



Date: October 2022



# OMG Systems Modeling Language™ (SysML®)

## Annex C: SysML v1 to SysML v2 Transformation

*Version 2.0*  
*Release 2022-09*

**Submitted in response to Systems Modeling Language (SysML®) v2 RFP (ad/  
2017-11-04) 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	Thematrix Partners

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

A .....	1
B .....	3
C Annex: SysML v1 to SysML v2 Transformation.....	5
C.1 General .....	5
C.1.1 Overview .....	5
C.1.2 Mapping Approach.....	5
C.2 Mappings .....	6
C.2.1 Overview .....	6
C.2.2 Mapping Helper and Library .....	6
C.2.2.1 Helper .....	6
C.2.2.2 SysML v1 Library .....	10
C.2.3 Generic Mappings .....	11
C.2.3.1 Overview .....	11
C.2.3.2 Generic Mappings To KerML .....	11
C.2.3.2.1 Overview.....	11
C.2.3.2.2 Mapping Specifications .....	12
C.2.3.2.2.1 GenericToAnnotatingElement_Mapping.....	12
C.2.3.2.2.2 GenericToAnnotation_Mapping.....	13
C.2.3.2.2.3 GenericToAssociation_Mapping.....	14
C.2.3.2.2.4 GenericToBehavior_Mapping.....	15
C.2.3.2.2.5 GenericToClassifier_Mapping.....	16
C.2.3.2.2.6 GenericToComment_Mapping.....	17
C.2.3.2.2.7 GenericToConjugation_Mapping.....	18
C.2.3.2.2.8 GenericToConnector_Mapping.....	19
C.2.3.2.2.9 GenericToDocumentation_Mapping.....	20
C.2.3.2.2.10 GenericToElement_Mapping.....	21
C.2.3.2.2.11 GenericToEndFeatureMembership_Mapping.....	21
C.2.3.2.2.12 GenericToExpression_Mapping.....	23
C.2.3.2.2.13 GenericToFeature_Mapping.....	24
C.2.3.2.2.14 GenericToFeatureChaining_Mapping.....	26
C.2.3.2.2.15 GenericToFeatureMembership_Mapping.....	26
C.2.3.2.2.16 GenericToFeatureReferenceExpression_Mapping.....	28
C.2.3.2.2.17 GenericToFeatureTyping_Mapping.....	29
C.2.3.2.2.18 GenericToFeatureValue_Mapping.....	30
C.2.3.2.2.19 GenericToFunction_Mapping.....	32
C.2.3.2.2.20 GenericToImport_Mapping.....	33
C.2.3.2.2.21 GenericToInvocationExpression_Mapping.....	34
C.2.3.2.2.22 GenericToMembership_Mapping.....	35
C.2.3.2.2.23 GenericToNamespace_Mapping.....	37
C.2.3.2.2.24 GenericToOwningMembership_Mapping.....	37
C.2.3.2.2.25 GenericToPackage_Mapping.....	37
C.2.3.2.2.26 GenericToParameterMembership_Mapping.....	38
C.2.3.2.2.27 GenericToPredicate_Mapping.....	39
C.2.3.2.2.28 GenericToRedefinition_Mapping.....	40
C.2.3.2.2.29 GenericToRelationship_Mapping.....	41
C.2.3.2.2.30 GenericToReturnParameterMembership_Mapping.....	42
C.2.3.2.2.31 GenericToSpecialization_Mapping.....	43
C.2.3.2.2.32 GenericToStep_Mapping.....	44
C.2.3.2.2.33 GenericToSubclassification_Mapping.....	45
C.2.3.2.2.34 GenericToSubsetting_Mapping.....	45
C.2.3.2.2.35 GenericToTextualRepresentation_Mapping.....	46

C.2.3.2.2.36 GenericToType_Mapping.....	47
C.2.3.2.2.37 GenericToTypeFeaturing_Mapping .....	48
C.2.3.3 Generic Mappings FromTo KerML .....	49
C.2.3.3.1 Overview.....	49
C.2.3.3.2 Mapping Specifications .....	49
C.2.3.3.2.1 CommonMembership_Mapping.....	49
C.2.3.3.2.2 CommonParameterReferenceUsageInMembership_Mapping .....	50
C.2.3.3.2.3 CommonParameterReferenceUsageIn_Mapping .....	52
C.2.3.3.2.4 CommonParameterReferenceUsageInUntyped_Mapping.....	54
C.2.3.3.2.5 CommonReferenceUsageInFeatureTyping_Mapping .....	55
C.2.3.3.2.6 CommonReferenceUsageInUntyped_Mapping.....	56
C.2.3.3.2.7 CommonReturnParameterFeature_Mapping .....	58
C.2.3.3.2.8 CommonReturnParameterFeatureTyping_Mapping .....	60
C.2.3.3.2.9 CommonReturnParameterFeatureUntyped_Mapping .....	61
C.2.3.3.2.10 CommonReturnParameterFeatureMembership_Mapping.....	62
C.2.3.3.2.11 CommonReturnParameterReferenceUsageMembership_Mapping.....	63
C.2.3.3.2.12 CommonReturnParameterReferenceUsage_Mapping.....	65
C.2.3.3.2.13 CommonParameterReferenceUsageInFeatureTyping_Mapping.....	66
C.2.3.3.2.14 CommonReturnParameterReferenceUsageFeatureTyping_Mapping .....	68
C.2.3.3.2.15 CommonReturnParameterReferenceUsageUntyped_Mapping .....	69
C.2.3.3.2.16 EmptyReturnParameterFeatureMembership_Mapping .....	71
C.2.3.4 Generic Mappings To Systems.....	72
C.2.3.4.1 Overview.....	72
C.2.3.4.2 Mapping Specifications .....	73
C.2.3.4.2.1 GenericToActionUsage_Mapping .....	73
C.2.3.4.2.2 GenericToAssignmentActionUsage_Mapping .....	74
C.2.3.4.2.3 GenericToConnectionUsage_Mapping .....	76
C.2.3.4.2.4 GenericToConjugatedPortDefinition_Mapping .....	77
C.2.3.4.2.5 GenericToConjugatedPortTyping_Mapping .....	78
C.2.3.4.2.6 GenericToConstraintDefinition_Mapping.....	80
C.2.3.4.2.7 GenericToDefinition_Mapping .....	81
C.2.3.4.2.8 GenericToEventOccurrenceUsage_Mapping .....	82
C.2.3.4.2.9 GenericToItemDefinition_Mapping .....	83
C.2.3.4.2.10 GenericToMetadataUsage_Mapping .....	84
C.2.3.4.2.11 GenericToOccurrenceDefinition_Mapping .....	86
C.2.3.4.2.12 GenericToOccurrenceUsage_Mapping .....	87
C.2.3.4.2.13 GenericToPartUsage_Mapping .....	88
C.2.3.4.2.14 GenericToPortConjugation_Mapping .....	90
C.2.3.4.2.15 GenericToPortDefinition_Mapping.....	91
C.2.3.4.2.16 GenericToReferenceUsage_Mapping.....	92
C.2.3.4.2.17 GenericToStateUsage_Mapping.....	93
C.2.3.4.2.18 GenericToUsage_Mapping.....	95
C.2.4 SysML v1.7 .....	96
C.2.4.1 Overview .....	96
C.2.4.2 Activities.....	96
C.2.4.2.1 Overview.....	96
C.2.4.2.2 Mapping Specifications .....	96
C.2.4.3 Allocations.....	96
C.2.4.3.1 Overview.....	97
C.2.4.3.2 Mapping Specifications .....	97
C.2.4.3.2.1 AllocationDefinition_Mapping.....	97
C.2.4.3.2.2 AllocationDefinitionToFeatureMembership_Mapping .....	98
C.2.4.3.2.3 AllocationDefinitionFromFeatureMembership_Mapping.....	99
C.2.4.3.2.4 AllocationDefinitionFromFeatureTyping_Mapping .....	101

C.2.4.3.2.5 AllocationDefinitionFromReferenceUsage_Mapping.....	102
C.2.4.3.2.6 AllocationDefinitionToFeatureTyping_Mapping.....	103
C.2.4.3.2.7 AllocationDefinitionToReferenceUsage_Mapping.....	105
C.2.4.3.2.8 AllocationUsage_Mapping.....	106
C.2.4.4 Blocks.....	108
C.2.4.4.1 Overview.....	108
C.2.4.4.2 SysML v1 Blocks elements not mapped.....	109
C.2.4.4.3 Mapping Specifications.....	109
C.2.4.4.3.1 AssociationBlock_Mapping.....	109
C.2.4.4.3.2 BindingConnector_Mapping.....	111
C.2.4.4.3.3 Block_Mapping.....	112
C.2.4.4.3.4 Part_Mapping.....	114
C.2.4.4.3.5 EncapsulatedBlock_Mapping.....	116
C.2.4.4.3.6 EncapsulatedBlockMetadataMembership_Mapping.....	117
C.2.4.4.3.7 EncapsulatedBlockMetadata_Mapping.....	118
C.2.4.4.3.8 EncapsulatedBlockMetadataFeatureMembership_Mapping.....	120
C.2.4.4.3.9 EncapsulatedBlockMetadataFeatureTyping_Mapping.....	121
C.2.4.4.3.10 EncapsulatedBlockMetadataReferenceUsage_Mapping.....	122
C.2.4.4.3.11 EncapsulatedBlockMetadataFeatureValue_Mapping.....	124
C.2.4.4.3.12 EncapsulatedBlockMetadataRedefinition_Mapping.....	125
C.2.4.5 Libraries.....	127
C.2.4.5.1 Requirements.....	127
C.2.4.5.1.1 VerdictKind.....	127
C.2.4.5.2 UnitAndQuantityKind.....	127
C.2.4.6 Model Elements.....	127
C.2.4.6.1 Overview.....	127
C.2.4.6.2 Mapping Specifications.....	127
C.2.4.6.2.1 ProblemRationaleMetadataUsage_Mapping.....	127
C.2.4.6.2.2 CommentToConcern_Mapping.....	129
C.2.4.6.2.3 CommentToConcernComment_Mapping.....	130
C.2.4.6.2.4 CommentToConcernDocumentation_Mapping.....	131
C.2.4.6.2.5 CommenttToConcernReturnParameter_Mapping.....	132
C.2.4.6.2.6 CommentToConcernReturnParameterMembership_Mapping.....	134
C.2.4.6.2.7 ProblemRationaleMetadataFeatureMembership_Mapping.....	135
C.2.4.6.2.8 ProblemRationaleMetadataFeatureTyping_Mapping.....	137
C.2.4.6.2.9 ProblemRationaleMetadataReferenceUsage_Mapping.....	138
C.2.4.6.2.10 ProblemRationaleMetadataFeatureValue_Mapping.....	140
C.2.4.6.2.11 ProblemRationaleMetadataMembership_Mapping.....	141
C.2.4.6.2.12 ElementGroup_Mapping.....	142
C.2.4.6.2.13 ElementGroupCriterion_Mapping.....	143
C.2.4.6.2.14 ElementGroupMetadaMembership_Mapping.....	145
C.2.4.6.2.15 ElementGroupMetadataFeatureMembership_Mapping.....	146
C.2.4.6.2.16 ElementGroupMetadataFeatureTyping_Mapping.....	147
C.2.4.6.2.17 ElementGroupMetadataFeatureValue_Mapping.....	149
C.2.4.6.2.18 ElementGroupMetadataRedefinition_Mapping.....	150
C.2.4.6.2.19 ElementGroupMetadataReferenceUsage_Mapping.....	151
C.2.4.6.2.20 ElementGroupMetadataUsage_Mapping.....	153
C.2.4.6.2.21 ElementGroupMembership_Mapping.....	155
C.2.4.6.2.22 ProblemRationale_Mapping.....	156
C.2.4.6.2.23 ProblemRationaleMetadataRedefinition_Mapping.....	158
C.2.4.6.2.24 ProblemRationaleMetadataFeatureValueString_Mapping.....	159
C.2.4.6.2.25 Stakeholder_Mapping.....	160
C.2.4.6.2.26 StakeholderMembership_Mapping.....	162
C.2.4.6.2.27 StakeholderPartUsage_Mapping.....	163

C.2.4.6.2.28 Viewpoint_Mapping .....	165
C.2.4.6.2.29 ViewpointPurposeMetadata_Mapping .....	166
C.2.4.6.2.30 ViewpointPurposeMetadataFeatureTyping_Mapping .....	168
C.2.4.6.2.31 ViewpointPurposeMetadataMembership_Mapping .....	169
C.2.4.6.2.32 ViewpointSubject_Mapping .....	170
C.2.4.6.2.33 ViewpointSubjectMembership_Mapping .....	172
C.2.4.7 PortsAndFlows .....	173
C.2.4.7.1 Overview .....	173
C.2.4.7.2 Mapping Specifications .....	174
C.2.4.7.2.1 AcceptChangeStructuralFeatureEventAction_Mapping .....	174
C.2.4.7.2.2 FullPort_Mapping .....	175
C.2.4.7.2.3 InterfaceBlock_Mapping .....	177
C.2.4.7.2.4 ItemFlow_Mapping .....	179
C.2.4.7.2.5 ItemFlowFeatureMembership_Mapping .....	180
C.2.4.7.2.6 ItemFlowItemFeature_Mapping .....	181
C.2.4.7.2.7 ItemFlowItemFeatureTyping_Mapping .....	182
C.2.4.7.2.8 ItemFlowSourceEndFeatureMembership_Mapping .....	183
C.2.4.7.2.9 ItemFlowSourceFeature_Mapping .....	185
C.2.4.7.2.10 ItemFlowSourceFeatureSubsetting_Mapping .....	186
C.2.4.7.2.11 ItemFlowTargetEndFeatureMembership_Mapping .....	187
C.2.4.7.2.12 ItemFlowTargetFeature_Mapping .....	188
C.2.4.7.2.13 ItemFlowTargetFeatureSubsetting_Mapping .....	190
C.2.4.7.2.14 OperationDirectedFeature_Mapping .....	191
C.2.4.8 Requirements .....	192
C.2.4.8.1 Overview .....	193
C.2.4.8.2 SysML v1 Requirements elements not mapped .....	193
C.2.4.8.3 Mapping Specifications .....	193
C.2.4.8.3.1 Requirement_Mapping .....	193
C.2.4.8.3.2 DeriveReq_Mapping .....	195
C.2.4.8.3.3 Refine_Mapping .....	196
C.2.4.8.3.4 RequirementDocumentation_Mapping .....	197
C.2.4.8.3.5 RequirementDocumentationMembership_Mapping .....	198
C.2.4.8.3.6 RequirementSubject_Mapping .....	199
C.2.4.8.3.7 RequirementSubjectMembership_Mapping .....	201
C.2.4.8.3.8 Satisfy_Mapping .....	203
C.2.4.8.3.9 SatisfyFeatureMembership_Mapping .....	204
C.2.4.8.3.10 SatisfyFeatureMembershipReferenceUsage_Mapping .....	206
C.2.4.8.3.11 SatisfyFeatureMembershipReferenceUsageFeatureTyping_Mapping .....	208
C.2.4.8.3.12 SatisfyFeatureTyping_Mapping .....	209
C.2.4.8.3.13 SatisfySubjectMembership_Mapping .....	210
C.2.4.8.3.14 SatisfySubjectMembershipFeatureValue_Mapping .....	211
C.2.4.8.3.15 SatisfySubjectMembershipFeatureValueFeatureReferenceExpression_Mapping .....	213
C.2.4.8.3.16	
SatisfySubjectMembershipFeatureValueFeatureReferenceExpressionMembership_Mapping .....	214
C.2.4.8.3.17	
SatisfySubjectMembershipFeatureValueFeatureReferenceExpressionReturnParameterMembership2_Mapping	
C.2.4.8.3.18	
SatisfySubjectMembershipFeatureValueFeatureReferenceExpressionReturnParameterMembership2_Mapping	
C.2.4.8.3.19 SatisfySubjectMembershipReferenceUsage_Mapping .....	218
C.2.4.8.3.20 TestCaseActivity_Mapping .....	220
C.2.4.8.3.21 TestCaseActivityReturnParameterMembership_Mapping .....	221
C.2.4.8.3.22 TestCaseVerifyObjectiveMembership_Mapping .....	223
C.2.4.8.3.23 TestCaseVerifyObjectiveRequirementUsage_Mapping .....	224
C.2.4.8.3.24 TestCaseVerifyRequirementUsageReferenceSubsetting_Mapping .....	226



C.2.4.8.3.25 TestCaseVerifyRequirementUsage_Mapping .....	227
C.2.4.8.3.26 Trace_Mapping .....	228
C.2.4.8.3.27 Verify_Mapping .....	229
C.2.5 UML4SysML .....	230
C.2.5.1 Overview .....	230
C.2.5.2 Actions .....	230
C.2.5.2.1 Overview .....	231
C.2.5.2.2 SysML v1 Activities elements not mapped .....	237
C.2.5.2.3 Mapping Specifications .....	237
C.2.5.2.3.1 Actions .....	237
C.2.5.2.3.1.1 CommonAction_Mapping .....	237
C.2.5.2.3.1.2 OpaqueAction_Mapping .....	239
C.2.5.2.3.1.3 OpaqueActionBody_Mapping .....	241
C.2.5.2.3.1.4 OpaqueActionBodyMembership_Mapping .....	242
C.2.5.2.3.1.5 Pin_Mapping .....	243
C.2.5.2.3.1.6 PinFeatureTyping_Mapping .....	245
C.2.5.2.3.1.7 UntypedPin_Mapping .....	246
C.2.5.2.3.1.8 ValuePin_Mapping .....	248
C.2.5.2.3.1.9 ValuePinFeatureValue_Mapping .....	249
C.2.5.2.3.1.10 ValuePinUntyped_Mapping .....	251
C.2.5.2.3.1.11 ValuePinValue_Mapping .....	252
C.2.5.2.3.2 Invocation Actions .....	254
C.2.5.2.3.2.1 CallBehaviorAction_Mapping .....	254
C.2.5.2.3.2.2 CallBehaviorFeatureTyping_Mapping .....	256
C.2.5.2.3.2.3 CallOperationAction_Mapping .....	257
C.2.5.2.3.2.4 CallOperationOutputPin_Mapping .....	259
C.2.5.2.3.2.5 CallOperationOutputPinFeature_Mapping .....	260
C.2.5.2.3.2.6 CallOperationOutputPinFeatureChainExpression_Mapping .....	261
C.2.5.2.3.2.7 CallOperationOutputPinFeatureChainExpressionMembership_Mapping .....	263
C.2.5.2.3.2.8 CallOperationOutputPinFeatureFeature_Mapping .....	264
C.2.5.2.3.2.9 CallOperationOutputPinFeatureFeatureMembership_Mapping .....	265
C.2.5.2.3.2.10 CallOperationOutputPinFeatureFeatureValue_Mapping .....	266
C.2.5.2.3.2.11 CallOperationOutputPinFeatureMembership_Mapping .....	268
C.2.5.2.3.2.12 CallOperationOutputPinFeatureReferenceExpression_Mapping .....	269
C.2.5.2.3.2.13 CallOperationOutputPinFeatureReferenceExpressionMembership_Mapping .....	271
C.2.5.2.3.2.14 CallOperationOutputPinParameterMembership_Mapping .....	272
C.2.5.2.3.2.15 CallOperationOutputPinReferenceUsage_Mapping .....	273
C.2.5.2.3.2.16 CallOperationOutputPinReferenceUsageFeatureValue_Mapping .....	275
C.2.5.2.3.2.17 SendSignalAction_Mapping .....	276
C.2.5.2.3.3 Link Actions .....	278
C.2.5.2.3.3.1 ClearAssociationAction_Mapping .....	278
C.2.5.2.3.3.2 CreateLinkAction_Mapping .....	279
C.2.5.2.3.3.3 DestroyLinkAction_Mapping .....	281
C.2.5.2.3.4 Object Actions .....	283
C.2.5.2.3.4.1 CreateObjectAction_Mapping .....	283
C.2.5.2.3.4.2 CreateObjectInvocationExpressionFeatureTyping_Mapping .....	285
C.2.5.2.3.4.3 CreateObjectInvocationExpression_Mapping .....	286
C.2.5.2.3.4.4 CreateObjectPin_Mapping .....	287
C.2.5.2.3.4.5 CreateObjectPinFeatureValue_Mapping .....	289
C.2.5.2.3.4.6 CreateObjectPinMembership_Mapping .....	290
C.2.5.2.3.4.7 ReadIsClassifiedObjectAction_Mapping .....	291
C.2.5.2.3.4.8 ReadIsClassifiedObjectActionFeatureValue_Mapping .....	292
C.2.5.2.3.4.9 ReadIsClassifiedObjectActionFeatureValueOperatorExpression_Mapping .....	294

C.2.5.2.3.4.10	
ReadIsClassifiedObjectActionFeatureValueOperatorExpressionFeature_Mapping .....	295
C.2.5.2.3.4.11	
ReadIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValue_Mapping .....	296
C.2.5.2.3.4.12	
ReadIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValueExpression_Mapping .....	298
C.2.5.2.3.4.13	
ReadIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValueExpressionMembership_Mapping .....	299
C.2.5.2.3.4.14	
ReadIsClassifiedObjectActionFeatureValueOperatorExpressionParameterMembership_Mapping .....	300
C.2.5.2.3.4.15 ReadIsClassifiedObjectActionOutputPin_Mapping .....	302
C.2.5.2.3.4.16 ReadExtentAction_Mapping .....	304
C.2.5.2.3.4.17 ReadExtentActionFeatureValue_Mapping .....	305
C.2.5.2.3.4.18 ReadExtentActionFeatureValueOperatorExpression_Mapping .....	307
C.2.5.2.3.4.19 ReadExtentActionFeatureValueOperatorExpressionFeature_Mapping .....	308
C.2.5.2.3.4.20 ReadExtentActionFeatureValueOperatorExpressionFeatureTyping_Mapping .....	309
C.2.5.2.3.4.21 ReadExtentActionFeatureValueOperatorExpressionMembership_Mapping .....	310
C.2.5.2.3.4.22 ReadExtentActionOutputPin_Mapping .....	312
C.2.5.2.3.4.23 ReadSelfAction_Mapping .....	314
C.2.5.2.3.4.24 ReadSelfActionFeatureValue_Mapping .....	315
C.2.5.2.3.4.25 ReadSelfActionFeatureValueFeatureReferenceExpression_Mapping .....	316
C.2.5.2.3.4.26	
ReadSelfActionFeatureValueFeatureReferenceExpressionMembership_Mapping .....	318
C.2.5.2.3.4.27 ReadSelfActionOutputPin_Mapping .....	319
C.2.5.2.3.4.28 TestIdentityAction_Mapping .....	321
C.2.5.2.3.4.29 EqualOperatorExpressionOperand_Mapping .....	323
C.2.5.2.3.4.30 CommonFeatureReferenceExpression_Mapping .....	324
C.2.5.2.3.4.31 CommonReferenceUsageIn_Mapping .....	326
C.2.5.2.3.4.32 CommonReferenceUsageInFeatureMembership_Mapping .....	327
C.2.5.2.3.4.33 TestIdentityActionOperator_Mapping .....	329
C.2.5.2.3.4.34 EqualOperatorExpressionFeature_Mapping .....	330
C.2.5.2.3.4.35 TestIdentityActionResultExpressionMembership_Mapping .....	331
C.2.5.2.3.4.36 ValueSpecificationAction_Mapping .....	333
C.2.5.2.3.4.37 ValueSpecificationActionOutputPin_Mapping .....	335
C.2.5.2.3.4.38 ValueSpecificationActionOutputPinFeatureValue_Mapping .....	336
C.2.5.2.3.5 Structural Feature Actions .....	338
C.2.5.2.3.5.1 AddStructuralFeatureValueAction_Mapping .....	338
C.2.5.2.3.5.2 AddStructuralFeatureValueActionAssignmentAction_Mapping .....	339
C.2.5.2.3.5.3 AddStructuralFeatureValueActionAssignmentActionMembership_Mapping .....	341
C.2.5.2.3.5.4 ReadStructuralFeatureAction_Mapping .....	342
C.2.5.2.3.6 Structured Actions .....	344
C.2.5.2.3.6.1 SequenceNode_Mapping .....	344
C.2.5.2.3.6.2 StructuredActivityNode_Mapping .....	346
C.2.5.2.3.7 Variable Actions .....	348
C.2.5.2.3.7.1 AddVariableValueAction_Mapping .....	348
C.2.5.2.3.7.2 AddVariableValueActionFeatureTyping_Mapping .....	350
C.2.5.2.3.7.3 ClearVariableAction_Mapping .....	351
C.2.5.2.3.7.4 ClearVariableActionFeatureMembership_Mapping .....	353
C.2.5.2.3.7.5 ClearVariableActionReferenceUsage_Mapping .....	354
C.2.5.2.3.7.6 ClearVariableActionReferenceUsageFeatureValue_Mapping .....	356
C.2.5.2.3.7.7 Null_Mapping .....	357
C.2.5.2.3.7.8 ReadVariableAction_Mapping .....	358
C.2.5.2.3.7.9 RemoveVariableValueAction_Mapping .....	360
C.2.5.2.3.7.10 RemoveVariableValueActionAssignmentAction_Mapping .....	362

C.2.5.2.3.7.11 RemoveVariableValueActionAssignmentActionMembership_Mapping .....	363
C.2.5.2.3.7.12 RemoveVariableValueActionAssignmentActionParameter_Mapping .....	365
C.2.5.2.3.7.13	
RemoveVariableValueActionAssignmentActionParameterFeatureMembership_Mapping .....	366
C.2.5.2.3.7.14	
RemoveVariableValueActionAssignmentActionParameterMembership_Mapping .....	368
C.2.5.2.3.7.15	
RemoveVariableValueActionAssignmentActionParameterReference_Mapping .....	369
C.2.5.2.3.7.16	
RemoveVariableValueActionAssignmentActionParameterReferenceFeatureMembership_Mapping .....	370
C.2.5.2.3.7.17	
RemoveVariableValueActionAssignmentActionParameterReferenceReference_Mapping .....	372
C.2.5.2.3.7.18 RemoveVariableValueActionAssignmentActionSecondParameter_Mapping .....	374
C.2.5.2.3.7.19	
RemoveVariableValueActionAssignmentActionSecondParameterMembership_Mapping .....	375
C.2.5.2.3.7.20 RemoveVariableValueActionExpressionMembership_Mapping .....	377
C.2.5.2.3.7.21 RemoveVariableValueActionExpressionParameter_Mapping .....	378
C.2.5.2.3.7.22	
RemoveVariableValueActionExpressionParameterFeatureReference_Mapping .....	379
C.2.5.2.3.7.23	
RemoveVariableValueActionExpressionParameterFeatureReferenceMembership_Mapping .....	381
C.2.5.2.3.7.24 RemoveVariableValueActionExpressionParameterMembership_Mapping .....	382
C.2.5.2.3.7.25 RemoveVariableValueActionExpressionParameterValue_Mapping .....	383
C.2.5.2.3.7.26 RemoveVariableValueActionExpressionReferenceUsage_Mapping .....	385
C.2.5.2.3.7.27	
RemoveVariableValueActionExpressionReferenceUsageFeatureValue_Mapping .....	386
C.2.5.2.3.7.28 RemoveVariableValueActionInvocationExpression_Mapping .....	387
C.2.5.2.3.7.29 RemoveVariableValueActionInvocationExpressionFeatureTyping_Mapping .....	389
C.2.5.3 Activities .....	390
C.2.5.3.1 Overview .....	390
C.2.5.3.2 SysML v1 Activities elements not mapped .....	394
C.2.5.3.3 Mapping Specifications .....	394
C.2.5.3.3.1 ActivityAsDefinition_Mapping .....	395
C.2.5.3.3.2 ActivityAsUsage_Mapping .....	396
C.2.5.3.3.3 ActivityEdgeMetadata_Mapping .....	397
C.2.5.3.3.4 ActivityEdgeMetadataFeatureMembership_Mapping .....	399
C.2.5.3.3.5 ActivityEdgeMetadataFeatureTyping_Mapping .....	400
C.2.5.3.3.6 ActivityEdgeMetadataFeatureValue_Mapping .....	401
C.2.5.3.3.7 ActivityEdgeMetadataMembership_Mapping .....	402
C.2.5.3.3.8 ActivityEdgeMetadataRedefinition_Mapping .....	404
C.2.5.3.3.9 ActivityEdgeMetadataReferenceUsage_Mapping .....	405
C.2.5.3.3.10 ActivityFinalNodeMembership_Mapping .....	407
C.2.5.3.3.11 CommonActivity_Mapping .....	408
C.2.5.3.3.12 CommonActivityEdgeSuccessionAsUsage_Mapping .....	409
C.2.5.3.3.13 CommonVariable_Mapping .....	411
C.2.5.3.3.14 ControlFlowTransitionUsage_Mapping .....	413
C.2.5.3.3.15 ActivityEdgeSourceEndFeature_Mapping .....	415
C.2.5.3.3.16 ActivityEdgeInitialNodeSourceEndFeatureMembership_Mapping .....	416
C.2.5.3.3.17 ActivityEdgeSourceInitialNode_Mapping .....	417
C.2.5.3.3.18 ActivityEdgeSourceEndFeatureMembership_Mapping .....	418
C.2.5.3.3.19 ActivityEdgeSourceInitialNodeSubsetting_Mapping .....	420
C.2.5.3.3.20 ActivityEdgeSourceEndSubsetting_Mapping .....	421
C.2.5.3.3.21 ControlFlowFinalNodeTargetEndFeatureMembership_Mapping .....	422
C.2.5.3.3.22 ControlFlowTargetFinalNodeSubsetting_Mapping .....	423

C.2.5.3.3.23	ControlFlowSuccessionAsUsage_Mapping .....	424
C.2.5.3.3.24	ControlFlowTargetFinalNode_Mapping .....	426
C.2.5.3.3.25	ControlFlowTargetEndFeature_Mapping .....	427
C.2.5.3.3.26	ControlFlowTargetEndFeatureMembership_Mapping .....	428
C.2.5.3.3.27	ControlFlowTargetEndSubsetting_Mapping .....	430
C.2.5.3.3.28	ControlFlowTransitionUsageFeatureMembership_Mapping .....	431
C.2.5.3.3.29	ControlFlowTransitionUsageFeatureReferenceExpression_Mapping .....	431
C.2.5.3.3.30	ControlFlowTransitionUsageFeatureReferenceExpressionMembership_Mapping .....	433
C.2.5.3.3.31	ActivityEdgeTransitionUsageSourceMembership_Mapping .....	434
C.2.5.3.3.32	DecisionNode_Mapping .....	435
C.2.5.3.3.33	ForkNode_Mapping .....	437
C.2.5.3.3.34	InitialNodeMembership_Mapping .....	438
C.2.5.3.3.35	JoinNode_Mapping .....	440
C.2.5.3.3.36	MergeNode_Mapping .....	441
C.2.5.3.3.37	ObjectFlow_Mapping .....	443
C.2.5.3.3.38	ObjectFlowFeatureMembership_Mapping .....	445
C.2.5.3.3.39	ObjectFlowGuardFeatureMembership_Mapping .....	446
C.2.5.3.3.40	ObjectFlowGuard_Mapping .....	447
C.2.5.3.3.41	ObjectFlowGuardSuccessionTargetEndFeature_Mapping .....	449
C.2.5.3.3.42	ObjectFlowGuardSuccessionTargetEndFeatureMembership_Mapping .....	450
C.2.5.3.3.43	ObjectFlowGuardSuccessionTargetEndSubsetting_Mapping .....	452
C.2.5.3.3.44	ObjectFlowItemFeature_Mapping .....	453
C.2.5.3.3.45	ObjectFlowItemFeatureMembership_Mapping .....	454
C.2.5.3.3.46	ObjectFlowItemFeatureTyping_Mapping .....	456
C.2.5.3.3.47	ObjectFlowItemFeatureUntyped_Mapping .....	457
C.2.5.3.3.48	ObjectFlowEndFeatureMembership_Mapping .....	458
C.2.5.3.3.49	ObjectFlowItemFlowEnd_Mapping .....	459
C.2.5.3.3.50	ObjectFlowItemFlowFeature_Mapping .....	460
C.2.5.3.3.51	ObjectFlowItemFlowFeatureMembership_Mapping .....	461
C.2.5.3.3.52	ObjectFlowItemFlowRedefinition_Mapping .....	462
C.2.5.3.3.53	ObjectFlowItemFlowSubsetting_Mapping .....	464
C.2.5.3.3.54	VariableAttribute_Mapping .....	465
C.2.5.3.3.55	VariableFeatureTyping_Mapping .....	467
C.2.5.3.3.56	VariableItem_Mapping .....	468
C.2.5.3.3.57	VariableMembership_Mapping .....	470
C.2.5.4	Classification .....	471
C.2.5.4.1	Overview .....	471
C.2.5.4.2	Mapping Specifications .....	473
C.2.5.4.2.1	BehavioralFeature_Mapping .....	474
C.2.5.4.2.2	Classifier_Mapping .....	475
C.2.5.4.2.3	DefaultLowerBound_Mapping .....	476
C.2.5.4.2.4	DefaultLowerBoundTyping_Mapping .....	478
C.2.5.4.2.5	DefaultMultiplicityBoundOwnership_Mapping .....	479
C.2.5.4.2.6	DefaultMultiplicityBoundTyping_Mapping .....	480
C.2.5.4.2.7	DefaultMultiplicityBoundValue_Mapping .....	482
C.2.5.4.2.8	DefaultMultiplicityElement_Mapping .....	483
C.2.5.4.2.9	DefaultMultiplicityLowerBoundOwnership_Mapping .....	484
C.2.5.4.2.10	DefaultMultiplicityLowerBoundValue_Mapping .....	486
C.2.5.4.2.11	DefaultMultiplicityMembership_Mapping .....	487
C.2.5.4.2.12	DefaultMultiplicityUpperBoundOwnership_Mapping .....	488
C.2.5.4.2.13	DefaultMultiplicityUpperBoundValue_Mapping .....	490
C.2.5.4.2.14	DefaultUpperBound_Mapping .....	491
C.2.5.4.2.15	DefaultUpperBoundTyping_Mapping .....	493
C.2.5.4.2.16	ElementFeatureMembership_Mapping .....	494

C.2.5.4.2.17 Generalization_Mapping .....	495
C.2.5.4.2.18 InstanceSpecificationLink_Mapping .....	497
C.2.5.4.2.19 InstanceSpecification_Mapping .....	498
C.2.5.4.2.20 InstanceSpecificationFeatureTyping_Mapping .....	500
C.2.5.4.2.21 InstanceValue_Mapping .....	501
C.2.5.4.2.22 InstanceValueInstanceSpecification_Mapping .....	503
C.2.5.4.2.23 LowerBoundValueOwnership_Mapping .....	504
C.2.5.4.2.24 MultiplicityElement_Mapping .....	506
C.2.5.4.2.25 MultiplicityLowerBoundOwnership_Mapping .....	507
C.2.5.4.2.26 MultiplicityMembership_Mapping .....	508
C.2.5.4.2.27 MultiplicityUpperBoundOwnership_Mapping .....	509
C.2.5.4.2.28 Operation_Mapping .....	511
C.2.5.4.2.29 Parameter_Mapping .....	513
C.2.5.4.2.30 ParameterMembership_Mapping .....	514
C.2.5.4.2.31 ParameterSet_Mapping .....	516
C.2.5.4.2.32 ParameterSetMembership_Mapping .....	517
C.2.5.4.2.33 ParameterSetParameterFeatureMembership_Mapping .....	519
C.2.5.4.2.34 ParameterSetParameterReferenceUsage_Mapping .....	520
C.2.5.4.2.35 ParameterSetParameterReferenceUsageFeatureValue_Mapping .....	522
C.2.5.4.2.36 ParameterSetParameterReferenceUsageFeatureValueExpression_Mapping .....	523
C.2.5.4.2.37	
ParameterSetParameterReferenceUsageFeatureValueExpressionMembership_Mapping .....	524
C.2.5.4.2.38 ParameterToFeatureTyping_Mapping .....	526
C.2.5.4.2.39 Property_Mapping .....	527
C.2.5.4.2.40 PropertyCommon_Mapping .....	528
C.2.5.4.2.41 PropertyDefaultValue_Mapping .....	530
C.2.5.4.2.42 PropertyDefaultValueOpaqueExpression_Mapping .....	532
C.2.5.4.2.43 PropertySubsetting_Mapping .....	533
C.2.5.4.2.44 PropertyUntyped_Mapping .....	534
C.2.5.4.2.45 Realization_Mapping .....	536
C.2.5.4.2.46 Slot_Mapping .....	537
C.2.5.4.2.47 SlotMembership_Mapping .....	538
C.2.5.4.2.48 SlotToFeatureTyping_Mapping .....	540
C.2.5.4.2.49 SlotValue_Mapping .....	541
C.2.5.4.2.50 StructuralFeature_Mapping .....	542
C.2.5.4.2.51 StructuralFeatureMembership_Mapping .....	544
C.2.5.4.2.52 StructuralFeatureToFeatureTyping_Mapping .....	545
C.2.5.4.2.53 TypedElementToFeatureTyping_Mapping .....	546
C.2.5.4.2.54 UpperBoundValueOwnership_Mapping .....	547
C.2.5.5 CommonBehavior .....	549
C.2.5.5.1 Overview .....	549
C.2.5.5.2 Mapping Specifications .....	550
C.2.5.5.2.1 Behavior_Mapping .....	550
C.2.5.5.2.2 ChangeEvent_Mapping .....	551
C.2.5.5.2.3 CommonOpaqueBehavior_Mapping .....	552
C.2.5.5.2.4 OpaqueBehaviorAsDefinition_Mapping .....	554
C.2.5.5.2.5 OpaqueBehaviorAsUsage_Mapping .....	555
C.2.5.5.2.6 OpaqueBehaviorMembership_Mapping .....	557
C.2.5.5.2.7 OpaqueBehaviorSpecification_Mapping .....	558
C.2.5.5.2.8 TimeEvent_Mapping .....	559
C.2.5.5.2.9 Trigger_Mapping .....	560
C.2.5.6 CommonStructure .....	561
C.2.5.6.1 Overview .....	561

C.2.5.6.2 Mapping Specifications .....	564
C.2.5.6.2.1 Abstraction_Mapping .....	564
C.2.5.6.2.2 Comment_Mapping .....	565
C.2.5.6.2.3 CommentToAnnotation_Mapping .....	566
C.2.5.6.2.4 Constraint_Mapping .....	568
C.2.5.6.2.5 ConstrainedElementFeatureMembership_Mapping .....	569
C.2.5.6.2.6 ConstraintUsageFeatureTyping_Mapping .....	570
C.2.5.6.2.7 ConstraintUsage_Mapping .....	571
C.2.5.6.2.8 Dependency_Mapping .....	573
C.2.5.6.2.9 DirectedRelationship_Mapping .....	574
C.2.5.6.2.10 ElementMain_Mapping .....	575
C.2.5.6.2.11 ElementMembership_Mapping .....	576
C.2.5.6.2.12 ElementOwnership_Mapping .....	577
C.2.5.6.2.13 ElementOwningMembership_Mapping .....	578
C.2.5.6.2.14 NamedElementMain_Mapping .....	580
C.2.5.6.2.15 Namespace_Mapping .....	581
C.2.5.6.2.16 Relationship_Mapping .....	582
C.2.5.6.2.17 Usage_Mapping .....	583
C.2.5.7 InformationFlows .....	584
C.2.5.7.1 Overview .....	584
C.2.5.7.2 Mapping Specifications .....	584
C.2.5.8 Interactions .....	584
C.2.5.8.1 Overview .....	584
C.2.5.8.2 Mapping Specifications .....	587
C.2.5.8.2.1 Interaction_Mapping .....	587
C.2.5.8.2.2 LifelineMembership_Mapping .....	588
C.2.5.8.2.3 LifelinePartUsage_Mapping .....	590
C.2.5.8.2.4 LifelineFeatureTyping_Mapping .....	591
C.2.5.8.2.5 MessageOccurrenceSpecification_Mapping .....	592
C.2.5.8.2.6 MessageOccurrenceSpecificationMembership_Mapping .....	594
C.2.5.9 Packages .....	596
C.2.5.9.1 Overview .....	596
C.2.5.9.2 UML4SysML Packages elements not mapped .....	598
C.2.5.9.3 Mapping Specifications .....	598
C.2.5.9.3.1 ElementImport_Mapping .....	598
C.2.5.9.3.2 Package_Mapping .....	600
C.2.5.9.3.3 PackageImport_Mapping .....	601
C.2.5.9.3.4 Model_Mapping .....	602
C.2.5.9.3.5 ModelViewpointMetadataUsage_Mapping .....	603
C.2.5.9.3.6 ModelViewpointMetadataFeatureMembership_Mapping .....	603
C.2.5.9.3.7 ModelViewpointMetadataReferenceUsage_Mapping .....	605
C.2.5.9.3.8 ModelViewpointMetadataFeatureTyping_Mapping .....	606
C.2.5.9.3.9 ModelViewpointMetadataMembership_Mapping .....	608
C.2.5.9.3.10 ModelViewpointMetadataFeatureValue_Mapping .....	609
C.2.5.9.3.11 ModelViewpointMetadataRedefinition_Mapping .....	610
C.2.5.9.3.12 ModelViewpointValue_Mapping .....	611
C.2.5.9.3.13 PackageURIMetadataUsage_Mapping .....	613
C.2.5.9.3.14 PackageURIFeatureMembership_Mapping .....	615
C.2.5.9.3.15 PackageURIFeatureTyping_Mapping .....	616
C.2.5.9.3.16 PackageURIMetadataReferenceUsage_Mapping .....	617
C.2.5.9.3.17 PackageURIMetadataFeatureValue_Mapping .....	619
C.2.5.9.3.18 PackageURIMetadataMembership_Mapping .....	620
C.2.5.9.3.19 PackageURIRedefinition_Mapping .....	622
C.2.5.9.3.20 PackageURIValue_Mapping .....	623

C.2.5.9.3.21 Profile_Mapping .....	624
C.2.5.9.3.22 StereotypeMetadataDefinition_Mapping.....	626
C.2.5.9.3.23 StereotypeMetadataDefinitionFeatureMembership_Mapping .....	627
C.2.5.9.3.24 StereotypeMetadataDefinitionMembership_Mapping .....	628
C.2.5.9.3.25 StereotypeMetadataDefinitionReferenceUsage_Mapping .....	630
C.2.5.9.3.26	
StereotypeMetadataDefinitionReferenceUsageFeatureMembershipReference_Mapping.....	631
C.2.5.9.3.27 StereotypeMetadataDefinitionReferenceUsageFeatureMembershipUsage_Mapping ...	633
C.2.5.9.3.28	
StereotypeMetadataDefinitionReferenceUsageFeatureMembershipUsageFeature_Mapping .....	634
C.2.5.9.3.29	
StereotypeMetadataDefinitionReferenceUsageFeatureMembershipUsageFeatureTyping_Mapping	635
C.2.5.9.3.30	
StereotypeMetadataDefinitionReferenceUsageFeatureReferenceExpression_Mapping .....	636
C.2.5.9.3.31	
StereotypeMetadataDefinitionReferenceUsageFeatureReferenceExpressionMembership_Mapping	638
C.2.5.9.3.32	
StereotypeMetadataDefinitionReferenceUsageFeatureReferenceExpressionReturnParameter_Mapping	639
C.2.5.9.3.33	
StereotypeMetadataDefinitionReferenceUsageFeatureReferenceExpressionReturnParameterMembership_Mapping	640
C.2.5.9.3.34 StereotypeMetadataDefinitionReferenceUsageFeatureValue_Mapping .....	642
C.2.5.9.3.35 StereotypeMetadataDefinitionReferenceUsageOperatorExpression_Mapping .....	643
C.2.5.9.3.36 StereotypeMetadataDefinitionReferenceUsageRedefinition_Mapping .....	645
C.2.5.9.3.37 StereotypeMetadataDefinitionSubclassification_Mapping .....	646
C.2.5.9.3.38 StereotypeOccurrenceUsage_Mapping.....	647
C.2.5.9.3.39 StereotypeOccurrenceDefinitionMembership_Mapping .....	649
C.2.5.9.3.40 StereotypeOccurrenceDefinition_Mapping .....	650
C.2.5.9.3.41 StereotypeOccurrenceUsageFeatureTyping_Mapping .....	651
C.2.5.9.3.42 StereotypeOccurrenceUsageMembership_Mapping.....	653
C.2.5.9.3.43 StereotypeOccurrenceUsageMultiplicityMembership_Mapping .....	654
C.2.5.9.3.44 StereotypeOccurrenceUsageMultiplicityRange_Mapping .....	655
C.2.5.9.3.45 StereotypeOccurrenceUsageMultiplicityRangeInfinity_Mapping .....	656
C.2.5.9.3.46 StereotypeOccurrenceUsageMultiplicityRangeInfinityReturnParameter_Mapping .....	658
C.2.5.9.3.47	
StereotypeOccurrenceUsageMultiplicityRangeInfinityReturnParameterMembership_Mapping .....	659
C.2.5.9.3.48 StereotypeOccurrenceUsageMultiplicityRangeMembership_Mapping .....	660
C.2.5.10 SimpleClassifiers .....	662
C.2.5.10.1 Overview.....	662
C.2.5.10.2 Mapping Specifications .....	662
C.2.5.10.2.1 Attribute_Mapping.....	662
C.2.5.10.2.2 AttributeRedefined_Mapping .....	664
C.2.5.10.2.3 AttributeRedefinedRedefinition_Mapping .....	666
C.2.5.10.2.4 AttributeRedefinedMembership_Mapping .....	667
C.2.5.10.2.5 AttributeRedefinedFeatureTyping_Mapping.....	668
C.2.5.10.2.6 BehavioredClassifier_Mapping .....	670
C.2.5.10.2.7 ClassifierBehaviorMembership_Mapping.....	671
C.2.5.10.2.8 BehavioredClassifierToFeatureTyping_Mapping .....	672
C.2.5.10.2.9 BehavioredClassifierToPerformActionUsage_Mapping.....	674
C.2.5.10.2.10 DataType_Mapping .....	675
C.2.5.10.2.11 Enumeration_Mapping .....	676
C.2.5.10.2.12 EnumerationLiteral_Mapping.....	677
C.2.5.10.2.13 EnumerationVariantMembership_Mapping .....	679
C.2.5.10.2.14 Interface_Mapping.....	680
C.2.5.10.2.15 InterfaceConjugatedPortDefinition_Mapping .....	681

C.2.5.10.2.16 InterfaceConjugatedPortDefinitionMembership_Mapping .....	682
C.2.5.10.2.17 InterfacePortConjugation_Mapping .....	684
C.2.5.10.2.18 InterfaceRealization_Mapping.....	685
C.2.5.10.2.19 PrimitiveType_Mapping .....	686
C.2.5.10.2.20 PrimitiveTypeString_Mapping .....	687
C.2.5.10.2.21 Reception_Mapping.....	688
C.2.5.10.2.22 ReceptionToFeatureTyping_Mapping.....	690
C.2.5.10.2.23 Signal_Mapping.....	691
C.2.5.11 StructuredClassifiers.....	692
C.2.5.11.1 Overview.....	693
C.2.5.11.2 Mapping Specifications .....	695
C.2.5.11.2.1 AssociationCommon_Mapping .....	695
C.2.5.11.2.2 AssociationClass_Mapping .....	697
C.2.5.11.2.3 AssociationToAnnotation_Mapping.....	698
C.2.5.11.2.4 AssociationToAnnotatingFeature_Mapping.....	699
C.2.5.11.2.5 AssociationToFeatureMembership_Mapping .....	700
C.2.5.11.2.6 AssociationToFeatureTyping_Mapping .....	702
C.2.5.11.2.7 AssociationToMetadataFeature_Mapping.....	703
C.2.5.11.2.8 AssociationToMetadataFeatureValue_Mapping .....	704
C.2.5.11.2.9 AssociationToMetadataMembership_Mapping.....	705
C.2.5.11.2.10 AssociationToRedefinition_Mapping.....	707
C.2.5.11.2.11 BehavoredClassifier_Mapping .....	708
C.2.5.11.2.12 BehavoredClassifierToFeatureTyping_Mapping .....	709
C.2.5.11.2.13 BehavoredClassifierToPerformActionUsage_Mapping.....	710
C.2.5.11.2.14 Class_Mapping .....	711
C.2.5.11.2.15 ClassifierBehaviorMembership_Mapping.....	713
C.2.5.11.2.16 ConnectionEndToSubsetting_Mapping.....	714
C.2.5.11.2.17 Connector_Mapping .....	715
C.2.5.11.2.18 ConnectorEndToFeatureCommon_Mapping.....	717
C.2.5.11.2.19 ConnectorEndToMembership_Mapping .....	718
C.2.5.11.2.20 ConnectorEndToOwnedFeature_Mapping.....	720
C.2.5.11.2.21 ConnectorEndToSubsettedFeature_Mapping.....	721
C.2.5.11.2.22 ConnectorEndToSubsettedFeatureMembership_Mapping.....	723
C.2.5.11.2.23 ConnectorMultiplicityMembership_Mapping .....	724
C.2.5.11.2.24 ConnectorType_Mapping .....	725
C.2.5.11.2.25 ConnectorTypeDerived_Mapping .....	727
C.2.5.11.2.26 End_Mapping.....	729
C.2.5.11.2.27 EndMembership_Mapping .....	731
C.2.5.11.2.28 NonOwnedEndSubsetting_Mapping .....	732
C.2.5.11.2.29 EndToSubsettedFeature_Mapping.....	733
C.2.5.11.2.30 EndToSubsettedFeatureChaining_Mapping .....	735
C.2.5.11.2.31 NonOwnedEndToSubsettedFeatureMembership_Mapping.....	736
C.2.5.11.2.32 NonOwnedEnd_Mapping .....	737
C.2.5.11.2.33 NonOwnedEndMembership_Mapping.....	739
C.2.5.11.2.34 NonOwnedEndSubsettingMembership_Mapping.....	740
C.2.5.11.2.35 OwnedEnd_Mapping .....	742
C.2.5.11.2.36 Port_Mapping .....	743
C.2.5.11.2.37 OwnedEndAttribute_Mapping.....	746
C.2.5.11.2.38 OwnedEndMembership_Mapping.....	748
C.2.5.11.2.39 PropertyToFeatureChaining_Mapping .....	749
C.2.5.12 UseCases.....	750
C.2.5.12.1 Overview.....	750
C.2.5.12.2 SysML v1 UseCases elements not mapped .....	751



C.2.5.12.3 Mapping Specifications .....	751
C.2.5.12.3.1 Actor_Mapping .....	751
C.2.5.12.3.2 CaseActor_Mapping .....	752
C.2.5.12.3.3 CaseActorFeatureTyping_Mapping .....	754
C.2.5.12.3.4 CaseActorMembership_Mapping .....	755
C.2.5.12.3.5 Include_Mapping .....	756
C.2.5.12.3.6 IncludeFeatureTyping_Mapping .....	758
C.2.5.12.3.7 IncludeMembership_Mapping .....	759
C.2.5.12.3.8 UseCase_Mapping .....	760
C.2.5.12.3.9 CaseObjectiveMembership_Mapping .....	762
C.2.5.12.3.10 CaseEmptySubjectReferenceUsage_Mapping .....	763
C.2.5.12.3.11 CaseObjectiveRequirementUsage_Mapping .....	765
C.2.5.12.3.12 CaseSubjectMembership_Mapping .....	767
C.2.5.12.3.13 CaseSubjectFeatureTyping_Mapping .....	768
C.2.5.12.3.14 CaseSubjectReferenceUsage_Mapping .....	769
C.2.5.13 Values .....	771
C.2.5.13.1 Overview .....	771
C.2.5.13.2 Mapping Specifications .....	773
C.2.5.13.2.1 CommonValueSpecification_Mapping .....	773
C.2.5.13.2.2 EqualOperatorExpressionFeatureValue_Mapping .....	775
C.2.5.13.2.3 Expression_Mapping .....	776
C.2.5.13.2.4 ExpressionElse_Mapping .....	778
C.2.5.13.2.5 ExpressionElseMembership_Mapping .....	780
C.2.5.13.2.6 ExpressionElseSpecification_Mapping .....	781
C.2.5.13.2.7 LiteralBoolean_Mapping .....	782
C.2.5.13.2.8 LiteralBooleanTrue_Mapping .....	784
C.2.5.13.2.9 LiteralInteger_Mapping .....	785
C.2.5.13.2.10 LiteralNull_Mapping .....	787
C.2.5.13.2.11 LiteralReal_Mapping .....	789
C.2.5.13.2.12 LiteralSpecification_Mapping .....	791
C.2.5.13.2.13 LiteralSpecificationCommon_Mapping .....	792
C.2.5.13.2.14 LiteralSpecificationTyping_Mapping .....	794
C.2.5.13.2.15 LiteralString_Mapping .....	795
C.2.5.13.2.16 LiteralUnlimitedToUnbounded_Mapping .....	797
C.2.5.13.2.17 LiteralUnlimitedToInteger_Mapping .....	798
C.2.5.13.2.18 OpaqueExpressionAsValue_Mapping .....	800
C.2.5.13.2.19 OpaqueExpression_Mapping .....	802
C.2.5.13.2.20 OpaqueExpressionFeature_Mapping .....	804
C.2.5.13.2.21 OpaqueExpressionFeatureFeature_Mapping .....	805
C.2.5.13.2.22 OpaqueExpressionFeatureFeatureMembership_Mapping .....	806
C.2.5.13.2.23 OpaqueExpressionFeatureValue_Mapping .....	807
C.2.5.13.2.24 OpaqueExpressionFeatureValueExpression_Mapping .....	808
C.2.5.13.2.25 OpaqueExpressionFeatureValueExpressionMembership_Mapping .....	810
C.2.5.13.2.26 OpaqueExpressionMembership_Mapping .....	811
C.2.5.13.2.27 OpaqueExpressionParameterMembership_Mapping .....	812
C.2.5.13.2.28 OpaqueExpressionReturnParameterMembershipReferenceUsage_Mapping .....	814
C.2.5.13.2.29 OpaqueExpressionReturnParameterReferenceUsage_Mapping .....	815
C.2.5.13.2.30 OpaqueExpressionReturnParameterReferenceUsageFeatureTyping_Mapping .....	817
C.2.5.13.2.31 OpaqueExpressionReturnParameterReferenceUsageUntyped_Mapping .....	818
C.2.5.13.2.32 OpaqueExpressionSpecification_Mapping .....	820
C.2.5.13.2.33 TimeExpression_Mapping .....	821
C.2.5.13.2.34 ValueSpecification_Mapping .....	822

# List of Tables

1. List of all Overview Mapping Specifications .....	11
2. List of all GenericToOwningMembership_Mapping Mapping Specifications .....	37
3. List of all GenericToSubclassification_Mapping Mapping Specifications .....	45
4. List of all Overview Mapping Specifications .....	72
5. List of all Overview Mapping Specifications .....	96
6. List of all Overview Mapping Specifications .....	97
7. List of all Overview Mapping Specifications .....	108
8. List of SysML v1 elements not mapped of this section .....	109
9. List of all Overview Mapping Specifications .....	127
10. List of all Overview Mapping Specifications .....	173
11. List of all Overview Mapping Specifications .....	193
12. List of SysML v1 elements not mapped of this section .....	193
13. List of all Overview Mapping Specifications .....	231
14. List of SysML v1 elements not mapped of this section .....	237
15. List of all Overview Mapping Specifications .....	390
16. List of SysML v1 elements not mapped of this section .....	394
17. List of all Overview Mapping Specifications .....	471
18. List of all Overview Mapping Specifications .....	549
19. List of all Trigger_Mapping Mapping Specifications .....	560
20. List of all Overview Mapping Specifications .....	561
21. List of all Overview Mapping Specifications .....	584
22. List of all Overview Mapping Specifications .....	584
23. List of all Overview Mapping Specifications .....	596
24. List of SysML v1 elements not mapped of this section .....	598
25. List of all Overview Mapping Specifications .....	662
26. List of all Overview Mapping Specifications .....	693
27. List of all Overview Mapping Specifications .....	750
28. List of SysML v1 elements not mapped of this section .....	751
29. List of all Overview Mapping Specifications .....	771





**A**



**B**





# C Annex: SysML v1 to SysML v2 Transformation

## C.1 General

### C.1.1 Overview

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

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

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

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

### C.1.2 Mapping Approach

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

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

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

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

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

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

It is possible to restrict the applicability of a mapping specification to a specific subset of objects. This is achieved by the "filter" static operation that is evaluated against each candidate object. Only objects for which this "filter"

operation returns “true” shall be translated according to the specifications of that mapping class. By default, the filter operation always returns “true”.

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

## C.2 Mappings

### C.2.1 Overview

### C.2.2 Mapping Helper and Library

#### C.2.2.1 Helper

##### Description

##### Operations

- `actionOwnedRelationship (in src : Element) : Relationship [0..*]`

```
bodyCondition:
result = let actionInputPin: Set(UML::Element) = src.ownedElement-
>select(e | e.ocIsTypeOf(UML::ActionInputPin)) in let triggers:
Set(UML::Element) = src.ownedElement->select(e |
e.ocIsKindOf(UML::Trigger)) in let toElementFMS: Set(UML::Element) =
src.ownedElement->select(e | e.ocIsKindOf(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..*]`

```
bodyCondition:
result = let initialNodes : Set(UML::Element) = src.ownedElement-
>select(e | e.ocIsKindOf(UML::InitialNode)) in let activityFinalNodes
: Set(UML::Element) = src.ownedElement->select(e |
e.ocIsKindOf(UML::ActivityFinalNode)) in let objectFlowsWithGuard :
Set(UML::ObjectFlow) = src.ownedElement->select(e |
e.ocIsKindOf(UML::ObjectFlow) and not
e.ocIsType(UML::ObjectFlow).guard.ocIsUndefined()) in let objectFlows
: Set(UML::ObjectFlow) = src.ownedElement->select(e |
e.ocIsKindOf(UML::ObjectFlow)) in let elementsFMS : Set(UML::Element)
= ((src.ownedElement->select(e | e.ocIsKindOf(UML::ControlNode) or
e.ocIsKindOf(UML::Action) or e.ocIsKindOf(UML::ControlFlow)) -
initialNodes) - activityFinalNodes) in let parameters:
Set(UML::Parameter) = src.ownedElement->select(e |
e.ocIsKindOf(UML::Parameter)) in let ignoreParameterNodes:
Set(UML::ActivityParameterNode) = src.ownedElement->select(e |
e.ocIsKindOf(UML::ActivityParameterNode)) in let
```

```

ignoreActivityPartition: Set(UML::ActivityPartition) =
src.ownedElement->select(e | e.ocIsKindOf(UML::ActivityPartition)) in
let ignoreInterruptibleActivityRegion:
Set(UML::InterruptibleActivityRegion) = src.ownedElement->select(e |
e.ocIsKindOf(UML::InterruptibleActivityRegion)) in let
ownedClassifier: Sequence(UML::Classifier) = src.ownedElement->select(e
| e.ocIsKindOf(UML::Classifier)) in let properties:
Sequence(UML::Property) = src.ownedElement->select(e |
e.ocIsKindOf(UML::Property)) in let variables: Sequence(UML::Variable)
= src.ownedElement->select(e | e.ocIsKindOf(UML::Variable)) in let
parameterSets: Set(UML::ParameterSet) = src.ownedElement->select(e |
e.ocIsKindOf(UML::ParameterSet)) in let elementsOMS: Set(UML::Element)
= (((((((((((src.ownedElement-initialNodes)-activityFinalNodes)-
objectFlowsWithGuard)-objectFlows)-elementsFMS)-parameters)-
ignoreParameterNodes)-ignoreActivityPartition)-
ignoreInterruptibleActivityRegion)-ownedClassifier)-properties)-
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(activityFinalNodes->collect(e |
ActivityFinalNodeMembership_Mapping.getMapped(e))) ->union(elementsFMS-
>collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(objectFlowsWithGuard->collect(e |
ObjectFlowGuardFeatureMembership_Mapping.getMapped(e)))
->union(objectFlows->collect(e |
ObjectFlowFeatureMembership_Mapping.getMapped(e))) ->union(properties-
>collect(e | PropertyMembership_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.ocIsUndefined() then memberships else
memberships-
>append(ClassifierBehaviorMembership_Mapping.getMapped(src)) endif
• createUUID () : String [1]
• getEnumerationType (in t : Enumeration) : EnumerationDefinition [1]

bodyCondition:
result = if t.name = 'VerdictKind' then
SYSML2::EnumerationDefinition.allInstances()->any(e | e.qualifiedName =
'VerificationCases::VerdictKind') else
SYSML2::EnumerationDefinition.allInstances()->any(e | e.qualifiedName =
'SysMLv1Library::Enumerations::' + t.name) endif
• getID (in src : Element) : String [1]
• getKerMLFeatureDirectionKind (in v : EnumerationLiteral) : FeatureDirectionKind [1]

bodyCondition:
result = if v.enumeration.qualifiedName =
'SysML::Ports&Flows::FeatureDirectionKind' or

```

```

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

```

- **getKerMLParameterDirectionKind** (in v : ParameterDirectionKind) : FeatureDirectionKind [1]

```

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

```
- **getKerMLVisibilityKind** (in v : VisibilityKind) : VisibilityKind [1]

```

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

```
- **getMetadataByName** (in mdName : String) : AttributeDefinition [1]

```

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

```
- **getScalarValueType** (in t : DataType) : DataType [1]

```

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

```
- **getScalarValueTypeByName** (in ptName : String) : DataType [1]

```

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

```
- **getSysMLv2EnumerationDefinition** (in v1Enumeration : Enumeration) : EnumerationDefinition [1]

```

bodyCondition:
result = if v1Enumeration = UML::ParameterDirectionKind then
KerML::FeatureDirectionKind else invalid endif

```
- **getTagValue** (in element : Element, in stereotypeName : String, in tagValueName : String) [1]

- `getTagValueAsElement` (in `element : Element`, in `stereotypeName : String`, in `tagValueName : String`) : `Element [1]`
- `getTagValueAsElementColl` (in `element : Element`, in `stereotypeName : String`, in `tagValueName : String`) : `Element [0..*]`
- `getTagValueAsString` (in `element : Element`, in `stereotypeName : String`, in `tagValueName : String`) : `String [1]`
- `getTagValueAsStringColl` (in `element : Element`, in `stereotypeName : String`, in `tagValueName : String`) : `String [0..*]`
- `globalNamespace ()` : `Namespace [1]`

`bodyCondition:`

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

- `hasStereotypeApplied` (in `element : Element`, in `stereotypeName : String`) : `Boolean [1]`
- `isConnectionDef` (in `association : Association`) : `Boolean [1]`

`bodyCondition:`

```
result = -- Case 1: composite association with multiplicity 1..1 on
owner side let case1: Boolean = association.memberEnd->exists(e | not
e.isComposite and e.lower=1) and association.memberEnd->exists(e |
e.isComposite) in -- Case 2: association is not composite and there is
no owned end with multiplicity 0..* let case2: Boolean = not
association.memberEnd->exists(e | e.isComposite) and not
association.ownedEnd->exists(e | e.lower = 0 and e.upper = -1) in
association.oclIsTypeOf(UML::AssociationClass) or case1 or case2
```

- `packageOwnedRelationship` (in `src : Element`) : `Relationship [0..*]`

`bodyCondition:`

```
result = let elementGroups: Set(UML::Comment) = src.ownedElement-
>select(e | Helper.hasStereotypeApplied(e,
'SysML::ModelElements::ElementGroup')) in let copyRelationship:
Set(UML::Abstraction) = src.ownedElement->select(e |
Helper.hasStereotypeApplied(e, 'SysML::Requirements::Copy')) in let
verifyRelationship: Set(UML::Abstraction) = src.ownedElement->select(e
| Helper.hasStereotypeApplied(e, 'SysML::Requirements::Verify')) in let
useCaseAssociations : Set(UML::Association) = src.ownedElement-
>select(e | e.oclIsKindOf(UML::Association))->collect(m | m.memberEnd)-
>flatten()->select( m | m.type.oclIsKindOf(UML::UseCase))->collect(a |
a.association) in let relationships: Set(SysMLv2::Relationship) =
(((src.ownedElement - elementGroups) - copyRelationship) -
verifyRelationship) - useCaseAssociations ->reject(e |
e.oclIsKindOf(UML::ProfileApplication) or
e.oclIsKindOf(UML::GeneralizationSet) or
e.oclIsKindOf(UML::SignalEvent) or e.oclIsKindOf(UML::CallEvent) or
e.oclIsKindOf(UML::ChangeEvent) or e.oclIsKindOf(UML::Extension) or
e.oclIsKindOf(UML::PackageMerge) or
(e.oclIsKindOf(UML::InstanceSpecification) and
e.oclAsType(UML::InstanceSpecification).classifier->size() = 0))
->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(elementGroups->collect(e |
ElementGroupMembership_Mapping.getMapped(e))) in if
```

```

src.URI.ocIsUndefined() or src.URI = '' then relationships else
relationships-
>including (PackageURIMetadataMembership_Mapping.getMapped(src)) endif
• stateOwnedRelationship (in src : Element) : Relationship [0..*]

bodyCondition:
result = let initialState : Set(UML::Element) = from.ownedElement-
>select(e | e.ocIsKindOf(UML::Pseudostate) and
e.ocIsType(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)))

```

### C.2.2.2 SysML v1 Library

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

```

package SysMLv1Library {

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

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

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

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

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

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

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

  metadata def ParameterSetData {
    doc /* Metadata definition to tag parameter that the mapping source of the parameter

```

```

        attribute isParameterSet : ScalarValue::Boolean;
    }

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

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

```

## C.2.3 Generic Mappings

### C.2.3.1 Overview

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

All of those generic mappings are abstract.

### C.2.3.2 Generic Mappings To KerML

#### C.2.3.2.1 Overview

**Table 1. List of all Overview Mapping Specifications**

Mapping Class	SysML v2 Concept
GenericToAnnotatingElement_Mapping	AnnotatingElement
GenericToAnnotation_Mapping	Annotation
GenericToAssociation_Mapping	Association
GenericToBehavior_Mapping	Behavior
GenericToClassifier_Mapping	Classifier
GenericToComment_Mapping	Comment
GenericToConjugation_Mapping	Conjugation
GenericToConnector_Mapping	Connector
GenericToDocumentation_Mapping	Documentation
GenericToElement_Mapping	Element
GenericToEndFeatureMembership_Mapping	EndFeatureMembership
GenericToExpression_Mapping	Expression

Mapping Class	SysML v2 Concept
GenericToFeature_Mapping	Feature
GenericToFeatureChaining_Mapping	FeatureChaining
GenericToFeatureMembership_Mapping	FeatureMembership
GenericToFeatureReferenceExpression_Mapping	FeatureReferenceExpression
GenericToFeatureTyping_Mapping	FeatureTyping
GenericToFeatureValue_Mapping	FeatureValue
GenericToFunction_Mapping	Function
GenericToImport_Mapping	Import
GenericToInvocationExpression_Mapping	InvocationExpression
GenericToMembership_Mapping	Membership
GenericToNamespace_Mapping	Namespace
GenericToOwningMembership_Mapping	OwningMembership
GenericToPackage_Mapping	Package
GenericToParameterMembership_Mapping	ParameterMembership
GenericToPredicate_Mapping	Predicate
GenericToRedefinition_Mapping	Redefinition
GenericToRelationship_Mapping	Relationship
GenericToReturnParameterMembership_Mapping	ReturnParameterMembership
GenericToSpecialization_Mapping	Specialization
GenericToStep_Mapping	Step
GenericToSubclassification_Mapping	Subclassification
GenericToSubsetting_Mapping	Subsetting
GenericToTextualRepresentation_Mapping	TextualRepresentation
GenericToType_Mapping	Type
GenericToTypeFeaturing_Mapping	TypeFeaturing

### C.2.3.2.2 Mapping Specifications

#### C.2.3.2.2.1 GenericToAnnotatingElement\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToElement\_Mapping

##### Mapping Source



### Mapping Target

AnnotatingElement

### Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- AnnotatingElement::annotation

Set { }

### C.2.3.2.2.2 GenericToAnnotation\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

GenericToRelationship\_Mapping

#### Mapping Source

#### Mapping Target

Annotation

#### Owned Mappings

(none)

#### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

#### Mapping rules

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

- Element::aliasId

Set { }

- Annotation::owningAnnotatedElement

null

- Element::name

null

- Annotation::annotatingElement  
*abstract rule*
- Element::shortName

null

- Annotation::annotatedElement  
*abstract rule*
- Element::elementId

Helper.createUUID()

- Element::ownedRelationship

Set{}

#### C.2.3.2.2.3 GenericToAssociation\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToRelationship\_Mapping  
GenericToClassifier\_Mapping

##### Mapping Source

##### Mapping Target

Association

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId

Set{}

- Type::isSufficient

false

- Element::name

null

- Element::shortName

null

- Type::isAbstract

false

- Element::elementId

Helper.createUUID()

- Element::ownedRelationship

Set{}

#### C.2.3.2.2.4 GenericToBehavior\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToClassifier\_Mapping

##### Mapping Source

##### Mapping Target

Behavior

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId

Set{}

- Type::isSufficient

false

- Element::name  
null
- Element::shortName  
null
- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Element::ownedRelationship  
Set{}

#### **C.2.3.2.2.5 GenericToClassifier\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToType\_Mapping

##### **Mapping Source**

##### **Mapping Target**

Classifier

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Element::aliasId  
Set{}
- Element::name  
null
- Element::shortName

null

- Element::elementId

    Helper.createUUID()

- Element::ownedRelationship

    Set{}

#### **C.2.3.2.2.6 GenericToComment\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToAnnotatingElement\_Mapping

##### **Mapping Source**

##### **Mapping Target**

Comment

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Element::aliasId

    Set{}

- Comment::locale

    null

- Element::name

    null

- Comment::body

*abstract rule*

- Element::shortName

    null

- Element::elementId

```
Helper.createUUID()
```

- Element::ownedRelationship

```
Set{ }
```

#### **C.2.3.2.2.7 GenericToConjugation\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToRelationship\_Mapping

##### **Mapping Source**

##### **Mapping Target**

Conjugation

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- Element::aliasId

```
Set{ }
```

- Conjugation::conjugatedType

*abstract rule*

- Conjugation::originalType

*abstract rule*

- Element::name

```
null
```

- Element::shortName

```
null
```

- Element::elementId

```
Helper.createUUID()
```

- Element::ownedRelationship

Set{}

#### C.2.3.2.2.8 GenericToConnector\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeature\_Mapping

GenericToRelationship\_Mapping

##### Mapping Source

##### Mapping Target

Connector

##### Owned Mappings

(none)

##### Applicable filters

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

(none)

##### Mapping rules

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

- Element::aliasId  
Set{}
- Type::isSufficient  
false
- Connector::isDirected  
false
- Element::name  
null
- Element::shortName  
null
- Type::isAbstract  
false
- Element::elementId

```
Helper.createUUID()
```

- Element::ownedRelationship

```
Set{}
```

#### **C.2.3.2.2.9 GenericToDocumentation\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToComment\_Mapping

##### **Mapping Source**

##### **Mapping Target**

Documentation

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- Element::aliasId

```
Set{}
```

- Element::name

```
null
```

- Element::shortName

```
null
```

- AnnotatingElement::annotation

```
Set{}
```

- Element::elementId

```
Helper.createUUID()
```

- Element::ownedRelationship

```
Set{}
```



#### **C.2.3.2.2.10 GenericToElement\_Mapping**

##### **Description**

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

##### **General Mappings**

No general mappings.

##### **Mapping Source**

##### **Mapping Target**

Element

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Element::aliasId  
`Set {}`
- Element::name  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set {}`

#### **C.2.3.2.2.11 GenericToEndFeatureMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureMembership\_Mapping

## Mapping Source

## Mapping Target

EndFeatureMembership

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
    null
- Element::shortName  
  
    null
- Element::elementId  
  
    Helper.createUUID()
- Element::aliasId  
  
    Set{ }
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
  
    null
- OwningMembership::ownedRelatedElement  
  
    Set{ self.ownedMemberElement() }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
  
    KerML::VisibilityKind::public
- Element::name

null

- Element::ownedRelationship

Set{}

#### **C.2.3.2.2.12 GenericToExpression\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToStep\_Mapping

##### **Mapping Source**

##### **Mapping Target**

Expression

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- Feature::isEnd

false

- Type::isSufficient

false

- Feature::isUnique

true

- Element::shortName

null

- Type::isAbstract

false

- Element::elementId

Helper.createUUID()

- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false
- Element::ownedRelationship  
Set{}

#### **C.2.3.2.2.13 GenericToFeature\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToType\_Mapping

##### **Mapping Source**

##### **Mapping Target**

Feature

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Feature::isEnd  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Feature::isReadOnly  
`false`
- Feature::direction  
`null`
- Element::name  
`null`
- Feature::isDerived  
`false`
- Feature::isComposite  
`false`
- Element::ownedRelationship  
`Set{}`

#### C.2.3.2.2.14 GenericToFeatureChaining\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToRelationship\_Mapping

##### Mapping Source

##### Mapping Target

FeatureChaining

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId  
`Set {}`
- Element::name  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- FeatureChaining::chainingFeature  
*abstract rule*
- Element::ownedRelationship  
`Set {}`

#### C.2.3.2.2.15 GenericToFeatureMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

## General Mappings

GenericToOwningMembership\_Mapping

GenericToTypeFeaturing\_Mapping

## Mapping Source

## Mapping Target

FeatureMembership

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Relationship::ownedRelatedElement  
`Set{ }`
- FeatureMembership::ownedRelatedElement  
`Set{ self.ownedMemberFeature() }`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Membership::memberElement  
*abstract rule*
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{ }`
- FeatureMembership::ownedMemberFeature  
*abstract rule*
- FeatureMembership::owningType  
*abstract rule*
- Membership::memberName

null

- Membership::visibility

KerML::VisibilityKind::public

- Element::name

null

- Element::ownedRelationship

Set{}

#### **C.2.3.2.2.16 GenericToFeatureReferenceExpression\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToExpression\_Mapping

##### **Mapping Source**

##### **Mapping Target**

FeatureReferenceExpression

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Feature::isEnd

false

- Type::isSufficient

false

- Feature::isUnique

true

- Element::shortName

null



- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false
- Element::ownedRelationship  
Set{}

#### **C.2.3.2.2.17 GenericToFeatureTyping\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToSpecialization\_Mapping

##### **Mapping Source**

##### **Mapping Target**

FeatureTyping

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Element::aliasId  
`Set {}`
- Relationship::ownedRelatedElement  
`Set {}`
- Relationship::source  
`Set {}`
- Element::name  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Relationship::target  
`Set {}`
- FeatureTyping::type  
*abstract rule*
- FeatureTyping::typedFeature  
*abstract rule*
- Element::ownedRelationship  
`Set {}`

### C.2.3.2.2.18 GenericToFeatureValue\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

GenericToOwningMembership\_Mapping

## Mapping Source

## Mapping Target

FeatureValue

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Relationship::ownedRelatedElement  
`Set{}`
- FeatureValue::isDefault  
`false`
- FeatureValue::isInitial  
`false`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Membership::memberElement  
*abstract rule*
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- FeatureValue::ownedRelatedElement  
`Set{self.value() }`
- Element::aliasId  
`Set{}`
- Membership::memberName  
`null`

- FeatureValue::featureWithValue  
*abstract rule*
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- FeatureValue::value  
*abstract rule*
- Element::ownedRelationship  
  
Set{ }

#### C.2.3.2.2.19 GenericToFunction\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToBehavior\_Mapping

##### Mapping Source

##### Mapping Target

Function

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId  
  
Set{ }
- Type::isSufficient  
  
false
- Element::name  
  
null

- Element::shortName  
null
- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Element::ownedRelationship  
Set{ }

#### C.2.3.2.2.20 GenericToImport\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToRelationship\_Mapping

##### Mapping Source

##### Mapping Target

Import

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId  
Set{ }
- Import::importedNamespace  
*abstract rule*
- Import::importedMemberName  
null
- Import::isRecursive  
false

- `Import::visibility`  
`KerML::VisibilityKind::public`
- `Element::name`  
`null`
- `Element::shortName`  
`null`
- `Element::elementId`  
`Helper.createUUID()`
- `Import::isImportAll`  
`false`
- `Element::ownedRelationship`  
`Set{}`

#### **C.2.3.2.2.21 GenericToInvocationExpression\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToExpression\_Mapping

##### **Mapping Source**

##### **Mapping Target**

InvocationExpression

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- `Feature::isEnd`  
`false`
- `Type::isSufficient`

- `false`
- `Feature::isUnique`  
`true`
- `Element::shortName`  
`null`
- `Type::isAbstract`  
`false`
- `Element::elementId`  
`Helper.createUUID()`
- `Feature::isOrdered`  
`false`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `Feature::isReadOnly`  
`false`
- `Feature::direction`  
`null`
- `Element::name`  
`null`
- `Feature::isDerived`  
`false`
- `Feature::isComposite`  
`false`
- `Element::ownedRelationship`  
`Set{}`

#### **C.2.3.2.2.22 GenericToMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

## General Mappings

GenericToRelationship\_Mapping

## Mapping Source

## Mapping Target

Membership

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Element::aliasId  
`Set{}`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Membership::memberName  
`null`
- Membership::memberElement  
*abstract rule*
- Membership::visibility  
`KerML::VisibilityKind::public`
- Element::name  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship



Set { }

#### **C.2.3.2.2.23 GenericToNamespace\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToElement\_Mapping

##### **Mapping Source**

##### **Mapping Target**

Namespace

##### **Owned Mappings**

(none)

#### **C.2.3.2.2.24 GenericToOwningMembership\_Mapping**

#### **C.2.3.2.2.25 GenericToPackage\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToNamespace\_Mapping

##### **Mapping Source**

##### **Mapping Target**

Package

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Element::aliasId

Set { }

- Element::name  
null
- Element::shortName  
null
- Element::elementId  
Helper.createUUID()
- Element::ownedRelationship  
Set{}

#### **C.2.3.2.2.26 GenericToParameterMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureMembership\_Mapping

##### **Mapping Source**

##### **Mapping Target**

ParameterMembership

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
null
- Element::shortName  
null
- Element::elementId  
Helper.createUUID()

- Element::aliasId  
Set{ }
- OwningMembership::ownedMemberElement  
*abstract rule*
- ParameterMembership::ownedMemberParameter  
null
- Membership::memberName  
null
- OwningMembership::ownedRelatedElement  
Set{ self.ownedMemberElement() }
- ParameterMembership::ownedRelatedElement  
Set{ self.ownedMemberParameter() }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
KerML::VisibilityKind::public
- Element::name  
null
- Element::ownedRelationship  
Set{ }

#### C.2.3.2.2.27 GenericToPredicate\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFunction\_Mapping

##### Mapping Source

##### Mapping Target

Predicate

##### Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  
`Set{}`
- Type::isSufficient  
`false`
- Element::name  
`null`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set{}`

#### C.2.3.2.2.28 GenericToRedefinition\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToSubsetting\_Mapping

##### Mapping Source

##### Mapping Target

Redefinition

##### Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  
`Set {}`
- Relationship::ownedRelatedElement  
`Set {}`
- Specialization::specific  
*abstract rule*
- Redefinition::redefinedFeature  
*abstract rule*
- Element::name  
`null`
- Element::shortName  
`null`
- Specialization::general  
*abstract rule*
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set {}`
- Redefinition::redefiningFeature  
*abstract rule*

#### C.2.3.2.2.29 GenericToRelationship\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToElement\_Mapping

##### Mapping Source

##### Mapping Target

Relationship

##### Owned Mappings

(none)

#### **Applicable filters**

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

(none)

#### **Mapping rules**

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

- Relationship::ownedRelatedElement

`Set { }`

- Relationship::source

`Set { }`

- Relationship::target

`Set { }`

### **C.2.3.2.2.30 GenericToReturnParameterMembership\_Mapping**

#### **Description**

\*\*\* not specified yet \*\*\*

#### **General Mappings**

GenericToParameterMembership\_Mapping

#### **Mapping Source**

#### **Mapping Target**

ReturnParameterMembership

#### **Owned Mappings**

(none)

#### **Applicable filters**

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

(none)

#### **Mapping rules**

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

- FeatureMembership::ownedRelatedElement

`Set { self.ownedMemberFeature ( ) }`

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName

null

- Element::shortName

null

- Element::elementId

Helper.createUUID()

- ReturnParameterMembership::isComposite

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

- Element::aliasId

Set{}

- FeatureMembership::ownedMemberFeature  
*abstract rule*

- FeatureMembership::owningType  
*abstract rule*

- Membership::memberName

null

- TypeFeaturing::featureOfType  
*abstract rule*

- TypeFeaturing::featuringType  
*abstract rule*

- Membership::visibility

KerML::VisibilityKind::public

- Element::name

null

- Element::ownedRelationship

Set{}

#### C.2.3.2.2.31 GenericToSpecialization\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToRelationship\_Mapping

**Mapping Source****Mapping Target**

Specialization

**Owned Mappings**

(none)

**Applicable filters**

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

(none)

**Mapping rules**

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

- Element::aliasId  
`Set{}`
- Specialization::specific  
*abstract rule*
- Element::name  
`null`
- Element::shortName  
`null`
- Specialization::general  
*abstract rule*
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set{}`

**C.2.3.2.2.32 GenericToStep\_Mapping****Description**

\*\*\* not specified yet \*\*\*

**General Mappings**

GenericToFeature\_Mapping

**Mapping Source****Mapping Target**



Step

### Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  
`Set{}`
- Type::isSufficient  
`false`
- Element::name  
`null`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set{}`

#### C.2.3.2.2.33 GenericToSubclassification\_Mapping

#### C.2.3.2.2.34 GenericToSubsetting\_Mapping

### Description

\*\*\* not specified yet \*\*\*

### General Mappings

GenericToSpecialization\_Mapping

### Mapping Source

### Mapping Target

Subsetting

### Owned Mappings

(none)

### Applicable filters

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

(none)

### Mapping rules

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

- Element::aliasId  
`Set {}`
- Relationship::source  
`Set {}`
- Subsetting::ownedRelatedElement  
`Set {}`
- Subsetting::subsettingFeature  
*abstract rule*
- Element::name  
`null`
- Subsetting::subsettingFeature  
*abstract rule*
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Relationship::target  
`Set {}`
- Element::ownedRelationship  
`Set {}`

#### C.2.3.2.2.35 GenericToTextualRepresentation\_Mapping

### Description

\*\*\* not specified yet \*\*\*

### General Mappings

GenericToAnnotatingElement\_Mapping

### Mapping Source

### Mapping Target

TextualRepresentation

### Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  
`Set {}`
- TextualRepresentation::language  
*abstract rule*
- Element::name  
`null`
- Element::shortName  
`null`
- TextualRepresentation::body  
*abstract rule*
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set {}`

#### C.2.3.2.2.36 GenericToType\_Mapping

### Description

\*\*\* not specified yet \*\*\*

### General Mappings

GenericToNamespace\_Mapping

### Mapping Source

## Mapping Target

Type

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Element::aliasId  
`Set{}`
- Type::isSufficient  
`false`
- Element::name  
`null`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set{}`

### C.2.3.2.2.37 GenericToTypeFeaturing\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

GenericToRelationship\_Mapping

#### Mapping Source

#### Mapping Target

TypeFeaturing

### Owned Mappings

(none)

### Applicable filters

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

(none)

### Mapping rules

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

- Element::aliasId  
`Set {}`
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Element::name  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set {}`

## C.2.3.3 Generic Mappings FromTo KerML

### C.2.3.3.1 Overview

### C.2.3.3.2 Mapping Specifications

#### C.2.3.3.2.1 CommonMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

### General Mappings

GenericToMembership\_Mapping

### Mapping Source

TypedElement

## Mapping Target

Membership

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Element::aliasId  
`Set {}`
- Relationship::ownedRelatedElement  
`Set {}`
- Relationship::source  
`Set {}`
- Membership::memberElement  
`from`
- Element::name  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Relationship::target  
`Set {}`
- Element::ownedRelationship  
`Set {}`

### C.2.3.3.2.2 CommonParameterReferenceUsageInMembership\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

## General Mappings

GenericToParameterMembership\_Mapping

## Mapping Source

Element

## Mapping Target

ParameterMembership

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- FeatureMembership::ownedRelatedElement  
`Set{self.ownedMemberFeature() }`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{ }`
- FeatureMembership::ownedMemberFeature  
*abstract rule*
- FeatureMembership::owningType  
*abstract rule*
- Membership::memberName  
`null`
- TypeFeaturing::featureOfType  
*abstract rule*

- TypeFeaturing::featuringType  
*abstract rule*
- ParameterMembership::ownedMemberParameter

```
if not from.ocIsKindOf(UML::TypedElement) then CommonParameterReferenceUsageIn_Mapping.getMa
else if from.ocIsType(UML::TypedElement).type.ocIsUndefined() then CommonParameterReference
else CommonParameterReferenceUsageInUntyped_Mapping.getMapped(from) endif endif
```

- Membership::visibility

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

- Element::name

```
null
```

- Element::ownedRelationship

```
Set{}
```

### C.2.3.3.2.3 CommonParameterReferenceUsageIn\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

CommonParameterReferenceUsageInUntyped\_Mapping

#### Mapping Source

Element

#### Mapping Target

ReferenceUsage

#### Owned Mappings

- commonParameterReferenceUsageInFeatureTyping :  
CommonParameterReferenceUsageInFeatureTyping\_Mapping

#### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

#### Mapping rules

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

- Feature::isEnd

```
false
```



- `Type::isSufficient`  
`false`
- `Feature::isUnique`  
`true`
- `Element::shortName`  
`null`
- `Type::isAbstract`  
`false`
- `Element::elementId`  
`Helper.createUUID()`
- `Feature::isOrdered`  
`false`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `Usage::isVariation`  
`false`
- `Feature::isReadOnly`  
`false`
- `ReferenceUsage::ownedRelationship`  
`if from.ocIsKindOf(UML::TypedElement) then Set{commonParameterReferenceUsageInFeatureTyping`
- `Feature::direction`  
`null`
- `Element::name`  
`null`
- `Feature::isDerived`  
`false`
- `Feature::isComposite`  
`false`

#### C.2.3.3.2.4 CommonParameterReferenceUsageInUntyped\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### 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:  
(none)

##### Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- ReferenceUsage::direction  
`KerML::FeatureDirectionKind::_in'`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`

- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Usage::isVariation  
false
- Feature::isReadOnly  
false
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false
- Element::ownedRelationship  
Set{}

#### **C.2.3.3.2.5 CommonReferenceUsageInFeatureTyping\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureTyping\_Mapping

##### **Mapping Source**

TypedElement

##### **Mapping Target**

FeatureTyping

##### **Owned Mappings**

- commonReferenceUsageIn : CommonReferenceUsageIn\_Mapping

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  

```
Set{}
```
- Relationship::ownedRelatedElement  

```
Set{}
```
- Specialization::specific  
*abstract rule*
- FeatureTyping::typedFeature  

```
commonReferenceUsageIn.to
```
- FeatureTyping::type  

```
if from.type.ocIsKindOf(UML::PrimitiveType) then
    Helper.getScalarValueType(from.type)
else
    from.type
endif
```
- Element::name  

```
null
```
- Element::shortName  

```
null
```
- Specialization::general  
*abstract rule*
- Element::elementId  

```
Helper.createUUID()
```
- Element::ownedRelationship  

```
Set{}
```

#### C.2.3.3.2.6 CommonReferenceUsageInUntyped\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToReferenceUsage\_Mapping

### Mapping Source

TypedElement

### Mapping Target

ReferenceUsage

### Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- ReferenceUsage::name  
`from.name`
- ReferenceUsage::direction  
`KerML::FeatureDirectionKind::_in'`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`

- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Usage::isVariation  
false
- Feature::isReadOnly  
false
- Feature::isDerived  
false
- Feature::isComposite  
false
- Element::ownedRelationship  
Set{}

#### **C.2.3.3.2.7 CommonReturnParameterFeature\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

CommonReturnParameterFeatureUntyped\_Mapping

##### **Mapping Source**

Element

##### **Mapping Target**

Feature

##### **Owned Mappings**

- commonReturnParameterFeatureTyping : CommonReturnParameterFeatureTyping\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Feature::ownedRelationship  
`if from.ocIsKindOf(UML::Property) then Set{commonReturnParameterFeatureTyping.to} else Set{}`
- Feature::isReadOnly  
`false`
- Feature::direction  
`null`
- Element::name  
`null`
- Feature::isDerived  
`false`
- Feature::isComposite

false

#### C.2.3.3.2.8 CommonReturnParameterFeatureTyping\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureTyping\_Mapping

##### Mapping Source

Element

##### Mapping Target

FeatureTyping

##### Owned Mappings

- commonReturnParameterFeature : CommonReturnParameterFeature\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId  

```
Set{}
```
- Relationship::ownedRelatedElement  

```
Set{}
```
- FeatureTyping::typedFeature  

```
commonReturnParameterFeature.to
```
- Specialization::specific  
*abstract rule*
- FeatureTyping::type  

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



- Element::name  
null
- Element::shortName  
null
- Specialization::general  
*abstract rule*
- Element::elementId  
Helper.createUUID()
- Element::ownedRelationship  
Set{}

#### C.2.3.3.2.9 CommonReturnParameterFeatureUntyped\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeature\_Mapping

##### Mapping Source

Element

##### Mapping Target

Feature

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId  
Set{}
- Type::isSufficient  
false

- Element::name  
null
- Feature::direction  
KerML::FeatureDirectionKind::\_'out'
- Element::shortName  
null
- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Element::ownedRelationship  
Set{}

#### **C.2.3.3.2.10 CommonReturnParameterFeatureMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToReturnParameterMembership\_Mapping

##### **Mapping Source**

Element

##### **Mapping Target**

ReturnParameterMembership

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Membership::membershipOwningNamespace  
*abstract rule*

- Membership::memberShortName

null

- Element::shortName

null

- Element::elementId

Helper.createUUID()

- Element::aliasId

Set{}

- FeatureMembership::owningType

*abstract rule*

- ReturnParameterMembership::ownedMemberParameter

```
if not from.ocIsKindOf(UML::TypedElement) then CommonReturnParameterFeatureUntyped_Mapping.g
else if from.ocIsType(UML::TypedElement).type.ocIsUndefined() then CommonReturnParameterFea
else CommonReturnParameterFeatureUntyped_Mapping.getMapped(from) endif endif
```

- Membership::memberName

null

- ParameterMembership::ownedRelatedElement

Set{self.ownedMemberParameter() }

- TypeFeaturing::featureOfType

*abstract rule*

- TypeFeaturing::featuringType

*abstract rule*

- Membership::visibility

KerML::VisibilityKind::public

- Element::name

null

- Element::ownedRelationship

Set{}

#### C.2.3.3.2.11 CommonReturnParameterReferenceUsageMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToReturnParameterMembership\_Mapping

## Mapping Source

Element

## Mapping Target

ReturnParameterMembership

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- ReturnParameterMembership::ownedMemberParameter

```
if not from.ocIsKindOf(UML::TypedElement) then CommonReturnParameterReferenceUsageUntyped_Ma
else if from.ocIsType(UML::TypedElement).type.ocIsUndefined() then CommonReturnParameterRef
else CommonReturnParameterReferenceUsageUntyped_Mapping.getMapped(from) endif endif
```

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName

```
null
```

- Element::shortName

```
null
```

- Element::elementId

```
Helper.createUUID()
```

- Element::aliasId

```
Set{}
```

- FeatureMembership::owningType  
*abstract rule*
- Membership::memberName

```
null
```

- ParameterMembership::ownedRelatedElement

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

- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- Element::ownedRelationship  
  
Set{ }

#### C.2.3.3.2.12 CommonReturnParameterReferenceUsage\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

CommonReturnParameterReferenceUsageUntyped\_Mapping

##### Mapping Source

Element

##### Mapping Target

ReferenceUsage

##### Owned Mappings

- commonReturnParameterReferenceUsageFeatureTyping :  
CommonReturnParameterReferenceUsageFeatureTyping\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Feature::isEnd  
  
false
- Type::isSufficient  
  
false
- Feature::isUnique



## General Mappings

GenericToFeatureTyping\_Mapping

## Mapping Source

Element

## Mapping Target

FeatureTyping

## Owned Mappings

- commonParameterReferenceUsageIn : CommonParameterReferenceUsageIn\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Element::aliasId  

```
Set{}
```
- Relationship::ownedRelatedElement  

```
Set{}
```
- FeatureTyping::typedFeature  

```
commonParameterReferenceUsageIn.to
```
- Specialization::specific  
*abstract rule*
- Element::name  

```
null
```
- Element::shortName  

```
null
```
- FeatureTyping::type  

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

- Specialization::general  
*abstract rule*
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set {}`

#### C.2.3.3.2.14 CommonReturnParameterReferenceUsageFeatureTyping\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureTyping\_Mapping

##### Mapping Source

Element

##### Mapping Target

FeatureTyping

##### Owned Mappings

- commonReturnParameterReferenceUsage : CommonReturnParameterReferenceUsage\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId  
`Set {}`
- Relationship::ownedRelatedElement  
`Set {}`
- Specialization::specific  
*abstract rule*
- FeatureTyping::typedFeature  
`commonReturnParameterReferenceUsage.to`
- FeatureTyping::type



```

    if from.ocIsKindOf(UML::TypedElement)
    then
    if from.ocAsType(UML::TypedElement).type.ocIsKindOf(UML::PrimitiveType) then
        Helper.getScalarValueType(from.ocAsType(UML::TypedElement).type)
    else
        from.ocAsType(UML::TypedElement).type
    endif
    else OclUndefined endif

```

- Element::name

```

    null

```

- Element::shortName

```

    null

```

- Specialization::general

*abstract rule*

- Element::elementId

```

    Helper.createUUID()

```

- Element::ownedRelationship

```

    Set{}

```

#### C.2.3.3.2.15 CommonReturnParameterReferenceUsageUntyped\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### 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:

(none)

##### Mapping rules

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

- ReferenceUsage::direction  
`KerML::FeatureDirectionKind::_out'`
- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Usage::isVariation  
`false`
- Feature::isReadOnly  
`false`
- Element::name  
`null`
- Feature::isDerived  
`false`
- Feature::isComposite

false

- Element::ownedRelationship

Set{}

#### **C.2.3.3.2.16 EmptyReturnParameterFeatureMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToReturnParameterMembership\_Mapping

##### **Mapping Source**

Element

##### **Mapping Target**

ReturnParameterMembership

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName

null

- Element::shortName

null

- Element::elementId

Helper.createUUID()

- Element::aliasId

Set{}

- FeatureMembership::owningType  
*abstract rule*

- ReturnParameterMembership::ownedMemberParameter  
`CommonReturnParameterFeatureUntyped_Mapping.getMapped(from)`
- Membership::memberName  
`null`
- ParameterMembership::ownedRelatedElement  
`Set{self.ownedMemberParameter() }`
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
`KerML::VisibilityKind::public`
- Element::name  
`null`
- Element::ownedRelationship  
`Set{ }`

### C.2.3.4 Generic Mappings To Systems

#### C.2.3.4.1 Overview

**Table 4. List of all Overview Mapping Specifications**

Mapping Class	SysML v2 Concept
GenericToActionUsage_Mapping	ActionUsage
GenericToAssignmentActionUsage_Mapping	AssignmentActionUsage
GenericToConjugatedPortDefinition_Mapping	ConjugatedPortDefinition
GenericToConjugatedPortTyping_Mapping	ConjugatedPortTyping
GenericToConnectionUsage_Mapping	ConnectionUsage
GenericToConstraintDefinition_Mapping	ConstraintDefinition
GenericToDefinition_Mapping	Definition
GenericToEventOccurrenceUsage_Mapping	EventOccurrenceUsage
GenericToItemDefinition_Mapping	ItemDefinition
GenericToMetadataUsage_Mapping	MetadataUsage
GenericToOccurrenceDefinition_Mapping	OccurrenceDefinition
GenericToOccurrenceUsage_Mapping	OccurrenceUsage
GenericToPartUsage_Mapping	PartUsage
GenericToPortConjugation_Mapping	PortConjugation

Mapping Class	SysML v2 Concept
GenericToPortDefinition_Mapping	PortDefinition
GenericToReferenceUsage_Mapping	ReferenceUsage
GenericToStateUsage_Mapping	StateUsage
GenericToUsage_Mapping	Usage

## C.2.3.4.2 Mapping Specifications

### C.2.3.4.2.1 GenericToActionUsage\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

GenericToUsage\_Mapping  
GenericToStep\_Mapping

#### Mapping Source

#### Mapping Target

ActionUsage

#### Owned Mappings

(none)

#### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

#### Mapping rules

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

- Feature::isEnd  
false
- Type::isSufficient  
false
- Feature::isUnique  
true
- Element::shortName  
null

- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- ActionUsage::isComposite  
true
- Element::ownedRelationship  
Set{}

#### **C.2.3.4.2.2 GenericToAssignmentActionUsage\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToActionUsage\_Mapping

##### **Mapping Source**

##### **Mapping Target**

AssignmentActionUsage

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Usage::isVariation  
`false`
- Feature::isReadOnly  
`false`
- Feature::direction

null

- Element::name

null

- Feature::isDerived

false

- Feature::isComposite

false

- Element::ownedRelationship

Set{}

#### **C.2.3.4.2.3 GenericToConnectionUsage\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToPartUsage\_Mapping

##### **Mapping Source**

##### **Mapping Target**

ConnectionUsage

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Feature::isEnd

false

- Type::isSufficient

false

- Feature::isUnique

true



- Element::shortName  
null
- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Usage::isVariation  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false
- Element::ownedRelationship  
Set{}

#### **C.2.3.4.2.4 GenericToConjugatedPortDefinition\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToPortDefinition\_Mapping

### Mapping Source

### Mapping Target

ConjugatedPortDefinition

### Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  
`Set{}`
- Type::isSufficient  
`false`
- Definition::isVariation  
`false`
- Element::name  
`null`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set{}`

#### C.2.3.4.2.5 GenericToConjugatedPortTyping\_Mapping

### Description

\*\*\* not specified yet \*\*\*

## General Mappings

GenericToFeatureTyping\_Mapping

## Mapping Source

## Mapping Target

ConjugatedPortTyping

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Element::aliasId  
`Set {}`
- Relationship::ownedRelatedElement  
`Set {}`
- Specialization::specific  
*abstract rule*
- ConjugatedPortTyping::conjugatedPortDefinition  
*abstract rule*
- ConjugatedPortTyping::portDefinition  
*abstract rule*
- Element::name  
`null`
- Element::shortName  
`null`
- Specialization::general  
*abstract rule*
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set {}`

#### C.2.3.4.2.6 GenericToConstraintDefinition\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToDefinition\_Mapping

##### Mapping Source

##### Mapping Target

ConstraintDefinition

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId  
`Set {}`
- Type::isSufficient  
`false`
- Element::name  
`null`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set {}`

#### C.2.3.4.2.7 GenericToDefinition\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToClassifier\_Mapping

##### Mapping Source

##### Mapping Target

Definition

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId  
`Set {}`
- Type::isSufficient  
`false`
- Definition::isVariation  
`false`
- Element::name  
`null`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship

Set{ }

#### C.2.3.4.2.8 GenericToEventOccurrenceUsage\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToOccurrenceUsage\_Mapping

##### Mapping Source

##### Mapping Target

EventOccurrenceUsage

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Feature::isEnd  
false
- Type::isSufficient  
false
- Feature::isUnique  
true
- Element::shortName  
null
- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Feature::isOrdered  
false

- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Usage::isVariation  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false
- Element::ownedRelationship  
Set{}

#### **C.2.3.4.2.9 GenericToItemDefinition\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToDefinition\_Mapping

##### **Mapping Source**

##### **Mapping Target**

ItemDefinition

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  
`Set{}`
- Type::isSufficient  
`false`
- Element::name  
`null`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set{}`

#### C.2.3.4.2.10 GenericToMetadataUsage\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToUsage\_Mapping

##### Mapping Source

##### Mapping Target

MetadataUsage

##### Owned Mappings

(none)

##### Applicable filters



This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Feature::isReadOnly  
`false`
- Feature::direction  
`null`
- Element::name  
`null`
- Feature::isDerived  
`false`

- Feature::isComposite  
false
- Element::ownedRelationship  
Set{ }

#### **C.2.3.4.2.11 GenericToOccurrenceDefinition\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToDefinition\_Mapping

##### **Mapping Source**

##### **Mapping Target**

OccurrenceDefinition

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Element::aliasId  
Set{ }
- Type::isSufficient  
false
- OccurrenceDefinition::isIndividual  
false
- Element::name  
null
- Element::shortName  
null
- Type::isAbstract

false

- Element::elementId

Helper.createUUID()

- Element::ownedRelationship

Set{}

#### **C.2.3.4.2.12 GenericToOccurrenceUsage\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToUsage\_Mapping

##### **Mapping Source**

##### **Mapping Target**

OccurrenceUsage

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Feature::isEnd

false

- Type::isSufficient

false

- OccurrenceUsage::portionKind

OclUndefined

- Feature::isUnique

true

- Element::shortName

null

- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false
- OccurrenceUsage::isIndividual  
false
- Element::ownedRelationship  
Set{}

#### **C.2.3.4.2.13 GenericToPartUsage\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToUsage\_Mapping

##### **Mapping Source**

## Mapping Target

PartUsage

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Feature::isReadOnly  
`false`
- Feature::direction

null

- Element::name

null

- Feature::isDerived

false

- Feature::isComposite

false

- Element::ownedRelationship

Set{}

#### **C.2.3.4.2.14 GenericToPortConjugation\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToConjugation\_Mapping

##### **Mapping Source**

##### **Mapping Target**

PortConjugation

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Element::aliasId

Set{}

- Relationship::ownedRelatedElement

Set{}

- Relationship::source

Set{}

- Element::name  
null
- Element::shortName  
null
- Element::elementId  
Helper.createUUID()
- Relationship::target  
Set{}
- PortConjugation::originalPortDefinition  
*abstract rule*
- Element::ownedRelationship  
Set{}

#### C.2.3.4.2.15 GenericToPortDefinition\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToDefinition\_Mapping

##### Mapping Source

##### Mapping Target

PortDefinition

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId  
Set{}
- Type::isSufficient  
false

- Element::name  
null
- Element::shortName  
null
- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Element::ownedRelationship  
Set{}

#### **C.2.3.4.2.16 GenericToReferenceUsage\_Mapping**

##### **Description**

Provides the basic features to map to a ReferenceUsage element.

##### **General Mappings**

GenericToUsage\_Mapping

##### **Mapping Source**

##### **Mapping Target**

ReferenceUsage

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Feature::isEnd  
false
- Type::isSufficient  
false
- Feature::isUnique



- `Element::shortName`

- `Type::isAbstract`

- `Element::elementId`

- Feature::isOrdered

- `Element::aliasId`

- Feature::isPortion

- Feature::isReadOnly

- Feature::direction

- `Element::name`

- Feature::isDerived

- Feature::isComposite

- Element::ownedRelationship

#### C.2.3.4.2.17 GenericToStateUsage\_Mapping

\*\*\* not specified yet \*\*\*

## GenericToActionUsage\_Mapping

### Mapping Source

### Mapping Target

StateUsage

### Owned Mappings

(none)

### Applicable filters

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

(none)

### Mapping rules

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

- Feature::isEnd  
false
- Type::isSufficient  
false
- Feature::isUnique  
true
- Element::shortName  
null
- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Usage::isVariation  
false

- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false
- Element::ownedRelationship  
Set{ }

#### **C.2.3.4.2.18 GenericToUsage\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeature\_Mapping

##### **Mapping Source**

##### **Mapping Target**

Usage

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Element::aliasId  
Set{ }
- Usage::isVariation

- false
- Type::isSufficient
   
false
- Element::name
   
null
- Element::shortName
   
null
- Type::isAbstract
   
false
- Element::elementId
   
Helper.createUUID()
- Element::ownedRelationship
   
Set{}

## C.2.4 SysML v1.7

### C.2.4.1 Overview

### C.2.4.2 Activities

#### C.2.4.2.1 Overview

Table 5. List of all Overview Mapping Specifications

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

#### C.2.4.2.2 Mapping Specifications

#### C.2.4.3 Allocations

### C.2.4.3.1 Overview

Table 6. List of all Overview Mapping Specifications

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

### C.2.4.3.2 Mapping Specifications

#### C.2.4.3.2.1 AllocationDefinition\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

Abstraction\_Mapping

##### Mapping Source

Dependency

##### Mapping Target

AllocationDefinition

##### Owned Mappings

- allocationDefinitionFromFeatureMembership : AllocationDefinitionFromFeatureMembership\_Mapping
- allocationDefinitionToFeatureMembership : AllocationDefinitionToFeatureMembership\_Mapping

##### Applicable filters

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

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

##### Mapping rules

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

- Element::aliasId  

```
Set{}
```
- Dependency::supplier  

```
from.target->collect(e | ElementMain_Mapping.getMapped(e))
```
- Dependency::name  

```
from.name
```
- AllocationDefinition::ownedRelationship

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

- Relationship::owningRelatedElement

```
ElementMain_Mapping.getMapped(from.owner)
```

- Dependency::client

```
from.source->collect(e | ElementMain_Mapping.getMapped(e))
```

- Element::elementId

```
Helper.getID(from)
```

- Relationship::ownedRelatedElement

```
from.relatedElement->select(e | from.ownedElement->includes(e))->collect(e | ElementMain_Mapping.getMapped(e))
```

- Element::shortName

```
null
```

#### C.2.4.3.2.2 AllocationDefinitionToFeatureMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureMembership\_Mapping

##### Mapping Source

Dependency

##### Mapping Target

FeatureMembership

##### Owned Mappings

- allocationDefinitionToReferenceUsage : AllocationDefinitionToReferenceUsage\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName

- FeatureMembership::memberName

```
'allocatedTo'
```

- `Element::shortName`

null

- `Element::elementId`

```
Helper.createUUID()
```

- `Element::aliasId`

Set { }

- `OwningMembership::ownedMemberElement`

*abstract rule*

- `OwningMembership::ownedRelatedElement`

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

- `TypeFeaturing::featureOfType`

*abstract rule*

- `TypeFeaturing::featuringType`

*abstract rule*

- Membership::visibility

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

- `Element::name`

null

- FeatureMembership::ownedMemberFeature

allocationDefinitionToReferenceUsage.to

- `Element::ownedRelationship`

Set { }

#### C.2.4.3.2.3 AllocationDefinitionFromFeatureMembership\_Mapping

### Description

\*\*\* not specified yet \*\*\*

## General Mappings

GenericToFeatureMembership\_Mapping

## Mapping Source

## Dependency

## Mapping Target

FeatureMembership

## Owned Mappings

- allocationDefinitionFromReferenceUsage : AllocationDefinitionFromReferenceUsage\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName

null

- FeatureMembership::memberName

'allocatedFrom'

- Element::shortName

null

- Element::elementId

Helper.createUUID()

- Element::aliasId

Set{}

- OwningMembership::ownedMemberElement  
*abstract rule*
- OwningMembership::ownedRelatedElement

Set{self.ownedMemberElement() }

- TypeFeaturing::featureOfType  
*abstract rule*

- TypeFeaturing::featuringType  
*abstract rule*

- Membership::visibility

KerML::VisibilityKind::public

- Element::name

null



- FeatureMembership::ownedMemberFeature  
allocationDefinitionFromReferenceUsage.to
- Element::ownedRelationship  
Set{ }

#### C.2.4.3.2.4 AllocationDefinitionFromFeatureTyping\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureTyping\_Mapping

##### Mapping Source

Dependency

##### Mapping Target

FeatureTyping

##### Owned Mappings

- allocationDefinitionFromReferenceUsage : AllocationDefinitionFromReferenceUsage\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId  
Set{ }
- Relationship::ownedRelatedElement  
Set{ }
- FeatureTyping::typedFeature  
allocationDefinitionFromReferenceUsage.to
- Specialization::specific  
*abstract rule*
- FeatureTyping::type  
from.source.get(0)

- Element::name  
null
- Element::shortName  
null
- Specialization::general  
*abstract rule*
- Element::elementId  
Helper.createUUID()
- Element::ownedRelationship  
Set{ }

#### **C.2.4.3.2.5 AllocationDefinitionFromReferenceUsage\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToReferenceUsage\_Mapping

##### **Mapping Source**

Dependency

##### **Mapping Target**

ReferenceUsage

##### **Owned Mappings**

- allocationDefinitionFromFeatureTyping : AllocationDefinitionFromFeatureTyping\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- ReferenceUsage::isEnd  
true
- Type::isSufficient  
false

- ReferenceUsage::ownedRelationship  
Set{allocationDefinitionFromFeatureTyping,to}
- Feature::isUnique  
true
- Element::shortName  
null
- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Usage::isVariation  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false

#### **C.2.4.3.2.6 AllocationDefinitionToFeatureTyping\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

## General Mappings

GenericToFeatureTyping\_Mapping

## Mapping Source

Dependency

## Mapping Target

FeatureTyping

## Owned Mappings

- allocationDefinitionToReferenceUsage : AllocationDefinitionToReferenceUsage\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Element::aliasId  
`Set{}`
- Relationship::ownedRelatedElement  
`Set{}`
- Specialization::specific  
*abstract rule*
- FeatureTyping::typedFeature  
`allocationDefinitionToReferenceUsage.to`
- Element::name  
`null`
- Element::shortName  
`null`
- Specialization::general  
*abstract rule*
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship

Set{}

- FeatureTyping::type

from.target.get(0)

#### **C.2.4.3.2.7 AllocationDefinitionToReferenceUsage\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToReferenceUsage\_Mapping

##### **Mapping Source**

Dependency

##### **Mapping Target**

ReferenceUsage

##### **Owned Mappings**

- allocationDefinitionToFeatureTyping : AllocationDefinitionToFeatureTyping\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Type::isSufficient

false

- Feature::isUnique

true

- Element::shortName

null

- Type::isAbstract

false

- ReferenceUsage::isEnd

true

- Element::elementId

```

    Helper.createUUID()

    • Feature::isOrdered
      false

    • Element::aliasId
      Set{}

    • Feature::isPortion
      false

    • Usage::isVariation
      false

    • Feature::isReadOnly
      false

    • Feature::direction
      null

    • Element::name
      null

    • Feature::isDerived
      false

    • Feature::isComposite
      false

    • ReferenceUsage::ownedRelationship
      Set{allocationDefinitionToFeatureTyping.to}

```

#### **C.2.4.3.2.8 AllocationUsage\_Mapping**

##### **Description**

A SysML::Allocate relationship is mapped to a SysMLv2::AllocationUsage.

##### **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(from, 'SysML::Allocations::Allocate') and not from.client->select(t | t
```

## Mapping rules

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

- Feature::isEnd  
`false`
- Dependency::name  
`from.name`
- Element::ownedRelationship  
`ElementOwnership_Mapping.getMappedColl(from.ownedElement)`
- Type::isSufficient  
`false`
- Relationship::owningRelatedElement  
`ElementMain_Mapping.getMapped(from.owner)`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`

- `Dependency::supplier`

```
from.target->collect(e | ElementMain_Mapping.getMapped(e))
```

- `Feature::isReadOnly`

```
false
```

- `Dependency::client`

```
from.source->collect(e | ElementMain_Mapping.getMapped(e))
```

- `Feature::direction`

```
null
```

- `Element::elementId`

```
Helper.getID(from)
```

- `Feature::isDerived`

```
false
```

- `Relationship::ownedRelatedElement`

```
from.relatedElement->select(e | from.ownedElement->includes(e))->collect(e | ElementMain_Mapping.getMapped(e))
```

- `Feature::isComposite`

```
false
```

#### C.2.4.4 Blocks

##### C.2.4.4.1 Overview

**Table 7. List of all Overview Mapping Specifications**

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



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

#### C.2.4.4.2 SysML v1 Blocks elements not mapped

Table 8. List of SysML v1 elements not mapped of this section

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

#### C.2.4.4.3 Mapping Specifications

##### C.2.4.4.3.1 AssociationBlock\_Mapping

###### Description

\*\*\* not specified yet \*\*\*

###### General Mappings

AssociationClass\_Mapping

###### Mapping Source

AssociationClass

###### Mapping Target

ConnectionDefinition

###### Owned Mappings

(none)

###### Applicable filters

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

```
from.memberEnd->select( m | m.type.ocIsKindOf(UML::UseCase)) ->isEmpty()
,
Helper.hasStereotypeApplied(from, 'SysML::Blocks::Block')
```

###### Mapping rules

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

- **Type::isSufficient**

false

- **Relationship::owningRelatedElement**

ElementMain\_Mapping.getMapped(from.owner)

- **Association::ownedRelationship**

```
let nonOwnedEnds: OrderedSet(UML::Property) = (from.memberEnd-from.ownedEnd)->asOrderedSet()
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.ocIsKindOf
let others: OrderedSet(UML::Element) = ((from.ownedElement-from.memberEnd)-generalizations)->
nonOwnedEnds->collect(e | NonOwnedEndMembership_Mapping.getMapped(e))
->union(from.ownedEnd->collect(e | OwnedEndMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))
->union(others->collect(e | ElementOwningMembership_Mapping.getMapped(e)))
->asOrderedSet()
```

- **Element::name**

from.name

- **Element::shortName**

null

- **Relationship::target**

Set{}

- **Element::aliasId**

Set{}

- **Classifier::isAbstract**

from.isAbstract

- **Namespace::ownedImport**

Set{}

- **Relationship::source**

Set{}

- **Element::elementId**

Helper.getID(from)

- **Relationship::ownedRelatedElement**

```
from.relatedElement->select(e | from.ownedElement->includes(e))->collect(e | ElementMain_Map
```

#### C.2.4.4.3.2 BindingConnector\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

Connector\_Mapping

##### Mapping Source

Connector

##### Mapping Target

BindingConnectorAsUsage

##### Owned Mappings

(none)

##### Applicable filters

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

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

##### Mapping rules

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

- Relationship::ownedRelatedElement  
`Set{}`
- Feature::isEnd  
`false`
- Element::ownedRelationship  
`ElementOwnership_Mapping.getMappedColl(from.ownedElement)`
- Type::isSufficient  
`false`
- Connector::isDirected  
`false`
- Feature::isUnique  
`true`
- Element::name

- from.name
- Element::shortName
  - null
- Type::isAbstract
  - false
- Feature::isOrdered
  - false
- Relationship::target
  - Set{}
- Element::aliasId
  - Set{}
- Feature::isPortion
  - false
- Feature::isReadOnly
  - false
- Relationship::source
  - Set{}
- Feature::direction
  - null
- Element::elementId
  - Helper.getID(from)
- Feature::isDerived
  - false
- Feature::isComposite
  - false

#### **C.2.4.4.3.3 Block\_Mapping**

##### **Description**

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

##### **General Mappings**

Class\_Mapping

## Mapping Source

Class

## Mapping Target

PartDefinition

## Owned Mappings

(none)

## Applicable filters

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

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

## Mapping rules

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

- Element::aliasId  

```
Set{}
```
- Classifier::isAbstract  

```
from.isAbstract
```
- Type::isSufficient  

```
false
```
- Namespace::ownedImport  

```
Set{}
```
- Classifier::ownedRelationship

```
let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | (e.oclIsKindOf(UML::Property)))
let redefinedAttributes: Set(UML::Element) = from.ownedElement->select(e | from.oclIsKindOf(UML::Class))
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.oclIsKindOf(UML::Generalization))
let constraints : Set(UML::Constraint) = UML::Constraint.allInstances()->select(c | c.constraintBlock = from)
let toElementOMS: Set(UML::Element) = (((from.ownedElement - toElementFMS) - redefinedAttributes) - constraints)
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(ClassifierBehavior_Mapping.getMapped(from.classifierBehavior))
```

- Element::elementId

```
Helper.getID(from)
```

- Element::name

```
from.name
```

- Element::shortName

```
null
```

#### **C.2.4.4.3.4 Part\_Mapping**

##### **Description**

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

##### **General Mappings**

Property\_Mapping

##### **Mapping Source**

Property

##### **Mapping Target**

PartUsage

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

```
let p: UML::Property = from.oclAsType(UML::Property) in
if p.type.oclIsUndefined() then false else Helper.hasStereotypeApplied(p.type, 'SysML::Blocks::
and (p.association.oclIsUndefined() or p.association.ownedEnd->excludes(p)) and p.aggregation =
```

##### **Mapping rules**

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

- Feature::isOrdered

```
from.isOrdered
```

- Type::isSufficient

```
false
```

- Feature::isComposite

```
from.isComposite
```

- Feature::ownedRelationship

```

let typing: KerML::FeatureTyping = StructuralFeatureToFeatureTyping_Mapping.getMapped(from)
let subsetting: Set(KerML::Subsetting) = from.subsettedProperty->collect(p | PropertySubsetting)
let subsettingMultiplicityTyping: Set(KerML::Relationship) = subsetting->union(if typing.ocIsTypeOf(UML::MultiplicityMembership_Mapping.getMapped(from))
    Set{MultiplicityMembership_Mapping.getMapped(from)}
else
    Set{MultiplicityMembership_Mapping.getMapped(from), typing}
endif)->asSet() in

let relationships: Set(KerML::Relationship) = if from.defaultValue.ocIsTypeOf(UML::OpaqueExpression) then
    if from.defaultValue.ocIsUndefined() then
        relationships
    else
        relationships->including(if from.defaultValue.ocIsTypeOf(UML::OpaqueExpression) then
            from.defaultValue
        else
            from.defaultValue
        endif)
    endif
endif

```

- **Feature::isAbstract**

```
false
```

- **Feature::isEnd**

```

if from.association.ocIsUndefined() then
    false
else
    from.association.ownedEnd->includes(from)
endif

```

- **Element::name**

```
from.name
```

- **Element::shortName**

```
null
```

- **Element::aliasId**

```
Set{}
```

- **Feature::isPortion**

```
false
```

- **Feature::isDerived**

```
from.isDerived
```

- **Feature::direction**

```
null
```

- **Element::elementId**

```
Helper.getID(from)
```

- **Feature::isUnique**

```
from.isUnique
```

- Feature::isReadOnly  
*abstract rule*

#### C.2.4.4.3.5 EncapsulatedBlock\_Mapping

##### Description

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

##### General Mappings

Block\_Mapping

##### Mapping Source

Class

##### Mapping Target

PartDefinition

##### Owned Mappings

(none)

##### Applicable filters

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

```
not Helper.hasStereotypeApplied(from, 'SysML::Requirements::Requirement') and not from.ocIsTypeOf(
,
not from.ocIsTypeOf(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

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

- Element::aliasId  
Set{}
- Classifier::isAbstract  
from.isAbstract
- Type::isSufficient  
false
- Namespace::ownedImport



```
Set{}
```

- **PartDefinition::ownedRelationship**

```
let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::Property))
let redefinedAttributes: Set(UML::Element) = from.ownedElement->select(e | from.ocIsKindOf(UML::Property) and e.ocIsKindOf(UML::Property))
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.ocIsKindOf(UML::Generalization))
let toElementOMS: Set(UML::Element) = (((from.ownedElement - toElementFMS) - redefinedAttributes) - generalizations)
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.ocIsUndefined() then relationships else relationships->append(ClassifierBehavior)
```

- **Element::elementId**

```
Helper.getID(from)
```

- **Element::name**

```
from.name
```

- **Element::shortName**

```
null
```

#### **C.2.4.4.3.6 EncapsulatedBlockMetadataMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToOwningMembership\_Mapping

##### **Mapping Source**

Class

##### **Mapping Target**

OwningMembership

##### **Owned Mappings**

- encapsulatedBlockMetadata : EncapsulatedBlockMetadata\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Relationship::ownedRelatedElement  
`Set{}`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Membership::memberElement  
*abstract rule*
- OwningMembership::ownedMemberElement  
`encapsulatedBlockMetadata.to`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{}`
- Membership::memberName  
`null`
- Membership::visibility  
`KerML::VisibilityKind::public`
- Element::name  
`null`
- Element::ownedRelationship  
`Set{}`

#### **C.2.4.4.3.7 EncapsulatedBlockMetadata\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToMetadataUsage\_Mapping

##### **Mapping Source**

Class

## Mapping Target

MetadataUsage

## Owned Mappings

- encapsulatedBlockMetadataFeatureMembership : EncapsulatedBlockMetadataFeatureMembership\_Mapping
- encapsulatedBlockMetadataFeatureTyping : EncapsulatedBlockMetadataFeatureTyping\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- MetadataUsage::ownedRelationship  
`Set{encapsulatedBlockMetadataFeatureMembership.to, encapsulatedBlockMetadataFeatureTyping.to}`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion

false

- Usage::isVariation

false

- Feature::isReadOnly

false

- Feature::direction

null

- Element::name

null

- Feature::isDerived

false

- Feature::isComposite

false

#### **C.2.4.4.3.8 EncapsulatedBlockMetadataFeatureMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureMembership\_Mapping

##### **Mapping Source**

Class

##### **Mapping Target**

FeatureMembership

##### **Owned Mappings**

- encapsulatedBlockMetadataReferenceUsage : EncapsulatedBlockMetadataReferenceUsage\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- FeatureMembership::ownedMemberFeature

- encapsulatedBlockMetadataReferenceUsage.to
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
  
null
- OwningMembership::ownedRelatedElement  
  
Set{self.ownedMemberElement() }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- Element::ownedRelationship  
  
Set{ }

#### **C.2.4.4.3.9 EncapsulatedBlockMetadataFeatureTyping\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureTyping\_Mapping

##### **Mapping Source**

Class

## Mapping Target

FeatureTyping

## Owned Mappings

- encapsulatedBlockMetadata : EncapsulatedBlockMetadata\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Element::aliasId  
`Set{}`
- Relationship::ownedRelatedElement  
`Set{}`
- Specialization::specific  
*abstract rule*
- Element::name  
`null`
- FeatureTyping::type  
`SysML2::MetadataDefinition.allInstances()->any(m | m.qualifiedName = 'SysMLv1Library::BlockD`
- Element::shortName  
`null`
- Specialization::general  
*abstract rule*
- Element::elementId  
`Helper.createUUID()`
- FeatureTyping::typedFeature  
`encapsulatedBlockMetadata.to`
- Element::ownedRelationship  
`Set{}`

### C.2.4.4.3.10 EncapsulatedBlockMetadataReferenceUsage\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

## General Mappings

GenericToReferenceUsage\_Mapping

## Mapping Source

Class

## Mapping Target

ReferenceUsage

## Owned Mappings

- encapsulatedBlockMetadataFeatureValue : EncapsulatedBlockMetadataFeatureValue\_Mapping
- encapsulatedBlockMetadataRedefinition : EncapsulatedBlockMetadataRedefinition\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- ReferenceUsage::ownedRelationship  
`Set{encapsulatedBlockMetadataRedefinition.to, encapsulatedBlockMetadataFeatureValue.to}`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered

- false
- Element::aliasId
  - Set{}
- Feature::isPortion
  - false
- Usage::isVariation
  - false
- Feature::isReadOnly
  - false
- Feature::direction
  - null
- Element::name
  - null
- Feature::isDerived
  - false
- Feature::isComposite
  - false

#### **C.2.4.4.3.11 EncapsulatedBlockMetadataFeatureValue\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureValue\_Mapping

##### **Mapping Source**

Class

##### **Mapping Target**

FeatureValue

##### **Owned Mappings**

- literalBooleanTrue : LiteralBooleanTrue\_Mapping

##### **Applicable filters**



This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName

    null

- Element::shortName

    null

- Element::elementId

    Helper.createUUID()

- Element::aliasId

    Set{}

- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName

    null

- OwningMembership::ownedRelatedElement

    Set{self.ownedMemberElement() }

- Membership::visibility

    KerML::VisibilityKind::public

- Element::name

    null

- Element::ownedRelationship

    Set{}

- FeatureValue::value

    literalBooleanTrue.to

#### C.2.4.4.3.12 EncapsulatedBlockMetadataRedefinition\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToRedefinition\_Mapping

## Mapping Source

Class

## Mapping Target

Redefinition

## Owned Mappings

- encapsulatedBlockMetadataReferenceUsage : EncapsulatedBlockMetadataReferenceUsage\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Element::aliasId

Set{ }

- Subsetting::ownedRelatedElement

Set{ }

- Subsetting::subsettingFeature

*abstract rule*

- Element::name

null

- Subsetting::subsettingFeature

*abstract rule*

- Redefinition::redefiningFeature

encapsulatedBlockMetadataReferenceUsage.to

- Element::shortName

null

- Redefinition::redefinedFeature

SysML2::AttributeUsage.allInstances()->any(m | m.qualifiedName = 'SysMLv1Library::BlockData:')

- Element::elementId

Helper.createUUID()

- Element::ownedRelationship

Set { }

## C.2.4.5 Libraries

### C.2.4.5.1 Requirements

#### C.2.4.5.1.1 VerdictKind

##### Description

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

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

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

##### Literals

- error
- fail
- inconclusive
- pass

### C.2.4.5.2 UnitAndQuantityKind

## C.2.4.6 Model Elements

### C.2.4.6.1 Overview

Table 9. List of all Overview Mapping Specifications

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

### C.2.4.6.2 Mapping Specifications

#### C.2.4.6.2.1 ProblemRationaleMetadataUsage\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

## General Mappings

GenericToMetadataUsage\_Mapping

## Mapping Source

Comment

## Mapping Target

MetadataUsage

## Owned Mappings

- problemRationaleMetadataFeatureTyping : ProblemRationaleMetadataFeatureTyping\_Mapping
- unnamed1 : Boolean

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`

- Feature::isPortion  
false
- Usage::isVariation  
false
- Feature::isReadOnly  
false
- MetadataUsage::ownedRelationship  
Set{problemRationaleMetadataFeatureTyping.to, ProblemRationaleMetadataFeatureMembership\_Mapping}
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false

#### **C.2.4.6.2.2 CommentToConcern\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

Comment\_Mapping

##### **Mapping Source**

Comment

##### **Mapping Target**

ConcernDefinition

##### **Owned Mappings**

- commentToConcernReturnParameterMembership :  
CommentToConcernReturnParameterMembership\_Mapping

##### **Applicable filters**

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

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

## Mapping rules

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

- Element::aliasId

```
Set{}
```

- Element::ownedRelationship

```
ElementOwnership_Mapping.getMappedColl(from.ownedElement)
```

- ConcernDefinition::ownedRelationship

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

- Element::elementId

```
Helper.getID(from)
```

- Element::name

```
null
```

- Element::shortName

```
null
```

- AnnotatingElement::annotation

```
Set{}
```

### C.2.4.6.2.3 CommentToConcernComment\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

GenericToAnnotatingElement\_Mapping

#### Mapping Source

Comment

#### Mapping Target

Comment

#### Owned Mappings

(none)

### Applicable filters

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

(none)

### Mapping rules

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

- Element::aliasId

`Set{}`

- Comment::body

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

- Element::name

`null`

- Element::shortName

`null`

- Element::elementId

`Helper.createUUID()`

- Element::ownedRelationship

`Set{}`

#### C.2.4.6.2.4 CommentToConcernDocumentation\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToAnnotation\_Mapping

##### Mapping Source

Comment

##### Mapping Target

Annotation

##### Owned Mappings

- commentToConcernComment : CommentToConcernComment\_Mapping

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  
`Set{}`
- Relationship::source  
`Set{}`
- Element::name  
`null`
- Annotation::ownedRelatedElement  
`Set{commentToConcernComment.to}`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Relationship::target  
`Set{}`
- Element::ownedRelationship  
`Set{}`

#### C.2.4.6.2.5 CommentToConcernReturnParameter\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToReferenceUsage\_Mapping

##### Mapping Source

Comment

##### Mapping Target

ReferenceUsage



## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Usage::isVariation  
`false`
- Feature::isReadOnly  
`false`
- Feature::direction

null

- Element::name

    null

- Feature::isDerived

    false

- Feature::isComposite

    false

- Element::ownedRelationship

    Set{ }

#### C.2.4.6.2.6 CommentToConcernReturnParameterMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToParameterMembership\_Mapping

##### Mapping Source

Comment

##### Mapping Target

ReturnParameterMembership

##### Owned Mappings

- commentToConcernDocumentation : CommentToConcernDocumentation\_Mapping
- commentToConcernReturnParameter : CommentToConcernReturnParameter\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- FeatureMembership::ownedRelatedElement  
    Set{self.ownedMemberFeature() }
- Membership::membershipOwningNamespace  
    *abstract rule*
- Membership::memberShortName

- `Element::shortName`

- `Element::elementId`

- `Element::aliasId`

- FeatureMembership::ownedMemberFeature

- FeatureMembership::owningType

- Membership::memberName

- `TypeFeaturing::featureOfType`

- `TypeFeaturing::featuringType`

- `ReturnParameterMembership::ownedRelatedElement`

- Membership::visibility

- `Element::name`

- `ReturnParameterMembership::ownedMemberParameter`

- Element::ownedRelationship

135

GenericToFeatureMembership\_Mapping

### Mapping Source

Comment

### Mapping Target

FeatureMembership

### Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
    null
- Element::shortName  
  
    null
- Element::elementId  
  
    Helper.createUUID()
- Element::aliasId  
  
    Set{}
- FeatureMembership::ownedMemberFeature  
  
    ProblemRationaleMetadataReferenceUsage\_Mapping.getMapped(from)
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
  
    null
- OwningMembership::ownedRelatedElement  
  
    Set{self.ownedMemberElement() }
- TypeFeaturing::featureOfType  
*abstract rule*

- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- Element::ownedRelationship  
  
Set{ }

#### C.2.4.6.2.8 ProblemRationaleMetadataFeatureTyping\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureTyping\_Mapping

##### Mapping Source

Comment

##### Mapping Target

FeatureTyping

##### Owned Mappings

- problemRationaleMetadataUsage : ProblemRationaleMetadataUsage\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId  
  
Set{ }
- Relationship::ownedRelatedElement  
  
Set{ }
- Specialization::specific  
*abstract rule*
- FeatureTyping::typedFeature

```
problemRationaleMetadataUsage.to
```

- FeatureTyping::type

```
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
```

- Element::name

```
null
```

- Element::shortName

```
null
```

- Specialization::general

```
abstract rule
```

- Element::elementId

```
Helper.createUUID()
```

- Element::ownedRelationship

```
Set{}
```

#### C.2.4.6.2.9 ProblemRationaleMetadataReferenceUsage\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToReferenceUsage\_Mapping

##### Mapping Source

Comment

##### Mapping Target

ReferenceUsage

##### Owned Mappings

- problemRationaleMetadataRedefinition : ProblemRationaleMetadataRedefinition\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Feature::isEnd  
false
- Type::isSufficient  
false
- ReferenceUsage::ownedRelationship  
Set{problemRationaleMetadataRedefinition.to, ProblemRationaleMetadataFeatureValue\_Mapping.ge
- Feature::isUnique  
true
- Element::shortName  
null
- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Usage::isVariation  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived

false

- Feature::isComposite

false

#### **C.2.4.6.2.10 ProblemRationaleMetadataFeatureValue\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureValue\_Mapping

##### **Mapping Source**

Comment

##### **Mapping Target**

FeatureValue

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName

null

- Element::shortName

null

- Element::elementId

Helper.createUUID()

- Element::aliasId

Set{}

- OwningMembership::ownedMemberElement  
*abstract rule*



- Membership::memberName  
null
- OwningMembership::ownedRelatedElement  
Set{self.ownedMemberElement() }
- Membership::visibility  
KerML::VisibilityKind::public
- Element::name  
null
- FeatureValue::value  
ProblemRationaleMetadataFeatureValueString\_Mapping.getMapped(from)
- Element::ownedRelationship  
Set{ }

#### **C.2.4.6.2.11 ProblemRationaleMetadataMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToOwningMembership\_Mapping

##### **Mapping Source**

Comment

##### **Mapping Target**

OwningMembership

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Relationship::ownedRelatedElement  
Set{ }

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- Membership::memberElement  
*abstract rule*
- OwningMembership::ownedMemberElement  
  
ProblemRationaleMetadataUsage\_Mapping.getMapped(from)
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- Membership::memberName  
  
null
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- Element::ownedRelationship  
  
Set{}

#### **C.2.4.6.2.12 ElementGroup\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToPackage\_Mapping

##### **Mapping Source**

Comment

##### **Mapping Target**

Package

## Owned Mappings

- elementGroupMetadaMembership : ElementGroupMetadaMembership\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Element::aliasId

```
Set{}
```

- Package::ownedRelationship

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

- Element::name

```
null
```

- Element::shortName

```
null
```

- Element::elementId

```
Helper.createUUID()
```

### C.2.4.6.2.13 ElementGroupCriterion\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

GenericToExpression\_Mapping

#### Mapping Source

Comment

#### Mapping Target

LiteralString

#### Owned Mappings

(none)

#### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- LiteralString::value  
`Helper.getTagValueAsString(from, 'SysML::ModelElements::ElementGroup', 'criterion')`
- Feature::isReadOnly  
`false`
- Feature::direction  
`null`
- Element::name  
`null`

- Feature::isDerived  
false
- Feature::isComposite  
false
- Element::ownedRelationship  
Set{ }

#### **C.2.4.6.2.14 ElementGroupMetadaMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToMembership\_Mapping

##### **Mapping Source**

Comment

##### **Mapping Target**

Membership

##### **Owned Mappings**

- elementGroupMetadataUsage : ElementGroupMetadataUsage\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Element::aliasId  
Set{ }
- Relationship::ownedRelatedElement  
Set{ }
- Relationship::source  
Set{ }
- Membership::ownedMemberElement  
elementGroupMetadataUsage.to

- Membership::memberName  
`'ElementGroupData'`
- Element::name  
`null`
- Membership::memberElement  
`self.ownedMemberElement()`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Relationship::target  
`Set{}`
- Element::ownedRelationship  
`Set{}`

#### **C.2.4.6.2.15 ElementGroupMetadataFeatureMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureMembership\_Mapping

##### **Mapping Source**

Comment

##### **Mapping Target**

FeatureMembership

##### **Owned Mappings**

- elementGroupMetadataReferenceUsage : ElementGroupMetadataReferenceUsage\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- FeatureMembership::ownedMemberFeature  
  
elementGroupMetadataReferenceUsage.to
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
  
null
- OwningMembership::ownedRelatedElement  
  
Set{self.ownedMemberElement() }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
  
KerML::VisibilityKind::public
- FeatureMembership::memberFeature  
  
self.ownedMemberFeature()
- Element::name  
  
null
- Element::ownedRelationship  
  
Set{}

#### **C.2.4.6.2.16 ElementGroupMetadataFeatureTyping\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

## General Mappings

GenericToFeatureTyping\_Mapping

## Mapping Source

Comment

## Mapping Target

FeatureTyping

## Owned Mappings

- elementGroupMetadataUsage : ElementGroupMetadataUsage\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Element::aliasId

```
Set{}
```

- Relationship::ownedRelatedElement

```
Set{}
```

- Specialization::specific

*abstract rule*

- Element::name

```
null
```

- Element::shortName

```
null
```

- FeatureTyping::type

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

- Specialization::general

*abstract rule*

- Element::elementId

```
Helper.createUUID()
```



- FeatureTyping::typedFeature  
     elementGroupMetadataUsage.to
- Element::ownedRelationship  
     Set{ }

#### **C.2.4.6.2.17 ElementGroupMetadataFeatureValue\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureValue\_Mapping

##### **Mapping Source**

Comment

##### **Mapping Target**

FeatureValue

##### **Owned Mappings**

- elementGroupCriterion : ElementGroupCriterion\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
 (none)

##### **Mapping rules**

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

- Membership::membershipOwningNamespace  
     *abstract rule*
- Membership::memberShortName  
     null
- Element::shortName  
     null
- Element::elementId  
     Helper.createUUID()
- Element::aliasId  
     Set{ }

- OwingMembership::ownedMemberElement  
*abstract rule*
- FeatureValue::value  
`elementGroupCriterion.to`
- Membership::memberName  
`null`
- OwingMembership::ownedRelatedElement  
`Set{self.ownedMemberElement() }`
- Membership::visibility  
`KerML::VisibilityKind::public`
- Element::name  
`null`
- Element::ownedRelationship  
`Set{ }`

#### **C.2.4.6.2.18 ElementGroupMetadataRedefinition\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToRedefinition\_Mapping

##### **Mapping Source**

Comment

##### **Mapping Target**

Redefinition

##### **Owned Mappings**

- elementGroupMetadataReferenceUsage : ElementGroupMetadataReferenceUsage\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Element::aliasId

```
Set{}
```

- Redefinition::redefinedFeature

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

- Subsetting::ownedRelatedElement

```
Set{}
```

- Subsetting::subsettingFeature

*abstract rule*

- Element::name

```
null
```

- Subsetting::subsettingFeature

*abstract rule*

- Redefinition::redefiningFeature

```
elementGroupMetadataReferenceUsage.to
```

- Element::shortName

```
null
```

- Element::elementId

```
Helper.createUUID()
```

- Element::ownedRelationship

```
Set{}
```

#### C.2.4.6.2.19 ElementGroupMetadataReferenceUsage\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToReferenceUsage\_Mapping

##### Mapping Source

Comment

##### Mapping Target

ReferenceUsage

### Owned Mappings

- elementGroupMetadataFeatureValue : ElementGroupMetadataFeatureValue\_Mapping
- elementGroupMetadataRedefinition : ElementGroupMetadataRedefinition\_Mapping

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Usage::isVariation  
`false`
- ReferenceUsage::ownedRelationship

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

- Feature::isReadOnly

```
false
```

- Feature::direction

```
null
```

- Element::name

```
null
```

- Feature::isDerived

```
false
```

- Feature::isComposite

```
false
```

#### **C.2.4.6.2.20 ElementGroupMetadataUsage\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToMetadataUsage\_Mapping

##### **Mapping Source**

Comment

##### **Mapping Target**

MetadataUsage

##### **Owned Mappings**

- elementGroupMetadataFeatureMembership : ElementGroupMetadataFeatureMembership\_Mapping
- elementGroupMetadataFeatureTyping : ElementGroupMetadataFeatureTyping\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Feature::isEnd

```
false
```

- `Type::isSufficient`  
`false`
- `Feature::isUnique`  
`true`
- `Element::shortName`  
`null`
- `Type::isAbstract`  
`false`
- `Element::elementId`  
`Helper.createUUID()`
- `Feature::isOrdered`  
`false`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `Usage::isVariation`  
`false`
- `Feature::isReadOnly`  
`false`
- `Feature::direction`  
`null`
- `Element::name`  
`null`
- `Feature::isDerived`  
`false`
- `MetadataUsage::ownedRelationship`  
`Set{elementGroupMetadataFeatureTyping.to, elementGroupMetadataFeatureMembership.to}`
- `Feature::isComposite`  
`false`

#### C.2.4.6.2.21 ElementGroupMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

ElementOwningMembership\_Mapping

##### Mapping Source

Element

##### Mapping Target

OwningMembership

##### Owned Mappings

- : Comment
- elementGroup : ElementGroup\_Mapping

##### Applicable filters

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

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

##### Mapping rules

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

- OwningMembership::ownedMemberElement

```
self.memberElement()
```
- Membership::memberShortName

```
null
```
- Element::shortName

```
null
```
- Element::elementId

```
Helper.createUUID()
```
- Membership::visibility

```
if (from.ocIsKindOf(UML::NamedElement)) then
  from.ocIsType(UML::NamedElement).visibility
else
  KerML::VisibilityKind::public
endif
```
- Element::aliasId

- Set{ }
- Relationship::target
  - OrderedSet{ElementMain\_Mapping.getMapped(from) }
- Relationship::source
  - OrderedSet{ElementMain\_Mapping.getMapped(from.owner) }
- OwningMembership::memberElement
  - elementGroup.to
- Membership::memberName
  - null
- OwningMembership::ownedRelatedElement
  - ```

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

```
- Membership::membershipOwningNamespace
  - Set{ElementMain\_Mapping(from) } -- will not be used since corresponding att is derived, but r
- Element::name
  - null
- Relationship::ownedRelatedElement
  - self.target()
- Membership::memberElement
  - ElementMain\_Mapping.getMapped(from)
- OwningMembership::memberName
  - Helper.getTagValueAsString(from, 'SysML::ModelElements::ElementGroup', 'name')
- Element::ownedRelationship
  - Set{ }

#### C.2.4.6.2.22 ProblemRationale\_Mapping

##### Description

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



```

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

```

## General Mappings

Comment\_Mapping

## Mapping Source

Comment

## Mapping Target

Comment

## Owned Mappings

(none)

## Applicable filters

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

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

## Mapping rules

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

- Element::aliasId
 

```
Set{}
```
- Element::ownedRelationship
 

```
ElementOwnership_Mapping.getMappedColl(from.ownedElement)
```
- Element::elementId
 

```
Helper.getID(from)
```
- Element::name
 

```
null
```
- Comment::ownedRelationship
 

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

```
null
```
- AnnotatingElement::annotation
 

```
Set{}
```

#### C.2.4.6.2.23 ProblemRationaleMetadataRedefinition\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToRedefinition\_Mapping

##### Mapping Source

Comment

##### Mapping Target

Redefinition

##### Owned Mappings

- problemRationaleMetadataReferenceUsage : ProblemRationaleMetadataReferenceUsage\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId

Set{}

- Redefinition::redefinedFeature

```
if Helper.hasStereotypeApplied(from, 'SysML::ModelElements::Problem') then
  SysML2::AttributeUsage.allInstances()->any(m | m.qualifiedName = 'ModelingMetadata::Issue:')
else if Helper.hasStereotypeApplied(from, 'SysML::ModelElements::Rationale') then
  SysML2::AttributeUsage.allInstances()->any(m | m.qualifiedName = 'ModelingMetadata::Rationale')
else
  OclUndefined
endif
endif
```

- Subsetting::ownedRelatedElement

Set{}

- Subsetting::subsettingFeature  
*abstract rule*

- Element::name

null

- Subsetting::subsettingFeature  
*abstract rule*
- Redefinition::redefiningFeature  
`problemRationaleMetadataReferenceUsage.to`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set{}`

#### **C.2.4.6.2.24 ProblemRationaleMetadataFeatureValueString\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToExpression\_Mapping

##### **Mapping Source**

Comment

##### **Mapping Target**

LiteralString

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`

- LiteralString::value  
from.body
- Feature::isUnique  
true
- Element::shortName  
null
- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false
- Element::ownedRelationship  
Set{}

#### **C.2.4.6.2.25 Stakeholder\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

## General Mappings

Class\_Mapping

## Mapping Source

Class

## Mapping Target

PartDefinition

## Owned Mappings

(none)

## Applicable filters

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

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

## Mapping rules

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

- Element::aliasId

```
Set{}
```

- Classifier::isAbstract

```
from.isAbstract
```

- Type::isSufficient

```
false
```

- Namespace::ownedImport

```
Set{}
```

- Element::elementId

```
Helper.getID(from)
```

- PartDefinition::ownedRelationship

```
let toClassifierMS: Sequence(UML::Element) = src.ownedElement->select(e | e.ocIsKindOf(UML::Classifier))
let excludeOwnedConcerns: Sequence(UML::Element) = src.ownedElement->select(e | e.ocIsKindOf(UML::Concern))
let toConcernMS: Sequence(UML::Element) = Helper.getTagValue(src, 'SysML::ModelElements::Stakeholder')
let toFeatureMS: Sequence(UML::Element) = src.ownedElement->select(e | e.ocIsKindOf(UML::Property))
let toElementOMS: Set(UML::Element) = ((src.ownedElement - toFeatureMS) - excludeOwnedConcerns)
let relationships: Sequence(UML::Element) =
  toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toFeatureMS->collect(e | PropertyMembership_Mapping.getMapped(e)))
```

```
->union(toClassifierMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))) in
if from.classifierBehavior.oclIsUndefined() then relationships else relationships->append(Cl
```

- Element::name

```
from.name
```

- Element::shortName

```
null
```

#### C.2.4.6.2.26 StakeholderMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToParameterMembership\_Mapping

##### Mapping Source

Classifier

##### Mapping Target

StakeholderMembership

##### Owned Mappings

- stakeholderPartUsage : StakeholderPartUsage\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- StakeholderMembership::ownedMemberParameter

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

- FeatureMembership::ownedRelatedElement

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

- Membership::membershipOwningNamespace  
*abstract rule*

- Membership::memberShortName

```
null
```

- Element::shortName  
null
- Element::elementId  
Helper.createUUID()
- Element::aliasId  
Set{}
- FeatureMembership::ownedMemberFeature  
*abstract rule*
- FeatureMembership::owningType  
*abstract rule*
- StakeholderMembership::memberName  
from.name
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
KerML::VisibilityKind::public
- Element::name  
null
- Element::ownedRelationship  
Set{}

#### C.2.4.6.2.27 StakeholderPartUsage\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToPartUsage\_Mapping

##### Mapping Source

Classifier

##### Mapping Target

PartUsage

##### Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Usage::isVariation  
`false`
- Feature::isReadOnly  
`false`
- Feature::direction  
`null`
- Element::name



null

- Feature::isDerived

    false

- Feature::isComposite

    false

- Element::ownedRelationship

    Set{}

#### **C.2.4.6.2.28 Viewpoint\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

Class\_Mapping

##### **Mapping Source**

Class

##### **Mapping Target**

ViewpointDefinition

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

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

##### **Mapping rules**

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

- Element::aliasId

    Set{}

- Classifier::isAbstract

    from.isAbstract

- Type::isSufficient

    false

- Namespace::ownedImport

```
Set{}
```

- ViewpointDefinition::ownedRelationship

```
let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::Property))
let redefinedAttributes: Set(UML::Element) = from.ownedElement->select(e | from.ocIsKindOf(UML::Attribute))
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.ocIsKindOf(UML::Generalization))
let toElementOMS: Set(UML::Element) = ((from.ownedElement - toElementFMS) - redefinedAttributes)
let relationships: Sequence(UML::Element) =
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(redefinedAttributes->collect(e | AttributeRedefinedMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))
->including(CommonReturnParameterReferenceUsageMembership_Mapping.getMapped(from))
->including(ViewpointSubjectMembership_Mapping.getMapped(from))
->including(ViewpointPurposeMetadataMembership_Mapping.getMapped(from)) in
if from.classifierBehavior.ocIsUndefined() then relationships else relationships->append(ClassifierBehavior)
```

- Element::elementId

```
Helper.getID(from)
```

- Element::name

```
from.name
```

- Element::shortName

```
null
```

#### C.2.4.6.2.29 ViewpointPurposeMetadata\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToMetadataUsage\_Mapping

##### Mapping Source

Class

##### Mapping Target

MetadataUsage

##### Owned Mappings

- viewpointPurposeMetadataFeatureTyping : ViewpointPurposeMetadataFeatureTyping\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- MetadataUsage::ownedRelationship  
`Set{viewpointPurposeMetadataFeatureTyping.to}`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Usage::isVariation  
`false`
- Feature::isReadOnly  
`false`
- Feature::direction  
`null`

- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false

#### **C.2.4.6.2.30 ViewpointPurposeMetadataFeatureTyping\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureTyping\_Mapping

##### **Mapping Source**

Class

##### **Mapping Target**

FeatureTyping

##### **Owned Mappings**

- viewpointPurposeMetadata : ViewpointPurposeMetadata\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Element::aliasId  
Set{ }
- Relationship::ownedRelatedElement  
Set{ }
- FeatureTyping::typedFeature  
viewpointPurposeMetadata.to
- Specialization::specific  
*abstract rule*

- Element::name

null

- Element::shortName

null

- Specialization::general

*abstract rule*

- FeatureTyping::type

`SysML2::MetadataDefinition.allInstances()->any(m | m.qualifiedName = 'SysMLv1Library::Viewpoint')`

- Element::elementId

`Helper.createUUID()`

- Element::ownedRelationship

`Set{}`

#### **C.2.4.6.2.31 ViewpointPurposeMetadataMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToOwningMembership\_Mapping

##### **Mapping Source**

Class

##### **Mapping Target**

OwningMembership

##### **Owned Mappings**

- viewpointPurposeMetadata : ViewpointPurposeMetadata\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Relationship::ownedRelatedElement

`Set{}`

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- Membership::memberElement  
*abstract rule*
- Element::shortName  
  
null
- OwningMembership::ownedMemberElement  
  
viewpointPurposeMetadata.to
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- Membership::memberName  
  
null
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- Element::ownedRelationship  
  
Set{}

#### **C.2.4.6.2.32 ViewpointSubject\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToReferenceUsage\_Mapping

##### **Mapping Source**

Class

##### **Mapping Target**

ReferenceUsage

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- ReferenceUsage::direction  
`KerML::FeatureDirectionKind::_in'`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Usage::isVariation  
`false`
- Feature::isReadOnly

false

- Element::name

null

- Feature::isDerived

false

- Feature::isComposite

false

- Element::ownedRelationship

Set{}

#### **C.2.4.6.2.33 ViewpointSubjectMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToParameterMembership\_Mapping

##### **Mapping Source**

Class

##### **Mapping Target**

SubjectMembership

##### **Owned Mappings**

- viewpointSubject : ViewpointSubject\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- FeatureMembership::ownedRelatedElement  
Set{self.ownedMemberFeature() }
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName



- SubjectMembership::ownedMemberParameter

- `Element::shortName`

- `Element::elementId`

- `Element::aliasId`

- FeatureMembership::ownedMemberFeature

- FeatureMembership::owningType

- Membership::memberName

- `TypeFeaturing::featureOfType`

- `TypeFeaturing::featuringType`

- Membership::visibility

- Element::name

- Element::ownedRelationship

### C.2.4.7.1 Overview

| SysML v1 Concept                   | SysML v2 Concept | Mapping Class             |
|------------------------------------|------------------|---------------------------|
| AcceptChangeStructuralFeatureEvent | Action           | *** not specified yet *** |
| AddFlowPropertyValueOnNestedPort   | Action           | *** not specified yet *** |
| ChangeStructuralFeatureEvent       |                  | *** not specified yet *** |
| DirectedFeature                    |                  | *** not specified yet *** |
| FlowProperty                       |                  | *** not specified yet *** |
| FullPort                           | PartUsage        | FullPort_Mapping          |

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

## C.2.4.7.2 Mapping Specifications

### C.2.4.7.2.1 AcceptChangeStructuralFeatureEventAction\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

AcceptEventAction\_Mapping

#### Mapping Source

AcceptEventAction

#### Mapping Target

AcceptActionUsage

#### Owned Mappings

(none)

#### Applicable filters

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

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

#### Mapping rules

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

- Feature::isEnd  
false
- Type::isSufficient  
false
- Feature::isUnique  
true

- ActionUsage::ownedRelationship  
Helper.actionOwnedRelationship(from)
- Element::name  
from.name
- Element::shortName  
null
- Type::isAbstract  
false
- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Usage::isVariation  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::elementId  
Helper.getID(from)
- ActionUsage::isComposite  
true
- Feature::isDerived  
false

#### **C.2.4.7.2.2 FullPort\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

Port\_Mapping

## Mapping Source

Port

## Mapping Target

PartUsage

## Owned Mappings

(none)

## Applicable filters

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

```
let p: UML::Property = from.oclAsType(UML::Property) in
if p.type.oclIsUndefined() then false else Helper.hasStereotypeApplied(p.type, 'SysML::Blocks::
and (p.association.oclIsUndefined() or p.association.ownedEnd->excludes(p)) and p.aggregation =
,
Helper.hasStereotypeApplied(src, 'SysML::Ports&Flows::FullPort')
```

## Mapping rules

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

- Feature::isOrdered

```
from.isOrdered
```

- Type::isSufficient

```
false
```

- Feature::isComposite

```
from.isComposite
```

- Feature::ownedRelationship

```
let typing: KerML::FeatureTyping = StructuralFeatureToFeatureTyping_Mapping.getMapped(from) in
let subsetting: Set(KerML::Subsetting) = from.subsettedProperty->collect(p | PropertySubsetting)
let subsettingMultiplicityTyping: Set(KerML::Relationship) = subsetting->union(if typing.oclIsTypeOf(KerML::Relationship)
Set{MultiplicityMembership_Mapping.getMapped(from)}
else
Set{MultiplicityMembership_Mapping.getMapped(from), typing}
endif)->asSet() in

let relationships: Set(KerML::Relationship) = if from.defaultValue.oclIsTypeOf(UML::OpaqueExpression) then
from.defaultValue.ownedEnd->collect(p | p.ownedEnd->excludes(p)) else
from.defaultValue.ownedEnd->collect(p | p.ownedEnd->excludes(p))
endif in

if from.defaultValue.oclIsUndefined() then
relationships
endif
```

```

else
    relationships->including(if from.defaultValue.ocIsTypeOf(UML::OpaqueExpression) then Pro
endif

```

- Feature::isAbstract

```

false

```

- Feature::isEnd

```

if from.association.ocIsUndefined() then
    false
else
    from.association.ownedEnd->includes(from)
endif

```

- Element::name

```

from.name

```

- Element::shortName

```

null

```

- Element::aliasId

```

Set{}

```

- Feature::isPortion

```

false

```

- Feature::isDerived

```

from.isDerived

```

- Feature::direction

```

null

```

- Element::elementId

```

Helper.getID(from)

```

- Feature::isUnique

```

from.isUnique

```

- Feature::isReadOnly

```

abstract rule

```

#### C.2.4.7.2.3 InterfaceBlock\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

Block\_Mapping

## Mapping Source

Class

## Mapping Target

PortDefinition

## Owned Mappings

(none)

## Applicable filters

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

```
not Helper.hasStereotypeApplied(from, 'SysML::Requirements::Requirement') and not from.ocIsTypeOf(
,
Helper.hasStereotypeApplied(from, 'SysML::Ports&Flows::InterfaceBlock')
```

## Mapping rules

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

- Element::aliasId  

```
Set{}
```
- Classifier::isAbstract  

```
from.isAbstract
```
- Type::isSufficient  

```
false
```
- Namespace::ownedImport  

```
Set{}
```
- Classifier::ownedRelationship

```
let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | (e.ocIsKindOf(UML::Prop
let redefinedAttributes: Set(UML::Element) = from.ownedElement->select(e | from.ocIsKindOf(U
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.ocIsKindOf
let constraints : Set(UML::Constraint) = UML::Constraint.allInstances()->select( c | c.constr
let toElementOMS: Set(UML::Element) = ((from.ownedElement - toElementFMS) - redefinedAttribu
let relationships: Sequence(KerML::Relationship) =
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(constraints->collect(e | ConstrainedElementFeatureMembership_Mapping.getMapped(e)))
->union(redefinedAttributes->collect(e | AttributeRedefinedMembership_Mapping.getMapped(e)))
```

```
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e))) in
if from.classifierBehavior.oclIsUndefined() then relationships else relationships->append(Cla
```

- Element::elementId

```
Helper.getID(from)
```

- Element::name

```
from.name
```

- Element::shortName

```
null
```

#### C.2.4.7.2.4 ItemFlow\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToRelationship\_Mapping  
NamedElementMain\_Mapping

##### Mapping Source

InformationFlow

##### Mapping Target

FlowConnectionUsage

##### Owned Mappings

- itemFlowFeatureMembership : ItemFlowFeatureMembership\_Mapping
- itemFlowSourceEndFeatureMembership : ItemFlowSourceEndFeatureMembership\_Mapping
- itemFlowTargetEndFeatureMembership : ItemFlowTargetEndFeatureMembership\_Mapping

##### Applicable filters

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

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

##### Mapping rules

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

- Element::aliasId

```
Set{}
```

- FlowConnectionUsage::target

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

- Element::elementId

Helper.getID(from)

- Element::name

null

- FlowConnectionUsage::ownedRelationship

Set{itemFlowFeatureMembership.to, itemFlowSourceEndFeatureMembership.to, itemFlowTargetEndFe

- Element::shortName

null

- FlowConnectionUsage::source

NamedElementMain\_Mapping.getMappedColl(from.informationSource)

#### **C.2.4.7.2.5 ItemFlowFeatureMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureMembership\_Mapping

##### **Mapping Source**

InformationFlow

##### **Mapping Target**

FeatureMembership

##### **Owned Mappings**

- itemFlowItemFeature : ItemFlowItemFeature\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- FeatureMembership::ownedMemberFeature

itemFlowItemFeature.to

- Membership::membershipOwningNamespace

*abstract rule*



- Membership::memberShortName  
null
- Element::shortName  
null
- Element::elementId  
Helper.createUUID()
- Element::aliasId  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
null
- OwningMembership::ownedRelatedElement  
Set{self.ownedMemberElement() }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
KerML::VisibilityKind::public
- Element::name  
null
- Element::ownedRelationship  
Set{}

#### C.2.4.7.2.6 ItemFlowItemFeature\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeature\_Mapping

##### Mapping Source

InformationFlow

##### Mapping Target

ItemFeature

### Owned Mappings

- itemFlowItemFeatureTyping : ItemFlowItemFeatureTyping\_Mapping

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  
`Set{}`
- Type::isSufficient  
`false`
- ItemFeature::ownedRelationship  
`Set{itemFlowItemFeatureTyping.to}`
- Element::name  
`null`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`

#### C.2.4.7.2.7 ItemFlowItemFeatureTyping\_Mapping

##### Description

Currently, only one conveyed item is supported

##### General Mappings

GenericToFeatureTyping\_Mapping

##### Mapping Source

InformationFlow

##### Mapping Target

FeatureTyping

### Owned Mappings

- itemFlowItemFeature : ItemFlowItemFeature\_Mapping

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  

```
Set{}
```
- Relationship::ownedRelatedElement  

```
Set{}
```
- FeatureTyping::type  

```
if from.conveyed->size() > 0 then
Classifier_Mapping.getMapped(from.conveyed.get(0))
else OclUndefined
endif
```
- Specialization::specific  
*abstract rule*
- Element::name  

```
null
```
- Element::shortName  

```
null
```
- Specialization::general  
*abstract rule*
- Element::elementId  

```
Helper.createUUID()
```
- Element::ownedRelationship  

```
Set{}
```
- FeatureTyping::typedFeature  

```
itemFlowItemFeature.to
```

#### C.2.4.7.2.8 ItemFlowSourceEndFeatureMembership\_Mapping

### Description

\*\*\* not specified yet \*\*\*

## General Mappings

GenericToEndFeatureMembership\_Mapping

## Mapping Source

InformationFlow

## Mapping Target

FeatureMembership

## Owned Mappings

- itemFlowSourceFeature : ItemFlowSourceFeature\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- FeatureMembership::ownedRelatedElement  
`Set{self.ownedMemberFeature() }`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{ }`
- FeatureMembership::owningType  
*abstract rule*
- Membership::memberName  
`null`
- TypeFeaturing::featureOfType  
*abstract rule*

- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
`KerML::VisibilityKind::public`
- Element::name  
`null`
- FeatureMembership::ownedMemberFeature  
`itemFlowSourceFeature.to`
- Element::ownedRelationship  
`Set{}`

#### C.2.4.7.2.9 ItemFlowSourceFeature\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeature\_Mapping

##### Mapping Source

InformationFlow

##### Mapping Target

ItemFlowEnd

##### Owned Mappings

- itemFlowSourceFeatureSubsetting : ItemFlowSourceFeatureSubsetting\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId  
`Set{}`
- ItemFlowEnd::isEnd  
`true`

- Type::isSufficient  
false
- ItemFlowEnd::ownedRelationship  
Set{itemFlowSourceFeatureSubsetting.to}
- Element::name  
null
- Element::shortName  
null
- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()

#### **C.2.4.7.2.10 ItemFlowSourceFeatureSubsetting\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToSubsetting\_Mapping

##### **Mapping Source**

InformationFlow

##### **Mapping Target**

Subsetting

##### **Owned Mappings**

- itemFlowSourceFeature : ItemFlowSourceFeature\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Element::aliasId  
Set{ }

- Relationship::ownedRelatedElement

```
Set{}
```

- Subsetting::subsettingFeature

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

- Specialization::specific

*abstract rule*

- Element::name

```
null
```

- Element::shortName

```
null
```

- Subsetting::subsettingFeature

```
itemFlowSourceFeature.to
```

- Specialization::general

*abstract rule*

- Element::elementId

```
Helper.createUUID()
```

- Element::ownedRelationship

```
Set{}
```

#### C.2.4.7.2.11 ItemFlowTargetEndFeatureMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToEndFeatureMembership\_Mapping

##### Mapping Source

InformationFlow

##### Mapping Target

FeatureMembership

##### Owned Mappings

- itemFlowTargetFeature : ItemFlowTargetFeature\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- FeatureMembership::ownedRelatedElement  
`Set{self.ownedMemberFeature() }`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{ }`
- FeatureMembership::owningType  
*abstract rule*
- Membership::memberName  
`null`
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
`KerML::VisibilityKind::public`
- Element::name  
`null`
- FeatureMembership::ownedMemberFeature  
`itemFlowTargetFeature.to`
- Element::ownedRelationship  
`Set{ }`

### C.2.4.7.2.12 ItemFlowTargetFeature\_Mapping

#### Description

\*\*\* not specified yet \*\*\*



## General Mappings

GenericToFeature\_Mapping

## Mapping Source

InformationFlow

## Mapping Target

ItemFlowEnd

## Owned Mappings

- itemFlowTargetFeatureSubsetting : ItemFlowTargetFeatureSubsetting\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Element::aliasId  
`Set{}`
- Type::isSufficient  
`false`
- ItemFlowEnd::ownedRelationship  
`Set{itemFlowTargetFeatureSubsetting.to}`
- ItemFlowEnd::isEnd  
`true`
- Element::name  
`null`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`

#### C.2.4.7.2.13 ItemFlowTargetFeatureSubsetting\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToSubsetting\_Mapping

##### Mapping Source

InformationFlow

##### Mapping Target

Subsetting

##### Owned Mappings

- itemFlowTargetFeature : ItemFlowTargetFeature\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId  
`Set{ }`
- Relationship::ownedRelatedElement  
`Set{ }`
- Specialization::specific  
*abstract rule*
- Subsetting::subsettingFeature  
`itemFlowTargetFeature.to`
- Subsetting::subsettingFeature  
`from.target.get(0)`
- Element::name  
`null`
- Element::shortName  
`null`

- Specialization::general  
*abstract rule*
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set{}`

#### **C.2.4.7.2.14 OperationDirectedFeature\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

Operation\_Mapping

##### **Mapping Source**

Operation

##### **Mapping Target**

PerformActionUsage

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

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

##### **Mapping rules**

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::name  
`from.name`

- `Element::shortName`  
`null`
- `Type::isAbstract`  
`false`
- `Feature::isOrdered`  
`false`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `Usage::isVariation`  
`false`
- `Namespace::ownedImport`  
`Set{}`
- `Feature::isReadOnly`  
`false`
- `Element::elementId`  
`Helper.getID(from)`
- `Feature::isDerived`  
`false`
- `ActionUsage::isComposite`  
`true`
- `PerformActionUsage::direction`  
`Helper.getKerMLFeatureDirectionKind(Helper.getTagValueAsElement(from, 'SysML::Ports&Flows::Di')`
- `Namespace::ownedRelationship`  
`from.ownedElement->collect(e | ElementOwningMembership_Mapping.getMapped(e))`

#### C.2.4.8 Requirements

### C.2.4.8.1 Overview

Table 11. List of all Overview Mapping Specifications

| SysML v1 Concept    | SysML v2 Concept                  | Mapping Class             |
|---------------------|-----------------------------------|---------------------------|
| AbstractRequirement |                                   | *** not specified yet *** |
| Copy                |                                   | *** not specified yet *** |
| DeriveReq           |                                   | DeriveReq_Mapping         |
| Refine              |                                   | Refine_Mapping            |
| Requirement         | RequirementUsage                  | Requirement_Mapping       |
| Satisfy             | SatisfyRequirementUsage           | Satisfy_Mapping           |
| TestCase            | VerificationCaseDefinition        | TestCaseActivity_Mapping  |
| Trace               | Dependency                        | Trace_Mapping             |
| Verify              | RequirementVerificationMembership | Verify_Mapping            |

### C.2.4.8.2 SysML v1 Requirements elements not mapped

Table 12. List of SysML v1 elements not mapped of this section

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

### C.2.4.8.3 Mapping Specifications

#### C.2.4.8.3.1 Requirement\_Mapping

##### Description

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

##### General Mappings

GenericToUsage\_Mapping  
NamedElementMain\_Mapping

##### Mapping Source

Class

##### Mapping Target

RequirementUsage

##### Owned Mappings

(none)

##### Applicable filters

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

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

## Mapping rules

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

- `Feature::isEnd`  
`false`
- `Type::isSufficient`  
`false`
- `Feature::isUnique`  
`true`
- `Element::shortName`  
`null`
- `Type::isAbstract`  
`false`
- `Feature::isOrdered`  
`false`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `RequirementUsage::reqId`  
`Helper.getTagValueAsString(from, 'SysML::Requirements::Requirement', 'id')`
- `Feature::isReadOnly`  
`false`
- `Feature::direction`  
`null`
- `Element::elementId`  
`Helper.getID(from)`
- `Element::name`  
`null`
- `Feature::isDerived`  
`false`

- Feature::isComposite

```
false
```

- RequirementUsage::ownedRelationship

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

#### C.2.4.8.3.2 DeriveReq\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

Abstraction\_Mapping

##### Mapping Source

Abstraction

##### Mapping Target

Dependency

##### Owned Mappings

(none)

##### Applicable filters

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

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

##### Mapping rules

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

- Element::aliasId

```
Set{}
```

- Dependency::supplier

```
from.target->collect(e | ElementMain_Mapping.getMapped(e))
```

- Dependency::name

```
from.name
```

- Element::ownedRelationship

```
ElementOwnership_Mapping.getMappedColl(from.ownedElement)
```

- Relationship::owningRelatedElement

```
ElementMain_Mapping.getMapped(from.owner)
```

- Dependency::client

```
from.source->collect(e | ElementMain_Mapping.getMapped(e))
```

- Element::elementId

```
Helper.getID(from)
```

- Relationship::ownedRelatedElement

```
from.relatedElement->select(e | from.ownedElement->includes(e))->collect(e | ElementMain_Mapping.getMapped(e))
```

- Element::shortName

```
null
```

#### C.2.4.8.3.3 Refine\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

Abstraction\_Mapping

##### Mapping Source

Abstraction

##### Mapping Target

Dependency

##### Owned Mappings

(none)

##### Applicable filters

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

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

##### Mapping rules

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

- Element::aliasId

```
Set{}
```

- Dependency::supplier



```
from.target->collect(e | ElementMain_Mapping.getMapped(e))
```

- **Dependency::name**

```
from.name
```

- **Element::ownedRelationship**

```
ElementOwnership_Mapping.getMappedColl(from.ownedElement)
```

- **Relationship::owningRelatedElement**

```
ElementMain_Mapping.getMapped(from.owner)
```

- **Dependency::client**

```
from.source->collect(e | ElementMain_Mapping.getMapped(e))
```

- **Element::elementId**

```
Helper.getID(from)
```

- **Relationship::ownedRelatedElement**

```
from.relatedElement->select(e | from.ownedElement->includes(e))->collect(e | ElementMain_Mapping.getMapped(e))
```

- **Element::shortName**

```
null
```

#### **C.2.4.8.3.4 RequirementDocumentation\_Mapping**

##### **Description**

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

##### **General Mappings**

GenericToDocumentation\_Mapping

##### **Mapping Source**

NamedElement

##### **Mapping Target**

Documentation

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- `Element::aliasId`  
`Set{}`
- `Documentation::body`  
`Helper.getTagValueAsString(from, 'SysML::Requirements::Requirement', 'text')`
- `Comment::locale`  
`null`
- `Element::name`  
`null`
- `Element::shortName`  
`null`
- `AnnotatingElement::annotation`  
`Set{}`
- `Element::elementId`  
`Helper.createUUID()`
- `Element::ownedRelationship`  
`Set{}`

### C.2.4.8.3.5 RequirementDocumentationMembership\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

GenericToOwningMembership\_Mapping

#### Mapping Source

NamedElement

#### Mapping Target

OwningMembership

#### Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Relationship::ownedRelatedElement  
`Set{}`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Membership::memberElement  
*abstract rule*
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{}`
- OwningMembership::ownedMemberElement  
`RequirementDocumentation_Mapping.getMapped(from)`
- Membership::memberName  
`null`
- Membership::visibility  
`KerML::VisibilityKind::public`
- Element::name  
`null`
- Element::ownedRelationship  
`Set{}`

#### C.2.4.8.3.6 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

NamedElement

### Mapping Target

ReferenceUsage

### Owned Mappings

(none)

### Applicable filters

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

(none)

### Mapping rules

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

- Feature::isEnd  
false
- Type::isSufficient  
false
- ReferenceUsage::direction  
KerML::FeatureDirectionKind::\_in'
- Feature::isUnique  
true
- Element::shortName  
null
- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Feature::isOrdered

- `Element::aliasId`

- Feature::isPortion

- Usage::isVariation

- Feature::isReadOnly

- `Element::name`

- Feature::isDerived

- Feature::isComposite

- Element::ownedRelationship

#### C.2.4.8.3.7 RequirementSubjectMembership\_Mapping

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

## GenericToParameterMembership\_Mapping

## NamedElement

## SubjectMembership

(none)

OMG Systems Modeling Language (SysML) v2.0, Submission

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- FeatureMembership::ownedRelatedElement  
`Set { self.ownedMemberFeature () }`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set {}`
- FeatureMembership::ownedMemberFeature  
*abstract rule*
- FeatureMembership::owningType  
*abstract rule*
- SubjectMembership::ownedMemberParameter  
`RequirementSubject_Mapping.getMapped(from)`
- Membership::memberName  
`null`
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
`KerML::VisibilityKind::public`
- Element::name  
`null`
- Element::ownedRelationship  
`Set {}`

#### C.2.4.8.3.8 Satisfy\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToOccurrenceUsage\_Mapping  
Abstraction\_Mapping

##### Mapping Source

Abstraction

##### Mapping Target

SatisfyRequirementUsage

##### Owned Mappings

- satisfyFeatureTyping : SatisfyFeatureTyping\_Mapping
- satisfySubjectMembership : SatisfySubjectMembership\_Mapping

##### Applicable filters

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

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

##### Mapping rules

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

- Feature::isEnd

```
false
```

- Dependency::name

```
from.name
```

- Type::isSufficient

```
false
```

- Relationship::owningRelatedElement

```
ElementMain_Mapping.getMapped(from.owner)
```

- Feature::isUnique

```
true
```

- SatisfyRequirementUsage::ownedRelationship

```
Set{satisfyFeatureTyping.to, satisfySubjectMembership.to, SatisfyFeatureMembership_Mapping.g
```

- `Element::shortName`  
`null`
- `Type::isAbstract`  
`false`
- `Feature::isOrdered`  
`false`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `Usage::isVariation`  
`false`
- `Dependency::supplier`  
`from.target->collect(e | ElementMain_Mapping.getMapped(e))`
- `Feature::isReadOnly`  
`false`
- `Dependency::client`  
`from.source->collect(e | ElementMain_Mapping.getMapped(e))`
- `Feature::direction`  
`null`
- `Element::elementId`  
`Helper.getID(from)`
- `Feature::isDerived`  
`false`
- `Relationship::ownedRelatedElement`  
`from.relatedElement->select(e | from.ownedElement->includes(e))->collect(e | ElementMain_Map`
- `Feature::isComposite`  
`false`

#### **C.2.4.8.3.9 SatisfyFeatureMembership\_Mapping**

##### **Description**



\*\*\* not specified yet \*\*\*

## General Mappings

GenericToFeatureMembership\_Mapping

## Mapping Source

Abstraction

## Mapping Target

FeatureMembership

## Owned Mappings

- satisfyFeatureMembershipReferenceUsage : SatisfyFeatureMembershipReferenceUsage\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- FeatureMembership::ownedMemberFeature  
  
satisfyFeatureMembershipReferenceUsage.to
- FeatureMembership::memberName  
  
'satisfyingFeature'
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{ }
- OwningMembership::ownedMemberElement  
*abstract rule*
- OwningMembership::ownedRelatedElement

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

- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- FeatureMembership::memberFeature

```
self.ownedMemberFeature()
```

- Membership::visibility

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

- Element::name

```
null
```

- Element::ownedRelationship

```
Set{ }
```

#### **C.2.4.8.3.10 SatisfyFeatureMembershipReferenceUsage\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToReferenceUsage\_Mapping

##### **Mapping Source**

Abstraction

##### **Mapping Target**

ReferenceUsage

##### **Owned Mappings**

- satisfyFeatureMembershipReferenceUsageFeatureTyping :  
SatisfyFeatureMembershipReferenceUsageFeatureTyping\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Feature::isEnd

```
false
```

- `Type::isSufficient`  
`false`
- `Feature::isUnique`  
`true`
- `ReferenceUsage::ownedRelationship`  
`Set{satisfyFeatureMembershipReferenceUsageFeatureTyping.to}`
- `Element::shortName`  
`null`
- `Type::isAbstract`  
`false`
- `Element::elementId`  
`Helper.createUUID()`
- `Feature::isOrdered`  
`false`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `Usage::isVariation`  
`false`
- `Feature::isReadOnly`  
`false`
- `Feature::direction`  
`null`
- `Element::name`  
`null`
- `Feature::isDerived`  
`false`
- `Feature::isComposite`  
`false`

#### C.2.4.8.3.11 SatisfyFeatureMembershipReferenceUsageFeatureTyping\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureTyping\_Mapping

##### Mapping Source

Abstraction

##### Mapping Target

FeatureTyping

##### Owned Mappings

- satisfyFeatureMembershipReferenceUsage : SatisfyFeatureMembershipReferenceUsage\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId

Set{ }

- Relationship::ownedRelatedElement

Set{ }

- Specialization::specific

*abstract rule*

- FeatureTyping::type

if Satisfy\_Mapping.getMapped(from).client->size() > 0 then Satisfy\_Mapping.getMapped(from).c

- FeatureTyping::typedFeature

satisfyFeatureMembershipReferenceUsage.to

- Element::name

null

- Element::shortName

null

- Specialization::general

*abstract rule*

- Element::elementId

`Helper.createUUID()`

- Element::ownedRelationship

`Set{}`

#### C.2.4.8.3.12 SatisfyFeatureTyping\_Mapping

##### Description

The type of the feature typing element is the client of the satisfy relationship. In SysML v1, the satisfy relationship can have only one client element. However, if there is more than one client element, the first one is taken and the others are ignored.

##### General Mappings

GenericToFeatureTyping\_Mapping

##### Mapping Source

Abstraction

##### Mapping Target

FeatureTyping

##### Owned Mappings

(none)

##### Applicable filters

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

(none)

##### Mapping rules

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

- Element::aliasId

`Set{}`

- Relationship::ownedRelatedElement

`Set{}`

- FeatureTyping::typedFeature

`Satisfy_Mapping.getMapped(from)`

- Specialization::specific

*abstract rule*

- Element::name

null

- Element::shortName

null

- Specialization::general

*abstract rule*

- Element::elementId

Helper.createUUID()

- FeatureTyping::type

if Satisfy\_Mapping.getMapped(from).supplier->size() > 0 then Satisfy\_Mapping.getMapped(from)

- Element::ownedRelationship

Set{}

#### C.2.4.8.3.13 SatisfySubjectMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToParameterMembership\_Mapping

##### Mapping Source

Abstraction

##### Mapping Target

SubjectMembership

##### Owned Mappings

- satisfySubjectMembershipReferenceUsage : SatisfySubjectMembershipReferenceUsage\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- FeatureMembership::ownedRelatedElement

Set{self.ownedMemberFeature() }

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- FeatureMembership::ownedMemberFeature  
*abstract rule*
- FeatureMembership::owningType  
*abstract rule*
- SubjectMembership::ownedMemberParameter  
  
satisfySubjectMembershipReferenceUsage.to
- SubjectMembership::memberParameter  
  
self.ownedMemberParameter()
- Membership::memberName  
  
null
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- Element::ownedRelationship  
  
Set{}

#### **C.2.4.8.3.14 SatisfySubjectMembershipFeatureValue\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureValue\_Mapping

### Mapping Source

Abstraction

### Mapping Target

FeatureValue

### Owned Mappings

- satisfySubjectMembershipFeatureValueFeatureReferenceExpression :  
SatisfySubjectMembershipFeatureValueFeatureReferenceExpression\_Mapping

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- FeatureValue::value  
`satisfySubjectMembershipFeatureValueFeatureReferenceExpression.to`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- FeatureValue::memberElement  
`satisfySubjectMembershipFeatureValueFeatureReferenceExpression.to`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{}`
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
`null`
- OwningMembership::ownedRelatedElement



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

- Membership::visibility

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

- Element::name

```
null
```

- Element::ownedRelationship

```
Set{ }
```

#### **C.2.4.8.3.15 SatisfySubjectMembershipFeatureValueFeatureReferenceExpression\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureReferenceExpression\_Mapping

##### **Mapping Source**

Abstraction

##### **Mapping Target**

FeatureReferenceExpression

##### **Owned Mappings**

- satisfySubjectMembershipFeatureValueFeatureReferenceExpressionMembership :  
SatisfySubjectMembershipFeatureValueFeatureReferenceExpressionMembership\_Mapping
- satisfySubjectMembershipFeatureValueFeatureReferenceExpressionReturnParameterMembership :  
SatisfySubjectMembershipFeatureValueFeatureReferenceExpressionReturnParameterMembership\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Feature::isEnd

```
false
```

- Type::isSufficient

```
false
```

- Feature::isUnique

- `Element::shortName`

- `Type::isAbstract`

- `Element::elementId`

- Feature::isOrdered

- `Element::aliasId`

- Feature::isPortion

- `Feature::isReadOnly`

- Feature::direction

- `Element::name`

- Feature::isDerived

- FeatureReferenceExpression::ownedRelationship

- Feature::isComposite

#### C.2.4.8.3.16 SatisfySubjectMembershipFeatureValueFeatureReferenceExpressionMembership\_Mapping

\*\*\* not specified yet \*\*\*

## GenericToMembership\_Mapping

## Mapping Source

Abstraction

## Mapping Target

Membership

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Element::aliasId  
`Set{}`
- Relationship::ownedRelatedElement  
`Set{}`
- Membership::memberElement  
`SatisfyFeatureMembershipReferenceUsage_Mapping.getMapped(from)`
- Relationship::source  
`Set{}`
- Element::name  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Relationship::target  
`Set{}`
- Element::ownedRelationship  
`Set{}`

#### C.2.4.8.3.17

### SatisfySubjectMembershipFeatureValueFeatureReferenceExpressionReturnParameterMembership\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

GenericToParameterMembership\_Mapping

#### Mapping Source

Abstraction

#### Mapping Target

ReturnParameterMembership

#### Owned Mappings

- satisfySubjectMembershipFeatureValueFeatureReferenceExpressionReturnParameterMembershipFeature  
: SatisfySubjectMembershipFeatureValueFeatureReferenceExpressionReturnParameterMembershipFeature\_Mapping

#### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

#### Mapping rules

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

- FeatureMembership::ownedRelatedElement  

```
Set{self.ownedMemberFeature() }
```
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  

```
null
```
- ReturnParameterMembership::ownedRelatedElement  

```
let member: KerML::Element = self.ownedMemberParameter() in
if member.ocIsUndefined() then
  Set{}
else
  Set{self.ownedMemberParameter() }
endif
```
- ReturnParameterMembership::memberParameter  

```
self.ownedMemberParameter()
```

- Element::shortName  
null
- Element::elementId  
Helper.createUUID()
- Element::aliasId  
Set{ }
- FeatureMembership::ownedMemberFeature  
*abstract rule*
- FeatureMembership::owningType  
*abstract rule*
- Membership::memberName  
null
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
KerML::VisibilityKind::public
- Element::name  
null
- ReturnParameterMembership::ownedMemberParameter  
satisfySubjectMembershipFeatureValueFeatureReferenceExpressionReturnParameterMembershipFeatu
- Element::ownedRelationship  
Set{ }

#### **C.2.4.8.3.18**

#### **SatisfySubjectMembershipFeatureValueFeatureReferenceExpressionReturnParameterMembershipFeature\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeature\_Mapping

##### **Mapping Source**

Abstraction

##### **Mapping Target**

Feature

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Element::aliasId  
`Set{}`
- Type::isSufficient  
`false`
- Element::name  
`null`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set{}`

### C.2.4.8.3.19 SatisfySubjectMembershipReferenceUsage\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

GenericToReferenceUsage\_Mapping

#### Mapping Source

Abstraction

#### Mapping Target

ReferenceUsage

## Owned Mappings

- satisfySubjectMembershipFeatureValue : SatisfySubjectMembershipFeatureValue\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- ReferenceUsage::direction  
`KerML::FeatureDirectionKind::_in'`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Usage::isVariation  
`false`
- ReferenceUsage::ownedRelationship

```
Set{satisfySubjectMembershipFeatureValue.to}
```

- Feature::isReadOnly

```
false
```

- Element::name

```
null
```

- Feature::isDerived

```
false
```

- Feature::isComposite

```
false
```

#### **C.2.4.8.3.20 TestCaseActivity\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

ActivityAsDefinition\_Mapping

##### **Mapping Source**

Activity

##### **Mapping Target**

VerificationCaseDefinition

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

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

```
,
```

```
true
```

##### **Mapping rules**

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

- Element::aliasId

```
Set{ }
```



- Classifier::isAbstract

```
from.isAbstract
```

- Type::isSufficient

```
false
```

- Namespace::ownedImport

```
Set{}
```

- Element::elementId

```
Helper.getID(from)
```

- Element::name

```
from.name
```

- Element::shortName

```
null
```

- VerificationCaseDefinition::ownedRelationship

```
let relationships : Set(KerML::Relationship) = Helper.activityOwnedRelationship(from) in
let verdictParameter : Set(UML::Parameter) = from.ownedElement->select(e | e.ocIsKindOf(UML::Parameter))
let parameters : Set(UML::Parameter) = ((from.ownedElement->select(e | e.ocIsKindOf(UML::Parameter)))
let verifyRelationships : Set(UML::Abstraction) = from.clientDependency->select(v | Helper.verifyRelationships
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)))
```

#### C.2.4.8.3.21 TestCaseActivityReturnParameterMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

ParameterMembership\_Mapping

##### Mapping Source

Parameter

##### Mapping Target

ReturnParameterMembership

##### Owned Mappings

(none)

### Applicable filters

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

(none)

### Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- FeatureMembership::owningType  
*abstract rule*
- ParameterMembership::ownedMemberParameter  
  
null
- Membership::memberName  
  
null
- ParameterMembership::ownedRelatedElement  
  
Set{self.ownedMemberParameter() }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- Element::ownedRelationship  
  
Set{}

#### C.2.4.8.3.22 TestCaseVerifyObjectiveMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

CaseObjectiveMembership\_Mapping

##### Mapping Source

Abstraction

##### Mapping Target

ObjectiveMembership

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- FeatureMembership::ownedRelatedElement  
`Set{self.ownedMemberFeature() }`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{ }`
- FeatureMembership::ownedMemberFeature  
*abstract rule*
- FeatureMembership::owningType  
*abstract rule*

- Membership::memberName  
null
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- ObjectiveMembership::ownedMemberFeature  
TestCaseVerifyObjectiveRequirementUsage\_Mapping.getMapped(from)
- Membership::visibility  
KerML::VisibilityKind::public
- Element::name  
null
- Element::ownedRelationship  
Set{}

#### **C.2.4.8.3.23 TestCaseVerifyObjectiveRequirementUsage\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

CaseObjectiveRequirementUsage\_Mapping

##### **Mapping Source**

Abstraction

##### **Mapping Target**

RequirementUsage

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Feature::isEnd

- false
- Type::isSufficient
  - false
- Feature::isUnique
  - true
- Element::shortName
  - null
- Type::isAbstract
  - false
- Element::elementId
  - Helper.createUUID()
- Feature::isOrdered
  - false
- RequirementUsage::ownedRelationship
  - Set{Verify\_Mapping.getMapped(from)}
- Element::aliasId
  - Set{}
- Feature::isPortion
  - false
- Usage::isVariation
  - false
- Feature::isReadOnly
  - false
- Feature::direction
  - null
- Element::name
  - null
- Feature::isDerived
  - false
- Feature::isComposite

false

- Element::ownedRelationship

Set{ }

#### C.2.4.8.3.24 TestCaseVerifyRequirementUsageReferenceSubsetting\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToSubsetting\_Mapping

##### Mapping Source

Abstraction

##### Mapping Target

ReferenceSubsetting

##### Owned Mappings

(none)

##### Applicable filters

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

(none)

##### Mapping rules

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

- Element::aliasId

Set{ }

- Relationship::ownedRelatedElement

Set{ }

- Specialization::specific

*abstract rule*

- ReferenceSubsetting::referencedFeature

from.supplier->get(0)

- Element::name

null

- Element::shortName

null

- Specialization::general  
*abstract rule*
- Element::elementId

Helper.createUUID()

- Element::ownedRelationship

Set{}

#### C.2.4.8.3.25 TestCaseVerifyRequirementUsage\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToUsage\_Mapping

##### Mapping Source

Abstraction

##### Mapping Target

RequirementUsage

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Feature::isEnd

false

- Type::isSufficient

false

- Feature::isUnique

true

- Element::shortName

- `Type::isAbstract`

- `Element::elementId`

- Feature::isOrdered

- RequirementUsage::ownedRelationship

- `Element::aliasId`

- Feature::isPortion

- `Feature::isReadOnly`

- Feature::direction

- `Element::name`

- Feature::isDerived

- Feature::isComposite

#### C.2.4.8.3.26 Trace\_Mapping

## Description

\*\*\* not specified yet \*\*\*

## General Mappings

## Abstraction\_Mapping

## Mapping Source

## Abstraction



## Mapping Target

Dependency

## Owned Mappings

(none)

## Applicable filters

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

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

## Mapping rules

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

- Element::aliasId

```
Set{}
```

- Dependency::supplier

```
from.target->collect(e | ElementMain_Mapping.getMapped(e))
```

- Dependency::name

```
from.name
```

- Element::ownedRelationship

```
ElementOwnership_Mapping.getMappedColl(from.ownedElement)
```

- Relationship::owningRelatedElement

```
ElementMain_Mapping.getMapped(from.owner)
```

- Dependency::client

```
from.source->collect(e | ElementMain_Mapping.getMapped(e))
```

- Element::elementId

```
Helper.getID(from)
```

- Relationship::ownedRelatedElement

```
from.relatedElement->select(e | from.ownedElement->includes(e))->collect(e | ElementMain_Mapping.getMapped(e))
```

- Element::shortName

```
null
```

### C.2.4.8.3.27 Verify\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

## General Mappings

GenericToRelationship\_Mapping

## Mapping Source

Abstraction

## Mapping Target

RequirementVerificationMembership

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Element::aliasId  
`Set{ }`
- RequirementVerificationMembership::ownedRelatedElement  
`Set{ TestCaseVerifyRequirementUsage_Mapping.getMapped(from) }`
- Element::name  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set{ }`

## C.2.5 UML4SysML

### C.2.5.1 Overview

### C.2.5.2 Actions

### C.2.5.2.1 Overview

**Table 13. List of all Overview Mapping Specifications**

| SysML v1 Concept                    | SysML v2 Concept                                                                                                              | Mapping Class                                                                                                                                                                                                                                                                                                                                                                      | Filter                                                                                                                                                                              |
|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AcceptCallAction                    | AcceptActionUsage                                                                                                             | AcceptCallAction_Mapping                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                                     |
| AcceptEventAction                   | FeatureTyping<br>ReferenceUsage<br>ParameterMembership<br>AcceptActionUsage                                                   | AcceptEventActionParameterFeatureTyping_Mapping<br>AcceptEventActionParameter_Mapping<br>AcceptEventActionParameterMembership_Mapping<br>AcceptEventAction_Mapping                                                                                                                                                                                                                 |                                                                                                                                                                                     |
| Action                              | ActionUsage                                                                                                                   | CommonAction_Mapping                                                                                                                                                                                                                                                                                                                                                               |                                                                                                                                                                                     |
| ActionInputPin                      | FeatureReferenceExpression<br>Feature<br>ReferenceUsage<br>FeatureTyping<br>FeatureValue<br>ParameterMembership<br>Membership | RemoveVariableValueActionExpressionParameterFeatureReference_Mapping<br>RemoveVariableValueActionExpressionParameter_Mapping<br>UntypedPin_Mapping<br>PinFeatureTyping_Mapping<br>RemoveVariableValueActionExpressionParameterValue_Mapping<br>RemoveVariableValueActionExpressionParameterMembership_Mapping<br>RemoveVariableValueActionExpressionParameterFeatureReferenceMembe | ActionInputPin.type.ocllsUndefined()<br>not<br>src.type.ocllsUndefined()<br>and<br>not(src.type.ocllsKindOf(UML::Enumeration))<br>and<br>Helper.getSysMLv2EnumerationDefinition(src |
| AddStructuralFeatureValueAssignment | ActionUsage<br>AssignmentActionUsage<br>FeatureMembership                                                                     | AddStructuralFeatureValueAction_Mapping<br>AddStructuralFeatureValueActionAssignmentAction_Mapping<br>AddStructuralFeatureValueActionAssignmentActionMembership_Mapping                                                                                                                                                                                                            |                                                                                                                                                                                     |
| AddVariableValueAction              | ActionUsage<br>FeatureTyping                                                                                                  | AddVariableValueAction_Mapping<br>AddVariableValueActionFeatureTyping_Mapping                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                     |
| BroadcastSignalAction               | ActionUsage                                                                                                                   | CommonAction_Mapping                                                                                                                                                                                                                                                                                                                                                               |                                                                                                                                                                                     |
| CallAction                          | ActionUsage                                                                                                                   | CommonAction_Mapping                                                                                                                                                                                                                                                                                                                                                               |                                                                                                                                                                                     |
| CallBehaviorAction                  | FeatureTyping<br>ActionUsage                                                                                                  | CallBehaviorFeatureTyping_Mapping<br>CallBehaviorAction_Mapping                                                                                                                                                                                                                                                                                                                    |                                                                                                                                                                                     |
| CallOperationAction                 | ActionUsage                                                                                                                   | CallOperationAction_Mapping                                                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                                                     |

| SysML v1 Concept             | SysML v2 Concept          | Mapping Class                                            | Filter |
|------------------------------|---------------------------|----------------------------------------------------------|--------|
| Clause                       | FeatureTyping             |                                                          |        |
|                              | FeatureTyping             | CommonParameterReferenceUsageInFeatureTyping_Mapping     |        |
|                              | Element                   | Mapping                                                  |        |
|                              | Feature                   | CommonReturnParameterFeatureUntyped_Mapping              |        |
|                              | FeatureTyping             | CommonReturnParameterFeatureTyping_Mapping               |        |
|                              | Relationship              | ElementOwnership_Mapping                                 |        |
|                              | Expression                | CommonValueSpecification_Mapping                         |        |
|                              | ReferenceUsage            | CommonParameterReferenceUsageInUntyped_Mapping           |        |
|                              | OwningMembership          | DefaultMultiplicityMembership_Mapping                    |        |
|                              | LiteralInteger            | CommonReturnParameterFeatureMembership_Mapping           |        |
|                              | ReturnParameterMembership | CommonParameterReferenceUsageInMembership_Mapping        |        |
|                              | ParameterMembership       | DefaultMultiplicityBoundOwnership_Mapping                |        |
|                              | FeatureMembership         | CommonReturnParameterReferenceUsageFeatureTyping_Mapping |        |
|                              | FeatureTyping             | DefaultUpperBound_Mapping                                |        |
|                              | LiteralInteger            | DefaultMultiplicityElement_Mapping                       |        |
|                              | MultiplicityRange         | CommonReturnParameterReferenceUsageUntyped_Mapping       |        |
|                              | ReferenceUsage            | DefaultLowerBound_Mapping                                |        |
|                              | LiteralInteger            | ElementMain_Mapping                                      |        |
|                              | Element                   | ElementMembership_Mapping                                |        |
|                              | Membership                | CommonReturnParameterReferenceUsageMembership_Mapping    |        |
|                              | ReturnParameterMembership | EmptyReturnParameterFeatureMembership_Mapping            |        |
|                              | ReturnParameterMembership |                                                          |        |
| ClearAssociationAction       | ActionUsage               | ClearAssociationAction_Mapping                           |        |
| ClearStructuralFeatureAction | ActionUsage               | CommonAction_Mapping                                     |        |
| ClearVariableAction          | FeatureMembership         | ClearVariableActionFeatureMembership_Mapping             |        |
|                              | ActionUsage               | ClearVariableAction_Mapping                              |        |
|                              | ReferenceUsage            | ClearVariableActionReferenceUsage_Mapping                |        |
|                              | FeatureValue              | ClearVariableActionReferenceUsageFeatureValue_Mapping    |        |
| ConditionalNode              | ActionUsage               | StructuredActivityNode_Mapping                           |        |
| CreateLinkAction             | ActionUsage               | CreateLinkAction_Mapping                                 |        |
| CreateLinkObjectAction       | ActionUsage               | CreateLinkAction_Mapping                                 |        |
| CreateObjectAction           | FeatureTyping             | CreateObjectInvocationExpressionFeatureTyping_Mapping    |        |
|                              | InvocationExpression      | CreateObjectInvocationExpression_Mapping                 |        |
|                              | ActionUsage               | CreateObjectAction_Mapping                               |        |
| DestroyLinkAction            | ActionUsage               | DestroyLinkAction_Mapping                                |        |
| DestroyObjectAction          | ActionUsage               | CommonAction_Mapping                                     |        |
| ExpansionRegion              | ActionUsage               | StructuredActivityNode_Mapping                           |        |

| SysML v1 Concept    | SysML v2 Concept                                                                                                                                                                                                                                                                                                                                                                                                    | Mapping Class                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Filter                                                                                                                                                                        |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| InputPin            | FeatureReferenceExpression<br>Feature<br>ReferenceUsage<br>FeatureTyping<br>FeatureValue<br>ParameterMembership<br>Membership                                                                                                                                                                                                                                                                                       | RemoveVariableValueActionExpressionParameterFeatureReference_Mapping<br>RemoveVariableValueActionExpressionParameter_Mapping<br>UntypedPin_Mapping<br>PinFeatureTyping_Mapping<br>RemoveVariableValueActionExpressionParameterValue_Mapping<br>RemoveVariableValueActionExpressionParameterMembership_Mapping<br>RemoveVariableValueActionExpressionParameterFeatureReferenceMembe                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | InputPin.type.oclIsUndefined()<br>not<br>src.type.oclIsUndefined()<br>and<br>not(src.type.oclIsKindOf(UML::Enumeration))<br>and<br>Helper.getSysMLv2EnumerationDefinition(src |
| InvocationAction    | ActionUsage                                                                                                                                                                                                                                                                                                                                                                                                         | CommonAction_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                                               |
| LinkAction          | ActionUsage                                                                                                                                                                                                                                                                                                                                                                                                         | CommonAction_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                                               |
| LinkEndCreationData | FeatureTyping<br>FeatureTyping<br>Element<br>Feature<br>FeatureTyping<br>Relationship<br>Expression<br>ReferenceUsage<br>OwningMembership<br>LiteralInteger<br>ReturnParameterMembership<br>ParameterMembership<br>FeatureMembership<br>FeatureTyping<br>LiteralInteger<br>MultiplicityRange<br>ReferenceUsage<br>LiteralInteger<br>Element<br>Membership<br>ReturnParameterMembership<br>ReturnParameterMembership | CommonParameterReferenceUsageInFeatureTyping_Mapping<br>Mapping<br>CommonReturnParameterFeatureUntyped_Mapping<br>CommonReturnParameterFeatureTyping_Mapping<br>ElementOwnership_Mapping<br>CommonValueSpecification_Mapping<br>CommonParameterReferenceUsageInUntyped_Mapping<br>DefaultMultiplicityMembership_Mapping<br>CommonReturnParameterFeatureMembership_Mapping<br>CommonParameterReferenceUsageInMembership_Mapping<br>DefaultMultiplicityBoundOwnership_Mapping<br>CommonReturnParameterReferenceUsageFeatureTyping_Mapping<br>DefaultUpperBound_Mapping<br>DefaultMultiplicityElement_Mapping<br>CommonReturnParameterReferenceUsageUntyped_Mapping<br>DefaultLowerBound_Mapping<br>ElementMain_Mapping<br>ElementMembership_Mapping<br>CommonReturnParameterReferenceUsageMembership_Mapping<br>EmptyReturnParameterFeatureMembership_Mapping |                                                                                                                                                                               |

| SysML v1 Concept       | SysML v2 Concept          | Mapping Class                                            | Filter |
|------------------------|---------------------------|----------------------------------------------------------|--------|
| LinkEndData            | FeatureTyping             |                                                          |        |
|                        | FeatureTyping             | CommonParameterReferenceUsageInFeatureTyping_Mapping     |        |
|                        | Element                   | Mapping                                                  |        |
|                        | Feature                   | CommonReturnParameterFeatureUntyped_Mapping              |        |
|                        | FeatureTyping             | CommonReturnParameterFeatureTyping_Mapping               |        |
|                        | Relationship              | ElementOwnership_Mapping                                 |        |
|                        | Expression                | CommonValueSpecification_Mapping                         |        |
|                        | ReferenceUsage            | CommonParameterReferenceUsageInUntyped_Mapping           |        |
|                        | OwningMembership          | DefaultMultiplicityMembership_Mapping                    |        |
|                        | LiteralInteger            | CommonReturnParameterFeatureMembership_Mapping           |        |
|                        | ReturnParameterMembership | CommonParameterReferenceUsageInMembership_Mapping        |        |
|                        | ParameterMembership       | DefaultMultiplicityBoundOwnership_Mapping                |        |
|                        | FeatureMembership         | CommonReturnParameterReferenceUsageFeatureTyping_Mapping |        |
|                        | FeatureTyping             | DefaultUpperBound_Mapping                                |        |
|                        | LiteralInteger            | DefaultMultiplicityElement_Mapping                       |        |
|                        | MultiplicityRange         | CommonReturnParameterReferenceUsageUntyped_Mapping       |        |
|                        | ReferenceUsage            | DefaultLowerBound_Mapping                                |        |
|                        | LiteralInteger            | ElementMain_Mapping                                      |        |
|                        | Element                   | ElementMembership_Mapping                                |        |
|                        | Membership                | CommonReturnParameterReferenceUsageMembership_Mapping    |        |
|                        | ReturnParameterMembership | EmptyReturnParameterFeatureMembership_Mapping            |        |
|                        | ReturnParameterMembership |                                                          |        |
| LinkEndDestructionData | FeatureTyping             |                                                          |        |
|                        | FeatureTyping             | CommonParameterReferenceUsageInFeatureTyping_Mapping     |        |
|                        | Element                   | Mapping                                                  |        |
|                        | Feature                   | CommonReturnParameterFeatureUntyped_Mapping              |        |
|                        | FeatureTyping             | CommonReturnParameterFeatureTyping_Mapping               |        |
|                        | Relationship              | ElementOwnership_Mapping                                 |        |
|                        | Expression                | CommonValueSpecification_Mapping                         |        |
|                        | ReferenceUsage            | CommonParameterReferenceUsageInUntyped_Mapping           |        |
|                        | OwningMembership          | DefaultMultiplicityMembership_Mapping                    |        |
|                        | LiteralInteger            | CommonReturnParameterFeatureMembership_Mapping           |        |
|                        | ReturnParameterMembership | CommonParameterReferenceUsageInMembership_Mapping        |        |
|                        | ParameterMembership       | DefaultMultiplicityBoundOwnership_Mapping                |        |
|                        | FeatureMembership         | CommonReturnParameterReferenceUsageFeatureTyping_Mapping |        |
|                        | FeatureTyping             | DefaultUpperBound_Mapping                                |        |
|                        | LiteralInteger            | DefaultMultiplicityElement_Mapping                       |        |
|                        | MultiplicityRange         | CommonReturnParameterReferenceUsageUntyped_Mapping       |        |
|                        | ReferenceUsage            | DefaultLowerBound_Mapping                                |        |
|                        | LiteralInteger            | ElementMain_Mapping                                      |        |
|                        | Element                   | ElementMembership_Mapping                                |        |
|                        | Membership                | CommonReturnParameterReferenceUsageMembership_Mapping    |        |
|                        | ReturnParameterMembership | EmptyReturnParameterFeatureMembership_Mapping            |        |
|                        | ReturnParameterMembership |                                                          |        |
| LoopNode               | ActionUsage               | StructuredActivityNode_Mapping                           |        |
| OpaqueAction           | ActionUsage               | OpaqueAction_Mapping                                     |        |
|                        | TextualRepresentation     | OpaqueActionBody_Mapping                                 |        |
|                        | OwningMembership          | OpaqueActionBodyMembership_Mapping                       |        |

| SysML v1 Concept     | SysML v2 Concept                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Mapping Class                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Filter                                                                                                                                                                                                                                                               |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OutputPin            | FeatureReferenceExpression<br>FeatureValue<br>FeatureMembership<br>ReferenceUsage<br>Membership<br>Feature<br>ParameterMembership<br>Membership<br>FeatureValue<br>Feature<br>FeatureChainExpression<br>ReferenceUsage<br>ReferenceUsage<br>Feature<br>ReferenceUsage<br>FeatureValue<br>OperatorExpression<br>ReferenceUsage<br>FeatureTyping<br>ReferenceUsage<br>Membership<br>ReferenceUsage<br>FeatureReferenceExpression<br>FeatureMembership<br>FeatureValue<br>FeatureMembership<br>FeatureValue<br>FeatureValue | CallOperationOutputPinFeatureReferenceExpression_Mapping<br>ReadSelfActionFeatureValue_Mapping<br>CallOperationOutputPinFeatureFeatureMembership_Mapping<br>CreateObjectPin_Mapping<br>ReadSelfActionFeatureValueFeatureReferenceExpressionMembership_Mapping<br>CallOperationOutputPinFeatureFeature_Mapping<br>CallOperationOutputPinParameterMembership_Mapping<br>CallOperationOutputPinFeatureReferenceExpressionMembership_Mapping<br>CreateObjectPinFeatureValue_Mapping<br>ReadExtentActionFeatureValueOperatorExpressionFeature_Mapping<br>CallOperationOutputPinFeatureChainExpression_Mapping<br>ReadIsClassifiedObjectActionOutputPin_Mapping<br>CallOperationOutputPin_Mapping<br>CallOperationOutputPinFeature_Mapping<br>CallOperationOutputPinReferenceUsage_Mapping<br>CallOperationOutputPinReferenceUsageFeatureValue_Mapping<br>ReadExtentActionFeatureValueOperatorExpression_Mapping<br>ReadExtentActionOutputPin_Mapping<br>ReadExtentActionFeatureValueOperatorExpressionFeatureTyping_Mapping<br>ValueSpecificationActionOutputPin_Mapping<br>CallOperationOutputPinFeatureChainExpressionMembership_Mapping<br>ReadSelfActionOutputPin_Mapping<br>ReadSelfActionFeatureValueFeatureReferenceExpression_Mapping<br>ReadExtentActionFeatureValueOperatorExpressionMembership_Mapping<br>ReadExtentActionFeatureValue_Mapping<br>CallOperationOutputPinFeatureMembership_Mapping<br>CallOperationOutputPinFeatureFeatureValue_Mapping<br>ValueSpecificationActionOutputPinFeatureValue_Mapping | OutputPin.owner.ocllsTypeOf(UML::CreateO<br>OutputPin.owner.ocllsTypeOf(UML::ReadIsC<br>OutputPin.owner.ocllsTypeOf(UML::CallOpe<br>OutputPin.owner.ocllsTypeOf(UML::ReadExt<br>OutputPin.owner.ocllsKindOf(UML::ValueSp<br>OutputPin.owner.ocllsKindOf(UML::ReadSel |
| Pin                  | FeatureReferenceExpression<br>Feature<br>ReferenceUsage<br>FeatureTyping<br>FeatureValue<br>ParameterMembership<br>Membership                                                                                                                                                                                                                                                                                                                                                                                            | RemoveVariableValueActionExpressionParameterFeatureReference_Mapp<br>RemoveVariableValueActionExpressionParameter_Mapping<br>UntypedPin_Mapping<br>PinFeatureTyping_Mapping<br>RemoveVariableValueActionExpressionParameterValue_Mapping<br>RemoveVariableValueActionExpressionParameterMembership_Mapping<br>RemoveVariableValueActionExpressionParameterFeatureReferenceMemb                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Pin.type.ocllsUndefined()<br>not<br>src.type.ocllsUndefined()<br>and<br>not(src.type.ocllsKindOf(UML::Enumeration))<br>and<br>Helper.getSysMLv2EnumerationDefinition(src                                                                                             |
| RaiseExceptionAction | ActionUsage                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | CommonAction_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                      |
| ReadExtentAction     | ActionUsage                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | ReadExtentAction_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                                      |

| SysML v1 Concept                   | SysML v2 Concept                                                                                                                                                                                                                                                                                                         | Mapping Class                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Filter |
|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| ReadIsClassifiedObjectAction       | FeatureValue<br>FeatureValue<br>Feature<br>ParameterMembership<br>FeatureReferenceExpression<br>Membership<br>ActionUsage<br>OperatorExpression                                                                                                                                                                          | ReadIsClassifiedObjectActionFeatureValue_Mapping<br>ReadIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValue_Mapping<br>ReadIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValue_Mapping<br>ReadIsClassifiedObjectActionFeatureValueOperatorExpressionParameterMembership_Mapping<br>ReadIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValue_Mapping<br>ReadIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValue_Mapping<br>ReadIsClassifiedObjectAction_Mapping<br>ReadIsClassifiedObjectActionFeatureValueOperatorExpression_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |
| ReadLinkAction                     | ActionUsage                                                                                                                                                                                                                                                                                                              | CommonAction_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |
| ReadLinkObjectEndAction            | ActionUsage                                                                                                                                                                                                                                                                                                              | CommonAction_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |
| ReadSelfAction                     | ActionUsage                                                                                                                                                                                                                                                                                                              | ReadSelfAction_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |        |
| ReadStructuralFeatureAction        | ActionUsage                                                                                                                                                                                                                                                                                                              | ReadStructuralFeatureAction_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |        |
| ReadVariableAction                 | ActionUsage                                                                                                                                                                                                                                                                                                              | ReadVariableAction_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |        |
| ReclassifyObjectAction             | ActionUsage                                                                                                                                                                                                                                                                                                              | CommonAction_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |
| ReduceAction                       | ActionUsage                                                                                                                                                                                                                                                                                                              | CommonAction_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |
| RemoveStructuralFeatureValueAction | ActionUsage                                                                                                                                                                                                                                                                                                              | CommonAction_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |
| RemoveVariableValueAction          | FeatureMembership<br>ParameterMembership<br>ReferenceUsage<br>AssignmentActionUsage<br>ParameterMembership<br>ReferenceUsage<br>FeatureValue<br>OwningMembership<br>ActionUsage<br>InvocationExpression<br>FeatureMembership<br>FeatureMembership<br>FeatureTyping<br>ReferenceUsage<br>ReferenceUsage<br>ReferenceUsage | RemoveVariableValueActionExpressionMembership_Mapping<br>RemoveVariableValueActionAssignmentActionParameterMembership_Mapping<br>RemoveVariableValueActionAssignmentActionParameterReferenceReference_Mapping<br>RemoveVariableValueActionAssignmentAction_Mapping<br>RemoveVariableValueActionAssignmentActionSecondParameterMemberships_Mapping<br>RemoveVariableValueActionExpressionReferenceUsage_Mapping<br>RemoveVariableValueActionExpressionReferenceUsageFeatureValue_Mapping<br>RemoveVariableValueActionAssignmentActionMembership_Mapping<br>RemoveVariableValueAction_Mapping<br>RemoveVariableValueActionInvocationExpression_Mapping<br>RemoveVariableValueActionAssignmentActionParameterFeatureMemberships_Mapping<br>RemoveVariableValueActionAssignmentActionParameterReferenceFeatureMemberships_Mapping<br>RemoveVariableValueActionInvocationExpressionFeatureTyping_Mapping<br>RemoveVariableValueActionAssignmentActionSecondParameter_Mapping<br>RemoveVariableValueActionAssignmentActionParameterReference_Mapping<br>RemoveVariableValueActionAssignmentActionParameter_Mapping |        |
| ReplyAction                        | ActionUsage                                                                                                                                                                                                                                                                                                              | CommonAction_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |
| SendObjectAction                   | ActionUsage                                                                                                                                                                                                                                                                                                              | CommonAction_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |
| SendSignalAction                   | ActionUsage                                                                                                                                                                                                                                                                                                              | SendSignalAction_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |        |
| SequenceNode                       | ActionUsage                                                                                                                                                                                                                                                                                                              | SequenceNode_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |
| StartClassifierBehaviorAction      | ActionUsage                                                                                                                                                                                                                                                                                                              | CommonAction_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |
| StartObjectBehaviorAction          | ActionUsage                                                                                                                                                                                                                                                                                                              | CommonAction_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |
| StructuralFeatureAction            | ActionUsage                                                                                                                                                                                                                                                                                                              | CommonAction_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |
| StructuredActivityNode             | ActionUsage                                                                                                                                                                                                                                                                                                              | StructuredActivityNode_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |        |



| SysML v1 Concept             | SysML v2 Concept                                                     | Mapping Class                                                                                                            | Filter                                                                      |
|------------------------------|----------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| TestIdentityAction           | ResultExpressionMembership<br>OperatorExpression<br>CalculationUsage | TestIdentityActionResultExpressionMembership_Mapping<br>TestIdentityActionOperator_Mapping<br>TestIdentityAction_Mapping |                                                                             |
| UnmarshallAction             | ActionUsage                                                          | CommonAction_Mapping                                                                                                     |                                                                             |
| ValuePin                     | ReferenceUsage<br>FeatureValue<br>Expression<br>ReferenceUsage       | ValuePin_Mapping<br>ValuePinFeatureValue_Mapping<br>ValuePinValue_Mapping<br>ValuePinUntyped_Mapping                     | not<br>ValuePin.type.ocllsUndefined()<br><br>ValuePin.type.ocllsUndefined() |
| ValueSpecificationAction     | ActionUsage                                                          | ValueSpecificationAction_Mapping                                                                                         |                                                                             |
| VariableAction               | ActionUsage                                                          | CommonAction_Mapping                                                                                                     |                                                                             |
| WriteLinkAction              | ActionUsage                                                          | CommonAction_Mapping                                                                                                     |                                                                             |
| WriteStructuralFeatureAction | ActionUsage                                                          | CommonAction_Mapping                                                                                                     |                                                                             |
| WriteVariableAction          | ActionUsage                                                          | CommonAction_Mapping                                                                                                     |                                                                             |

#### C.2.5.2.2 SysML v1 Activities elements not mapped

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

| SysML v1 Concept          | Rationale                                                                                                           |
|---------------------------|---------------------------------------------------------------------------------------------------------------------|
| ReclassifyObjectAction    | The SysMLv1::ReclassifyObjectAction is not supported by SysML v2. It is mapped to a action usage that does nothing. |
| StartObjectBehaviorAction | The SysMLv1::StartObjectBehaviorAction is not supported by SysML v2.                                                |

#### C.2.5.2.3 Mapping Specifications

##### C.2.5.2.3.1 Actions

##### C.2.5.2.3.1.1 CommonAction\_Mapping

##### Description

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

##### General Mappings

GenericToActionUsage\_Mapping  
NamedElementMain\_Mapping

##### Mapping Source

Action

##### Mapping Target

ActionUsage

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- ActionUsage::ownedRelationship  
`Helper.actionOwnedRelationship(from)`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Usage::isVariation  
`false`
- Feature::isReadOnly  
`false`
- Feature::direction

- `Element::elementId`

```
Helper.getID(from)
```

- ActionUsage::isComposite

true

- Element::name

null

- Feature::isDerived

false

- Feature::isComposite

false

#### C.2.5.2.3.1.2 OpaqueAction\_Mapping

### Description

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

```

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

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

```

## General Mappings

## CommonAction\_Mapping

## Mapping Source

## OpaqueAction

## Mapping Target

## ActionUsage

## Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Feature::isEnd  
`false`
- Element::ownedRelationship  
`ElementOwnership_Mapping.getMappedColl(from.ownedElement)`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::name  
`from.name`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Usage::isVariation  
`false`
- Feature::isReadOnly  
`false`
- ActionUsage::ownedRelationship  
`if from.body->size() > 0 then`

```

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

```

- Feature::direction

```

    null

```

- Element::elementId

```

    Helper.getID(from)

```

- Feature::isDerived

```

    false

```

- ActionUsage::isComposite

```

    true

```

#### C.2.5.2.3.1.3 OpaqueActionBody\_Mapping

##### Description

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

##### General Mappings

GenericToAnnotatingElement\_Mapping

##### Mapping Source

OpaqueAction

##### Mapping Target

TextualRepresentation

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId

```

    Set {}

```

- TextualRepresentation::body

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

- TextualRepresentation::language

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

- Element::name

```
null
```

- Element::shortName

```
null
```

- Element::elementId

```
Helper.createUUID()
```

- Element::ownedRelationship

```
Set{}
```

#### C.2.5.2.3.1.4 OpaqueActionBodyMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToOwningMembership\_Mapping

##### Mapping Source

OpaqueAction

##### Mapping Target

OwningMembership

##### Owned Mappings

(none)

##### Applicable filters

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

(none)

##### Mapping rules

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

- Relationship::ownedRelatedElement

```
Set{}
```

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- Membership::memberElement  
*abstract rule*
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- OwningMembership::ownedMemberElement  
  
OpaqueActionBody\_Mapping.getMapped(from)
- Membership::memberName  
  
null
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- Element::ownedRelationship  
  
Set{}

#### **C.2.5.2.3.1.5 Pin\_Mapping**

##### **Description**

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

##### **General Mappings**

UntypedPin\_Mapping

##### **Mapping Source**

Pin

##### **Mapping Target**

ReferenceUsage

## Owned Mappings

- `pinFeatureTyping : PinFeatureTyping_Mapping`

## Applicable filters

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

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

## Mapping rules

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

- `Feature::isEnd`  
`false`
- `Element::ownedRelationship`  
`ElementOwnership_Mapping.getMappedColl(from.ownedElement)`
- `Type::isSufficient`  
`false`
- `Feature::isUnique`  
`true`
- `Element::name`  
`from.name`
- `Element::shortName`  
`null`
- `Type::isAbstract`  
`false`
- `Feature::isOrdered`  
`false`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `Usage::isVariation`  
`false`
- `ReferenceUsage::ownedRelationship`



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

- Feature::isReadOnly

```
false
```

- Feature::direction

```
null
```

- Element::elementId

```
Helper.getID(from)
```

- Feature::isDerived

```
false
```

- Feature::isComposite

```
false
```

#### **C.2.5.2.3.1.6 PinFeatureTyping\_Mapping**

##### **Description**

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

##### **General Mappings**

TypedElementToFeatureTyping\_Mapping

##### **Mapping Source**

Pin

##### **Mapping Target**

FeatureTyping

##### **Owned Mappings**

- pin : Pin\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Element::aliasId

```
Set{ }
```

- Relationship::ownedRelatedElement

```

    Set {}

    • FeatureTyping::typedFeature
      pin.to

    • Element::name
      null

    • Element::shortName
      null

    • Element::elementId
      Helper.createUUID()

    • FeatureTyping::type
      abstract rule
    • FeatureTyping::typedFeature
      abstract rule
    • Element::ownedRelationship

    Set {}

```

#### C.2.5.2.3.1.7 UntypedPin\_Mapping

##### Description

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

##### General Mappings

GenericToReferenceUsage\_Mapping  
NamedElementMain\_Mapping

##### Mapping Source

Pin

##### Mapping Target

ReferenceUsage

##### Owned Mappings

(none)

##### Applicable filters

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

```
from.type.ocIsUndefined()
```

##### Mapping rules

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

- Feature::isEnd

false

- Type::isSufficient

false

- Feature::isUnique

true

- Element::shortName

null

- Type::isAbstract

false

- Feature::isOrdered

false

- Element::aliasId

Set{}

- Feature::isPortion

false

- Usage::isVariation

false

- Feature::isReadOnly

false

- ReferenceUsage::ownedRelationship

ElementOwnership\_Mapping.getMappedColl(from.ownedElement)->including(MultiplicityMembership\_

- ReferenceUsage::direction

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

- Element::elementId

Helper.getID(from)

- Element::name

null

- Feature::isDerived

false

- Feature::isComposite

false

#### **C.2.5.2.3.1.8 ValuePin\_Mapping**

##### **Description**

Mapping of UML4SysML::ValuePin with a specified type.

##### **General Mappings**

Pin\_Mapping

##### **Mapping Source**

ValuePin

##### **Mapping Target**

ReferenceUsage

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

```
from.type.ocIsUndefined()
```

##### **Mapping rules**

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

- Feature::isEnd

false

- Type::isSufficient

false

- Feature::isUnique

true

- Element::name

from.name

- Element::shortName



GenericToFeatureValue\_Mapping

### Mapping Source

ValuePin

### Mapping Target

FeatureValue

### Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
    null
- Element::shortName  
  
    null
- FeatureValue::value  
  
    ValuePinValue\_Mapping.getMapped(from)
- Element::elementId  
  
    Helper.createUUID()
- Element::aliasId  
  
    Set{ }
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
  
    null
- OwningMembership::ownedRelatedElement  
  
    Set{ self.ownedMemberElement() }
- Membership::visibility

`KerML::VisibilityKind::public`

- `Element::name`

`null`

- `Element::ownedRelationship`

`Set{}`

#### **C.2.5.2.3.1.10 ValuePinUntyped\_Mapping**

##### **Description**

Mapping of UML4SysML::ValuePin without a specified type.

##### **General Mappings**

UntypedPin\_Mapping

##### **Mapping Source**

ValuePin

##### **Mapping Target**

ReferenceUsage

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- `Feature::isEnd`

`false`

- `Element::ownedRelationship`

`ElementOwnership_Mapping.getMappedColl(from.ownedElement)`

- `Type::isSufficient`

`false`

- `Feature::isUnique`

`true`

- `Element::name`

```

    from.name

    • Element::shortName
        null

    • Type::isAbstract
        false

    • Feature::isOrdered
        false

    • Element::aliasId
        Set{}

    • Feature::isPortion
        false

    • Usage::isVariation
        false

    • Feature::isReadOnly
        false

    • Feature::direction
        null

    • Element::elementId
        Helper.getID(from)

    • Feature::isDerived
        false

    • Feature::isComposite
        false

    • ReferenceUsage::ownedRelationship
        Set{ValuePinFeatureValue_Mapping.getMapped(from), MultiplicityMembership_Mapping.getMapped(f

```

#### C.2.5.2.3.1.11 ValuePinValue\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToExpression\_Mapping



## Mapping Source

ValuePin

## Mapping Target

Expression

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Feature::isReadOnly

false

- Feature::direction

null

- Element::name

null

- Feature::isDerived

false

- Feature::isComposite

false

- Element::ownedRelationship

Set{ }

#### **C.2.5.2.3.2 Invocation Actions**

##### **C.2.5.2.3.2.1 CallBehaviorAction\_Mapping**

###### **Description**

\*\*\* not specified yet \*\*\*

###### **General Mappings**

CommonAction\_Mapping

###### **Mapping Source**

CallBehaviorAction

###### **Mapping Target**

ActionUsage

###### **Owned Mappings**

- callBehaviorFeatureTyping : CallBehaviorFeatureTyping\_Mapping

###### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

###### **Mapping rules**

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

- Feature::isEnd

false

- Element::ownedRelationship  
     ElementOwnership\_Mapping.getMappedColl(from.ownedElement)
- Type::isSufficient  
     false
- Feature::isUnique  
     true
- Element::name  
     from.name
- Element::shortName  
     null
- Type::isAbstract  
     false
- Feature::isOrdered  
     false
- Element::aliasId  
     Set{ }
- Feature::isPortion  
     false
- Usage::isVariation  
     false
- Feature::isReadOnly  
     false
- ActionUsage::ownedRelationship  
     Helper.actionOwnedRelationship(from)->append(callBehaviorFeatureTyping.to)
- Feature::direction  
     null
- Element::elementId  
     Helper.getID(from)
- Feature::isDerived  
     false

- ActionUsage::isComposite

true

#### C.2.5.2.3.2.2 CallBehaviorFeatureTyping\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureTyping\_Mapping

##### Mapping Source

CallBehaviorAction

##### Mapping Target

FeatureTyping

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId

Set{}

- Relationship::ownedRelatedElement

Set{}

- FeatureTyping::type

from.behavior

- Specialization::specific

*abstract rule*

- Element::name

null

- FeatureTyping::typedFeature

from

- Element::shortName  
null
- Specialization::general  
*abstract rule*
- Element::elementId  
Helper.createUUID()
- Element::ownedRelationship  
Set{}

#### C.2.5.2.3.2.3 CallOperationAction\_Mapping

##### Description

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

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

##### General Mappings

CommonAction\_Mapping

##### Mapping Source

CallOperationAction

##### Mapping Target

ActionUsage

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Feature::isEnd  
false

- Element::ownedRelationship  
 ElementOwnership\_Mapping.getMappedColl(from.ownedElement)
- Type::isSufficient  
 false
- Feature::isUnique  
 true
- Element::name  
 from.name
- Element::shortName  
 null
- Type::isAbstract  
 false
- Feature::isOrdered  
 false
- Element::aliasId  
 Set{}
- Feature::isPortion  
 false
- Usage::isVariation  
 false
- Feature::isReadOnly  
 false
- Feature::direction  
 null
- Element::elementId  
 Helper.getID(from)
- Feature::isDerived  
 false
- ActionUsage::isComposite  
 true

#### C.2.5.2.3.2.4 CallOperationOutputPin\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

Pin\_Mapping

##### Mapping Source

OutputPin

##### Mapping Target

ReferenceUsage

##### Owned Mappings

(none)

##### Applicable filters

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

```
from.owner.ocIsTypeOf(UML::CallOperationAction)
,
from.type.ocIsUndefined()
```

##### Mapping rules

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

- Feature::isEnd  
false
- Type::isSufficient  
false
- Feature::isUnique  
true
- Element::name  
from.name
- Element::shortName  
null
- Type::isAbstract

- false
- Feature::isOrdered
  - false
- Element::aliasId
  - Set{}
- Feature::isPortion
  - false
- Usage::isVariation
  - false
- ReferenceUsage::ownedRelationship
  - Set{CallOperationOutputPinFeatureMembership\_Mapping.getMapped(from), pinFeatureTyping.to, Mu
- Feature::isReadOnly
  - false
- ReferenceUsage::ownedRelationship
  - ElementOwnership\_Mapping.getMappedColl(from.ownedElement)->including(MultiplicityMembership\_
- ReferenceUsage::direction
  - ```

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

```
- Element::elementId
  - Helper.getID(from)
- Feature::isDerived
  - false
- Feature::isComposite
  - false

#### C.2.5.2.3.2.5 CallOperationOutputPinFeature\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeature\_Mapping



## Mapping Source

OutputPin

## Mapping Target

Feature

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Element::aliasId  
`Set{}`
- Type::isSufficient  
`false`
- Element::name  
`null`
- Feature::ownedRelationship  
`Set{CallOperationOutputPinFeatureFeatureValue_Mapping.getMapped(from), CallOperationOutputPinFeatureFeatureValue_Mapping.getMapped(to)}`
- Element::shortName  
`null`
- Feature::direction  
`KerML::FeatureDirectionKind::_in'`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`

### C.2.5.2.3.2.6 CallOperationOutputPinFeatureChainExpression\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

## General Mappings

GenericToInvocationExpression\_Mapping

## Mapping Source

OutputPin

## Mapping Target

FeatureChainExpression

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Feature::isEnd

false

- Type::isSufficient

false

- Feature::isUnique

true

- Element::shortName

null

- Type::isAbstract

false

- Element::elementId

Helper.createUUID()

- Feature::isOrdered

false

- FeatureChainExpression::ownedRelationship

Set{CallOperationOutputPinParameterMembership\_Mapping.getMapped(from), CallOperationOutputPi

- Element::aliasId

Set{}

- Feature::isPortion

false

- Feature::isReadOnly

false

- Feature::direction

null

- Element::name

null

- Feature::isDerived

false

- Feature::isComposite

false

#### **C.2.5.2.3.2.7 CallOperationOutputPinFeatureChainExpressionMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToMembership\_Mapping

##### **Mapping Source**

OutputPin

##### **Mapping Target**

Membership

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- Element::aliasId

Set{}

- Membership::memberElement

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

- Relationship::ownedRelatedElement

Set{}

- Relationship::source

Set{}

- Element::name

null

- Element::shortName

null

- Element::elementId

Helper.createUUID()

- Relationship::target

Set{}

- Element::ownedRelationship

Set{}

#### **C.2.5.2.3.2.8 CallOperationOutputPinFeatureFeature\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeature\_Mapping

##### **Mapping Source**

OutputPin

##### **Mapping Target**

Feature

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  
`Set{}`
- Type::isSufficient  
`false`
- Element::name  
`null`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set{}`

#### C.2.5.2.3.2.9 CallOperationOutputPinFeatureFeatureMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureMembership\_Mapping

##### Mapping Source

OutputPin

##### Mapping Target

FeatureMembership

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- FeatureMembership::ownedMemberFeature  
  
CallOperationOutputPinFeatureFeature\_Mapping.getMapped(from)
- Membership::memberName  
  
null
- OwningMembership::ownedRelatedElement  
  
Set{self.ownedMemberElement() }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- Element::ownedRelationship  
  
Set{}

#### C.2.5.2.3.2.10 CallOperationOutputPinFeatureFeatureValue\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

## General Mappings

GenericToFeatureValue\_Mapping

## Mapping Source

OutputPin

## Mapping Target

FeatureValue

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- FeatureValue::value  
  
CallOperationOutputPinFeatureReferenceExpression\_Mapping.getMapped(from)
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
  
null
- OwningMembership::ownedRelatedElement

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

- Membership::visibility

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

- Element::name

```
null
```

- Element::ownedRelationship

```
Set{ }
```

#### **C.2.5.2.3.2.11 CallOperationOutputPinFeatureMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureMembership\_Mapping

##### **Mapping Source**

OutputPin

##### **Mapping Target**

FeatureMembership

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- Membership::membershipOwningNamespace

*abstract rule*

- Membership::memberShortName

```
null
```

- Element::shortName

```
null
```

- Element::elementId



- Helper.createUUID()
- Element::aliasId
  - Set{}
- OwningMembership::ownedMemberElement
  - abstract rule*
- Membership::memberName
  - null
- OwningMembership::ownedRelatedElement
  - Set{self.ownedMemberElement() }
- TypeFeaturing::featureOfType
  - abstract rule*
- TypeFeaturing::featuringType
  - abstract rule*
- Membership::visibility
  - KerML::VisibilityKind::public
- Element::name
  - null
- FeatureMembership::ownedMemberFeature
  - CallOperationOutputPinReferenceUsage\_Mapping.getMapped(from)
- Element::ownedRelationship
  - Set{}

#### C.2.5.2.3.2.12 CallOperationOutputPinFeatureReferenceExpression\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureReferenceExpression\_Mapping

##### Mapping Source

OutputPin

##### Mapping Target

FeatureReferenceExpression

##### Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- FeatureReferenceExpression::ownedRelationship  
`Set{CallOperationOutputPinFeatureReferenceExpressionMembership_Mapping.getMapped(from), EmptySet}`
- Feature::isReadOnly  
`false`
- Feature::direction  
`null`
- Element::name

null

- Feature::isDerived

false

- Feature::isComposite

false

#### **C.2.5.2.3.2.13 CallOperationOutputPinFeatureReferenceExpressionMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToMembership\_Mapping

##### **Mapping Source**

OutputPin

##### **Mapping Target**

Membership

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- Element::aliasId

Set { }

- Relationship::ownedRelatedElement

Set { }

- Relationship::source

Set { }

- Membership::memberElement

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

- Element::name

null

- Element::shortName

    null

- Element::elementId

    Helper.createUUID()

- Relationship::target

    Set{}

- Element::ownedRelationship

    Set{}

#### C.2.5.2.3.2.14 CallOperationOutputPinParameterMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToParameterMembership\_Mapping

##### Mapping Source

OutputPin

##### Mapping Target

ParameterMembership

##### Owned Mappings

(none)

##### Applicable filters

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

(none)

##### Mapping rules

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

- FeatureMembership::ownedRelatedElement

    Set{self.ownedMemberFeature() }

- Membership::membershipOwningNamespace  
    *abstract rule*
- Membership::memberShortName

- `Element::shortName`

- `ParameterMembership::visibility`

- `Element::elementId`

- `Element::aliasId`

- FeatureMembership::ownedMemberFeature

- FeatureMembership::owningType

- ParameterMembership::ownedMemberParameter

- Membership::memberName

- `TypeFeaturing::featureOfType`

- `TypeFeaturing::featuringType`

- `Element::name`

- Element::ownedRelationship

#### C.2.5.2.3.2.15 CallOperationOutputPinReferenceUsage\_Mapping

\*\*\* not specified yet \*\*\*

## GenericToReferenceUsage\_Mapping

## OutputPin

**OMG Systems Modeling Language (SysML) v2.0, Submission**

ReferenceUsage

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Feature::isEnd  
`false`
- ReferenceUsage::ownedRelationship  
`Set{CallOperationOutputPinReferenceUsageFeatureValue_Mapping.getMapped(from)}`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Usage::isVariation  
`false`

- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false

#### C.2.5.2.3.2.16 CallOperationOutputPinReferenceUsageFeatureValue\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureValue\_Mapping

##### Mapping Source

OutputPin

##### Mapping Target

FeatureValue

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
null

- Element::shortName  
null
- Element::elementId  
Helper.createUUID()
- Element::aliasId  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
null
- OwningMembership::ownedRelatedElement  
Set{self.ownedMemberElement() }
- Membership::visibility  
KerML::VisibilityKind::public
- Element::name  
null
- FeatureValue::value  
CallOperationOutputPinFeatureChainExpression\_Mapping.getMapped(from)
- Element::ownedRelationship  
Set{}

#### C.2.5.2.3.2.17 SendSignalAction\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

CommonAction\_Mapping

##### Mapping Source

SendSignalAction

##### Mapping Target

ActionUsage

##### Owned Mappings



(none)

### Applicable filters

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

(none)

### Mapping rules

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

- Feature::isEnd  
`false`
- Element::ownedRelationship  
`ElementOwnership_Mapping.getMappedColl(from.ownedElement)`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::name  
`from.name`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Usage::isVariation  
`false`
- Feature::isReadOnly  
`false`

- Feature::direction  
null
- Element::elementId  
Helper.getID(from)
- Feature::isDerived  
false
- ActionUsage::isComposite  
true

### C.2.5.2.3.3 Link Actions

#### C.2.5.2.3.3.1 ClearAssociationAction\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

CommonAction\_Mapping

##### Mapping Source

ClearAssociationAction

##### Mapping Target

ActionUsage

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Feature::isEnd  
false
- Element::ownedRelationship  
ElementOwnership\_Mapping.getMappedColl(from.ownedElement)
- Type::isSufficient

- `false`
- `Feature::isUnique`  
`true`
- `Element::name`  
`from.name`
- `Element::shortName`  
`null`
- `Type::isAbstract`  
`false`
- `Feature::isOrdered`  
`false`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `Usage::isVariation`  
`false`
- `Feature::isReadOnly`  
`false`
- `Feature::direction`  
`null`
- `Element::elementId`  
`Helper.getID(from)`
- `Feature::isDerived`  
`false`
- `ActionUsage::isComposite`  
`true`

#### **C.2.5.2.3.3.2 CreateLinkAction\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

## General Mappings

CommonAction\_Mapping

## Mapping Source

CreateLinkAction

## Mapping Target

ActionUsage

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Feature::isEnd  
`false`
- Element::ownedRelationship  
`ElementOwnership_Mapping.getMappedColl(from.ownedElement)`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::name  
`from.name`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Feature::isOrdered  
`false`
- Element::aliasId

- Set{}
- Feature::isPortion
  - false
- Usage::isVariation
  - false
- Feature::isReadOnly
  - false
- Feature::direction
  - null
- Element::elementId
  - Helper.getID(from)
- Feature::isDerived
  - false
- ActionUsage::isComposite
  - true
- ActionUsage::ownedRelationship

```

let actionInputPin: Set(UML::Element) = src.ownedElement->select(e | e.ocIsTypeOf(UML::ActionInputPin))
let outputPin: Set(UML::Element) = src.ownedElement->select(e | e.ocIsTypeOf(UML::OutputPin))
let triggers: Set(UML::Element) = src.ownedElement->select(e | e.ocIsTypeOf(UML::Trigger))
let linkData: Set(UML::Element) = src.ownedElement->select(e | e.ocIsKindOf(UML::LinkEndCreation))
let toElementFMS: Set(UML::Element) = src.ownedElement->select(e | e.ocIsTypeOf(UML::InputPin))
let toElementOMS: Set(UML::Element) = (((src.ownedElement - toElementFMS) - outputPin) - actionUsage)
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(outputPin->collect(e | OutputPinMembership_Mapping.getMapped(e)))

```

#### C.2.5.2.3.3.3 DestroyLinkAction\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

CommonAction\_Mapping

##### Mapping Source

DestroyLinkAction

##### Mapping Target

ActionUsage

### Owned Mappings

(none)

### Applicable filters

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

(none)

### Mapping rules

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

- Feature::isEnd  
`false`
- Element::ownedRelationship  
`ElementOwnership_Mapping.getMappedColl(from.ownedElement)`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::name  
`from.name`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Usage::isVariation  
`false`

- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::elementId  
Helper.getID(from)
- Feature::isDerived  
false
- ActionUsage::isComposite  
true

#### **C.2.5.2.3.4 Object Actions**

##### **C.2.5.2.3.4.1 CreateObjectAction\_Mapping**

###### **Description**

\*\*\* not specified yet \*\*\*

###### **General Mappings**

CommonAction\_Mapping

###### **Mapping Source**

CreateObjectAction

###### **Mapping Target**

ActionUsage

###### **Owned Mappings**

(none)

###### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

###### **Mapping rules**

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

- Feature::isEnd  
false
- Element::ownedRelationship

```
ElementOwnership_Mapping.getMappedColl(from.ownedElement)
```

- **Type::isSufficient**  
false
- **Feature::isUnique**  
true
- **Element::name**  
from.name
- **Element::shortName**  
null
- **Type::isAbstract**  
false
- **Feature::isOrdered**  
false
- **Element::aliasId**  
Set{ }
- **Feature::isPortion**  
false
- **Usage::isVariation**  
false
- **Feature::isReadOnly**  
false
- **Feature::direction**  
null
- **Element::elementId**  
Helper.getID(from)
- **Feature::isDerived**  
false
- **ActionUsage::isComposite**  
true



#### C.2.5.2.3.4.2 CreateObjectInvocationExpressionFeatureTyping\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureTyping\_Mapping

##### Mapping Source

CreateObjectAction

##### Mapping Target

FeatureTyping

##### Owned Mappings

- createObjectInvocationExpression : CreateObjectInvocationExpression\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId  
`Set{}`
- Relationship::ownedRelatedElement  
`Set{}`
- FeatureTyping::typedFeature  
`createObjectInvocationExpression.to`
- Specialization::specific  
*abstract rule*
- FeatureTyping::type  
`from.classifier`
- Element::name  
`null`
- Element::shortName  
`null`

- Specialization::general  
*abstract rule*
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set{}`

#### C.2.5.2.3.4.3 CreateObjectInvocationExpression\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToInvocationExpression\_Mapping

##### Mapping Source

CreateObjectAction

##### Mapping Target

InvocationExpression

##### Owned Mappings

- createObjectInvocationExpressionFeatureTyping :  
CreateObjectInvocationExpressionFeatureTyping\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`

- `Type::isAbstract`  
`false`
- `Element::elementId`  
`Helper.createUUID()`
- `Feature::isOrdered`  
`false`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `InvocationExpression::ownedRelationship`  
`Set{createObjectInvocationExpressionFeatureTyping.to, CommonReturnParameterFeatureMembership_`
- `Feature::isReadOnly`  
`false`
- `Feature::direction`  
`null`
- `Element::name`  
`null`
- `Feature::isDerived`  
`false`
- `Feature::isComposite`  
`false`

#### **C.2.5.2.3.4.4 CreateObjectPin\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

Pin\_Mapping

##### **Mapping Source**

OutputPin

##### **Mapping Target**

ReferenceUsage

## Owned Mappings

(none)

## Applicable filters

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

```
from.owner.ocIsTypeOf(UML::CreateObjectAction)
```

```
,
```

```
from.type.ocIsUndefined()
```

## Mapping rules

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

- Feature::isEnd  
false
- Type::isSufficient  
false
- ReferenceUsage::ownedRelationship  
Set{pinFeatureTyping.to, CreateObjectPinFeatureValue\_Mapping.getMapped(from)}
- Feature::isUnique  
true
- Element::name  
from.name
- Element::shortName  
null
- Type::isAbstract  
false
- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false

- Usage::isVariation

false

- Feature::isReadOnly

false

- ReferenceUsage::ownedRelationship

ElementOwnership\_Mapping.getMappedColl(from.ownedElement)->including(MultiplicityMembership\_

- ReferenceUsage::direction

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

- Element::elementId

Helper.getID(from)

- Feature::isDerived

false

- Feature::isComposite

false

#### C.2.5.2.3.4.5 CreateObjectPinFeatureValue\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureValue\_Mapping

##### Mapping Source

OutputPin

##### Mapping Target

FeatureValue

##### Owned Mappings

(none)

##### Applicable filters

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

(none)

## Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- FeatureValue::value  
  
CreateObjectInvocationExpression\_Mapping.getMapped(from.owner)
- Membership::memberName  
  
null
- OwningMembership::ownedRelatedElement  
  
Set{self.ownedMemberElement() }
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- Element::ownedRelationship  
  
Set{ }

### C.2.5.2.3.4.6 CreateObjectPinMembership\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

Mapping Class Specification Description

### Mapping Source

No source element.

### Mapping Target

No target element.

### Owned Mappings

(none)

#### C.2.5.2.3.4.7 ReadIsClassifiedObjectAction\_Mapping

### Description

\*\*\* not specified yet \*\*\*

### General Mappings

CommonAction\_Mapping

### Mapping Source

ReadIsClassifiedObjectAction

### Mapping Target

ActionUsage

### Owned Mappings

(none)

### Applicable filters

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

(none)

### Mapping rules

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

- Feature::isEnd

false

- Element::ownedRelationship

ElementOwnership\_Mapping.getMappedColl(from.ownedElement)

- Type::isSufficient

false

- Feature::isUnique

true

- Element::name  
from.name
- Element::shortName  
null
- Type::isAbstract  
false
- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Usage::isVariation  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::elementId  
Helper.getID(from)
- Feature::isDerived  
false
- ActionUsage::isComposite  
true

#### **C.2.5.2.3.4.8 ReadIsClassifiedObjectActionFeatureValue\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureValue\_Mapping

##### **Mapping Source**



ReadIsClassifiedObjectAction

## Mapping Target

FeatureValue

## Owned Mappings

- readIsClassifiedObjectActionFeatureValueOperatorExpression :  
ReadIsClassifiedObjectActionFeatureValueOperatorExpression\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- FeatureValue::value  
  
readIsClassifiedObjectActionFeatureValueOperatorExpression.to
- Element::aliasId  
  
Set{ }
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
  
null
- OwningMembership::ownedRelatedElement  
  
Set{ self.ownedMemberElement() }
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name

null

- Element::ownedRelationship

Set{}

#### **C.2.5.2.3.4.9 ReadIsClassifiedObjectActionFeatureValueOperatorExpression\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToExpression\_Mapping

##### **Mapping Source**

ReadIsClassifiedObjectAction

##### **Mapping Target**

OperatorExpression

##### **Owned Mappings**

- readIsClassifiedObjectActionFeatureValueOperatorExpressionParameterMembership :  
ReadIsClassifiedObjectActionFeatureValueOperatorExpressionParameterMembership\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Feature::isEnd

false

- Type::isSufficient

false

- OperatorExpression::operator

if from.isDirect then 'istype' else 'hastype' endif

- Feature::isUnique

true

- Element::shortName

null

- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Feature::isReadOnly  
false
- OperatorExpression::ownedRelationship  
Set{readIsClassifiedObjectActionFeatureValueOperatorExpressionParameterMembership.to}
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false

#### **C.2.5.2.3.4.10 ReadIsClassifiedObjectActionFeatureValueOperatorExpressionFeature\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeature\_Mapping

##### **Mapping Source**

ReadIsClassifiedObjectAction

##### **Mapping Target**

Feature

### Owned Mappings

- `readIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValue :`  
`ReadIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValue_Mapping`

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- `Element::aliasId`  
`Set {}`
- `Type::isSufficient`  
`false`
- `Element::name`  
`null`
- `Feature::direction`  
`KerML::FeatureDirectionKind::_in'`
- `Element::shortName`  
`null`
- `Type::isAbstract`  
`false`
- `Feature::ownedRelationship`  
`Set {readIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValue.to}`
- `Element::elementId`  
`Helper.createUUID()`

#### C.2.5.2.3.4.11 ReadIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValue\_Mapping

### Description

\*\*\* not specified yet \*\*\*

### General Mappings

GenericToFeatureValue\_Mapping

## Mapping Source

ReadIsClassifiedObjectAction

## Mapping Target

FeatureValue

## Owned Mappings

- readIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValueExpression :  
ReadIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValueExpression\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{ }
- OwningMembership::ownedMemberElement  
*abstract rule*
- FeatureValue::value  
  
readIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValueExpression.to
- Membership::memberName  
  
null
- OwningMembership::ownedRelatedElement  
  
Set{self.ownedMemberElement() }
- Membership::visibility  
  
KerML::VisibilityKind::public

- Element::name  
null
- Element::ownedRelationship  
Set{ }

#### C.2.5.2.3.4.12

#### ReadIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValueExpression\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToExpression\_Mapping

##### Mapping Source

ReadIsClassifiedObjectAction

##### Mapping Target

FeatureReferenceExpression

##### Owned Mappings

- readIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValueExpressionMembership :  
ReadIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValueExpressionMembership\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Feature::isEnd  
false
- Type::isSufficient  
false
- FeatureReferenceExpression::ownedRelationship  
Set{ readIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValueExpressionMembership
- Feature::isUnique  
true

- Element::shortName  
null
- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false

#### **C.2.5.2.3.4.13**

#### **ReadIsClassifiedObjectActionFeatureValueOperatorExpressionFeatureValueExpressionMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToMembership\_Mapping

##### **Mapping Source**

ReadIsClassifiedObjectAction

## Mapping Target

Membership

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Element::aliasId  
`Set {}`
- Relationship::ownedRelatedElement  
`Set {}`
- Relationship::source  
`Set {}`
- Element::name  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Relationship::target  
`Set {}`
- Element::ownedRelationship  
`Set {}`

### C.2.5.2.3.4.14

#### ReadIsClassifiedObjectActionFeatureValueOperatorExpressionParameterMembership\_Mapping

## Description

\*\*\* not specified yet \*\*\*

## General Mappings



GenericToParameterMembership\_Mapping

### Mapping Source

ReadIsClassifiedObjectAction

### Mapping Target

ParameterMembership

### Owned Mappings

- readIsClassifiedObjectActionFeatureValueOperatorExpressionFeature :  
ReadIsClassifiedObjectActionFeatureValueOperatorExpressionFeature\_Mapping

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- FeatureMembership::ownedRelatedElement  
`Set{self.ownedMemberFeature() }`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- ParameterMembership::visibility  
`KerML::VisibilityKind::private`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{ }`
- FeatureMembership::ownedMemberFeature  
*abstract rule*
- FeatureMembership::owningType  
*abstract rule*
- Membership::memberName  
`null`

- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- ParameterMembership::ownedMemberParameter  
  
readIsClassifiedObjectActionFeatureValueOperatorExpressionFeature.to
- Element::name  
  
null
- Element::ownedRelationship  
  
Set { }

#### C.2.5.2.3.4.15 ReadIsClassifiedObjectActionOutputPin\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

Pin\_Mapping

##### Mapping Source

OutputPin

##### Mapping Target

ReferenceUsage

##### Owned Mappings

(none)

##### Applicable filters

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

```
from.type.ocIsUndefined()
,
from.owner.ocIsTypeOf(UML::ReadIsClassifiedObjectAction)
```

##### Mapping rules

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

- Feature::isEnd  
  
false
- Type::isSufficient

```

false

```

- Feature::isUnique

```

true

```
- Element::name

```

from.name

```
- Element::shortName

```

null

```
- Type::isAbstract

```

false

```
- Feature::isOrdered

```

false

```
- Element::aliasId

```

Set{}

```
- Feature::isPortion

```

false

```
- Usage::isVariation

```

false

```
- Feature::isReadOnly

```

false

```
- ReferenceUsage::ownedRelationship

```

ElementOwnership_Mapping.getMappedColl(from.ownedElement)->including(MultiplicityMembership_

```
- ReferenceUsage::direction

```

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

```
- Element::elementId

```

Helper.getID(from)

```
- ReferenceUsage::ownedRelationship

```

Set{pinFeatureTyping.to, ReadIsClassifiedObjectActionFeatureValue_Mapping.getMapped(from.own

```
- Feature::isDerived

```

false

```

- Feature::isComposite

false

#### **C.2.5.2.3.4.16 ReadExtentAction\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

CommonAction\_Mapping

##### **Mapping Source**

ReadExtentAction

##### **Mapping Target**

ActionUsage

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- Feature::isEnd

false

- Element::ownedRelationship

ElementOwnership\_Mapping.getMappedColl(from.ownedElement)

- Type::isSufficient

false

- Feature::isUnique

true

- Element::name

from.name

- Element::shortName

null

- `Type::isAbstract`  
`false`
- `Feature::isOrdered`  
`false`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `Usage::isVariation`  
`false`
- `Feature::isReadOnly`  
`false`
- `Feature::direction`  
`null`
- `Element::elementId`  
`Helper.getID(from)`
- `Feature::isDerived`  
`false`
- `ActionUsage::isComposite`  
`true`
- `ActionUsage::ownedRelationship`  
`Helper.actionOwnedRelationship(from)`

#### **C.2.5.2.3.4.17 ReadExtentActionFeatureValue\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureValue\_Mapping

##### **Mapping Source**

OutputPin

##### **Mapping Target**

FeatureValue

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
    null
- Element::shortName  
  
    null
- Element::elementId  
  
    Helper.createUUID()
- FeatureValue::value  
  
    ReadExtentActionFeatureValueOperatorExpression\_Mapping.getMapped(from)
- Element::aliasId  
  
    Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
  
    null
- OwningMembership::ownedRelatedElement  
  
    Set{self.ownedMemberElement() }
- Membership::visibility  
  
    KerML::VisibilityKind::public
- Element::name  
  
    null
- Element::ownedRelationship  
  
    Set{}

#### C.2.5.2.3.4.18 ReadExtentActionFeatureValueOperatorExpression\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToExpression\_Mapping

##### Mapping Source

OutputPin

##### Mapping Target

OperatorExpression

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- OperatorExpression::operator  
`'all'`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`

- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Feature::isReadOnly  
false
- OperatorExpression::ownedRelationship  
Set{ReadExtentActionFeatureValueOperatorExpressionMembership\_Mapping.getMapped(from), Common
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false

#### C.2.5.2.3.4.19 ReadExtentActionFeatureValueOperatorExpressionFeature\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeature\_Mapping

##### Mapping Source

OutputPin

##### Mapping Target

Feature

##### Owned Mappings

- readExtentActionFeatureValueOperatorExpressionFeatureTyping :  
ReadExtentActionFeatureValueOperatorExpressionFeatureTyping\_Mapping



### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  
`Set{ }`
- Type::isSufficient  
`false`
- Element::name  
`null`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Feature::ownedRelationship  
`Set{ readExtentActionFeatureValueOperatorExpressionFeatureTyping.to }`
- Element::elementId  
`Helper.createUUID()`

#### C.2.5.2.3.4.20 ReadExtentActionFeatureValueOperatorExpressionFeatureTyping\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureTyping\_Mapping

##### Mapping Source

OutputPin

##### Mapping Target

FeatureTyping

## Owned Mappings

- readExtentActionFeatureValueOperatorExpressionFeature :  
ReadExtentActionFeatureValueOperatorExpressionFeature\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Element::aliasId  
`Set{}`
- Relationship::ownedRelatedElement  
`Set{}`
- Specialization::specific  
*abstract rule*
- FeatureTyping::typedFeature  
`readExtentActionFeatureValueOperatorExpressionFeature.to`
- Element::name  
`null`
- Element::shortName  
`null`
- Specialization::general  
*abstract rule*
- Element::elementId  
`Helper.createUUID()`
- FeatureTyping::type  
`from.owner.classifier`
- Element::ownedRelationship  
`Set{}`

### C.2.5.2.3.4.21 ReadExtentActionFeatureValueOperatorExpressionMembership\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

GenericToFeatureMembership\_Mapping

### Mapping Source

OutputPin

### Mapping Target

FeatureMembership

### Owned Mappings

- readExtentActionFeatureValueOperatorExpressionFeature :  
ReadExtentActionFeatureValueOperatorExpressionFeature\_Mapping

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{ }
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
  
null
- OwningMembership::ownedRelatedElement  
  
Set{ self.ownedMemberElement() }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility

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

- Element::name

```
null
```

- FeatureMembership::ownedMemberFeature

```
readExtentActionFeatureValueOperatorExpressionFeature
```

- Element::ownedRelationship

```
Set{}
```

#### **C.2.5.2.3.4.22 ReadExtentActionOutputPin\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

Pin\_Mapping

##### **Mapping Source**

OutputPin

##### **Mapping Target**

ReferenceUsage

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

```
from.owner.ocIsTypeOf(UML::ReadExtentAction)
```

```
,
```

```
from.type.ocIsUndefined()
```

##### **Mapping rules**

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

- Feature::isEnd

```
false
```

- Type::isSufficient

```
false
```

- Feature::isUnique  
true
- Element::name  
from.name
- Element::shortName  
null
- Type::isAbstract  
false
- Feature::isOrdered  
false
- ReferenceUsage::ownedRelationship  
Set{pinFeatureTyping.to, ReadExtentActionFeatureValue\_Mapping.getMapped(from), MultiplicityM
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Usage::isVariation  
false
- Feature::isReadOnly  
false
- ReferenceUsage::ownedRelationship  
ElementOwnership\_Mapping.getMappedColl(from.ownedElement)->including(MultiplicityMembership
- ReferenceUsage::direction  
  
if src.oclIsTypeOf(UML::InputPin) then KerML::FeatureDirectionKind::\_'in'  
else if src.oclIsTypeOf(UML::OutputPin) then KerML::FeatureDirectionKind::\_'out'  
else OclUndefined endif endif
- Element::elementId  
Helper.getID(from)
- Feature::isDerived  
false
- Feature::isComposite

false

#### C.2.5.2.3.4.23 ReadSelfAction\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

CommonAction\_Mapping

##### Mapping Source

ReadSelfAction

##### Mapping Target

ActionUsage

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Feature::isEnd

false

- Element::ownedRelationship

ElementOwnership\_Mapping.getMappedColl(from.ownedElement)

- Type::isSufficient

false

- Feature::isUnique

true

- Element::name

from.name

- Element::shortName

null

- Type::isAbstract

- false
- Feature::isOrdered
 

false
- Element::aliasId
 

Set{}
- Feature::isPortion
 

false
- Usage::isVariation
 

false
- Feature::isReadOnly
 

false
- Feature::direction
 

null
- Element::elementId
 

Helper.getID(from)
- Feature::isDerived
 

false
- ActionUsage::isComposite
 

true

#### **C.2.5.2.3.4.24 ReadSelfActionFeatureValue\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureValue\_Mapping

##### **Mapping Source**

OutputPin

##### **Mapping Target**

FeatureValue

##### **Owned Mappings**

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
  
null
- OwningMembership::ownedRelatedElement  
  
Set{self.ownedMemberElement() }
- FeatureValue::value  
  
ReadSelfActionFeatureValueFeatureReferenceExpression\_Mapping.getMapped(from)
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- Element::ownedRelationship  
  
Set{}

#### C.2.5.2.3.4.25 ReadSelfActionFeatureValueFeatureReferenceExpression\_Mapping

##### Description

\*\*\* not specified yet \*\*\*



## General Mappings

GenericToFeatureReferenceExpression\_Mapping

## Mapping Source

OutputPin

## Mapping Target

FeatureReferenceExpression

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion

false

- Feature::isReadOnly

false

- Feature::direction

null

- Element::name

null

- Feature::isDerived

false

- Feature::isComposite

false

- FeatureReferenceExpression::ownedRelationship

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

#### **C.2.5.2.3.4.26 ReadSelfActionFeatureValueFeatureReferenceExpressionMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToMembership\_Mapping

##### **Mapping Source**

OutputPin

##### **Mapping Target**

Membership

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- `Element::aliasId`  
`Set{}`
- `Relationship::ownedRelatedElement`  
`Set{}`
- `Relationship::source`  
`Set{}`
- `Membership::memberElement`  
`SYSMML2::Feature.allInstances()->any(e | e.qualifiedName = 'Occurrences::Occurrence::this')`
- `Element::name`  
`null`
- `Element::shortName`  
`null`
- `Element::elementId`  
`Helper.createUUID()`
- `Relationship::target`  
`Set{}`
- `Element::ownedRelationship`  
`Set{}`

#### **C.2.5.2.3.4.27 ReadSelfActionOutputPin\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

Pin\_Mapping

##### **Mapping Source**

OutputPin

##### **Mapping Target**

ReferenceUsage

##### **Owned Mappings**

(none)

### Applicable filters

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

```
from.owner.ocIsKindOf(UML::ReadSelfAction)
```

,

```
from.type.ocIsUndefined()
```

### Mapping rules

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

- Feature::isEnd

```
false
```

- Type::isSufficient

```
false
```

- Element::name

```
from.name
```

- Element::shortName

```
null
```

- Feature::isOrdered

```
false
```

- Element::aliasId

```
Set{}
```

- Feature::isPortion

```
false
```

- Usage::isVariation

```
false
```

- Feature::isReadOnly

```
false
```

- ReferenceUsage::ownedRelationship

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

- ReferenceUsage::direction

```

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

```

- ReferenceUsage::isUnique

```
false
```

- ReferenceUsage::ownedRelationship

```
Set{pinFeatureTyping.to, ReadSelfActionFeatureValue_Mapping.getMapped(from), MultiplicityMen
```

- Element::elementId

```
Helper.getID(from)
```

- ReferenceUsage::isAbstract

```
true
```

- Feature::isDerived

```
false
```

- Feature::isComposite

```
false
```

#### C.2.5.2.3.4.28 TestIdentityAction\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

CommonAction\_Mapping

##### Mapping Source

TestIdentityAction

##### Mapping Target

CalculationUsage

##### Owned Mappings

(none)

##### Applicable filters

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

(none)

##### Mapping rules

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

- `Feature::isEnd`  
`false`
- `Element::ownedRelationship`  
`ElementOwnership_Mapping.getMappedColl(from.ownedElement)`
- `Type::isSufficient`  
`false`
- `Feature::isUnique`  
`true`
- `Element::name`  
`from.name`
- `Element::shortName`  
`null`
- `Type::isAbstract`  
`false`
- `Feature::isOrdered`  
`false`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `Usage::isVariation`  
`false`
- `Feature::isReadOnly`  
`false`
- `CalculationUsage::ownedRelationship`  
`Helper.actionOwnedRelationship(from)`  
`->including(TestIdentityActionResultExpressionMembership_Mapping.getMapped(from))`
- `Feature::direction`  
`null`
- `Element::elementId`

```
Helper.getID(from)
```

- Feature::isDerived

```
false
```

- ActionUsage::isComposite

```
true
```

#### C.2.5.2.3.4.29 EqualOperatorExpressionOperand\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToParameterMembership\_Mapping

##### Mapping Source

TypedElement

##### Mapping Target

ParameterMembership

##### Owned Mappings

(none)

##### Applicable filters

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

(none)

##### Mapping rules

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

- FeatureMembership::ownedRelatedElement

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

- Membership::membershipOwningNamespace

*abstract rule*

- Membership::memberShortName

```
null
```

- ParameterMembership::ownedMemberParameter

```
EqualOperatorExpressionFeature_Mapping.getMapped(from)
```

- Element::shortName

- `ParameterMembership::visibility`  
`KerML::VisibilityKind::private`
- `Element::elementId`  
`Helper.createUUID()`
- `Element::aliasId`  
`Set{}`
- `FeatureMembership::ownedMemberFeature`  
*abstract rule*
- `FeatureMembership::owningType`  
*abstract rule*
- `Membership::memberName`  
`null`
- `TypeFeaturing::featureOfType`  
*abstract rule*
- `TypeFeaturing::featuringType`  
*abstract rule*
- `Element::name`  
`null`
- `Element::ownedRelationship`  
`Set{}`

### Description

## General Mappings

## Mapping Source

## Mapping Target

## Owned Mappings

### Applicable filters



This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- FeatureReferenceExpression::ownedRelationship  
`Set{CommonMembership_Mapping.getMapped(from), CommonReturnParameterFeatureMembership_Mapping}`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Feature::isReadOnly  
`false`
- Feature::direction  
`null`
- Element::name  
`null`

- Feature::isDerived

false

- Feature::isComposite

false

#### **C.2.5.2.3.4.31 CommonReferenceUsageIn\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

CommonReferenceUsageInUntyped\_Mapping

##### **Mapping Source**

TypedElement

##### **Mapping Target**

ReferenceUsage

##### **Owned Mappings**

- commonReferenceUsageInFeatureTyping : CommonReferenceUsageInFeatureTyping\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Feature::isEnd

false

- Type::isSufficient

false

- ReferenceUsage::ownedRelationship

Set{commonReferenceUsageInFeatureTyping.to}

- Feature::isUnique

true

- Element::shortName

null

- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Usage::isVariation  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false

#### **C.2.5.2.3.4.32 CommonReferenceUsageInFeatureMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureMembership\_Mapping

##### **Mapping Source**

TypedElement

##### **Mapping Target**

FeatureMembership

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- FeatureMembership::ownedMemberFeature

```
if from.type.ocIsUndefined() then CommonReferenceUsageInUntyped_Mapping.getMapped(from) els
```

- Membership::membershipOwningNamespace

*abstract rule*

- Membership::memberShortName

```
null
```

- Element::shortName

```
null
```

- Element::elementId

```
Helper.createUUID()
```

- Element::aliasId

```
Set{}
```

- OwningMembership::ownedMemberElement

*abstract rule*

- Membership::memberName

```
null
```

- OwningMembership::ownedRelatedElement

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

- TypeFeaturing::featureOfType

*abstract rule*

- TypeFeaturing::featuringType

*abstract rule*

- Membership::visibility

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

- Element::name

```
null
```

- Element::ownedRelationship

Set {}

#### C.2.5.2.3.4.33 TestIdentityActionOperator\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToExpression\_Mapping

##### Mapping Source

TestIdentityAction

##### Mapping Target

OperatorExpression

##### Owned Mappings

(none)

##### Applicable filters

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

(none)

##### Mapping rules

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

- Feature::isEnd

false

- Type::isSufficient

false

- OperatorExpression::operator

'=='

- OperatorExpression::ownedRelationship

Set {EqualOperatorExpressionOperand\_Mapping.getMapped(from.first), EqualOperatorExpressionOperand\_Mapping.getMapped(from.second)}

- Feature::isUnique

true

- Element::shortName

null

- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false

#### **C.2.5.2.3.4.34 EqualOperatorExpressionFeature\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeature\_Mapping

##### **Mapping Source**

TypedElement

##### **Mapping Target**

Feature

##### **Owned Mappings**

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  
`Set { }`
- Type::isSufficient  
`false`
- Element::name  
`null`
- Feature::ownedRelationship  
`Set { EqualOperatorExpressionFeatureValue_Mapping.getMapped (from) }`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID ( )`

#### C.2.5.2.3.4.35 TestIdentityActionResultExpressionMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureMembership\_Mapping

##### Mapping Source

TestIdentityAction

##### Mapping Target

ResultExpressionMembership

##### Owned Mappings

(none)

### Applicable filters

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

(none)

### Mapping rules

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

- ResultExpressionMembership::ownedMemberFeature  
`TestIdentityActionOperator_Mapping.getMapped(from)`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{}`
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
`null`
- OwningMembership::ownedRelatedElement  
`Set{self.ownedMemberElement() }`
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
`KerML::VisibilityKind::public`
- Element::name  
`null`
- Element::ownedRelationship  
`Set{}`



#### C.2.5.2.3.4.36 ValueSpecificationAction\_Mapping

##### Description

The expected SysML v2 textual notation of a SysMLv1::ValueSpecificationAction is as follows:

```
action thisIsAValueSpecificationAction {
  out result : ScalarValues::Integer = 42;
}

action thisIsAnotherValueSpecificationAction {
  out result = thisIsAnOpaqueExpression.result;
  calc thisIsAnOpaqueExpression {
    language "Math"
    /*
     * 42 + 23
     */
  }
}
```

##### General Mappings

CommonAction\_Mapping

##### Mapping Source

ValueSpecificationAction

##### Mapping Target

ActionUsage

##### Owned Mappings

(none)

##### Applicable filters

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

(none)

##### Mapping rules

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

- Feature::isEnd  
false
- Element::ownedRelationship  
ElementOwnership\_Mapping.getMappedColl(from.ownedElement)
- Type::isSufficient  
false

- ActionUsage::ownedRelationship

```
let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::Pin))
let toElementOMS: Set(UML::Element) = (from.ownedElement - toElementFMS) in
toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e))
->union(toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e)))
```

- Feature::isUnique

```
true
```

- Element::name

```
from.name
```

- Element::shortName

```
null
```

- Type::isAbstract

```
false
```

- Feature::isOrdered

```
false
```

- Element::aliasId

```
Set{}
```

- Feature::isPortion

```
false
```

- Usage::isVariation

```
false
```

- Feature::isReadOnly

```
false
```

- Feature::direction

```
null
```

- Element::elementId

```
Helper.getID(from)
```

- Feature::isDerived

```
false
```

- ActionUsage::isComposite

```
true
```

#### C.2.5.2.3.4.37 ValueSpecificationActionOutputPin\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

Pin\_Mapping

##### Mapping Source

OutputPin

##### Mapping Target

ReferenceUsage

##### Owned Mappings

(none)

##### Applicable filters

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

```
from.owner.ocIsKindOf(UML::ValueSpecificationAction)
,
from.type.ocIsUndefined()
```

##### Mapping rules

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

- Feature::isEnd  
false
- Type::isSufficient  
false
- ReferenceUsage::ownedRelationship  
Set{pinFeatureTyping.to, ValueSpecificationActionOutputPinFeatureValue\_Mapping.getMapped(from)}
- Feature::isUnique  
true
- Element::name  
from.name
- Element::shortName

- `Type::isAbstract`

false

- Feature::isOrdered

false

- `Element::aliasId`

Set { }

- `Feature::isPortion`

false

- Usage::isVariation

false

- `Feature::isReadOnly`

false

- ReferenceUsage::ownedRelationship

```
ElementOwnership Mapping.getMappedColl(from.ownedElement)->including(MultiplicityMembership
```

- ReferenceUsage::direction

```

if src.oclIsTypeOf(UML::InputPin) then KerML::FeatureDirectionKind::_in'
else if src.oclIsTypeOf(UML::OutputPin) then KerML::FeatureDirectionKind::_out'
else OclUndefined endif endif

```

- `Element::elementId`

```
Helper.getID(from)
```

- Feature::isDerived

false

- Feature::isComposite

false

#### C.2.5.2.3.4.38 ValueSpecificationActionOutputPinFeatureValue\_Mapping

### Description

\*\*\* not specified yet \*\*\*

## General Mappings

## GenericToFeatureValue\_Mapping

## Mapping Source

OutputPin

## Mapping Target

FeatureValue

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Membership::membershipOwningNamespace

*abstract rule*

- Membership::memberShortName

null

- Element::shortName

null

- FeatureValue::value

```
if from.owner.value.ocIsTypeOf(UML::OpaqueExpression) then OpaqueExpressionAsValue_Mapping.
```

- Element::elementId

```
Helper.createUUID()
```

- Element::aliasId

```
Set{}
```

- OwningMembership::ownedMemberElement

*abstract rule*

- Membership::memberName

null

- OwningMembership::ownedRelatedElement

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

- Membership::visibility

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

- Element::name

null

- Element::ownedRelationship

Set{}

#### **C.2.5.2.3.5 Structural Feature Actions**

##### **C.2.5.2.3.5.1 AddStructuralFeatureValueAction\_Mapping**

###### **Description**

\*\*\* not specified yet \*\*\*

###### **General Mappings**

CommonAction\_Mapping

###### **Mapping Source**

AddStructuralFeatureValueAction

###### **Mapping Target**

ActionUsage

###### **Owned Mappings**

- addStructuralFeatureValueActionAssignActionMembership :  
AddStructuralFeatureValueActionAssignmentActionMembership\_Mapping

###### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

###### **Mapping rules**

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

- Feature::isEnd

false

- Element::ownedRelationship

ElementOwnership\_Mapping.getMappedColl(from.ownedElement)

- Type::isSufficient

false

- ActionUsage::ownedRelationship

Helper.actionOwnedRelationship(from)  
->including(addStructuralFeatureValueActionAssignActionMembership.to)

- Feature::isUnique

- `Element::name`  
`from.name`
- `Element::shortName`  
`null`
- `Type::isAbstract`  
`false`
- `Feature::isOrdered`  
`false`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `Usage::isVariation`  
`false`
- `Feature::isReadOnly`  
`false`
- `Feature::direction`  
`null`
- `Element::elementId`  
`Helper.getID(from)`
- `Feature::isDerived`  
`false`
- `ActionUsage::isComposite`  
`true`

### Description

## General Mappings

OMG Systems Modeling Language (SysML) v2.0, Submission

## Mapping Source

AddStructuralFeatureValueAction

## Mapping Target

AssignmentActionUsage

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Usage::isVariation



false

- Feature::isReadOnly

false

- Feature::direction

null

- Element::name

null

- Feature::isDerived

false

- ActionUsage::isComposite

true

- Element::ownedRelationship

Set{}

#### **C.2.5.2.3.5.3 AddStructuralFeatureValueActionAssignmentActionMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureMembership\_Mapping

##### **Mapping Source**

AddStructuralFeatureValueAction

##### **Mapping Target**

FeatureMembership

##### **Owned Mappings**

- addStructuralFeatureValueActionAssignmentAction :  
AddStructuralFeatureValueActionAssignmentAction\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
  
null
- OwningMembership::ownedRelatedElement  
  
Set{self.ownedMemberElement() }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- FeatureMembership::ownedMemberFeature  
  
addStructuralFeatureValueActionAssignmentAction.to
- FeatureMembership::memberFeature  
  
self.ownedMemberFeature()
- Element::ownedRelationship  
  
Set{}

#### C.2.5.2.3.5.4 ReadStructuralFeatureAction\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

## General Mappings

CommonAction\_Mapping

## Mapping Source

ReadStructuralFeatureAction

## Mapping Target

ActionUsage

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Feature::isEnd  
`false`
- Element::ownedRelationship  
`ElementOwnership_Mapping.getMappedColl(from.ownedElement)`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::name  
`from.name`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Feature::isOrdered  
`false`
- Element::aliasId

Set{}

- Feature::isPortion  
false
- Usage::isVariation  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::elementId  
Helper.getID(from)
- Feature::isDerived  
false
- ActionUsage::isComposite  
true

#### **C.2.5.2.3.6 Structured Actions**

##### **C.2.5.2.3.6.1 SequenceNode\_Mapping**

###### **Description**

\*\*\* not specified yet \*\*\*

###### **General Mappings**

CommonAction\_Mapping  
StructuredActivityNode\_Mapping

###### **Mapping Source**

SequenceNode

###### **Mapping Target**

ActionUsage

###### **Owned Mappings**

(none)

###### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- ActionUsage::ownedRelationship  
`Helper.actionOwnedRelationship(from)`
- Element::name  
`from.name`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Usage::isVariation  
`false`
- Namespace::ownedImport  
`Set{}`
- Feature::isReadOnly  
`false`

- Feature::direction  
null
- Element::elementId  
Helper.getID(from)
- ActionUsage::isComposite  
true
- Feature::isDerived  
false
- Namespace::ownedRelationship  
from.ownedElement->collect(e | ElementOwningMembership\_Mapping.getMapped(e))

#### **C.2.5.2.3.6.2 StructuredActivityNode\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

Namespace\_Mapping  
CommonAction\_Mapping

##### **Mapping Source**

StructuredActivityNode

##### **Mapping Target**

ActionUsage

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Feature::isEnd  
false
- Element::ownedRelationship

```
ElementOwnership_Mapping.getMappedColl(from.ownedElement)
```

- **ActionUsage::ownedRelationship**

```
let valuePin: Set(UML::Element) = src.ownedElement->select(e | e.ocIsTypeOf(UML::ValuePin))
let inputPin: Set(UML::Element) = src.ownedElement->select(e | e.ocIsTypeOf(UML::InputPin))
let actionInputPin: Set(UML::Element) = src.ownedElement->select(e | e.ocIsTypeOf(UML::ActionInputPin))
let outputPin: Set(UML::Element) = src.ownedElement->select(e | e.ocIsTypeOf(UML::OutputPin))
let toInitialNode: Set(UML::Element) = src.ownedElement->select(e | e.ocIsKindOf(UML::InitialNode))
let toActivityFinalNode: Set(UML::Element) = src.ownedElement->select(e | e.ocIsKindOf(UML::ActivityFinalNode))
let toObjectFlow: Set(UML::Element) = src.ownedElement->select(e | e.ocIsKindOf(UML::ObjectFlow))
let ignoreActivityPartition: Set(UML::Element) = src.ownedElement->select(e | e.ocIsKindOf(UML::IgnoreActivityPartition))
let ignoreInterruptibleActivityRegion: Set(UML::Element) = src.ownedElement->select(e | e.ocIsKindOf(UML::IgnoreInterruptibleActivityRegion))
let toClassifierMS: Sequence(UML::Element) = src.ownedElement->select(e | e.ocIsKindOf(UML::ClassifierMS))
let variables: Sequence(UML::Element) = src.ownedElement->select(e | e.ocIsKindOf(UML::Variable))
let toElementFMS : Set(UML::Element) = src.ownedElement->select(e | e.ocIsKindOf(UML::ControlFlow))
let toElementOMS: Set(UML::Element) = ((((((((((src.ownedElement-toElementFMS)-variables)-toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toInitialNode->collect(e | InitialNode_Mapping.getMapped(e)))
->union(toActivityFinalNode->collect(e | ActivityFinalNode_Mapping.getMapped(e)))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(toObjectFlow->collect(e | ObjectFlowMembership_Mapping.getMapped(e)))
->union(variables->collect(e | VariableMembership_Mapping.getMapped(e)))
->union(toClassifierMS->collect(e | ElementOwningMembership_Mapping.getMapped(e)))
->union(inputPin->collect(e | InputPinMembership_Mapping.getMapped(e)))
->union(valuePin->collect(e | ValuePinMembership_Mapping.getMapped(e)))
->union(outputPin->collect(e | OutputPinMembership_Mapping.getMapped(e)))
```

- **Type::isSufficient**

```
false
```

- **Feature::isUnique**

```
true
```

- **Element::name**

```
from.name
```

- **Element::shortName**

```
null
```

- **Type::isAbstract**

```
false
```

- **Feature::isOrdered**

```
false
```

- **Element::aliasId**

```
Set{}
```

- **Feature::isPortion**

```
false
```

- Usage::isVariation  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::elementId  
Helper.getID(from)
- Feature::isDerived  
false
- ActionUsage::isComposite  
true

#### **C.2.5.2.3.7 Variable Actions**

##### **C.2.5.2.3.7.1 AddVariableValueAction\_Mapping**

###### **Description**

\*\*\* not specified yet \*\*\*

###### **General Mappings**

CommonAction\_Mapping

###### **Mapping Source**

AddVariableValueAction

###### **Mapping Target**

ActionUsage

###### **Owned Mappings**

- addVariableValueActionFeatureTyping : AddVariableValueActionFeatureTyping\_Mapping

###### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

###### **Mapping rules**

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

- Feature::isEnd



- false
- Element::ownedRelationship
  - ElementOwnership\_Mapping.getMappedColl(from.ownedElement)
- Type::isSufficient
  - false
- Feature::isUnique
  - true
- Element::name
  - from.name
- Element::shortName
  - null
- Type::isAbstract
  - false
- Feature::isOrdered
  - false
- Element::aliasId
  - Set{}
- Feature::isPortion
  - false
- Usage::isVariation
  - false
- Feature::isReadOnly
  - false
- Feature::direction
  - null
- Element::elementId
  - Helper.getID(from)
- Feature::isDerived
  - false
- ActionUsage::isComposite

true

- ActionUsage::ownedRelationship

Helper.actionOwnedRelationship(from)->including(addVariableValueActionFeatureTyping.to)

#### C.2.5.2.3.7.2 AddVariableValueActionFeatureTyping\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureTyping\_Mapping

##### Mapping Source

AddVariableValueAction

##### Mapping Target

FeatureTyping

##### Owned Mappings

- addVariableValueAction : AddVariableValueAction\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId

Set{ }

- Relationship::ownedRelatedElement

Set{ }

- FeatureTyping::type

SysML2::ActionUsage.allInstances()->any(m | m.qualifiedName = 'Actions::AssignmentAction')

- FeatureTyping::typedFeature

addVariableValueAction.to

- Specialization::specific

*abstract rule*

- Element::name

null

- Element::shortName

    null

- Specialization::general  
    *abstract rule*
- Element::elementId

    Helper.createUUID()

- Element::ownedRelationship

    Set{}

### C.2.5.2.3.7.3 ClearVariableAction\_Mapping

#### Description

The expected SysML v2 textual notation of a SysMLv1::ClearVariableAction is as follows

```
action thisIsAClearVariableAction {  
    thisIsAVariable = null;  
}
```

#### General Mappings

CommonAction\_Mapping

#### Mapping Source

ClearVariableAction

#### Mapping Target

ActionUsage

#### Owned Mappings

(none)

#### Applicable filters

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

(none)

#### Mapping rules

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

- Feature::isEnd

    false

- Element::ownedRelationship

```
ElementOwnership_Mapping.getMappedColl(from.ownedElement)
```

- Type::isSufficient

```
false
```

- ActionUsage::ownedRelationship

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

- Feature::isUnique

```
true
```

- Element::name

```
from.name
```

- Element::shortName

```
null
```

- Type::isAbstract

```
false
```

- Feature::isOrdered

```
false
```

- Element::aliasId

```
Set{}
```

- Feature::isPortion

```
false
```

- Usage::isVariation

```
false
```

- Feature::isReadOnly

```
false
```

- Feature::direction

```
null
```

- Element::elementId

```
Helper.getID(from)
```

- Feature::isDerived

```
false
```

- ActionUsage::isComposite

true

#### C.2.5.2.3.7.4 ClearVariableActionFeatureMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureMembership\_Mapping

##### Mapping Source

ClearVariableAction

##### Mapping Target

FeatureMembership

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
null
- FeatureMembership::ownedMemberFeature  
ClearVariableActionReferenceUsage\_Mapping.getMapped(from)
- Element::shortName  
null
- Element::elementId  
Helper.createUUID()
- Element::aliasId  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*

- Membership::memberName  
null
- OwningMembership::ownedRelatedElement  
Set { self.ownedMemberElement () }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
KerML::VisibilityKind::public
- Element::name  
null
- Element::ownedRelationship  
Set { }

#### C.2.5.2.3.7.5 ClearVariableActionReferenceUsage\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToReferenceUsage\_Mapping

##### Mapping Source

ClearVariableAction

##### Mapping Target

ReferenceUsage

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Feature::isEnd

- false
- Type::isSufficient
  - false
- Feature::isUnique
  - true
- Element::shortName
  - null
- Type::isAbstract
  - false
- Element::elementId
  - Helper.createUUID()
- Feature::isOrdered
  - false
- Element::aliasId
  - Set{}
- Feature::isPortion
  - false
- Usage::isVariation
  - false
- ReferenceUsage::ownedRelationship
  - Set{ClearVariableActionReferenceUsageFeatureValue\_Mapping.getMapped(from)}
- Feature::isReadOnly
  - false
- ReferenceUsage::name
  - from.variable.name
- Feature::direction
  - null
- Feature::isDerived
  - false
- Feature::isComposite

false

#### C.2.5.2.3.7.6 ClearVariableActionReferenceUsageFeatureValue\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureValue\_Mapping

##### Mapping Source

ClearVariableAction

##### Mapping Target

FeatureValue

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- FeatureValue::value  
  
Null\_Mapping.getMapped(from)
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*



- Membership::memberName  
null
- OwningMembership::ownedRelatedElement  
Set{self.ownedMemberElement() }
- Membership::visibility  
KerML::VisibilityKind::public
- Element::name  
null
- Element::ownedRelationship  
Set{ }

#### **C.2.5.2.3.7.7 Null\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

CommonValueSpecification\_Mapping

##### **Mapping Source**

Element

##### **Mapping Target**

NullExpression

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Feature::isEnd  
false
- Type::isSufficient  
false

- Feature::isUnique  
true
- Element::shortName  
null
- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false
- Element::ownedRelationship  
Set{}

#### **C.2.5.2.3.7.8 ReadVariableAction\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

CommonAction\_Mapping

### Mapping Source

ReadVariableAction

### Mapping Target

ActionUsage

### Owned Mappings

(none)

### Applicable filters

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

(none)

### Mapping rules

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

- Feature::isEnd  
`false`
- Element::ownedRelationship  
`ElementOwnership_Mapping.getMappedColl(from.ownedElement)`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::name  
`from.name`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`

- Feature::isPortion  
false
- Usage::isVariation  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::elementId  
Helper.getID(from)
- Feature::isDerived  
false
- ActionUsage::isComposite  
true

#### **C.2.5.2.3.7.9 RemoveVariableValueAction\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

CommonAction\_Mapping

##### **Mapping Source**

RemoveVariableValueAction

##### **Mapping Target**

ActionUsage

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- `Feature::isEnd`  
`false`
- `Element::ownedRelationship`  
`ElementOwnership_Mapping.getMappedColl(from.ownedElement)`
- `Type::isSufficient`  
`false`
- `Feature::isUnique`  
`true`
- `Element::name`  
`from.name`
- `Element::shortName`  
`null`
- `Type::isAbstract`  
`false`
- `Feature::isOrdered`  
`false`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `Usage::isVariation`  
`false`
- `Feature::isReadOnly`  
`false`
- `Feature::direction`  
`null`
- `Element::elementId`  
`Helper.getID(from)`
- `Feature::isDerived`

false

- ActionUsage::ownedRelationship

```
Helper.actionOwnedRelationship(from)
->including(RemoveVariableValueActionExpressionMembership_Mapping.getMapped(from))
->including(RemoveVariableValueActionExpressionParameterMembership_Mapping.getMapped(from.rem
->including(RemoveVariableValueActionExpressionParameterMembership_Mapping.getMapped(from.val
->including(EmptyReturnParameterFeatureMembership_Mapping.getMapped(from))
->including(RemoveVariableValueActionAssignmentActionMembership_Mapping.getMapped(from))
```

- ActionUsage::isComposite

true

#### C.2.5.2.3.7.10 RemoveVariableValueActionAssignmentAction\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToAssignmentActionUsage\_Mapping

##### Mapping Source

RemoveVariableValueAction

##### Mapping Target

AssignmentActionUsage

##### Owned Mappings

(none)

##### Applicable filters

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

(none)

##### Mapping rules

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

- Feature::isEnd

false

- Type::isSufficient

false

- Feature::isUnique

true

- Element::shortName  
null
- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Feature::isOrdered  
false
- AssignmentActionUsage::ownedRelationship  
Set{RemoveVariableValueActionAssignmentActionParameterMembership\_Mapping.getMapped(from), Re
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Usage::isVariation  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- ActionUsage::isComposite  
true

#### **C.2.5.2.3.7.11 RemoveVariableValueActionAssignmentActionMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToOwningMembership\_Mapping

### Mapping Source

RemoveVariableValueAction

### Mapping Target

OwningMembership

### Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Relationship::ownedRelatedElement  
`Set{ }`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Membership::memberElement  
*abstract rule*
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{ }`
- Membership::memberName  
`null`
- OwningMembership::ownedMemberElement  
`RemoveVariableValueActionAssignmentAction_Mapping.getMapped(from)`
- Membership::visibility



KerML::VisibilityKind::public

- Element::name

null

- Element::ownedRelationship

Set{}

#### **C.2.5.2.3.7.12 RemoveVariableValueActionAssignmentActionParameter\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToReferenceUsage\_Mapping

##### **Mapping Source**

RemoveVariableValueAction

##### **Mapping Target**

ReferenceUsage

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- Feature::isEnd

false

- Type::isSufficient

false

- Feature::isUnique

true

- Element::shortName

null

- Type::isAbstract

- false
- Element::elementId
  - Helper.createUUID()
- Feature::isOrdered
  - false
- Element::aliasId
  - Set{}
- Feature::isPortion
  - false
- Usage::isVariation
  - false
- Feature::isReadOnly
  - false
- ReferenceUsage::ownedRelationship
  - Set{RemoveVariableValueActionAssignmentActionParameterFeatureMembership\_Mapping.getMapped(fr
- ReferenceUsage::direction
  - KerML::FeatureDirectionKind::\_'in'
- Element::name
  - null
- Feature::isDerived
  - false
- Feature::isComposite
  - false

#### **C.2.5.2.3.7.13 RemoveVariableValueActionAssignmentActionParameterFeatureMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureMembership\_Mapping

##### **Mapping Source**

RemoveVariableValueAction

## Mapping Target

FeatureMembership

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName

null

- Element::shortName

null

- Element::elementId

Helper.createUUID()

- Element::aliasId

Set{}

- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName

null

- OwningMembership::ownedRelatedElement

Set{self.ownedMemberElement() }

- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility

KerML::VisibilityKind::public

- Element::name

null

- FeatureMembership::ownedMemberFeature

```
RemoveVariableValueActionAssignmentActionParameterReference_Mapping.getMapped(from)
```

- Element::ownedRelationship

```
Set{ }
```

#### **C.2.5.2.3.7.14 RemoveVariableValueActionAssignmentActionParameterMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToParameterMembership\_Mapping

##### **Mapping Source**

RemoveVariableValueAction

##### **Mapping Target**

ParameterMembership

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- FeatureMembership::ownedRelatedElement

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName

```
null
```

- Element::shortName

```
null
```

- Element::elementId

```
Helper.createUUID()
```

- Element::aliasId  
Set{ }
- FeatureMembership::ownedMemberFeature  
*abstract rule*
- FeatureMembership::owningType  
*abstract rule*
- Membership::memberName  
null
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- ParameterMembership::ownedMemberParameter  
RemoveVariableValueActionAssignmentActionParameter\_Mapping.getMapped(from)
- Membership::visibility  
KerML::VisibilityKind::public
- Element::name  
null
- Element::ownedRelationship  
Set{ }

#### **C.2.5.2.3.7.15 RemoveVariableValueActionAssignmentActionParameterReference\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToReferenceUsage\_Mapping

##### **Mapping Source**

RemoveVariableValueAction

##### **Mapping Target**

ReferenceUsage

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- ReferenceUsage::ownedRelationship  
`Set{RemoveVariableValueActionAssignmentActionParameterReferenceFeatureMembership_Mapping.get`
- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Usage::isVariation  
`false`
- Feature::isReadOnly  
`false`
- Feature::direction  
`null`

- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false

#### C.2.5.2.3.7.16

#### RemoveVariableValueActionAssignmentActionParameterReferenceFeatureMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureMembership\_Mapping

##### Mapping Source

RemoveVariableValueAction

##### Mapping Target

FeatureMembership

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
null
- FeatureMembership::ownedMemberFeature  
RemoveVariableValueActionAssignmentActionParameterReferenceReference\_Mapping.getMapped(from)
- Element::shortName  
null

- Element::elementId  
Helper.createUUID()
- Element::aliasId  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
null
- OwningMembership::ownedRelatedElement  
Set{self.ownedMemberElement() }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
KerML::VisibilityKind::public
- Element::name  
null
- Element::ownedRelationship  
Set{ }

#### **C.2.5.2.3.7.17 RemoveVariableValueActionAssignmentActionParameterReferenceReference\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToReferenceUsage\_Mapping

##### **Mapping Source**

RemoveVariableValueAction

##### **Mapping Target**

ReferenceUsage

##### **Owned Mappings**

(none)

##### **Applicable filters**



This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Usage::isVariation  
`false`
- Feature::isReadOnly  
`false`
- Feature::direction  
`null`
- Element::name  
`null`

- Feature::isDerived  
false
- Feature::isComposite  
false
- Element::ownedRelationship  
Set{ }

#### **C.2.5.2.3.7.18 RemoveVariableValueActionAssignmentActionSecondParameter\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToReferenceUsage\_Mapping

##### **Mapping Source**

RemoveVariableValueAction

##### **Mapping Target**

ReferenceUsage

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Feature::isEnd  
false
- Type::isSufficient  
false
- Feature::isUnique  
true
- Element::shortName  
null

- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Usage::isVariation  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false
- Element::ownedRelationship  
Set{}

#### **C.2.5.2.3.7.19 RemoveVariableValueActionAssignmentActionSecondParameterMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToParameterMembership\_Mapping

##### **Mapping Source**

RemoveVariableValueAction

## Mapping Target

ParameterMembership

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- ParameterMembership::ownedMemberParameter  
`RemoveVariableValueActionAssignmentActionSecondParameter_Mapping.getMapped(from)`
- FeatureMembership::ownedRelatedElement  
`Set{self.ownedMemberFeature() }`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{ }`
- FeatureMembership::ownedMemberFeature  
*abstract rule*
- FeatureMembership::owningType  
*abstract rule*
- Membership::memberName  
`null`
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*

- Membership::visibility  
KerML::VisibilityKind::public
- Element::name  
null
- Element::ownedRelationship  
Set{ }

#### C.2.5.2.3.7.20 RemoveVariableValueActionExpressionMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureMembership\_Mapping

##### Mapping Source

RemoveVariableValueAction

##### Mapping Target

FeatureMembership

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
null
- Element::shortName  
null
- Element::elementId  
Helper.createUUID()

- Element::aliasId  
`Set {}`
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
`null`
- OwningMembership::ownedRelatedElement  
`Set { self.ownedMemberElement () }`
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- FeatureMembership::ownedMemberFeature  
`RemoveVariableValueActionExpressionReferenceUsage_Mapping.getMapped (from)`
- Membership::visibility  
`KerML::VisibilityKind::public`
- Element::name  
`null`
- Element::ownedRelationship  
`Set {}`

#### **C.2.5.2.3.7.21 RemoveVariableValueActionExpressionParameter\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeature\_Mapping

##### **Mapping Source**

Pin

##### **Mapping Target**

Feature

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  
`Set{ }`
- Type::isSufficient  
`false`
- Feature::ownedRelationship  
`Set{ RemoveVariableValueActionExpressionParameterValue_Mapping.getMapped(from) }`
- Element::name  
`null`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`

#### C.2.5.2.3.7.22 RemoveVariableValueActionExpressionParameterFeatureReference\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureReferenceExpression\_Mapping

##### Mapping Source

Pin

##### Mapping Target

FeatureReferenceExpression

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- FeatureReferenceExpression::ownedRelationship  
`Set{RemoveVariableValueActionExpressionParameterFeatureReferenceMembership_Mapping.getMapped}`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Feature::isReadOnly  
`false`
- Feature::direction  
`null`
- Element::name  
`null`



- Feature::isDerived  
false
- Feature::isComposite  
false

#### **C.2.5.2.3.7.23 RemoveVariableValueActionExpressionParameterFeatureReferenceMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToMembership\_Mapping

##### **Mapping Source**

Pin

##### **Mapping Target**

Membership

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Element::aliasId  
Set{ }
- Relationship::ownedRelatedElement  
Set{ }
- Relationship::source  
Set{ }
- Element::name  
null
- Element::shortName  
null

- Membership::memberElement

from

- Element::elementId

Helper.createUUID()

- Relationship::target

Set{ }

- Element::ownedRelationship

Set{ }

#### **C.2.5.2.3.7.24 RemoveVariableValueActionExpressionParameterMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToParameterMembership\_Mapping

##### **Mapping Source**

Pin

##### **Mapping Target**

ParameterMembership

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- ParameterMembership::ownedMemberParameter

RemoveVariableValueActionExpressionParameter\_Mapping.getMapped(from)

- FeatureMembership::ownedRelatedElement

Set{self.ownedMemberFeature() }

- Membership::membershipOwningNamespace

*abstract rule*

- Membership::memberShortName  
null
- Element::shortName  
null
- Element::elementId  
Helper.createUUID()
- Element::aliasId  
Set{}
- FeatureMembership::ownedMemberFeature  
*abstract rule*
- FeatureMembership::owningType  
*abstract rule*
- Membership::memberName  
null
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
KerML::VisibilityKind::public
- Element::name  
null
- Element::ownedRelationship  
Set{}

#### **C.2.5.2.3.7.25 RemoveVariableValueActionExpressionParameterValue\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureValue\_Mapping

##### **Mapping Source**

Pin

##### **Mapping Target**

FeatureValue

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
  
null
- OwningMembership::ownedRelatedElement  
  
Set{self.ownedMemberElement() }
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- FeatureValue::value  
  
RemoveVariableValueActionExpressionParameterFeatureReference\_Mapping.getMapped(from)
- Element::ownedRelationship  
  
Set{}

#### C.2.5.2.3.7.26 RemoveVariableValueActionExpressionReferenceUsage\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToReferenceUsage\_Mapping

##### Mapping Source

RemoveVariableValueAction

##### Mapping Target

ReferenceUsage

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Feature::isEnd  
false
- Type::isSufficient  
false
- Feature::isUnique  
true
- Element::shortName  
null
- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Feature::isOrdered  
false

- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `Usage::isVariation`  
`false`
- `Feature::isReadOnly`  
`false`
- `ReferenceUsage::name`  
`from.variable.name`
- `Feature::direction`  
`null`
- `Feature::isDerived`  
`false`
- `ReferenceUsage::ownedRelationship`  
`Set{RemoveVariableValueActionExpressionReferenceUsageFeatureValue_Mapping.getMapped(from)}`
- `Feature::isComposite`  
`false`

#### **C.2.5.2.3.7.27 RemoveVariableValueActionExpressionReferenceUsageFeatureValue\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureValue\_Mapping

##### **Mapping Source**

RemoveVariableValueAction

##### **Mapping Target**

FeatureValue

##### **Owned Mappings**

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
  
null
- OwningMembership::ownedRelatedElement  
  
Set{self.ownedMemberElement() }
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- FeatureValue::value  
  
RemoveVariableValueActionInvocationExpression\_Mapping.getMapped(from)
- Element::ownedRelationship  
  
Set{}

### C.2.5.2.3.7.28 RemoveVariableValueActionInvocationExpression\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

## General Mappings

GenericToInvocationExpression\_Mapping

## Mapping Source

RemoveVariableValueAction

## Mapping Target

InvocationExpression

## Owned Mappings

- removeVariableValueActionInvocationExpressionFeatureTyping :  
RemoveVariableValueActionInvocationExpressionFeatureTyping\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{ }`



- Feature::isPortion  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- InvocationExpression::ownedRelationship  
Set{removeVariableValueActionInvocationExpressionFeatureTyping.to}
- Feature::isComposite  
false

#### **C.2.5.2.3.7.29 RemoveVariableValueActionInvocationExpressionFeatureTyping\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureTyping\_Mapping

##### **Mapping Source**

RemoveVariableValueAction

##### **Mapping Target**

FeatureTyping

##### **Owned Mappings**

- removeVariableValueActionInvocationExpression :  
RemoveVariableValueActionInvocationExpression\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Element::aliasId  
`Set{}`
- Relationship::ownedRelatedElement  
`Set{}`
- Specialization::specific  
*abstract rule*
- FeatureTyping::type  
`KerML::Function.allInstances()->any(m | m.qualifiedName = 'SequenceFunctions::excluding')`
- Element::name  
`null`
- Element::shortName  
`null`
- Specialization::general  
*abstract rule*
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set{}`
- FeatureTyping::typedFeature  
`removeVariableValueActionInvocationExpression.to`

### C.2.5.3 Activities

#### C.2.5.3.1 Overview

**Table 15. List of all Overview Mapping Specifications**

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
Activity	Behavior	CommonActivity_Mapping	true
ActivityEdge	SuccessionAsUsage	CommonActivityEdgeSuccessionAsUsage_Mapping	
	FeatureMembership	ActivityEdgeMetadataFeatureMembership_Mapping	
	OwningMembership	ActivityEdgeMetadataMembership_Mapping	
	FeatureTyping	ActivityEdgeMetadataFeatureTyping_Mapping	
	Redefinition	ActivityEdgeMetadataRedefinition_Mapping	
	MetadataUsage	ActivityEdgeMetadata_Mapping	
	ReferenceUsage	ActivityEdgeMetadataReferenceUsage_Mapping	
	FeatureValue	ActivityEdgeMetadataFeatureValue_Mapping	
ActivityFinalNode	Membership	ActivityFinalNodeMembership_Mapping	

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
ActivityGroup	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	
ActivityNode	Membership Feature FeatureMembership Feature Redefinition Subsetting Feature FeatureMembership Subsetting FeatureMembership FeatureMembership FeatureMembership ItemFlowEnd Feature Subsetting Subsetting Subsetting ItemFlowFeature	ActivityEdgeTransitionUsageSourceMembership_Mapping ControlFlowTargetEndFeature_Mapping ControlFlowTargetEndFeatureMembership_Mapping ActivityEdgeSourceEndFeature_Mapping ObjectFlowItemFlowRedefinition_Mapping ControlFlowTargetFinalNodeSubsetting_Mapping ControlFlowTargetFinalNode_Mapping ObjectFlowItemFlowFeatureMembership_Mapping ActivityEdgeSourceEndSubsetting_Mapping ActivityEdgeSourceEndFeatureMembership_Mapping ActivityEdgeInitialNodeSourceEndFeatureMembership_Mapping ControlFlowFinalNodeTargetEndFeatureMembership_Mapping ObjectFlowEndFeatureMembership_Mapping ObjectFlowItemFlowEnd_Mapping ActivityEdgeSourceInitialNode_Mapping ActivityEdgeSourceInitialNodeSubsetting_Mapping ObjectFlowItemFlowSubsetting_Mapping ControlFlowTargetEndSubsetting_Mapping ObjectFlowItemFlowFeature_Mapping	
ActivityParameterNode	FeatureTyping ItemFeature	ObjectFlowItemFeatureTyping_Mapping ObjectFlowItemFeatureUntyped_Mapping	not src.type.oclIsUndefined() and not(src.type.oclIsKindOf(UML::Enumeration)) and Helper.getSysMLv2EnumerationDefinition(src.type)
ActivityPartition	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	
CentralBufferNode	FeatureTyping ItemFeature	ObjectFlowItemFeatureTyping_Mapping ObjectFlowItemFeatureUntyped_Mapping	not src.type.oclIsUndefined() and not(src.type.oclIsKindOf(UML::Enumeration)) and Helper.getSysMLv2EnumerationDefinition(src.type)

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
ControlFlow	TransitionUsage SuccessionAsUsage FeatureReferenceExpression TransitionFeatureMembership Membership	ControlFlowTransitionUsage_Mapping ControlFlowSuccessionAsUsage_Mapping ControlFlowTransitionUsageFeatureReferenceExpression_Mapping ControlFlowTransitionUsageFeatureMembership_Mapping ControlFlowTransitionUsageFeatureReferenceExpressionMembership_Mapping	not ControlFlow.guard.ocllsUndefined()
ControlNode	Membership Feature FeatureMembership Feature Redefinition Subsetting Feature FeatureMembership Subsetting FeatureMembership FeatureMembership FeatureMembership ItemFlowEnd Feature Subsetting Subsetting Subsetting ItemFlowFeature	ActivityEdgeTransitionUsageSourceMembership_Mapping ControlFlowTargetEndFeature_Mapping ControlFlowTargetEndFeatureMembership_Mapping ActivityEdgeSourceEndFeature_Mapping ObjectFlowItemFlowRedefinition_Mapping ControlFlowTargetFinalNodeSubsetting_Mapping ControlFlowTargetFinalNode_Mapping ObjectFlowItemFlowFeatureMembership_Mapping ActivityEdgeSourceEndSubsetting_Mapping ActivityEdgeSourceEndFeatureMembership_Mapping ActivityEdgeInitialNodeSourceEndFeatureMembership_Mapping ControlFlowFinalNodeTargetEndFeatureMembership_Mapping ObjectFlowEndFeatureMembership_Mapping ObjectFlowItemFlowEnd_Mapping ActivityEdgeSourceInitialNode_Mapping ActivityEdgeSourceInitialNodeSubsetting_Mapping ObjectFlowItemFlowSubsetting_Mapping ControlFlowTargetEndSubsetting_Mapping ObjectFlowItemFlowFeature_Mapping	
DataStoreNode	FeatureTyping ItemFeature	ObjectFlowItemFeatureTyping_Mapping ObjectFlowItemFeatureUntyped_Mapping	not src.type.ocllsUndefined() and not(src.type.ocllsKindOf(UML::Enumeration)) and Helper.getSysMLv2EnumerationDefinition(src.type)
DecisionNode	DecisionNode	DecisionNode_Mapping	

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
ExceptionHandler	FeatureTyping	CommonParameterReferenceUsageInFeatureTyping_Mapping	
	FeatureTyping	Mapping	
	Element	CommonReturnParameterFeatureUntyped_Mapping	
	Feature	CommonReturnParameterFeatureTyping_Mapping	
	FeatureTyping	ElementOwnership_Mapping	
	Relationship	CommonValueSpecification_Mapping	
	Expression	CommonParameterReferenceUsageInUntyped_Mapping	
	ReferenceUsage	DefaultMultiplicityMembership_Mapping	
	OwningMembership	CommonReturnParameterFeatureMembership_Mapping	
	LiteralInteger	CommonParameterReferenceUsageInMembership_Mapping	
	ReturnParameterMembership	DefaultMultiplicityBoundOwnership_Mapping	
	ParameterMembership	CommonReturnParameterReferenceUsageFeatureTyping_Mapping	
	FeatureMembership	DefaultUpperBound_Mapping	
	FeatureTyping	DefaultMultiplicityElement_Mapping	
	LiteralInteger	CommonReturnParameterReferenceUsageUntyped_Mapping	
	MultiplicityRange	DefaultLowerBound_Mapping	
	ReferenceUsage	ElementMain_Mapping	
	LiteralInteger	ElementMembership_Mapping	
	Element	CommonReturnParameterReferenceUsageMembership_Mapping	
	Membership	EmptyReturnParameterFeatureMembership_Mapping	
	ReturnParameterMembership	ReturnParameterMembership	
ExecutableNode	Membership	ActivityEdgeTransitionUsageSourceMembership_Mapping	
	Feature	ControlFlowTargetEndFeature_Mapping	
	FeatureMembership	ControlFlowTargetEndFeatureMembership_Mapping	
	Feature	ActivityEdgeSourceEndFeature_Mapping	
	Redefinition	ObjectFlowItemFlowRedefinition_Mapping	
	Subsetting	ControlFlowTargetFinalNodeSubsetting_Mapping	
	Feature	ControlFlowTargetFinalNode_Mapping	
	FeatureMembership	ObjectFlowItemFlowFeatureMembership_Mapping	
	Subsetting	ActivityEdgeSourceEndSubsetting_Mapping	
	FeatureMembership	ActivityEdgeSourceEndFeatureMembership_Mapping	
	FeatureMembership	ActivityEdgeInitialNodeSourceEndFeatureMembership_Mapping	
	FeatureMembership	ControlFlowFinalNodeTargetEndFeatureMembership_Mapping	
	FeatureMembership	ObjectFlowEndFeatureMembership_Mapping	
	ItemFlowEnd	ObjectFlowItemFlowEnd_Mapping	
	Feature	ActivityEdgeSourceInitialNode_Mapping	
	Subsetting	ActivityEdgeSourceInitialNodeSubsetting_Mapping	
	Subsetting	ObjectFlowItemFlowSubsetting_Mapping	
	Subsetting	ControlFlowTargetEndSubsetting_Mapping	
	ItemFlowFeature	ObjectFlowItemFlowFeature_Mapping	
FinalNode	Membership	ActivityFinalNodeMembership_Mapping	
FlowFinalNode	Membership	ActivityFinalNodeMembership_Mapping	
ForkNode	ForkNode	ForkNode_Mapping	
InitialNode	Membership	InitialNodeMembership_Mapping	

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
InterruptibleActivityRegion	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	
JoinNode	JoinNode	JoinNode_Mapping	
MergeNode	MergeNode	MergeNode_Mapping	
ObjectFlow	TransitionUsage Feature FeatureMembership FeatureMembership FeatureMembership FeatureMembership Subsetting SuccessionFlowConnectionUsage	ObjectFlowGuard_Mapping ObjectFlowGuardSuccessionTargetEndFeature_Mapping ObjectFlowFeatureMembership_Mapping ObjectFlowGuardSuccessionTargetEndFeatureMembership_Mapping ObjectFlowItemFeatureMembership_Mapping ObjectFlowGuardFeatureMembership_Mapping ObjectFlowGuardSuccessionTargetEndSubsetting_Mapping ObjectFlow_Mapping	
ObjectNode	FeatureTyping ItemFeature	ObjectFlowItemFeatureTyping_Mapping ObjectFlowItemFeatureUntyped_Mapping	not src.type.ocIsUndefined() and not(src.type.ocIsKindOf(UML::Enumeration)) and Helper.getSysMLv2EnumerationDefinition(src.type.oc)
Variable	FeatureTyping FeatureMembership Feature	VariableFeatureTyping_Mapping VariableMembership_Mapping CommonVariable_Mapping	not src.type.ocIsUndefined() and not(src.type.ocIsKindOf(UML::Enumeration)) and Helper.getSysMLv2EnumerationDefinition(src.type.oc)

### C.2.5.3.2 SysML v1 Activities elements not mapped

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

SysML v1 Concept	Rationale
ActivityParameterNode	The parameter of the activity is mapped from SysML v1 to SysML v2. The additional concept of the activity parameter node is necessary for the token semantic of SysML v1 activities, which is not part of SysML v2. Therefore, the additional concept of the activity parameter node is not mapped to SysML v2.
FlowFinalNode	The flow final node is required for the token semantic, which is not part of SysML v2. Therefore, the element FlowFinalNode is not mapped.

### C.2.5.3.3 Mapping Specifications

#### C.2.5.3.3.1 ActivityAsDefinition\_Mapping

##### Description

A UML4SysML::Activity is mapped to a SysMLv2::ActionDefinition.

##### 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.ocIsKindOf(UML::Package)
```

##### Mapping rules

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

- Element::aliasId  

```
Set{}
```
- Classifier::isAbstract  

```
from.isAbstract
```
- Type::isSufficient  

```
false
```
- Namespace::ownedImport  

```
Set{}
```
- Behavior::ownedRelationship

```
let parameters: Set(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::Parameter))
let parameterSets: Set(UML::Element) = src.ownedElement->select(e | e.ocIsKindOf(UML::ParameterSet))
let features: Set(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::Property))
let elementsOMS: Set(UML::Element) = (((from.ownedElement - parameters) parameterSets) - features)
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)))
```

- Element::elementId

```
Helper.getID(from)
```

- Element::name

```
from.name
```

- Element::shortName

```
null
```

#### **C.2.5.3.3.2 ActivityAsUsage\_Mapping**

##### **Description**

A UML4SysML::Activity is mapped to a SysMLv2::ActionDefinition.

##### **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.ocIsKindOf(UML::Package)
```

##### **Mapping rules**

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

- Element::aliasId

```
Set{}
```

- Classifier::isAbstract

```
from.isAbstract
```

- Type::isSufficient

```
false
```



- Namespace::ownedImport

```
Set{}
```

- Behavior::ownedRelationship

```
let parameters: Set(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::Parameter))
let parameterSets: Set(UML::Element) = src.ownedElement->select(e | e.ocIsKindOf(UML::ParameterSet))
let features: Set(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::Property))
let elementsOMS: Set(UML::Element) = (((from.ownedElement - parameters) parameterSets) - features)
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)))
```

- Element::elementId

```
Helper.getID(from)
```

- Element::name

```
from.name
```

- Element::shortName

```
null
```

### C.2.5.3.3.3 ActivityEdgeMetadata\_Mapping

#### Description

Adds metadata to the target elements of UML::ControlFlow and UML::ObjectFlow to map the UML::ActivityEdge::weight property which has no direct target in SysML v2.

#### General Mappings

GenericToMetadataUsage\_Mapping

#### Mapping Source

ActivityEdge

#### Mapping Target

MetadataUsage

#### Owned Mappings

- activityEdgeMetadataFeatureMembership : ActivityEdgeMetadataFeatureMembership\_Mapping
- activityEdgeMetadataFeatureTyping : ActivityEdgeMetadataFeatureTyping\_Mapping

#### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

#### Mapping rules

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

- `Feature::isEnd`  
`false`
- `Type::isSufficient`  
`false`
- `Feature::isUnique`  
`true`
- `MetadataUsage::name`  
`'weight'`
- `Element::shortName`  
`null`
- `Type::isAbstract`  
`false`
- `Element::elementId`  
`Helper.createUUID()`
- `Feature::isOrdered`  
`false`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `Usage::isVariation`  
`false`
- `Feature::isReadOnly`  
`false`
- `MetadataUsage::ownedRelationship`  
`Set{activityEdgeMetadataFeatureTyping.to, activityEdgeMetadataFeatureMembership.to}`
- `Feature::direction`  
`null`
- `Feature::isDerived`

false

- Feature::isComposite

false

#### **C.2.5.3.3.4 ActivityEdgeMetadataFeatureMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureMembership\_Mapping

##### **Mapping Source**

ActivityEdge

##### **Mapping Target**

FeatureMembership

##### **Owned Mappings**

- activityEdgeMetadataReferenceUsage : ActivityEdgeMetadataReferenceUsage\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName

null

- Element::shortName

null

- Element::elementId

Helper.createUUID()

- Element::aliasId

Set{}

- OwningMembership::ownedMemberElement  
*abstract rule*

- FeatureMembership::ownedMemberFeature  
activityEdgeMetadataReferenceUsage.to
- Membership::memberName  
null
- OwningMembership::ownedRelatedElement  
Set { self.ownedMemberElement () }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
KerML::VisibilityKind::public
- Element::name  
null
- Element::ownedRelationship  
Set { }

#### C.2.5.3.3.5 ActivityEdgeMetadataFeatureTyping\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureTyping\_Mapping

##### Mapping Source

ActivityEdge

##### Mapping Target

FeatureTyping

##### Owned Mappings

- activityEdgeMetadata : ActivityEdgeMetadata\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId

Set{}

- Relationship::ownedRelatedElement

Set{}

- Specialization::specific

*abstract rule*

- FeatureTyping::typedFeature

activityEdgeMetadata.to

- FeatureTyping::type

SysML2::MetadataDefinition.allInstances()->any(m | m.qualifiedName = 'SysMLv1Library::Activi

- Element::name

null

- Element::shortName

null

- Specialization::general

*abstract rule*

- Element::elementId

Helper.createUUID()

- Element::ownedRelationship

Set{}

#### C.2.5.3.3.6 ActivityEdgeMetadataFeatureValue\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureValue\_Mapping

##### Mapping Source

ActivityEdge

##### Mapping Target

FeatureValue

##### Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- FeatureValue::value  
  
from.weight
- Membership::memberName  
  
null
- OwningMembership::ownedRelatedElement  
  
Set{self.ownedMemberElement() }
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- Element::ownedRelationship  
  
Set{}

#### C.2.5.3.3.7 ActivityEdgeMetadataMembership\_Mapping

##### Description

Creates the owning membership relationship for the target of ActivityEdgeMetadata\_Mapping.

### General Mappings

GenericToOwningMembership\_Mapping

### Mapping Source

ActivityEdge

### Mapping Target

OwningMembership

### Owned Mappings

- activityEdgeMetadata : ActivityEdgeMetadata\_Mapping

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Relationship::ownedRelatedElement  
`Set{ }`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Membership::memberElement  
*abstract rule*
- OwningMembership::ownedMemberElement  
`activityEdgeMetadata.to`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{ }`
- Membership::memberName

null

- Membership::visibility

    KerML::VisibilityKind::public

- Element::name

    null

- Element::ownedRelationship

    Set{}

#### **C.2.5.3.3.8 ActivityEdgeMetadataRedefinition\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToRedefinition\_Mapping

##### **Mapping Source**

ActivityEdge

##### **Mapping Target**

Redefinition

##### **Owned Mappings**

- activityEdgeMetadataReferenceUsage : ActivityEdgeMetadataReferenceUsage\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Element::aliasId

    Set{}

- Redefinition::redefiningFeature

    activityEdgeMetadataReferenceUsage.to

- Redefinition::redefinedFeature

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

- Subsetting::ownedRelatedElement



```

Set {}

• Subsetting::subsettingFeature
  abstract rule
• Element::name

  null

• Subsetting::subsettingFeature
  abstract rule
• Element::shortName

  null

• Element::elementId

  Helper.createUUID()

• Element::ownedRelationship

  Set {}

```

#### C.2.5.3.3.9 ActivityEdgeMetadataReferenceUsage\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToReferenceUsage\_Mapping

##### Mapping Source

ActivityEdge

##### Mapping Target

ReferenceUsage

##### Owned Mappings

- activityEdgeMetadataFeatureValue : ActivityEdgeMetadataFeatureValue\_Mapping
- activityEdgeMetadataRedefinition : ActivityEdgeMetadataRedefinition\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Feature::isEnd

 

```

false

```

- `Type::isSufficient`  
`false`
- `ReferenceUsage::ownedRelationship`  
`Set{activityEdgeMetadataRedefinition.to, activityEdgeMetadataFeatureValue.to}`
- `Feature::isUnique`  
`true`
- `Element::shortName`  
`null`
- `Type::isAbstract`  
`false`
- `Element::elementId`  
`Helper.createUUID()`
- `Feature::isOrdered`  
`false`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `Usage::isVariation`  
`false`
- `Feature::isReadOnly`  
`false`
- `Feature::direction`  
`null`
- `Element::name`  
`null`
- `Feature::isDerived`  
`false`
- `Feature::isComposite`  
`false`

#### C.2.5.3.3.10 ActivityFinalNodeMembership\_Mapping

##### Description

The mapping creates a membership relationship to the action usage library element Systems Library::Actions::Action::done.

##### General Mappings

GenericToMembership\_Mapping

##### Mapping Source

FinalNode

##### Mapping Target

Membership

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId  
`Set{}`
- Relationship::ownedRelatedElement  
`Set{}`
- Relationship::source  
`Set{}`
- Membership::memberElement  
`SysMLv2::ActionUsage.allInstances()->any(e | e.qualifiedName = 'Actions::Action::done')`
- Element::name  
`null`
- Element::shortName  
`null`
- Element::elementId

```
Helper.createUUID()
```

- Relationship::target

```
Set{}
```

- Element::ownedRelationship

```
Set{}
```

#### **C.2.5.3.3.11 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**

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

```
not Helper.hasStereotypeApplied(from, 'SysML::Requirements::Requirement') and not from.oclIsTypeOf(
```

##### **Mapping rules**

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

- Element::aliasId

```
Set{}
```

- Classifier::isAbstract

```
from.isAbstract
```

- Type::isSufficient

```
false
```

- Namespace::ownedImport

```
Set{}
```

- Behavior::ownedRelationship

```
let relationships : Set(KerML::Relationship) = Helper.activityOwnedRelationship(from) in
let parameters : Set(UML::Paramter) = from.ownedElement->select(e | e.ocIsKindOf(UML::Paramter))
relationships->union(parameters->collect(p | ParameterMembership_Mapping.getMapped(p)))
```

- Classifier::ownedRelationship

```
let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | (e.ocIsKindOf(UML::Propert
let redefinedAttributes: Set(UML::Element) = from.ownedElement->select(e | from.ocIsKindOf(U
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.ocIsKindOf
let constraints : Set(UML::Constraint) = UML::Constraint.allInstances()->select( c | c.constr
let toElementOMS: Set(UML::Element) = ((from.ownedElement - toElementFMS) - redefinedAttribu
let relationships: Sequence(KerML::Relationship) =
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(constraints->collect(e | ConstrainedElementFeatureMembership_Mapping.getMapped(e)))
->union(redefinedAttributes->collect(e | AttributeRedefinedMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e))) in
if from.classifierBehavior.ocIsUndefined() then relationships else relationships->append(Cla
```

- Element::elementId

```
Helper.getID(from)
```

- Element::name

```
from.name
```

- Element::shortName

```
null
```

#### C.2.5.3.3.12 CommonActivityEdgeSuccessionAsUsage\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToConnector\_Mapping

##### Mapping Source

ActivityEdge

##### Mapping Target

SuccessionAsUsage

##### Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Relationship::ownedRelatedElement

```
Set{}
```

- Feature::isEnd

```
false
```

- Type::isSufficient

```
false
```

- SuccessionAsUsage::ownedRelationship

```
Set{if from.source.ocIsKindOf(UML::InitialNode) then ActivityEdgeInitialNodeSourceEndFeatur  
->union(let relationship : KerML::Relationship = if from.ocIsKindOf(UML::ObjectFlow) then OK  
if from.target.ocIsKindOf(UML::FinalNode) then ControlFlowFinalNodeTargetEndFeatureMembershi  
if from.guard.ocIsUndefined() then Set{relationship} else Set{relationship, ElementFeatureMe
```

- Feature::isUnique

```
true
```

- Element::shortName

```
null
```

- Type::isAbstract

```
false
```

- Element::elementId

```
Helper.createUUID()
```

- Feature::isOrdered

```
false
```

- Relationship::target

```
Set{}
```

- Element::aliasId

```
Set{}
```

- Feature::isPortion

```
false
```

- Feature::isReadOnly

false

- Relationship::source

Set{}

- Feature::direction

null

- Element::name

null

- Feature::isDerived

false

- Feature::isComposite

false

#### **C.2.5.3.3.13 CommonVariable\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

PropertyCommon\_Mapping

##### **Mapping Source**

Variable

##### **Mapping Target**

Feature

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- Feature::isDerived

false

- **Feature::isEnd**  
false
- **Feature::isOrdered**  
from.isOrdered
- **Type::isSufficient**  
false
- **Feature::isAbstract**  
false
- **Element::shortName**  
null
- **Feature::ownedRelationship**  

```

let typing: KerML::FeatureTyping = StructuralFeatureToFeatureTyping_Mapping.getMapped(from)
if typing.oclIsUndefined() then
    Set{MultiplicityMembership_Mapping.getMapped(from)}
else
    Set{MultiplicityMembership_Mapping.getMapped(from), typing}
endif

```
- **Feature::ownedRelationship**  

```

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

```
- **Element::elementId**  
Helper.createUUID()
- **Feature::isComposite**  
false
- **Element::aliasId**  
Set{}
- **Feature::isPortion**  
false
- **Feature::isEnd**  
false
- **Feature::direction**



null

- Element::name

null

- Feature::isDerived

false

- Feature::isComposite

false

- Feature::isUnique

from.isUnique

- Feature::isReadOnly

*abstract rule*

#### **C.2.5.3.3.14 ControlFlowTransitionUsage\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToUsage\_Mapping

NamedElementMain\_Mapping

##### **Mapping Source**

ControlFlow

##### **Mapping Target**

TransitionUsage

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

```
not from.guard.oclIsUndefined()
```

##### **Mapping rules**

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

- Feature::isEnd

false

- Type::isSufficient

false

- TransitionUsage::ownedRelationship

```
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.ocIsTypeOf(UML::Opaque
if from.weight.ocIsUndefined() then relationshipsWithGuard else relationshipsWithGuard->incl
```

- Feature::isUnique

true

- TransitionUsage::isComposite

true

- Element::shortName

null

- Type::isAbstract

false

- Feature::isOrdered

false

- Element::aliasId

Set{}

- Feature::isPortion

false

- Feature::isReadOnly

false

- Feature::direction

null

- Element::elementId

Helper.getID(from)

- Element::name

null

- Feature::isDerived

false

#### **C.2.5.3.3.15 ActivityEdgeSourceEndFeature\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeature\_Mapping

##### **Mapping Source**

ActivityNode

##### **Mapping Target**

Feature

##### **Owned Mappings**

- activityEdgeSourceEndSubsetting : ActivityEdgeSourceEndSubsetting\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Element::aliasId

Set{}

- Feature::isEnd

true

- Type::isSufficient

false

- Feature::ownedRelationship

Set{activityEdgeSourceEndSubsetting.to}

- Element::name

null

- Element::shortName

null

- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()

#### C.2.5.3.3.16 ActivityEdgeInitialNodeSourceEndFeatureMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToEndFeatureMembership\_Mapping

##### Mapping Source

ActivityNode

##### Mapping Target

FeatureMembership

##### Owned Mappings

- activityEdgeSourceInitialNode : ActivityEdgeSourceInitialNode\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- FeatureMembership::ownedRelatedElement  
Set{self.ownedMemberFeature() }
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
null
- FeatureMembership::ownedMemberFeature  
activityEdgeSourceInitialNode.to
- Element::shortName  
null

- Element::elementId  
Helper.createUUID()
- Element::aliasId  
Set{}
- FeatureMembership::owningType  
*abstract rule*
- Membership::memberName  
null
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
KerML::VisibilityKind::public
- Element::name  
null
- Element::ownedRelationship  
Set{}

#### C.2.5.3.3.17 ActivityEdgeSourceInitialNode\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeature\_Mapping

##### Mapping Source

ActivityNode

##### Mapping Target

Feature

##### Owned Mappings

- activityEdgeSourceInitialNodeSubsetting : ActivityEdgeSourceInitialNodeSubsetting\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- `Element::aliasId`  
`Set{}`
- `Feature::isEnd`  
`true`
- `Type::isSufficient`  
`false`
- `Element::name`  
`null`
- `Element::shortName`  
`null`
- `Type::isAbstract`  
`false`
- `Element::elementId`  
`Helper.createUUID()`
- `Feature::ownedRelationship`  
`Set{activityEdgeSourceInitialNodeSubsetting.to}`

### C.2.5.3.3.18 ActivityEdgeSourceEndFeatureMembership\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

GenericToEndFeatureMembership\_Mapping

#### Mapping Source

ActivityNode

#### Mapping Target

FeatureMembership

#### Owned Mappings

- `activityEdgeSourceEndFeature` : `ActivityEdgeSourceEndFeature_Mapping`

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- FeatureMembership::ownedRelatedElement  
`Set { self.ownedMemberFeature () }`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set {}`
- FeatureMembership::owningType  
*abstract rule*
- FeatureMembership::ownedMemberFeature  
`activityEdgeSourceEndFeature.to`
- Membership::memberName  
`null`
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
`KerML::VisibilityKind::public`
- Element::name  
`null`
- Element::ownedRelationship  
`Set {}`

#### C.2.5.3.3.19 ActivityEdgeSourceInitialNodeSubsetting\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToSubsetting\_Mapping

##### Mapping Source

ActivityNode

##### Mapping Target

Subsetting

##### Owned Mappings

- activityEdgeSourceInitialNode : ActivityEdgeSourceInitialNode\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId  
`Set{}`
- Relationship::ownedRelatedElement  
`Set{}`
- Subsetting::subsettingFeature  
`activityEdgeSourceInitialNode.to`
- Subsetting::subsettingFeature  
`SysML2::ActionUsage.allInstances()->any(m | m.qualifiedName = 'Actions::Action::start')`
- Specialization::specific  
*abstract rule*
- Element::name  
`null`
- Element::shortName  
`null`



- Specialization::general  
*abstract rule*
- Element::elementId  
  
Helper.createUUID()
- Element::ownedRelationship  
  
Set{ }

#### C.2.5.3.3.20 ActivityEdgeSourceEndSubsetting\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToSubsetting\_Mapping

##### Mapping Source

ActivityNode

##### Mapping Target

Subsetting

##### Owned Mappings

- activityEdgeSourceEndFeature : ActivityEdgeSourceEndFeature\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId  
  
Set{ }
- Relationship::ownedRelatedElement  
  
Set{ }
- Specialization::specific  
*abstract rule*
- Subsetting::subsettingFeature  
  
from
- Element::name

null

- Element::shortName

null

- Subsetting::subsettingFeature

activityEdgeSourceEndFeature.to

- Specialization::general  
*abstract rule*

- Element::elementId

Helper.createUUID()

- Element::ownedRelationship

Set{}

#### **C.2.5.3.3.21 ControlFlowFinalNodeTargetEndFeatureMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToEndFeatureMembership\_Mapping

##### **Mapping Source**

ActivityNode

##### **Mapping Target**

FeatureMembership

##### **Owned Mappings**

- controlFlowTargetFinalNode : ControlFlowTargetFinalNode\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- FeatureMembership::ownedRelatedElement

Set{self.ownedMemberFeature() }

- Membership::membershipOwningNamespace  
*abstract rule*

- Membership::memberShortName  
null
- Element::shortName  
null
- Element::elementId  
Helper.createUUID()
- Element::aliasId  
Set{}
- FeatureMembership::owningType  
*abstract rule*
- Membership::memberName  
null
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- FeatureMembership::ownedMemberFeature  
controlFlowTargetFinalNode.to
- Membership::visibility  
KerML::VisibilityKind::public
- Element::name  
null
- Element::ownedRelationship  
Set{}

#### C.2.5.3.3.22 ControlFlowTargetFinalNodeSubsetting\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToSubsetting\_Mapping

##### Mapping Source

ActivityNode

##### Mapping Target

Subsetting

### Owned Mappings

- controlFlowTargetFinalNode : ControlFlowTargetFinalNode\_Mapping

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  
`Set {}`
- Relationship::ownedRelatedElement  
`Set {}`
- Specialization::specific  
*abstract rule*
- Subsetting::subsettingFeature  
`controlFlowTargetFinalNode.to`
- Element::name  
`null`
- Subsetting::subsettingFeature  
`SysML2::ActionUsage.allInstances()->any(m | m.qualifiedName = 'Actions::Action::done')`
- Element::shortName  
`null`
- Specialization::general  
*abstract rule*
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set {}`

#### C.2.5.3.3.23 ControlFlowSuccessionAsUsage\_Mapping

### Description

\*\*\* not specified yet \*\*\*

### General Mappings

NamedElementMain\_Mapping  
CommonActivityEdgeSuccessionAsUsage\_Mapping

### Mapping Source

ControlFlow

### Mapping Target

SuccessionAsUsage

### Owned Mappings

(none)

### Applicable filters

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

```
from.guard.oclIsUndefined()
```

### Mapping rules

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

- Relationship::ownedRelatedElement

```
Set{}
```

- Feature::isEnd

```
false
```

- Type::isSufficient

```
false
```

- Connector::isDirected

```
false
```

- SuccessionAsUsage::ownedRelationship

```
let relationships : Set(KerML::Relationship) = Set{  
  if from.source.oclIsKindOf(UML::InitialNode) then ActivityEdgeInitialNodeSourceEndFeatureMembershi  
  if from.target.oclIsKindOf(UML::FinalNode) then ControlFlowFinalNodeTargetEndFeatureMembershi  
  let relationshipsWithGuard : Set(KerML::Relationship) = if src.guard.oclIsUndefined() then re  
  if src.weight.oclIsUndefined() then relationshipsWithGuard else relationshipsWithGuard->inclu
```

- Feature::isUnique

```
true
```

- Element::shortName

```
null
```

- `Type::isAbstract`  
`false`
- `Feature::isOrdered`  
`false`
- `Relationship::target`  
`Set{}`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `Feature::isReadOnly`  
`false`
- `Relationship::source`  
`Set{}`
- `Feature::direction`  
`null`
- `Element::elementId`  
`Helper.getID(from)`
- `Element::name`  
`null`
- `Feature::isDerived`  
`false`
- `Feature::isComposite`  
`false`

#### **C.2.5.3.3.24 ControlFlowTargetFinalNode\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeature\_Mapping

##### **Mapping Source**

ActivityNode

## Mapping Target

Feature

## Owned Mappings

- controlFlowTargetFinalNodeSubsetting : ControlFlowTargetFinalNodeSubsetting\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Element::aliasId  
`Set{}`
- Type::isSufficient  
`false`
- Element::name  
`null`
- Feature::ownedRelationship  
`Set{controlFlowTargetFinalNodeSubsetting.to}`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isEnd  
`true`

### C.2.5.3.3.25 ControlFlowTargetEndFeature\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

GenericToFeature\_Mapping

### Mapping Source

ActivityNode

### Mapping Target

Feature

### Owned Mappings

- controlFlowTargetEndSubsetting : ControlFlowTargetEndSubsetting\_Mapping

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  
`Set{}`
- Type::isSufficient  
`false`
- Feature::ownedRelationship  
`Set{controlFlowTargetEndSubsetting.to}`
- Feature::isEnd  
`true`
- Element::name  
`null`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`

#### C.2.5.3.3.26 ControlFlowTargetEndFeatureMembership\_Mapping

### Description



\*\*\* not specified yet \*\*\*

## General Mappings

GenericToEndFeatureMembership\_Mapping

## Mapping Source

ActivityNode

## Mapping Target

FeatureMembership

## Owned Mappings

- controlFlowTargetEndFeature : ControlFlowTargetEndFeature\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- FeatureMembership::ownedRelatedElement  

```
Set{self.ownedMemberFeature() }
```
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  

```
null
```
- Element::shortName  

```
null
```
- Element::elementId  

```
Helper.createUUID()
```
- FeatureMembership::ownedMemberFeature  

```
controlFlowTargetEndFeature.to
```
- Element::aliasId  

```
Set{ }
```
- FeatureMembership::owningType  
*abstract rule*
- Membership::memberName

null

- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility

    KerML::VisibilityKind::public

- Element::name

    null

- Element::ownedRelationship

    Set{ }

#### **C.2.5.3.3.27 ControlFlowTargetEndSubsetting\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToSubsetting\_Mapping

##### **Mapping Source**

ActivityNode

##### **Mapping Target**

Subsetting

##### **Owned Mappings**

- controlFlowTargetEndFeature : ControlFlowTargetEndFeature\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Element::aliasId

    Set{ }

- Relationship::ownedRelatedElement

    Set{ }

- Specialization::specific  
*abstract rule*
- Subsetting::subsettingFeature  
  
from
- Element::name  
  
null
- Subsetting::subsettingFeature  
  
controlFlowTargetEndFeature.to
- Element::shortName  
  
null
- Specialization::general  
*abstract rule*
- Element::elementId  
  
Helper.createUUID()
- Element::ownedRelationship  
  
Set{}

#### **C.2.5.3.3.28 ControlFlowTransitionUsageFeatureMembership\_Mapping**

#### **C.2.5.3.3.29 ControlFlowTransitionUsageFeatureReferenceExpression\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureReferenceExpression\_Mapping

##### **Mapping Source**

ControlFlow

##### **Mapping Target**

FeatureReferenceExpression

##### **Owned Mappings**

- controlFlowTransitionUsageFeatureReferenceExpressionMembership :  
ControlFlowTransitionUsageFeatureReferenceExpressionMembership\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- `Feature::isEnd`  
`false`
- `FeatureReferenceExpression::ownedRelationship`  
`Set{controlFlowTransitionUsageFeatureReferenceExpressionMembership.to, CommonReturnParameter}`
- `Type::isSufficient`  
`false`
- `Feature::isUnique`  
`true`
- `Element::shortName`  
`null`
- `Type::isAbstract`  
`false`
- `Element::elementId`  
`Helper.createUUID()`
- `Feature::isOrdered`  
`false`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `Feature::isReadOnly`  
`false`
- `Feature::direction`  
`null`
- `Element::name`  
`null`
- `Feature::isDerived`  
`false`

- Feature::isComposite

false

### C.2.5.3.3.30 ControlFlowTransitionUsageFeatureReferenceExpressionMembership\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

GenericToMembership\_Mapping

#### Mapping Source

ControlFlow

#### Mapping Target

Membership

#### Owned Mappings

(none)

#### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

#### Mapping rules

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

- Element::aliasId  
Set{}
- Relationship::ownedRelatedElement  
Set{}
- Relationship::source  
Set{}
- Element::name  
null
- Element::shortName  
null
- Membership::memberElement  
from.guard

- Element::elementId  
Helper.createUUID()
- Relationship::target  
Set{ }
- Element::ownedRelationship  
Set{ }

#### **C.2.5.3.3.31 ActivityEdgeTransitionUsageSourceMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToMembership\_Mapping

##### **Mapping Source**

ActivityNode

##### **Mapping Target**

Membership

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Element::aliasId  
Set{ }
- Relationship::ownedRelatedElement  
Set{ }
- Relationship::source  
Set{ }
- Element::name  
null

- Element::shortName  
null
- Element::elementId  
Helper.createUUID()
- Relationship::target  
Set{}
- Membership::memberElement  
if from.ocIsTypeOf(UML::ActivityParameterNode) then from.parameter else from endif
- Element::ownedRelationship  
Set{}

### C.2.5.3.32 DecisionNode\_Mapping

#### Description

There is no suitable element in SysML v2 for the else condition of an outgoing SysMLv1::ActivityEdge. Therefore, it is mapped to a TextualRepresentation with language "SysML v1" and body "else" (see ExpressionElse\_Mapping class). The expected SysML v2 textual notation of a SysMLv1::DecisionNode is as follows

```
decide thisIsADecisionNode;
    succession flow1 first thisIsADecisionNode if {
        return : ScalarValues::Boolean;
        // guard expression, for example, opaque expression
    }.result then nextAction;
    succession flow2 first thisIsADecisionNode if {
        return : ScalarValues::Boolean;
        language "SysMLv1"
        /*
         * else
         */
    }.result then nextAction;
```

#### General Mappings

GenericToUsage\_Mapping  
NamedElementMain\_Mapping

#### Mapping Source

DecisionNode

#### Mapping Target

DecisionNode

#### Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Feature::isEnd  
`false`
- Element::ownedRelationship  
`ElementOwnership_Mapping.getMappedColl(from.ownedElement)`
- DecisionNode::isComposite  
`true`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Feature::isReadOnly  
`false`
- Feature::direction  
`null`
- Element::elementId



Helper.getID(from)

- Element::name

null

- Feature::isDerived

false

### C.2.5.3.3.33 ForkNode\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

GenericToUsage\_Mapping  
NamedElementMain\_Mapping

#### Mapping Source

ForkNode

#### Mapping Target

ForkNode

#### Owned Mappings

(none)

#### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

#### Mapping rules

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

- Feature::isEnd

false

- Element::ownedRelationship

ElementOwnership\_Mapping.getMappedColl(from.ownedElement)

- Type::isSufficient

false

- Feature::isUnique

true

- Element::shortName  
null
- Type::isAbstract  
false
- Feature::isOrdered  
false
- Element::aliasId  
Set{ }
- Feature::isPortion  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::elementId  
Helper.getID(from)
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false

#### **C.2.5.3.3.34 InitialNodeMembership\_Mapping**

##### **Description**

The InitialNode\_Mapping class creates a membership relationship to reference the action usage "start" from the system library. The mapping is called in the ownedRelationship() operation of the Activitiy\_Mapping class.

##### **General Mappings**

GenericToMembership\_Mapping

##### **Mapping Source**

InitialNode

## Mapping Target

Membership

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Element::aliasId  
`Set{}`
- Membership::memberName  
`if from.name = '' then null else from.name endif`
- Relationship::ownedRelatedElement  
`Set{}`
- Membership::memberElement  
`SysMLv2::ActionUsage.allInstances()->any(e | e.qualifiedName = 'Actions::Action::start')`
- Relationship::source  
`Set{}`
- Element::name  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Relationship::target  
`Set{}`
- Element::ownedRelationship  
`Set{}`

### C.2.5.3.3.35 JoinNode\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

GenericToUsage\_Mapping  
NamedElementMain\_Mapping

#### Mapping Source

JoinNode

#### Mapping Target

JoinNode

#### Owned Mappings

(none)

#### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

#### Mapping rules

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

- Feature::isEnd  
`false`
- Element::ownedRelationship  
`ElementOwnership_Mapping.getMappedColl(from.ownedElement)`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Feature::isOrdered

- `false`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `Feature::isReadOnly`  
`false`
- `Feature::direction`  
`null`
- `Element::elementId`  
`Helper.getID(from)`
- `Element::name`  
`null`
- `Feature::isDerived`  
`false`
- `Feature::isComposite`  
`false`

#### **C.2.5.3.3.36 MergeNode\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToUsage\_Mapping  
NamedElementMain\_Mapping

##### **Mapping Source**

MergeNode

##### **Mapping Target**

MergeNode

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Feature::isEnd  
`false`
- Element::ownedRelationship  
`ElementOwnership_Mapping.getMappedColl(from.ownedElement)`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Feature::isReadOnly  
`false`
- Feature::direction  
`null`
- Element::elementId  
`Helper.getID(from)`
- Element::name  
`null`

- Feature::isDerived

false

- Feature::isComposite

false

### C.2.5.3.3.37 ObjectFlow\_Mapping

#### Description

A UML4SysML::ObjectFlow is mapped to a SysMLv2::SuccessionFlowConnectionUsage. The expected SysML v2 textual syntax of a mapped object flow between two pins is as follows.

```
succession flow of1 of BlockA from action1.outputValue to action2.inputValue;

action action1 {
    out outputValue : BlockA;
}
action action2 {
    in inputValue : BlockA;
}
part def BlockA;
```

The mapping does not yet support the case where the source node of the object flow is not an object node.

#### General Mappings

GenericToConnector\_Mapping

#### Mapping Source

ObjectFlow

#### Mapping Target

SuccessionFlowConnectionUsage

#### Owned Mappings

(none)

#### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

#### Mapping rules

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

- Relationship::ownedRelatedElement

Set{}

- Feature::isEnd

```

    false

    • Type::isSufficient

        false

    • Feature::isUnique

        true

    • Element::shortName

        null

    • Type::isAbstract

        false

    • Element::elementId

        Helper.createUUID()

    • Feature::isOrdered

        false

    • Relationship::target

        Set{}

    • Element::aliasId

        Set{}

    • Feature::isPortion

        false

    • Feature::isReadOnly

        false

    • SuccessionFlowConnectionUsage::ownedRelationship

        if from.source.oclIsKindOf(UML::ObjectNode) then
        Set{ObjectFlowItemFeatureMembership_Mapping.getMapped(from), ObjectFlowEndFeatureMembership_M
        else Set{ObjectFlowEndFeatureMembership_Mapping.getMapped(from.source), ObjectFlowEndFeatureM

    • Relationship::source

        Set{}

    • Feature::direction

        null

    • Feature::isDerived

        false

```



- SuccessionFlowConnectionUsage::name

`from.name`

- Feature::isComposite

`false`

#### **C.2.5.3.38 ObjectFlowFeatureMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureMembership\_Mapping

##### **Mapping Source**

ObjectFlow

##### **Mapping Target**

FeatureMembership

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Membership::membershipOwningNamespace

*abstract rule*

- Membership::memberShortName

`null`

- Element::shortName

`null`

- FeatureMembership::ownedMemberFeature

`ObjectFlow_Mapping.getMapped(from)`

- Element::elementId

`Helper.createUUID()`

- Element::aliasId  
`Set{}`
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
`null`
- OwningMembership::ownedRelatedElement  
`Set{self.ownedMemberElement() }`
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
`KerML::VisibilityKind::public`
- Element::name  
`null`
- Element::ownedRelationship  
`Set{}`

#### **C.2.5.3.3.39 ObjectFlowGuardFeatureMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureMembership\_Mapping

##### **Mapping Source**

ObjectFlow

##### **Mapping Target**

FeatureMembership

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- FeatureMembership::ownedMemberFeature  
  
ObjectFlowGuard\_Mapping.getMapped(from)
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
  
null
- OwningMembership::ownedRelatedElement  
  
Set{self.ownedMemberElement() }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- Element::ownedRelationship  
  
Set{ }

### C.2.5.3.3.40 ObjectFlowGuard\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

## General Mappings

GenericToUsage\_Mapping

## Mapping Source

ObjectFlow

## Mapping Target

TransitionUsage

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- TransitionUsage::ownedRelationship  

```
Set{ActivityEdgeTransitionUsageSourceMembership_Mapping.getMapped(from.source),  
CommonParameterReferenceUsageInMembership_Mapping.getMapped(from.source),  
ObjectFlowGuardSuccessionTargetEndFeatureMembership_Mapping.getMapped(from),  
CommonActivityEdgeSuccessionAsUsage_Mapping.getMapped(from),  
CommonReturnParameterReferenceUsageMembership_Mapping.getMapped(from)  
}
```
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`

- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false

#### **C.2.5.3.3.41 ObjectFlowGuardSuccessionTargetEndFeature\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeature\_Mapping

##### **Mapping Source**

ObjectFlow

##### **Mapping Target**

Feature

##### **Owned Mappings**

- objectFlowGuardSuccessionTargetEndSubsetting :  
ObjectFlowGuardSuccessionTargetEndSubsetting\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  
`Set{}`
- Feature::isEnd  
`true`
- Type::isSufficient  
`false`
- Feature::ownedRelationship  
`Set{objectFlowGuardSuccessionTargetEndSubsetting.to}`
- Element::name  
`null`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`

#### C.2.5.3.3.42 ObjectFlowGuardSuccessionTargetEndFeatureMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToEndFeatureMembership\_Mapping

##### Mapping Source

ObjectFlow

##### Mapping Target

FeatureMembership

## Owned Mappings

- objectFlowGuardSuccessionTargetEndFeature : ObjectFlowGuardSuccessionTargetEndFeature\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- FeatureMembership::ownedRelatedElement  
`Set{self.ownedMemberFeature() }`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{ }`
- FeatureMembership::owningType  
*abstract rule*
- Membership::memberName  
`null`
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
`KerML::VisibilityKind::public`
- Element::name  
`null`
- FeatureMembership::ownedMemberFeature  
`objectFlowGuardSuccessionTargetEndFeature.to`
- Element::ownedRelationship

Set{ }

#### C.2.5.3.3.43 ObjectFlowGuardSuccessionTargetEndSubsetting\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToSubsetting\_Mapping

##### Mapping Source

ObjectFlow

##### Mapping Target

Subsetting

##### Owned Mappings

- objectFlowGuardSuccessionTargetEndFeature : ObjectFlowGuardSuccessionTargetEndFeature\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId  
Set{ }
- Relationship::ownedRelatedElement  
Set{ }
- Specialization::specific  
*abstract rule*
- Subsetting::subsettingFeature  
objectFlowGuardSuccessionTargetEndFeature.to
- Element::name  
null
- Element::shortName  
null
- Specialization::general  
*abstract rule*



- Subsetting::subsettingFeature  
ObjectFlow\_Mapping.getMapped(from)
- Element::elementId  
Helper.createUUID()
- Element::ownedRelationship  
Set{ }

#### C.2.5.3.3.44 ObjectFlowItemFeature\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

ObjectFlowItemFeatureUntyped\_Mapping

##### Mapping Source

ObjectNode

##### Mapping Target

ItemFeature

##### Owned Mappings

- objectFlowItemFeatureTyping : ObjectFlowItemFeatureTyping\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Feature::isEnd  
false
- Type::isSufficient  
false
- ItemFeature::ownedRelationship  
Set{objectFlowItemFeatureTyping.to}
- Feature::isUnique  
true

- Element::shortName  
null
- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false

#### **C.2.5.3.3.45 ObjectFlowItemFeatureMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureMembership\_Mapping

##### **Mapping Source**

ObjectFlow

##### **Mapping Target**

FeatureMembership

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- FeatureMembership::ownedMemberFeature

```
if from.source.type.oclIsUndefined() then ObjectFlowItemFeatureUntyped_Mapping.getMapped(from)
```

- Membership::membershipOwningNamespace

*abstract rule*

- Membership::memberShortName

null

- Element::shortName

null

- Element::elementId

```
Helper.createUUID()
```

- Element::aliasId

```
Set{}
```

- OwningMembership::ownedMemberElement

*abstract rule*

- Membership::memberName

null

- OwningMembership::ownedRelatedElement

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

- TypeFeaturing::featureOfType

*abstract rule*

- TypeFeaturing::featuringType

*abstract rule*

- Membership::visibility

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

- Element::name

null

- Element::ownedRelationship

Set{ }

#### C.2.5.3.3.46 ObjectFlowItemFeatureTyping\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

TypedElementToFeatureTyping\_Mapping

##### Mapping Source

ObjectNode

##### Mapping Target

FeatureTyping

##### Owned Mappings

- objectFlowItemFeature : ObjectFlowItemFeature\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId

Set{ }

- Relationship::ownedRelatedElement

Set{ }

- FeatureTyping::typedFeature

objectFlowItemFeature.to

- Element::name

null

- Element::shortName

null

- Element::elementId

Helper.createUUID()

- FeatureTyping::type  
*abstract rule*
- FeatureTyping::typedFeature  
*abstract rule*
- Element::ownedRelationship

Set { }

#### **C.2.5.3.3.47 ObjectFlowItemFeatureUntyped\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeature\_Mapping

##### **Mapping Source**

ObjectNode

##### **Mapping Target**

ItemFeature

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Element::aliasId  
Set { }
- Type::isSufficient  
false
- Element::name  
null
- Element::shortName  
null
- Type::isAbstract

false

- Element::elementId

Helper.createUUID()

- Element::ownedRelationship

Set{}

#### C.2.5.3.3.48 ObjectFlowEndFeatureMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToEndFeatureMembership\_Mapping

##### Mapping Source

ActivityNode

##### Mapping Target

FeatureMembership

##### Owned Mappings

(none)

##### Applicable filters

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

(none)

##### Mapping rules

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

- FeatureMembership::ownedRelatedElement

Set{self.ownedMemberFeature() }

- Membership::membershipOwningNamespace

*abstract rule*

- Membership::memberShortName

null

- FeatureMembership::ownedMemberFeature

ObjectFlowItemFlowEnd\_Mapping.getMapped(from)

- Element::shortName

- `Element::elementId`

- `Element::aliasId`

- FeatureMembership::owningType

- Membership::memberName

- `TypeFeaturing::featureOfType`

- `TypeFeaturing::featuringType`

- Membership::visibility

- Element::name

- `Element::ownedRelationship`

#### C.2.5.3.3.49 ObjectFlowItemFlowEnd\_Mapping

\*\*\* not specified yet \*\*\*

## GenericToFeature\_Mapping

## ActivityNode

## ItemFlowEnd

- objectFlowItemFlowSubsetting : ObjectFlowItemFlowSubsetting\_Mapping

**OMG Systems Modeling Language (SysML) v2.0, Submission**

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId

`Set{}`

- Type::isSufficient

`false`

- ItemFlowEnd::ownedRelationship

`Set{objectFlowItemFlowSubsetting.to, ObjectFlowItemFlowFeatureMembership_Mapping.getMapped(f`

- Element::name

`null`

- Element::shortName

`null`

- Type::isAbstract

`false`

- Element::elementId

`Helper.createUUID()`

#### C.2.5.3.3.50 ObjectFlowItemFlowFeature\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeature\_Mapping

##### Mapping Source

ActivityNode

##### Mapping Target

ItemFlowFeature

##### Owned Mappings

(none)

##### Applicable filters



This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  
`Set{}`
- ItemFlowFeature::ownedRelationship  
`Set{ObjectFlowItemFlowRedefinition_Mapping.getMapped(from)}`
- Type::isSufficient  
`false`
- Element::name  
`null`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`

#### C.2.5.3.3.51 ObjectFlowItemFlowFeatureMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureMembership\_Mapping

##### Mapping Source

ActivityNode

##### Mapping Target

FeatureMembership

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- FeatureMembership::ownedMemberFeature  
`ObjectFlowItemFlowFeature_Mapping.getMapped(from)`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{}`
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
`null`
- OwningMembership::ownedRelatedElement  
`Set{self.ownedMemberElement() }`
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
`KerML::VisibilityKind::public`
- Element::name  
`null`
- Element::ownedRelationship  
`Set{}`

#### C.2.5.3.3.52 ObjectFlowItemFlowRedefinition\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

## General Mappings

GenericToRedefinition\_Mapping

## Mapping Source

ActivityNode

## Mapping Target

Redefinition

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Element::aliasId  
`Set {}`
- Subsetting::ownedRelatedElement  
`Set {}`
- Subsetting::subsettingFeature  
*abstract rule*
- Element::name  
`null`
- Subsetting::subsettingFeature  
*abstract rule*
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set {}`

#### C.2.5.3.3.53 ObjectFlowItemFlowSubsetting\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToSubsetting\_Mapping

##### Mapping Source

ActivityNode

##### Mapping Target

Subsetting

##### Owned Mappings

- objectFlowItemFlowEnd : ObjectFlowItemFlowEnd\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId  
`Set{ }`
- Relationship::ownedRelatedElement  
`Set{ }`
- Specialization::specific  
*abstract rule*
- Subsetting::subsettingFeature  
`objectFlowSourceItemFlowEnd.to`
- Element::name  
`null`
- Element::shortName  
`null`
- Specialization::general  
*abstract rule*
- Element::elementId

```
Helper.createUUID()
```

- Subsetting::subsettingFeature

```
if from.ocIsKindOf(UML::ActivityParameterNode) then Parameter_Mapping.getMapped(from.parameter)
else if from.ocIsKindOf(UML::Pin) then CommonAction_Mapping.getMapped(from.owner)
else if from.ocIsKindOf(UML::InitialNode) then SysMLv2::ActionUsage.allInstances()->any(e |
else if from.ocIsKindOf(UML::ActivityFinalNode) then SysMLv2::ActionUsage.allInstances()->any(e |
else from endif endif endif endif
```

- Element::ownedRelationship

```
Set{}
```

#### C.2.5.3.3.54 VariableAttribute\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

NamedElementMain\_Mapping  
CommonVariable\_Mapping

##### Mapping Source

Variable

##### Mapping Target

AttributeUsage

##### Owned Mappings

(none)

##### Applicable filters

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

```
from.type.ocIsKindOf(UML::DataType)
```

##### Mapping rules

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

- Feature::isOrdered

```
from.isOrdered
```

- Type::isSufficient

```
false
```

- Feature::isComposite

```
from.isComposite
```

- **Feature::ownedRelationship**

```
let typing: KerML::FeatureTyping = StructuralFeatureToFeatureTyping_Mapping.getMapped(from)
let subsetting: Set(KerML::Subsetting) = from.subsettedProperty->collect(p | PropertySubsetting)
let subsettingMultiplicityTyping: Set(KerML::Relationship) = subsetting->union(if typing.ocll
    Set{MultiplicityMembership_Mapping.getMapped(from)}
else
    Set{MultiplicityMembership_Mapping.getMapped(from), typing}
endif)->asSet() in

let relationships: Set(KerML::Relationship) = if from.defaultValue.ocllIsTypeOf(UML::OpaqueExp

if from.defaultValue.ocllIsUndefined() then
    relationships
else
    relationships->including(if from.defaultValue.ocllIsTypeOf(UML::OpaqueExpression) then Pro
endif
```

- **Feature::isAbstract**

```
false
```

- **Feature::isEnd**

```
if from.association.ocllIsUndefined() then
    false
else
    from.association.ownedEnd->includes(from)
endif
```

- **Element::shortName**

```
null
```

- **Element::aliasId**

```
Set{}
```

- **Feature::isPortion**

```
false
```

- **Feature::isDerived**

```
from.isDerived
```

- **Feature::direction**

```
null
```

- **Element::elementId**

```
Helper.getID(from)
```

- **Element::name**

```
null
```

- Feature::isUnique

`from.isUnique`

- Feature::isReadOnly

*abstract rule*

#### **C.2.5.3.3.55 VariableFeatureTyping\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

TypedElementToFeatureTyping\_Mapping

##### **Mapping Source**

Variable

##### **Mapping Target**

FeatureTyping

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- Element::aliasId

`Set{ }`

- Relationship::ownedRelatedElement

`Set{ }`

- Element::name

`null`

- Element::shortName

`null`

- Element::elementId

`Helper.createUUID()`

- FeatureTyping::type  
*abstract rule*
- FeatureTyping::typedFeature  
*abstract rule*
- Element::ownedRelationship

Set{}

#### C.2.5.3.3.56 VariableItem\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

NamedElementMain\_Mapping

CommonVariable\_Mapping

##### Mapping Source

Variable

##### Mapping Target

ItemUsage

##### Owned Mappings

(none)

##### Applicable filters

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

```
not from.type.ocIsKindOf(UML::DataType)
```

##### Mapping rules

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

- Feature::isOrdered  
`from.isOrdered`
- Type::isSufficient  
`false`
- Feature::isComposite  
`from.isComposite`
- Feature::ownedRelationship

```
let typing: KerML::FeatureTyping = StructuralFeatureToFeatureTyping_Mapping.getMapped(from) i
```



```

let subsetting: Set(KerML::Subsetting) = from.subsettedProperty->collect(p | PropertySubsetting)
let subsettingMultiplicityTyping: Set(KerML::Relationship) = subsetting->union(if typing.oclc
    Set{MultiplicityMembership_Mapping.getMapped(from)}
else
    Set{MultiplicityMembership_Mapping.getMapped(from), typing}
endif)->asSet() in

let relationships: Set(KerML::Relationship) = if from.defaultValue.oclcIsTypeOf(UML::OpaqueExp

if from.defaultValue.oclcIsUndefined() then
    relationships
else
    relationships->including(if from.defaultValue.oclcIsTypeOf(UML::OpaqueExpression) then Pro
endif

```

- **Feature::isAbstract**

```

false

```

- **Feature::isEnd**

```

if from.association.oclcIsUndefined() then
    false
else
    from.association.ownedEnd->includes(from)
endif

```

- **Element::shortName**

```

null

```

- **Element::aliasId**

```

Set{}

```

- **Feature::isPortion**

```

false

```

- **Feature::isDerived**

```

from.isDerived

```

- **Feature::direction**

```

null

```

- **Element::elementId**

```

Helper.getID(from)

```

- **Element::name**

```

null

```

- **Feature::isUnique**

```

from.isUnique

```

- Feature::isReadOnly  
*abstract rule*

### C.2.5.3.3.57 VariableMembership\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

ElementFeatureMembership\_Mapping

#### Mapping Source

Variable

#### Mapping Target

FeatureMembership

#### Owned Mappings

(none)

#### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

#### Mapping rules

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

- FeatureMembership::ownedRelatedElement  
`Set { self.ownedMemberFeature() }`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- FeatureMembership::visibility  
`KerML::VisibilityKind::private`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId

Set{}

- FeatureMembership::ownedMemberFeature  
*abstract rule*
- FeatureMembership::owningType  
*abstract rule*
- Membership::memberName

null

- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility

KerML::VisibilityKind::public

- Element::name

null

- Element::ownedRelationship

Set{}

#### C.2.5.4 Classification

##### C.2.5.4.1 Overview

**Table 17. List of all Overview Mapping Specifications**

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
BehavioralFeature	Usage	BehavioralFeature_Mapping	
Classifier	ObjectiveMembership FeatureTyping SubjectMembership PartUsage RequirementUsage ReferenceUsage Classifier StakeholderMembership	CaseObjectiveMembership_Mapping CaseSubjectFeatureTyping_Mapping CaseSubjectMembership_Mapping StakeholderPartUsage_Mapping CaseObjectiveRequirementUsage_Mapping CaseEmptySubjectReferenceUsage_Mapping Classifier_Mapping StakeholderMembership_Mapping	
Feature	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	
Generalization	Subclassification	Generalization_Mapping	

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
GeneralizationSet	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	
InstanceSpecification	PartUsage Membership FeatureTyping ConnectionUsage	InstanceSpecification_Mapping InstanceValueInstanceSpecification_Mapping InstanceSpecificationFeatureTyping_Mapping InstanceSpecificationLink_Mapping	InstanceSpecification.classifier->select( c   c.oclIsTypeOf(UML::Association))->size() = 0 InstanceSpecification.classifier->select( c   c.oclIsTypeOf(UML::Association))->size() > 0
InstanceValue	FeatureReferenceExpression	InstanceValue_Mapping	
Operation	PerformActionUsage	Operation_Mapping	
Parameter	FeatureValue FeatureTyping FeatureReferenceExpression ParameterMembership Membership ReferenceUsage ReferenceUsage	ParameterSetParameterReferenceUsageFeatureValue_Mapping ParameterToFeatureTyping_Mapping ParameterSetParameterReferenceUsageFeatureValueExpression_Mapping ParameterMembership_Mapping ParameterSetParameterReferenceUsageFeatureValueExpressionMembership_Mapping Parameter_Mapping ParameterSetParameterReferenceUsage_Mapping	not src.type.oclIsUndefined() and not src.type.oclIsKindOf(UML::Enumeration) and Helper.getSysMLv2EnumerationDefinition(src.type)
ParameterSet	ReferenceUsage FeatureMembership FeatureMembership	ParameterSet_Mapping ParameterSetMembership_Mapping ParameterSetParameterFeatureMembership_Mapping	



#### C.2.5.4.2.1 BehavioralFeature\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToUsage\_Mapping  
Namespace\_Mapping

##### Mapping Source

BehavioralFeature

##### Mapping Target

Usage

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Feature::isEnd  
`false`
- Element::ownedRelationship  
`ElementOwnership_Mapping.getMappedColl(from.ownedElement)`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::name  
`from.name`
- Element::shortName  
`null`
- Type::isAbstract

- false
- Feature::isOrdered
  - false
- Element::aliasId
  - Set{}
- Feature::isPortion
  - false
- Feature::isReadOnly
  - false
- Feature::direction
  - null
- Element::elementId
  - Helper.getID(from)
- Feature::isDerived
  - false
- Feature::isComposite
  - false

#### **C.2.5.4.2.2 Classifier\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToClassifier\_Mapping  
Namespace\_Mapping

##### **Mapping Source**

Classifier

##### **Mapping Target**

Classifier

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId

```
Set{}
```

- Classifier::isAbstract

```
from.isAbstract
```

- Element::ownedRelationship

```
ElementOwnership_Mapping.getMappedColl(from.ownedElement)
```

- Type::isSufficient

```
false
```

- Element::elementId

```
Helper.getID(from)
```

- Classifier::ownedRelationship

```
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.ocIsKindOf(UML::Generalization))
let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::FeatureMembership))
let toElementOMS: Set(UML::Element) = (from.ownedElement - toElementFMS) - generalizations
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))
```

- Element::name

```
from.name
```

- Element::shortName

```
null
```

#### C.2.5.4.2.3 DefaultLowerBound\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToExpression\_Mapping

##### Mapping Source

Element



## Mapping Target

LiteralInteger

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Feature::isEnd

false

- LiteralInteger::ownedRelationship

```
let ownerships: Set(SYSML2::Relationship) = ElementOwnership_Mapping.getMappedColl(from.ownedRelationships)
->including(CommonReturnParameterFeatureMembership_Mapping.getMapped(from)) in
```

ownerships

- Type::isSufficient

false

- Feature::isUnique

true

- Element::shortName

null

- Type::isAbstract

false

- Element::elementId

Helper.createUUID()

- Feature::isOrdered

false

- Element::aliasId

Set{}

- Feature::isPortion

- false
- Feature::isReadOnly
 

false
- LiteralInteger::value
 

1
- Feature::direction
 

null
- Element::name
 

null
- Feature::isDerived
 

false
- Feature::isComposite
 

false

#### C.2.5.4.2.4 DefaultLowerBoundTyping\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

\_DefaultMultiplicityBoundTyping\_Mapping

##### Mapping Source

Element

##### Mapping Target

FeatureTyping

##### Owned Mappings

- defaultMultiplicityLowerBoundValue : \_DefaultMultiplicityLowerBoundValue\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId

- Set {}
- Relationship::ownedRelatedElement
  - Set {}
- FeatureTyping::typedFeature
  - self.defaultMultiplicityLowerBoundValue.to
- Element::name
  - null
- Element::shortName
  - null
- Element::elementId
  - Helper.createUUID()
- FeatureTyping::type
  - abstract rule*
- FeatureTyping::typedFeature
  - abstract rule*
- Element::ownedRelationship
  - Set {}

#### C.2.5.4.2.5 DefaultMultiplicityBoundOwnership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### 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:

(none)

## Mapping rules

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

- Membership::membershipOwningNamespace

*abstract rule*

- Membership::memberShortName

null

- Element::shortName

null

- Element::elementId

Helper.createUUID()

- Element::aliasId

Set{}

- OwningMembership::ownedMemberElement

*abstract rule*

- Membership::memberName

null

- OwningMembership::ownedRelatedElement

Set{self.ownedMemberElement() }

- TypeFeaturing::featureOfType

*abstract rule*

- TypeFeaturing::featuringType

*abstract rule*

- FeatureMembership::isComposite

true

- Membership::visibility

KerML::VisibilityKind::public

- Element::name

null

- Element::ownedRelationship

Set{}

### C.2.5.4.2.6 DefaultMultiplicityBoundTyping\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

## 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:  
(none)

## Mapping rules

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

- Element::aliasId  
`Set{}`
- Relationship::ownedRelatedElement  
`Set{}`
- Specialization::specific  
*abstract rule*
- FeatureTyping::typedFeature  
*abstract rule*
- Element::name  
`null`
- FeatureTyping::type  
`Helper.getScalarValueTypeByName('Integer')`
- Element::shortName  
`null`
- Specialization::general  
*abstract rule*
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship

Set{ }

#### C.2.5.4.2.7 DefaultMultiplicityBoundValue\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToExpression\_Mapping

##### Mapping Source

Element

##### Mapping Target

LiteralInteger

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Feature::isEnd  
false
- LiteralInteger::value  
1
- Type::isSufficient  
false
- Feature::isUnique  
true
- Element::shortName  
null
- Type::isAbstract  
false
- Element::elementId

Helper.createUUID()

- Feature::isOrdered

false

- Element::aliasId

Set{}

- Feature::isPortion

false

- Feature::isReadOnly

false

- Feature::direction

null

- Element::name

null

- Feature::isDerived

false

- Feature::isComposite

false

- Element::ownedRelationship

Set{}

#### **C.2.5.4.2.8 DefaultMultiplicityElement\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeature\_Mapping

##### **Mapping Source**

Element

##### **Mapping Target**

MultiplicityRange

##### **Owned Mappings**

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId

Set{}

- Type::isSufficient

false

- MultiplicityRange::isUnique

true

- MultiplicityRange::ownedRelationship

OrderedSet{DefaultMultiplicityLowerBoundOwnership\_Mapping.getMapped(from), DefaultMultiplicityLowerBoundOwnership\_Mapping.getMapped(to)}

- Element::shortName

null

- Type::isAbstract

false

- MultiplicityRange::name

'defaultMultiplicity'

- Element::elementId

Helper.createUUID()

#### C.2.5.4.2.9 DefaultMultiplicityLowerBoundOwnership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

DefaultMultiplicityBoundOwnership\_Mapping

##### Mapping Source

Element

##### Mapping Target

FeatureMembership



## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- FeatureMembership::ownedRelatedElement  
`Set{self.ownedMemberFeature() }`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{ }`
- FeatureMembership::owningType  
*abstract rule*
- Membership::memberName  
`null`
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
`KerML::VisibilityKind::public`
- Element::name  
`null`
- FeatureMembership::ownedMemberFeature  
`DefaultLowerBound_Mapping.getMapped(from)`
- Element::ownedRelationship

Set { }

#### C.2.5.4.2.10 DefaultMultiplicityLowerBoundValue\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

\_DefaultMultiplicityBoundValue\_Mapping

##### Mapping Source

Element

##### Mapping Target

LiteralInteger

##### Owned Mappings

- defaultLowerBoundTyping : \_DefaultLowerBoundTyping\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Feature::isEnd  
false
- Type::isSufficient  
false
- Feature::isUnique  
true
- Element::shortName  
null
- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Feature::isOrdered

- `LiteralInteger::ownedRelationship`  
`Set{self.defaultLowerBoundTyping.to}`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `Feature::isReadOnly`  
`false`
- `Feature::direction`  
`null`
- `Feature::isDerived`  
`false`
- `Feature::isComposite`  
`false`
- `LiteralInteger::name`  
`'lowerBound'`

### Description

## General Mappings

## Mapping Source

## Mapping Target

## Owned Mappings

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Relationship::ownedRelatedElement  
`Set{}`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Membership::memberElement  
*abstract rule*
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{}`
- OwningMembership::ownedMemberElement  
`DefaultMultiplicityElement_Mapping.getMapped(from)`
- Membership::memberName  
`null`
- Membership::visibility  
`KerML::VisibilityKind::public`
- Element::name  
`null`
- Element::ownedRelationship  
`Set{}`

#### C.2.5.4.2.12 DefaultMultiplicityUpperBoundOwnership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

DefaultMultiplicityBoundOwnership\_Mapping

### Mapping Source

Element

### Mapping Target

FeatureMembership

### Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- FeatureMembership::ownedRelatedElement  
`Set{self.ownedMemberFeature() }`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- FeatureMembership::ownedMemberFeature  
`DefaultUpperBound_Mapping.getMapped(from)`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{ }`
- FeatureMembership::owningType  
*abstract rule*
- Membership::memberName  
`null`
- TypeFeaturing::featureOfType  
*abstract rule*

- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- Element::ownedRelationship  
  
Set{ }

#### C.2.5.4.2.13 DefaultMultiplicityUpperBoundValue\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

\_DefaultMultiplicityBoundValue\_Mapping

##### Mapping Source

Element

##### Mapping Target

LiteralInteger

##### Owned Mappings

- defaultUpperBoundTyping : \_DefaultUpperBoundTyping\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Feature::isEnd  
  
false
- Type::isSufficient  
  
false
- LiteralInteger::ownedRelationship  
  
Set{self.defaultUpperBoundTyping.to}

- Feature::isUnique  
true
- Element::shortName  
null
- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Feature::isReadOnly  
false
- LiteralInteger::name  
'upperBound'
- Feature::direction  
null
- Feature::isDerived  
false
- Feature::isComposite  
false

#### **C.2.5.4.2.14 DefaultUpperBound\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToExpression\_Mapping

##### **Mapping Source**

Element

## Mapping Target

LiteralInteger

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- LiteralInteger::value  
`1`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`



- LiteralInteger::ownedRelationship

```
let ownerships: Set(SYSML2::Relationship) = ElementOwnership_Mapping.getMappedColl(from.ownedRelationships
->including(CommonReturnParameterFeatureMembership_Mapping.getMapped(from)) in
```

```
ownerships
```

- Feature::isReadOnly

```
false
```

- Feature::direction

```
null
```

- Element::name

```
null
```

- Feature::isDerived

```
false
```

- Feature::isComposite

```
false
```

#### **C.2.5.4.2.15 DefaultUpperBoundTyping\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

\_DefaultMultiplicityBoundTyping\_Mapping

##### **Mapping Source**

Element

##### **Mapping Target**

FeatureTyping

##### **Owned Mappings**

- defaultMultiplicityUpperBoundValue : \_DefaultMultiplicityUpperBoundValue\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- `Element::aliasId`  
`Set{}`
- `Relationship::ownedRelatedElement`  
`Set{}`
- `FeatureTyping::typedFeature`  
`self.defaultMultiplicityUpperBoundValue.to`
- `Element::name`  
`null`
- `Element::shortName`  
`null`
- `Element::elementId`  
`Helper.createUUID()`
- `FeatureTyping::type`  
*abstract rule*
- `FeatureTyping::typedFeature`  
*abstract rule*
- `Element::ownedRelationship`  
`Set{}`

#### **C.2.5.4.2.16 ElementFeatureMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureMembership\_Mapping

##### **Mapping Source**

NamedElement

##### **Mapping Target**

FeatureMembership

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- FeatureMembership::ownedMemberFeature  
  
NamedElementMain\_Mapping.getMapped(from)
- Element::aliasId  
  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- FeatureMembership::visibility  
  
Helper.getKerMLVisibilityKind(from.oclAsType(UML::NamedElement).visibility)
- Membership::memberName  
  
null
- OwningMembership::ownedRelatedElement  
  
Set{self.ownedMemberElement() }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Element::name  
  
null
- Element::ownedRelationship  
  
Set{}

#### C.2.5.4.2.17 Generalization\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

## General Mappings

GenericToSpecialization\_Mapping  
ElementMain\_Mapping

## Mapping Source

Generalization

## Mapping Target

Subclassification

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Element::aliasId  
`Set{}`
- Relationship::ownedRelatedElement  
`Set{}`
- Relationship::source  
`Set{}`
- Subclassification::subclassifier  
`Classifier_Mapping.getMapped(from.specific)`
- Element::name  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Subclassification::superclassifier

```

    if from.general.oclIsTypeOf(UML::PrimitiveType) and not (Helper.getScalarValueType(from.general)
        Helper.getScalarValueType(from.general)
    else
        Classifier_Mapping.getMapped(from.general)
    endif

```

- Relationship::target

```
Set{}
```

- Element::ownedRelationship

```
Set{}
```

#### C.2.5.4.2.18 InstanceSpecificationLink\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

NamedElementMain\_Mapping  
GenericToConnectionUsage\_Mapping

##### Mapping Source

InstanceSpecification

##### Mapping Target

ConnectionUsage

##### Owned Mappings

(none)

##### Applicable filters

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

```
from.classifier->select( c | c.oclIsTypeOf(UML::Association))->size() > 0
```

##### Mapping rules

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

- Feature::isEnd

```
false
```

- Type::isSufficient

```
false
```

- ConnectionUsage::ownedRelationship

```
SlotMembership_Mapping.getMappedColl(from.slot)
->union(from.classifier->collect(g | InstanceSpecificationFeatureTyping_Mapping.getMapped(fro
```

- Feature::isUnique

```
true
```

- Element::shortName

```
null
```

- Type::isAbstract

```
false
```

- Feature::isOrdered

```
false
```

- Element::aliasId

```
Set{}
```

- Feature::isPortion

```
false
```

- Usage::isVariation

```
false
```

- Feature::isReadOnly

```
false
```

- Feature::direction

```
null
```

- Element::elementId

```
Helper.getID(from)
```

- Element::name

```
null
```

- Feature::isDerived

```
false
```

- Feature::isComposite

```
false
```

#### C.2.5.4.2.19 InstanceSpecification\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

## General Mappings

NamedElementMain\_Mapping  
GenericToPartUsage\_Mapping

## Mapping Source

InstanceSpecification

## Mapping Target

PartUsage

## Owned Mappings

(none)

## Applicable filters

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

```
from.classifier->select( c | c.ocIsTypeOf(UML::Association))->size() = 0
```

## Mapping rules

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

- Feature::isEnd

false

- Type::isSufficient

false

- PartUsage::ownedFeatureMembership

```
from.classifier->collect( c | InstanceSpecificationToGeneralization_Mapping.getMapped(from, c
```

- Feature::isUnique

true

- PartUsage::ownedRelationship

```
SlotMembership_Mapping.getMappedColl(from.slot)  
->union(from.classifier->collect(g | InstanceSpecificationFeatureTyping_Mapping.getMapped(fro
```

- Element::shortName

null

- Type::isAbstract

false

- Feature::isOrdered

- false
- Element::aliasId
  - Set{}
- Feature::isPortion
  - false
- Usage::isVariation
  - false
- Feature::isReadOnly
  - false
- Feature::direction
  - null
- Element::elementId
  - Helper.getID(from)
- Element::name
  - null
- Feature::isDerived
  - false
- Feature::isComposite
  - false

#### **C.2.5.4.2.20 InstanceSpecificationFeatureTyping\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureTyping\_Mapping

##### **Mapping Source**

InstanceSpecification

##### **Mapping Target**

FeatureTyping with qualifier: classifier:Classifier

##### **Owned Mappings**

(none)



### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  
`Set{}`
- Relationship::ownedRelatedElement  
`Set{}`
- Specialization::specific  
*abstract rule*
- FeatureTyping::typedFeature  
`InstanceSpecification_Mapping.getMapped(from)`
- Element::name  
`null`
- Element::shortName  
`null`
- Specialization::general  
*abstract rule*
- Element::elementId  
`Helper.createUUID()`
- FeatureTyping::type  
`Classifier_Mapping.getMapped(classifier)`
- Element::ownedRelationship  
`Set{}`

#### C.2.5.4.2.21 InstanceValue\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

ValueSpecification\_Mapping

##### Mapping Source

InstanceValue

## Mapping Target

FeatureReferenceExpression

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Feature::isEnd  
`false`
- Element::ownedRelationship  
`ElementOwnership_Mapping.getMappedColl(from.ownedElement)`
- Type::isSufficient  
`false`
- FeatureReferenceExpression::ownedRelationship  
`ElementOwnership_Mapping.getMappedColl(from.ownedElement)  
->including(InstanceValueInstanceSpecification_Mapping.getMapped(from.instance))`
- Feature::isUnique  
`true`
- Element::name  
`from.name`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{ }`

- Feature::isPortion  
false
- Feature::isReadOnly  
false
- Expression::ownedRelationship  
ElementOwnership\_Mapping.getMappedColl(from.ownedElement)->append(CommonReturnParameterFeatu
- Feature::direction  
null
- Element::elementId  
Helper.getID(from)
- Feature::isDerived  
false
- Feature::isComposite  
false

#### C.2.5.4.2.22 InstanceValueInstanceSpecification\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToMembership\_Mapping

##### Mapping Source

InstanceSpecification

##### Mapping Target

Membership

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- `Element::aliasId`  
`Set {}`
- `Relationship::ownedRelatedElement`  
`Set {}`
- `Membership::memberElement`  
`from`
- `Relationship::source`  
`Set {}`
- `Element::name`  
`null`
- `Element::shortName`  
`null`
- `Element::elementId`  
`Helper.createUUID()`
- `Relationship::target`  
`Set {}`
- `Element::ownedRelationship`  
`Set {}`

#### **C.2.5.4.2.23 LowerBoundValueOwnership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureMembership\_Mapping

##### **Mapping Source**

MultiplicityElement

##### **Mapping Target**

FeatureMembership

##### **Owned Mappings**

(none)

### Applicable filters

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

(none)

### Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- FeatureMembership::ownedMemberFeature  
  
LiteralInteger\_Mapping.getMapped(from.lowerValue)
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
  
null
- OwningMembership::ownedRelatedElement  
  
Set{self.ownedMemberElement() }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- Element::ownedRelationship  
  
Set{}

#### C.2.5.4.2.24 MultiplicityElement\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeature\_Mapping

##### Mapping Source

MultiplicityElement

##### Mapping Target

MultiplicityRange

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId

Set{ }

- MultiplicityRange::ownedRelationship

OrderedSet{MultiplicityLowerBoundOwnership\_Mapping.getMapped(from), MultiplicityUpperBoundOwnership\_Mapping.getMapped(from)}

- Type::isSufficient

false

- MultiplicityRange::isUnique

from.isUnique

- Element::shortName

null

- Type::isAbstract

false

- Element::elementId

Helper.createUUID()

- MultiplicityRange::name

`'multiplicity'`

#### C.2.5.4.2.25 MultiplicityLowerBoundOwnership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToOwningMembership\_Mapping

##### Mapping Source

MultiplicityElement

##### Mapping Target

OwningMembership

##### Owned Mappings

(none)

##### Applicable filters

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

(none)

##### Mapping rules

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

- Relationship::ownedRelatedElement

`Set{ }`

- OwningMembership::memberName

`'lowerBound'`

- Membership::membershipOwningNamespace

*abstract rule*

- Membership::memberShortName

`null`

- Membership::memberElement

*abstract rule*

- OwningMembership::ownedMemberElement

`if from.lowerValue.oclIsUndefined() then DefaultLowerBound_Mapping.getMapped(from) else Elen`

- Element::shortName

null

- Element::elementId

    Helper.createUUID()

- Element::aliasId

    Set{}

- Membership::visibility

    KerML::VisibilityKind::public

- Element::name

    null

- Element::ownedRelationship

    Set{}

#### **C.2.5.4.2.26 MultiplicityMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToOwningMembership\_Mapping

##### **Mapping Source**

MultiplicityElement

##### **Mapping Target**

OwningMembership

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Relationship::ownedRelatedElement

    Set{}



- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- Membership::memberElement  
*abstract rule*
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- OwningMembership::ownedMemberElement  
  
MultiplicityElement\_Mapping.getMapped(from)
- Membership::memberName  
  
null
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- Element::ownedRelationship  
  
Set{}

#### **C.2.5.4.2.27 MultiplicityUpperBoundOwnership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToOwningMembership\_Mapping

##### **Mapping Source**

MultiplicityElement

##### **Mapping Target**

OwningMembership

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- `OwningMembership::ownedMemberElement`

```
if from.upperValue.ocIsUndefined() then
    DefaultUpperBound_Mapping.getMapped(from)
else
    ElementMain_Mapping.getMapped(from.upperValue)
endif
```
- `Relationship::ownedRelatedElement`

```
Set{}
```
- `Membership::membershipOwningNamespace`  
*abstract rule*
- `Membership::memberShortName`

```
null
```
- `Membership::memberElement`  
*abstract rule*
- `Element::shortName`

```
null
```
- `Element::elementId`

```
Helper.createUUID()
```
- `Element::aliasId`

```
Set{}
```
- `Membership::visibility`

```
KerML::VisibilityKind::public
```
- `Element::name`

```
null
```
- `OwningMembership::memberName`

```
'upperBound'
```
- `Element::ownedRelationship`

```
Set{}
```

#### **C.2.5.4.2.28 Operation\_Mapping**

##### **Description**

The expected SysML v2 textual syntax of a mapped UML4SysML::Operation is as follows.

```
part def ThisIsABlock {  
  perform action thisIsAnOperation {  
    in parIn : ScalarValues::Boolean;  
    inout parInOut [0..*] : ScalarValues::String;  
    out parOut;  
    out result : ScalarValues::Integer;  
  }  
}
```

##### **General Mappings**

BehavioralFeature\_Mapping

GenericToActionUsage\_Mapping

##### **Mapping Source**

Operation

##### **Mapping Target**

PerformActionUsage

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- Feature::isEnd  
false
- Type::isSufficient  
false
- Feature::isUnique  
true
- Element::name

```

    from.name

• Element::shortName

    null

• Type::isAbstract

    false

• Feature::isOrdered

    false

• Element::aliasId

    Set{}

• Feature::isPortion

    false

• Usage::isVariation

    false

• Namespace::ownedImport

    Set{}

• Feature::isReadOnly

    false

• Feature::direction

    null

• Element::elementId

    Helper.getID(from)

• Feature::isDerived

    false

• Feature::isComposite

    false

• PerformActionUsage::ownedRelationship

let parameters: Set(UML::Element) = src.ownedElement->select(e | e.ocIsKindOf(UML::Parameter
let parameterSets: Set(UML::Element) = src.ownedElement->select(e | e.ocIsKindOf(UML::Param
let elementsOMS: Set(UML::Element) = ((src.ownedElement - parameters) - parameterSets) in
elementsOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(parameters->collect(e | ParameterMembership_Mapping.getMapped(e)))
->union(parameterSets->collect(e | ParameterSetMembership_Mapping.getMapped(e)))

```

#### C.2.5.4.2.29 Parameter\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToReferenceUsage\_Mapping  
NamedElementMain\_Mapping

##### Mapping Source

Parameter

##### Mapping Target

ReferenceUsage

##### Owned Mappings

- parameterToFeatureTyping : ParameterToFeatureTyping\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Feature::isEnd  

```
false
```
- Type::isSufficient  

```
false
```
- Feature::isUnique  

```
true
```
- ReferenceUsage::ownedRelationship  

```
let typing: KerML::FeatureTyping = parameterToFeatureTyping.to in
if typing.oclIsUndefined() then
  Set{MultiplicityMembership_Mapping.getMapped(from)}
else
  Set{MultiplicityMembership_Mapping.getMapped(from), typing}
endif
```
- Element::shortName  

```
null
```
- Type::isAbstract

- Feature::isOrdered

- `Element::aliasId`

- Feature::isPortion

- Usage::isVariation

- Feature::isReadOnly

- ReferenceUsage::direction

- `Element::elementId`

- Element::name

- Feature::isDerived

- Feature::isComposite

#### C.2.5.4.2.30 ParameterMembership\_Mapping

\*\*\* not specified yet \*\*\*

## GenericToParameterMembership\_Mapping

## Parameter

## ParameterMembership

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- FeatureMembership::ownedRelatedElement  
`Set{self.ownedMemberFeature() }`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{ }`
- FeatureMembership::ownedMemberFeature  
*abstract rule*
- ParameterMembership::ownedMemberParameter  
`Parameter_Mapping.getMapped(from)`
- FeatureMembership::owningType  
*abstract rule*
- Membership::memberName  
`null`
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
`KerML::VisibilityKind::public`
- Element::name  
`null`

- Element::ownedRelationship

Set{ }

#### **C.2.5.4.2.31 ParameterSet\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToReferenceUsage\_Mapping

##### **Mapping Source**

ParameterSet

##### **Mapping Target**

ReferenceUsage

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- Feature::isEnd

false

- Type::isSufficient

false

- ReferenceUsage::name

from.name

- Feature::isUnique

true

- Element::shortName

null

- Type::isAbstract

false



- ReferenceUsage::ownedRelationship

```
from.parameter->collect(p | ParameterSetParameterFeatureMembership_Mapping.getMapped(from, p
```

- Element::elementId

```
Helper.createUUID()
```

- Feature::isOrdered

```
false
```

- Element::aliasId

```
Set{}
```

- Feature::isPortion

```
false
```

- Usage::isVariation

```
false
```

- Feature::isReadOnly

```
false
```

- Feature::direction

```
null
```

- Feature::isDerived

```
false
```

- Feature::isComposite

```
false
```

#### **C.2.5.4.2.32 ParameterSetMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureMembership\_Mapping

##### **Mapping Source**

ParameterSet

##### **Mapping Target**

FeatureMembership

##### **Owned Mappings**

(none)

### Applicable filters

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

(none)

### Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- FeatureMembership::ownedMemberFeature  
  
ParameterSet\_Mapping.getMapped(from)
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
  
null
- OwningMembership::ownedRelatedElement  
  
Set{self.ownedMemberElement() }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- Element::ownedRelationship  
  
Set{}

#### C.2.5.4.2.33 ParameterSetParameterFeatureMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureMembership\_Mapping

##### Mapping Source

ParameterSet

##### Mapping Target

FeatureMembership with qualifier: parameter:Parameter

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
  
null
- OwningMembership::ownedRelatedElement

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

- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility

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

- Element::name

```
null
```

- FeatureMembership::ownedMemberFeature

```
ParameterSetParameterReferenceUsage_Mapping.getMapped(parameter)
```

- Element::ownedRelationship

```
Set{ }
```

#### **C.2.5.4.2.34 ParameterSetParameterReferenceUsage\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToReferenceUsage\_Mapping

##### **Mapping Source**

Parameter

##### **Mapping Target**

ReferenceUsage

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Feature::isEnd

```
false
```

- `Type::isSufficient`  
`false`
- `Feature::isUnique`  
`true`
- `Element::shortName`  
`null`
- `Type::isAbstract`  
`false`
- `Element::elementId`  
`Helper.createUUID()`
- `Feature::isOrdered`  
`false`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `Usage::isVariation`  
`false`
- `ReferenceUsage::ownedRelationship`  
`Set{ParameterSetParameterReferenceUsageFeatureValue_Mapping.getMapped(from), MultiplicityMen`
- `Feature::isReadOnly`  
`false`
- `Feature::direction`  
`null`
- `Element::name`  
`null`
- `Feature::isDerived`  
`false`
- `Feature::isComposite`  
`false`

#### C.2.5.4.2.35 ParameterSetParameterReferenceUsageFeatureValue\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureValue\_Mapping

##### Mapping Source

Parameter

##### Mapping Target

FeatureValue

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- FeatureValue::value  
  
ParameterSetParameterReferenceUsageFeatureValueExpression\_Mapping.getMapped(from)
- Element::aliasId  
  
Set{ }
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName

null

- OwningMembership::ownedRelatedElement

    Set { self.ownedMemberElement () }

- Membership::visibility

    KerML::VisibilityKind::public

- Element::name

    null

- Element::ownedRelationship

    Set {}

#### **C.2.5.4.2.36 ParameterSetParameterReferenceUsageFeatureValueExpression\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureReferenceExpression\_Mapping

##### **Mapping Source**

Parameter

##### **Mapping Target**

FeatureReferenceExpression

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- Feature::isEnd

    false

- Type::isSufficient

    false

- Feature::isUnique

```

    true

    • Element::shortName
        null

    • Type::isAbstract
        false

    • Element::elementId
        Helper.createUUID()

    • Feature::isOrdered
        false

    • Element::aliasId
        Set{}

    • Feature::isPortion
        false

    • Feature::isReadOnly
        false

    • Feature::direction
        null

    • Element::name
        null

    • Feature::isDerived
        false

    • Feature::isComposite
        false

    • FeatureReferenceExpression::ownedRelationship
        Set{ParameterSetParameterReferenceUsageFeatureValueExpressionMembership_Mapping.getMapped(fr

```

#### **C.2.5.4.2.37 ParameterSetParameterReferenceUsageFeatureValueExpressionMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToMembership\_Mapping



## Mapping Source

Parameter

## Mapping Target

Membership

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Element::aliasId  
`Set {}`
- Membership::memberElement  
`from`
- Relationship::ownedRelatedElement  
`Set {}`
- Relationship::source  
`Set {}`
- Element::name  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Relationship::target  
`Set {}`
- Element::ownedRelationship  
`Set {}`

#### C.2.5.4.2.38 ParameterToFeatureTyping\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

TypedElementToFeatureTyping\_Mapping

##### Mapping Source

Parameter

##### Mapping Target

FeatureTyping

##### Owned Mappings

- parameter : Parameter\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId  
`Set{ }`
- Relationship::ownedRelatedElement  
`Set{ }`
- FeatureTyping::typedFeature  
`parameter.to`
- Element::name  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- FeatureTyping::type  
*abstract rule*

- FeatureTyping::typedFeature  
*abstract rule*
- Element::ownedRelationship

Set{}

#### C.2.5.4.2.39 Property\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

PropertyCommon\_Mapping  
NamedElementMain\_Mapping

##### Mapping Source

Property

##### Mapping Target

Feature

##### Owned Mappings

(none)

##### Applicable filters

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

```
if from.type.ocIsUndefined() then false
else
let p: UML::Property = src.ocAsType(UML::Property) in
not p.ocIsUndefined() and
not p.type.ocIsKindOf(UML::DataType) and
not (p.name.indexOf('base_') > 0) and
(p.association.ocIsUndefined() or p.association.ownedEnd->excludes(p))
endif
```

##### Mapping rules

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

- Feature::isEnd  
false
- Feature::isOrdered  
from.isOrdered
- Type::isSufficient  
false

- Feature::isAbstract  
false
- Element::shortName  
null
- Feature::ownedRelationship  

```

let typing: KerML::FeatureTyping = StructuralFeatureToFeatureTyping_Mapping.getMapped(from)
if typing.ocllIsUndefined() then
    Set{MultiplicityMembership_Mapping.getMapped(from)}
else
    Set{MultiplicityMembership_Mapping.getMapped(from), typing}
endif

```
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Feature::direction  
null
- Element::elementId  
Helper.getID(from)
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false
- Feature::isUnique  
from.isUnique
- Feature::isReadOnly  
*abstract rule*

#### **C.2.5.4.2.40 PropertyCommon\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

StructuralFeature\_Mapping

## Mapping Source

Property

## Mapping Target

Feature

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Type::isSufficient

```
false
```

- Feature::isComposite

```
from.isComposite
```

- Feature::ownedRelationship

```
let typing: KerML::FeatureTyping = StructuralFeatureToFeatureTyping_Mapping.getMapped(from)
let subsetting: Set(KerML::Subsetting) = from.subsettedProperty->collect(p | PropertySubsetting)
let subsettingMultiplicityTyping: Set(KerML::Relationship) = subsetting->union(if typing.ocIsMultiplicityMembershipMapping then
    Set{MultiplicityMembership_Mapping.getMapped(from)}
else
    Set{MultiplicityMembership_Mapping.getMapped(from), typing}
endif)->asSet() in
```

```
let relationships: Set(KerML::Relationship) = if from.defaultValue.ocIsTypeOf(UML::OpaqueExpression) then
    if from.defaultValue.ocIsUndefined() then
        relationships
    else
        relationships->including(if from.defaultValue.ocIsTypeOf(UML::OpaqueExpression) then Pro
    endif
endif
```

- Feature::isEnd

```
if from.association.ocIsUndefined() then
    false
else
    from.association.ownedEnd->includes(from)
endif
```

- Feature::isUnique  
true
- Element::shortName  
null
- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Feature::isReadOnly  
false
- Feature::isDerived  
from.isDerived
- Feature::direction  
null
- Element::name  
null
- Element::ownedRelationship  
Set{}

#### **C.2.5.4.2.41 PropertyDefaultValue\_Mapping**

##### **Description**

The expected SysML v2 textual syntax of a mapped SysML v2 default value is as follows:

```
attribute value : ScalarValues::String default := "thisIsTheDefaultValue";
```

##### **General Mappings**

GenericToFeatureValue\_Mapping

### Mapping Source

ValueSpecification

### Mapping Target

FeatureValue

### Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- FeatureValue::isDefault

```
if from.oclIsUndefined() then false else true endif
```
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName

```
null
```
- Element::shortName

```
null
```
- Element::elementId

```
Helper.createUUID()
```
- Element::aliasId

```
Set{ }
```
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName

```
null
```
- OwningMembership::ownedRelatedElement

```
Set{ self.ownedMemberElement() }
```
- Membership::visibility

KerML::VisibilityKind::public

- Element::name

null

- FeatureValue::value

ValueSpecification\_Mapping.getMapped(from)

- Element::ownedRelationship

Set{}

#### **C.2.5.4.2.42 PropertyDefaultValueOpaqueExpression\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

PropertyDefaultValue\_Mapping

##### **Mapping Source**

OpaqueExpression

##### **Mapping Target**

FeatureValue

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- FeatureValue::isDefault

false

- FeatureValue::isInitial

false

- Membership::membershipOwningNamespace

*abstract rule*

- Membership::memberShortName



- `Element::shortName`

- `Element::elementId`

- FeatureValue::ownedRelatedElement

- `Element::aliasId`

- Membership::memberName

- FeatureValue::value

- FeatureValue::featureWithValue

- Membership::visibility

- `Element::name`

- FeatureValue::value

- `Element::ownedRelationship`

#### 4.2.43 PropertySubsetting\_Mapping

\*\*\* not specified yet \*\*\*

## GenericToSubsetting\_Mapping

### Property

OMG Systems Modeling Language (SysML) v2.0, Submission

Subsetting with qualifier: subsettedProperty:Property

### Owned Mappings

(none)

### Applicable filters

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

(none)

### Mapping rules

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

- Element::aliasId  
`Set {}`
- Relationship::ownedRelatedElement  
`Set {}`
- Subsetting::subsettingFeature  
`Property_Mapping.getMapped(from)`
- Specialization::specific  
*abstract rule*
- Element::name  
`null`
- Subsetting::subsettedFeature  
`Property_Mapping.getMapped(subsettedProperty)`
- Element::shortName  
`null`
- Specialization::general  
*abstract rule*
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set {}`

#### C.2.5.4.2.44 PropertyUntyped\_Mapping

### Description

\*\*\* not specified yet \*\*\*

### General Mappings

PropertyCommon\_Mapping  
GenericToReferenceUsage\_Mapping  
NamedElementMain\_Mapping

### Mapping Source

Property

### Mapping Target

ReferenceUsage

### Owned Mappings

(none)

### Applicable filters

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

```
from.type.oclIsUndefined() and not from.oclIsKindOf(UML::Port)
```

### Mapping rules

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

- Feature::isEnd  

```
false
```
- Feature::isOrdered  

```
from.isOrdered
```
- Type::isSufficient  

```
false
```
- Feature::isAbstract  

```
false
```
- Element::shortName  

```
null
```
- Feature::ownedRelationship  

```
let typing: KerML::FeatureTyping = StructuralFeatureToFeatureTyping_Mapping.getMapped(from)
if typing.oclIsUndefined() then
  Set{MultiplicityMembership_Mapping.getMapped(from)}
else
  Set{MultiplicityMembership_Mapping.getMapped(from), typing}
endif
```
- Element::aliasId  

```
Set{}
```

- Feature::isPortion  
false
- Usage::isVariation  
false
- Feature::direction  
null
- Element::elementId  
Helper.getID(from)
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false
- Feature::isUnique  
from.isUnique
- Feature::isReadOnly  
*abstract rule*

#### **C.2.5.4.2.45 Realization\_Mapping**

##### **Description**

\*\*\* issue \*\*\* This mapping is not appropriate since the Realization can have more than one client and more than one supplier and that the semantics defined in UML is much more informal than those of a generalization

##### **General Mappings**

Abstraction\_Mapping

##### **Mapping Source**

Realization

##### **Mapping Target**

Dependency

##### **Owned Mappings**

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Element::aliasId

```
Set{ }
```

- Dependency::supplier

```
from.target->collect(e | ElementMain_Mapping.getMapped(e))
```

- Dependency::name

```
from.name
```

- Element::ownedRelationship

```
ElementOwnership_Mapping.getMappedColl(from.ownedElement)
```

- Relationship::owningRelatedElement

```
ElementMain_Mapping.getMapped(from.owner)
```

- Dependency::client

```
from.source->collect(e | ElementMain_Mapping.getMapped(e))
```

- Element::elementId

```
Helper.getID(from)
```

- Relationship::ownedRelatedElement

```
from.relatedElement->select(e | from.ownedElement->includes(e))->collect(e | ElementMain_Map
```

- Element::shortName

```
null
```

### C.2.5.4.2.46 Slot\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

GenericToFeature\_Mapping  
ElementMain\_Mapping

#### Mapping Source

Slot

### Mapping Target

Feature

### Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  
`Set{}`
- Type::isSufficient  
`false`
- Element::name  
`null`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set{}`

#### C.2.5.4.2.47 SlotMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureMembership\_Mapping

##### Mapping Source

Slot

## Mapping Target

FeatureMembership

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
    null
- Element::shortName  
  
    null
- Element::elementId  
  
    Helper.createUUID()
- Element::aliasId  
  
    Set{ }
- OwningMembership::ownedMemberElement  
*abstract rule*
- FeatureMembership::isReadOnly  
  
    from.isReadOnly
- OwningMembership::ownedRelatedElement  
  
    Set{ self.ownedMemberElement() }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- FeatureMembership::ownedMemberFeature  
  
    from
- FeatureMembership::memberName

`from.definingFeature.name`

- Membership::visibility

`KerML::VisibilityKind::public`

- Element::name

`null`

- Element::ownedRelationship

`Set{}`

#### **C.2.5.4.2.48 SlotToFeatureTyping\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureTyping\_Mapping

##### **Mapping Source**

Slot

##### **Mapping Target**

FeatureTyping

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- Element::aliasId

`Set{}`

- Relationship::ownedRelatedElement

`Set{}`

- FeatureTyping::type

`ElementMain_Mapping.getMapped(from)`



- Specialization::specific  
*abstract rule*
- Element::name  
  
null
- Element::shortName  
  
null
- FeatureTyping::typedFeature  
  
Slot\_Mapping.getMapped(from)
- Specialization::general  
*abstract rule*
- Element::elementId  
  
Helper.createUUID()
- Element::ownedRelationship  
  
Set{}

#### **C.2.5.4.2.49 SlotValue\_Mapping**

##### **Description**

Issue here since a KerML feature cannot have more than one FeatureValue while a UML::Slot can. How to manage collection of values?

##### **General Mappings**

GenericToFeatureValue\_Mapping

##### **Mapping Source**

ValueSpecification

##### **Mapping Target**

FeatureValue

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

```
src.owner.ocIsKindOf(UML::Slot)
```

##### **Mapping rules**

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- Element::shortName  
  
null
- FeatureValue::value  
  
ValueSpecification\_Mapping.getMapped(from)
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
  
null
- OwningMembership::ownedRelatedElement  
  
Set{self.ownedMemberElement() }
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- FeatureValue::featureWithValue  
  
Slot\_Mapping.getMapped(from.owner)
- Element::ownedRelationship  
  
Set{}

#### **C.2.5.4.2.50 StructuralFeature\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeature\_Mapping

## Mapping Source

StructuralFeature

## Mapping Target

Feature

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Element::aliasId  
`Set{}`
- Feature::isOrdered  
`from.isOrdered`
- Type::isSufficient  
`false`
- Feature::isAbstract  
`false`
- Element::name  
`null`
- Element::shortName  
`null`
- Feature::isUnique  
`from.isUnique`
- Feature::ownedRelationship  

```
let typing: KerML::FeatureTyping = StructuralFeatureToFeatureTyping_Mapping.getMapped(from)
if typing.oclIsUndefined() then
    Set{MultiplicityMembership_Mapping.getMapped(from)}
else
    Set{MultiplicityMembership_Mapping.getMapped(from), typing}
endif
```

- Element::elementId  
`Helper.createUUID()`
- Feature::isReadOnly  
*abstract rule*

#### **C.2.5.4.2.51 StructuralFeatureMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureMembership\_Mapping

##### **Mapping Source**

StructuralFeature

##### **Mapping Target**

FeatureMembership

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
 (none)

##### **Mapping rules**

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Element::shortName  
`null`
- FeatureMembership::visibility  

```

if (from.ocIsKindOf(UML::NamedElement)) then
  Helper.getKerMLVisibilityKind(from.ocAsType(UML::NamedElement).visibility)
else
  KerML::VisibilityKind::public
endif

```
- Element::elementId

```
Helper.createUUID()
```

- Element::aliasId

```
Set{}
```

- OwningMembership::ownedMemberElement  
*abstract rule*
- FeatureMembership::ownedMemberFeature

```
NamedElementMain_Mapping.getMapped(from)
```

- Membership::memberName

```
null
```

- OwningMembership::ownedRelatedElement

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

- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Element::name

```
null
```

- Element::ownedRelationship

```
Set{}
```

#### **C.2.5.4.2.52 StructuralFeatureToFeatureTyping\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

TypedElementToFeatureTyping\_Mapping

##### **Mapping Source**

StructuralFeature

##### **Mapping Target**

FeatureTyping

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  
`Set{}`
- Relationship::ownedRelatedElement  
`Set{}`
- FeatureTyping::typedFeature  
`ElementMain_Mapping.getMapped(from)`
- Element::name  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- FeatureTyping::type  
*abstract rule*
- FeatureTyping::typedFeature  
*abstract rule*
- Element::ownedRelationship  
`Set{}`

#### C.2.5.4.2.53 TypedElementToFeatureTyping\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### 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.ocIsUndefined()
  and not (src.type.ocIsKindOf(UML::Enumeration) and Helper.getSysMLv2EnumerationDefinition(src.type))
```

### Mapping rules

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

- Element::aliasId  
`Set{}`
- Relationship::ownedRelatedElement  
`Set{}`

- FeatureTyping::typedFeature  
*abstract rule*
- Specialization::specific  
*abstract rule*
- Element::name  
`null`

- Element::shortName  
`null`

- Specialization::general  
*abstract rule*
- Element::elementId  
`Helper.createUUID()`

- Element::ownedRelationship  
`Set{}`

- FeatureTyping::type

```
let sysmlv1PrimitiveType : SysMLv2::DataType = if from.type.ocIsKindOf(UML::PrimitiveType) then
let sysmlv1EnumerationType : SysMLv2::Enumeration = if from.type.ocIsKindOf(UML::Enumeration) then
if not sysmlv1PrimitiveType.ocIsUndefined() then sysmlv1PrimitiveType else
if not sysmlv1EnumerationType.ocIsUndefined() then sysmlv1EnumerationType else
Classifier_Mapping.getMapped(from.type) endif endif
```

#### C.2.5.4.2.54 UpperBoundValueOwnership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

## General Mappings

GenericToFeatureMembership\_Mapping

## Mapping Source

MultiplicityElement

## Mapping Target

FeatureMembership

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
    null
- Element::shortName  
  
    null
- Element::elementId  
  
    Helper.createUUID()
- Element::aliasId  
  
    Set{ }
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
  
    null
- OwningMembership::ownedRelatedElement  
  
    Set{ self.ownedMemberElement() }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*



- FeatureMembership::ownedMemberFeature

```

if from.upper <> -1 then
    LiteralUnlimitedToInteger_Mapping.getMapped(from.upperValue)
else
    LiteralUnlimitedToUnbounded_Mapping.getMapped(from.upperValue)
endif

```

- Membership::visibility

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

- Element::name

```
null
```

- Element::ownedRelationship

```
Set{}
```

## C.2.5.5 CommonBehavior

### C.2.5.5.1 Overview

**Table 18. List of all Overview Mapping Specifications**

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
AnyReceiveEvent	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	
Behavior	Behavior	Behavior_Mapping	true
CallEvent	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	
ChangeEvent	TextualRepresentation	ChangeEvent_Mapping	
Event	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	
FunctionBehavior	TextualRepresentation OwningMembership Behavior	OpaqueBehaviorSpecification_Mapping OpaqueBehaviorMembership_Mapping CommonOpaqueBehavior_Mapping	

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
MessageEvent	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	
OpaqueBehavior	TextualRepresentation OwningMembership Behavior	OpaqueBehaviorSpecification_Mapping OpaqueBehaviorMembership_Mapping CommonOpaqueBehavior_Mapping	
SignalEvent	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	
TimeEvent	TextualRepresentation	TimeEvent_Mapping	
Trigger	AcceptActionUsage	Trigger_Mapping	

## C.2.5.5.2 Mapping Specifications

### C.2.5.5.2.1 Behavior\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

GenericToBehavior\_Mapping  
Class\_Mapping

#### Mapping Source

Behavior

#### Mapping Target

Behavior

#### Owned Mappings

(none)

#### Applicable filters

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

true

#### Mapping rules

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

- Element::aliasId

```
Set{}
```

- Classifier::isAbstract

```
from.isAbstract
```

- Type::isSufficient

```
false
```

- Namespace::ownedImport

```
Set{}
```

- Behavior::ownedRelationship

```
let parameters: Set(UML::Element) = from.ownedElement->select(e | e.oclIsKindOf(UML::Parameter))
let parameterSets: Set(UML::Element) = src.ownedElement->select(e | e.oclIsKindOf(UML::ParameterSet))
let features: Set(UML::Element) = from.ownedElement->select(e | e.oclIsKindOf(UML::Property))
let elementsOMS: Set(UML::Element) = (((from.ownedElement - parameters) parameterSets) - features)
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)))
```

- Element::elementId

```
Helper.getID(from)
```

- Element::name

```
from.name
```

- Element::shortName

```
null
```

#### C.2.5.5.2.2 ChangeEvent\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToTextualRepresentation\_Mapping  
NamedElementMain\_Mapping

##### Mapping Source

ChangeEvent

##### Mapping Target

TextualRepresentation

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Element::aliasId

```
Set{}
```

- Element::ownedRelationship

```
ElementOwnership_Mapping.getMappedColl(from.ownedElement)
```

- TextualRepresentation::body

```
if from.changeExpression.ocIsKindOf(UML::OpaqueExpression)
then if from.changeExpression.ocIsType(UML::OpaqueExpression).body.ocIsUndefined() then OclU
else OclUndefined
endif
```

- Element::elementId

```
Helper.getID(from)
```

- Element::name

```
null
```

- TextualRepresentation::language

```
if from.changeExpression.ocIsKindOf(UML::OpaqueExpression)
then if from.changeExpression.ocIsType(UML::OpaqueExpression).language->size() = 0 then OclU
else OclUndefined
endif
```

- Element::shortName

```
null
```

- AnnotatingElement::annotation

```
Set{}
```

### C.2.5.5.2.3 CommonOpaqueBehavior\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

## General Mappings

Behavior\_Mapping

## Mapping Source

OpaqueBehavior

## Mapping Target

Behavior

## Owned Mappings

(none)

## Applicable filters

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

```
not Helper.hasStereotypeApplied(from, 'SysML::Requirements::Requirement') and not from.ocIsTypeOf(
```

## Mapping rules

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

- Element::aliasId

```
Set{}
```

- Classifier::isAbstract

```
from.isAbstract
```

- Type::isSufficient

```
false
```

- Namespace::ownedImport

```
Set{}
```

- Behavior::ownedRelationship

```
let parameters : Set(UML::Parameter) = from.ownedElement->select(e | e.ocIsKindOf(UML::Parameter))
let parameterSets : Set(UML::ParameterSet) = from.ownedElement->select(e | e.ocIsKindOf(UML::ParameterSet))
let features : Set(UML::Property) = from.ownedElement->select(e | e.ocIsKindOf(UML::Property))
let elementsOMS: Set(UML::Element) = (((from.ownedElement - parameters) - parameterSets) - features)
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)))
```

- Classifier::ownedRelationship

```

let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | (e.ocIsKindOf(UML::Prop
let redefinedAttributes: Set(UML::Element) = from.ownedElement->select(e | from.ocIsKindOf(U
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.ocIsKindOf
let constraints : Set(UML::Constraint) = UML::Constraint.allInstances()->select( c | c.constr
let toElementOMS: Set(UML::Element) = (((from.ownedElement - toElementFMS) - redefinedAttribu
let relationships: Sequence(KerML::Relationship) =
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(constraints->collect(e | ConstrainedElementFeatureMembership_Mapping.getMapped(e)))
->union(redefinedAttributes->collect(e | AttributeRedefinedMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e))) in
if from.classifierBehavior.ocIsUndefined() then relationships else relationships->append(Cla

```

- Element::elementId

```
Helper.getID(from)
```

- Element::name

```
from.name
```

- Element::shortName

```
null
```

#### C.2.5.5.2.4 OpaqueBehaviorAsDefinition\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

```

GenericToDefinition_Mapping
CommonOpaqueBehavior_Mapping

```

##### Mapping Source

```
OpaqueBehavior
```

##### Mapping Target

```
ActionDefinition
```

##### Owned Mappings

```
(none)
```

##### Applicable filters

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

```
src.owner.ocIsKindOf(UML::Package)
```

##### Mapping rules

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

- Element::aliasId

Set{}

- Classifier::isAbstract

from.isAbstract

- Type::isSufficient

false

- Namespace::ownedImport

Set{}

- Behavior::ownedRelationship

```
let parameters: Set(UML::Element) = from.ownedElement->select(e | e.oclIsKindOf(UML::Parameter))
let parameterSets: Set(UML::Element) = src.ownedElement->select(e | e.oclIsKindOf(UML::ParameterSet))
let features: Set(UML::Element) = from.ownedElement->select(e | e.oclIsKindOf(UML::Property))
let elementsOMS: Set(UML::Element) = (((from.ownedElement - parameters) parameterSets) - features)
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)))
```

- Element::elementId

Helper.getID(from)

- Element::name

from.name

- Element::shortName

null

#### C.2.5.5.2.5 OpaqueBehaviorAsUsage\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

CommonOpaqueBehavior\_Mapping  
GenericToActionUsage\_Mapping

##### Mapping Source

OpaqueBehavior

##### Mapping Target

ActionUsage

## Owned Mappings

(none)

## Applicable filters

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

```
not src.owner.ocIsKindOf(UML::Package)
```

## Mapping rules

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

- Feature::isEnd

```
false
```

- Type::isSufficient

```
false
```

- Behavior::ownedRelationship

```
let parameters: Set(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::Parameter))
let parameterSets: Set(UML::Element) = src.ownedElement->select(e | e.ocIsKindOf(UML::ParameterSet))
let features: Set(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::Property))
let elementsOMS: Set(UML::Element) = (((from.ownedElement - parameters) parameterSets) - features)
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)))
```

- Feature::isUnique

```
true
```

- Element::name

```
from.name
```

- Element::shortName

```
null
```

- Feature::isOrdered

```
false
```

- Element::aliasId

```
Set{}
```

- Classifier::isAbstract

```
from.isAbstract
```



- Feature::isPortion  
false
- Usage::isVariation  
false
- Namespace::ownedImport  
Set{}
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::elementId  
Helper.getID(from)
- Feature::isDerived  
false
- Feature::isComposite  
false

#### **C.2.5.5.2.6 OpaqueBehaviorMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToOwningMembership\_Mapping

##### **Mapping Source**

OpaqueBehavior

##### **Mapping Target**

OwningMembership with qualifier: language:String

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Relationship::ownedRelatedElement  
`Set{}`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Membership::memberElement  
*abstract rule*
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- OwningMembership::ownedMemberElement  
`OpaqueBehaviorSpecification_Mapping.getMapped(from, language)`
- Element::aliasId  
`Set{}`
- Membership::memberName  
`null`
- Membership::visibility  
`KerML::VisibilityKind::public`
- Element::name  
`null`
- Element::ownedRelationship  
`Set{}`

### C.2.5.5.2.7 OpaqueBehaviorSpecification\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

GenericToTextualRepresentation\_Mapping

## Mapping Source

OpaqueBehavior

## Mapping Target

TextualRepresentation with qualifier: language:String

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Element::aliasId  

```
Set{}
```
- TextualRepresentation::body  

```
let index:Integer = from.language->indexOf(language) in  
from._'body'->at(index)
```
- Element::name  

```
null
```
- Element::shortName  

```
null
```
- AnnotatingElement::annotation  

```
Set{}
```
- Element::elementId  

```
Helper.createUUID()
```
- TextualRepresentation::language  

```
language
```
- Element::ownedRelationship  

```
Set{}
```

### C.2.5.5.2.8 TimeEvent\_Mapping

#### Description

tbd - just a placeholder yet

### General Mappings

NamedElementMain\_Mapping  
GenericToTextualRepresentation\_Mapping

### Mapping Source

TimeEvent

### Mapping Target

TextualRepresentation

### Owned Mappings

(none)

### Applicable filters

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

(none)

### Mapping rules

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

- Element::aliasId  
`Set{}`
- Element::ownedRelationship  
`ElementOwnership_Mapping.getMappedColl(from.ownedElement)`
- TextualRepresentation::body  
`'tbd timeevent'`
- Element::elementId  
`Helper.getID(from)`
- Element::name  
`null`
- Element::shortName  
`null`
- AnnotatingElement::annotation  
`Set{}`

#### C.2.5.5.2.9 Trigger\_Mapping

## C.2.5.6 CommonStructure

### C.2.5.6.1 Overview

**Table 20. List of all Overview Mapping Specifications**

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
Abstraction	ReturnParameterMembership	SatisfySubjectMembership	FeatureValueFeatureReferenceExpressionReturn
	ReferenceUsage	SatisfyFeatureMembership	ReferenceUsage_Mapping
	Feature	SatisfySubjectMembership	FeatureValueFeatureReferenceExpressionReturn
	ReferenceUsage	SatisfySubjectMembership	ReferenceUsage_Mapping
	FeatureReferenceExpression	SatisfySubjectMembership	FeatureValueFeatureReferenceExpression_Mapping
	FeatureValue	SatisfySubjectMembership	FeatureValue_Mapping
	RequirementUsage	TestCaseVerifyObjective	RequirementUsage_Mapping
	SubjectMembership	SatisfySubjectMembership	Membership_Mapping
	FeatureTyping	SatisfyFeatureTyping	Typing_Mapping
	Membership	SatisfySubjectMembership	FeatureValueFeatureReferenceExpressionMembership_Mapping
	FeatureMembership	SatisfyFeatureMembership	Membership_Mapping
	ReferenceSubsetting	TestCaseVerifyRequirement	UsageReferenceSubsetting_Mapping
	Dependency	Abstraction_Mapping	Helper.hasStereotypeApplied(Abstraction, 'SysML::Requirements::Verify')
	RequirementUsage	TestCaseVerifyRequirement	Usage_Mapping
	ObjectiveMembership	TestCaseVerifyObjective	Membership_Mapping
	FeatureTyping	SatisfyFeatureMembership	ReferenceUsageFeatureTyping_Mapping
	RequirementVerificationMembership	Membership_Mapping	
	SatisfyRequirementUsage	Satisfy_Mapping	Helper.hasStereotypeApplied(Abstraction, 'SysML::Requirements::Satisfy')
Comment	FeatureValue	ElementGroupMetadata	FeatureValue_Mapping
	Comment	Comment_Mapping	Helper.hasStereotypeApplied(Comment, 'SysML::Requirements::ElementGroup')
	OwningMembership	ProblemRationaleMetadata	Membership_Mapping
	Redefinition	ProblemRationaleMetadata	Redefinition_Mapping
	Package	ElementGroup	Package_Mapping
	MetadataUsage	ProblemRationaleMetadata	Usage_Mapping
	FeatureTyping	ProblemRationaleMetadata	FeatureTyping_Mapping
	FeatureMembership	ProblemRationaleMetadata	FeatureMembership_Mapping
	FeatureValue	ProblemRationaleMetadata	FeatureValue_Mapping
	Comment	CommentToConcern	Comment_Mapping
	Annotation	CommentToAnnotation	Annotation_Mapping
	Redefinition	ElementGroupMetadata	Redefinition_Mapping
	MetadataUsage	ElementGroupMetadata	Usage_Mapping
	ReturnParameterMembership	CommentToConcern	ReturnParameterMembership_Mapping
	Membership	ElementGroupMetadata	Membership_Mapping
	ReferenceUsage	CommentToConcern	ReturnParameter_Mapping
	LiteralString	ProblemRationaleMetadata	FeatureValueString_Mapping
	Annotation	CommentToConcern	Documentation_Mapping
	FeatureTyping	ElementGroupMetadata	FeatureTyping_Mapping
	FeatureMembership	ElementGroupMetadata	FeatureMembership_Mapping
	ReferenceUsage	ProblemRationaleMetadata	ReferenceUsage_Mapping
	ReferenceUsage	ElementGroupMetadata	ReferenceUsage_Mapping
	LiteralString	ElementGroupCriterion	Criterion_Mapping

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
Constraint	AssertConstraintUsage ConstraintDefinition FeatureTyping FeatureMembership	ConstraintUsage_Mapping Constraint_Mapping ConstraintUsageFeatureTyping_Mapping ConstrainedElementFeatureMembership_Mapping	
Dependency	FeatureMembership Dependency AllocationUsage FeatureTyping ReferenceUsage AllocationDefinition	AllocationDefinitionFromFeatureMembership_Mapping Dependency_Mapping AllocationUsage_Mapping AllocationDefinitionFromFeatureTyping_Mapping AllocationDefinitionFromReferenceUsage_Mapping AllocationDefinition_Mapping	Helper.hasStereotypeApplied(Dependency, 'SysML::Allocations::Allocate') and not Dependency.client->select(t   t.oclIsKindOf(UML::Type))->notEmpty() Helper.hasStereotypeApplied(Dependency, 'SysML::Allocations::Allocate') and Dependency.client->select(t   t.oclIsKindOf(UML::Type))->notEmpty()
DirectedRelationship	Relationship	DirectedRelationship_Mapping	
Element	FeatureTyping FeatureTyping Element Feature FeatureTyping Relationship Expression ReferenceUsage OwningMembership LiteralInteger ReturnParameterMembership ParameterMembership FeatureMembership FeatureTyping LiteralInteger MultiplicityRange ReferenceUsage LiteralInteger Element Membership ReturnParameterMembership ReturnParameterMembership	CommonParameterReferenceUsageInFeatureTyping_Mapping Mapping CommonReturnParameterFeatureUntyped_Mapping CommonReturnParameterFeatureTyping_Mapping ElementOwnership_Mapping CommonValueSpecification_Mapping CommonParameterReferenceUsageInUntyped_Mapping DefaultMultiplicityMembership_Mapping CommonReturnParameterFeatureMembership_Mapping CommonParameterReferenceUsageInMembership_Mapping DefaultMultiplicityBoundOwnership_Mapping CommonReturnParameterReferenceUsageFeatureTyping_Mapping DefaultUpperBound_Mapping DefaultMultiplicityElement_Mapping CommonReturnParameterReferenceUsageUntyped_Mapping DefaultLowerBound_Mapping ElementMain_Mapping ElementMembership_Mapping CommonReturnParameterReferenceUsageMembership_Mapping EmptyReturnParameterFeatureMembership_Mapping	
ElementImport	Membership	ElementImport_Mapping	

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
MultiplicityElement	OwningMembership OwningMembership OwningMembership FeatureMembership MultiplicityRange FeatureMembership	MultiplicityUpperBoundOwnership_Mapping MultiplicityLowerBoundOwnership_Mapping MultiplicityMembership_Mapping UpperBoundValueOwnership_Mapping MultiplicityElement_Mapping LowerBoundValueOwnership_Mapping	
NamedElement	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	
Namespace	Namespace	Namespace_Mapping	
PackageableElement	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	
PackageImport	Import	PackageImport_Mapping	
ParameterableElement	FeatureTyping FeatureTyping Element Feature FeatureTyping Relationship Expression ReferenceUsage OwningMembership LiteralInteger ReturnParameterMembership ParameterMembership FeatureMembership FeatureTyping LiteralInteger MultiplicityRange ReferenceUsage LiteralInteger Element Membership ReturnParameterMembership ReturnParameterMembership	CommonParameterReferenceUsageInFeatureTyping_Mapping Mapping CommonReturnParameterFeatureUntyped_Mapping CommonReturnParameterFeatureTyping_Mapping ElementOwnership_Mapping CommonValueSpecification_Mapping CommonParameterReferenceUsageInUntyped_Mapping DefaultMultiplicityMembership_Mapping CommonReturnParameterFeatureMembership_Mapping CommonParameterReferenceUsageInMembership_Mapping DefaultMultiplicityBoundOwnership_Mapping CommonReturnParameterReferenceUsageFeatureTyping_Mapping DefaultUpperBound_Mapping DefaultMultiplicityElement_Mapping CommonReturnParameterReferenceUsageUntyped_Mapping DefaultLowerBound_Mapping ElementMain_Mapping ElementMembership_Mapping CommonReturnParameterReferenceUsageMembership_Mapping EmptyReturnParameterFeatureMembership_Mapping	
Realization	Dependency	Realization_Mapping	not Helper.hasStereotypeApplied(Realization, 'SysML::Requirements::Verify')
Relationship	Relationship	Relationship_Mapping	

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
Type	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	
TypedElement	FeatureMembership FeatureTyping FeatureTyping FeatureValue ParameterMembership FeatureReferenceExpression Membership Feature ReferenceUsage	CommonReferenceUsageInFeatureMembership_Mapping CommonReferenceUsageInFeatureTyping_Mapping TypedElementToFeatureTyping_Mapping EqualOperatorExpressionFeatureValue_Mapping EqualOperatorExpressionOperand_Mapping CommonFeatureReferenceExpression_Mapping CommonMembership_Mapping EqualOperatorExpressionFeature_Mapping CommonReferenceUsageInUntyped_Mapping	not src.type.ocIsUndefined() and not src.type.ocIsKindOf(UML::Enumeration) and Helper.getSysMLv2EnumerationDefinition(src)
Usage	Dependency	Usage_Mapping	

## C.2.5.6.2 Mapping Specifications

### C.2.5.6.2.1 Abstraction\_Mapping

#### Description

There is no way to represent the "mapping" property on the target metaclass

#### General Mappings

Dependency\_Mapping

#### Mapping Source

Abstraction

#### Mapping Target

Dependency

#### Owned Mappings

(none)

#### Applicable filters

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

```
not Helper.hasStereotypeApplied(from, 'SysML::Requirements::Verify')
```



## Mapping rules

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

- `Element::aliasId`

```
Set{}
```

- `Element::ownedRelationship`

```
ElementOwnership_Mapping.getMappedColl(from.ownedElement)
```

- `Relationship::target`

```
from.target->collect(e | ElementMain_Mapping.getMapped(e))
```

- `Relationship::owningRelatedElement`

```
ElementMain_Mapping.getMapped(from.owner)
```

- `Element::elementId`

```
Helper.getID(from)
```

- `Element::name`

```
null
```

- `Relationship::source`

```
from.source->collect(e | ElementMain_Mapping.getMapped(e))
```

- `Relationship::ownedRelatedElement`

```
from.relatedElement->select(e | from.ownedElement->includes(e))->collect(e | ElementMain_Mapping.getMapped(e))
```

- `Element::shortName`

```
null
```

### C.2.5.6.2.2 Comment\_Mapping

#### Description

test

#### General Mappings

ElementMain\_Mapping

GenericToAnnotatingElement\_Mapping

#### Mapping Source

Comment

#### Mapping Target

Comment

## Owned Mappings

(none)

## Applicable filters

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

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

## Mapping rules

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

- Element::aliasId  

```
Set{}
```
- Comment::annotation  

```
from.annotatedElement->collect(e | CommentToAnnotation_Mapping.getMapped(from, e))
```
- Comment::ownedRelationship  

```
self.annotation()
```
- Element::name  

```
null
```
- Comment::body  

```
if from.body->isEmpty() then '' else from.body endif
```
- Element::shortName  

```
null
```
- Element::elementId  

```
Helper.createUUID()
```
- Element::ownedRelationship  

```
Set{}
```

### C.2.5.6.2.3 CommentToAnnotation\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

GenericToAnnotation\_Mapping

#### Mapping Source

Comment

### Mapping Target

Annotation with qualifier: annotatedElement:Element

### Owned Mappings

(none)

### Applicable filters

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

(none)

### Mapping rules

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

- Element::aliasId  
`Set{}`
- Relationship::ownedRelatedElement  
`Set{}`
- Annotation::annotatedElement  
`ElementMain_Mapping.getMapped(annotatedElement)`
- Annotation::annotatingElement  
`Comment_Mapping.getMapped(from)`
- Relationship::source  
`Set{}`
- Element::name  
`null`
- Element::shortName  
`null`
- Annotation::owningAnnotatedElement  
`null`
- Element::elementId  
`Helper.createUUID()`
- Relationship::target  
`Set{}`

- Element::ownedRelationship

Set{ }

#### C.2.5.6.2.4 Constraint\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToConstraintDefinition\_Mapping  
NamedElementMain\_Mapping

##### Mapping Source

Constraint

##### Mapping Target

ConstraintDefinition

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId

Set{ }

- Type::isSufficient

false

- ConstraintDefinition::ownedRelationship

Set{ElementFeatureMembership\_Mapping.getMapped(from.specification), CommonReturnParameterRef

- Definition::isVariation

false

- Element::elementId

Helper.getID(from)

- Element::name

null

- Element::shortName

null

- Type::isAbstract

false

#### **C.2.5.6.2.5 ConstrainedElementFeatureMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureMembership\_Mapping

##### **Mapping Source**

Constraint

##### **Mapping Target**

FeatureMembership

##### **Owned Mappings**

- constraintUsage : ConstraintUsage\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- FeatureMembership::ownedMemberFeature

constraintUsage.to

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName

null

- Element::shortName

null

- Element::elementId

- Helper.createUUID()
- Element::aliasId
  - Set{ }
- OwningMembership::ownedMemberElement
  - abstract rule*
- Membership::memberName
  - null
- OwningMembership::ownedRelatedElement
  - Set{ self.ownedMemberElement() }
- TypeFeaturing::featureOfType
  - abstract rule*
- TypeFeaturing::featuringType
  - abstract rule*
- Membership::visibility
  - KerML::VisibilityKind::public
- Element::name
  - null
- Element::ownedRelationship
  - Set{ }

#### C.2.5.6.2.6 ConstraintUsageFeatureTyping\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureTyping\_Mapping

##### Mapping Source

Constraint

##### Mapping Target

FeatureTyping

##### Owned Mappings

- constraintUsage : ConstraintUsage\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  
`Set{}`
- Relationship::ownedRelatedElement  
`Set{}`
- Specialization::specific  
*abstract rule*
- FeatureTyping::type  
`from`
- Element::name  
`null`
- Element::shortName  
`null`
- Specialization::general  
*abstract rule*
- FeatureTyping::typedFeature  
`constraintUsage.to`
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set{}`

#### C.2.5.6.2.7 ConstraintUsage\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToUsage\_Mapping

##### Mapping Source

Constraint

##### Mapping Target

AssertConstraintUsage

### Owned Mappings

- constraintUsageFeatureTyping : ConstraintUsageFeatureTyping\_Mapping

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Feature::isEnd  
false
- Type::isSufficient  
false
- Feature::isUnique  
true
- AssertConstraintUsage::ownedRelationship  
Set{constraintUsageFeatureTyping.to, CommonReturnParameterReferenceUsageMembership\_Mapping.g
- Element::shortName  
null
- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Feature::isReadOnly  
false



- AssertConstraintUsage::name

```
'assert_' + from.name
```

- Feature::direction

```
null
```

- Feature::isDerived

```
false
```

- Feature::isComposite

```
false
```

#### **C.2.5.6.2.8 Dependency\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

DirectedRelationship\_Mapping

##### **Mapping Source**

Dependency

##### **Mapping Target**

Dependency

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- Element::aliasId

```
Set{ }
```

- Dependency::supplier

```
from.target->collect(e | ElementMain_Mapping.getMapped(e))
```

- Dependency::name

```
from.name
```

- Element::ownedRelationship

```
ElementOwnership_Mapping.getMappedColl(from.ownedElement)
```

- Relationship::source

```
Set{}
```

- Relationship::owningRelatedElement

```
ElementMain_Mapping.getMapped(from.owner)
```

- Dependency::client

```
from.source->collect(e | ElementMain_Mapping.getMapped(e))
```

- Element::elementId

```
Helper.getID(from)
```

- Relationship::ownedRelatedElement

```
from.relatedElement->select(e | from.ownedElement->includes(e))->collect(e | ElementMain_Mapping.getMapped(e))
```

- Element::shortName

```
null
```

- Relationship::target

```
Set{}
```

#### C.2.5.6.2.9 DirectedRelationship\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

Relationship\_Mapping

##### Mapping Source

DirectedRelationship

##### Mapping Target

Relationship

##### Owned Mappings

(none)

##### Applicable filters

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

(none)

## Mapping rules

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

- `Element::aliasId`  
`Set{}`
- `Relationship::ownedRelatedElement`  
`Set{}`
- `Element::ownedRelationship`  
`ElementOwnership_Mapping.getMappedColl(from.ownedElement)`
- `Relationship::target`  
`from.target->collect(e | ElementMain_Mapping.getMapped(e))`
- `Element::elementId`  
`Helper.getID(from)`
- `Element::name`  
`null`
- `Relationship::source`  
`from.source->collect(e | ElementMain_Mapping.getMapped(e))`
- `Element::shortName`  
`null`

### C.2.5.6.2.10 ElementMain\_Mapping

#### Description

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

#### General Mappings

GenericToElement\_Mapping

#### Mapping Source

Element

#### Mapping Target

Element

#### Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::ownedRelationship  
`ElementOwnership_Mapping.getMappedColl(from.ownedElement)`
- Element::elementId  
`Helper.getID(from)`

#### C.2.5.6.2.11 ElementMembership\_Mapping

### Description

\*\*\* not specified yet \*\*\*

### General Mappings

GenericToMembership\_Mapping

### Mapping Source

Element

### Mapping Target

Membership

### Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  
`Set {}`
- Relationship::ownedRelatedElement  
`Set {}`
- Relationship::source

```
Set{}
```

- Membership::membershipOwningNamespace

```
Set{ElementMain_Mapping(from)} -- will not be used since corresponding att is derived, but r
```

- Element::name

```
null
```

- Element::shortName

```
null
```

- Element::elementId

```
Helper.createUUID()
```

- Relationship::target

```
Set{}
```

- Membership::memberElement

```
ElementMain_Mapping.getMapped(from)
```

- Element::ownedRelationship

```
Set{}
```

- Membership::visibility

```
if (from.ocIsKindOf(UML::NamedElement)) then
  from.ocAsType(UML::NamedElement).visibility
else
  KerML::VisibilityKind::public
endif
```

#### C.2.5.6.2.12 ElementOwnership\_Mapping

##### Description

##### General Mappings

GenericToRelationship\_Mapping

##### Mapping Source

Element

##### Mapping Target

Relationship

##### Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  
`Set{}`
- Relationship::target  
`OrderedSet{ElementMain_Mapping.getMapped(from)}`
- Relationship::source  
`OrderedSet{ElementMain_Mapping.getMapped(from.owner)}`
- Element::name  
`null`
- Relationship::ownedRelatedElement  
`self.target()`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set{}`

#### C.2.5.6.2.13 ElementOwningMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

ElementMembership\_Mapping

ElementOwnership\_Mapping

##### Mapping Source

Element

##### Mapping Target

OwningMembership

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Relationship::ownedRelatedElement

Set{}

- Membership::membershipOwningNamespace

*abstract rule*

- OwningMembership::membershipOwningNamespace

Set{ElementMain\_Mapping(from)} -- will not be used since corresponding att is derived, but r

- Membership::memberShortName

null

- Membership::memberElement

*abstract rule*

- Element::shortName

null

- Element::elementId

Helper.createUUID()

- Element::aliasId

Set{}

- OwningMembership::ownedMemberElement

ElementMain\_Mapping.getMapped(from)

- Membership::memberName

null

- OwningMembership::ownedRelatedElement

Set{self.ownedMemberElement() }

- Membership::visibility

KerML::VisibilityKind::public

- Element::name  
null
- Element::ownedRelationship  
Set{ }

#### **C.2.5.6.2.14 NamedElementMain\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

ElementMain\_Mapping

##### **Mapping Source**

NamedElement

##### **Mapping Target**

Element

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Element::aliasId  
Set{ }
- Element::name  
from.name
- Element::shortName  
null
- Element::elementId  
Helper.createUUID()
- Element::ownedRelationship  
Set{ }



#### C.2.5.6.2.15 Namespace\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToNamespace\_Mapping  
NamedElementMain\_Mapping

##### Mapping Source

Namespace

##### Mapping Target

Namespace

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId  
`Set{ }`
- Namespace::ownedImport  
`Set{ }`
- Element::elementId  
`Helper.getID(from)`
- Element::name  
`null`
- Element::shortName  
`null`
- Namespace::ownedRelationship  
`from.ownedElement->collect(e | ElementOwningMembership_Mapping.getMapped(e))`

#### C.2.5.6.2.16 Relationship\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToRelationship\_Mapping  
ElementMain\_Mapping

##### Mapping Source

Relationship

##### Mapping Target

Relationship

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId

Set{ }

- Relationship::owningRelatedElement

ElementMain\_Mapping.getMapped(from.owner)

- Element::name

null

- Relationship::ownedRelatedElement

from.relatedElement->select(e | from.ownedElement->includes(e))->collect(e | ElementMain\_Mapping.getMapped(e))

- Element::shortName

null

- Element::elementId

Helper.createUUID()

- Element::ownedRelationship

Set{ }

#### C.2.5.6.2.17 Usage\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

Dependency\_Mapping

##### Mapping Source

Usage

##### Mapping Target

Dependency

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId  

```
Set{ }
```
- Element::ownedRelationship  

```
ElementOwnership_Mapping.getMappedColl(from.ownedElement)
```
- Relationship::target  

```
from.target->collect(e | ElementMain_Mapping.getMapped(e) )
```
- Relationship::owningRelatedElement  

```
ElementMain_Mapping.getMapped(from.owner)
```
- Element::elementId  

```
Helper.getID(from)
```
- Element::name  

```
null
```
- Relationship::source

```
from.source->collect(e | ElementMain_Mapping.getMapped(e))
```

- Relationship::ownedRelatedElement

```
from.relatedElement->select(e | from.ownedElement->includes(e))->collect(e | ElementMain_Mapping.getMapped(e))
```

- Element::shortName

```
null
```

## C.2.5.7 InformationFlows

### C.2.5.7.1 Overview

**Table 21. List of all Overview Mapping Specifications**

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
InformationFlow	FeatureMembership FeatureMembership Subsetting FeatureTyping Subsetting FeatureMembership ItemFlowEnd ItemFlowEnd FlowConnectionUsage ItemFeature	ItemFlowFeatureMembership_Mapping ItemFlowTargetEndFeatureMembership_Mapping ItemFlowSourceFeatureSubsetting_Mapping ItemFlowItemFeatureTyping_Mapping ItemFlowTargetFeatureSubsetting_Mapping ItemFlowSourceEndFeatureMembership_Mapping ItemFlowSourceFeature_Mapping ItemFlowTargetFeature_Mapping ItemFlow_Mapping ItemFlowItemFeature_Mapping	Helper.hasStereotypeApplied('SysML::Ports&Flows::ItemFlow')
InformationItem	ObjectiveMembership FeatureTyping SubjectMembership PartUsage RequirementUsage ReferenceUsage Classifier StakeholderMembership	CaseObjectiveMembership_Mapping CaseSubjectFeatureTyping_Mapping CaseSubjectMembership_Mapping StakeholderPartUsage_Mapping CaseObjectiveRequirementUsage_Mapping CaseEmptySubjectReferenceUsage_Mapping Classifier_Mapping StakeholderMembership_Mapping	

### C.2.5.7.2 Mapping Specifications

### C.2.5.8 Interactions

#### C.2.5.8.1 Overview

**Table 22. List of all Overview Mapping Specifications**

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
ActionExecutionSpecification	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
BehaviorExecutionSpecification	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	
CombinedFragment	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	
ConsiderIgnoreFragment	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	
Continuation	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	
DestructionOccurrenceSpecification	FeatureMembership EventOccurrenceUsage	MessageOccurrenceSpecificationMembership_Mapping MessageOccurrenceSpecification_Mapping	
ExecutionOccurrenceSpecification	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	
ExecutionSpecification	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	
Gate	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
GeneralOrdering	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	
Interaction	OccurrenceDefinition	Interaction_Mapping	
InteractionConstraint	AssertConstraintUsage ConstraintDefinition FeatureTyping FeatureMembership	ConstraintUsage_Mapping Constraint_Mapping ConstraintUsageFeatureTyping_Mapping ConstrainedElementFeatureMembership_Mapping	
InteractionFragment	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	
InteractionOperand	Namespace	Namespace_Mapping	
InteractionUse	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	
Lifeline	PartUsage FeatureMembership FeatureTyping	LifelinePartUsage_Mapping LifelineMembership_Mapping LifelineFeatureTyping_Mapping	
Message	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	
MessageEnd	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	
MessageOccurrenceSpecification	FeatureMembership EventOccurrenceUsage	MessageOccurrenceSpecificationMembership_Mapping MessageOccurrenceSpecification_Mapping	

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
OccurrenceSpecification	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	
PartDecomposition	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	
StateInvariant	FeatureMembership ReferenceUsage OwningMembership Element Documentation SubjectMembership	ElementFeatureMembership_Mapping RequirementSubject_Mapping RequirementDocumentationMembership_Mapping NamedElementMain_Mapping RequirementDocumentation_Mapping RequirementSubjectMembership_Mapping	

### C.2.5.8.2 Mapping Specifications

#### C.2.5.8.2.1 Interaction\_Mapping

##### Description

A UML4SysML::Interaction is mapped to a SysMLv2::Interaction.

##### General Mappings

ElementMain\_Mapping  
GenericToOccurrenceDefinition\_Mapping

##### Mapping Source

Interaction

##### Mapping Target

OccurrenceDefinition

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId

```
Set{}
```

- Type::isSufficient

```
false
```

- Definition::isVariation

```
false
```

- Element::name

```
null
```

- Element::shortName

```
null
```

- Type::isAbstract

```
false
```

- Element::elementId

```
Helper.createUUID()
```

- OccurrenceDefinition::ownedRelationship

```
let lifelines: Set(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::Lifeline))
let messageOccurrences: Set(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::MessageOccurrence))
let elements: Set(UML::Element) = ((from.ownedElement - lifelines) - messageOccurrences) in
elements->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(lifelines->collect(e | LifelineMembership_Mapping.getMapped(e)))
```

- Element::ownedRelationship

```
Set{}
```

#### **C.2.5.8.2.2 LifelineMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureMembership\_Mapping

##### **Mapping Source**

Lifeline

##### **Mapping Target**



## FeatureMembership

### Owned Mappings

- lifelinePartUsage : LifelinePartUsage\_Mapping

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- FeatureMembership::memberFeature  
  
self.ownedMemberFeature()
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- FeatureMembership::ownedMemberFeature  
  
lifelinePartUsage.to
- FeatureMembership::memberName  
  
from.name
- OwningMembership::ownedRelatedElement  
  
Set{self.ownedMemberElement() }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
  
KerML::VisibilityKind::public

- Element::name

null

- Element::ownedRelationship

Set{}

#### **C.2.5.8.2.3 LifelinePartUsage\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToPartUsage\_Mapping

##### **Mapping Source**

Lifeline

##### **Mapping Target**

PartUsage

##### **Owned Mappings**

- lifelineFeatureTyping : LifelineFeatureTyping\_Mapping
- messageOccurrenceSpecificationMembership : MessageOccurrenceSpecificationMembership\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Feature::isEnd

false

- Type::isSufficient

false

- Feature::isUnique

true

- Element::shortName

null

- Type::isAbstract

- false
- Element::elementId
  - Helper.createUUID()
- Feature::isOrdered
  - false
- Element::aliasId
  - Set{}
- Feature::isPortion
  - false
- Usage::isVariation
  - false
- PartUsage::ownedRelationship
  - Set{lifelineFeatureTyping.to, messageOccurrenceSpecificationMembership.to}
- Feature::isReadOnly
  - false
- Feature::direction
  - null
- Element::name
  - null
- Feature::isDerived
  - false
- Feature::isComposite
  - false

#### **C.2.5.8.2.4 LifelineFeatureTyping\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureTyping\_Mapping

##### **Mapping Source**

Lifeline

## Mapping Target

FeatureTyping

## Owned Mappings

- lifelinePartUsage : LifelinePartUsage\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Element::aliasId  
`Set {}`
- Relationship::ownedRelatedElement  
`Set {}`

- Specialization::specific  
*abstract rule*
- FeatureTyping::type

`from.represents.type`

- Element::name  
`null`
- Element::shortName  
`null`

- Specialization::general  
*abstract rule*
- Element::elementId

`Helper.createUUID()`

- Element::ownedRelationship  
`Set {}`
- FeatureTyping::typedFeature

`lifelinePartUsage.to`

### C.2.5.8.2.5 MessageOccurrenceSpecification\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

## General Mappings

GenericToEventOccurrenceUsage\_Mapping

## Mapping Source

MessageOccurrenceSpecification

## Mapping Target

EventOccurrenceUsage

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- OccurrenceUsage::portionKind  
`OclUndefined`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId

```

    Set{}

    • Feature::isPortion

      false

    • Usage::isVariation

      false

    • Feature::isReadOnly

      false

    • Feature::direction

      null

    • Element::name

      null

    • Feature::isDerived

      false

    • Feature::isComposite

      false

    • OccurrenceUsage::isIndividual

      false

    • Element::ownedRelationship

      Set{}

```

#### **C.2.5.8.2.6 MessageOccurrenceSpecificationMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureMembership\_Mapping

##### **Mapping Source**

MessageOccurrenceSpecification

##### **Mapping Target**

FeatureMembership

##### **Owned Mappings**

- messageOccurrenceSpecification : MessageOccurrenceSpecification\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
    null
- Element::shortName  
  
    null
- Element::elementId  
  
    Helper.createUUID()
- Element::aliasId  
  
    Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
  
    null
- OwningMembership::ownedRelatedElement  
  
    Set{self.ownedMemberElement() }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
  
    KerML::VisibilityKind::public
- Element::name  
  
    null
- FeatureMembership::memberFeature  
  
    self.ownedMemberFeature()
- FeatureMembership::ownedMemberFeature  
  
    messageOccurrenceSpecification.to
- Element::ownedRelationship

Set { }

## C.2.5.9 Packages

### C.2.5.9.1 Overview

**Table 23. List of all Overview Mapping Specifications**

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
Extension	FeatureMembership FeatureMembership Redefinition MetadataFeature FeatureTyping Association Feature FeatureValue Annotation	AssociationToMetadataMembership_Mapping AssociationToFeatureMembership_Mapping AssociationToRedefinition_Mapping AssociationToAnnotatingFeature_Mapping AssociationToFeatureTyping_Mapping AssociationCommon_Mapping AssociationToMetadataFeature_Mapping AssociationToMetadataFeatureValue_Mapping AssociationToAnnotation_Mapping	Extension.memberEnd- >select( m   m.type.ocIsKindOf(UML::UseCase))- >isEmpty()
ExtensionEnd	FeatureTyping FeatureMembership FeatureChaining Subsetting AttributeUsage FeatureChaining OwningMembership EndFeatureMembership Subsetting ActorMembership Redefinition PartUsage Feature	CaseActorFeatureTyping_Mapping NonOnedEndToSubsettedFeatureMembership_Mapping EndToSubsettedFeatureChaining_Mapping NonOwnedEndSubsetting_Mapping OwnedEndAttribute_Mapping PropertyToFeatureChaining_Mapping NonOwnedEndSubsettingMembership_Mapping EndMembership_Mapping PropertySubsetting_Mapping CaseActorMembership_Mapping AttributeRedefinedRedefinition_Mapping CaseActor_Mapping PropertyCommon_Mapping	src.ocIsKindOf(UML::Property) and not src.ocAsType(UML::Property).association.oc UML::Property = src.ocAsType(UML::Property) >includes(p)) and (not p.type.ocIsKindOf(UML::DataType))



SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
Image	FeatureTyping	CommonParameterReferenceUsageInFeatureTyping_Mapping	
	FeatureTyping	Mapping	
	Element	CommonReturnParameterFeatureUntyped_Mapping	
	Feature	CommonReturnParameterFeatureTyping_Mapping	
	FeatureTyping	ElementOwnership_Mapping	
	Relationship	CommonValueSpecification_Mapping	
	Expression	CommonParameterReferenceUsageInUntyped_Mapping	
	ReferenceUsage	DefaultMultiplicityMembership_Mapping	
	OwningMembership	CommonReturnParameterFeatureMembership_Mapping	
	LiteralInteger	CommonParameterReferenceUsageInMembership_Mapping	
	ReturnParameterMembership	DefaultMultiplicityBoundOwnership_Mapping	
	ParameterMembership	CommonReturnParameterReferenceUsageFeatureTyping_Mapping	
	FeatureMembership	DefaultUpperBound_Mapping	
	FeatureTyping	DefaultMultiplicityElement_Mapping	
	LiteralInteger	CommonReturnParameterReferenceUsageUntyped_Mapping	
	MultiplicityRange	DefaultLowerBound_Mapping	
	ReferenceUsage	ElementMain_Mapping	
	LiteralInteger	ElementMembership_Mapping	
	Element	CommonReturnParameterReferenceUsageMembership_Mapping	
	Membership	EmptyReturnParameterFeatureMembership_Mapping	
	ReturnParameterMembership	ReturnParameterMembership	
Model	LiteralString	ModelViewpointValue_Mapping	
	Redefinition	ModelViewpointMetadataRedefinition_Mapping	
	FeatureTyping	ModelViewpointMetadataFeatureTyping_Mapping	
	Package	Model_Mapping	
	FeatureValue	ModelViewpointMetadataFeatureValue_Mapping	
	ReferenceUsage	ModelViewpointMetadataReferenceUsage_Mapping	
	FeatureMembership	ModelViewpointMetadataFeatureMembership_Mapping	
	MetadataUsage	ModelViewpointMetadataUsage_Mapping	
	OwningMembership	ModelViewpointMetadataMembership_Mapping	
Package	FeatureMembership	PackageURIFeatureMembership_Mapping	
	FeatureValue	PackageURIMetadataFeatureValue_Mapping	
	FeatureTyping	PackageURIFeatureTyping_Mapping	
	OwningMembership	PackageURIMetadataMembership_Mapping	
	Package	Package_Mapping	
	MetadataUsage	PackageURIMetadataUsage_Mapping	
	LiteralString	PackageURIValue_Mapping	
	Redefinition	PackageURIRedefinition_Mapping	
	ReferenceUsage	PackageURIMetadataReferenceUsage_Mapping	
PackageMerge	Relationship	DirectedRelationship_Mapping	
Profile	Package	Profile_Mapping	
ProfileApplication	Relationship	DirectedRelationship_Mapping	

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
Stereotype	OccurrenceDefinition	StereotypeOccurrenceDefinition_Mapping	
	OwningMembership	StereotypeMetadataDefinitionMembership_Mapping	
	Subclassification	StereotypeMetadataDefinitionSubclassification_Mapping	
	ReturnParameterMembership	StereotypeOccurrenceUsageMultiplicityRangeInfinityReturnParameterMem	
	FeatureValue	StereotypeMetadataDefinitionReferenceUsageFeatureValue_Mapping	
	FeatureTyping	StereotypeOccurrenceUsageFeatureTyping_Mapping	
	Membership	StereotypeMetadataDefinitionReferenceUsageFeatureReferenceExpressionl	
	Membership	StereotypeOccurrenceUsageMembership_Mapping	
	Feature	StereotypeMetadataDefinitionReferenceUsageFeatureReferenceExpressionl	
	FeatureReferenceExpression	StereotypeMetadataDefinitionReferenceUsageFeatureReferenceExpressionl	
	FeatureMembership	StereotypeMetadataDefinitionReferenceUsageFeatureMembershipUsage_M	
	FeatureMembership	StereotypeMetadataDefinitionFeatureMembership_Mapping	
	OwningMembership	StereotypeOccurrenceDefinitionMembership_Mapping	
	Redefinition	StereotypeMetadataDefinitionReferenceUsageRedefinition_Mapping	
	FeatureTyping	StereotypeMetadataDefinitionReferenceUsageFeatureMembershipUsageFe	
	OccurrenceUsage	StereotypeOccurrenceUsage_Mapping	
	OperatorExpression	StereotypeMetadataDefinitionReferenceUsageOperatorExpression_Mappin	
	LiteralInfinity	StereotypeOccurrenceUsageMultiplicityRangeInfinity_Mapping	
	Membership	StereotypeOccurrenceUsageMultiplicityRangeMembership_Mapping	
	FeatureMembership	StereotypeMetadataDefinitionReferenceUsageFeatureMembershipReferenc	
	MultiplicityRange	StereotypeOccurrenceUsageMultiplicityRange_Mapping	
	MetadataDefinition	StereotypeMetadataDefinition_Mapping	
	ReferenceUsage	StereotypeMetadataDefinitionReferenceUsage_Mapping	
	ReturnParameterMembership	StereotypeMetadataDefinitionReferenceUsageFeatureReferenceExpressionl	
	Feature	StereotypeOccurrenceUsageMultiplicityRangeInfinityReturnParameter_Ma	
	Membership	StereotypeOccurrenceUsageMultiplicityMembership_Mapping	
	Feature	StereotypeMetadataDefinitionReferenceUsageFeatureMembershipUsageFe	

#### C.2.5.9.2 UML4SysML Packages elements not mapped

Table 24. List of SysML v1 elements not mapped of this section

SysML v1 Concept	Rationale
Extension	The mapping of the extension relationship is performed in the context of Stereotype_Mapping.
ExtensionEnd	The mapping of the extension end property is performed in the context of Stereotype_Mapping.
PackageMerge	The concept of the PackageMerge relationship is not supported by SysML v2.

#### C.2.5.9.3 Mapping Specifications

##### C.2.5.9.3.1 ElementImport\_Mapping

###### Description

\*\*\* not specified yet \*\*\*

###### General Mappings

GenericToMembership\_Mapping

DirectedRelationship\_Mapping

## Mapping Source

ElementImport

## Mapping Target

Membership

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Membership::memberElement  
`ElementMain_Mapping.getMapped(from.importedElement)`
- Element::ownedRelationship  
`ElementOwnership_Mapping.getMappedColl(from.ownedElement)`
- Relationship::owningRelatedElement  
`ElementMain_Mapping.getMapped(from.owner)`
- Element::shortName  
`null`
- Relationship::target  
`Set{}`
- Element::aliasId  
`Set{}`
- Membership::visibility  
`Helper.getKerMLVisibilityKind(from.visibility)`
- Membership::aliases  
`from.alias->asSet()`
- Membership::membershipOwningPackage  
`Namespace_Mapping.getMapped(from.importingNamespace)`
- Relationship::source

```
Set{}
```

- Element::elementId

```
Helper.getID(from)
```

- Element::name

```
null
```

- Relationship::ownedRelatedElement

```
from.relatedElement->select(e | from.ownedElement->includes(e))->collect(e | ElementMain_Map
```

- Membership::memberName

```
from.importedElement.name
```

### C.2.5.9.3.2 Package\_Mapping

#### Description

A UML::Package is mapped to a SysMLv2::Package. The property "URI" is mapped to a metadata if it has a value. The expected SysML v2 textual notation of a SysMLv1::Package is as follows:

```
package ThisIsAPackageWithURI {  
  metadata SysMLv1Library::PackageData {URI="https://omg.org";}   
}
```

#### General Mappings

Namespace\_Mapping

#### Mapping Source

Package

#### Mapping Target

Package

#### Owned Mappings

(none)

#### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

#### Mapping rules

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

- Element::aliasId

```
Set{}
```

- Element::ownedRelationship

```
ElementOwnership_Mapping.getMappedColl(from.ownedElement)
```

- Package::ownedRelationship

```
Helper.packageOwnedRelationship(from)
```

- Element::elementId

```
Helper.getID(from)
```

- Element::name

```
from.name
```

- Element::shortName

```
null
```

### C.2.5.9.3.3 PackageImport\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

DirectedRelationship\_Mapping

#### Mapping Source

PackageImport

#### Mapping Target

Import

#### Owned Mappings

(none)

#### Applicable filters

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

(none)

#### Mapping rules

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

- Import::importOwningPackage

```
Namespace_Mapping.getMapped(from.importingNamespace)
```

- Element::ownedRelationship

```
ElementOwnership_Mapping.getMappedColl(from.ownedElement)
```

- Relationship::owningRelatedElement

- `Import::importedPackage`

- `Element::shortName`

- Import::visibility

- Relationship::target

- `Element::aliasId`

- Relationship::source

- `Element::elementId`

- `Element::name`

- Relationship::ownedRelatedElement

```
from.relatedElement->select(e | from.ownedElement->includes(e))->collect(e | ElementMain_Map
```

### Description

```
package ThisIsAModel {
  metadata SysMLLv1Library::PackageData {URI="https://omg.org";}
  metadata SysMLLv1Library::ModelData {'viewpoint'="thisIsTheViewpointOfTheModel";}
}
```

## Package\_Mapping

### Mapping Source

Model

## Mapping Target

Package

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Element::aliasId

Set{}

- Package::ownedRelationship

```
let relationships : Set(KerML::Relationship) = Helper.packageOwnedRelationship(from) in
if from.viewpoint.ocIsUndefined() or from.viewpoint = '' then
    relationships
else
    relationships->including(ModelViewpointMetadataMembership_Mapping.getMapped(from))
endif
```

- Namespace::ownedImport

Set{}

- Element::elementId

Helper.getID(from)

- Element::name

from.name

- Element::shortName

null

- Namespace::ownedRelationship

```
from.ownedElement->collect(e | ElementOwningMembership_Mapping.getMapped(e))
```

### C.2.5.9.3.5 ModelViewpointMetadataUsage\_Mapping

### C.2.5.9.3.6 ModelViewpointMetadataFeatureMembership\_Mapping

## Description

The mapping class creates the feature membership relationship for the metadata feature to store the UML::Model::viewpoint property.

### General Mappings

GenericToFeatureMembership\_Mapping

### Mapping Source

Model

### Mapping Target

FeatureMembership

### Owned Mappings

(none)

### Applicable filters

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

(none)

### Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- FeatureMembership::ownedMemberFeature  
  
ModelViewpointMetadataReferenceUsage\_Mapping.getMapped(from)
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
  
null
- OwningMembership::ownedRelatedElement



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

- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility

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

- Element::name

```
null
```

- Element::ownedRelationship

```
Set{ }
```

#### **C.2.5.9.3.7 ModelViewpointMetadataReferenceUsage\_Mapping**

##### **Description**

The mapping class creates the MetadataFeature for the mapping of the property UML::Model::viewpoint.

##### **General Mappings**

GenericToReferenceUsage\_Mapping

##### **Mapping Source**

Model

##### **Mapping Target**

ReferenceUsage

##### **Owned Mappings**

- modelViewpointMetadataRedefinition : ModelViewpointMetadataRedefinition\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Feature::isEnd  

```
false
```
- Type::isSufficient  

```
false
```

- Feature::isUnique  
true
- Element::shortName  
null
- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Usage::isVariation  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false
- ReferenceUsage::ownedRelationship  
Set{modelViewpointMetadataRedefinition.to, ModelViewpointMetadataFeatureValue\_Mapping.getMap

#### **C.2.5.9.3.8 ModelViewpointMetadataFeatureTyping\_Mapping**

##### **Description**

The mapping class creates the FeatureTyping relationship for the AnnotatingFeature for the metadata to store the UML::Model::viewpoint property.

### General Mappings

GenericToFeatureTyping\_Mapping

### Mapping Source

Model

### Mapping Target

FeatureTyping

### Owned Mappings

- modelViewpointMetadataUsage : ModelViewpointMetadataUsage\_Mapping

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId

Set{ }

- Relationship::ownedRelatedElement

Set{ }

- FeatureTyping::typedFeature

modelViewpointMetadataUsage.to

- Specialization::specific

*abstract rule*

- Element::name

null

- Element::shortName

null

- Specialization::general

*abstract rule*

- FeatureTyping::type

SysMLv2::MetadataDefinition.allInstances()->any(m | m.qualifiedName = 'SysMLv1Library::Model')

- Element::elementId

```
Helper.createUUID()
```

- Element::ownedRelationship

```
Set{}
```

#### **C.2.5.9.3.9 ModelViewpointMetadataMembership\_Mapping**

##### **Description**

The mapping class creates a membership relationship for the metadata feature value for the UML::Model::viewpoint property.

##### **General Mappings**

GenericToOwningMembership\_Mapping

##### **Mapping Source**

Model

##### **Mapping Target**

OwningMembership

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- Relationship::ownedRelatedElement

```
Set{}
```

- Membership::membershipOwningNamespace

*abstract rule*

- Membership::memberShortName

```
null
```

- Membership::memberElement

*abstract rule*

- OwningMembership::ownedMemberElement

```
ModelViewpointMetadataUsage_Mapping.getMapped(from)
```

- Element::shortName

```
null
```

- Element::elementId  
Helper.createUUID()
- Element::aliasId  
Set{}
- Membership::memberName  
null
- Membership::visibility  
KerML::VisibilityKind::public
- Element::name  
null
- Element::ownedRelationship  
Set{}

#### **C.2.5.9.3.10 ModelViewpointMetadataFeatureValue\_Mapping**

##### **Description**

The mapping class maps the value of the property UML::Model::viewpoint.

##### **General Mappings**

GenericToFeatureValue\_Mapping

##### **Mapping Source**

Model

##### **Mapping Target**

FeatureValue

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Membership::membershipOwningNamespace  
*abstract rule*

- Membership::memberShortName  
null
- Element::shortName  
null
- Element::elementId  
Helper.createUUID()
- Element::aliasId  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
null
- OwningMembership::ownedRelatedElement  
Set{self.ownedMemberElement() }
- Membership::visibility  
KerML::VisibilityKind::public
- Element::name  
null
- FeatureValue::value  
ModelViewpointValue\_Mapping.getMapped(from)
- Element::ownedRelationship  
Set{}

#### **C.2.5.9.3.11 ModelViewpointMetadataRedefinition\_Mapping**

##### **Description**

The mapping class creates the redefinition of the attribute for the metadata UML::Model::viewpoint.

##### **General Mappings**

GenericToRedefinition\_Mapping

##### **Mapping Source**

Model

##### **Mapping Target**

Redefinition

## Owned Mappings

- `modelViewpointMetadataReferenceUsage : ModelViewpointMetadataReferenceUsage_Mapping`

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- `Element::aliasId`

```
Set{}
```

- `Redefinition::redefinedFeature`

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

- `Redefinition::redefiningFeature`

```
modelViewpointMetadataReferenceUsage.to
```

- `Subsetting::ownedRelatedElement`

```
Set{}
```

- `Subsetting::subsettingFeature`

*abstract rule*

- `Element::name`

```
null
```

- `Subsetting::subsettingFeature`

*abstract rule*

- `Element::shortName`

```
null
```

- `Element::elementId`

```
Helper.createUUID()
```

- `Element::ownedRelationship`

```
Set{}
```

### C.2.5.9.3.12 ModelViewpointValue\_Mapping

## Description

The mapping class maps the value expression of the property `UML::Model::viewpoint`.

## General Mappings

GenericToExpression\_Mapping

### Mapping Source

Model

### Mapping Target

LiteralString

### Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`



- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false
- Element::ownedRelationship  
Set{}
- LiteralString::value  
from.viewpoint

#### **C.2.5.9.3.13 PackageURIMetadataUsage\_Mapping**

##### **Description**

The mapping class creates the annotating feature to annotate the generated Package element with metadata to store the UML::Package::URI property.

##### **General Mappings**

GenericToMetadataUsage\_Mapping

##### **Mapping Source**

Package

##### **Mapping Target**

MetadataUsage

##### **Owned Mappings**

- packageURIFeatureTyping : PackageURIFeatureTyping\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- `Feature::isEnd`  
`false`
- `Type::isSufficient`  
`false`
- `MetadataUsage::name`  
`'URI'`
- `Feature::isUnique`  
`true`
- `Element::shortName`  
`null`
- `Type::isAbstract`  
`false`
- `Element::elementId`  
`Helper.createUUID()`
- `Feature::isOrdered`  
`false`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `Usage::isVariation`  
`false`
- `Feature::isReadOnly`  
`false`
- `Feature::direction`  
`null`
- `MetadataUsage::ownedRelationship`  
`Set{packageURIFeatureTyping.to, PackageURIFeatureMembership_Mapping.getMapped(from)}`
- `Feature::isDerived`

false

- Feature::isComposite

false

#### C.2.5.9.3.14 PackageURIFeatureMembership\_Mapping

##### Description

The mapping class creates the feature membership relationship for the metadata feature to store the UML::Package::URI property.

##### General Mappings

GenericToFeatureMembership\_Mapping

##### Mapping Source

Package

##### Mapping Target

FeatureMembership

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- FeatureMembership::ownedMemberFeature

`PackageURIMetadataReferenceUsage_Mapping.getMapped(from)`

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName

null

- Element::shortName

null

- Element::elementId

`Helper.createUUID()`

- Element::aliasId

```

Set {}

• OwningMembership::ownedMemberElement
  abstract rule
• Membership::memberName

  null

• OwningMembership::ownedRelatedElement

  Set { self.ownedMemberElement () }

• TypeFeaturing::featureOfType
  abstract rule
• TypeFeaturing::featuringType
  abstract rule
• Membership::visibility

  KerML::VisibilityKind::public

• Element::name

  null

• Element::ownedRelationship

  Set {}

```

#### C.2.5.9.3.15 PackageURIFeatureTyping\_Mapping

##### Description

The mapping class creates the FeatureTyping relationship for the AnnotatingFeature for the metadata to store the UML::Package::URI property.

##### General Mappings

GenericToFeatureTyping\_Mapping

##### Mapping Source

Package

##### Mapping Target

FeatureTyping

##### Owned Mappings

- packageURIMetadataUsage : PackageURIMetadataUsage\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId
 

```
Set{}
```
- Relationship::ownedRelatedElement
 

```
Set{}
```
- Specialization::specific
 

*abstract rule*
- FeatureTyping::type
 

```
let m: SysMLv2::Membership = SysMLv2::AttributeDefinition.allInstances()
->collect(dt | dt.owningRelationship)
->select(r | r.ocIsKindOf(SysMLv2::Membership))
->any(m | m.memberName = 'PackageData' ) in

if (m.ocIsUndefined()) then
    invalid
else
    m.memberElement
endif
```
- Element::name
 

```
null
```
- Element::shortName
 

```
null
```
- Specialization::general
 

*abstract rule*
- FeatureTyping::typedFeature
 

```
packageURIMetadataUsage.to
```
- Element::elementId
 

```
Helper.createUUID()
```
- Element::ownedRelationship
 

```
Set{}
```

#### C.2.5.9.3.16 PackageURIMetadataReferenceUsage\_Mapping

##### Description

The mapping class creates the MetadataFeature for the mapping of the property UML::Package::URI.

##### General Mappings

GenericToReferenceUsage\_Mapping

##### Mapping Source

Package

## Mapping Target

ReferenceUsage

## Owned Mappings

- packageURIMetadataFeatureValue : PackageURIMetadataFeatureValue\_Mapping
- packageURIRedefinition : PackageURIRedefinition\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Usage::isVariation

- false
- Feature::isReadOnly
  - false
- ReferenceUsage::ownedRelationship
  - Set{packageURIRedefinition.to, packageURIMetadataFeatureValue.to}
- Feature::direction
  - null
- Element::name
  - null
- Feature::isDerived
  - false
- Feature::isComposite
  - false

#### **C.2.5.9.3.17 PackageURIMetadataFeatureValue\_Mapping**

##### **Description**

The mapping class maps the value of the property UML::Package::URI.

##### **General Mappings**

GenericToFeatureValue\_Mapping

##### **Mapping Source**

Package

##### **Mapping Target**

FeatureValue

##### **Owned Mappings**

- packageURIMetadataReferenceUsage : PackageURIMetadataReferenceUsage\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- FeatureValue::featureWithValue

- packageURIMetadataReferenceUsage.to
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
null
- Element::shortName  
null
- Element::elementId  
Helper.createUUID()
- FeatureValue::value  
PackageURIValue\_Mapping.getMapped(from)
- Element::aliasId  
Set{ }
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
null
- OwningMembership::ownedRelatedElement  
Set{ self.ownedMemberElement() }
- Membership::visibility  
KerML::VisibilityKind::public
- Element::name  
null
- Element::ownedRelationship  
Set{ }

#### **C.2.5.9.3.18 PackageURIMetadataMembership\_Mapping**

##### **Description**

The mapping class creates a membership relationship for the metadata feature value for the UML::Package::URI property.

##### **General Mappings**

GenericToOwningMembership\_Mapping

##### **Mapping Source**



Package

## Mapping Target

OwningMembership

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Relationship::ownedRelatedElement  
`Set{}`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Membership::memberElement  
*abstract rule*
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{}`
- OwningMembership::ownedMemberElement  
`PackageURIMetadataUsage_Mapping.getMapped(from)`
- Membership::memberName  
`null`
- Membership::visibility  
`KerML::VisibilityKind::public`
- Element::name  
`null`

- Element::ownedRelationship

```
Set{}
```

### C.2.5.9.3.19 PackageURIRedefinition\_Mapping

#### Description

The mapping class creates the redefinition of the attribute for the metadata UML::Package::URI.

#### General Mappings

GenericToRedefinition\_Mapping

#### Mapping Source

Package

#### Mapping Target

Redefinition

#### Owned Mappings

- packageURIMetadataReferenceUsage : PackageURIMetadataReferenceUsage\_Mapping

#### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

#### Mapping rules

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

- Element::aliasId

```
Set{}
```

- Redefinition::redefinedFeature

```
let m : SysMLv2::Membership = SysMLv2::AttributeUsage.allInstances()->collect(dt | dt.owning
if (m.oclisUndefined()) then invalid else m.memberElement endif
```

- Subsetting::ownedRelatedElement

```
Set{}
```

- Redefinition::redefiningFeature

```
packageURIMetadataReferenceUsage.to
```

- Subsetting::subsettingFeature

*abstract rule*

- Element::name

```
null
```

- Subsetting::subsettingFeature  
*abstract rule*
- Element::shortName

null

- Element::elementId

Helper.createUUID()

- Element::ownedRelationship

Set{}

#### **C.2.5.9.3.20 PackageURIValue\_Mapping**

##### **Description**

The mapping class maps the value expression of the property UML::Package::URI.

##### **General Mappings**

GenericToExpression\_Mapping

##### **Mapping Source**

Package

##### **Mapping Target**

LiteralString

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- LiteralString::value

from.URI

- Feature::isEnd

false

- Type::isSufficient

false

- Feature::isUnique  
true
- Element::shortName  
null
- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false
- Element::ownedRelationship  
Set{}

#### **C.2.5.9.3.21 Profile\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

Package\_Mapping

## Mapping Source

Profile

## Mapping Target

Package

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Element::aliasId

```
Set{}
```

- Package::ownedRelationship

```
let stereotypes: Sequence(UML::Element) = src.ownedElement->select(e | e.oclIsKindOf(UML::Stereotype))
let relationships: Set(SysMLv2::Relationship) = from.ownedElement
->reject(e | e.oclIsKindOf(UML::ProfileApplication)
or e.oclIsKindOf(UML::GeneralizationSet)
or e.oclIsKindOf(UML::SignalEvent)
or e.oclIsKindOf(UML::CallEvent)
or e.oclIsKindOf(UML::Extension)
or e.oclIsKindOf(UML::PackageMerge)
or e.oclIsKindOf(UML::Stereotype))
->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(stereotypes->collect(e | StereotypeMetadataDefinitionMembership_Mapping.getMapped(e)))
->union(stereotypes->collect(e | StereotypeOccurrenceDefinitionMembership_Mapping.getMapped(e)))
->union(stereotypes->collect(e | StereotypeOccurrenceUsageMembership_Mapping.getMapped(e))) in

if from.URI.oclIsUndefined() then
    relationships
else
    relationships->including(PackageURIMetadataMembership_Mapping.getMapped(from))
endif
```

- Namespace::ownedImport

```
Set{}
```

- Element::elementId

```
Helper.getID(from)
```

- Element::name

```
from.name
```

- Element::shortName

```
null
```

- Namespace::ownedRelationship

```
from.ownedElement->collect(e | ElementOwningMembership_Mapping.getMapped(e))
```

#### C.2.5.9.3.22 StereotypeMetadataDefinition\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToClassifier\_Mapping

##### Mapping Source

Stereotype

##### Mapping Target

MetadataDefinition

##### Owned Mappings

- stereotypeMetadataDefinitionFeatureMembership :  
StereotypeMetadataDefinitionFeatureMembership\_Mapping
- stereotypeMetadataDefinitionSubclassification : StereotypeMetadataDefinitionSubclassification\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId

```
Set{}
```

- Type::isSufficient

```
false
```

- MetadataDefinition::name

```
if from.name.ocIsUndefined() then 'UnnamedMetadata' else from.name + 'Metadata' endif
```

- MetadataDefinition::ownedRelationship

```
Set{stereotypeMetadataDefinitionSubclassification.to, stereotypeMetadataDefinitionFeatureMem
```

- Type::isAbstract

```
false
```

- Element::elementId

```
Helper.createUUID()
```

- MetadataDefinition::humanId

```
from.name.substring(1,1).toLowerCase() + from.name.substring(2, from.name.size())
```

### C.2.5.9.3.23 StereotypeMetadataDefinitionFeatureMembership\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

GenericToFeatureMembership\_Mapping

#### Mapping Source

Stereotype

#### Mapping Target

FeatureMembership

#### Owned Mappings

- stereotypeMetadataDefinitionReferenceUsage : StereotypeMetadataDefinitionReferenceUsage\_Mapping

#### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

#### Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
null
- Element::shortName

- `Element::elementId`

- FeatureMembership::ownedMemberFeature

- `Element::aliasId`

- `OwningMembership::ownedMemberElement`

- Membership::memberName

- `OwningMembership::ownedRelatedElement`

- `TypeFeaturing::featureOfType`

- `TypeFeaturing::featuringType`

- Membership::visibility

- `Element::name`

- `Element::ownedRelationship`

#### C.2.5.9.3.24 StereotypeMetadataDefinitionMembership\_Mapping

### Description

\*\*\* not specified yet \*\*\*

## General Mappings

## ElementOwningMembership Mapping

## Mapping Source

## Stereotype

## Mapping Target

## OwningMembership



## Owned Mappings

- stereotypeMetadataDefinition : StereotypeMetadataDefinition\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Membership::memberShortName

```
null
```

- Element::shortName

```
null
```

- Element::elementId

```
Helper.createUUID()
```

- Membership::visibility

```
if (from.ocIsKindOf(UML::NamedElement)) then
    from.ocAsType(UML::NamedElement).visibility
else
    KerML::VisibilityKind::public
endif
```

- Element::aliasId

```
Set{}
```

- Relationship::target

```
OrderedSet{ElementMain_Mapping.getMapped(from)}
```

- Relationship::source

```
OrderedSet{ElementMain_Mapping.getMapped(from.owner)}
```

- Membership::memberName

```
null
```

- Membership::membershipOwningNamespace

```
Set{ElementMain_Mapping(from)} -- will not be used since corresponding att is derived, but n
```

- Element::name

```
null
```

- Relationship::ownedRelatedElement

```
self.target()
```

- OwningMembership::ownedMemberElement

```
stereotypeMetadataDefinition.to
```

- Membership::memberElement

```
ElementMain_Mapping.getMapped(from)
```

- Element::ownedRelationship

```
Set{}
```

#### **C.2.5.9.3.25 StereotypeMetadataDefinitionReferenceUsage\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToReferenceUsage\_Mapping

##### **Mapping Source**

Stereotype

##### **Mapping Target**

ReferenceUsage

##### **Owned Mappings**

- stereotypeMetadataDefinitionReferenceUsageFeatureValue :  
StereotypeMetadataDefinitionReferenceUsageFeatureValue\_Mapping
- stereotypeMetadataDefinitionReferenceUsageRedefinition :  
StereotypeMetadataDefinitionReferenceUsageRedefinition\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Feature::isEnd

```
false
```

- Type::isSufficient

```
false
```

- ReferenceUsage::ownedRelationship

Set{stereotypeMetadataDefinitionReferenceUsageRedefinition.to, stereotypeMetadataDefinitionR

- Feature::isUnique

true

- Element::shortName

null

- Type::isAbstract

false

- Element::elementId

Helper.createUUID()

- Feature::isOrdered

false

- Element::aliasId

Set{}

- Feature::isPortion

false

- Usage::isVariation

false

- Feature::isReadOnly

false

- Feature::direction

null

- Element::name

null

- Feature::isDerived

false

- Feature::isComposite

false

#### **C.2.5.9.3.26 StereotypeMetadataDefinitionReferenceUsageFeatureMembershipReference\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

## General Mappings

GenericToFeatureMembership\_Mapping

## Mapping Source

Stereotype

## Mapping Target

FeatureMembership

## Owned Mappings

- stereotypeMetadataDefinitionReferenceUsageFeatureReferenceExpression :  
StereotypeMetadataDefinitionReferenceUsageFeatureReferenceExpression\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- FeatureMembership::visibility  
`KerML::VisibilityKind::private`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{}`
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
`null`
- OwningMembership::ownedRelatedElement  
`Set{self.ownedMemberElement() }`

- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Element::name  
  
null
- FeatureMembership::ownedMemberFeature  
  
stereotypeMetadataDefinitionReferenceUsageFeatureReferenceExpression.to
- Element::ownedRelationship  
  
Set{ }

#### **C.2.5.9.3.27 StereotypeMetadataDefinitionReferenceUsageFeatureMembershipUsage\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureMembership\_Mapping

##### **Mapping Source**

Stereotype

##### **Mapping Target**

FeatureMembership

##### **Owned Mappings**

- stereotypeMetadataDefinitionReferenceUsageFeatureMembershipUsageFeature :  
StereotypeMetadataDefinitionReferenceUsageFeatureMembershipUsageFeature\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- FeatureMembership::ownedMemberFeature  
  
stereotypeMetadataDefinitionReferenceUsageFeatureMembershipUsageFeature.to

- Element::shortName  
null
- Element::elementId  
Helper.createUUID()
- Element::aliasId  
Set{ }
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
null
- OwningMembership::ownedRelatedElement  
Set{ self.ownedMemberElement() }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
KerML::VisibilityKind::public
- Element::name  
null
- Element::ownedRelationship  
Set{ }

#### **C.2.5.9.3.28 StereotypeMetadataDefinitionReferenceUsageFeatureMembershipUsageFeature\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeature\_Mapping

##### **Mapping Source**

Stereotype

##### **Mapping Target**

Feature

## Owned Mappings

- stereotypeMetadataDefinitionReferenceUsageFeatureMembershipUsageFeatureTyping :  
StereotypeMetadataDefinitionReferenceUsageFeatureMembershipUsageFeatureTyping\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Element::aliasId  
`Set{}`
- Type::isSufficient  
`false`
- Feature::ownedRelationship  
`Set{stereotypeMetadataDefinitionReferenceUsageFeatureMembershipUsageFeatureTyping.to}`
- Element::name  
`null`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`

### C.2.5.9.3.29 StereotypeMetadataDefinitionReferenceUsageFeatureMembershipUsageFeatureTyping\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

GenericToFeatureTyping\_Mapping

#### Mapping Source

Stereotype

#### Mapping Target

FeatureTyping

## Owned Mappings

- stereotypeMetadataDefinitionReferenceUsageFeatureMembershipUsageFeature :  
StereotypeMetadataDefinitionReferenceUsageFeatureMembershipUsageFeature\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Element::aliasId  

```
Set {}
```
- Relationship::ownedRelatedElement  

```
Set {}
```
- Specialization::specific  
*abstract rule*
- FeatureTyping::typedFeature  

```
stereotypeMetadataDefinitionReferenceUsageFeatureMembershipUsageFeature.to
```
- Element::name  

```
null
```
- Element::shortName  

```
null
```
- FeatureTyping::type  

```
SysMLv2::Package.allInstances()->collect(dt | dt.owningRelationship)->select(r | r.ocIsKindOf)
```
- Specialization::general  
*abstract rule*
- Element::elementId  

```
Helper.createUUID()
```
- Element::ownedRelationship  

```
Set {}
```

### C.2.5.9.3.30 StereotypeMetadataDefinitionReferenceUsageFeatureReferenceExpression\_Mapping

#### Description

\*\*\* not specified yet \*\*\*



## General Mappings

GenericToFeatureReferenceExpression\_Mapping

## Mapping Source

Stereotype

## Mapping Target

FeatureReferenceExpression

## Owned Mappings

- stereotypeMetadataDefinitionReferenceUsageFeatureReferenceExpressionMembership :  
StereotypeMetadataDefinitionReferenceUsageFeatureReferenceExpressionMembership\_Mapping
- stereotypeMetadataDefinitionReferenceUsageFeatureReferenceExpressionReturnParameterMembership :  
StereotypeMetadataDefinitionReferenceUsageFeatureReferenceExpressionReturnParameterMembership\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- FeatureReferenceExpression::ownedRelationship  
`Set{stereotypeMetadataDefinitionReferenceUsageFeatureReferenceExpressionMembership.to, stereotypeMetadataDefinitionReferenceUsageFeatureReferenceExpressionReturnParameterMembership.to}`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered

false

- Element::aliasId

Set{ }

- Feature::isPortion

false

- Feature::isReadOnly

false

- Feature::direction

null

- Element::name

null

- Feature::isDerived

false

- Feature::isComposite

false

#### **C.2.5.9.3.31**

#### **StereotypeMetadataDefinitionReferenceUsageFeatureReferenceExpressionMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToMembership\_Mapping

##### **Mapping Source**

Stereotype

##### **Mapping Target**

Membership

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- `Element::aliasId`  
`Set{}`
- `Membership::memberElement`  
`StereotypeOccurrenceUsageMembership_Mapping.getMapped(from)`
- `Relationship::ownedRelatedElement`  
`Set{}`
- `Relationship::source`  
`Set{}`
- `Element::name`  
`null`
- `Element::shortName`  
`null`
- `Element::elementId`  
`Helper.createUUID()`
- `Relationship::target`  
`Set{}`
- `Element::ownedRelationship`  
`Set{}`

### C.2.5.9.3.32

#### **StereotypeMetadataDefinitionReferenceUsageFeatureReferenceExpressionReturnParameter\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeature\_Mapping

##### **Mapping Source**

Stereotype

##### **Mapping Target**

Feature

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Element::aliasId  
`Set{}`
- Type::isSufficient  
`false`
- Feature::direction  
`SysMLv2::FeatureDirectionKind::out`
- Element::name  
`null`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set{}`

### C.2.5.9.3.33

#### StereotypeMetadataDefinitionReferenceUsageFeatureReferenceExpressionReturnParameterMembership\_Mapping

## Description

\*\*\* not specified yet \*\*\*

## General Mappings

GenericToReturnParameterMembership\_Mapping

## Mapping Source

Stereotype

## Mapping Target

ReturnParameterMembership

## Owned Mappings

- stereotypeMetadataDefinitionReferenceUsageFeatureReferenceExpressionReturnParameter :  
StereotypeMetadataDefinitionReferenceUsageFeatureReferenceExpressionReturnParameter\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- ReturnParameterMembership::ownedMemberParameter  
  
stereotypeMetadataDefinitionReferenceUsageFeatureReferenceExpressionReturnParameter.to
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{ }
- FeatureMembership::owningType  
*abstract rule*
- Membership::memberName  
  
null
- ParameterMembership::ownedRelatedElement  
  
Set{ self.ownedMemberParameter() }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility

`KerML::VisibilityKind::public`

- `Element::name`

`null`

- `Element::ownedRelationship`

`Set{}`

#### **C.2.5.9.3.34 StereotypeMetadataDefinitionReferenceUsageFeatureValue\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

`GenericToFeatureValue_Mapping`

##### **Mapping Source**

`Stereotype`

##### **Mapping Target**

`FeatureValue`

##### **Owned Mappings**

- `stereotypeMetadataDefinitionReferenceUsageOperatorExpression :`  
`StereotypeMetadataDefinitionReferenceUsageOperatorExpression_Mapping`

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- `Membership::membershipOwningNamespace`  
*abstract rule*
- `Membership::memberShortName`

`null`

- `Element::shortName`

`null`

- `Element::elementId`

`Helper.createUUID()`

- `Element::aliasId`

Set{}

- FeatureValue::value

Set{stereotypeMetadataDefinitionReferenceUsageOperatorExpression.to}

- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName

null

- OwningMembership::ownedRelatedElement

Set{self.ownedMemberElement() }

- Membership::visibility

KerML::VisibilityKind::public

- Element::name

null

- Element::ownedRelationship

Set{}

#### C.2.5.9.3.35 StereotypeMetadataDefinitionReferenceUsageOperatorExpression\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToExpression\_Mapping

##### Mapping Source

Stereotype

##### Mapping Target

OperatorExpression

##### Owned Mappings

- stereotypeMetadataDefinitionReferenceUsageFeatureMembershipReference :  
StereotypeMetadataDefinitionReferenceUsageFeatureMembershipReference\_Mapping
- stereotypeMetadataDefinitionReferenceUsageFeatureMembershipUsage :  
StereotypeMetadataDefinitionReferenceUsageFeatureMembershipUsage\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- `Feature::isEnd`  
`false`
- `Type::isSufficient`  
`false`
- `OperatorExpression::ownedRelationship`  
`Set{stereotypeMetadataDefinitionReferenceUsageFeatureMembershipReference.to, stereotypeMeta`
- `Feature::isUnique`  
`true`
- `Element::shortName`  
`null`
- `Type::isAbstract`  
`false`
- `Element::elementId`  
`Helper.createUUID()`
- `Feature::isOrdered`  
`false`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `Feature::isReadOnly`  
`false`
- `OperatorExpression::operator`  
`'as'`
- `Feature::direction`  
`null`
- `Element::name`  
`null`



- Feature::isDerived

false

- Feature::isComposite

false

#### C.2.5.9.3.36 StereotypeMetadataDefinitionReferenceUsageRedefinition\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToRedefinition\_Mapping

##### Mapping Source

Stereotype

##### Mapping Target

Redefinition

##### Owned Mappings

- stereotypeMetadataDefinitionReferenceUsage : StereotypeMetadataDefinitionReferenceUsage\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId

Set{ }

- Subsetting::ownedRelatedElement

Set{ }

- Subsetting::subsettingFeature

*abstract rule*

- Redefinition::redefinedFeature

SysMLv2::Feature.allInstances()->collect(dt | dt.owningRelationship)->select(r | r.ocIsKindOf)

- Element::name

null

- Subsetting::subsettingFeature

*abstract rule*

- Element::shortName

null

- Element::elementId

Helper.createUUID()

- Element::ownedRelationship

Set{}

- Redefinition::redefiningFeature

stereotypeMetadataDefinitionReferenceUsage.to

### **C.2.5.9.3.37 StereotypeMetadataDefinitionSubclassification\_Mapping**

#### **Description**

\*\*\* not specified yet \*\*\*

#### **General Mappings**

GenericToSubclassification\_Mapping

#### **Mapping Source**

Stereotype

#### **Mapping Target**

Subclassification

#### **Owned Mappings**

- stereotypeMetadataDefinition : StereotypeMetadataDefinition\_Mapping

#### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

#### **Mapping rules**

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

- Element::aliasId

Set{}

- Relationship::ownedRelatedElement

Set{}

- Specialization::specific  
*abstract rule*
- Subclassification::subclassifier

```
stereotypeMetadataDefinition.to
```

- Element::name

```
null
```

- Element::shortName

```
null
```

- Subclassification::superclassifier

```
SysMLv2::Metaclass.allInstances()->collect(dt | dt.owningRelationship)->select(r | r.oclIsKi
```

- Specialization::general  
*abstract rule*

- Element::elementId

```
Helper.createUUID()
```

- Element::ownedRelationship

```
Set{}
```

#### **C.2.5.9.3.38 StereotypeOccurrenceUsage\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToOccurrenceUsage\_Mapping

##### **Mapping Source**

Stereotype

##### **Mapping Target**

OccurrenceUsage

##### **Owned Mappings**

- stereotypeOccurrenceUsageFeatureTyping : StereotypeOccurrenceUsageFeatureTyping\_Mapping
- stereotypeOccurrenceUsageMultiplicityMembership :  
StereotypeOccurrenceUsageMultiplicityMembership\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- `Feature::isEnd`  
`false`
- `Type::isSufficient`  
`false`
- `OccurrenceUsage::ownedRelationship`  
`Set{stereotypeOccurenceUsageFeatureTyping.to, stereotypeOccurenceUsageMultiplicityMembership`
- `Feature::isUnique`  
`true`
- `Element::shortName`  
`null`
- `Type::isAbstract`  
`false`
- `Element::elementId`  
`Helper.createUUID()`
- `Feature::isOrdered`  
`false`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `Usage::isVariation`  
`false`
- `Feature::isReadOnly`  
`false`
- `Feature::direction`  
`null`
- `Element::name`  
`null`

- Feature::isDerived

false

- Feature::isComposite

false

#### **C.2.5.9.3.39 StereotypeOccurrenceDefinitionMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

ElementOwningMembership\_Mapping

##### **Mapping Source**

Stereotype

##### **Mapping Target**

OwningMembership

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Membership::memberShortName

null

- Element::shortName

null

- OwningMembership::ownedMemberElement

StereotypeOccurrenceDefinition\_Mapping.getMapped(from)

- Element::elementId

Helper.createUUID()

- Membership::visibility

```

    if (from.ocIsKindOf(UML::NamedElement)) then
        from.ocAsType(UML::NamedElement).visibility
    else
        KerML::VisibilityKind::public
    endif

```

- Element::aliasId

```
Set{}
```

- Relationship::target

```
OrderedSet{ElementMain_Mapping.getMapped(from)}
```

- Relationship::source

```
OrderedSet{ElementMain_Mapping.getMapped(from.owner)}
```

- Membership::memberName

```
null
```

- Membership::membershipOwningNamespace

```
Set{ElementMain_Mapping(from)} -- will not be used since corresponding att is derived, but r
```

- Element::name

```
null
```

- Relationship::ownedRelatedElement

```
self.target()
```

- Membership::memberElement

```
ElementMain_Mapping.getMapped(from)
```

- Element::ownedRelationship

```
Set{}
```

#### C.2.5.9.3.40 StereotypeOccurenceDefinition\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToOccurenceDefinition\_Mapping

##### Mapping Source

Stereotype

##### Mapping Target

OccurrenceDefinition

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Element::aliasId

```
Set{}
```

- Type::isSufficient

```
false
```

- OccurrenceDefinition::name

```
from.name
```

- Definition::isVariation

```
false
```

- OccurrenceDefinition::ownedRelationship

```
let baseProperties : Sequence(UML::Element) = src.ownedElement->select(e | e.name.indexOf('ba') > -1)
let properties: Sequence(UML::Element) = src.ownedElement->select(e | e.ocliIsKindOf(UML::Property))
let elements: Set(UML::Element) = (src.ownedElement-properties) - baseProperties in
elements->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(properties->collect(e | PropertyMembership_Mapping.getMapped(e)))
```

- Element::shortName

```
null
```

- Type::isAbstract

```
false
```

- Element::elementId

```
Helper.createUUID()
```

### C.2.5.9.3.41 StereotypeOccurenceUsageFeatureTyping\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

## General Mappings

GenericToFeatureTyping\_Mapping

## Mapping Source

Stereotype

## Mapping Target

FeatureTyping

## Owned Mappings

- stereotypeOccurrenceUsage : StereotypeOccurrenceUsage\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Element::aliasId  
`Set{}`
- Relationship::ownedRelatedElement  
`Set{}`
- Specialization::specific  
*abstract rule*
- FeatureTyping::type  
`StereotypeOccurrenceDefinition_Mapping.getMapped(from)`
- FeatureTyping::typedFeature  
`stereotypeOccurrenceUsage.to`
- Element::name  
`null`
- Element::shortName  
`null`
- Specialization::general  
*abstract rule*
- Element::elementId  
`Helper.createUUID()`



- Element::ownedRelationship

```
Set{}
```

#### C.2.5.9.3.42 StereotypeOccurenceUsageMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToMembership\_Mapping

##### Mapping Source

Stereotype

##### Mapping Target

Membership

##### Owned Mappings

- stereotypeOccurenceUsage : StereotypeOccurenceUsage\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId

```
Set{}
```

- Relationship::ownedRelatedElement

```
Set{}
```

- Membership::memberName

```
from.name.substring(1,1).toLowerCase() + from.name.substring(2,from.name.size()) + 's'
```

- Membership::memberElement

```
self.ownedMemberElement()
```

- Relationship::source

```
Set{}
```

- Element::name

```
null
```

- Membership::ownedMemberElement

stereotypeOccurrenceUsage.to

- Element::shortName

null

- Element::elementId

Helper.createUUID()

- Relationship::target

Set{}

- Element::ownedRelationship

Set{}

#### **C.2.5.9.3.43 StereotypeOccurrenceUsageMultiplicityMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToMembership\_Mapping

##### **Mapping Source**

Stereotype

##### **Mapping Target**

Membership

##### **Owned Mappings**

- stereotypeOccurrenceUsageMultiplicityRange : StereotypeOccurrenceUsageMultiplicityRange\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Element::aliasId

Set{}

- Relationship::ownedRelatedElement

Set{}

- Relationship::source  
`Set{}`
- Membership::ownedMemberElement  
`stereotypeOccurenceUsageMultiplicityRange.to`
- Element::name  
`null`
- Membership::memberElement  
`self.ownedMemberElement()`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Relationship::target  
`Set{}`
- Element::ownedRelationship  
`Set{}`

#### **C.2.5.9.3.44 StereotypeOccurenceUsageMultiplicityRange\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeature\_Mapping

##### **Mapping Source**

Stereotype

##### **Mapping Target**

MultiplicityRange

##### **Owned Mappings**

- stereotypeOccurenceUsageMultiplicityRangeMembership :  
StereotypeOccurenceUsageMultiplicityRangeMembership\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  
`Set{}`
- Type::isSufficient  
`false`
- Element::name  
`null`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- MultiplicityRange::ownedRelationship  
`Set{stereotypeOccurenceUsageMultiplicityRangeMembership.to}`

#### C.2.5.9.3.45 StereotypeOccurenceUsageMultiplicityRangeInfinity\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToExpression\_Mapping

##### Mapping Source

Stereotype

##### Mapping Target

LiteralInfinity

##### Owned Mappings

- stereotypeOccurenceUsageMultiplicityRangeInfinityReturnParameterMembership :  
StereotypeOccurenceUsageMultiplicityRangeInfinityReturnParameterMembership\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Feature::isEnd  
`false`
- LiteralInfinity::ownedRelationship  
`Set{stereotypeOccurenceUsageMultiplicityRangeInfinityReturnParameterMembership.to}`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Feature::isReadOnly  
`false`
- Feature::direction  
`null`
- Element::name

null

- Feature::isDerived

false

- Feature::isComposite

false

#### **C.2.5.9.3.46 StereotypeOccurrenceUsageMultiplicityRangeInfinityReturnParameter\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeature\_Mapping

##### **Mapping Source**

Stereotype

##### **Mapping Target**

Feature

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- Element::aliasId

Set{ }

- Type::isSufficient

false

- Feature::direction

SysMLv2::FeatureDirectionKind::out

- Element::name

null

- Element::shortName

null

- Type::isAbstract

false

- Element::elementId

Helper.createUUID()

- Element::ownedRelationship

Set{}

#### **C.2.5.9.3.47 StereotypeOccurenceUsageMultiplicityRangeInfinityReturnParameterMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToReturnParameterMembership\_Mapping

##### **Mapping Source**

Stereotype

##### **Mapping Target**

ReturnParameterMembership

##### **Owned Mappings**

- stereotypeOccurenceUsageMultiplicityRangeInfinityReturnParameter :  
StereotypeOccurenceUsageMultiplicityRangeInfinityReturnParameter\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName

null

- ReturnParameterMembership::ownedMemberParameter

stereotypeOccurenceUsageMultiplicityRangeInfinityReturnParameter.to

- ReturnParameterMembership::memberParameter

- ```
self.ownedMemberParameter()
```
- Element::shortName
  - ```
null
```
- Element::elementId
  - ```
Helper.createUUID()
```
- Element::aliasId
  - ```
Set{}
```
- FeatureMembership::owningType
  - ```
abstract rule
```
- ReturnParameterMembership::ownedRelatedElement
  - ```
let member: KerML::Element = self.ownedMemberParameter() in
if member.oclIsUndefined() then
  Set{}
else
  Set{self.ownedMemberParameter()}
endif
```
- Membership::memberName
  - ```
null
```
- TypeFeaturing::featureOfType
  - ```
abstract rule
```
- TypeFeaturing::featuringType
  - ```
abstract rule
```
- Membership::visibility
  - ```
KerML::VisibilityKind::public
```
- Element::name
  - ```
null
```
- Element::ownedRelationship
  - ```
Set{}
```

#### C.2.5.9.3.48 StereotypeOccurrenceUsageMultiplicityRangeMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToMembership\_Mapping

##### Mapping Source

Stereotype



## Mapping Target

Membership

## Owned Mappings

- stereotypeOccurenceUsageMultiplicityRangeInfinity :  
StereotypeOccurenceUsageMultiplicityRangeInfinity\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Element::aliasId  
`Set {}`
- Relationship::ownedRelatedElement  
`Set {}`
- Membership::memberElement  
`self.ownedMemberElement()`
- Relationship::source  
`Set {}`
- Element::name  
`null`
- Membership::ownedMemberElement  
`stereotypeOccurenceUsageMultiplicityRangeInfinity.to`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Relationship::target  
`Set {}`
- Element::ownedRelationship  
`Set {}`

## C.2.5.10 SimpleClassifiers

### C.2.5.10.1 Overview

This chapter specifies the mapping of the metaclasses defined in the UML specification in the SimpleClassifiers chapter, which are part of the UML4SysML subset.

**Table 25. List of all Overview Mapping Specifications**

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
BehavioredClassifier	PerformActionUsage Classifier FeatureTyping FeatureMembership	BehavioredClassifierToPerformActionUsage_Mapping BehavioredClassifier_Mapping BehavioredClassifierToFeatureTyping_Mapping ClassifierBehaviorMembership_Mapping	
DataType	AttributeDefinition	DataType_Mapping	
Enumeration	EnumerationDefinition	Enumeration_Mapping	
EnumerationLiteral	EnumerationUsage VariantMembership	EnumerationLiteral_Mapping EnumerationVariantMembership_Mapping	EnumerationLiteral.classifier->select( c   coclIsTypeOf(UML::Association))->size() = 0
Interface	PortConjugation OwningMembership ConjugatedPortDefinition PortDefinition	InterfacePortConjugation_Mapping InterfaceConjugatedPortDefinitionMembership_Mapping InterfaceConjugatedPortDefinition_Mapping Interface_Mapping	
InterfaceRealization	Subclassification	InterfaceRealization_Mapping	
PrimitiveType	AttributeDefinition	PrimitiveType_Mapping	
Reception	FeatureTyping AttributeUsage	ReceptionToFeatureTyping_Mapping Reception_Mapping	
Signal	AttributeDefinition	Signal_Mapping	

### C.2.5.10.2 Mapping Specifications

#### C.2.5.10.2.1 Attribute\_Mapping

##### Description

An UML::SimpleClassifiers::Property is mapped to a SysMLv2::Systems::Attributes::AttributeUsage.

##### General Mappings

PropertyCommon\_Mapping  
NamedElementMain\_Mapping

##### Mapping Source

Property

##### Mapping Target

AttributeUsage

## Owned Mappings

(none)

## Applicable filters

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

```
if src.ocIsTypeOf(UML::Property) and (src.ocAsType(UML::Property).redefinedElement->size() = 0) then
    let p: UML::Property = src.ocAsType(UML::Property) in
        p.type.ocIsKindOf(UML::DataType)
else
    false
endif
```

## Mapping rules

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

- Feature::isEnd

false

- Feature::isOrdered

from.isOrdered

- Type::isSufficient

false

- Feature::isAbstract

false

- Element::shortName

null

- Feature::ownedRelationship

```
let typing: KerML::FeatureTyping = StructuralFeatureToFeatureTyping_Mapping.getMapped(from)
if typing.ocIsUndefined() then
    Set{MultiplicityMembership_Mapping.getMapped(from)}
else
    Set{MultiplicityMembership_Mapping.getMapped(from), typing}
endif
```

- Element::aliasId

Set{}

- Feature::isPortion

false

- Feature::direction

null

- Element::elementId  
Helper.getID(from)
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false
- Feature::isUnique  
from.isUnique
- Feature::isReadOnly  
*abstract rule*

#### **C.2.5.10.2.2 AttributeRedefined\_Mapping**

##### **Description**

An UML::SimpleClassifiers::Property is mapped to a SysMLv2::Systems::Attributes::AttributeUsage.

##### **General Mappings**

PropertyCommon\_Mapping

##### **Mapping Source**

Property

##### **Mapping Target**

ReferenceUsage

##### **Owned Mappings**

- attributeRedefinedFeatureTyping : AttributeRedefinedFeatureTyping\_Mapping
- attributeRedefinedRedefinition : AttributeRedefinedRedefinition\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Feature::isEnd

false

- Feature::isOrdered

from.isOrdered

- Type::isSufficient

false

- Feature::isAbstract

false

- ReferenceUsage::ownedRelationship

```
let typing: KerML::FeatureTyping = attributeRedefinedFeatureTyping.to in
let subsetting: Set(KerML::Subsetting) = from.subsettedProperty->collect(p | PropertySubsetting)
let subsettingMultiplicityTyping: Set(KerML::Relationship) = subsetting->union(Set{attributeF
    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
```

- Element::shortName

null

- Feature::ownedRelationship

```
let typing: KerML::FeatureTyping = StructuralFeatureToFeatureTyping_Mapping.getMapped(from)
if typing.oclIsUndefined() then
    Set{MultiplicityMembership_Mapping.getMapped(from)}
else
    Set{MultiplicityMembership_Mapping.getMapped(from), typing}
endif
```

- Element::elementId

Helper.createUUID()

- Element::aliasId

Set{}

- Feature::isPortion

false

- Feature::direction

null

- Element::name

null

- Feature::isDerived

    false

- Feature::isComposite

    false

- Feature::isUnique

    from.isUnique

- Feature::isReadOnly

*abstract rule*

#### **C.2.5.10.2.3 AttributeRedefinedRedefinition\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToRedefinition\_Mapping

##### **Mapping Source**

Property

##### **Mapping Target**

Redefinition

##### **Owned Mappings**

- attributeRedefined : AttributeRedefined\_Mapping

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- Element::aliasId

    Set{ }

- Redefinition::redefiningFeature

    attributeRedefined.to

- Subsetting::ownedRelatedElement

```

Set {}

• Subsetting::subsettingFeature
  abstract rule
• Element::name

  null

• Subsetting::subsettingFeature
  abstract rule
• Redefinition::redefinedFeature

  from.redefinedProperty.get(0)

• Element::shortName

  null

• Element::elementId

  Helper.createUUID()

• Element::ownedRelationship

  Set {}

```

#### C.2.5.10.2.4 AttributeRedefinedMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

ElementFeatureMembership\_Mapping

##### Mapping Source

NamedElement

##### Mapping Target

FeatureMembership

##### Owned Mappings

- attributeRedefined : AttributeRedefined\_Mapping

##### Applicable filters

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

```
from.ocIsKindOf(UML::Property) and (from.ocAsType(UML::Property).redefinedElement->size() > 0)
```

##### Mapping rules

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

- FeatureMembership::ownedMemberFeature  
`attributeRedefined.to`
- FeatureMembership::ownedRelatedElement  
`Set{self.ownedMemberFeature() }`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{ }`
- FeatureMembership::ownedMemberFeature  
*abstract rule*
- FeatureMembership::owningType  
*abstract rule*
- Membership::memberName  
`null`
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
`KerML::VisibilityKind::public`
- Element::name  
`null`
- Element::ownedRelationship  
`Set{ }`

#### **C.2.5.10.2.5 AttributeRedefinedFeatureTyping\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*



## General Mappings

StructuralFeatureToFeatureTyping\_Mapping

## Mapping Source

StructuralFeature

## Mapping Target

FeatureTyping

## Owned Mappings

- attributeRedefined : AttributeRedefined\_Mapping

## Applicable filters

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

```
not src.type.ocIsUndefined()  
and not (src.type.ocIsKindOf(UML::Enumeration) and Helper.getSysMLv2EnumerationDefinition(src.type))
```

## Mapping rules

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

- Element::aliasId  

```
Set{}
```
- Relationship::ownedRelatedElement  

```
Set{}
```
- FeatureTyping::typedFeature  

```
attributeRedefined.to
```
- FeatureTyping::typedFeature  
*abstract rule*
- Element::name  

```
null
```
- Element::shortName  

```
null
```
- Element::elementId  

```
Helper.createUUID()
```
- Element::ownedRelationship  

```
Set{}
```
- FeatureTyping::type

```

let sysmlv1PrimitiveType : SysMLv2::DataType = if from.type.ocIsKindOf(UML::PrimitiveType) t
let sysmlv1EnumerationType : SysMLv2::Enumeration = if from.type.ocIsKindOf(UML::Enumeration
if not sysmlv1PrimitiveType.ocIsUndefined() then sysmlv1PrimitiveType else
if not sysmlv1EnumerationType.ocIsUndefined() then sysmlv1EnumerationType else
Classifier_Mapping.getMapped(from.type) endif endif

```

#### C.2.5.10.2.6 BehavioredClassifier\_Mapping

##### Description

The abstract mapping class BehavioredClassifier\_Mapping maps the abstract metaclass UML::SimpleClassifiers::BehavioredClassifiers to a SysMLv2::Core::Classifiers::Classifier. The mapping class is used by concrete mapping classes, for example, Block\_Mapping.

##### General Mappings

Classifier\_Mapping

##### Mapping Source

BehavioredClassifier

##### Mapping Target

Classifier

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId  
Set{ }
- Type::isSufficient  
false
- Namespace::ownedImport  
Set{ }
- Classifier::ownedRelationship

```

let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | (e.ocIsKindOf(UML::Prope
let redefinedAttributes: Set(UML::Element) = from.ownedElement->select(e | from.ocIsKindOf(U

```

```

let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.ocIsKindOf
let constraints : Set(UML::Constraint) = UML::Constraint.allInstances()->select( c | c.constr
let toElementOMS: Set(UML::Element) = ((from.ownedElement - toElementFMS) - redefinedAttribu
let relationships: Sequence(KerML::Relationship) =
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(constraints->collect(e | ConstrainedElementFeatureMembership_Mapping.getMapped(e)))
->union(redefinedAttributes->collect(e | AttributeRedefinedMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e))) in
if from.classifierBehavior.ocIsUndefined() then relationships else relationships->append(Cla

```

- Element::elementId

```
Helper.getID(from)
```

- Element::name

```
from.name
```

- Element::shortName

```
null
```

- Type::isAbstract

```
false
```

- Namespace::ownedRelationship

```
from.ownedElement->collect(e | ElementOwningMembership_Mapping.getMapped(e))
```

#### C.2.5.10.2.7 ClassifierBehaviorMembership\_Mapping

##### Description

The ClassifierBehaviorMembership\_Mapping class creates a membership relationship for a PerformActionUsage element to call the transformed SysML v1 classifier behavior.

##### General Mappings

GenericToFeatureMembership\_Mapping

##### Mapping Source

BehavioredClassifier

##### Mapping Target

FeatureMembership

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{ }
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
  
null
- OwningMembership::ownedRelatedElement  
  
Set{ self.ownedMemberElement() }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- FeatureMembership::ownedMemberFeature  
  
BehavioredClassifierToPerformActionUsage\_Mapping.getMapped(from)
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- Element::ownedRelationship  
  
Set{ }

### C.2.5.10.2.8 BehavioredClassifierToFeatureTyping\_Mapping

#### Description

The BehavioredClassifierToFeatureTyping\_Mapping creates the relationship from the PerformActionUsage element to its type which is the transformed SysML v1 classifier behavior.

## General Mappings

GenericToFeatureTyping\_Mapping

## Mapping Source

BehavioredClassifier

## Mapping Target

FeatureTyping

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Element::aliasId  
`Set {}`
- Relationship::ownedRelatedElement  
`Set {}`
- Specialization::specific  
*abstract rule*
- Element::name  
`null`
- Element::shortName  
`null`
- Specialization::general  
*abstract rule*
- Element::elementId  
`Helper.createUUID()`
- FeatureTyping::type  
`from`
- Element::ownedRelationship  
`Set {}`

#### C.2.5.10.2.9 BehavioredClassifierToPerformActionUsage\_Mapping

##### Description

The BehavioredClassifierToPerformActionUsage\_Mapping class creates a PerformActionUsage element to call the transformed SysML v1 classifier behavior.

##### General Mappings

GenericToFeature\_Mapping

##### Mapping Source

BehavioredClassifier

##### Mapping Target

PerformActionUsage

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId  
`Set{ }`
- Type::isSufficient  
`false`
- PerformActionUsage::isComposite  
`true`
- PerformActionUsage::ownedRelationship  
`Set{ BehavioredClassifierToFeatureTyping_Mapping.getMapped(from) }`
- PerformActionUsage::name  
`'classifierBehavior'`
- Element::shortName  
`null`
- Type::isAbstract

false

- Element::elementId

Helper.createUUID()

#### **C.2.5.10.2.10 DataType\_Mapping**

##### **Description**

A UML::SimpleClassifiers::DataType is mapped to a SysMLv2::Systems::Attributes::AttributeDefinition. The mapping also cover the transformation of UML4SysML::PrimitiveType elements.

##### **General Mappings**

Classifier\_Mapping

##### **Mapping Source**

DataType

##### **Mapping Target**

AttributeDefinition

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Element::aliasId

Set{ }

- Type::isSufficient

false

- Namespace::ownedImport

Set{ }

- Element::elementId

Helper.getID(from)

- Element::name

from.name

- Element::shortName

null

- Type::isAbstract

false

- Namespace::ownedRelationship

from.ownedElement->collect(e | ElementOwningMembership\_Mapping.getMapped(e))

#### **C.2.5.10.2.11 Enumeration\_Mapping**

##### **Description**

A UML4SysML::Enumeration is mapped to a SysMLv2::EnumerationDefinition.

##### **General Mappings**

DataType\_Mapping

##### **Mapping Source**

Enumeration

##### **Mapping Target**

EnumerationDefinition

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- Element::aliasId

Set{}

- Classifier::isAbstract

from.isAbstract

- Type::isSufficient

false

- Namespace::ownedImport

Set{}



- EnumerationDefinition::isVariation

true

- Element::elementId

Helper.getID(from)

- Element::name

from.name

- Element::shortName

null

- EnumerationDefinition::ownedRelationship

```
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.ocIsKindOf(UML::Generalization))
let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::Property))
let literals: Set(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::EnumerationLiteral))
let toElementOMS: Set(UML::Element) = (((from.ownedElement - toElementFMS) - generalizations) - literals)
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))
->union(literals->collect(e | EnumerationVariantMembership_Mapping.getMapped(e)))
```

#### C.2.5.10.2.12 EnumerationLiteral\_Mapping

##### Description

A UML4SysML::EnumerationLiteral is mapped to a SysMLv2::EnumerationUsage.

##### General Mappings

GenericToFeature\_Mapping  
InstanceSpecification\_Mapping

##### Mapping Source

EnumerationLiteral

##### Mapping Target

EnumerationUsage

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- `Feature::isEnd`  
`false`
- `Element::ownedRelationship`  
`ElementOwnership_Mapping.getMappedColl(from.ownedElement)`
- `Type::isSufficient`  
`false`
- `Feature::isUnique`  
`true`
- `Element::name`  
`from.name`
- `Element::shortName`  
`null`
- `Type::isAbstract`  
`false`
- `Feature::isOrdered`  
`false`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `Usage::isVariation`  
`false`
- `Feature::isReadOnly`  
`false`
- `Feature::direction`  
`null`
- `Element::elementId`  
`Helper.getID(from)`
- `Feature::isDerived`

false

- Feature::isComposite

false

#### **C.2.5.10.2.13 EnumerationVariantMembership\_Mapping**

##### **Description**

The EnumerationVariantMembership\_Mapping class creates the variant membership relationship between the enumeration definition and a enumeration usage.

##### **General Mappings**

GenericToMembership\_Mapping

##### **Mapping Source**

EnumerationLiteral

##### **Mapping Target**

VariantMembership

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Element::aliasId

Set{ }

- Relationship::ownedRelatedElement

Set{ }

- Relationship::source

Set{ }

- Element::name

null

- Element::shortName

null

- VariantMembership::ownedMemberElement  
EnumerationLiteral\_Mapping.getMapped(from)
- Element::elementId  
Helper.createUUID()
- Relationship::target  
Set{ }
- Element::ownedRelationship  
Set{ }

#### **C.2.5.10.2.14 Interface\_Mapping**

##### **Description**

A UML4SysML::Interface is mapped to a SysMLv2::PortDefinition. The mapping also includes the generation of an appropriate ConjugatedPortDefinition. That mappings is performed by the mapping classes InterfaceConjugatedPortDefinitionMembership\_Mapping, InterfacePortConjugation\_Mapping, and InterfaceConjugatedPortDefinition\_Mapping.

##### **General Mappings**

GenericToPortDefinition\_Mapping  
Classifier\_Mapping

##### **Mapping Source**

Interface

##### **Mapping Target**

PortDefinition

##### **Owned Mappings**

- conjugatedPortDefinitionMembership : InterfaceConjugatedPortDefinitionMembership\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Element::aliasId  
Set{ }
- Type::isSufficient

- Namespace::ownedImport

- Namespace::ownedImport

Set { }

- Definition::isVariation

false

- `Element::elementId`

```
Helper.getID(from)
```

- `Element::name`

```
from.name
```

- PortDefinition::ownedRelationship

```

let properties: Set(UML::Element) = from.ownedElement->select(e | e.oclcIsKindOf(UML::Property))
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.oclcIsKindOf(UML::Generalization))
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)

```

- `Element::shortName`

null

- `Type::isAbstract`

false

- Namespace::ownedRelationship

```
from.ownedElement->collect(e | ElementOwningMembership Mapping.getMapped(e))
```

#### C.2.5.10.2.15 InterfaceConjugatedPortDefinition\_Mapping

### Description

As part of the mapping from a `UML4SysML::Interface` to a `SysMLv2::PortDefinition`, this mapping class is used to create the appropriate `ConjugatedPortDefinition`.

## General Mappings

## GenericToPortDefinition Mapping

## Mapping Source

## Interface

## Mapping Target

### ConjugatedPortDefinition

## Owned Mappings

- portConjugation : InterfacePortConjugation\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Element::aliasId  
`Set{}`
- Type::isSufficient  
`false`
- ConjugatedPortDefinition::ownedRelationship  
`Set{portConjugation}`
- Definition::isVariation  
`false`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- ConjugatedPortDefinition::name  
`'~'+from.name`
- Element::elementId  
`Helper.createUUID()`

### C.2.5.10.2.16 InterfaceConjugatedPortDefinitionMembership\_Mapping

#### Description

As part of the mapping from a UML4SysML::Interface to a SysMLv2::PortDefinition, this mapping class is used to create the membership relationship for the ConjugatedPortDefinition.

#### General Mappings

GenericToOwningMembership\_Mapping

#### Mapping Source

Interface

## Mapping Target

OwningMembership

## Owned Mappings

- conjugatedPortDefinitionMapping : InterfaceConjugatedPortDefinition\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Relationship::ownedRelatedElement  
`Set{}`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- OwningMembership::ownedMemberElement  
`conjugatedPortDefinitionMapping.to`
- Membership::memberElement  
*abstract rule*
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- OwningMembership::ownedRelationship  
`Set{portConjugation}`
- Element::aliasId  
`Set{}`
- Membership::memberName  
`null`
- Membership::visibility  
`KerML::VisibilityKind::public`

- Element::name

null

#### **C.2.5.10.2.17 InterfacePortConjugation\_Mapping**

##### **Description**

As part of the mapping from a UML4SysML::Interface to a SysMLv2::PortDefinition, this mapping class is used to create the appropriate PortConjugation relationship.

##### **General Mappings**

GenericToRelationship\_Mapping

##### **Mapping Source**

Interface

##### **Mapping Target**

PortConjugation

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- Element::aliasId

Set{ }

- PortConjugation::originalPortDefinition

from

- PortConjugation::conjugatedType

SysMLv2::ConjugatedPortDefinition.allInstances()->collect(cpd | cpd.owningRelationship)->sel

- Element::name

null

- Element::shortName

null

- Element::elementId



```
Helper.createUUID()
```

- Element::ownedRelationship

```
Set{}
```

#### **C.2.5.10.2.18 InterfaceRealization\_Mapping**

##### **Description**

A UML4SysML::InterfaceRealization is mapped to a SysMLv2::Superclassing.

##### **General Mappings**

GenericToSpecialization\_Mapping

##### **Mapping Source**

InterfaceRealization

##### **Mapping Target**

Subclassification

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- Element::aliasId

```
Set{}
```

- Relationship::ownedRelatedElement

```
Set{}
```

- Relationship::source

```
Set{}
```

- Element::name

```
null
```

- Subclassification::subclassifier

```
Classifier_Mapping.getMapped(from.specific)
```

- Element::shortName

null

- Element::elementId

    Helper.createUUID()

- Relationship::target

    Set{}

- Element::ownedRelationship

    Set{}

- Subclassification::superclassifier

    Classifier\_Mapping.getMapped(from.general)

#### **C.2.5.10.2.19 PrimitiveType\_Mapping**

##### **Description**

The PrimitiveType\_Mapping class maps a UML4SysML::PrimitiveType to a SysML v2 AttributeDefinition.

##### **General Mappings**

DataType\_Mapping

##### **Mapping Source**

PrimitiveType

##### **Mapping Target**

AttributeDefinition

##### **Owned Mappings**

(none)

##### **Applicable filters**

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

(none)

##### **Mapping rules**

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

- Element::aliasId

    Set{}

- Classifier::isAbstract

    from.isAbstract

- Type::isSufficient

false

- Namespace::ownedImport

Set{}

- Element::elementId

Helper.getID(from)

- Classifier::ownedRelationship

```
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.ocIsKindOf(UML::Generalization))
let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::FeatureMembership))
let toElementOMS: Set(UML::Element) = (from.ownedElement - toElementFMS) - generalizations
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))
```

- Element::name

from.name

- Element::shortName

null

#### C.2.5.10.2.20 PrimitiveTypeString\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

PrimitiveType\_Mapping

##### Mapping Source

PrimitiveType

##### Mapping Target

LiteralString

##### Owned Mappings

(none)

##### Applicable filters

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

```
(from.name = 'String')
```

##### Mapping rules

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

- Element::aliasId

```
Set{}
```

- Classifier::isAbstract

```
from.isAbstract
```

- Type::isSufficient

```
false
```

- Namespace::ownedImport

```
Set{}
```

- LiteralString::value

```
from
```

- Element::elementId

```
Helper.getID(from)
```

- Classifier::ownedRelationship

```
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.ocIsKindOf(UML::Generalization))
let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::FeatureMembership))
let toElementOMS: Set(UML::Element) = (from.ownedElement - toElementFMS) - generalizations
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))
```

- Element::name

```
from.name
```

- Element::shortName

```
null
```

#### C.2.5.10.2.21 Reception\_Mapping

##### Description

A UML4SysML::Reception is mapped to a SysMLv2::AttributeUsage with feature direction "in".

##### General Mappings

BehavioralFeature\_Mapping

##### Mapping Source

Reception

##### Mapping Target

AttributeUsage

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- AttributeUsage::direction  
`SysMLv2::FeatureDirectionKind::in`
- Feature::isUnique  
`true`
- Element::name  
`from.name`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- AttributeUsage::ownedRelationship  
`Set{ReceptionToFeatureTyping_Mapping.getMapped(from)}`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`

- Usage::isVariation  
false
- Namespace::ownedImport  
Set{}
- Feature::isReadOnly  
false
- Element::elementId  
Helper.getID(from)
- Feature::isDerived  
false
- Feature::isComposite  
false

#### **C.2.5.10.2.22 ReceptionToFeatureTyping\_Mapping**

##### **Description**

A UML4SysML::Reception is mapped to SysMLv2::AttributeUsage. The ReceptionToFeatureTyping\_Mapping class creates the type of the AttributeUsage which is the Signal of the Reception.

##### **General Mappings**

TypedElementToFeatureTyping\_Mapping

##### **Mapping Source**

Reception

##### **Mapping Target**

FeatureTyping

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Element::aliasId

- Set{}
- Relationship::ownedRelatedElement
  - Set{}
- FeatureTyping::typedFeature
  - Reception\_Mapping.getMapped(from)
- Element::name
  - null
- FeatureTyping::type
  - Classifier\_Mapping.getMapped(from.signal)
- Element::shortName
  - null
- Element::elementId
  - Helper.createUUID()
- FeatureTyping::type
  - abstract rule*
- FeatureTyping::typedFeature
  - abstract rule*
- Element::ownedRelationship
  - Set{}

#### C.2.5.10.2.23 Signal\_Mapping

##### Description

A UML4SysML::Signal is mapped to a SysMLv2::AttributeDefinition.

##### General Mappings

DataType\_Mapping

##### Mapping Source

Signal

##### Mapping Target

AttributeDefinition

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- `Element::aliasId`

`Set{}`

- `Classifier::isAbstract`

`from.isAbstract`

- `Type::isSufficient`

`false`

- `Namespace::ownedImport`

`Set{}`

- `Element::elementId`

`Helper.getID(from)`

- `Classifier::ownedRelationship`

```
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.ocIsKindOf(UML::Generalization))
let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::FeatureMembership))
let toElementOMS: Set(UML::Element) = (from.ownedElement - toElementFMS) - generalizations
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))
```

- `Element::name`

`from.name`

- `Element::shortName`

`null`

### C.2.5.11 StructuredClassifiers



### C.2.5.11.1 Overview

**Table 26. List of all Overview Mapping Specifications**

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
Association	FeatureMembership FeatureMembership Redefinition MetadataFeature FeatureTyping Association Feature FeatureValue Annotation	AssociationToMetadataMembership_Mapping AssociationToFeatureMembership_Mapping AssociationToRedefinition_Mapping AssociationToAnnotatingFeature_Mapping AssociationToFeatureTyping_Mapping AssociationCommon_Mapping AssociationToMetadataFeature_Mapping AssociationToMetadataFeatureValue_Mapping AssociationToAnnotation_Mapping	Association.memberEnd->select( m   m.type.oclIsKindOf(UML::UseCase))->isEmpty()
AssociationClass	ConnectionDefinition	AssociationClass_Mapping	not Helper.hasStereotypeApplied(AssociationClass, 'SysML::Blocks::Block')
Class	ViewpointDefinition SubjectMembership FeatureTyping MetadataUsage FeatureValue FeatureTyping FeatureMembership ReferenceUsage OwningMembership RequirementUsage ReferenceUsage OwningMembership OccurrenceDefinition MetadataUsage Redefinition	Viewpoint_Mapping ViewpointSubjectMembership_Mapping EncapsulatedBlockMetadataFeatureTyping_Mapping EncapsulatedBlockMetadata_Mapping EncapsulatedBlockMetadataFeatureValue_Mapping ViewpointPurposeMetadataFeatureTyping_Mapping EncapsulatedBlockMetadataFeatureMembership_Mapping ViewpointSubject_Mapping ViewpointPurposeMetadataMembership_Mapping Requirement_Mapping EncapsulatedBlockMetadataReferenceUsage_Mapping EncapsulatedBlockMetadataMembership_Mapping Class_Mapping ViewpointPurposeMetadata_Mapping EncapsulatedBlockMetadataRedefinition_Mapping	Helper.hasStereotypeApplied(Class, 'SysML::ModelElements::Viewpoint')          Helper.hasStereotypeApplied(Class, 'SysML::Requirements::Requirement')          Helper.hasStereotypeApplied(Class, 'SysML::Requirements::Requirement')    Class.oclIsTypeOf(UML::AssociationClass)

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
ConnectableElement	FeatureMembership FeatureTyping FeatureTyping FeatureValue ParameterMembership FeatureReferenceExpression Membership Feature ReferenceUsage FeatureTyping FeatureTyping Element Feature FeatureTyping Relationship Expression ReferenceUsage OwningMembership LiteralInteger ReturnParameterMembership ParameterMembership FeatureMembership FeatureTyping LiteralInteger MultiplicityRange ReferenceUsage LiteralInteger Element Membership ReturnParameterMembership ReturnParameterMembership	CommonReferenceUsageInFeatureMembership_Mapping CommonReferenceUsageInFeatureTyping_Mapping TypedElementToFeatureTyping_Mapping EqualOperatorExpressionFeatureValue_Mapping EqualOperatorExpressionOperand_Mapping CommonFeatureReferenceExpression_Mapping CommonMembership_Mapping EqualOperatorExpressionFeature_Mapping CommonReferenceUsageInUntyped_Mapping CommonParameterReferenceUsageInFeatureTyping_Mapping Mapping CommonReturnParameterFeatureUntyped_Mapping CommonReturnParameterFeatureTyping_Mapping ElementOwnership_Mapping CommonValueSpecification_Mapping CommonParameterReferenceUsageInUntyped_Mapping DefaultMultiplicityMembership_Mapping CommonReturnParameterFeatureMembership_Mapping CommonParameterReferenceUsageInMembership_Mapping DefaultMultiplicityBoundOwnership_Mapping CommonReturnParameterReferenceUsageFeatureTyping_Mapping DefaultUpperBound_Mapping DefaultMultiplicityElement_Mapping CommonReturnParameterReferenceUsageUntyped_Mapping DefaultLowerBound_Mapping ElementMain_Mapping ElementMembership_Mapping CommonReturnParameterReferenceUsageMembership_Mapping EmptyReturnParameterFeatureMembership_Mapping	not src.type.ocllsUndefined() and not src.type.ocllsKindOf(UML::Enumeration) and Helper.getSysMLv2EnumerationDefinition(src)
Connector	ConnectionUsage OwningMembership	Connector_Mapping ConnectorMultiplicityMembership_Mapping	
ConnectorEnd	Feature Subsetting EndFeatureMembership EndFeatureMembership	ConnectorEndToFeatureCommon_Mapping ConnectionEndToSubsetting_Mapping ConnectorEndToSubsettedFeatureMembership_Mapping ConnectorEndToMembership_Mapping	
EncapsulatedClassifier	ObjectiveMembership FeatureTyping SubjectMembership PartUsage RequirementUsage ReferenceUsage Classifier StakeholderMembership	CaseObjectiveMembership_Mapping CaseSubjectFeatureTyping_Mapping CaseSubjectMembership_Mapping StakeholderPartUsage_Mapping CaseObjectiveRequirementUsage_Mapping CaseEmptySubjectReferenceUsage_Mapping Classifier_Mapping StakeholderMembership_Mapping	

SysML v1 Concept	SysML v2 Concept	Mapping Class	Filter
Port	PortUsage	Port_Mapping	<pre> result = not   Helper.hasStereotypeApplied(Port.owner,     'SysML::ConstraintBlocks::ConstraintBlock') or   ((Port.type.ocIsUndefined() or   Helper.hasStereotypeApplied(Port.type,     'SysML::Ports&amp;Flows::InterfaceBlock')) and not   (Helper.hasStereotypeApplied(Port,     'SysML::Ports&amp;Flows::FullPort') or   (Port.type.ocIsKindOf(UML::Classifier) and not   Helper.hasStereotypeApplied(Port.type,     'SysML::Ports&amp;Flows::InterfaceBlock')))) </pre>
StructuredClassifier	ObjectiveMembership FeatureTyping SubjectMembership PartUsage RequirementUsage ReferenceUsage Classifier StakeholderMembership	CaseObjectiveMembership_Mapping CaseSubjectFeatureTyping_Mapping CaseSubjectMembership_Mapping StakeholderPartUsage_Mapping CaseObjectiveRequirementUsage_Mapping CaseEmptySubjectReferenceUsage_Mapping Classifier_Mapping StakeholderMembership_Mapping	

## C.2.5.11.2 Mapping Specifications

### C.2.5.11.2.1 AssociationCommon\_Mapping

#### Description

A UML4SysML::Association is mapped to a SysMLv2::ConnectionDefinition. The UML4SysML::Association::isDerived property is not supported in SysML v2. To preserve the information, it is stored in a metadata annotation.

#### General Mappings

Classifier\_Mapping  
Relationship\_Mapping

#### Mapping Source

Association

#### Mapping Target

Association

#### Owned Mappings

(none)

#### Applicable filters

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

```
from.memberEnd->select( m | m.type.ocIsKindOf(UML::UseCase)) ->isEmpty()
```

## Mapping rules

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

- Relationship::ownedRelatedElement

```
Set{}
```

- Type::isSufficient

```
false
```

- Association::ownedRelationship

```
let nonOwnedEnds: OrderedSet(UML::Property) = (from.memberEnd-from.ownedEnd)->asOrderedSet()
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.ocIsKindOf
let others: OrderedSet(UML::Element) = ((from.ownedElement-from.memberEnd)-generalizations)->
nonOwnedEnds->collect(e | NonOwnedEndMembership_Mapping.getMapped(e))
->union(from.ownedEnd->collect(e | OwnedEndMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))
->union(others->collect(e | ElementOwningMembership_Mapping.getMapped(e)))
->asOrderedSet()
```

- Element::name

```
from.name
```

- Element::shortName

```
null
```

- Type::isAbstract

```
false
```

- Relationship::target

```
Set{}
```

- Element::aliasId

```
Set{}
```

- Namespace::ownedImport

```
Set{}
```

- Relationship::source

```
Set{}
```

- Element::elementId

```
Helper.getID(from)
```

- Namespace::ownedRelationship

```
from.ownedElement->collect(e | ElementOwningMembership_Mapping.getMapped(e))
```

#### C.2.5.11.2.2 AssociationClass\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

AssociationCommon\_Mapping

##### Mapping Source

AssociationClass

##### Mapping Target

ConnectionDefinition

##### Owned Mappings

(none)

##### Applicable filters

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

```
not Helper.hasStereotypeApplied(from, 'SysML::Blocks::Block')
```

##### Mapping rules

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

- Type::isSufficient  
false
- Relationship::owningRelatedElement  
ElementMain\_Mapping.getMapped(from.owner)
- Element::name  
from.name
- Element::shortName  
null
- Relationship::target  
Set{}
- Element::aliasId

```
Set{}
```

- Classifier::isAbstract

```
from.isAbstract
```

- Namespace::ownedImport

```
Set{}
```

- Relationship::source

```
Set{}
```

- ConnectionDefinition::ownedRelationship

```
let nonOwnedEnds: OrderedSet(UML::Property) = (from.memberEnd-from.ownedEnd)->asOrderedSet()
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.ocIsKindOf
let others: OrderedSet(UML::Element) = ((from.ownedElement-from.memberEnd)-generalizations)->
nonOwnedEnds->collect(e | NonOwnedEndMembership_Mapping.getMapped(e))
->union(from.ownedEnd->collect(e | OwnedEndMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))
->union(others->collect(e | ElementOwningMembership_Mapping.getMapped(e)))
->asOrderedSet()
```

- Element::elementId

```
Helper.getID(from)
```

- Classifier::ownedRelationship

```
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.ocIsKindOf
let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::Feature
let toElementOMS: Set(UML::Element) = (from.ownedElement - toElementFMS) - generalizations
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))
```

- Relationship::ownedRelatedElement

```
from.relatedElement->select(e | from.ownedElement->includes(e))->collect(e | ElementMain_Map
```

#### C.2.5.11.2.3 AssociationToAnnotation\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToAnnotation\_Mapping

##### Mapping Source

Association

##### Mapping Target

Annotation

### Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  
`Set {}`
- Relationship::ownedRelatedElement  
`Set {}`
- Relationship::source  
`Set {}`
- Element::name  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Relationship::target  
`Set {}`
- Element::ownedRelationship  
`Set {}`

#### C.2.5.11.2.4 AssociationToAnnotatingFeature\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToAnnotatingElement\_Mapping

##### Mapping Source

Association

### Mapping Target

MetadataFeature

### Owned Mappings

- associationToAnnotation : AssociationToAnnotation\_Mapping
- associationToFeatureMembership : AssociationToFeatureMembership\_Mapping
- associationToFeatureTyping : AssociationToFeatureTyping\_Mapping

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  

```
Set{}
```
- MetadataFeature::name  

```
'isDerived'
```
- MetadataFeature::ownedRelationship  

```
Set{associationToFeatureMembership.to, associationToAnnotation.to, associationToFeatureTyping.to}
```
- Element::shortName  

```
null
```
- Element::elementId  

```
Helper.createUUID()
```

#### C.2.5.11.2.5 AssociationToFeatureMembership\_Mapping

### Description

\*\*\* not specified yet \*\*\*

### General Mappings

GenericToFeatureMembership\_Mapping

### Mapping Source

Association

### Mapping Target



FeatureMembership

### Owned Mappings

- associationToMetadataFeature : AssociationToMetadataFeature\_Mapping

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- FeatureMembership::ownedMemberFeature  
  
self.associationToMetadataFeatureValue.to
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
  
null
- OwningMembership::ownedRelatedElement  
  
Set{self.ownedMemberElement() }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null

- Element::ownedRelationship

Set{}

#### C.2.5.11.2.6 AssociationToFeatureTyping\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureTyping\_Mapping

##### Mapping Source

Association

##### Mapping Target

FeatureTyping

##### Owned Mappings

- associationToAnnotatingFeature : AssociationToAnnotatingFeature\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId

Set{}

- Relationship::ownedRelatedElement

Set{}

- Specialization::specific

*abstract rule*

- FeatureTyping::typedFeature

self.associationToAnnotatingFeature.to

- Element::name

null

- Element::shortName

null

- Specialization::general  
*abstract rule*
- FeatureTyping::type

```

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

```

- Element::elementId  

```

        Helper.createUUID()

```
- Element::ownedRelationship  

```

        Set{}

```

#### C.2.5.11.2.7 AssociationToMetadataFeature\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeature\_Mapping

##### Mapping Source

Association

##### Mapping Target

Feature

##### Owned Mappings

- associationToMetadataFeatureValue : AssociationToMetadataFeatureValue\_Mapping
- associationToRedefinition : AssociationToRedefinition\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId  

```

        Set{}

```
- Type::isSufficient

false

- Feature::ownedRelationship

Set{self.associationToRedefinition.to, self.associationToMetadataFeatureValue.to}

- Element::name

null

- Element::shortName

null

- Type::isAbstract

false

- Element::elementId

Helper.createUUID()

#### **C.2.5.11.2.8 AssociationToMetadataFeatureValue\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureValue\_Mapping

##### **Mapping Source**

Association

##### **Mapping Target**

FeatureValue

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName

- FeatureValue::ownedMemberElement

- `Element::shortName`

- `Element::elementId`

- `Element::aliasId`

- FeatureValue::value

- Membership::memberName

- `OwningMembership::ownedRelatedElement`

- Membership::visibility

- `Element::name`

- Element::ownedRelationship

#### C.2.5.11.2.9 AssociationToMetadataMembership\_Mapping

### Description

\*\*\* not specified yet \*\*\*

## General Mappings

GenericToFeatureMembership\_Mapping

## Mapping Source

## Association

## Mapping Target

## FeatureMembership

## Owned Mappings

- associationToAnnotatingFeature : AssociationToAnnotatingFeature\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
  
null
- OwningMembership::ownedRelatedElement  
  
Set{self.ownedMemberElement() }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- FeatureMembership::ownedMemberFeature  
  
self.associationToAnnotatingFeature.to
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- Element::ownedRelationship

Set{}

#### C.2.5.11.2.10 AssociationToRedefinition\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToRedefinition\_Mapping

##### Mapping Source

Association

##### Mapping Target

Redefinition

##### Owned Mappings

- associationToMetadataFeature : AssociationToMetadataFeature\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId

Set{}

- Redefinition::redefiningFeature

self.associationToMetadataFeatureValue.to

- Redefinition::redefinedFeature

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

- Subsetting::ownedRelatedElement

Set{}

- Subsetting::subsettingFeature

*abstract rule*

- Element::name

null

- Subsetting::subsettingFeature  
*abstract rule*
- Element::shortName

null

- Element::elementId

Helper.createUUID()

- Element::ownedRelationship

Set{}

#### C.2.5.11.2.11 BehavioeredClassifier\_Mapping

##### Description

The abstract mapping class BehavioeredClassifier\_Mapping maps the abstract metaclass UML::SimpleClassifiers::BehavioeredClassifiers to a SysMLv2::Core::Classifiers::Classifier. The mapping class is used by concrete mapping classes, for example, Block\_Mapping.

##### General Mappings

Classifier\_Mapping

##### Mapping Source

BehavioeredClassifier

##### Mapping Target

Classifier

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId

Set{}

- Type::isSufficient

false



- Namespace::ownedImport

Set{}

- Classifier::ownedRelationship

```
let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | (e.oclIsKindOf(UML::Property)))
let redefinedAttributes: Set(UML::Element) = from.ownedElement->select(e | from.oclIsKindOf(UML::Property) and e.oclIsKindOf(UML::Property))
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.oclIsKindOf(UML::Generalization))
let constraints : Set(UML::Constraint) = UML::Constraint.allInstances()->select( c | c.constraintName = from.name)
let toElementOMS: Set(UML::Element) = ((from.ownedElement - toElementFMS) - redefinedAttributes)
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(ClassifierBehavior_Mapping.getMapped(from.classifierBehavior))
```

- Element::elementId

Helper.getID(from)

- Element::name

from.name

- Element::shortName

null

- Type::isAbstract

false

- Namespace::ownedRelationship

from.ownedElement->collect(e | ElementOwningMembership\_Mapping.getMapped(e))

#### C.2.5.11.2.12 BehavioredClassifierToFeatureTyping\_Mapping

##### Description

The BehavioredClassifierToFeatureTyping\_Mapping creates the relationship from the PerformActionUsage element to its type which is the transformed SysML v1 classifier behavior.

##### General Mappings

GenericToFeatureTyping\_Mapping

##### Mapping Source

BehavioredClassifier

##### Mapping Target

FeatureTyping

## Owned Mappings

(none)

## Applicable filters

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

(none)

## Mapping rules

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

- Element::aliasId  
`Set{}`
- Relationship::ownedRelatedElement  
`Set{}`
- Specialization::specific  
*abstract rule*
- Element::name  
`null`
- Element::shortName  
`null`
- Specialization::general  
*abstract rule*
- Element::elementId  
`Helper.createUUID()`
- FeatureTyping::type  
`from`
- Element::ownedRelationship  
`Set{}`

### C.2.5.11.2.13 BehavoredClassifierToPerformActionUsage\_Mapping

#### Description

The BehavoredClassifierToPerformActionUsage\_Mapping class creates a PerformActionUsage element to call the transformed SysML v1 classifier behavior.

#### General Mappings

GenericToFeature\_Mapping

#### Mapping Source

BehavioredClassifier

### Mapping Target

PerformActionUsage

### Owned Mappings

(none)

### Applicable filters

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

(none)

### Mapping rules

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

- Element::aliasId  
`Set{ }`
- Type::isSufficient  
`false`
- PerformActionUsage::isComposite  
`true`
- PerformActionUsage::ownedRelationship  
`Set{ BehavioredClassifierToFeatureTyping_Mapping.getMapped(from) }`
- PerformActionUsage::name  
`'classifierBehavior'`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`

#### C.2.5.11.2.14 Class\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

BehavioredClassifier\_Mapping

### Mapping Source

Class

### Mapping Target

OccurrenceDefinition

### Owned Mappings

(none)

### Applicable filters

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

```
not Helper.hasStereotypeApplied(from, 'SysML::Requirements::Requirement') and not from.ocIsTypeOf(
```

### Mapping rules

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

- Element::aliasId  

```
Set{}
```
- Classifier::isAbstract  

```
from.isAbstract
```
- Type::isSufficient  

```
false
```
- Namespace::ownedImport  

```
Set{}
```
- Element::elementId  

```
Helper.getID(from)
```
- Classifier::ownedRelationship

```
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.ocIsKindOf(UML::Generalization))
let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::FeatureMembership))
let toElementOMS: Set(UML::Element) = (from.ownedElement - toElementFMS) - generalizations
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))
```

- Element::name  

```
from.name
```

- Element::shortName

null

#### C.2.5.11.2.15 ClassifierBehaviorMembership\_Mapping

##### Description

The ClassifierBehaviorMembership\_Mapping class creates a membership relationship for a PerformActionUsage element to call the transformed SysML v1 classifier behavior.

##### General Mappings

GenericToFeatureMembership\_Mapping

##### Mapping Source

BehavioredClassifier

##### Mapping Target

FeatureMembership

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName

null

- Element::shortName

null

- Element::elementId

Helper.createUUID()

- Element::aliasId

Set{ }

- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName

null

- OwningMembership::ownedRelatedElement

Set { self.ownedMemberElement () }

- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- FeatureMembership::ownedMemberFeature

BehavioeredClassifierToPerformActionUsage\_Mapping.getMapped (from)

- Membership::visibility

KerML::VisibilityKind::public

- Element::name

null

- Element::ownedRelationship

Set { }

#### C.2.5.11.2.16 ConnectionEndToSubsetting\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToSubsetting\_Mapping

##### Mapping Source

ConnectorEnd

##### Mapping Target

Subsetting

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

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

- Element::aliasId

```
Set{}
```

- Relationship::ownedRelatedElement

```
Set{}
```

- Specialization::specific

*abstract rule*

- Subsetting::subsettingFeature

```
let propertyPath: OrderedSet(UML::Property) = Helper.getTagValueAsElementColl(src, 'SysML::E
if propertyPath->isEmpty() then
    ElementMain_Mapping.getMapped(from.role)
else
    ConnectorEndToSubsettingFeature_Mapping.getMapped(from)
endif
```

- Subsetting::ownedRelationship

```
let propertyPath: OrderedSet(UML::Property) = Helper.getTagValueAsElementColl(from, 'SysML::
if propertyPath->notEmpty() then
    OrderedSet{ConnectorEndToSubsettingFeatureMembership_Mapping.getMapped(from)}
else
    OrderedSet{}
endif
```

- Element::name

```
null
```

- Subsetting::subsettingFeature

```
ConnectorEndToOwnedFeature_Mapping.getMapped(from)
```

- Element::shortName

```
null
```

- Specialization::general

*abstract rule*

- Element::elementId

```
Helper.createUUID()
```

#### C.2.5.11.2.17 Connector\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

NamedElementMain\_Mapping

GenericToConnector\_Mapping

## Mapping Source

Connector

## Mapping Target

ConnectionUsage

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

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

- Relationship::ownedRelatedElement

Set {}

- Feature::isEnd

false

- Type::isSufficient

false

- Feature::isUnique

true

- Element::shortName

null

- Type::isAbstract

false

- Feature::isOrdered

false

- Relationship::target

Set {}

- Element::aliasId

Set {}

- Feature::isPortion



false

- Feature::isReadOnly

false

- ConnectionUsage::ownedRelationship

```
from.end->collect(e | ConnectorEndToMembership_Mapping.getMapped(e))  
->including(ConnectorMultiplicityMembership_Mapping.getMapped(from))
```

- Relationship::source

Set{}

- Feature::direction

null

- Element::elementId

Helper.getID(from)

- Element::name

null

- Feature::isDerived

false

- Feature::isComposite

false

#### **C.2.5.11.2.18 ConnectorEndToFeatureCommon\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeature\_Mapping

##### **Mapping Source**

ConnectorEnd

##### **Mapping Target**

Feature

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

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

- Element::aliasId  
`Set{}`
- Type::isSufficient  
`false`
- Feature::isOrdered  
`from.isOrdered`
- Element::name  
`null`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set{}`

#### C.2.5.11.2.19 ConnectorEndToMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureMembership\_Mapping

##### Mapping Source

ConnectorEnd

##### Mapping Target

EndFeatureMembership

##### Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:

(none)

### Mapping rules

The following lists the mapping rules for the target element properties.

- EndFeatureMembership::ownedMemberFeature  
`ConnectorEndToOwnedFeature_Mapping.getMapped(from)`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{}`
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
`null`
- OwningMembership::ownedRelatedElement  
`Set{self.ownedMemberElement() }`
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
`KerML::VisibilityKind::public`
- Element::name  
`null`
- Element::ownedRelationship  
`Set{}`

#### C.2.5.11.2.20 ConnectorEndToOwnedFeature\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

ConnectorEndToFeatureCommon\_Mapping  
ElementMain\_Mapping

##### Mapping Source

ConnectorEnd

##### Mapping Target

Feature

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isEnd  

```
false
```
- Type::isSufficient  

```
false
```
- Feature::ownedRelationship  

```
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
```
- Feature::isUnique  

```
true
```
- Element::shortName  

```
null
```
- Type::isAbstract

- false
- Element::elementId
  - Helper.createUUID()
- Feature::isOrdered
  - false
- Element::aliasId
  - Set{}
- Feature::isPortion
  - false
- Feature::isReadOnly
  - false
- Feature::direction
  - null
- Element::name
  - null
- Feature::isDerived
  - false
- Feature::isComposite
  - false
- Element::ownedRelationship
  - Set{}

#### **C.2.5.11.2.21 ConnectorEndToSubsettedFeature\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

ConnectorEndToFeatureCommon\_Mapping

##### **Mapping Source**

ConnectorEnd

##### **Mapping Target**

Feature

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
let propertyPath: OrderedSet(UML::Property) = Helper.getTagValueAsElementColl(src, 'SysML::Blocks::N
propertyPath->notEmpty()
```

## Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isEnd

false

- Type::isSufficient

false

- Feature::ownedRelationship

```
let propertyPath: OrderedSet(UML::Property) = Helper.getTagValueAsElementColl(from, 'SysML::
let chain: OrderedSet(KerML::FeatureChaining) = propertyPath->collect(p | PropertyToFeatureCh
->asOrderedSet())->including(PropertyToFeatureChaining_Mapping.getMapped(from.role)) in
chain->union(OrderedSet{MultiplicityMembership_Mapping.getMapped(from)})
```

- Feature::isUnique

true

- Element::shortName

null

- Type::isAbstract

false

- Element::elementId

Helper.createUUID()

- Feature::isOrdered

false

- Element::aliasId

Set{}

- Feature::isPortion

false

- Feature::isReadOnly

- false
- Feature::direction
  - null
- Feature::isDerived
  - false
- Feature::isComposite
  - false
- Feature::name
  - 'featureChain'

#### C.2.5.11.2.22 ConnectorEndToSubsettedFeatureMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureMembership\_Mapping

##### Mapping Source

ConnectorEnd

##### Mapping Target

EndFeatureMembership

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

The following lists the mapping rules for the target element properties.

- Membership::membershipOwningNamespace
  - abstract rule*
- Membership::memberShortName
  - null
- Element::shortName

- `Element::elementId`

- `Element::aliasId`

- `OwningMembership::ownedMemberElement`  
*abstract rule*

null

- TypeFeaturing::featureOfType  
*abstract rule*

- TypeFeaturing::featuringType  
*abstract rule*

```
KerML::VisibilityKind::public
```

null

```
ConnectorEndToSubsettedFeature Mapping.getMapped(from)
```

Set { }



## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:

(none)

## Mapping rules

The following lists the mapping rules for the target element properties.

- Element::aliasId  
`Set{}`
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- OwningMembership::memberName  
`from.name+'_Connector_multiplicity'`
- OwningMembership::ownedRelatedElement  
`Set{self.ownedMemberElement() }`
- Membership::visibility  
`KerML::VisibilityKind::public`
- Element::name  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set{}`

### C.2.5.11.2.24 ConnectorType\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

## General Mappings

AssociationCommon\_Mapping

## Mapping Source

Association

## Mapping Target

ConnectionDefinition

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
(from.memberEnd->select( m | m.type.ocIsKindOf(UML::UseCase))->isEmpty()) and
(let this: UML::Association = src.oclAsType(UML::Association) in
if this.ocIsUndefined() then
    false
else
    not this.isDerived and
    not this.ocIsTypeOf(UML::AssociationClass) and
    Helper.isConnectionDef(this)
endif)
```

## Mapping rules

The following lists the mapping rules for the target element properties.

- Type::isSufficient  
false
- Relationship::owningRelatedElement  
ElementMain\_Mapping.getMapped(from.owner)
- Element::name  
from.name
- Element::shortName  
null
- Relationship::target  
Set{}
- Element::aliasId  
Set{}

- Classifier::isAbstract

```
from.isAbstract
```

- Namespace::ownedImport

```
Set{}
```

- Relationship::source

```
Set{}
```

- Element::elementId

```
Helper.getID(from)
```

- Classifier::ownedRelationship

```
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.ocIsKindOf(UML::Generalization))
let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::FeatureMembership))
let toElementOMS: Set(UML::Element) = (from.ownedElement - toElementFMS) - generalizations
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))
```

- Relationship::ownedRelatedElement

```
from.relatedElement->select(e | from.ownedElement->includes(e))->collect(e | ElementMain_Mapping.getMapped(e))
```

#### C.2.5.11.2.25 ConnectorTypeDerived\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

AssociationCommon\_Mapping

##### Mapping Source

Association

##### Mapping Target

ConnectionDefinition

##### Owned Mappings

- associationToMetadataMembership : AssociationToMetadataMembership\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
(from.memberEnd->select( m | m.type.ocIsKindOf(UML::UseCase))->isEmpty()) and
(let this: UML::Association = src.ocAsType(UML::Association) in
```

```

if this.ocIsUndefined() then
  false
else
  this.isDerived and
  not this.ocIsTypeOf(UML::AssociationClass) and
  Helper.isConnectionDef(this)
endif)

```

## Mapping rules

The following lists the mapping rules for the target element properties.

- **Type::isSufficient**

```
false
```

- **Relationship::owningRelatedElement**

```
ElementMain_Mapping.getMapped(from.owner)
```

- **ConnectionDefinition::ownedRelationship**

```

let nonOwnedEnds: OrderedSet(UML::Property) = (from.memberEnd-from.ownedEnd)->asOrderedSet()
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.ocIsKindOf
let others: OrderedSet(UML::Element) = ((from.ownedElement-from.memberEnd)-generalizations)->
nonOwnedEnds->collect(e | NonOwnedEndMembership_Mapping.getMapped(e))
->union(from.ownedEnd->collect(e | OwnedEndMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))
->union(others->collect(e | ElementOwningMembership_Mapping.getMapped(e)))
->asOrderedSet()
->append(self.associationToMetadataMembership.to)

```

- **Element::name**

```
from.name
```

- **Element::shortName**

```
null
```

- **Relationship::target**

```
Set{}
```

- **Element::aliasId**

```
Set{}
```

- **Classifier::isAbstract**

```
from.isAbstract
```

- **Namespace::ownedImport**

```
Set{}
```

- **Relationship::source**

```
Set{}
```

- Element::elementId

```
Helper.getID(from)
```

- Classifier::ownedRelationship

```
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.ocIsKindOf
let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::Featur
let toElementOMS: Set(UML::Element) = (from.ownedElement - toElementFMS) - generalizations
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))
```

- Relationship::ownedRelatedElement

```
from.relatedElement->select(e | from.ownedElement->includes(e))->collect(e | ElementMain_Map
```

### C.2.5.11.2.26 End\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

PropertyCommon\_Mapping

#### Mapping Source

Property

#### Mapping Target

Feature

#### Owned Mappings

(none)

#### Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
src.ocIsKindOf(UML::Property) and not src.ocAsType(UML::Property).association.ocIsUndefined()
```

#### Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isEnd

```
false
```

- Feature::isOrdered

- from.isOrdered
- Type::isSufficient
  - false
- Feature::isAbstract
  - false
- Element::shortName
  - null
- Feature::ownedRelationship
  - ```

let typing: KerML::FeatureTyping = StructuralFeatureToFeatureTyping_Mapping.getMapped(from)
if typing.oclIsUndefined() then
    Set{MultiplicityMembership_Mapping.getMapped(from)}
else
    Set{MultiplicityMembership_Mapping.getMapped(from), typing}
endif

```
- Element::elementId
  - Helper.createUUID()
- Element::aliasId
  - Set{}
- Feature::isPortion
  - false
- Feature::direction
  - null
- Element::name
  - null
- Feature::isDerived
  - false
- Feature::isComposite
  - false
- Feature::isEnd
  - true
- Feature::isUnique
  - from.isUnique

- Feature::isReadOnly  
*abstract rule*

#### **C.2.5.11.2.27 EndMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

StructuralFeatureMembership\_Mapping

##### **Mapping Source**

Property

##### **Mapping Target**

EndFeatureMembership

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedRelatedElement  
`Set{self.ownedMemberFeature() }`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{ }`

- FeatureMembership::ownedMemberFeature  
*abstract rule*
- FeatureMembership::owningType  
*abstract rule*
- Membership::memberName

null

- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility

KerML::VisibilityKind::public

- Element::name

null

- Element::ownedRelationship

Set{}

#### **C.2.5.11.2.28 NonOwnedEndSubsetting\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToSubsetting\_Mapping

##### **Mapping Source**

Property

##### **Mapping Target**

Subsetting

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:

(none)

##### **Mapping rules**

The following lists the mapping rules for the target element properties.

- Element::aliasId



Set{}

- Relationship::ownedRelatedElement

Set{}

- Subsetting::subsettingFeature

Property\_Mapping.getMapped(from)

- Specialization::specific  
*abstract rule*
- Element::name

null

- Element::shortName

null

- Specialization::general  
*abstract rule*
- Element::elementId

Helper.createUUID()

- Element::ownedRelationship

Set{}

#### C.2.5.11.2.29 EndToSubsettingFeature\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

PropertyCommon\_Mapping

##### Mapping Source

Property

##### Mapping Target

Feature

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```

let property: UML::Property = src.oclAsType(UML::Property) in
not property.association.oclIsUndefined()
and property.association.ownedEnd->excludes(property)

```

## Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isEnd

```
false
```

- Feature::isOrdered

```
from.isOrdered
```

- Type::isSufficient

```
false
```

- Feature::isAbstract

```
false
```

- Element::shortName

```
null
```

- Feature::ownedRelationship

```

let typing: KerML::FeatureTyping = StructuralFeatureToFeatureTyping_Mapping.getMapped(from)
if typing.oclIsUndefined() then
    Set{MultiplicityMembership_Mapping.getMapped(from)}
else
    Set{MultiplicityMembership_Mapping.getMapped(from), typing}
endif

```

- Element::elementId

```
Helper.createUUID()
```

- Element::aliasId

```
Set{}
```

- Feature::isPortion

```
false
```

- Feature::ownedRelationship

```

let chain: OrderedSet(KerML::FeatureChaining) = OrderedSet{EndToSubsettedFeatureChaining_Map
chain->including(MultiplicityMembership_Mapping.getMapped(from))

```

- Feature::direction

```
null
```

- Element::name

null

- Feature::isDerived

false

- Feature::isComposite

false

- Feature::isUnique

from.isUnique

- Feature::isReadOnly

*abstract rule*

#### **C.2.5.11.2.30 EndToSubsettedFeatureChaining\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToRelationship\_Mapping

##### **Mapping Source**

Property

##### **Mapping Target**

FeatureChaining

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:

(none)

##### **Mapping rules**

The following lists the mapping rules for the target element properties.

- Element::aliasId

Set{ }

- FeatureChaining::name

'featureChain'

- FeatureChaining::chainingFeature

```
Property_Mapping.getMapped(from)
```

- Element::shortName

```
null
```

- Element::elementId

```
Helper.createUUID()
```

- Element::ownedRelationship

```
Set{}
```

#### C.2.5.11.2.31 NonOnedEndToSubsettedFeatureMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### 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.ocIsKindOf(UML::Property) and not src.ocIsType(UML::Property).association.ocIsUndefined()
```

##### Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedMemberFeature

```
EndToSubsettedFeature_Mapping.getMapped(from)
```

- Membership::membershipOwningNamespace

*abstract rule*

- Membership::memberShortName

```
null
```

- Element::shortName

- `Element::elementId`

- `Element::elementId`

```
Helper.createUUID()
```

- `Element::aliasId`

Set { }

- `OwningMembership::ownedMemberElement`  
*abstract rule*

*abstract rule*

- Membership::memberName

null

- `OwningMembership::ownedRelatedElement`

```
Set { self.ownedMemberElement () }
```

- `TypeFeaturing::featureOfType`  
*abstract rule*

*abstract rule*

- `TypeFeaturing::featuringType`  
*abstract rule*

*abstract rule*

- Membership::visibility

```
KerML::VisibilityKind::public
```

- `Element::name`

null

- Element::ownedRelationship

Set { }

#### C.2.5.11.2.32 NonOwnedEnd\_Mapping

### Description

\*\*\* not specified yet \*\*\*

## General Mappings

End\_Mapping

## Mapping Source

### Property

## Mapping Target

Feature

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isOrdered  

```
from.isOrdered
```
- Type::isSufficient  

```
false
```
- Feature::isComposite  

```
from.isComposite
```
- Feature::isAbstract  

```
false
```
- Feature::isEnd  

```
if from.association.oclIsUndefined() then  
    false  
else  
    from.association.ownedEnd->includes(from)  
endif
```
- Element::shortName  

```
null
```
- Element::elementId  

```
Helper.createUUID()
```
- Feature::ownedRelationship  

```
Set{MultiplicityMembership_Mapping.getMapped(from)  
    ,StructuralFeatureToFeatureTyping_Mapping.getMapped(from)  
    ,NonOwnedEndSubsettingMembership_Mapping.getMapped(from)  
    ,NonOwnedEndToSubsettingFeatureMembership_Mapping.getMapped(from)}
```
- Element::aliasId  

```
Set{}
```
- Feature::isPortion  

```
false
```
- Feature::isDerived  

```
from.isDerived
```

- Feature::direction  
null
- Feature::isUnique  
from.isUnique
- Feature::isReadOnly  
*abstract rule*
- Feature::name  
'nonOwnedEnd'

#### C.2.5.11.2.33 NonOwnedEndMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### 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.ocIsKindOf(UML::Property)
and not src.ocAsType(UML::Property).association.ocIsUndefined()
and src.ocAsType(UML::Property).association.ownedEnd->excludes(src)
```

##### Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedRelatedElement  
Set{self.ownedMemberFeature() }
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
null

- EndFeatureMembership::ownedMemberFeature  
NonOwnedEnd\_Mapping.getMapped(from)
- Element::shortName  
null
- FeatureMembership::visibility  
if (from.ocllsKindOf(UML::NamedElement)) then  
    Helper.getKerMLVisibilityKind(from.oclAsType(UML::NamedElement).visibility)  
else  
    KerML::VisibilityKind::public  
endif
- Element::elementId  
Helper.createUUID()
- Element::aliasId  
Set{}
- FeatureMembership::owningType  
*abstract rule*
- Membership::memberName  
null
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Element::name  
null
- Element::ownedRelationship  
Set{}

#### C.2.5.11.2.34 NonOwnedEndSubsettingMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToOwningMembership\_Mapping

##### Mapping Source

Property

##### Mapping Target



OwningMembership

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:

(none)

## Mapping rules

The following lists the mapping rules for the target element properties.

- Relationship::ownedRelatedElement  
`Set{}`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Membership::memberElement  
*abstract rule*
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{}`
- Membership::memberName  
`null`
- OwningMembership::ownedMemberElement  
`NonOwnedEndSubsetting_Mapping.getMapped(from)`
- Membership::visibility  
`KerML::VisibilityKind::public`
- Element::name  
`null`
- Element::ownedRelationship  
`Set{}`

### C.2.5.11.2.35 OwnedEnd\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

End\_Mapping

NamedElementMain\_Mapping

#### Mapping Source

Property

#### Mapping Target

Feature

#### Owned Mappings

(none)

#### Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
let p: UML::Property = src.oclAsType(UML::Property) in
not p.oclIsUndefined() and
(not p.association.oclIsUndefined() and p.association.ownedEnd->includes(p)) and
(not p.association.memberEnd->select( m | (not m.type.oclIsUndefined()) and m.type.oclIsTypeOf(UML::
```

#### Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isOrdered  
`from.isOrdered`
- Type::isSufficient  
`false`
- Feature::isComposite  
`from.isComposite`
- Feature::ownedRelationship

```
let typing: KerML::FeatureTyping = StructuralFeatureToFeatureTyping_Mapping.getMapped(from) in
let subsetting: Set(KerML::Subsetting) = from.subsettedProperty->collect(p | PropertySubsetting)
let subsettingMultiplicityTyping: Set(KerML::Relationship) = subsetting->union(if typing.occl
Set{MultiplicityMembership_Mapping.getMapped(from)}
else
Set{MultiplicityMembership_Mapping.getMapped(from), typing}
endif)->asSet() in
```

```

let relationships: Set(KerML::Relationship) = if from.defaultValue.ocIsTypeOf(UML::OpaqueExp

if from.defaultValue.ocIsUndefined() then
    relationships
else
    relationships->including(if from.defaultValue.ocIsTypeOf(UML::OpaqueExpression) then Pro
endif

```

- Feature::isAbstract

```
false
```

- Feature::isEnd

```

if from.association.ocIsUndefined() then
    false
else
    from.association.ownedEnd->includes(from)
endif

```

- Element::shortName

```
null
```

- Element::aliasId

```
Set{}
```

- Feature::isPortion

```
false
```

- Feature::isDerived

```
from.isDerived
```

- Feature::direction

```
null
```

- Element::elementId

```
Helper.getID(from)
```

- Element::name

```
null
```

- Feature::isUnique

```
from.isUnique
```

- Feature::isReadOnly

```
abstract rule
```

#### C.2.5.11.2.36 Port\_Mapping

##### Description

A port which is untyped or typed by an interface block is mapped to a SysMLv2::PortUsage.

## General Mappings

Part\_Mapping

## Mapping Source

Port

## Mapping Target

PortUsage

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
if from.type.ocIsUndefined() then false
else
let p: UML::Property = src.ocAsType(UML::Property) in
not p.ocIsUndefined() and
not p.type.ocIsKindOf(UML::DataType) and
not (p.name.indexOf('base_') > 0) and
(p.association.ocIsUndefined() or p.association.ownedEnd->excludes(p))
endif

,

result =
not Helper.hasStereotypeApplied(from.owner, 'SysML::ConstraintBlocks::ConstraintBlock') or
((from.type.ocIsUndefined() or Helper.hasStereotypeApplied(from.type, 'SysML::Ports&Flows::Interface')
and not (Helper.hasStereotypeApplied(from, 'SysML::Ports&Flows::FullPort') or (from.type.ocIsKindOf
```

## Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isOrdered

```
from.isOrdered
```

- Type::isSufficient

```
false
```

- Feature::isComposite

```
from.isComposite
```

- Feature::ownedRelationship

```
let typing: KerML::FeatureTyping = StructuralFeatureToFeatureTyping_Mapping.getMapped(from) i
```

```

let subsetting: Set(KerML::Subsetting) = from.subsettedProperty->collect(p | PropertySubsetting)
let subsettingMultiplicityTyping: Set(KerML::Relationship) = subsetting->union(if typing.oclc
    Set{MultiplicityMembership_Mapping.getMapped(from)}
else
    Set{MultiplicityMembership_Mapping.getMapped(from), typing}
endif)->asSet() in

let relationships: Set(KerML::Relationship) = if from.defaultValue.oclcIsTypeOf(UML::OpaqueExp

if from.defaultValue.oclcIsUndefined() then
    relationships
else
    relationships->including(if from.defaultValue.oclcIsTypeOf(UML::OpaqueExpression) then Pro
endif

```

- **Feature::isAbstract**

```

false

```

- **Feature::isEnd**

```

if from.association.oclcIsUndefined() then
    false
else
    from.association.ownedEnd->includes(from)
endif

```

- **Element::name**

```

from.name

```

- **Element::shortName**

```

null

```

- **Element::aliasId**

```

Set{}

```

- **Feature::isPortion**

```

false

```

- **Feature::isDerived**

```

from.isDerived

```

- **Feature::direction**

```

null

```

- **Element::elementId**

```

Helper.getID(from)

```

- **Feature::isUnique**

```

from.isUnique

```

- Feature::isReadOnly  
*abstract rule*

### C.2.5.11.2.37 OwnedEndAttribute\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

OwnedEnd\_Mapping  
Attribute\_Mapping

#### Mapping Source

Property

#### Mapping Target

AttributeUsage

#### 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.type.oclIsUndefined() and p.type.oclIsKindOf(UML::DataType))

,

src.oclIsKindOf(UML::Property) and not src.oclAsType(UML::Property).association.oclIsUndefined()
```

#### Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isOrdered  
*from.isOrdered*
- Type::isSufficient  
*false*
- Feature::isComposite  
*from.isComposite*
- Feature::ownedRelationship

```

let typing: KerML::FeatureTyping = StructuralFeatureToFeatureTyping_Mapping.getMapped(from)
let subsetting: Set(KerML::Subsetting) = from.subsettedProperty->collect(p | PropertySubsetting)
let subsettingMultiplicityTyping: Set(KerML::Relationship) = subsetting->union(if typing.ocIsTypeOf(UML::MultiplicityMembership_Mapping.getMapped(from))
    Set{MultiplicityMembership_Mapping.getMapped(from)}
else
    Set{MultiplicityMembership_Mapping.getMapped(from), typing}
endif)->asSet() in

let relationships: Set(KerML::Relationship) = if from.defaultValue.ocIsTypeOf(UML::OpaqueExpression) then
    from.defaultValue
else
    from.defaultValue
endif

if from.defaultValue.ocIsUndefined() then
    relationships
else
    relationships->including(if from.defaultValue.ocIsTypeOf(UML::OpaqueExpression) then from.defaultValue)
endif

```

- **Feature::isAbstract**

false

- **Element::name**

from.name

- **Element::shortName**

null

- **Element::aliasId**

Set{}

- **Feature::isPortion**

false

- **Feature::isDerived**

from.isDerived

- **Feature::direction**

null

- **Element::elementId**

Helper.getID(from)

- **Feature::isEnd**

true

- **Feature::isUnique**

from.isUnique

- **Feature::isReadOnly**

*abstract rule*

#### C.2.5.11.2.38 OwnedEndMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### 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.ocIsKindOf(UML::Property)
and not src.ocAsType(UML::Property).association.ocIsUndefined()
and src.ocAsType(UML::Property).association.ownedEnd->includes(src)
```

##### Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedRelatedElement

```
Set{self.ownedMemberFeature() }
```
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName

```
null
```
- Element::shortName

```
null
```
- FeatureMembership::visibility

```
if (from.ocIsKindOf(UML::NamedElement)) then
    Helper.getKerMLVisibilityKind(from.ocAsType(UML::NamedElement).visibility)
else
    KerML::VisibilityKind::public
endif
```
- Element::elementId



Helper.createUUID()

- Element::aliasId

Set{}

- FeatureMembership::owningType  
*abstract rule*
- Membership::memberName

null

- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Element::name

null

- EndFeatureMembership::ownedMemberFeature

OwnedEnd\_Mapping.getMapped(from)

- Element::ownedRelationship

Set{}

#### C.2.5.11.2.39 PropertyToFeatureChaining\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToRelationship\_Mapping

##### Mapping Source

Property

##### Mapping Target

FeatureChaining

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

The following lists the mapping rules for the target element properties.

- Element::aliasId  
Set{}
- FeatureChaining::chainingFeature  
ElementMain\_Mapping.getMapped(from)
- Element::name  
null
- Element::shortName  
null
- Element::elementId  
Helper.createUUID()
- Element::ownedRelationship  
Set{}

## C.2.5.12 UseCases

### C.2.5.12.1 Overview

**Table 27. List of all Overview Mapping Specifications**

| SysML v1 Concept | SysML v2 Concept                                                                                                         | Mapping Class                                                                                                                                                                                                                                        | Filter |
|------------------|--------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| Actor            | PartDefinition                                                                                                           | Actor_Mapping                                                                                                                                                                                                                                        |        |
| Extend           | Relationship<br>FeatureMembership<br>ReferenceUsage<br>OwningMembership<br>Element<br>Documentation<br>SubjectMembership | DirectedRelationship_Mapping<br>ElementFeatureMembership_Mapping<br>RequirementSubject_Mapping<br>RequirementDocumentationMembership_Mapping<br>NamedElementMain_Mapping<br>RequirementDocumentation_Mapping<br>RequirementSubjectMembership_Mapping |        |
| ExtensionPoint   | FeatureMembership<br>ReferenceUsage<br>OwningMembership<br>Element<br>Documentation<br>SubjectMembership                 | ElementFeatureMembership_Mapping<br>RequirementSubject_Mapping<br>RequirementDocumentationMembership_Mapping<br>NamedElementMain_Mapping<br>RequirementDocumentation_Mapping<br>RequirementSubjectMembership_Mapping                                 |        |
| Include          | FeatureMembership<br>IncludeUseCaseUsage<br>FeatureTyping                                                                | IncludeMembership_Mapping<br>Include_Mapping<br>IncludeFeatureTyping_Mapping                                                                                                                                                                         |        |
| UseCase          | UseCaseDefinition                                                                                                        | UseCase_Mapping                                                                                                                                                                                                                                      |        |

### C.2.5.12.2 SysML v1 UseCases elements not mapped

Table 28. List of SysML v1 elements not mapped of this section

| SysML v1 Concept | Rationale                                                                                                                                                        |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Extend           | The semantics of the UML4SysML::Extend relationship is not supported by SysML v2.                                                                                |
| ExtensionPoint   | The semantics of the UML4SysML::Extend relationship is not supported by SysML v2 Therefore, UML4SysML::ExtensionPoint is also not covered by the transformation. |

### C.2.5.12.3 Mapping Specifications

#### C.2.5.12.3.1 Actor\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

ElementMain\_Mapping

BehavioredClassifier\_Mapping

##### Mapping Source

Actor

##### Mapping Target

PartDefinition

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:

(none)

##### Mapping rules

The following lists the mapping rules for the target element properties.

- Element::aliasId  
Set { }
- Classifier::isAbstract  
from.isAbstract
- Type::isSufficient

false

- Namespace::ownedImport

Set{}

- Element::elementId

Helper.getID(from)

- Classifier::ownedRelationship

```
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.ocIsKindOf(UML::Generalization))
let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::FeatureMembership))
let toElementOMS: Set(UML::Element) = (from.ownedElement - toElementFMS) - generalizations
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))
```

- Element::name

from.name

- Element::shortName

null

#### C.2.5.12.3.2 CaseActor\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToPartUsage\_Mapping

##### Mapping Source

Property

##### Mapping Target

PartUsage

##### Owned Mappings

- useCaseActorFeatureTyping : CaseActorFeatureTyping\_Mapping

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

The following lists the mapping rules for the target element properties.

- `Feature::isEnd`  
`false`
- `Type::isSufficient`  
`false`
- `PartUsage::ownedRelationship`  
`Set{useCaseActorFeatureTyping.to}`
- `Feature::isUnique`  
`true`
- `Element::shortName`  
`null`
- `Type::isAbstract`  
`false`
- `Element::elementId`  
`Helper.createUUID()`
- `Feature::isOrdered`  
`false`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `Usage::isVariation`  
`false`
- `Feature::isReadOnly`  
`false`
- `Feature::direction`  
`null`
- `PartUsage::name`  
`from.name`
- `Feature::isDerived`

false

- Feature::isComposite

false

### C.2.5.12.3.3 CaseActorFeatureTyping\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

GenericToFeatureTyping\_Mapping

#### Mapping Source

Property

#### Mapping Target

FeatureTyping

#### Owned Mappings

- useCaseActor : CaseActor\_Mapping

#### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

#### Mapping rules

The following lists the mapping rules for the target element properties.

- Element::aliasId  

Set { }
- Relationship::ownedRelatedElement  

Set { }
- Specialization::specific  
*abstract rule*
- FeatureTyping::typedFeature  

useCaseActor.to
- FeatureTyping::type  

from.type
- Element::name

null

- Element::shortName

null

- Specialization::general  
*abstract rule*
- Element::elementId

Helper.createUUID()

- Element::ownedRelationship

Set{}

#### C.2.5.12.3.4 CaseActorMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToParameterMembership\_Mapping

##### Mapping Source

Property

##### Mapping Target

ActorMembership

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedRelatedElement

Set{self.ownedMemberFeature() }

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName

null

- Element::shortName  
null
- Element::elementId  
Helper.createUUID()
- ActorMembership::ownedMemberParameter  
CaseActor\_Mapping.getMapped(from)
- Element::aliasId  
Set{}
- FeatureMembership::ownedMemberFeature  
*abstract rule*
- FeatureMembership::owningType  
*abstract rule*
- Membership::memberName  
null
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
KerML::VisibilityKind::public
- Element::name  
null
- Element::ownedRelationship  
Set{}

#### C.2.5.12.3.5 Include\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToOccurrenceUsage\_Mapping

##### Mapping Source

Include

##### Mapping Target

IncludeUseCaseUsage



## Owned Mappings

- includeFeatureTyping : IncludeFeatureTyping\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Usage::isVariation  
`false`
- Feature::isReadOnly  
`false`
- Feature::direction

null

- Element::name

    null

- Feature::isDerived

    false

- Feature::isComposite

    false

- Element::ownedRelationship

    Set{}

#### **C.2.5.12.3.6 IncludeFeatureTyping\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureTyping\_Mapping

##### **Mapping Source**

Include

##### **Mapping Target**

FeatureTyping

##### **Owned Mappings**

- includeUsage : Include\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

The following lists the mapping rules for the target element properties.

- Element::aliasId

    Set{}

- Relationship::ownedRelatedElement

    Set{}

- FeatureTyping::typedFeature

- includeUsage.to
- Specialization::specific  
*abstract rule*
- Element::name  
  
null
- Element::shortName  
  
null
- FeatureTyping::type  
  
from.addition
- Specialization::general  
*abstract rule*
- Element::elementId  
  
Helper.createUUID()
- Element::ownedRelationship  
  
Set{ }

#### **C.2.5.12.3.7 IncludeMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureMembership\_Mapping

##### **Mapping Source**

Include

##### **Mapping Target**

FeatureMembership

##### **Owned Mappings**

- includeUsage : Include\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

The following lists the mapping rules for the target element properties.

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName

    null

- Element::shortName

    null

- Element::elementId

    Helper.createUUID()

- Element::aliasId

    Set{}

- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName

    null

- FeatureMembership::ownedMemberFeature

    includeUsage.to

- OwningMembership::ownedRelatedElement

    Set{self.ownedMemberElement() }

- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility

    KerML::VisibilityKind::public

- Element::name

    null

- Element::ownedRelationship

    Set{}

#### C.2.5.12.3.8 UseCase\_Mapping

##### Description

The expected SysML v2 textual syntax of a mapped UML4SysML::UseCase with a defined subject is as follows.

```
use case def ThisIsAUseCase {  
  subject subject_ThisIsABlock : ThisIsABlock;  
}
```

Currently, only one use case subject is supported by the mapping class. Since the UML4SysML::Extend relationship is not considered by the SysML v1 to SysML v2 transformation, the extension points of a use case are also not mapped.

### General Mappings

BehavioredClassifier\_Mapping

NamedElementMain\_Mapping

### Mapping Source

UseCase

### Mapping Target

UseCaseDefinition

### Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:

(none)

### Mapping rules

The following lists the mapping rules for the target element properties.

- Element::aliasId

```
Set{}
```

- Classifier::isAbstract

```
from.isAbstract
```

- Type::isSufficient

```
false
```

- Namespace::ownedImport

```
Set{}
```

- UseCaseDefinition::ownedRelationship

```
let properties : Set(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::Property))
let actors : Set(UML::Property) = UML::Association.allInstances()->collect(m | m.memberEnd)->select(e | e.ocIsKindOf(UML::Property))
let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.ocIsKindOf(UML::Generalization))
let extensionPoints : Sequence(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::ExtensionPoint))
let extend : Sequence(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::Extend))
let include : Sequence(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::Include))
let elements : Set(UML::Element) = (((((from.ownedElement-properties) - generalizations) - extensionPoints) - extend) - include)
let relationships : Sequence(KerML::Relationship) =
elements->collect(e | ElementOwningMembership_Mapping.getMapped(e))
```

```

->union(properties->collect(e | PropertyMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))
->including(CaseSubjectMembership_Mapping.getMapped(from))
->union(actors->collect(e | CaseActorMembership_Mapping.getMapped(e)))
->including(CaseObjectiveMembership_Mapping.getMapped(from))
->including(CommonReturnParameterReferenceUsageMembership_Mapping.getMapped(from)) in
if from.classifierBehavior.oclIsUndefined() then relationships else relationships->including

```

- Element::elementId

```

Helper.getID(from)

```

- Classifier::ownedRelationship

```

let generalizations : Set(UML::Generalization) = from.ownedElement->select(e | e.ocIsKindOf
let toElementFMS: Set(UML::Element) = from.ownedElement->select(e | e.ocIsKindOf(UML::Featur
let toElementOMS: Set(UML::Element) = (from.ownedElement - toElementFMS) - generalizations i
toElementOMS->collect(e | ElementOwningMembership_Mapping.getMapped(e))
->union(toElementFMS->collect(e | ElementFeatureMembership_Mapping.getMapped(e)))
->union(generalizations->collect(e | Generalization_Mapping.getMapped(e)))

```

- Element::name

```

from.name

```

- Element::shortName

```

null

```

#### C.2.5.12.3.9 CaseObjectiveMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureMembership\_Mapping

##### Mapping Source

Classifier

##### Mapping Target

ObjectiveMembership

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

The following lists the mapping rules for the target element properties.

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
    null
- Element::shortName  
  
    null
- Element::elementId  
  
    Helper.createUUID()
- Element::aliasId  
  
    Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- ObjectiveMembership::ownedMemberFeature  
  
    CaseObjectiveRequirementUsage\_Mapping.getMapped(from)
- Membership::memberName  
  
    null
- OwningMembership::ownedRelatedElement  
  
    Set{self.ownedMemberElement() }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
  
    KerML::VisibilityKind::public
- Element::name  
  
    null
- Element::ownedRelationship  
  
    Set{}

#### C.2.5.12.3.10 CaseEmptySubjectReferenceUsage\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

## General Mappings

GenericToReferenceUsage\_Mapping

## Mapping Source

Classifier

## Mapping Target

ReferenceUsage

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isEnd  
false
- Type::isSufficient  
false
- Feature::isUnique  
true
- Element::shortName  
null
- Type::isAbstract  
false
- Element::elementId  
Helper.createUUID()
- Feature::isOrdered  
false
- Element::aliasId  
Set{ }



- Feature::isPortion  
false
- Usage::isVariation  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false
- Element::ownedRelationship  
Set{ }

#### **C.2.5.12.3.11 CaseObjectiveRequirementUsage\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToUsage\_Mapping

##### **Mapping Source**

Classifier

##### **Mapping Target**

RequirementUsage

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

The following lists the mapping rules for the target element properties.

- `Feature::isEnd`  
`false`
- `Type::isSufficient`  
`false`
- `RequirementUsage::ownedRelationship`  
`Set{CaseSubjectMembership_Mapping.getMapped(from), CommonReturnParameterReferenceUsageMember`
- `Feature::isUnique`  
`true`
- `Element::shortName`  
`null`
- `Type::isAbstract`  
`false`
- `Element::elementId`  
`Helper.createUUID()`
- `Feature::isOrdered`  
`false`
- `Element::aliasId`  
`Set{}`
- `Feature::isPortion`  
`false`
- `Feature::isReadOnly`  
`false`
- `Feature::direction`  
`null`
- `Element::name`  
`null`
- `Feature::isDerived`  
`false`

- Feature::isComposite

false

### C.2.5.12.3.12 CaseSubjectMembership\_Mapping

#### Description

The current version only supports one specified subject.

#### General Mappings

GenericToParameterMembership\_Mapping

#### Mapping Source

Classifier

#### Mapping Target

SubjectMembership

#### Owned Mappings

(none)

#### Applicable filters

This mapping applies only if the following (OCL) condition is verified:

(none)

#### Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedRelatedElement

```
Set{self.ownedMemberFeature() }
```

- Membership::membershipOwningNamespace

*abstract rule*

- Membership::memberShortName

null

- SubjectMembership::ownedMemberParameter

```
if (from.ocIsTypeOf(UML::UseCase)) and (from.ocAsType(UML::UseCase).subject->size() > 0) t
```

- Element::shortName

null

- Element::elementId

```
Helper.createUUID()
```

- Element::aliasId  
`Set {}`
- FeatureMembership::ownedMemberFeature  
*abstract rule*
- FeatureMembership::owningType  
*abstract rule*
- Membership::memberName  
`null`
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
`KerML::VisibilityKind::public`
- Element::name  
`null`
- Element::ownedRelationship  
`Set {}`

### C.2.5.12.3.13 CaseSubjectFeatureTyping\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

GenericToFeatureTyping\_Mapping

#### Mapping Source

Classifier

#### Mapping Target

FeatureTyping

#### Owned Mappings

- useCaseSubjectReferenceUsage : CaseSubjectReferenceUsage\_Mapping

#### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

#### Mapping rules

The following lists the mapping rules for the target element properties.

- Element::aliasId  
`Set{}`
- Relationship::ownedRelatedElement  
`Set{}`
- Specialization::specific  
*abstract rule*
- FeatureTyping::type  
`if from->size() > 0 then from->get(0) else OclUndefined endif`
- Element::name  
`null`
- Element::shortName  
`null`
- Specialization::general  
*abstract rule*
- FeatureTyping::typedFeature  
`useCaseSubjectReferenceUsage.to`
- Element::elementId  
`Helper.createUUID()`
- Element::ownedRelationship  
`Set{}`

#### **C.2.5.12.3.14 CaseSubjectReferenceUsage\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

CaseEmptySubjectReferenceUsage\_Mapping

##### **Mapping Source**

Classifier

##### **Mapping Target**

ReferenceUsage

## Owned Mappings

- useCaseSubjectFeatureTyping : CaseSubjectFeatureTyping\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- ReferenceUsage::ownedRelationship  
`Set{useCaseSubjectFeatureTyping.to}`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Usage::isVariation  
`false`
- Feature::isReadOnly

false

- Feature::direction

null

- Feature::isDerived

false

- ReferenceUsage::name

'subject\_' + from->get(0).name

- Feature::isComposite

false

### C.2.5.13 Values

#### C.2.5.13.1 Overview

**Table 29. List of all Overview Mapping Specifications**

| SysML v1 Concept    | SysML v2 Concept                                                                                         | Mapping Class                                                                                                                                                                                                        | Filter                          |
|---------------------|----------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|
| Duration            | Expression<br>FeatureValue<br>FeatureValue                                                               | ValueSpecification_Mapping<br>PropertyDefaultValue_Mapping<br>SlotValue_Mapping                                                                                                                                      | src.owner.ocIsKindOf(UML::Slot) |
| DurationConstraint  | AssertConstraintUsage<br>ConstraintDefinition<br>FeatureTyping<br>FeatureMembership                      | ConstraintUsage_Mapping<br>Constraint_Mapping<br>ConstraintUsageFeatureTyping_Mapping<br>ConstrainedElementFeatureMembership_Mapping                                                                                 |                                 |
| DurationInterval    | Expression<br>FeatureValue<br>FeatureValue                                                               | ValueSpecification_Mapping<br>PropertyDefaultValue_Mapping<br>SlotValue_Mapping                                                                                                                                      | src.owner.ocIsKindOf(UML::Slot) |
| DurationObservation | FeatureMembership<br>ReferenceUsage<br>OwningMembership<br>Element<br>Documentation<br>SubjectMembership | ElementFeatureMembership_Mapping<br>RequirementSubject_Mapping<br>RequirementDocumentationMembership_Mapping<br>NamedElementMain_Mapping<br>RequirementDocumentation_Mapping<br>RequirementSubjectMembership_Mapping |                                 |
| Expression          | OperatorExpression<br>OwningMembership<br>TextualRepresentation                                          | Expression_Mapping<br>ExpressionElseMembership_Mapping<br>ExpressionElseSpecification_Mapping                                                                                                                        |                                 |
| Interval            | Expression<br>FeatureValue<br>FeatureValue                                                               | ValueSpecification_Mapping<br>PropertyDefaultValue_Mapping<br>SlotValue_Mapping                                                                                                                                      | src.owner.ocIsKindOf(UML::Slot) |
| IntervalConstraint  | AssertConstraintUsage<br>ConstraintDefinition<br>FeatureTyping<br>FeatureMembership                      | ConstraintUsage_Mapping<br>Constraint_Mapping<br>ConstraintUsageFeatureTyping_Mapping<br>ConstrainedElementFeatureMembership_Mapping                                                                                 |                                 |
| LiteralBoolean      | LiteralBoolean                                                                                           | LiteralBoolean_Mapping                                                                                                                                                                                               |                                 |

| SysML v1 Concept        | SysML v2 Concept                                                                                                                                                                                                                                                                                                                             | Mapping Class                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Filter                                                                                                                                                                       |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LiteralInteger          | LiteralInteger                                                                                                                                                                                                                                                                                                                               | LiteralInteger_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                              |
| LiteralNull             | NullExpression                                                                                                                                                                                                                                                                                                                               | LiteralNull_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                              |
| LiteralReal             | LiteralRational                                                                                                                                                                                                                                                                                                                              | LiteralReal_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                              |
| LiteralSpecification    | LiteralExpression<br>LiteralExpression<br>FeatureTyping                                                                                                                                                                                                                                                                                      | LiteralSpecificationCommon_Mapping<br>LiteralSpecification_Mapping<br>LiteralSpecificationTyping_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | not src.type.ocllsUndefined()<br>and<br>not(src.type.ocllsKindOf(UML::Enumeration))<br>and<br>Helper.getSysMLv2EnumerationDefinition(src.type.ocllsKindOf(UML::Enumeration)) |
| LiteralString           | LiteralString                                                                                                                                                                                                                                                                                                                                | LiteralString_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                              |
| LiteralUnlimitedNatural | LiteralInfinity<br>LiteralInteger                                                                                                                                                                                                                                                                                                            | LiteralUnlimitedToUnbounded_Mapping<br>LiteralUnlimitedToInteger_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | (LiteralUnlimitedNatural.value >= 0)<br>and<br>(LiteralUnlimitedNatural.value < -1)                                                                                          |
| Observation             | FeatureMembership<br>ReferenceUsage<br>OwningMembership<br>Element<br>Documentation<br>SubjectMembership                                                                                                                                                                                                                                     | ElementFeatureMembership_Mapping<br>RequirementSubject_Mapping<br>RequirementDocumentationMembership_Mapping<br>NamedElementMain_Mapping<br>RequirementDocumentation_Mapping<br>RequirementSubjectMembership_Mapping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                                                                                                                              |
| OpaqueExpression        | OwningMembership<br>ReferenceUsage<br>FeatureMembership<br>ReturnParameterMembership<br>Feature<br>Feature<br>Membership<br>FeatureMembership<br>FeatureChainExpression<br>FeatureTyping<br>ParameterMembership<br>FeatureValue<br>CalculationUsage<br>FeatureValue<br>TextualRepresentation<br>FeatureReferenceExpression<br>ReferenceUsage | OpaqueExpressionMembership_Mapping<br>OpaqueExpressionReturnParameterReferenceUsage_Mapping<br>ObjectFlowTESTGaurdFeatureMembership_Mapping<br>OpaqueExpressionReturnParameterMembershipReferenceUsage_Mapping<br>OpaqueExpressionFeatureFeature_Mapping<br>OpaqueExpressionFeature_Mapping<br>OpaqueExpressionFeatureValueExpressionMembership_Mapping<br>OpaqueExpressionFeatureFeatureMembership_Mapping<br>OpaqueExpressionAsValue_Mapping<br>OpaqueExpressionReturnParameterReferenceUsageFeatureTyping_Mapping<br>OpaqueExpressionParameterMembership_Mapping<br>OpaqueExpressionFeatureValue_Mapping<br>OpaqueExpression_Mapping<br>PropertyDefaultValueOpaqueExpression_Mapping<br>OpaqueExpressionSpecification_Mapping<br>OpaqueExpressionFeatureValueExpression_Mapping<br>OpaqueExpressionReturnParameterReferenceUsageUntyped_Mapping | not src.type.ocllsUndefined()<br>and<br>not(src.type.ocllsKindOf(UML::Enumeration))<br>and<br>Helper.getSysMLv2EnumerationDefinition(src.type.ocllsKindOf(UML::Enumeration)) |



| SysML v1 Concept   | SysML v2 Concept            | Mapping Class                                            | Filter                           |
|--------------------|-----------------------------|----------------------------------------------------------|----------------------------------|
| StringExpression   | OperatorExpression          | Expression_Mapping                                       |                                  |
|                    | OwningMembership            | ExpressionElseMembership_Mapping                         |                                  |
|                    | TextualRepresentation       | ExpressionElseSpecification_Mapping                      |                                  |
|                    | FeatureTyping               | CommonParameterReferenceUsageInFeatureTyping_Mapping     |                                  |
|                    | FeatureTyping               | Mapping                                                  |                                  |
|                    | Element                     | CommonReturnParameterFeatureUntyped_Mapping              |                                  |
|                    | Feature                     | CommonReturnParameterFeatureTyping_Mapping               |                                  |
|                    | FeatureTyping               | ElementOwnership_Mapping                                 |                                  |
|                    | Relationship                | CommonValueSpecification_Mapping                         |                                  |
|                    | Expression                  | CommonParameterReferenceUsageInUntyped_Mapping           |                                  |
|                    | ReferenceUsage              | DefaultMultiplicityMembership_Mapping                    |                                  |
|                    | OwningMembership            | CommonReturnParameterFeatureMembership_Mapping           |                                  |
|                    | LiteralInteger              | CommonParameterReferenceUsageInMembership_Mapping        |                                  |
|                    | ReturnParameterMembership   | DefaultMultiplicityBoundOwnership_Mapping                |                                  |
|                    | ParameterMembership         | CommonReturnParameterReferenceUsageFeatureTyping_Mapping |                                  |
|                    | FeatureMembership           | DefaultUpperBound_Mapping                                |                                  |
|                    | FeatureTyping               | DefaultMultiplicityElement_Mapping                       |                                  |
|                    | LiteralInteger              | CommonReturnParameterReferenceUsageUntyped_Mapping       |                                  |
|                    | MultiplicityRange           | DefaultLowerBound_Mapping                                |                                  |
|                    | ReferenceUsage              | ElementMain_Mapping                                      |                                  |
|                    | LiteralInteger              | ElementMembership_Mapping                                |                                  |
|                    | Element                     | CommonReturnParameterReferenceUsageMembership_Mapping    |                                  |
|                    | Membership                  | EmptyReturnParameterFeatureMembership_Mapping            |                                  |
|                    | ReturnParameterMembership   | ReturnParameterMembership                                |                                  |
| TimeConstraint     | AssertConstraintUsage       | ConstraintUsage_Mapping                                  |                                  |
|                    | ConstraintDefinition        | Constraint_Mapping                                       |                                  |
|                    | FeatureTyping               | ConstraintUsageFeatureTyping_Mapping                     |                                  |
|                    | FeatureMembership           | ConstrainedElementFeatureMembership_Mapping              |                                  |
| TimeExpression     | TriggerInvocationExpression | TimeExpression_Mapping                                   |                                  |
| TimeInterval       | Expression                  | ValueSpecification_Mapping                               |                                  |
|                    | FeatureValue                | PropertyDefaultValue_Mapping                             |                                  |
|                    | FeatureValue                | SlotValue_Mapping                                        | src.owner.oclIsKindOf(UML::Slot) |
| TimeObservation    | FeatureMembership           | ElementFeatureMembership_Mapping                         |                                  |
|                    | ReferenceUsage              | RequirementSubject_Mapping                               |                                  |
|                    | OwningMembership            | RequirementDocumentationMembership_Mapping               |                                  |
|                    | Element                     | NamedElementMain_Mapping                                 |                                  |
|                    | Documentation               | RequirementDocumentation_Mapping                         |                                  |
|                    | SubjectMembership           | RequirementSubjectMembership_Mapping                     |                                  |
| ValueSpecification | Expression                  | ValueSpecification_Mapping                               |                                  |
|                    | FeatureValue                | PropertyDefaultValue_Mapping                             |                                  |
|                    | FeatureValue                | SlotValue_Mapping                                        | src.owner.oclIsKindOf(UML::Slot) |

### C.2.5.13.2 Mapping Specifications

#### C.2.5.13.2.1 CommonValueSpecification\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

## General Mappings

GenericToExpression\_Mapping

## Mapping Source

Element

## Mapping Target

Expression

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:

(none)

## Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion

false

- Feature::isReadOnly

false

- Expression::ownedRelationship

ElementOwnership\_Mapping.getMappedColl(from.ownedElement)->append(CommonReturnParameterFeatu

- Feature::direction

null

- Element::name

null

- Feature::isDerived

false

- Feature::isComposite

false

#### **C.2.5.13.2.2 EqualOperatorExpressionFeatureValue\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToFeatureValue\_Mapping

##### **Mapping Source**

TypedElement

##### **Mapping Target**

FeatureValue

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:

(none)

##### **Mapping rules**

The following lists the mapping rules for the target element properties.

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
  
null
- OwningMembership::ownedRelatedElement  
  
Set{self.ownedMemberElement() }
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- FeatureValue::value  
  
CommonFeatureReferenceExpression\_Mapping.getMapped(from)
- Element::ownedRelationship  
  
Set{}

#### **C.2.5.13.2.3 Expression\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToExpression\_Mapping  
NamedElementMain\_Mapping

##### **Mapping Source**

Expression

### Mapping Target

OperatorExpression

### Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:

(none)

### Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isEnd  
`false`
- Element::ownedRelationship  
`ElementOwnership_Mapping.getMappedColl(from.ownedElement)`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- OperatorExpression::operator  
`from.symbol`

- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::elementId  
Helper.getID(from)
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false

#### **C.2.5.13.2.4 ExpressionElse\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

Expression\_Mapping

##### **Mapping Source**

Expression

##### **Mapping Target**

OperatorExpression

##### **Owned Mappings**

- expressionElseMembership : ExpressionElseMembership\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:

```
from.symbol = 'else'
```

##### **Mapping rules**

The following lists the mapping rules for the target element properties.

- Feature::isEnd

- false
- Type::isSufficient
  - false
- Feature::isUnique
  - true
- Element::name
  - from.name
- Element::shortName
  - null
- Type::isAbstract
  - false
- Feature::isOrdered
  - false
- Element::aliasId
  - Set{}
- Feature::isPortion
  - false
- OperatorExpression::ownedRelationship
  - Set{expressionElseMembership.to}
- Feature::isReadOnly
  - false
- Feature::direction
  - null
- Element::elementId
  - Helper.getID(from)
- Feature::isDerived
  - false
- Feature::isComposite
  - false

### C.2.5.13.2.5 ExpressionElseMembership\_Mapping

#### Description

Creates the membership relationship for the textual representation for the else guard condition specification.

#### General Mappings

GenericToOwningMembership\_Mapping

#### Mapping Source

Expression

#### Mapping Target

OwningMembership

#### Owned Mappings

- expressionElseSpecification : ExpressionElseSpecification\_Mapping

#### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

#### Mapping rules

The following lists the mapping rules for the target element properties.

- Relationship::ownedRelatedElement  
`Set{ }`
- OwningMembership::ownedMemberElement  
`expressionElseSpecification.to`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Membership::memberElement  
*abstract rule*
- Element::shortName  
`null`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId



Set{}

- Membership::memberName

null

- Membership::visibility

KerML::VisibilityKind::public

- Element::name

null

- Element::ownedRelationship

Set{}

#### C.2.5.13.2.6 ExpressionElseSpecification\_Mapping

##### Description

Creates the textual representation for the else guard condition specification.

##### General Mappings

GenericToTextualRepresentation\_Mapping

##### Mapping Source

Expression

##### Mapping Target

TextualRepresentation

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:

(none)

##### Mapping rules

The following lists the mapping rules for the target element properties.

- Element::aliasId

Set{}

- Element::name

null

- TextualRepresentation::language

- 'SysMLv1'
- TextualRepresentation::body
  - 'else'
- Element::shortName
  - null
- AnnotatingElement::annotation
  - Set{}
- Element::elementId
  - Helper.createUUID()
- Element::ownedRelationship
  - Set{}

#### C.2.5.13.2.7 LiteralBoolean\_Mapping

##### Description

Maps the UML4SysML::LiteralBoolean to the SysMLv2::LiteralBoolean.

##### General Mappings

LiteralSpecificationCommon\_Mapping  
ElementMain\_Mapping

##### Mapping Source

LiteralBoolean

##### Mapping Target

LiteralBoolean

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isEnd
  - false

- **Type::isSufficient**  
false
- **Feature::isUnique**  
true
- **Element::shortName**  
null
- **Type::isAbstract**  
false
- **Element::elementId**  
Helper.createUUID()
- **Feature::isOrdered**  
false
- **Element::aliasId**  
Set{}
- **Feature::isPortion**  
false
- **Feature::isReadOnly**  
false
- **LiteralBoolean::ownedRelationship**  

```

let ownerships: Set(SYSML2::Relationship) = ElementOwnership_Mapping.getMappedColl(from.ownedRelationships->including(CommonReturnParameterFeatureMembership_Mapping.getMapped(from)) in
if from.type.ocIsUndefined() then
    ownerships
else
    ownerships->including(LiteralSpecificationTyping_Mapping.getMapped(from))
endif

```
- **LiteralBoolean::value**  
from.value
- **Feature::direction**  
null
- **Element::name**  
null
- **Feature::isDerived**

false

- Feature::isComposite

false

- Element::ownedRelationship

Set{}

#### **C.2.5.13.2.8 LiteralBooleanTrue\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

LiteralSpecificationCommon\_Mapping

##### **Mapping Source**

Element

##### **Mapping Target**

LiteralBoolean

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:

(none)

##### **Mapping rules**

The following lists the mapping rules for the target element properties.

- Feature::isEnd

false

- Type::isSufficient

false

- LiteralBoolean::value

true

- Feature::isUnique

true

- Element::shortName

- `Type::isAbstract`

- `Element::elementId`

- Feature::isOrdered

- `Element::aliasId`

- Feature::isPortion

- `Feature::isReadOnly`

- Feature::direction

- Element::name

- Feature::isDerived

- Feature::isComposite

- `Element::ownedRelationship`

#### C.2.5.13.2.9 LiteralInteger\_Mapping

### Description

Maps the UML4SysML::LiteralInteger to the SysMLv2::LiteralInteger.

## General Mappings

## LiteralSpecificationCommon\_Mapping

## ElementMain\_Mapping

### Mapping Source

LiteralInteger

### Mapping Target

LiteralInteger

### Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:

(none)

### Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isEnd

false

- LiteralInteger::ownedRelationship

```
let ownerships: Set(SYSML2::Relationship) = ElementOwnership_Mapping.getMappedColl(from.ownedRelationships->including(CommonReturnParameterFeatureMembership_Mapping.getMapped(from)) in
if from.type.ocIsUndefined() then
  ownerships
else
  ownerships->including(LiteralSpecificationTyping_Mapping.getMapped(from))
endif
```

- Type::isSufficient

false

- Feature::isUnique

true

- Element::shortName

null

- Type::isAbstract

false

- Element::elementId

Helper.createUUID()

- Feature::isOrdered

false

- Element::aliasId

- Set{}
- Feature::isPortion  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- LiteralInteger::value  
from.value
- Feature::isComposite  
false
- Element::ownedRelationship  
Set{}

#### **C.2.5.13.2.10 LiteralNull\_Mapping**

##### **Description**

Maps the UML4SysML::LiteralNull to the SysMLv2::LiteralNull.

##### **General Mappings**

LiteralSpecificationCommon\_Mapping  
ElementMain\_Mapping

##### **Mapping Source**

LiteralNull

##### **Mapping Target**

NullExpression

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

### Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isEnd

false

- Type::isSufficient

false

- Feature::isUnique

true

- Element::shortName

null

- Type::isAbstract

false

- NullExpression::ownedRelationship

```
let ownerships: Set(SYSML2::Relationship) = ElementOwnership_Mapping.getMappedColl(from.ownedRelationships)
->including(CommonReturnParameterFeatureMembership_Mapping.getMapped(from)) in
if from.type.oclIsUndefined() then
  ownerships
else
  ownerships->including(LiteralSpecificationTyping_Mapping.getMapped(from))
endif
```

- Element::elementId

Helper.createUUID()

- Feature::isOrdered

false

- Element::aliasId

Set{}

- Feature::isPortion

false

- Feature::isReadOnly

false

- Feature::direction



null

- Element::name

    null

- Feature::isDerived

    false

- Feature::isComposite

    false

- Element::ownedRelationship

    Set{}

### C.2.5.13.2.11 LiteralReal\_Mapping

#### Description

Maps the UML4SysML::LiteralReal to the SysMLv2::LiteralReal.

#### General Mappings

LiteralSpecificationCommon\_Mapping  
ElementMain\_Mapping

#### Mapping Source

LiteralReal

#### Mapping Target

LiteralRational

#### Owned Mappings

(none)

#### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

#### Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isEnd

    false

- Type::isSufficient

    false

- **LiteralRational::ownedRelationship**

```

    let ownerships: Set(SYSML2::Relationship) = ElementOwnership_Mapping.getMappedColl(from.ownedRelationships
        ->including(CommonReturnParameterFeatureMembership_Mapping.getMapped(from)) in
    if from.type.oclIsUndefined() then
        ownerships
    else
        ownerships->including(LiteralSpecificationTyping_Mapping.getMapped(from))
    endif

```

- **LiteralRational::value**

```

    from.value

```

- **Feature::isUnique**

```

    true

```

- **Element::shortName**

```

    null

```

- **Type::isAbstract**

```

    false

```

- **Element::elementId**

```

    Helper.createUUID()

```

- **Feature::isOrdered**

```

    false

```

- **Element::aliasId**

```

    Set{}

```

- **Feature::isPortion**

```

    false

```

- **Feature::isReadOnly**

```

    false

```

- **Feature::direction**

```

    null

```

- **Element::name**

```

    null

```

- **Feature::isDerived**

```

    false

```

- **Feature::isComposite**

false

- Element::ownedRelationship

Set{}

#### **C.2.5.13.2.12 LiteralSpecification\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

ElementMain\_Mapping

LiteralSpecificationCommon\_Mapping

##### **Mapping Source**

LiteralSpecification

##### **Mapping Target**

LiteralExpression

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:

(none)

##### **Mapping rules**

The following lists the mapping rules for the target element properties.

- Feature::isEnd

false

- Type::isSufficient

false

- Feature::isUnique

true

- Element::shortName

null

- Type::isAbstract

false

- Element::elementId  
Helper.createUUID()
- Feature::isOrdered  
false
- Element::aliasId  
Set{}
- Feature::isPortion  
false
- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false
- Element::ownedRelationship  
Set{}

#### **C.2.5.13.2.13 LiteralSpecificationCommon\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToExpression\_Mapping

##### **Mapping Source**

LiteralSpecification

##### **Mapping Target**

LiteralExpression

##### **Owned Mappings**

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:

(none)

### Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isEnd

false

- Type::isSufficient

false

- Feature::isUnique

true

- Element::shortName

null

- Type::isAbstract

false

- Element::elementId

Helper.createUUID()

- Feature::isOrdered

false

- Element::aliasId

Set{}

- Feature::isPortion

false

- LiteralExpression::ownedRelationship

```
let ownerships: Set(SYSML2::Relationship) = ElementOwnership_Mapping.getMappedColl(from.ownedRelationships)
->including(CommonReturnParameterFeatureMembership_Mapping.getMapped(from)) in
if from.type.ocIsUndefined() then
  ownerships
else
  ownerships->including(LiteralSpecificationTyping_Mapping.getMapped(from))
endif
```

- Feature::isReadOnly

false

- Feature::direction

null

- Element::name

null

- Feature::isDerived

false

- Feature::isComposite

false

#### **C.2.5.13.2.14 LiteralSpecificationTyping\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

TypedElementToFeatureTyping\_Mapping

##### **Mapping Source**

LiteralSpecification

##### **Mapping Target**

FeatureTyping

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

The following lists the mapping rules for the target element properties.

- Element::aliasId

Set{ }

- Relationship::ownedRelatedElement

Set{ }

- FeatureTyping::typedFeature

```

    from
    • Element::name
      null
    • Element::shortName
      null
    • Element::elementId
      Helper.createUUID()
    • FeatureTyping::type
      abstract rule
    • FeatureTyping::typedFeature
      abstract rule
    • Element::ownedRelationship
      Set{}

```

#### C.2.5.13.2.15 LiteralString\_Mapping

##### Description

Maps the UML4SysML::LiteralString to the SysMLv2::LiteralString.

##### General Mappings

LiteralSpecificationCommon\_Mapping  
ElementMain\_Mapping

##### Mapping Source

LiteralString

##### Mapping Target

LiteralString

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

The following lists the mapping rules for the target element properties.

```

    • Feature::isEnd
      false

```

- **Type::isSufficient**  
false
- **Feature::isUnique**  
true
- **Element::shortName**  
null
- **Type::isAbstract**  
false
- **Element::elementId**  
Helper.createUUID()
- **Feature::isOrdered**  
false
- **Element::aliasId**  
Set{}
- **Feature::isPortion**  
false
- **Feature::isReadOnly**  
false
- **Feature::direction**  
null
- **Element::name**  
null
- **LiteralString::ownedRelationship**  

```

let ownerships: Set(SYSML2::Relationship) = ElementOwnership_Mapping.getMappedColl(from.ownedRelationships
->including(CommonReturnParameterFeatureMembership_Mapping.getMapped(from)) in
if from.type.ocIsUndefined() then
    ownerships
else
    ownerships->including(LiteralSpecificationTyping_Mapping.getMapped(from))
endif

```
- **Feature::isDerived**  
false
- **Feature::isComposite**



```
false
```

- LiteralString::value

```
if from.value.oclIsUndefined() then '' else from.value endif
```

- Element::ownedRelationship

```
Set{}
```

### C.2.5.13.2.16 LiteralUnlimitedToUnbounded\_Mapping

#### Description

Maps the UML4SysML::LiteralUnlimited to the SysMLv2::LiteralInfinity if it is the unlimited value.

#### General Mappings

LiteralSpecificationCommon\_Mapping

ElementMain\_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 following lists the mapping rules for the target element properties.

- Feature::isEnd

```
false
```

- Type::isSufficient

```
false
```

- LiteralInfinity::ownedRelationship

```
let ownerships: Set(SysML2::Relationship) = ElementOwnership_Mapping.getMappedColl(from.ownedRelationships)
->including(CommonReturnParameterFeatureMembership_Mapping.getMapped(from)) in
if from.type.oclIsUndefined() then
  ownerships
else
```

```

        ownerships->including(LiteralSpecificationTyping_Mapping.getMapped(from))
    endif

```

- Feature::isUnique

```

        true

```
- Element::shortName

```

        null

```
- Type::isAbstract

```

        false

```
- Element::elementId

```

        Helper.createUUID()

```
- Feature::isOrdered

```

        false

```
- Element::aliasId

```

        Set{}

```
- Feature::isPortion

```

        false

```
- Feature::isReadOnly

```

        false

```
- Feature::direction

```

        null

```
- Element::name

```

        null

```
- Feature::isDerived

```

        false

```
- Feature::isComposite

```

        false

```
- Element::ownedRelationship

```

        Set{}

```

#### **C.2.5.13.2.17 LiteralUnlimitedToInteger\_Mapping**

##### **Description**

Maps the UML4SysML::LiteralUnlimited to the SysMLv2::LiteralInteger if it is not the unlimited value.

## General Mappings

LiteralSpecificationCommon\_Mapping  
ElementMain\_Mapping

## Mapping Source

LiteralUnlimitedNatural

## Mapping Target

LiteralInteger

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:

```
(from.value <> -1)
```

## Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isEnd

```
false
```

- Type::isSufficient

```
false
```

- LiteralInteger::value

```
from.value
```

- LiteralInteger::ownedRelationship

```
let ownerships: Set(SYSML2::Relationship) = ElementOwnership_Mapping.getMappedColl(from.ownedRelationships)
->including(CommonReturnParameterFeatureMembership_Mapping.getMapped(from)) in
if from.type.ocIsUndefined() then
  ownerships
else
  ownerships->including(LiteralSpecificationTyping_Mapping.getMapped(from))
endif
```

- Feature::isUnique

```
true
```

- Element::shortName

```
null
```

- Type::isAbstract

- false
- Element::elementId
  - Helper.createUUID()
- Feature::isOrdered
  - false
- Element::aliasId
  - Set{}
- Feature::isPortion
  - false
- Feature::isReadOnly
  - false
- Feature::direction
  - null
- Element::name
  - null
- Feature::isDerived
  - false
- Feature::isComposite
  - false
- Element::ownedRelationship
  - Set{}

#### **C.2.5.13.2.18 OpaqueExpressionAsValue\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

CommonValueSpecification\_Mapping

##### **Mapping Source**

OpaqueExpression

##### **Mapping Target**

FeatureChainExpression

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:

(none)

## Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isEnd

false

- Type::isSufficient

false

- Feature::isUnique

true

- Element::shortName

null

- Type::isAbstract

false

- Element::elementId

Helper.createUUID()

- Feature::isOrdered

false

- FeatureChainExpression::ownedRelationship

Set{OpaqueExpressionParameterMembership\_Mapping.getMapped(from), CommonReturnParameterFeatur

- Element::aliasId

Set{}

- Feature::isPortion

false

- Feature::isReadOnly

false

- Feature::direction

null

- Element::name

null

- Feature::isDerived

false

- Feature::isComposite

false

- Element::ownedRelationship

Set{}

#### C.2.5.13.2.19 OpaqueExpression\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

CommonAction\_Mapping

ValueSpecification\_Mapping

##### Mapping Source

OpaqueExpression

##### Mapping Target

CalculationUsage

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:

(none)

##### Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isEnd

false

- CalculationUsage::ownedRelationship

Set{OpaqueExpressionMembership\_Mapping.getMapped(from), OpaqueExpressionReturnParameterMembe

- Element::ownedRelationship  
 ElementOwnership\_Mapping.getMappedColl(from.ownedElement)
- Type::isSufficient  
 false
- Feature::isUnique  
 true
- Element::name  
 from.name
- Element::shortName  
 null
- Type::isAbstract  
 false
- Feature::isOrdered  
 false
- Element::aliasId  
 Set{ }
- Feature::isPortion  
 false
- Usage::isVariation  
 false
- Feature::isReadOnly  
 false
- Expression::ownedRelationship  
 ElementOwnership\_Mapping.getMappedColl(from.ownedElement)->append(CommonReturnParameterFeatu
- Feature::direction  
 null
- Element::elementId  
 Helper.getID(from)
- Feature::isDerived  
 false

- ActionUsage::isComposite

true

#### C.2.5.13.2.20 OpaqueExpressionFeature\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeature\_Mapping

##### Mapping Source

OpaqueExpression

##### Mapping Target

Feature

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:

(none)

##### Mapping rules

The following lists the mapping rules for the target element properties.

- Element::aliasId

Set{ }

- Type::isSufficient

false

- Element::name

null

- Element::shortName

null

- Type::isAbstract

false

- Element::elementId

Helper.createUUID()



- Feature::ownedRelationship

```
Set{OpaqueExpressionFeatureValue_Mapping.getMapped(from), OpaqueExpressionFeatureFeatureMemk
```

#### C.2.5.13.2.21 OpaqueExpressionFeatureFeature\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeature\_Mapping

##### Mapping Source

OpaqueExpression

##### Mapping Target

Feature

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:

(none)

##### Mapping rules

The following lists the mapping rules for the target element properties.

- Element::aliasId

```
Set{}
```

- Type::isSufficient

```
false
```

- Element::name

```
null
```

- Element::shortName

```
null
```

- Type::isAbstract

```
false
```

- Element::elementId

```
Helper.createUUID()
```

- Element::ownedRelationship

Set { }

#### C.2.5.13.2.22 OpaqueExpressionFeatureFeatureMembership\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureMembership\_Mapping

##### Mapping Source

OpaqueExpression

##### Mapping Target

FeatureMembership

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:

(none)

##### Mapping rules

The following lists the mapping rules for the target element properties.

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
null
- FeatureMembership::ownedMemberFeature  
OpaqueExpressionFeatureFeature\_Mapping.getMapped(from)
- Element::shortName  
null
- Element::elementId  
Helper.createUUID()
- Element::aliasId  
Set { }

- OwingMembership::ownedMemberElement  
*abstract rule*
- Membership::memberName  
  
null
- OwingMembership::ownedRelatedElement  
  
Set { self.ownedMemberElement () }
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- Element::ownedRelationship  
  
Set { }

#### C.2.5.13.2.23 OpaqueExpressionFeatureValue\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToFeatureValue\_Mapping

##### Mapping Source

OpaqueExpression

##### Mapping Target

FeatureValue

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

The following lists the mapping rules for the target element properties.

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
  
null
- Element::shortName  
  
null
- Element::elementId  
  
Helper.createUUID()
- Element::aliasId  
  
Set{}
- OwningMembership::ownedMemberElement  
*abstract rule*
- FeatureValue::value  
  
OpaqueExpressionFeatureValueExpression\_Mapping.getMapped(from)
- Membership::memberName  
  
null
- OwningMembership::ownedRelatedElement  
  
Set{self.ownedMemberElement() }
- Membership::visibility  
  
KerML::VisibilityKind::public
- Element::name  
  
null
- Element::ownedRelationship  
  
Set{}

#### **C.2.5.13.2.24 OpaqueExpressionFeatureValueExpression\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToExpression\_Mapping

##### **Mapping Source**

OpaqueExpression

### Mapping Target

FeatureReferenceExpression

### Owned Mappings

(none)

### Applicable filters

This mapping applies only if the following (OCL) condition is verified:

(none)

### Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isEnd

false

- Type::isSufficient

false

- Feature::isUnique

true

- Element::shortName

null

- Type::isAbstract

false

- Element::elementId

Helper.createUUID()

- Feature::isOrdered

false

- Element::aliasId

Set{}

- Feature::isPortion

false

- FeatureReferenceExpression::ownedRelationship

Set{OpaqueExpressionFeatureValueExpressionMembership\_Mapping.getMapped(from), EmptyReturnPar

- Feature::isReadOnly  
false
- Feature::direction  
null
- Element::name  
null
- Feature::isDerived  
false
- Feature::isComposite  
false

#### **C.2.5.13.2.25 OpaqueExpressionFeatureValueExpressionMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToMembership\_Mapping

##### **Mapping Source**

OpaqueExpression

##### **Mapping Target**

Membership

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

The following lists the mapping rules for the target element properties.

- Element::aliasId  
Set { }
- Relationship::ownedRelatedElement  
Set { }

- Relationship::source  
Set{}
- Membership::memberElement  
from
- Element::name  
null
- Element::shortName  
null
- Element::elementId  
Helper.createUUID()
- Relationship::target  
Set{}
- Element::ownedRelationship  
Set{}

#### **C.2.5.13.2.26 OpaqueExpressionMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToOwningMembership\_Mapping

##### **Mapping Source**

OpaqueExpression

##### **Mapping Target**

OwningMembership

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

The following lists the mapping rules for the target element properties.

- Relationship::ownedRelatedElement  
`Set{}`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Membership::memberElement  
*abstract rule*
- Element::shortName  
`null`
- OwningMembership::ownedMemberElement  
`OpaqueExpressionSpecification_Mapping.getMapped(from)`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{}`
- Membership::memberName  
`null`
- Membership::visibility  
`KerML::VisibilityKind::public`
- Element::name  
`null`
- Element::ownedRelationship  
`Set{}`

#### **C.2.5.13.2.27 OpaqueExpressionParameterMembership\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

GenericToParameterMembership\_Mapping

##### **Mapping Source**



OpaqueExpression

## Mapping Target

ParameterMembership

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:

(none)

## Mapping rules

The following lists the mapping rules for the target element properties.

- FeatureMembership::ownedRelatedElement  
`Set{self.ownedMemberFeature() }`
- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
`null`
- Element::shortName  
`null`
- ParameterMembership::ownedMemberParameter  
`OpaqueExpressionFeature_Mapping.getMapped(from)`
- Element::elementId  
`Helper.createUUID()`
- Element::aliasId  
`Set{ }`
- FeatureMembership::ownedMemberFeature  
*abstract rule*
- FeatureMembership::owningType  
*abstract rule*
- Membership::memberName  
`null`
- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*

- Membership::visibility  
KerML::VisibilityKind::public
- Element::name  
null
- Element::ownedRelationship  
Set{ }

#### C.2.5.13.2.28 OpaqueExpressionReturnParameterMembershipReferenceUsage\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToReturnParameterMembership\_Mapping

##### Mapping Source

OpaqueExpression

##### Mapping Target

ReturnParameterMembership

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

The following lists the mapping rules for the target element properties.

- Membership::membershipOwningNamespace  
*abstract rule*
- Membership::memberShortName  
null
- Element::shortName  
null
- Element::elementId  
Helper.createUUID()

- Element::aliasId

Set{}

- FeatureMembership::owningType  
*abstract rule*
- ReturnParameterMembership::ownedMemberParameter

if from.type.oclIsUndefined() then OpaqueExpressionReturnParameterReferenceUsageUntyped\_Mapping

- Membership::memberName

null

- ParameterMembership::ownedRelatedElement

Set{self.ownedMemberParameter() }

- TypeFeaturing::featureOfType  
*abstract rule*
- TypeFeaturing::featuringType  
*abstract rule*
- Membership::visibility

KerML::VisibilityKind::public

- Element::name

null

- Element::ownedRelationship

Set{}

#### C.2.5.13.2.29 OpaqueExpressionReturnParameterReferenceUsage\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToReferenceUsage\_Mapping

##### Mapping Source

OpaqueExpression

##### Mapping Target

ReferenceUsage

##### Owned Mappings

- opaqueExpressionReturnParameterReferenceUsageFeatureTyping :  
OpaqueExpressionReturnParameterReferenceUsageFeatureTyping\_Mapping

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

## Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::direction  
`KerML::FeatureDirectionKind::_'out'`
- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Element::elementId  
`Helper.createUUID()`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Usage::isVariation  
`false`
- Feature::isReadOnly  
`false`
- ReferenceUsage::ownedRelationship

```
Set{opaqueExpressionReturnParameterReferenceUsageFeatureTyping.to}
```

- Element::name

```
null
```

- Feature::isDerived

```
false
```

- Feature::isComposite

```
false
```

#### **C.2.5.13.2.30 OpaqueExpressionReturnParameterReferenceUsageFeatureTyping\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

TypedElementToFeatureTyping\_Mapping

##### **Mapping Source**

OpaqueExpression

##### **Mapping Target**

FeatureTyping

##### **Owned Mappings**

- opaqueExpressionReturnParameterReferenceUsage :  
OpaqueExpressionReturnParameterReferenceUsage\_Mapping

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### **Mapping rules**

The following lists the mapping rules for the target element properties.

- Element::aliasId

```
Set{ }
```

- Relationship::ownedRelatedElement

```
Set{ }
```

- FeatureTyping::typedFeature

```
opaqueExpressionReturnParameterReferenceUsage.to
```

- Element::name  
null
- Element::shortName  
null
- Element::elementId  
Helper.createUUID()
- FeatureTyping::type  
*abstract rule*
- FeatureTyping::typedFeature  
*abstract rule*
- Element::ownedRelationship  
Set{ }

#### C.2.5.13.2.31 OpaqueExpressionReturnParameterReferenceUsageUntyped\_Mapping

##### Description

\*\*\* not specified yet \*\*\*

##### General Mappings

GenericToReferenceUsage\_Mapping

##### Mapping Source

OpaqueExpression

##### Mapping Target

ReferenceUsage

##### Owned Mappings

(none)

##### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

##### Mapping rules

The following lists the mapping rules for the target element properties.

- ReferenceUsage::direction  
KerML::FeatureDirectionKind::\_out'
- Feature::isEnd

- false
- Type::isSufficient
  - false
- Feature::isUnique
  - true
- Element::shortName
  - null
- Type::isAbstract
  - false
- Element::elementId
  - Helper.createUUID()
- Feature::isOrdered
  - false
- Element::aliasId
  - Set{}
- Feature::isPortion
  - false
- Usage::isVariation
  - false
- Feature::isReadOnly
  - false
- Element::name
  - null
- Feature::isDerived
  - false
- Feature::isComposite
  - false
- Element::ownedRelationship
  - Set{}

### C.2.5.13.2.32 OpaqueExpressionSpecification\_Mapping

#### Description

\*\*\* not specified yet \*\*\*

#### General Mappings

GenericToTextualRepresentation\_Mapping

#### Mapping Source

OpaqueExpression

#### Mapping Target

TextualRepresentation

#### Owned Mappings

(none)

#### Applicable filters

This mapping applies only if the following (OCL) condition is verified:  
(none)

#### Mapping rules

The following lists the mapping rules for the target element properties.

- Element::aliasId  
`Set{}`
- TextualRepresentation::body  
`if from.body->size() = 0 then OclUndefined else from.body.get(0) endif`
- Element::name  
`null`
- TextualRepresentation::language  
`if from.language->size() = 0 then OclUndefined else from.language.get(0) endif`
- Element::shortName  
`null`
- AnnotatingElement::annotation  
`Set{}`
- Element::elementId  
`Helper.createUUID()`



- Element::ownedRelationship

Set { }

#### **C.2.5.13.2.33 TimeExpression\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

ValueSpecification\_Mapping

##### **Mapping Source**

TimeExpression

##### **Mapping Target**

TriggerInvocationExpression

##### **Owned Mappings**

(none)

##### **Applicable filters**

This mapping applies only if the following (OCL) condition is verified:

(none)

##### **Mapping rules**

The following lists the mapping rules for the target element properties.

- Feature::isEnd

false

- Element::ownedRelationship

ElementOwnership\_Mapping.getMappedColl(from.ownedElement)

- Type::isSufficient

false

- Feature::isUnique

true

- Element::name

from.name

- Element::shortName

null

- `Type::isAbstract`  
`false`
- `Feature::isOrdered`  
`false`
- `Element::aliasId`  
`Set{ }`
- `Feature::isPortion`  
`false`
- `TriggerInvocationExpression::kind`  
`SysMLv2::TriggerKind::at`
- `Feature::isReadOnly`  
`false`
- `Expression::ownedRelationship`  
`ElementOwnership_Mapping.getMappedColl(from.ownedElement)->append(CommonReturnParameterFeatu`
- `Feature::direction`  
`null`
- `Element::elementId`  
`Helper.getID(from)`
- `Feature::isDerived`  
`false`
- `Feature::isComposite`  
`false`

#### **C.2.5.13.2.34 ValueSpecification\_Mapping**

##### **Description**

\*\*\* not specified yet \*\*\*

##### **General Mappings**

`CommonValueSpecification_Mapping`  
`NamedElementMain_Mapping`

##### **Mapping Source**

`ValueSpecification`

## Mapping Target

Expression

## Owned Mappings

(none)

## Applicable filters

This mapping applies only if the following (OCL) condition is verified:

(none)

## Mapping rules

The following lists the mapping rules for the target element properties.

- Feature::isEnd  
`false`
- Type::isSufficient  
`false`
- Feature::isUnique  
`true`
- Element::shortName  
`null`
- Type::isAbstract  
`false`
- Feature::isOrdered  
`false`
- Element::aliasId  
`Set{}`
- Feature::isPortion  
`false`
- Feature::isReadOnly  
`false`
- Feature::direction  
`null`
- Element::elementId

```
Helper.getID(from)
```

- Element::name

```
null
```

- Feature::isDerived

```
false
```

- Expression::ownedRelationship

```
if from.type.oclIsUndefined() then
    Set{CommonReturnParameterFeatureMembership_Mapping.getMapped(from)}
else
    Set{LiteralSpecificationTyping_Mapping.getMapped(from), CommonReturnParameterFeatureMembe
endif
```

- Feature::isComposite

```
false
```