   thisIsAVariable = null;
}
```

General Mappings

CommonAction_Mapping

Mapping Source

ClearVariableAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ActionUsage::ownedRelationship () : Relationship [0..*]

  ```
  Helper.actionOwnedRelationship(from)->including(ClearVariableActionFeatureMembership_Mapping
  ```

C.2.5.2.3.9.4 ClearVariableActionFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *ClearVariableAction* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ClearVariableAction

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

  ```
  ClearVariableActionReferenceUsage_Mapping.getMapped(from)
  ```

C.2.5.2.3.9.5 ClearVariableActionReferenceUsage_Mapping

Description

Creates a reference usage for the *ClearVariableAction* mapping.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

ClearVariableAction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

  ```
  Set{ClearVariableActionReferenceUsageFeatureValue_Mapping.getMapped(from)}
  ```

- ReferenceUsage::name () : String [0..1]

  ```
  from.variable.name
  ```

C.2.5.2.3.9.6 ClearVariableActionReferenceUsageFeatureValue_Mapping

Description

Creates a feature value relationship for the mapping class *ClearVariableAction*

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

ClearVariableAction

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

```
Null_Mapping.getMapped(from)
```

C.2.5.2.3.9.7 Null_Mapping

Description

*** not specified yet ***

General Mappings

CommonValueSpecification_Mapping

Mapping Source

Element

Mapping Target

NullExpression

Owned Mappings

(none)

C.2.5.2.3.9.8 ReadVariableAction_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping

Mapping Source

ReadVariableAction

Mapping Target

ActionUsage

Owned Mappings

(none)

C.2.5.2.3.9.9 RemoveVariableValueAction_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping

Mapping Source

RemoveVariableValueAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ActionUsage::ownedRelationship () : Relationship [0..*]

```
let relationships : Set(KerML::Relationship) = Helper.actionOwnedRelationship(from)
->including(RemoveVariableValueActionExpressionMembership_Mapping.getMapped(from))
->including(EmptyReturnParameterFeatureMembership_Mapping.getMapped(from))
->including(RemoveVariableValueActionAssignmentActionMembership_Mapping.getMapped(from)) in
let relationshipsWithRemoveAt : Set(KerML::Relationship) = if from.removeAt.oclIsUndefined()
if from.value.oclIsUndefined() then relationshipsWithRemoveAt else relationshipsWithRemoveAt-
```

C.2.5.2.3.9.10 RemoveVariableValueActionAssignmentAction_Mapping

Description

*** not specified yet ***

General Mappings

GenericToAssignmentActionUsage_Mapping

Mapping Source

RemoveVariableValueAction

Mapping Target

AssignmentActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- AssignmentActionUsage::ownedRelationship () : Relationship [0..*]

```
Set{RemoveVariableValueActionAssignmentActionParameterMembership_Mapping.getMapped(from), Re
```

C.2.5.2.3.9.11 RemoveVariableValueActionAssignmentActionMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *RemoveVariableValueAction* mapping.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

RemoveVariableValueAction

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

```
RemoveVariableValueActionAssignmentAction_Mapping.getMapped(from)
```

C.2.5.2.3.9.12 RemoveVariableValueActionAssignmentActionParameter_Mapping

Description

*** not specified yet ***

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

RemoveVariableValueAction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

  ```
  Set{RemoveVariableValueActionAssignmentActionParameterFeatureMembership_Mapping.getMapped(fr
  ```

- ReferenceUsage::direction () : FeatureDirectionKind [0..1]

  ```
  KerML::FeatureDirectionKind::_'in'
  ```

C.2.5.2.3.9.13 RemoveVariableValueActionAssignmentActionParameterFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *RemoveVariableValueAction* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

RemoveVariableValueAction

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
RemoveVariableValueActionAssignmentActionParameterReference_Mapping.getMapped(from)
```

C.2.5.2.3.9.14 RemoveVariableValueActionAssignmentActionParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *RemoveVariableValueAction* mapping.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

RemoveVariableValueAction

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ParameterMembership::ownedMemberParameter () : Feature [1]

```
RemoveVariableValueActionAssignmentActionParameter_Mapping.getMapped(from)
```

C.2.5.2.3.9.15 RemoveVariableValueActionAssignmentActionParameterReference_Mapping

Description

*** not specified yet ***

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

RemoveVariableValueAction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
Set{RemoveVariableValueActionAssignmentActionParameterReferenceFeatureMembership_Mapping.get
```

C.2.5.2.3.9.16 RemoveVariableValueActionAssignmentActionParameterReferenceFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *RemoveVariableValueAction* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

RemoveVariableValueAction

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
RemoveVariableValueActionAssignmentActionParameterReferenceReference_Mapping.getMapped(from)
```

C.2.5.2.3.9.17 RemoveVariableValueActionAssignmentActionParameterReferenceReference_Mapping

Description

*** not specified yet ***

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

RemoveVariableValueAction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

C.2.5.2.3.9.18 RemoveVariableValueActionAssignmentActionSecondParameter_Mapping

Description

*** not specified yet ***

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

RemoveVariableValueAction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

C.2.5.2.3.9.19 RemoveVariableValueActionAssignmentActionSecondParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *RemoveVariableValueAction* mapping.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

RemoveVariableValueAction

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ParameterMembership::ownedMemberParameter () : Feature [1]

```
RemoveVariableValueActionAssignmentActionSecondParameter_Mapping.getMapped(from)
```

C.2.5.2.3.9.20 RemoveVariableValueActionExpressionMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *RemoveVariableValueAction* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

RemoveVariableValueAction

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
RemoveVariableValueActionExpressionReferenceUsage_Mapping.getMapped(from)
```

C.2.5.2.3.9.21 RemoveVariableValueActionExpressionParameter_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeature_Mapping

Mapping Source

Pin

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::ownedRelationship () : Relationship [0..*]

```
Set{RemoveVariableValueActionExpressionParameterValue_Mapping.getMapped(from)}
```

C.2.5.2.3.9.22 RemoveVariableValueActionExpressionParameterFeatureReference_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeatureReferenceExpression_Mapping

Mapping Source

Pin

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureReferenceExpression::ownedRelationship () : Relationship [0..*]

```
Set{RemoveVariableValueActionExpressionParameterFeatureReferenceMembership_Mapping.getMappe
```

C.2.5.2.3.9.23 RemoveVariableValueActionExpressionParameterFeatureReferenceMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Pin* mapping.

General Mappings

GenericToMembership_Mapping

Mapping Source

Pin

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Membership::memberElement () : Element [1]

```
from
```

C.2.5.2.3.9.24 RemoveVariableValueActionExpressionParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Pin* mapping.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

Pin

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ParameterMembership::ownedMemberParameter () : Feature [1]

  ```
  RemoveVariableValueActionExpressionParameter_Mapping.getMapped(from)
  ```

C.2.5.2.3.9.25 RemoveVariableValueActionExpressionParameterValue_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Pin

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

  ```
  RemoveVariableValueActionExpressionParameterFeatureReference_Mapping.getMapped(from)
  ```

C.2.5.2.3.9.26 RemoveVariableValueActionExpressionReferenceUsage_Mapping

Description

Creates a reference usage for the *RemoveVariableValueAction* mapping.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

RemoveVariableValueAction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::name () : String [0..1]

```
from.variable.name
```

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
Set{RemoveVariableValueActionExpressionReferenceUsageFeatureValue_Mapping.getMapped(from)}
```

C.2.5.2.3.9.27 RemoveVariableValueActionExpressionReferenceUsageFeatureValue_Mapping

Description

Creates a feature value relationship for the mapping class *RemoveVariableValueAction*

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

RemoveVariableValueAction

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

```
RemoveVariableValueActionInvocationExpression_Mapping.getMapped(from)
```

C.2.5.2.3.9.28 RemoveVariableValueActionInvocationExpression_Mapping

Description

*** not specified yet ***

General Mappings

GenericToInvocationExpression_Mapping

Mapping Source

RemoveVariableValueAction

Mapping Target

InvocationExpression

Owned Mappings

- removeVariableValueActionInvocationExpressionFeatureTyping : RemoveVariableValueActionInvocationExpressionFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- InvocationExpression::ownedRelationship () : Relationship [0..*]

```
Set{removeVariableValueActionInvocationExpressionFeatureTyping.to}
```

C.2.5.2.3.9.29 RemoveVariableValueActionInvocationExpressionFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *RemoveVariableValueAction* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

RemoveVariableValueAction

Mapping Target

FeatureTyping

Owned Mappings

- removeVariableValueActionInvocationExpression : RemoveVariableValueActionInvocationExpression_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
KerML::Function.allInstances()->any(m | m.qualifiedName = 'SequenceFunctions::excluding')
```

- FeatureTyping::typedFeature () : Feature [1]

```
removeVariableValueActionInvocationExpression.to
```

C.2.5.3 Activities

C.2.5.3.1 Overview

Table 17. List of all Overview Mapping Specfications

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter

Activity	VerificationCaseDefinition
	ActionUsage
	ActionDefinition
		TestCaseActivity_Mapping
		ActivityAsUsage_Mapping
		ActivityAsDefinition_Mapping
			Helper.hasStereotypeApplied(from, 'SysML::Requirements::TestCase')
			(not from.owner.oclIsKindOf(UML::Package)) and (not Helper.hasStereotypeApplied(from, 'SysML::Requirements::TestCase'))
			from.owner.oclIsKindOf(UML::Package)

ActivityFinalNode			
ActivityParameterNode			
ActivityPartition			
CentralBufferNode	ActionUsage ActionUsage	DataStoreNode_Mapping CentralBufferNode_Mapping	
ControlFlow	TransitionUsage SuccessionAsUsage	ControlFlowTransitionUsage_Mapping ControlFlowSuccessionAsUsage_Mapping	not from.guard.oclIsUndefined() from.guard.oclIsUndefined()
DataStoreNode	ActionUsage	DataStoreNode_Mapping	
DecisionNode	DecisionNode	DecisionNode_Mapping	
ExceptionHandler			

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
FlowFinalNode			
ForkNode	ForkNode	ForkNode_Mapping	
InitialNode			
InterruptibleActivityRegion			
JoinNode	JoinNode	JoinNode_Mapping	
MergeNode	MergeNode	MergeNode_Mapping	
ObjectFlow	TransitionUsage SuccessionFlowConnectionUsage	ObjectFlowGuard_Mapping ObjectFlow_Mapping	not from.guard.oclIsUndefined() from.guard.oclIsUndefined()
Variable	AttributeUsage ItemUsage	VariableAttribute_Mapping VariableItem_Mapping	from.type.oclIsKindOf(UML::DataType) not from.type.oclIsKindOf(UML::DataType)

C.2.5.3.2 SysML v1 Activities elements not mapped

Table 18. List of SysML v1 elements not mapped of this section

SysML v1 Concept	Rationale
ActivityParameterNode	The parameter of the activity is mapped from SysML v1 to SysML v2. The additional concept of the activity parameter node is necessary for the token semantic of SysML v1 activities, which is not part of SysML v2. Therefore, the additional concept of the activity parameter node is not mapped to SysML v2.

C.2.5.3.3 Mapping Specifications

C.2.5.3.3.1 ActivityAsDefinition_Mapping

Description

A UML4SysML::Activity is mapped to a SysMLv2::ActionDefinition if the owner of the activity is a UML4SysML::Package. The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action def SysMLv1Activity {
  in parIn : SysMLv1Block;
  out parOut;
  out parReturn;
}
```

General Mappings

CommonActivity_Mapping

Mapping Source

Activity

Mapping Target

ActionDefinition

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
from.owner.oclIsKindOf(UML::Package)
```

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

C.2.5.3.3.2 ActivityAsUsage_Mapping

Description

A UML4SysML::Activity is mapped to a SysMLv2::ActionUsage if the owner of the activity is not a UML4SysML::Package. To follow the informal naming convention that usage elements start with a lowercase letter, the first letter of the activity's name is converted to a lowercase letter. The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
part def SysMLv1Block {
  action sysMLv1Activity {
    in parIn : SysMLv1Enumeration;
    out parOut : ScalarValues::Integer;
  }
}
enum def SysMLv1Enumeration;
```

General Mappings

CommonActivity_Mapping

Mapping Source

Activity

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
(not from.owner.oclIsKindOf(UML::Package)) and  (not Helper.hasStereotypeApplied(from, 'SysML::Requi
```

Mapping rules

The following lists the mapping rules for the target element properties.

- ActionUsage::name () : String [0..1]

```
if from.name.size() > 1 then from.name.substring(1,1).toLowerCase().concat(from.name.substri
```

C.2.5.3.3.3 ActivityEdgeMetadata_Mapping

Description

Adds metadata to the transformation target elements of UML::ControlFlow and UML::ObjectFlow to map the UML::ActivityEdge::weight property which has no direct target in SysML v2.

General Mappings

GenericToMetadataUsage_Mapping

Mapping Source

ActivityEdge

Mapping Target

MetadataUsage

Owned Mappings

- activityEdgeMetadataFeatureTyping : ActivityEdgeMetadataFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- MetadataUsage::ownedRelationship () : Relationship [0..*]

```
Set{activityEdgeMetadataFeatureTyping.to, ActivityEdgeMetadataFeatureMembership_Mapping.getM
```

- MetadataUsage::name () : String [0..1]

```
'weight'
```

C.2.5.3.3.4 ActivityEdgeMetadataFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *ActivityEdge* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ActivityEdge

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
ActivityEdgeMetadataReferenceUsage_Mapping.getMapped(from)
```

C.2.5.3.3.5 ActivityEdgeMetadataFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *ActivityEdge* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

ActivityEdge

Mapping Target

FeatureTyping

Owned Mappings

- activityEdgeMetadata : ActivityEdgeMetadata_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

```
activityEdgeMetadata.to
```

- FeatureTyping::type () : Type [1]

```
SYSML2::MetadataDefinition.allInstances()->any(m | m.qualifiedName = 'SysMLv1Library::Activi
```

C.2.5.3.3.6 ActivityEdgeMetadataFeatureValue_Mapping

Description

Creates a feature value relationship for the mapping class *ActivityEdge*

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

ActivityEdge

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

```
Helper.mappedValueSpecification(from.weight)
```

C.2.5.3.3.7 ActivityEdgeMetadataOwningMembership_Mapping

Description

Creates a owning membership relationship for *ownedMemberElement()* for the *ActivityEdge* mapping.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

ActivityEdge

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

```
ActivityEdgeMetadata_Mapping.getMapped(from)
```

C.2.5.3.3.8 ActivityEdgeMetadataRedefinition_Mapping

Description

Creates a redefinition relationship for the *redefiningFeature()* and the *redefinedFeature()* for the *ActivityEdge* mapping.

General Mappings

GenericToRedefinition_Mapping

Mapping Source

ActivityEdge

Mapping Target

Redefinition

Owned Mappings

- activityEdgeMetadataReferenceUsage : ActivityEdgeMetadataReferenceUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Redefinition::redefiningFeature () : Feature [1]

```
activityEdgeMetadataReferenceUsage.to
```

- Redefinition::redefinedFeature () : Feature [1]

```
SYSML2::AttributeUsage.allInstances()->any(m | m.qualifiedName = 'SysMLv1Library::ActivityEd
```

C.2.5.3.3.9 ActivityEdgeMetadataReferenceUsage_Mapping

Description

Creates a reference usage for the *ActivityEdge* mapping.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

ActivityEdge

Mapping Target

ReferenceUsage

Owned Mappings

- activityEdgeMetadataRedefinition : ActivityEdgeMetadataRedefinition_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
Set{activityEdgeMetadataRedefinition.to, ActivityEdgeMetadataFeatureValue_Mapping.getMapped(
```

C.2.5.3.3.10 ActivityEdgeSourceEndFeature_Mapping

Description

Creates a SysML v2 feature for the source activity node of the SysML v1 activity edge which subsets the SysML v2 target element of the source activity node.

General Mappings

GenericToFeature_Mapping

Mapping Source

ActivityNode

Mapping Target

Feature

Owned Mappings

- activityEdgeSourceEndSubsetting : ActivityEdgeSourceEndSubsetting_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isEnd () : Boolean [1]

  ```
  true
  ```

- Feature::ownedRelationship () : Relationship [0..*]

  ```
  Set{activityEdgeSourceEndSubsetting.to}
  ```

C.2.5.3.3.11 ActivityEdgeInitialNodeSourceEndFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *InitialNode* mapping.

General Mappings

GenericToEndFeatureMembership_Mapping

Mapping Source

InitialNode

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

  ```
  ActivityEdgeSourceInitialNode_Mapping.getMapped(from)
  ```

C.2.5.3.3.12 ActivityEdgeSourceInitialNode_Mapping

Description

The SysMLv1::InitialNode is mapped to a subsetted feature of the SysML v2 Actions::start feature.

General Mappings

GenericToFeature_Mapping

Mapping Source

InitialNode

Mapping Target

Feature

Owned Mappings

- activityEdgeSourceInitialNodeSubsetting : ActivityEdgeSourceInitialNodeSubsetting_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isEnd () : Boolean [1]

```
true
```

- Feature::ownedRelationship () : Relationship [0..*]

```
Set{activityEdgeSourceInitialNodeSubsetting.to}
```

C.2.5.3.3.13 ActivityEdgeSourceEndFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *ActivityNode* mapping.

General Mappings

GenericToEndFeatureMembership_Mapping

Mapping Source

ActivityNode

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
ActivityEdgeSourceEndFeature_Mapping.getMapped(from)
```

C.2.5.3.3.14 ActivityEdgeSourceInitialNodeSubsetting_Mapping

Description

Creates a subsetting relationship for the *subsettingFeature()* and the *subsettedFeature()* for the *InitialNode* mapping.

General Mappings

GenericToSubsetting_Mapping

Mapping Source

InitialNode

Mapping Target

Subsetting

Owned Mappings

- activityEdgeSourceInitialNode : ActivityEdgeSourceInitialNode_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Subsetting::subsettingFeature () : Feature [1]

```
activityEdgeSourceInitialNode.to
```

- Subsetting::subsettedFeature () : Feature [1]

```
SYSML2::ActionUsage.allInstances()->any(m | m.qualifiedName = 'Actions::Action::start')
```

C.2.5.3.3.15 ActivityEdgeSourceEndSubsetting_Mapping

Description

Creates a subsetting relationship for the *subsettingFeature()* and the *subsettedFeature()* for the *ActivityNode* mapping.

General Mappings

GenericToSubsetting_Mapping

Mapping Source

ActivityNode

Mapping Target

Subsetting

Owned Mappings

- activityEdgeSourceEndFeature : ActivityEdgeSourceEndFeature_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Subsetting::subsettedFeature () : Feature [1]

  ```
  from
  ```

- Subsetting::subsettingFeature () : Feature [1]

  ```
  activityEdgeSourceEndFeature.to
  ```

C.2.5.3.3.16 ActivityFinalNodeMembership_Mapping

Description

The mapping class creates a membership relationship to the action usage library element Actions::Action::done.

General Mappings

GenericToMembership_Mapping

Mapping Source

FinalNode

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Membership::memberElement () : Element [1]

```
 SysMLv2::ActionUsage.allInstances()->any(e | e.qualifiedName =  'Actions::Action::done')
```

C.2.5.3.3.17 CommonActivity_Mapping

Description

Abstract mapping class for UML4SysML::Activity. A UML4SysML::Activity is mapped to a SysMLv2::ActionDefinition or SysMLv2::ActionUsage. See specialized mapping classes for the specific mapping rules.

General Mappings

Behavior_Mapping

Mapping Source

Activity

Mapping Target

Behavior

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Behavior::ownedRelationship () : Relationship [0..*]

```
let relationships : Set(KerML::Relationship) = Helper.activityOwnedRelationship(from) in
let parameters : Set(UML::Paramter) = from.ownedElement->select(e | e.oclIsKindOf(UML::Parame
relationships->union(parameters->collect(p | ParameterMembership_Mapping.getMapped(p)))
```

C.2.5.3.3.18 CommonActivityEdgeSuccessionAsUsage_Mapping

Description

The mapping class provides a common mapping of a SysMLv1::ActivityEdge to a SysMLv2::SucessionAsUsage. The mapping used for SysMLv1::ControlFlows and SysMLv2::ObjectFlows.

General Mappings

GenericToConnector_Mapping

Mapping Source

ActivityEdge

Mapping Target

SuccessionAsUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- SuccessionAsUsage::ownedRelationship () : Relationship [0..*]

```
let relationships : Set(KerML::Relationship) = Set{
if from.source.oclIsKindOf(UML::InitialNode) then ActivityEdgeInitialNodeSourceEndFeatureMemb
if from.oclIsKindOf(UML::ObjectFlow) then ObjectFlowGuardSuccessionTargetEndFeatureMembership
if from.target.oclIsKindOf(UML::FinalNode) then ControlFlowFinalNodeTargetEndFeatureMembershi
if src.guard.oclIsUndefined() then relationships else relationships->including(ElementFeature
```

C.2.5.3.3.19 CommonVariable_Mapping

Description

Abstract mapping class for UML4SysML::Variable which is defined in the context of UML4SysML::Activity. A UML4SysML::Variable is mapped to a SysMLv2::AttributeUsage or SysMLv2::ItemUsage. See specialized mapping classes for the specific mapping rules.

General Mappings

PropertyCommon_Mapping

Mapping Source

Variable

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isDerived () : Boolean [1]

```
false
```

- Feature::isEnd () : Boolean [1]

```
false
```

- Feature::ownedRelationship () : Relationship [0..*]

```
let typing: KerML::FeatureTyping = VariableFeatureTyping_Mapping.getMapped(from) in
if typing.oclIsUndefined() then
    Set{MultiplicityMembership_Mapping.getMapped(from)}
else
    Set{MultiplicityMembership_Mapping.getMapped(from), typing}
endif
```

- Feature::isComposite () : Boolean [1]

```
false
```

C.2.5.3.3.20 ControlFlowTransitionUsage_Mapping

Description

A UML4SysML::ControlFlow with a guard condition is mapped to a SysMLv2::TransitionUsage. The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
succession controlFlowName first action1 if guardCondition.result then action2 {
  calc guardCondition {
    return : ScalarValues::Boolean;
    language "English"
    /*
     * thisIsAGuard
     */
  }
}
```

General Mappings

GenericToUsage_Mapping
NamedElementMain_Mapping

Mapping Source

ControlFlow

Mapping Target

TransitionUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
not from.guard.oclIsUndefined()
```

Mapping rules

The following lists the mapping rules for the target element properties.

- TransitionUsage::ownedRelationship () : Relationship [0..*]

```
let relationships : Set(KerML::Relationship) =
Set{ActivityEdgeTransitionUsageSourceMembership_Mapping.getMapped(from.source)}
->including(CommonParameterReferenceUsageInMembership_Mapping.getMapped(from.source))
->including(ControlFlowTransitionUsageFeatureMembership_Mapping.getMapped(from))
->including(CommonActivityEdgeSuccessionAsUsage_Mapping.getMapped(from))
->including(CommonReturnParameterReferenceUsageMembership_Mapping.getMapped(from)) in
let relationshipsWithGuard : Set(KerML::Relationship) = if from.guard.oclIsTypeOf(UML::Opaque
if from.weight.oclIsUndefined() then relationshipsWithGuard else relationshipsWithGuard->incl
```

- TransitionUsage::isComposite () : Boolean [1]

```
 true
```

C.2.5.3.3.21 CentralBufferNode_Mapping

Description

The mapping of the SysMLv1::CentralBufferNode is not defined in detail yet. It will be an action usage which contains the behavior of a central buffer node.

General Mappings

GenericToActionUsage_Mapping
NamedElementMain_Mapping

Mapping Source

CentralBufferNode

Mapping Target

ActionUsage

Owned Mappings

(none)

C.2.5.3.3.22 ControlFlowFinalNodeTargetEndFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *ActivityNode* mapping.

General Mappings

GenericToEndFeatureMembership_Mapping

Mapping Source

ActivityNode

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

  ```
  ControlFlowTargetFinalNode_Mapping.getMapped(from)
  ```

C.2.5.3.3.23 ControlFlowTargetFinalNodeSubsetting_Mapping

Description

Creates a subsetting relationship for the *subsettingFeature()* and the *subsettedFeature()* for the *FinalNode* mapping.

General Mappings

GenericToSubsetting_Mapping

Mapping Source

FinalNode

Mapping Target

Subsetting

Owned Mappings

- controlFlowTargetFinalNode : ControlFlowTargetFinalNode_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Subsetting::subsettingFeature () : Feature [1]

```
controlFlowTargetFinalNode.to
```

- Subsetting::subsettedFeature () : Feature [1]

```
SYSML2::ActionUsage.allInstances()->any(m | m.qualifiedName = 'Actions::Action::done')
```

C.2.5.3.3.24 ControlFlowSuccessionAsUsage_Mapping

Description

A UML4SysML::ControlFlow without a guard condition is mapped to a SysMLv2::SuccessionAsUsage. The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
succession controlFlowName first action1 then action2;
```

General Mappings

NamedElementMain_Mapping
CommonActivityEdgeSuccessionAsUsage_Mapping

Mapping Source

ControlFlow

Mapping Target

SuccessionAsUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
from.guard.oclIsUndefined()
```

Mapping rules

The following lists the mapping rules for the target element properties.

- SuccessionAsUsage::ownedRelationship () : Relationship [0..*]

```
let relationships : Set(KerML::Relationship) = Set{
if from.source.oclIsKindOf(UML::InitialNode) then ActivityEdgeInitialNodeSourceEndFeatureMemb
if from.oclIsKindOf(UML::ObjectFlow) then ObjectFlowGuardSuccessionTargetEndFeatureMembership
if from.target.oclIsKindOf(UML::FinalNode) then ControlFlowFinalNodeTargetEndFeatureMembershi
let relationshipsWithGuard : Set(KerML::Relationship) = if src.guard.oclIsUndefined() then re
if from.weight.oclIsUndefined() then relationshipsWithGuard else relationshipsWithGuard->incl
```

C.2.5.3.3.25 ControlFlowTargetFinalNode_Mapping

Description

The mapping class maps a UML4SysML::FinalNode to a Feature which will be subsetted by Actions::Action::done. The subsetting is created by the mapping class ControlFlowTargetFinalNodeSubsetting_Mapping.

General Mappings

GenericToFeature_Mapping

Mapping Source

FinalNode

Mapping Target

Feature

Owned Mappings

- controlFlowTargetFinalNodeSubsetting : ControlFlowTargetFinalNodeSubsetting_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::ownedRelationship () : Relationship [0..*]

```
Set{controlFlowTargetFinalNodeSubsetting.to}
```

- Feature::isEnd () : Boolean [1]

```
true
```

C.2.5.3.3.26 ControlFlowTargetEndFeature_Mapping

Description

The mapping class maps the UML4SysML::ActivityNode to a Feature which is subsetted by the mapping target of the UML4SysML::ActivityNode. The subsetting is created by the mapping class ControlFlowTargetEndSubsetting_Mapping.

General Mappings

GenericToFeature_Mapping

Mapping Source

ActivityNode

Mapping Target

Feature

Owned Mappings

- controlFlowTargetEndSubsetting : ControlFlowTargetEndSubsetting_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::ownedRelationship () : Relationship [0..*]

  ```
  Set{controlFlowTargetEndSubsetting.to}
  ```

- Feature::isEnd () : Boolean [1]

  ```
  true
  ```

C.2.5.3.3.27 ControlFlowTargetEndFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *ActivityNode* mapping.

General Mappings

GenericToEndFeatureMembership_Mapping

Mapping Source

ActivityNode

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

  ```
  ControlFlowTargetEndFeature_Mapping.getMapped(from)
  ```

C.2.5.3.3.28 ControlFlowTargetEndSubsetting_Mapping

Description

Creates a subsetting relationship for the *subsettingFeature()* and the *subsettedFeature()* for the *ActivityNode* mapping.

General Mappings

GenericToSubsetting_Mapping

Mapping Source

ActivityNode

Mapping Target

Subsetting

Owned Mappings

- controlFlowTargetEndFeature : ControlFlowTargetEndFeature_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Subsetting::subsettedFeature () : Feature [1]

  ```
  from
  ```

- Subsetting::subsettingFeature () : Feature [1]

  ```
  controlFlowTargetEndFeature.to
  ```

C.2.5.3.3.29 ControlFlowTransitionUsageFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *ControlFlow* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ControlFlow

Mapping Target

TransitionFeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- TransitionFeatureMembership::kind () : TransitionFeatureKind [1]

```
KerML::TransitionFeatureKind::guard
```

- TransitionFeatureMembership::ownedMemberFeature () : Feature [1]

```
if from.guard.oclIsKindOf(UML::OpaqueExpression) then OpaqueExpressionAsValue_Mapping.getMapp
else from.guard endif
```

C.2.5.3.3.30 ActivityEdgeTransitionUsageSourceMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *ActivityNode* mapping.

General Mappings

GenericToMembership_Mapping

Mapping Source

ActivityNode

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Membership::memberElement () : Element [1]

```
if from.oclIsTypeOf(UML::ActivityParameterNode) then from.parameter else from endif
```

C.2.5.3.3.31 DataStoreNode_Mapping

Description

The mapping of the SysMLv1::DataStoreNode is not defined in detail yet. It will an action usage which contains the behavior of a data store node.

General Mappings

CentralBufferNode_Mapping

Mapping Source

DataStoreNode

Mapping Target

ActionUsage

Owned Mappings

(none)

C.2.5.3.3.32 DecisionNode_Mapping

Description

The SysMLv1::DecisionNode is mapped to a SysMLv2::DecisionNode. There is no suitable element in SysML v2 for the else condition of an outgoing SysMLv1::ActivityEdge. Therefore, it is mapped to a TextualRepresentation with language "SysML v1" and body "else" (see ExpressionElse_Mapping class). The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
decide sysMLv1DecisionNode;
        succession flow1 first sysMLv1DecisionNode if {
                return : ScalarValues::Boolean;
                // guard expression, for example, opaque expression
        }.result then nextAction;
        succession flow2 first sysMLv1DecisionNode if {
                return : ScalarValues::Boolean;
                language "SysMLv1"
                /*
                 * else
                  */
        }.result then nextAction;
```

General Mappings

GenericToUsage_Mapping
NamedElementMain_Mapping

Mapping Source

DecisionNode

Mapping Target

DecisionNode

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- DecisionNode::isComposite () : Boolean [1]

  ```
  true
  ```

C.2.5.3.3.33 ForkNode_Mapping

Description

The SysMLv1::ForkNode is mapped to a SysMLv2::ForkNode. The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action def SysMLv1Activity {
  first start;
  action sysMLv1ActionA;
  then fork sysMLv1ForkNode;
    then sysMLv1ActionB;
    then sysMLv1ActionC;
  action sysMLv1ActionB;
  then sysMLv1JoinNode;
  action sysMLv1ActionC;
  then sysMLv1JoinNode;

  join sysMLv1JoinNode;
  then done;
}
```

General Mappings

GenericToUsage_Mapping
NamedElementMain_Mapping

Mapping Source

ForkNode

Mapping Target

ForkNode

Owned Mappings

(none)

C.2.5.3.3.34 InitialNodeMembership_Mapping

Description

The mapping class creates a membership relationship to the action usage library element Actions::Action::start.

General Mappings

GenericToMembership_Mapping

Mapping Source

InitialNode

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Membership::memberName () : String [0..1]

  ```
  if from.name = '' then null else from.name endif
  ```

- Membership::memberElement () : Element [1]

  ```
  SysMLv2::ActionUsage.allInstances()->any(e | e.qualifiedName =  'Actions::Action::start')
  ```

C.2.5.3.3.35 JoinNode_Mapping

Description

The SysMLv1::JoinNode is mapped to a SysMLv2::JoinNode. The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action def SysMLv1Activity {
  first start;
  action sysMLv1ActionA;
  then fork sysMLv1ForkNode;
    then sysMLv1ActionB;
    then sysMLv1ActionC;
  action sysMLv1ActionB;
  then sysMLv1JoinNode;
  action sysMLv1ActionC;
  then sysMLv1JoinNode;

  join sysMLv1JoinNode;
```

```
  then done;
}
```

General Mappings

GenericToUsage_Mapping
NamedElementMain_Mapping

Mapping Source

JoinNode

Mapping Target

JoinNode

Owned Mappings

(none)

C.2.5.3.3.36 MergeNode_Mapping

Description

The SysMLv1::MergeNode is mapped to a SysMLv2::MergeNode. The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
tbd
```

General Mappings

GenericToUsage_Mapping
NamedElementMain_Mapping

Mapping Source

MergeNode

Mapping Target

MergeNode

Owned Mappings

(none)

C.2.5.3.3.37 ObjectFlow_Mapping

Description

A UML4SysML::ObjectFlowFlow without a guard condition is mapped to a SysMLv2::SuccessionFlowConnectionUsage. The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
succession flow objectFlowName of ScalarValues::String from action1.outputValue to action2.inputValu
```

General Mappings

GenericToConnector_Mapping
NamedElementMain_Mapping

Mapping Source

ObjectFlow

Mapping Target

SuccessionFlowConnectionUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
from.guard.oclIsUndefined()
```

Mapping rules

The following lists the mapping rules for the target element properties.

- SuccessionFlowConnectionUsage::ownedRelationship () : Relationship [0..*]

  ```
  let relationships : Set(KerML::Relationship) = if from.source.oclIsKindOf(UML::ObjectNode) th
  Set{ObjectFlowItemFeatureMembership_Mapping.getMapped(from), ObjectFlowEndFeatureMembership_M
  else Set{ObjectFlowEndFeatureMembership_Mapping.getMapped(from.source), ObjectFlowEndFeatureM
  if from.weight.oclIsUndefined() then relationships else relationships->including(ActivityEdge
  ```

C.2.5.3.3.38 ObjectFlowFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *ObjectFlow* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ObjectFlow

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
ObjectFlow_Mapping.getMapped(from)
```

C.2.5.3.3.39 ObjectFlowGuardFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *ObjectFlow* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ObjectFlow

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
ObjectFlowGuard_Mapping.getMapped(from)
```

C.2.5.3.3.40 ObjectFlowGuard_Mapping

Description

A UML4SysML::ObjectFlowFlow with a guard condition is mapped to a combined SysMLv2::TransitionUsage and SysMLv2::SuccessionFlowConnectionUsage. The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
first action1 if guardCondition.result then objectFlowName {
  calc guardCondition {
    return : ScalarValues::Boolean;
    language "English"
    /*
     * guard says ok
     */
  }
}
succession flow objectFlowName of AClassifier from action1.outputValue to action2.inputValue;
```

General Mappings

GenericToUsage_Mapping
NamedElementMain_Mapping

Mapping Source

ObjectFlow

Mapping Target

TransitionUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
not from.guard.oclIsUndefined()
```

Mapping rules

The following lists the mapping rules for the target element properties.

- TransitionUsage::ownedRelationship () : Relationship [0..*]

  ```
   Set{
  ActivityEdgeTransitionUsageSourceMembership_Mapping.getMapped(from.source),
  CommonParameterReferenceUsageInMembership_Mapping.getMapped(from.source),
  ObjectFlowTransitionUsageFeatureMembership_Mapping.getMapped(from),
  ObjectFlowGuardSuccessionTargetEndFeatureMembership_Mapping.getMapped(from),
  CommonActivityEdgeSuccessionAsUsage_Mapping.getMapped(from),
  CommonReturnParameterReferenceUsageMembership_Mapping.getMapped(from)
  }
  ```

C.2.5.3.3.41 ObjectFlowGuardSuccessionTargetEndFeature_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeature_Mapping

Mapping Source

ObjectFlow

Mapping Target

Feature

Owned Mappings

- objectFlowGuardSuccessionTargetEndSubsetting : ObjectFlowGuardSuccessionTargetEndSubsetting_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isEnd () : Boolean [1]

  ```
  true
  ```

- Feature::ownedRelationship () : Relationship [0..*]

  ```
  Set{objectFlowGuardSuccessionTargetEndSubsetting.to}
  ```

C.2.5.3.3.42 ObjectFlowGuardSuccessionTargetEndFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *ObjectFlow* mapping.

General Mappings

GenericToEndFeatureMembership_Mapping

Mapping Source

ObjectFlow

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

  ```
  ObjectFlowGuardSuccessionTargetEndFeature_Mapping.getMapped(from)
  ```

C.2.5.3.3.43 ObjectFlowGuardSuccessionTargetEndSubsetting_Mapping

Description

Creates a subsetting relationship for the *subsettingFeature()* and the *subsettedFeature()* for the *ObjectFlow* mapping.

General Mappings

GenericToSubsetting_Mapping

Mapping Source

ObjectFlow

Mapping Target

Subsetting

Owned Mappings

- objectFlowGuardSuccessionTargetEndFeature : ObjectFlowGuardSuccessionTargetEndFeature_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Subsetting::subsettingFeature () : Feature [1]

  ```
  objectFlowGuardSuccessionTargetEndFeature.to
  ```

- Subsetting::subsettedFeature () : Feature [1]

  ```
  ObjectFlow_Mapping.getMapped(from)
  ```

C.2.5.3.3.44 ObjectFlowItemFeature_Mapping

Description

The mapping class maps the source UML4SysML::ObjectNode to a ItemFeature which is typed by the UML4SysML::ObjectNode type.

General Mappings

ObjectFlowItemFeatureUntyped_Mapping

Mapping Source

ObjectNode

Mapping Target

ItemFeature

Owned Mappings

- objectFlowItemFeatureTyping : ObjectFlowItemFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ItemFeature::ownedRelationship () : Relationship [0..*]

```
Set{objectFlowItemFeatureTyping.to}
```

C.2.5.3.3.45 ObjectFlowItemFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *ObjectFlow* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ObjectFlow

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
        if from.source.type.oclIsUndefined() then ObjectFlowItemFeatureUntyped_Mapping.getMapped(fro
```

C.2.5.3.3.46 ObjectFlowItemFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *ObjectNode* mapping.

General Mappings

TypedElementToFeatureTyping_Mapping

Mapping Source

ObjectNode

Mapping Target

FeatureTyping

Owned Mappings

- objectFlowItemFeature : ObjectFlowItemFeature_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

```
objectFlowItemFeature.to
```

C.2.5.3.3.47 ObjectFlowItemFeatureUntyped_Mapping

Description

The mapping class maps the source UML4SysML::ObjectNode to a ItemFeature without a type.

General Mappings

GenericToFeature_Mapping

Mapping Source

ObjectNode

Mapping Target

ItemFeature

Owned Mappings

(none)

C.2.5.3.3.48 ObjectFlowEndFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *ActivityNode* mapping.

General Mappings

GenericToEndFeatureMembership_Mapping

Mapping Source

ActivityNode

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
ObjectFlowItemFlowEnd_Mapping.getMapped(from)
```

C.2.5.3.3.49 ObjectFlowItemFlowEnd_Mapping

Description

The mapping class maps a UML4SysML::ActivityNode to a ItemFlowEnd which is subsetted by the transformation target of the UML4SysML::ActivityNode.

General Mappings

GenericToFeature_Mapping

Mapping Source

ActivityNode

Mapping Target

ItemFlowEnd

Owned Mappings

- objectFlowItemFlowSubsetting : ObjectFlowItemFlowSubsetting_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ItemFlowEnd::ownedRelationship () : Relationship [0..*]

```
Set{objectFlowItemFlowSubsetting.to, ObjectFlowItemFlowFeatureMembership_Mapping.getMapped(f
```

C.2.5.3.3.50 ObjectFlowItemFlowFeature_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeature_Mapping

Mapping Source

ActivityNode

Mapping Target

ItemFeature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ItemFeature::ownedRelationship () : Relationship [0..*]

```
Set{ObjectFlowItemFlowRedefinition_Mapping.getMapped(from)}
```

C.2.5.3.3.51 ObjectFlowItemFlowFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *ActivityNode* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ActivityNode

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
ObjectFlowItemFlowFeature_Mapping.getMapped(from)
```

C.2.5.3.3.52 ObjectFlowItemFlowRedefinition_Mapping

Description

Creates a redefinition relationship for the *redefiningFeature()* and the *redefinedFeature()* for the *ActivityNode* mapping.

General Mappings

GenericToRedefinition_Mapping

Mapping Source

ActivityNode

Mapping Target

Redefinition

Owned Mappings

(none)

C.2.5.3.3.53 ObjectFlowItemFlowSubsetting_Mapping

Description

Creates a subsetting relationship for the *subsettingFeature()* and the *subsettedFeature()* for the *ActivityNode* mapping.

General Mappings

GenericToSubsetting_Mapping

Mapping Source

ActivityNode

Mapping Target

Subsetting

Owned Mappings

- objectFlowItemFlowEnd : ObjectFlowItemFlowEnd_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Subsetting::subsettingFeature () : Feature [1]

```
objectFlowSourceItemFlowEnd.to
```

- Subsetting::subsettedFeature () : Feature [1]

```
if from.oclIsKindOf(UML::ActivityParameterNode) then Parameter_Mapping.getMapped(from.paramet
else if from.oclIsKindOf(UML::Pin) then CommonAction_Mapping.getMapped(from.owner)
else if from.oclIsKindOf(UML::InitialNode) then SysMLv2::ActionUsage.allInstances()->any(e |
else if from.oclIsKindOf(UML::FinalNode) then SysMLv2::ActionUsage.allInstances()->any(e | e.
else from endif endif endif endif
```

C.2.5.3.3.54 ObjectFlowTransitionUsageFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *ObjectFlow* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ObjectFlow

Mapping Target

TransitionFeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- TransitionFeatureMembership::kind () : TransitionFeatureKind [1]

  ```
  KerML::TransitionFeatureKind::guard
  ```

- TransitionFeatureMembership::ownedMemberFeature () : Feature [1]

  ```
  if from.guard.oclIsKindOf(UML::OpaqueExpression) then OpaqueExpressionAsValue_Mapping.getMapp
  else from.guard endif
  ```

C.2.5.3.3.55 VariableAttribute_Mapping

Description

A UML4SysML::Variable is mapped to a SysMLv2::AttributeUsage if the type of the variable is of kind UML4SysML::DataType. The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action def SysMLv1Activity {
  private attribute sysmlv1Variable : ScalarValues::Integer;
}
```

General Mappings

NamedElementMain_Mapping
CommonVariable_Mapping

Mapping Source

Variable

Mapping Target

AttributeUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
from.type.oclIsKindOf(UML::DataType)
```

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

C.2.5.3.3.56 VariableFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *Variable* mapping.

General Mappings

TypedElementToFeatureTyping_Mapping

Mapping Source

Variable

Mapping Target

FeatureTyping

Owned Mappings

(none)

C.2.5.3.3.57 VariableItem_Mapping

Description

A UML4SysML::Variable is mapped to a SysMLv2::ItemUsage if the type of the variable is not of kind UML4SysML::DataType. The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action def SysMLv1Activity {
  private item sysmlv1Variable : SysMLv1Block;
}
part def SysMLv1Block;
```

General Mappings

NamedElementMain_Mapping
CommonVariable_Mapping

Mapping Source

Variable

Mapping Target

ItemUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
not from.type.oclIsKindOf(UML::DataType)
```

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

C.2.5.3.3.58 VariableMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Variable* mapping.

General Mappings

ElementFeatureMembership_Mapping

Mapping Source

Variable

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::visibility () : VisibilityKind [1]

  ```
  KerML::VisibilityKind::private
  ```

C.2.5.4 Classification

C.2.5.4.1 Overview

Table 19. List of all Overview Mapping Specfications

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
Generalization	Subclassification	Generalization_Mapping	
GeneralizationSet			

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter

InstanceSpecification	PartUsage
	ConnectionUsage
	EnumerationUsage
		InstanceSpecification_Mapping
		InstanceSpecificationLink_Mapping
		EnumerationLiteral_Mapping
			from.classifier->select(c | c.oclIsTypeOf(UML::Association))->size() = 0
			from.classifier->select(c | c.oclIsTypeOf(UML::Association))->size() > 0
			from.classifier->select(c | c.oclIsTypeOf(UML::Association))->size() = 0

InstanceValue	FeatureReferenceExpression	InstanceValue_Mapping	
Operation	PerformActionUsage PerformActionUsage	Operation_Mapping OperationDirectedFeature_Mapping	Helper.hasStereotypeApplied(src, 'SysML::Ports&Flows::DirectedFeature')
Parameter	ReferenceUsage	Parameter_Mapping	
ParameterSet			

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter

Property	PartUsage
	PortUsage
	ReferenceUsage
	PortUsage
	AttributeUsage
	Feature
	PartUsage
	AttributeUsage
	Feature
	ItemUsage
	PartUsage
	AttributeUsage
		Part_Mapping
		PortUntyped_Mapping
		PropertyUntyped_Mapping
		Port_Mapping
		VariableAttribute_Mapping
		OwnedEnd_Mapping
		FullPortUntyped_Mapping
		ConstraintParameter_Mapping
		Property_Mapping
		VariableItem_Mapping
		FullPort_Mapping
		Attribute_Mapping
			if src.oclIsKindOf(UML::Property) then let p: UML::Property = src.oclAsType(UML::Property) in not p.type.oclIsUndefined() and Helper.hasStereotypeApplied(p.type, 'SysML::Blocks::Block') and (p.association.oclIsUndefined() or p.association.ownedEnd->excludes(p)) and p.aggregation = UML::AggregationKind::composite else false endif
			from.type.oclIsUndefined()
			from.type.oclIsUndefined()
			if src.oclIsKindOf(UML::Property) then let p: UML::Property = src.oclAsType(UML::Property) in if p.type.oclIsUndefined() then false else not p.type.oclIsKindOf(UML::DataType) and not (p.name.indexOf('base_') > 0) and (p.association.oclIsUndefined() or p.association.ownedEnd->excludes(p)) endif else false endif
			from.type.oclIsKindOf(UML::DataType)
			let p: UML::Property = src.oclAsType(UML::Property) in not p.oclIsUndefined() and (not p.association.oclIsUndefined() and p.association.ownedEnd->includes(p)) and (not p.association.memberEnd->select(m | (not m.type.oclIsUndefined()) and m.type.oclIsTypeOf(UML::UseCase))-

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
			>notEmpty()) from.type.oclIsUndefined() and Helper.hasStereotypeApplied(src, 'SysML::Ports&Flows::FullPort') Helper.hasStereotypeApplied(src.owner, 'SysML::ConstraintBlocks::ConstraintBlock') if src.oclIsKindOf(UML::Property) then let p: UML::Property = src.oclAsType(UML::Property) in if p.type.oclIsUndefined() then false else not p.type.oclIsKindOf(UML::DataType) and not (p.name.indexOf('base_') > 0) and (p.association.oclIsUndefined() or p.association.ownedEnd->excludes(p)) endif else false endif not from.type.oclIsKindOf(UML::DataType) (not from.type.oclIsUndefined()) and Helper.hasStereotypeApplied(src, 'SysML::Ports&Flows::FullPort') if src.oclIsKindOf(UML::Property) then let p: UML::Property = src.oclAsType(UML::Property) in if p.type.oclIsUndefined() then false else p.type.oclIsKindOf(UML::DataType) and (p.association.oclIsUndefined() or p.association.ownedEnd->excludes(p)) endif else false endif
Slot	Feature	Slot_Mapping	
Substitution			

C.2.5.4.2 Mapping Specifications

C.2.5.4.2.1 BehavioralFeature_Mapping

Description

*** not specified yet ***

General Mappings

GenericToUsage_Mapping
Namespace_Mapping

Mapping Source

BehavioralFeature

Mapping Target

Usage

Owned Mappings

(none)

C.2.5.4.2.2 Classifier_Mapping

Description

*** not specified yet ***

General Mappings

GenericToClassifier_Mapping
Namespace_Mapping

Mapping Source

Classifier

Mapping Target

Classifier

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Classifier::isAbstract () : Boolean [1]

```
from.isAbstract
```

- Classifier::ownedRelationship () : Relationship [0..*]

```
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.oclIsKindOf(
let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | e.oclIsKindOf(UML::Featur
let toElementOMS: Set(UML::Element) = (from.ownedElement - toElementFMS) - generalizations  i
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))
```

C.2.5.4.2.3 DefaultLowerBound_Mapping

Description

*** not specified yet ***

General Mappings

GenericToExpression_Mapping

Mapping Source

Element

Mapping Target

LiteralInteger

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- LiteralInteger::ownedRelationship () : Relationship [0..*]

```
Set{CommonReturnParameterFeatureMembership_Mapping.getMapped(from)}
```

- LiteralInteger::value () : Integer [1]

```
1
```

C.2.5.4.2.4 DefaultMultiplicityBoundOwnership_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Element

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::isComposite () : Boolean [1]

  ```
  true
  ```

C.2.5.4.2.5 DefaultMultiplicityElement_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeature_Mapping

Mapping Source

Element

Mapping Target

MultiplicityRange

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- MultiplicityRange::isUnique () : Boolean [1]

```
true
```

- MultiplicityRange::ownedRelationship () : Relationship [0..*]

```
OrderedSet{DefaultMultiplicityLowerBoundOwnership_Mapping.getMapped(from), DefaultMultiplici
```

- MultiplicityRange::name () : String [0..1]

```
'defaultMultiplicity'
```

C.2.5.4.2.6 DefaultMultiplicityLowerBoundOwnership_Mapping

Description

*** not specified yet ***

General Mappings

DefaultMultiplicityBoundOwnership_Mapping

Mapping Source

Element

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : MultiplicityRange [1]

```
DefaultLowerBound_Mapping.getMapped(from)
```

C.2.5.4.2.7 DefaultMultiplicityMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Element* mapping.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Element

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

```
DefaultMultiplicityElement_Mapping.getMapped(from)
```

C.2.5.4.2.8 DefaultMultiplicityUpperBoundOwnership_Mapping

Description

*** not specified yet ***

General Mappings

DefaultMultiplicityBoundOwnership_Mapping

Mapping Source

Element

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : MultiplicityRange [1]

```
DefaultUpperBound_Mapping.getMapped(from)
```

C.2.5.4.2.9 DefaultUpperBound_Mapping

Description

*** not specified yet ***

General Mappings

GenericToExpression_Mapping

Mapping Source

Element

Mapping Target

LiteralInteger

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- LiteralInteger::ownedRelationship () : Relationship [0..*]

  ```
  Set{CommonReturnParameterFeatureMembership_Mapping.getMapped(from)}
  ```

- LiteralInteger::value () : Integer [1]

  ```
  1
  ```

C.2.5.4.2.10 DefaultValueMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Property* mapping.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Property

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

  ```
  DefaultValue_Mapping.getMapped(from)
  ```

C.2.5.4.2.11 ElementFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *NamedElement* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

NamedElement

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::visibility () : VisibilityKind [1]

  ```
  Helper.getKerMLVisibilityKind(from.oclAsType(UML::NamedElement).visibility)
  ```

- FeatureMembership::ownedMemberFeature () : Feature [1]

  ```
  NamedElementMain_Mapping.getMapped(from)
  ```

C.2.5.4.2.12 Generalization_Mapping

Description

*** not specified yet ***

General Mappings

GenericToSpecialization_Mapping
ElementMain_Mapping

Mapping Source

Generalization

Mapping Target

Subclassification

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Subclassification::subclassifier () : Classifier [1]

```
Classifier_Mapping.getMapped(from.specific)
```

- Subclassification::superclassifier () : Classifier [1]

```
if from.general.oclIsTypeOf(UML::PrimitiveType) and not (Helper.getScalarValueType(from.gener
    Helper.getScalarValueType(from.general)
else
    Classifier_Mapping.getMapped(from.general)
endif
```

C.2.5.4.2.13 InstanceSpecificationLink_Mapping

Description

*** not specified yet ***

General Mappings

NamedElementMain_Mapping
GenericToConnectionUsage_Mapping

Mapping Source

InstanceSpecification

Mapping Target

ConnectionUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
from.classifier->select( c | c.oclIsTypeOf(UML::Association))->size() > 0
```

Mapping rules

The following lists the mapping rules for the target element properties.

- ConnectionUsage::ownedRelationship () : Relationship [0..*]

```
 ElementOwnership_Mapping.getMappedColl(from.ownedComment)
->union(SlotMembership_Mapping.getMappedColl(from.slot))
->union(from.classifier->collect(g | InstanceSpecificationFeatureTyping_Mapping.getMapped(fro
```

C.2.5.4.2.14 InstanceSpecification_Mapping

Description

*** not specified yet ***

General Mappings

NamedElementMain_Mapping
GenericToPartUsage_Mapping

Mapping Source

InstanceSpecification

Mapping Target

PartUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
from.classifier->select( c | c.oclIsTypeOf(UML::Association))->size() = 0
```

Mapping rules

The following lists the mapping rules for the target element properties.

- PartUsage::ownedFeatureMembership () : FeatureMembership [0..*]

```
from.classifier->collect(c | InstanceSpecificationToGeneralization_Mapping.getMapped(from, c
```

- PartUsage::ownedRelationship () : Relationship [0..*]

```
 SlotMembership_Mapping.getMappedColl(from.slot)
->union(from.classifier->collect(g | InstanceSpecificationFeatureTyping_Mapping.getMapped(fro
```

C.2.5.4.2.15 InstanceSpecificationFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *InstanceSpecification* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

InstanceSpecification

Mapping Target

FeatureTyping with qualifier: classifier:Classifier

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Type [1]

```
InstanceSpecification_Mapping.getMapped(from)
```

- FeatureTyping::type (in classifier : Classifier) : Type [1]

```
Classifier_Mapping.getMapped(classifier)
```

C.2.5.4.2.16 InstanceValue_Mapping

Description

*** not specified yet ***

General Mappings

ValueSpecification_Mapping

Mapping Source

InstanceValue

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureReferenceExpression::ownedRelationship () : Relationship [0..*]

```
 ElementOwnership_Mapping.getMappedColl(from.ownedComment)
->including(InstanceValueInstanceSpecification_Mapping.getMapped(from.instance))
->including(EmptyReturnParameterFeatureMembership_Mapping.getMapped(from))
```

C.2.5.4.2.17 InstanceValueInstanceSpecification_Mapping

Description

*** not specified yet ***

General Mappings

GenericToMembership_Mapping

Mapping Source

InstanceSpecification

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Membership::memberElement () : Element [1]

```
from
```

C.2.5.4.2.18 LowerBoundValueOwnership_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

MultiplicityElement

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [0..1]

```
LiteralInteger_Mapping.getMapped(from.lowerValue)
```

C.2.5.4.2.19 MultiplicityElement_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeature_Mapping

Mapping Source

MultiplicityElement

Mapping Target

MultiplicityRange

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- MultiplicityRange::ownedRelationship () : Relationship [0..*]

  ```
  OrderedSet{MultiplicityLowerBoundOwnership_Mapping.getMapped(from), MultiplicityUpperBoundOw
  ```

- MultiplicityRange::isUnique () : Boolean [1]

  ```
  from.isUnique
  ```

- MultiplicityRange::name () : String [0..1]

  ```
  'multiplicity'
  ```

C.2.5.4.2.20 MultiplicityLowerBoundOwnership_Mapping

Description

*** not specified yet ***

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

MultiplicityElement

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OwningMembership::memberName () : String [0..1]

  ```
  'lowerBound'
  ```

- OwningMembership::ownedMemberElement () : Element [1]

```
        if from.lowerValue.oclIsUndefined() then DefaultLowerBound_Mapping.getMapped(from) else
```

C.2.5.4.2.21 MultiplicityMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *MultiplicityElement* mapping.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

MultiplicityElement

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

```
MultiplicityElement_Mapping.getMapped(from)
```

C.2.5.4.2.22 MultiplicityUpperBoundOwnership_Mapping

Description

*** not specified yet ***

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

MultiplicityElement

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

```
 if from.upperValue.oclIsUndefined() then
    DefaultUpperBound_Mapping.getMapped(from)
else
    Helper.mappedValueSpecification(from.upperValue)
endif
```

- OwningMembership::memberName () : String [0..1]

```
 'upperBound'
```

C.2.5.4.2.23 Operation_Mapping

Description

The expected SysML v2 textual syntax of a mapped UML4SysML::Operation is as follows.

```
part def ThisIsABlock {
  perform action thisIsAnOperation {
    in parIn : ScalarValues::Boolean;
    inout parInOut [0..*] : ScalarValues::String;
    out parOut;
    out result : ScalarValues::Integer;
  }
}
```

General Mappings

BehavioralFeature_Mapping
GenericToActionUsage_Mapping

Mapping Source

Operation

Mapping Target

PerformActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- PerformActionUsage::ownedRelationship () : Relationship [0..*]

```
let parameters: Set(UML::Element) = src.ownedElement->select(e | e.oclIsKindOf(UML::Parameter
let parameterSets: Set(UML::Element) = src.ownedElement->select(e | e.oclIsKindOf(UML::Parame
ElementOwnership_Mapping.getMappedColl(from.ownedComment)
->union(parameters->collect(e | ParameterMembership_Mapping.getMapped(e)))
->union(parameterSets->collect(e | ParameterSetMembership_Mapping.getMapped(e)))
```

C.2.5.4.2.24 Parameter_Mapping

Description

*** not specified yet ***

General Mappings

GenericToReferenceUsage_Mapping
NamedElementMain_Mapping

Mapping Source

Parameter

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::name () : String [0..1]

```
if from.direction = UML::ParameterDirectionKind::return then 'result' else from.name endif
```

- ReferenceUsage::direction () : FeatureDirectionKind [0..1]

```
Helper.getKerMLParameterDirectionKind(from.direction)
```

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
let typings: Set(KerML::FeatureTyping) = if from.type.oclIsUndefined() then Set{} else Set{Pa
let multiplicities: Set(KerML::Relationship) = Set{MultiplicityMembership_Mapping.getMapped(f
let defaultValues: Set(KerML::Relationship) = if from.defaultValue.oclIsUndefined() then Set{
ElementOwnership_Mapping.getMappedColl(from.ownedComment)->asSet()
->union(typings)
->union(multiplicities)
->union(defaultValues)
```

C.2.5.4.2.25 ParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Parameter* mapping.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

Parameter

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ParameterMembership::ownedMemberParameter () : Feature [1]

  ```
  Parameter_Mapping.getMapped(from)
  ```

C.2.5.4.2.26 ParameterSet_Mapping

Description

*** not specified yet ***

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

ParameterSet

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::name () : String [0..1]

  ```
  from.name
  ```

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

  ```
  from.parameter->collect(p | ParameterSetParameterFeatureMembership_Mapping.getMapped(from, p
  ```

C.2.5.4.2.27 ParameterSetMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *ParameterSet* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ParameterSet

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
ParameterSet_Mapping.getMapped(from)
```

C.2.5.4.2.28 ParameterSetParameterFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *ParameterSet* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ParameterSet

Mapping Target

FeatureMembership with qualifier: parameter:Parameter

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature (in parameter : Parameter) : Feature [1]

```
ParameterSetParameterReferenceUsage_Mapping.getMapped(parameter)
```

C.2.5.4.2.29 ParameterSetParameterReferenceUsage_Mapping

Description

Creates a reference usage for the *Parameter* mapping.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Parameter

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

  ```
  Set{ParameterSetParameterReferenceUsageFeatureValue_Mapping.getMapped(from), MultiplicityMem
  ```

C.2.5.4.2.30 ParameterSetParameterReferenceUsageFeatureValue_Mapping

Description

Creates a feature value relationship for the mapping class *Parameter*

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Parameter

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

  ```
  ParameterSetParameterReferenceUsageFeatureValueExpression_Mapping.getMapped(from)
  ```

C.2.5.4.2.31 ParameterSetParameterReferenceUsageFeatureValueExpression_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeatureReferenceExpression_Mapping

Mapping Source

Parameter

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureReferenceExpression::ownedRelationship () : Relationship [0..*]

```
Set{ParameterSetParameterReferenceUsageFeatureValueExpressionMembership_Mapping.getMapped(fr
```

C.2.5.4.2.32 ParameterSetParameterReferenceUsageFeatureValueExpressionMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Parameter* mapping.

General Mappings

GenericToMembership_Mapping

Mapping Source

Parameter

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Membership::memberElement () : Element [1]

```
from
```

C.2.5.4.2.33 ParameterToFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *Parameter* mapping.

General Mappings

TypedElementToFeatureTyping_Mapping

Mapping Source

Parameter

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

```
parameter.to
```

C.2.5.4.2.34 Property_Mapping

Description

*** not specified yet ***

General Mappings

PropertyCommon_Mapping
NamedElementMain_Mapping

Mapping Source

Property

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
if src.oclIsKindOf(UML::Property) then
    let p: UML::Property = src.oclAsType(UML::Property) in
    if p.type.oclIsUndefined() then
        false
    else
        not p.type.oclIsKindOf(UML::DataType) and
        not (p.name.indexOf('base_') > 0) and
        (p.association.oclIsUndefined() or p.association.ownedEnd->excludes(p))
    endif
else
    false
endif
```

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

C.2.5.4.2.35 PropertyCommon_Mapping

Description

*** not specified yet ***

General Mappings

StructuralFeature_Mapping

Mapping Source

Property

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isComposite () : Boolean [1]

```
 from.isComposite
```

- Feature::ownedRelationship () : Relationship [0..*]

```
let typings: Set(KerML::FeatureTyping) = if from.type.oclIsUndefined() then
    Set{}
else
    Set{StructuralFeatureToFeatureTyping_Mapping.getMapped(from)}
endif in
let subsettings: Set(KerML::Subsetting) = from.subsettedProperty
    ->collect(p | PropertySubsetting_Mapping.getMapped(from, p))->asSet() in
let defaultValue: Set(KerML::OwningMembership) = if from.defaultValue.oclIsUndefined() then
    Set{}
else
    Set{DefaultValue_Mapping.getMapped(from)}
endif in
typings->union(subsettings)->union(defaultValue)
->including(MultiplicityMembership_Mapping.getMapped(from))->asSet()
```

- Feature::isEnd () : Boolean [1]

```
 if from.association.oclIsUndefined() then
    false
else
    from.association.ownedEnd->includes(from)
endif
```

- Feature::isDerived () : Boolean [1]

```
 from.isDerived
```

C.2.5.4.2.36 DefaultValue_Mapping

Description

The expected SysML v2 textual syntax of a mapped SysML v2 default value is as follows:

```
attribute value : ScalarValues::String default := "thisIsTheDefaultValue";
```

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Property

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::isDefault () : Boolean [1]

  ```
  true
  ```

- FeatureValue::value () : Expression [1]

  ```
  if from.defaultValue.oclIsKindOf(UML::LiteralSpecification) then Helper.mappedValueSpecifica
  ```

C.2.5.4.2.37 PropertySubsetting_Mapping

Description

Creates a subsetting relationship for the *subsettingFeature()* and the *subsettedFeature()* for the *Property* mapping.

General Mappings

GenericToSubsetting_Mapping

Mapping Source

Property

Mapping Target

Subsetting with qualifier: subsettedProperty:Property

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Subsetting::subsettingFeature () : Feature [1]

  ```
  Property_Mapping.getMapped(from)
  ```

- Subsetting::subsettedFeature (in subsettedProperty : Property) : Feature [1]

  ```
  Property_Mapping.getMapped(subsettedProperty)
  ```

C.2.5.4.2.38 PropertyUntyped_Mapping

Description

*** not specified yet ***

General Mappings

PropertyCommon_Mapping
GenericToReferenceUsage_Mapping
NamedElementMain_Mapping

Mapping Source

Property

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
from.type.oclIsUndefined()
```

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

C.2.5.4.2.39 Realization_Mapping

Description

*** issue *** This mapping is not appropriate since the Realization can have more than one client and more than one supplier and that the semantics defined in UML is much more informal than those of a generalization

General Mappings

Abstraction_Mapping

Mapping Source

Realization

Mapping Target

Dependency

Owned Mappings

(none)

C.2.5.4.2.40 Slot_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeature_Mapping
ElementMain_Mapping

Mapping Source

Slot

Mapping Target

Feature

Owned Mappings

(none)

C.2.5.4.2.41 SlotMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Slot* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Slot

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::isReadOnly () : Boolean [1]

  ```
  from.isReadOnly
  ```

- FeatureMembership::ownedMemberFeature () : Feature [1]

  ```
  from
  ```

- FeatureMembership::memberName () : String [0..1]

```
from.definingFeature.name
```

C.2.5.4.2.42 SlotToFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *Slot* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Slot

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

  ```
  ElementMain_Mapping.getMapped(from)
  ```

- FeatureTyping::typedFeature () : Feature [1]

  ```
  Slot_Mapping.getMapped(from)
  ```

C.2.5.4.2.43 SlotValue_Mapping

Description

Issue here since a KerML feature cannot have more than one FeatureValue while a UML::Slot can. How to manage collection of values?

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

ValueSpecification

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
src.owner.oclIsKindOf(UML::Slot)
```

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

  ```
  Helper.mappedValueSpecification(from)
  ```

- FeatureValue::featureWithValue () : Feature [1]

  ```
  Slot_Mapping.getMapped(from.owner)
  ```

C.2.5.4.2.44 StructuralFeature_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeature_Mapping

Mapping Source

StructuralFeature

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isOrdered () : Boolean [1]

  ```
  from.isOrdered
  ```

- Feature::isAbstract () : Boolean [1]

```
 false
```

- Feature::isUnique () : Boolean [1]

```
 from.isUnique
```

- Feature::ownedRelationship () : Relationship [0..*]

```
 let typing: KerML::FeatureTyping = StructuralFeatureToFeatureTyping_Mapping.getMapped(from)
if typing.oclIsUndefined() then
    Set{MultiplicityMembership_Mapping.getMapped(from)}
else
    Set{MultiplicityMembership_Mapping.getMapped(from), typing}
endif
```

- Feature::isReadOnly () : Boolean [1]
  abstract rule

C.2.5.4.2.45 StructuralFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *StructuralFeature* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

StructuralFeature

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [0..1]

```
 NamedElementMain_Mapping.getMapped(from)
```

- FeatureMembership::visibility () : VisibilityKind [1]

```
 if (from.oclIsKindOf(UML::NamedElement)) then
    Helper.getKerMLVisibilityKind(from.oclAsType(UML::NamedElement).visibility)
```

```
        else
            KerML::VisibilityKind::public
        endif
```

C.2.5.4.2.46 StructuralFeatureToFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *StructuralFeature* mapping.

General Mappings

TypedElementToFeatureTyping_Mapping

Mapping Source

StructuralFeature

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

```
ElementMain_Mapping.getMapped(from)
```

C.2.5.4.2.47 TypedElementToFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *TypedElement* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

TypedElement

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
not src.type.oclIsUndefined()
    and not(src.type.oclIsKindOf(UML::Enumeration) and Helper.getSysMLv2EnumerationDefinition(src.ty
```

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
 if from.type.oclIsKindOf(UML::PrimitiveType) then
    Helper.getScalarValueType(from.type)
else if from.type.oclIsKindOf(UML::Enumeration) then
    Helper.getEnumerationType(from.type)
else
    Classifier_Mapping.getMapped(from.type)
endif endif
```

C.2.5.4.2.48 UpperBoundValueOwnership_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

MultiplicityElement

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [0..1]

```
 if from.upper <> -1 then
    LiteralUnlimitedToInteger_Mapping.getMapped(from.upperValue)
else
    LiteralUnlimitedToUnbounded_Mapping.getMapped(from.upperValue)
endif
```

C.2.5.5 CommonBehavior

C.2.5.5.1 Overview

Table 20. List of all Overview Mapping Specfications

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
AnyReceiveEvent	Element	AnyReceiveEvent_Mapping	
CallEvent			
ChangeEvent	TextualRepresentation	ChangeEvent_Mapping	
FunctionBehavior			
OpaqueBehavior	ActionDefinition ActionUsage	OpaqueBehaviorAsDefinition_Mapping OpaqueBehaviorAsUsage_Mapping	src.owner.oclIsKindOf(UML::Package) not src.owner.oclIsKindOf(UML::Package)
SignalEvent			
TimeEvent	TextualRepresentation	TimeEvent_Mapping	
Trigger	AcceptActionUsage	Trigger_Mapping	

C.2.5.5.2 UML4SysML CommonBehavior elements not mapped

Table 21. List of SysML v1 elements not mapped of this section

SysML v1 Concept	Rationale
CallEvent	The concept of a CallEvent is not supported by SysML v2.

C.2.5.5.3 Mapping Specifications

C.2.5.5.3.1 AnyReceiveEvent_Mapping

Description

*** not specified yet ***

General Mappings

NamedElementMain_Mapping

Mapping Source

AnyReceiveEvent

Mapping Target

Element

Owned Mappings

(none)

C.2.5.5.3.2 Behavior_Mapping

Description

*** not specified yet ***

General Mappings

GenericToBehavior_Mapping
Class_Mapping

Mapping Source

Behavior

Mapping Target

Behavior

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
true
```

Mapping rules

The following lists the mapping rules for the target element properties.

- Behavior::ownedRelationship () : Relationship [0..*]

```
let parameters: Set(UML::Element) = from.ownedElement->select(e | e.oclIsKindOf(UML::Paramete
let parameterSets: Set(UML::Element) = src.ownedElement->select(e | e.oclIsKindOf(UML::Parame
let features: Set(UML::Element) = from.ownedElement->select(e | e.oclIsKindOf(UML::Property))
let elementsOMS: Set(UML::Element) = (((from.ownedElement - parameters) parameterSets) - feat
elementsOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(features->collect(e | PropertyMembership_Mapping.getMapped(e)))
->union(parameters->collect(e | ParameterMembership_Mapping.getMapped(e)))
->union(parameterSets->collect(e | ParameterSetMembership_Mapping.getMapped(e)))
```

C.2.5.5.3.3 ChangeEvent_Mapping

Description

*** not specified yet ***

General Mappings

GenericToTextualRepresentation_Mapping
NamedElementMain_Mapping

Mapping Source

ChangeEvent

Mapping Target

TextualRepresentation

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- TextualRepresentation::body () : String [1]

```
if from.changeExpression.oclIsKindOf(UML::OpaqueExpression)
then if from.changeExpression.oclAsType(UML::OpaqueExpression).body.oclIsUndefined() then Ocl
else OclUndefined
endif
```

- TextualRepresentation::language () : String [1]

```
if from.changeExpression.oclIsKindOf(UML::OpaqueExpression)
then if from.changeExpression.oclAsType(UML::OpaqueExpression).language->size() = 0 then OclU
else OclUndefined
endif
```

C.2.5.5.3.4 CommonOpaqueBehavior_Mapping

Description

*** not specified yet ***

General Mappings

Behavior_Mapping

Mapping Source

OpaqueBehavior

Mapping Target

Behavior

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Behavior::ownedRelationship () : Relationship [0..*]

```
let parameters : Set(UML::Parameter) = from.ownedElement->select(e | e.oclIsKindOf(UML::Param
let parameterSets : Set(UML::ParameterSet) = from.ownedElement->select(e | e.oclIsKindOf(UML:
let features : Set(UML::Property) = from.ownedElement->select(e | e.oclIsKindOf(UML::Property
let elementsOMS: Set(UML::Element) = (((from.ownedElement - parameters) - parameterSets) - fe
elementsOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(features->collect(e | PropertyMembership_Mapping.getMapped(e)))
->union(parameters->collect(e | ParameterMembership_Mapping.getMapped(e)))
->union(parameterSets->collect(e | ParameterSetMembership_Mapping.getMapped(e)))
->union(from.language->collect(l | OpaqueBehaviorMembership_Mapping.getMapped(from, l)))
```

C.2.5.5.3.5 OpaqueBehaviorAsDefinition_Mapping

Description

*** not specified yet ***

General Mappings

GenericToDefinition_Mapping
CommonOpaqueBehavior_Mapping

Mapping Source

OpaqueBehavior

Mapping Target

ActionDefinition

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
src.owner.oclIsKindOf(UML::Package)
```

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

C.2.5.5.3.6 OpaqueBehaviorAsUsage_Mapping

Description

*** not specified yet ***

General Mappings

CommonOpaqueBehavior_Mapping
GenericToActionUsage_Mapping

Mapping Source

OpaqueBehavior

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
 not src.owner.oclIsKindOf(UML::Package)
```

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

C.2.5.5.3.7 OpaqueBehaviorMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *OpaqueBehavior* mapping.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

OpaqueBehavior

Mapping Target

OwningMembership with qualifier: language:String

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OwningMembership::ownedMemberElement (in language : String) : Element [1]

  ```
  OpaqueBehaviorSpecification_Mapping.getMapped(from, language)
  ```

C.2.5.5.3.8 OpaqueBehaviorSpecification_Mapping

Description

*** not specified yet ***

General Mappings

GenericToTextualRepresentation_Mapping

Mapping Source

OpaqueBehavior

Mapping Target

TextualRepresentation with qualifier: language:String

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- TextualRepresentation::body () : String [1]

  ```
   let index:Integer = from.language->indexOf(language) in
  from._'body'->at(index)
  ```

- TextualRepresentation::language () : String [1]

  ```
  language
  ```

C.2.5.5.3.9 TimeEvent_Mapping

Description

tbd - just a placeholder yet

General Mappings

NamedElementMain_Mapping
GenericToTextualRepresentation_Mapping

Mapping Source

TimeEvent

Mapping Target

TextualRepresentation

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- TextualRepresentation::body () : String [1]

  ```
  'tbd timeevent'
  ```

C.2.5.5.3.10 Trigger_Mapping

C.2.5.6 CommonStructure

C.2.5.6.1 Overview

Table 23. List of all Overview Mapping Specfications

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter

Abstraction	Dependency
	Dependency
	Dependency
	SatisfyRequirementUsage
	AllocationDefinition
	Dependency
	Dependency
		Realization_Mapping
		Trace_Mapping
		Refine_Mapping
		Satisfy_Mapping
		AllocationDefinition_Mapping
		Abstraction_Mapping
		DeriveReqt_Mapping
			Helper.hasStereotypeApplied(from, 'SysML::Requirements::Trace')
			Helper.hasStereotypeApplied(from, 'SysML::Requirements::Refine')
			let satisfy: UML::Abstraction = src.oclAsType(UML::Abstraction) in if satisfy.oclIsUndefined() then false else Helper.hasStereotypeApplied(satisfy, 'SysML::Requirements::Satisfy') and satisfy.client->exists(c | not c.oclIsKindOf(UML::Classifier)) endif
			Helper.hasStereotypeApplied(from, 'SysML::Allocations::Allocate') and from.client->select(t | t.oclIsKindOf(UML::Type))->notEmpty()
			
			Helper.hasStereotypeApplied(src, 'SysML::Requirements::DeriveReqt')

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
Comment	Comment ConcernDefinition Comment	Comment_Mapping CommentToConcern_Mapping ProblemRationale_Mapping	not Helper.hasStereotypeApplied(from, 'SysML::ModelElements::ElementGroup') (not Helper.hasStereotypeApplied(from, 'SysML::ModelElements::ElementGroup')) and UML::Classifier.allInstances()->select(s | Helper.hasStereotypeApplied(s, 'SysML::ModelElements::Stakeholder'))->collect(c | Helper.getTagValue(c, 'SysML::ModelElements::Stakeholder', 'concernList'))->flatten()->includes(from) (not Helper.hasStereotypeApplied(from, 'SysML::ModelElements::ElementGroup')) and (Helper.hasStereotypeApplied(from, 'SysML::ModelElements::Problem') or Helper.hasStereotypeApplied(from, 'SysML::ModelElements::Rationale'))
Constraint	ConstraintDefinition	Constraint_Mapping	

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter

Dependency	Dependency
	Dependency
	Dependency
	Dependency
	SatisfyRequirementUsage
	AllocationDefinition
	Dependency
	Dependency
	Dependency
		Realization_Mapping
		Trace_Mapping
		Dependency_Mapping
		Refine_Mapping
		Satisfy_Mapping
		AllocationDefinition_Mapping
		Abstraction_Mapping
		DeriveReqt_Mapping
		Usage_Mapping
			Helper.hasStereotypeApplied(from, 'SysML::Requirements::Trace')
			Helper.hasStereotypeApplied(from, 'SysML::Requirements::Refine')
			let satisfy: UML::Abstraction = src.oclAsType(UML::Abstraction) in if satisfy.oclIsUndefined() then false else Helper.hasStereotypeApplied(satisfy, 'SysML::Requirements::Satisfy') and satisfy.client->exists(c | not c.oclIsKindOf(UML::Classifier)) endif
			Helper.hasStereotypeApplied(from, 'SysML::Allocations::Allocate') and from.client->select(t | t.oclIsKindOf(UML::Type))->notEmpty()
			Helper.hasStereotypeApplied(src, 'SysML::Requirements::DeriveReqt')

ElementImport	Membership	ElementImport_Mapping	
PackageImport	Import	PackageImport_Mapping	
Realization	Dependency	Realization_Mapping	
Usage	Dependency	Usage_Mapping	

C.2.5.6.2 Mapping Specifications

C.2.5.6.2.1 Abstraction_Mapping

Description

There is no way to represent the "mapping" property on the target metaclass

General Mappings

Dependency_Mapping

Mapping Source

Abstraction

Mapping Target

Dependency

Owned Mappings

(none)

C.2.5.6.2.2 Comment_Mapping

Description

test

General Mappings

ElementMain_Mapping
GenericToAnnotatingElement_Mapping

Mapping Source

Comment

Mapping Target

Comment

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
not Helper.hasStereotypeApplied(from, 'SysML::ModelElements::ElementGroup')
```

Mapping rules

The following lists the mapping rules for the target element properties.

- Comment::annotation () : Annotation [0..*]

  ```
   from.annotatedElement->collect(e | CommentToAnnotation_Mapping.getMapped(from, e))
  ```

- Comment::ownedRelationship () : Relationship [0..*]

  ```
   ElementOwnership_Mapping.getMappedColl(from.ownedComment)
  ->union(self.annotation())
  ```

- Comment::body () : String [1]

  ```
   if from.body->isEmpty() then '' else from.body endif
  ```

C.2.5.6.2.3 CommentToAnnotation_Mapping

Description

*** not specified yet ***

General Mappings

GenericToAnnotation_Mapping

Mapping Source

Comment

Mapping Target

Annotation with qualifier: annotatedElement:Element

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Annotation::annotatedElement (in annotatedElement : Element) : Element [1]

```
ElementMain_Mapping.getMapped(annotatedElement)
```

- Annotation::annotatingElement () : AnnotatingElement [1]

```
Comment_Mapping.getMapped(from)
```

- Annotation::owningAnnotatedElement () : Element [0..1]

```
null
```

C.2.5.6.2.4 Constraint_Mapping

Description

*** not specified yet ***

General Mappings

GenericToConstraintDefinition_Mapping
NamedElementMain_Mapping

Mapping Source

Constraint

Mapping Target

ConstraintDefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ConstraintDefinition::ownedRelationship () : Relationship [0..*]

```
 ElementOwnership_Mapping.getMappedColl(from.ownedComment)->asSet()
->union(Set{ElementFeatureMembership_Mapping.getMapped(from.specification), CommonReturnParam
```

C.2.5.6.2.5 ConstrainedElementFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *Constraint* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Constraint

Mapping Target

FeatureMembership

Owned Mappings

- constraintUsage : ConstraintUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
 constraintUsage.to
```

C.2.5.6.2.6 ConstraintUsageFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *Constraint* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Constraint

Mapping Target

FeatureTyping

Owned Mappings

- constraintUsage : ConstraintUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
from
```

- FeatureTyping::typedFeature () : Feature [1]

```
constraintUsage.to
```

C.2.5.6.2.7 ConstraintUsage_Mapping

Description

*** not specified yet ***

General Mappings

GenericToUsage_Mapping

Mapping Source

Constraint

Mapping Target

AssertConstraintUsage

Owned Mappings

- constraintUsageFeatureTyping : ConstraintUsageFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- AssertConstraintUsage::name () : String [0..1]

```
'assert_' + from.name
```

- AssertConstraintUsage::ownedRelationship () : Relationship [0..*]

```
 ElementOwnership_Mapping.getMappedColl(from.ownedComment)->asSet()
->union(Set{constraintUsageFeatureTyping.to, CommonReturnParameterReferenceUsageMembership_Ma
```

C.2.5.6.2.8 Dependency_Mapping

Description

*** not specified yet ***

General Mappings

DirectedRelationship_Mapping

Mapping Source

Dependency

Mapping Target

Dependency

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Dependency::supplier () : Element [0..*]

```
from.target->collect(e | ElementMain_Mapping.getMapped(e))
```

- Dependency::name () : String [0..1]

```
from.name
```

- Dependency::client () : Element [0..*]

```
from.source->collect(e | ElementMain_Mapping.getMapped(e))
```

C.2.5.6.2.9 DirectedRelationship_Mapping

Description

*** not specified yet ***

General Mappings

Relationship_Mapping

Mapping Source

DirectedRelationship

Mapping Target

Relationship

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Relationship::target () : Element [0..*]

  ```
  from.target->collect(e | ElementMain_Mapping.getMapped(e))
  ```

- Relationship::source () : Element [0..*]

  ```
  from.source->collect(e | ElementMain_Mapping.getMapped(e))
  ```

C.2.5.6.2.10 ElementMain_Mapping

Description

This is the general abstract class to be used as an ancestor for any class mapping specification.

General Mappings

GenericToElement_Mapping
MainMapping

Mapping Source

Element

Mapping Target

Element

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Element::ownedRelationship () : Relationship [0..*]

  ```
  ElementOwnership_Mapping.getMappedColl(from.ownedComment)
  ```

- Element::elementId () : String [1]

  ```
  Helper.getID(from)
  ```

C.2.5.6.2.11 ElementMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Element* mapping.

General Mappings

GenericToMembership_Mapping

Mapping Source

Element

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Membership::membershipOwningNamespace () : Element [0..*]

  ```
  Set{ElementMain_Mapping(from)} -- will not be used since corresponding att is derived, but r
  ```

- Membership::memberElement () : Element [1]

  ```
  ElementMain_Mapping.getMapped(from)
  ```

- Membership::visibility () : VisibilityKind [1]

  ```
   if (from.oclIsKindOf(UML::NamedElement)) then
      from.oclAsType(UML::NamedElement).visibility
  else
  ```

```
            KerML::VisibilityKind::public
        endif
```

C.2.5.6.2.12 ElementOwnership_Mapping

Description

General Mappings

GenericToRelationship_Mapping

Mapping Source

Element

Mapping Target

Relationship

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Relationship::target () : Element [0..*]

  ```
  OrderedSet{ElementMain_Mapping.getMapped(from)}
  ```

- Relationship::source () : Element [0..*]

  ```
  OrderedSet{ElementMain_Mapping.getMapped(from.owner)}
  ```

- Relationship::ownedRelatedElement () : Element [0..*]

  ```
  self.target()
  ```

C.2.5.6.2.13 ElementOwningMembership_Mapping

Description

Creates a owning membership relationship for *ownedMemberElement()* for the *Element* mapping.

General Mappings

ElementMembership_Mapping
ElementOwnership_Mapping

Mapping Source

Element

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

  ```
  ElementMain_Mapping.getMapped(from)
  ```

- OwningMembership::membershipOwningNamespace () : Element [0..*]

  ```
  Set{ElementMain_Mapping(from)} -- will not be used since corresponding att is derived, but r
  ```

- OwningMembership::ownedRelatedElement () : Element [0..*]

  ```
  Set{self.ownedMemberElement()}
  ```

C.2.5.6.2.14 NamedElementMain_Mapping

Description

*** not specified yet ***

General Mappings

ElementMain_Mapping

Mapping Source

NamedElement

Mapping Target

Element

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Element::name () : String [0..1]

```
from.name
```

C.2.5.6.2.15 Namespace_Mapping

Description

*** not specified yet ***

General Mappings

GenericToNamespace_Mapping
NamedElementMain_Mapping

Mapping Source

Namespace

Mapping Target

Namespace

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Namespace::ownedImport () : Import [0..*]

```
Set{}
```

C.2.5.6.2.16 Relationship_Mapping

Description

*** not specified yet ***

General Mappings

GenericToRelationship_Mapping
ElementMain_Mapping

Mapping Source

Relationship

Mapping Target

Relationship

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Relationship::owningRelatedElement () : Element [0..1]

  ```
  ElementMain_Mapping.getMapped(from.owner)
  ```

- Relationship::ownedRelatedElement () : Element [0..*]

  ```
  from.relatedElement->select(e | from.ownedElement->includes(e))->collect(e | ElementMain_Map
  ```

C.2.5.6.2.17 Usage_Mapping

Description

*** not specified yet ***

General Mappings

Dependency_Mapping

Mapping Source

Usage

Mapping Target

Dependency

Owned Mappings

(none)

C.2.5.7 InformationFlows

C.2.5.7.1 Overview

Table 24. List of all Overview Mapping Specfications

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
InformationFlow	FlowConnectionUsage FlowConnectionDefinition	ItemFlow_Mapping InformationFlow_Mapping	Helper.hasStereotypeApplied(from, 'SysML::Ports&Flows::ItemFlow')
InformationItem	ItemDefinition	InformationItem_Mapping	

C.2.5.7.2 Mapping Specifications

C.2.5.7.2.1 InformationFlow_Mapping

Description

*** not specified yet ***

General Mappings

Relationship_Mapping

Mapping Source

InformationFlow

Mapping Target

FlowConnectionDefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FlowConnectionDefinition::ownedRelationship () : Relationship [0..*]

```
 from.source->collect(s | InformationFlowSourceMembership_Mapping.getMapped(from, s))
->union(from.target->collect(t | InformationFlowTargetMembership_Mapping.getMapped(from, t)))
->asOrderedSet()
```

C.2.5.7.2.2 InformationFlowEndCommonMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *InformationFlow* mapping.

General Mappings

GenericToElement_Mapping

Mapping Source

InformationFlow

Mapping Target

Element

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Element::ownedMemberFeature (in end : NamedElement) : Feature [1]
  abstract rule
- Element::ownedRelatedElement () : Element [0..*]

```
Set{self.ownedMemberFeature()}
```

- Element::memberName () : String [0..1]

```
null
```

- Element::visibility () : VisibilityKind [1]

```
KerML::VisibilityKind::public
```

- Element::memberShortName () : String [0..1]

```
null
```

C.2.5.7.2.3 InformationFlowSource_Mapping

Description

*** not specified yet ***

General Mappings

GenericToElement_Mapping

Mapping Source

InformationFlow

Mapping Target

Feature with qualifier: source:NamedElement

Owned Mappings

- informationFlowSourceTyping : InformationFlowSourceTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isEnd () : Boolean [1]

```
true
```

- Feature::name (in source : NamedElement) : String [0..1]

```
'source'
```

- Feature::ownedRelationship () : Relationship [0..*]

```
Set{informationFlowSourceTyping.to}
```

C.2.5.7.2.4 InformationFlowSourceMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *InformationFlow* mapping.

General Mappings

InformationFlowEndCommonMembership_Mapping

Mapping Source

InformationFlow

Mapping Target

FeatureMembership with qualifier: source:NamedElement

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature (in source : NamedElement) : Feature [1]

```
InformationFlowSource_Mapping.getMapped(from, source)
```

C.2.5.7.2.5 InformationFlowSourceTyping_Mapping

Description

*** not specified yet ***

General Mappings

GenericToElement_Mapping

Mapping Source

InformationFlow

Mapping Target

FeatureTyping with qualifier: source:NamedElement

Owned Mappings

- informationFlowSource : InformationFlowSource_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature (in source : NamedElement) : Feature [1]

```
InformationFlowSource_Mapping.getMapped(from, source)
```

- FeatureTyping::type (in source : NamedElement) : Type [1]

```
ElementMain_Mapping.getMapped(source)
```

C.2.5.7.2.6 InformationFlowTarget_Mapping

Description

*** not specified yet ***

General Mappings

GenericToElement_Mapping

Mapping Source

InformationFlow

Mapping Target

Feature with qualifier: target:NamedElement

Owned Mappings

- informationFlowTargetTyping : InformationFlowTargetTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::ownedRelationship () : Relationship [0..*]

```
Set{informationFlowTargetTyping.to}
```

- Feature::name (in target : NamedElement) : String [0..1]

```
'target_'+target.name
```

- Feature::isEnd () : Boolean [1]

```
true
```

C.2.5.7.2.7 InformationFlowTargetMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *InformationFlow* mapping.

General Mappings

InformationFlowEndCommonMembership_Mapping

Mapping Source

InformationFlow

Mapping Target

FeatureMembership with qualifier: target:NamedElement

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature (in target : NamedElement) : Feature [1]

```
InformationFlowTarget_Mapping.getMapped(from, target)
```

C.2.5.7.2.8 InformationFlowTargetTyping_Mapping

Description

*** not specified yet ***

General Mappings

GenericToElement_Mapping

Mapping Source

InformationFlow

Mapping Target

FeatureTyping with qualifier: target:NamedElement

Owned Mappings

- informationTarget : InformationFlowTarget_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::type (in target : NamedElement) : Type [1]

```
ElementMain_Mapping.getMapped(target)
```

- FeatureTyping::typedFeature (in target : NamedElement) : Feature [1]

```
InformationFlowTarget_Mapping.getMapped(from, target)
```

C.2.5.7.2.9 InformationItem_Mapping

Description

*** not specified yet ***

General Mappings

Classifier_Mapping

Mapping Source

InformationItem

Mapping Target

ItemDefinition

Owned Mappings

(none)

C.2.5.8 Interactions

C.2.5.8.1 Overview

Table 25. List of all Overview Mapping Specfications

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
ActionExecutionSpecification	ActionUsage	ActionExecutionSpecification_Mapping	
BehaviorExecutionSpecification	ActionUsage	BehaviorExecutionSpecification_Mapping	
CombinedFragment	Interaction	CombinedFragment_Mapping	
ConsiderIgnoreFragment			
Continuation			
DestructionOccurrenceSpecification			
ExecutionOccurrenceSpecification			
Gate			
GeneralOrdering			
Interaction	Interaction	Interaction_Mapping	
InteractionConstraint			
InteractionOperand	Interaction	InteractionOperand_Mapping	
InteractionUse	Step	InteractionUse_Mapping	
Lifeline	PartUsage	LifelinePartUsage_Mapping	
Message	ItemFlow	Message_Mapping	
MessageOccurrenceSpecification			
OccurrenceSpecification			
PartDecomposition			
StateInvariant	Invariant	StateInvariant_Mapping	

C.2.5.8.2 Mapping Specifications

C.2.5.8.2.1 ActionExecutionSpecification_Mapping

Description

*** not specified yet ***

General Mappings

GenericToActionUsage_Mapping
NamedElementMain_Mapping

Mapping Source

ActionExecutionSpecification

Mapping Target

ActionUsage

Owned Mappings

(none)

C.2.5.8.2.2 BehaviorExecutionSpecification_Mapping

Description

*** not specified yet ***

General Mappings

GenericToActionUsage_Mapping
NamedElementMain_Mapping

Mapping Source

BehaviorExecutionSpecification

Mapping Target

ActionUsage

Owned Mappings

(none)

C.2.5.8.2.3 CombinedFragment_Mapping

Description

A UML4SysML::Interaction is mapped to a SysMLv2::Interaction.

General Mappings

NamedElementMain_Mapping
GenericToInteraction_Mapping

Mapping Source

CombinedFragment

Mapping Target

Interaction

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Interaction::ownedRelationship () : Relationship [0..*]

```
let operands: Set(UML::Element) = from.ownedElement->select(e | e.oclIsKindOf(UML::Interactio
let occurrencesSpecs: Set(UML::Element) = from.ownedElement->select(e | e.oclIsKindOf(UML::Oc

let elements: Set(UML::Element) = (from.ownedElement - operands) - occurrencesSpecs in
elements->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(operands->collect(e | InteractionOperandMembership_Mapping.getMapped(e)))
```

C.2.5.8.2.4 CombinedFragmentMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *CombinedFragment* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

CombinedFragment

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [0..1]

  ```
  self.memberFeature()
  ```

- FeatureMembership::memberFeature () : Feature [1]

  ```
  ElementMain_Mapping.getMapped(from)
  ```

C.2.5.8.2.5 ExecutionSpecificationMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *ExecutionSpecification* mapping.

General Mappings

GenericToEndFeatureMembership_Mapping

Mapping Source

ExecutionSpecification

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [0..1]

  ```
  self.memberFeature()
  ```

- FeatureMembership::memberFeature () : Feature [1]

  ```
  ElementMain_Mapping.getMapped(from)
  ```

C.2.5.8.2.6 Interaction_Mapping

Description

A UML4SysML::Interaction is mapped to a SysMLv2::Interaction.

General Mappings

Namespace_Mapping
GenericToInteraction_Mapping

Mapping Source

Interaction

Mapping Target

Interaction

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Interaction::ownedRelationship () : Relationship [0..*]

```
let lifelines: Set(UML::Element) = from.lifeline in
let messageOccurrences: Set(UML::Element) = from.ownedElement->select(e | e.oclIsKindOf(UML::
let executionOccurrences: Set(UML::Element) = from.fragment->select(e | e.oclIsKindOf(UML::Ex
let occurrencesSpecs: Set(UML::Element) = from.fragment->select(e | e.oclIsKindOf(UML::Occurr
let messages: Set(UML::Element) = from.message in
let invariants: Set(UML::Element) = from.fragment->select(e | e.oclIsKindOf(UML::StateInvaria
let interactionUsages: Set(UML::Element) = from.fragment->select(e | e.oclIsKindOf(UML::Inter
let combinedFragments: Set(UML::Element) = from.ownedElement->select( e | e.oclIsKindOf(UML::
let continuations: Set(UML::Element) = from.ownedElement->select(e | e.oclIsKindOf(UML::Conti
let elements: Set(UML::Element) = (((((((from.ownedElement - lifelines) - messageOccurrences
    - executionOccurrences) - occurrencesSpecs) - messages) - combinedFragments) - invariants
    - interactionUsages) - continuations in
elements->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(lifelines->collect(e | LifelineMembership_Mapping.getMapped(e)))
->union(executionOccurrences->collect(e | ExecutionSpecificationMembership_Mapping.getMapped(
->union(messages->collect(e | MessageMembership_Mapping.getMapped(e)))
->union(combinedFragments->collect(e | CombinedFragmentMembership_Mapping.getMapped(e)))
->union(invariants->collect(e | StateInvariantMembership_Mapping.getMapped(e)))
->union(interactionUsages->collect(e | InteractionUseMembership_Mapping.getMapped(e)))
```

C.2.5.8.2.7 InteractionOperand_Mapping

Description

A UML4SysML::Interaction is mapped to a SysMLv2::Interaction.

General Mappings

NamedElementMain_Mapping
GenericToInteraction_Mapping

Mapping Source

InteractionOperand

Mapping Target

Interaction

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Interaction::ownedRelationship () : Relationship [0..*]

```
let executionOccurrences: Set(UML::Element) = from.ownedElement->select(e | e.oclIsKindOf(UMI
let occurrencesSpecs: Set(UML::Element) = from.ownedElement->select(e | e.oclIsKindOf(UML::Oc
let continuations: Set(UML::Element) = from.ownedElement->select(e | e.oclIsKindOf(UML::Conti
let elements: Set(UML::Element) = ((from.ownedElement - executionOccurrences) - occurrencesSp
elements->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(executionOccurrences->collect(e |ExecutionSpecificationMembership_Mapping.getMapped(e
```

C.2.5.8.2.8 InteractionOperandMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *InteractionOperand* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

InteractionOperand

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [0..1]

```
self.memberFeature()
```

- FeatureMembership::memberFeature () : Feature [1]

```
ElementMain_Mapping.getMapped(from)
```

C.2.5.8.2.9 InteractionUse_Mapping

Description

*** not specified yet ***

General Mappings

GenericToStep_Mapping
Namespace_Mapping

Mapping Source

InteractionUse

Mapping Target

Step

Owned Mappings

- interactionUseTyping : InteractionUseTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Step::ownedRelationship () : Relationship [0..*]

```
Set{interactionUseTyping.to}
```

C.2.5.8.2.10 InteractionUseMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *InteractionUse* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

InteractionUse

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [0..1]

```
self.memberFeature()
```

- FeatureMembership::memberFeature () : Feature [1]

```
ElementMain_Mapping.getMapped(from)
```

C.2.5.8.2.11 InteractionUseTyping_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

InteractionUse

Mapping Target

FeatureTyping

Owned Mappings

- interactionUse : InteractionUse_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

```
interactionUse.to
```

- FeatureTyping::type () : Type [1]

```
ElementMain_Mapping.getMapped(from.refersTo)
```

C.2.5.8.2.12 LifelineMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Lifeline* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Lifeline

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [0..1]

  ```
  self.memberFeature()
  ```

- FeatureMembership::memberFeature () : Feature [1]

  ```
  ElementMain_Mapping.getMapped(from)
  ```

C.2.5.8.2.13 LifelinePartUsage_Mapping

Description

*** not specified yet ***

General Mappings

GenericToPartUsage_Mapping
NamedElementMain_Mapping

Mapping Source

Lifeline

Mapping Target

PartUsage

Owned Mappings

- lifelineFeatureTyping : LifelineFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- PartUsage::ownedRelationship () : Relationship [0..*]

  ```
  Set{lifelineFeatureTyping.to}
  ```

C.2.5.8.2.14 LifelineFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *Lifeline* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Lifeline

Mapping Target

FeatureTyping

Owned Mappings

- lifelinePartUsage : LifelinePartUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

  ```
  ElementMain_Mapping.getMapped(from.represents.type)
  ```

- FeatureTyping::typedFeature () : Feature [1]

  ```
  lifelinePartUsage.to
  ```

C.2.5.8.2.15 Message_Mapping

Description

*** not specified yet ***

General Mappings

GenericToItemFlow_Mapping
NamedElementMain_Mapping

Mapping Source

Message

Mapping Target

ItemFlow

Owned Mappings

(none)

C.2.5.8.2.16 MessageMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Message* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Message

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::memberFeature () : Feature [1]

  ```
  ElementMain_Mapping.getMapped(from)
  ```

- FeatureMembership::ownedMemberFeature () : Feature [0..1]

  ```
  self.memberFeature()
  ```

C.2.5.8.2.17 StateInvariant_Mapping

Description

*** not specified yet ***

General Mappings

GenericToExpression_Mapping
Namespace_Mapping

Mapping Source

StateInvariant

Mapping Target

Invariant

Owned Mappings

- stateInvariantTyping : StateInvariantTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Invariant::ownedRelationship () : Relationship [0..*]

```
Set{stateInvariantTyping.to}
```

C.2.5.8.2.18 StateInvariantMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *StateInvariant* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

StateInvariant

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::memberFeature () : Feature [1]

```
ElementMain_Mapping.getMapped(from)
```

- FeatureMembership::ownedMemberFeature () : Feature [0..1]

```
self.memberFeature()
```

C.2.5.8.2.19 StateInvariantTyping_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

StateInvariant

Mapping Target

FeatureTyping

Owned Mappings

- stateInvariant : StateInvariant_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
ElementMain_Mapping.getMapped(from.invariant)
```

- FeatureTyping::typedFeature () : Feature [1]

```
stateInvariant.to
```

C.2.5.9 Packages

C.2.5.9.1 Overview

Table 26. List of all Overview Mapping Specfications

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
Extension			

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
ExtensionEnd			
Image			
Model	Package	Model_Mapping	
Package	Package Package Package	Package_Mapping Profile_Mapping Model_Mapping	
PackageMerge			
Profile	Package	Profile_Mapping	
ProfileApplication			
Stereotype	MetadataDefinition	StereotypeMetadataDefinition_Mapping	not Helper.hasStereotypeApplied(from, 'SysML::Requirements::Requirement') and not from.oclIsTypeOf(UML::AssociationClass)

C.2.5.9.2 UML4SysML Packages elements not mapped

Table 27. List of SysML v1 elements not mapped of this section

SysML v1 Concept	Rationale
Extension	The mapping of the extension relationship is performed in the context of Stereotype_Mapping.
ExtensionEnd	The mapping of the extension end property is performed in the context of Stereotype_Mapping.
PackageMerge	The concept of the PackageMerge relationship is not supported by SysML v2.

C.2.5.9.3 Mapping Specifications

C.2.5.9.3.1 ElementImport_Mapping

Description

*** not specified yet ***

General Mappings

GenericToMembership_Mapping
DirectedRelationship_Mapping

Mapping Source

ElementImport

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Membership::memberElement () : Element [1]

  ```
  ElementMain_Mapping.getMapped(from.importedElement)
  ```

- Membership::visibility () : VisibilityKind [0..1]

  ```
  Helper.getKerMLVisibilityKind(from.visibility)
  ```

- Membership::aliases () : String [0..*]

  ```
  from.alias->asSet()
  ```

- Membership::membershipOwningPackage () : Namespace [1]

  ```
  Namespace_Mapping.getMapped(from.importingNamespace)
  ```

- Membership::memberName () : String [0..1]

  ```
  from.importedElement.name
  ```

C.2.5.9.3.2 Package_Mapping

Description

A UML::Package is mapped to a SysMLv2::Package. The property "URI" is mapped to a metadata if it has a value. The expected SysML v2 textual notation of a SysMLv1::Package is as follows:

```
package ThisIsAPackageWithURI {
  metadata SysMLv1Library::PackageData {URI="https://omg.org";}
}
```

General Mappings

Namespace_Mapping

Mapping Source

Package

Mapping Target

Package

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Package::ownedRelationship () : Relationship [0..*]

  ```
  Helper.packageOwnedRelationship(from)
  ```

C.2.5.9.3.3 PackageImport_Mapping

Description

*** not specified yet ***

General Mappings

DirectedRelationship_Mapping

Mapping Source

PackageImport

Mapping Target

Import

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Import::importOwningPackage () : Namespace [1]

  ```
  Namespace_Mapping.getMapped(from.importingNamespace)
  ```

- Import::importedPackage () : Namespace [1]

  ```
  Namespace_Mapping.getMapped(from.importedPackage)
  ```

- Import::visibility () : VisibilityKind [0..1]

  ```
  Helper.getKerMLVisibilityKind(from.visibility)
  ```

C.2.5.9.3.4 Model_Mapping

Description

SysMLv2 has no explicit model element for a model. The SysMLv1::Model element is mapped to a SysMLv2::Package. The property "viewpoint" is mapped to a metadata defined in the SysML v1 library. The expected SysML v2 textual notation of a SysMLv1::Model with URI and viewpoint is as follows. If URI or viewpoint are not set in the source model, the metadata is not generated.

```
package ThisIsAModel {
  metadata SysMLv1Library::PackageData {URI="https://omg.org";}
  metadata SysMLv1Library::ModelData {'viewpoint'="thisIsTheViewpointOfTheModel";}
}
```

General Mappings

Package_Mapping

Mapping Source

Model

Mapping Target

Package

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Package::ownedRelationship () : Relationship [0..*]

```
 let relationships : Set(KerML::Relationship) = Helper.packageOwnedRelationship(from) in
if from.viewpoint.oclIsUndefined() or from.viewpoint = '' then
    relationships
else
    relationships->including(ModelViewpointMetadataMembership_Mapping.getMapped(from))
endif
```

C.2.5.9.3.5 ModelViewpointMetadataUsage_Mapping

C.2.5.9.3.6 ModelViewpointMetadataFeatureMembership_Mapping

Description

The mapping class creates the feature membership relationship for the metadata feature to store the UML::Model::viewpoint property.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Model

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [0..1]

```
ModelViewpointMetadataReferenceUsage_Mapping.getMapped(from)
```

C.2.5.9.3.7 ModelViewpointMetadataReferenceUsage_Mapping

Description

The mapping class creates the MetadataFeature for the mapping of the property UML::Model::viewpoint.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Model

Mapping Target

ReferenceUsage

Owned Mappings

- modelViewpointMetadataRedefinition : ModelViewpointMetadataRedefinition_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
Set{modelViewpointMetadataRedefinition.to, ModelViewpointMetadataFeatureValue_Mapping.getMap
```

C.2.5.9.3.8 ModelViewpointMetadataFeatureTyping_Mapping

Description

The mapping class creates the FeatureTyping relationship for the AnnotatingFeature for the metadata to store the UML::Model::viewpoint property.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Model

Mapping Target

FeatureTyping

Owned Mappings

- modelViewpointMetadataUsage : ModelViewpointMetadataUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

```
modelViewpointMetadataUsage.to
```

- FeatureTyping::type () : Type [1]

```
SysMLv2::MetadataDefinition.allInstances()->any(m | m.qualifiedName = 'SysMLv1Library::Model
```

C.2.5.9.3.9 ModelViewpointMetadataMembership_Mapping

Description

The mapping class creates a membership relationship for the metadata feature value for the UML::Model::viewpoint property.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Model

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

```
ModelViewpointMetadataUsage_Mapping.getMapped(from)
```

C.2.5.9.3.10 ModelViewpointMetadataFeatureValue_Mapping

Description

The mapping class maps the value of the property UML::Model::viewpoint.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Model

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

```
ModelViewpointValue_Mapping.getMapped(from)
```

C.2.5.9.3.11 ModelViewpointMetadataRedefinition_Mapping

Description

The mapping class creates the redefinition of the attribute for the metadata UML::Model::viewpoint.

General Mappings

GenericToRedefinition_Mapping

Mapping Source

Model

Mapping Target

Redefinition

Owned Mappings

- modelViewpointMetadataReferenceUsage : ModelViewpointMetadataReferenceUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Redefinition::redefinedFeature () : Feature [1]

```
 let m : SYSML2::Membership = SYSML2::AttributeUsage.allInstances()->collect(dt | dt.owningRe
if (m.oclIsUndefined()) then OclUndefined else m.memberElement endif
```

- Redefinition::redefiningFeature () : Feature [1]

```
 modelViewpointMetadataReferenceUsage.to
```

C.2.5.9.3.12 ModelViewpointValue_Mapping

Description

The mapping class maps the value expression of the property UML::Model::viewpoint.

General Mappings

GenericToExpression_Mapping

Mapping Source

Model

Mapping Target

LiteralString

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- LiteralString::value () : String [1]

```
from.viewpoint
```

C.2.5.9.3.13 PackageURIMetadataUsage_Mapping

Description

The mapping class creates the annotating feature to annotate the generated Package element with metadata to store the UML::Package::URI property.

General Mappings

GenericToMetadataUsage_Mapping

Mapping Source

Package

Mapping Target

MetadataUsage

Owned Mappings

- packageURIFeatureTyping : PackageURIFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- MetadataUsage::name () : String [0..1]

```
'URI'
```

- MetadataUsage::ownedRelationship () : Relationship [0..*]

```
Set{packageURIFeatureTyping.to, PackageURIFeatureMembership_Mapping.getMapped(from)}
```

C.2.5.9.3.14 PackageURIFeatureMembership_Mapping

Description

The mapping class creates the feature membership relationship for the metadata feature to store the UML::Package::URI property.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Package

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
PackageURIMetadataReferenceUsage_Mapping.getMapped(from)
```

C.2.5.9.3.15 PackageURIFeatureTyping_Mapping

Description

The mapping class creates the FeatureTyping relationship for the AnnotatingFeature for the metadata to store the UML::Package::URI property.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Package

Mapping Target

FeatureTyping

Owned Mappings

- packageURIMetadataUsage : PackageURIMetadataUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
 let m: SysMLv2::Membership = SysMLv2::AttributeDefinition.allInstances()
->collect(dt | dt.owningRelationship)
->select(r | r.oclIsKindOf(SysMLv2::Membership))
->any(m | m.memberName =  'PackageData' ) in

if (m.oclIsUndefined()) then
          invalid
else
    m.memberElement
endif
```

- FeatureTyping::typedFeature () : Feature [1]

```
packageURIMetadataUsage.to
```

C.2.5.9.3.16 PackageURIMetadataReferenceUsage_Mapping

Description

The mapping class creates the MetadataFeature for the mapping of the property UML::Package::URI.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Package

Mapping Target

ReferenceUsage

Owned Mappings

- packageURIMetadataFeatureValue : PackageURIMetadataFeatureValue_Mapping
- packageURIRedefinition : PackageURIRedefinition_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
Set{packageURIRedefinition.to, packageURIMetadataFeatureValue.to}
```

C.2.5.9.3.17 PackageURIMetadataFeatureValue_Mapping

Description

The mapping class maps the value of the property UML::Package::URI.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Package

Mapping Target

FeatureValue

Owned Mappings

- packageURIMetadataReferenceUsage : PackageURIMetadataReferenceUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::featureWithValue () : Feature [1]

```
packageURIMetadataReferenceUsage.to
```

- FeatureValue::value () : Expression [1]

```
PackageURIValue_Mapping.getMapped(from)
```

C.2.5.9.3.18 PackageURIMetadataMembership_Mapping

Description

The mapping class creates a membership relationship for the metadata feature value for the UML::Package::URI property.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Package

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

```
 PackageURIMetadataUsage_Mapping.getMapped(from)
```

C.2.5.9.3.19 PackageURIRedefinition_Mapping

Description

The mapping class creates the redefinition of the attribute for the metadata UML::Package::URI.

General Mappings

GenericToRedefinition_Mapping

Mapping Source

Package

Mapping Target

Redefinition

Owned Mappings

- packageURIMetadataReferenceUsage : PackageURIMetadataReferenceUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Redefinition::redefinedFeature () : Feature [1]

```
 let m : SysMLv2::Membership = SysMLv2::AttributeUsage.allInstances()->collect(dt | dt.owning
if (m.oclIsUndefined()) then invalid else m.memberElement endif
```

- Redefinition::redefiningFeature () : Feature [1]

```
 packageURIMetadataReferenceUsage.to
```

C.2.5.9.3.20 PackageURIValue_Mapping

Description

The mapping class maps the value expression of the property UML::Package::URI.

General Mappings

GenericToExpression_Mapping

Mapping Source

Package

Mapping Target

LiteralString

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- LiteralString::value () : String [1]

  ```
  from.URI
  ```

C.2.5.9.3.21 Profile_Mapping

Description

*** not specified yet ***

General Mappings

Package_Mapping

Mapping Source

Profile

Mapping Target

Package

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Package::ownedRelationship () : Relationship [0..*]

```
Helper.packageOwnedRelationship(from)->including(ProfileMetadataMembership_Mapping.getMappec
```

C.2.5.9.3.22 ProfileMetadataMembership_Mapping

Description

The mapping class creates a membership relationship for the metadata feature value for the UML::Model::viewpoint property.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Profile

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

```
ProfileMetadataUsage_Mapping.getMapped(from)
```

C.2.5.9.3.23 ProfileMetadataUsage_Mapping

Description

The mapping class creates the annotating feature to annotate the generated Package element with metadata to store the UML::Model::viewpoint property.

General Mappings

GenericToMetadataUsage_Mapping

Mapping Source

Profile

Mapping Target

MetadataUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- MetadataUsage::name () : String [0..1]

```
'Profile'
```

C.2.5.9.3.24 StereotypeMetadataDefinition_Mapping

Description

*** not specified yet ***

General Mappings

Class_Mapping

Mapping Source

Stereotype

Mapping Target

MetadataDefinition

Owned Mappings

(none)

C.2.5.9.3.25 StereotypeMetadataDefinitionMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Stereotype* mapping.

General Mappings

ElementOwningMembership_Mapping

Mapping Source

Stereotype

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [0..1]

```
ElementMain_Mapping.getMapped(from)
```

C.2.5.9.3.26 StereotypeMetadataDefinitionReferenceUsage_Mapping

Description

Creates a reference usage for the *Stereotype* mapping.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Stereotype

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
Set{stereotypeMetadataDefinitionReferenceUsageRedefinition.to, stereotypeMetadataDefinitionR
```

C.2.5.9.3.27 StereotypeOccurenceUsage_Mapping

Description

*** not specified yet ***

General Mappings

GenericToOccurrenceUsage_Mapping

Mapping Source

Stereotype

Mapping Target

OccurrenceUsage

Owned Mappings

- stereotypeOccurenceUsageFeatureTyping : StereotypeOccurenceUsageFeatureTyping_Mapping
- stereotypeOccurenceUsageMultiplicityMembership : StereotypeOccurenceUsageMultiplicityMembership_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OccurrenceUsage::ownedRelationship () : Relationship [0..*]

```
Set{stereotypeOccurenceUsageFeatureTyping.to, stereotypeOccurenceUsageMultiplicityMembership
```

C.2.5.9.3.28 StereotypeOccurenceUsageFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *Stereotype* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Stereotype

Mapping Target

FeatureTyping

Owned Mappings

- stereotypeOccurenceUsage : StereotypeOccurenceUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
StereotypeOccurenceDefinition_Mapping.getMapped(from)
```

- FeatureTyping::typedFeature () : Feature [1]

```
stereotypeOccurenceUsage.to
```

C.2.5.9.3.29 StereotypeOccurenceUsageMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Stereotype* mapping.

General Mappings

GenericToMembership_Mapping

Mapping Source

Stereotype

Mapping Target

Membership

Owned Mappings

- stereotypeOccurenceUsage : StereotypeOccurenceUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Membership::memberName () : String [0..1]

```
from.name.substring(1,1).toLowerCase() + from.name.substring(2,from.name.size()) + 's'
```

- Membership::memberElement () : Element [1]

```
self.ownedMemberElement()
```

- Membership::ownedMemberElement () : Element [0..1]

```
stereotypeOccurenceUsage.to
```

C.2.5.9.3.30 StereotypeOccurenceUsageMultiplicityMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Stereotype* mapping.

General Mappings

GenericToMembership_Mapping

Mapping Source

Stereotype

Mapping Target

Membership

Owned Mappings

- stereotypeOccurenceUsageMultiplicityRange : StereotypeOccurenceUsageMultiplicityRange_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Membership::ownedMemberElement () : Element [0..1]

```
stereotypeOccurenceUsageMultiplicityRange.to
```

- Membership::memberElement () : Element [1]

```
self.ownedMemberElement()
```

C.2.5.9.3.31 StereotypeOccurenceUsageMultiplicityRange_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeature_Mapping

Mapping Source

Stereotype

Mapping Target

MultiplicityRange

Owned Mappings

- stereotypeOccurenceUsageMultiplicityRangeMembership : StereotypeOccurenceUsageMultiplicityRangeMembership_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- MultiplicityRange::ownedRelationship () : Relationship [0..*]

```
Set{stereotypeOccurenceUsageMultiplicityRangeMembership.to}
```

C.2.5.9.3.32 StereotypeOccurenceUsageMultiplicityRangeInfinity_Mapping

Description

*** not specified yet ***

General Mappings

GenericToExpression_Mapping

Mapping Source

Stereotype

Mapping Target

LiteralInfinity

Owned Mappings

- stereotypeOccurenceUsageMultiplicityRangeInfinityReturnParameterMembership : StereotypeOccurenceUsageMultiplicityRangeInfinityReturnParameterMembership_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- LiteralInfinity::ownedRelationship () : Relationship [0..*]

```
Set{stereotypeOccurenceUsageMultiplicityRangeInfinityReturnParameterMembership.to}
```

C.2.5.9.3.33 StereotypeOccurenceUsageMultiplicityRangeInfinityReturnParameter_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeature_Mapping

Mapping Source

Stereotype

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::direction () : FeatureDirectionKind [0..1]

```
SysMLv2::FeatureDirectionKind::out
```

C.2.5.9.3.34 StereotypeOccurenceUsageMultiplicityRangeInfinityReturnParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Stereotype* mapping.

General Mappings

GenericToReturnParameterMembership_Mapping

Mapping Source

Stereotype

Mapping Target

ReturnParameterMembership

Owned Mappings

- stereotypeOccurenceUsageMultiplicityRangeInfinityReturnParameter : StereotypeOccurenceUsageMultiplicityRangeInfinityReturnParameter_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReturnParameterMembership::ownedRelatedElement () : Element [0..*]

```
 let member: KerML::Element = self.ownedMemberParameter() in
if member.oclIsUndefined() then
    Set{}
else
    Set{self.ownedMemberParameter()}
endif
```

- ReturnParameterMembership::ownedMemberParameter () : Feature [0..1]

```
stereotypeOccurenceUsageMultiplicityRangeInfinityReturnParameter.to
```

- ReturnParameterMembership::memberParameter () : Feature [1]

```
self.ownedMemberParameter()
```

C.2.5.9.3.35 StereotypeOccurenceUsageMultiplicityRangeMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Stereotype* mapping.

General Mappings

GenericToMembership_Mapping

Mapping Source

Stereotype

Mapping Target

Membership

Owned Mappings

- stereotypeOccurenceUsageMultiplicityRangeInfinity : StereotypeOccurenceUsageMultiplicityRangeInfinity_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Membership::memberElement () : Element [1]

```
self.ownedMemberElement()
```

- Membership::ownedMemberElement () : Element [0..1]

```
stereotypeOccurenceUsageMultiplicityRangeInfinity.to
```

C.2.5.10 SimpleClassifiers

C.2.5.10.1 Overview

This chapter specifies the mapping of the metaclasses defined in the UML specification in the SimpleClassifiers chapter, which are part of the UML4SysML subset.

Table 28. List of all Overview Mapping Specfications

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
DataType	EnumerationDefinition AttributeDefinition AttributeDefinition	Enumeration_Mapping PrimitiveType_Mapping DataType_Mapping	
Enumeration	EnumerationDefinition	Enumeration_Mapping	
EnumerationLiteral	EnumerationUsage	EnumerationLiteral_Mapping	from.classifier->select(c | c.oclIsTypeOf(UML::Association))->size() = 0
Interface	PortDefinition	Interface_Mapping	
InterfaceRealization			
PrimitiveType	AttributeDefinition	PrimitiveType_Mapping	
Reception	ItemUsage	Reception_Mapping	
Signal	ItemDefinition	Signal_Mapping	

C.2.5.10.2 Mapping Specifications

C.2.5.10.2.1 Attribute_Mapping

Description

An UML::SimpleClassifiers::Property is mapped to a SysMLv2::Systems::Attributes::AttributeUsage.

General Mappings

PropertyCommon_Mapping
NamedElementMain_Mapping

Mapping Source

Property

Mapping Target

AttributeUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
if src.oclIsKindOf(UML::Property) then
    let p: UML::Property = src.oclAsType(UML::Property) in
    if p.type.oclIsUndefined() then
        false
    else
        p.type.oclIsKindOf(UML::DataType) and
        (p.association.oclIsUndefined() or p.association.ownedEnd->excludes(p))
    endif
else
    false
endif
```

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

C.2.5.10.2.2 AttributeRedefined_Mapping

Description

An UML::SimpleClassifiers::Property is mapped to a SysMLv2::Systems::Attributes::AttributeUsage.

General Mappings

PropertyCommon_Mapping

Mapping Source

Property

Mapping Target

ReferenceUsage

Owned Mappings

- attributeRedefinedFeatureTyping : AttributeRedefinedFeatureTyping_Mapping
- attributeRedefinedRedefinition : AttributeRedefinedRedefinition_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
let typing: KerML::FeatureTyping = attributeRedefinedFeatureTyping.to in
let subsetting: Set(KerML::Subsetting) = from.subsettedProperty->collect(p | PropertySubsetti
let subsettingMultiplicityTyping: Set(KerML::Relationship) = subsetting->union(Set{attributeR
    Set{MultiplicityMembership_Mapping.getMapped(from)}
else
    Set{MultiplicityMembership_Mapping.getMapped(from), typing}
endif)->asSet() in
if from.defaultValue.oclIsUndefined() then
    subsettingMultiplicityTyping
```

```
        else
            subsettingMultiplicityTyping->including(PropertyDefaultValue_Mapping.getMapped(from))
        endif
```

C.2.5.10.2.3 AttributeRedefinedRedefinition_Mapping

Description

Creates a redefinition relationship for the *redefiningFeature()* and the *redefinedFeature()* for the *Property* mapping.

General Mappings

GenericToRedefinition_Mapping

Mapping Source

Property

Mapping Target

Redefinition

Owned Mappings

- attributeRedefined : AttributeRedefined_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Redefinition::redefiningFeature () : Feature [1]

```
attributeRedefined.to
```

- Redefinition::redefinedFeature () : Feature [1]

```
from.redefinedProperty.get(0)
```

C.2.5.10.2.4 AttributeRedefinedMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *NamedElement* mapping.

General Mappings

ElementFeatureMembership_Mapping

Mapping Source

NamedElement

Mapping Target

FeatureMembership

Owned Mappings

- attributeRedefined : AttributeRedefined_Mapping

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
from.oclIsKindOf(UML::Property) and (from.oclAsType(UML::Property).redefinedElement->size() > 0)
```

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [0..1]

```
attributeRedefined.to
```

C.2.5.10.2.5 AttributeRedefinedFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *StructuralFeature* mapping.

General Mappings

StructuralFeatureToFeatureTyping_Mapping

Mapping Source

StructuralFeature

Mapping Target

FeatureTyping

Owned Mappings

- attributeRedefined : AttributeRedefined_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

```
attributeRedefined.to
```

C.2.5.10.2.6 BehavioredClassifier_Mapping

Description

The abstract mapping class BehavioredClassifier_Mapping maps the abstract metaclass UML::SimpleClassifiers::BehavioredClassifiers to a SysMLv2::Core::Classifiers::Classifier. The mapping class is used by concrete mapping classes, for example, Block_Mapping.

General Mappings

Classifier_Mapping

Mapping Source

BehavioredClassifier

Mapping Target

Classifier

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Classifier::ownedRelationship () : Relationship [0..*]

```
let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | (e.oclIsKindOf(UML::Prope
let redefinedAttributes: Set(UML::Element) = from.ownedElement->select(e | from.oclIsKindOf(U
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.oclIsKindOf(
let constraints : Set(UML::Constraint) = UML::Constraint.allInstances()->select( c | c.constr
let toElementOMS: Set(UML::Element) = (((from.ownedElement - toElementFMS) - redefinedAttribu
let relationships: Sequence(KerML::Relationship) =
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(constraints->collect(e | ConstrainedElementFeatureMembership_Mapping.getMapped(e)))
->union(redefinedAttributes->collect(e | AttributeRedefinedMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e))) in
if from.classifierBehavior.oclIsUndefined() then relationships else relationships->append(Cla
```

C.2.5.10.2.7 ClassifierBehaviorMembership_Mapping

Description

The ClassifierBehaviorMemberhship_Mapping class creates a membership relationship for a PerformActionUsage element to call the transformed SysML v1 classifier behavior.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

BehavioredClassifier

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [0..1]

```
BehavioredClassifierToPerformActionUsage_Mapping.getMapped(from)
```

C.2.5.10.2.8 BehavioredClassifierToFeatureTyping_Mapping

Description

The BehavioredClassifierToFeatureTyping_Mapping creates the relationship from the PerformActionUsage element to its type which is the transformed SysML v1 classifier behavior.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

BehavioredClassifier

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
from
```

C.2.5.10.2.9 BehavioredClassifierToPerformActionUsage_Mapping

Description

The BehavioredClassifierToPerformActionUsage_Mapping class creates a PerformActionUsage element to call the transformed SysML v1 classifier behavior.

General Mappings

GenericToFeature_Mapping

Mapping Source

BehavioredClassifier

Mapping Target

PerformActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- PerformActionUsage::isComposite () : Boolean [1]

```
true
```

- PerformActionUsage::ownedRelationship () : Relationship [0..*]

```
Set{BehavioredClassifierToFeatureTyping_Mapping.getMapped(from)}
```

- PerformActionUsage::name () : String [0..1]

```
'classifierBehavior'
```

C.2.5.10.2.10 DataType_Mapping

Description

A UML::SimpleClassifiers::DataType is mapped to a SysMLv2::Systems::Attributes::AttributeDefinition. The mapping also cover the transformation of UML4SysML::PrimitiveType elements.

General Mappings

Classifier_Mapping

Mapping Source

DataType

Mapping Target

AttributeDefinition

Owned Mappings

(none)

C.2.5.10.2.11 Enumeration_Mapping

Description

A UML4SysML::Enumeration is mapped to a SysMLv2::EnumerationDefinition.

General Mappings

DataType_Mapping

Mapping Source

Enumeration

Mapping Target

EnumerationDefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- EnumerationDefinition::isVariation () : Boolean [1]

  ```
   true
  ```

- EnumerationDefinition::ownedRelationship () : Relationship [0..*]

  ```
  let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.oclIsKindOf(
  let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | e.oclIsKindOf(UML::Proper
  let literals: Set(UML::Element) = from.ownedElement->select(e | e.oclIsKindOf(UML::Enumeratic
  let toElementOMS: Set(UML::Element) = (((from.ownedElement - toElementFMS) - generalizations)
  toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
  ->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
  ```

```
        ->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))
        ->union(literals->collect(e | EnumerationVariantMembership_Mapping.getMapped(e)))
```

C.2.5.10.2.12 EnumerationLiteral_Mapping

Description

A UML4SysML::EnumerationLiteral is mapped to a SysMLv2::EnumerationUsage.

General Mappings

GenericToFeature_Mapping
InstanceSpecification_Mapping

Mapping Source

EnumerationLiteral

Mapping Target

EnumerationUsage

Owned Mappings

(none)

C.2.5.10.2.13 EnumerationVariantMembership_Mapping

Description

The EnumerationVariantMembership_Mapping class creates the variant membership relationship between the enumeration definition and a enumeration usage.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

EnumerationLiteral

Mapping Target

VariantMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- VariantMembership::ownedMemberElement () : Element [1]

  ```
  from
  ```

C.2.5.10.2.14 Interface_Mapping

Description

A UML4SysML::Interface is mapped to a SysMLv2::PortDefinition. The mapping also includes the generation of an appropriate ConjugatedPortDefinition. That mappings is performed by the mapping classes InterfaceConjugatedPortDefinitionMembership_Mapping, InterfacePortConjugation_Mapping, and InterfaceConjugatedPortDefinition_Mapping.

General Mappings

GenericToPortDefinition_Mapping
Classifier_Mapping

Mapping Source

Interface

Mapping Target

PortDefinition

Owned Mappings

- conjugatedPortDefinitionMembership : InterfaceConjugatedPortDefinitionMembership_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- PortDefinition::ownedRelationship () : Relationship [0..*]

  ```
  let properties: Set(UML::Element) = from.ownedElement->select(e | e.oclIsKindOf(UML::Property
  let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.oclIsKindOf(
  let elements: Set(UML::Element) = (from.ownedElement - properties) - generalizations  in
  elements->collect(e | ElementOwningMembership_Mapping.getMapped(e))
  ->union(properties->collect(e | PropertyMembership_Mapping.getMapped(e)))
  ->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))
  ->append(conjugatedPortDefinitionMembership)
  ```

C.2.5.10.2.15 InterfaceConjugatedPortDefinition_Mapping

Description

As part of the mapping from a UML4SysML::Interface to a SysMLv2::PortDefinition, this mapping class is used to create the appropriate ConjugatedPortDefinition.

General Mappings

GenericToPortDefinition_Mapping

Mapping Source

Interface

Mapping Target

ConjugatedPortDefinition

Owned Mappings

- portConjugation : InterfacePortConjugation_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ConjugatedPortDefinition::ownedRelationship () : Relationship [0..*]

  ```
  Set{portConjugation}
  ```

- ConjugatedPortDefinition::name () : String [0..1]

  ```
  '~'+from.name
  ```

C.2.5.10.2.16 InterfaceConjugatedPortDefinitionMembership_Mapping

Description

As part of the mapping from a UML4SysML::Interface to a SysMLv2::PortDefinition, this mapping class is used to create the membership relationship for the ConjugatedPortDefinition.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Interface

Mapping Target

OwningMembership

Owned Mappings

- conjugatedPortDefinitionMapping : InterfaceConjugatedPortDefinition_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

  ```
  conjugatedPortDefinitionMapping.to
  ```

- OwningMembership::ownedRelationship () : Relationship [0..*]

  ```
  Set{portConjugation}
  ```

C.2.5.10.2.17 InterfacePortConjugation_Mapping

Description

As part of the mapping from a UML4SysML::Interface to a SysMLv2::PortDefinition, this mapping class is used to create the appropriate PortConjugation relationship.

General Mappings

GenericToRelationship_Mapping

Mapping Source

Interface

Mapping Target

PortConjugation

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- PortConjugation::originalPortDefinition () : PortDefinition [1]

  ```
  from
  ```

- PortConjugation::conjugatedType () : Type [1]

  ```
  SysMLv2::ConjugatedPortDefinition.allInstances()->collect(cpd | cpd.owningRelationship)->sel
  ```

C.2.5.10.2.18 InterfaceRealization_Mapping

Description

A UML4SysML::InterfaceRealization is mapped to a SysMLv2::Superclassing.

General Mappings

GenericToSpecialization_Mapping

Mapping Source

InterfaceRealization

Mapping Target

Subclassification

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Subclassification::subclassifier () : Type [1]

  ```
  Classifier_Mapping.getMapped(from.specific)
  ```

- Subclassification::superclassifier () : Type [1]

  ```
  Classifier_Mapping.getMapped(from.general)
  ```

C.2.5.10.2.19 PrimitiveType_Mapping

Description

The PrimitiveType_Mapping class maps a UML4SysML::PrimitiveType to a SysML v2 AttributeDefinition.

General Mappings

DataType_Mapping

Mapping Source

PrimitiveType

Mapping Target

AttributeDefinition

Owned Mappings

(none)

C.2.5.10.2.20 Reception_Mapping

Description

A UML4SysML::Reception is mapped to a SysMLv2::AttributeUsage with feature direction "in".

General Mappings

BehavioralFeature_Mapping

Mapping Source

Reception

Mapping Target

ItemUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ItemUsage::direction () : FeatureDirectionKind [0..1]

```
SysMLv2::FeatureDirectionKind::in
```

- ItemUsage::ownedRelationship () : Relationship [0..*]

```
Set{ReceptionToFeatureTyping_Mapping.getMapped(from)}
```

C.2.5.10.2.21 ReceptionToFeatureTyping_Mapping

Description

A UML4SysML::Reception is mapped to SysMLv2::AttributeUsage. The ReceptionToFeatureTyping_Mapping class creates the type of the AttributeUsage which is the Signal of the Reception.

General Mappings

TypedElementToFeatureTyping_Mapping

Mapping Source

Reception

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

  ```
  Reception_Mapping.getMapped(from)
  ```

- FeatureTyping::type () : Type [1]

  ```
  Classifier_Mapping.getMapped(from.signal)
  ```

C.2.5.10.2.22 Signal_Mapping

Description

A UML4SysML::Signal is mapped to a SysMLv2::AttributeDefinition.

General Mappings

Classifier_Mapping

Mapping Source

Signal

Mapping Target

ItemDefinition

Owned Mappings

(none)

C.2.5.11 StructuredClassifiers

C.2.5.11.1 Overview

Table 29. List of all Overview Mapping Specfications

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
Association	ConnectionDefinition ConnectionDefinition ConnectionDefinition ConnectionDefinition	ConnectorType_Mapping AssociationClass_Mapping ConnectorTypeDerived_Mapping AssociationBlock_Mapping	let this: UML::Association = src.oclAsType(UML::Association) in if this.oclIsUndefined() then false else not from.memberEnd->exists(m | m.type.oclIsKindOf(UML::UseCase)) and not this.isDerived and not this.oclIsTypeOf(UML::AssociationClass) and Helper.isConnectionDef(this) endif not Helper.hasStereotypeApplied(from, 'SysML::Blocks::Block') (from.memberEnd->select(m | m.type.oclIsKindOf(UML::UseCase))->isEmpty()) and (let this: UML::Association = src.oclAsType(UML::Association) in if this.oclIsUndefined() then false else this.isDerived and not this.oclIsTypeOf(UML::AssociationClass) and Helper.isConnectionDef(this) endif) Helper.hasStereotypeApplied(from, 'SysML::Blocks::Block')
AssociationClass	ConnectionDefinition ConnectionDefinition	AssociationClass_Mapping AssociationBlock_Mapping	not Helper.hasStereotypeApplied(from, 'SysML::Blocks::Block') Helper.hasStereotypeApplied(from, 'SysML::Blocks::Block')

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter

Class	PartDefinition
	OccurrenceDefinition
	PartDefinition
	VerificationCaseDefinition
	ActionUsage
	PartDefinition
	ViewpointDefinition
	ActionDefinition
	PortDefinition
	ConstraintDefinition
	ActionDefinition
	MetadataDefinition
	ActionUsage
	StateDefinition
		Block_Mapping
		Class_Mapping
		Stakeholder_Mapping
		TestCaseActivity_Mapping
		ActivityAsUsage_Mapping
		EncapsulatedBlock_Mapping
		Viewpoint_Mapping
		ActivityAsDefinition_Mapping
		InterfaceBlock_Mapping
		ConstraintBlock_Mapping
		OpaqueBehaviorAsDefinition_Mapping
		StereotypeMetadataDefinition_Mapping
		OpaqueBehaviorAsUsage_Mapping
		StateDefinition_Mapping
			not from.oclIsTypeOf(UML::AssociationClass) and Helper.hasStereotypeApplied(src, 'SysML::Blocks::Block') and not Helper.hasStereotypeApplied(src, 'SysML::ConstraintBlocks::ConstraintBlock') and not Helper.hasStereotypeApplied(src, 'SysML::Ports&Flows::InterfaceBlock')
			
			not Helper.hasStereotypeApplied(from, 'SysML::Requirements::Requirement') and not from.oclIsTypeOf(UML::AssociationClass)
			Helper.hasStereotypeApplied(from, 'SysML::ModelElements::Stakeholder')
			Helper.hasStereotypeApplied(from, 'SysML::Requirements::TestCase')
			(not from.owner.oclIsKindOf(UML::Package)) and (not Helper.hasStereotypeApplied(from, 'SysML::Requirements::TestCase'))
			not from.oclIsTypeOf(UML::AssociationClass) and Helper.hasStereotypeApplied(src, 'SysML::Blocks::Block') and not Helper.hasStereotypeApplied(src, 'SysML::ConstraintBlocks::ConstraintBlock') and not Helper.hasStereotypeApplied(src, 'SysML::Ports&Flows::InterfaceBlock') and Helper.getTagValue(src, 'SysML::Blocks::Block', 'isEncapsulated')
			Helper.hasStereotypeApplied(from, 'SysML::ModelElements::Viewpoint')
			from.owner.oclIsKindOf(UML::Package)
			Helper.hasStereotypeApplied(from, 'SysML::Ports&Flows::InterfaceBlock')
			Helper.hasStereotypeApplied(from, 'SysML::ConstraintBlocks::ConstraintBlock')
			src.owner.oclIsKindOf(UML::Package)
			not Helper.hasStereotypeApplied(from,

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
			'SysML::Requirements::Requirement') and not from.oclIsTypeOf(UML::AssociationClass) not src.owner.oclIsKindOf(UML::Package) src.owner.oclIsKindOf(UML::Package)
Connector	ConnectionUsage BindingConnectorAsUsage	Connector_Mapping BindingConnector_Mapping	Helper.hasStereotypeApplied(from, 'SysML::Blocks::BindingConnector')
ConnectorEnd	Feature	ConnectorEndToOwnedFeature_Mapping	

Port	PortUsage
	PortUsage
	PartUsage
	PartUsage
		PortUntyped_Mapping
		Port_Mapping
		FullPortUntyped_Mapping
		FullPort_Mapping
			from.type.oclIsUndefined()
			if src.oclIsKindOf(UML::Property) then let p: UML::Property = src.oclAsType(UML::Property) in if p.type.oclIsUndefined() then false else not p.type.oclIsKindOf(UML::DataType) and not (p.name.indexOf('base_') > 0) and (p.association.oclIsUndefined() or p.association.ownedEnd->excludes(p)) endif else false endif
			from.type.oclIsUndefined() and Helper.hasStereotypeApplied(src, 'SysML::Ports&Flows::FullPort')
			(not from.type.oclIsUndefined()) and Helper.hasStereotypeApplied(src, 'SysML::Ports&Flows::FullPort')

C.2.5.11.2 Mapping Specifications

C.2.5.11.2.1 AssociationCommon_Mapping

Description

A UML4SysML::Association is mapped to a SysMLv2::ConnectionDefinition. The UML4SysML::Association::isDerived property is not supported in SysML v2. To preserve the information, it is stored in a metadata annotation.

General Mappings

Classifier_Mapping
Relationship_Mapping

Mapping Source

Association

Mapping Target

Association

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
from.memberEnd->select( m | m.type.oclIsKindOf(UML::UseCase))->isEmpty()
```

Mapping rules

The following lists the mapping rules for the target element properties.

- Association::ownedRelationship () : Relationship [0..*]

```
let nonOwnedEnds: OrderedSet(UML::Property) = (from.memberEnd-from.ownedEnd)->asOrderedSet()
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.oclIsKindOf(
let others: OrderedSet(UML::Element) = ((from.ownedElement-from.memberEnd)-generalizations)->
nonOwnedEnds->collect(e | NonOwnedEndMembership_Mapping.getMapped(e))
->union(from.ownedEnd->collect(e | OwnedEndMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))
->union(others->collect(e | ElementOwningMembership_Mapping.getMapped(e)))
->asOrderedSet()
```

C.2.5.11.2.2 AssociationClass_Mapping

Description

*** not specified yet ***

General Mappings

AssociationCommon_Mapping

Mapping Source

AssociationClass

Mapping Target

ConnectionDefinition

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
 not Helper.hasStereotypeApplied(from, 'SysML::Blocks::Block')
```

Mapping rules

The following lists the mapping rules for the target element properties.

- ConnectionDefinition::ownedRelationship () : Relationship [0..*]

```
let nonOwnedEnds: OrderedSet(UML::Property) = (from.memberEnd-from.ownedEnd)->asOrderedSet()
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.oclIsKindOf(
let others: OrderedSet(UML::Element) = ((from.ownedElement-from.memberEnd)-generalizations)->
nonOwnedEnds->collect(e | NonOwnedEndMembership_Mapping.getMapped(e))
->union(from.ownedEnd->collect(e | OwnedEndMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))
->union(others->collect(e | ElementOwningMembership_Mapping.getMapped(e)))
->asOrderedSet()
```

C.2.5.11.2.3 AssociationToAnnotation_Mapping

Description

*** not specified yet ***

General Mappings

GenericToAnnotation_Mapping

Mapping Source

Association

Mapping Target

Annotation

Owned Mappings

(none)

C.2.5.11.2.4 AssociationToAnnotatingFeature_Mapping

Description

*** not specified yet ***

General Mappings

GenericToAnnotatingElement_Mapping

Mapping Source

Association

Mapping Target

MetadataFeature

Owned Mappings

- associationToAnnotation : AssociationToAnnotation_Mapping
- associationToFeatureMembership : AssociationToFeatureMembership_Mapping
- associationToFeatureTyping : AssociationToFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- MetadataFeature::name () : String [0..1]

  ```
  'isDerived'
  ```

- MetadataFeature::ownedRelationship () : Relationship [0..*]

  ```
  Set{associationToFeatureMembership.to, associationToAnnotation.to, associationToFeatureTypin
  ```

C.2.5.11.2.5 AssociationToFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *Association* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Association

Mapping Target

FeatureMembership

Owned Mappings

- associationToMetadataFeature : AssociationToMetadataFeature_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
 self.associationToMetadataFeatureValue.to
```

C.2.5.11.2.6 AssociationToFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *Association* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Association

Mapping Target

FeatureTyping

Owned Mappings

- associationToAnnotatingFeature : AssociationToAnnotatingFeature_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

```
 self.associationToAnnotatingFeature.to
```

- FeatureTyping::type () : Type [1]

```
 let m : SYSML2::Membership = SYSML2::AttributeDefinition.allInstances()->collect(dt | dt.owr
if (m.oclIsUndefined()) then
       OclUndefined
else
       m.memberElement
endif
```

C.2.5.11.2.7 AssociationToMetadataFeature_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeature_Mapping

Mapping Source

Association

Mapping Target

Feature

Owned Mappings

- associationToMetadataFeatureValue : AssociationToMetadataFeatureValue_Mapping
- associationToRedefinition : AssociationToRedefinition_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::ownedRelationship () : Relationship [0..*]

```
Set{self.associationToRedefinition.to, self.associationToMetadataFeatureValue.to}
```

C.2.5.11.2.8 AssociationToMetadataFeatureValue_Mapping

Description

Creates a feature value relationship for the mapping class *Association*

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Association

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

```
Helper.mappedValueSpecification(from.isDerived)
```

- FeatureValue::ownedMemberElement () : Element [1]

```
Helper.getScalarValueTypeByName('Boolean')
```

C.2.5.11.2.9 AssociationToMetadataMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Association* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Association

Mapping Target

FeatureMembership

Owned Mappings

- associationToAnnotatingFeature : AssociationToAnnotatingFeature_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
self.associationToAnnotatingFeature.to
```

C.2.5.11.2.10 AssociationToRedefinition_Mapping

Description

Creates a redefinition relationship for the *redefiningFeature()* and the *redefinedFeature()* for the *Association* mapping.

General Mappings

GenericToRedefinition_Mapping

Mapping Source

Association

Mapping Target

Redefinition

Owned Mappings

- associationToMetadataFeature : AssociationToMetadataFeature_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Redefinition::redefiningFeature () : Feature [1]

```
 self.associationToMetadataFeatureValue.to
```

- Redefinition::redefinedFeature () : Feature [1]

```
 let m : SYSML2::Membership = SYSML2::AttributeUsage.allInstances()->collect(dt | dt.owningRe
if (m.oclIsUndefined()) then
        OclUndefined
else
        m.memberElement
endif
```

C.2.5.11.2.11 BehavioredClassifier_Mapping

Description

The abstract mapping class BehavioredClassifier_Mapping maps the abstract metaclass UML::SimpleClassifiers::BehavioredClassifiers to a SysMLv2::Core::Classifiers::Classifier. The mapping class is used by concrete mapping classes, for example, Block_Mapping.

General Mappings

Classifier_Mapping

Mapping Source

BehavioredClassifier

Mapping Target

Classifier

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Classifier::ownedRelationship () : Relationship [0..*]

```
let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | (e.oclIsKindOf(UML::Prope
let redefinedAttributes: Set(UML::Element) = from.ownedElement->select(e | from.oclIsKindOf(U
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.oclIsKindOf(
let constraints : Set(UML::Constraint) = UML::Constraint.allInstances()->select( c | c.constr
let toElementOMS: Set(UML::Element) = (((from.ownedElement - toElementFMS) - redefinedAttribu
let relationships: Sequence(KerML::Relationship) =
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(constraints->collect(e | ConstrainedElementFeatureMembership_Mapping.getMapped(e)))
->union(redefinedAttributes->collect(e | AttributeRedefinedMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e))) in
if from.classifierBehavior.oclIsUndefined() then relationships else relationships->append(Cla
```

C.2.5.11.2.12 BehavioredClassifierToFeatureTyping_Mapping

Description

The BehavioredClassifierToFeatureTyping_Mapping creates the relationship from the PerformActionUsage element to its type which is the transformed SysML v1 classifier behavior.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

BehavioredClassifier

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
from
```

C.2.5.11.2.13 BehavioredClassifierToPerformActionUsage_Mapping

Description

The BehavioredClassifierToPerformActionUsage_Mapping class creates a PerformActionUsage element to call the transformed SysML v1 classifier behavior.

General Mappings

GenericToFeature_Mapping

Mapping Source

BehavioredClassifier

Mapping Target

PerformActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- PerformActionUsage::isComposite () : Boolean [1]

  ```
  true
  ```

- PerformActionUsage::ownedRelationship () : Relationship [0..*]

  ```
  Set{BehavioredClassifierToFeatureTyping_Mapping.getMapped(from)}
  ```

- PerformActionUsage::name () : String [0..1]

  ```
  'classifierBehavior'
  ```

C.2.5.11.2.14 Class_Mapping

Description

*** not specified yet ***

General Mappings

BehavioredClassifier_Mapping

Mapping Source

Class

Mapping Target

OccurrenceDefinition

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
 not Helper.hasStereotypeApplied(from, 'SysML::Requirements::Requirement') and not from.oclIsTypeOf(
```

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

C.2.5.11.2.15 ClassifierBehaviorMembership_Mapping

Description

The ClassifierBehaviorMemberhship_Mapping class creates a membership relationship for a PerformActionUsage element to call the transformed SysML v1 classifier behavior.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

BehavioredClassifier

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [0..1]

  ```
  BehavioredClassifierToPerformActionUsage_Mapping.getMapped(from)
  ```

C.2.5.11.2.16 ConnectionEndToSubsetting_Mapping

Description

Creates a subsetting relationship for the *subsettingFeature()* and the *subsettedFeature()* for the *ConnectorEnd* mapping.

General Mappings

GenericToSubsetting_Mapping

Mapping Source

ConnectorEnd

Mapping Target

Subsetting

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Subsetting::subsettedFeature () : Feature [1]

```
 let propertyPath: OrderedSet(UML::Property) = Helper.getTagValueAsElementColl(src, 'SysML::
if propertyPath->isEmpty() then
    ElementMain_Mapping.getMapped(from.role)
else
    ConnectorEndToSubsettedFeature_Mapping.getMapped(from)
endif
```

- Subsetting::ownedRelationship () : Relationship [0..*]

```
 let propertyPath: OrderedSet(UML::Property) = Helper.getTagValueAsElementColl(from, 'SysML::
if propertyPath->notEmpty() then
        OrderedSet{ConnectorEndToSubsettedFeatureMembership_Mapping.getMapped(from)}
else
        OrderedSet{}
endif
```

- Subsetting::subsettingFeature () : Feature [1]

```
 ConnectorEndToOwnedFeature_Mapping.getMapped(from)
```

C.2.5.11.2.17 Connector_Mapping

Description

*** not specified yet ***

General Mappings

NamedElementMain_Mapping
GenericToConnector_Mapping

Mapping Source

Connector

Mapping Target

ConnectionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ConnectionUsage::ownedRelationship () : Relationship [0..*]

```
from.end->collect(e | ConnectorEndToMembership_Mapping.getMapped(e))
   ->including(ConnectorMultiplicityMembership_Mapping.getMapped(from))
```

C.2.5.11.2.18 ConnectorEndToFeatureCommon_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeature_Mapping

Mapping Source

ConnectorEnd

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isOrdered () : Boolean [1]

```
from.isOrdered
```

C.2.5.11.2.19 ConnectorEndToMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *ConnectorEnd* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ConnectorEnd

Mapping Target

EndFeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- EndFeatureMembership::ownedMemberFeature () : Feature [1]

```
ConnectorEndToOwnedFeature_Mapping.getMapped(from)
```

C.2.5.11.2.20 ConnectorEndToOwnedFeature_Mapping

Description

*** not specified yet ***

General Mappings

ConnectorEndToFeatureCommon_Mapping
ElementMain_Mapping

Mapping Source

ConnectorEnd

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::ownedRelationship () : Relationship [0..*]

```
 let subsetting: KerML::Subsetting = ConnectionEndToSubsetting_Mapping.getMapped(from) in
if subsetting.oclIsUndefined() then
    OrderedSet{MultiplicityMembership_Mapping.getMapped(from)}
else
    OrderedSet{MultiplicityMembership_Mapping.getMapped(from), subsetting}
endif
```

C.2.5.11.2.21 ConnectorEndToSubsettedFeature_Mapping

Description

*** not specified yet ***

General Mappings

ConnectorEndToFeatureCommon_Mapping

Mapping Source

ConnectorEnd

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
let propertyPath: OrderedSet(UML::Property) = Helper.getTagValueAsElementColl(src, 'SysML::Blocks::N
propertyPath->notEmpty()
```

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::ownedRelationship () : Relationship [0..*]

```
 let propertyPath: OrderedSet(UML::Property) = Helper.getTagValueAsElementColl(from, 'SysML::
let chain: OrderedSet(KerML::FeatureChaining) = propertyPath->collect(p | PropertyToFeatureCh
->asOrderedSet()->including(PropertyToFeatureChaining_Mapping.getMapped(from.role)) in
chain->union(OrderedSet{MultiplicityMembership_Mapping.getMapped(from)})
```

- Feature::name () : String [0..1]

```
'featureChain'
```

C.2.5.11.2.22 ConnectorEndToSubsettedFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *ConnectorEnd* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ConnectorEnd

Mapping Target

EndFeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- EndFeatureMembership::ownedMemberFeature () : Feature [1]

```
ConnectorEndToSubsettedFeature_Mapping.getMapped(from)
```

C.2.5.11.2.23 ConnectorMultiplicityMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Connector* mapping.

General Mappings

DefaultMultiplicityMembership_Mapping

Mapping Source

Connector

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OwningMembership::memberName () : String [0..1]

```
from.name+'_Connector_multiplicity'
```

C.2.5.11.2.24 ConnectorType_Mapping

Description

*** not specified yet ***

General Mappings

AssociationCommon_Mapping

Mapping Source

Association

Mapping Target

ConnectionDefinition

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
let this: UML::Association = src.oclAsType(UML::Association) in
if this.oclIsUndefined() then
    false
else
    not from.memberEnd->exists( m | m.type.oclIsKindOf(UML::UseCase)) and
    not this.isDerived and
    not this.oclIsTypeOf(UML::AssociationClass) and
    Helper.isConnectionDef(this)
endif
```

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

C.2.5.11.2.25 ConnectorTypeDerived_Mapping

Description

*** not specified yet ***

General Mappings

AssociationCommon_Mapping

Mapping Source

Association

Mapping Target

ConnectionDefinition

Owned Mappings

- associationToMetadataMembership : AssociationToMetadataMembership_Mapping

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
(from.memberEnd->select( m | m.type.oclIsKindOf(UML::UseCase))->isEmpty()) and
(let this: UML::Association = src.oclAsType(UML::Association) in
if this.oclIsUndefined() then
    false
else
    this.isDerived and
    not this.oclIsTypeOf(UML::AssociationClass) and
    Helper.isConnectionDef(this)
endif)
```

Mapping rules

The following lists the mapping rules for the target element properties.

- ConnectionDefinition::ownedRelationship () : Relationship [0..*]

```
let nonOwnedEnds: OrderedSet(UML::Property) = (from.memberEnd-from.ownedEnd)->asOrderedSet()
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.oclIsKindOf(
let others: OrderedSet(UML::Element) = ((from.ownedElement-from.memberEnd)-generalizations)->
nonOwnedEnds->collect(e | NonOwnedEndMembership_Mapping.getMapped(e))
->union(from.ownedEnd->collect(e | OwnedEndMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))
->union(others->collect(e | ElementOwningMembership_Mapping.getMapped(e)))
->asOrderedSet()
->append(self.associationToMetadataMembership.to)
```

C.2.5.11.2.26 End_Mapping

Description

*** not specified yet ***

General Mappings

PropertyCommon_Mapping

Mapping Source

Property

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
src.oclIsKindOf(UML::Property) and not src.oclAsType(UML::Property).association.oclIsUndefined()
```

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isEnd () : Boolean [1]

  ```
  true
  ```

C.2.5.11.2.27 EndMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Property* mapping.

General Mappings

StructuralFeatureMembership_Mapping

Mapping Source

Property

Mapping Target

EndFeatureMembership

Owned Mappings

(none)

C.2.5.11.2.28 NonOwnedEndSubsetting_Mapping

Description

Creates a subsetting relationship for the *subsettingFeature()* and the *subsettedFeature()* for the *Property* mapping.

General Mappings

GenericToSubsetting_Mapping

Mapping Source

Property

Mapping Target

Subsetting

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Subsetting::subsettedFeature () : Feature [1]

  ```
  Property_Mapping.getMapped(from)
  ```

C.2.5.11.2.29 EndToSubsettedFeature_Mapping

Description

*** not specified yet ***

General Mappings

PropertyCommon_Mapping

Mapping Source

Property

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
let property: UML::Property = src.oclAsType(UML::Property) in
not property.association.oclIsUndefined()
and property.association.ownedEnd->excludes(property)
```

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::ownedRelationship () : Relationship [0..*]

  ```
   let chain: OrderedSet(KerML::FeatureChaining) = OrderedSet{EndToSubsettedFeatureChaining_Map
  chain->including(MultiplicityMembership_Mapping.getMapped(from))
  ```

C.2.5.11.2.30 EndToSubsettedFeatureChaining_Mapping

Description

*** not specified yet ***

General Mappings

GenericToRelationship_Mapping

Mapping Source

Property

Mapping Target

FeatureChaining

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureChaining::name () : String [0..1]

  ```
  'featureChain'
  ```

- FeatureChaining::chainingFeature () : Feature [1]

  ```
  Property_Mapping.getMapped(from)
  ```

C.2.5.11.2.31 NonOnedEndToSubsettedFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *Property* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Property

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
src.oclIsKindOf(UML::Property) and not src.oclAsType(UML::Property).association.oclIsUndefined()
```

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

  ```
  EndToSubsettedFeature_Mapping.getMapped(from)
  ```

C.2.5.11.2.32 NonOwnedEnd_Mapping

Description

*** not specified yet ***

General Mappings

End_Mapping

Mapping Source

Property

Mapping Target

Feature

Owned Mappings

- nonOwnedEndTyping : NonOwnedEndTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::ownedRelationship () : Relationship [0..*]

```
Set{MultiplicityMembership_Mapping.getMapped(from)
   ,nonOwnedEndTyping.to
   ,NonOwnedEndSubsettingMembership_Mapping.getMapped(from)
   ,NonOwnedEndToSubsettedFeatureMembership_Mapping.getMapped(from)}
   ->union(from.qualifier->collect(q | ElementFeatureMembership_Mapping.getMapped(q))->asSet
```

- Feature::name () : String [0..1]

```
'nonOwnedEnd'
```

C.2.5.11.2.33 NonOwnedEndMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Property* mapping.

General Mappings

EndMembership_Mapping

Mapping Source

Property

Mapping Target

EndFeatureMembership

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
src.oclIsKindOf(UML::Property)
    and not src.oclAsType(UML::Property).association.oclIsUndefined()
    and src.oclAsType(UML::Property).association.ownedEnd->excludes(src)
```

Mapping rules

The following lists the mapping rules for the target element properties.

- EndFeatureMembership::ownedMemberFeature () : Feature [1]

```
NonOwnedEnd_Mapping.getMapped(from)
```

C.2.5.11.2.34 NonOwnedEndSubsettingMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Property* mapping.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Property

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

  ```
  NonOwnedEndSubsetting_Mapping.getMapped(from)
  ```

C.2.5.11.2.35 NonOwnedEndTyping_Mapping

Description

*** not specified yet ***

General Mappings

StructuralFeatureToFeatureTyping_Mapping

Mapping Source

Property

Mapping Target

FeatureTyping

Owned Mappings

- nonOwnedEnd : NonOwnedEnd_Mapping

C.2.5.11.2.36 OwnedEnd_Mapping

Description

*** not specified yet ***

General Mappings

End_Mapping
NamedElementMain_Mapping

Mapping Source

Property

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
let p: UML::Property = src.oclAsType(UML::Property) in
not p.oclIsUndefined() and
(not p.association.oclIsUndefined() and p.association.ownedEnd->includes(p)) and
(not p.association.memberEnd->select( m | (not m.type.oclIsUndefined()) and m.type.oclIsTypeOf(UML::
```

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::ownedRelationship () : Relationship [0..*]

```
let qualifiers: Set(KerML::FeatureMembership) = from.qualifier->collect(q | ElementFeatureMem
let typing: KerML::FeatureTyping = StructuralFeatureToFeatureTyping_Mapping.getMapped(from) i
let subsetting: Set(KerML::Subsetting) = from.subsettedProperty->collect(p | PropertySubsetti
let subsettingMultiplicityTyping: Set(KerML::Relationship) = subsetting->union(if typing.oclI
    Set{MultiplicityMembership_Mapping.getMapped(from)}
else
    Set{MultiplicityMembership_Mapping.getMapped(from), typing}
endif)->asSet() in
let relationships: Set(KerML::Relationship) = qualifiers->union(
    if from.defaultValue.oclIsTypeOf(UML::OpaqueExpression) then
        subsettingMultiplicityTyping->including(ElementOwningMembership_Mapping.getMapped(fro
    else
        subsettingMultiplicityTyping
    endif) in

if from.defaultValue.oclIsUndefined() then
    relationships
else
    relationships->including(if from.defaultValue.oclIsTypeOf(UML::OpaqueExpression) then Def
endif
```

C.2.5.11.2.37 OwnedEndMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Property* mapping.

General Mappings

EndMembership_Mapping

Mapping Source

Property

Mapping Target

EndFeatureMembership

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
src.oclIsKindOf(UML::Property)
    and not src.oclAsType(UML::Property).association.oclIsUndefined()
    and src.oclAsType(UML::Property).association.ownedEnd->includes(src)
```

Mapping rules

The following lists the mapping rules for the target element properties.

- EndFeatureMembership::ownedMemberFeature () : Feature [1]

  ```
  OwnedEnd_Mapping.getMapped(from)
  ```

C.2.5.11.2.38 Port_Mapping

Description

A port which is untyped or typed by an interface block is mapped to a SysMLv2::PortUsage. The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
port port1 : sysMLv1InterfaceBlock;
```

General Mappings

Property_Mapping

Mapping Source

Port

Mapping Target

PortUsage

Owned Mappings

(none)

C.2.5.11.2.39 PortUntyped_Mapping

Description

*** not specified yet ***

General Mappings

PropertyUntyped_Mapping

Mapping Source

Port

Mapping Target

PortUsage

Owned Mappings

(none)

C.2.5.11.2.40 PropertyToFeatureChaining_Mapping

Description

*** not specified yet ***

General Mappings

GenericToRelationship_Mapping

Mapping Source

Property

Mapping Target

FeatureChaining

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureChaining::chainingFeature () : Feature [1]

```
ElementMain_Mapping.getMapped(from)
```

C.2.5.11.2.41 QualifierMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *StructuralFeature* mapping.

General Mappings

StructuralFeatureMembership_Mapping

Mapping Source

StructuralFeature

Mapping Target

FeatureMembership

Owned Mappings

(none)

C.2.5.12 UseCases

C.2.5.12.1 Overview

Table 30. List of all Overview Mapping Specfications

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
Actor	ItemDefinition	Actor_Mapping	
Extend			
ExtensionPoint			
Include			
UseCase	UseCaseDefinition	UseCase_Mapping	

C.2.5.12.2 SysML v1 UseCases elements not mapped

Table 31. List of SysML v1 elements not mapped of this section

SysML v1 Concept	Rationale
Extend	The semantics of the UML4SysML::Extend relationship is not supported by SysML v2.
ExtensionPoint	The semantics of the UML4SysML::Extend relationship is not supported by SysML v2 Therefore, UML4SysML::ExtensionPoint is also not covered by the transformation.

C.2.5.12.3 Mapping Specifications

C.2.5.12.3.1 Actor_Mapping

Description

*** not specified yet ***

General Mappings

ElementMain_Mapping
BehavioredClassifier_Mapping

Mapping Source

Actor

Mapping Target

ItemDefinition

Owned Mappings

(none)

C.2.5.12.3.2 UseCaseActor_Mapping

Description

*** not specified yet ***

General Mappings

GenericToPartUsage_Mapping

Mapping Source

Property

Mapping Target

PartUsage

Owned Mappings

- useCaseActorFeatureTyping : UseCaseActorFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- PartUsage::ownedRelationship () : Relationship [0..*]

```
Set{useCaseActorFeatureTyping.to}
```

- PartUsage::name () : String [0..1]

```
from.name
```

C.2.5.12.3.3 UseCaseActorFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *Property* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Property

Mapping Target

FeatureTyping

Owned Mappings

- useCaseActor : UseCaseActor_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

```
useCaseActor.to
```

- FeatureTyping::type () : Type [1]

```
from.type
```

C.2.5.12.3.4 UseCaseActorMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Property* mapping.

General Mappings

GenericToActorMembership_Mapping

Mapping Source

Property

Mapping Target

ActorMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ActorMembership::ownedMemberParameter () : Feature [1]

```
UseCaseActor_Mapping.getMapped(from)
```

C.2.5.12.3.5 Include_Mapping

Description

*** not specified yet ***

General Mappings

GenericToOccurrenceUsage_Mapping

Mapping Source

Include

Mapping Target

IncludeUseCaseUsage

Owned Mappings

- includeFeatureTyping : IncludeFeatureTyping_Mapping

C.2.5.12.3.6 IncludeFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *Include* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Include

Mapping Target

FeatureTyping

Owned Mappings

- includeUsage : Include_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

  ```
  includeUsage.to
  ```

- FeatureTyping::type () : Type [1]

  ```
  from.addition
  ```

C.2.5.12.3.7 IncludeMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Include* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Include

Mapping Target

FeatureMembership

Owned Mappings

- includeUsage : Include_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
includeUsage.to
```

C.2.5.12.3.8 UseCase_Mapping

Description

The expected SysML v2 textual syntax of a mapped UML4SysML::UseCase with a defined subject is as follows.

```
use case def ThisIsAUseCase {
  subject subject_ThisIsABlock : ThisIsABlock;
}
```

Currently, only one use case subject is supported by the mapping class. Since the UML4SysML::Extend relationship is not considered by the SysML v1 to SysML v2 transformation, the extension points of a use case are also not mapped.

General Mappings

BehavioredClassifier_Mapping
NamedElementMain_Mapping

Mapping Source

UseCase

Mapping Target

UseCaseDefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- UseCaseDefinition::ownedRelationship () : Relationship [0..*]

```
let properties : Set(UML::Element) = from.ownedElement->select(e | e.oclIsKindOf(UML::Propert
let actors : Set(UML::Property) = UML::Association.allInstances()->collect(m | m.memberEnd)->
let extensionPoints : Sequence(UML::Element) = from.ownedElement->select(e | e.oclIsKindOf(UM
let extend : Sequence(UML::Element) = from.ownedElement->select(e | e.oclIsKindOf(UML::Extend
let include : Sequence(UML::Element) = from.ownedElement->select(e | e.oclIsKindOf(UML::Inclu
let elements : Set(UML::Element) = ((((from.ownedElement-properties) - extensionPoints) - ext
let relationships : Sequence(KerML::Relationship) =
elements->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(properties->collect(e | PropertyMembership_Mapping.getMapped(e)))
->including(UseCaseSubjectMembership_Mapping.getMapped(from))
```

```
        ->including(UseCaseObjectiveMembership_Mapping.getMapped(from))
        ->including(CommonReturnParameterReferenceUsageMembership_Mapping.getMapped(from))
        ->union(actors->collect(e | UseCaseActorMembership_Mapping.getMapped(e))) in
        if from.classifierBehavior.oclIsUndefined() then relationships else relationships->including(
```

C.2.5.12.3.9 CaseObjectiveMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *Classifier* mapping.

General Mappings

GenericToObjectiveMembership_Mapping

Mapping Source

Classifier

Mapping Target

ObjectiveMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ObjectiveMembership::ownedMemberFeature () : Feature [1]

```
CaseObjectiveRequirementUsage_Mapping.getMapped(from)
```

C.2.5.12.3.10 CaseEmptySubjectReferenceUsage_Mapping

Description

Creates a reference usage for the *Classifier* mapping.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Classifier

Mapping Target

ReferenceUsage

Owned Mappings

(none)

C.2.5.12.3.11 CaseObjectiveRequirementUsage_Mapping

Description

*** not specified yet ***

General Mappings

GenericToRequirementUsage_Mapping

Mapping Source

Classifier

Mapping Target

RequirementUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- RequirementUsage::ownedRelationship () : Relationship [0..*]

```
Set{CaseSubjectMembership_Mapping.getMapped(from), CommonReturnParameterReferenceUsageMember
```

C.2.5.12.3.12 CaseSubjectMembership_Mapping

Description

The current version only supports one specified subject.

General Mappings

GenericToSubjectMembership_Mapping

Mapping Source

Classifier

Mapping Target

SubjectMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- SubjectMembership::ownedMemberParameter () : Feature [0..1]

```
if (from.oclIsTypeOf(UML::UseCase)) and (from.oclAsType(UML::UseCase).subject->size() > 0) t
```

C.2.5.12.3.13 CaseSubjectFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *Classifier* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Classifier

Mapping Target

FeatureTyping

Owned Mappings

- useCaseSubjectReferenceUsage : CaseSubjectReferenceUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
if from->size() > 0 then from->get(0) else OclUndefined endif
```

- FeatureTyping::typedFeature () : Feature [1]

```
useCaseSubjectReferenceUsage.to
```

C.2.5.12.3.14 CaseSubjectReferenceUsage_Mapping

Description

Creates a reference usage for the *Classifier* mapping.

General Mappings

CaseEmptySubjectReferenceUsage_Mapping

Mapping Source

Classifier

Mapping Target

ReferenceUsage

Owned Mappings

- useCaseSubjectFeatureTyping : CaseSubjectFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
Set{useCaseSubjectFeatureTyping.to}
```

- ReferenceUsage::name () : String [0..1]

```
'subject_' + from->get(0).name
```

C.2.5.12.3.15 UseCaseEmptySubjectReferenceUsage_Mapping

Description

Creates a reference usage for the *UseCase* mapping.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

UseCase

Mapping Target

ReferenceUsage

Owned Mappings

(none)

C.2.5.12.3.16 UseCaseObjectiveMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *UseCase* mapping.

General Mappings

GenericToObjectiveMembership_Mapping

Mapping Source

UseCase

Mapping Target

ObjectiveMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ObjectiveMembership::ownedMemberFeature () : Feature [1]

```
UseCaseObjectiveRequirementUsage_Mapping.getMapped(from)
```

C.2.5.12.3.17 UseCaseObjectiveRequirementUsage_Mapping

Description

*** not specified yet ***

General Mappings

GenericToRequirementUsage_Mapping

Mapping Source

UseCase

Mapping Target

RequirementUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- RequirementUsage::ownedRelationship () : Relationship [0..*]

```
Set{UseCaseObjectiveSubjectMembership_Mapping.getMapped(from), CommonReturnParameterReferenc
```

C.2.5.12.3.18 UseCaseObjectiveSubjectMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *UseCase* mapping.

General Mappings

GenericToSubjectMembership_Mapping

Mapping Source

UseCase

Mapping Target

SubjectMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- SubjectMembership::ownedMemberParameter () : Feature [1]

```
UseCaseEmptySubjectReferenceUsage_Mapping.getMapped(from)
```

C.2.5.12.3.19 UseCaseSubjectFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *UseCase* mapping.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

UseCase

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

  ```
  useCaseSubjectReferenceUsage.to
  ```

- FeatureTyping::type () : Type [1]

  ```
  if from.subject->size() > 0 then from.subject->get(0) else OclUndefined endif
  ```

C.2.5.12.3.20 UseCaseSubjectMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *UseCase* mapping.

General Mappings

GenericToSubjectMembership_Mapping

Mapping Source

UseCase

Mapping Target

SubjectMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- SubjectMembership::ownedMemberParameter () : Feature [1]

  ```
  if from.subject->size() > 0 then UseCaseSubjectReferenceUsage_Mapping.getMapped(from) else U
  ```

C.2.5.12.3.21 UseCaseSubjectReferenceUsage_Mapping

Description

Creates a reference usage for the *UseCase* mapping.

General Mappings

UseCaseEmptySubjectReferenceUsage_Mapping

Mapping Source

UseCase

Mapping Target

ReferenceUsage

Owned Mappings

- useCaseSubjectFeatureTyping : UseCaseSubjectFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::name () : String [0..1]

  ```
  'subject_' + from.subject->get(0).name
  ```

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

  ```
  Set{useCaseSubjectFeatureTyping.to}
  ```

C.2.5.13 Values

C.2.5.13.1 Overview

Table 32. List of all Overview Mapping Specfications

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
Duration			
DurationConstraint			

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
DurationInterval			
DurationObservation			
Expression	OperatorExpression OperatorExpression	ExpressionElse_Mapping Expression_Mapping	from.symbol = 'else'
Interval			
IntervalConstraint			
LiteralBoolean			
LiteralInteger			
LiteralNull			
LiteralReal			
LiteralString			
LiteralUnlimitedNatural			
OpaqueExpression	CalculationUsage	OpaqueExpression_Mapping	
StringExpression			
TimeConstraint			
TimeExpression	TriggerInvocationExpression	TimeExpression_Mapping	
TimeInterval			
TimeObservation			

C.2.5.13.2 Mapping Specifications

C.2.5.13.2.1 CommonValueSpecification_Mapping

Description

*** not specified yet ***

General Mappings

GenericToExpression_Mapping

Mapping Source

Element

Mapping Target

Expression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Expression::ownedRelationship () : Relationship [0..*]

  ```
  Set{CommonReturnParameterFeatureMembership_Mapping.getMapped(from)}
  ```

C.2.5.13.2.2 EqualOperatorExpressionFeatureValue_Mapping

Description

Creates a feature value relationship for the mapping class *TypedElement*

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

TypedElement

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

  ```
  CommonFeatureReferenceExpression_Mapping.getMapped(from)
  ```

C.2.5.13.2.3 Expression_Mapping

Description

*** not specified yet ***

General Mappings

GenericToExpression_Mapping
NamedElementMain_Mapping

Mapping Source

Expression

Mapping Target

OperatorExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OperatorExpression::operator () : String [1]

  ```
  from.symbol
  ```

C.2.5.13.2.4 ExpressionElse_Mapping

Description

*** not specified yet ***

General Mappings

Expression_Mapping

Mapping Source

Expression

Mapping Target

OperatorExpression

Owned Mappings

- expressionElseMembership : ExpressionElseMembership_Mapping

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
from.symbol = 'else'
```

Mapping rules

The following lists the mapping rules for the target element properties.

- OperatorExpression::ownedRelationship () : Relationship [0..*]

```
Set{expressionElseMembership.to}
```

C.2.5.13.2.5 ExpressionElseMembership_Mapping

Description

Creates the membership relationship for the textual representation for the else guard condition specification.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Expression

Mapping Target

OwningMembership

Owned Mappings

- expressionElseSpecification : ExpressionElseSpecification_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

```
expressionElseSpecification.to
```

C.2.5.13.2.6 ExpressionElseSpecification_Mapping

Description

Creates the textual representation for the else guard condition specification.

General Mappings

GenericToTextualRepresentation_Mapping

Mapping Source

Expression

Mapping Target

TextualRepresentation

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- TextualRepresentation::language () : String [1]

  ```
  'SysMLv1'
  ```

- TextualRepresentation::body () : String [1]

  ```
  'else'
  ```

C.2.5.13.2.7 LiteralBoolean_Mapping

Description

Maps the UML4SysML::LiteralBoolean to the SysMLv2::LiteralBoolean.

General Mappings

LiteralSpecificationCommon_Mapping

Mapping Source

LiteralBoolean

Mapping Target

LiteralBoolean

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- LiteralBoolean::value () : Boolean [1]

  ```
  from.value
  ```

C.2.5.13.2.8 LiteralBooleanTrue_Mapping

Description

*** not specified yet ***

General Mappings

LiteralSpecificationCommon_Mapping

Mapping Source

Element

Mapping Target

LiteralBoolean

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- LiteralBoolean::value () : Boolean [1]

  ```
  true
  ```

C.2.5.13.2.9 LiteralInteger_Mapping

Description

Maps the UML4SysML::LiteralInteger to the SysMLv2::LiteralInteger.

General Mappings

LiteralSpecificationCommon_Mapping

Mapping Source

LiteralInteger

Mapping Target

LiteralInteger

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- LiteralInteger::value () : Integer [1]

  ```
  from.value
  ```

C.2.5.13.2.10 LiteralNull_Mapping

Description

Maps the UML4SysML::LiteralNull to the SysMLv2::LiteralNull.

General Mappings

LiteralSpecificationCommon_Mapping

Mapping Source

LiteralNull

Mapping Target

NullExpression

Owned Mappings

(none)

C.2.5.13.2.11 LiteralReal_Mapping

Description

Maps the UML4SysML::LiteralReal to the SysMLv2::LiteralReal.

General Mappings

LiteralSpecificationCommon_Mapping

Mapping Source

LiteralReal

Mapping Target

LiteralRational

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- LiteralRational::value () : Real [1]

```
 from.value
```

C.2.5.13.2.12 LiteralSpecificationCommon_Mapping

Description

*** not specified yet ***

General Mappings

GenericToExpression_Mapping

Mapping Source

LiteralSpecification

Mapping Target

LiteralExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- LiteralExpression::ownedRelationship () : Relationship [0..*]

```
 let ownerships: Set(SYSML2::Relationship) = ElementOwnership_Mapping.getMappedColl(from.owne
    ->including(CommonReturnParameterFeatureMembership_Mapping.getMapped(from)) in
if from.type.oclIsUndefined() then
    ownerships
else
    ownerships->including(LiteralSpecificationTyping_Mapping.getMapped(from))
endif
```

C.2.5.13.2.13 LiteralSpecificationTyping_Mapping

Description

*** not specified yet ***

General Mappings

TypedElementToFeatureTyping_Mapping

Mapping Source

LiteralSpecification

Mapping Target

FeatureTyping

Owned Mappings

(none)

C.2.5.13.2.14 LiteralString_Mapping

Description

Maps the UML4SysML::LiteralString to the SysMLv2::LiteralString.

General Mappings

LiteralSpecificationCommon_Mapping

Mapping Source

LiteralString

Mapping Target

LiteralString

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- LiteralString::value () : String [1]

```
if from.value.oclIsUndefined() then '' else from.value endif
```

C.2.5.13.2.15 LiteralUnlimitedToUnbounded_Mapping

Description

Maps the UML4SysML::LiteralUnlimited to the SysMLv2::LiteralInfinity if it is the unlimited value.

General Mappings

LiteralSpecificationCommon_Mapping

Mapping Source

LiteralUnlimitedNatural

Mapping Target

LiteralInfinity

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

C.2.5.13.2.16 LiteralUnlimitedToInteger_Mapping

Description

Maps the UML4SysML::LiteralUnlimited to the SysMLv2::LiteralInteger if it is not the unlimited value.

General Mappings

LiteralSpecificationCommon_Mapping

Mapping Source

LiteralUnlimitedNatural

Mapping Target

LiteralInteger

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- LiteralInteger::value () : Integer [1]

  ```
  from.value
  ```

C.2.5.13.2.17 OpaqueExpressionAsValue_Mapping

Description

*** not specified yet ***

General Mappings

CommonValueSpecification_Mapping

Mapping Source

OpaqueExpression

Mapping Target

FeatureChainExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureChainExpression::ownedRelationship () : Relationship [0..*]

  ```
  Set{OpaqueExpressionParameterMembership_Mapping.getMapped(from), CommonReturnParameterFeatur
  ```

C.2.5.13.2.18 OpaqueExpression_Mapping

Description

*** not specified yet ***

General Mappings

CommonAction_Mapping
ValueSpecification_Mapping

Mapping Source

OpaqueExpression

Mapping Target

CalculationUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- CalculationUsage::ownedRelationship () : Relationship [0..*]

  ```
  Set{OpaqueExpressionMembership_Mapping.getMapped(from), OpaqueExpressionReturnParameterMembe
  ```

C.2.5.13.2.19 OpaqueExpressionFeature_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeature_Mapping

Mapping Source

OpaqueExpression

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::ownedRelationship () : Relationship [0..*]

  ```
  Set{OpaqueExpressionFeatureValue_Mapping.getMapped(from), OpaqueExpressionFeatureFeatureMemb
  ```

C.2.5.13.2.20 OpaqueExpressionFeatureFeature_Mapping

Description

*** not specified yet ***

General Mappings

GenericToFeature_Mapping

Mapping Source

OpaqueExpression

Mapping Target

Feature

Owned Mappings

(none)

C.2.5.13.2.21 OpaqueExpressionFeatureFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*for the *OpaqueExpression* mapping.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

OpaqueExpression

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

  ```
  OpaqueExpressionFeatureFeature_Mapping.getMapped(from)
  ```

C.2.5.13.2.22 OpaqueExpressionFeatureValue_Mapping

Description

Creates a feature value relationship for the mapping class *OpaqueExpression*

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

OpaqueExpression

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

```
OpaqueExpressionFeatureValueExpression_Mapping.getMapped(from)
```

C.2.5.13.2.23 OpaqueExpressionFeatureValueExpression_Mapping

Description

*** not specified yet ***

General Mappings

GenericToExpression_Mapping

Mapping Source

OpaqueExpression

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureReferenceExpression::ownedRelationship () : Relationship [0..*]

```
Set{OpaqueExpressionFeatureValueExpressionMembership_Mapping.getMapped(from), EmptyReturnPar
```

C.2.5.13.2.24 OpaqueExpressionFeatureValueExpressionMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *OpaqueExpression* mapping.

General Mappings

GenericToMembership_Mapping

Mapping Source

OpaqueExpression

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Membership::memberElement () : Element [1]

  ```
  from
  ```

C.2.5.13.2.25 OpaqueExpressionMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *OpaqueExpression* mapping.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

OpaqueExpression

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

```
OpaqueExpressionSpecification_Mapping.getMapped(from)
```

C.2.5.13.2.26 OpaqueExpressionParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()* for the *OpaqueExpression* mapping.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

OpaqueExpression

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ParameterMembership::ownedMemberParameter () : Feature [1]

```
OpaqueExpressionFeature_Mapping.getMapped(from)
```

C.2.5.13.2.27 OpaqueExpressionReturnParameterMembershipReferenceUsage_Mapping

Description

Creates a reference usage for the *OpaqueExpression* mapping.

General Mappings

GenericToReturnParameterMembership_Mapping

Mapping Source

OpaqueExpression

Mapping Target

ReturnParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReturnParameterMembership::ownedMemberParameter () : Feature [1]

```
if from.type.oclIsUndefined() then OpaqueExpressionReturnParameterReferenceUsageUntyped_Mappi
```

C.2.5.13.2.28 OpaqueExpressionReturnParameterReferenceUsage_Mapping

Description

Creates a reference usage for the *OpaqueExpression* mapping.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

OpaqueExpression

Mapping Target

ReferenceUsage

Owned Mappings

- opaqueExpressionReturnParameterReferenceUsageFeatureTyping : OpaqueExpressionReturnParameterReferenceUsageFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::direction () : FeatureDirectionKind [0..1]

```
KerML::FeatureDirectionKind::_'out'
```

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
Set{opaqueExpressionReturnParameterReferenceUsageFeatureTyping.to}
```

C.2.5.13.2.29 OpaqueExpressionReturnParameterReferenceUsageFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*and typed by *type()* for the *OpaqueExpression* mapping.

General Mappings

TypedElementToFeatureTyping_Mapping

Mapping Source

OpaqueExpression

Mapping Target

FeatureTyping

Owned Mappings

- opaqueExpressionReturnParameterReferenceUsage : OpaqueExpressionReturnParameterReferenceUsage_Mapping

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureTyping::typedFeature () : Feature [1]

```
opaqueExpressionReturnParameterReferenceUsage.to
```

C.2.5.13.2.30 OpaqueExpressionReturnParameterReferenceUsageUntyped_Mapping

Description

*** not specified yet ***

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

OpaqueExpression

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::direction () : FeatureDirectionKind [0..1]

```
KerML::FeatureDirectionKind::_'out'
```

C.2.5.13.2.31 OpaqueExpressionSpecification_Mapping

Description

*** not specified yet ***

General Mappings

GenericToTextualRepresentation_Mapping

Mapping Source

OpaqueExpression

Mapping Target

TextualRepresentation

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- TextualRepresentation::body () : String [1]

```
if from.body->size() = 0 then OclUndefined else from.body.get(0) endif
```

- TextualRepresentation::language () : String [1]

```
if from.language->size() = 0 then OclUndefined else from.language.get(0) endif
```

C.2.5.13.2.32 TimeExpression_Mapping

Description

*** not specified yet ***

General Mappings

ValueSpecification_Mapping

Mapping Source

TimeExpression

Mapping Target

TriggerInvocationExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- TriggerInvocationExpression::kind () : TriggerKind [1]

```
SysMLv2::TriggerKind::at
```

C.2.5.13.2.33 ValueSpecification_Mapping

Description

*** not specified yet ***

General Mappings

CommonValueSpecification_Mapping
NamedElementMain_Mapping

Mapping Source

ValueSpecification

Mapping Target

Expression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

The following lists the mapping rules for the target element properties.

- Expression::ownedRelationship () : Relationship [0..*]

```
 if from.type.oclIsUndefined() then
    Set{CommonReturnParameterFeatureMembership_Mapping.getMapped(from)}
else
    Set{LiteralSpecificationTyping_Mapping.getMapped(from), CommonReturnParameterFeatureMembe
endif
```