Date: March 2023

OMG Systems Modeling Language ™ (SysML®)

Version 2.0

SysML v1 to SysML v2 Transformation

Release 2023-02

Fourth Revised Submission **(with errata corrected)**
OMG Document Number: ad/2023-03-06

Machine Readable Files:
SysML v1 to v2 Transformation Model (XMI) ad/2023-02-19

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 LLC

Copyright © 2019-2023, 88solutions Corporation
Copyright © 2019-2023, Airbus
Copyright © 2019-2023, Aras Corporation
Copyright © 2019-2023, Association of Universities for Research in Astronomy (AURA)
Copyright © 2019-2023, BigLever Software
Copyright © 2019-2023, Boeing
Copyright © 2022-2023, Budapest University of Technology and Economics
Copyright © 2021-2023, Commissariat à l'énergie atomique et aux énergies alternatives (CEA)
Copyright © 2019-2023, Contact Software GmbH
Copyright © 2019-2023, Dassault Systèmes (No Magic)
Copyright © 2019-2023, DSC Corporation
Copyright © 2020-2023, DEKonsult
Copyright © 2020-2023, Delligatti Associates LLC
Copyright © 2019-2023, The Charles Stark Draper Laboratory, Inc.
Copyright © 2020-2023, ESTACA
Copyright © 2022-2023, Galois, Inc.
Copyright © 2019-2023, GfSE e.V.
Copyright © 2019-2023, George Mason University
Copyright © 2019-2023, IBM
Copyright © 2019-2023, Idaho National Laboratory
Copyright © 2019-2023, INCOSE
Copyright © 2019-2023, Intercax LLC
Copyright © 2019-2023, Jet Propulsion Laboratory (California Institute of Technology)
Copyright © 2019-2023, Kenntnis LLC
Copyright © 2020-2023, Kungliga Tekniska högskolon (KTH)
Copyright © 2019-2023, LightStreet Consulting LLC
Copyright © 2019-2023, Lockheed Martin Corporation
Copyright © 2019-2023, Maplesoft
Copyright © 2021-2023, MID GmbH
Copyright © 2020-2023, MITRE
Copyright © 2019-2023, Model Alchemy Consulting
Copyright © 2019-2023, Model Driven Solutions, Inc.
Copyright © 2019-2023, Model Foundry Pty. Ltd.
Copyright © 2019-2023, On-Line Application Research Corporation (OAC)
Copyright © 2019-2023, oose Innovative Informatik eG
Copyright © 2019-2023, Østfold University College
Copyright © 2019-2023, PTC
Copyright © 2020-2023, Qualtech Systems, Inc.
Copyright © 2019-2023, SAF Consulting
Copyright © 2019-2023, Simula Research Laboratory AS
Copyright © 2019-2023, System Strategy, Inc.
Copyright © 2019-2023, Thematix Partners, LLC
Copyright © 2019-2023, Tom Sawyer
Copyright © 2022-2023, Tucson Embedded Systems, Inc.
Copyright © 2019-2023, Universidad de Cantabria
Copyright © 2019-2023, University of Alabama in Huntsville
Copyright © 2019-2023, University of Detroit Mercy
Copyright © 2019-2023, University of Kaiserslauten
Copyright © 2020-2023, Willert Software Tools GmbH (SodiusWillert)

Each of the entities listed above: (i) grants to the Object Management Group, Inc. (OMG) a nonexclusive, royalty-free, paid up, worldwide license to copy and distribute this document and to modify this document and distribute copies of the modified version, and (ii) grants to each member of the OMG a nonexclusive, royalty-free, paid up, worldwide license to make up to fifty (50) copies of this document for internal review purposes only and not for distribution, and (iii) has agreed that no person shall be deemed to have infringed the copyright in the included material of any such copyright holder by reason of having used any OMG specification that may be based hereon or having conformed any computer software to such specification.

Table of Contents

List of Tables

0 Submission Introduction

0.1 Submission Overview

The SysML Submission Team (SST) is making two submissions in response to the Systems Modeling Language (SysML®) v2 Request for Proposals (RFP) (ad/2017-11-04).

1. *Kernel Modeling Language (KerML), Version 1.0* (ad/2023-02-02)
2. Part 1: *OMG Systems Modeling Language (SysML), Version 2.0* (ad/2023-02-09)
   Part 2: *SysML v1 to v2 Transformation* (ad/2023-02-10)

This document is the second part of the second submission, specifying an optional but normative SysML v1 to v2 Transformation. This specification formally satisfies RFP Requirement 1.1.3 (see 0.3), but it will also be useful in general for supporting migration from SysML v1 to SysML v2 by projects and organizations.

0.2 Submitters

The following OMG member organizations are jointly submitting this proposed specification:

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

The submitters also thankfully acknowledge the support of many other organizations that participated in the SysML v2 Submission Team (SST).

The contacts for this joint submission (acting for all submitters) are:

- Sandy Friedenthal, SAF Consulting/INCOSE, safriedenthal@gmail.com
- Ed Seidewitz, Model Driven Solutions, ed-s@modeldriven.com

0.3 Issues to be discussed

6.7.1 Proposals shall describe a proof of concept implementation that can successfully execute the test cases that are required in 6.5.4.

The SST generated an implementation of the SysML v1 to v2 Transformation model presented in this specification, with the transformations themselves implemented using the Eclipse ATL transformation language. This prototype implementation was used to validate the transformation model as it was developed,.

6.7.2 Proposals shall provide a requirements traceability matrix that demonstrates how each requirement in the RFP is satisfied. It is recognized that the requirements will be evaluated in more detail as part of the submission process. Rationale should be included in the matrix to support any proposed changes to these requirements.

The SysML v1 to v2 Transformation model proposed in this specification specifically satisfies SysML v2 RFP Requirement LNG 1.1.3:

> Proposals for SysML v2 shall be specified as a SysML v2 profile of UML that includes, as a minimum, the functional capabilities of the SysML v1.x profile, and a mapping to the SysML v2 metamodel.
>
> **Supporting information:** Equivalent functional capability can be demonstrated by mapping the UML metaclasses and SysML stereotypes between SysML v2 and SysML v1.

The SysML v1 to v2 Transformation model proposed in this specification effectively allows the SysML v1.7 profile to also be used as a profile for the subset of SysML v2 functional capabilities that have equivalent capabilities in SysML v1, minimally meeting the RFP requirement (per the "supporting information"). The SST developed an initial UML profile for a portion of SysML v2, and determined that the profile was difficult to use and implement. As a result, the SST decided not to propose a more extensive UML profile for SysML v2.

6.7.3 Proposals shall include a description of how OMG technologies are leveraged and what proposed changes to these technologies are needed to support the specification.

The SysML v1 to v2 Transformation is modeled as a Unified Modeling Language [UML] model mapping between the normative Meta Object Facility [MOF] model of UML and the SysML v1.7 [SysMLv1] profile and the normative MOF models of KerML [KerML] and SysML v2 [SysML v2]. The details of these mappings are specified using the Object Constraint Language [OCL].

1 Scope

This specification describes a transformation for a semantic translation from SysML v1 [SysMLv1] to SysML v2 [SysMLv2] in a precise way. (In this document, "SysML v1" refers to SysML v1.7, the last version of SysML prior to v2.0, and "SysML v2" refers to SysML v2.0, or whatever version corresponds to the current version of 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 provide "SysMLv2-like" read-only access to its SysMLv1 models for external applications.

2 Conformance

A tool shall demonstrate *conformance* with this specification by meeting all of the following requirements.

1. The tool shall implement the UML4SysML abstract syntax and SysML v1 profile conformant with [SysMLv1]. The tool should, but is not required, to provide the ability to import a SysML v1 model using standard XMI Model Interchange format [XMI].
2. The tool shall implement the SysML v2 abstract syntax conformant with [SysML v2]. The tool should, but is not required, to provide the ability to export a SysML v2 model KerML-standard model interchange project (see [KerML], Clause 10; see also [SysML v2], Clause 2).
3. The tool shall implement a transformation from an abstract syntax representation of an input SysML v1 model to the abstract syntax representation of an output SysML v2, as specified in Clause 7 of this specification.

A tool may claim *partial conformance* with this specification by satisfying the first two requirements above, but only implementing an identified subset of the mappings specified in 7.7 and 7.8. (Note that care must also be taken that certain mappings depend on other mappings, and so cannot reasonably be implemented separately.)

Note. A tool that conforms to [SysMLv2] is not required to necessarily implement a transformation conformant with this specification, or it may implement a SysML v1 to v2 transformation that is not claimed to conform with the transformation defined in this specification.

3 Normative References

The following normative documents contain provisions which, through reference in this text, constitute provisions of this specification.

[KerML] *Kernel Modeling Language (KerML),* Version 1.0
https://www.omg.org/spec/KerML/1.0

[MOF] *Meta Object Facility,* Version 2.5.1
https://www.omg.org/spec/MOF/2.5.1

[OCL] *Object Constraint Language,* Version 2.4
https://www.omg.org/spec/OCL/2.4

[SysML v1] *OMG Systems Modeling Language (SysML),* Version 1.7
https://www.omg.org/spec/SysML/1.7

[SysML v2] *OMG Systems Modeling Language (SysML),* Version 2.0
https://www.omg.org/spec/SysML/2.0

[UML] *Unified Modeling Language (UML),* Version 2.5.1
https://www.omg.org/spec/UML/2.5.1

[XMI] *XML Metadata Interchange,* Version 2.5.1
https://www.omg.org/spec/XMI/2.5.1

4 Terms and Definitions

Various terms and definitions are specified throughout the body of this specification.

5 Symbols

No special symbols are defined in this specification.

6 Introduction

6.1 Mapping Approach

The SysML v1 to v2 transformation is specified by directional mappings between UML metaclasses or stereotypes that are part of the SysML v1 specification [SysMLv1] (referenced below as the "SysML v1 scope") on the one hand, and the set of the metaclasses defined in the KerML [KerML] and SysMLv2 [SysMLv2] specifications (referenced below as "SysML v2") in the other hand. Some library classes are also involved.

Each mapping is a directed relationship that reifies a semantic link between a concept belonging to the SysML v1 scope on the source side and one concept belonging to SysML v2 (or one conforming library element) on the target side. As a set, those mappings constitute a declarative specification of a formal transformation that describes how the information encoded by the SysML v1 concepts can be reliably represented using constructs of SysML v2 metaclass 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 values of non-derived properties of the target metaclass instance have to be computed based on property 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 the properties of the target object shall be instances of SysML v2 metaclasses, coming themselves from transformations of SysMLv1 objects to SysMLv2 objects. Since the specification is declarative, the order in which the individual transformations shall happen is not imposed. It is up to a conforming implementation to deal with this. Instead, the `getMapped` static operation is provided for referring to the result of a transformation from within an OCL rule. It returns a (possibly undefined) value, that is typed by the target metaclass of the mapping class from which it is invoked.

Each mapping class 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 for the class `A` (i.e., it 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 of the appropriate type for which this `filter` operation returns `true` shall be translated according to the specifications of that mapping class. The default `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 reflects the specific property 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 actual combination of value of those properties for a given instance of object of the source type, assuming they pass the applicability filter as described above.

6.2 Acknowledgements

The primary authors of this specification document (and also developers of a proof-of-concept implementation of it) are:

- Yves Bernard, Airbus
- Tim Weilkiens, oose

The specification was formally submitted for standardization by the following organizations:

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

However, work on the specification was also supported by over 170 people in over 70 organizations that participated in the SysML v2 Submission Team (SST), by contributing use cases, providing critical review and comment, and validating the language design. The following individuals had leadership roles in the SST:

- Manas Bajaj, Intercax LLC (API and services development lead)
- Yves Bernard, Airbus (v1 to v2 transformation co-lead)
- Bjorn Cole, Lockheed Martin Corporation (metamodel development co-lead)
- Sanford Friedenthal, SAF Consulting (SST co-lead, requirements V&V lead)
- Charles Galey, Lockheed Martin Corporation (metamodel development co-lead)
- Karen Ryan, Siemens (metamodel development co-lead)
- Ed Seidewitz, Model Driven Solutions (SST co-lead, pilot implementation lead)
- Tim Weilkiens, oose (v1 to v2 transformation co-lead)

The specification was prepared using CATIA No Magic modeling tools and the OpenMBEE system for model publication (http://www.openmbee.org), with the invaluable support of the following individuals:

- Tyler Anderson, No Magic/Dassault Systèmes
- Christopher Delp, Jet Propulsion Laboratory
- Ivan Gomes, Twingineer
- Doris Lam, Jet Propulsion Laboratory
- Robert Karban, Jet Propulsion Laboratory
- Christopher Klotz, No Magic/Dassault Systèmes
- John Watson, Lightstreet Consulting

7 Mappings

7.1 Overview

This Clause is organized in order to match the packages that subdivide the model of the transformation. The `Foundations` package gathers the abstract classes that represent the concepts on top of which the mapping approach is built. The next subclause presents a utility class named `Helper` that provides reusable operations that simplify the OCL statements defining the computation rules of target properties and make them more readable. Libraries play an important role in SysML v2, and a specific one has been created in order to represent semantics equivalent to those of UML/SysML concepts, where needed. It is presented in this subclause as well.

The three next subclauses are dedicated to initializers, factories and generic mappings, respectively. They do not specify mappings, strictly speaking. Instead, they factorize more or less advanced OCL code that will be reused by the actual mapping specifications that are contained in the two last subclauses. The first of them is dedicated to UML metaclass from the UML4SYSML scope, while the second deals with SysML stereotypes more specifically.

7.2 Foundations

7.2.1 Overview

The concepts defined by KerML/SysML v2 are relatively similar to those of UML/SysML v1, but the ways they are built are different. This makes the specification of the global transformation quite complex. In order to keep it manageable, specific kinds of foundational classes are provided. They represent concepts on which classical "model to model" transformation technologies rely:

- The mappings built on top of the abstract class `Mapping` shall be executed only when they are explicitly called. Each call shall produce a new target element, whatever the source element. It specifies a `from` property typed by the `UML::CommonStructure::Element` metaclass that shall be redefined by any of its subclass for specifying the convenient type of source element. Also it specifies a default (neutral) filter and a set of `getMapped` operations for various purposes: regular mapping result, qualified mapping result and mapping result for a collection of elements.
- The mappings built on top of the abstract class `UniqueMapping`, specified as a specialization of the `Mapping` class, shall produce only one target element for a given source element, whatever the number of time they are called.
- The mappings built on top of the abstract class `MainMapping`, specified as a specialization of the `UniqueMapping` class, shall be systematically executed (i.e. implicitly called) for all the elements that match both theirs source type and filter. There can be at most one main mapping for a given source type and only one target element shall be produced for a given source element.

The corresponding classes are located the the `Foundations` package.

Sometimes, it is necessary to be able to generate elements in the target model without having to provide an explicit link with a source element. In such a case, a mapping class is not appropriate. Instead the mapping framework provides the concept of a `Factory`.

Last, the concept of an `Initializer` allows the factorization of the specification of properties' default values that can be inherited by mappings and factories, as convenient.

In the model of the transformation that is specified here, all of the abstract classes of this `Foundations` package are subject to direct or indirect subclassing. In other words, this specification is built as a set of interrelated initializers, factories, regular, unique and main mappings, where the initializers' operation factorizes the specification of default values for their target element, wherever possible. Those "default operations" are either used as-is or redefined by mappings or factories that can inherit for a specific initializer, as appropriate.

7.2.2 Foundational class specifications

7.2.2.1 Factory

Description

Similarly to the well-known to the homonyms software design pattern, a Factory can be used for specifying the production of a target element without any link with a source element. Factories have in common with mapping classes the operations that specify how the properties of the target element shall be computed and the "to" property that specifies the type of the target element. However factories do not define source element. Instead, they can have parameters. Those parameters, if any, shall be specified by properties with appropriate types and multiplicities. Factories are expected to provide a "create" operation with parameters matching in type and multiplicity the properties that are intended to specify them.

Generalizations

- Initializer (from Foundations)

7.2.2.2 Initializer

Description

The abstract class Initializer is the common ancestor of Mapping and Factory. It specifies a "to" property typed by the KerML::Root::Element metaclass that shall be redefined by any of its subclass for specifying the convenient type of target element. Initializers are intended to specify reusable properties' computation rules, mainly for initializing them with default values. Those rules will be inherited or redefined by the sub-classes, as appropriate.

Attributes

- /inputs [0..*]

Association Ends

- to : Element [1]

7.2.2.3 MainMapping

Description

The mappings built on top of the abstract class MainMapping are a specific kind of UniqueMappings class that are always implicitly called for any element in the source model that match both their source type (as specified by their "from" property) and their filter condition. If more than one main mapping is specified for a given source type, they shall have filters that specify mutually exclusive conditions. Also, as with any unique mapping, only one target element shall be produced for a given source element.

Generalizations

- UniqueMapping (from Foundations)

7.2.2.4 Mapping

Description

This is the generic abstract class that provides the basic features of any mapping class mapping. The mappings built on top of the abstract class Mapping are intended to be executed only when explicitly called (e.g. by the rule of another mapping class). It specifies a "from" property typed by the UML::CommonStructure::Element metaclass that shall be redefined by any of its subclass for specifying the convenient type of source element. Also it specifies a default (neutral) filter and a set of getMapped operations for various purposes: regular mapping result, qualified mapping result and mapping result for a collection of elements. Each call to the getMapped operation shall produce a new target element, whatever the source element provided. Instances of Mapping class are represent a link between one source element and the target element produced by the transformation specified by that mapping class.

Generalizations

- Initializer (from Foundations)

Association Ends

- from : Element [1]

Operations

- filter (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)

  ```
   true
  ```

- getMapped (in fromVar : Element) : Element [1]

  postConditions:

  ```
  self.filter(fromVar) and
  self.to.allFeatures()->selectByKind(UML::Property)->reject(isDerived)
  ->forAll(p | let ops: Operation = self.allFeatures()
      ->selectByKind(UML::Operation)->any(o | o.name = p.name) in
      p = ops()) and
  result = self.to
  ```

- getMapped (in fromVar : Element, in qual : Element) : Element [1]

  postConditions:

  ```
  self.filter(fromVar) and
  self.to.allFeatures()->selectByKind(UML::Property)->reject(isDerived)
  ->forAll(p | let ops: Operation = self.allFeatures()
      ->selectByKind(UML::Operation)->any(o | o.name = p.name) in
      if ops.ownedParameter
          ->select(p | p.direction = UML::ParameterDirectionKind::_'in')
          ->size()=1 then
          p = ops(qual)
      else if ops.ownedParameter
          ->select(p | p.direction = UML::ParameterDirectionKind::_'in')
          ->size()=0 then
          p = ops()
      else
          invalid
  ```

```
    endif endif) and
result = self.to
```

- getMappedColl (in fromColl : Element) : Element [0..*]

  postConditions:

  ```
  result = fromColl->collect(e | self.getMapped(e))
  ```

7.2.2.5 UniqueMapping

Description

The mappings built on top of the abstract class UniqueMapping are a specific kind of Mappings that are intended to produce only one target element for a given source element, whatever the number of time they are called.If a getMapped is called several time with the same source element, the target element returned shall always be the same.

Generalizations

- Mapping (from Foundations)

7.3 Mapping Helper and Library

7.3.1 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 UML4SysML::Action mapping.

  ```
  let actionInputPin: Set(UML::Element) =
      src.ownedElement->select(e | e.oclIsTypeOf(UML::ActionInputPin)) in
  let triggers: Set(UML::Element) =
      src.ownedElement->select(e | e.oclIsKindOf(UML::Trigger)) in
  let toElementFMS: Set(UML::Element) =
      src.ownedElement->select(e | e.oclIsKindOf(UML::Pin)) in
  let toElementOMS: Set(UML::Element) =
      (((src.ownedElement - toElementFMS) - actionInputPin) - triggers) in
  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 UML4SysML::Activity mapping.

```
let initialNodes : Set(UML::Element) =
    src.ownedElement->select(e | e.oclIsKindOf(UML::InitialNode)) in
let finalNodes : Set(UML::Element) =
    src.ownedElement->select(e | e.oclIsKindOf(UML::FinalNode)) in
let elementsFMS : Set(UML::Element) =
    ((src.ownedElement->select(e | e.oclIsKindOf(UML::ControlNode) or
    e.oclIsKindOf(UML::Action) or e.oclIsKindOf(UML::ControlFlow) or
    e.oclIsKindOf(UML::ObjectFlow) or e.oclIsKindOf(UML::Property))
    - initialNodes) - finalNodes) in
let parameters: Set(UML::Parameter) =
    src.ownedElement->select(e | e.oclIsKindOf(UML::Parameter)) in
let ignoreParameterNodes: Set(UML::ActivityParameterNode) =
    src.ownedElement->select(e | e.oclIsKindOf(UML::ActivityParameterNode)) in
let ignoreActivityPartition: Set(UML::ActivityPartition) =
    src.ownedElement->select(e | e.oclIsKindOf(UML::ActivityPartition)) in
let ignoreInterruptibleActivityRegion: Set(UML::InterruptibleActivityRegion) =
    src.ownedElement
    ->select(e | e.oclIsKindOf(UML::InterruptibleActivityRegion)) in
let ownedClassifier: Sequence(UML::Classifier) =
    src.ownedElement->select(e | e.oclIsKindOf(UML::Classifier)) in
let variables: Sequence(UML::Variable) =
    src.ownedElement->select(e | e.oclIsKindOf(UML::Variable)) in
let parameterSets: Set(UML::ParameterSet) =
    src.ownedElement->select(e | e.oclIsKindOf(UML::ParameterSet)) in
let elementsOMS: Set(UML::Element) =
    (((((((((src.ownedElement-initialNodes)-finalNodes)-elementsFMS)-parameters)-
    ignoreParameterNodes)-ignoreActivityPartition)-
    ignoreInterruptibleActivityRegion)-ownedClassifier)-variables)-
    parameterSets)-Set{from.classifierBehavior}) in
let memberships : Sequence(UML::Element) =
elementsOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(initialNodes->collect(e | InitialNodeMembership_Mapping.getMapped(e)))
->union(finalNodes->collect(e | FlowFinalNodeMembership_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(ClassifierBehaviorFeatureMembership_Mapping.getMapped(src))
endif
```

- 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 UML4SysM::Enumeration to the appropriate SysML v2 EnumerationDefinition.

```
 let enum: SYSML2::EnumerationDefinition =
    Enumeration_Mapping.getMapped(t) in
```

```
if enum.oclIsKindOf(SYSML2::EnumerationDefinition) then
    enum
else if t.name = 'VerdictKind' then
        SYSML2::EnumerationDefinition.allInstances()
        ->any(e | e.qualifiedName = 'VerificationCases::VerdictKind')

     else if t = UML::ParameterDirectionKind then
        KerML::FeatureDirectionKind

        else if t.qualifiedName =
            'SysML::Libraries::ControlValues::ControlValueKind' then
            SYSML2::EnumerationDefinition.allInstances()
            ->any(e | e.qualifiedName =
                'SysMLv1Library::Enumerations::ControlValueKind')

            else
                SYSML2::EnumerationDefinition.allInstances()
                ->any(e | e.qualifiedName =
                    'SysMLv1Library::Enumerations::' + t.name)
            endif
        endif
    endif
endif
```

- getFlowDirectionKind (in v : EnumerationLiteral) : FeatureDirectionKind [1]
  Maps a given SysMLv1 feature direction enumeration literal to a SysML v2 FeatureDirectionKind enumeration literal.

```
 if v.enumeration.qualifiedName =
    'SysML::Ports&Flows::FlowDirectionKind' then
    if v = SysML::FlowDirectionKind::_'out' then
        KerML::FeatureDirectionKind::_'out'
    else if (v = SysML::FlowDirectionKind::_'in') then
        KerML::FeatureDirectionKind::_'in'
    else if (v = SysML::FlowDirectionKind::inout) then
        KerML::FeatureDirectionKind::inout
    else
        invalid
    endif endif endif
else
    invalid
endif
```

- getID (in src : Element) : String [1]
  Returns the identifier of a UML4SysML::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 SysML v2 FeatureDirectionKind enumeration literal.

```
 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 SysML v2 FeatureDirectionKind enumeration literal.

```
 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 UML4SysML::VisibilityKind enumeration literal to a SysML v2 VisibilityKind enumeration literal.

```
 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.

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

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

```
 let stereotypes: Set(UML::Stereotype) =
    Helper.getAppliedStereotypes(element) in
```

```
stereotypes->any(s | s.general->collect(g | g.qualifiedName)
->includes('SysML::Requirements::AbstractRequirement'))
```

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

```
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.

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

- 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.

- 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.

- globalNamespace () : Namespace [1]

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

- hasMainMapping (in element : Element) : Boolean [1]
- 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 UML4SysML::Association is mapped to a SysML v2 ConnectionDefinition.

```
-- 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
```

- isInScope (in element : Element) : Boolean [1]
  The isInScope operation is intended to define the scope on which the transformation will apply. If the isInScope operation return "true" for a given model element, this element shall be consider by the transformation. Especially, main mappings - if any - will apply to it. It shall be ignored otherwise.
- isRequirement (in element : Element) : Boolean [1]
  Checks whether the stereotype AbstractRequirement is applied to the given element.

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

- packageOwnedRelationship (in src : Element) : Relationship [0..*]
  Reusable mapping rule for owned relationships of a UML4SysML::Package mapping.

```
let useCaseAssociations : Set(UML::Association) =
    src.ownedType->select(e | e.oclIsKindOf(UML::Association))
    ->select(a | a.memberEnd->exists(e | e.type.oclIsKindOf(UML::UseCase))) in
let unmappedAssociations : Set(UML::Association) =
    src.ownedType->select(e | e.oclIsKindOf(UML::Association))
    ->reject(a | Helper.isConnectionDef(a)) in
let imports: Set(UML::PackageImport) =
    src.packageImport->select(pi | Helper.isInScope(pi.importedPackage)) in
let relationships: Set(SysMLv2::Relationship) =
    src.ownedComment->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(((src.ownedType-useCaseAssociations)-unmappedAssociations)
->collect(e | ElementOwningMembership_Mapping.getMapped(e)))
->union(imports->collect(i | PackageImport_Mapping.getMapped(i)))
->union(src.ownedElement->select(e | e.oclIsKindOf(UML::Dependency) or
e.oclIsKindOf(UML::Package)
or (e.oclIsKindOf(UML::InstanceSpecification) and
e.oclAsType(UML::InstanceSpecification).classifier->notEmpty()))
->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 UML4SysML::State mapping.

```
let initialState : Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Pseudostate) and
    e.oclAsType(UML::Pseudostate).kind = UML::PseudostateKind::initial) in
let toElementOMS : Set(UML::Element) = from.ownedElement - initialState in
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(initialState->collect(e | InitialStateMembership_Mapping.getMapped(e)))
```

7.3.2 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 library elements and metadata for
         * SysML elements which cannot mapped to a SysML v2 element.
         */

        // Library elements

        action def AddValueAction {
                in insertAt : ScalarValues::Natural [0..1];
                in value : ScalarValues::Integer;
                in isReplaceAll : ScalarValues::Boolean = false;
                in target;

                if isReplaceAll {
                        if insertAt == * {
                                assign target := SequenceFunctions::including(target, value);
                        }
                        else {
                                assign target :=
                                        SequenceFunctions::includingAt(target, value, insertAt);
                        }
                } else {
                        target := value;
                }
        }

        action def AddStructuralFeatureValueAction :> AddValueAction {
                in object;
        }
        action def RemoveVariableValueAction :> Actions::AssignmentAction {
                in removeAt: ScalarValues::Natural [0..1];
                in value : ScalarValues::Integer;
                in isRemoveDuplicates : ScalarValues::Boolean = false;
                in variable;

                // isRemoveDuplicates not covered yet

                if removeAt {
                        assign variable :=
                                SequenceFunctions::excludingAt(variable, value, removeAt);
                } else {
                        assign variable := SequenceFunctions::excluding(variable, value);
                }
        }


        // Metadata

        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 property mapping
                 */
        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 */
            :> annotatedElement : SysML::Package;
            attribute 'viewpoint' : ScalarValues::String;
    }

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

        metadata def ParameterSetData {
                doc /* Metadata definition for tagging parameters
                 * mapped from a UML::ParameterSet
                 */
                attribute isParameterSet : ScalarValues::Boolean;
        }

    metadata def PortData {
            doc /* Metadata definition for tagging SysML v2 ports
             * mapped from a SysML::Ports&Flows::FullPort element
             */
            :> annotatedElement : SysML::PartUsage;
            attribute isFullPort : ScalarValues::Boolean;
    }

    metadata def ProbabilityData {
            doc /* Metadata definition for SysML::Activities::Probability stereotype */
            attribute probability : ScalarValues::Real;
    }

    metadata def RateData {
            doc /* Metadata definition for SysML::Activities::Rate and
             * specialized Discrete and Continuous stereotypes
             */
            :> annotatedElement : SysML::PartUsage;
            part rate;
            attribute isDiscrete : ScalarValues::Boolean;
            attribute isConcrete : ScalarValues::Boolean;
    }

    metadata def RefineData {
            doc /* Metadata definition for tagging SysML v2 dependencies
             * mapped from a SysML::Requirements::Refine relationship
             */
            :> annotatedElement : SysML::Dependency;
            attribute isRefine : ScalarValues::Boolean;
    }

    metadata def StakeholderData {
            doc /* Metadata definition for tagging SysML v2 item definitions
             * mapped from a SysML::ModelElements::Stakeholder element
             */
            :> annotatedElement : SysML::ItemDefinition;
            attribute isStakeholder : ScalarValues::Boolean;
    }

    metadata def traceData {
            doc /* Metadata definition for tagging SysML v2 dependencies
             * mapped from a SysML::Requirements::Trace relationship
             */
            :> annotatedElement : SysML::Dependency;
            attribute isTrace : ScalarValues::Boolean;
    }

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

    package Enumerations {
            enum def ControlValueKind {
                    doc /* The ControlValueKind enumeration is a type for
```

```
                     * treating control values as data and for UML control pins.
                     */
                    enum disable;
                    enum enable;
            }
    }
}
```

7.4 Initializers

7.4.1 Overview

The classes presented in this subclause provide set of rules that provide default values for all non-derived features of their target metaclasses. Intentionally, initializers do not specify any "source" element. This makes them easier to specialize but prevents them from being able to provide a computation algorithm for some target features. In such a case, the operation matching the feature will be specified as abstract.

7.4.2 Mapping Specifications

7.4.2.1 KerML Initializers

7.4.2.1.1 AnnotatingElement_Init

Description

Initializes the properties of the SysML v2 element AnnotatingElement.

Generalizations

- Element_Init (from KerMLInitializers)

Association Ends

- to : AnnotatingElement [1]
  (redefines: Element_Init::to)

Operations

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

```
Set{}
```

7.4.2.1.2 Annotation_Init

Description

Initializes the properties of the SysML v2 element Annotation.

Generalizations

- Relationship_Init (from KerMLInitializers)

Attributes

- to : Annotation [1]

Operations

- annotatedElement () : Element [1] {redefines target, abstract}
- annotatingElement () : AnnotatingElement [1] {redefines source, abstract}
- owningAnnotatedElement () : Element [0..1]

```
null
```

7.4.2.1.3 Association_Init

Description

Initializes the properties of the SysML v2 element Association.

Generalizations

- Classifier_Init (from KerMLInitializers)
- Relationship_Init (from KerMLInitializers)

Attributes

- to : Association [1]

7.4.2.1.4 Behavior_Init

Description

Initializes the properties of the SysML v2 element Behavior.

Generalizations

- Classifier_Init (from KerMLInitializers)

Attributes

- to : Behavior [1]

7.4.2.1.5 Classifier_Init

Description

Initializes the properties of the SysML v2 element Classifier.

Generalizations

- Type_Init (from KerMLInitializers)

Attributes

- to : Classifier [1]

7.4.2.1.6 Comment_Init

Description

Initializes the properties of the SysML v2 element Comment.

Generalizations

- AnnotatingElement_Init (from KerMLInitializers)

Association Ends

- to : Comment [1]
  (redefines: AnnotatingElement_Init::to)

Operations

- body () : String [1]{abstract}
- locale () : String [1]

```
null
```

7.4.2.1.7 Conjugation_Init

Description

Initializes the properties of the SysML v2 element Conjugation.

Generalizations

- Relationship_Init (from KerMLInitializers)

Attributes

- to : Conjugation [1]

Operations

- conjugatedType () : Type [1] {redefines source, abstract}
- originalType () : Type [1] {redefines target, abstract}

7.4.2.1.8 Connector_Init

Description

Initializes the properties of the SysML v2 element Connector.

Generalizations

- Feature_Init (from KerMLInitializers)
- Relationship_Init (from KerMLInitializers)

Attributes

- to : Connector [1]

Operations

- isDirected () : Boolean [1]

```
false
```

7.4.2.1.9 Documentation_Init

Description

Initializes the properties of the SysML v2 element Documentation.

Generalizations

- Comment_Init (from KerMLInitializers)

Attributes

- to : Documentation [1]

7.4.2.1.10 Element_Init

Description

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

Generalizations

- Initializer (from Foundations)

Association Ends

- to : Element [1]
  (redefines: Initializer::to)

Operations

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

```
Set{}
```

- declaredName () : String [0..1]

```
null
```

- elementId () : String [1]

```
Helper.createUUID()
```

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

```
Set{}
```

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

```
null
```

7.4.2.1.11 EndFeatureMembership_Init

Description

Initializes the properties of the SysML v2 element EndFeatureMembership.

Generalizations

- FeatureMembership_Init (from KerMLInitializers)

Attributes

- to : EndFeatureMembership [1]

7.4.2.1.12 Expression_Init

Description

Initializes the properties of the SysML v2 element Expression.

Generalizations

- Step_Init (from KerMLInitializers)

Attributes

- to : Expression [1]

7.4.2.1.13 Feature_Init

Description

Initializes the properties of the SysML v2 element Feature.

Generalizations

- Type_Init (from KerMLInitializers)

Attributes

- to : Feature [1]

Operations

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

```
null
```

- isComposite () : Boolean [1]

```
false
```

- isDerived () : Boolean [1]

```
false
```

- isEnd () : Boolean [1]

```
false
```

- isOrdered () : Boolean [1]

```
false
```

- isPortion () : Boolean [1]

```
false
```

- isReadOnly () : Boolean [1]

```
false
```

- isUnique () : Boolean [1]

```
true
```

7.4.2.1.14 FeatureChainExpression_Init

Description

Initializes the properties of the SysML v2 element FeatureChainExpression.

Generalizations

- OperatorExpression_Init (from KerMLInitializers)

Attributes

- to : FeatureChainExpression [1]

7.4.2.1.15 FeatureChaining_Init

Description

Initializes the properties of the SysML v2 element FeatureChaining.

Generalizations

- Relationship_Init (from KerMLInitializers)

Attributes

- to : FeatureChaining [1]

Operations

- chainingFeature () : Feature [1] {redefines target, abstract}

7.4.2.1.16 FeatureMembership_Init

Description

Initializes the properties of the SysML v2 element FeatureMembership.

Generalizations

- OwningMembership_Init (from KerMLInitializers)
- TypeFeaturing_Init (from KerMLInitializers)

Attributes

- to : FeatureMembership [1]

Operations

- ownedMemberFeature () : Feature [1] {redefines ownedMemberElement, abstract}
- ownedRelatedElement () : Element [0..*] {redefines ownedRelatedElement}

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

7.4.2.1.17 FeatureReferenceExpression_Init

Description

Initializes the properties of the SysML v2 element FeatureReferenceExpression.

Generalizations

- Expression_Init (from KerMLInitializers)

Attributes

- to : FeatureReferenceExpression [1]

7.4.2.1.18 FeatureTyping_Init

Description

Initializes the properties of the SysML v2 element FeatureTyping.

Generalizations

- Specialization_Init (from KerMLInitializers)

Attributes

- to : FeatureTyping [1]

Operations

- type () : Type [1] {redefines general, abstract}
- typedFeature () : Feature [1] {redefines specific, abstract}

7.4.2.1.19 FeatureValue_Init

Description

Initializes the properties of the SysML v2 element FeatureValue.

Generalizations

- OwningMembership_Init (from KerMLInitializers)

Attributes

- to : FeatureValue [1]

Operations

- featureWithValue () : Feature [1] {redefines ownedMemberElement, abstract}
- isDefault () : Boolean [1]

```
false
```

- isInitial () : Boolean [1]

```
false
```

- ownedRelatedElement () : Element [0..*] {redefines ownedRelatedElement}

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

- value () : Expression [1] {redefines ownedMemberElement, abstract}

7.4.2.1.20 Function_Init

Description

Initializes the properties of the SysML v2 element Function.

Generalizations

- Behavior_Init (from KerMLInitializers)

Attributes

- to : Function [1]

7.4.2.1.21 Import_Init

Description

Initializes the properties of the SysML v2 element Import.

Generalizations

- Relationship_Init (from KerMLInitializers)

Attributes

- to : Import [1]

Operations

- importedMemberName () : String [0..1]

```
null
```

- isImportAll () : Boolean [1]

```
false
```

- isRecursive () : Boolean [1]

```
false
```

- source () : Element [1] {redefines source, abstract}
- target () : Element [1] {redefines target, abstract}
- visibility () : VisibilityKind [1]

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

7.4.2.1.22 Interaction_Init

Description

Initializes the properties of the SysML v2 element Interaction.

Generalizations

- Association_Init (from KerMLInitializers)
- Behavior_Init (from KerMLInitializers)

Attributes

- to : Interaction [1]

7.4.2.1.23 InvocationExpression_Init

Description

Initializes the properties of the SysML v2 element InvocationExpression.

Generalizations

- Expression_Init (from KerMLInitializers)

Attributes

- to : InvocationExpression [1]

7.4.2.1.24 ItemFlow_Init

Description

Initializes the properties of the SysML v2 element ItemFlow.

Generalizations

- Connector_Init (from KerMLInitializers)

Attributes

- to : ItemFlow [1]

7.4.2.1.25 Membership_Init

Description

Initializes the properties of the SysML v2 element Membership.

Generalizations

- Relationship_Init (from KerMLInitializers)

Attributes

- to : Membership [1]

Operations

- memberElement () : Element [1] {redefines target, abstract}
- memberName () : String [0..1]

```
null
```

- memberShortName () : String [0..1]

```
null
```

- membershipOwningNamespace () : Element [0..*] {redefines source, abstract}

- visibility () : VisibilityKind [1]

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

7.4.2.1.26 MembershipImport_Init

Description

Initializes the properties of the SysML v2 element MembershipImport.

Generalizations

- Import_Init (from KerMLInitializers)

Attributes

- to : MembershipImport [1]

Operations

- importedMembership () : Namespace [1] {redefines target, abstract}

7.4.2.1.27 Namespace_Init

Description

Initializes the properties of the SysML v2 element Namespace.

Generalizations

- Element_Init (from KerMLInitializers)

Association Ends

- to : Namespace [1]
  (redefines: Element_Init::to)

7.4.2.1.28 NamespaceImport_Init

Description

Initializes the properties of the SysML v2 element NamespaceImport.

Generalizations

- Import_Init (from KerMLInitializers)

Attributes

- to : NamespaceImport [1]

Operations

- importedNamespace () : Namespace [1] {redefines target, abstract}

7.4.2.1.29 OperatorExpression_Init

Description

Initializes the properties of the SysML v2 element OperatorExpression.

Generalizations

- Expression_Init (from KerMLInitializers)

Attributes

- to : OperatorExpression [1]

Operations

- operator () : String [1]{abstract}

7.4.2.1.30 OwningMembership_Init

Description

Initializes the properties of the SysML v2 element OwningMembership.

Generalizations

- Membership_Init (from KerMLInitializers)

Attributes

- to : OwningMembership [1]

Operations

- ownedMemberElement () : Element [1] {redefines memberElement, abstract}
- ownedRelatedElement () : Element [0..*] {redefines ownedRelatedElement}

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

7.4.2.1.31 Package_Init

Description

Initializes the properties of the SysML v2 element Package.

Generalizations

- Namespace_Init (from KerMLInitializers)

Attributes

- to : Package [1]

7.4.2.1.32 ParameterMembership_Init

Description

Initializes the properties of the SysML v2 element ParameterMembership.

Generalizations

- FeatureMembership_Init (from KerMLInitializers)

Attributes

- to : ParameterMembership [1]

Operations

- ownedMemberParameter () : Feature [1] {redefines ownedMemberFeature, abstract}
- ownedRelatedElement () : Element [0..*] {redefines ownedRelatedElement}

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

7.4.2.1.33 Predicate_Init

Description

Initializes the properties of the SysML v2 element Predicate.

Generalizations

- Function_Init (from KerMLInitializers)

Attributes

- to : Predicate [1]

7.4.2.1.34 Redefinition_Init

Description

Initializes the properties of the SysML v2 element Redefinition.

Generalizations

- Subsetting_Init (from KerMLInitializers)

Attributes

- to : Redefinition [1]

Operations

- redefinedFeature () : Feature [1] {redefines subsettedFeature, abstract}
- redefiningFeature () : Feature [1] {redefines subsettingFeature, abstract}

7.4.2.1.35 ReferenceSubsetting_Init

Description

Initializes the properties of the SysML v2 element ReferenceSubsetting.

Generalizations

- Subsetting_Init (from KerMLInitializers)

Attributes

- to : ReferenceSubsetting [1]

Operations

- referencedFeature () : Feature [1] {redefines subsettedFeature, abstract}

7.4.2.1.36 Relationship_Init

Description

Initializes the properties of the SysML v2 element Relationship.

Generalizations

- Element_Init (from KerMLInitializers)

Association Ends

- to : Relationship [1]
  (redefines: Element_Init::to)

Operations

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

  ```
  Set{}
  ```

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

  ```
  Set{}
  ```

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

  ```
  Set{}
  ```

7.4.2.1.37 ReturnParameterMembership_Init

Description

Initializes the properties of the SysML v2 element ReturnParameterMembership.

Generalizations

- ParameterMembership_Init (from KerMLInitializers)

Attributes

- to : ReturnParameterMembership [1]

Operations

- 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
```

7.4.2.1.38 Specialization_Init

Description

Initializes the properties of the SysML v2 element Specialization.

Generalizations

- Relationship_Init (from KerMLInitializers)

Attributes

- to : Specialization [1]

Operations

- general () : Type [1] {redefines target, abstract}
- specific () : Type [1] {redefines source, abstract}

7.4.2.1.39 Step_Init

Description

Initializes the properties of the SysML v2 element Step.

Generalizations

- Feature_Init (from KerMLInitializers)

Attributes

- to : Step [1]

7.4.2.1.40 Subclassification_Init

Description

Initializes the properties of the SysML v2 element Subclassification.

Generalizations

- Specialization_Init (from KerMLInitializers)

Attributes

- to : Subclassification [1]

Operations

- subclassifier () : Classifier [1]{abstract}
- superclassifier () : Classifier [1]{abstract}

7.4.2.1.41 Subsetting_Init

Description

Initializes the properties of the SysML v2 element Subsetting.

Generalizations

- Specialization_Init (from KerMLInitializers)

Attributes

- to : Subsetting [1]

Operations

- subsettedFeature () : Feature [1] {redefines general, abstract}
- subsettingFeature () : Feature [1] {redefines specific, abstract}

7.4.2.1.42 Succession_Init

Description

Initializes the properties of the SysML v2 element Succession.

Generalizations

- Connector_Init (from KerMLInitializers)

Attributes

- to : Succession [1]

7.4.2.1.43 SuccessionItemFlow_Init

Description

Initializes the properties of the SysML v2 element SuccessionItemFlow.

Generalizations

- ItemFlow_Init (from KerMLInitializers)

- Succession_Init (from KerMLInitializers)

Attributes

- to : SuccessionItemFlow [1]

7.4.2.1.44 TextualRepresentation_Init

Description

Initializes the properties of the SysML v2 element TextualRepresentation.

Generalizations

- AnnotatingElement_Init (from KerMLInitializers)

Attributes

- to : TextualRepresentation [1]

Operations

- body () : String [1]{abstract}
- language () : String [1]{abstract}

7.4.2.1.45 Type_Init

Description

Initializes the properties of the SysML v2 element Type.

Generalizations

- Namespace_Init (from KerMLInitializers)

Attributes

- to : Type [1]

Operations

- isAbstract () : Boolean [1]

```
false
```

- isSufficient () : Boolean [1]

```
false
```

7.4.2.1.46 TypeFeaturing_Init

Description

Initializes the properties of the SysML v2 element TypeFeaturing.

Generalizations

- Relationship_Init (from KerMLInitializers)

Attributes

- to : TypeFeaturing [1]

Operations

- featureOfType () : Feature [1] {redefines source, abstract}
- featuringType () : Type [1] {redefines target, abstract}

7.4.2.2 System Initializers

7.4.2.2.1 ActionUsage_Init

Description

Initializes the properties of the SysML v2 element ActionUsage.

Generalizations

- Step_Init (from KerMLInitializers)
- Usage_Init (from SystemInitializers)

Attributes

- to : ActionUsage [1]

Operations

- isComposite () : Boolean [1] {redefines isComposite}

```
true
```

7.4.2.2.2 ActorMembership_Init

Description

Initializes the properties of the SysML v2 element ActorMembership.

Generalizations

- ParameterMembership_Init (from KerMLInitializers)

Attributes

- to : ActorMembership [1]

7.4.2.2.3 AssignmentActionUsage_Init

Description

Initializes the properties of the SysML v2 element AssignmentActionUsage.

Generalizations

- ActionUsage_Init (from SystemInitializers)

Attributes

- to : AssignmentActionUsage [1]

7.4.2.2.4 ConjugatedPortDefinition_Init

Description

Initializes the properties of the SysML v2 element ConjugatedPortDefinition.

Generalizations

- PortDefinition_Init (from SystemInitializers)

Attributes

- to : ConjugatedPortDefinition [1]

7.4.2.2.5 ConjugatedPortTyping_Init

Description

Initializes the properties of the SysML v2 element ConjugatedPortTyping.

Generalizations

- FeatureTyping_Init (from KerMLInitializers)

Attributes

- to : ConjugatedPortTyping [1]

Operations

- conjugatedPortDefinition () : ConjugatedPortDefinition [1] {redefines type, abstract}
- portDefinition () : PortDefinition [1]{abstract}

7.4.2.2.6 ConnectionUsage_Init

Description

Initializes the properties of the SysML v2 element ConnectionUsage.

Generalizations

- PartUsage_Init (from SystemInitializers)

Attributes

- to : ConnectionUsage [1]

7.4.2.2.7 ConstraintDefinition_Init

Description

Initializes the properties of the SysML v2 element ConstraintDefinition.

Generalizations

- Definition_Init (from SystemInitializers)

Attributes

- to : ConstraintDefinition [1]

7.4.2.2.8 ConstraintUsage_Init

Description

Initializes the properties of the SysML v2 element ConstraintUsage.

Generalizations

- Usage_Init (from SystemInitializers)

Attributes

- to : ConstraintUsage [1]

7.4.2.2.9 Definition_Init

Description

Initializes the properties of the SysML v2 element Definition.

Generalizations

- Classifier_Init (from KerMLInitializers)

Attributes

- to : Definition [1]

Operations

- isVariation () : Boolean [1]

```
false
```

7.4.2.2.10 EventOccurerenceUsage_Init

Description

Initializes the properties of the SysML v2 element EventOccurrenceUsage.

Generalizations

- OccurrenceUsage_Init (from SystemInitializers)

Attributes

- to : EventOccurrenceUsage [1]

7.4.2.2.11 ItemDefinition_Init

Description

Initializes the properties of the SysML v2 element ItemDefinition.

Generalizations

- Definition_Init (from SystemInitializers)

Attributes

- to : ItemDefinition [1]

7.4.2.2.12 MetadataUsage_Init

Description

Initializes the properties of the SysML v2 element MetadataUsage.

Generalizations

- Usage_Init (from SystemInitializers)

Attributes

- to : MetadataUsage [1]

7.4.2.2.13 ObjectiveMembership_Init

Description

Initializes the properties of the SysML v2 element ObjectiveMembership.

Generalizations

- FeatureMembership_Init (from KerMLInitializers)

Attributes

- to : ObjectiveMembership [1]

7.4.2.2.14 OccurenceDefinition_Init

Description

Initializes the properties of the SysML v2 element OccurrenceDefinition.

Generalizations

- Definition_Init (from SystemInitializers)

Attributes

- to : OccurrenceDefinition [1]

Operations

- isIndividual () : Boolean [1]

```
false
```

7.4.2.2.15 OccurrenceUsage_Init

Description

Initializes the properties of the SysML v2 element OccurrenceUsage.

Generalizations

- Usage_Init (from SystemInitializers)

Attributes

- to : OccurrenceUsage [1]

Operations

- isIndividual () : Boolean [1]

```
false
```

- portionKind () : PortionKind [1]{abstract}

7.4.2.2.16 PartUsage_Init

Description

Initializes the properties of the SysML v2 element PartUsage.

Generalizations

- Usage_Init (from SystemInitializers)

Attributes

- to : PartUsage [1]

7.4.2.2.17 PortConjugation_Init

Description

Initializes the properties of the SysML v2 element PortConjugation.

Generalizations

- Conjugation_Init (from KerMLInitializers)

Attributes

- to : PortConjugation [1]

Operations

- originalPortDefinition () : PortDefinition [1] {redefines originalType, abstract}

7.4.2.2.18 PortDefinition_Init

Description

Initializes the properties of the SysML v2 element PortDefinition.

Generalizations

- Definition_Init (from SystemInitializers)

Attributes

- to : PortDefinition [1]

7.4.2.2.19 ReferenceUsage_Init

Description

Provides the basic features to map to a ReferenceUsage element.

Generalizations

- Usage_Init (from SystemInitializers)

Attributes

- to : ReferenceUsage [1]

7.4.2.2.20 RequirementUsage_Init

Description

Initializes the properties of the SysML v2 element RequirementUsage.

Generalizations

- Usage_Init (from SystemInitializers)

Attributes

- to : RequirementUsage [1]

7.4.2.2.21 StateUsage_Init

Description

Initializes the properties of the SysML v2 element StateUsage.

Generalizations

- ActionUsage_Init (from SystemInitializers)

Attributes

- to : StateUsage [1]

7.4.2.2.22 SubjectMembership_Init

Description

Initializes the properties of the SysML v2 element SubjectMembership.

Generalizations

- ParameterMembership_Init (from KerMLInitializers)

Attributes

- to : SubjectMembership [1]

7.4.2.2.23 Usage_Init

Description

Initializes the properties of the SysML v2 element Usage.

Generalizations

- Feature_Init (from KerMLInitializers)

Attributes

- to : Usage [1]

Operations

- isVariation () : Boolean [1]

```
false
```

7.5 Factories

7.5.1 Overview

The classes presented in this subclause specify facilities for creating elements in the target model form an arbitrary set of zero to many input parameters. After the target element is created, no link between it and an the value of inputs parameter (if any) will be preserved.

7.5.2 Mapping Specifications

7.5.2.1 EmptySubject_Factory

Description

Factory class to create a reference usage representing a subject without a source in the SysML v1 model.

Generalizations

- Factory (from Foundations)
- ReferenceUsage_Init (from SystemInitializers)

Operations

- create () : ReferenceUsage [1]
- direction () : FeatureDirectionKind [0..1] {redefines direction}

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

7.5.2.2 EmptySubjectMembership_Factory

Description

Factory class to create a memberhsip relationship for a reference usage representing a subject without a source in the SysML v1 model.

Generalizations

- Factory (from Foundations)
- SubjectMembership_Init (from SystemInitializers)

Operations

- create () : SubjectMembership [1]
- ownedMemberParameter () : Feature [1] {redefines ownedMemberParameter}

```
EmptySubject_Factory.create()
```

7.5.2.3 LiteralBoolean_Factory

Description

Factory class to create a LiteralBoolean element.

Generalizations

- Expression_Init (from KerMLInitializers)
- Factory (from Foundations)

Association Ends

- boolean : Boolean [1]
- to : LiteralBoolean [1]
  (redefines: Expression_Init::to)

Operations

- create (in boolean : Boolean) : LiteralBoolean [1]
- ownedRelationship () : Relationship [0..*] {redefines ownedRelationship}

```
Set{ReturnParameterFeatureMembership_Factory.create()}
```

7.5.2.4 LiteralNull_Factory

Description

Factory class to create a LiteralNull element.

Generalizations

- Expression_Init (from KerMLInitializers)
- Factory (from Foundations)

Association Ends

- to : NullExpression [1]
  (redefines: Expression_Init::to)

Operations

- create () : NullExpression [1]

7.5.2.5 LiteralRational_Factory

Description

Factory class to create a LiteralRational element.

Generalizations

- Expression_Init (from KerMLInitializers)
- Factory (from Foundations)

Association Ends

- real : Real [1]
- to : LiteralRational [1]
  (redefines: Expression_Init::to)

Operations

- create (in real : Real) : LiteralReal [1]
- ownedRelationship () : Relationship [0..*] {redefines ownedRelationship}

```
Set{ReturnParameterFeatureMembership_Factory.create()}
```

7.5.2.6 LiteralString_Factory

Description

Factory class to create a LiteralString element.

Generalizations

- Expression_Init (from KerMLInitializers)
- Factory (from Foundations)

Association Ends

- string : String [1]
- to : LiteralString [1]
  (redefines: Expression_Init::to)

Operations

- create (in string : String) : LiteralString [1]
- ownedRelationship () : Relationship [0..*] {redefines ownedRelationship}

```
Set{ReturnParameterFeatureMembership_Factory.create()}
```

7.5.2.7 ReturnParameterFeature_Factory

Description

Factory class to create a feature element with direction 'out' representing a return parameter.

Generalizations

- Factory (from Foundations)
- Feature_Init (from KerMLInitializers)

Operations

- create () : Feature [1]
- direction () : FeatureDirectionKind [0..1] {redefines direction}

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

7.5.2.8 ReturnParameterFeatureMembership_Factory

Description

Factory class to create a feature membership relationship for a feature element with direction 'out' representing a return parameter.

Generalizations

- Factory (from Foundations)
- ReturnParameterMembership_Init (from KerMLInitializers)

Operations

- create () : ReturnParameterMembership [1]
- ownedMemberParameter () : Feature [1] {redefines ownedMemberParameter}

```
ReturnParameterFeature_Factory.create()
```

7.5.2.9 StringParameterFeature_Factory

Description

Factory class to create a feature element representing a string.

Generalizations

- Factory (from Foundations)
- Feature_Init (from KerMLInitializers)

Association Ends

- string : String [1]

Operations

- create (in string : String) : Feature [1]
- ownedRelationship () : Relationship [0..*] {redefines ownedRelationship}

```
Set{StringParameterFeatureValue_Factory.create(string)}
```

7.5.2.10 StringParameterFeatureValue_Factory

Description

Factory class to create a string feature value relationship for a feature element.

Generalizations

- Factory (from Foundations)
- FeatureValue_Init (from KerMLInitializers)

Association Ends

- string : String [1]

Operations

- create (in string : String) : FeatureValue [1]
- value () : Expression [1] {redefines value}

```
LiteralString_Factory.create(string)
```

7.5.2.11 StringParameterMembership_Factory

Description

Factory class to create a parameter membership relationship for a feature element representing a string.

Generalizations

- Factory (from Foundations)
- ParameterMembership_Init (from KerMLInitializers)

Association Ends

- string : String [1]

Operations

- create (in string : String) : ParameterMembership [1]
- ownedMemberParameter () : Feature [1] {redefines ownedMemberParameter}

```
StringParameterFeature_Factory.create(string)
```

7.5.2.12 SubjectMembership_Factory

Description

Factory class to create a subject membership relationship for a given subject.

Generalizations

- Factory (from Foundations)
- SubjectMembership_Init (from SystemInitializers)

Association Ends

- subject : Type [1]

Operations

- create (in subject : Type) : SubjectMembership [1]
- ownedMemberParameter () : Feature [1] {redefines ownedMemberParameter}

```
subject
```

7.6 Generic Mappings

7.6.1 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. They are similar to initializers, except that they have a source element defined. 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 these generic mappings are abstract.

7.6.2 Common Mappings

7.6.2.1 CommonAssignmentActionUsage_Mapping

Description

Common mapping class to create an assignment action usage.

General Mappings

GenericToAssignmentActionUsage_Mapping

Mapping Source

Action

Mapping Target

AssignmentActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{CommonAssignmentActionUsageReplacementParameterMembership_Mapping.getMapped(from),
CommonAssignmentActionUsageTargetParameterMembership_Mapping.getMapped(from)}
```

7.6.2.2 CommonAssignmentActionUsageOwningMembership_Mapping

Description

Creates a owning membership relationship for *ownedMemberElement()*.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Action

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
CommonAssignmentActionUsage_Mapping.getMapped(from)
```

7.6.2.3 CommonAssignmentActionUsageReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Action

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.6.2.4 CommonAssignmentActionUsageReferenceUsage2_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Action

Mapping Target

ReferenceUsage

Owned Mappings

(none)

7.6.2.5 CommonAssignmentActionUsageReplacementParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

Action

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
CommonAssignmentActionUsageReplacementReferenceUsage_Mapping.getMapped(from)
```

7.6.2.6 CommonAssignmentActionUsageReplacementReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Action

Mapping Target

ReferenceUsage

Owned Mappings

(none)

7.6.2.7 CommonAssignmentActionUsageTargetFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Action

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
CommonAssignmentActionUsageTargetReferenceUsage_Mapping.getMapped(from)
```

7.6.2.8 CommonAssignmentActionUsageTargetParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

Action

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
CommonAssignmentActionUsageReferenceUsage_Mapping.getMapped(from)
```

7.6.2.9 CommonAssignmentActionUsageTargetReferenceFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Action

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
CommonAssignmentActionUsageReferenceUsage2_Mapping.getMapped(from)
```

7.6.2.10 CommonAssignmentActionUsageTargetReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Action

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.6.2.11 CommonFeatureReferenceExpression_Mapping

Description

Common mapping class for a feature reference expression.

General Mappings

GenericToFeatureReferenceExpression_Mapping

Mapping Source

TypedElement

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{CommonMembership_Mapping.getMapped(from),
CommonReturnParameterFeatureMembership_Mapping.getMapped(from)}
```

7.6.2.12 CommonMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToMembership_Mapping

Mapping Source

TypedElement

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 from
```

7.6.2.13 CommonParameterReferenceUsageInMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

Element

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.6.2.14 CommonParameterReferenceUsageIn_Mapping

Description

Common mapping class that creates a parameter reference usage element with direction 'in' and with a type.

General Mappings

CommonParameterReferenceUsageInUntyped_Mapping

Mapping Source

Element

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 if from.oclIsKindOf(UML::TypedElement) then
Set{CommonParameterReferenceUsageInFeatureTyping_Mapping.getMapped(from)}
else Set{} endif
```

7.6.2.15 CommonParameterReferenceUsageInFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Element

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- 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
```

7.6.2.16 CommonParameterReferenceUsageInUntyped_Mapping

Description

Common mapping class that creates a parameter reference usage element with direction 'in' and without a type.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Element

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.6.2.17 CommonReturnParameterFeature_Mapping

Description

Common mapping class that creates a parameter feature element with a type.

General Mappings

CommonReturnParameterFeatureUntyped_Mapping

Mapping Source

Element

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 if from.oclIsKindOf(UML::Property) then
    Set{CommonReturnParameterFeatureTyping_Mapping.getMapped(from)}
else
    Set{}
endif
```

7.6.2.18 CommonReturnParameterFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Element

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- 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
```

7.6.2.19 CommonReturnParameterFeatureUntyped_Mapping

Description

Common mapping class that creates a parameter feature element without a type.

General Mappings

GenericToFeature_Mapping

Mapping Source

Element

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.6.2.20 CommonReturnParameterFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToReturnParameterMembership_Mapping

Mapping Source

Element

Mapping Target

ReturnParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.6.2.21 CommonReturnParameterReferenceUsageMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToReturnParameterMembership_Mapping

Mapping Source

Element

Mapping Target

ReturnParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.6.2.22 CommonReturnParameterReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

CommonReturnParameterReferenceUsageUntyped_Mapping

Mapping Source

Element

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 if from.oclIsKindOf(UML::TypedElement) then
Set{CommonReturnParameterReferenceUsageFeatureTyping_Mapping.getMapped(from)}
else Set{} endif
```

7.6.2.23 CommonReturnParameterReferenceUsageFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Element

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- 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
```

7.6.2.24 CommonReturnParameterReferenceUsageUntyped_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Element

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.6.2.25 CommonReferenceUsageIn_Mapping

Description

Common mapping class that creates a reference usage element with direction 'in'.

General Mappings

CommonReferenceUsageInUntyped_Mapping

Mapping Source

TypedElement

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  Common mapping class that creates a reference usage element with direction 'in'.

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

7.6.2.26 CommonReferenceUsageInFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

TypedElement

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
   if from.type.oclIsUndefined() then
      CommonReferenceUsageInUntyped_Mapping.getMapped(from)
  else
      CommonReferenceUsageIn_Mapping.getMapped(from)
  endif
  ```

7.6.2.27 CommonReferenceUsageInFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

TypedElement

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.6.2.28 CommonReferenceUsageInUntyped_Mapping

Description

Common mapping class that creates an untyped reference usage element with direction 'in'.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

TypedElement

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  from.name
  ```

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

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

7.6.3 Generic Mappings To KerML

7.6.3.1 GenericToAnnotatingElement_Mapping

Description

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

General Mappings

GenericToElement_Mapping

Mapping Source

Element

Mapping Target

AnnotatingElement

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  Set{}
  ```

7.6.3.2 GenericToAnnotation_Mapping

Description

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

General Mappings

GenericToRelationship_Mapping

Mapping Source

Element

Mapping Target

Annotation

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific 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

7.6.3.3 GenericToAssociation_Mapping

Description

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

General Mappings

GenericToRelationship_Mapping
GenericToClassifier_Mapping

Mapping Source

Element

Mapping Target

Association

Owned Mappings

(none)

7.6.3.4 GenericToBehavior_Mapping

Description

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

General Mappings

GenericToClassifier_Mapping

Mapping Source

Element

Mapping Target

Behavior

Owned Mappings

(none)

7.6.3.5 GenericToClassifier_Mapping

Description

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

General Mappings

GenericToType_Mapping

Mapping Source

Element

Mapping Target

Classifier

Owned Mappings

(none)

7.6.3.6 GenericToComment_Mapping

Description

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

General Mappings

GenericToAnnotatingElement_Mapping

Mapping Source

Element

Mapping Target

Comment

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  null
  ```

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

7.6.3.7 GenericToConjugation_Mapping

Description

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

General Mappings

GenericToRelationship_Mapping

Mapping Source

Element

Mapping Target

Conjugation

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

7.6.3.8 GenericToConnector_Mapping

Description

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

General Mappings

GenericToFeature_Mapping
GenericToRelationship_Mapping

Mapping Source

Element

Mapping Target

Connector

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
false
```

7.6.3.9 GenericToDocumentation_Mapping

Description

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

General Mappings

GenericToComment_Mapping

Mapping Source

Element

Mapping Target

Documentation

Owned Mappings

(none)

7.6.3.10 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

Element

Mapping Target

Element

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  Helper.createUUID()
  ```

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

  ```
  Set{}
  ```

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

  ```
  null
  ```

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

  ```
  null
  ```

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

```
Set{}
```

7.6.3.11 GenericToEndFeatureMembership_Mapping

Description

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

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Element

Mapping Target

EndFeatureMembership

Owned Mappings

(none)

7.6.3.12 GenericToExpression_Mapping

Description

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

General Mappings

GenericToStep_Mapping

Mapping Source

Element

Mapping Target

Expression

Owned Mappings

(none)

7.6.3.13 GenericToFeature_Mapping

Description

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

General Mappings

GenericToType_Mapping

Mapping Source

Element

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  false
  ```

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

  ```
  false
  ```

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

  ```
  false
  ```

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

  ```
  null
  ```

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

  ```
  false
  ```

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

  ```
  false
  ```

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

  ```
  true
  ```

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

  ```
  false
  ```

7.6.3.14 GenericToFeatureChainExpression_Mapping

Description

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

General Mappings

GenericToOperatorExpression_Mapping

Mapping Source

Element

Mapping Target

FeatureChainExpression

Owned Mappings

(none)

7.6.3.15 GenericToFeatureChaining_Mapping

Description

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

General Mappings

GenericToRelationship_Mapping

Mapping Source

Element

Mapping Target

FeatureChaining

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

7.6.3.16 GenericToFeatureMembership_Mapping

Description

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

General Mappings

GenericToOwningMembership_Mapping
GenericToTypeFeaturing_Mapping

Mapping Source

Element

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.6.3.17 GenericToFeatureReferenceExpression_Mapping

Description

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

General Mappings

GenericToExpression_Mapping

Mapping Source

Element

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

7.6.3.18 GenericToFeatureTyping_Mapping

Description

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

General Mappings

GenericToSpecialization_Mapping

Mapping Source

Element

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

7.6.3.19 GenericToFeatureValue_Mapping

Description

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

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Element

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  false
  ```

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

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

- FeatureValue::value () : Expression [1]
  abstract rule
- FeatureValue::isDefault () : Boolean [1]

  ```
  false
  ```

7.6.3.20 GenericToFunction_Mapping

Description

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

General Mappings

GenericToBehavior_Mapping

Mapping Source

Element

Mapping Target

Function

Owned Mappings

(none)

7.6.3.21 GenericToImport_Mapping

Description

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

General Mappings

GenericToRelationship_Mapping

Mapping Source

Element

Mapping Target

Import

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Import::importedMemberName () : String [0..1]

  ```
  null
  ```

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

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

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

  ```
  false
  ```

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

  ```
  false
  ```

7.6.3.22 GenericToInvocationExpression_Mapping

Description

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

General Mappings

GenericToExpression_Mapping

Mapping Source

Element

Mapping Target

InvocationExpression

Owned Mappings

(none)

7.6.3.23 GenericToInteraction_Mapping

Description

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

General Mappings

GenericToBehavior_Mapping
GenericToAssociation_Mapping

Mapping Source

Element

Mapping Target

Interaction

Owned Mappings

(none)

7.6.3.24 GenericToItemFlow_Mapping

Description

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

General Mappings

GenericToConnector_Mapping

Mapping Source

Element

Mapping Target

ItemFlow

Owned Mappings

(none)

7.6.3.25 GenericToMembership_Mapping

Description

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

General Mappings

GenericToRelationship_Mapping

Mapping Source

Element

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  null
  ```

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

  ```
  null
  ```

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

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

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

7.6.3.26 GenericToMembershipImport_Mapping

Description

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

General Mappings

GenericToImport_Mapping

Mapping Source

Element

Mapping Target

MembershipImport

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- MembershipImport::importedMembership () : Namespace [1]
  abstract rule

7.6.3.27 GenericToNamespace_Mapping

Description

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

General Mappings

GenericToElement_Mapping

Mapping Source

Element

Mapping Target

Namespace

Owned Mappings

(none)

7.6.3.28 GenericToNamespaceImport_Mapping

Description

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

General Mappings

GenericToImport_Mapping

Mapping Source

Element

Mapping Target

NamespaceImport

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- NamespaceImport::importedNamespace () : Namespace [1]
  abstract rule

7.6.3.29 GenericToOperatorExpression_Mapping

Description

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

General Mappings

GenericToExpression_Mapping

Mapping Source

Element

Mapping Target

OperatorExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- OperatorExpression::operator () : String [1]
  abstract rule

7.6.3.30 GenericToOwningMembership_Mapping

Description

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

General Mappings

GenericToMembership_Mapping

Mapping Source

Element

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]
  abstract rule
- OwningMembership::ownedRelatedElement () : Element [0..*]

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

7.6.3.31 GenericToPackage_Mapping

Description

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

General Mappings

GenericToNamespace_Mapping

Mapping Source

Element

Mapping Target

Package

Owned Mappings

(none)

7.6.3.32 GenericToParameterMembership_Mapping

Description

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

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Element

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  null
  ```

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

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

7.6.3.33 GenericToPredicate_Mapping

Description

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

General Mappings

GenericToFunction_Mapping

Mapping Source

Element

Mapping Target

Predicate

Owned Mappings

(none)

7.6.3.34 GenericToRedefinition_Mapping

Description

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

General Mappings

GenericToSubsetting_Mapping

Mapping Source

Element

Mapping Target

Redefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

7.6.3.35 GenericToReferenceSubsetting_Mapping

Description

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

General Mappings

GenericToSubsetting_Mapping

Mapping Source

Element

Mapping Target

ReferenceSubsetting

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

7.6.3.36 GenericToRelationship_Mapping

Description

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

General Mappings

GenericToElement_Mapping

Mapping Source

Element

Mapping Target

Relationship

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  Set{}
  ```

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

  ```
  Set{}
  ```

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

  ```
  Set{}
  ```

7.6.3.37 GenericToReturnParameterMembership_Mapping

Description

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

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

Element

Mapping Target

ReturnParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific 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
  ```

7.6.3.38 GenericToSpecialization_Mapping

Description

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

General Mappings

GenericToRelationship_Mapping

Mapping Source

Element

Mapping Target

Specialization

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

7.6.3.39 GenericToStep_Mapping

Description

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

General Mappings

GenericToFeature_Mapping

Mapping Source

Element

Mapping Target

Step

Owned Mappings

(none)

7.6.3.40 GenericToSubclassification_Mapping

Description

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

General Mappings

GenericToSpecialization_Mapping

Mapping Source

Element

Mapping Target

Subclassification

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Subclassification::subclassifier () : Classifier [1]

  ```
  null
  ```

- Subclassification::superclassifier () : Classifier [1]

  ```
  null
  ```

7.6.3.41 GenericToSubsetting_Mapping

Description

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

General Mappings

GenericToSpecialization_Mapping

Mapping Source

Element

Mapping Target

Subsetting

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Subsetting::subsettedFeature () : Feature [1]
  abstract rule
- Subsetting::ownedRelatedElement () : Element [0..*]

```
Set{}
```

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

```
from
```

7.6.3.42 GenericToSuccession_Mapping

Description

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

General Mappings

GenericToConnector_Mapping

Mapping Source

Element

Mapping Target

Succession

Owned Mappings

(none)

7.6.3.43 GenericToSuccessionItemFlow_Mapping

Description

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

General Mappings

GenericToSuccession_Mapping
GenericToItemFlow_Mapping

Mapping Source

Element

Mapping Target

SuccessionItemFlow

Owned Mappings

(none)

7.6.3.44 GenericToTextualRepresentation_Mapping

Description

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

General Mappings

GenericToAnnotatingElement_Mapping

Mapping Source

Element

Mapping Target

TextualRepresentation

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

7.6.3.45 GenericToType_Mapping

Description

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

General Mappings

GenericToNamespace_Mapping

Mapping Source

Element

Mapping Target

Type

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
false
```

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

```
false
```

7.6.3.46 GenericToTypeFeaturing_Mapping

Description

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

General Mappings

GenericToRelationship_Mapping

Mapping Source

Element

Mapping Target

TypeFeaturing

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

7.6.4 Generic Mappings to Systems

7.6.4.1 GenericToActionUsage_Mapping

Description

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

General Mappings

GenericToUsage_Mapping
GenericToStep_Mapping

Mapping Source

Element

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  true
  ```

7.6.4.2 GenericToActorMembership_Mapping

Description

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

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

Element

Mapping Target

ActorMembership

Owned Mappings

(none)

7.6.4.3 GenericToAssignmentActionUsage_Mapping

Description

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

General Mappings

GenericToActionUsage_Mapping

Mapping Source

Element

Mapping Target

AssignmentActionUsage

Owned Mappings

(none)

7.6.4.4 GenericToConnectionUsage_Mapping

Description

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

General Mappings

GenericToPartUsage_Mapping

Mapping Source

Element

Mapping Target

ConnectionUsage

Owned Mappings

(none)

7.6.4.5 GenericToConjugatedPortDefinition_Mapping

Description

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

General Mappings

GenericToPortDefinition_Mapping

Mapping Source

Element

Mapping Target

ConjugatedPortDefinition

Owned Mappings

(none)

7.6.4.6 GenericToConjugatedPortTyping_Mapping

Description

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

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Element

Mapping Target

ConjugatedPortTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

7.6.4.7 GenericToConstraintDefinition_Mapping

Description

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

General Mappings

GenericToDefinition_Mapping

Mapping Source

Element

Mapping Target

ConstraintDefinition

Owned Mappings

(none)

7.6.4.8 GenericToConstraintUsage_Mapping

Description

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

General Mappings

GenericToUsage_Mapping

Mapping Source

Element

Mapping Target

ConstraintUsage

Owned Mappings

(none)

7.6.4.9 GenericToDefinition_Mapping

Description

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

General Mappings

GenericToClassifier_Mapping

Mapping Source

Element

Mapping Target

Definition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
false
```

7.6.4.10 GenericToEventOccurerenceUsage_Mapping

Description

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

General Mappings

GenericToOccurrenceUsage_Mapping

Mapping Source

Element

Mapping Target

EventOccurrenceUsage

Owned Mappings

(none)

7.6.4.11 GenericToItemDefinition_Mapping

Description

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

General Mappings

GenericToDefinition_Mapping

Mapping Source

Element

Mapping Target

ItemDefinition

Owned Mappings

(none)

7.6.4.12 GenericToMetadataUsage_Mapping

Description

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

General Mappings

GenericToUsage_Mapping

Mapping Source

Element

Mapping Target

MetadataUsage

Owned Mappings

(none)

7.6.4.13 GenericToObjectiveMembership_Mapping

Description

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

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Element

Mapping Target

ObjectiveMembership

Owned Mappings

(none)

7.6.4.14 GenericToOccurenceDefinition_Mapping

Description

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

General Mappings

GenericToDefinition_Mapping

Mapping Source

Element

Mapping Target

OccurrenceDefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
false
```

7.6.4.15 GenericToOccurrenceUsage_Mapping

Description

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

General Mappings

GenericToUsage_Mapping

Mapping Source

Element

Mapping Target

OccurrenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  OclUndefined
  ```

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

  ```
  false
  ```

7.6.4.16 GenericToPartUsage_Mapping

Description

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

General Mappings

GenericToUsage_Mapping

Mapping Source

Element

Mapping Target

PartUsage

Owned Mappings

(none)

7.6.4.17 GenericToPortConjugation_Mapping

Description

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

General Mappings

GenericToConjugation_Mapping

Mapping Source

Element

Mapping Target

PortConjugation

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

7.6.4.18 GenericToPortDefinition_Mapping

Description

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

General Mappings

GenericToDefinition_Mapping

Mapping Source

Element

Mapping Target

PortDefinition

Owned Mappings

(none)

7.6.4.19 GenericToReferenceUsage_Mapping

Description

Provides the basic features to map to a ReferenceUsage element.

General Mappings

GenericToUsage_Mapping

Mapping Source

Element

Mapping Target

ReferenceUsage

Owned Mappings

(none)

7.6.4.20 GenericToRequirementUsage_Mapping

Description

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

General Mappings

GenericToUsage_Mapping

Mapping Source

Element

Mapping Target

RequirementUsage

Owned Mappings

(none)

7.6.4.21 GenericToStateUsage_Mapping

Description

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

General Mappings

GenericToActionUsage_Mapping

Mapping Source

Element

Mapping Target

StateUsage

Owned Mappings

(none)

7.6.4.22 GenericToSubjectMembership_Mapping

Description

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

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

Element

Mapping Target

SubjectMembership

Owned Mappings

(none)

7.6.4.23 GenericToUsage_Mapping

Description

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

General Mappings

GenericToFeature_Mapping

Mapping Source

Element

Mapping Target

Usage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
false
```

7.7 Mappings from UML4SysML metaclasses

This chapter lists all mapping specifications of UML4SysML model elements.

7.7.1 Overview

`UML4SysML` is the subset of UML containing all model elements that are reused by SysML. The complete list of model elements is defined in [SysMLv1], subclause 4.1.

7.7.2 Actions

This chapter lists all mapping specifications of UML4SysML::Actions model elements.

7.7.2.1 Overview

The following table gives an overview of which SysML v2 elements the UML4SysML::Actions elements are transformed with which mapping class. The mapping details are in 7.7.2.3.

The justifications for the elements without mapping are given in 7.7.2.2.

Table 1. List of all mappings

SysML v1 Concept	SysML v2 Concept
AcceptCallAction	AcceptActionUsage
AcceptEventAction	AcceptActionUsage AcceptActionUsage AcceptActionUsage
ActionInputPin	
AddStructuralFeatureValueAction	ActionUsage
AddVariableValueAction	ActionUsage
BroadcastSignalAction	ActionUsage
CallBehaviorAction	ActionUsage
CallOperationAction	ActionUsage
Clause	
ClearAssociationAction	ActionUsage
ClearStructuralFeatureAction	ActionUsage
ClearVariableAction	ActionUsage
ConditionalNode	
CreateLinkAction	ActionUsage
CreateLinkObjectAction	ActionUsage
CreateObjectAction	ActionUsage
DestroyLinkAction	ActionUsage
DestroyObjectAction	ActionUsage
InputPin	ReferenceUsage ReferenceUsage
LinkEndCreationData	
LinkEndData	
LinkEndDestructionData	
LoopNode	ActionUsage
OpaqueAction	ActionUsage

SysML v1 Concept	SysML v2 Concept
OutputPin	ReferenceUsage ReferenceUsage ReferenceUsage ReferenceUsage ReferenceUsage ReferenceUsage ReferenceUsage
RaiseExceptionAction	ActionUsage
ReadExtentAction	ActionUsage
ReadIsClassifiedObjectAction	ActionUsage
ReadLinkAction	ActionUsage
ReadLinkObjectEndAction	ActionUsage
ReadSelfAction	ActionUsage
ReadStructuralFeatureAction	ActionUsage
ReadVariableAction	ActionUsage
ReclassifyObjectAction	ActionUsage
ReduceAction	ActionUsage
RemoveStructuralFeatureValueAction	ActionUsage
RemoveVariableValueAction	ActionUsage
ReplyAction	ActionUsage
SendObjectAction	ActionUsage
SendSignalAction	ActionUsage ActionUsage
SequenceNode	ActionUsage
StartClassifierBehaviorAction	ActionUsage
StartObjectBehaviorAction	ActionUsage
StructuredActivityNode	ActionUsage ActionUsage ActionUsage
TestIdentityAction	CalculationUsage
UnmarshallAction	ActionUsage
ValuePin	ReferenceUsage ReferenceUsage
ValueSpecificationAction	ActionUsage

7.7.2.2 UML4SysML::Actions elements not mapped

Table 2. 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.
Clause	Mapping is not specified yet.
ConditionalNode	Mapping is not specified yet.
LinkEndCreationData	Mapping is not specified yet.
LinkEndData	Mapping is not specified yet.
LinkEndDestructionData	Mapping is not specified yet.
ReclassifyObjectAction	The UML4SysML::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 UML4SysML::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 UML4SysML::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.
UnmarshallAction	Mapping is not specified yet.

7.7.2.3 Mapping Specifications

The mapping specifications are divided into sections corresponding to the structure in which the UML 2.5.1 specification groups the actions.

The name of the primary mapping class is "<SysML v1 action name>_Mapping". The secondary mapping classes for the further mapping structures start with an abbreviation of the SysML v1 action name to reduce the overall length of the names.

7.7.2.3.1 Accept Event Actions

7.7.2.3.1.1 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)

7.7.2.3.1.2 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 acceptEventActionSignalEvent1 accept : SysMLv1Signal via sysMLv1Port;
action acceptEventActionChangeEvent1 accept when when changeExpression.result {
        calc changeExpression {
                return : ScalarValues::Boolean;
                language "OCL"
                        /*
                         * x > 0
                         */
        }
}
```

General Mappings

CommonAction_Mapping

Mapping Source

AcceptEventAction

Mapping Target

AcceptActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
let relationships : Set(KerML::Relationship) = Helper.actionOwnedRelationship(from)
->including(AEAParameterMembership_Mapping.getMapped(from))
->including(AEAReceiverParameterMembership_Mapping.getMapped(from)) in
if from.trigger.get(0).event.oclIsTypeOf(UML::ChangeEvent) then
relationships->including(ElementFeatureMembership_Mapping.getMapped(
    from.trigger.get(0).event.oclAsType(UML::ChangeEvent).changeExpression))
else relationships
endif
```

7.7.2.3.1.3 AEAChangeExpressionMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

AcceptEventAction

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
from.trigger.get(0).event.oclAsType(UML::ChangeEvent).changeExpression
```

7.7.2.3.1.4 AEAChangeParameter_Mapping

Description

The mapping class transforms the change event specified at the AcceptEventAction.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

AcceptEventAction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

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

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

7.7.2.3.1.5 AEAChangeParameterFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

AcceptEventAction

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  AEAChangeParameterTrigger_Mapping.getMapped(from)
  ```

7.7.2.3.1.6 AEAChangeParameterTrigger_Mapping

Description

The mapping class creates a TriggerInvocationExpression from the change event specified at the AcceptEventAction.

General Mappings

GenericToInvocationExpression_Mapping

Mapping Source

AcceptEventAction

Mapping Target

TriggerInvocationExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.2.3.1.7 AEAChangeParameterTriggerExpression_Mapping

Description

The mapping class creates the trigger expression element for the change parameter of the SysML v2 AcceptActionUsage element.

General Mappings

GenericToExpression_Mapping

Mapping Source

AcceptEventAction

Mapping Target

Expression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.2.3.1.8 AEAChangeParameterResultExpressionMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

AcceptEventAction

Mapping Target

ResultExpressionMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
AEAChangeParameterFeatureChainExpression_Mapping.getMapped(from)
```

7.7.2.3.1.9 AEAChangeParameterFeatureChainExpression_Mapping

Description

The mapping class creates the feature chain expression element for the change parameter of the SysML v2 AcceptActionUsage element.

General Mappings

GenericToInvocationExpression_Mapping

Mapping Source

AcceptEventAction

Mapping Target

FeatureChainExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.2.3.1.10 AEAChangeParameterFeature_Mapping

Description

The mapping class creates the feature for the feature chain expression element for the change parameter of the SysML v2 AcceptActionUsage element.

General Mappings

GenericToFeature_Mapping

Mapping Source

AcceptEventAction

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.2.3.1.11 AEAChangeParameterExpressionFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

AcceptEventAction

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
AEAChangeParameterFeatureReferenceExpression_Mapping.getMapped(from)
```

7.7.2.3.1.12 AEAChangeParameterFeatureReferenceExpression_Mapping

Description

The mapping class creates the feature reference expression for the feature chain expression element for the change parameter of the SysML v2 AcceptActionUsage element.

General Mappings

GenericToFeatureReferenceExpression_Mapping

Mapping Source

AcceptEventAction

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.2.3.1.13 AEAChangeParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToMembership_Mapping

Mapping Source

AcceptEventAction

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
from.trigger.get(0).event.oclAsType(UML::ChangeEvent).changeExpression
```

7.7.2.3.1.14 AEAChangeParameterParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

AcceptEventAction

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
AEAChangeParameterFeature_Mapping.getMapped(from)
```

7.7.2.3.1.15 AEAReceiverParameter_Mapping

Description

The mapping class creates the reference usage element for the receiver parameter of the SysML v2 AcceptActionUsage element.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

AcceptEventAction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
if from.trigger.get(0).port->size() > 0
then Set{AEAReceiverFeatureValue_Mapping.getMapped(from)}
else Set{}
endif
```

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

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

7.7.2.3.1.16 AEAReceiverParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

AcceptEventAction

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
AEAReceiverParameter_Mapping.getMapped(from)
```

7.7.2.3.1.17 AEAReceiverFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

AcceptEventAction

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
ACAReceiverFeatureReferenceExpression_Mapping.getMapped(from)
```

7.7.2.3.1.18 AEASignalParameter_Mapping

Description

The mapping class creates the reference usage element for the signal parameter of the SysML v2 AcceptActionUsage element.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

AcceptEventAction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

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

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

7.7.2.3.1.19 AEASignalParameterFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

AcceptEventAction

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- 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 OclUndefined endif
```

7.7.2.3.1.20 AEAParameterMembership_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 UML4SysML::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

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 if from.trigger.get(0).event.oclIsTypeOf(UML::SignalEvent) then
    AEASignalParameter_Mapping.getMapped(from)
else if from.trigger.get(0).event.oclIsTypeOf(UML::ChangeEvent) then
    AEAChangeParameter_Mapping.getMapped(from)
else
    OclUndefined
endif endif
```

7.7.2.3.1.21 AEAReceiverFeatureReferenceExpression_Mapping

Description

The mapping class creates the feature reference expression for the reference usage element for the receiver parameter of the SysML v2 AcceptActionUsage element.

General Mappings

GenericToFeatureReferenceExpression_Mapping

Mapping Source

AcceptEventAction

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{AEAReceiverFeatureReferenceExpressionMembership_Mapping.getMapped(from),
ReturnParameterFeatureMembership_Factory.create()}
```

7.7.2.3.1.22 AEAReceiverFeatureReferenceExpressionMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToMembership_Mapping

Mapping Source

AcceptEventAction

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 if from.trigger.get(0).port->size() > 0 then
    from.trigger.get(0).port.get(0)
else
    OclUndefined
endif
```

7.7.2.3.1.23 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)

7.7.2.3.1.24 UnmarshallAction_Mapping

Description

The mapping of UML4SysML::UnmarshallAction is not specified yet. It is currently mapped to an empty action usage to keep the connections within the activity respectively action definition.

General Mappings

CommonAction_Mapping

Mapping Source

UnmarshallAction

Mapping Target

ActionUsage

Owned Mappings

(none)

7.7.2.3.2 Actions

7.7.2.3.2.1 CommonAction_Mapping

Description

Base mapping class for model elements of kind UML4SysML::Action. The target element is a SysML v2 ActionUsage.

General Mappings

GenericToActionUsage_Mapping
NamedElementMain_Mapping

Mapping Source

Action

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  true
  ```

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

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

7.7.2.3.2.2 InputPin_Mapping

Description

A UML4SysML::InputPin is mapped to a SysMLv2 ReferenceUsage with direction 'in'. This mapping class transforms input pins with a specified type.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action sysMLv1Action {
  in parIn : SysMLv1Block;
}
part def SysMLv1Block;
```

General Mappings

Pin_Mapping

Mapping Source

InputPin

Mapping Target

ReferenceUsage

Owned Mappings

(none)

7.7.2.3.2.3 InputPinUntyped_Mapping

Description

A UML4SysML::InputPin is mapped to a SysMLv2 ReferenceUsage with direction 'in'. This mapping class transforms input pins without a specified type.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action sysMLv1Action {
  in parIn;
}
```

General Mappings

UntypedPin_Mapping

Mapping Source

InputPin

Mapping Target

ReferenceUsage

Owned Mappings

(none)

7.7.2.3.2.4 OpaqueAction_Mapping

Description

The UML4SysML::OpaqueAction is mapped to a SysML v2 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

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
if from.body->size() > 0 then
Helper.actionOwnedRelationship(from)->append(OABodyMembership_Mapping.getMapped(from))
else
Helper.actionOwnedRelationship(from)
endif
```

7.7.2.3.2.5 OABody_Mapping

Description

The languages and bodies of a UML4SysML::OpaqueAction are mapped to SysMLv2 TextualRepresentations.

General Mappings

GenericToAnnotatingElement_Mapping

Mapping Source

OpaqueAction

Mapping Target

TextualRepresentation

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific 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
```

7.7.2.3.2.6 OABodyMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

OpaqueAction

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
OABody_Mapping.getMapped(from)
```

7.7.2.3.2.7 OutputPin_Mapping

Description

A UML4SysML::OutputPin is mapped to a SysMLv2 ReferenceUsage with direction 'out'. This mapping class transforms output pins with a specified type.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action sysMLv1Action {
  out parIn : SysMLv1Block;
}
part def SysMLv1Block;
```

General Mappings

Pin_Mapping

Mapping Source

OutputPin

Mapping Target

ReferenceUsage

Owned Mappings

(none)

7.7.2.3.2.8 OutputPinUntyped_Mapping

Description

A UML4SysML::OutputPin is mapped to a SysMLv2 ReferenceUsage with direction 'in'. This mapping class transforms output pins without a specified type.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action sysMLv1Action {
  out parIn;
}
```

General Mappings

UntypedPin_Mapping

Mapping Source

OutputPin

Mapping Target

ReferenceUsage

Owned Mappings

(none)

7.7.2.3.2.9 Pin_Mapping

Description

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

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action def SysMLv1Activity {
        action sysMLv1Action {
                in sysMLv1TypedPin : ScalarValues::Integer;
        }
}
```

General Mappings

UntypedPin_Mapping

Mapping Source

Pin

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

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

```
not src.type.oclIsUndefined()
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{PinFeatureTyping_Mapping.getMapped(from),
MultiplicityMembership_Mapping.getMapped(from)}
```

7.7.2.3.2.10 PinFeatureTyping_Mapping

Description

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

General Mappings

TypedElementFeatureTyping_Mapping

Mapping Source

Pin

Mapping Target

FeatureTyping

Owned Mappings

(none)

7.7.2.3.2.11 UntypedPin_Mapping

Description

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

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action def SysMLv1Activity {
        action sysMLv1Action {
                in sysMLv1UntypedPin;
        }
}
```

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:

```
src.type.oclIsUndefined()
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

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

```
 ElementOwnership_Mapping.getMappedColl(from.ownedComment)
->including(MultiplicityMembership_Mapping.getMapped(from))
```

7.7.2.3.2.12 ValuePin_Mapping

Description

A UML4SysML::ValuePin is mapped to a SysML v2 ReferenceUsage with assigned value.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action sysMLv1Action {
        in sysMLv1ValuePin1 : ScalarValues::Integer = 42;
        in sysMLv1ValuePin2 = {
                return result;
                language "English"
                /*
                 * this is a opaque expression
                 */
                }.result;
}
```

General Mappings

Pin_Mapping

Mapping Source

ValuePin

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{PinFeatureTyping_Mapping.getMapped(from),
ValuePinFeatureValue_Mapping.getMapped(from),
MultiplicityMembership_Mapping.getMapped(from)}
```

7.7.2.3.2.13 ValuePinFeatureValue_Mapping

Description

The mapping class creates the value expression for the reference usage element.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

ValuePin

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
if from.value.oclIsUndefined() then OclUndefined else from.value endif
```

7.7.2.3.2.14 ValuePinUntyped_Mapping

Description

Same as ValuePin_Mapping, but for UML4SysML::ValuePins without a specified type.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action sysMLv1Action {
        in sysMLv1ValuePin1 = 42;
}
```

General Mappings

UntypedPin_Mapping

Mapping Source

ValuePin

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.2.3.3 Invocation Actions

7.7.2.3.3.1 BroadcastSignalAction_Mapping

Description

The UML4SysML::BroadcastSignalAction is mapped to a SysML v2 ActionUsage. The details of the mapping are not defined yet.

General Mappings

CommonAction_Mapping

Mapping Source

BroadcastSignalAction

Mapping Target

ActionUsage

Owned Mappings

(none)

7.7.2.3.3.2 CallBehaviorAction_Mapping

Description

A UML4SysML::CallBehaviorAction is mapped to a SysML v2 ActionUsage.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action def SysMLv1Activity1 {
        action sysMLv1CallBehaviorAction : SysMLv1Activity2;
}
action def SysMLv1Activity2;
```

General Mappings

CommonAction_Mapping

Mapping Source

CallBehaviorAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Helper.actionOwnedRelationship(from)
->append(CBAFeatureTyping_Mapping.getMapped(from))
```

7.7.2.3.3.3 CBAFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

CallBehaviorAction

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
        from.behavior
```

7.7.2.3.3.4 CallOperationAction_Mapping

Description

A UML4SysML::CallOperationAction is mapped to a SysML v2 ActionUsage which calls the operation.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action sysMLv1CallOperationAction {
  in paramIn;
  in target : ThisIsABlock;
  out paramReturn = target.sysMLv1Operation;
}
```

General Mappings

CommonAction_Mapping

Mapping Source

CallOperationAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Helper.actionOwnedRelationship(from)
->including(COAPerformActionFeatureMembership_Mapping.getMapped(from))
```

7.7.2.3.3.5 COAOutputPinFeature_Mapping

Description

The mapping class creates the feature element for the output parameter.

General Mappings

GenericToFeature_Mapping

Mapping Source

OutputPin

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{COAOutputPinFeatureFeatureValue_Mapping.getMapped(from),
COAOutputPinFeatureFeatureMembership_Mapping.getMapped(from)}
```

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

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

7.7.2.3.3.6 COAOutputPinFeatureChainExpression_Mapping

Description

The mapping class creates the feature chain expression for the output parameter feature value.

General Mappings

GenericToInvocationExpression_Mapping

Mapping Source

OutputPin

Mapping Target

FeatureChainExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{COAOutputPinParameterMembership_Mapping.getMapped(from),
COAOutputPinFeatureChainExpressionMembership_Mapping.getMapped(from),
ReturnParameterFeatureMembership_Factory.create()}
```

7.7.2.3.3.7 COAOutputPinFeatureChainExpressionMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToMembership_Mapping

Mapping Source

OutputPin

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.2.3.3.8 COAOutputPinFeatureFeature_Mapping

Description

Creates a feature element for the UML4SysML::CallOperationAction mapping.

General Mappings

GenericToFeature_Mapping

Mapping Source

OutputPin

Mapping Target

Feature

Owned Mappings

(none)

7.7.2.3.3.9 COAOutputPinFeatureFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

OutputPin

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
COAOutputPinFeatureFeature_Mapping.getMapped(from)
```

7.7.2.3.3.10 COAOutputPinFeatureFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

OutputPin

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
COAOutputPinFeatureReferenceExpression_Mapping.getMapped(from)
```

7.7.2.3.3.11 COAOutputPinFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

OutputPin

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
        COAOutputPinReferenceUsage_Mapping.getMapped(from)
```

7.7.2.3.3.12 COAOutputPinFeatureReferenceExpression_Mapping

Description

The mapping class creates the feature reference expression for the output parameter.

General Mappings

GenericToFeatureReferenceExpression_Mapping

Mapping Source

OutputPin

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{COAOutputPinFeatureReferenceExpressionMembership_Mapping.getMapped(from),
ReturnParameterFeatureMembership_Factory.create()}
```

7.7.2.3.3.13 COAOutputPinFeatureReferenceExpressionMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToMembership_Mapping

Mapping Source

OutputPin

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.2.3.3.14 COAOutputPinParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

OutputPin

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
COAOutputPinFeature_Mapping.getMapped(from)
```

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

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

7.7.2.3.3.15 COAOutputPinReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

OutputPin

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.2.3.3.16 COAOutputPinReferenceUsageFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

OutputPin

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
COAOutputPinFeatureChainExpression_Mapping.getMapped(from)
```

7.7.2.3.3.17 COAPerformAction_Mapping

Description

The mapping class creates the PerformActionUsage element.

General Mappings

GenericToActionUsage_Mapping

Mapping Source

CallOperationAction

Mapping Target

PerformActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.2.3.3.18 COAPerformActionFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToEndFeatureMembership_Mapping

Mapping Source

CallOperationAction

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
COAPerformAction_Mapping.getMapped(from)
```

7.7.2.3.3.19 COAPerformActionReferenceSubsetting_Mapping

Description

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

General Mappings

GenericToReferenceSubsetting_Mapping

Mapping Source

CallOperationAction

Mapping Target

ReferenceSubsetting

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.2.3.3.20 COAPerformActionFeature_Mapping

Description

The mapping class creates the feature element for the perform action usage.

General Mappings

GenericToFeature_Mapping

Mapping Source

CallOperationAction

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
   Set{COAPerformActionFeatureChainingTarget_Mapping.getMapped(from),
  COAPerformActionFeatureChainingOperation_Mapping.getMapped(from)}
  ```

7.7.2.3.3.21 COAPerformActionFeatureChainingOperation_Mapping

Description

The mapping class creates the feature chaining element for the operation of the perform action usage.

General Mappings

GenericToFeatureChaining_Mapping

Mapping Source

CallOperationAction

Mapping Target

FeatureChaining

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
from.operation
```

7.7.2.3.3.22 COAPerformActionFeatureChainingTarget_Mapping

Description

The mapping class creates the feature chaining element for the target element of the perform action usage.

General Mappings

GenericToFeatureChaining_Mapping

Mapping Source

CallOperationAction

Mapping Target

FeatureChaining

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
from.target
```

7.7.2.3.3.23 SendObjectAction_Mapping

Description

A UML4SysML::SendObjectAction is mapped to a SysMLv2 ActionUsage that includes a SendActionUsage.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action sysMLv1SendObjectAction {
        in target : SysMLv1Block;
        send SysMLv1Objectl() to target;
}
part def SysMLv1Block;
item def SysMLv1Object;
```

General Mappings

SendSignalAction_Mapping

Mapping Source

SendObjectAction

Mapping Target

ActionUsage

Owned Mappings

(none)

7.7.2.3.3.24 SendSignalAction_Mapping

Description

A UML4SysML::SendSignalAction is mapped to a SysMLv2 ActionUsage that includes a SendActionUsage.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action sysMLv1SendSignalAction {
        in target : SysMLv1Block;
        send SysMLv1Signal() to target;
}
part def SysMLv1Block;
item def SysMLv1Signal;
```

General Mappings

CommonAction_Mapping

Mapping Source

SendSignalAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Helper.actionOwnedRelationship(from)
->including(SSAFeatureMembership_Mapping.getMapped(from))
```

7.7.2.3.3.25 SSAFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

InvocationAction

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
SSASendActionUsage_Mapping.getMapped(from)
```

7.7.2.3.3.26 SSAParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

InvocationAction

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  SSAReferenceUsage_Mapping.getMapped(from)
  ```

7.7.2.3.3.27 SSAReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

InvocationAction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.2.3.3.28 SSAItemParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

InvocationAction

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
SSAItemReferenceUsage_Mapping.getMapped(from)
```

7.7.2.3.3.29 SSAItemReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

InvocationAction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

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

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

7.7.2.3.3.30 SSAItemReferenceUsageFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

InvocationAction

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
        SSAItemReferenceUsageInvocationExpression_Mapping.getMapped(from)
```

7.7.2.3.3.31 SSAItemReferenceUsageFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

InvocationAction

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 if from.oclIsTypeOf(UML::SendSignalAction) then
    from.signal
else if from.oclIsTypeOf(UML::SendObjectAction) then
    from.request
else
    OclUndefined
endif endif
```

7.7.2.3.3.32 SSAItemReferenceUsageInvocationExpression_Mapping

Description

The mapping class creates the invocation expression for the SysML v2 SendActionUsage.

General Mappings

GenericToInvocationExpression_Mapping

Mapping Source

InvocationAction

Mapping Target

InvocationExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{SSAItemReferenceUsageFeatureTyping_Mapping.getMapped(from),
ReturnParameterFeatureMembership_Factory.create()}
```

7.7.2.3.3.33 SSATargetParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

InvocationAction

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
SSATargetReferenceUsage_Mapping.getMapped(from)
```

7.7.2.3.3.34 SSATargetReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

InvocationAction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

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

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

7.7.2.3.3.35 SSATargetReferenceUsageFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

InvocationAction

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
SSATargetReferenceUsageFeatureValueExpression_Mapping.getMapped(from)
```

7.7.2.3.3.36 SSATargetReferenceUsageFeatureValueMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToMembership_Mapping

Mapping Source

InvocationAction

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
from.target
```

7.7.2.3.3.37 SSATargetReferenceUsageFeatureValueExpression_Mapping

Description

The mapping class creates the feature reference expression for the target reference usage element of the SysML v2 SendActionUsage.

General Mappings

GenericToFeatureReferenceExpression_Mapping

Mapping Source

InvocationAction

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{SSATargetReferenceUsageFeatureValueMembership_Mapping.getMapped(from),
ReturnParameterFeatureMembership_Factory.create()}
```

7.7.2.3.3.38 SSASendActionUsage_Mapping

Description

The mapping class creates the SysML v2 element SendActionUsage for the UML4SysML::SendSignalAction mapping.

General Mappings

GenericToActionUsage_Mapping

Mapping Source

InvocationAction

Mapping Target

SendActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{SSAItemParameterMembership_Mapping.getMapped(from),
SSAParameterMembership_Mapping.getMapped(from),
SSATargetParameterMembership_Mapping.getMapped(from)}
```

7.7.2.3.3.39 StartClassifierBehaviorAction_Mapping

Description

The UML4SysML::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)

7.7.2.3.3.40 StartObjectBehaviorAction_Mapping

Description

The UML4SysML::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)

7.7.2.3.4 Link Actions

7.7.2.3.4.1 ClearAssociationAction_Mapping

Description

The UML4SysML::ClearAssociationAction is mapped to a SysML v2 ActionUsage. The details of the mapping are not defined yet.

General Mappings

CommonAction_Mapping

Mapping Source

ClearAssociationAction

Mapping Target

ActionUsage

Owned Mappings

(none)

7.7.2.3.4.2 CreateLinkAction_Mapping

Description

The UML4SysML::CreateLinkAction is mapped to a SysML v2 ActionUsage. The details of the mapping are not completely defined yet.

General Mappings

CommonAction_Mapping

Mapping Source

CreateLinkAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
let linkEndCreationData : Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsTypeOf(UML::LinkEndCreationData)) in
let actionInputPin: Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsTypeOf(UML::ActionInputPin)) in
let triggers: Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Trigger)) in
let toElementFMS: Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Pin)) in
let toElementOMS: Set(UML::Element) =
    ((((from.ownedElement - toElementFMS) - actionInputPin)
        - triggers) - linkEndCreationData) in
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
```

7.7.2.3.4.3 CreateLinkObjectAction_Mapping

Description

A UML4SysML::CreateLinkObjectAction is mapped to a SysML v2 ActionUsage. The details of the mapping are not defined yet.

General Mappings

CommonAction_Mapping

Mapping Source

CreateLinkObjectAction

Mapping Target

ActionUsage

Owned Mappings

(none)

7.7.2.3.4.4 DestroyLinkAction_Mapping

Description

The UML4SysML::DestroyLinkAction is mapped to a SysML v2 ActionUsage. The details of the mapping are not completely defined yet.

General Mappings

CommonAction_Mapping

Mapping Source

DestroyLinkAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 let actionInputPin: Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsTypeOf(UML::ActionInputPin)) in
let triggers: Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Trigger)) in
let linkData: Set(UML::Element) =
    from.ownedElement->select( e | e.oclIsKindOf(UML::LinkEndData) or
    e.oclIsKindOf(UML::LinkEndDestructionData)) in
let toElementFMS: Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Pin)) in
let toElementOMS: Set(UML::Element) =
    (((from.ownedElement - toElementFMS) - actionInputPin)
       - triggers) - linkData) in
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
```

7.7.2.3.4.5 ReadLinkAction_Mapping

Description

The UML4SysML::ReadLinkAction is mapped to a SysML v2 ActionUsage. The details of the mapping are not completely defined yet.

General Mappings

CommonAction_Mapping

Mapping Source

ReadLinkAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 let actionInputPin: Set(UML::Element) =
     from.ownedElement->select(e | e.oclIsTypeOf(UML::ActionInputPin)) in
let triggers: Set(UML::Element) =
     from.ownedElement->select(e | e.oclIsKindOf(UML::Trigger)) in
let linkData: Set(UML::Element) =
     from.ownedElement->select( e | e.oclIsKindOf(UML::LinkEndData)) in
let toElementFMS: Set(UML::Element) =
     from.ownedElement->select(e | e.oclIsKindOf(UML::Pin)) in
let toElementOMS: Set(UML::Element) =
     ((((from.ownedElement - toElementFMS) - actionInputPin)
         - triggers) - linkData) in
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
```

7.7.2.3.4.6 ReadLinkObjectEndAction_Mapping

Description

The UML4SysML::ReadLinkObjectEndAction is mapped to a SysML v2 ActionUsage. The details of the mapping are not defined yet.

General Mappings

CommonAction_Mapping

Mapping Source

ReadLinkObjectEndAction

Mapping Target

ActionUsage

Owned Mappings

(none)

7.7.2.3.4.7 ReadLinkObjectEndQualifierAction_Mapping

Description

The UML4SysML::ReadLinkObjectEndQualifierAction is mapped to a SysML v2 ActionUsage. The details of the mapping are not defined yet.

General Mappings

CommonAction_Mapping

Mapping Source

ReadLinkObjectEndQualifierAction

Mapping Target

ActionUsage

Owned Mappings

(none)

7.7.2.3.5 Object Actions

7.7.2.3.5.1 CreateObjectAction_Mapping

Description

A UML4SysML::CreateObjectAction is mapped to a SysML v2 ActionUsage.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action def SysMLv1Activity {
        action sysMLv1CreateObjectAction {
                 out result : SysMLv1Block = SysMLv1Block();
        }
}
part def SysMLv1Block;
```

General Mappings

CommonAction_Mapping

Mapping Source

CreateObjectAction

Mapping Target

ActionUsage

Owned Mappings

(none)

7.7.2.3.5.2 COAInvocationExpessionFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

CreateObjectAction

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
from.classifier
```

7.7.2.3.5.3 COAInvocationExpression_Mapping

Description

The mapping class creates the invocation expression to create the object.

General Mappings

GenericToInvocationExpression_Mapping

Mapping Source

CreateObjectAction

Mapping Target

InvocationExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{COAInvocationExpessionFeatureTyping_Mapping.getMapped(from),
CommonReturnParameterFeatureMembership_Mapping.getMapped(from.result)}
```

7.7.2.3.5.4 COAPin_Mapping

Description

The mapping class creates the output parameter of the ActionUsage for the mapping of UML4SysML::CreateObjectAction.

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:

```
src.owner.oclIsTypeOf(UML::CreateObjectAction)
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{PinFeatureTyping_Mapping.getMapped(from),
COAPinFeatureValue_Mapping.getMapped(from)}
```

7.7.2.3.5.5 COAPinFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

OutputPin

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
COAInvocationExpression_Mapping.getMapped(from.owner)
```

7.7.2.3.5.6 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 def SysMLv1Activity {
        action sysMLv1DestroyObjectAction {
                in target : SysMLv1Block;
                  action : OccurrenceFunctions::destroy {
                        in occ = target;
                }
        }
}
part def SysMLv1Block;
```

General Mappings

CommonAction_Mapping

Mapping Source

DestroyObjectAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Helper.actionOwnedRelationship(from)
->including(DOADestroyFeatureMembership_Mapping.getMapped(from))
```

7.7.2.3.5.7 DOADestroyActionUsage_Mapping

Description

The mapping class creates the action usage for the destroy function.

General Mappings

GenericToActionUsage_Mapping

Mapping Source

DestroyObjectAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{DOADestroyActionUsageFeatureTyping_Mapping.getMapped(from),
DOADestroyActionUsageFeatureMembership_Mapping.getMapped(from)}
```

7.7.2.3.5.8 DOADestroyActionUsageFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

DestroyObjectAction

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  DOADestroyActionUsageReferenceUsage_Mapping.getMapped(from)
  ```

7.7.2.3.5.9 DOADestroyActionUsageFeatureReferenceExpression_Mapping

Description

The mapping class creates the feature reference expression for the UML4SysML::DestroyObjectAction mapping.

General Mappings

GenericToFeatureReferenceExpression_Mapping

Mapping Source

DestroyObjectAction

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{DOADestroyActionUsageMembership_Mapping.getMapped(from),
ReturnParameterFeatureMembership_Factory.create()}
```

7.7.2.3.5.10 DOADestroyActionUsageMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToMembership_Mapping

Mapping Source

DestroyObjectAction

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
from.target
```

7.7.2.3.5.11 DOADestroyActionUsageFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

DestroyObjectAction

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 SysMLv2::Function.allInstances(
)->any(e | e.qualifiedName =  'OccurrenceFunctions::destroy')
```

7.7.2.3.5.12 DOADestroyActionUsageFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

DestroyObjectAction

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
DOADestroyActionUsageFeatureReferenceExpression_Mapping.getMapped(from)
```

7.7.2.3.5.13 DOADestroyActionUsageReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

DestroyObjectAction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.2.3.5.14 DOADestroyFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

DestroyObjectAction

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
DOADestroyActionUsage_Mapping.getMapped(from)
```

7.7.2.3.5.15 ReadIsClassifiedObjectAction_Mapping

Description

The UML4SysML::ReadIsClassifiedObjectAction is conceptually mapped to a SysML v2 ActionUsage.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action def SysMLv1Activity {
        action sysMLv1ReadIsClassifiedObjectActionDirect {
                in object;
                out result : ScalarValues::Boolean =
                        object istype ThisIsABlock;
        }

        action sysMLv1ReadIsClassifiedObjectActionNonDirect {
                in object;
                out result : ScalarValues::Boolean =
                        object hastype ThisIsABlock;
        }
}
```

General Mappings

CommonAction_Mapping

Mapping Source

ReadIsClassifiedObjectAction

Mapping Target

ActionUsage

Owned Mappings

(none)

7.7.2.3.5.16 RICOAFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

ReadIsClassifiedObjectAction

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
RICOAFeatureValueOperatorExpression_Mapping.getMapped(from)
```

7.7.2.3.5.17 RICOAFeatureValueOperatorExpression_Mapping

Description

The mapping class creates the operator expression for the UML4SysML::ReadIsClassifiedObjectAction mapping.

General Mappings

GenericToOperatorExpression_Mapping

Mapping Source

ReadIsClassifiedObjectAction

Mapping Target

OperatorExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- OperatorExpression::operator () : String [1]

  ```
  if from.isDirect then 'istype' else 'hastype' endif
  ```

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

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

7.7.2.3.5.18 RICOAFeatureValueOperatorExpressionFeature_Mapping

Description

The mapping class creates the feature for the operator expression of the UML4SysML::ReadIsClassifiedObjectAction mapping.

General Mappings

GenericToFeature_Mapping

Mapping Source

ReadIsClassifiedObjectAction

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

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

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

7.7.2.3.5.19 RICOAFeatureValueOperatorExpressionFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

ReadIsClassifiedObjectAction

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
RICOAFeatureValueOperatorFeatureReferenceExpression_Mapping.getMappeD(from)
```

7.7.2.3.5.20 RICOAFeatureValueOperatorFeatureReferenceExpression_Mapping

Description

The mapping class creates the feature reference expression for the UML4SysML::ReadIsClassifiedObjectAction mapping.

General Mappings

GenericToFeatureReferenceExpression_Mapping

Mapping Source

ReadIsClassifiedObjectAction

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{RICOAFeatureValueOperatorMembership_Mapping.getMapped(from),
CommonReturnParameterFeatureMembership_Mapping.getMapped(from)}
```

7.7.2.3.5.21 RICOAFeatureValueOperatorMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToMembership_Mapping

Mapping Source

ReadIsClassifiedObjectAction

Mapping Target

Membership

Owned Mappings

(none)

7.7.2.3.5.22 RICOAFeatureValueOperatorParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

ReadIsClassifiedObjectAction

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

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

  ```
  RICOAFeatureValueOperatorExpressionFeature_Mapping.getMapped(from)
  ```

7.7.2.3.5.23 RICOAOutputPin_Mapping

Description

The mapping class creates the output parameter of the ActionUsage element for the UML4SysML::ReadIsClassifiedObjectAction mapping.

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:

```
src.owner.oclIsTypeOf(UML::ReadIsClassifiedObjectAction)
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{PinFeatureTyping_Mapping.getMapped(from),
RICOAFeatureValue_Mapping.getMapped(from.owner),
MultiplicityMembership_Mapping.getMapped(from)}
```

7.7.2.3.5.24 ReadExtentAction_Mapping

Description

A UML4SysML::ReadExtentAction is mapped to a SysML v2 ActionUsage.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action def SysMLv1Activity {
        action sysMLv1ReadExtentAction {
                out thisIsTheOutputPin : SysMLv1Block =
                        all SysMLv1Block;
        }
}
part def SysMLv1Block;
```

General Mappings

CommonAction_Mapping

Mapping Source

ReadExtentAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.2.3.5.25 REAFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

OutputPin

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
REAFeatureValueOperatorExpression_Mapping.getMapped(from)
```

7.7.2.3.5.26 REAFeatureValueOperatorExpression_Mapping

Description

The mapping class creates the operator expression for the UML4SysML::ReadExtentAction mapping.

General Mappings

GenericToOperatorExpression_Mapping

Mapping Source

OutputPin

Mapping Target

OperatorExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- OperatorExpression::operator () : String [1]

```
'all'
```

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

```
 Set{REAFeatureValueOperatorExpressionMembership_Mapping.getMapped(from),
CommonReturnParameterFeatureMembership_Mapping.getMapped(from)}
```

7.7.2.3.5.27 REAFeatureValueOperatorExpressionFeature_Mapping

Description

The mapping class creates the feature for the operator expression for the UML4SysML::ReadExtentAction mapping.

General Mappings

GenericToFeature_Mapping

Mapping Source

OutputPin

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.2.3.5.28 REAFeatureValueOperatorExpressionFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

OutputPin

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
from.owner.classifier
```

7.7.2.3.5.29 REAFeatureValueOperatorExpressionMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

OutputPin

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 REAFeatureValueOperatorExpressionFeature_Mapping.getMapped(from)
```

7.7.2.3.5.30 REAOutputPin_Mapping

Description

The mapping class creates the output parameter of the ActionUsage for the mapping of UML4SysML::ReadExtentAction.

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:

```
src.owner.oclIsTypeOf(UML::ReadExtentAction)
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{PinFeatureTyping_Mapping.getMapped(from),
REAFeatureValue_Mapping.getMapped(from),
MultiplicityMembership_Mapping.getMapped(from)}
```

7.7.2.3.5.31 ReadSelfAction_Mapping

Description

A UML4SysML::ReadSelfAction is mapped to a SysML v2 ActionUsage.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action def SysMLv1Activity {
        action sysMLv1ReadSelfAction {
                out : Base::Anything = this;
        }
}
```

General Mappings

CommonAction_Mapping

Mapping Source

ReadSelfAction

Mapping Target

ActionUsage

Owned Mappings

(none)

7.7.2.3.5.32 RSAFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

OutputPin

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
RSAFeatureValueFeatureReferenceExpression_Mapping.getMapped(from)
```

7.7.2.3.5.33 RSAFeatureValueFeatureReferenceExpression_Mapping

Description

The mapping class creates the feature reference expression for the mapping of UML4SysML::ReadSelfAction.

General Mappings

GenericToFeatureReferenceExpression_Mapping

Mapping Source

OutputPin

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{RSAFeatureValueMembership_Mapping.getMapped(from),
CommonReturnParameterFeatureMembership_Mapping.getMapped(from)}
```

7.7.2.3.5.34 RSAFeatureValueMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToMembership_Mapping

Mapping Source

OutputPin

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 SYSML2::Feature.allInstances()
->any(e | e.qualifiedName = 'Occurrences::Occurrence::this')
```

7.7.2.3.5.35 RSAOutputPin_Mapping

Description

The mapping class creates the output parameter of the ActionUsage for the mapping of UML4SysML::ReadSelfAction.

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:

```
src.owner.oclIsKindOf(UML::ReadSelfAction)
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
true
```

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

```
false
```

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

```
 Set{PinFeatureTyping_Mapping.getMapped(from),
RSAFeatureValue_Mapping.getMapped(from),
MultiplicityMembership_Mapping.getMapped(from)}
```

7.7.2.3.5.36 ReclassifyObjectAction_Mapping

Description

The UML4SysML::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)

7.7.2.3.5.37 TestIdentityAction_Mapping

Description

A UML4SysML::TestIdentityAction is mapped to a SysML v2 ActionUsage.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action def SysMLv1Activity {
        action sysMLv1TestIdentityAction {
                in firstParameter;
                in secondParameter;
                out result : ScalarValues::Boolean =
                        firstParameter == secondParameter;
        }
}
```

General Mappings

CommonAction_Mapping

Mapping Source

TestIdentityAction

Mapping Target

CalculationUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Helper.actionOwnedRelationship(from)
->including(TIAResultExpressionMembership_Mapping.getMapped(from))
```

7.7.2.3.5.38 TIAOperatorExpression_Mapping

Description

The mapping class creates the operator expression for the UML4SysML::TestIdentityAction mapping.

General Mappings

GenericToOperatorExpression_Mapping

Mapping Source

TestIdentityAction

Mapping Target

OperatorExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- OperatorExpression::operator () : String [1]

```
'=='
```

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

```
 Set{EqualOperatorExpressionOperandParameterMembership_Mapping.getMapped(from.first),
EqualOperatorExpressionOperand_Mapping.getMapped(from.second),
CommonReturnParameterFeatureMembership_Mapping.getMapped(from.result)}
```

7.7.2.3.5.39 TIAResultExpressionMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

TestIdentityAction

Mapping Target

ResultExpressionMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
TIAOperatorExpression_Mapping.getMapped(from)
```

7.7.2.3.5.40 ValueSpecificationAction_Mapping

Description

A UML4SysML::ValueSpecificationAction is mapped to a SysML v2 ActionUsage.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action def SysMLv1Acticity {
        action sysMLv1ValueSpecificationAction1 {
                out result : ScalarValues::Integer = 42;
        }

        action sysMLv1ValueSpecificationAction2 {
                out result = sysMLv1OpaqueExpression.result;
                        calc sysMLv1OpaqueExpression {
                            language "Math"
```

```
                                /*
                                 * 42 + 23
                                */
                              }
          }
}
```

General Mappings

CommonAction_Mapping

Mapping Source

ValueSpecificationAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 let toElementFMS: Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Pin)) in
let toElementOMS: Set(UML::Element) =
    (from.ownedElement - toElementFMS) - Set{from.value} in
toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e))
->union(toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e)))
```

7.7.2.3.5.41 VSAOutputPin_Mapping

Description

The mapping class creates the output parameter of the ActionUsage for the mapping of UML4SysML::ValueSpecificationAction.

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:

```
src.owner.oclIsKindOf(UML::ValueSpecificationAction)
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{PinFeatureTyping_Mapping.getMapped(from),
VSAOutputPinFeatureValue_Mapping.getMapped(from),
MultiplicityMembership_Mapping.getMapped(from)}
```

7.7.2.3.5.42 VSAOutputPinFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

OutputPin

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific 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
    from.owner.value
endif
```

7.7.2.3.6 Other Actions

7.7.2.3.6.1 RaiseExceptionAction_Mapping

Description

The UML4SysML::RaiseExceptionAction is mapped to a SysML v2 ActionUsage. The details of the mapping are not defined yet.

General Mappings

CommonAction_Mapping

Mapping Source

RaiseExceptionAction

Mapping Target

ActionUsage

Owned Mappings

(none)

7.7.2.3.6.2 ReduceAction_Mapping

Description

The UML4SysML::ReduceAction is mapped to a SysML v2 ActionUsage. The details of the mapping are not defined yet.

General Mappings

CommonAction_Mapping

Mapping Source

ReduceAction

Mapping Target

ActionUsage

Owned Mappings

(none)

7.7.2.3.7 Structural Feature Actions

7.7.2.3.7.1 AddStructuralFeatureValueAction_Mapping

Description

A UML4SysML::AddStructuralFeatureValueAction is mapped to a SysML v2 ActionUsage defined by the SysML v1 library action definition SysMLv1Library::AddStructuralFeatureValueAction.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action sysMLv1AddStructuralFeatureValueAction :
    SysMLv1Library::AddStructuralFeatureValueAction {
        in insertAt;
        in value;
        in target := object.sysMLv1Property;
        in object : SysMLv1Block;
        in isReplaceAll := true;
}
part def SysMLv1Block {
        attribute sysMLv1Property;
}
```

General Mappings

CommonAction_Mapping

Mapping Source

AddStructuralFeatureValueAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Helper.actionOwnedRelationship(from)
->including(ASFVAFeatureTyping_Mapping.getMapped(from))
->including(ASFVATargetFeatureMembership_Mapping.getMapped(from))
```

7.7.2.3.7.2 ASFVAFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

AddStructuralFeatureValueAction

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.2.3.7.3 ASFVATargetFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

AddStructuralFeatureValueAction

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
ASFVATargetReferenceUsage_Mapping.getMapped(from)
```

7.7.2.3.7.4 ASFVATargetReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

AddStructuralFeatureValueAction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
'target'
```

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

```
 Set{ASFVATargetFeatureValue_Mapping.getMapped(from),
ASFVATargetOwningMembership_Mapping.getMapped(from)}
```

7.7.2.3.7.5 ASFVATargetAsignmentActionUsage_Mapping

Description

The mapping class creates the assignment action for the UML4SysML::AddStructuralFeatureValueAction mapping.

General Mappings

GenericToAssignmentActionUsage_Mapping

Mapping Source

AddStructuralFeatureValueAction

Mapping Target

AssignmentActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.2.3.7.6 ASFVATargetActionParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

AddStructuralFeatureValueAction

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
ASFVATargetActionReferenceUsage_Mapping.getMapped(from)
```

7.7.2.3.7.7 ASFVATargetActionReferenceUsage_Mapping

Description

The mapping class creates the reference usage element for the target element of the UML4SysML::AddStructuralFeatureValueAction mapping.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

AddStructuralFeatureValueAction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

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

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

7.7.2.3.7.8 ASFVATargetActionReferenceUsageReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

AddStructuralFeatureValueAction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.2.3.7.9 ASFVATargetActionReferenceUsageFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

AddStructuralFeatureValueAction

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
ASFVATargetActionReferenceUsageFeature_Mapping.getMapped(from)
```

7.7.2.3.7.10 ASFVATargetActionFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

AddStructuralFeatureValueAction

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
ASFVATargetActionReferenceUsageReferenceUsage_Mapping.getMapped(from)
```

7.7.2.3.7.11 ASFVATargetActionReferenceUsageFeature_Mapping

Description

The mapping class creates the reference usage element for the target action of the UML4SysML::AddStructuralFeatureValueAction mapping.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

AddStructuralFeatureValueAction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

7.7.2.3.7.12 ASFVATargetFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

AddStructuralFeatureValueAction

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  true
  ```

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

  ```
  ASFVATargetFeatureChainExpression_Mapping.getMapped(from)
  ```

7.7.2.3.7.13 ASFVATargetFeatureChainExpression_Mapping

Description

The mapping class creates the feature chain expression element for the target element of the UML4SysML::AddStructuralFeatureValueAction mapping.

General Mappings

GenericToFeatureChainExpression_Mapping

Mapping Source

AddStructuralFeatureValueAction

Mapping Target

FeatureChainExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{ASFVATargetParameterMembership_Mapping.getMapped(from),
ASFVATargetFeatureValueExpressionMembership_Mapping.getMapped(from),
ReturnParameterFeatureMembership_Factory.create()}
```

7.7.2.3.7.14 ASFVATargetParameterFeature_Mapping

Description

The mapping class creates the feature element for the target element of the UML4SysML::AddStructuralFeatureValueAction mapping.

General Mappings

GenericToFeature_Mapping

Mapping Source

AddStructuralFeatureValueAction

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{ASFVATargetParameterFeatureValue_Mapping.getMapped(from),
ASFVATargetParameterExpressionFeatureMembership_Mapping.getMapped(from)}
```

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

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

7.7.2.3.7.15 ASFVATargetParameterExpressionFeature_Mapping

Description

The mapping class creates the feature element of the feature reference expression for the target element of the UML4SysML::AddStructuralFeatureValueAction mapping.

General Mappings

GenericToFeature_Mapping

Mapping Source

AddStructuralFeatureValueAction

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.2.3.7.16 ASFVATargetParameterExpressionFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

AddStructuralFeatureValueAction

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
ASFVATargetParameterExpressionFeature_Mapping.getMapped(from)
```

7.7.2.3.7.17 ASFVATargetParameterFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

AddStructuralFeatureValueAction

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 ASFVATargetParameterFeatureReferenceExpression_Mapping.getMapped(from)
```

7.7.2.3.7.18 ASFVATargetParameterFeatureReferenceExpression_Mapping

Description

The mapping class creates the feature reference expression element for the target element of the UML4SysML::AddStructuralFeatureValueAction mapping.

General Mappings

GenericToFeatureReferenceExpression_Mapping

Mapping Source

AddStructuralFeatureValueAction

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{ASFVATargetParameterExpressionMembership_Mapping.getMapped(from),
ReturnParameterFeatureMembership_Factory.create()}
```

7.7.2.3.7.19 ASFVATargetParameterExpressionMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToMembership_Mapping

Mapping Source

AddStructuralFeatureValueAction

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
from.object
```

7.7.2.3.7.20 ASFVATargetParameterFeatureExpressionMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToMembership_Mapping

Mapping Source

AddStructuralFeatureValueAction

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
from.structuralFeature
```

7.7.2.3.7.21 ASFVATargetParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

AddStructuralFeatureValueAction

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

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

  ```
  ASFVATargetParameterFeature_Mapping.getMapped(from)
  ```

7.7.2.3.7.22 ASFVATargetOwningMembership_Mapping

Description

Creates a owning membership relationship for *ownedMemberElement()*.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

AddStructuralFeatureValueAction

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
ASFVATargetAsignmentActionUsage_Mapping.getMapped(from)
```

7.7.2.3.7.23 ClearStructuralFeatureAction_Mapping

Description

The UML4SysML::ClearStructuralFeatureAction is mapped to a SysML v2 ActionUsage. The details of the mapping are not defined yet.

General Mappings

CommonAction_Mapping

Mapping Source

ClearStructuralFeatureAction

Mapping Target

ActionUsage

Owned Mappings

(none)

7.7.2.3.7.24 RSFAReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

ReadStructuralFeatureAction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

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

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

7.7.2.3.7.25 RSFAReferenceUsageFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ReadStructuralFeatureAction

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
ReadStructuralFeatureActionReferenceUsage_Mapping.getMapped(from)
```

7.7.2.3.7.26 RSFAReferenceUsageFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

ReadStructuralFeatureAction

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
RSFAReferenceUsageFeatureChainExpression_Mapping.getMapped(from)
```

7.7.2.3.7.27 RSFAReferenceUsageFeatureChainExpression_Mapping

Description

The mapping class creates the feature chain expression element for the reference usage of the UML4SysML::ReadStructuralFeatureValueAction mapping.

General Mappings

GenericToFeatureChainExpression_Mapping

Mapping Source

ReadStructuralFeatureAction

Mapping Target

FeatureChainExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{RSFAReferenceUsageParameterMembership_Mapping.getMapped(from),
RSFAReferenceUsageMembership_Mapping.getMapped(from),
ReturnParameterFeatureMembership_Factory.create()}
```

7.7.2.3.7.28 RSFAReferenceUsageExpressionFeature_Mapping

Description

The mapping class creates the feature of the feature chain expression for the reference usage of the UML4SysML::ReadStructuralFeatureValueAction mapping.

General Mappings

GenericToFeature_Mapping

Mapping Source

ReadStructuralFeatureAction

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{RSFAReferenceUsageExpressionFeatureValue_Mapping.getMapped(from),
RSFAReferenceUsageExpressionFeatureMembership_Mapping.getMapped(from)}
```

7.7.2.3.7.29 RSFAReferenceUsageFeatureChainExpressionFeature_Mapping

Description

The mapping class creates the feature element for the feature chain expression for the UML4SysML::RemoveStructuralFeatureValueAction mapping.

General Mappings

GenericToFeature_Mapping

Mapping Source

ReadStructuralFeatureAction

Mapping Target

Feature

Owned Mappings

(none)

7.7.2.3.7.30 RSFAReferenceUsageExpressionFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ReadStructuralFeatureAction

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
RSFAReferenceUsageFeatureChainExpressionFeature_Mapping.getMapped(from)
```

7.7.2.3.7.31 RSFAReferenceUsageExpressionFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

ReadStructuralFeatureAction

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
RSFAReferenceUsageExpressionFeatureReferenceExpression_Mapping.getMapped(from)
```

7.7.2.3.7.32 RSFAReferenceUsageExpressionFeatureReferenceExpression_Mapping

Description

The mapping class creates the feature reference expression element for the UML4SysML::RemoveStructuralFeatureValueAction mapping.

General Mappings

GenericToFeatureReferenceExpression_Mapping

Mapping Source

ReadStructuralFeatureAction

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{RSFAReferenceUsageExpressionFeatureMembership_Mapping.getMapped(from),
ReturnParameterFeatureMembership_Factory.create()}
```

7.7.2.3.7.33 RSFAReferenceUsageMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToMembership_Mapping

Mapping Source

ReadStructuralFeatureAction

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
from.object
```

7.7.2.3.7.34 RSFAReferenceUsageFeatureChainExpressionMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToMembership_Mapping

Mapping Source

ReadStructuralFeatureAction

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
from.structuralFeature
```

7.7.2.3.7.35 RSFAReferenceUsageParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

ReadStructuralFeatureAction

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  RSFAReferenceUsageExpressionFeature_Mapping.getMapped(from)
  ```

7.7.2.3.7.36 ReadStructuralFeatureAction_Mapping

Description

A UML4SysML::ReadStructuralFeatureAction is mapped to a SysML v2 ActionUsage that returns the value of the specified structural feature of the given object.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action def SysMLv1Activity {
        action sysMLv1ReadStructuralFeatureAction {
                in object : SysMLv1Block;
                out result = object.sysMLv1Property;
        }
}
part def SysMLv1Block {
        attribute sysMLv1Property;
}
```

General Mappings

CommonAction_Mapping

Mapping Source

ReadStructuralFeatureAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
         Helper.actionOwnedRelationship(from)
        ->including(RSFAReferenceUsageFeatureMembership_Mapping.getMapped(from))
```

7.7.2.3.7.37 RemoveStructuralFeatureValueAction_Mapping

Description

The UML4SysML::RemoveStructuralFeatureValueAction is mapped to a SysML v2 ActionUsage. The details of the mapping are not defined yet.

General Mappings

CommonAction_Mapping

Mapping Source

RemoveStructuralFeatureValueAction

Mapping Target

ActionUsage

Owned Mappings

(none)

7.7.2.3.8 Structured Actions

7.7.2.3.8.1 LoopNode_Mapping

Description

The UML4SysML::LoopNode is mapped to a SysML v2 ActionUsage. The details of the mapping are not defined yet.

General Mappings

StructuredActivityNode_Mapping

Mapping Source

LoopNode

Mapping Target

ActionUsage

Owned Mappings

(none)

7.7.2.3.8.2 SequenceNode_Mapping

Description

The UML4SysML::SequenceNode is mapped to a SysML v2 ActionUsage. The details of the mapping are not defined yet.

General Mappings

CommonAction_Mapping
StructuredActivityNode_Mapping

Mapping Source

SequenceNode

Mapping Target

ActionUsage

Owned Mappings

(none)

7.7.2.3.8.3 StructuredActivityNode_Mapping

Description

The UML4SysML::StructuredActivityNode is mapped to a SysML v2 ActionUsage. The details of the mapping are not defined yet.

General Mappings

CommonAction_Mapping

Mapping Source

StructuredActivityNode

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 let initialNodes : Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::InitialNode)) in
let finalNodes : Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::FinalNode)) in
let objectFlowsWithGuard : Set(UML::ObjectFlow) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::ObjectFlow)
        and not e.oclAsType(UML::ObjectFlow).guard.oclIsUndefined()) in
let objectFlows : Set(UML::ObjectFlow) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::ObjectFlow))  in
let ignoreInterruptibleActivityRegion: Set(UML::InterruptibleActivityRegion) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::InterruptibleActivityRegion)) in
let elementsFMS : Set(UML::Element) =
    ((from.ownedElement->select(e | e.oclIsKindOf(UML::ControlNode) or
        e.oclIsKindOf(UML::Action) or (e.oclIsKindOf(UML::ControlFlow) or
        e.oclIsKindOf(UML::Pin))) - initialNodes) - finalNodes) in
let elementsOMS: Set(UML::Element) =
    ((((((from.ownedElement-initialNodes)-finalNodes)-objectFlowsWithGuard)
        -objectFlows)-elementsFMS)-ignoreInterruptibleActivityRegion) in
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 | FlowFinalNodeMembership_Mapping.getMapped(e)))
->union(objectFlowsWithGuard
    ->collect(e | ObjectFlowGuardFeatureMembership_Mapping.getMapped(e)))
->union(objectFlows->collect(e | ObjectFlowFeatureMembership_Mapping.getMapped(e)))
```

7.7.2.3.9 Variable Actions

7.7.2.3.9.1 AddVariableValueAction_Mapping

Description

A UML4SysML::AddVariableValueAction is mapped to a SysML v2 ActionUsage defined by the SysML v1 library action definition SysMLv1Library::AddValueAction. 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 sysMLv1Variable1 : ScalarValues::Integer;
        private attribute sysMLv1Variable2 [0..*] : ScalarValues::Integer;

        action sysMLv1AddVariableValueAction1 : SysMLv1Library::AddValueAction {
                :>> target := sysMLv1Variable1;
        }

        action sysMLv1AddVariableValueAction1 : SysMLv1Library::AddValueAction {
                :>> target := thisIsAVariable;
                :>> isReplaceAll := true;
        }
}
```

General Mappings

CommonAction_Mapping

Mapping Source

AddVariableValueAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Helper.actionOwnedRelationship(from)
->including(AVVAFeatureTyping_Mapping.getMapped(from))
->including(AVVAVariableFeatureMembership_Mapping.getMapped(from))
```

7.7.2.3.9.2 AVVAFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Action

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.2.3.9.3 AVVAVariable_Mapping

Description

The mapping class creates a reference usage element for the UML4SysML::AddVariableValueAction mapping.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Action

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{AVVARedefinition_Mapping.getMapped(from),
AVVAFeatureValue_Mapping.getMapped(from),
CommonAssignmentActionUsageOwningMembership_Mapping.getMapped(from)}
```

7.7.2.3.9.4 AVVAVariableFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Action

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 AVVAVariable_Mapping.getMapped(from)
```

7.7.2.3.9.5 AVVARedefinition_Mapping

Description

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

General Mappings

GenericToRedefinition_Mapping

Mapping Source

Action

Mapping Target

Redefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 SYSML2::ReferenceUsage.allInstances()
->any(m | m.qualifiedName = 'SysMLv1Library::AddValueAction::target')
```

7.7.2.3.9.6 AVVAFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Action

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  AddValueActionValueFeatureReferenceExpression_Mapping.getMapped(from)
  ```

7.7.2.3.9.7 AVVAValueFeatureReferenceExpression_Mapping

Description

The mapping class creates the feature reference expression element for the UML4SysML::AddStructuralFeatureValueAction mapping.

General Mappings

GenericToFeatureReferenceExpression_Mapping

Mapping Source

Action

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{AVVAValueExpressionMembership_Mapping.getMapped(from),
ReturnParameterFeatureMembership_Factory.create()}
```

7.7.2.3.9.8 AVVAValueExpressionMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToMembership_Mapping

Mapping Source

Action

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 from
```

7.7.2.3.9.9 ClearVariableAction_Mapping

Description

The UML4SysML::ClearVariableAction is mapped to a SysML v2 ActionUsage that sets the attribute usage representing the variable to null.

The expected SysML v2 textual notation of a SysMLv1::ClearVariableAction is as follows

```
action def SysMLv1Activity {
        private attribute sysMLv1Variable : ScalarValues::Integer;
        action sysMLv1ClearVariableAction {
                sysMLv1Variable := null;
        }
}
```

General Mappings

CommonAction_Mapping

Mapping Source

ClearVariableAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Helper.actionOwnedRelationship(from)
->including(CVAFeatureMembership_Mapping.getMapped(from))
```

7.7.2.3.9.10 CVAFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ClearVariableAction

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
CVAReferenceUsage_Mapping.getMapped(from)
```

7.7.2.3.9.11 CVAReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

ClearVariableAction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{CVAReferenceUsageFeatureValue_Mapping.getMapped(from),
CommonAssignmentActionUsageOwningMembership_Mapping.getMapped(from)}
```

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

```
from.variable.name
```

7.7.2.3.9.12 CVAReferenceUsageFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

ClearVariableAction

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
LiteralNull_Factory.create()
```

7.7.2.3.9.13 ReadVariableAction_Mapping

Description

A UML4SysML::ReadVariableValueAction is mapped to a SysML v2 ActionUsage with an out parameter that returns the value of the attribute usage that is the transformation target of the UML4SysML::Variable.

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;

        action sysMLv1ReadVariableAction {
                out result : ScalarValues::Integer = sysMLv1Variable;
        }
}
```

General Mappings

CommonAction_Mapping

Mapping Source

ReadVariableAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.2.3.9.14 RVAFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ReadVariableAction

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 RVAReferenceUsage_Mapping.getMapped(from.result)
```

7.7.2.3.9.15 RVAReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Pin

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 let featureTyping : Set(KerML::FeatureTyping) =
    if from.type.oclIsUndefined() then
        Set{}
    else
        Set{RVAReferenceUsageFeatureTyping_Mapping.getMapped(from)}
    endif in
featureTyping
->including(RVAReferenceUsageFeatureValue_Mapping.getMapped(from))
```

7.7.2.3.9.16 RVAReferenceUsageFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

TypedElementFeatureTyping_Mapping

Mapping Source

Pin

Mapping Target

FeatureTyping

Owned Mappings

(none)

7.7.2.3.9.17 RVAReferenceUsageFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Pin

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
RVAReferenceUsageFeatureReferenceExpression_Mapping.getMapped(from)
```

7.7.2.3.9.18 RVAReferenceUsageFeatureReferenceExpression_Mapping

Description

The mapping class creates the feature reference expression element for the UML4SysML::ReadVariableAction mapping.

General Mappings

GenericToFeatureReferenceExpression_Mapping

Mapping Source

Pin

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
Set{RVAReferenceUsageExpressionMembership_Mapping.getMapped(from),
ReturnParameterFeatureMembership_Factory.create()}
```

7.7.2.3.9.19 RVAReferenceUsageExpressionMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToMembership_Mapping

Mapping Source

Pin

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 from.owner.oclAsType(UML::ReadVariableAction).variable
```

7.7.2.3.9.20 RemoveVariableValueAction_Mapping

Description

A UML4SysML::RemoveVariableValueAction is mapped to a SysML v2 ActionUsage defined by the SysML v1 library action definition SysMLv1Library::RemoveVariableValueAction.

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 sysMLv1Variable : ScalarValues::Integer;

        action sysMLv1RemoveVariableValueAction
            : SysMLv1Library::RemoveVariableValueAction {
                :>> variable := sysMLv1Variable;
        }
}
```

General Mappings

CommonAction_Mapping

Mapping Source

RemoveVariableValueAction

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Helper.actionOwnedRelationship(from)
->including(RVVAFeatureTyping_Mapping.getMapped(from))
->including(RVVAVariableFeatureMembership_Mapping.getMapped(from))
```

7.7.2.3.9.21 RVVAVariableFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

RemoveVariableValueAction

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
RVVAVariable_Mapping.getMapped(from)
```

7.7.2.3.9.22 RVVAVariableFeatureReferenceExpression_Mapping

Description

The mapping class creates the feature reference expression element for the UML4SysML::RemoveVariableValueAction mapping.

General Mappings

GenericToFeatureReferenceExpression_Mapping

Mapping Source

RemoveVariableValueAction

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{RVVAVariableExpressionMembership_Mapping.getMapped(from),
EmptyReturnParameterFeatureMembership_Mapping.getMapped(from)}
```

7.7.2.3.9.23 RVVAVariableExpressionMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToMembership_Mapping

Mapping Source

RemoveVariableValueAction

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
from.variable
```

7.7.2.3.9.24 RVVAVariableFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

RemoveVariableValueAction

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
RVVAVariableFeatureReferenceExpression_Mapping.getMapped(from)
```

7.7.2.3.9.25 RVVAVariable_Mapping

Description

The mapping class creates a reference usage element for the UML4SysML::RemoveVariableValueAction mapping.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

RemoveVariableValueAction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{RVVAVariableRedefinition_Mapping.getMapped(from),
RVVAVariableFeatureValue_Mapping.getMapped(from),
CommonAssignmentActionUsageOwningMembership_Mapping.getMapped(from)}
```

7.7.2.3.9.26 RVVAVariableRedefinition_Mapping

Description

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

General Mappings

GenericToRedefinition_Mapping

Mapping Source

RemoveVariableValueAction

Mapping Target

Redefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 SYSML2::ReferenceUsage.allInstances()
->any(m | m.qualifiedName = 'SysMLv1Library::RemoveVariableValueAction::variable')
```

7.7.2.3.9.27 RVVAFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

RemoveVariableValueAction

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.3 Activities

This chapter lists all mapping specifications of UML4SysML::Activities model elements.

7.7.3.1 Overview

The following table gives an overview of which SysML v2 elements the UML4SysML::Activities elements are transformed with which mapping class. The mapping details are in 7.7.3.3.

The justifications for the elements without mapping are given in 7.7.3.2.

Table 3. List of all mappings

SysML v1 Concept	SysML v2 Concept
Activity	VerificationCaseDefinition ActionUsage ActionDefinition
ActivityFinalNode	
ActivityParameterNode	
ActivityPartition	
CentralBufferNode	ActionUsage ActionUsage
ControlFlow	TransitionUsage SuccessionAsUsage
DataStoreNode	ActionUsage
DecisionNode	DecisionNode

SysML v1 Concept	SysML v2 Concept
ExceptionHandler	
FlowFinalNode	
ForkNode	ForkNode
InitialNode	
InterruptibleActivityRegion	
JoinNode	JoinNode
MergeNode	MergeNode
ObjectFlow	TransitionUsage SuccessionFlowConnectionUsage
Variable	AttributeUsage ItemUsage

7.7.3.2 UML4SysML::Activities elements not mapped

Table 4. List of SysML v1 elements not mapped of this section

SysML v1 Concept	Rationale
ActivityFinalNode	Mapping is not specified yet.
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.
ActivityPartition	Mapping is not specified yet.
ExceptionHandler	Mapping is not specified yet.
InterruptibleActivityRegion	Mapping is not specified yet.

7.7.3.3 Mapping Specifications

7.7.3.3.1 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:

```
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.

7.7.3.3.2 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 src.owner.oclIsKindOf(UML::Package))
and (not Helper.hasStereotypeApplied(src, 'SysML::Requirements::TestCase'))
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ActionUsage::declaredName () : String [0..1]

```
 if from.name.size() > 1 then
    from.name.substring(1,1).toLowerCase().
        concat(from.name.substring(2, from.name.size()))
else
    from.name
endif
```

7.7.3.3.3 ActivityEdgeInitialNodeFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToEndFeatureMembership_Mapping

Mapping Source

InitialNode

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 ActivityEdgeSourceInitialNode_Mapping.getMapped(from)
```

7.7.3.3.4 ActivityEdgeMetadata_Mapping

Description

Adds metadata to the transformation target elements of UML4SysML::ControlFlow and UML::ObjectFlow to map the UML4SysML::ActivityEdge::weight property which has no direct target in SysML v2.

General Mappings

GenericToMetadataUsage_Mapping

Mapping Source

ActivityEdge

Mapping Target

MetadataUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- MetadataUsage::declaredName () : String [0..1]

```
 'weight'
```

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

```
 Set{ActivityEdgeMetadataFeatureTyping_Mapping.getMapped(from),
ActivityEdgeMetadataFeatureMembership_Mapping.getMapped(from)}
```

7.7.3.3.5 ActivityEdgeMetadataFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ActivityEdge

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
ActivityEdgeMetadataReferenceUsage_Mapping.getMapped(from)
```

7.7.3.3.6 ActivityEdgeMetadataFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

ActivityEdge

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.3.3.7 ActivityEdgeMetadataFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

ActivityEdge

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
from.weight
```

7.7.3.3.8 ActivityEdgeMetadataOwningMembership_Mapping

Description

Creates a owning membership relationship for *ownedMemberElement()*.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

ActivityEdge

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 ActivityEdgeMetadata_Mapping.getMapped(from)
```

7.7.3.3.9 ActivityEdgeMetadataRedefinition_Mapping

Description

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

General Mappings

GenericToRedefinition_Mapping

Mapping Source

ActivityEdge

Mapping Target

Redefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.3.3.10 ActivityEdgeMetadataReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

ActivityEdge

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{ActivityEdgeMetadataRedefinition_Mapping.getMapped(from),
ActivityEdgeMetadataFeatureValue_Mapping.getMapped(from)}
```

7.7.3.3.11 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

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  true
  ```

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

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

7.7.3.3.12 ActivityEdgeSourceInitialNode_Mapping

Description

The UML4SysML::InitialNode is mapped to a subsetted feature of the SysML v2 library element Actions::start.

General Mappings

GenericToFeature_Mapping

Mapping Source

InitialNode

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  true
  ```

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

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

7.7.3.3.13 ActivityEdgeSourceEndFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToEndFeatureMembership_Mapping

Mapping Source

ActivityNode

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  ActivityEdgeSourceEndFeature_Mapping.getMapped(from)
  ```

7.7.3.3.14 ActivityEdgeSourceInitialNodeSubsetting_Mapping

Description

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

General Mappings

GenericToSubsetting_Mapping

Mapping Source

InitialNode

Mapping Target

Subsetting

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 SYSML2::ActionUsage.allInstances()
->any(m | m.qualifiedName = 'Actions::Action::start')
```

7.7.3.3.15 ActivityEdgeSourceEndSubsetting_Mapping

Description

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

General Mappings

GenericToSubsetting_Mapping

Mapping Source

ActivityNode

Mapping Target

Subsetting

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 from
```

7.7.3.3.16 ActivityEdgeTransitionUsageSourceMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToMembership_Mapping

Mapping Source

ActivityNode

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 if from.oclIsTypeOf(UML::ActivityParameterNode) then
    from.parameter
else
    from
endif
```

7.7.3.3.17 CentralBufferNode_Mapping

Description

The mapping of the UML4SysML::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)

7.7.3.3.18 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

In addition to the inherited rules, the following lists the mapping class specific 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::Parameter)) in
relationships->union(parameters
    ->collect(p | ParameterMembership_Mapping.getMapped(p))
)
```

7.7.3.3.19 CommonActivityEdgeSuccessionAsUsage_Mapping

Description

The mapping class provides a common mapping of a UML4SysML::ActivityEdge to a SysMLv2 SucessionAsUsage. The mapping is used for UML4SysML::ControlFlows and UML4SysML::ObjectFlows.

General Mappings

GenericToConnector_Mapping

Mapping Source

ActivityEdge

Mapping Target

SuccessionAsUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 let relationships : Set(KerML::Relationship) = Set{
if from.source.oclIsKindOf(UML::InitialNode) then
    ActivityEdgeInitialNodeFeatureMembership_Mapping.getMapped(from.source)
else
    ActivityEdgeSourceEndFeatureMembership_Mapping.getMapped(from.source)
endif,
if from.oclIsKindOf(UML::ObjectFlow) then
    ObjectFlowGuardSuccessionTargetEndFeatureMembership_Mapping.getMapped(from)
else if from.target.oclIsKindOf(UML::FinalNode) then
        ControlFlowFinalNodeFeatureMembership_Mapping.getMapped(from.target)
    else
        ControlFlowTargetFeatureMembership_Mapping.getMapped(from.target)
    endif
endif} in
if from.guard.oclIsUndefined() then
    relationships
else
    relationships
    ->including(ElementFeatureMembership_Mapping.getMapped(from.guard))
endif
```

7.7.3.3.20 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

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- 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::isDerived () : Boolean [1]

```
false
```

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

```
false
```

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

```
false
```

7.7.3.3.21 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.

```
action def SysMLv1Activity {
        action sysMLv1Action1;
        succession sysMLv1ControlFlow first sysMLv1Action1
                if guardCondition.result then sysMLv1Action2 {
                  calc guardCondition {
                    return : ScalarValues::Boolean;
                    language "English"
                    /*
                     * thisIsAGuard
                     */
                  }
                }
        action sysMLv1Action2;
}
```

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 src.guard.oclIsUndefined()
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
   true
  ```

- 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::OpaqueExpression) then
      relationships
      ->including(ElementFeatureMembership_Mapping.getMapped(from.guard))
  else
      relationships
  endif in
  let relationshipsConsideringWeight : Set(KerML::Relationship) =
  if from.weight.oclIsUndefined() then
      relationshipsWithGuard
  else
      relationshipsWithGuard
  ```

```
            ->including(ActivityEdgeMetadataOwningMembership_Mapping.getMapped(from))
        endif in
        if Helper.hasStereotypeApplied(from, 'SysML::Activities::Probability') then
            relationshipsConsideringWeight
            ->including(ProbabilityOwningMembership_Mapping.getMapped(from))
        else
            relationshipsConsideringWeight
        endif
```

7.7.3.3.22 ControlFlowFinalNodeFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToEndFeatureMembership_Mapping

Mapping Source

ActivityNode

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
ControlFlowTargetFinalNode_Mapping.getMapped(from)
```

7.7.3.3.23 ControlFlowTargetFinalNodeSubsetting_Mapping

Description

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

General Mappings

GenericToSubsetting_Mapping

Mapping Source

FinalNode

Mapping Target

Subsetting

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 SYSML2::ActionUsage.allInstances()
->any(m | m.qualifiedName = 'Actions::Action::done')
```

7.7.3.3.24 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.

```
action def SysMLv1Activity {
        action sysMLv1Action1;
        succession sysMLv1ControlFlow
                first sysMLv1Action1 then sysMLv1Action2;
        action sysMLv1Action2;
}
```

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:

```
src.guard.oclIsUndefined()
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 let relationships : Set(KerML::Relationship) = Set{
if from.source.oclIsKindOf(UML::InitialNode) then
    ActivityEdgeInitialNodeSourceEndFeatureMembership_Mapping.getMapped(from.source)
else
    ActivityEdgeSourceEndFeatureMembership_Mapping.getMapped(from.source)
endif,
if from.oclIsKindOf(UML::ObjectFlow) then
    ObjectFlowGuardSuccessionTargetEndFeatureMembership_Mapping.getMapped(from)
else if from.target.oclIsKindOf(UML::FinalNode) then
        ControlFlowFinalNodeTargetEndFeatureMembership_Mapping.getMapped(from.target)
     else
        ControlFlowTargetEndFeatureMembership_Mapping.getMapped(from.target)
     endif
endif} in
let relationshipsWithGuard : Set(KerML::Relationship) =
if from.guard.oclIsUndefined() then
    relationships
else
    relationships
    ->including(ElementFeatureMembership_Mapping.getMapped(from.guard))
endif in
let relationshipsConsideringWeight : Set(KerML::Relationship) =
if from.weight.oclIsUndefined() then
    relationshipsWithGuard
else
    relationshipsWithGuard
    ->including(ActivityEdgeMetadataOwningMembership_Mapping.getMapped(from))
endif in

if Helper.hasStereotypeApplied(from, 'SysML::Activities::Probability') then
    relationshipsConsideringWeight
    ->including(ProbabilityOwningMembership_Mapping.getMapped(from))
else
    relationshipsConsideringWeight
endif
```

7.7.3.3.25 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

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  true
  ```

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

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

7.7.3.3.26 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

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
true
```

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

```
Set{controlFlowTargetEndSubsetting.to}
```

7.7.3.3.27 ControlFlowTargetFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToEndFeatureMembership_Mapping

Mapping Source

ActivityNode

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
ControlFlowTargetEndFeature_Mapping.getMapped(from)
```

7.7.3.3.28 ControlFlowTargetEndSubsetting_Mapping

Description

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

General Mappings

GenericToSubsetting_Mapping

Mapping Source

ActivityNode

Mapping Target

Subsetting

Owned Mappings

- controlFlowTargetEndFeature : ControlFlowTargetEndFeature_Mapping

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  controlFlowTargetEndFeature.to
  ```

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

  ```
  from
  ```

7.7.3.3.29 ControlFlowTransitionUsageFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ControlFlow

Mapping Target

TransitionFeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 if from.guard.oclIsKindOf(UML::OpaqueExpression) then
    OpaqueExpressionAsValue_Mapping.getMapped(from.guard)
else
    from.guard
endif
```

- TransitionFeatureMembership::kind () : TransitionFeatureKind [1]

```
 KerML::TransitionFeatureKind::guard
```

7.7.3.3.30 DataStoreNode_Mapping

Description

The mapping of the UML4SysML::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)

7.7.3.3.31 DecisionNode_Mapping

Description

The UML4SysML::DecisionNode is mapped to a SysMLv2 DecisionNode.

There is no suitable element in SysML v2 for the else condition of an outgoing UML4SysML::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.

```
action def SysMLv1Activity {
        action sysMLv1Action1;
        succession sysMLv1ControlFlow1 first sysMLv1Action1 then sysMLv1DecisionNode;
        decide sysMLv1DecisionNode;
        succession sysMLv1ControlFlow2 first sysMLv1DecisionNode if {
                return : ScalarValues::Boolean;
                // guard expression, for example, opaque expression
        }.result then sysMLv1Action2;
        succession flow2 first sysMLv1DecisionNode if {
                return : ScalarValues::Boolean;
                language "SysMLv1"
                /*
                 * else
                   */
        }.result then sysMLv1Action2;
        action sysMLv1Action2;
}
```

General Mappings

GenericToUsage_Mapping
NamedElementMain_Mapping

Mapping Source

DecisionNode

Mapping Target

DecisionNode

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
true
```

7.7.3.3.32 FlowFinalNodeMembership_Mapping

Description

The mapping class creates a membership relationship to the action usage library element Actions::Action::done.

General Mappings

GenericToMembership_Mapping

Mapping Source

FlowFinalNode

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 SysMLv2::ActionUsage.allInstances()
->any(e | e.qualifiedName =  'Actions::Action::done')
```

7.7.3.3.33 ForkNode_Mapping

Description

The UML4SysML::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 sysMLv1Action1;

          then fork sysMLv1ForkNode;

          then sysMLv1Action2;
          then sysMLv1Action3;
          action sysMLv1Action2;
          then sysMLv1JoinNode;
          action sysMLv1Action3;
          then sysMLv1JoinNode;

          join sysMLv1JoinNode;

          then done;
}
```

General Mappings

GenericToUsage_Mapping
NamedElementMain_Mapping

Mapping Source

ForkNode

Mapping Target

ForkNode

Owned Mappings

(none)

7.7.3.3.34 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

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 SysMLv2::ActionUsage.allInstances()
->any(e | e.qualifiedName =  'Actions::Action::start')
```

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

```
 if from.name = '' then null else from.name endif
```

7.7.3.3.35 JoinNode_Mapping

Description

The UML4SysML::JoinNode is mapped to a SysMLv2JoinNode.

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 sysMLv1Action1;

          then fork sysMLv1ForkNode;

          then sysMLv1Action2;
          then sysMLv1Action3;
          action sysMLv1Action2;
          then sysMLv1JoinNode;
          action sysMLv1Action3;
          then sysMLv1JoinNode;

          join sysMLv1JoinNode;

          then done;
}
```

General Mappings

GenericToUsage_Mapping
NamedElementMain_Mapping

Mapping Source

JoinNode

Mapping Target

JoinNode

Owned Mappings

(none)

7.7.3.3.36 MergeNode_Mapping

Description

The UML4SysML::MergeNode is mapped to a SysMLv2 MergeNode.

General Mappings

GenericToUsage_Mapping
NamedElementMain_Mapping

Mapping Source

MergeNode

Mapping Target

MergeNode

Owned Mappings

(none)

7.7.3.3.37 ObjectFlow_Mapping

Description

A UML4SysML::ObjectFlowFlow without a guard condition is mapped to a SysMLv2SuccessionFlowConnectionUsage.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action def SysMLv1Acticity {
        action sysMLv1Action1 {
                out outputValue;
        }
        succession flow sysMLv1ObjectFlow of ScalarValues::String
                from sysMLv1Action1.outputValue to sysMLv1Action1.inputValue;
        action sysMLv1Action2 {
                out inputValue;
        }
}
```

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:

```
src.guard.oclIsUndefined()
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 let relationships : Set(KerML::Relationship) =
if from.source.oclIsKindOf(UML::ObjectNode) then
    Set{ObjectFlowItemFeatureMembership_Mapping.getMapped(from),
    ObjectFlowEndFeatureMembership_Mapping.getMapped(from.source),
    ObjectFlowEndFeatureMembership_Mapping.getMapped(from.target)}
else
    Set{ObjectFlowEndFeatureMembership_Mapping.getMapped(from.source),
    ObjectFlowEndFeatureMembership_Mapping.getMapped(from.target)}
endif in
let relationshipsConsideringWeight : Set(KerML::Relationship) =
if from.weight.oclIsUndefined() then
    relationships
else
    relationships
    ->including(ActivityEdgeMetadataOwningMembership_Mapping.getMapped(from))
endif in
let relationshipsConsideringRate : Set(KerML::Relationship) =
if (Helper.hasStereotypeApplied(from, 'SysML::Activities::Rate') or
    Helper.hasStereotypeApplied(from, 'SysML::Activities::Discrete') or
    Helper.hasStereotypeApplied(from, 'SysML::Activities::Continuous')) then

    relationshipsConsideringWeight
    ->including(RateOwningMembership_Mapping.getMapped(from))
else
    relationshipsConsideringWeight
endif in

if Helper.hasStereotypeApplied(from, 'SysML::Activities::Probability') then
    relationshipsConsideringRate
    ->including(ProbabilityOwningMembership_Mapping.getMapped(from))
else
    relationshipsConsideringRate
endif
```

7.7.3.3.38 ObjectFlowFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ObjectFlow

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  ObjectFlow_Mapping.getMapped(from)
  ```

7.7.3.3.39 ObjectFlowGuardFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ObjectFlow

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  ObjectFlowGuard_Mapping.getMapped(from)
  ```

7.7.3.3.40 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.

```
action def SysMLv1Activity {
        action sysMLv1Action1 {
                out outputValue;
        }

        first sysMLv1Action1 if guardCondition.result then sysMLv1ObjectFlow {
          calc guardCondition {
            return : ScalarValues::Boolean;
            language "English"
            /*
             * guard says ok
             */
          }
        }
        succession flow sysMLv1ObjectFlow of SysMLv1Block from
                sysMLv1Action1.outputValue to sysMLv1Action2.inputValue;

        action sysMLv1Action2 {
                out 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 src.guard.oclIsUndefined()
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific 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)
}
```

7.7.3.3.41 ObjectFlowGuardSuccessionTargetEndFeature_Mapping

Description

Creates a feature element for the UML4SysML::ObjectFlow mapping.

General Mappings

GenericToFeature_Mapping

Mapping Source

ObjectFlow

Mapping Target

Feature

Owned Mappings

- objectFlowGuardSuccessionTargetEndSubsetting : ObjectFlowGuardSuccessionTargetEndSubsetting_Mapping

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
Set{objectFlowGuardSuccessionTargetEndSubsetting.to}
```

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

```
true
```

7.7.3.3.42 ObjectFlowGuardSuccessionTargetEndFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToEndFeatureMembership_Mapping

Mapping Source

ObjectFlow

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
ObjectFlowGuardSuccessionTargetEndFeature_Mapping.getMapped(from)
```

7.7.3.3.43 ObjectFlowGuardSuccessionTargetEndSubsetting_Mapping

Description

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

General Mappings

GenericToSubsetting_Mapping

Mapping Source

ObjectFlow

Mapping Target

Subsetting

Owned Mappings

- objectFlowGuardSuccessionTargetEndFeature : ObjectFlowGuardSuccessionTargetEndFeature_Mapping

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
ObjectFlow_Mapping.getMapped(from)
```

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

```
objectFlowGuardSuccessionTargetEndFeature.to
```

7.7.3.3.44 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

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.3.3.45 ObjectFlowItemFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ObjectFlow

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 if from.source.type.oclIsUndefined() then
    ObjectFlowItemFeatureUntyped_Mapping.getMapped(from.source)
else
    ObjectFlowItemFeature_Mapping.getMapped(from.source)
endif
```

7.7.3.3.46 ObjectFlowItemFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

TypedElementFeatureTyping_Mapping

Mapping Source

ObjectNode

Mapping Target

FeatureTyping

Owned Mappings

(none)

7.7.3.3.47 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)

7.7.3.3.48 ObjectFlowEndFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToEndFeatureMembership_Mapping

Mapping Source

ActivityNode

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
ObjectFlowItemFlowEnd_Mapping.getMapped(from)
```

7.7.3.3.49 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

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{ObjectFlowItemFlowSubsetting_Mapping.getMapped(from),
ObjectFlowItemFlowFeatureMembership_Mapping.getMapped(from)}
```

7.7.3.3.50 ObjectFlowItemFlowFeature_Mapping

Description

Creates a feature element for the UML4SysML::ObjectFlow mapping.

General Mappings

GenericToFeature_Mapping

Mapping Source

ActivityNode

Mapping Target

ItemFeature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.3.3.51 ObjectFlowItemFlowFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ActivityNode

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
ObjectFlowItemFlowFeature_Mapping.getMapped(from)
```

7.7.3.3.52 ObjectFlowItemFlowRedefinition_Mapping

Description

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

General Mappings

GenericToRedefinition_Mapping

Mapping Source

ActivityNode

Mapping Target

Redefinition

Owned Mappings

(none)

7.7.3.3.53 ObjectFlowItemFlowSubsetting_Mapping

Description

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

General Mappings

GenericToSubsetting_Mapping

Mapping Source

ActivityNode

Mapping Target

Subsetting

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 if from.oclIsKindOf(UML::ActivityParameterNode) then
    Parameter_Mapping.getMapped(from.parameter)
else if from.oclIsKindOf(UML::Pin) then
        CommonAction_Mapping.getMapped(from.owner)
    else if from.oclIsKindOf(UML::InitialNode) then
            SysMLv2::ActionUsage.allInstances()
            ->any(e | e.qualifiedName =  'Actions::Action::start')
        else if from.oclIsKindOf(UML::FinalNode) then
                SysMLv2::ActionUsage.allInstances()
                ->any(e | e.qualifiedName =  'Actions::Action::done')
            else
                from
            endif
        endif
    endif
endif
```

7.7.3.3.54 ObjectFlowTransitionUsageFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ObjectFlow

Mapping Target

TransitionFeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific 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.getMapped(from.guard)
else
    from.guard
endif
```

7.7.3.3.55 VariableAttribute_Mapping

Description

A UML4SysML::Variable is mapped to a SysML v2 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:

```
src.type.oclIsKindOf(UML::DataType)
```

Mapping rules

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

7.7.3.3.56 VariableFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

TypedElementFeatureTyping_Mapping

Mapping Source

Variable

Mapping Target

FeatureTyping

Owned Mappings

(none)

7.7.3.3.57 VariableItem_Mapping

Description

A UML4SysML::Variable is mapped to a SysML v2 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 src.type.oclIsKindOf(UML::DataType)
```

Mapping rules

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

7.7.3.3.58 VariableMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

ElementFeatureMembership_Mapping

Mapping Source

Variable

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.4 Classification

7.7.4.1 Overview

The following table gives an overview of which SysML v2 elements the UML4SysML::Classification elements are transformed with which mapping class. The mapping details are in 7.7.4.3.

The justifications for the elements without mapping are given in 7.7.4.2.

Table 5. List of all mappings

SysML v1 Concept	SysML v2 Concept
Generalization	Subclassification
GeneralizationSet	
InstanceSpecification	PartUsage ConnectionUsage EnumerationUsage
InstanceValue	FeatureReferenceExpression
Operation	PerformActionUsage PerformActionUsage
Parameter	ReferenceUsage
ParameterSet	

SysML v1 Concept	SysML v2 Concept
Property	PartUsage PortUsage ReferenceUsage PortUsage AttributeUsage Feature PartUsage AttributeUsage Feature ItemUsage PartUsage AttributeUsage
Slot	Feature
Substitution	

7.7.4.2 UML4SysML::Classifications elements not mapped

7.7.4.3 Mapping Specifications

7.7.4.3.1 BehavioralFeature_Mapping

Description

The mapping class is the abstract base class for UML4SysML::BehavioralFeature mappings.

General Mappings

GenericToUsage_Mapping
Namespace_Mapping

Mapping Source

BehavioralFeature

Mapping Target

Usage

Owned Mappings

(none)

7.7.4.3.2 Classifier_Mapping

Description

The mapping class is the abstract base class for all mapping classes that map specializations of UML4SysML::Classifier elements.

General Mappings

GenericToClassifier_Mapping
Namespace_Mapping

Mapping Source

Classifier

Mapping Target

Classifier

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 let generalizations : Set(UML::Generalization) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Generalization)) in
let toElementFMS: Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Feature)) in
let toElementOMS: Set(UML::Element) =
    (from.ownedElement - toElementFMS) - generalizations  in
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))
```

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

```
 from.isAbstract
```

7.7.4.3.3 DefaultLowerBound_Mapping

Description

The mapping class creates the default lower bound of a multiplicity element.

General Mappings

GenericToExpression_Mapping

Mapping Source

Element

Mapping Target

LiteralInteger

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- LiteralInteger::value () : Integer [1]

  ```
  1
  ```

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

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

7.7.4.3.4 DefaultMultiplicityBoundFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Element

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  true
  ```

7.7.4.3.5 DefaultMultiplicityElement_Mapping

Description

The mapping class creates a feature element representing the default multiplicity.

General Mappings

GenericToFeature_Mapping

Mapping Source

Element

Mapping Target

MultiplicityRange

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- MultiplicityRange::declaredName () : String [0..1]

  ```
  'defaultMultiplicity'
  ```

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

  ```
  true
  ```

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

  ```
   OrderedSet{DefaultMultiplicityLowerBoundFeatureMembership_Mapping.getMapped(from),
  DefaultMultiplicityUpperBoundFeatureMembership_Mapping.getMapped(from)}
  ```

7.7.4.3.6 DefaultMultiplicityLowerBoundFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

DefaultMultiplicityBoundFeatureMembership_Mapping

Mapping Source

Element

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
DefaultLowerBound_Mapping.getMapped(from)
```

7.7.4.3.7 DefaultMultiplicityMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Element

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
DefaultMultiplicityElement_Mapping.getMapped(from)
```

7.7.4.3.8 DefaultMultiplicityUpperBoundFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

DefaultMultiplicityBoundFeatureMembership_Mapping

Mapping Source

Element

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
DefaultUpperBound_Mapping.getMapped(from)
```

7.7.4.3.9 DefaultUpperBound_Mapping

Description

The mapping class creates the default upper bound of a multiplicity element.

General Mappings

GenericToExpression_Mapping

Mapping Source

Element

Mapping Target

LiteralInteger

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- LiteralInteger::value () : Integer [1]

  ```
  1
  ```

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

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

7.7.4.3.10 DefaultValue_Mapping

Description

The expected SysML v2 textual syntax of a mapped SysML v2 default value is as follows:

```
attribute sysMLv1Property : ScalarValues::String default := "default value";
```

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Property

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  true
  ```

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

```
        from.defaultValue
```

7.7.4.3.11 ElementFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Element

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [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
```

7.7.4.3.12 Generalization_Mapping

Description

A UML4SysML::Generalization relationship is mapped to a SysML v2 Subclassification.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
part def SysMLv1BlockGeneral;
part def SysMLv1BlockSpecial :> SysMLv1BlockGeneral;
```

General Mappings

GenericToSpecialization_Mapping
ElementMain_Mapping

Mapping Source

Generalization

Mapping Target

Subclassification

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Subclassification::superclassifier () : Classifier [1]

```
 if from.general.oclIsTypeOf(UML::PrimitiveType)
           and not (Helper.getScalarValueType(from.general)
               = OclUndefined) then
    Helper.getScalarValueType(from.general)
else
    Classifier_Mapping.getMapped(from.general)
endif
```

- Subclassification::subclassifier () : Classifier [1]

```
 Classifier_Mapping.getMapped(from.specific)
```

7.7.4.3.13 InstanceSpecificationLink_Mapping

Description

The UML4SysML::InstanceSpecification that is a link is mapped to a SysMLv2 ConnectionUsage.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
part def SysMLv1Block1;
part def SysMLv1Block2;
connection def SysMLv1Association {
        end : SysMLv1Block1[1];
        end : SysMLv1Block2[1];
}
part sysMLv1InstanceSpecification1 : SysMLv1Block1;
part sysMLv1InstanceSpecification2 : SysMLv1Block2;
connection sysMLv1Link : SysMLv1Association
        connect sysMLv1InstanceSpecification1 to sysMLv1InstanceSpecification2;
```

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:

```
src.classifier->select( c | c.oclIsTypeOf(UML::Association))->size() > 0
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific 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(from, g)))
    ->asSet()
```

7.7.4.3.14 InstanceSpecification_Mapping

Description

The UML4SysML::InstanceSpecification that is not a link is mapped to a SysMLv2 PartDefinition.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
part def SysMLv1Block {
        attribute sysMLv1ValueProperty : ScalarValues::String;
}

part sysMLv1InstanceSpecification : SysMLv1Block {
        redefines sysMLv1ValueProperty = "Hello InstanceSpecification";
}
```

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:

```
src.classifier->select( c | c.oclIsTypeOf(UML::Association))->size() = 0
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 SlotMembership_Mapping.getMappedColl(from.slot)
->union(from.classifier
    ->collect(g | InstanceSpecificationFeatureTyping_Mapping.getMapped(from, g)))
    ->asSet()
```

- PartUsage::ownedFeatureMembership () : FeatureMembership [0..*]

```
 from.classifier
->collect(c | InstanceSpecificationToGeneralization_Mapping.getMapped(from, c))
```

7.7.4.3.15 InstanceSpecificationFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

InstanceSpecification

Mapping Target

FeatureTyping with qualifier: classifier:Classifier

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureTyping::type (in classifier : Classifier) : Type [1]

```
Classifier_Mapping.getMapped(classifier)
```

7.7.4.3.16 InstanceValue_Mapping

Description

The UML4SysML::InstanceValue is mapped to a SysMLv2 FeatureReferenceExpression.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
part def SysMLv1Block1;
part sysMLv1InstanceSpecification : SysMLv1Block1;
part def SysMLv1Block2 {
        part sysMLv1PartProperty : SysMLv1Block1
                = sysMLv1InstanceSpecification;
}
```

General Mappings

ValueSpecification_Mapping

Mapping Source

InstanceValue

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 ElementOwnership_Mapping.getMappedColl(from.ownedComment)
->including(InstanceValueMembership_Mapping.getMapped(from.instance))
->including(ReturnParameterFeatureMembership_Factory.create())
```

7.7.4.3.17 InstanceValueMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToMembership_Mapping

Mapping Source

InstanceSpecification

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
from
```

7.7.4.3.18 LowerBoundValueFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

MultiplicityElement

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  LiteralInteger_Mapping.getMapped(from.lowerValue)
  ```

7.7.4.3.19 MultiplicityElement_Mapping

Description

A UML4SysML::MultiplicityElement is mapped to a SysML v2 MultiplicityRange.

General Mappings

GenericToFeature_Mapping

Mapping Source

MultiplicityElement

Mapping Target

MultiplicityRange

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 from.isUnique
```

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

```
 OrderedSet{MultiplicityLowerBoundOwningMembership_Mapping.getMapped(from),
MultiplicityUpperBoundOwningMembership_Mapping.getMapped(from)}
```

- MultiplicityRange::declaredName () : String [0..1]

```
 'multiplicity'
```

7.7.4.3.20 MultiplicityLowerBoundOwningMembership_Mapping

Description

Creates a owning membership relationship for *ownedMemberElement()*.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

MultiplicityElement

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 if from.lowerValue.oclIsUndefined() then
    DefaultLowerBound_Mapping.getMapped(from)
else
    from.lowerValue
endif
```

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

```
 'lowerBound'
```

7.7.4.3.21 MultiplicityMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

MultiplicityElement

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  MultiplicityElement_Mapping.getMapped(from)
  ```

7.7.4.3.22 MultiplicityUpperBoundOwningMembership_Mapping

Description

Creates a owning membership relationship for *ownedMemberElement()*.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

MultiplicityElement

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 'upperBound'
```

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

```
 if from.upperValue.oclIsUndefined() then
    DefaultUpperBound_Mapping.getMapped(from)
else
    from.upperValue
endif
```

7.7.4.3.23 Operation_Mapping

Description

A UML4SysML::Operation is mapped to a SysML v2 PerformActionUsage.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
part def SysMLv1Block {
          perform action sysMLv1Operation {
                  in parIn : ScalarValues::Boolean;
                 out result : ScalarValues::Integer;
          }
}
```

General Mappings

BehavioralFeature_Mapping
GenericToActionUsage_Mapping

Mapping Source

Operation

Mapping Target

PerformActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
let parameters: Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Parameter)) in
let parameterSets: Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::ParameterSet)) in
ElementOwnership_Mapping.getMappedColl(from.ownedComment)
->union(parameters->collect(e | ParameterMembership_Mapping.getMapped(e)))
->union(parameterSets->collect(e | ParameterSetMembership_Mapping.getMapped(e)))
```

7.7.4.3.24 Parameter_Mapping

Description

A UML4SysML::Parameter is mapped to a SysML v2 ReferenceUsage.

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 : ScalarValues::Boolean;
}
```

General Mappings

GenericToReferenceUsage_Mapping
NamedElementMain_Mapping

Mapping Source

Parameter

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
Helper.getKerMLParameterDirectionKind(from.direction)
```

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

```
 if from.direction = UML::ParameterDirectionKind::return then 'result' else from.name endif
```

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

```
let typings: Set(KerML::FeatureTyping) =
    if from.type.oclIsUndefined() then
        Set{}
    else
        Set{ParameterToFeatureTyping_Mapping.getMapped(from)}
    endif in
let multiplicities: Set(KerML::Relationship) =
    Set{MultiplicityMembership_Mapping.getMapped(from)} in
let defaultValues: Set(KerML::Relationship) =
    if from.defaultValue.oclIsUndefined() then
        Set{}
    else
        Set{ParameterDefaultValue_Mapping.getMapped(from)}
    endif in
ElementOwnership_Mapping.getMappedColl(from.ownedComment)->asSet()
->union(typings)
->union(multiplicities)
->union(defaultValues)
```

7.7.4.3.25 ParameterDefaultValue_Mapping

Description

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
attribute value : ScalarValues::String default := "default value";
```

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Parameter

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  from.defaultValue
  ```

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

  ```
  true
  ```

7.7.4.3.26 ParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

Parameter

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  Parameter_Mapping.getMapped(from)
  ```

7.7.4.3.27 ParameterSet_Mapping

Description

A UML4SysML::ParameterSet is mapped to a SysML v2 ReferenceUsage.

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 [0..1];
```

```
        inout parInOut [0..1];
        out parOut [0..1];
        out parReturn [0..1];

        sysMLv1ParameterSet1 [1] {
                ref parIn = SysMLv1Activity::parIn;
                assert constraint sysMLv1ParameterSet1Condition {
                        language "English"
                        /*
                         * opaque expression parameter set 1
                        */
                }
        }
        sysMLv1ParameterSet2  [1]  {
                ref parInOut = SysMLv1Activity::parInOut;
                ref parOut = SysMLv1Activity::parOut;
                ref parReturn = SysMLv1Activity::parReturn;
        }
}
```

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

ParameterSet

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 from.parameter
->collect(p | ParameterSetParameterFeatureMembership_Mapping.getMapped(from, p))
->asSet()
```

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

```
 from.name
```

7.7.4.3.28 ParameterSetMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ParameterSet

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
ParameterSet_Mapping.getMapped(from)
```

7.7.4.3.29 ParameterSetParameterFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ParameterSet

Mapping Target

FeatureMembership with qualifier: parameter:Parameter

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature (in parameter : Parameter) : Feature [1]

```
ParameterSetParameterReferenceUsage_Mapping.getMapped(parameter)
```

7.7.4.3.30 ParameterSetParameterReferenceUsage_Mapping

Description

The mapping class creates the reference usage element for the UML4SysML::ParameterSet mapping.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Parameter

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{ParameterSetParameterReferenceUsageFeatureValue_Mapping.getMapped(from),
MultiplicityMembership_Mapping.getMapped(from)}
```

7.7.4.3.31 ParameterSetParameterReferenceUsageFeatureValue_Mapping

Description

The mapping class creates the feature reference expression for the reference usage element of the UML4SysML::ParameterSet mapping.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Parameter

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  ParameterSetParameterReferenceUsageFeatureValueExpression_Mapping.getMapped(from)
  ```

7.7.4.3.32 ParameterSetParameterReferenceUsageFeatureValueExpression_Mapping

Description

The mapping class creates the feature reference expression for the UML4SysML::ParameterSet mapping.

General Mappings

GenericToFeatureReferenceExpression_Mapping

Mapping Source

Parameter

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{ParameterSetParameterReferenceUsageMembership_Mapping.getMapped(from),
CommonReturnParameterFeatureMembership_Mapping.getMapped(from)}
```

7.7.4.3.33 ParameterSetParameterReferenceUsageMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToMembership_Mapping

Mapping Source

Parameter

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
from
```

7.7.4.3.34 ParameterToFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

TypedElementFeatureTyping_Mapping

Mapping Source

Parameter

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
parameter.to
```

7.7.4.3.35 Property_Mapping

Description

A UML4SysML::Property is mapped to a SysML v2 Feature. The mapping class maps properties with a type.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
part def SysMLv1Block {
        attribute sysMLv1Property [0..1] : ScalarValues::Integer;
}
```

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) and not
    Helper.hasStereotypeApplied(src.owner, 'SysML::ConstraintBlocks::ConstraintBlock') 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.

7.7.4.3.36 PropertyCommon_Mapping

Description

The mapping class is the abstract base class for UML4SysML::Property mappings.

General Mappings

StructuralFeature_Mapping

Mapping Source

Property

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  from.isDerived
  ```

- 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::isComposite () : Boolean [1]

```
 from.isComposite
```

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

```
 if from.association.oclIsUndefined() then
    false
else
    from.association.ownedEnd->includes(from)
endif
```

7.7.4.3.37 PropertySubsetting_Mapping

Description

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

General Mappings

GenericToSubsetting_Mapping

Mapping Source

Property

Mapping Target

Subsetting with qualifier: subsettedProperty:Property

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Subsetting::subsettedFeature (in subsettedProperty : Property) : Feature [1]

  ```
  Property_Mapping.getMapped(subsettedProperty)
  ```

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

  ```
  Property_Mapping.getMapped(from)
  ```

7.7.4.3.38 PropertyUntyped_Mapping

Description

A UML4SysML::Property is mapped to a SysML v2 Feature. The mapping class maps properties without a type.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
part def SysMLv1Block {
        attribute sysMLv1Property;
}
```

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:

```
src.type.oclIsUndefined() and not
Helper.hasStereotypeApplied(src.owner, 'SysML::ConstraintBlocks::ConstraintBlock')
```

Mapping rules

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

7.7.4.3.39 Realization_Mapping

Description

A UML4SysML::Realization relationship is mapped to a SysML v2 Dependency.

General Mappings

Abstraction_Mapping

Mapping Source

Realization

Mapping Target

Dependency

Owned Mappings

(none)

7.7.4.3.40 Slot_Mapping

Description

A UML4SysML::Slot is mapped to a SysML v2 Feature.

General Mappings

GenericToFeature_Mapping
ElementMain_Mapping

Mapping Source

Slot

Mapping Target

Feature

Owned Mappings

(none)

7.7.4.3.41 SlotMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Slot

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  from.isReadOnly
  ```

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

  ```
  from.definingFeature.name
  ```

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

  ```
  from
  ```

7.7.4.3.42 SlotFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Slot

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  ElementMain_Mapping.getMapped(from)
  ```

7.7.4.3.43 SlotValue_Mapping

Description

Issue here since a KerML feature cannot have more than one FeatureValue while a UML4SysML::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

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  from
  ```

- FeatureValue::featureWithValue () : Feature [1]

  ```
  Slot_Mapping.getMapped(from.owner)
  ```

7.7.4.3.44 StructuralFeature_Mapping

Description

The mapping class is the abstract base class for all UML4SysML::StructuralFeature mappings.

General Mappings

GenericToFeature_Mapping

Mapping Source

StructuralFeature

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Feature::isReadOnly () : Boolean [1]
  abstract rule
- Feature::isAbstract () : Boolean [1]

```
 false
```

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

```
 from.isUnique
```

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

```
 let typing: KerML::FeatureTyping =
    StructuralFeatureToFeatureTyping_Mapping.getMapped(from) in
if typing.oclIsUndefined() then
    Set{MultiplicityMembership_Mapping.getMapped(from)}
else
    Set{MultiplicityMembership_Mapping.getMapped(from), typing}
endif
```

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

```
 from.isOrdered
```

7.7.4.3.45 StructuralFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

StructuralFeature

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific 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
```

7.7.4.3.46 StructuralFeatureToFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

TypedElementFeatureTyping_Mapping

Mapping Source

StructuralFeature

Mapping Target

FeatureTyping

Owned Mappings

(none)

7.7.4.3.47 TypedElementFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

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()
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific 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
```

7.7.4.3.48 UpperBoundValueFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

MultiplicityElement

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific 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
```

7.7.5 CommonBehavior

7.7.5.1 Overview

The following table gives an overview of which SysML v2 elements the UML4SysML::CommonBehavior elements are transformed with which mapping class. The mapping details are in 7.7.5.3.

The justifications for the elements without mapping are given in 7.7.5.2.

Table 7. List of all mappings

SysML v1 Concept	SysML v2 Concept
AnyReceiveEvent	
CallEvent	
ChangeEvent	TextualRepresentation
FunctionBehavior	
OpaqueBehavior	ActionDefinition ActionUsage
SignalEvent	
TimeEvent	TextualRepresentation
Trigger	AcceptActionUsage

7.7.5.2 UML4SysML::CommonBehavior elements not mapped

Table 8. 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.

7.7.5.3 Mapping Specifications

7.7.5.3.1 Behavior_Mapping

Description

The mapping class is the abstract base class for all UML4SysML::Behavior mappings.

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

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
let parameters: Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Parameter)) in
let parameterSets: Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::ParameterSet)) in
let features: Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Property)) in
let elementsOMS: Set(UML::Element) =
    (((from.ownedElement - parameters) parameterSets) - features) in
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)))
```

7.7.5.3.2 ChangeEvent_Mapping

Description

T#3 meeting, 2022-12-14: Do not use automatic rules! Events are not single elements in SysML v2. Consider it in the transformation for AcceptEventAction, Transition

General Mappings

GenericToTextualRepresentation_Mapping
NamedElementMain_Mapping

Mapping Source

ChangeEvent

Mapping Target

TextualRepresentation

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific 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
        OclUndefined
    else
        from.changeExpression.oclAsType(UML::OpaqueExpression).body.get(0)
    endif
else
    OclUndefined
endif
```

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

```
 if from.changeExpression.oclIsKindOf(UML::OpaqueExpression) then
    if from.changeExpression.
        oclAsType(UML::OpaqueExpression).language->size() = 0 then
        OclUndefined
    else
        from.changeExpression.oclAsType(UML::OpaqueExpression).language.get(0)
```

```
            endif
        else
            OclUndefined
        endif
```

7.7.5.3.3 CommonOpaqueBehavior_Mapping

Description

The mapping class is the abstract base class for UML4SysML::OpaqueBehavior mappings.

General Mappings

Behavior_Mapping

Mapping Source

OpaqueBehavior

Mapping Target

Behavior

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
let parameters : Set(UML::Parameter) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Parameter)) in
let parameterSets : Set(UML::ParameterSet) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::ParameterSet)) in
let features : Set(UML::Property) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Property)) in
let elementsOMS: Set(UML::Element) =
    (((from.ownedElement - parameters) - parameterSets) - features) in
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)))
```

7.7.5.3.4 OpaqueBehaviorAsDefinition_Mapping

Description

A UML4SysML::OpaqueBehavior that is owned by a package is mapped to a SysML v2 ActionDefinition.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action def SysMLv1OpaqueBehavior {
        language "Built-in Math"
        /*
         * result = 42 + 23;
        */
}
```

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.

7.7.5.3.5 OpaqueBehaviorAsUsage_Mapping

Description

A UML4SysML::OpaqueBehavior that is not owned by a package is mapped to a SysML v2 ActionUsage.

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 sysMLv1OpaqueBehavior {
                language "Built-in Math"
                /*
                 * result = 42 + 23;
                */
        }
}
```

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.

7.7.5.3.6 OpaqueBehaviorMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

OpaqueBehavior

Mapping Target

OwningMembership with qualifier: language:String

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- OwningMembership::ownedMemberElement (in language : String) : Element [1]

```
 OpaqueBehaviorSpecification_Mapping.getMapped(from, language)
```

7.7.5.3.7 OpaqueBehaviorSpecification_Mapping

Description

The mapping class creates the SysML v2 TextualRepresentation elements from the languages and bodies properties of the given UML4SysML::OpaqueBehavior.

General Mappings

GenericToTextualRepresentation_Mapping

Mapping Source

OpaqueBehavior

Mapping Target

TextualRepresentation with qualifier: language:String

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 language
```

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

```
 let index:Integer = from.language->indexOf(language) in
from._'body'->at(index)
```

7.7.5.3.8 TimeEvent_Mapping

Description

T#3 meeting, 2022-12-14: Do not use automatic rules! Events are not single elements in SysML v2. Consider it in the transformation for AcceptEventAction, Transition

General Mappings

NamedElementMain_Mapping
GenericToTextualRepresentation_Mapping

Mapping Source

TimeEvent

Mapping Target

TextualRepresentation

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
'tbd timeevent'
```

7.7.5.3.9 Trigger_Mapping

7.7.6 CommonStructure

7.7.6.1 Overview

The following table gives an overview of which SysML v2 elements the UML4SysML::CommonStructure elements are transformed with which mapping class. The mapping details are in 7.7.6.2.

Table 10. List of all mappings

SysML v1 Concept	SysML v2 Concept
Abstraction	Dependency Dependency Dependency SatisfyRequirementUsage AllocationUsage AllocationDefinition Dependency ConnectionUsage
Comment	Comment Package ConcernUsage Comment
Constraint	ConstraintDefinition

SysML v1 Concept	SysML v2 Concept
Dependency	Dependency Dependency Dependency Dependency SatisfyRequirementUsage AllocationUsage AllocationDefinition Dependency ConnectionUsage Dependency
ElementImport	MembershipImport
PackageImport	NamespaceImport
Realization	Dependency
Usage	Dependency

7.7.6.2 Mapping Specifications

7.7.6.2.1 Abstraction_Mapping

Description

A UML4SysML::Abstraction relationship is mapped to a SysML v2 Dependency relationship.

General Mappings

Dependency_Mapping

Mapping Source

Abstraction

Mapping Target

Dependency

Owned Mappings

(none)

7.7.6.2.2 Comment_Mapping

Description

A UML4SysML::Comment is mapped to a SysML v2 Comment.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
part def SysMLv1Block1;
part def SysMLv1Block2;
action def SysMLv1Activitiy {
        comment about SysMLv1Activity, SysMLv1Block1
                /* comment body */
}
comment about SysMLv1Block1, SysMLv1Block /* comment body */
```

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(src, 'SysML::ModelElements::ElementGroup')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
   if from.body->isEmpty() then '' else from.body endif
  ```

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

  ```
    from.annotatedElement
  ->collect(e | CommentAnnotation_Mapping.getMapped(from, e))
  ```

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

  ```
   ElementOwnership_Mapping.getMappedColl(from.ownedComment)
  ->union(self.annotation())
  ```

7.7.6.2.3 CommentAnnotation_Mapping

Description

The mapping class creates the annotation relationship for the UML4SysML::Comment mapping.

General Mappings

GenericToAnnotation_Mapping

Mapping Source

Comment

Mapping Target

Annotation with qualifier: annotatedElement:Element

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Annotation::annotatedElement (in annotatedElement : Element) : Element [1]

```
ElementMain_Mapping.getMapped(annotatedElement)
```

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

```
null
```

- Annotation::annotatingElement () : AnnotatingElement [1]

```
Comment_Mapping.getMapped(from)
```

7.7.6.2.4 Constraint_Mapping

Description

A UML4SysML::Constraint is mapped to a SysML v2 ConstraintDefinition and AssertConstraintUsages for the constrained elements.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
part def SysMLv1Block {
        constraint def SysMLv1Constraint {
                calc sysMLv1Constraint {
                         language "English"
                        /*
                         * constraint specification
                         */
                 }
        }
        assert constraint assert_sysMLv1Constraint : SysMLv1Constraint;
}
```

General Mappings

GenericToConstraintDefinition_Mapping
NamedElementMain_Mapping

Mapping Source

Constraint

Mapping Target

ConstraintDefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 ElementOwnership_Mapping.getMappedColl(from.ownedComment)->asSet()
->union(Set{ElementFeatureMembership_Mapping.getMapped(from.specification),
CommonReturnParameterReferenceUsageMembership_Mapping.getMapped(from.specification)})
```

7.7.6.2.5 ConstrainedElementFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Constraint

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
ConstraintUsage_Mapping.getMapped(from)
```

7.7.6.2.6 ConstraintUsageFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Constraint

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
from
```

7.7.6.2.7 ConstraintUsage_Mapping

Description

The mapping class creates the SysML v2 AssertConstraintUsage elements for the constrained elements of the UML4SysML::Constraint mapping.

General Mappings

GenericToUsage_Mapping

Mapping Source

Constraint

Mapping Target

AssertConstraintUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- AssertConstraintUsage::declaredName () : String [0..1]

```
 'assert_' + from.name
```

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

```
 ElementOwnership_Mapping.getMappedColl(from.ownedComment)->asSet()
->union(Set{ConstraintUsageFeatureTyping_Mapping.getMapped(from),
CommonReturnParameterReferenceUsageMembership_Mapping.getMapped(from)})
```

7.7.6.2.8 Dependency_Mapping

Description

A UML4SysML::Dependency relationship is mapped to a SysML v2 Dependency relationship.

General Mappings

DirectedRelationship_Mapping

Mapping Source

Dependency

Mapping Target

Dependency

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Dependency::supplier () : Element [0..*]

```
from.target->collect(e | ElementMain_Mapping.getMapped(e))
```

- Dependency::client () : Element [0..*]

```
from.source->collect(e | ElementMain_Mapping.getMapped(e))
```

- Dependency::declaredName () : String [0..1]

```
from.name
```

7.7.6.2.9 DirectedRelationship_Mapping

Description

The mapping class is the abstract base class for all UML4SysML::DirectedRelationship mappings.

General Mappings

Relationship_Mapping

Mapping Source

DirectedRelationship

Mapping Target

Relationship

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific 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))
```

7.7.6.2.10 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

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
Helper.getID(from)
```

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

```
ElementOwnership_Mapping.getMappedColl(from.ownedComment)
```

7.7.6.2.11 ElementMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToMembership_Mapping

Mapping Source

Element

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 ElementMain_Mapping.getMapped(from)
```

- Membership::membershipOwningNamespace () : Element [0..*]

```
 Set{ElementMain_Mapping(from)}
-- will not be used since corresponding attribute is derived,
-- but required for redefinition
```

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

```
 if (from.oclIsKindOf(UML::NamedElement)) then
    from.oclAsType(UML::NamedElement).visibility
else
    KerML::VisibilityKind::public
endif
```

7.7.6.2.12 ElementOwnership_Mapping

Description

The mapping class is the abstract base class for mappings that target ownership relationships.

General Mappings

GenericToRelationship_Mapping
UniqueMapping

Mapping Source

Element

Mapping Target

Relationship

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
OrderedSet{ElementMain_Mapping.getMapped(from.owner)}
```

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

```
OrderedSet{ElementMain_Mapping.getMapped(from)}
```

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

```
self.target()
```

7.7.6.2.13 ElementOwningMembership_Mapping

Description

Creates a owning membership relationship for *ownedMemberElement()*.

General Mappings

ElementMembership_Mapping
ElementOwnership_Mapping

Mapping Source

Element

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific 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 attribute is derived,
-- but required for redefinition
```

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

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

7.7.6.2.14 NamedElementMain_Mapping

Description

The mapping class is the abstract base class for mappings of UML4SysML::NamedElements.

General Mappings

ElementMain_Mapping

Mapping Source

NamedElement

Mapping Target

Element

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 from.name
```

7.7.6.2.15 Namespace_Mapping

Description

The mapping class is the abstract base class for UML4SysML::Namespace mappings.

General Mappings

GenericToNamespace_Mapping
NamedElementMain_Mapping

Mapping Source

Namespace

Mapping Target

Namespace

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Namespace::ownedImport () : Import [0..*]

```
Set{}
```

7.7.6.2.16 Relationship_Mapping

Description

Th mapping class is the abstract base class for UML4SysML::Relationship mappings.

General Mappings

GenericToRelationship_Mapping
ElementMain_Mapping

Mapping Source

Relationship

Mapping Target

Relationship

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 from.relatedElement->select(e | from.ownedElement->includes(e))
->collect(e | ElementMain_Mapping.getMapped(e))
```

- Relationship::owningRelatedElement () : Element [0..1]

```
ElementMain_Mapping.getMapped(from.owner)
```

7.7.6.2.17 Usage_Mapping

Description

A UML4SysML::Usage relationship is mapped to a SysML v2 Dependency relationship.

General Mappings

Dependency_Mapping

Mapping Source

Usage

Mapping Target

Dependency

Owned Mappings

(none)

7.7.7 InformationFlows

7.7.7.1 Overview

The following table gives an overview of which SysML v2 elements the UML4SysML::InformationFlows elements are transformed with which mapping class. The mapping details are in 7.7.7.2.

Table 11. List of all mappings

SysML v1 Concept	SysML v2 Concept
InformationFlow	FlowConnectionUsage FlowConnectionDefinition
InformationItem	ItemDefinition

7.7.7.2 Mapping Specifications

7.7.7.2.1 InformationFlow_Mapping

Description

A UML4SysML::InformationFlow is mapped to a SysML v2 FlowConnectionDefinition.

General Mappings

Relationship_Mapping

Mapping Source

InformationFlow

Mapping Target

FlowConnectionDefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific 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()
```

7.7.7.2.2 InformationFlowEndCommonMembership_Mapping

Description

The mapping class is the abstract base class for the concrete mapping classes for the source and the target membership relationships of the FlowConnectionDefinition for the UML4SysML::InformationFlow mapping.

General Mappings

GenericToElement_Mapping

Mapping Source

InformationFlow

Mapping Target

Element

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Element::ownedMemberFeature (in end : NamedElement) : Feature [1]
  abstract rule
- Element::memberShortName () : String [0..1]

  ```
  null
  ```

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

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

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

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

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

  ```
  null
  ```

7.7.7.2.3 InformationFlowSource_Mapping

Description

The mapping class creates the source feature of the FlowConnectionDefinition for the mapping of UML4SysML::InformationFlow.

General Mappings

GenericToElement_Mapping

Mapping Source

InformationFlow

Mapping Target

Feature with qualifier: source:NamedElement

Owned Mappings

- informationFlowSourceFeatureTyping : InformationFlowSourceFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Feature::name (in source : NamedElement) : String [0..1]

```
'source'
```

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

```
Set{informationFlowSourceFeatureTyping.to}
```

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

```
true
```

7.7.7.2.4 InformationFlowSourceMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

InformationFlowEndCommonMembership_Mapping

Mapping Source

InformationFlow

Mapping Target

FeatureMembership with qualifier: source:NamedElement

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature (in source : NamedElement) : Feature [1]

```
InformationFlowSource_Mapping.getMapped(from, source)
```

7.7.7.2.5 InformationFlowSourceFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToElement_Mapping

Mapping Source

InformationFlow

Mapping Target

FeatureTyping with qualifier: source:NamedElement

Owned Mappings

- informationFlowSource : InformationFlowSource_Mapping

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureTyping::type (in source : NamedElement) : Type [1]

  ```
  ElementMain_Mapping.getMapped(source)
  ```

- FeatureTyping::typedFeature (in source : NamedElement) : Feature [1]

  ```
  InformationFlowSource_Mapping.getMapped(from, source)
  ```

7.7.7.2.6 InformationFlowTarget_Mapping

Description

The mapping class creates the target feature of the FlowConnectionDefinition for the mapping of UML4SysML::InformationFlow.

General Mappings

GenericToElement_Mapping

Mapping Source

InformationFlow

Mapping Target

Feature with qualifier: target:NamedElement

Owned Mappings

- informationFlowTargetFeatureTyping : InformationFlowTargetFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
Set{informationFlowTargetFeatureTyping.to}
```

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

```
true
```

- Feature::name (in target : NamedElement) : String [0..1]

```
'target_'+target.name
```

7.7.7.2.7 InformationFlowTargetMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

InformationFlowEndCommonMembership_Mapping

Mapping Source

InformationFlow

Mapping Target

FeatureMembership with qualifier: target:NamedElement

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature (in target : NamedElement) : Feature [1]

```
InformationFlowTarget_Mapping.getMapped(from, target)
```

7.7.7.2.8 InformationFlowTargetFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToElement_Mapping

Mapping Source

InformationFlow

Mapping Target

FeatureTyping with qualifier: target:NamedElement

Owned Mappings

- informationTarget : InformationFlowTarget_Mapping

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureTyping::typedFeature (in target : NamedElement) : Feature [1]

```
InformationFlowTarget_Mapping.getMapped(from, target)
```

- FeatureTyping::type (in target : NamedElement) : Type [1]

```
ElementMain_Mapping.getMapped(target)
```

7.7.7.2.9 InformationItem_Mapping

Description

A UML4SysML::InformationItem is mapped to a SysML v2 ItemDefinition.

General Mappings

Classifier_Mapping

Mapping Source

InformationItem

Mapping Target

ItemDefinition

Owned Mappings

(none)

7.7.8 Interactions

7.7.8.1 Overview

The following table gives an overview of which SysML v2 elements the UML4SysML::Interactions elements are transformed with which mapping class. The mapping details are in 7.7.8.3.

The justifications for the elements without mapping are given in 7.7.8.2.

Table 12. List of all mappings

SysML v1 Concept	SysML v2 Concept
ActionExecutionSpecification	ActionUsage
BehaviorExecutionSpecification	ActionUsage
CombinedFragment	Interaction
ConsiderIgnoreFragment	
Continuation	
DestructionOccurrenceSpecification	
ExecutionOccurrenceSpecification	
Gate	
GeneralOrdering	
Interaction	Interaction
InteractionConstraint	
InteractionOperand	Interaction
InteractionUse	Step
Lifeline	PartUsage
Message	ItemFlow
MessageOccurrenceSpecification	
OccurrenceSpecification	
PartDecomposition	
StateInvariant	Invariant

7.7.8.2 UML4SysML::Interactions elements not mapped

Table 13. List of SysML v1 elements not mapped of this section

SysML v1 Concept	Rationale
ConsiderIgnoreFragment	Mapping is not specified yet.
Continuation	Mapping is not specified yet.
DestructionOccurrenceSpecification	Mapping is not specified yet.
ExecutionOccurrenceSpecification	Mapping is not specified yet.
Gate	Mapping is not specified yet.
GeneralOrdering	Mapping is not specified yet.

SysML v1 Concept	Rationale
InteractionConstraint	Mapping is not specified yet.
MessageOccurrenceSpecification	Mapping is not specified yet.
OccurrenceSpecification	Mapping is not specified yet.
PartDecomposition	Mapping is not specified yet.

7.7.8.3 Mapping Specifications

7.7.8.3.1 ActionExecutionSpecification_Mapping

Description

A UML4SysML::ActionExecutionSpecification is mapped to a SysML v2 ActionUsage.

General Mappings

GenericToActionUsage_Mapping
NamedElementMain_Mapping

Mapping Source

ActionExecutionSpecification

Mapping Target

ActionUsage

Owned Mappings

(none)

7.7.8.3.2 BehaviorExecutionSpecification_Mapping

Description

A UML4SysML::BehaviorExecutionSpecification is mapped to a SysML v2 ActionUsage.

General Mappings

GenericToActionUsage_Mapping
NamedElementMain_Mapping

Mapping Source

BehaviorExecutionSpecification

Mapping Target

ActionUsage

Owned Mappings

(none)

7.7.8.3.3 CombinedFragment_Mapping

Description

A UML4SysML::CombinedFragment 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

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
let operands: Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::InteractionOperand)) in
let occurrencesSpecs: Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::OccurrenceSpecification)) in
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)))
```

7.7.8.3.4 CombinedFragmentMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

CombinedFragment

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  self.memberFeature()
  ```

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

  ```
  ElementMain_Mapping.getMapped(from)
  ```

7.7.8.3.5 ExecutionSpecificationMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToEndFeatureMembership_Mapping

Mapping Source

ExecutionSpecification

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
self.memberFeature()
```

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

```
ElementMain_Mapping.getMapped(from)
```

7.7.8.3.6 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

In addition to the inherited rules, the following lists the mapping class specific 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::MessageOccurrenceSpecification)) in
let executionOccurrences: Set(UML::Element) =
    from.fragment->select(e | e.oclIsKindOf(UML::ExecutionSpecification)) in
let occurrencesSpecs: Set(UML::Element) =
    from.fragment->select(e | e.oclIsKindOf(UML::OccurrenceSpecification)) in
let messages: Set(UML::Element) = from.message in
let invariants: Set(UML::Element) =
    from.fragment->select(e | e.oclIsKindOf(UML::StateInvariant)) in
let interactionUsages: Set(UML::Element) =
    from.fragment->select(e | e.oclIsKindOf(UML::InteractionUse)) in
let combinedFragments: Set(UML::Element) =
    from.ownedElement->select( e | e.oclIsKindOf(UML::CombinedFragment)) in
let continuations: Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Continuation)) in
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(e)))
->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)))
```

7.7.8.3.7 InteractionOperand_Mapping

Description

A UML4SysML::InteractionOperand is mapped to a SysML v2 Interaction.

General Mappings

NamedElementMain_Mapping
GenericToInteraction_Mapping

Mapping Source

InteractionOperand

Mapping Target

Interaction

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
let executionOccurrences: Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::ExecutionSpecification)) in
let occurrencesSpecs: Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::OccurrenceSpecification)) in
let continuations: Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Continuation)) in
let elements: Set(UML::Element) =
    ((from.ownedElement - executionOccurrences) - occurrencesSpecs) -
    continuations in
elements->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(executionOccurrences
    ->collect(e |ExecutionSpecificationMembership_Mapping.getMapped(e)))
```

7.7.8.3.8 InteractionOperandMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

InteractionOperand

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  self.memberFeature()
  ```

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

  ```
  ElementMain_Mapping.getMapped(from)
  ```

7.7.8.3.9 InteractionUse_Mapping

Description

A UML4SysML::InteractionUse is mapped to a SysML v2 Step.

General Mappings

GenericToStep_Mapping
Namespace_Mapping

Mapping Source

InteractionUse

Mapping Target

Step

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.8.3.10 InteractionUseMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

InteractionUse

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  self.memberFeature()
  ```

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

  ```
  ElementMain_Mapping.getMapped(from)
  ```

7.7.8.3.11 InteractionUseFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

InteractionUse

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  ElementMain_Mapping.getMapped(from.refersTo)
  ```

7.7.8.3.12 LifelineMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Lifeline

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  self.memberFeature()
  ```

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

  ```
  ElementMain_Mapping.getMapped(from)
  ```

7.7.8.3.13 LifelinePartUsage_Mapping

Description

A UML4SysML::Lifeline is mapped to a SysML v2 PartUsage.

General Mappings

GenericToPartUsage_Mapping
NamedElementMain_Mapping

Mapping Source

Lifeline

Mapping Target

PartUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.8.3.14 LifelineFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Lifeline

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
ElementMain_Mapping.getMapped(from.represents.type)
```

7.7.8.3.15 Message_Mapping

Description

A UML4SysML::Message is mapped to a SysML v2 ItemFlow.

General Mappings

GenericToItemFlow_Mapping
NamedElementMain_Mapping

Mapping Source

Message

Mapping Target

ItemFlow

Owned Mappings

(none)

7.7.8.3.16 MessageMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Message

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  self.memberFeature()
  ```

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

  ```
  ElementMain_Mapping.getMapped(from)
  ```

7.7.8.3.17 StateInvariant_Mapping

Description

A UML4SysML::StateInvariant is mapped to a SysML v2 Invariant.

General Mappings

GenericToExpression_Mapping
Namespace_Mapping

Mapping Source

StateInvariant

Mapping Target

Invariant

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.8.3.18 StateInvariantMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

StateInvariant

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  ElementMain_Mapping.getMapped(from)
  ```

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

  ```
  self.memberFeature()
  ```

7.7.8.3.19 StateInvariantFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

StateInvariant

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  ElementMain_Mapping.getMapped(from.invariant)
  ```

7.7.9 Packages

7.7.9.1 Overview

The following table gives an overview of which SysML v2 elements the UML4SysML::Packages elements are transformed with which mapping class. The mapping details are in 7.7.9.3.

The justifications for the elements without mapping are given in 7.7.9.2.

Table 14. List of all mappings

SysML v1 Concept	SysML v2 Concept
Extension	
ExtensionEnd	
Image	
Model	Package
Package	Package Package Package
PackageMerge	
Profile	Package
ProfileApplication	
Stereotype	MetadataDefinition

7.7.9.2 UML4SysML::Packages elements not mapped

Table 15. 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.
Image	Mapping is not specified yet.
PackageMerge	The concept of the PackageMerge relationship is not supported by SysML v2.

7.7.9.3 Mapping Specifications

7.7.9.3.1 ElementImport_Mapping

Description

A UML4SysML::ElementImport is mapped to a SysMLv2 MembershipImport. The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
package SysMLv1Package1 {
    import SysMLv1Package2::SysMLv1Block;
    import SysMLv1Package2::SysMLv1ValueType;
}
package SysMLv1Package2 {
   part def SysMLv1Block;
   attribute def SysMLv1ValueType;
}
```

General Mappings

GenericToMembershipImport_Mapping
NamedElementMain_Mapping

Mapping Source

ElementImport

Mapping Target

MembershipImport

Owned Mappings

(none)

Applicable filters

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

```
if src.oclIsKindOf(UML::ElementImport) then
    Helper.hasMainMapping(src.oclAsType(UML::ElementImport).importedElement)
else
    false
endif
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- MembershipImport::importedMemberName () : String [0..1]

  ```
  from.alias
  ```

- MembershipImport::importedMembership () : Namespace [1]

  ```
  ElementOwningMembership_Mapping.getMapped(from.importedElement)
  ```

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

  ```
  Helper.getKerMLVisibilityKind(from.visibility)
  ```

7.7.9.3.2 Model_Mapping

Description

SysMLv2 has no explicit model element for a model. The UML4SysML::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 UML4SysML::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 SysMLv1Model {
  @SysMLv1Library::PackageData {URI="https://omg.org";}
  @SysMLv1Library::ModelData {'viewpoint'="The viewpoint of the model element.";}
}
```

General Mappings

Package_Mapping

Mapping Source

Model

Mapping Target

Package

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific 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
```

7.7.9.3.3 ModelViewpointMetadataUsage_Mapping

7.7.9.3.4 ModelViewpointMetadataFeatureMembership_Mapping

Description

The mapping class creates the feature membership relationship for the metadata feature to store the UML4SysML::Model::viewpoint property.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Model

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
ModelViewpointMetadataReferenceUsage_Mapping.getMapped(from)
```

7.7.9.3.5 ModelViewpointMetadataReferenceUsage_Mapping

Description

The mapping class creates the MetadataFeature for the mapping of the property UML4SysML::Model::viewpoint.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Model

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{ModelViewpointMetadataRedefinition_Mapping.getMapped(from),
ModelViewpointMetadataFeatureValue_Mapping.getMapped(from)}
```

7.7.9.3.6 ModelViewpointMetadataFeatureTyping_Mapping

Description

The mapping class creates the FeatureTyping relationship for the AnnotatingFeature for the metadata to store the UML4SysML::Model::viewpoint property.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Model

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 SysMLv2::MetadataDefinition.allInstances()
->any(m | m.qualifiedName = 'SysMLv1Library::ModelData')
```

7.7.9.3.7 ModelViewpointMetadataMembership_Mapping

Description

The mapping class creates a membership relationship for the metadata feature value for the UML4SysML::Model::viewpoint property.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Model

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  ModelViewpointMetadataUsage_Mapping.getMapped(from)
  ```

7.7.9.3.8 ModelViewpointMetadataFeatureValue_Mapping

Description

The mapping class maps the value of the property UML4SysML::Model::viewpoint.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Model

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  ModelViewpointValue_Mapping.getMapped(from)
  ```

7.7.9.3.9 ModelViewpointMetadataRedefinition_Mapping

Description

The mapping class creates the redefinition of the attribute for the metadata UML4SysML::Model::viewpoint.

General Mappings

GenericToRedefinition_Mapping

Mapping Source

Model

Mapping Target

Redefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 let m : SYSML2::Membership =
    SYSML2::AttributeUsage.allInstances()
    ->collect(dt | dt.owningRelationship)
    ->select(r | r.oclIsKindOf(SYSML2::Membership))
    ->any(m | m.memberName = 'viewpoint') in
if (m.oclIsUndefined()) then
    OclUndefined
else
    m.memberElement
endif
```

7.7.9.3.10 ModelViewpointValue_Mapping

Description

The mapping class maps the value expression of the property UML4SysML::Model::viewpoint.

General Mappings

GenericToExpression_Mapping

Mapping Source

Model

Mapping Target

LiteralString

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  LiteralString_Factory.create(from.viewpoint)
  ```

7.7.9.3.11 Package_Mapping

Description

A UML4SysML::Package is mapped to a SysML v2 Package. The property "URI" is mapped to a metadata if it has a value. The expected SysML v2 textual notation of a UML4SysML::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

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  Helper.packageOwnedRelationship(from)
  ```

7.7.9.3.12 PackageImport_Mapping

Description

A UML4SysML::PackageImport is mapped to a SysML v2 NamespaceImport. The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
import SysMLv1Package::*;
```

General Mappings

GenericToNamespaceImport_Mapping
ElementMain_Mapping

Mapping Source

PackageImport

Mapping Target

NamespaceImport

Owned Mappings

(none)

Applicable filters

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

```
if src.oclIsKindOf(UML::PackageImport) then
    Helper.isInScope(src.oclAsType(UML::PackageImport).importedPackage)
else
    false
endif
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- NamespaceImport::visibility () : VisibilityKind [0..1]

  ```
  Helper.getKerMLVisibilityKind(from.visibility)
  ```

- NamespaceImport::importedNamespace () : Namespace [1]

  ```
  Namespace_Mapping.getMapped(from.importedPackage)
  ```

7.7.9.3.13 PackageURIMetadataUsage_Mapping

Description

The mapping class creates the annotating feature to annotate the generated Package element with metadata to store the UML4SysML::Package::URI property.

General Mappings

GenericToMetadataUsage_Mapping

Mapping Source

Package

Mapping Target

MetadataUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{PackageURIFeatureTyping_Mapping.getMapped(from),
PackageURIFeatureMembership_Mapping.getMapped(from)}
```

- MetadataUsage::declaredName () : String [0..1]

```
 'URI'
```

7.7.9.3.14 PackageURIFeatureMembership_Mapping

Description

The mapping class creates the feature membership relationship for the metadata feature to store the UML4SysML::Package::URI property.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Package

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 PackageURIMetadataReferenceUsage_Mapping.getMapped(from)
```

7.7.9.3.15 PackageURIFeatureTyping_Mapping

Description

The mapping class creates the FeatureTyping relationship for the AnnotatingFeature for the metadata to store the UML4SysML::Package::URI property.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Package

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific 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
```

7.7.9.3.16 PackageURIMetadataReferenceUsage_Mapping

Description

The mapping class creates the MetadataFeature for the mapping of the property UML4SysML::Package::URI.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Package

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{PackageURIRedefinition_Mapping.getMapped(from),
PackageURIMetadataFeatureValue_Mapping.getMapped(from)}
```

7.7.9.3.17 PackageURIMetadataFeatureValue_Mapping

Description

The mapping class maps the value of the property UML4SysML::Package::URI.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Package

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureValue::featureWithValue () : Feature [1]

  ```
  packageURIMetadataReferenceUsage.to
  ```

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

  ```
  PackageURIValue_Mapping.getMapped(from)
  ```

7.7.9.3.18 PackageURIMetadataMembership_Mapping

Description

The mapping class creates a membership relationship for the metadata feature value for the UML4SysML::Package::URI property.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Package

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  PackageURIMetadataUsage_Mapping.getMapped(from)
  ```

7.7.9.3.19 PackageURIRedefinition_Mapping

Description

The mapping class creates the redefinition of the attribute for the metadata UML4SysML::Package::URI.

General Mappings

GenericToRedefinition_Mapping

Mapping Source

Package

Mapping Target

Redefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 let m : SysMLv2::Membership =
    SysMLv2::AttributeUsage.allInstances()
    ->collect(dt | dt.owningRelationship)
    ->select(r | r.oclIsKindOf(SYSML2::Membership))
    ->any(m | m.memberName = 'URI') in
if (m.oclIsUndefined()) then
    invalid
else
    m.memberElement
endif
```

7.7.9.3.20 PackageURIValue_Mapping

Description

The mapping class maps the value expression of the property UML4SysML::Package::URI.

General Mappings

GenericToExpression_Mapping

Mapping Source

Package

Mapping Target

LiteralString

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 from.URI
```

7.7.9.3.21 Profile_Mapping

Description

A UML4SysML::Profile is mapped to a SysML v2 Package.

General Mappings

Package_Mapping

Mapping Source

Profile

Mapping Target

Package

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Helper.packageOwnedRelationship(from)
->including(ProfileMetadataMembership_Mapping.getMapped(from))
```

7.7.9.3.22 ProfileMetadataMembership_Mapping

Description

The mapping class creates a membership relationship for the metadata feature value for the UML4SysML::Model::viewpoint property.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Profile

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
ProfileMetadataUsage_Mapping.getMapped(from)
```

7.7.9.3.23 ProfileMetadataUsage_Mapping

Description

The mapping class creates the annotating feature to annotate the generated Package element with metadata to store the UML4SysML::Model::viewpoint property.

General Mappings

GenericToMetadataUsage_Mapping

Mapping Source

Profile

Mapping Target

MetadataUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- MetadataUsage::declaredName () : String [0..1]

  ```
  'Profile'
  ```

7.7.9.3.24 StereotypeMetadataDefinition_Mapping

Description

A UML4SysML::Stereotype is mapped to a SysML v2 MetadataDefinition.

General Mappings

Class_Mapping

Mapping Source

Stereotype

Mapping Target

MetadataDefinition

Owned Mappings

(none)

7.7.9.3.25 StereotypeMetadataDefinitionMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

ElementOwningMembership_Mapping

Mapping Source

Stereotype

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 ElementMain_Mapping.getMapped(from)
```

7.7.9.3.26 StereotypeOccurenceUsage_Mapping

Description

The mapping class maps the usage of a stereotype to a SysML v2 OccurrenceUsage.

General Mappings

GenericToOccurrenceUsage_Mapping

Mapping Source

Stereotype

Mapping Target

OccurrenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{StereotypeOccurenceUsageFeatureTyping_Mapping.getMapped(from),
StereotypeOccurenceUsageMultiplicityMembership_Mapping.getMapped(from)}
```

7.7.9.3.27 StereotypeOccurenceUsageFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Stereotype

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
StereotypeOccurenceDefinition_Mapping.getMapped(from)
```

7.7.9.3.28 StereotypeOccurenceUsageMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToMembership_Mapping

Mapping Source

Stereotype

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
StereotypeOccurenceUsage_Mapping.getMapped(from)
```

7.7.9.3.29 StereotypeOccurenceUsageMultiplicityMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToMembership_Mapping

Mapping Source

Stereotype

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
self.ownedMemberElement()
```

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

```
StereotypeOccurenceUsageMultiplicityRange_Mapping.getMapped(from)
```

7.7.9.3.30 StereotypeOccurenceUsageMultiplicityRange_Mapping

Description

The mapping class creates the multiplicity range element for the UML4SysML::Stereotype mapping.

General Mappings

GenericToFeature_Mapping

Mapping Source

Stereotype

Mapping Target

MultiplicityRange

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.9.3.31 StereotypeOccurenceUsageMultiplicityRangeInfinity_Mapping

Description

The mapping class creates the literal infinity element for the multiplicity range element for the UML4SysML::Stereotype mapping.

General Mappings

GenericToExpression_Mapping

Mapping Source

Stereotype

Mapping Target

LiteralInfinity

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.9.3.32 StereotypeOccurenceUsageInfinityReturnParameter_Mapping

Description

The mapping class creates the return parameter relationship for the literal infinity element for the multiplicity range element for the UML4SysML::Stereotype mapping.

General Mappings

GenericToFeature_Mapping

Mapping Source

Stereotype

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
        SysMLv2::FeatureDirectionKind::out
```

7.7.9.3.33 StereotypeOccurenceUsageInfinityReturnParameterMembership_Mapping

Description

General Mappings

GenericToReturnParameterMembership_Mapping

Mapping Source

Stereotype

Mapping Target

ReturnParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 StereotypeOccurenceUsageInfinityReturnParameter_Mapping.getMapped(from)
```

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

```
 self.ownedMemberParameter()
```

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

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

7.7.9.3.34 StereotypeOccurenceUsageMultiplicityRangeMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToMembership_Mapping

Mapping Source

Stereotype

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  StereotypeOccurenceUsageMultiplicityRangeInfinity_Mapping.getMapped(from)
  ```

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

  ```
  self.ownedMemberElement()
  ```

7.7.10 SimpleClassifiers

7.7.10.1 Overview

The following table gives an overview of which SysML v2 elements the UML4SysML::SimpleClassifiers elements are transformed with which mapping class. The mapping details are in 7.7.10.2.

Table 16. List of all mappings

SysML v1 Concept	SysML v2 Concept
DataType	EnumerationDefinition AttributeDefinition AttributeDefinition
Enumeration	EnumerationDefinition
EnumerationLiteral	EnumerationUsage
Interface	PortDefinition
InterfaceRealization	
PrimitiveType	AttributeDefinition
Reception	ItemUsage
Signal	ItemDefinition

7.7.10.2 Mapping Specifications

7.7.10.2.1 Attribute_Mapping

Description

An UML4SysML::Property is mapped to a SysMLv2 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) and not
    Helper.hasStereotypeApplied(src.owner,
        'SysML::ConstraintBlocks::ConstraintBlock') 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.

7.7.10.2.2 AttributeRedefined_Mapping

Description

An UML4SysML::SimpleClassifiers::Property is mapped to a SysML v2 AttributeUsage.

General Mappings

PropertyCommon_Mapping

Mapping Source

Property

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
let typing: KerML::FeatureTyping =
    AssociationToFeatureTyping_Mapping.getMapped(from) in
let subsetting: Set(KerML::Subsetting) =
    from.subsettedProperty
    ->collect(p | PropertySubsetting_Mapping.getMapped(from, p))->asSet() in
let subsettingMultiplicityTyping: Set(KerML::Relationship) =
    subsetting
    ->union(Set{AttributeRedefinedRedefinition_Mapping.getMapped(from)})->union(
        if typing.oclIsUndefined() then
            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
```

7.7.10.2.3 AttributeRedefinedRedefinition_Mapping

Description

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

General Mappings

GenericToRedefinition_Mapping

Mapping Source

Property

Mapping Target

Redefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
from.redefinedProperty.get(0)
```

7.7.10.2.4 AttributeRedefinedMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

ElementFeatureMembership_Mapping

Mapping Source

Element

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

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

```
src.oclIsKindOf(UML::Property)
and (src.oclAsType(UML::Property).redefinedElement->size() > 0)
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
AttributeRedefined_Mapping.getMapped(from)
```

7.7.10.2.5 AttributeRedefinedFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

StructuralFeatureToFeatureTyping_Mapping

Mapping Source

StructuralFeature

Mapping Target

FeatureTyping

Owned Mappings

(none)

7.7.10.2.6 BehavioredClassifier_Mapping

Description

The abstract mapping class maps the abstract metaclass UML4SysML::BehavioredClassifiers to a SysMLv2 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

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
let toElementFMS: Set(UML::Element) =
    from.ownedElement->select(e | (e.oclIsKindOf(UML::Property) and
        (e.oclAsType(UML::Property).redefinedProperty->size() = 0)) or
            e.oclIsKindOf(UML::Operation)) in
let redefinedAttributes: Set(UML::Element) =
    from.ownedElement->select(e | from.oclIsKindOf(UML::DataType) and
        (e.oclAsType(UML::Property).redefinedProperty->size() > 0)) in
let generalizations : Set(UML::Generalization) =
    from.ownedElement
    ->select(e | e.oclIsKindOf(UML::Generalization)) in
let constraints : Set(UML::Constraint) =
    UML::Constraint.allInstances()
    ->select( c | c.constrainedElement->includes(from)) in
let toElementOMS: Set(UML::Element) =
    (((from.ownedElement - toElementFMS) - redefinedAttributes) -
    generalizations)  in
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(BehavioredClassifierFeatureMembership_Mapping.getMapped(from))
endif
```

7.7.10.2.7 BehavioredClassifierFeatureMembership_Mapping

Description

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

BehavioredClassifier

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
BehavioredClassifierActionUsage_Mapping.getMapped(from)
```

7.7.10.2.8 BehavioredClassifierFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

BehavioredClassifier

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
from
```

7.7.10.2.9 BehavioredClassifierActionUsage_Mapping

Description

The BehavioredClassifierToPerformActionUsage_Mapping class creates a PerformActionUsage element to call the transformed SysML v1 classifier behavior.

General Mappings

GenericToActionUsage_Mapping

Mapping Source

BehavioredClassifier

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

- ActionUsage::declaredName () : String [0..1]

  ```
  'classifierBehavior'
  ```

7.7.10.2.10 DataType_Mapping

Description

A UML4SysML::SimpleClassifiers::DataType is mapped to a SysML v2 AttributeDefinition. The mapping also cover the transformation of UML4SysML::PrimitiveType elements.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
part def SysMLv1Block {
        attribute sysMLv1Property : ScalarValues::Integer;
}
```

General Mappings

Classifier_Mapping

Mapping Source

DataType

Mapping Target

AttributeDefinition

Owned Mappings

(none)

7.7.10.2.11 Enumeration_Mapping

Description

A UML4SysML::Enumeration is mapped to a SysML v2 EnumerationDefinition.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
enum def SysMLv1Enumeration {
        enum sysMLv1Literal1;
        enum sysMLv1Literal2;
}
```

General Mappings

DataType_Mapping

Mapping Source

Enumeration

Mapping Target

EnumerationDefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific 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(UML::Generalization)) in
  let toElementFMS: Set(UML::Element) =
      from.ownedElement->select(e | e.oclIsKindOf(UML::Property)) in
  let literals: Set(UML::Element) =
      from.ownedElement->select(e | e.oclIsKindOf(UML::EnumerationLiteral)) in
  let toElementOMS: Set(UML::Element) =
      (((from.ownedElement - toElementFMS) - generalizations) - literals)  in
  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)))
  ```

7.7.10.2.12 EnumerationLiteral_Mapping

Description

A UML4SysML::EnumerationLiteral is mapped to a SysML v2 EnumerationUsage.

General Mappings

GenericToFeature_Mapping
InstanceSpecification_Mapping

Mapping Source

EnumerationLiteral

Mapping Target

EnumerationUsage

Owned Mappings

(none)

7.7.10.2.13 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

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
from
```

7.7.10.2.14 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.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
port def SysMLv1Interface {
        attribute sysMLv1Property;
}
```

General Mappings

GenericToPortDefinition_Mapping
Classifier_Mapping

Mapping Source

Interface

Mapping Target

PortDefinition

Owned Mappings

- conjugatedPortDefinitionMembership : InterfaceConjugatedPortDefinitionMembership_Mapping

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
let properties: Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Property)) in
let generalizations : Set(UML::Generalization) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Generalization)) in
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)
```

7.7.10.2.15 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

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ConjugatedPortDefinition::declaredName () : String [0..1]

  ```
  '~'+from.name
  ```

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

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

7.7.10.2.16 InterfaceConjugatedPortDefinitionMembership_Mapping

Description

As part of the mapping from a UML4SysML::Interface to a SysML v2 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

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
InterfaceConjugatedPortDefinition_Mapping.getMapped(from)
```

7.7.10.2.17 InterfacePortConjugation_Mapping

Description

As part of the mapping from a UML4SysML::Interface to a SysML v2 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

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
from
```

- PortConjugation::conjugatedType () : Type [1]

```
 SysMLv2::ConjugatedPortDefinition.allInstances()
->collect(cpd | cpd.owningRelationship)
->select(r | r.oclIsKindOf(SysMLv2::Membership))
->any(m | m.memberName = from.name)
```

7.7.10.2.18 InterfaceRealization_Mapping

Description

A UML4SysML::InterfaceRealization is mapped to a SysMLv2 Subclassification relationship.

General Mappings

GenericToSpecialization_Mapping

Mapping Source

InterfaceRealization

Mapping Target

Subclassification

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific 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)
  ```

7.7.10.2.19 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)

7.7.10.2.20 Reception_Mapping

Description

A UML4SysML::Reception is mapped to a SysML v2 AttributeUsage with feature direction "in".

General Mappings

BehavioralFeature_Mapping

Mapping Source

Reception

Mapping Target

ItemUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

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

  ```
  SysMLv2::FeatureDirectionKind::in
  ```

7.7.10.2.21 ReceptionFeatureTyping_Mapping

Description

A UML4SysML::Reception is mapped to SysML v2 AttributeUsage. The ReceptionToFeatureTyping_Mapping class creates the type of the AttributeUsage which is the Signal of the Reception.

General Mappings

TypedElementFeatureTyping_Mapping

Mapping Source

Reception

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
Classifier_Mapping.getMapped(from.signal)
```

7.7.10.2.22 Signal_Mapping

Description

A UML4SysML::Signal is mapped to a SysML v2 AttributeDefinition.

General Mappings

Classifier_Mapping

Mapping Source

Signal

Mapping Target

ItemDefinition

Owned Mappings

(none)

7.7.11 StateMachines

7.7.11.1 Overview

The following table gives an overview of which SysML v2 elements the UML4SysML::StateMachines elements are transformed with which mapping class. The mapping details are in 7.7.11.3.

The justifications for the elements without mapping are given in 7.7.11.2.

Table 17. List of all mappings

SysML v1 Concept	SysML v2 Concept
ConnectionPointReference	StateUsage
FinalState	StateUsage
Pseudostate	StateUsage
Region	StateUsage
State	StateUsage StateUsage
StateMachine	StateDefinition
Transition	TransitionUsage

7.7.11.2 UML4SysML::StateMachines elements not mapped

7.7.11.3 Mapping Specifications

7.7.11.3.1 ConnectionPointReference_Mapping

Description

A UML4SysML::ConnectionPointReference element is mapped to a SysML v2 StateUsage.

General Mappings

Namespace_Mapping
GenericToStateUsage_Mapping

Mapping Source

ConnectionPointReference

Mapping Target

StateUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
false
```

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

```
let toFeatureMS : Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Region)) in
let toElementOMS : Set(UML::Element) =
    from.ownedElement - toFeatureMS in
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toFeatureMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
```

7.7.11.3.2 FinalState_Mapping

Description

A UML4SysML::FinalState is mapped to a SysML v2 StateUsage. The details of the mapping are not defined yet.

General Mappings

State_Mapping

Mapping Source

FinalState

Mapping Target

StateUsage

Owned Mappings

(none)

Applicable filters

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

```
src.oclIsTypeOf(UML::FinalState)
```

Mapping rules

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

7.7.11.3.3 PseudoState_Mapping

Description

A UML4SysML::PseudoState is mapped to a SysML v2 StateUsage.

General Mappings

Namespace_Mapping
GenericToStateUsage_Mapping

Mapping Source

Pseudostate

Mapping Target

StateUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
let toFeatureMS : Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Region)) in
let toElementOMS : Set(UML::Element) =
    from.ownedElement - toFeatureMS in
toElementOMS
->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toFeatureMS
->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
```

7.7.11.3.4 Region_Mapping

Description

A UML4SysML::Region is mapped to SysML v2 StateUsage.

General Mappings

Namespace_Mapping
GenericToStateUsage_Mapping

Mapping Source

Region

Mapping Target

StateUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
let toFeatureMS : Set(UML::Element) =
    from.ownedElement
    ->select(e | e.oclIsKindOf(UML::State) or e.oclIsKindOf(UML::Transition)) in
let toElementOMS : Set(UML::Element) =
    from.ownedElement - toFeatureMS in
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toFeatureMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
```

7.7.11.3.5 State_Mapping

Description

A UML4SysML::State is mapped to a SysML v2 StateUsage.

General Mappings

Namespace_Mapping
GenericToStateUsage_Mapping

Mapping Source

State

Mapping Target

StateUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
let toFeatureMS : Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Region)) in
let toElementOMS : Set(UML::Element) =
    from.ownedElement - toFeatureMS in
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toFeatureMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
```

7.7.11.3.6 StateDefinition_Mapping

Description

A UML4SysML::StateMachine is mapped to a SysML v2 StateDefinition.

General Mappings

Behavior_Mapping

Mapping Source

StateMachine

Mapping Target

StateDefinition

Owned Mappings

(none)

Applicable filters

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

```
src.owner.oclIsKindOf(UML::Package)
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- StateDefinition::isParallel () : Boolean [1]

  ```
  from.region->size() > 1
  ```

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

  ```
  let initialState : Set(UML::Element) =
      from.ownedElement
      ->select(e | e.oclIsKindOf(UML::Pseudostate) and
      e.oclAsType(UML::Pseudostate).kind = UML::PseudostateKind::initial) in
  let toParameterMS : Set(UML::Element) =
      from.ownedElement->select(e | e.oclIsKindOf(UML::Parameter)) in
  let parameterSets: Set(UML::Element) =
      from.ownedElement->select(e | e.oclIsKindOf(UML::ParameterSet)) in
  let toFeatureMS : Set(UML::Element) =
      from.ownedElement->select(e | e.oclIsKindOf(UML::Region)) in
  let toElementOMS : Set(UML::Element) =
      ((from.ownedElement - toFeatureMS) - toParameterMS) - initialState in
  toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
  ->union(toFeatureMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
  ->union(toParameterMS->collect(e | ParameterMembership_Mapping.getMapped(e)))
  ->union(parameterSets->collect(e | ParameterSetMembership_Mapping.getMapped(e)))
  ->union(initialState->collect(e | InitialStateMembership_Mapping.getMapped(e)))
  ```

7.7.11.3.7 Transition_Mapping

Description

A UML4SysML::Transition is mapped to a SysML v2 TransitionUsage.

General Mappings

Namespace_Mapping

Mapping Source

Transition

Mapping Target

TransitionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 from.ownedElement->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->including(TransitionSuccession_Mapping.getMapped(from))
```

- TransitionUsage::target () : ActionUsage [1]

```
from.target
```

- TransitionUsage::source () : ActionUsage [1]

```
from.source
```

7.7.11.3.8 TransitionSuccession_Mapping

Description

The mapping class creates the source Feature element of the Succession that is part of the TransitionUsage that is the target element of the UML4SysML::Transition mapping.

General Mappings

GenericToConnector_Mapping
GenericToMembership_Mapping

Mapping Source

Transition

Mapping Target

Succession

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 OrderedSet{TransitionSuccessionSourceMembership_Mapping.getMapped(from),
TransitionSuccessionTargetMembership_Mapping.getMapped(from)}
```

7.7.11.3.9 TransitionSourceToSubsetting_Mapping

Description

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

General Mappings

GenericToSubsetting_Mapping

Mapping Source

Transition

Mapping Target

Subsetting

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
ElementMain_Mapping.getMapped(from.source)
```

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

```
TransitionSuccessionSource_Mapping.getMapped(from)
```

7.7.11.3.10 TransitionSuccessionSource_Mapping

Description

The mapping class creates the Succession element that is part of the TransitionUsage that is the target element of the UML4SysML::Transition mapping.

General Mappings

GenericToFeature_Mapping

Mapping Source

Transition

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Feature::declaredName () : String [0..1]

```
'source'
```

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

```
true
```

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

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

7.7.11.3.11 TransitionSuccessionSourceMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToEndFeatureMembership_Mapping

Mapping Source

Transition

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
TransitionSuccessionSource_Mapping.getMapped(from)
```

7.7.11.3.12 TransitionSuccessionTarget_Mapping

Description

The mapping class creates the target Feature element of the Succession that is part of the TransitionUsage that is the target element of the UML4SysML::Transition mapping.

General Mappings

GenericToFeature_Mapping

Mapping Source

Transition

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

- Feature::declaredName () : String [0..1]

  ```
  'target'
  ```

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

  ```
  true
  ```

7.7.11.3.13 TransitionSuccessionTargetMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToEndFeatureMembership_Mapping

Mapping Source

Transition

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  TransitionSuccessionTarget_Mapping.getMapped(from)
  ```

7.7.11.3.14 TransitionTargetToSubsetting_Mapping

Description

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

General Mappings

GenericToSubsetting_Mapping

Mapping Source

Transition

Mapping Target

Subsetting

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  TransitionSuccessionTarget_Mapping.getMapped(from)
  ```

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

  ```
  ElementMain_Mapping.getMapped(from.target)
  ```

7.7.12 StructuredClassifiers

7.7.12.1 Overview

The following table gives an overview of which SysML v2 elements the UML4SysML::StructuredClassifiers elements are transformed with which mapping class. The mapping details are in 7.7.12.2.

Table 19. List of all mappings

SysML v1 Concept	SysML v2 Concept
Association	ConnectionDefinition ConnectionDefinition ConnectionDefinition ConnectionDefinition
AssociationClass	ConnectionDefinition ConnectionDefinition

SysML v1 Concept	SysML v2 Concept
Class	PartDefinition OccurrenceDefinition ItemDefinition VerificationCaseDefinition ActionUsage PartDefinition ViewDefinition ActionDefinition RequirementUsage PortDefinition ConstraintDefinition ActionDefinition MetadataDefinition ActionUsage StateDefinition
Connector	ConnectionUsage BindingConnectorAsUsage
ConnectorEnd	Feature
Port	PortUsage PortUsage PartUsage PartUsage

7.7.12.2 Mapping Specifications

7.7.12.2.1 AssociationClass_Mapping

Description

A UML4SysML::AssociationClass is mapped to a SysML v2 ConnectionDefinition. The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
part def SysMLv1Block1;
part def SysMLv1Block2;
connection def SysMLv1AssociationBlock {
        end : SysMLv1Block1;
        end : SysMLv1Block2;
}
```

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(src, 'SysML::Blocks::Block')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
let nonOwnedEnds: OrderedSet(UML::Property) =
    (from.memberEnd-from.ownedEnd)->asOrderedSet() in
let generalizations : Set(UML::Generalization) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Generalization)) in
let others: OrderedSet(UML::Element) =
    ((from.ownedElement-from.memberEnd)-generalizations)->asOrderedSet() in
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()
```

7.7.12.2.2 AssociationCommon_Mapping

Description

A UML4SysML::Association is mapped to a SysML v2 ConnectionDefinition. This is the abstract base class of all concrete association mapping classes.

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:

```
src.memberEnd->select( m | m.type.oclIsKindOf(UML::UseCase))->isEmpty()
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
let nonOwnedEnds: OrderedSet(UML::Property) =
    (from.memberEnd-from.ownedEnd)->asOrderedSet() in
let generalizations : Set(UML::Generalization) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Generalization)) in
let others: OrderedSet(UML::Element) =
    ((from.ownedElement-from.memberEnd)-generalizations)->asOrderedSet() in
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()
```

7.7.12.2.3 AssociationMetadataUsage_Mapping

Description

The mapping class creates the MetadataUsage element to annotate a ConnectionDefinition that its mapping source element is a derived association.

General Mappings

GenericToMetadataUsage_Mapping

Mapping Source

Association

Mapping Target

MetadataUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{AssociationToFeatureTyping_Mapping.getMapped(from),
AssociationMetadataUsageFeatureMembership_Mapping.getMapped(from)}
```

7.7.12.2.4 AssociationMetadataUsageFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Association

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
AssociationMetadataUsageFeature_Mapping.getMapped(from)
```

7.7.12.2.5 AssociationMetadataUsageFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Association

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.12.2.6 AssociationMetadataUsageFeature_Mapping

Description

The mapping class creates the feature of the MetadataUsage.

General Mappings

GenericToFeature_Mapping

Mapping Source

Association

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{AssociationMetadataUsageRedefinition_Mapping.getMapped(from),
AssociationMetadataUsageFeatureValue_Mapping.getMapped(from)}
```

7.7.12.2.7 AssociationMetadataUsageFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Association

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
LiteralBoolean_Factory.create(from.isDerived)
```

7.7.12.2.8 AssociationMetadataUsageMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Association

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 AssociationMetadataUsage_Mapping.getMapped(from)
```

7.7.12.2.9 AssociationMetadataUsageRedefinition_Mapping

Description

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

General Mappings

GenericToRedefinition_Mapping

Mapping Source

Association

Mapping Target

Redefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.12.2.10 Class_Mapping

Description

A UML4SysML::Class is mapped to a SysML v2 OccurrenceDefinition. The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
occurrence def UML4SysMLClass;
```

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.isRequirement(src) and not src.oclIsTypeOf(UML::AssociationClass)
```

Mapping rules

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

7.7.12.2.11 ConnectionEndToSubsetting_Mapping

Description

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

General Mappings

GenericToSubsetting_Mapping

Mapping Source

ConnectorEnd

Mapping Target

Subsetting

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 let propertyPath: OrderedSet(UML::Property) =
    Helper.getTagValueAsElementColl
        (from, 'SysML::Blocks::NestedConnectorEnd','propertyPath')
    ->asOrderedSet() in
if propertyPath->notEmpty() then
        OrderedSet{ConnectorEndToSubsettedFeatureMembership_Mapping.getMapped(from)}
else
        OrderedSet{}
endif
```

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

```
 let propertyPath: OrderedSet(UML::Property) =
    Helper.getTagValueAsElementColl
    (src, 'SysML::Blocks::NestedConnectorEnd','propertyPath')
    ->asOrderedSet() in
if propertyPath->isEmpty() then
    ElementMain_Mapping.getMapped(from.role)
else
    ConnectorEndToSubsettedFeature_Mapping.getMapped(from)
endif
```

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

```
 ConnectorEndToOwnedFeature_Mapping.getMapped(from)
```

7.7.12.2.12 Connector_Mapping

Description

A UML4SysML::Connector is mapped to a SysMLv2 ConnectionUsage. The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
part def SysMLv1Block3 {
        part sysMLv1PartProperty1 : SysMLv1Block1;
        part sysMLv1PartProperty2 : SysMLv1Block2;
        connection sysMLv1Connector connect sysMLv1PartProperty1 to sysMLv1PartProperty2;
}
part def SysMLv1Block1;
part def SysMLv1Block2;
```

General Mappings

NamedElementMain_Mapping
GenericToConnector_Mapping

Mapping Source

Connector

Mapping Target

ConnectionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
from.end->collect(e | ConnectorEndToMembership_Mapping.getMapped(e))
   ->including(ConnectorMultiplicityMembership_Mapping.getMapped(from))
```

7.7.12.2.13 ConnectorEndToFeatureCommon_Mapping

Description

The mapping class is the abstract base class for UML4SysML::ConnectorEnd mapping classes.

General Mappings

GenericToFeature_Mapping

Mapping Source

ConnectorEnd

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
from.isOrdered
```

7.7.12.2.14 ConnectorEndToMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ConnectorEnd

Mapping Target

EndFeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
ConnectorEndToOwnedFeature_Mapping.getMapped(from)
```

7.7.12.2.15 ConnectorEndToOwnedFeature_Mapping

Description

The mapping class creates the SysML v2 Feature element for the UML4SysML::ConnectorEnd mapping.

General Mappings

ConnectorEndToFeatureCommon_Mapping
ElementMain_Mapping

Mapping Source

ConnectorEnd

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific 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
```

7.7.12.2.16 ConnectorEndToSubsettedFeature_Mapping

Description

The mapping class maps UML4SysML::ConnectorEnd that are part of a SysML::Ports&Flows::NestedConnectorEnd.

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::NestedConnectorEnd','propertyPath')
->asOrderedSet() in
propertyPath->notEmpty()
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 let propertyPath: OrderedSet(UML::Property) =
    Helper.getTagValueAsElementColl
    (from, 'SysML::Blocks::NestedConnectorEnd','propertyPath')
    ->asOrderedSet() in
let chain: OrderedSet(KerML::FeatureChaining) =
    propertyPath->collect(p | PropertyToFeatureChaining_Mapping.getMapped(p))
    ->asOrderedSet()
    ->including(PropertyToFeatureChaining_Mapping.getMapped(from.role)) in
chain->union(OrderedSet{MultiplicityMembership_Mapping.getMapped(from)})
```

- Feature::declaredName () : String [0..1]

```
'featureChain'
```

7.7.12.2.17 ConnectorEndToSubsettedFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

ConnectorEnd

Mapping Target

EndFeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
ConnectorEndToSubsettedFeature_Mapping.getMapped(from)
```

7.7.12.2.18 ConnectorMultiplicityMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

DefaultMultiplicityMembership_Mapping

Mapping Source

Connector

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  from.name+'_Connector_multiplicity'
  ```

7.7.12.2.19 ConnectorType_Mapping

Description

A UML4SysML::Association is mapped to a SysML v2 ConnectionDefinition.

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 src.memberEnd->exists( m | m.type.oclIsKindOf(UML::UseCase)) and
    not src.isDerived and
    not src.oclIsTypeOf(UML::AssociationClass) and
    Helper.isConnectionDef(src)
endif
```

Mapping rules

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

7.7.12.2.20 ConnectorTypeDerived_Mapping

Description

The mapping class is a concrete mapping class of the abstract AssociationCommon_Mapping class for mappings of derived associations. The UML4SysML::Association::isDerived property is not supported in SysML v2. To preserve the information, it is stored in a metadata annotation.

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:

```
(src.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

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
let nonOwnedEnds: OrderedSet(UML::Property) =
    (from.memberEnd-from.ownedEnd)->asOrderedSet() in
let generalizations : Set(UML::Generalization) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Generalization)) in
let others: OrderedSet(UML::Element) =
    ((from.ownedElement-from.memberEnd)-generalizations)->asOrderedSet() in
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()
->including(AssociationMetadataUsageMembership_Mapping.getMapped(from))
```

7.7.12.2.21 End_Mapping

Description

The mapping class is the abstract base class of mapping classes for properties that are defined by association ends.

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

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
true
```

7.7.12.2.22 EndMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

StructuralFeatureMembership_Mapping

Mapping Source

Property

Mapping Target

EndFeatureMembership

Owned Mappings

(none)

7.7.12.2.23 EndToSubsettedFeature_Mapping

Description

The mapping class creates a feature element for the UML4SysML::ConnectorEnd mapping.

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

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 let chain: OrderedSet(KerML::FeatureChaining) =
    OrderedSet{EndToSubsettedFeatureChaining_Mapping.getMapped(from)} in
chain->including(MultiplicityMembership_Mapping.getMapped(from))
```

7.7.12.2.24 EndToSubsettedFeatureChaining_Mapping

Description

The mapping class creates a feature chaining element for the UML4SysML::ConnectorEnd mapping.

General Mappings

GenericToRelationship_Mapping

Mapping Source

Property

Mapping Target

FeatureChaining

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureChaining::declaredName () : String [0..1]

```
'featureChain'
```

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

```
Property_Mapping.getMapped(from)
```

7.7.12.2.25 NonOwnedEndSubsetting_Mapping

Description

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

General Mappings

GenericToSubsetting_Mapping

Mapping Source

Property

Mapping Target

Subsetting

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  Property_Mapping.getMapped(from)
  ```

7.7.12.2.26 NonOwnedEndToSubsettedFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

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

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 EndToSubsettedFeature_Mapping.getMapped(from)
```

7.7.12.2.27 NonOwnedEnd_Mapping

Description

The mapping class maps UML4SysML::Property elements that are not owned by an association to a SysML v2 Feature element.

General Mappings

End_Mapping

Mapping Source

Property

Mapping Target

Feature

Owned Mappings

- nonOwnedEndTyping : NonOwnedEndFeatureTyping_Mapping

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Feature::declaredName () : String [0..1]

```
 'nonOwnedEnd'
```

- 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())
```

7.7.12.2.28 NonOwnedEndMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

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

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  NonOwnedEnd_Mapping.getMapped(from)
  ```

7.7.12.2.29 NonOwnedEndSubsettingMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Property

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
NonOwnedEndSubsetting_Mapping.getMapped(from)
```

7.7.12.2.30 NonOwnedEndFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

StructuralFeatureToFeatureTyping_Mapping

Mapping Source

Property

Mapping Target

FeatureTyping

Owned Mappings

- nonOwnedEnd : NonOwnedEnd_Mapping

7.7.12.2.31 OwnedEnd_Mapping

Description

The mapping class maps UML4SysML::Property elements that are owned by an association to a SysML v2 Feature element.

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::UseCase))->notEmpty())
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
let qualifiers: Set(KerML::FeatureMembership) =
    from.qualifier
    ->collect(q | ElementFeatureMembership_Mapping.getMapped(q))->asSet() in
let typing: KerML::FeatureTyping =
    StructuralFeatureToFeatureTyping_Mapping.getMapped(from) in
let subsetting: Set(KerML::Subsetting) =
    from.subsettedProperty
    ->collect(p | PropertySubsetting_Mapping.getMapped(from, p))->asSet() in
let subsettingMultiplicityTyping: Set(KerML::Relationship) =
    subsetting->union(if typing.oclIsUndefined() then
                        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(from.defaultValue))
    else
        subsettingMultiplicityTyping
    endif) in

if from.defaultValue.oclIsUndefined() then
    relationships
else
    relationships->including(
        if from.defaultValue.oclIsTypeOf(UML::OpaqueExpression) then
            DefaultValueOpaqueExpression_Mapping.getMapped(from.defaultValue)
        else
            DefaultValue_Mapping.getMapped(from.defaultValue)
        endif)
endif
```

7.7.12.2.32 OwnedEndMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

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

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  OwnedEnd_Mapping.getMapped(from)
  ```

7.7.12.2.33 Port_Mapping

Description

A UML4SysML::Port that is typed by an interface block is mapped to a SysML v2 PortUsage.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
port sysMLv1Port : SysMLv1InterfaceBlock;
port def SysMLv1InterfaceBlock
```

General Mappings

Property_Mapping

Mapping Source

Port

Mapping Target

PortUsage

Owned Mappings

(none)

7.7.12.2.34 PortUntyped_Mapping

Description

A UML4SysML::Port that is untyped is mapped to a SysML v2 PortUsage.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
port sysMLv1Port;
```

General Mappings

PropertyUntyped_Mapping

Mapping Source

Port

Mapping Target

PortUsage

Owned Mappings

(none)

7.7.12.2.35 PropertyToFeatureChaining_Mapping

Description

The mapping class creates the SysML v2 FeatureChaining for the UML4SysML::Property mapping.

General Mappings

GenericToRelationship_Mapping

Mapping Source

Property

Mapping Target

FeatureChaining

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
ElementMain_Mapping.getMapped(from)
```

7.7.12.2.36 QualifierMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

StructuralFeatureMembership_Mapping

Mapping Source

StructuralFeature

Mapping Target

FeatureMembership

Owned Mappings

(none)

7.7.13 UseCases

7.7.13.1 Overview

The following table gives an overview of which SysML v2 elements the UML4SysML::UseCases elements are transformed with which mapping class. The mapping details are in 7.7.13.3.

The justifications for the elements without mapping are given in 7.7.13.2.

Table 20. List of all mappings

SysML v1 Concept	SysML v2 Concept
Actor	ItemDefinition
Extend	
ExtensionPoint	
Include	IncludeUseCaseUsage
UseCase	UseCaseDefinition

7.7.13.2 UML4SysML::UseCases elements not mapped

Table 21. 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.

7.7.13.3 Mapping Specifications

7.7.13.3.1 Actor_Mapping

Description

A UML4SysML::Actor is mapped to a SysML v2 ItemDefinition. The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
item def SysMLv1Actor;
```

General Mappings

ElementMain_Mapping
BehavioredClassifier_Mapping

Mapping Source

Actor

Mapping Target

ItemDefinition

Owned Mappings

(none)

7.7.13.3.2 Include_Mapping

Description

A UML4SysML::Include is mapped to a SysML v2 IncludeUseCaseUsage. The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
use case def SysMLv1UseCase1 {
        include use case : SysMLv1UseCase2;
}
use case def SysMLv1UseCase2;
```

General Mappings

GenericToOccurrenceUsage_Mapping
NamedElementMain_Mapping

Mapping Source

Include

Mapping Target

IncludeUseCaseUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{IncludeFeatureTyping_Mapping.getMapped(from),
ReturnParameterFeatureMembership_Factory.create(),
EmptySubjectMembership_Factory.create()}
```

7.7.13.3.3 IncludeFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Include

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
from.addition
```

7.7.13.3.4 UseCase_Mapping

Description

A UML4SysML::UseCase is mapped to a SysML v2 UseCaseDefinition. The expected SysML v2 textual syntax of a mapped UML4SysML::UseCase with a defined subject is as follows.

```
use case def SysMLv1UseCase {
  subject subject_SysMLv1Block : SysMLv1Block;
}
part def SysMLv1Block;
```

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

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
let properties : Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Property) and
        e.oclAsType(UML::Property).association.oclIsUndefined()) in
let actors : Set(UML::Property) =
    UML::Association.allInstances()
        ->collect(m | m.memberEnd)
        ->flatten()
        ->select( m | m.type = from)->collect(a | a.owningAssociation)
        ->collect( p | p.memberEnd->select( m | not (m.type = from) ))->flatten() in
let extensionPoints : Sequence(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::ExtensionPoint)) in
let extend : Sequence(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Extend)) in
let include : Sequence(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Include)) in
let elements : Set(UML::Element) =
    ((((from.ownedElement-properties) - extensionPoints) - extend) - include) in
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(ClassifierBehaviorFeatureMembership_Mapping.getMapped(from))
endif
```

7.7.13.3.5 UseCaseActor_Mapping

Description

The mapping class creates the PartUsage representing an actor of the use case.

General Mappings

GenericToPartUsage_Mapping

Mapping Source

Property

Mapping Target

PartUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- PartUsage::declaredName () : String [0..1]

```
from.name
```

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

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

7.7.13.3.6 UseCaseActorFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Property

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
from.type
```

7.7.13.3.7 UseCaseActorMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToActorMembership_Mapping

Mapping Source

Property

Mapping Target

ActorMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
UseCaseActor_Mapping.getMapped(from)
```

7.7.13.3.8 UseCaseEmptySubjectReferenceUsage_Mapping

Description

The mapping class creates an "empty" ReferenceUsage for the subject, if the subject is not given at the SysML v1 UseCase element.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

UseCase

Mapping Target

ReferenceUsage

Owned Mappings

(none)

7.7.13.3.9 UseCaseObjectiveMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToObjectiveMembership_Mapping

Mapping Source

UseCase

Mapping Target

ObjectiveMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
UseCaseObjectiveRequirementUsage_Mapping.getMapped(from)
```

7.7.13.3.10 UseCaseObjectiveRequirementUsage_Mapping

Description

The mapping class creates the RequirementUsage element for the use case objective. The element is not set by an element from the SysML v1 UseCase.

General Mappings

GenericToRequirementUsage_Mapping

Mapping Source

UseCase

Mapping Target

RequirementUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{UseCaseObjectiveSubjectMembership_Mapping.getMapped(from),
CommonReturnParameterReferenceUsageMembership_Mapping.getMapped(from)}
```

7.7.13.3.11 UseCaseObjectiveSubjectMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToSubjectMembership_Mapping

Mapping Source

UseCase

Mapping Target

SubjectMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
UseCaseEmptySubjectReferenceUsage_Mapping.getMapped(from)
```

7.7.13.3.12 UseCaseSubjectFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

UseCase

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
if from.subject->size() > 0 then from.subject->get(0) else OclUndefined endif
```

7.7.13.3.13 UseCaseSubjectMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToSubjectMembership_Mapping

Mapping Source

UseCase

Mapping Target

SubjectMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
if from.subject->size() > 0 then
   UseCaseSubjectReferenceUsage_Mapping.getMapped(from)
```

```
        else
            UseCaseEmptySubjectReferenceUsage_Mapping.getMapped(from)
        endif
```

7.7.13.3.14 UseCaseSubjectReferenceUsage_Mapping

Description

The mapping class creates the ReferenceUsage element for the subject.

General Mappings

UseCaseEmptySubjectReferenceUsage_Mapping

Mapping Source

UseCase

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

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

  ```
  'subject_' + from.subject->get(0).name
  ```

7.7.14 Values

7.7.14.1 Overview

The following table gives an overview of which SysML v2 elements the UML4SysML::Values elements are transformed with which mapping class. The mapping details are in 7.7.14.3.

The justifications for the elements without mapping are given in 7.7.14.2.

Table 22. List of all mappings

SysML v1 Concept	SysML v2 Concept
Duration	

SysML v1 Concept	SysML v2 Concept
DurationConstraint	
DurationInterval	
DurationObservation	
Expression	OperatorExpression OperatorExpression
Interval	
IntervalConstraint	
LiteralBoolean	LiteralBoolean
LiteralInteger	LiteralInteger
LiteralNull	NullExpression
LiteralReal	LiteralRational
LiteralString	LiteralString
LiteralUnlimitedNatural	LiteralInfinity LiteralInteger
OpaqueExpression	CalculationUsage
StringExpression	
TimeConstraint	
TimeExpression	TriggerInvocationExpression
TimeInterval	
TimeObservation	

7.7.14.2 UML4SysML::Values elements not mapped

Table 23. List of SysML v1 elements not mapped of this section

SysML v1 Concept	Rationale
Duration	Mapping is not specified yet.
DurationConstraint	Mapping is not specified yet.
DurationInterval	Mapping is not specified yet.
DurationObservation	Mapping is not specified yet.
Interval	Mapping is not specified yet.
IntervalConstraint	Mapping is not specified yet.
StringExpression	Mapping is not specified yet.
TimeConstraint	Mapping is not specified yet.
TimeInterval	Mapping is not specified yet.
TimeObservation	Mapping is not specified yet.

7.7.14.3 Mapping Specifications

7.7.14.3.1 EqualOperatorExpressionFeature_Mapping

Description

The mapping class creates the feature element for the equal operator.

General Mappings

GenericToFeature_Mapping

Mapping Source

TypedElement

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.14.3.2 EqualOperatorExpressionFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

TypedElement

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  CommonFeatureReferenceExpression_Mapping.getMapped(from)
  ```

7.7.14.3.3 EqualOperatorExpressionOperandParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

TypedElement

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  EqualOperatorExpressionFeature_Mapping.getMapped(from)
  ```

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

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

7.7.14.3.4 Expression_Mapping

Description

A UML4SysML::Expression element is mapped to a SysML v2 OperatorExpression element.

General Mappings

GenericToExpression_Mapping
NamedElementMain_Mapping

Mapping Source

Expression

Mapping Target

OperatorExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- OperatorExpression::operator () : String [1]

```
from.symbol
```

7.7.14.3.5 ExpressionElse_Mapping

Description

A UML4SysML::Expression element with operator "else" is mapped to a SysML v2 TextualRepresentation element with language set to "SysMLv1" and body set to "else".

General Mappings

Expression_Mapping

Mapping Source

Expression

Mapping Target

OperatorExpression

Owned Mappings

(none)

Applicable filters

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

```
src.symbol = 'else'
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.14.3.6 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

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  ExpressionElseSpecification_Mapping.getMapped(from)
  ```

7.7.14.3.7 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

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
  'SysMLv1'
  ```

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

  ```
  'else'
  ```

7.7.14.3.8 LiteralBoolean_Mapping

Description

The mapping class maps UML4SysML::LiteralBoolean to SysML v2 LiteralBoolean.

General Mappings

LiteralSpecificationCommon_Mapping

Mapping Source

LiteralBoolean

Mapping Target

LiteralBoolean

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- LiteralBoolean::value () : Boolean [1]

```
from.value
```

7.7.14.3.9 LiteralInteger_Mapping

Description

The mapping class maps UML4SysML::LiteralInteger to SysML v2 LiteralInteger.

General Mappings

LiteralSpecificationCommon_Mapping

Mapping Source

LiteralInteger

Mapping Target

LiteralInteger

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- LiteralInteger::value () : Integer [1]

```
from.value
```

7.7.14.3.10 LiteralNull_Mapping

Description

The mapping class maps UML4SysML::LiteralNull to SysML v2 NullExpression.

General Mappings

LiteralSpecificationCommon_Mapping

Mapping Source

LiteralNull

Mapping Target

NullExpression

Owned Mappings

(none)

7.7.14.3.11 LiteralReal_Mapping

Description

The mapping class maps UML4SysML::LiteralReal to SysML v2 LiteralRational.

General Mappings

LiteralSpecificationCommon_Mapping

Mapping Source

LiteralReal

Mapping Target

LiteralRational

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- LiteralRational::value () : Real [1]

  ```
  from.value
  ```

7.7.14.3.12 LiteralSpecificationCommon_Mapping

Description

The mapping class the is abstract base class for all concrete UML4SysML::LiteralSpecification mappings.

General Mappings

ValueSpecification_Mapping

Mapping Source

LiteralSpecification

Mapping Target

LiteralExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 let ownerships: Set(SYSML2::Relationship) =
    ElementOwnership_Mapping.getMappedColl(from.ownedComment)
    ->including(CommonReturnParameterFeatureMembership_Mapping.getMapped(from)) in
if from.type.oclIsUndefined() then
    ownerships
else
    ownerships->including(LiteralSpecificationTyping_Mapping.getMapped(from))
endif
```

7.7.14.3.13 LiteralSpecificationFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

TypedElementFeatureTyping_Mapping

Mapping Source

LiteralSpecification

Mapping Target

FeatureTyping

Owned Mappings

(none)

7.7.14.3.14 LiteralString_Mapping

Description

The mapping class maps UML4SysML::LiteralString to the SysML v2 LiteralString.

General Mappings

LiteralSpecificationCommon_Mapping

Mapping Source

LiteralString

Mapping Target

LiteralString

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
if from.value.oclIsUndefined() then '' else from.value endif
```

7.7.14.3.15 LiteralUnlimitedUnbounded_Mapping

Description

The mapping class maps UML4SysML::LiteralUnlimited to SysML v2 LiteralInfinity if it is the unlimited value.

General Mappings

LiteralUnlimitedInteger_Mapping

Mapping Source

LiteralUnlimitedNatural

Mapping Target

LiteralInfinity

Owned Mappings

(none)

Applicable filters

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

```
(from.value = -1)
```

Mapping rules

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

7.7.14.3.16 LiteralUnlimitedInteger_Mapping

Description

The mapping class maps UML4SysML::LiteralUnlimited to SysML v2 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

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- LiteralInteger::value () : Integer [1]

```
from.value
```

7.7.14.3.17 OpaqueExpressionAsValue_Mapping

Description

The mapping class maps a UML4SysML::OpaqueExpression if it is used as a value to a SysML v2 FeatureChainExpression.

General Mappings

GenericToExpression_Mapping

Mapping Source

OpaqueExpression

Mapping Target

FeatureChainExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{OpaqueExpressionParameterMembership_Mapping.getMapped(from),
CommonReturnParameterFeatureMembership_Mapping.getMapped(from)}
```

7.7.14.3.18 OpaqueExpression_Mapping

Description

A UML4SysML::OpaqueExpression element is mapped to a SysMLv2 CalculationUsage element.. The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
calc sysMLv1OpaqueExpression {
        return result : ScalarValues::Integer;
        language "Built-in Math"
        /*
         * result = 42 + 23;
        */
}
```

General Mappings

CommonAction_Mapping
ValueSpecification_Mapping

Mapping Source

OpaqueExpression

Mapping Target

CalculationUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{OpaqueExpressionMembership_Mapping.getMapped(from),
OpaqueExpressionReferenceUsageReturnParameterMembership_Mapping.getMapped(from)}
```

7.7.14.3.19 OpaqueExpressionFeature_Mapping

Description

The mapping class creates the feature of the FeatureChainExpression.

General Mappings

GenericToFeature_Mapping

Mapping Source

OpaqueExpression

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{OpaqueExpressionFeatureValue_Mapping.getMapped(from),
OpaqueExpressionFeatureFeatureMembership_Mapping.getMapped(from)}
```

7.7.14.3.20 OpaqueExpressionFeatureFeature_Mapping

Description

The mapping class creates the Feature of the FeatureReferenceExpression.

General Mappings

GenericToFeature_Mapping

Mapping Source

OpaqueExpression

Mapping Target

Feature

Owned Mappings

(none)

7.7.14.3.21 OpaqueExpressionFeatureFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

OpaqueExpression

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
OpaqueExpressionFeatureFeature_Mapping.getMapped(from)
```

7.7.14.3.22 OpaqueExpressionFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

OpaqueExpression

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
OpaqueExpressionFeatureValueExpression_Mapping.getMapped(from)
```

7.7.14.3.23 OpaqueExpressionFeatureValueExpression_Mapping

Description

The mapping class creates the value of the FeatureChainExpression that is a FeatureReferenceExpression.

General Mappings

GenericToExpression_Mapping

Mapping Source

OpaqueExpression

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{OpaqueExpressionFeatureValueExpressionMembership_Mapping.getMapped(from),
EmptyReturnParameterFeatureMembership_Mapping.getMapped(from)}
```

7.7.14.3.24 OpaqueExpressionFeatureValueExpressionMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToMembership_Mapping

Mapping Source

OpaqueExpression

Mapping Target

Membership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
from
```

7.7.14.3.25 OpaqueExpressionMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

OpaqueExpression

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
OpaqueExpressionSpecification_Mapping.getMapped(from)
```

7.7.14.3.26 OpaqueExpressionParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

OpaqueExpression

Mapping Target

ParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
OpaqueExpressionFeature_Mapping.getMapped(from)
```

7.7.14.3.27 OpaqueExpressionReferenceUsageReturnParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToReturnParameterMembership_Mapping

Mapping Source

OpaqueExpression

Mapping Target

ReturnParameterMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 if from.type.oclIsUndefined() then
    OpaqueExpressionReferenceUsageUntyped_Mapping.getMapped(from)
else
    OpaqueExpressionReferenceUsage_Mapping.getMapped(from)
endif
```

7.7.14.3.28 OpaqueExpressionReferenceUsage_Mapping

Description

The mapping class creates the return parameter reference usage of the calculation usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

OpaqueExpression

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

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

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

7.7.14.3.29 OpaqueExpressionReferenceUsageFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

TypedElementFeatureTyping_Mapping

Mapping Source

OpaqueExpression

Mapping Target

FeatureTyping

Owned Mappings

(none)

7.7.14.3.30 OpaqueExpressionReferenceUsageUntyped_Mapping

Description

The mapping class creates the return parameter reference usage of the calculation usage, if the UML4SysML::OpaqueExpression is untyped.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

OpaqueExpression

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.7.14.3.31 OpaqueExpressionSpecification_Mapping

Description

The mapping class creates the specification of the calculation usage based on the language and body of the UML4SysML::OpaqueExpression.

General Mappings

GenericToTextualRepresentation_Mapping

Mapping Source

OpaqueExpression

Mapping Target

TextualRepresentation

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
if from.language->size() = 0 then OclUndefined else from.language.get(0) endif
```

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

```
        if from.body->size() = 0 then OclUndefined else from.body.get(0) endif
```

7.7.14.3.32 TimeExpression_Mapping

Description

A UML4SysML::TimeExpression is mapped to a SysML v2 TriggerInvocationExpression. The details of the mapping are not specified yet.

General Mappings

ValueSpecification_Mapping

Mapping Source

TimeExpression

Mapping Target

TriggerInvocationExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- TriggerInvocationExpression::kind () : TriggerKind [1]

```
SysMLv2::TriggerKind::at
```

7.7.14.3.33 ValueSpecification_Mapping

Description

The mapping class is the abstract base class of all mapping classes for special value specifications.

General Mappings

NamedElementMain_Mapping
GenericToExpression_Mapping

Mapping Source

ValueSpecification

Mapping Target

Expression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific 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),
    CommonReturnParameterFeatureMembership_Mapping.getMapped(from)}
endif
```

7.8 Mappings from SysML v1.7 stereotypes

This chapter lists all mapping specifications of SysML v1 stereotype model elements.

7.8.1 Overview

The following subclauses of 7.8 are organized according to the main packages of SysML v1.

7.8.2 Activities

This chapter lists all mapping specifications of SysML::Activities model elements.

7.8.2.1 Overview

The following table gives an overview of which SysML v2 elements the SysML::Activities elements are transformed with which mapping class. The mapping details are specified in 7.8.2.3.

The justifications for the elements without mapping are given in 7.8.2.2.

Table 24. List of all mappings

SysML v1 Concept	SysML v2 Concept
Continuous	MetadataUsage
ControlOperator	
Discrete	MetadataUsage
NoBuffer	
Optional	
Overwrite	
Probability	MetadataUsage

SysML v1 Concept	SysML v2 Concept
Rate	MetadataUsage

7.8.2.2 SysML::Activities elements not mapped

Table 25. List of SysML v1 elements not mapped of this section

SysML v1 Concept	Rationale
ControlOperator	The concept that an action can control other actions is not supported by SysML v2.
NoBuffer	Mapping is not specified yet.
Optional	The stereotype states that the lower multiplicity of the parameter is 0. Since the multiplicity of the parameter is transformed, the additional statement that the parameter is optional is redundant. Therefore, the stereotype is not considered in the transformation.
Overwrite	Mapping is not specified yet.

7.8.2.3 Mapping Specifications

7.8.2.3.1 ProbabilityMetadataUsage_Mapping

Description

A SysML::Activities::Probability is mapped to a SysML v2 MetadataUsage owned by the appropriate target element of the UML4SysML::ActivityEdge or UML4SysML::ParameterSet.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action def SysMLv1Activity {
    action sysMLv1Action1;
    succession sysMLv1ControlFlow1 first sysMLv1Action1 then sysMLv1Action2 {
        @SysMLv1Library::ProbabilityData {probability = 0.42;}
    }
    action sysMLv1Action2;
}
```

General Mappings

GenericToMetadataUsage_Mapping

Mapping Source

Element

Mapping Target

MetadataUsage

Owned Mappings

(none)

Applicable filters

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

```
Helper.hasStereotypeApplied(src, 'SysML::Activities::Probability')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
 Set{ProbabilityMetadataUsageFeatureTyping_Mapping.getMapped(from),
ProbabilityMetadataUsageFeatureMembership_Mapping.getMapped(from)}
```

7.8.2.3.2 ProbabilityMetadataUsageFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Element

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

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

```
Helper.hasStereotypeApplied(src, 'SysML::Activities::Probability')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
ProbabilityMetadataUsageReferenceUsage_Mapping.getMapped(from)
```

7.8.2.3.3 ProbabilityMetadataUsageFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Element

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

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

```
Helper.hasStereotypeApplied(src, 'SysML::Activities::Probability')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.8.2.3.4 ProbabilityMetadataUsageReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Element

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

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

```
Helper.hasStereotypeApplied(src, 'SysML::Activities::Probability')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
   Set{ProbabilityMetadataUsageReferenceUsageRedefinition_Mapping.getMapped(from),
  ProbabilityMetadataUsageReferenceUsageFeatureValue_Mapping.getMapped(from)}
  ```

7.8.2.3.5 ProbabilityMetadataUsageReferenceUsageFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Element

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

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

```
Helper.hasStereotypeApplied(src, 'SysML::Activities::Probability')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
   let probability : OclAny =
  Helper.getTagValue(from, 'SysML::Activities::Probability', 'probability') in
  LiteralRational_Factory.create(probability)
  ```

7.8.2.3.6 ProbabilityMetadataUsageReferenceUsageRedefinition_Mapping

Description

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

General Mappings

GenericToRedefinition_Mapping

Mapping Source

Element

Mapping Target

Redefinition

Owned Mappings

(none)

Applicable filters

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

```
Helper.hasStereotypeApplied(src, 'SysML::Activities::Probability')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.8.2.3.7 ProbabilityOwningMembership_Mapping

Description

Creates a owning membership relationship for *ownedMemberElement()*.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Element

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

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

```
Helper.hasStereotypeApplied(src, 'SysML::Activities::Probability')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
ProbabilityMetadataUsage_Mapping.getMapped(from)
```

7.8.2.3.8 RateMetadataUsage_Mapping

Description

A SysML::Activities::Rate and the specializations SysML::Activities::Discrete and SysML::Activities::Continuous are mapped to a SysML v2 MetadataUsage owned by the appropriate target element of the UML4SysML::ActivityEdge or UML4SysML::Parameter.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
succession flow sysMLv1ObjectFlow of SysMLv1Block
    from sysMLv1Action1.outputValue to sysMLv1Action1.inputValue {
        @SysMLv1Library::RateData {isDiscrete = true;}
}
```

The mapping of the rate instance value is not supported yet.

General Mappings

GenericToMetadataUsage_Mapping

Mapping Source

Element

Mapping Target

MetadataUsage

Owned Mappings

(none)

Applicable filters

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

```
Helper.hasStereotypeApplied(src, 'SysML::Activities::Rate')
or Helper.hasStereotypeApplied(src, 'SysML::Activities::Continuous')
or Helper.hasStereotypeApplied(src, 'SysML::Activities::Discrete')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
let relationships : Set(KerML::Relationship) =
    Set{RateMetadataUsageFeatureTyping_Mapping.getMapped(from)} in
if Helper.hasStereotypeApplied(from, 'SysML::Activities::Discrete') then
    relationships
    ->including(
        RateMetadataUsageDiscreteFeatureMembership_Mapping.getMapped(from))
else if Helper.hasStereotypeApplied(from, 'SysML::Activities::Continuous') then
        relationships
        ->including(
            RateMetadataUsageContinuousFeatureMembership_Mapping.getMapped(from))
     else
        relationships
     endif
endif
```

7.8.2.3.9 RateMetadataUsageContinuousFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Element

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

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

```
Helper.hasStereotypeApplied(src, 'SysML::Activities::Continuous')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
RateMetadataUsageContinuousReferenceUsage_Mapping.getMapped(from)
```

7.8.2.3.10 RateMetadataUsageFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Element

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

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

```
Helper.hasStereotypeApplied(src, 'SysML::Activities::Rate')
or Helper.hasStereotypeApplied(src, 'SysML::Activities::Continuous')
or Helper.hasStereotypeApplied(src, 'SysML::Activities::Discrete')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
LiteralBoolean_Factory.create(true)
```

7.8.2.3.11 RateMetadataUsageContinuousReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Element

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

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

```
Helper.hasStereotypeApplied(src, 'SysML::Activities::Continuous')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

  ```
   Set{RateMetadataUsageContinuousReferenceUsageRedefinition_Mapping.getMapped(from),
  RateMetadataUsageFeatureValue_Mapping.getMapped(from)}
  ```

7.8.2.3.12 RateMetadataUsageContinuousReferenceUsageRedefinition_Mapping

Description

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

General Mappings

GenericToRedefinition_Mapping

Mapping Source

Element

Mapping Target

Redefinition

Owned Mappings

(none)

Applicable filters

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

```
Helper.hasStereotypeApplied(src, 'SysML::Activities::Continuous')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

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

7.8.2.3.13 RateMetadataUsageDiscreteFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Element

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

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

```
Helper.hasStereotypeApplied(src, 'SysML::Activities::Discrete')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

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

```
RateMetadataUsageDiscreteReferenceUsage_Mapping.getMapped(from)
```

7.8.2.3.14 RateMetadataUsageDiscreteReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Element

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

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

```
Helper.hasStereotypeApplied(src, 'SysML::Activities::Discrete')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

  ```
   Set{RateMetadataUsageDiscreteReferenceUsageRedefinition_Mapping.getMapped(from),
  RateMetadataUsageFeatureValue_Mapping.getMapped(from)}
  ```

7.8.2.3.15 RateMetadataUsageDiscreteReferenceUsageRedefinition_Mapping

Description

Creates a redefinition relationship for the *redefiningFeature()* and the *redefinedFeature()*.

General Mappings

GenericToRedefinition_Mapping

Mapping Source

Element

Mapping Target

Redefinition

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
Helper.hasStereotypeApplied(src, 'SysML::Activities::Discrete')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Redefinition::redefinedFeature () : Feature [1]

```
 SYSML2::AttributeUsage.allInstances()
->any(m | m.qualifiedName = 'SysMLv1Library::RateData::isDiscrete')
```

7.8.2.3.16 RateMetadataUsageFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Element

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
Helper.hasStereotypeApplied(src, 'SysML::Activities::Rate')
or Helper.hasStereotypeApplied(src, 'SysML::Activities::Continuous')
or Helper.hasStereotypeApplied(src, 'SysML::Activities::Discrete')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
 SYSML2::MetadataDefinition.allInstances()
->any(m | m.qualifiedName = 'SysMLv1Library::RateData')
```

7.8.2.3.17 RateOwningMembership_Mapping

Description

Creates a owning membership relationship for *ownedMemberElement()*.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Element

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
Helper.hasStereotypeApplied(src, 'SysML::Activities::Rate')
or Helper.hasStereotypeApplied(src, 'SysML::Activities::Continuous')
or Helper.hasStereotypeApplied(src, 'SysML::Activities::Discrete')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

  ```
  RateMetadataUsage_Mapping.getMapped(from)
  ```

7.8.2.3.18 Model Libraries

7.8.2.3.18.1 ControlValues

The SysML v1 model library ControlValues contains the enumeration ControlValueKind.

7.8.2.3.18.1.1 ControlValueKind

The enumeration ControlValueKind is mapped to the SysML v2 enumeration definition SysMLv1Library::Enumerations::ControlValueKind (see 7.3.2).

7.8.3 Allocations

This chapter lists all mapping specifications of SysML::Allocations model elements.

7.8.3.1 Overview

The following table gives an overview of which SysML v2 elements the SysML::Allocations elements are transformed with which mapping class. The mapping details are in 7.8.3.3.

The justifications for the elements without mapping are given in 7.8.3.2.

Table 26. List of all mappings

SysML v1 Concept	SysML v2 Concept
Allocate	AllocationUsage
AllocateActivityPartition	

7.8.3.2 SysML::Allocations elements not mapped

Table 27. List of SysML v1 elements not mapped of this section

SysML v1 Concept	Rationale
AllocateActivityPartition	Mapping is not specified yet.

7.8.3.3 Mapping Specifications

7.8.3.3.1 AllocationDefinition_Mapping

Description

A SysML::Allocations::Allocate is mapped to a SysML v2 AllocationDefinition if it is an allocation between definition elements.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
action def SysMLv1Activity;
part def SysMLv1Block;

allocation def SysMLv1Allocation {
        end : SysMLv1Activity;
        end : SysMLv1Block;
}
```

General Mappings

Abstraction_Mapping

Mapping Source

Dependency

Mapping Target

AllocationDefinition

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
 (Helper.hasStereotypeApplied(src, 'SysML::Allocations::Allocate')) and
src.client->any(c | true).oclIsKindOf(UML::Type)
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- AllocationDefinition::ownedRelationship () : Relationship [0..*]

```
 Set{AllocationDefinitionFromFeatureMembership_Mapping.getMapped(from),
AllocationDefinitionToFeatureMembership_Mapping.getMapped(from)}
```

7.8.3.3.2 AllocationDefinitionToFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Dependency

Mapping Target

FeatureMembership

Owned Mappings

- allocationDefinitionToReferenceUsage : AllocationDefinitionToReferenceUsage_Mapping

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureMembership::memberName () : String [0..1]

```
'allocatedTo'
```

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
allocationDefinitionToReferenceUsage.to
```

7.8.3.3.3 AllocationDefinitionFromFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Dependency

Mapping Target

FeatureMembership

Owned Mappings

- allocationDefinitionFromReferenceUsage : AllocationDefinitionFromReferenceUsage_Mapping

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureMembership::memberName () : String [0..1]

```
'allocatedFrom'
```

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
allocationDefinitionFromReferenceUsage.to
```

7.8.3.3.4 AllocationDefinitionFromFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Dependency

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
from.source.get(0)
```

7.8.3.3.5 AllocationDefinitionFromReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Dependency

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
Set{AllocationDefinitionFromFeatureTyping_Mapping.getMapped(from)}
```

- ReferenceUsage::isEnd () : Boolean [1]

```
true
```

7.8.3.3.6 AllocationDefinitionToFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Dependency

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
from.target.get(0)
```

7.8.3.3.7 AllocationDefinitionToReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Dependency

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

  ```
  Set{AllocationDefinitionToFeatureTyping_Mapping.getMapped(from)}
  ```

- ReferenceUsage::isEnd () : Boolean [1]

  ```
  true
  ```

7.8.3.3.8 AllocationUsage_Mapping

Description

A SysML::Allocations::Allocate is mapped to a SysML v2 AllocationUsage if it is an allocation between usage elements.

The details of the mapping is not defined yet.

General Mappings

GenericToUsage_Mapping
Abstraction_Mapping

Mapping Source

Dependency

Mapping Target

AllocationUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
 (Helper.hasStereotypeApplied(src, 'SysML::Allocations::Allocate')) and
not src.client->any(c | true).oclIsKindOf(UML::Type)
```

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

7.8.4 Blocks

This chapter lists all mapping specifications of SysML::Blocks model elements.

7.8.4.1 Overview

The following table gives an overview of which SysML v2 elements the SysML::Blocks elements are transformed with which mapping class. The mapping details are in 7.8.4.3.

The justifications for the elements without mapping are given in 7.8.4.2.

Table 28. List of all mappings

SysML v1 Concept	SysML v2 Concept
AdjunctProperty	
BindingConnector	BindingConnectorAsUsage
Block	PartDefinition PartDefinition
BoundReference	
ClassifierBehaviorProperty	
ConnectorProperty	
DistributedProperty	
EndPathMultiplicity	
NestedConnectorEnd	
ParticipantProperty	
PropertySpecificType	
ValueType	AttributeDefinition

7.8.4.2 SysML::Blocks elements not mapped

Table 29. 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.
BoundReference	Mapping is not specified yet.
ClassifierBehaviorProperty	The classifier behavior is already mapped to a property which also plays the role of the classifier behavior property. Therefore, there is no explicit mapping of a classifier behavior property.
ConnectorProperty	The connector property is a special case of an adjunct property and is not mapped, just like the adjunct property.
DirectedRelationshipPropertyPath	The stereotype is abstract is therefore not mapped. The concept of the DirectedRelationshipPropertyPath is included in the SysML v2 language.
DistributedProperty	Mapping is not specified yet.

SysML v1 Concept	Rationale
ElementPropertyPath	The stereotype is abstract is therefore not mapped. The concept of the ElementPropertyPath is included in the SysML v2 language.
EndPathMultiplicity	Mapping is not specified yet.
NestedConnectorEnd	The concept of NestedConnectorEnd is already included in the SysML v2 language. It is not required to do an explicit mapping.
ParticipantProperty	Mapping is not specified yet.
PropertySpecificType	Mapping is not specified yet.

7.8.4.3 Mapping Specifications

7.8.4.3.1 AssociationBlock_Mapping

Description

An AssociationBlock is mapped to a SysML v2 ConnectionDefinition.

The SysML::Blocks::ParticipantProperties transformation is not defined yet. Therefore, the mapping is currently identical with the mapping of UML4SysML::AssociationClass.

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(src, 'SysML::Blocks::Block')
```

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

7.8.4.3.2 BindingConnector_Mapping

Description

A SysML::Blocks::BindingConnector is mapped to a SysML v2 BindingConnectorAsUsage.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
part def SysMLv1Block1 {
        part sysMLv1PartProperty1 : SysMLv1Block2;
        part sysMLv1PartProperty2 : SysMLv1Block2;

        binding sysMLv1BindingConnector
            bind sysMLv1PartProperty1 = sysMLv1PartProperty2;
}
part def SysMLv1Block2;
```

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(src, 'SysML::Blocks::BindingConnector')
```

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

7.8.4.3.3 Block_Mapping

Description

A SysML::Blocks::Block is mapped to a SysML v2 PartDefinition.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
part definition SysMLv1Block;
```

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 src.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')
```

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

7.8.4.3.4 EncapsulatedBlock_Mapping

Description

A SysML::Block with *isEncapsulated=true* is mapped to a SysML v2 PartDefinition, and, additionally, gets a metadata feature defined by the SysML v1 library which represents the SysML v1 isEncapsulated property.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
part def SysMLv1EncapsulatedBlock {
  @SysMLv1Library::BlockData {isEncapsulated = true;}
}
```

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 src.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')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- PartDefinition::ownedRelationship () : Relationship [0..*]

```
let toElementFMS: Set(UML::Element) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Property) and
    (e.oclAsType(UML::Property).redefinedProperty->size() = 0)) in
let redefinedAttributes: Set(UML::Element) =
    from.ownedElement->select(e | from.oclIsKindOf(UML::DataType) and
    (e.oclAsType(UML::Property).redefinedProperty->size() > 0)) in
let generalizations : Set(UML::Generalization) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Generalization)) in
let toElementOMS: Set(UML::Element) =
    (((from.ownedElement - toElementFMS) - redefinedAttributes) -
    generalizations)  in
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(ClassifierBehaviorFeatureMembership_Mapping.getMapped(from))
endif
```

7.8.4.3.5 EncapsulatedBlockMetadataMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Class

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

```
 EncapsulatedBlockMetadata_Mapping.getMapped(from)
```

7.8.4.3.6 EncapsulatedBlockMetadata_Mapping

Description

The mapping class creates the metadata for the property SysML::Blocks::Block::isEncapsulated.

General Mappings

GenericToMetadataUsage_Mapping

Mapping Source

Class

Mapping Target

MetadataUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- MetadataUsage::ownedRelationship () : Relationship [0..*]

```
 Set{EncapsulatedBlockMetadataFeatureTyping_Mapping.getMapped(from),
EncapsulatedBlockMetadataFeatureMembership_Mapping.getMapped(from)}
```

7.8.4.3.7 EncapsulatedBlockMetadataFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Class

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [0..1]

```
EncapsulatedBlockMetadataReferenceUsage_Mapping.getMapped(from)
```

7.8.4.3.8 EncapsulatedBlockMetadataFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Class

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
 SYSML2::MetadataDefinition.allInstances()
->any(m | m.qualifiedName = 'SysMLv1Library::BlockData')
```

7.8.4.3.9 EncapsulatedBlockMetadataReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Class

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
 Set{EncapsulatedBlockMetadataRedefinition_Mapping.getMapped(from),
EncapsulatedBlockMetadataFeatureValue_Mapping.getMapped(from)}
```

7.8.4.3.10 EncapsulatedBlockMetadataFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Class

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

```
LiteralBoolean_Factory.create(true)
```

7.8.4.3.11 EncapsulatedBlockMetadataRedefinition_Mapping

Description

Creates a redefinition relationship for the *redefiningFeature()* and the *redefinedFeature()*.

General Mappings

GenericToRedefinition_Mapping

Mapping Source

Class

Mapping Target

Redefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Redefinition::redefinedFeature () : Feature [1]

```
 SYSML2::AttributeUsage.allInstances()
->any(m | m.qualifiedName = 'SysMLv1Library::BlockData::isEncapsulated')
```

7.8.4.3.12 Part_Mapping

Description

A UML4SysML::Property with composite aggregation kind which is typed by a block is mapped to a SysML::PartUsage.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
part def SysMLv1Block1 {
        part sysMLv1PartProperty1 : SysMLv1Block2;
}
part def SysMLv1Block2;
```

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) and not src.oclIsKindOf(UML::Port) 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.

7.8.4.3.13 Model Libraries

The Blocks section of the SysML v1 specification contains two model libraries.

7.8.4.3.13.1 PrimitiveValueTypes

The SysML v1 model library PrimitiveValueTypes contains primitive types that are mapped to the appropriate scalar values in SysML v2.

7.8.4.3.13.1.1 Boolean

The SysML v1 primitive type Boolean is mapped to the SysML v2 ScalarValues::Boolean element.

7.8.4.3.13.1.2 Complex

The SysML v1 primitive type Complex is mapped to the SysML v2 ScalarValues::Complex element.

7.8.4.3.13.1.3 Integer

The SysML v1 primitive type Integer is mapped to the SysML v2 ScalarValues::Integer element.

7.8.4.3.13.1.4 Number

The SysML v1 primitive type Number is abstract. Therefore, no mapping is defined for it.

7.8.4.3.13.1.5 Real

The SysML v1 primitive type Real is mapped to the SysML v2 ScalarValues::Real element.

7.8.4.3.13.1.6 String

The SysML v1 primitive type String is mapped to the SysML v2 ScalarValues::String element.

7.8.4.3.13.2 UnitAndQuantityKind

The SysML v1 model library UnitAndQuantityKind contains the blocks Unit and QuantityKind.

7.8.4.3.13.2.1 QuantityKind

The mapping of the SysML v1 QuantityKind element is not specified yet.

7.8.4.3.13.2.2 Unit

The mapping of the SysML v1 QuantityKind element is not specified yet.

7.8.5 ConstraintBlocks

7.8.5.1 Overview

The following table gives an overview of which SysML v2 elements the SysML::ConstraintBlocks elements are transformed with which mapping class. The mapping details are in 7.8.5.2.

Table 30. List of all mappings

SysML v1 Concept	SysML v2 Concept
ConstraintBlock	ConstraintDefinition

7.8.5.2 Mapping Specifications

7.8.5.2.1 ConstraintBlock_Mapping

Description

A SysML::ConstraintBlocks::ConstraintBlock is mapped to a SysML v2 ConstraintDefinition.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
onstraint def SysMLv1ConstraintBlock {
        in attribute a : ScalarValues::Integer;
        in attribute b : ScalarValues::Integer;
        in attribute c : ScalarValues::Integer;

        constraint constraintExpression {
                language "OCL2.0"
                /*
                 * c == a + b
                 */
         }
}
```

General Mappings

Class_Mapping

Mapping Source

Class

Mapping Target

ConstraintDefinition

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
 Helper.hasStereotypeApplied(src, 'SysML::ConstraintBlocks::ConstraintBlock')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ConstraintDefinition::ownedRelationship () : Relationship [0..*]

```
let generalizations : Set(UML::Generalization) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Generalization)) in
let toElementFMS : Set(UML::Element) =
    from.ownedElement
    ->select(e | e.oclIsKindOf(UML::Property) or e.oclIsKindOf(UML::Constraint)) in
let toElementOMS: Set(UML::Element) =
    (from.ownedElement - generalizations) - toElementFMS  in
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))
->including(CommonReturnParameterReferenceUsageMembership_Mapping.getMapped(from))
```

7.8.5.2.2 ConstraintParameter_Mapping

Description

The mapping class maps SysML v1 constraint parameter to SysML v2 attribute usages.

General Mappings

Property_Mapping

Mapping Source

Property

Mapping Target

AttributeUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
Helper.hasStereotypeApplied(src.owner, 'SysML::ConstraintBlocks::ConstraintBlock')
```

Mapping rules

The mapping class only has inherited rules. See the mapping classes in the general mapping section for details.

7.8.6 Model Elements

This chapter lists all mapping specifications of SysML::ModelElements model elements.

7.8.6.1 Overview

The following table gives an overview of which SysML v2 elements the SysML::ModelElements elements are transformed with which mapping class. The mapping details are in 7.8.6.3.

The justifications for the elements without mapping are given in 7.8.6.2.

Table 31. List of all mappings

SysML v1 Concept	SysML v2 Concept
Conform	

SysML v1 Concept	SysML v2 Concept
ElementGroup	Package
Expose	
Problem	Comment
Rationale	Comment
Stakeholder	ItemDefinition
View	
Viewpoint	

7.8.6.2 SysML::ModelElements elements not mapped

Table 32. List of SysML v1 elements not mapped of this section

SysML v1 Concept	Rationale
Conform	Mapping is not specified yet.
Expose	Mapping is not specified yet.
View	Mapping is not specified yet.

7.8.6.3 Mapping Specifications

7.8.6.3.1 ProblemRationaleMetadataFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Comment

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [0..1]

```
ProblemRationaleMetadataReferenceUsage_Mapping.getMapped(from)
```

7.8.6.3.2 ProblemRationaleMetadataFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Comment

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
if Helper.hasStereotypeApplied(from, 'SysML::ModelElements::Problem') then
  SYSML2::MetadataDefinition.allInstances()
    ->any(m | m.qualifiedName = 'ModelingMetadata::Issue')
else if Helper.hasStereotypeApplied(from, 'SysML::ModelElements::Rationale') then
  SYSML2::MetadataDefinition.allInstances()
    ->any(m | m.qualifiedName = 'ModelingMetadata::Rationale')
else OclUndefined endif endif
```

7.8.6.3.3 ProblemRationaleMetadataReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Comment

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
 Set{ProblemRationaleMetadataRedefinition_Mapping.getMapped(from),
ProblemRationaleMetadataFeatureValue_Mapping.getMapped(from)}
```

7.8.6.3.4 ProblemRationaleMetadataFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Comment

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

```
LiteralString_Factory.create(from.body)
```

7.8.6.3.5 ProblemRationaleMetadataMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Comment

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

```
ProblemRationaleMetadataUsage_Mapping.getMapped(from)
```

7.8.6.3.6 Concern_Mapping

Description

The concern comments of a SysML::ModelElements::Stakeholder or a SysML::ModelElements::Viewpoint are mapped to SysML v2 ConcernUsages. The concern comments of the stakeholder are mapped to ConcernUsages which reference the stakeholder item definition.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
item def SysMLv1Stakeholder {
        @SysMLv1Library::StakeholderData {isStakeholder = true;}
}
concern concernCommentXMI_ID {
        doc /* concern string */
        stakeholder : SysMLv1Stakeholder;
}
```

General Mappings

Comment_Mapping

Mapping Source

Comment

Mapping Target

ConcernUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
(not Helper.hasStereotypeApplied(src, '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(src)) or
(UML::Classifier.allInstances()
->select(s |
    Helper.hasStereotypeApplied(s, 'SysML::ModelElements::Viewpoint'))
->collect(c |
    Helper.getTagValue(c, 'SysML::ModelElements::Viewpoint', 'concernList'))
->flatten()->includes(src)))
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ConcernUsage::ownedRelationship () : Relationship [0..*]

```
let toStakeholderMS : Set(UML::Classifier) =
    UML::Classifier.allInstances()
    ->select(s |
        Helper.hasStereotypeApplied(s, 'SysML::ModelElements::Stakeholder'))
    ->select(s |
        Helper.getTagValue(s, 'SysML::ModelElements::Stakeholder', 'concernList')
    ->flatten()->includes(from)) in
```

```
toStakeholderMS
->collect(e | ConcernStakeholderMembership_Mapping.getMapped(e))
->including(ConcernOwningMembership_Mapping.getMapped(from))
->including(
    CommonReturnParameterReferenceUsageMembership_Mapping.getMapped(from))
->including(EmptySubjectMembership_Factory.create())
```

7.8.6.3.7 ConcernDocumentation_Mapping

Description

The mapping class creates the documentation element with the body string of the UML4SysML::Comment model element representing a concern.

General Mappings

GenericToDocumentation_Mapping

Mapping Source

Comment

Mapping Target

Documentation

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Documentation::body () : String [1]

```
from.body
```

7.8.6.3.8 ConcernOwningMembership_Mapping

Description

Creates a owning membership relationship for *ownedMemberElement()*.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Comment

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

```
ConcernDocumentation_Mapping.getMapped(from)
```

7.8.6.3.9 ConcernStakeholderMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToParameterMembership_Mapping

Mapping Source

Classifier

Mapping Target

StakeholderMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- StakeholderMembership::ownedMemberParameter () : Feature [1]

```
 ConcernStakeholderPartUsage_Mapping.getMapped(from)
```

7.8.6.3.10 ConcernStakeholderPartUsage_Mapping

Description

In SysML v1, the stakeholder element has concerns. In SysML v2, the Concern element has stakeholders. This mapping class creates a PartUsage of the type of the stakeholder for the concern element.

General Mappings

GenericToPartUsage_Mapping

Mapping Source

Classifier

Mapping Target

PartUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- PartUsage::ownedRelationship () : Relationship [0..*]

```
 Set{ConcernStakeholderPartUsageFeatureTyping_Mapping.getMapped(from),
ConcernStakeholderPartUsageOwningMembership_Mapping.getMapped(from)}
```

7.8.6.3.11 ConcernStakeholderPartUsageFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Classifier

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

  ```
  from
  ```

7.8.6.3.12 ConcernStakeholderPartUsageOwningMembership_Mapping

Description

Creates a owning membership relationship for *ownedMemberElement()*.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Classifier

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

```
        ConcernStakeholderPartUsageFeature_Mapping.getMapped(from)
```

7.8.6.3.13 ConcernStakeholderPartUsageFeature_Mapping

Description

The mapping class creates a feature element for the concern stakeholder part usage.

General Mappings

GenericToFeature_Mapping

Mapping Source

Classifier

Mapping Target

Multiplicity

Owned Mappings

(none)

7.8.6.3.14 ElementGroup_Mapping

Description

A SysML::ModelElements::ElementGroup element is mapped to a SysML v2 Package with membership import relationships representing the grouping.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
package ElementGroupModel {
    part def SysMLv1Block1;
    attribute def SysMLv1ValueType;
    part def SysMLv1Block2 {
        part sysMLv1PartProperty:SysMLv1Block1;
    }
}

package SysMLv1ElementGroup {
    import ElementGroupModel::SysMLv1Block1;
    import ElementGroupModel::SysMLv1ValueType;
    import ElementGroupModel::SysMLv1Block2::sysMLv1PartProperty;

    @SysMLv1Library::ElementGroupData {criterion = "criterion string";}
}
```

General Mappings

Comment_Mapping

Mapping Source

Comment

Mapping Target

Package

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
Helper.hasStereotypeApplied(src, 'SysML::ModelElements::ElementGroup')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Package::ownedRelationship () : Relationship [0..*]

```
let elements : Set(KerML::Relationahip) =
    Helper.getTagValueAsElementColl(from,
        'SysML::ModelElements::ElementGroup', 'member')
    ->collect(e | CommonElementImport_Mapping.getMapped(e)) in
elements->including(ElementGroupMetadaMembership_Mapping.getMapped(from))
```

- Package::declaredName () : String [0..1]

```
Helper.getTagValueAsString(from, 'SysML::ModelElements::ElementGroup', 'name')
```

7.8.6.3.15 ElementGroupMetadaMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Comment

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

  ```
  ElementGroupMetadataUsage_Mapping.getMapped(from)
  ```

7.8.6.3.16 ElementGroupMetadataFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Comment

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

  ```
  ElementGroupMetadataReferenceUsage_Mapping.getMapped(from)
  ```

7.8.6.3.17 ElementGroupMetadataFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Comment

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
 SYSML2::MetadataDefinition.allInstances()
->any(m | m.qualifiedName = 'SysMLv1Library::ElementGroupData')
```

7.8.6.3.18 ElementGroupMetadataFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Comment

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

```
 LiteralString_Factory.create(
Helper.getTagValueAsString(from, 'SysML::ModelElements::ElementGroup', 'criterion')
)
```

7.8.6.3.19 ElementGroupMetadataRedefinition_Mapping

Description

Creates a redefinition relationship for the *redefiningFeature()* and the *redefinedFeature()*.

General Mappings

GenericToRedefinition_Mapping

Mapping Source

Comment

Mapping Target

Redefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Redefinition::redefinedFeature () : Feature [1]

```
 let m : SYSML2::Membership =
    SYSML2::AttributeUsage.allInstances()
    ->collect(dt | dt.owningRelationship)
    ->select(r | r.oclIsKindOf(SYSML2::Membership))
    ->any(m | m.memberName = 'criterion') in
if (m.oclIsUndefined()) then
        OclUndefined
else
    m.memberElement
endif
```

7.8.6.3.20 ElementGroupMetadataReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Comment

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
 Set{ElementGroupMetadataRedefinition_Mapping.getMapped(from),
ElementGroupMetadataFeatureValue_Mapping.getMapped(from)}
```

7.8.6.3.21 ElementGroupMetadataUsage_Mapping

Description

The mapping class creates the metadata usage element for the SysML::ModelElements::ElementGroup mapping.

General Mappings

GenericToMetadataUsage_Mapping

Mapping Source

Comment

Mapping Target

MetadataUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- MetadataUsage::ownedRelationship () : Relationship [0..*]

```
  Set{ElementGroupMetadataFeatureTyping_Mapping.getMapped(from),
ElementGroupMetadataFeatureMembership_Mapping.getMapped(from)}
```

7.8.6.3.22 ProblemRationale_Mapping

Description

The mapping class combines the mapping of SysML::ModelElements::Problem and SysML::ModelElements::Rationale. The SysML::ModelElements::Problem is mapped to the library element ModelingMetadata::Issue and the SysML::ModelElements::Rationale is mapped to ModelingMetadata::Rationale.

The expected SysML v2 textual syntax of the mapping is as follows.

```
@ModelingMetadata::Issue {text = "This is a problem statement";}

@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(src, 'SysML::ModelElements::ElementGroup')) and
(Helper.hasStereotypeApplied(src, 'SysML::ModelElements::Problem') or
Helper.hasStereotypeApplied(src, 'SysML::ModelElements::Rationale'))
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Comment::ownedRelationship () : Relationship [0..*]

```
  self.annotation()
->append(ProblemRationaleMetadataMembership_Mapping.getMapped(from))
```

7.8.6.3.23 ProblemRationaleMetadataRedefinition_Mapping

Description

Creates a redefinition relationship for the *redefiningFeature()* and the *redefinedFeature()*.

General Mappings

GenericToRedefinition_Mapping

Mapping Source

Comment

Mapping Target

Redefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific 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::text')
else if Helper.hasStereotypeApplied(from, 'SysML::ModelElements::Rationale') then
  SYSML2::AttributeUsage.allInstances()
  ->any(m | m.qualifiedName = 'ModelingMetadata::Rationale::text')
else
  OclUndefined
endif
endif
```

7.8.6.3.24 ProblemRationaleMetadataUsage_Mapping

Description

The mapping class creates the metadata usage element for the SysML::ModelElements::Problem and SysML::ModelElements::Rationale transformation target.

General Mappings

GenericToMetadataUsage_Mapping

Mapping Source

Comment

Mapping Target

MetadataUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- MetadataUsage::ownedRelationship () : Relationship [0..*]

```
 Set{ProblemRationaleMetadataFeatureTyping_Mapping.getMapped(from),
ProblemRationaleMetadataFeatureMembership_Mapping.getMapped(from)}
```

7.8.6.3.25 Stakeholder_Mapping

Description

A SysML::ModelElements::Stakeholder is mapped to a SysML v2 ItemDefinition with metadata to tag it as a stakeholder. The concern comments of the stakeholder are mapped to ConcernUsages which reference the stakeholder item definition.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
item def SysMLv1Stakeholder {@SysMLv1Library::StakeholderData {isStakeholder = true;}}
concern concernCommentXMI_ID {
        doc /* concern string */
        stakeholder : SysMLv1Stakeholder;
}
```

General Mappings

Class_Mapping

Mapping Source

Class

Mapping Target

ItemDefinition

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
Helper.hasStereotypeApplied(src, 'SysML::ModelElements::Stakeholder')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ItemDefinition::ownedRelationship () : Relationship [0..*]

```
let toElementFMS: Set(UML::Element) =
    from.ownedElement
    ->select(e | (e.oclIsKindOf(UML::Property) and
    (e.oclAsType(UML::Property).redefinedProperty->size() = 0)) or
    e.oclIsKindOf(UML::Operation)) in
let redefinedAttributes: Set(UML::Element) =
    from.ownedElement
    ->select(e | from.oclIsKindOf(UML::DataType) and
    (e.oclAsType(UML::Property).redefinedProperty->size() > 0)) in
let generalizations : Set(UML::Generalization) =
    from.ownedElement
    ->select(e | e.oclIsKindOf(UML::Generalization)) in
let constraints : Set(UML::Constraint) =
    UML::Constraint.allInstances()
    ->select( c | c.constrainedElement->includes(from)) in
let toElementOMS: Set(UML::Element) =
    (((from.ownedElement - toElementFMS) - redefinedAttributes) -
    generalizations)  in
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)))
->including(StakeholderMetadataOwningMembership_Mapping.getMapped(from)) in
if from.classifierBehavior.oclIsUndefined() then
    relationships
else
    relationships->append(ClassifierBehaviorFeatureMembership_Mapping.getMapped(from))
endif
```

7.8.6.3.26 StakeholderMetadataUsage_Mapping

Description

The mapping class creates the metadata usage element for the SysML::ModelElements::Stakeholder mapping.

General Mappings

GenericToMetadataUsage_Mapping

Mapping Source

Classifier

Mapping Target

MetadataUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- MetadataUsage::ownedRelationship () : Relationship [0..*]

```
 Set{StakeholderMetadataFeatureTyping_Mapping.getMapped(from),
StakeholderMetadataFeatureMembership_Mapping.getMapped(from)}
```

7.8.6.3.27 StakeholderMetadataFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Classifier

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
 StakeholderMetadataReferenceUsage_Mapping.getMapped(from)
```

7.8.6.3.28 StakeholderMetadataFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Classifier

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
 SYSML2::MetadataDefinition.allInstances()
->any(m | m.qualifiedName = 'SysMLv1Library::StakeholderData')
```

7.8.6.3.29 StakeholderMetadataOwningMembership

Description

Creates a owning membership relationship for *ownedMemberElement()*.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Classifier

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

```
StakeholderMetadataUsage_Mapping.getMapped(from)
```

7.8.6.3.30 StakeholderMetadataReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Classifier

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
 Set{StakeholderMetadataReferenceUsageRedefinition_Mapping.getMapped(from),
StakeholderMetadataReferenceUsageFeatureValue_Mapping.getMapped(from)}
```

7.8.6.3.31 StakeholderMetadataReferenceUsageFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Classifier

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

```
LiteralBoolean_Factory.create(true)
```

7.8.6.3.32 StakeholderMetadataReferenceUsageRedefinition_Mapping

Description

Creates a redefinition relationship for the *redefiningFeature()* and the *redefinedFeature()*.

General Mappings

GenericToRedefinition_Mapping

Mapping Source

Classifier

Mapping Target

Redefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Redefinition::redefinedFeature () : Feature [1]

```
 SYSML2::AttributeUsage.allInstances()
->any(m | m.qualifiedName = 'SysMLv1Library::StakeholderData::isStakeholder')
```

7.8.6.3.33 Viewpoint_Mapping

Description

A SysML::ModelElements::Viewpoint is mapped to a SysML v2 ViewDefinition with an owned SysML v2 ViewpointUsage. In SysML v1, the viewpoint combines the purpose and stakeholder concerns as well as presentation information. This is covered by a SysML v2 ViewDefinition with owned SysML v2 ViewpointUsage.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
view def SysMLv1Viewpoint {
        viewpoint sysMLv1Viewpoint {
                frame concern1XmiID1;
                frame concern2XmiID2;
                metadata SysMLv1Library::ViewpointData {
                        languages = ("language1","language2");
                        presentations = ("presentation1", "presentation2");
                }
                require constraint {
                        doc /* thisIsThePurpose */
                }
        }
        satisfy sysMLv1Viewpoint;
        rendering {
                action : SysMLv1ViewpointMethodBehavior1;
                action : SysMLv1ViewpointMethodBehavior2;
        }
}
action def SysMLv1ViewpointMethodBehavior1;
action def SysMLv1ViewpointMethodBehavior2;

item def SysMLv1Stakeholder {@SysMLv1Library::StakeholderData {isStakeholder = true;}}

concern concern1XmiID1 {
        doc /* Concern1 */
        stakeholder : SysMLv1Stakeholder;
}
concern concern2XmiID2 {
        doc /* Concern2 */
        stakeholder : SysMLv1Stakeholder;
}
```

General Mappings

Class_Mapping

Mapping Source

Class

Mapping Target

ViewDefinition

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
 Helper.hasStereotypeApplied(src, 'SysML::ModelElements::Viewpoint')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ViewDefinition::ownedRelationship () : Relationship [0..*]

```
let toElementFMS: Set(UML::Element) =
    from.ownedElement->select(e | (e.oclIsKindOf(UML::Property) and
        (e.oclAsType(UML::Property).redefinedProperty->size() = 0)) or
        e.oclIsKindOf(UML::Comment)) in
let redefinedAttributes: Set(UML::Element) =
    from.ownedElement->select(e | from.oclIsKindOf(UML::DataType) and
        (e.oclAsType(UML::Property).redefinedProperty->size() > 0)) in
let generalizations : Set(UML::Generalization) =
    from.ownedElement->select(e | e.oclIsKindOf(UML::Generalization)) in
let toElementOMS: Set(UML::Element) =
    (((from.ownedElement - toElementFMS) - redefinedAttributes) -
    generalizations)  in
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(ViewpointViewpointUsageFeatureMembership_Mapping.getMapped(from))
->including(ViewpointSatisfyFeatureMembership_Mapping.getMapped(from))
```

```
        ->including(ViewpointRenderingFeatureMembership_Mapping.getMapped(from))
        ->including(
            CommonReturnParameterReferenceUsageMembership_Mapping.getMapped(from)) in
        if from.classifierBehavior.oclIsUndefined() then
            relationships
        else
            relationships
            ->append(ClassifierBehaviorFeatureMembership_Mapping.getMapped(from))
        endif
```

7.8.6.3.34 ViewpointConcernReferenceSubsetting_Mapping

Description

Creates a subsetting relationship for the *subsettingFeature()* and the *subsettedFeature()*.

General Mappings

GenericToReferenceSubsetting_Mapping

Mapping Source

Comment

Mapping Target

ReferenceSubsetting

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ReferenceSubsetting::referencedFeature () : Feature [1]

```
from
```

7.8.6.3.35 ViewpointConcernUsage_Mapping

Description

The mapping class creates the concern usage element for the SysML::ModelElements::Viewpoint mapping.

General Mappings

GenericToRequirementUsage_Mapping

Mapping Source

Comment

Mapping Target

ConcernUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ConcernUsage::ownedRelationship () : Relationship [0..*]

```
 Set{ViewpointConcernReferenceSubsetting_Mapping.getMapped(from),
EmptySubjectMembership_Factory.create(),
CommonReturnParameterReferenceUsageMembership_Mapping.getMapped(from)}
```

7.8.6.3.36 ViewpointConstraintUsage_Mapping

Description

The mapping class creates the constraint usage element for the SysML::ModelElements::Viewpoint mapping.

General Mappings

GenericToConstraintUsage_Mapping

Mapping Source

Class

Mapping Target

ConstraintUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ConstraintUsage::ownedRelationship () : Relationship [0..*]

```
 Set{ViewpointConstraintUsageOwningMembership_Mapping.getMapped(from),
ReturnParameterFeatureMembership_Factory.create()}
```

7.8.6.3.37 ViewpointConstraintUsageDocumentation_Mapping

Description

The mapping class creates the documentation element for the SysML::ModelElements::Viewpoint mapping.

General Mappings

GenericToDocumentation_Mapping

Mapping Source

Class

Mapping Target

Documentation

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Documentation::body () : String [1]

```
Helper.getTagValueAsString(from, 'SysML::ModelElements::Viewpoint', 'purpose')
```

7.8.6.3.38 ViewpointConstraintUsageOwningMembership_Mapping

Description

Creates a owning membership relationship for *ownedMemberElement()*.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Class

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

```
ViewpointConstraintUsageDocumentation_Mapping.getMapped(from)
```

7.8.6.3.39 ViewpointFramedConcernMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Comment

Mapping Target

FramedConcernMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FramedConcernMembership::ownedMemberFeature () : Feature [1]

```
ViewpointConcernUsage_Mapping.getMapped(from)
```

7.8.6.3.40 ViewpointLanguagesMetadataFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Class

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
ViewpointLanguagesMetadataReferenceUsage_Mapping.getMapped(from)
```

7.8.6.3.41 ViewpointLanguagesMetadataFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Class

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

```
 ViewpointLanguagesMetadataOperatorExpression_Mapping.getMapped(from)
```

7.8.6.3.42 ViewpointLanguagesMetadataRedefinition_Mapping

Description

Creates a redefinition relationship for the *redefiningFeature()* and the *redefinedFeature()*.

General Mappings

GenericToRedefinition_Mapping

Mapping Source

Class

Mapping Target

Redefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Redefinition::redefinedFeature () : Feature [1]

```
 SYSML2::AttributeUsage.allInstances()
->any(m | m.qualifiedName = 'SysMLv1Library::ViewpointData::languages')
```

7.8.6.3.43 ViewpointLanguagesMetadataReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Class

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
 Set{ViewpointLanguagesMetadataRedefinition_Mapping.getMapped(from),
ViewpointLanguagesMetadataFeatureValue_Mapping.getMapped(from)}
```

7.8.6.3.44 ViewpointMetadataFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Class

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
 SYSML2::MetadataDefinition.allInstances()
->any(m | m.qualifiedName = 'SysMLv1Library::ViewpointData')
```

7.8.6.3.45 ViewpointLanguagesMetadataOperatorExpression_Mapping

Description

The mapping class creates the operator expression for the list of languages of the SysML::ModelElements::Viewpoint mapping.

General Mappings

GenericToOperatorExpression_Mapping

Mapping Source

Class

Mapping Target

OperatorExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- OperatorExpression::operator () : String [1]

```
 ','
```

- OperatorExpression::ownedRelationship () : Relationship [0..*]

```
 Helper.getTagValueAsStringColl(from, 'SysML::ModelElements::Viewpoint', 'language')
->collect(e | StringParameterMembership_Factory.create(e))
```

7.8.6.3.46 ViewpointMetadataOwningMembership_Mapping

Description

Creates a owning membership relationship for *ownedMemberElement()*.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Class

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

```
ViewpointMetadataUsage_Mapping.getMapped(from)
```

7.8.6.3.47 ViewpointMetadataUsage_Mapping

Description

The mapping class creates the metadata usage element for the SysML::ModelElements::Viewpoint mapping.

General Mappings

GenericToMetadataUsage_Mapping

Mapping Source

Class

Mapping Target

MetadataUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- MetadataUsage::ownedRelationship () : Relationship [0..*]

```
 Set{ViewpointMetadataFeatureTyping_Mapping.getMapped(from),
ViewpointLanguagesMetadataFeatureMembership_Mapping.getMapped(from),
ViewpointPresentationsMetadataFeatureMembership_Mapping.getMapped(from)}
```

7.8.6.3.48 ViewpointPresentationsMetadataFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Class

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
ViewpointPresentationsMetadataReferenceUsage_Mapping.getMapped(from)
```

7.8.6.3.49 ViewpointPresentationsMetadataFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Class

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

```
ViewpointPresentationsMetadataOperatorExpression_Mapping.getMapped(from)
```

7.8.6.3.50 ViewpointPresentationsMetadataOperatorExpression_Mapping

Description

The mapping class creates the operator expression for the list of presentations of the SysML::ModelElements::Viewpoint mapping.

General Mappings

GenericToOperatorExpression_Mapping

Mapping Source

Class

Mapping Target

OperatorExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- OperatorExpression::ownedRelationship () : Relationship [0..*]

```
Helper.getTagValueAsStringColl(from,
   'SysML::ModelElements::Viewpoint', 'presentation')
   ->collect(e | StringParameterMembership_Factory.create(e))
```

- OperatorExpression::operator () : String [1]

```
','
```

7.8.6.3.51 ViewpointPresentationsMetadataRedefinition_Mapping

Description

Creates a redefinition relationship for the *redefiningFeature()* and the *redefinedFeature()*.

General Mappings

GenericToRedefinition_Mapping

Mapping Source

Class

Mapping Target

Redefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Redefinition::redefinedFeature () : Feature [1]

```
 SYSML2::AttributeUsage.allInstances()
->any(m | m.qualifiedName = 'SysMLv1Library::ViewpointData::presentations')
```

7.8.6.3.52 ViewpointPresentationsMetadataReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Class

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
 Set{ViewpointPresentationsMetadataRedefinition_Mapping.getMapped(from),
ViewpointPresentationsMetadataFeatureValue_Mapping.getMapped(from)}
```

7.8.6.3.53 ViewpointRenderingFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Class

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
ViewpointRenderingUsage_Mapping.getMapped(from)
```

7.8.6.3.54 ViewpointRenderingUsage_Mapping

Description

The mapping class creates the rendering usage element for the SysML::ModelElements::Viewpoint mapping class.

General Mappings

GenericToPartUsage_Mapping

Mapping Source

Class

Mapping Target

RenderingUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- RenderingUsage::ownedRelationship () : Relationship [0..*]

```
 from.ownedOperation
->select( o | Helper.hasStereotypeApplied(o, 'Create') )
->collect( e |
    ViewpointRenderingUsageActionUsageFeatureMembership_Mapping.getMapped(e))
```

7.8.6.3.55 ViewpointRenderingUsageActionUsage_Mapping

Description

The mapping class creates the action usage element for the rendering usage element for the SysML::ModelElements::Viewpoint mapping class.

General Mappings

GenericToActionUsage_Mapping

Mapping Source

Class

Mapping Target

ActionUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ActionUsage::ownedRelationship () : Relationship [0..*]

  ```
  Set{ViewpointRenderingUsageActionUsageFeatureTyping_Mapping.getMapped(from)}
  ```

7.8.6.3.56 ViewpointRenderingUsageActionUsageFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Class

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

  ```
  ViewpointRenderingUsageActionUsage_Mapping.getMapped(from)
  ```

7.8.6.3.57 ViewpointRenderingUsageActionUsageFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Class

Mapping Target

FeatureTyping

Owned Mappings

(none)

7.8.6.3.58 ViewpointRequirementConstraintMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Class

Mapping Target

RequirementConstraintMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- RequirementConstraintMembership::ownedMemberFeature () : Feature [1]

```
ViewpointConstraintUsage_Mapping.getMapped(from)
```

7.8.6.3.59 ViewpointSatisfyFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Class

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
ViewpointSatisfyRequirementUsage_Mapping.getMapped(from)
```

7.8.6.3.60 ViewpointSatisfyRequirementUsage_Mapping

Description

The mapping class creates the satisfy requirement usage element for the SysML::ModelElements::Viewpoint mapping.

General Mappings

GenericToRequirementUsage_Mapping

Mapping Source

Class

Mapping Target

SatisfyRequirementUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- SatisfyRequirementUsage::ownedRelationship () : Relationship [0..*]

```
 Set{ViewpointSatisfyRequirementUsageReferenceSubsetting_Mapping.getMapped(from),
EmptySubjectMembership_Factory.create(),
ReturnParameterFeatureMembership_Factory.create()}
```

7.8.6.3.61 ViewpointSatisfyRequirementUsageReferenceSubsetting_Mapping

Description

Creates a subsetting relationship for the *subsettingFeature()* and the *subsettedFeature()*.

General Mappings

GenericToReferenceSubsetting_Mapping

Mapping Source

Class

Mapping Target

ReferenceSubsetting

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ReferenceSubsetting::referencedFeature () : Feature [1]

```
ViewpointViewpointUsage_Mapping.getMapped(from)
```

7.8.6.3.62 ViewpointViewpointUsage_Mapping

Description

The mapping class creates the embedded viewpoint usage for the SysML::ModelElements::Viewpoint mapping.

General Mappings

GenericToUsage_Mapping

Mapping Source

Class

Mapping Target

ViewpointUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ViewpointUsage::ownedRelationship () : Relationship [0..*]

```
 Helper.getTagValueAsElementColl(
    from, 'SysML::ModelElements::Viewpoint', 'concernList')
->collect(e | ViewpointFramedConcernMembership_Mapping.getMapped(e))
->including(ViewpointMetadataOwningMembership_Mapping.getMapped(from))
->including(EmptySubjectMembership_Factory.create())
->including(ViewpointRequirementConstraintMembership_Mapping.getMapped(from))
```

- ViewpointUsage::declaredName () : String [0..1]

```
from.name.substring(1,1).toLowerCase() + from.name.substring(2, from.name.size())
```

7.8.6.3.63 ViewpointViewpointUsageFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Class

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

  ```
  ViewpointViewpointUsage_Mapping.getMapped(from)
  ```

7.8.7 PortsAndFlows

This chapter lists all mapping specifications of SysML::PortsAndFlows model elements.

7.8.7.1 Overview

The following table gives an overview of which SysML v2 elements the SysML::Ports&Flows elements are transformed with which mapping class. The mapping details are in 7.8.7.3.

The justifications for the elements without mapping are given in 7.8.7.2.

Table 33. List of all mappings

SysML v1 Concept	SysML v2 Concept
AcceptChangeStructuralFeatureEventAction	AcceptActionUsage
AddFlowPropertyValueOnNestedPortAction	
ChangeStructuralFeatureEvent	
DirectedFeature	PerformActionUsage
FlowProperty	
FullPort	PartUsage
InterfaceBlock	PortDefinition
InvocationOnNestedPortAction	
ItemFlow	FlowConnectionUsage
ProxyPort	
TriggerOnNestedPort	
~InterfaceBlock	

7.8.7.2 SysML::Ports&Flows elements not mapped

Table 34. List of SysML v1 elements not mapped of this section

SysML v1 Concept	Rationale
AddFlowPropertyValueOnNestedPortAction	Mapping is not specified yet.

SysML v1 Concept	Rationale
ChangeStructuralFeatureEvent	Mapping is not specified yet.
FlowProperty	Mapping is not specified yet.
InvocationOnNestedPortAction	Mapping is not specified yet.
TriggerOnNestedPort	Mapping is not specified yet.
~InterfaceBlock	Mapping is not specified yet.

7.8.7.3 Mapping Specifications

7.8.7.3.1 AcceptChangeStructuralFeatureEventAction_Mapping

Description

The SysML::PortsAndFlows::AcceptChangeStructuralFeatureEventAction element is mapped to SysML v2 AcceptActionUsage. The details of the mapping are not defined 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.

7.8.7.3.2 CommonFullPort_Mapping

Description

The abstract mapping class is the base class of the mapping classes for the SysML::Ports&Flows::FullPort mappings.

General Mappings

PropertyCommon_Mapping

Mapping Source

Port

Mapping Target

PartUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific 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))
```

7.8.7.3.3 FeatureDirectionKind

The SysML::Ports&Flows::FeatureDirectionKind enumeration is mapped to the SysML v2 FeatureDirectionKind enumeration according to the following mapping table.

SysML::Ports&Flows::FeatureDirectionKind	SysML v2 FeatureDirectionKind
provided	out
required	in
provreqd	inout

7.8.7.3.4 FlowDirectionKind

The SysML::Ports&Flows::FlowDirectionKind enumeration is mapped to the SysML v2 FeatureDirectionKind enumeration according to the following mapping table.

SysML::Ports&Flows::FlowDirectionKind	SysML v2 FeatureDirectionKind
out	out
in	in
inout	inout

7.8.7.3.5 FullPort_Mapping

Description

A SysML::Ports&Flows::FullPort element is mapped to a part usage in SysML v2 with metadata that marks the part usage as a full port. The metadata is defined in the SysML v1 library for SysML v2.

The mapping class FullPortUntyped_Mapping does the same for full ports that have no type.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
part sysMLv1FullPort : SysMLv1Block {SysMLv1Library::PortData {isFullPort = true;}}
```

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 src.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.

7.8.7.3.6 FullPortMetadata_Mapping

Description

Create the metadata usage element to annotate a port with the information that its SysML v1 mapping source element is a SysML v1 full port element.

General Mappings

GenericToMetadataUsage_Mapping

Mapping Source

Port

Mapping Target

MetadataUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- MetadataUsage::ownedRelationship () : Relationship [0..*]

```
 Set{FullPortMetadataFeatureTyping_Mapping.getMapped(from),
FullPortMetadataFeatureMembership_Mapping.getMapped(from)}
```

7.8.7.3.7 FullPortMetadataFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Port

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
 FullPortMetadataReferenceUsage_Mapping.getMapped(from)
```

7.8.7.3.8 FullPortMetadataFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Port

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
 SYSML2::MetadataDefinition.allInstances()
->any(m | m.qualifiedName = 'SysMLv1Library::PortData')
```

7.8.7.3.9 FullPortMetadataOwningMembership_Mapping

Description

Creates a owning membership relationship for *ownedMemberElement()*.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Port

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

  ```
  FullPortMetadata_Mapping.getMapped(from)
  ```

7.8.7.3.10 FullPortMetadataReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Port

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
 Set{FullPortMetadataReferenceUsageRedefinition_Mapping.getMapped(from),
FullPortMetadataReferenceUsageFeatureValue_Mapping.getMapped(from)}
```

7.8.7.3.11 FullPortMetadataReferenceUsageFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Port

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

```
LiteralBoolean_Factory.create(true)
```

7.8.7.3.12 FullPortMetadataReferenceUsageRedefinition_Mapping

Description

Creates a redefinition relationship for the *redefiningFeature()* and the *redefinedFeature()*.

General Mappings

GenericToRedefinition_Mapping

Mapping Source

Port

Mapping Target

Redefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Redefinition::redefinedFeature () : Feature [1]

```
 SYSML2::AttributeUsage.allInstances()
->any(m | m.qualifiedName = 'SysMLv1Library::PortData::isFullPort')
```

7.8.7.3.13 FullPortUntyped_Mapping

Description

A SysML::Ports&Flows::FullPort element is mapped to a part usage in SysML v2 with metadata that marks the part usage as a full port. The metadata is defined in the SysML v1 library for SysML v2.

The mapping class FullPort_Mapping does the same for full ports with a type.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
part sysMLv1FullPort {SysMLv1Library::PortData {isFullPort = true;}}
```

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:

```
src.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.

7.8.7.3.14 InterfaceBlock_Mapping

Description

A SysML::Ports&Flows::InterfaceBlock element is mapped to a SysML v2 PortDefinition.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
port def SysMLv1InterfaceBlock;
```

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(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.

7.8.7.3.15 ItemFlow_Mapping

Description

A SysML::Ports&Flows::ItemFlow element is mapped to a SysML v2 FlowConnectionUsage.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
part sysMLv1PartProperty1 : SysMLv1Block1;
part sysMLv1PartProperty2 : SysMLv1Block2;

connection sysMLv1Connector connect sysMLv1PartProperty1 to sysMLv1PartProperty2;
message sysMLv1ItemFlow :> sysMLv1Connector
    of SysMLv1Block3
    from sysMLv1PartProperty1 to sysMLv1PartProperty2;

part def SysMLv1Block1;
part def SysMLv1Block2;
part def SysMLv1Block3;
```

General Mappings

InformationFlow_Mapping

Mapping Source

InformationFlow

Mapping Target

FlowConnectionUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
 Helper.hasStereotypeApplied(src, 'SysML::Ports&Flows::ItemFlow')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FlowConnectionUsage::ownedRelationship () : Relationship [0..*]

  ```
   Set{ItemFlowFeatureMembership_Mapping.getMapped(from),
  ItemFlowSourceEndFeatureMembership_Mapping.getMapped(from),
  ItemFlowTargetEndFeatureMembership_Mapping.getMapped(from)}
  ```

- FlowConnectionUsage::source () : Element [0..*]

  ```
  NamedElementMain_Mapping.getMappedColl(from.informationSource)
  ```

- FlowConnectionUsage::target () : Element [0..*]

  ```
  NamedElementMain_Mapping.getMappedColl(from.informationTarget)
  ```

7.8.7.3.16 ItemFlowFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

InformationFlow

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
ItemFlowItemFeature_Mapping.getMapped(from)
```

7.8.7.3.17 ItemFlowItemFeature_Mapping

Description

The mapping class creates the item feature element for the item that flows.

General Mappings

GenericToFeature_Mapping

Mapping Source

InformationFlow

Mapping Target

ItemFeature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ItemFeature::ownedRelationship () : Relationship [0..*]

```
Set{ItemFlowItemFeatureTyping_Mapping.getMapped(from)}
```

7.8.7.3.18 ItemFlowItemFeatureTyping_Mapping

Description

Currently, only one conveyed item is supported

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

InformationFlow

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific 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
```

7.8.7.3.19 ItemFlowSourceEndFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToEndFeatureMembership_Mapping

Mapping Source

InformationFlow

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
ItemFlowSourceFeature_Mapping.getMapped(from)
```

7.8.7.3.20 ItemFlowSourceFeature_Mapping

Description

The mapping class creates the source item flow end for the item flow mapping.

General Mappings

GenericToFeature_Mapping

Mapping Source

InformationFlow

Mapping Target

ItemFlowEnd

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ItemFlowEnd::isEnd () : Boolean [1]

```
true
```

- ItemFlowEnd::ownedRelationship () : Relationship [0..*]

```
Set{ItemFlowSourceFeatureSubsetting_Mapping.getMapped(from)}
```

7.8.7.3.21 ItemFlowSourceFeatureSubsetting_Mapping

Description

Creates a subsetting relationship for the *subsettingFeature()* and the *subsettedFeature()*.

General Mappings

GenericToSubsetting_Mapping

Mapping Source

InformationFlow

Mapping Target

Subsetting

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Subsetting::subsettedFeature () : Feature [1]

```
from.source.get(0)
```

7.8.7.3.22 ItemFlowTargetEndFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToEndFeatureMembership_Mapping

Mapping Source

InformationFlow

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
ItemFlowTargetFeature_Mapping.getMapped(from)
```

7.8.7.3.23 ItemFlowTargetFeature_Mapping

Description

The mapping class creates the target item flow end for the item flow mapping.

General Mappings

GenericToFeature_Mapping

Mapping Source

InformationFlow

Mapping Target

ItemFlowEnd

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ItemFlowEnd::isEnd () : Boolean [1]

```
true
```

- ItemFlowEnd::ownedRelationship () : Relationship [0..*]

```
Set{ItemFlowTargetFeatureSubsetting_Mapping.getMapped(from)}
```

7.8.7.3.24 ItemFlowTargetFeatureSubsetting_Mapping

Description

Creates a subsetting relationship for the *subsettingFeature()* and the *subsettedFeature()*.

General Mappings

GenericToSubsetting_Mapping

Mapping Source

InformationFlow

Mapping Target

Subsetting

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Subsetting::subsettedFeature () : Feature [1]

```
from.target.get(0)
```

7.8.7.3.25 OperationDirectedFeature_Mapping

Description

The mapping class sets the direction of the perform action usage if the SysML v1 mapping source operation has the stereotype SysML::Ports&Flows::DirectedFeature applied.

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

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- PerformActionUsage::direction () : FeatureDirectionKind [0..1]

```
 Helper.getKerMLFeatureDirectionKind(
Helper.getTagValueAsElement(
from,'SysML::Ports&Flows::DirectedFeature', 'featureDirection'
))
```

7.8.8 Requirements

This chapter lists all mapping specifications of SysML::Requirements model elements.

7.8.8.1 Overview

The following table gives an overview of which SysML v2 elements the SysML::Requirements elements are transformed with which mapping class. The mapping details are in 7.8.8.3.

The justifications for the elements without mapping are given in 7.8.8.2.

Table 35. List of all mappings

SysML v1 Concept	SysML v2 Concept
Copy	
DeriveReqt	ConnectionUsage
Refine	Dependency
Requirement	RequirementUsage
Satisfy	SatisfyRequirementUsage
TestCase	VerificationCaseDefinition
Trace	Dependency
Verify	RequirementVerificationMembership

7.8.8.2 SysML::Requirements elements not mapped

Table 36. List of SysML v1 elements not mapped of this section

SysML v1 Concept	Rationale
Copy	The copy relationship is not covered by SysML v2.

7.8.8.3 Mapping Specifications

7.8.8.3.1 DeriveReqt_Mapping

Description

A SysML::Requirements::DeriveReqt relationship is mapped to a SysML v2 DerivationConnections::Derivation model library element.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
requirement <'id1'> SysMLv1Requirement {
        doc /*
               * requirement text
               */
}
requirement <'id2'> SysMLv1RequirementDerived {
        doc /*
               * requirement text
               */
}
connection : DerivationConnections::Derivation
    connect SysMLv1RequirementDerived to SysMLv1Requirement;
```

General Mappings

Abstraction_Mapping
GenericToConnectionUsage_Mapping

Mapping Source

Abstraction

Mapping Target

ConnectionUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
Helper.hasStereotypeApplied(src, 'SysML::Requirements::DeriveReqt')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ConnectionUsage::ownedRelationship () : Relationship [0..*]

```
 Set{DeriveReqtFeatureTyping_Mapping.getMapped(from),
DeriveReqtSourceEndFeatureMembership_Mapping.getMapped(from),
DeriveReqtTargetEndFeatureMembership_Mapping.getMapped(from)}
```

7.8.8.3.2 DeriveReqtFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Dependency

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
 SYSML2::ConnectionDefinition.allInstances()
->any(m | m.qualifiedName = 'DerivationConnections::Derivation')
```

7.8.8.3.3 DeriveReqtSourceEndFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToEndFeatureMembership_Mapping

Mapping Source

Dependency

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
DeriveReqtSourceFeature_Mapping.getMapped(from)
```

7.8.8.3.4 DeriveReqtSourceFeature_Mapping

Description

The mapping class creates the source feature of the ConnectionUsage relationship for the mapping of the SysML v1 deriveReqt relationship.

General Mappings

GenericToFeature_Mapping

Mapping Source

Dependency

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Feature::ownedRelationship () : Relationship [0..*]

```
Set{DeriveReqtSourceFeatureReferenceSubsetting_Mapping.getMapped(from)}
```

7.8.8.3.5 DeriveReqtSourceFeatureReferenceSubsetting_Mapping

Description

Creates a subsetting relationship for the *subsettingFeature()* and the *subsettedFeature()*.

General Mappings

GenericToReferenceSubsetting_Mapping

Mapping Source

Dependency

Mapping Target

ReferenceSubsetting

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ReferenceSubsetting::referencedFeature () : Feature [1]

  ```
  from.client->any(c | true)
  ```

7.8.8.3.6 DeriveReqtTargetEndFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToEndFeatureMembership_Mapping

Mapping Source

Dependency

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
DeriveReqtTargetFeature_Mapping.getMapped(from)
```

7.8.8.3.7 DeriveReqtTargetFeature_Mapping

Description

The mapping class creates the target feature of the ConnectionUsage relationship for the mapping of the SysML v1 deriveReqt relationship.

General Mappings

GenericToFeature_Mapping

Mapping Source

Dependency

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Feature::ownedRelationship () : Relationship [0..*]

```
Set{DeriveReqtTargetFeatureReferenceSubsetting_Mapping.getMapped(from)}
```

7.8.8.3.8 DeriveReqtTargetFeatureReferenceSubsetting_Mapping

Description

Creates a subsetting relationship for the *subsettingFeature()* and the *subsettedFeature()*.

General Mappings

GenericToReferenceSubsetting_Mapping

Mapping Source

Dependency

Mapping Target

ReferenceSubsetting

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ReferenceSubsetting::referencedFeature () : Feature [1]

```
from.supplier->any(c | true)
```

7.8.8.3.9 Refine_Mapping

Description

A SysML::Requirements::Refine relationship is mapped to a SysML v2 Dependency relationship annotated with a metadata usage tagging it as a former SysML v1 refine relationship.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
requirement <'id1'> SysMLv1Requirement {
        doc /*
               * requirement text
               */
}
use case def SysMLv1UseCase;

dependency from SysMLv1UseCase to SysMLv1Requirement {
        @SysMLv1Library::RefineData {isRefine = true;}
}
```

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::Refine')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Dependency::ownedRelationship () : Relationship [0..*]

```
 ElementOwnership_Mapping.getMappedColl(from.ownedComment)
->including(RefineAnnotation_Mapping.getMapped(from))
```

7.8.8.3.10 RefineAnnotation_Mapping

Description

The mapping class creates the annotation relationship for the SysML::Requirements::Refine mapping.

General Mappings

GenericToAnnotation_Mapping

Mapping Source

Abstraction

Mapping Target

Annotation

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Annotation::annotatingElement () : AnnotatingElement [1]

```
RefineMetadataUsage_Mapping.getMapped(from)
```

7.8.8.3.11 RefineMetadataFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Abstraction

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
RefineMetadataReferenceUsage_Mapping.getMapped(from)
```

7.8.8.3.12 RefineMetadataReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Abstraction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
 Set{RefineMetadataReferenceUsageRedefinition_Mapping.getMapped(from),
RefineMetadataReferenceUsageFeatureValue_Mapping.getMapped(from)}
```

7.8.8.3.13 RefineMetadataReferenceUsageFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Abstraction

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

```
LiteralBoolean_Factory.create(true)
```

7.8.8.3.14 RefineMetadataReferenceUsageRedefinition_Mapping

Description

Creates a redefinition relationship for the *redefiningFeature()* and the *redefinedFeature()*.

General Mappings

GenericToRedefinition_Mapping

Mapping Source

Abstraction

Mapping Target

Redefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Redefinition::redefinedFeature () : Feature [1]

```
 SYSML2::AttributeUsage.allInstances()
->any(m | m.qualifiedName = 'SysMLv1Library::RefineData::isRefine')
```

7.8.8.3.15 RefineMetadataUsage_Mapping

Description

Create the metadata usage element to annotate a dependency relationship with the information that its SysML v1 mapping source element is a SysML v1 refine relationship.

General Mappings

GenericToMetadataUsage_Mapping

Mapping Source

Abstraction

Mapping Target

MetadataUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- MetadataUsage::ownedRelationship () : Relationship [0..*]

```
 Set{RefineMetadataUsageFeatureTyping_Mapping.getMapped(from),
RefineMetadataFeatureMembership_Mapping.getMapped(from)}
```

7.8.8.3.16 RefineMetadataUsageFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Abstraction

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
 SYSML2::MetadataDefinition.allInstances()
->any(m | m.qualifiedName = 'SysMLv1Library::RefineData')
```

7.8.8.3.17 Requirement_Mapping

Description

A SysML::Requirement is mapped to a SysML v2 RequirementUsage.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
requirement <'id1'> SysMLv1Requirement {
        doc /*
                      * requirement text
               */

        requirement <'id2'> SysMLv1NestedRequirement {
                doc /*
                             * requirement text
                                */
        }
}
```

General Mappings

NamedElementMain_Mapping
GenericToRequirementUsage_Mapping

Mapping Source

Class

Mapping Target

RequirementUsage

Owned Mappings

(none)

Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
 Helper.isRequirement(src)
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- RequirementUsage::reqId () : String [1]

  ```
   let stereotype: UML::Stereotype = Helper.getRequirementStereotype(from) in
  Helper.getTagValueAsString(from, stereotype.qualifiedName, 'id')
  ```

- RequirementUsage::ownedRelationship () : Relationship [0..*]

  ```
  from.ownedElement->collect(e | ElementOwningMembership_Mapping.getMapped(e))
  ->including(CommonReturnParameterReferenceUsageMembership_Mapping.getMapped(from))
  ```

```
        ->including(RequirementDocumentationMembership_Mapping.getMapped(from))
        ->including(RequirementSubjectMembership_Mapping.getMapped(from))
```

7.8.8.3.18 RequirementDocumentation_Mapping

Description

The mapping class creates a Comment contained in a Requirement which contains the SysML::Requirements::AbstractRequirement::text property.

General Mappings

GenericToDocumentation_Mapping

Mapping Source

Class

Mapping Target

Documentation

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Documentation::body () : String [1]

```
 let stereotype: UML::Stereotype = Helper.getRequirementStereotype(from) in
Helper.getTagValueAsString(from, stereotype.qualifiedName, 'text')
```

7.8.8.3.19 RequirementDocumentationMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Class

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

```
RequirementDocumentation_Mapping.getMapped(from)
```

7.8.8.3.20 RequirementSubject_Mapping

Description

The mapping class creates the subject reference usage element of the requirement. It is not used since the concept does not exist SysML v1.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Class

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ReferenceUsage::direction () : FeatureDirectionKind [0..1]

```
KerML::FeatureDirectionKind::_'in'
```

7.8.8.3.21 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

Class

Mapping Target

SubjectMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- SubjectMembership::ownedMemberParameter () : Feature [0..1]

```
RequirementSubject_Mapping.getMapped(from)
```

7.8.8.3.22 Satisfy_Mapping

Description

A SysML::Requirements::Satisfy relationship is mapped to a SysML v2 SatisfyRequirementUsage.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
// satisfy relationship from a block
part def SysMLv1Block {
        part sysMLv1PartProperty;
}
requirement <'ReqId1'> SysMLv1Requirement { doc /* requirement text */ }

ref :SysMLv1Block = all SysMLv1Block {
        satisfy requirement SysMLv1Requirement by self;
}

// satisfy relationship from a part property
satisfy SysMLv1Requirement by sysML1BlockUsage.sysMLv1PartProperty {
        sysMLv1BlockUsage : SysMLv1Block;
}
```

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')
    endif
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- SatisfyRequirementUsage::ownedRelationship () : Relationship [0..*]

```
let relationships : Set(KerML::Relationship) =
    ElementOwnership_Mapping.getMappedColl(from.ownedComment)
->including(SatisfyFeatureTyping_Mapping.getMapped(from))
->including(SatisfySubjectSubjectMembership_Mapping.getMapped(from))
->including(CommonReturnParameterReferenceUsageMembership_Mapping.getMapped(from)) in
if from.client->any(c | true).oclIsKindOf(UML::Property) then
    relationships
    ->including(SatisfyReferenceUsageFeatureMembership_Mapping.getMapped(from))
else
    relationships
endif
```

7.8.8.3.23 SatisfyReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Abstraction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ReferenceUsage::declaredName () : String [0..1]

```
 from.client
->any(c | true).owner.name.substring(1,1).toLowerCase()
+ from.client
->any(c | true).owner.name.
substring(2,from.client->any(c | true).owner.name.size())
+ 'SatisfyClientUsage'
```

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
 Set{SatisfyReferenceUsageFeatureTyping_Mapping.getMapped(from)}
```

7.8.8.3.24 SatisfyReferenceUsageFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Abstraction

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
SatisfyReferenceUsage_Mapping.getMapped(from)
```

7.8.8.3.25 SatisfySubjectReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Abstraction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
Set{SatisfySubjectReferenceUsageFeatureValue_Mapping.getMapped(from)}
```

- ReferenceUsage::direction () : FeatureDirectionKind [0..1]

```
KerML::FeatureDirectionKind::_'in'
```

7.8.8.3.26 SatisfySubjectReferenceUsageValue_Mapping

Description

The mapping class create the feature reference expression for the subject of the SatisfyRequirementUsage element.

General Mappings

GenericToFeatureReferenceExpression_Mapping

Mapping Source

Abstraction

Mapping Target

FeatureReferenceExpression

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureReferenceExpression::ownedRelationship () : Relationship [0..*]

```
 Set{SatisfySubjectReferenceUsageValueOwningMembership_Mapping.getMapped(from),
ReturnParameterFeatureMembership_Factory.create()}
```

7.8.8.3.27 SatisfySubjectReferenceUsageValueFeature_Mapping

Description

The mapping class creates the feature element for the feature reference expression of the subject of the SatisRequirementUsage element.

General Mappings

GenericToFeature_Mapping

Mapping Source

Abstraction

Mapping Target

Feature

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Feature::ownedRelationship () : Relationship [0..*]

```
 Set{SatisfySubjectReferenceUsageFeatureChaining_Mapping.getMapped(from),
SatisfySubjectReferenceUsageValueFeatureChainingProperty_Mapping.getMapped(from)}
```

7.8.8.3.28 SatisfySubjectReferenceUsageFeatureChaining_Mapping

Description

The mapping class creates the feature chaining element from SysML v2 SatisfyRequirementUsage's reference usage element.

General Mappings

GenericToFeatureChaining_Mapping

Mapping Source

Abstraction

Mapping Target

FeatureChaining

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureChaining::chainingFeature () : Feature [1]

```
SatisfyReferenceUsage_Mapping.getMapped(from)
```

7.8.8.3.29 SatisfySubjectReferenceUsageValueFeatureChainingProperty_Mapping

Description

The mapping class creates the feature chaining element from the source element of the SysML v1 satisfy relationship.

General Mappings

GenericToFeatureChaining_Mapping

Mapping Source

Abstraction

Mapping Target

FeatureChaining

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureChaining::chainingFeature () : Feature [1]

```
from.client->any(c | true)
```

7.8.8.3.30 SatisfySubjectReferenceUsageFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Abstraction

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

  ```
  SatisfySubjectReferenceUsageValue_Mapping.getMapped(from)
  ```

7.8.8.3.31 SatisfySubjectReferenceUsageValueOwningMembership_Mapping

Description

Creates a owning membership relationship for *ownedMemberElement()*.

General Mappings

GenericToOwningMembership_Mapping

Mapping Source

Abstraction

Mapping Target

OwningMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- OwningMembership::ownedMemberElement () : Element [1]

  ```
  SatisfySubjectReferenceUsageValueFeature_Mapping.getMapped(from)
  ```

7.8.8.3.32 SatisfySubjectSubjectMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

GenericToSubjectMembership_Mapping

Mapping Source

Abstraction

Mapping Target

SubjectMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- SubjectMembership::ownedMemberParameter () : Feature [1]

```
SatisfySubjectReferenceUsage_Mapping.getMapped(from)
```

7.8.8.3.33 SatisfyFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Abstraction

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
from.supplier->any(s | true)
```

7.8.8.3.34 SatisfyReferenceUsageFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Abstraction

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
from.client->any(c | true).owner
```

7.8.8.3.35 TestCaseActivity_Mapping

Description

A SysML::Requirements::TestCase applied to an activity is mapped to a SysML v2 VerificationCaseDefinition element.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
verification def SysMLv1ActivityTestCase {
        return verdict : VerificationCases::VerdictKind;
}
```

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(src, 'SysML::Requirements::TestCase')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific 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::Parameter) and
    (e.oclAsType(UML::Parameter).type.name = 'VerdictKind')) in
let parameters : Set(UML::Paramter) =
    ((from.ownedElement->select(e | e.oclIsKindOf(UML::Parameter))) -
    verdictParameter) in
let verifyRelationships : Set(UML::Abstraction) =
    from.clientDependency
    ->select( v |
        Helper.hasStereotypeApplied(v, 'SysML::Requirements::Verify')) in
relationships
->union(parameters->collect(p | ParameterMembership_Mapping.getMapped(p)))
->union(verdictParameter
    ->collect(vp |
        TestCaseActivityReturnParameterMembership_Mapping.getMapped(vp)))
->including(CaseSubjectMembership_Mapping.getMapped(from))
->including(CaseObjectiveMembership_Mapping.getMapped(from))
->union(verifyRelationships->collect(v | Verify_Mapping.getMapped(v)))
```

7.8.8.3.36 TestCaseActivityReturnParameterMembership_Mapping

Description

Creates a membership relationship for *memberElement()*.

General Mappings

ParameterMembership_Mapping

Mapping Source

Parameter

Mapping Target

ReturnParameterMembership

Owned Mappings

(none)

7.8.8.3.37 TestCaseVerifyObjectiveMembership_Mapping

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ownedMemberFeature () : Feature [1]

```
TestCaseVerifyObjectiveRequirementUsage_Mapping.getMapped(from)
```

7.8.8.3.38 TestCaseVerifyObjectiveRequirementUsage_Mapping

Description

The mapping class creates the objective requirements usage of the SysML v2 test case.

General Mappings

No general mappings.

Mapping Source

Abstraction

Mapping Target

No target element.

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ownedRelationship () : Relationship [0..*]

```
Set{Verify_Mapping.getMapped(from)}
```

7.8.8.3.39 TestCaseVerifyRequirementUsageReferenceSubsetting_Mapping

Description

Creates a subsetting relationship for the *subsettingFeature()* and the *subsettedFeature()*.

General Mappings

GenericToSubsetting_Mapping

Mapping Source

Abstraction

Mapping Target

ReferenceSubsetting

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ReferenceSubsetting::referencedFeature () : Feature [1]

```
from.supplier->get(0)
```

7.8.8.3.40 TestCaseVerifyRequirementUsage_Mapping

Description

The mapping class creates the requirements usage of the SysML v2 test case for the verify relationship.

General Mappings

GenericToUsage_Mapping

Mapping Source

Abstraction

Mapping Target

RequirementUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- RequirementUsage::ownedRelationship () : Relationship [0..*]

```
 Set{TestCaseVerifyRequirementUsageReferenceSubsetting_Mapping.getMapped(from),
CaseSubjectMembership_Mapping.getMapped(from.client),
CommonReturnParameterReferenceUsageMembership_Mapping.getMapped(from)}
```

7.8.8.3.41 Trace_Mapping

Description

A SysML::Requirements::Trace relationship is mapped to a SysML v2 Dependency relationship annotated with a metadata usage tagging it as a former SysML v1 trace relationship.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
requirement <'id1'> SysMLv1Requirement1 {
        doc /*
               * requirement text
               */
}
requirement <'id2'> SysMLv1Requirement2 {
        doc /*
               * requirement text
               */
}
dependency from SysMLv1Requirement1 to SysMLv1Requirement2 {
        @SysMLv1Library::TraceData {isTrace = true;}
}
```

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::Trace')
```

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Dependency::ownedRelationship () : Relationship [0..*]

```
 ElementOwnership_Mapping.getMappedColl(from.ownedComment)
->including(TraceAnnotation_Mapping.getMapped(from))
```

7.8.8.3.42 TraceAnnotation_Mapping

Description

The mapping class creates the annotation relationship for the SysML::Requirements::Trace mapping.

General Mappings

GenericToAnnotation_Mapping

Mapping Source

Abstraction

Mapping Target

Annotation

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Annotation::annotatingElement () : AnnotatingElement [1]

```
TraceMetadataUsage_Mapping.getMapped(from)
```

7.8.8.3.43 TraceMetadataFeatureMembership_Mapping

Description

Creates a feature membership relationship for *ownedMemberFeature()*.

General Mappings

GenericToFeatureMembership_Mapping

Mapping Source

Abstraction

Mapping Target

FeatureMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature () : Feature [1]

```
TraceMetadataReferenceUsage_Mapping.getMapped(from)
```

7.8.8.3.44 TraceMetadataReferenceUsage_Mapping

Description

Creates a reference usage.

General Mappings

GenericToReferenceUsage_Mapping

Mapping Source

Abstraction

Mapping Target

ReferenceUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- ReferenceUsage::ownedRelationship () : Relationship [0..*]

```
 Set{TraceMetadataReferenceUsageRedefinition_Mapping.getMapped(from),
TraceMetadataReferenceUsageFeatureValue_Mapping.getMapped(from)}
```

7.8.8.3.45 TraceMetadataReferenceUsageFeatureValue_Mapping

Description

Creates a feature value relationship.

General Mappings

GenericToFeatureValue_Mapping

Mapping Source

Abstraction

Mapping Target

FeatureValue

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureValue::value () : Expression [1]

```
LiteralBoolean_Factory.create(true)
```

7.8.8.3.46 TraceMetadataReferenceUsageRedefinition_Mapping

Description

Creates a redefinition relationship for the *redefiningFeature()* and the *redefinedFeature()*.

General Mappings

GenericToRedefinition_Mapping

Mapping Source

Abstraction

Mapping Target

Redefinition

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- Redefinition::redefinedFeature () : Feature [1]

```
 SYSML2::AttributeUsage.allInstances()
->any(m | m.qualifiedName = 'SysMLv1Library::TraceData::isTrace')
```

7.8.8.3.47 TraceMetadataUsage_Mapping

Description

Create the metadata usage element to annotate a dependency relationship with the information that its SysML v1 mapping source element is a SysML v1 trace relationship.

General Mappings

GenericToMetadataUsage_Mapping

Mapping Source

Abstraction

Mapping Target

MetadataUsage

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- MetadataUsage::ownedRelationship () : Relationship [0..*]

```
 Set{TraceMetadataUsageFeatureTyping_Mapping.getMapped(from),
TraceMetadataFeatureMembership_Mapping.getMapped(from)}
```

7.8.8.3.48 TraceMetadataUsageFeatureTyping_Mapping

Description

Creates a feature typing relationship owned by the element *typedFeature()*.

General Mappings

GenericToFeatureTyping_Mapping

Mapping Source

Abstraction

Mapping Target

FeatureTyping

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- FeatureTyping::type () : Type [1]

```
 SYSML2::MetadataDefinition.allInstances()
->any(m | m.qualifiedName = 'SysMLv1Library::TraceData')
```

7.8.8.3.49 Verify_Mapping

Description

A SysML::Requirements::Verify relationship is mapped to a SysML v2 RequirementVerificationMembership relationship.

The following shows an example of what the textual SysML v2 syntax of the result of the transformation may look like.

```
requirement <'id1'> SysMLv1Requirement {
        doc /*
               * requirement text
               */
}
verification def SysMLv1TestCase {
        objective objective_SysMLv1TestCase {
                verify SysMLv1Requirement;
        }
        return verdict : VerificationCases::VerdictKind;
}
```

General Mappings

GenericToRelationship_Mapping

Mapping Source

Abstraction

Mapping Target

RequirementVerificationMembership

Owned Mappings

(none)

Applicable filters

(none)

Mapping rules

In addition to the inherited rules, the following lists the mapping class specific mapping rules for the target element properties.

- RequirementVerificationMembership::ownedRelatedElement () : Element [0..*]

```
Set{TestCaseVerifyRequirementUsage_Mapping.getMapped(from)}
```

7.8.8.3.50 Model Libraries

7.8.8.3.50.1 Verdicts

7.8.8.3.50.1.1 VerdictKind

The enumeration VerdictKind is mapped to the SysML v2 VerificationCases::VerdictKind model library element.