



Seventh Framework Programme

CallFP7-ICT-2013-10

Project Acronym: **S-CASE**

Grant Agreement N°: **610717**

Project Type: **COLLABORATIVE PROJECT**

Project Full Title: **Scaffolding Scalable Software Services**

D2.3.2 Ontology to MOF Models Transformation Mechanism	
Nature:	R
Dissemination Level:	PU
Version #:	1.0
Date:	30 June 2015
WP number and Title:	WP 2 Ontology to MOF Models Transformation Mechanism
Deliverable Leader:	AUTH
Author(s):	Christoforos Zolotas (AUTH), Themistoklis Diamantopoulos (AUTH)
Reviewers:	Kate Byrne (UEDIN), Marin Orlic (ENT)
Status:	Submitted

Document History

Version ¹	Issue Date	Status ²	Content and changes
0.1	5 May 2015	Draft	Initial Template Document
0.2	2 June 2015	Draft	Defined extended structure of all document chapters
0.3	5 June 2015	Draft	Completed CIM definition in chapter 4
0.4	9 June 2015	Draft	Completed PIM definition in chapter 5
0.5	11 June 2015	Draft	Completed PSM definition in chapter 6
0.6	16 June 2015	Draft	Completed CIM To PIM transformation definition in chapter 8
0.7	19 June 2015	Draft	Completed PIM To PSM transformation definition in chapter 9
0.8	23 June 2015	Draft	Added chapter 7 – ontology to Yaml mapping
0.9	25 June 2015	Draft	Added introductory chapters 1, 2 and 3
1.0	30 June 2015	Final	Added UEDIN comments/Sent to Commission

Peer Review History³

Version	Peer Review Date	Reviewed By
0.9	25 June 2015	Kate Byrne (UEDIN)

{Comment: every text surrounded by either '<' and '>', or curly parentheses like the present are prescriptions and guidance, respectively, for the reader – that is, the author of a deliverable in progress. All such text should be deleted during the process of writing the deliverable while replaced by the relevant text.}



This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 610717.

¹ Please use a new number for each new version of the deliverable. Use "0.#" for Draft and Peer-Reviewed. "x.#" for Submitted and Approved", where x>=1. Add the date when this version was issued and list the items that have been added or changed.

² A deliverable can be in one of these stages: Draft, Peer-Reviewed, Submitted and Approved.

³ Only for deliverables that have to be peer-reviewed

Table of contents

Document History.....	2
Table of contents	3
List of Figures.....	13
List of Tables.....	16
Abbreviations and Acronyms	27
Executive Summary	28
1 Introduction	29
1.1 WP2 - Task 2.3 Description and Objectives	29
1.2 Deliverable Structure.....	30
1.3 Intended Audience	30
2 Related work	32
2.1 REST Architecture.....	32
2.2 Ruby on Rails.....	32
2.3 EMF-Rest.....	33
2.4 Django – REST-framework	35
2.5 Other Evolving Frameworks.....	36
2.6 S-CASE MDE Engine Beyond the State of the Art.....	36
3 S-CASE MDE Engine Introduction.....	38
3.1 MDE Process Overview	38
3.2 Associated Technologies and Frameworks.....	39
3.2.1 Ecore meta-model	39
3.2.2 Object Constraint Language (OCL).....	40
3.2.3 Eclipse Modelling Framework (EMF)	40
3.2.4 Atlas Transformation Language (ATL).....	41
3.2.5 Acceleo code generation framework.....	42
3.3 S-CASE 2D MDE Engine Architecture	42
4 Computational Independent Model UML Profile	47
4.1 Core CIM UML Profile	47
4.1.1 CIM UML Profile Design Goals	47
4.1.2 CIM Ecore Meta-model Definition.....	48
4.1.2.1 Introduction	48
4.1.2.2 CIM Ecore Meta-model Elements	48
4.1.2.2.1 CIM Custom Data Types Definition.....	48
4.1.2.2.2 RESTfulServiceCIM Element.....	49
4.1.2.2.3 Resource Element.....	51
4.1.2.2.4 CRUActivity Element	53
4.1.2.2.5 Property Element.....	53
4.1.2.2.6 InputRepresentation Element	54
4.1.2.2.7 OutputRepresentation Element	54
4.2 CIM UML Profile Extensions	55
4.2.1 CIM UML Profile Extensions Goals.....	55
4.2.1.1 CIM Authentication Extension Goals	55
4.2.1.2 CIM Database Searching Extension Goals.....	56
4.2.1.3 CIM External Service Composition Wrapping Extension Goals	56
4.2.1.4 CIM ABAC Authorization Extension Goals.....	57

4.2.2 CIM Authentication Ecore Meta-model Definition.....	57
4.2.2.1 Introduction	57
4.2.2.2 CIM Authentication Ecore Meta-model Elements	58
4.2.2.2.1 AnnotationModel Element	58
4.2.2.2.2 AnnotatedElement Element	59
4.2.2.2.3 Annotation Element	60
4.2.2.2.4 AnnCRUDActivity Element.....	60
4.2.2.2.5 AnnProperty Element	61
4.2.2.2.6 AnnResource Element	62
4.2.2.2.7 AuthenticationModel Element	62
4.2.2.2.8 AuthenticationToken Element.....	63
4.2.2.2.9 Password Element	64
4.2.2.2.10 AuthenticationMode Element.....	64
4.2.2.2.11 GuestMode Element	65
4.2.2.2.12 BothMode Element	66
4.2.2.2.13 AuthenticationOnlyMode Element	66
4.2.3 CIM Database Searching Ecore Meta-model Definition	67
4.2.3.1 Introduction	67
4.2.3.2 CIM Database Searching Ecore Meta-model Elements	67
4.2.3.2.1 AnnotationModel Element	67
4.2.3.2.2 Annotation Element	68
4.2.3.2.3 AnnProperty Element	69
4.2.3.2.4 AnnAlgoResource Element	69
4.2.3.2.5 AnnCRUDResource Element	70
4.2.3.2.6 SearchResource Element.....	71
4.2.3.2.7 SearchableResource Element	71
4.2.3.2.8 SearchableProperty Element.....	72
4.2.4 CIM External Service Composition Ecore Meta-model Definition.....	73
4.2.4.1 Introduction	73
4.2.4.2 CIM External Service Composition Ecore Meta-model Elements	74
4.2.4.2.1 AnnotationModel Element	74
4.2.4.2.2 AnnotatedElement Element	74
4.2.4.2.3 Annotation Element	75
4.2.4.2.4 AnnCRUDResource Element	75
4.2.4.2.5 AnnAlgoResource Element	76
4.2.4.2.6 RESTClientResource Element.....	76
4.2.4.2.7 TargetRESTService Element.....	78
4.2.4.2.8 QueryParam Element	79
4.2.4.2.9 InputDataModel Element	80
4.2.4.2.10 OutputDataModel Element.....	80
4.2.4.2.11 Representation Element.....	82
4.2.4.2.12 Property Element	82
4.2.4.2.13 NonPersistentOutput Element.....	83
4.2.4.2.14 AutoPersistentOutput Element	83
4.2.4.2.15 ExistentsCRUDPersistentOutput Element.....	84
4.2.5 CIM ABAC Authorization Ecore Meta-model Definition	85
4.2.5.1 Introduction	85
4.2.5.2 CIM ABAC Authorization Ecore Meta-model Elements	85
4.2.5.2.1 AnnotationModel Element	85
4.2.5.2.2 AnnotatedElement Element	86
4.2.5.2.3 Annotation Element	86
4.2.5.2.4 AnnCRUDActivity Element	87
4.2.5.2.5 AnnProperty Element	87
4.2.5.2.6 AnnResource Element	88
4.2.5.2.7 AuthorizationSubject Element.....	88
4.2.5.2.8 SubjectAttribute Element	89

4.2.5.2.9	AuthorizableResource Element	90
4.2.5.2.10	ResourceAccessPolicySet Element	91
4.2.5.2.11	ResourceAccessPolicy Element	92
4.2.5.2.12	RuleCombiningAlgorithm Element	94
4.2.5.2.13	DenyOverridesAlgorithm Element	94
4.2.5.2.14	PermitOverridesAlgorithm Element	95
4.2.5.2.15	ResourceAccessRule Element	95
4.2.5.2.16	ResourceAccessPermitRule Element	96
4.2.5.2.17	ResourceAccessDenyRule Element	97
4.2.5.2.18	MatchedSubjectAttribute Element	97
4.2.5.2.19	AllowedAction Element	99
4.2.5.2.20	MatchedResourceAttribute Element	100
4.2.5.2.21	MatchedContextAttribute Element	101
4.2.5.2.22	ContextResource Element	102
5	Platform Independent Model UML Profile Extensions	104
5.1	PIM UML Profile Extension Goals	104
5.2	PIM Authentication Ecore Meta-model Definition	104
5.2.1.1	Introduction	104
5.2.1.2	PIM Authentication Ecore Meta-model Elements	104
5.2.1.2.1	AnnotationModel Element	104
5.2.1.2.2	AnnotatedElement Element	106
5.2.1.2.3	Annotation Element	106
5.2.1.2.4	AnnCRUDActivityHandler Element	106
5.2.1.2.5	AnnCRUDActivity Element	107
5.2.1.2.6	AnnDatabaseController Element	108
5.2.1.2.7	AuthenticationPerformer Element	108
5.2.1.2.8	AuthenticationToken Element	110
5.2.1.2.9	Password Element	111
5.2.1.2.10	AuthenticationMode Element	111
5.2.1.2.11	GuestMode Element	113
5.2.1.2.12	AuthenticationOnlyMode Element	113
5.2.1.2.13	BothMode Element	114
5.3	PIM Database Searching Ecore Meta-model Definition	114
5.3.1.1	Introduction	114
5.3.1.2	PIM Database Searching Ecore Meta-model Elements	115
5.3.1.2.1	AnnotationModel Element	115
5.3.1.2.2	AnnotatedElement Element	116
5.3.1.2.3	Annotation Element	116
5.3.1.2.4	AnnPIMComponentProperty Element	117
5.3.1.2.5	AnnResourceModel Element	117
5.3.1.2.6	AnnCRUDActivityHandler Element	118
5.3.1.2.7	AnnCRUDActivity Element	119
5.3.1.2.8	AnnAlgoResourceController Element	119
5.3.1.2.9	SearchController Element	120
5.3.1.2.10	SearchCRUDActivity Element	121
5.3.1.2.11	SearchCRUDActivityHandler Element	122
5.3.1.2.12	SearchableResourceModel Element	123
5.3.1.2.13	SearchableProperty Element	124
5.4	PIM External Service Composition Ecore Meta-model Definition	126
5.4.1.1	Introduction	126
5.4.1.2	PIM External Service Composition Ecore Meta-model Elements	126
5.4.1.2.1	AnnotationModel Element	126
5.4.1.2.2	AnnotatedElement Element	127
5.4.1.2.3	Annotation Element	128
5.4.1.2.4	AnnAlgoResourceModel Element	128
5.4.1.2.5	AnnResourceModel Element	129

5.4.1.2.6	AnnCRUDActivityHandler Element	129
5.4.1.2.7	AnnCRUDActivity Element	130
5.4.1.2.8	AnnAlgoResourceController Element	130
5.4.1.2.9	RESTClientController Element	131
5.4.1.2.10	RESTClientCRUDActivity Element	132
5.4.1.2.11	RESTClientCRUDActivityHandler Element	134
5.4.1.2.12	QueryParam Element	135
5.4.1.2.13	RESTClientModel Element	136
5.4.1.2.14	InputDataModel Element	137
5.4.1.2.15	OutputDataModel Element	138
5.4.1.2.16	Property Element	139
5.4.1.2.17	Representation Element	140
5.4.1.2.18	NonPersistentOutput Element	141
5.4.1.2.19	AutoPersistentOutput Element	141
5.4.1.2.20	ExistentCRUDPersistentOutput Element	142
5.5	PIM ABAC Authorization Ecore Meta-model Definition.....	143
5.5.1.1	Introduction	143
5.5.1.2	PIM ABAC Authorization Ecore Meta-model Elements	143
5.5.1.2.1	AnnotationModel Element	143
5.5.1.2.2	AnnotatedElement Element	144
5.5.1.2.3	Annotation Element	144
5.5.1.2.4	AnnResourceControllerCRUDActivity Element	145
5.5.1.2.5	AnnPIMComponentProperty Element	145
5.5.1.2.6	AnnDatabaseController Element	146
5.5.1.2.7	AnnCRUDActivityHandler Element	147
5.5.1.2.8	AnnResourceModelManager Element	147
5.5.1.2.9	AnnAlgoResourceModel Element	148
5.5.1.2.10	AnnResourceModel Element	148
5.5.1.2.11	AuthorizationSubject Element	149
5.5.1.2.12	SubjectAttribute Element	150
5.5.1.2.13	AuthorizableResource Element	151
5.5.1.2.14	ResourceAccessPolicySet Element	153
5.5.1.2.15	RuleCombiningAlgorithm Element	154
5.5.1.2.16	DenyOverridesAlgorithm Element	155
5.5.1.2.17	PermitOverridesAlgorithm Element	155
5.5.1.2.18	ResourceAccessPolicy Element	156
5.5.1.2.19	ResourceAccessRule Element	157
5.5.1.2.20	ResourceAccessDenyRule Element	159
5.5.1.2.21	ResourceAccessPermitRule Element	159
5.5.1.2.22	AllowedAction Element	160
5.5.1.2.23	MatchedResourceAttribute Element	161
5.5.1.2.24	MatchedContextAttribute Element	162
5.5.1.2.25	MatchedSubjectAttribute Element	164
5.5.1.2.26	AuthorizationPerformer Element	166
5.5.1.2.27	AuthorizationPolicyEvaluator Element	167
5.5.1.2.28	AuthorizationDataHandler Element	167
5.5.1.2.29	AuthorizationDataTable Element	169
6	Platform Specific Model UML Profile Extensions.....	171
6.1	PSM UML Profile Extension Goals	171
6.2	PSM Authentication Ecore Meta-model Definition	171
6.2.1.1	Introduction	171
6.2.1.2	PSM Authentication Ecore Meta-model Elements	171
6.2.1.2.1	AnnotationModel Element	171
6.2.1.2.2	AnnotatedElement Element	173
6.2.1.2.3	Annotation Element	173
6.2.1.2.4	AnnHTTPActivity Element	173

6.2.1.2.5	AnnHTTPActivityHandler Element	174
6.2.1.2.6	AnnJPAController Element	175
6.2.1.2.7	AuthenticationPerformer Element	175
6.2.1.2.8	AuthenticationToken Element	177
6.2.1.2.9	Password Element	177
6.2.1.2.10	AuthenticationMode Element	178
6.2.1.2.11	GuestMode Element	179
6.2.1.2.12	BothMode Element	180
6.2.1.2.13	AuthenticationOnlyMode Element	180
6.3	PSM Database Searching Ecore Meta-model Definition	181
6.3.1.1	Introduction	181
6.3.1.2	PSM Database Searching Ecore Meta-model Elements	181
6.3.1.2.1	PSM Database Searching Custom Data Types Definition	181
6.3.1.2.2	AnnotationModel Element	181
6.3.1.2.3	AnnotatedElement Element	183
6.3.1.2.4	Annotation Element	183
6.3.1.2.5	AnnPSMComponentProperty Element	184
6.3.1.2.6	AnnJavaResourceModel Element	184
6.3.1.2.7	AnnHTTPActivityHandler Element	185
6.3.1.2.8	AnnHTTPActivity Element	185
6.3.1.2.9	AnnJavaAlgoController Element	186
6.3.1.2.10	SearchController Element	186
6.3.1.2.11	SearchHTTPActivity Element	187
6.3.1.2.12	SearchHTTPHandler Element	188
6.3.1.2.13	SearchableJavaResourceModel Element	190
6.3.1.2.14	SearchableProperty Element	191
6.4	PSM External Service Composition Ecore Meta-model Definition	192
6.4.1.1	Introduction	192
6.4.1.2	PSM External Service Composition Ecore Meta-model Elements	193
6.4.1.2.1	AnnotationModel Element	193
6.4.1.2.2	AnnotatedElement Element	194
6.4.1.2.3	Annotation Element	194
6.4.1.2.4	AnnJavaAlgoModel Element	194
6.4.1.2.5	AnnJavaResourceModel Element	195
6.4.1.2.6	AnnHTTPActivityHandler Element	196
6.4.1.2.7	AnnHTTPActivity Element	196
6.4.1.2.8	AnnJavaAlgoController Element	197
6.4.1.2.9	JavaRESTClientController Element	197
6.4.1.2.10	JavaRESTClientHTTPActivity Element	199
6.4.1.2.11	QueryParam Element	200
6.4.1.2.12	JavaRESTClientHTTPActivityHandler Element	201
6.4.1.2.13	JavaRESTClientModel Element	202
6.4.1.2.14	JavaInputDataModel Element	204
6.4.1.2.15	JavaOutputDataModel Element	205
6.4.1.2.16	Property Element	206
6.4.1.2.17	Representation Element	207
6.4.1.2.18	NonPersistentOutput Element	207
6.4.1.2.19	AutoPersistentOutput Element	208
6.4.1.2.20	ExistentJavaModelPersistentOutput Element	208
6.5	PSM ABAC Authorization Ecore Meta-model Definition	210
6.5.1.1	Introduction	210
6.5.1.2	PSM ABAC Authorization Ecore Meta-model Elements	210
6.5.1.2.1	AnnotationModel Element	210
6.5.1.2.2	AnnotatedElement Element	211
6.5.1.2.3	Annotation Element	212
6.5.1.2.4	AnnHTTPActivity Element	212
6.5.1.2.5	AnnPSMComponentProperty Element	213

6.5.1.2.6	AnnJPAController Element	213
6.5.1.2.7	AnnHTTPActivityHandler Element	214
6.5.1.2.8	AnnJavaResourceModelManager Element	214
6.5.1.2.9	AnnJavaAlgoResourceModel Element	215
6.5.1.2.10	AnnJavaResourceModel Element	216
6.5.1.2.11	AuthorizationSubject Element	216
6.5.1.2.12	SubjectAttribute Element	217
6.5.1.2.13	AuthorizableResource Element	218
6.5.1.2.14	ResourceAccessPolicySet Element	220
6.5.1.2.15	ResourceAccessPolicy Element	221
6.5.1.2.16	RuleCombiningAlgorithm Element	223
6.5.1.2.17	DenyOverridesAlgorithm Element	223
6.5.1.2.18	PermitOverridesAlgorithm Element	224
6.5.1.2.19	ResourceAccessRule Element	224
6.5.1.2.20	ResourceAccessDenyRule Element	226
6.5.1.2.21	ResourceAccessPermitRule Element	227
6.5.1.2.22	AllowedAction Element	227
6.5.1.2.23	MatchedResourceAttribute Element	228
6.5.1.2.24	MatchedSubjectAttribute Element	230
6.5.1.2.25	MatchedContextAttribute Element	232
6.5.1.2.26	JPAAnnotation Element	233
6.5.1.2.27	AuthorizationPerformer Element	234
6.5.1.2.28	AuthorizationDataHandler Element	235
6.5.1.2.29	AuthorizationPolicyEvaluator Element	236
7	Ontology Software Artefacts to MOF models transformation	237
7.1	Overview	237
7.2	Intermediary YAML Representation	237
7.3	Ontology to YAML Representation Transformation Steps	238
7.3.1	Ontology to YAML Transformation	238
7.3.2	Example Transformation	239
7.4	Intermediary YAML to CIM Transformation Steps	242
8	S-CASE MDE Engine Transformation Definitions	243
8.1	Core CIM to Core PIM Transformation Steps	243
8.1.1	RESTfulServiceCIM element transformation	244
8.1.2	AlgoResourceModel element introduction	245
8.1.3	AlgoResourceController element introduction	247
8.1.4	ResourceModel element introduction	247
8.1.5	ResourceModelManager element introduction	250
8.1.6	ResourceController element introduction	251
8.1.7	ResourceControllerManager element introduction	252
8.1.8	InputRepresentation element transformation (case 1)	252
8.1.9	OutputRepresentation element transformation (case 1)	253
8.1.10	InputRepresentation element transformation (case 2)	253
8.1.11	OutputRepresentation element transformation (case 2)	254
8.1.12	InputRepresentation element transformation (case 3)	254
8.1.13	OutputRepresentation element transformation (case 3)	254
8.1.14	Property element transformation	255
8.1.15	SetterFunction element introduction	256
8.1.16	FunctionParameter element introduction	256
8.1.17	GetterFunction element introduction	257
8.1.18	RDBMSTable element introduction	257
8.1.19	RDBMSTableColumn element introduction (case 1)	258
8.1.20	RDBMSTableColumn element transformation (case 2)	258

8.1.21	RDBMSTableColumn element introduction (case 3).....	259
8.1.22	RepresentationParseFunction element introduction.....	259
8.1.23	RepresentationMarshallingFunction element introduction.....	260
8.1.24	PIMComponentProperty element introduction (case 2).....	260
8.1.25	SetterFunction element introduction (case 2)	261
8.1.26	GetterFunction element introduction (case 2).....	261
8.1.27	PIMComponentProperty element introduction (case 3).....	262
8.1.28	SetterFunction element introduction (case 3)	262
8.1.29	GetterFunction element introduction (case 3).....	263
8.1.30	ResourceControllerCRUDActivity element introduction (case 1).....	263
8.1.31	ResourceControllerCRUDActivity element introduction (case 2).....	264
8.1.32	ResourceControllerCRUDActivity element introduction (case 3).....	265
8.1.33	CRUDActivityHandler element introduction (case 1)	265
8.1.34	CRUDActivityHandler element introduction (case 2)	266
8.1.35	DatabaseController element introduction	267
8.1.36	RDBMSActivity element introduction	267
8.1.37	CreateHypermediaPIMFunction element introduction	268
8.1.38	HypermediaLink element introduction	268
8.2	Authentication Extension CIM to PIM ATL Rules	269
8.2.1	AnnotationModel element transformation.....	269
8.2.2	AnnCRUDActivity element transformation.....	270
8.2.3	AnnCRUDActivityhandler element introduction	270
8.2.4	AnnDatabaseController element introduction.....	271
8.2.5	AuthenticationPerformer element introduction	271
8.2.6	AuthenticationToken element transformation	272
8.2.7	Password element transformation	272
8.2.8	GuestMode element transformation	273
8.2.9	AuthenticationOnlyMode element transformation	274
8.2.10	BothMode element transformation	274
8.3	Database Searching Extension CIM to PIM ATL Rules	275
8.3.1	AnnotationModel element transformation.....	275
8.3.2	AnnPIMComponentProperty element transformation	276
8.3.3	AnnAlgoResourceController element transformation	276
8.3.4	AnnCRUDActivity element introduction	276
8.3.5	AnnCRUDActivityHandler element introduction	277
8.3.6	AnnResourceModel element transformation	277
8.3.7	SearchController element transformation	278
8.3.8	SearchCRUDActivity element introduction.....	278
8.3.9	SearchCRUDActivityHandler element introduction.....	278
8.3.10	SearchableResourceModel element transformation	279
8.3.11	SearchableProperty element transformation	280
8.4	External Service Composition Extension CIM to PIM ATL Rules	280
8.4.1	AnnotationModel element transformation.....	280
8.4.2	AnnAlgoResource element transformation.....	281
8.4.3	AnnCRUDResource element transformation.....	281
8.4.4	AnnCRUDActivity element introduction	282
8.4.5	AnnCRUDActivityHandler element introduction	282
8.4.6	AnnAlgoResourceModel element introduction	283
8.4.7	RESTClientResource element transformation	283
8.4.8	RESTClientCRUDActivity element introduction	284
8.4.9	RESTClientModel element introduction	284
8.4.10	TargetRESTService element transformation	285

8.4.11	InputDataModel element transformation	285
8.4.12	Representation element transformation	286
8.4.13	Property element transformation	287
8.4.14	QueryParam element transformation	287
8.4.15	NonPersistentOutput element transformation	288
8.4.16	AutoPersistentOutput element transformation	288
8.4.17	ExistentCRUDPersistentOutput element transformation	289
8.5	ABAC Authorization Extension CIM to PIM ATL Rules.....	290
8.5.1	AnnotationModel element transformation.....	290
8.5.2	AnnResource element transformation (case 1).....	291
8.5.3	AnnResource element transformation (case 2).....	291
8.5.4	AnnResource element transformation (case 3).....	292
8.5.5	AnnProperty element transformation.....	293
8.5.6	AnnCRUDActivity element transformation.....	293
8.5.7	AnnCRUDActivityHandler element introduction	294
8.5.8	AnnDatabaseController element introduction	294
8.5.9	AuthorizationSubject element transformation	294
8.5.10	SubjectAttribute element transformation	295
8.5.11	AuthorizableResource element transformation.....	295
8.5.12	AuthorizationPerformer element introduction	296
8.5.13	AuthorizationPolicyEvaluator element introduction.....	297
8.5.14	ResourceAccessPolicySet element transformation	297
8.5.15	ResourceAccessPolicy element transformation	298
8.5.16	DenyOverridesAlgorithm element transformation	299
8.5.17	PermitOverridesAlgorithm element transformation	299
8.5.18	ResourceAccessPermitRule element transformation	299
8.5.19	ResourceAccessDenyRule element transformation	300
8.5.20	AllowedAction element transformation.....	301
8.5.21	MatchedResourceAttribute element transformation	302
8.5.22	MatchedContextAttribute element transformation	302
8.5.23	MatchedSubjectAttribute element transformation	303
8.5.24	AuthorizationDataHandler element introduction	304
8.5.25	AuthorizationDataTable element introduction	304
8.5.26	AuthorizationDataTableColumn element introduction	305
9	PIM extensions to PSM extensions Transformation Definition	307
9.1	PIM extensions to PSM extensions Transformation Steps	307
9.2	Authentication Extension PIM to PSM ATL Rules	307
9.2.1	AnnotationModel element transformation.....	307
9.2.2	AnnDatabaseController element transformation	308
9.2.3	AnnCRUDActivcity element transformation	308
9.2.4	AnnCRUDActivityHandler element transformation	308
9.2.5	AuthenticationPerformer element transformation.....	309
9.2.6	AuthenticationToken element transformation	310
9.2.7	Password element transformation	310
9.2.8	GuestMode element transformation	311
9.2.9	AuthenticationOnlyMode element transformation	311
9.2.10	BothMode element transformation	312
9.3	Database Searching Extension PIM to PSM ATL Rules	312
9.3.1	AnnotationModel element transformation.....	312
9.3.2	AnnPIMComponentProperty element transformation	313
9.3.3	AnnAlgoResourceController element transformation	314

9.3.4	AnnCRUDActivity element transformation.....	314
9.3.5	AnnCRUDActivityHandler element transformation.....	315
9.3.6	AnnResourceModel element transformation	315
9.3.7	SearchController element transformation	316
9.3.8	SearchCRUDActivity element transformation	316
9.3.9	SearchCRUDActivityHandler element transformation	317
9.3.10	SearchableResourceModel element transformation	317
9.3.11	SearchableProperty element transformation	318
9.4	External Service Composition Extension PIM to PSM ATL Rules.....	318
9.4.1	AnnotationModel element transformation.....	318
9.4.2	AnnAlgoResourceModel element transformation	319
9.4.3	AnnResourceModel element transformation	320
9.4.4	AnnAlgoResourceController element transformation	320
9.4.5	AnnCRUDActivity element transformation.....	321
9.4.6	AnnCRUDActivityHandler element transformation.....	321
9.4.7	RESTClientController element transformation.....	321
9.4.8	RESTClientCRUDActivity element transformation.....	322
9.4.9	RESTClientCRUDActivityHandler element transformation	323
9.4.10	QueryParam element transformation.....	323
9.4.11	RESTClientModel element transformation	324
9.4.12	InputDataModel element transformation	325
9.4.13	Property element transformation	325
9.4.14	Representation element transformation	326
9.4.15	NonPersistentOutput element transformation.....	326
9.4.16	AutoPersistentOutput element transformation.....	327
9.4.17	ExistenCRUDPersistentOutput element transformation	327
9.5	ABAC Authorization Extension PIM to PSM ATL Rules.....	328
9.5.1	AnnotationModel element transformation.....	328
9.5.2	AnnResourceModel element transformation	329
9.5.3	AnnAlgoResourceModel element transformation	330
9.5.4	AnnResourceModelManager element transformation.....	330
9.5.5	AnnCRUDActivityHandler element transformation.....	331
9.5.6	AnnDatabaseController element transformation	331
9.5.7	AnnPIMComponentProperty element transformation	332
9.5.8	AnnResourceControllerCRUDActivity element transformation	332
9.5.9	AuthorizationSubject element transformation	333
9.5.10	SubjectAttribute element transformation	333
9.5.11	AuthorizableResource element transformation	334
9.5.12	AuthorizationPerformer element transformation	335
9.5.13	AuthorizationPolicyEvaluator element transformation	335
9.5.14	ResourceAccessPolicySet element transformation	336
9.5.15	ResourceAccessPolicy element transformation	336
9.5.16	DenyOverridesAlgorithm element transformation	337
9.5.17	PermitOverridesAlgorithm element transformation	337
9.5.18	ResourceAccessDenyRule element transformation	338
9.5.19	ResourceAccessPermitRule element transformation	339
9.5.20	AllowedAction element transformation.....	340
9.5.21	MatchedResourceAttribute element transformation	340
9.5.22	MatchedContextAttribute element transformation	341
9.5.23	MatchedSubjectAttribute element transformation	342
9.5.24	AuthorizationDataHandler element transformation.....	343
9.5.25	AuthorizationTable element transformation	343

9.5.26 AuthorizationDataTableColumn element transformation	343
10 PSM to Code Transformation Definition	345
10.1 PSM to Code Transformation.....	345
10.2 PSM to Code Acceleo Templates	345
References.....	356
A. Annex A – OCL Constraints Full List	357
B. Annex B – ATL/Acceleo Transformations Full List.....	358

List of Figures

Figure 2-1 Web service friendliness Richardson's Maturity Model	32
Figure 2-2 Basic structure of a Rails service (http://binaryhash.com/ruby-on-rails).....	33
Figure 2-3 AtlanMod overview presentation of EMF-REST at EclipseCon Europe 2013 Symposium ...	34
Figure 2-4 Left: example JSON response. Right: example of RESTful API	34
Figure 2-5 Django-REST-Framework serializers.....	35
Figure 2-6 Service response to a GET HTTP request	35
Figure 3-1 The four phases of Model Driven Engineering.....	39
Figure 3-2 Simplified subset of the Ecore meta-model.....	40
Figure 3-3 The Eclipse Modelling Framework unifies Java, XML and UML.....	41
Figure 3-4 ATL Operational Context	41
Figure 3-5 Abstract from an Acceleo code template used in S-CASE MDE engine to produce code....	42
Figure 3-6 Conceptual design of an extension meta-model	44
Figure 3-7 Indicative Core CIM extension meta-model.	44
Figure 3-8 Transformations within the 2D MDE mechanism	45
Figure 4-1 REST abstract concepts of S-CASE CIM meta-model.....	47
Figure 4-2 S-CASE Core CIM meta-model.....	50
Figure 4-3 An example resource that could be used as authenticaiton model	56
Figure 4-4 Searchable Resource Properties	56
Figure 4-5 Authenticaiton CIM extension meta-model.....	58
Figure 4-6 Database Searching CIM extension meta-model.....	67
Figure 4-7 External Service Composition CIM extension meta-model.....	73
Figure 4-8 RESTClientResource annotation and its relations.....	77
Figure 4-9 TargetRESTService element and its annotations	78
Figure 4-10 InputDataModel element and its relations.....	80
Figure 4-11 OutputDataModel element and its relations.....	81
Figure 4-12 Authorization Subject annotation and its relations	88
Figure 4-13 AuthorizableResource annotation and its relations	90
Figure 4-14 ResourceAccessPolicySet element and its relations	91
Figure 4-15 ResourceAccessPolicy element and its relations	93
Figure 4-16 ResourceAccessRule element and its relations.....	95
Figure 4-17 MatchedSubjectAttribute element and its relations	98
Figure 4-18 AllowedAction element and its relations.....	99
Figure 4-19 MatchedResourceAttribute element and its relations.	100
Figure 4-20 MatchedContextAttribute element and its relations.	101

Figure 5-1 Authentication PIM extension meta-model.....	104
Figure 5-2 AuthenticationPerformer annotation and its relations.	109
Figure 5-3 AuthenticationModel annotation and its relations.	112
Figure 5-4 Database Searching PIM extension meta-model.....	114
Figure 5-5 SearchController annotation and its relations.....	120
Figure 5-6 SearchCRUDActivity	121
Figure 5-7 SearchCRUDActivityHandler.....	122
Figure 5-8 SearchableResourceModel	123
Figure 5-9 SearchableProperty annotation and its relations.....	124
Figure 5-10 External Service Composition meta-model.....	126
Figure 5-11 RESTClientController annotation and its relations.	131
Figure 5-12 RESTClientCRUDActivity element and its annotations.....	133
Figure 5-13 RESTClientCRUDActivityHandler	134
Figure 5-14 RESTClientModel annotation and its relations.....	136
Figure 5-15 InputDataModel element and its relations.....	137
Figure 5-16 OutputDataModel element and its annotations.....	139
Figure 5-17 AuthorizationSubject annotation and its relations.....	149
Figure 5-18 AuthorizableResource annotation and its relations	151
Figure 5-19 ResourceAccessPolicySet element and its relations	153
Figure 5-20 ResourceAccessPolicy element and its relations	156
Figure 5-21 ResourceAccessRule element and its relations.....	158
Figure 5-22 AllowedAction element and its relations.....	160
Figure 5-23 MatchedResourceAttribute element and its relations	161
Figure 5-24 MatchedContextAttribute element and its relations	163
Figure 5-25 MatchedSubjectAttribute element and its relations.	164
Figure 5-26 AuthorizationPerformer annotation and its relations.	166
Figure 5-27 AuthorizationDataHandler annotation and its relations.	168
Figure 6-1 Authentication PSM extension meta-model.....	171
Figure 6-2 AuthenticationPerformer annotation and its relations	176
Figure 6-3 AuthenticationMode annotation and its relations	178
Figure 6-4 Database Searching PSM extension meta-model	181
Figure 6-5 SearchController annotation and its relations.....	187
Figure 6-6 SearchHTTPActivity	188
Figure 6-7 SearchHTTPHandler annotation and its relations.....	189
Figure 6-8 SearchableJavaResourceModel annotation and its relations.....	190
Figure 6-9 SearchableProperty.....	191

Figure 6-10 External Service Composition PSM extension meta-model.....	192
Figure 6-11 JavaRESTClientController annotation and its relations.....	198
Figure 6-12 JavaRESTClientHTTPActivity element and its relations.....	199
Figure 6-13 JavaRESTClientHTTPActivityHandler	201
Figure 6-14 JavaRESTClientModel annotation and its relations.	202
Figure 6-15 JavaInputDataModel element and its relations.....	204
Figure 6-16 JavaOutputDataModel element and its relations.....	205
Figure 6-17 ExistentalJavaModelPersistentOutput annotation and its relations.....	208
Figure 6-18 AuthorizationSubject annotation and its relations	216
Figure 6-19 AuthorizableResource annotation and its relations	218
Figure 6-20 ResourceAccessPolicySet element and its relations	220
Figure 6-21 ResourceAccessPolicy element and its relations	222
Figure 6-22 ResourceAccessRule element and its relations.....	225
Figure 6-23 AllowedAction element and its relations.....	227
Figure 6-24 MatchedResourceAttribute element and its relations.	229
Figure 6-25 MatchedSubjectAttribute element and its relations.	230
Figure 6-26 MatchedContextAttribute.....	232
Figure 6-27 AuthorizationPerformer annotation and its relations.	234
Figure 6-28 AuthorizationDataHandler	235
Figure 7-1 Schema of the Intermediary YAML Representation	237
Figure 7-2 Example YAML file for Project Restmarks	240
Figure 7-3 User-modified YAML file for Project Restmarks.....	242
Figure 8-1 Indicative S-CASE transformation example introduced in D2.2.....	243
Figure 10-1 Maven Standard Layout Directory of WSAT system	346

List of Tables

Table 4-1 Supported Media Types	49
Table 4-2 CRUDVerb Data Type.....	49
Table 4-3 RESTfulServiceCIM 's Properties.....	50
Table 4-4 RESTfulServiceCIM's Relations	51
Table 4-5 Resource's Properties.....	51
Table 4-6 Resource's Relations.....	52
Table 4-7 CRUDActivity's Element.....	53
Table 4-8 Property's Properties.....	53
Table 4-9 InputRepresentation's Properties	54
Table 4-10 OutputRepresentation's Properties	55
Table 4-11 AnnotationModel's Properties	58
Table 4-12 AnnotationModel's Relations.....	58
Table 4-13 AnnCRUDActivity's Properties	60
Table 4-14 AnnCRUDActivity's Relations.....	61
Table 4-15 AnnProperty's Relations	61
Table 4-16 AnnResource's Relations	62
Table 4-17 AuthenticationModel's Relations	62
Table 4-18 AuthenticationToken's Properties.....	63
Table 4-19 AuthenticationToken's Relations	64
Table 4-20 AuthenticationMode's Relations.....	65
Table 4-21 AnnotationModel's Properties	67
Table 4-22 AnnotationModel's Relations.....	67
Table 4-23 AnnProperty's Relations	69
Table 4-24 AnnAlgoResource's Relations	70
Table 4-25 AnnCRUDResource's Relations	70
Table 4-26 SearchResource's Relations.....	71
Table 4-27 SearchableResource's Relations	72
Table 4-28 SearchableProperty's Relations.....	73
Table 4-29 AnnotationModel's Properties	74
Table 4-30 AnnotationModel's Relations	74
Table 4-31 AnnCRUDResource Relations	76
Table 4-32 AnnAlgoResource's Relations	76
Table 4-33 RESTClientResource's Relations	77
Table 4-34 TargetRESTService's Properties.....	78

Table 4-35 TargetRESTService's Relations.....	78
Table 4-36 QueryParam's Properties	79
Table 4-37 InputDataModel's Relations.....	80
Table 4-38 OutputDataModel's Relations.....	81
Table 4-39 Representation's Properties.....	82
Table 4-40 Property's Properties.....	82
Table 4-41 ExistentsCRUDPersistentOutput's Relations.....	84
Table 4-42 AnnotationModel's Properties.....	85
Table 4-43 AnnotationModel's Relations.....	85
Table 4-44 AuthorizationSubject's Relations	89
Table 4-45 SubjectAttribute's Properties	89
Table 4-46 AuthorizableResource's Properties	90
Table 4-47 AuthorizableResource's Relations.....	91
Table 4-48 ResourceAccessPolicySet's Relations	92
Table 4-49 ResourceAccessPolicy's Relations	93
Table 4-50 ResourceAccessRule's Relations.....	96
Table 4-51 MatchedSubjectAttribute's Properties	98
Table 4-52 MatchedSubjectAttribute's Relations	99
Table 4-53 AllowedActions's Relations	100
Table 4-54 MatchedResourceAttribute's Properties	101
Table 4-55 MatchedResourceAttribute's Relations	101
Table 4-56 MatchedContextAttribute's Properties.....	102
Table 4-57 MatchedContextAttribute's Relations.....	102
Table 4-58 ContextResource's Relations.....	103
Table 5-1 AnnotationModel's Properties	105
Table 5-2 AnnotationModel's Relations.....	105
Table 5-3 AnnotatedElement's Properties	106
Table 5-4 AnnCRUDActivityHandler's Relations.....	107
Table 5-5 AnnCRUDActivity's Relations.....	107
Table 5-6 AnnDatabaseController's Relations.....	108
Table 5-7 AuthenticationPerformer's Properties	109
Table 5-8 AuthenticationPerformer's Relations.....	109
Table 5-9 AuthenticationToken's Properties.....	110
Table 5-10 AuthenticationMode's Relations.....	112
Table 5-11 AnnotationModel's Properties	115
Table 5-12 AnnotationModel's Relations.....	115

Table 5-13 AnnPIMComponentProperty's Relations	117
Table 5-14 AnnResourceModel's Relations.....	118
Table 5-15 AnnCRUDActivityHandler's Relations.....	118
Table 5-16 AnnCRUDActivity's Relations.....	119
Table 5-17 AnnAlgoResourceController's Relations.....	119
Table 5-18 SearchController's Relations	120
Table 5-19 SearchCRUDActivity's Relations	121
Table 5-20 SearchCRUDActivityHandler's Relations	122
Table 5-21 SearchableResourceModel's Relations	124
Table 5-22 SearchableProperty's Relations.....	125
Table 5-23 AnnotationModel's Properties	126
Table 5-24 AnnotationModel's Relations.....	127
Table 5-25 AnnAlgoResourceModel's Relations	129
Table 5-26 AnnResourceModel's Relations.....	129
Table 5-27 AnnCRUDActivityHandler's Relations.....	130
Table 5-28 AnnCRUDActivity's Relations.....	130
Table 5-29 AnnAlgoResourceController's Relations.....	131
Table 5-30 RESTClientController's Relations	132
Table 5-31 RESTClientCRUDActivity's Relations	133
Table 5-32 RESTClientCRUDActivityHandler's Properties	134
Table 5-33 RESTClientCRUDActivityHandler's Relations	135
Table 5-34QueryParam's Properties	135
Table 5-35 RESTClientModel's Relations.....	136
Table 5-36 InputDataModel's Relations.....	138
Table 5-37 OutputDataModel's Relations.....	138
Table 5-38 Property's Properties.....	139
Table 5-39 Representation's Properties.....	140
Table 5-40 ExistentsCRUDPersistentOutput's Relations.....	142
Table 5-41 AnnotationModel's Properties	143
Table 5-42 AnnotationModel's Relations.....	143
Table 5-43 AnnResourceControllerCRUDActivity's Relations	145
Table 5-44 AnnPIMComponentProperty's Relations	146
Table 5-45 AnnDatabaseController's Relations.....	146
Table 5-46 AnnCRUDActivityHandler's Relations.....	147
Table 5-47 AnnResourceModelManager's Relations	147
Table 5-48 AnnAlgoResourceModel's Relations	148

Table 5-49 AnnResourceModel's Relations.....	149
Table 5-50 AuthorizationSubject's Relations	150
Table 5-51 SubjectAttribute's Properties.....	150
Table 5-52 AuthorizableResource's Properties	152
Table 5-53 AuthorizableResource's Relations.....	152
Table 5-54 ResourceAccessPolicySet's Relations	154
Table 5-55 ResourceAccessPolicy's Relations	156
Table 5-56 ResourceAccessRule's Relations.....	158
Table 5-57 AllowedAction's Relations.....	160
Table 5-58 MatchedResourceAttribute's Properties	162
Table 5-59 MatchedResourceAttribute's Relations	162
Table 5-60 MatchedContextAttribute's Properties.....	163
Table 5-61 MatchedContextAttribute's Relations.....	163
Table 5-62 MatchedSubjectAttribute's Properties	165
Table 5-63 MatchedSubjectAttribute's Relations	165
Table 5-64 AuthorizationPerformer's Relations.....	166
Table 5-65 AuthorizationPolicyEvaluator's Relations.....	167
Table 5-66 AuthorizationDataHandler's Relations.....	168
Table 5-67 AuthorizationDataTable's Relations	169
Table 6-1 AnnotationModel's Properties	172
Table 6-2 AnnotationModel's Relations.....	172
Table 6-3 AnnHTTPActivity's Relations.....	174
Table 6-4 AnnHTTPActivityHandler's Relations.....	174
Table 6-5 AnnJPAController's Relations	175
Table 6-6 AuthenticationPerformer's Properties	176
Table 6-7 AuthenticationPerformer's Relations.....	176
Table 6-8 AuthenticationToken's Properties.....	177
Table 6-9 AuthenticationMode's Relations	179
Table 6-10 AnnotationModel's Properties	182
Table 6-11 AnnotationModel's Relations	182
Table 6-12 AnnPSMComponentProperty's Relations	184
Table 6-13 AnnJavaResourceModel's Relations.....	184
Table 6-14 AnnHTTPActivityHandler's Relations.....	185
Table 6-15 AnnHTTPActivity's Relations.....	186
Table 6-16 AnnJavaAlgoController's Relations.....	186
Table 6-17 SearchController's Relations	187

Table 6-18 SearchHTTPActivity's Relations	188
Table 6-19 SearchHTTPHandler's Relations	189
Table 6-20 SearchableJavaResourceModel's Relations	190
Table 6-21 SearchableProperty's Relations.....	191
Table 6-22 AnnotationModel's Properties	193
Table 6-23 AnnotationModel's Relations.....	193
Table 6-24 AnnJavaAlgoModel's Relations.....	195
Table 6-25 AnnJavaResourceModel's Relations.....	195
Table 6-26 AnnHTTPActivityHandler's Relations.....	196
Table 6-27 AnnHTTPActivity's Relations.....	197
Table 6-28 AnnJavaAlgoController's Relations.....	197
Table 6-29 JavaRESTClientController's Relations	198
Table 6-30 JavaRESTClientHTTPActivity's Relations.....	200
Table 6-31QueryParam's Properties	200
Table 6-32 JavaRESTClientHTTPActivityHandler's Properties	201
Table 6-33 JavaRESTClientHTTPActivityHandler's Relations	202
Table 6-34 JavaRESTClientModel's Relations	203
Table 6-35 JavaInputDataModel's Relations	204
Table 6-36 JavaOutputDataModel's Relations	206
Table 6-37 Property's Properties.....	206
Table 6-38 Representation's Properties.....	207
Table 6-39 ExistentalJavaModelPersistentOutput's Relations.....	209
Table 6-40 AnnotationModel's Properties	210
Table 6-41 AnnotationModel's Relations	210
Table 6-42 AnnHTTPActivity's Relations.....	212
Table 6-43 AnnPSMComponentProperty's Relations	213
Table 6-44 AnnJPAController's Relations	213
Table 6-45 AnnHTTPActivityHandler's Relations.....	214
Table 6-46 AnnJavaResourceModelManager's Relations	215
Table 6-47 AnnJavaAlgoResourceModel's Relations.....	215
Table 6-48 AnnJavaResourceModel's Relations.....	216
Table 6-49 AuthorizationSubject's Relations	217
Table 6-50 SubjectAttribute's Properties	217
Table 6-51 AuthorizableResource's Properties	219
Table 6-52 AuthorizableResource's Relations	219
Table 6-53 ResourceAccessPolicySet's Relations	221

Table 6-54 ResourceAccessPolicy's Relations	222
Table 6-55 ResourceAccessRule's Relations.....	225
Table 6-56 AllowedAction's Relations	228
Table 6-57 MatchedResourceAttribute's Properties	229
Table 6-58 MatchedResourceAttribute's Relations	229
Table 6-59 MatchedSubjectAttribute's Properties	231
Table 6-60 MatchedSubjectAttribute's Relations	231
Table 6-61 MatchedContextAttribute's Properties.....	232
Table 6-62 MatchedContextAttribute's Relations.....	232
Table 6-63 JPAnnotation's Properties.....	233
Table 6-64 AuthorizationPerformer's Relations.....	234
Table 6-65 AuthorizationDataHandler's Relations	236
Table 8-1 CIMToPIMService ATL rule	244
Table 8-2 createAlgoResourceModel ATL rule.....	245
Table 8-3 createAlgoResourceController ATL rule.....	247
Table 8-4 createCRUDModel ATL rule.....	248
Table 8-5 createCRUDRMMManager ATL rule	250
Table 8-6 createRController ATL rule	251
Table 8-7 CreateRCManager ATL rule	252
Table 8-8 addResourceModelIRepresentation ATL rule	253
Table 8-9 addRModelOutputRepresentation ATL rule.....	253
Table 8-10 addAlgoModelIRepresentation ATL rule	253
Table 8-11 addAlgoModelORepresentation ATL rule	254
Table 8-12 addRMMManagerIRepresentation ATL rule.....	254
Table 8-13 addRMMManagerIRepresentation ATL rule.....	255
Table 8-14 addRModelProperties ATL rule	255
Table 8-15 addSetterFunction ATL rule.....	256
Table 8-16 createFunctionParameter ATL rule	256
Table 8-17 addGetterFunction ATL rule	257
Table 8-18 createRDBMSTable ATL rule.....	257
Table 8-19 createRDBMSTableColumn ATL rule	258
Table 8-20 createRDBMSTablePrimaryKey ATL rule	258
Table 8-21 createRDBMSTableForeignKey ATL rule.....	259
Table 8-22 createParseFunction ATL rule	260
Table 8-23 createMarshalFunction ATL rule	260
Table 8-24 createRModelPrimaryIdentifier ATL rule	260

Table 8-25 createRModelPrimaryIdSetter ATL rule	261
Table 8-26 createRModelPrimaryIdGetter ATL rule.....	261
Table 8-27 createLinkListProperty ATL rule	262
Table 8-28 createLinkListPropertySetter ATL rule.....	262
Table 8-29 createLinkListPropertyGetter ATL rule.....	263
Table 8-30 createAlgoControllerCRUDActivity ATL rule.....	263
Table 8-31 addRControllerCRUDActivity ATL rule.....	264
Table 8-32 addRCManagerCRUDActivity ATL rule	265
Table 8-33 createCRUDActivityHandler ATL rule	265
Table 8-34 createRCManagerCRUDActivityHandler ATL rule	266
Table 8-35 createDatabaseController ATL rule	267
Table 8-36 createRDBMSActivity ATL rule	268
Table 8-37 createAlgoResourceHypermediaFunction ATL rule	268
Table 8-38 createHypermediaLink ATL rule	269
Table 8-39 CIMtoPIMAuthentication ATL rule	269
Table 8-40 createPIMAnnCRUDActivity ATL rule	270
Table 8-41 createPIMAnnCRUDActivityHandler ATL rule	271
Table 8-42 createAnnDatabaseController ATL rule.....	271
Table 8-43 createAuthenticationPerformer ATL rule.....	271
Table 8-44 createPIMAAuthenticationToken ATL rule	272
Table 8-45 createPIMAAuthenticationPasswordToken ATL rule	273
Table 8-46 createGuestMode ATL rule	273
Table 8-47 createAuthenticationOnlyMode ATL rule	274
Table 8-48 createBothMode ATL rule	274
Table 8-49 SearchLayerCIMToPIM ATL rule	275
Table 8-50 createAnnPIMComponentProperty ATL rule	276
Table 8-51 createAnnAlgoResourceController ATL rule	276
Table 8-52 createAnnCRUDActivity ATL rule.....	277
Table 8-53 createAnnCRUDActivityHandler ATL rule.....	277
Table 8-54 createAnnResourceModel ATL rule.....	277
Table 8-55 createPIMSearchController ATL rule.....	278
Table 8-56 createPIMSearchCRUDActivity ATL rule.....	278
Table 8-57 createPIMSearchCRUDActivityHandler ATL rule.....	279
Table 8-58 createSearchableResourceModel ATL rule	279
Table 8-59 createPIMSearchableProperty ATL rule	280
Table 8-60 ExternalServiceCIMToPIM ATL rule	280

Table 8-61 createAnnAlgoResourceController ATL rule	281
Table 8-62 createAnnCRUDResource ATL rule.....	282
Table 8-63 createAnnCRUDActivity ATL rule.....	282
Table 8-64 createAnnCRUDActivityHandler ATL rule.....	282
Table 8-65 createAnnAlgoResourceModel ATL rule	283
Table 8-66 createRESTClientController ATL rule	283
Table 8-67 createRESTClientCRUDActivity ATL rule.....	284
Table 8-68 createRESTClientModel ATL rule.....	284
Table 8-69 createRESTClientCRUDActivityHandler ATL rule	285
Table 8-70 createInputDataModel ATL rule.....	286
Table 8-71 createRepresentation ATL rule.....	286
Table 8-72 createProperty ATL rule	287
Table 8-73 createQueryParam ATL rule	287
Table 8-74 createNonPersistentOutput ATL rule	288
Table 8-75 createAutoPersistentOutput ATL rule.....	289
Table 8-76 createExistentCRUDPersistentOutput ATL rule.....	289
Table 8-77 AuthorizationCIMToPIM ATL rule.....	290
Table 8-78 createAnnResourceModel ATL rule.....	291
Table 8-79 createAnnResourceModel ATL rule.....	292
Table 8-80 createAnnResourceModelManager ATL rule	292
Table 8-81 createAnnAlgoResourceModel ATL rule	292
Table 8-82 createAnnPIMComponentProperty ATL rule	293
Table 8-83 createAnnResourceControllerCRUDActivity ATL rule	293
Table 8-84 createAnnCRUDActivityHandler ATL rule.....	294
Table 8-85 createAnnDatabaseController ATL rule.....	294
Table 8-86 createAuthorizationSubject ATL rule	295
Table 8-87 createSubjectAttribute ATL rule.....	295
Table 8-88 createAuthorizableResource ATL rule	296
Table 8-89 createAuthorizationPerformer ATL rule.....	296
Table 8-90 createAuthorizationPolicyEvaluator ATL rule	297
Table 8-91 createResourceAccessPolicySet ATL rule	297
Table 8-92 createResourceAccessPolicy ATL rule	298
Table 8-93 createResourceAccessPermitRule ATL rule.....	299
Table 8-94 createResourceAccessDenyRule ATL rule	300
Table 8-95 createAllowedAction ATL rule.....	301
Table 8-96 createMatchedResourceAttribute ATL rule	302

Table 8-97 createMatchedContextAttribute ATL rule	303
Table 8-98 createMatchedSubjectAttribute ATL rule	303
Table 8-99 createAuthorizationDataHandler ATL rule.....	304
Table 8-100 createAuthorizationDataTable ATL rule	305
Table 8-101 createAuthorizationDataTableColumn ATL rule	305
Table 9-1 PIMToPSMAuthentication ATL rule.....	307
Table 9-2 createAnnJPAController ATL rule	308
Table 9-3 createAnnHTTPActivity ATL rule	308
Table 9-4 createAnnHTTPActivityHandler ATL rule.....	309
Table 9-5 createAuthenticationPerformer ATL rule.....	309
Table 9-6 createAuthenticationToken ATL rule	310
Table 9-7 createPassword ATL rule	310
Table 9-8 createGuestMode ATL rule	311
Table 9-9 createAuthenticationOnlyMode ATL rule	311
Table 9-10 createBothMode ATL rule	312
Table 9-11 SearchLayerPIMToPSM ATL rule	313
Table 9-12 createAnnPSMComponentProperty ATL rule	314
Table 9-13 createAnnJavaAlgoController ATL rule.....	314
Table 9-14 createAnnHTTPActivity ATL rule	314
Table 9-15 createAnnHTTPActivityHandler ATL rule.....	315
Table 9-16 createAnnJavaResourceModel ATL rule.....	315
Table 9-17 createPSMSearchController ATL rule	316
Table 9-18 createSearchHTTPActivity ATL rule	316
Table 9-19 createSearchHTTPActivityHandler ATL rule	317
Table 9-20 createPSMSearchableResource ATL rule	317
Table 9-21 createPSMSearchableProperty ATL rule	318
Table 9-22 ExternalServicePIMToPSM ATL rule	318
Table 9-23 createAnnJavaAlgoModel ATL rule	319
Table 9-24 createAnnJavaResourceModel ATL rule.....	320
Table 9-25 createAnnJavaAlgoController ATL rule.....	320
Table 9-26 createAnnHTTPActivity ATL rule	321
Table 9-27 createAnnHTTPActivityHandler ATL rule.....	321
Table 9-28 createJavaRESTClientController ATL rule	322
Table 9-29 createJavaRESTClientHTTPActivity ATL rule.....	322
Table 9-30 createJavaRESTClientHTTPActivityHandler ATL rule	323
Table 9-31 createQueryParam ATL rule	324

Table 9-32 createJavaRESTClientModel ATL rule	324
Table 9-33 createJavaInputDataModel ATL rule	325
Table 9-34 createProperty ATL rule	325
Table 9-35 createRepresentation ATL rule	326
Table 9-36 createNonPersistentOutput ATL rule	326
Table 9-37 createAutoPersistentOutput ATL rule	327
Table 9-38 createExistentJavaModelPersistentOutput ATL rule	328
Table 9-39 AuthorizationPIMToPSM ATL rule	329
Table 9-40 createAnnJavaResourceModel ATL rule	330
Table 9-41 createAnnJavaAlgoResourceModel ATL rule	330
Table 9-42 createAnnJavaResourceModelManager ATL rule	331
Table 9-43 createAnnHTTPActivityHandler ATL rule	331
Table 9-44 createAnnJPAController ATL rule	331
Table 9-45 createAnnPSMComponentProperty ATL rule	332
Table 9-46 createAnnHTTPActivity ATL rule	332
Table 9-47 createAuthorizationSubject ATL rule	333
Table 9-48 createSubjectAttribute ATL rule	333
Table 9-49 createAuthorizableResource ATL rule	334
Table 9-50 createAuthorizationPerformer ATL rule	335
Table 9-51 createResourceAccessPolicySet ATL rule	336
Table 9-52 createResourceAccessPolicy ATL rule	337
Table 9-53 createResourceAccessDenyRule ATL rule	338
Table 9-54 createResourceAccessPermitRule ATL rule	339
Table 9-55 createAllowedAction ATL rule	340
Table 9-56 createMatchedResourceAttribute ATL rule	340
Table 9-57 createMatchedContextAttribute ATL rule	341
Table 9-58 createMatchedSubjectAttribute ATL rule	342
Table 9-59 createAuthorizationDataHandler ATL rule	343
Table 10-1 generate.mtl input/output	346
Table 10-2 hibernateConfigurationFile.mtl input/output	347
Table 10-3 javaJAXRSPublisher.mtl input/output	348
Table 10-4 javaAlgoResourceController.mtl input/output	348
Table 10-5 javaAlgoResourceModel.mtl input/output	348
Table 10-6 javaHibernateController.mtl input/output	348
Table 10-7 javaHibernateUtil.mtl input/output	349
Table 10-8 javaHTTPActivityHandler.mtl input/output	349

Table 10-9 javaHypermediaLink.mtl input/output.....	350
Table 10-10 javaResourceController.mtl input/output.....	350
Table 10-11 javaResourceControllerManager.mtl input/output	350
Table 10-12 javaResourceModel.mtl input/output	350
Table 10-13 javaResourceModelManager.mtl input/output.....	351
Table 10-14 mavenConfigurationFile.mtl input/output.....	351
Table 10-15 webXMLConfigurationFile.mtl input/output.....	351
Table 10-16 javaPersistentUtil.mtl input/output.....	351
Table 10-17 javaLuceneStringSetBrigde.mtl input/output	351
Table 10-18 persistenceXML.mtl input/output.....	352
Table 10-19 JavaOutputModel.mtl input/output	352
Table 10-20 AuthorizationPolicyEvaluator.mtl input/output	352
Table 10-21 putResourceHandlerSingleParent.mtl input/output.....	352
Table 10-22 generateSearchHTTPHandler.mtl input/output.....	353
Table 10-23 javaRESTClientHTTPAcitivityHandler.mtl input/output	353
Table 10-24 ResourceAccessPolicySet.mtl input/output	354
Table 10-25 javaResourceAccessPolicy.mtl input/output.....	354
Table 10-26 javaResourceAccessRule.mtl input/output	354
Table 10-27 javaAllowedAction.mtl input/output	354
Table 10-28 javaMatchedResourceAttribute.mtl input/output.....	355

Abbreviations and Acronyms

ATL: Atlas Transformation Language (a declarative M2M transformation language)

CIM: Computational Independent Model

EMF: Eclipse Modelling Framework

M2M: Model to Model transformation

MDE: Model Driven Engineering

MOF: Meta-Object Facility

MVC: Model View Controller design pattern

OCL: Object Constraint Language

OMG: Object Management Group

PIM: Platform Independent Model

PSM: Platform Specific Model

QVT: Query View Transformation language

REST: Representational State Transfer

RMM: Richardson Maturity Model

URI: Uniform Resource Identifier

XMI: XML Metadata Interchange

XML: Extensible Markup Language

Executive Summary

The overall objective of WP2 is to provide upper and lower CASE-related functionalities to the S-CASE system in order to allow the successful, automated handling of software-artefact meta-information, as well as its transformation to a RESTful web service upon developer request.

REST architectural style has specific features a web service must present in order to be considered RESTful. Richardson's Maturity Model (RMM) captures these principal features as the decomposition of web services to simple resources that are addressed via a unique URI, the proper use of the HTTP verbs by respecting their semantics as well as the semantic interconnection of a web service's resources, known as the *hypermedia*.

Although there are numerous frameworks in the domain of RESTful development, most of them do not satisfy all of the RMM's criteria and therefore produce 2nd level RMM services. This is because, either most of them do no embed web service resources interconnection at all, or in the few cases that some *hypermedia* are introduced, are of limited use or require developer intervention.

In this context, this document presents advances and achievements within WP2 with respect to the state of the art, which are:

- The full incorporation of the REST architectural style to the S-CASE produced web services, including automated semantic interweaving of their resources with hypermedia links.
- The application of the Model Driven Engineering paradigm to RESTful services development, which provides a comprehensive mechanism that is extensible so as to support multiple platforms instead of a specific one (as most other frameworks do).
- The introduction of a 2D MDE engine extension mechanism that allows non-intrusive extensions of existing MDE engines.
- Extended functionality introduction using the 2D MDE engine within the domain of *Authentication*, *Database Searching*, interaction with *External Service Compositions* and *ABAC Authorization* mechanisms
- Transparent transformations within the MDE phases by using the declarative programming paradigm with ATL, which leads to increased visibility and traceability of the whole process. More over, with 2D MDE, the ATL transformations alongside the Ecore meta-models are compartmentalised per supported sub-domain.
- A list of fully formalized OCL constraints that provides validation capabilities for every S-CASE produced web service, in terms of compliance to the REST architectural style as well as their structural and behavioural consistency for all the extended functionality, in all the intermediary stages.

This document initially presents the state of the art regarding RESTful services development and then proceeds with the explanation of the innovative 2D Model Driven Engineering mechanism that allows S-CASE to further extend its automation capacity. Then it defines the Core CIM meta-model, which comprises a set of *Resources* that may have *Properties*, *Representations* and *CRUDActivities* following the REST architectural design style. This meta-model forms the base upon which the presented *Authentication*, *Database Searching*, *External Service Compositions* and *ABAC Authorization* functionality layers build in order to increase the fully automatable subset of services by the S-CASE MDE engine.

Thereafter, all the Model-to-Model transformations within the current 2D MDE mechanism are defined, namely Core CIM to Core PIM, CIM to PIM and PIM to PSM extension transformations. Finally, it documents a list of accompanying Acceleo templates that form the code generation engine of the S-CASE MDE engine.

1 Introduction

1.1 WP2 - Task 2.3 Description and Objectives

The overall objective of WP2 is to provide upper and lower CASE-related functionality to the S-CASE system in order to allow the successful storing and querying of meta-information for software artefacts, as well as the transformation of this meta-information to RESTful web services upon developer request. The most important objectives of WP2 as a whole are to:

1. Define the structure and constraints of the Platform Independent Model (PIM) following the MDE paradigm so as to be able to create the abstract design of envisioned RESTful services that satisfy the multi-modal user requirements.
2. Define the structure and constraints of the Platform Specific Model (PSM) based on the MDE paradigm so as to be able to specify the abstract PIM design to a concrete set of technologies that will implement the desired functionality.
3. Design the interfaces to the S-CASE ontology in order to support automated storing and retrieval of software artefacts to be used for querying, matching and mining tasks.
4. Develop an automated CIM to PIM Model-to-Model (M2M) transformation mechanism.
5. Develop an automated PIM to PSM M2M transformation mechanism.
6. Develop an automated PSM to code Model-to-Text (M2T) transformation mechanism.
7. Develop mechanisms for transforming 3rd party services into models and into S-CASE services.
8. Design and develop mining mechanisms for discovering associations between models.

This deliverable is primarily concerned with the objectives of Task 2.3. These are objectives 3, 4 and 6 from the above list. Specifically, the goal of this task is to define the (formal) scheme for generating the abstract S-CASE Computational Independent Model, as well as the M2M transformation to the corresponding Platform Independent Model, which is already defined in D.2.2, and the automated production of code. The S-CASE CIM and the transformation mechanism to its according PIM, will serve as **a single point of reference for mapping requirements to functionality, denoting associations between related software artefacts, and describing web service architecture details** within the S-CASE framework. The CIM has been designed in order to **foster the incorporation of all needed information** as described by the S-CASE ontology and as imposed by the abstract RESTful Services concept domain, while the PIM design introduces the abstract envisioned system's design that follows the REST Architectural style. Finally the PSM design fully specifies the PIM one with a **set of concrete widespread technologies and standards**, in order to lead **to automated code generation**.

Moreover, this deliverable presents a set of meta-models that extend the Core REST CIM, PIM and PSM meta-models that are either defined in this document or have been defined in D2.2. These extensions will allow the S-CASE MDE engine to embed to the envisioned systems authentication mechanisms, database searching capabilities, cooperation with 3rd party external compositions of web services and Attribute Based Access Controller Authorization mechanisms. As a result, the automatically produced envisioned systems will satisfy real world needs within the RESTful services realm.

Regarding the formal scheme of the CIM, PIM and PSM models, this deliverable introduces **thirteen Ecore meta-models that enable the interweaving of parsed software artefacts so as to produce a RESTful web service and its data storage**. The outcome of this task along with the outcome of task T.2.2 provide the basis for scaffolding modern RESTful services within S-CASE framework.

1.2 Deliverable Structure

This deliverable is divided into ten main sections. The first three lay the ground regarding terminology and context that is used in the rest sections, where the CIM, PIM, PSM, their 4 layers of extensions and the transformation between them are fully defined.

More specifically, section 1 discusses the general goals of WP2 of the S-CASE project as well as the more specific goals related to Task 2.3 and this deliverable. Then, it identifies the intended audience of this document and the reading guidelines to various stakeholder groups. Section 2 presents state-of-the art frameworks for RESTful web services development and discusses the way the S-CASE MDE engine is related to them, as well as how our work advances beyond the state of the art. Next, section 3 completes the introductory part by presenting the MDE paradigm as applied by S-CASE. Additionally, it provides a brief outline of the various frameworks used in PIM, PSM and PIM to PSM transformation definitions. Finally, section 3 introduces the way additional layers of functionality are incorporated in the existing S-CASE MDE engine so as to expand its automation capabilities.

The second part of this document, sections 4, 5 and 6, delve deep into the definition of the Task 2.3 related artefacts. Therefore, section 4 fully defines the Core CIM meta-model both structurally and behaviourally, taking into account the design goals and restrictions imposed by the S-CASE project. Moreover, section 4 fully defines the *Authentication*, *Database Searching*, *External Service Composition* and *ABA Authorization* Core CIM extension layers. Section 5 builds upon the Core PIM meta-model that has been presented in D2.2 and defines its extensions to provide the abstract envisioned system's design that embeds the extended functionality of Core CIM extensions. These are also defined structurally and behaviourally. Finally, section 6 fully defines the analogous Core PSM extensions and their relations to it, as it has been defined in D2.2.

The third part of this document comprises sections 7, 8, 9 and 10. These sections describe in detail the way any parsed software artefacts in the S-CASE ontology is transformed so as to produce the desired system. Section 7 defines the structure of the ontology output in yaml format, which is then parsed by the MDE engine in order to produce the envisioned systems CIM. Then, section 8, defines all the ATL transformations needed to create the Core PIM, the Authentication PIM extension, the Database Searching PIM extension, the 3rd party service composition PIM extension as well as the ABAC authorization PIM extension. Followin, section 9 defines the analogous steps, for both the core and extension functionality layers, in order to produce the envisioned systems PSM. Finally, section 10 discusses the way the S-CASE MDE engine uses the Acceleo code engine, in order to produce the envisioned system's code as well as the general output project structure.

The document concludes with a list of references and several appendices that point to the actual implementations of the CIM, PIM and PSM meta-model definitions and their extensions, which are introduced in sections 4 and 5 and 6, as well as the full transformation code following the ATL declarative paradigm and Acceleo template code generation.

1.3 Intended Audience

This document is considerably technical and therefore it primarily targets stakeholders interested in understanding the way S-CASE automates the development of RESTful web services or even extending it.

Readers who wish to gain an overview of the goals related to WP2 of the S-CASE project should mainly focus on the introductory sections 1, 2 and 3, which discuss the overall state of the art on RESTful services development, an overview of the way S-CASE attempts to automate it as well as the way S-CASE goes beyond the state of the art regarding RESTful services semi-automated production.

Readers who wish to understand the CIM, PIM and PSM meta-models should focus on Sections 4, 5 and 6 respectively, since they define syntactically and semantically all the 13 introduced meta-

models. Moreover, anyone who wishes to define and implement a new PSM meta-model in order to expand the pool of the S-CASE target execution platforms, and thus its user groups, should focus on understanding the PIM meta-model definition extensions on top of the Core PIM meta-model definition that has been presented in D2.2. The full Ecore/OCL definition of the CIM, PIM and PSM meta-models, as well as the full ATL transformations definition and Acceleo code templates that can be found in appendix should be of great help for anyone in this direction.

2 Related work

This section provides an overview of the related projects and frameworks for RESTful services development. Initially it provides a short description of the REST architectural style so as to lay the minimum background in respect to which other tools and S-CASE's MDE engine should be compared. The last subsection provides an overview of the way S-CASE relates to these frameworks and of the relevant advances made against the state of the art regarding WP2.

2.1 REST Architecture

In the year 2000, Roy Fielding introduced a new web architectural style in order to cope with the increasing complexity and heterogeneity of web services. The *Representational State Transfer* or *REST* [1] exhibits some distinct characteristics against which a service can be checked for compliance. *Richardson's Maturity Model* (RMM) [10] captures these principal REST characteristics that a web service must have in order to be considered RESTful. According to RMM, a web service is RESTful if:

- It comprises resources each of which is identified by a unique URI. This means that any two distinct resources must have different URIs. This REST constraint refers to level 1 of RMM (Figure 2-1).
- Every such URI is accessed with the common HTTP verbs and these verbs must be used in the intended way. This means that in any case, the POST HTTP verb is used to create a new resource, the GET verb is used to retrieve an existing resource (or a list of resources), PUT is used to create or update an existing resource and DELETE to delete one. This REST constraint refers to level 2 of RMM.
- Thirdly, the service resources are interconnected with hypermedia links, which provide the clients the next possible actions from any given point of interaction with it. That is, once a client makes a request to a RESTful service, along with its response, it will receive a list of links that allow the client to ponder its next options without prior knowledge.

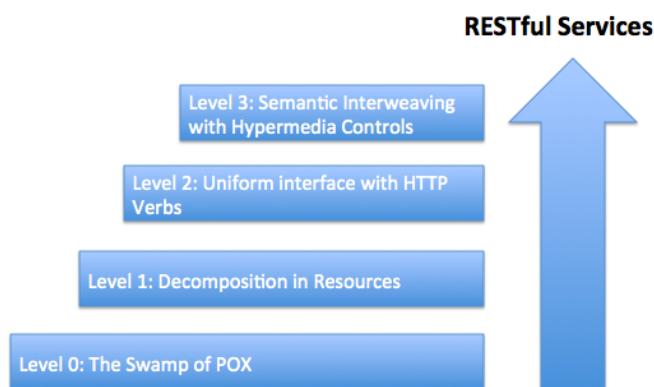


Figure 2-1 Web service friendliness Richardson's Maturity Model

2.2 Ruby on Rails

Ruby on Rails (Rails) [2] is an open-source framework for building web applications with the *Ruby* programming language. It was initially released in 2005 and has evolved and gained a lot of popularity since then. Some of the main features that make Rails popular are:

- The Ruby language is of simple nature, which makes it an easier starting point against more complex languages such as PHP and Java.
- It embeds some well-known software engineering paradigm/patterns such as the MVC pattern, the active record pattern and the convention over configuration paradigm.
- It is open source.
- It provides speed and agility, lowering costs and time needed to build services

Ruby on Rails

Web Applications

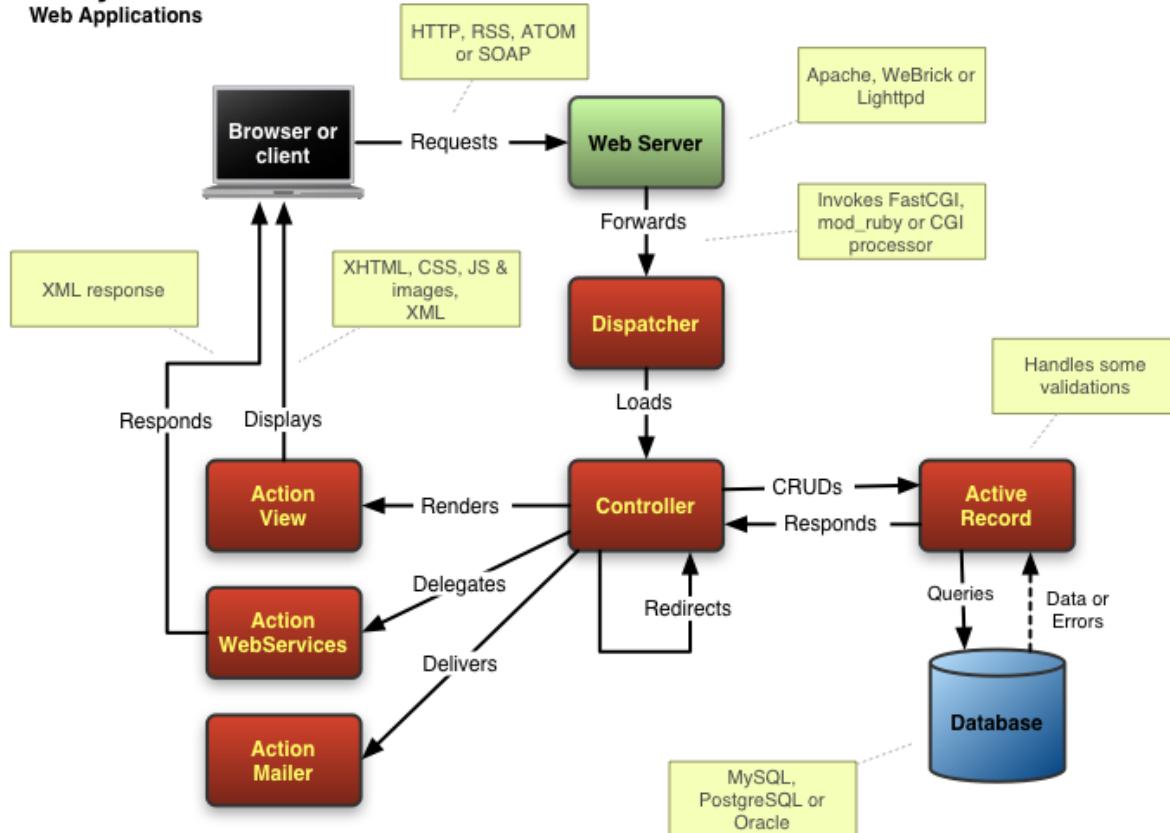


Figure 2-2 Basic structure of a Rails service (<http://binaryhash.com/ruby-on-rails>)

As Figure 2-2 demonstrates, the core of a Rails service is built around the MVC design pattern. Some of its core elements are therefore *Models*, *Controllers* and *Views*. Models store data and map to tables of the service database. Controllers receive and respond to external requests by providing a set of actions. Although not compulsory, Rails emphasizes RESTful actions, which means that controllers preferably respond to create/new, edit/update, destroy, show/index. Finally, the *Views* are by default *erb* files, which are compiled to HTML.

Ruby in Rails has also received some criticism primarily regarding performance and scalability, which has made some companies switch from Rails implementations to other frameworks/solutions. However, it still retains a substantial portion of the RESTful development framework pie and some well-known web services use it, at least at some level, such as Twitter, Groupon and Scribd.

2.3 EMF-Rest

EMF-Rest [3] is an Eclipse-based framework that enables web service development by using Ecore meta-models. As Figure 2-3 demonstrates, the developer starts by modelling his/her web service using the Ecore meta-model in the form of the familiar class diagram. Once this is done, the EMF-

Rest framework scaffolds the corresponding service and provides a RESTful API by means of JAX-RS, JSON serializers and JavaScript API. Figure 2-4 demonstrates a RESTful API produced from an example “Family” Ecore meta-model, whilst at the left there is a JSON response produced by such a service. As it is observed in the JSON response EMF-REST produces no Hypermedia Controls and therefore, it does not achieve the maximum web-friendliness as it is defined in RMM.

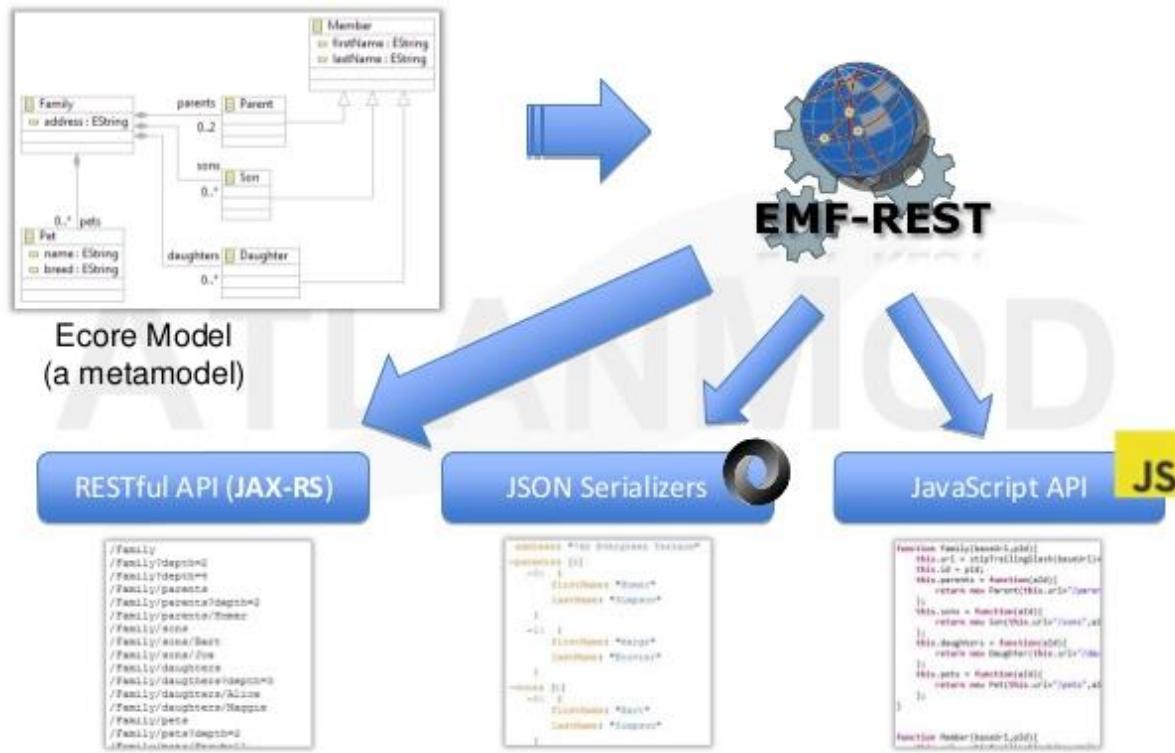


Figure 2-3 AtlanMod overview presentation of EMF-REST at EclipseCon Europe 2013 Symposium



Figure 2-4 Left: example JSON response. Right: example of RESTful API

2.4 Django – REST-framework

Django-REST-Framework [4] is another web application framework for scaffolding REST web services. This one is based on the python language. The principal elements/activities that a developer has to set-up in order to build a REST API are *Models*, *Serializers*, *Views* and naming URLs. In the rest of this subsection an overview of the development process using Django-REST-Framework is provided.

Initially, the developer has to define some sort of *Model* that represents the data with which the REST interface will interact with and then creates some serializers to serialize/deserialize it (Figure 2-5). Then the developer has to define views, which define the way clients can interact with the API (requests/responses). The final step in order to include *hyperlinking* is to setup a python file with URL templates to be used. Figure 2-6 demonstrates a server response to a GET request. As it is stated in the documentation though, Django does not automatically produce Hypermedia Controls.

```
from django.contrib.auth.models import User, Group
from rest_framework import serializers

class UserSerializer(serializers.HyperlinkedModelSerializer):
    class Meta:
        model = User
        fields = ('url', 'username', 'email', 'groups')

class GroupSerializer(serializers.HyperlinkedModelSerializer):
    class Meta:
        model = Group
        fields = ('url', 'name')
```

Figure 2-5 Django-REST-Framework serializers

```
http http://127.0.0.1:8000/snippets/
HTTP/1.1 200 OK
...
[
  {
    "id": 1,
    "title": "",
    "code": "foo = \"bar\"\n",
    "linenos": false,
    "language": "python",
    "style": "friendly"
  },
  {
    "id": 2,
    "title": "",
    "code": "print \"hello, world\"\n",
    "linenos": false,
    "language": "python",
    "style": "friendly"
  }
]
```

Figure 2-6 Service response to a GET HTTP request

2.5 Other Evolving Frameworks

Apart the aforementioned, well established, related frameworks, there are numerous others that more or less aspire to automate the production of web services, either RESTful or not. One such framework is Cloudfier [14]. Cloudfier is a platform for rapid development and deployment of business apps. Its meta-modeling arsenal relies on the structural and behavioural expressiveness of UML. Actually, Cloudfier programs are UML models in an executable textual form. Although still in pretty early stage of development, Cloudfier aspires to provide:

- Business centric development
- Instant prototyping for early concept validation
- Gapless code generation
- Online, open source, development platform

Another framework that is currently in early development stage is *thegrid.io* [15]. Thegrid.io uses graphical layout filters backed up with AI techniques in order to automate the website design without requiring coding skills. The produced websites are supposed to be self-evolvable and adaptable depending on the multi-modal media that the user wants to host in his website. However, this framework is still in an early stage without a significant release.

2.6 S-CASE MDE Engine Beyond the State of the Art

The previous sections presented an overview of an indicative subset of the major attempts to facilitate the development of RESTful services that have been made so far. Some of them tend to achieve higher levels of web-friendliness in terms of RMM, whilst others attempt to achieve higher automation levels. The three indicative frameworks presented, as well as most of the other major attempts, belong to the second level of the RMM since they break down the service into a set of resources, each of which is addressed uniquely with a URI, and respect the semantics of the HTTP verbs as well. However, in most cases they do not interweave their resources with hypermedia controls and thus fail to achieve the maximum web-friendliness. Even in the few cases in which hypermedia links are introduced in some way, their usage is either limited or requires developer intervention since it is not automated.

In terms of automation, most of the existing frameworks either produce a REST API as a database front end or yield a basic structure to which the developer has to add a great part of functionality manually. The automated part usually embeds the needed functionality to handle data from/towards the underlying service database, lacking behavioural aspects.

In contrast, since the S-CASE goal is to automate the development of RESTful services as much as possible, a lot of effort has been made to ensure that the produced services will belong to the 3rd level of RMM and the automated functionality will not be limited to an underlying database handling. In this context, the achievements of the designed and developed S-CASE MDE engine in respect of the state of the art are:

1. The full incorporation of the REST architectural style to the produced S-CASE web services. The generated resources are semantically interwoven to each other with hypermedia controls. Thus, whenever clients interact with them, they receive, additionally to the expected HTTP response, a complete list with hypermedia links that list all the possible next actions that the client can take in its next request. These links, according to the definition of the S-CASE MDE meta-models, include the URI of every resource with which an interaction

has become viable. Additionally, they include the HTTP verb, which can be used to interact with each resource, as well as the relation of every target resource with the action that has been just taken.

2. The application of the Model Driven Engineering paradigm to RESTful services development. That is, the MDE engine of S-CASE follows all the MDE steps (presented in section 3) and thus embodies the corresponding benefits. The S-CASE MDE engine provides a comprehensive mechanism that is extensible to support multiple platforms instead of a single combination of some programming language and two or three 3rd party frameworks like most other frameworks do.
3. Moreover, the S-CASE MDE engine introduces an innovative meta-model extension mechanism that allows expanding the automation capabilities of it without modifying the original meta-models. As explained in detail in section 3, instead of modifying the original meta-models and risking introducing errors and complexity to them, a two dimensional MDE paradigm is proposed, where every extension is built on top of the existing core.
4. Due to the 2D MDE mechanism, the S-CASE MDE engine is able to automatically embed to the produced envisioned systems other than basic database handling and REST API, also authentication, database searching, 3rd party service compositions and ABAC authorization functionality.
5. The building of transparent transformations within the MDE phases by using the declarative programming paradigm with ATL leads to increased visibility and traceability of the whole transformation process.
6. Moreover, the 2D MDE mechanism, which this document introduces, allows the ATL transformations to be broken down to smaller and simpler ones that extent the core functionality without the need to modify the existing ones.
7. The developed meta-models for CIM, PIM and PSM as well as the accompanying four layers of functionality extensions embed a formalized set of OCL constraints that provide validation capability for every produced web service by the S-CASE platform in terms of compliance to the REST architectural style, as well as structural and behavioural consistency of all the intermediary and final artefacts.

3 S-CASE MDE Engine Introduction

3.1 MDE Process Overview

This subsection introduces the cornerstone technology adopted in S-CASE for automation, which is the *Model Driven Engineering* or in short *MDE* [5]. The Object Management Group (OMG) has introduced MDE since 2000, which aspires to change the software design and construction paradigm. Its principal concept is the transition of code-centric software engineering to a model-based one, so as to achieve higher consistency, increased level of automation and productivity. In a nutshell, it aspires to deliver software in less time and at a lower cost. Within S-CASE, MDE is one of the technologies used in order to automate the production of RESTful services. MDE comprises four distinct phases each of which is introduced and briefly explained subsequently.

During the first phase of MDE the developer formulates the *Computational Independent Model* (CIM). This model is the most abstract view of the underlying system. It comprises strictly application domain concepts, properties and entities. During this phase any design, architecture and implementation details are irrelevant. That is, the developer models the system to be built in terms of abstract domain concepts, their properties and relations. In S-CASE the domain of application is the web services that follow the *REST* architectural style (RESTful domain). Therefore, the principal concepts of the S-CASE MDE CIM are *resources*, *web interfaces*, *resource properties*, *resource relations* etc. The CIM meta-model of the S-CASE MDE engine will be defined in deliverable D2.3 *Ontology to MOF models transformation*.

Once the CIM of an envisioned service is formulated, a “model to model” (M2M) transformation takes place that transforms the CIM into the corresponding *Platform Independent Model* or PIM. The PIM introduces the abstract design of the system without any implementation details. Therefore, during the transformation that takes place from the CIM to the corresponding PIM, the abstract architecture of the system is applied to every CIM concept in order to form the complete design that satisfies the system requirements. Section 4 of this document fully defines the language of S-CASE MDE’s PIM, the *PIM meta-model*, along with the architecture it introduces.

The third phase of the MDE process involves the transformation of the PIM to one of its possible *Platform Specific Models* (PSM). The PSM conforms to the design that PIM introduces but specifies the way it will be implemented by introducing specific implementation language details, as well as several external frameworks, libraries and/or programming language specific design patterns or idioms. All these together form the target platform that the PSM models. Thus, the PIM to PSM transformation specifies each PIM element using the concrete technologies that form the PSM. It must be noted that there is a one-to-many relationship between a PIM model and the compatible PSMs. That is, every abstract design of a system that the PIM introduces can be implemented with numerous distinct mixtures of specific technologies that form different PSMs. Section 5 of this document fully defines the language of S-CASE MDE PSM, the *PSM meta-model*, along with the concrete technologies that are used in it. Thereafter, section 6, conceptually defines the PIM to PSM M2M transformation that is implemented with the *Atlas Transformation Language* (ATL).

Finally, the MDE engine produces the code by filling code templates with the meta-data that is available in the PSM. Some parts of the code may be fully implemented, whilst others may need extra intervention by developers. The code templates in par with the code generator of S-CASE MDE engine will be defined in D2.3 along with the CIM. Figure 3-1 summarizes the MDE phases and transformations.

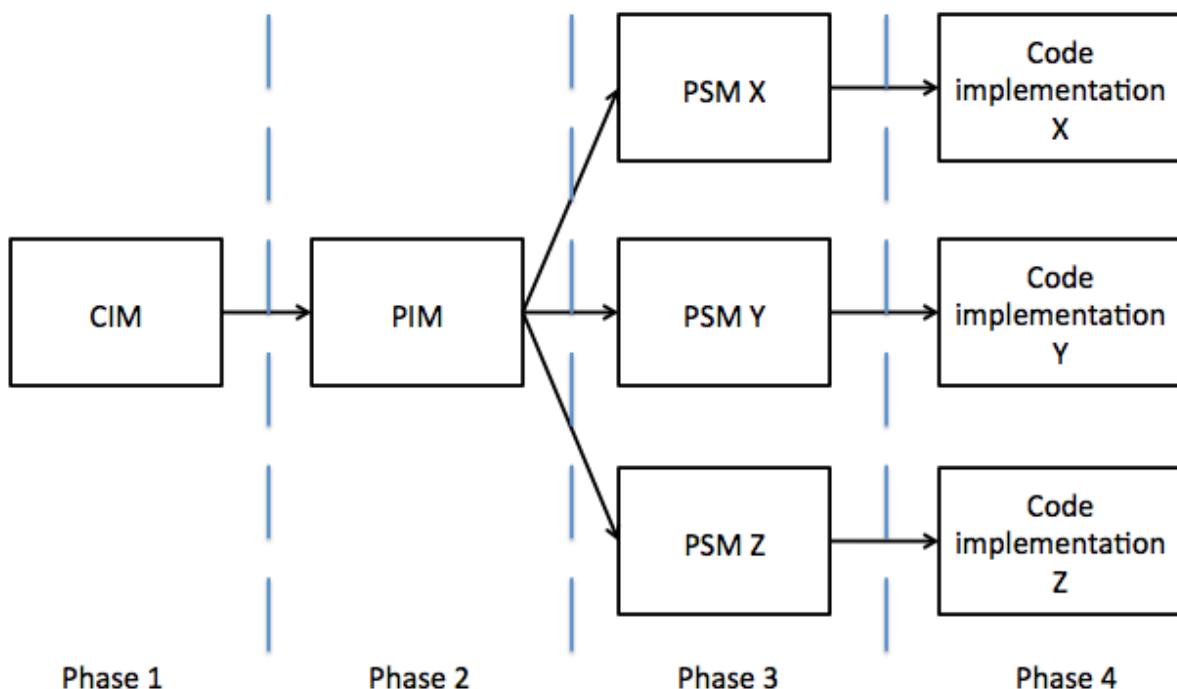


Figure 3-1 The four phases of Model Driven Engineering

3.2 Associated Technologies and Frameworks

This section briefly presents the technologies used to define and implement the PIM and PSM meta-models as well as the M2M transformation from the PIM to the PSM.

3.2.1 Ecore meta-model

In the definition of *Meta-Object Facility 2.0* MOF, apart from the *Complete Meta-Object Facility* or *CMOF*, OMG introduced an “essential” subset of it, the *Essential Meta-Object Facility* or *EMOF* that aims to model primarily structural aspects of systems, whilst the CMOF models behavioural ones as well. The Eclipse framework implementation of the EMOF is the Ecore meta-model [6]. The Ecore meta-model is the cornerstone of many modelling applications and plugins within the Eclipse ecosystem.

Since S-CASE aims to offer developers tools within the Eclipse framework, the Ecore meta-model has been selected as the core upon which the S-CASE profiles will be created. That is, the CIM, PIM and PSM meta-models are all extensions of the Ecore meta-model. Therefore, S-CASE produced artefacts can be used with other tools and Eclipse plugins that conform to Ecore. This selection impacts positively on the potential S-CASE users group due to the large Eclipse user pool.

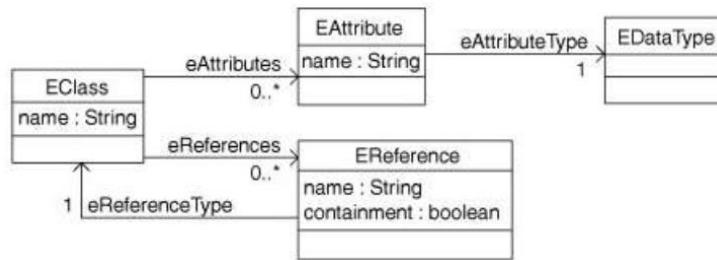


Figure 3-2 Simplified subset of the Ecore meta-model

Figure 3-2 depicts a simplified subset of the Ecore meta-model. The *EClass* element is used to represent a modeled class. It has a name, and zero or more *EAttributes* or *EReferences*. The *EAttribute* is used to represent a modeled attribute of an *EClass* and has a *name* and a *type*. Thereafter, *EReferences* represent associations between *EClasses* either of containment type or not and lastly, the *EDataType* represents either primitive or custom datatypes. The S-CASE CIM, PIM and PSM meta-models extend these Ecore elements in order to define the S-CASE profiles.

3.2.2 Object Constraint Language (OCL)

The *Object Constraint Language* (OCL) [7] is a formal language used to describe expressions on MOF models. Such expressions typically describe invariant constraints that must hold within the system being modelled or queries to its objects. OCL can be used primarily:

- As a query language
- To specify invariants on classes and types of a class model
- To describe pre- and post-conditions on operations and methods

Within the S-CASE MDE engine, OCL is used to specify invariants that form concrete validation schemes of CIM, PIM and PSM meta-model instances. Sections 4 and 5 of this document specify all the OCL constraints that an instance model has to conform to in order to be a valid one of the S-CASE MDE PIM or PSM meta-model respectively.

3.2.3 Eclipse Modelling Framework (EMF)

The *Eclipse Modelling Framework* (EMF) [6] is a framework that unifies many of the Eclipse ecosystem facilities. As Figure 3-3 depicts, EMF unifies the *Java* language with the *XML* specification and the *UML* one as well. This means that instances of any of the three languages can be interpreted transparently to any of the others offering thus, a robust, reliable connection among different representations of the same model. Therefore an Eclipse EMF user does not need any more to, for example, manually convert *UML* diagrams to *Java* instances and therefore risk introducing inconsistency among them.

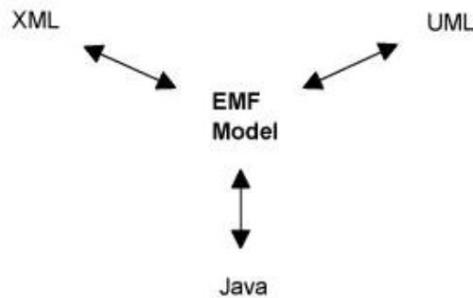


Figure 3-3 The Eclipse Modelling Framework unifies Java, XML and UML

Among others, EMF embeds a model validation scheme against an Ecore meta-model. The MDE engine of S-CASE, therefore, uses the EMF in order to validate instances of CIM, PIM and PSM meta-models. This happens against both structural and behavioural aspects. The structural ones are imposed by the structure of the CIM, PIM and PSM Ecore meta-models of S-CASE, whilst the behavioural ones are embedded in them as OCL invariant constraints using the OCLinEclipse facility.

3.2.4 Atlas Transformation Language (ATL)

The *Atlas Transformation Language* (ATL) [8] is a hybrid language (declarative and imperative) that is used in M2M transformation definitions. The AtlanMod research group introduced ATL as a response to the OMG's QVT request for proposal. In the context of MDE, ATL provides developers the means to specify the way to produce a number of target models from a set of source models. Figure 3-4 demonstrates the mechanism by means of which the ATL language performs M2M transformations.

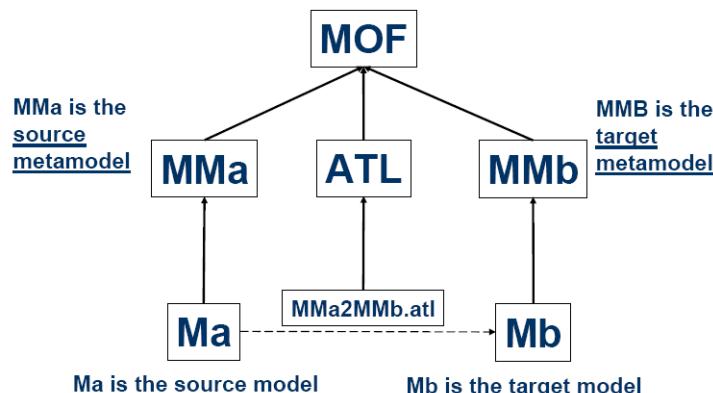


Figure 3-4 ATL Operational Context

ATL conforms itself to the MOF meta-meta-model and performs transformations among models that are instances of meta-models that also conform to MOF. For example, following a set of ATL rules, a source model *Ma*, which conforms to the *MMa* meta-model, is transformed, element-by-element, to an *Mb* model that conforms to the *MMb* meta-model. In the same way S-CASE MDE Engine uses sets of ATL rules to transform elements of CIM, PIM and PSM meta-models to elements of other models. This is possible since the CIM, PIM and PSM meta-models are defined using the Ecore meta-model, which is an implementation of the EMOF that in turn is a subset of CMOF. Section 6 of this document conceptually defines all the implemented ATL rules in order to perform the PIM to PSM transformation within the S-CASE MDE engine.

ATL as a hybrid language provides a variety of ways to define such rules. These are:

- **Matched Rules:** The matched rules are declarative rules. Such rules enable matching on some elements from the source model and generate a number of distinct target model elements. The ATL transformation engine calls such rules implicitly whenever a source model element, which is “matched” in a rule, is found and then the target element is generated. In order to provide some execution order control, ATL provides also a subcategory of the matched rules, the *Lazy* ones. These are also declarative, but are called only by other rules explicitly. Both of them are usually more suitable in order to perform transformations to target model elements that do have source model counterparts.
- **Called Rules:** The called rules on the other hand are imperative rules. These rules must be invoked by other rules and do not have any source element as input. Such rules are usually suitable in order to perform transformation rules that generate target model elements that do not have source model counterparts.

3.2.5 Acceleo code generation framework

Acceleo is a pragmatic implementation of the OMG MOF Model To Text Language (MTL or M2T) standard. More specifically it is a template language based on an OCL superset that allows the developer to produce generic code templates. These templates take as input one or more models that comply with one or more meta-models and have all the needed information to fill in the templates. One such code template is demonstrated in Figure 3-5 below.

```
[for (jRModel : JavaResourceModel | anAnnotationStack.hasCorePSM.hasJavaRModel)]
import eu.fp7.scase.[anAnnotationStack.hasCorePSM.name.toLowerCase()/.[jRModel.parentName/].[jRModel.name/];
[/for]

/* This class follows the singleton pattern in order to build once and provide a unique hibernate session instance*/
public class HibernateUtil{

    private static final SessionFactory sessionFactory = buildSessionFactory();

    private static SessionFactory buildSessionFactory(){
        try {
            /* Create the unique hibernate session. All resource models should be added here.*/
            return new AnnotationConfiguration().configure()
[for (jRModel : JavaResourceModel | anAnnotationStack.hasCorePSM.hasJavaRModel)]
            .addAnnotatedClass([jRModel.name/].class)
[/for]
            .buildSessionFactory();
        }
        catch (Throwable ex){
            System.err.println("Initial SessionFactory creation failed." + ex);
            throw new ExceptionInInitializerError(ex);
        }
    }
}
```

Figure 3-5 Abstract from an Acceleo code template used in S-CASE MDE engine to produce code

3.3 S-CASE 2D MDE Engine Architecture

As section 3.1 has already presented MDE aspires to increase the level of automation in software production so as to increase productivity and consistency while lowering the time and cost needed to produce software. However, since the product of MDE is some sort of software, both the outcome products as well as the MDE mechanisms are exposed to continuous needs for expansion and evolution to accommodate evolving needs in the domain they are applied. This implies the need to continuously modify any MDE artefacts used in the MDE mechanism, namely meta-models and therefore Model-To-Model transformations as well. Moreover, these modifications and expansions to the initial MDE artefacts entail the risk of introducing inherent complexity of any all in one approach and errors, which highlights the need to go through a new debugging and testing cycle to reconfirm proper functioning of the MDE mechanism.

This section introduces a new MDE mechanism expansion approach that embeds the following quality design characteristics:

- Genericity: The extension mechanism is not limited to the S-CASE MDE engine meta-models but can be applied to any existing meta-model in order to extend or alter the functionality and domain knowledge it embeds.
- Non-intrusiveness: The extensions made with the proposed mechanism do not alter the original meta-model and the original Model-to-Model transformations. This way, a new round of testing and debugging cycle of the extended meta-model and its Model-To-Model transformations is not needed.
- Backwards Compatibility: The fact that the extension mechanism is non-intrusive implies that it remains compatible after the extensions with any existing software systems that are built using the original meta-models and their Model-To-Model transformations. Thus, there is no need to go through an update cycle of all the existing systems' models, since the MDE mechanism is still able to use them as before.
- Dynamic utilization and reuse: However, if updating an existing system with the new functionality is desirable, the MDE mechanism user needs to fill in only the extension of the original meta-model whilst the initial model is remains and is used unchanged.
- Rollback capability: Additionally, if after updating the user decides to rollback to the initial version, he only needs to re-run the MDE mechanism with the original model as input.
- Design-level Domain Compartmentalisation: In terms of development and MDE mechanism design, the 2D MDE approach this section presents, allows the MDE mechanism architect to focus on the design and the implementation of each sub-domain introduced separately. That is, in S-CASE case, the introduced extensions of the original meta-models are done by introducing *EMF/Ecore* models and ATL transformations that are only a fraction in terms of size and complexity compared with the original meta-model. Therefore, the whole process of meta-model extension is conceptually easier to manage.
- Usage-level Domain Compartmentalisation: In terms of using the MDE mechanism, the 2D MDE approach this section presents, allows both the MDE mechanism user, to focus on each functionality sub-domain introduced separately. That is, in S-CASE case, a REST expert could design the core of the envisioned system by creating an instance of the Core CIM model (introduced in chapter 4), whilst a security expert can then focus only on the automated application of an authentication and an ABAC authorization mechanism. This implies that the level of expertise needed to use the MDE mechanism is lowered since there is no need for cross-domain experts, making it thus usable by a wider range of developers.

The main concept of the MDE engine extension mechanism is the use of annotations to existing concepts of the original meta-models. These annotations can either alter the meaning and/or the design of existing concepts or introduce new concepts that should be related with existing ones expanding thus the supported functionality. These extension meta-models have the general conceptual design, which is demonstrated in Figure 3-6. So in general, every *AnnotationModel* has several *AnnotatedElements*, which are references to original meta-model concepts, and *Annotations* that are associated with those *AnnotatedElements* in order to either alter their meaning, structure and/or introduce related new concepts.

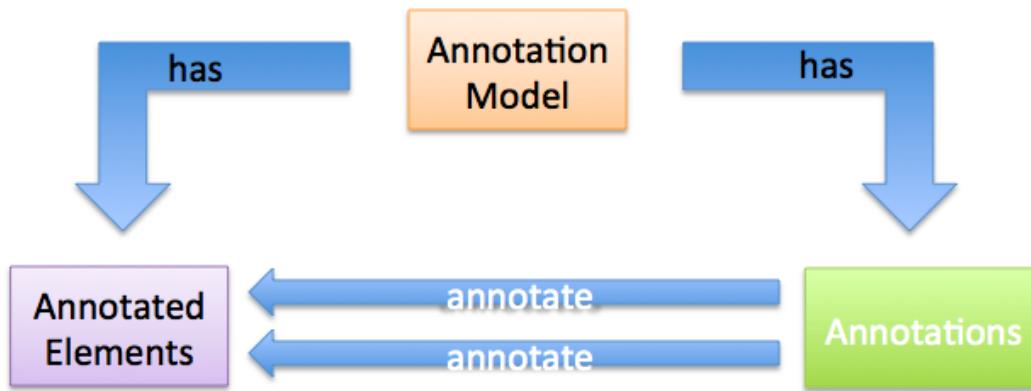


Figure 3-6 Conceptual design of an extension meta-model

Figure 3-7 demonstrates one such extension of the original S-CASE CIM meta-model that is introduced in chapter 4 of this document. In this case, it is the *External Service Composition* CIM extension meta-model, which is defined in chapter four of this document. This extension meta-model has two *AnnotatedElements*, the *AnnCRUDResource* and *AnnAlgoResource* ones. These reference (both in this case) the *Resource* concept of the original Core CIM meta-model. However each of these reference a different subset of *Resources*.

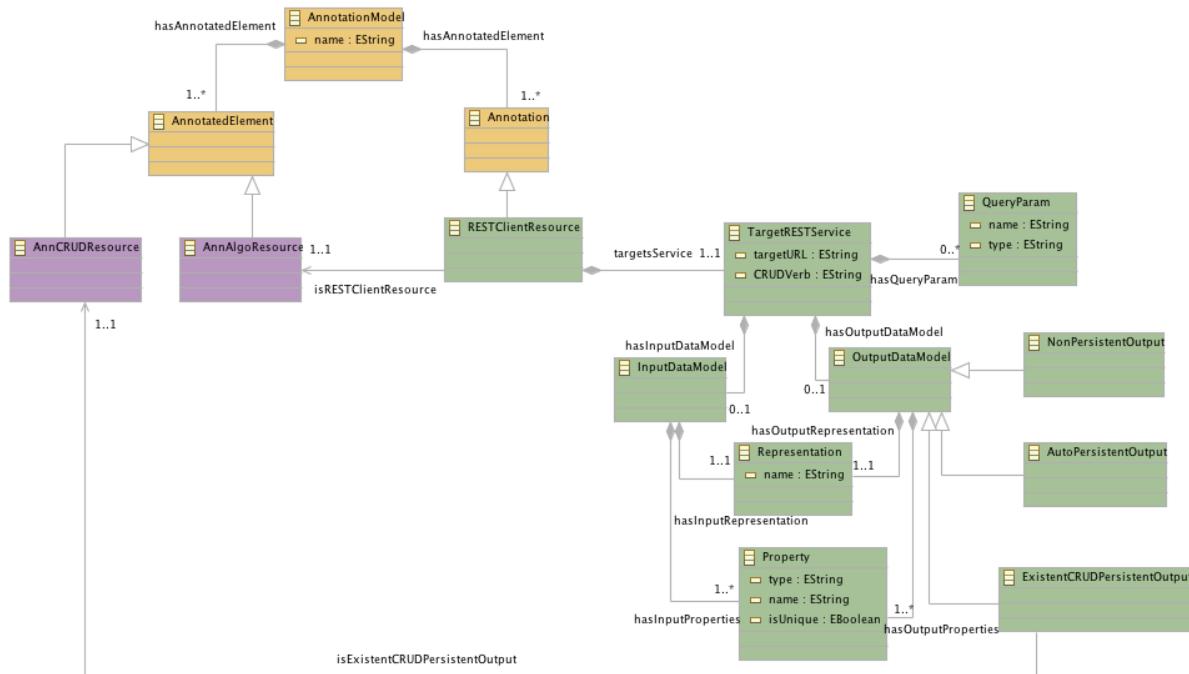


Figure 3-7 Indicative Core CIM extension meta-model.

Furthermore, this extension meta-model introduces one *Annotation* element, which expands the structure and the meaning of the referenced Core CIM *Resource* concept. In this case, the annotated *Resource* will be handled by the MDE mechanism as a *RESTful* client that will be able to interact with a composition of external services. The needed information to interact with that composition,

accompany the *RESTClientResource Annotation* element. That information includes the *URL* at which the service composition is reachable alongside with the verb that should be used to interact with. Additionally, there may exist several query parameters that have to be used in the request and some sort of input/output models of data (*InputDataModel/OutputDataModel*) that should be formatted using a specific media format (*Representation*). This way, whilst the original Core CIM meta-model did not embed the needed information for a part of the envisioned system, with this extension any envisioned system is able to have one or more *Resources* that interact with external service compositions.

Since the S-CASE MDE engine follows the MDA paradigm, for every functionality layer that is added, one corresponding CIM, PIM and PSM extension model is added, which annotates the original Core CIM, PIM and PSM meta-models. Therefore, four CIM, PIM and PSM extensions are presented in this document to model the *Authentication*, *Database Searching*, *External Service Composition* and *ABAC Authorization* extensions on top of the original Core CIM, PIM and PSM meta-models.

Figure 3-8 demonstrates conceptually the whole transformation process of the 2D MDE mechanism. One may observe that the four MDA phases are still present, namely the CIM, PIM, PSM and code generation phase. However, as previously explained, each phase has multiple meta-model artefacts. For example within the CIM zone exist the Core CIM meta-model and its CIM Extension A. In a similar way, PIM zone comprises Core PIM meta-model alongside its PIM Extension A and PSM zone comprises Core PSM meta-model and its PSM Extension A.

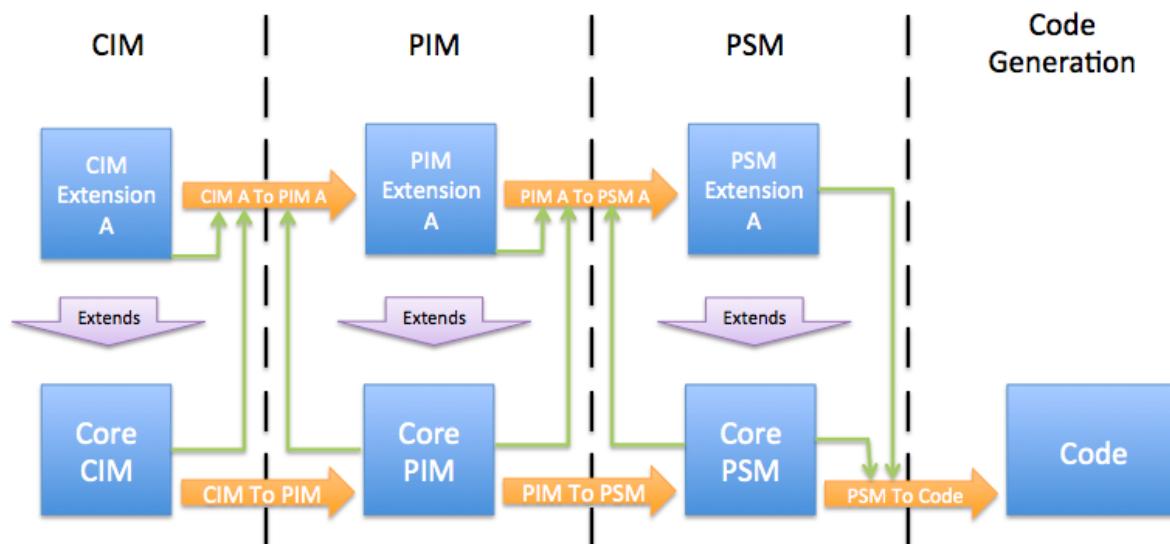


Figure 3-8 Transformations within the 2D MDE mechanism

The orange arrows model the ATL Model-To-Model transformations within the 2D MDE mechanism, whilst the green arrows model the input models of each extension transformation. The transformations of the Core CIM to Core PIM and Core PIM to Core PSM models remain unchanged. The extra transformations needed are completely independent from the original ones. For example, the ATL *CIM A To PIM A* transformation takes as input the Core CIM model of the envisioned system along side its *CIM Extension A*, in order to create the *PIM Extension A* model, and the Core PIM model in order to create the needed references from *PIM Extension A* model's *AnnotatedElements* references to the annotated concepts of the Core PIM's model. The same applies for the PIM to PSM Model-To-Model transformation. In the last step, the Core PSM model of the envisioned system as well as any extension models of it, form the input package of the code generation process.

In the same manner, there could be more annotations instead of one. This is the case of S-CASE 2D MDE engine in order to be able to produce envisioned systems that embed the original core RESTful

functionality as well as any combination of the *Authentication*, *Database Searching*, *External Service Compositions* and *ABAC Authorization* extensions.

4 Computational Independent Model UML Profile

4.1 Core CIM UML Profile

4.1.1 CIM UML Profile Design Goals

The Computational Independent Model, as section 3.1 explains, comprises strictly abstract domain concepts, their properties and their relations to each other. Since S-CASE domain is the domain of RESTful services development, the principal goal of the S-CASE Core CIM meta-model is to sufficiently embed the main REST concepts and at the same time abiding by the 3rd level of Richardson's Maturity Model to ensure maximum web friendliness and compliance to REST.

Figure 4-1 demonstrates a simplified version of the abstract REST concepts that the S-CASE CIM embeds. Since REST services have as building blocks *Resources*, the core abstract S-CASE CIM concept is the *Resource* one. It models any core entity of the envisioned system. In other words it is the REST analogous of a *Class* in the object oriented design paradigm. As imposed by the RMM, each such resource has a unique *URI* by the means of which it is addressable. Depending on the case, each such *Resource* may have also some properties that model the data it encapsulates and some possible input/output representations that can serve various client needs in terms of media formats such as *application/JSON*.

Additionally, each *Resource* may have some *CRUDActivities* that are the possible actions a client of this resource can perform on its data. However since S-CASE aspires to be fully compatible with the REST architectural style, these *CRUDActivities* are restricted to be the abstract form of the HTTP verbs that are used as the common web interface of *RESTful* services. There are four such verbs, the *POST*, *GET*, *UPDATE* and *DELETE*. However since the CIM comprises of abstract concepts, independent of underlying implementation technologies, the analogous abstract forms of the *HTTP* verbs, the *CRUD* verbs, are used instead. The *CREATE* activity is the abstract form of the *HTTP POST* verb, that is used to model the action of creating a new *Resource*. The *READ* activity, as the abstract form of the *HTTP GET*, retrieves a specific *Resource*. Finally the *UPDATE* and *DELETE* activities are the abstract forms of the *HTTP PUT* and *DELETE* verbs and they model the actions of updating or deleting existing *Resources* respectively.



Figure 4-1 REST abstract concepts of S-CASE CIM meta-model

In order to comply with the 3rd principal REST architectural design, the hypermedias as the engine of application state, the S-CASE CIM meta-model also introduces *Relations* among *Resources*. These relations allow the semantic interweaving of an envisioned system that the S-CASE will produce and form the needed infrastructure so as to be able to return to each client a set of links with all the next possible actions along with the results of its request.

Therefore, since the S-CASE CIM comprises concepts that model all the needed REST architectural qualities, the S-CASE Core CIM meta-model that is structurally and behaviouraly defined in the next sections, coupled with the PIM and PSM ones that have already been defined in D2.2, satisfies the principal S-CASE requirements to be able to produce REST services.

4.1.2 CIM Ecore Meta-model Definition

4.1.2.1 Introduction

In order to fully define the Core Ecore CIM meta-model (Figure 4-2), its various aspects must be explained, demonstrated as well as documented. Each of the remaining subsections of this section precisely defines the structure of each Core CIM meta-model element, its properties and their behavioural constraints.

Each such subsection begins with a meta-model element overview that provides a high level description. Depending on the complexity of each element, one or more Ecore class diagrams may be included to clearly demonstrate the relations of it with the rest of the elements of the Core CIM meta-model.

For each element two tables are provided. The first table defines the properties of each Core CIM meta-model. Therefore, they provide the *name*, the *type* as well as an explanation of every such property. The second table defines the relations of the meta-model element. This table comprises the name of the related elements, the type of relation (e.g. composition, association etc.) and the multiplicity of that relation. The last column of such tables, defines the structural constraints of this element within the Core CIM meta-model. Finally, the last part of each element subsection includes a list of the OCL constraints conceptual descriptions, which apply to it. These constraints define the behavioural constraints of each element as well as the well-formness rules it must comply with.

4.1.2.2 CIM Ecore Meta-model Elements

This section defines all the Core CIM meta-model elements using the format that is presented in section 4.1.2.1.

4.1.2.2.1 CIM Custom Data Types Definition

This subsection defines the custom data types included in the Core CIM meta-model.

MediaType

The *MediaType* data type is an enumeration that models the possible input/output media types that are supported within S-CASE. These can be found in the following table:

Table 4-1 Supported Media Types

MediaType	Relative HTTP Header value	Default Value	Explanation
JSON	application/JSON	Yes	When the JSON media type is selected as input or output representation of a resource, its web interface expects as input data format or produces the output data respectively, in application/JSON format.
XML	application/XML	No	When the XML media type is selected as input or output representation of a resource, its web interface expects as input data format or produces the output data respectively, in application/XML format.

CRUDVerb

The *CRUDVerb* data type is an enumeration that models the four *CRUD* verbs, *Create*, *Read*, *Update*, *Delete*. They are defined in the following table.

Table 4-2 CRUDVerb Data Type

CRUDVerb	Default Value	Explanation
CREATE	Yes	The CREATE verb is used to model the abstract form of the HTTP POST verb within the CIM meta-model. Wherever it is used, it respects the REST architectural style and thus denotes the action of creating a new resource.
READ	No	The READ verb is used to model the abstract form of the HTTP GET verb within the CIM meta-model. Wherever it is used, it respects the REST architectural style and thus denotes the action of retrieving either a specific existing resource or a list of existing resources.
UPDATE	No	The UPDATE verb is used to model the abstract form of the HTTP PUT verb within the CIM meta-model. Wherever it is used, it respects the REST architectural style and thus denotes the action of updating a specific existing resource. Moreover, since S-CASE produces web services that produce automatically resource identifiers, the UPDATE verb is not used to also create resources.
DELETE	No	The DELETE verb is used to model the abstract form of the HTTP DELETE verb within the CIM meta-model. Wherever it is used, it respects the REST architectural style and thus denotes the action of deleting a specific existing resource as well as any subsequence resources related to it.

4.1.2.2 RESTfulServiceCIM Element

Overview

RESTfulServiceCIM is the root element of the CIM meta-model. There may exist exactly one *RESTfulServiceCIM* element in any CIM produced by S-CASE. It comprises the abstract form of all the elements that form a RESTful service and is associated with them by means of a composition

association. This suggests that any other element of the CIM must be contained by this specific root element. The rest of this subsection provides information on its properties, relations and their structural and behavioural restrictions. Figure 4-2 demonstrates the RESTfulServiceCIM element with its properties and relations in the context of the Core CIM meta-model. It is actually the full S-CASE Core CIM meta-model.



Figure 4-2 S-CASE Core CIM meta-model

Properties

Table 4-3 RESTfulServiceCIM 's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the provided service name by the S-CASE developer. Among others, it is used to produce the service folder structure and build files.
serviceOutputPath	EString	1	This is the desired output path of the envisioned system to be produced.
serviceDatabaseIP	EString	1	This is the database IP of the database server, which is going to be used to store the envisioned system's data.
serviceDatabasePort	EString	1	This is the database port of the database server, which is going to be used to store the envisioned system's data.
serviceDatabaseUsername	EString	1	This is the username of the user account to be used by the envisioned system in order to interact with the database server.
ServiceDatabasePassword	EString	1	This is the password of the user account to be used by the envisioned system in order to interact with the database server.

Relations

Table 4-4 RESTfulServiceCIM's Relations

Relation With CIM Element	Type	Multiplicity	Structural Constraints
Resource	composition	1.. *	The <i>RESTfulServiceCIM</i> element may be associated with one or more <i>Resource</i> elements. This relation models the fact that its CIM comprises of resources since this service is a RESTful one.

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of a *RESTfulServiceCIM* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the CIM meta-model that can be found in appendix A.1.

- uniqueResourceNames: This OCL constraint validates the uniqueness of the names among the resources that are associated with this *RESTfulServiceCIM* element.

4.1.2.2.3 Resource Element

Overview

The *Resource* element models a RESTful resource. It is the core building block of a RESTful service, thus the cornerstone element of the S-CASE Core CIM meta-model. This element aims to capture the name and the nature of a desired resource.

Properties

Table 4-5 Resource's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the provided resource name. Among others, it is used to define code class names and project file structure.
isAlgorithmic	EBoolean	1	This attribute defines the nature of the <i>Resource</i> element. If it is <i>false</i> , then the <i>Resource</i> element describes a simple data-holding resource like the <i>EJB</i> classes. If it is <i>true</i> , however, it models a <i>Resource</i> element that encapsulates some sort of algorithm.

Relations

Table 4-6 Resource's Relations

Relation With CIM Element	Type	Multiplicity	Structural Constraints
Resource	reference	0..*	The <i>Resource</i> element may be associated with zero or more <i>Resources</i> . This association models the fact that a <i>Resource</i> may have zero or more related <i>Resources</i> .
Resource	reference	0..*	The <i>Resource</i> element may be associated of zero or more <i>Resources</i> . This association models the fact that a <i>Resource</i> may be related <i>Resource</i> of zero or more other <i>Resources</i> .
InputRepresentation	composition	1..*	This composition association models the fact that each <i>Resource</i> element may have one or more <i>InputRepresentation</i> elements that encapsulate its possible input representations.
OutputRepresentation	composition	1..*	This composition association models the fact that each <i>Resource</i> element may have one or more <i>OutputRepresentation</i> elements that encapsulate its possible output representations.
Property	composition	0..*	This composition association models the fact that each <i>Resource</i> element may have zero or more <i>Property</i> elements that encapsulate the <i>Resource</i> 's data.
CRUDActivity	composition	1..4	This composition association models the fact that each <i>Resource</i> element may have at least one and at most four <i>CRUDActivity</i> elements that encapsulate the possible actions on this <i>Resource</i> 's data or algorithm.

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of a *Resource* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the CIM meta-model that can be found in appendix A.1.

- *uniqueNamingProperty*: This OCL constraint checks whether the associated *CRUDActivity* elements of this *Resource*, have unique *CRUDVerbs*.
- *uniqueInputMediaTypes*: This OCL constraint checks whether the associated *InputRepresentation* elements of this *Resource*, have unique *MediaTypes*.
- *uniqueOutputMediaTypes*: This OCL constraint checks whether the associated *OutputRepresentation* elements of this *Resource*, have unique *MediaTypes*.
- *uniqueNamingProperty*: This OCL constraint checks whether exactly one of the associated *Property* elements of this *Resource*, is *naming property*.

- *algoResHaveNoProperties*: This OCL constraint validates that a *Resource* element that is algorithmic, does not have any property.
- *CRUDResHaveAtLeastOneProperty*: This OCL constraint validates that a *Resource* element that is not algorithmic, has at least one associated *Property* element.

4.1.2.2.4 CRUActivity Element

Overview

The *CRUActivity* element models a possible action that can be performed on the data or an algorithm that is encapsulated by a *Resource* element. The possible actions are restricted to the *CRUD* ones.

Properties

Table 4-7 CRUActivity's Element

Name	Type	Multiplicity	Explanation
CRUDVerb	CRUDVerb	1	This attribute of the <i>CRUActivity</i> element holds its type, which can be any of the <i>CRUD</i> verbs modelled by the <i>CRUDVerb</i> media type.

Relations

The *CRUActivity* element of the Core CIM meta-model does not have any relation.

Behavioural Restrictions

The *CRUActivity* element of the Core CIM meta-model does not have any behavioural restriction.

4.1.2.2.5 Property Element

Overview

The *Property* element models the properties of a *Resource* element. It encapsulates the name, type, multiplicity and other aspects of each *Resource*'s property.

Properties

Table 4-8 Property's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This attribute models the name of the <i>Property</i> element.
type	EString	1	This attribute models the type of the <i>Property</i> element e.g. String, Integer etc.
isUnique	EBoolean	1	This attribute models the multiplicity of the <i>Property</i> . If it is <i>true</i> , then the <i>Property</i> element is of multiplicity one. Otherwise, it is of multiplicity many.
isNamingProperty	EBoolean	1	This attribute specifies whether an attribute has

			naming capabilities or not. This attribute is useful for clients in order to pick a specific resource out of a list and then retrieve it using the accompanying <i>Hypermedia Link</i> .
--	--	--	--

Relations

The *Property* element of the Core CIM meta-model does not have any relation.

Behavioural Restrictions

The *Property* element of the Core CIM meta-model does not have any behavioural restriction.

4.1.2.2.6 InputRepresentation Element

Overview

The *InputRepresentation* element models one possible input media format that is acceptable by a *Resource*'s interface.

Properties

Table 4-9 InputRepresentation's Properties

Name	Type	Multiplicity	Explanation
inputMediaType	MediaType	1	This attribute of type <i>MediaType</i> encapsulates one acceptable input media format by the <i>Resource</i> element with which it is associated.

Relations

The *InputRepresentation* element of the Core CIM meta-model does not have any relation.

Behavioural Restrictions

The *InputRepresentation* element of the Core CIM meta-model does not have any behavioural restriction.

4.1.2.2.7 OutputRepresentation Element

Overview

The *OutputRepresentation* element models one output media format that a *Resource* element can package its data before sending to a client.

Properties

Table 4-10 OutputRepresentation's Properties

Name	Type	Multiplicity	Explanation
outputMediaType	MediaType	1	This attribute of type <i>MediaType</i> encapsulates one possible output media format by the <i>Resource</i> element with which it is associated.

Relations

The *OutputRepresentation* element of the Core CIM meta-model does not have any relation.

Behavioural Restrictions

The *OutputRepresentation* element of the Core CIM meta-model does not have any behavioural restriction.

4.2 CIM UML Profile Extensions

The following subsections describe the extra functionality that is automatically embedded to the envisioned systems that S-CASE produces. Following the meta-model extension mechanism that section 3.1 presents, the following Core CIM extensions allow the S-CASE MDE engine to embed to the systems it produces authentication and *ABAC* authorization mechanisms, external 3rd party service compositions as well as database searching capabilities. Section 4.2.1.1 to 4.2.1.4 provide more information regarding the extended functionality.

4.2.1 CIM UML Profile Extensions Goals

4.2.1.1 CIM Authentication Extension Goals

The *Authentication* CIM extension meta-model comprises the needed annotations so as to be able to embed to the produced services a *Basic Authentication* mechanism [17]. This authentication layer (defined in section 4.2.2), allows the S-CASE developer to select a *Resource* of his envisioned system to be the authentication model. That model comprises of a *Password* and a *Username*. As Figure 4-3 demonstrates, such resources may have more properties (e.g. email) as well as different names for the password/username pair. The user will have to select which of the resource's properties should be used by the system as a password and which should be used as a username.

Once the authentication model is selected the user is able to define the desired authentication mode for every *CRUDActivity* of the system. Currently the *Authentication* CIM extension meta-model includes three options. The first is the *guest* mode, which allows clients that are not authenticated to access the annotated resource action. The second is the authenticated only mode, which restricts the access to the annotated resource to clients that have provided a valid pair of username and password. Finally, there is also the *both* mode, which allows either of the aforementioned client categories to access the desired resource action.

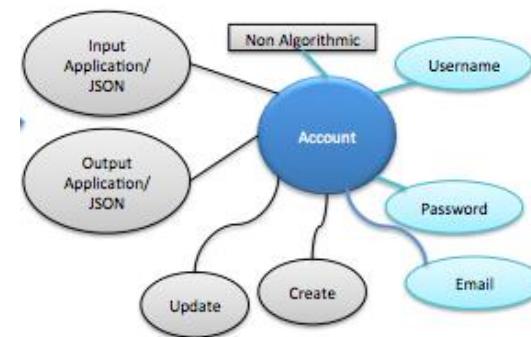


Figure 4-3 An example resource that could be used as authentication model

4.2.1.2 CIM Database Searching Extension Goals

The *Database Searching* CIM extension meta-model comprises the needed annotations so as the MDE engine is able to produce envisioned systems that embed automated keyword searching capabilities to their local database. Section 4.2.3 defines structurally and behaviourally this extension meta-model.

With this extension the S-CASE developer is able to select any of his envisioned system resource that models some sort of algorithm and annotate it as a search resource. Once a resource is annotated as a search resource, it will embed the needed functionality to perform database queries, given a keyword, and return the results to its client in hypermedia links form. After the S-CASE developer selects a resource to be annotated as a search resource, he is able to select any combination of his envisioned system properties that this specific search resource should look up everytime a client makes a search request. It must be noted, that if a resource property is annotated as either *Password* or *Username* it is not selectable. Figure 4-4 demonstrates a simple case with 3 resources and their selectable properties.

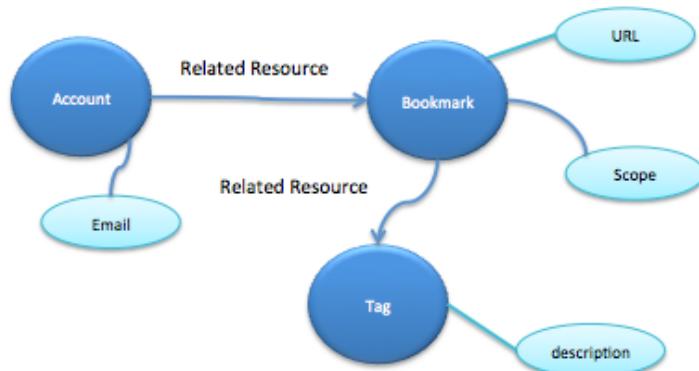


Figure 4-4 Searchable Resource Properties

4.2.1.3 CIM External Service Composition Wrapping Extension Goals

The *External Service Composition* CIM extension meta-model introduces the needed annotations so as to be able to embed the needed functionality to produce systems in order to interact with REST compositions of external services. These compositions could be either semi-automated by the WP4 modules or any other existing composition that the user wishes to use. In order to successfully

embed an external services composition to his envisioned system, he must provide the following information:

- A *Resource* of his envisioned system that models some sort of algorithm, which he wishes to interact with the external services composition.
- The *URL* of the external services composition.
- The CRUD verb that should be used to interact with the composition at the provided *URL*.
- The input and output data models that the composition uses and their respective representations
- Optionally the S-CASE developer may select to enable his envisioned system to store any data outcome of the external composition to its local database for future use.

Section 4.2.4 defines the *External Service Compositon* CIM extension both structurally and behaviourally.

4.2.1.4 CIM ABAC Authorization Extension Goals

The Attribute Based Access Control authorization scheme [18] defines a new access control paradigm whereby access rights are granted to users through the use of policies, which combine attributes together. The policies can use any type of attributes (user attributes, resource attributes, context attributes etc.). Whenever a client makes an access request to an action of a resource, the set of its access rules is evaluated in order to conclude whether the client should be granted access or not. The *ABAC Authorization* CIM extension meta-model is structurally and behaviourally defined in section 4.2.5. This ABAC meta-model is a custom simplified subset of the XACML ABAC language [19].

This extension enables the S-CASE developer to define authorization policies per resource of his envisioned system. More specifically, he is able to create a resource access policy set for every resource of his system, which comprises resource access policies. Each such resource access policy comprises in turn several access rules. These rules embed a user defined combination of attributes and their respective values that must be satisfied in runtime so as to grant access to the underlying allowed actions.

4.2.2 CIM Authentication Ecore Meta-model Definition

4.2.2.1 Introduction

In order to fully define the Authentication Ecore CIM extension meta-model (Figure 4-5), its various aspects must be explained, demonstrated as well as documented. This is done in the following subsections following the format section 4.1.2.1 describes.

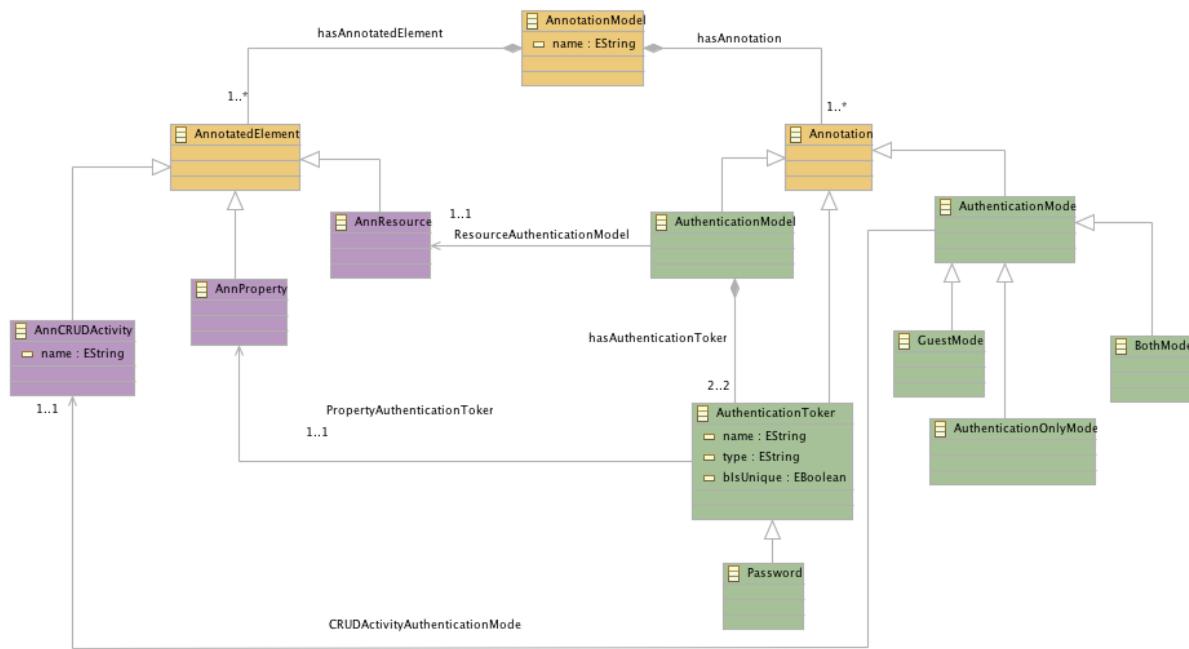


Figure 4-5 Authentication CIM extension meta-model

4.2.2.2 CIM Authentication Ecore Meta-model Elements

This section defines all the elements of the Authentication CIM extension meta-model using the format described in section 4.2.2.1.

4.2.2.2.1 *AnnotationModel* Element

Overview

The *AnnotationModel* element is the root element of the *Authentication* CIM extension meta-model and comprises of all the introduced *Annotations* and the existing Core CIM elements to which these annotations are attached in order to alter their semantic meaning, design and/or configuration.

Properties

Table 4-11 *AnnotationModel*'s Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the name of the <i>Annotation</i> model.

Relations

Table 4-12 *AnnotationModel*'s Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
AnnotatedElement	composition	1..*	This composition association models the fact that every <i>AnnotationModel</i> element must have at least one <i>AnnotatedElement</i> .

Annotation	composition	1..*	This composition association models the fact that every <i>AnnotationModel</i> element must have at least one <i>Annotation</i> .
------------	-------------	------	---

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of an *AnnotationModel* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the *Authentication* CIM extension meta-model that can be found in appendix A.1.

- *atLeastOneResourceIsAnnotated*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnResource*.
- *atLeastTwoPropertiesAreAnnotated*: This OCL constraint checks whether there exist at least two *AnnotatedElements* that are associated with this *AnnotationModel* that are of type *AnnProperty*.
- *atLeastOneCRUDActivityIsAnnotated*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnCRUDActivity*.
- *authenticationModelExists*: This OCL constraint checks whether there exists exactly one *Annotation* element that is associated with this *AnnotationModel* that is of type *AuthenticationModel*.
- *twoAuthenticationTokensExist*: This OCL constraint checks whether there exist exactly two *Annotation* elements that are associated with this *AnnotationModel* that are of type *AuthenticationToken*.
- *exactlyOneAuthenticationTokenIsPassword*: This OCL constraint checks whether there exists exactly one *Annotation* element that is associated with this *AnnotationModel* that is of type *Password*.
- *atLeastOneAuthenticationModeAnnotationExists*: This OCL constraint checks whether there exists at least one *Annotation* element that is associated with this *AnnotationModel* that is of type *GuestMode*, or *AuthenticationOnlyMode*, or *BothMode*.

4.2.2.2 AnnotatedElement Element

Overview

The *AnnotatedElement* element models an existent Core CIM meta-model element that can be annotated by the *Authentication* Core CIM extension in order to extend its semantic meaning and/or design. It is always further specialized to more specific categories of annotated elements.

Properties

The *AnnotatedElement* element of the *Authentication* CIM extension meta-model does not have any property.

Relations

The *AnnotatedElement* element of the *Authentication* CIM extension meta-model does not have any relation.

Behavioural Restrictions

The *AnnotatedElement* of the *Authentication* CIM extension meta-model does not have any behavioural restriction.

4.2.2.2.3 Annotation Element

Overview

The *Annotation* element models any new semantic meaning or design that is attached to a Core CIM meta-model element. It is always further specialized to more specific categories of annotations.

Properties

The *Annotation* element of the *Authentication* CIM extension meta-model does not have any property.

Relations

The *Annotation* element of the *Authentication* CIM extension meta-model does not have any relation.

Behavioural Restrictions

The *Annotation* element of the *Authentication* CIM extension meta-model does not have any behavioural restriction.

4.2.2.2.4 AnnCRUDActivity Element

Overview

The *AnnCRUDActivity* element models an existent *CRUDActivity* of the Core CIM model that can be annotated by the *Authentication* CIM extension meta-model, in order to alter its semantic meaning and/or design.

Properties

Table 4-13 AnnCRUDActivity's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the name of the Core CIM model <i>CRUDActivity</i> element to be annotated.

Relations

Table 4-14 AnnCRUDActivity's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
<i>CRUDActivity</i>	reference	1	The <i>AnnCRUDActivity</i> element must always reference exactly one <i>CRUDActivity</i> element of the Core CIM meta-model. It is the <i>CRUDActivity</i> to which some annotation will be attached.

Behavioural Restrictions

The *AnnCRUDActivity* element of the *Authentication* CIM extension meta-model does not have any behavioural restriction.

4.2.2.2.5 AnnProperty Element

Overview

The *AnnProperty* element models an existent *Property* of the Core CIM model that can be annotated by the *Authentication* CIM extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnProperty* element of the *Authentication* CIM extension meta-model does not have any property.

Relations

Table 4-15 AnnProperty's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
<i>Property</i>	reference	1	The <i>AnnProperty</i> element must reference exactly one Core CIM model <i>Property</i> element.

Behavioural Restrictions

The *AnnProperty* element of the *Authentication* CIM extension meta-model does not have any behavioural restriction.

4.2.2.2.6 *AnnResource Element*

Overview

The *AnnResource* element models an existent *Resource* of the Core CIM model that can be annotated by the *Authentication* CIM extension meta-model, in order to alter its semantic meaning and/or design. In this case, a *Resource* element is annotated as an *Authentication* model.

Properties

The *AnnResource* element of the *Authentication* CIM extension meta-model does not have any property.

Relations

Table 4-16 AnnResource's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
Resource	reference	1	The <i>AnnResource</i> element must reference exactly one <i>Resource</i> element of the Core CIM model.

Behavioural Restrictions

The *AnnResource* element of the *Authentication* CIM extension meta-model does not have any behavioural restriction.

4.2.2.2.7 *AuthenticationModel Element*

Overview

The *AuthenticationModel* element models an annotation of the *Authentication* CIM extension meta-model that is intended to annotate an existent Core CIM *Resource*. With this annotation, the *Resource* is intended to be the means by which a user should get authenticated by the envisioned web service system.

Properties

The *AuthenticationModel* element of the *Authentication* CIM meta-model does not have any property.

Relations

Table 4-17 AuthenticationModel's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
AuthenticationToken	composition	2	The <i>AuthenticationModel</i> element must have composition association with exactly two <i>AuthenticationToken</i> elements. These two <i>AuthenticationToken</i> elements are intended to be the credentials that must be used to

login in the envisioned web service system.			
AnnResource	association	1	The <i>AuthenticationModel</i> element must have exactly one association with one <i>AnnResource</i> element. That <i>AnnResource</i> element references the Core CIM <i>Resource</i> element that will be assigned the authentication model role.

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of an *AuthenticationModel* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the Authentication CIM extension meta-model that can be found in appendix A.1.

- *exactlyOneAuthenticationTokenIsPassword*: This OCL constraint checks whether there exists exactly one *Annotation* element that is associated with this *AuthenticationModel* that is of type *Password*.

4.2.2.2.8 AuthenticationToken Element

Overview

The *AuthenticationToken* element models an annotation of the Authentication CIM extension meta-model that is intended to annotate an existent Core CIM *Property*. With this annotation, the *Property* is intended to be used as an authentication token (either password or username) by the envisioned web service system.

Properties

Table 4-18 AuthenticationToken's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the name of the Core CIM <i>Property</i> element that is annotated as <i>AuthenticationToken</i> element.
type	EString	1	This is the type of the Core CIM <i>Property</i> element that is annotated as <i>AuthenticationToken</i> element.
isUnique	EBoolean	1	This Boolean property indicates whether the annotated as <i>AuthenticationToken</i> Core CIM <i>Property</i> element is unique (true) or may have multiple values (false).

Relations

Table 4-19 AuthenticationToken's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
AnnProperty	association	1	The <i>AuthenticationToken</i> element must have exactly one association with an <i>AnnProperty</i> element. That <i>AnnProperty</i> element references the Core CIM <i>Property</i> element that will be assigned the <i>AuthenticationToken</i> role.

Behavioural Restrictions

The *AuthenticationToken* element of the *Authentication* CIM extension meta-model does not have any behavioural restriction.

4.2.2.2.9 Password Element

Overview

The *Password* element models an annotation of the *Authentication* CIM extension meta-model that is intended to annotate an existent Core CIM *Property*. With this annotation, the *Property* is intended to be the *Password Authentication Token* that will be used by the envisioned web service system in order to authenticate its users.

Properties

The *Password* element does not have any extra properties other than the ones inherited from the *AuthenticationToken* element.

Relations

The *Password* element does not have any extra relations other than the ones inherited from the *AuthenticationToken* element.

Behavioural Restrictions

The *Password* element of the *Authentication* CIM extension meta-model does not have any behavioural restrictions.

4.2.2.2.10 AuthenticationMode Element

Overview

The *AuthenticationMode* element models an annotation of the *Authentication* CIM extension meta-model that is intended to annotate an existent Core CIM *CRUDActivity*. Depending on the type of this

annotation, the *CRUDActivity* will allow only authenticated users to access it, only guests, or both. The *AuthenticationMode* element is an abstract annotation and is always further specialized as either *GuestMode*, *AuthenticationOnlyMode* or *BothMode*.

Properties

The *AuthenticationMode* element of the *Authentication* CIM meta-model does not have any properties.

Relations

Table 4-20 AuthenticationMode's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
AnnCRUDActivity	association	1	The <i>AuthenticationMode</i> element must have exactly one association with an <i>AnnCRUDActivity</i> element. That <i>AnnCRUDActivity</i> references a Core CIM model <i>CRUDActivity</i> that will have this specific type of authentication mode.

Behavioural Restrictions

The *AuthenticationMode* element of the *Authentication* CIM extension meta-model does not have any behavioural restriction.

4.2.2.2.11 GuestMode Element

Overview

The *GuestMode* element models an annotation of the *Authentication* CIM extension meta-model that is intended to annotate an existent Core CIM *CRUDActivity*. With this annotation, the *CRUDActivity* will only allow guests of the envisioned web service system to access it.

Properties

The *GuestMode* element of the *Authentication* CIM meta-model does not have any properties.

Relations

The *GuestMode* element does not have any extra relations other than the ones inherited from the *AuthenticationMode* element.

Behavioural Restrictions

The *GuestMode* element of the *Authentication* CIM extension meta-model does not have any

behavioural restrictions.

4.2.2.2.12 *BothMode Element*

Overview

The *BothMode* element models an annotation of the *Authentication* CIM extension meta-model that is intended to annotate an existent Core CIM *CRUDActivity*. With this annotation, the *CRUDActivity* will either authenticated users or guests of the envisioned web service system to access it.

Properties

The *BothMode* element of the *Authentication* CIM meta-model does not have any properties.

Relations

The *BothMode* element does not have any extra relations other than the ones inherited from the *AuthenticationMode* element.

Behavioural Restrictions

The *BothMode* element of the *Authentication* CIM extension meta-model does not have any behavioural restrictions.

4.2.2.2.13 *AuthenticationOnlyMode Element*

Overview

The *AuthenticationOnlyMode* element models an annotation of the *Authentication* CIM extension meta-model that is intended to annotate an existent Core CIM *CRUDActivity*. With this annotation, the *CRUDActivity* will only allow authenticated users of the envisioned web service system to access it.

Properties

The *AuthenticationOnlyMode* element of the *Authentication* CIM meta-model does not have any properties.

Relations

The *AuthenticationOnlyMode* element does not have any extra relations other than the ones inherited from the *AuthenticationMode* element.

Behavioural Restrictions

The *AuthenticationOnlyMode* element of the *Authentication* CIM extension meta-model does not have any behavioural restrictions.

4.2.3 CIM Database Searching Ecore Meta-model Definition

4.2.3.1 Introduction

In order to fully define the *Database Searching* Ecore CIM extension meta-model (Figure 4-6), its various aspects must be explained, demonstrated as well as documented. This is done in the following subsections following the format section 4.1.2.1 describes.

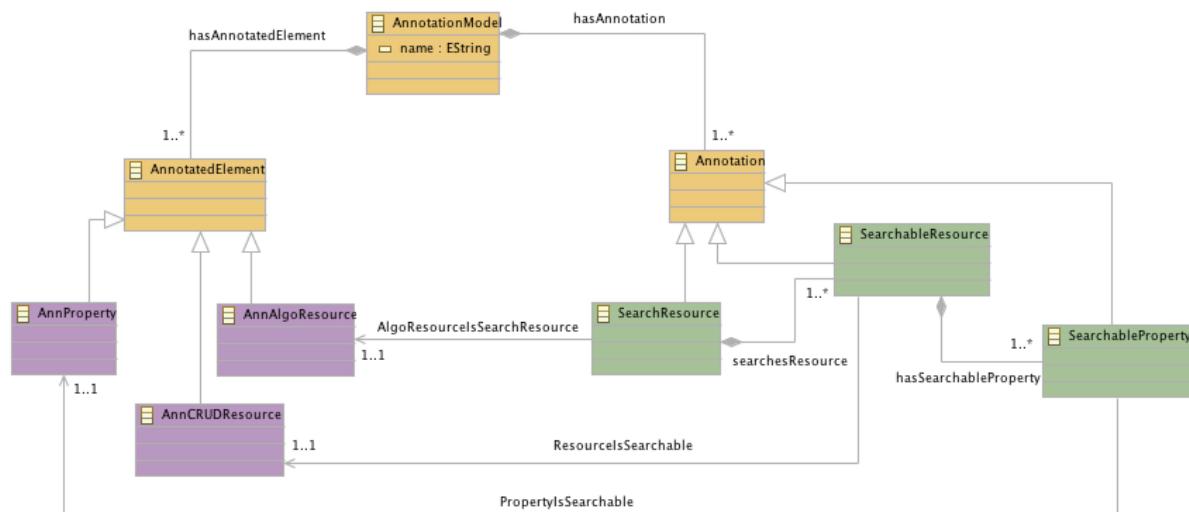


Figure 4-6 Database Searching CIM extension meta-model

4.2.3.2 CIM Database Searching Ecore Meta-model Elements

4.2.3.2.1 AnnotationModel Element

Overview

The *AnnotationModel* element is the root element of the *Database Searching* CIM extension meta-model and comprises of all the introduced *Annotations* and the existing Core CIM elements to which these annotations are attached in order to alter their semantic meaning, design and/or configuration.

Properties

Table 4-21 AnnotationModel's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the name of the <i>Annotation</i> model.

Relations

Table 4-22 AnnotationModel's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
AnnotatedElement	composition	1..*	This composition association models the fact that every <i>AnnotationModel</i> element must have at least one <i>AnnotatedElement</i> .

Annotation	composition	1..*	This composition association models the fact that every <i>AnnotationModel</i> element must have at least one <i>Annotation</i> .
------------	-------------	------	---

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of an *AnnotationModel* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the *Database Searching* CIM extension meta-model that can be found in appendix A.1.

- *atLeastOneAnnAlgoResourceExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnAlgoResource*.
- *atLeastOneAnnCRUDResourceExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnCRUDResource*.
- *atLeastOneAnnPropertyExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnProperty*.
- *atLeastOneSearchResourceExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *SearchResource*.

Overview

The *AnnotatedElement* element models any existent Core CIM meta-model element that can be annotated by the *Database Searching* Core CIM extension in order to extend its semantic meaning and/or design. It is always further specialized to more specific categories of annotated elements.

Properties

The *AnnotatedElement* element of the *Database Searching* CIM extension meta-model does not have any property.

Relations

The *AnnotatedElement* element of the *Database Searching* CIM extension meta-model does not have any relation.

Behavioural Restrictions

The *AnnotatedElement* of the *Database Searching* CIM extension meta-model does not have any behavioural restriction.

4.2.3.2.2 Annotation Element

Overview

The *Annotation* element models any new semantic meaning or design that is attached to a Core CIM meta-model element. It is always further specialized to more specific categories of annotations.

Properties

The *Annotation* element of the *Database Searching CIM* extension meta-model does not have any property.

Relations

The *Annotation* element of the *Database Searching CIM* extension meta-model does not have any relation.

Behavioural Restrictions

The *Annotation* element of the *Database Searching CIM* extension meta-model does not have any behavioural restriction.

4.2.3.2.3 AnnProperty Element

Overview

The *AnnProperty* element models an existent *Property* of the Core CIM model that can be annotated by the *Database Searching CIM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnProperty* element of the *Database Searching CIM* extension meta-model does not have any property.

Relations

Table 4-23 AnnProperty's Relations

Relation With CIM Element	Type	Multiplicity	Structural Constraints
Property	reference	1	The <i>AnnProperty</i> element must reference exactly one Core CIM <i>Property</i> element.

Behavioural Restrictions

The *AnnProperty* element of the *Database Searching CIM* extension meta-model does not have any behavioural restriction.

4.2.3.2.4 AnnAlgoResource Element

Overview

The *AnnAlgoResource* element models an existent *Resource* of the Core CIM model, which has its *isAlgorithmic* property set to true and that can be annotated by the *Database Searching CIM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnAlgoResource* element of the *Database Searching* CIM extension meta-model does not have any property.

Relations

Table 4-24 AnnAlgoResource's Relations

Relation With CIM Element	Type	Multiplicity	Structural Constraints
Resource	reference	1	The <i>AnnAlgoResource</i> element must have exactly one reference to a Core CIM <i>Resource</i> one, which has its <i>isAlgorithmic</i> property set to true.

Behavioural Restrictions

The *AnnAlgoResource* element of the *Database Searching* CIM extension meta-model does not have any behavioural restriction.

4.2.3.2.5 AnnCRUDResource Element

Overview

The *AnnCRUDResource* element models an existent *Resource* of the Core CIM model, which has its *isAlgorithmic* property set to false and that can be annotated by the *Database Searching* CIM extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnCRUDResource* element of the *Database Searching* CIM extension meta-model does not have any property.

Relations

Table 4-25 AnnCRUDResource's Relations

Relation With CIM Element	Type	Multiplicity	Structural Constraints
Resource	reference	1	The <i>AnnCRUDResource</i> element must have exactly one reference to a Core CIM <i>Resource</i> element, which has its <i>isAlgorithmic</i> property set to false.

Behavioural Restrictions

The *AnnCRUDResource* element of the *Database Searching* CIM extension meta-model does not have any behavioural restriction.

4.2.3.2.6 SearchResource Element

Overview

The *SearchResource* element models an annotation of the *Database Searching* CIM extension meta-model that is intended to annotate an existent Core CIM *Resource*, which has its *isAlgorithmic* property set to true. With this annotation, this *Resource* will be able to search the database given a *keyword* by its client and return any results to them by providing hypermedia links.

Properties

The *SearchResource* element of the *Database Searching* CIM extension meta-model does not have any property.

Relations

Table 4-26 SearchResource's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
AnnAlgoResource	association	1	The <i>SearchResource</i> element must have an association with exactly one <i>AnnAlgoResource</i> , which references a Core CIM model <i>Resource</i> . That <i>Resource</i> will be able to search the envisioned web service local database by keywords provided by its clients and then return to them any results as hypermedia links.
SearchableResource	composition	1..*	The <i>SearchResource</i> element must have a composition association with at least one <i>SearchableResource</i> element. This association models the fact that a <i>SearchResource</i> must be able to search at least one Core CIM <i>Resource</i> , which has its <i>isAlgorithmic</i> property set to false.

Behavioural Restrictions

The *SearchResource* element of the *Database Searching* CIM extension meta-model does not have any behavioural restriction.

4.2.3.2.7 SearchableResource Element

Overview

The *SearchableResource* element models an annotation of the *Database Searching* CIM extension meta-model that is intended to annotate an existent Core CIM *Resource*, which has its *isAlgorithmic* property set to false. With this annotation, that *Resource* will be indexed so as to be searchable by other *Resources*, which have been annotated with a *SearchResource* annotation.

Properties

The *SearchResource* element of the *Database Searching* CIM extension meta-model does not have any property.

Relations

Table 4-27 SearchableResource's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
AnnCRUDResource	association	1	The <i>SearchableResource</i> element must have exactly one association with an <i>AnnCRUDResource</i> element, which references a Core CIM <i>Resource</i> . This <i>Resource</i> be indexed so as to be searchable by other <i>Resources</i> that are annotated with the <i>SearchResource</i> annotation.
SearchableProperty	composition	1..*	The <i>SearchableResource</i> element must have a composition association with at least one <i>SearchableProperty</i> element. This association models the fact that any Core CIM <i>Resource</i> that is annotated as <i>SearchableResource</i> must have at least one <i>Property</i> to be searched using a keyword.

Behavioural Restrictions

The *SearchableResource* element of the *Database Searching* CIM extension meta-model does not have any behavioural restriction.

4.2.3.2.8 SearchableProperty Element

Overview

The *SearchableProperty* element models an annotation of the *Database Searching* CIM extension meta-model that is intended to annotate an existent Core CIM *Property*. With this annotation, this *Property* will be indexed, so as to be searchable by *Resources* that are annotated with the *SearchResource* annotation by using a keyword.

Properties

The *SearchableProperty* element of the *Database Searching* CIM extension meta-model does not have any property.

Relations

Table 4-28 SearchableProperty's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
AnnProperty	association	1	The <i>SearchableProperty</i> element must have an association with exactly one <i>AnnProperty</i> element that references a Core CIM <i>Property</i> . That <i>Property</i> will then be indexed so as to be searchable by using a keyword.

Behavioural Restrictions

The *SearchableProperty* element of the *Database Searching* CIM extension meta-model does not have any behavioural restriction.

4.2.4 CIM External Service Composition Ecore Meta-model Definition

4.2.4.1 Introduction

In order to fully define the *External Service Composition* Ecore CIM extension meta-model (Figure 4-7), its various aspects must be explained, demonstrated as well as documented. This is done in the following subsections following the format section 4.1.2.1 describes.

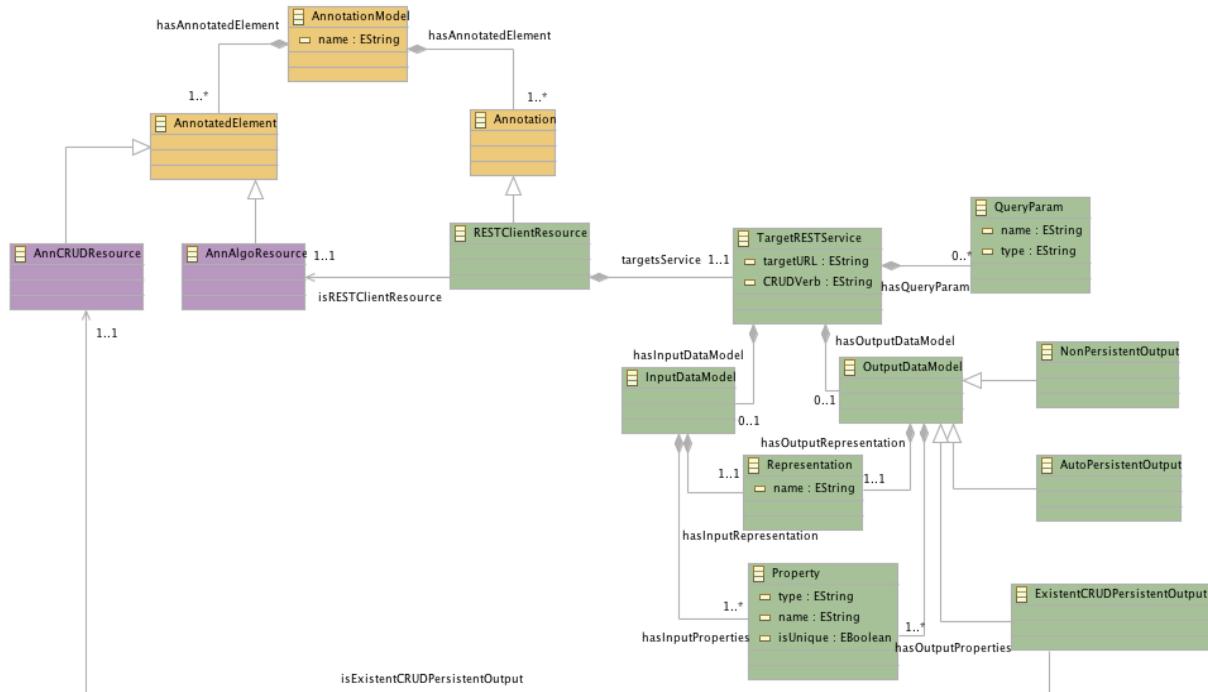


Figure 4-7 External Service Composition CIM extension meta-model

4.2.4.2 CIM External Service Composition Ecore Meta-model Elements

4.2.4.2.1 AnnotationModel Element

Overview

The *AnnotationModel* element is the root element of the *External Service Composition* CIM extension meta-model and comprises of all the introduced *Annotations* and the existing Core CIM elements to which these annotations are attached in order to alter their semantic meaning, design and/or configuration.

Properties

Table 4-29 AnnotationModel's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the name of the <i>Annotation</i> model.

Relations

Table 4-30 AnnotationModel's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
AnnotatedElement	composition	1..*	This composition association models the fact that every <i>AnnotationModel</i> element must have at least one <i>AnnotatedElement</i> .
Annotation	composition	1..*	This composition association models the fact that every <i>AnnotationModel</i> element must have at least one <i>Annotation</i> .

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of an *AnnotationModel* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the *External Service Composition* CIM extension meta-model that can be found in appendix A.1.

- *atLeastOneAlgoResourcesAnnotated*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnAlgoResource*.

4.2.4.2.2 AnnotatedElement Element

Overview

The *AnnotatedElement* element models any existent Core CIM meta-model element that can be annotated by the *External Service Composition* Core CIM extension in order to extend its semantic meaning and/or design. It is always further specialized to more specific categories of annotated elements.

Properties

The *AnnotatedElement* element of the *External Service Composition CIM* extension meta-model does not have any property.

Relations

The *AnnotatedElement* element of the *External Service Composition CIM* extension meta-model does not have any relation.

Behavioural Restrictions

The *AnnotatedElement* element of the *External Service Composition CIM* extension meta-model does not have any behavioural restriction.

4.2.4.2.3 Annotation Element

Overview

The *Annotation* element models any new semantic meaning or design that is attached to a Core CIM meta-model element. It is always further specialized to more specific categories of annotations.

Properties

The *Annotation* element of the *External Service Composition CIM* extension meta-model does not have any property.

Relations

The *Annotation* element of the *External Service Composition CIM* extension meta-model does not have any behavioural restriction.

Behavioural Restrictions

The *Annotation* element of the *External Service Composition CIM* extension meta-model does not have any relation.

4.2.4.2.4 AnnCRUDResource Element

Overview

The *AnnCRUDResource* element models an existent *Resource* of the Core CIM model, which has its *isAlgorithmic* property set to *false*, that can be annotated by the *External Service Composition CIM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnCRUDResource* element of the *External Service Composition* CIM extension meta-model does not have any property.

Relations

Table 4-31 AnnCRUDResource Relations

Relation With CIM Element	Type	Multiplicity	Structural Constraints
Resource	reference	1	The <i>AnnCRUDResource</i> element must reference exactly one Core CIM <i>Resource</i> element, which has its <i>isAlgorithmic</i> property set to <i>false</i> .

Behavioural Restrictions

The *AnnCRUDResource* element of the *External Service Composition* CIM extension meta-model does not have any behavioural restriction.

4.2.4.2.5 AnnAlgoResource Element

Overview

The *AnnAlgoResource* element models an existent *Resource* of the Core CIM model, which has its *isAlgorithmic* property set to *true*, that can be annotated by the *External Service Composition* CIM extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnAlgoResource* element of the *External Service Composition* CIM extension meta-model does not have any property.

Relations

Table 4-32 AnnAlgoResource's Relations

Relation With CIM Element	Type	Multiplicity	Structural Constraints
Resource	reference	1	The <i>AnnAlgoResource</i> element must reference exactly one Core CIM <i>Resource</i> element, which has its <i>isAlgorithmic</i> property set to <i>true</i> .

Behavioural Restrictions

The *AnnAlgoResource* element of the *External Service Composition* CIM extension meta-model does not have any behavioural restriction.

4.2.4.2.6 RESTClientResource Element

Overview

The *RESTClientResource* element models an annotation of the *External Service Composition* CIM extension meta-model that is intended to annotate an existent Core CIM *Resource*, which has its *isAlgorithmic* property set to *true*. With this annotation, that *Resource* will be designed as a client that will interact with an external RESTful service composition. Figure 4-8 demonstrates the

RESTClientResource element of the *External Service Composition* CIM extension meta-model and its relations.

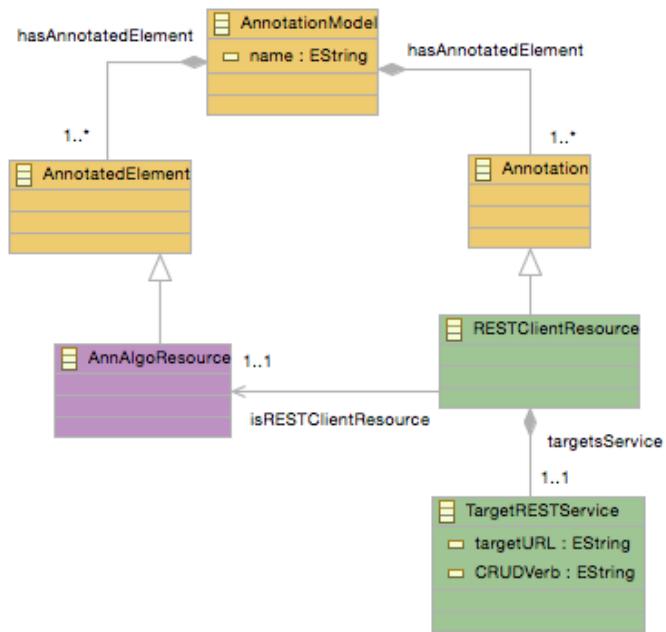


Figure 4-8 RESTClientResource annotation and its relations.

Properties

The *RESTClientResource* element of the *External Service Composition* CIM extension meta-model does not have any property.

Relations

Table 4-33 RESTClientResource's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
TargetRESTService	composition	1	The <i>RESTClientResource</i> element must have exactly one composition association with a <i>TargetRESTService</i> element. This association models the fact that every <i>RESTClientResource</i> is intended to interact with exactly one external RESTful service composition.
AnnAlgoResource	association	1	The <i>RESTClientResource</i> element must have exactly one associated <i>AnnAlgoResource</i> that references a Core CIM Resource, which has its <i>isAlgorithmic</i> property set to <i>true</i> . This association models the intension to design that specific <i>Resource</i> with REST client capabilities.

Behavioural Restrictions

The *RESTClientResource* element of the *External Service Composition* CIM extension meta-model

does not have any behavioural restriction.

4.2.4.2.7 TargetRESTService Element

Overview

The *TargetRESTService* element models the target external RESTful service composition with which the *RESTClientResource* is intended to interact. Figure 4-9 demonstrates the *TargetRestService* element of the *External Service Composition* CIM extension meta-model and its relations.

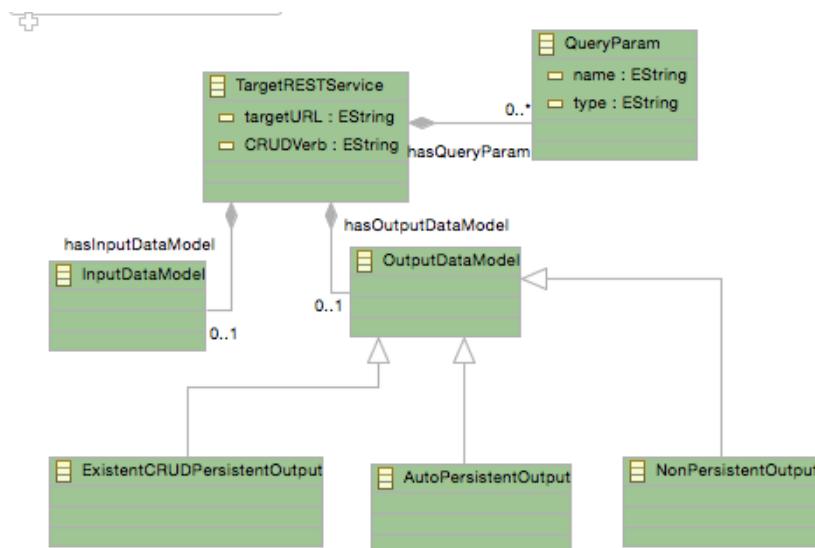


Figure 4-9 TargetRESTService element and its annotations

Properties

Table 4-34 TargetRESTService's Properties

Name	Type	Multiplicity	Explanation
<i>targetURL</i>	EString	1	This property contains the target URL of the external service composition with which the envisioned system will interact by the means of an automated REST client.
<i>CRUDVerb</i>	EString	1	This models the <i>CRUDVerb</i> that must be used when interacting with the external RESTful service composition. The <i>CRUDVerb</i> can be either <i>Create</i> , <i>Read</i> , <i>Update</i> or <i>Delete</i> .

Relations

Table 4-35 TargetRESTService's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
<i>QueryParam</i>	composition	0..*	The <i>TargetRESTService</i> can have a composition association with zero or more <i>QueryParam</i> elements. These <i>QueryParam</i> elements are the needed query params that must be used when the envisioned system interacts with the

			external RESTful service composition.
OutputDataModel	composition	0..1	The <i>TargetRESTService</i> can have zero or one composition association with an <i>OutputDataModel</i> element. That <i>OutputDataModel</i> models the expected output of the external RESTful service composition.
InputDataModel	composition	0..1	The <i>TargetRESTService</i> can have zero or one composition association with an <i>InputDataModel</i> element. That <i>InputDataModel</i> models the expected input that should be sent to the external RESTful service composition when the envisioned system interacts with it.

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of a *TargetRESTService* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the CIM extension meta-model that can be found in appendix A.1.

- *specializedOutputModel*: This OCL constraint checks whether the *OutputModel* that is associated with this *TargetRESTService* element (if there exists one), is specialized to either *NonPersistentOutput*, *AutoPersistentOutput* or *ExistentCRUDPersistentOutput*.

4.2.4.2.8QueryParam Element

Overview

The *QueryParam* element models an annotation of the *External Service Composition* CIM extension meta-model that models a query param that must be used when the envisioned system interacts with the external RESTful service composition.

Properties

Table 4-36 QueryParam's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the name of the <i>QueryParam</i> element.
type	EString	1	This is the type of the <i>QueryParam</i> element.

Relations

The *QueryParam* element of the *External Service Composition* CIM extension meta-model does not have any relation.

Behavioural Restrictions

The *QueryParam* element of the *External Service Composition* CIM extension meta-model does not

have any behavioural restriction.

4.2.4.2.9 *InputDataModel* Element

Overview

The *InputDataModel* element models the expected input data model that should be used when the envisioned system interacts with the external RESTful service composition. Figure 4-10 demonstrates the *InputDataModel* element of the *External Service Composition* CIM extension meta-model and its relations.

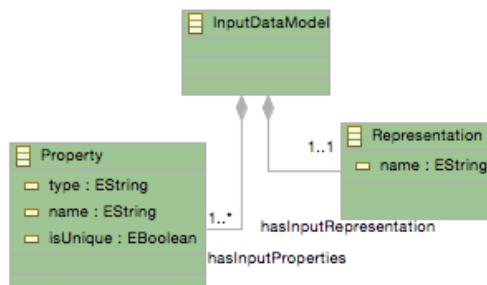


Figure 4-10 *InputDataModel* element and its relations.

Properties

The *InputDataModel* element of the *External Service Composition* CIM extension meta-model does not have any property.

Relations

Table 4-37 *InputDataModel*'s Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
Representation	composition	1..1	The <i>InputDataModel</i> must have exactly one composition association with a <i>Representation</i> element. That <i>Representation</i> models the expected input media format that should be used e.g. "application/JSON" etc.
Property	composition	1..*	The <i>InputDataModel</i> must have at least one composition association with a <i>Property</i> element. Each such <i>Property</i> element models a property of the <i>InputDataModel</i> .

Behavioural Restrictions

The *InputDataModel* element of the *External Service Composition* CIM extension meta-model does not have any behavioural restriction.

4.2.4.2.10 *OutputDataModel* Element

Overview

The *OutputDataModel* element models the expected output data model from the external RESTful service composition, when the envisioned system interacts with it. The *OutputDataModel* is always further specialized to a *NonPersistentElement*, *AutoPersistentOutput* or *ExistentCRUDPersistentOutput*. Figure 4-11 demonstrates the *OutputDataModel* element of the *External Service Composition* CIM extension meta-model and its relations.

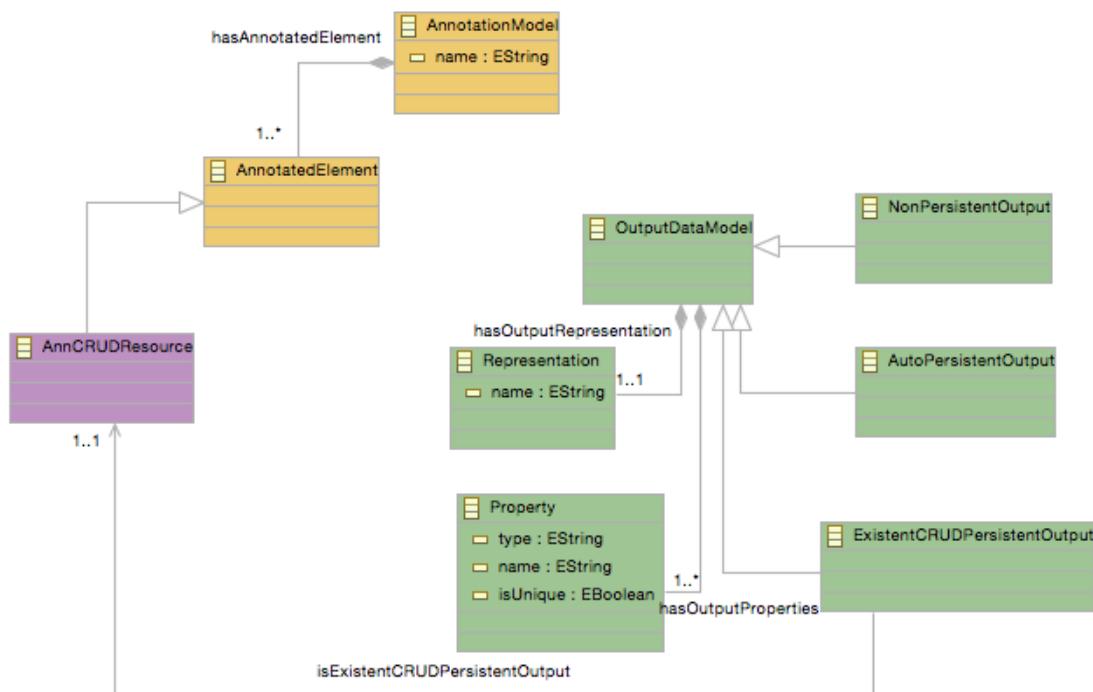


Figure 4-11 OutputDataModel element and its relations.

Properties

The *OutputDataModel* element of the *External Service Composition* CIM extension meta-model does not have any property.

Relations

Table 4-38 OutputDataModel's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
Representation	composition	1..1	The <i>OutputDataModel</i> element must have exactly one composition association with a <i>Representation</i> element. That <i>Representation</i> models the output media format that should be expected when the envisioned system interacts with the external RESTful service composition.
Property	composition	1..*	The <i>OutputDataModel</i> element must have at least one composition association with a <i>Property</i> element. That <i>Property</i> models a property of the <i>OutputDataModel</i> .

Behavioural Restrictions

The *OutputDataModel* element of the *External Service Composition* CIM extension meta-model does not have any behavioural restriction.

4.2.4.2.11 Representation Element

Overview

The *Representation* element models the representation of either the *InputDataModel* or the *OutputDataModel* that must be used/expected when the envisioned system interacts with the external RESTful service composition.

Properties

Table 4-39 Representation's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This property contains the <i>EString</i> that describes the modelled representation. The values can be <i>application/JSON</i> or <i>application/XML</i>

Relations

The *Representation* element of the *External Service Composition* CIM extension meta-model does not have any relation.

Behavioural Restrictions

The *Representation* element of the *External Service Composition* CIM extension meta-model does not have any behavioural restriction.

4.2.4.2.12 Property Element

Overview

The *Property* element models a property of either an *InputDataModel* or an *OutputDataModel* that must be used/expected when the envisioned system interacts with the external RESTful service composition.

Properties

Table 4-40 Property's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the name of the <i>Property</i> element.
type	EString	1	This is the type of the <i>Property</i> element.
isUnique	EBoolean	1	The <i>isUnique</i> property models the multiplicity of the <i>Property</i> . If it is set to <i>true</i> then the <i>Property</i> element has

			multiplicity one, otherwise it is an array with many values.
--	--	--	--

Relations

The *Property* element of the *External Service Composition* CIM extension meta-model does not have any relation.

Behavioural Restrictions

The *Property* element of the *External Service Composition* CIM extension meta-model does not have any behavioural restriction.

4.2.4.2.13 NonPersistentOutput Element

Overview

The *NonPersistentOutput* element models a specialization of the *OutputDataModel* class. When an *OutputDataModel* element is specialized as *NonPersistentOutput* then the output data that is received by the envisioned system when it interacts with the external RESTful service composition, is not stored in the local system database but only returned to its client.

Properties

The *NonPersistentOutput* element of the *External Service Composition* CIM extension meta-model does not have any property.

Relations

The *NonPersistentOutput* element of the *External Service Composition* CIM extension meta-model does not have any relation.

Behavioural Restrictions

The *NonPersistentOutput* element of the *External Service Composition* CIM extension meta-model does not have any behavioural restriction.

4.2.4.2.14 AutoPersistentOutput Element

Overview

The *AutoPersistentOutput* element models a specialization of the *OutputDataModel* element. When an *OutputDataModel* element is specialized as *AutoPersistentOutput* then the output data that is received by the envisioned system when it interacts with the external RESTful service composition, is stored in the local database as well as sent back to its client.

Properties

The *AutoPersistentOutput* element of the *External Service Composition* CIM extension meta-model does not have any property.

Relations

The *AutoPersistentOutput* element of the *External Service Composition* CIM extension meta-model does not have any relation.

Behavioural Restrictions

The *AutoPersistentOutput* element of the *External Service Composition* CIM extension meta-model does not have any behavioural restriction.

4.2.4.2.15 *ExistentCRUDPersistentOutput* Element

Overview

The *ExistentCRUDPersistentOutput* element models a specialization of the *OutputDataModel* element. If an *OutputDataModel* is specialized as *ExistentCRUDPersistentOutput* then the expected output data model from the external RESTful service composition, is a data model that is already defined in the Core CIM. In this case, when the envisioned system interacts with the external RESTful service composition will store the output data as well as send it to its client.

Properties

The *ExistentCRUDPersistentOutput* element of the *External Service Composition* CIM extension meta-model does not have any property.

Relations

Table 4-41 ExistentCRUDPersistentOutput's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
AnnCRUDResource	association	1	The <i>ExistentCRUDPersistentOutput</i> must have exactly one association with an <i>AnnCRUDResource</i> element that references a Core CIM <i>Resource</i> . That <i>Resource</i> will be used as an output data model when the envisioned system interacts with the external RESTful service composition.

Behavioural Restrictions

The *ExistentCRUDPersistentOutput* element of the *External Service Composition* CIM extension meta-model does not have any behavioural restriction.

4.2.5 CIM ABAC Authorization Ecore Meta-model Definition

4.2.5.1 Introduction

In order to fully define the Authentication Ecore CIM extension meta-model, its various aspects must be explained, demonstrated as well as documented. This is done in the following subsections following the format section 4.1.2.1 describes. Due to space limitations, the full *ABAC Authorization* meta-model can be found using the following link <https://github.com/s-case/mde> .

4.2.5.2 CIM ABAC Authorization Ecore Meta-model Elements

4.2.5.2.1 *AnnotationModel* Element

Overview

The *AnnotationModel* element is the root element of the *ABAC Authorization* CIM extension meta-model and comprises of all the introduced *Annotations* and the existing Core CIM elements to which these annotations are attached in order to alter their semantic meaning, design and/or configuration.

Properties

Table 4-42 *AnnotationModel*'s Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the name of the <i>Annotation</i> model.

Relations

Table 4-43 *AnnotationModel*'s Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
AnnotatedElement	composition	1..*	This composition association models the fact that every <i>AnnotationModel</i> element must have at least one <i>AnnotatedElement</i> .
Annotation	composition	1..*	This composition association models the fact that every <i>AnnotationModel</i> element must have at least one <i>Annotation</i> .

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of an *AnnotationModel* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the *ABAC Authorization* CIM extension meta-model that can be found in appendix A.1.

- *atLeastTwoAnnResourceExist*: This OCL constraint checks whether there exist at least two *AnnotatedElements* that are associated with this *AnnotationModel* that are of type *AnnResource*.
- *atLeastOneAnnCRUDActivityExists*: This OCL constraint checks whether there exists at least one *AnnCRUDActivity* that are associated with this *AnnotationModel* that are of type *AnnCRUDActivity*.
- *exactlyOneAuthorizationSubjectExists*: This OCL constraint checks whether there exists at least one *Annotation* element that is associated with this *AnnotationModel* that is of type *AuthorizationSubject*.
- *atLeastOneAuthorizableResourceExists*: This OCL constraint checks whether there exists at least one *Annotation* element that is associated with this *AnnotationModel* that is of type *AuthorizableResource*.

4.2.5.2.2 AnnotatedElement Element

Overview

The *AnnotatedElement* element models any existent Core CIM meta-model element that can be annotated by the *ABAC Authorization* Core CIM extension in order to extend its semantic meaning and/or design. It is always further specialized to more specific categories of annotated elements.

Properties

The *AnnotatedElement* element of the *ABAC Authorization* CIM extension meta-model does not have any property.

Relations

The *AnnotatedElement* element of the *ABAC Authorization* CIM extension meta-model does not have any relation.

Behavioural Restrictions

The *AnnotatedElement* of the *ABAC Authorization* CIM extension meta-model does not have any behavioural restriction.

4.2.5.2.3 Annotation Element

Overview

The *Annotation* element models any new semantic meaning or design that is attached to a Core CIM meta-model element. It is always further specialized to more specific categories of annotations.

Properties

The *Annotation* element of the *ABAC Authorization* CIM extension meta-model does not have any property.

Relations

The *Annotation* element of the *ABAC Authorization* CIM extension meta-model does not have any relation.

Behavioural Restrictions

The *Annotation* element of the ABAC Authorization CIM extension meta-model does not have any behavioural restriction.

4.2.5.2.4 AnnCRUDActivity Element

Overview

The *AnnCRUDActivity* element models an existent *CRUDActivity* of the Core CIM model that can be annotated by the *ABAC Authorization* CIM extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnCRUDActivity* element of the *ABAC Authorization* CIM extension meta-model does not have any property.

Relations

The *AnnCRUDActivity* element of the *ABAC Authorization* CIM extension meta-model does not have any relation

Behavioural Restrictions

The *AnnCRUDActivity* element of the *ABAC Authorization* CIM extension meta-model does not have any behavioural restriction.

4.2.5.2.5 AnnProperty Element

Overview

The *AnnProperty* element models an existent *Property* of the Core CIM model that can be annotated by the *ABAC Authorization* CIM extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnProperty* element of the *ABAC Authorization* CIM extension meta-model does not have any property.

Relations

The *AnnProperty* element of the *ABAC Authorization* CIM extension meta-model does not have any relation.

Behavioural Restrictions

The *AnnProperty* element of the *ABAC Authorization* CIM extension meta-model does not have any

behavioural restriction.

4.2.5.2.6 *AnnResource Element*

Overview

The *AnnResource* element models an existent *Resource* of the Core CIM model that can be annotated by the *ABAC Authorization* CIM extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnResource* element of the *ABAC Authorization* CIM extension meta-model does not have any property.

Relations

The *AnnResource* element of the *ABAC Authorization* CIM extension meta-model does not have any relation.

Behavioural Restrictions

The *AnnResource* element of the *ABAC Authorization* CIM extension meta-model does not have any behavioural restriction.

4.2.5.2.7 *AuthorizationSubject Element*

Overview

The *AuthorizationSubject* element models an annotation of the *ABAC Authorization* CIM extension meta-model that is intended to annotate an existent Core CIM *Resource*, which has its *isAlgorithmic* property set to *false*. With this annotation, that *Resource* will be used as an authorization model by the envisioned system. Figure 4-12 demonstrates the *AuthorizationSubject* element of the *ABAC Authorization* CIM extension meta-model and its relations.

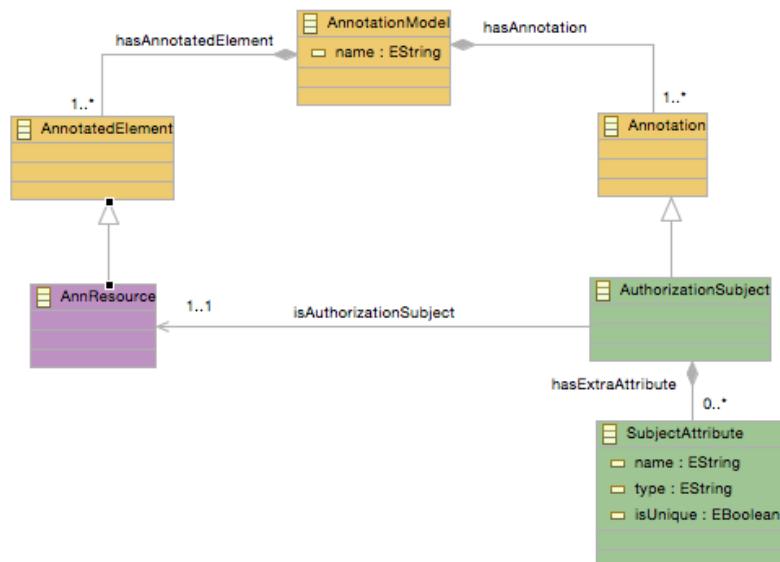


Figure 4-12 Authorization Subject annotation and its relations

Properties

The *AuthorizationSubject* element of the *ABAC Authorization* CIM extension meta-model does not have any property.

Relations

Table 4-44 AuthorizationSubject's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
SubjectAttribute	composition	0..*	The <i>AuthorizationSubject</i> can have zero or more composition associations with <i>SubjectAttribute</i> elements. These <i>SubjectAttributes</i> model extra properties that can be added to the existent Core CIM <i>Resource</i> element, which will be used as authorization model.
AnnResource	association	1	The <i>AuthorizationSubject</i> must have exactly one association with an <i>AnnResource</i> element, which references a Core CIM <i>Resource</i> . The association with this Core CIM <i>Resource</i> models the fact that this specific <i>Resource</i> will be used as an authorization model.

Behavioural Restrictions

The *AuthorizationSubject* element of the *ABAC Authorization* CIM extension meta-model does not have any behavioural restriction.

4.2.5.2.8 SubjectAttribute Element

Overview

The *SubjectAttribute* element models an extra property that can be added to the existing ones of the Core CIM *Resource* element that is annotated with the *AuthorizationSubject* annotation of the ABAC Authorization CIM extension meta-model.

Properties

Table 4-45 SubjectAttribute's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the name of the <i>SubjectAttribute</i> .
type	EString	1	This is the type of the <i>SubjectAttribute</i> .
isUnique	EBoolean	1	This property models the multiplicity of the <i>SubjectAttribute</i> . If it is set to <i>true</i> , then the <i>SubjectAttribute</i> has multiplicity one. Otherwise, it is an array that can have multiple values.

Relations

The *SubjectAttribute* element of the *ABAC Authorization* CIM extension meta-model does not have any relation.

Behavioural Restrictions

The *SubjectAttribute* element of the *ABAC Authorization* CIM extension meta-model does not have any behavioural restriction.

4.2.5.2.9 AuthorizableResource Element

Overview

The *AuthorizableResource* element models an annotation of the *ABAC Authorization* CIM extension meta-model that is intended to annotate an existent Core CIM *Resource*. With this annotation, that *Resource* will be accessible by an envisioned system's client only if its authorization rules are satisfied. Figure 4-13 demonstrates *AuthorizableResource* element of the *ABAC Authorization* CIM extension meta-model and its relations.

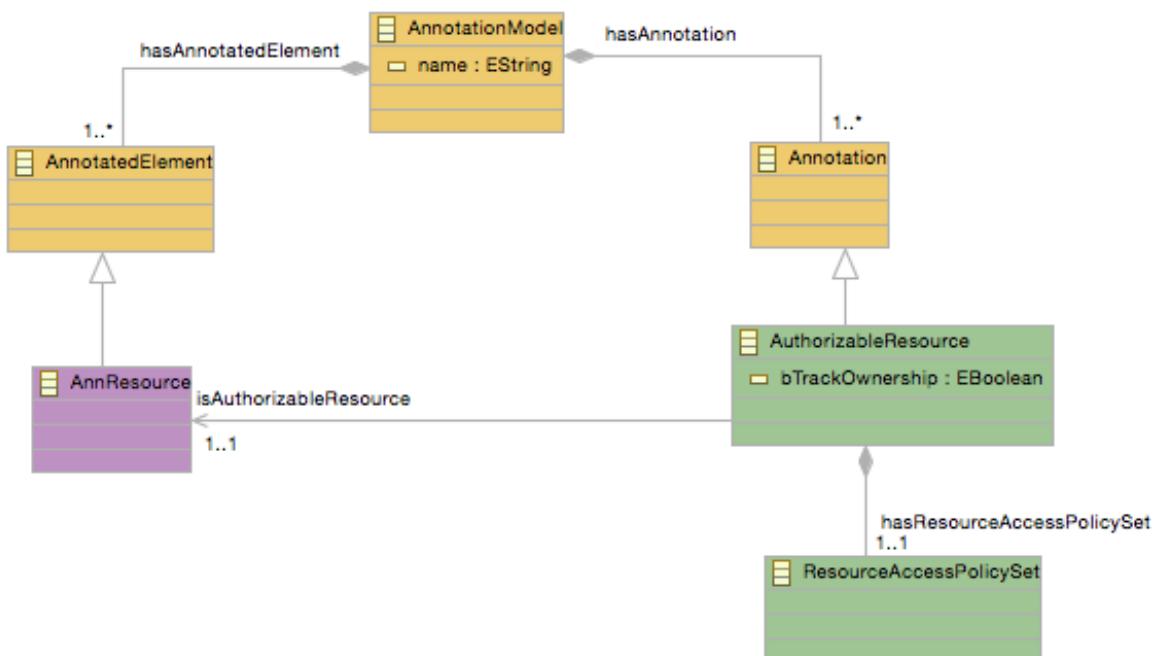


Figure 4-13 AuthorizableResource annotation and its relations

Properties

Table 4-46 AuthorizableResource's Properties

Name	Type	Multiplicity	Explanation
bTrackOwnership	EBoolean	1	When this <i>EBoolean</i> property is set to true, then the ownership of the Core CIM <i>Resource</i> that is annotated with this <i>AuthorizableResource</i> annotation will be tracked. This can only be true if the annotated Core CIM <i>Resource</i> has its <i>isAlgorithmic</i> property set to false.

Relations

Table 4-47 AuthorizableResource's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
ResourceAccessPolicy	composition	1	The <i>AuthorizableResource</i> must have a composition association with exactly one <i>ResourceAccessPolicy</i> element. This defines the authorization policy of the annotated Core CIM <i>Resource</i> .
AnnResource	association	1	The <i>AuthorizableResource</i> must have an association with exactly one <i>AnnResource</i> element, which references a Core CIM <i>Resource</i> . That <i>Resource</i> will then be accessed only if the corresponding <i>ResourceAccessPolicy</i> is satisfied.

Behavioural Restrictions

The *AuthorizableResource* element of the *ABAC Authorization* CIM extension meta-model does not have any behavioural restriction.

4.2.5.2.10 ResourceAccessPolicySet Element

Overview

The *ResourceAccessPolicySet* element models the set of access policies that can be used in order to define under which conditions a Core CIM *Resource* can be accessed. Figure 4-14 demonstrates the *ResourceAccessPolicySet* element of the *ABAC Authorization* CIM extension meta-model and its relations.



Figure 4-14 ResourceAccessPolicySet element and its relations

Properties

The *ResourceAccessPolicySet* element of the *ABAC Authorization* CIM extension meta-model does not have any property.

Relations

Table 4-48 ResourceAccessPolicySet's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
RuleCombiningAlgorithm	composition	1	The <i>ResourceAccessPolicySet</i> must have a composition association with exactly one <i>RuleCombiningAlgorithm</i> element. That <i>RuleCombiningAlgorithm</i> specifies the way that the rules of the <i>ResourceAccessPolicySet</i> are evaluated in order to permit or deny access to a specific resource.
ResourceAccessPolicy	composition	1..*	The <i>ResourceAccessPolicySet</i> must have at least one composition association with a <i>ResourceAccessPolicy</i> element. That <i>ResourceAccessPolicy</i> specifies a set of rules that partially define the access policy of a <i>Resource</i> .

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of a *ResourceAccessPolicySet* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the *ABAC Authorization* CIM meta-model that can be found in appendix A.1.

- *hasEitherDenyOrPermitRule*: This OCL constraint checks whether the *RuleCombiningAlgorithm* of any *ResourceAccessPolicySet* is specialized to either *PermitOverridesRule* or *DenyOverridesRule*.

4.2.5.2.11 ResourceAccessPolicy Element

Overview

The *ResourceAccessPolicy* element models one access policy of a resource that can be used in order to define under which conditions a Core CIM *Resource* can be accessed. Figure 4-15 demonstrates the *ResourceAccessPolicy* element of the *ABAC Authorization* CIM extension meta-model and its relations.



Figure 4-15 ResourceAccessPolicy element and its relations.

Properties

The *ResourceAccessPolicy* element of the *ABAC Authorization* CIM extension meta-model does not have any property.

Relations

Table 4-49 ResourceAccessPolicy's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
RuleCombiningAlgorithm	composition	1	The <i>ResourceAccessPolicy</i> must have a composition association with exactly one <i>RuleCombiningAlgorithm</i> element. That <i>RuleCombiningAlgorithm</i> specifies the way that the rules of the <i>ResourceAccessPolicy</i> are evaluated in order to permit or deny access to a specific resource.
ResourceAccessRule	composition	1..*	The <i>ResourceAccessPolicy</i> must have at least one composition association with a <i>ResourceAccessRule</i> element. These <i>ResourceAccessRules</i> specify the conditions that must be met in order to grant access permission to a client.

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of a *ResourceAccessPolicy* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the *ABAC Authorization* CIM meta-model that can be found in appendix A.1.

- *hasEitherDenyOrPermitRule*: This OCL constraint checks whether the *RuleCombiningAlgorithm* of any *ResourceAccessPolicy* is specialized to either *PermitOverridesRule* or *DenyOverridesRule*.

4.2.5.2.12 ***RuleCombiningAlgorithm Element***

Overview

The *RuleCombiningAlgorithm* element models the access rules evaluation algorithm that will be used when evaluating either a *ResourceAccessPolicy* or a *ResourceAccessPolicySet*. It is always specialized to either a *DenyOverridesAlgorithm* or a *PermitOverridesAlgorithm*.

Properties

The *RuleCombiningAlgorithm* element of the *ABAC Authorization* CIM extension meta-model does not have any property.

Relations

The *RuleCombiningAlgorithm* element of the *ABAC Authorization* CIM extension meta-model does not have any relation.

Behavioural Restrictions

The *RuleCombiningAlgorithm* element of the *ABAC Authorization* CIM extension meta-model does not have any behavioural restriction.

4.2.5.2.13 ***DenyOverridesAlgorithm Element***

Overview

The *DenyOverridesAlgorithm* element models a rule set evaluation algorithm which denies access to a corresponding resource when at least one of the rules yields denial of access for any specific access request.

Properties

The *DenyOverridesAlgorithm* element of the *ABAC Authorization* CIM extension meta-model does not have any property.

Relations

The *DenyOverridesAlgorithm* element of the *ABAC Authorization* CIM extension meta-model does not have any relation.

Behavioural Restrictions

The *DenyOverridesAlgorithm* element of the *ABAC Authorization* CIM extension meta-model does not have any behavioural restriction.

4.2.5.2.14 PermitOverridesAlgorithm Element

Overview

The *PermitOverridesAlgorithm* element models a rule set evaluation algorithm that permits the access to a specific resource whenever there exists at least one rule that yields access permission.

Properties

The *PermitOverridesAlgorithm* element of the *ABAC Authorization* CIM extension meta-model does not have any property.

Relations

The *PermitOverridesAlgorithm* element of the *ABAC Authorization* CIM extension meta-model does not have any relation.

Behavioural Restrictions

The *PermitOverridesAlgorithm* element of the *ABAC Authorization* CIM extension meta-model does not have any behavioural restriction.

4.2.5.2.15 ResourceAccessRule Element

Overview

The *ResourceAccessRule* element models an access rule of a specific resource. The evaluation of every rule can either yield a denial of access or permission. The *ResourceAccessRule* element is always specialized to either *ResourceAccessPermitRule* or *ResourceAccessDenyRule*. Figure 4-16 demonstrates the *ResourceAccessRule* element of the *ABAC Authorization* CIM extension meta-model and its relations.

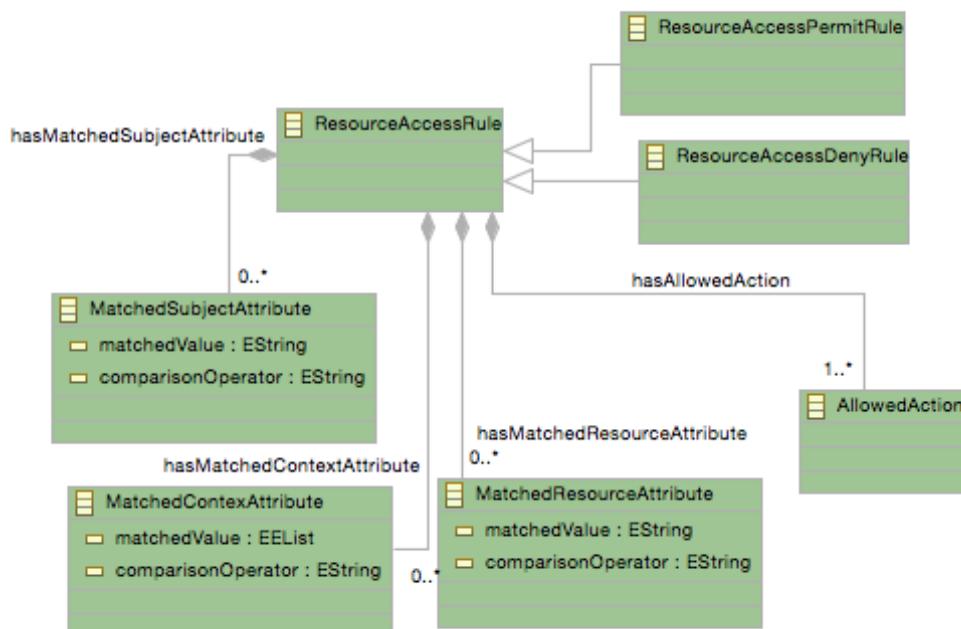


Figure 4-16 ResourceAccessRule element and its relations

Properties

The *ResourceAccessRule* element of the *ABAC Authorization* CIM extension meta-model does not have any property.

Relations

Table 4-50 ResourceAccessRule's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
AllowedAction	composition	1..*	The <i>ResourceAccessRule</i> must have at least one composition association with an <i>AllowedAction</i> element. This association models the fact that any specific resource access rule may permit or deny the access to a set of resource actions.
MatchedResourceAttribute	composition	0..*	The <i>ResourceAccessRule</i> can have zero or more composition associations with <i>MatchedResourceAttributes</i> . These <i>MatchedResourceAttributes</i> are a set of properties that a resource to which access is requested, which must be satisfied, in order to be accessed as it is dictated by the ABAC authorization scheme.
MatchedContextAttribute	composition	0..*	The <i>ResourceAccessRule</i> can have zero or more composition associations with <i>MatchedContextAttributes</i> . These <i>MatchedContextAttributes</i> are a set of properties of the resource context that are also taken into account when evaluating denial or permission of access to a resource. For example context attributes can be properties of related resources of any specific resource.
MatchedSubjectAttribute	composition	0..*	The <i>ResourceAccessRule</i> can have zero or more composition associations with <i>MatchedSubjectAttributes</i> . These are properties either of the Core CIM <i>Resource</i> that is used as an authorization model, or extra properties added as <i>SubjectAttributes</i> .

Behavioural Restrictions

The *ResourceAccessRule* element of the *ABAC Authorization* CIM extension meta-model does not have any behavioural restriction.

4.2.5.2.16 ResourceAccessPermitRule Element

Overview

The *ResourceAccessPermitRule* element models a *ResourceAccessRule* specialization. If a *ResourceAccessRule* is specialized as *ResourceAccessPermitRule*, then, when that rule evaluates positively in regards to an access request, access is granted to the client.

Properties

The *ResourceAccessPermitRule* element of the *ABAC Authorization* CIM extension meta-model does not have any property.

Relations

The *ResourceAccessPermitRule* element of the *ABAC Authorization* CIM extension meta-model does not have any relation.

Behavioural Restrictions

The *ResourceAccessPermitRule* element of the *ABAC Authorization* CIM extension meta-model does not have any behavioural restriction.

4.2.5.2.17 *ResourceAccessDenyRule* Element

Overview

The *ResourceAccessDenyRule* element models a specialization of a *ResourceAccessRule* element. If a *ResourceAccessRule* is specialized as *ResourceAccessDenyRule*, then, when this rule evaluates positively in regard with a specific client request, it denies access to it.

Properties

The *ResourceAccessDenyRule* element of the *ABAC Authorization* CIM extension meta-model does not have any property.

Relations

The *ResourceAccessDenyRule* element of the *ABAC Authorization* CIM extension meta-model does not have any relation.

Behavioural Restrictions

The *ResourceAccessDenyRule* element of the *ABAC Authorization* CIM extension meta-model does not have any behavioural restriction.

4.2.5.2.18 *MatchedSubjectAttribute* Element

Overview

The *MatchedSubjectAttribute* element models a property of the authorization model that is used by the envisioned system for authorization. That property can be either an existing Core CIM *Resource* property of the *Resource* that is annotated with the *AuthorizationSubject* annotation, or an extra property that is modelled as a *SubjectAttribute*. Figure 4-17 demonstrates the *MatchedSubjectAttribute* element of the ABAC Authorization CIM extension meta-model and its relations.

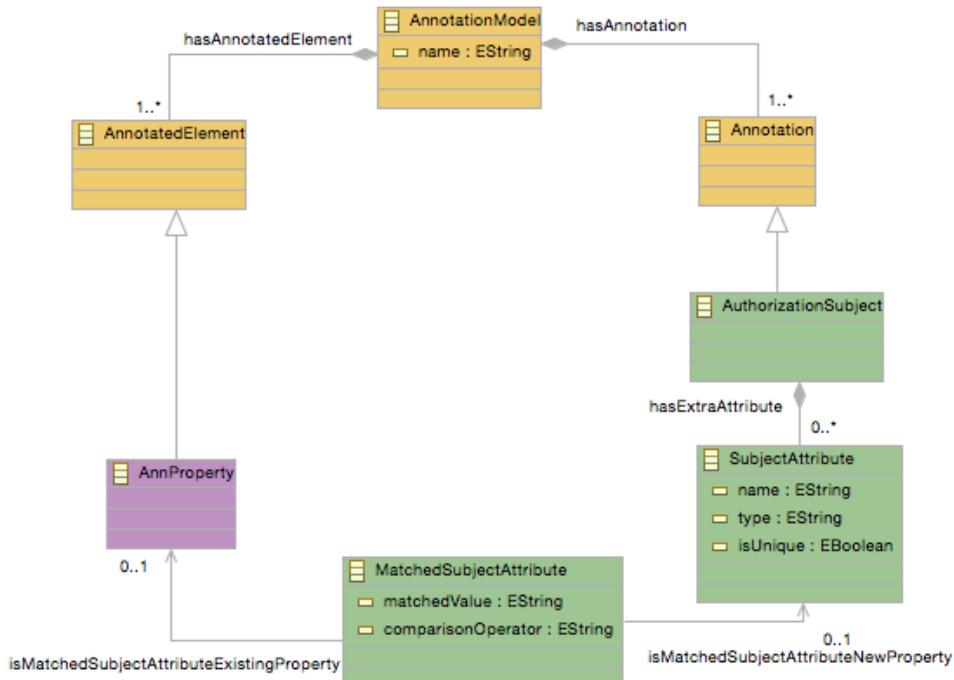


Figure 4-17 MatchedSubjectAttribute element and its relations

Properties

Table 4-51 MatchedSubjectAttribute's Properties

Name	Type	Multiplicity	Explanation
matchedValue	EString	1..*	Every <i>MatchedSubjectAttribute</i> has a list of values, which are compared by using the <i>comparisonOperator</i> of it, with the runtime value of the corresponding property of the client's authorization model property.
comparisonOperator	EString	1	Every <i>MatchedSubjectAttribute</i> has exactly one <i>comparisonOperator</i> that is used to compare the runtime value of a client's authorization property value with the possible <i>matchedValues</i> of a <i>MatchedSubjectAttribute</i> , every time the overlying rule is evaluated for access.

Relations

Table 4-52 MatchedSubjectAttribute's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
SubjectAttribute	association	0..1	The <i>MatchedSubjectAttribute</i> may have zero or one association with <i>SubjectAttribute</i> elements. If it has one, then this specific <i>MatchedSubjectAttribute</i> has a list of values that will be compared with the runtime value of a <i>SubjectAttribute</i> in order to evaluate an overlying rule for resource access.
AnnProperty	association	0..1	The <i>MatchedSubjectAttribute</i> may have zero or one associated <i>AnnProperty</i> elements. If it has one, then this specific <i>MatchedSubjectAttribute</i> has a list of values that will be compared with the runtime value of an existing Core CIM Resource <i>Property</i> that is annotated with the <i>AuthorizationSubject</i> annotation, in order to evaluate an overlying authorization rule for resource access.

Behavioural Restrictions

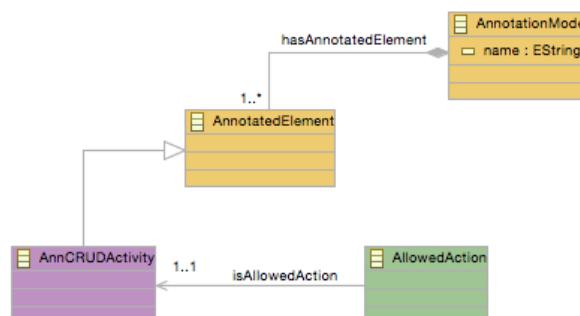
This subsection lists all the behavioural restrictions of the properties and relations of a *MatchedSubjectAttribute* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the *ABAC Authorization* CIM meta-model that can be found in appendix A.1.

- *isEitherSubjectOrCorePropertyAttribute*: This OCL constraint checks whether the *MatchedSubjectAttribute* has only an association with a *SubjectAttribute* but not with an *AnnProperty*, or the opposite. In any case, it verifies that a *MatchedSubjectAttribute* cannot be associated with both.

4.2.5.2.19 AllowedAction Element

Overview

The *AllowedAction* element models an action that can be allowed or denied when a resource access request is evaluated. Figure 4-18 demonstrates the *AllowedAction* element of the *ABAC Authorization* CIM extension meta-model and its relations.

**Figure 4-18 AllowedAction element and its relations.**

Properties

The *AllowedAction* element of the *ABAC Authorization* CIM extension meta-model does not have any property.

Relations

Table 4-53 AllowedActions's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
AnnCRUDActivity	association	1	The <i>AllowedAction</i> must be associated with exactly one <i>AnnCRUDActivity</i> element, which references a Core CIM <i>CRUDActivity</i> . Access to that <i>CRUDActivity</i> could be granted depending on the evaluation of the overlying access rule.

Behavioural Restrictions

The *AllowedAction* element of the *ABAC Authorization* CIM extension meta-model does not have any behavioural restriction.

4.2.5.2.20 *MatchedResourceAttribute* Element

Overview

The *MatchedResourceAttribute* element models a property of a resource to which access is requested, that is used when an overlying rule is evaluated. Figure 4-19 demonstrates the *MatchedResourceAttribute* element of the *ABAC Authorization* CIM extension meta-model and its relations.

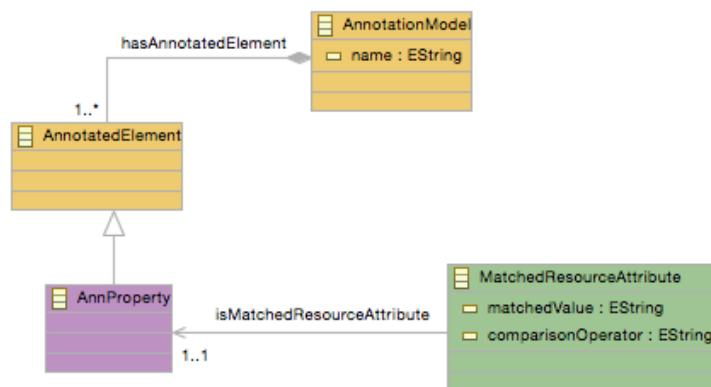


Figure 4-19 MatchedResourceAttribute element and its relations.

Properties

Table 4-54 MatchedResourceAttribute's Properties

Name	Type	Multiplicity	Explanation
matchedValue	EString	1..*	This is a list with values with which the runtime of a resource property will be compared when an overlying access rule is evaluated. The comparison is done using the according <i>comparisonOperator</i> .
comparisonOperator	EString	1	The <i>MatchedResourceAttribute</i> has exactly one <i>comparisonOperator</i> property, that specifies the way the runtime value of a resource property will be compared with the <i>matchValues</i> when an overlying access rule is evaluated.

Relations

Table 4-55 MatchedResourceAttribute's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
AnnProperty	association	1	The <i>MatchedResourceAttribute</i> must be associated with exactly one <i>AnnProperty</i> element, which references a Core CIM <i>Property</i> of a <i>Resource</i> .

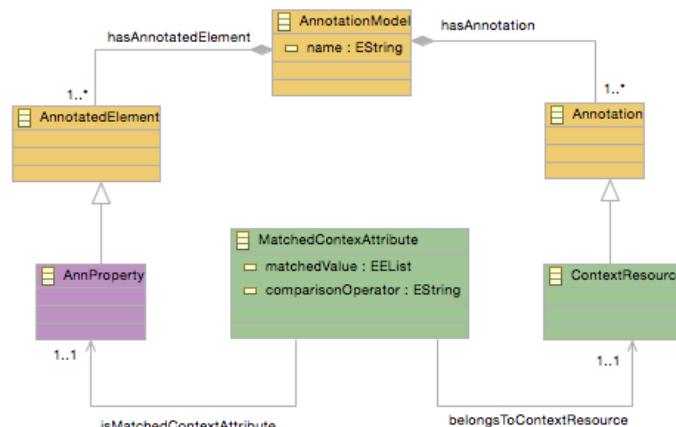
Behavioural Restrictions

The *MatchedResourceAttribute* element of the *ABAC Authorization* CIM extension meta-model does not have any behavioural restriction.

4.2.5.2.21 MatchedContextAttribute Element

Overview

The *MatchedContextAttribute* element models a property of a resource that belongs in the execution context of a resource to which access request is sent. Figure 4-20 demonstrates the *MatchedResourceAttribute* element of the *ABAC Authorization* CIM extension meta-model and its relations.

**Figure 4-20 MatchedContextAttribute element and its relations.**

Properties

Table 4-56 MatchedContextAttribute's Properties

Name	Type	Multiplicity	Explanation
matchedValue	EString	1..*	This is a list of values with which the runtime value of a resource's property that belongs in the execution context of a resource to which a client requests access, must be compared. The comparison is done using the corresponding <i>comparisonOperator</i> .
comparisonOperator	EString	1	The <i>MatchedContextAttribute</i> has exactly one <i>comparisonOperator</i> that is used to compare the runtime value of a resource's property that belongs in the execution context of a resource to which a client requests access, when the overlying rule is evaluated for access permission.

Relations

Table 4-57 MatchedContextAttribute's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
ContextResource	association	1	The <i>MatchedContextAttribute</i> must have an association with exactly one <i>ContextResource</i> element. This is the Core CIM Resource to which the property modelled by the <i>MatchedContextAttribute</i> , belongs to.
AnnProperty	association	1	The <i>MatchedContextAttribute</i> must have an association with exactly one <i>AnnProperty</i> , which references a Core CIM Property.

Behavioural Restrictions

The *MatchedContextAttribute* element of the *ABAC Authorization* CIM extension meta-model does not have any behavioural restriction.

4.2.5.2.22 *ContextResource* Element

Overview

The *ContextResource* element models an annotation of the *ABAC Authorization* CIM extension meta-model that is intended to annotate an existent Core CIM Resource, which has its *isAlgorithmic* property set to *false*. If a Core CIM Resource is annotated as a *ContextResource*, it is then added to the execution context of an *AuthorizableResource* and its runtime value are used when an overlying access rule is evaluated in order to grant access permission to the client.

Properties

The *ContextResource* element of the *ABAC Authorization* CIM extension meta-model does not have any property.

Relations

Table 4-58 ContextResource's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
AnnResource	association	1	The <i>ContextResource</i> must have an association with exactly one <i>AnnResource</i> element, which references a Core CIM Resource. This Resource will then be considered to belong in the execution context of an <i>AuthorizableResource</i> , an underlying access rule of which is evaluated in order to grant permission access to the client.

Behavioural Restrictions

The *ContextResource* element of the *ABAC Authorization* CIM extension meta-model does not have any behavioural restriction.

5 Platform Independent Model UML Profile Extensions

5.1 PIM UML Profile Extension Goals

The following subsections describe the abstract design of the extra functionality that is automatically embedded to the envisioned systems that S-CASE produces. Following the meta-model extension mechanism that section 3.1 presents, the following Core PIM extensions allow the S-CASE MDE engine to embed to the systems it produces authentication and ABAC authorization mechanisms, external 3rd party service compositions as well as database searching capabilities.

5.2 PIM Authentication Ecore Meta-model Definition

5.2.1.1 Introduction

In order to fully define the Authentication Ecore PIM extension meta-model (Figure 5-1), its various aspects must be explained, demonstrated as well as documented. This is done in the following subsections following the format section 4.1.2.1 describes.

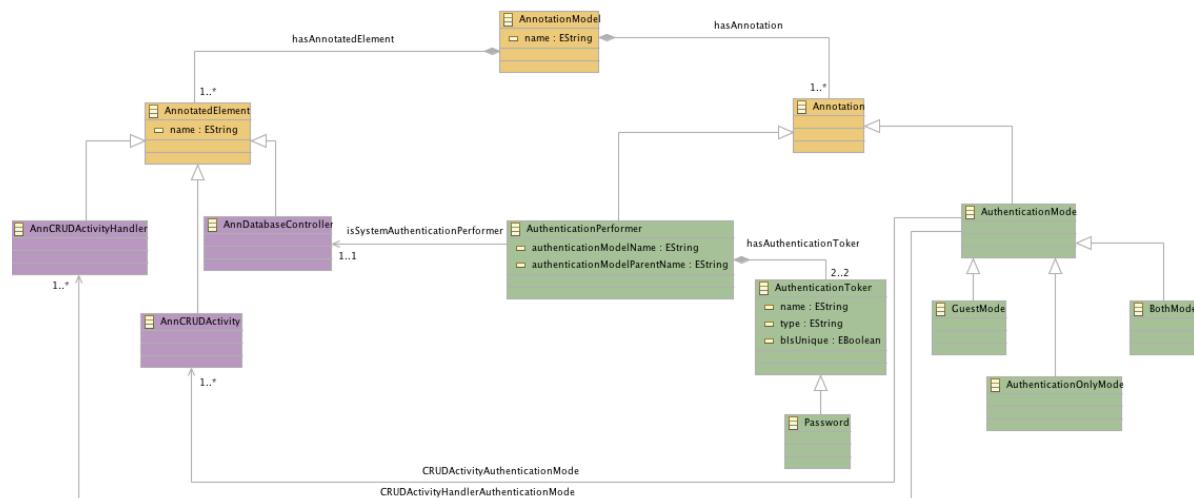


Figure 5-1 Authentication PIM extension meta-model

5.2.1.2 PIM Authentication Ecore Meta-model Elements

5.2.1.2.1 AnnotationModel Element

Overview

The *AnnotationModel* element is the root element of the *Authentication* PIM extension meta-model and comprises of all the introduced *Annotations* and the existing Core PIM elements to which these annotations are attached in order to alter their semantic meaning, design and/or configuration.

Properties

Table 5-1 AnnotationModel's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the name of the <i>Annotation</i> model.

Relations

Table 5-2 AnnotationModel's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
AnnotatedElement	composition	1..*	This composition association models the fact that every <i>AnnotationModel</i> element must have at least one <i>AnnotatedElement</i> .
Annotation	composition	1..*	This composition association models the fact that every <i>AnnotationModel</i> element must have at least one <i>Annotation</i> .

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of an *AnnotationModel* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the *Authorization* PIM extension meta-model that can be found in appendix A.1.

- *exactlyOneAnnDatabaseControllerExists*: This OCL constraint checks whether there exists exactly one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnDatabaseController*.
- *atLeastOneAnnCRUDActivityExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that are of type *AnnCRUDActivity*.
- *atLeastOneAnnCRUDActivityHandler*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnCRUDActivityHandler*.
- *exactlyOneAuthenticationPerformerExists*: This OCL constraint checks whether there exists exactly one *Annotation* element that is associated with this *AnnotationModel* that is of type *AuthenticationPerformer*.
- *atLeastOneAuthenticationModeExists*: This OCL constraint checks whether there exists at least one *Annotation* element that is associated with this *AnnotationModel* that is either of type *GuestMode*, *AuthenticationOnlyMode* or *BothMode*.
- *allAuthenticationModesAreSpecialized*: This OCL constraint checks whether every *Annotation* element that is associated with this *AnnotationModel* that is of type *AuthenticationMode* is further specialized to either *GuestMode*, *AuthenticationOnlyMode* or *BothMode*.

5.2.1.2.2 *AnnotatedElement* Element

Overview

The *AnnotatedElement* element models any existent Core PIM meta-model element that can be annotated by the *Authentication* Core PIM extension in order to extend its semantic meaning and/or design. It is always further specialized to more specific categories of annotated elements.

Properties

Table 5-3 AnnotatedElement's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the name of the <i>AnnotatedElement</i> .

Relations

The *AnnotatedElement* element of the *Authentication PIM* extension meta-model does not have any relation.

Behavioural Restrictions

The *AnnotatedElement* element of the *Authentication PIM* extension meta-model does not have any behavioural restriction.

5.2.1.2.3 *Annotation* Element

Overview

The *Annotation* element models any new semantic meaning or design that is attached to a Core PIM meta-model element. It is always further specialized to more specific categories of annotations.

Properties

The *Annotation* element of the *Authentication PIM* extension meta-model does not have any property.

Relations

The *Annotation* element of the *Authentication PIM* extension meta-model does not have any relation.

Behavioural Restrictions

The *Annotation* element of the *Authentication PIM* extension meta-model does not have any behavioural restriction.

5.2.1.2.4 *AnnCRUDActivityHandler* Element

Overview

The *AnnCRUDActivityHandler* element models an existent *CRUDActivityHandler* of the Core PIM model that can be annotated by the *Authentication* PIM extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnCRUDActivityHandler* element of the *Authentication PIM* extension meta-model does not have any property.

Relations

Table 5-4 AnnCRUDActivityHandler's Relations

Relation With PIM Element	Type	Multiplicity	Structural Constraints
CRUDActivityHandler	reference	1	The <i>AnnCRUDActivityHandler</i> must have exactly one reference to one <i>CRUDActivityHandler</i> of the Core PIM model.

Behavioural Restrictions

The *AnnCRUDActivityHandler* element of the *Authentication PIM* extension meta-model does not have any behavioural restriction.

5.2.1.2.5 AnnCRUDActivity Element

Overview

The *AnnCRUDActivity* element models an existent *CRUDActivity* of the Core PIM model that can be annotated by the *Authentication PIM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnCRUDActivity* element of the *Authentication PIM* extension meta-model does not have any property.

Relations

Table 5-5 AnnCRUDActivity's Relations

Relation With PIM Element	Type	Multiplicity	Structural Constraints
CRUDActivity	reference	1	The <i>AnnCRUDActivity</i> must reference exactly one <i>CRUDActivity</i> element of the Core PIM meta-model.

Behavioural Restrictions

The *AnnCRUDActivity* element of the *Authentication PIM* extension meta-model does not have any behavioural restriction.

5.2.1.2.6 *AnnDatabaseController Element*

Overview

The *AnnDatabaseController* element models an existent *DatabaseController* of the Core PIM model that can be annotated by the *Authentication* PIM extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnDatabaseController* element of the *Authentication PIM* extension meta-model does not have any property.

Relations

Table 5-6 AnnDatabaseController's Relations

Relation With PIM Element	Type	Multiplicity	Structural Constraints
DatabaseController	reference	1	The <i>AnnDatabaseController</i> must reference exactly one <i>DatabaseController</i> of the Core PIM meta-model.

Behavioural Restrictions

The *AnnDatabaseController* element of the *Authentication PIM* extension meta-model does not have any behavioural restriction.

5.2.1.2.7 *AuthenticationPerformer Element*

Overview

The *AuthenticationPerformer* element models an annotation of the *Authentication* PIM extension meta-model that is intended to annotate an existent Core PIM *DatabaseController*. With this annotation, the *DatabaseController* will also embed the needed functionality to query the local envisioned system's database requested authentication data. Figure 5-2 demonstrates the *AuthenticationPerformer* element of the *Authentication* PIM extension meta-model and its relations.

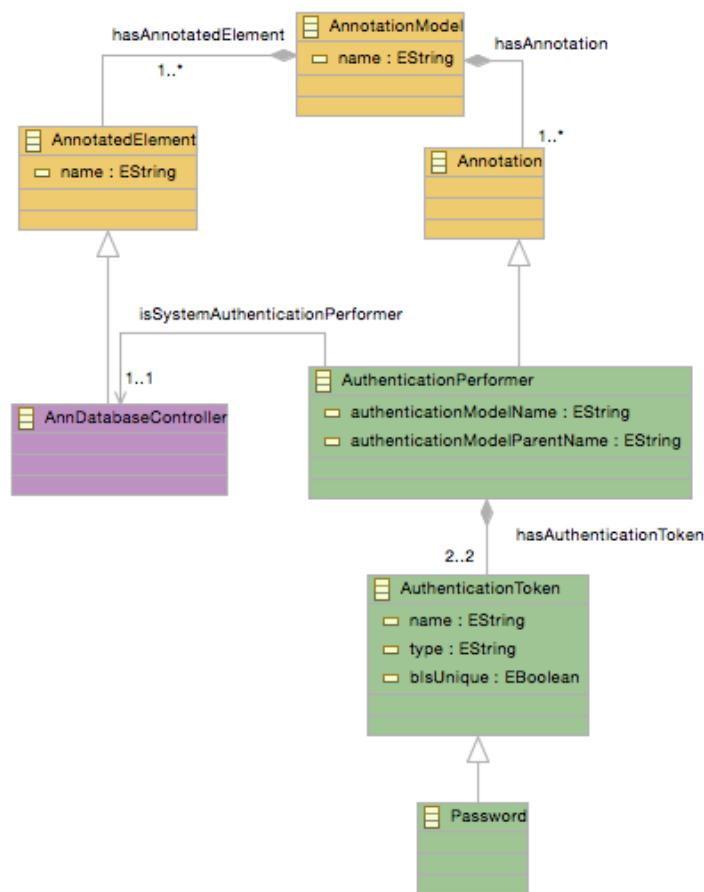


Figure 5-2 AuthenticationPerformer annotation and its relations.

Properties

Table 5-7 AuthenticationPerformer's Properties

Name	Type	Multiplicity	Explanation
authenticationmodelName	EString	1	This is the name of the Core PIM meta-model <i>ResourceModel</i> that is used as authentication model.
authenticationModelParentName	EString	1	This is the name of the Core PIM meta-model <i>ResourceModel</i> 's parent, which is used as authentication model.

Relations

Table 5-8 AuthenticationPerformer's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
AnnDatabaseController	association	1	The <i>AuthenticationPerformer</i> must be associated with exactly one <i>AnnDatabaseController</i> element, which references the Core PIM <i>DatabaseController</i> .

			With this annotation, the <i>DatabaseController</i> will also embed the needed functionality to query the envisioned system's local database for authentication data.
AuthenticationToken	composition	2	The <i>AuthenticationPerformer</i> must have composition associations with exactly two <i>AuthenticationTokens</i> . These will be used as credentials (password/username) for authentication.

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of a *AuthenticationPerformer* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the PIM meta-model that can be found in appendix A.1.

- *hasExactlyOnePasswordToken*: This OCL constraint checks whether exactly one of the two *AuthenticationTokens* with which the *AuthenticationPerformer* is associated, is of type *Password*.

5.2.1.2.8 AuthenticationToken Element

Overview

The *AuthenticationToken* element models an authentication token such as a password or username that will be used as credentials in the authentication process of the envisioned system.

Properties

Table 5-9 AuthenticationToken's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	The <i>name</i> property is the name of the authentication token.
type	EString	1	The <i>type</i> property is the type of the authentication token.
isUnique	EString	1	The <i>isUnique</i> property models the authentication token's multiplicity. If it is <i>true</i> , then the multiplicity is one.

Relations

The *AuthenticationToken* element of the *Authentication* PIM extension meta-model does not have any relation.

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of a *AuthenticationToken* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the PIM meta-model that can be found in appendix A.1.

- *hasMultiplicityOne*: This OCL constraint checks whether the selected *AuthenticationToken* is of multiplicity one.

5.2.1.2.9 Password Element

Overview

The *Password* element models a specialization of the *AuthenticationToken* element, which alters further its meaning as password.

Properties

The *Password* element of the *Authentication PIM* extension meta-model does not have any property other than those inherited from the *AuthenticationToken* element.

Relations

The *Password* element of the *Authentication PIM* extension meta-model does not have any relation other than those inherited from the *AuthenticationToken* element.

Behavioural Restrictions

The *Password* element of the *Authentication PIM* extension meta-model does not have any behavioural restriction.

5.2.1.2.10 AuthenticationMode Element

Overview

The *AuthenticationMode* element models an annotation of the *Authentication PIM* extension meta-model that is intended to annotate an existent Core PIM *CRUDActivity* or *CRUDActivityHandler*. This annotation embeds the needed information of the authentication mode to be used when a client makes a request to a specific action of the envisioned system's resource. The *AuthenticationMode* element is always further specialized to *GuestMode*, *AuthorizationOnlyMode* or *BothMode*. Figure 5-3 demonstrates the *AuthenticationMode* element of the *Authentication PIM* extension meta-model and its relations.

Properties

The *AuthenticationMode* element of the *Authentication PIM* extension meta-model does not have any property.

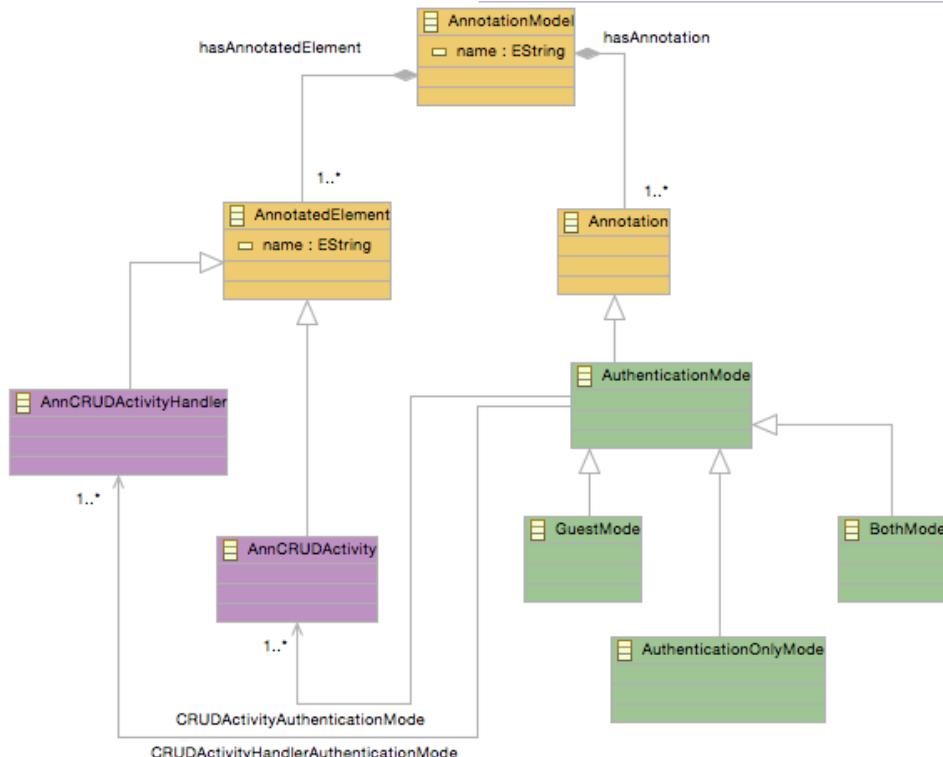


Figure 5-3 AuthenticationModel annotation and its relations.

Relations

Table 5-10 AuthenticationMode's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
AnnCRUDActivity	association	1	The <i>AuthenticationMode</i> must be associated with exactly one <i>AnnCRUDActivity</i> element, which references a Core PIM <i>CRUDActivity</i> . That <i>CRUDActivity</i> will then embed the requested authentication type.
AnnCRUDActivityHandler	association	1	The <i>AuthenticationMode</i> must be associated with exactly one <i>AnnCRUDActivityHandler</i> element, which references a Core PIM <i>CRUDActivityHandler</i> . That <i>CRUDActivityHandler</i> will then embed the requested authentication type.

Behavioural Restrictions

The *AuthenticationMode* element of the *Authentication PIM* extension meta-model does not have any behavioural restriction.

5.2.1.2.11 *GuestMode Element*

Overview

The *GuestMode* element models an annotation of the *Authentication* PIM extension meta-model that is intended to annotate an existent Core PIM *CRUDActivity* or *CRUDActivityHandler*. With this annotation, the *CRUDActivity* or the *CRUDActivityHandler* will only allow guest users of the envisioned web service system to access their underlying resource.

Properties

The *GuestMode* element of the *Authentication* PIM extension meta-model does not have any property other than those inherited from the *AuthenticationMode* element.

Relations

The *GuestMode* element of the *Authentication* PIM extension meta-model does not have any relation other than those inherited from the *AuthenticationMode* element.

Behavioural Restrictions

The *GuestMode* element of the *Authentication* PIM extension meta-model does not have any behavioural restriction.

5.2.1.2.12 *AuthenticationOnlyMode Element*

Overview

The *AuthenticationOnlyMode* element models a specialization of the *AuthenticationMode* annotation of the *Authentication* PIM extension meta-model that is intended to annotate an existent Core PIM *CRUDActivity* or *CRUDActivityHandler*. With this annotation, the *CRUDActivity* or the *CRUDActivityHandler* will only allow authenticated users of the envisioned web service system to access the underlying resource.

Properties

The *AuthenticationOnlyMode* element of the *Authentication* PIM extension meta-model does not have any property other than those inherited from the *AuthenticationMode* element.

Relations

The *AuthenticationOnlyMode* element of the *Authentication* PIM extension meta-model does not have any relation other than those inherited from the *AuthenticationMode* element.

Behavioural Restrictions

The *AuthenticationOnlyMode* element of the *Authentication* PIM extension meta-model does not have any behavioural restriction.

5.2.1.2.13 BothMode Element

Overview

The *BothMode* element models a specialization of the *AuthenticationMode* annotation of the *Authentication PIM* extension meta-model that is intended to annotate an existent Core PIM *CRUDActivity* or *CRUDActivityHandler*. With this annotation, the *CRUDActivity* or the *CRUDActivityHandler* will allow any type of user, authenticated or guest, of the envisioned web service system to access the underlying resource.

Properties

The *BothMode* element of the *Authentication PIM* extension meta-model does not have any property other than those inherited from the *AuthenticationMode* element.

Relations

The *BothMode* element of the *Authentication PIM* extension meta-model does not have any relation other than those inherited from the *AuthenticationMode* element.

Behavioural Restrictions

The *BothMode* element of the *Authentication PIM* extension meta-model does not have any behavioural restriction.

5.3 PIM Database Searching Ecore Meta-model Definition

5.3.1.1 Introduction

In order to fully define the *Database Searching* Ecore PIM extension meta-model (Figure 5-4), its various aspects must be explained, demonstrated as well as documented. This is done in the following subsections following the format section 4.1.2.1 describes.

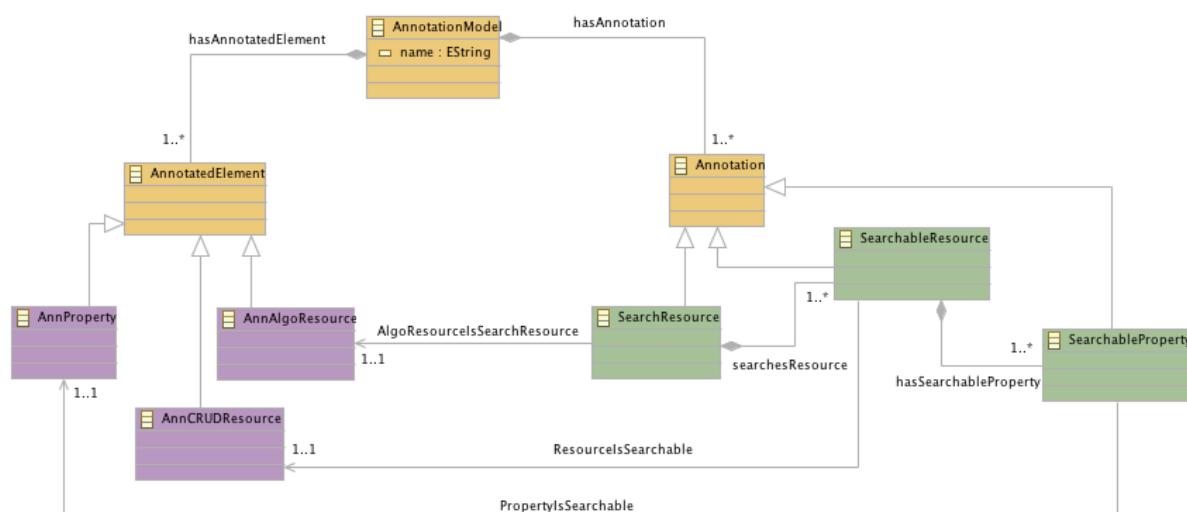


Figure 5-4 Database Searching PIM extension meta-model

5.3.1.2 PIM Database Searching Ecore Meta-model Elements

5.3.1.2.1 *AnnotationModel Element*

Overview

The *AnnotationModel* element is the root element of the *Database Searching* PIM extension meta-model and comprises of all the introduced *Annotations* and the existing Core PIM elements to which these annotations are attached in order to alter their semantic meaning, design and/or configuration.

Properties

Table 5-11 *AnnotationModel's Properties*

Name	Type	Multiplicity	Explanation
name	EString	1	This is the name of the <i>Annotation</i> model.

Relations

Table 5-12 *AnnotationModel's Relations*

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
AnnotatedElement	composition	1..*	This composition association models the fact that every <i>AnnotationModel</i> element must have at least one <i>AnnotatedElement</i> .
Annotation	composition	1..*	This composition association models the fact that every <i>AnnotationModel</i> element must have at least one <i>Annotation</i> .

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of an *AnnotationModel* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the *Database Searching* PIM extension meta-model that can be found in appendix A.1.

- *atLeastOneAnnAlgoResourceControllerExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnAlgoResourceController*.
- *atLeastOneAnnCRUDActivityExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnCRUDActivity*.
- *AnnCRUDActivityAndAnnResourceControllerAndAnnCRUDActivityHandlerAreEqual*: This OCL constraint checks whether the number of the *AnnCRUDActivities* that exist is the same with

the number of *AnnAlgoResourceControllers* and the number of the *AnnCRUDActivityHandlers*.

- *atLeastOneAnnCRUDActivityHandlerExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* element that is associated with this *AnnotationModel* that is of type *AnnCRUDActivityHandler*.
- *atLeastOneAnnResourceModelExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* element that is associated with this *AnnotationModel* that is of type *AnnResourceModel*.
- *atLeastOneAnnPIMComponentPropertyExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* element that is associated with this *AnnotationModel* that is of type *AnnPIMComponentProperty*.
- *atLeastOneSearchResourceExists*: This OCL constraint checks whether there exists at least one *Annotation* element that is associated with this *AnnotationModel* that is of type *SearchResource*.
- *atLeastOneSearchCRUDActivityExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* element that is associated with this *AnnotationModel* that is of type *SearchCRUDActivity*.
- *atLeastOneSearchCRUDActivityHandlerExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* element that is associated with this *AnnotationModel* that is of type *SearchCRUDActivityHandler*.

5.3.1.2.2 *AnnotatedElement* Element

Overview

The *AnnotatedElement* element models any existent Core PIM meta-model element that can be annotated by the *Database Searching* PIM extension in order to extend its semantic meaning and/or design. It is always further specialized to more specific categories of annotated elements.

Properties

The *AnnotatedElement* element of the *Database Searching PIM* extension meta-model does not have any property.

Relations

The *AnnotatedElement* element of the *Database Searching PIM* extension meta-model does not have any relation.

Behavioural Restrictions

The *AnnotatedElement* of the *Database Searching PIM* extension meta-model does not have any behavioural restriction.

5.3.1.2.3 *Annotation* Element

Overview

The *Annotation* element models any new semantic meaning or design that is attached to a Core PIM meta-model element. It is always further specialized to more specific categories of annotations.

Properties

The *Annotation* element of the *Database Searching PIM* extension meta-model does not have any property.

Relations

The *Annotation* element of the *Database Searching PIM* extension meta-model does not have any relation.

Behavioural Restrictions

The *Annotation* element of the *Database Searching PIM* extension meta-model does not have any behavioural restriction.

5.3.1.2.4 AnnPIMComponentProperty Element

Overview

The *AnnPIMComponentProperty* element models an existent *PIMComponentProperty* of the Core PIM model that can be annotated by the *Database Searching PIM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnPIMComponentProperty* element of the *Database Searching PIM* extension meta-model does not have any property.

Relations

Table 5-13 AnnPIMComponentProperty's Relations

Relation With PIM Element	Type	Multiplicity	Structural Constraints
PIMComponentProperty	reference	1	The <i>AnnPIMComponentProperty</i> must reference exactly one <i>PIMComponentProperty</i> of the Core PIM model.

Behavioural Restrictions

The *AnnPIMComponentProperty* element of the *Database Searching PIM* extension meta-model does not have any behavioural restriction.

5.3.1.2.5 AnnResourceModel Element

Overview

The *AnnResourceModel* element models an existent *ResourceModel* of the Core PIM model that can be annotated by the *Database Searching PIM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnResourceModel* element of the *Database Searching* PIM extension meta-model does not have any property.

Relations

Table 5-14 AnnResourceModel's Relations

Relation With PIM Element	Type	Multiplicity	Structural Constraints
ResourceModel	reference	1	The <i>AnnResourceModel</i> must reference exactly one <i>ResourceModel</i> of the Core PIM model.

Behavioural Restrictions

The *AnnResourceModel* element of the *Database Searching* PIM extension meta-model does not have any behavioural restriction.

5.3.1.2.6 AnnCRUDActivityHandler Element

Overview

The *AnnCRUDActivityHandler* element models an existent *CRUDActivityHandler* of the Core PIM model that can be annotated by the *Database Searching* PIM extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnCRUDActivityHandler* element of the *Database Searching* PIM extension meta-model does not have any property.

Relations

Table 5-15 AnnCRUDActivityHandler's Relations

Relation With PIM Element	Type	Multiplicity	Structural Constraints
CRUDActivityHandler	reference	1	The <i>AnnCRUDActivityHandler</i> must reference exactly one <i>CRUDActivityHandler</i> of the Core PIM model.

Behavioural Restrictions

The *AnnCRUDActivityHandler* element of the *Database Searching* PIM extension meta-model does not have any behavioural restriction.

5.3.1.2.7 *AnnCRUDActivity* Element

Overview

The *AnnCRUDActivity* element models an existent *CRUDActivity* of the Core PIM model that can be annotated by the *Database Searching* PIM extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnCRUDActivity* element of the *Database Searching* PIM extension meta-model does not have any property.

Relations

Table 5-16 *AnnCRUDActivity*'s Relations

Relation With PIM Element	Type	Multiplicity	Structural Constraints
CRUDActivity	reference	1	The <i>AnnCRUDActivity</i> must reference exactly one <i>CRUDActivity</i> of the Core PIM model.

Behavioural Restrictions

The *AnnCRUDActivity* element of the *Database Searching* PIM extension meta-model does not have any behavioural restriction.

5.3.1.2.8 *AnnAlgoResourceController* Element

Overview

The *AnnAlgoResourceController* element models an existent *AlgoResourceController* of the Core PIM model that can be annotated by the *Database Searching* PIM extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnAlgoResourceController* element of the *Database Searching* PIM extension meta-model does not have any property.

Relations

Table 5-17 *AnnAlgoResourceController*'s Relations

Relation With PIM Element	Type	Multiplicity	Structural Constraints
AlgoResourceController	reference	1	The <i>AnnAlgoResourceController</i> must reference exactly one <i>AlgoResourceController</i> of the Core

			PIM model.
--	--	--	------------

Behavioural Restrictions

The *AnnAlgoResourceController* element of the *Database Searching* PIM extension meta-model does not have any behavioural restriction.

5.3.1.2.9 SearchController Element

Overview

The *SearchController* element models an annotation of the *Database Searching* PIM extension meta-model that is intended to annotate an existent Core PIM *AlgoResourceController*. With this annotation, the *AlgoResourceController* will embed the appropriate web API, in order to be able to handle incoming database searching requests, as well as the needed functionality to call underlying software artefacts to satisfy the client request. Figure 5-5 demonstrates the *SearchController* element of the *Database Searching* PIM extension meta-model and its relations.

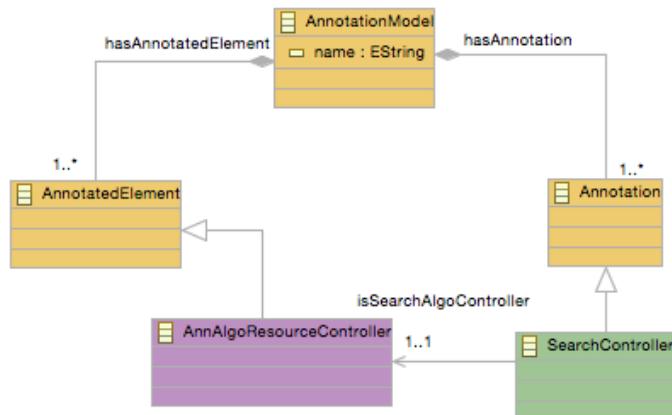


Figure 5-5 SearchController annotation and its relations.

Properties

The *SearchController* element of the *Database Searching* PIM extension meta-model does not have any property.

Relations

Table 5-18 SearchController's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
AnnAlgoResourceController	association	1	The <i>SearchController</i> must be associated with exactly one <i>AnnAlgoResourceController</i> element, which references an <i>AlgoResourceController</i> of the Core PIM model. That <i>AlgoResourceController</i> will then embed the needed code in order to handle incoming search requests.

Behavioural Restrictions

The *SearchResource* element of the *Database Searching* PIM extension meta-model does not have any behavioural restriction.

5.3.1.2.10 SearchCRUDActivity Element

Overview

The *SearchCRUDActivity* element models an annotation of the *Database Searching* PIM extension meta-model that is intended to annotate an existent Core PIM *CRUDActivity*. With this annotation, the *CRUDActivity* will implement the needed web API infrastructure in order to be able to accept search requests. Figure 5-6 demonstrates the *SearchCRUDActivity* element of the *Database Searching* PIM extension meta-model and its relations.

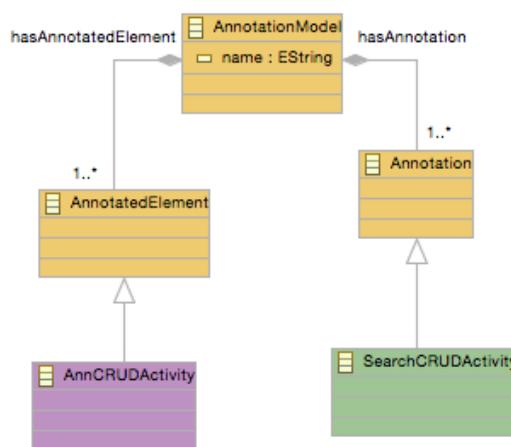


Figure 5-6 SearchCRUDActivity

Properties

The *SearchCRUDActivity* element of the *Database Searching* PIM extension meta-model does not have any property.

Relations

Table 5-19 SearchCRUDActivity's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
AnnCRUDActivity	association	1	The <i>SearchCRUDActivity</i> must be associated with exactly one <i>AnnCRUDActivity</i> , which references a <i>CRUDActivity</i> of the Core PIM model. With this annotation the <i>CRUDActivity</i> will implement the needed web API infrastructure in order to be able to accept search requests.

Behavioural Restrictions

The *SearchCRUDActivity* element of the *Database Searching* PIM extension meta-model does not have any behavioural restriction.

5.3.1.2.11 *SearchCRUDActivityHandler* Element

Overview

The *SearchCRUDActivityHandler* element models an annotation of the *Database Searching* PIM extension meta-model that is intended to annotate an existent Core PIM *CRUDActivityHandler*. With this annotation, the *CRUDActivityHandler* will embed the needed functionality in order to perform low level tasks (e.g. database queries) that are needed so as to answer client's search request. Figure 5-7 demonstrates the *SearchCRUDActivityHandler* element of the *Database Searching* PIM extension meta-model and its relations.

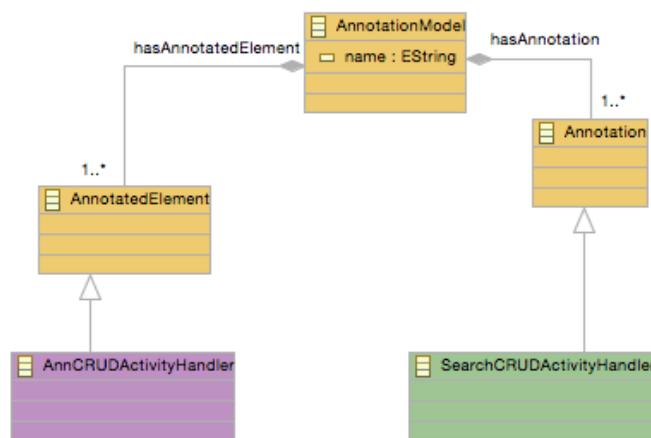


Figure 5-7 SearchCRUDActivityHandler

Properties

The *SearchCRUDActivityHandler* element of the *Database Searching* PIM extension meta-model does not have any property.

Relations

Table 5-20 SearchCRUDActivityHandler's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
AnnCRUDActivityHandler	association	1	The <i>SearchCRUDActivityHandler</i> must be associated with exactly one <i>AnnCRUDActivityHandler</i> , which references one <i>CRUDActivityHandler</i> of the Core PIM model. With this annotation, the <i>CRUDActivityHandler</i> will embed the needed functionality to fulfil low level tasks, such as database queries, in order to answer client's search requests.
SearchableResourceModel	composition	1..*	The <i>SearchCRUDActivityHandler</i> must have at least one composition association with a

			<i>SearchableResourceModel</i> . This association models the fact that a <i>SearchCRUDActivityHandler</i> must search at least one Core PIM <i>ResourceModel</i> in order to be meaningful.
--	--	--	---

Behavioural Restrictions

The *SearchCRUDActivityHandler* element of the *Database Searching* PIM extension meta-model does not have any behavioural restriction.

5.3.1.2.12 *SearchableResourceModel* Element

Overview

The *SearchableResourceModel* element models an annotation of the *Database Searching* PIM extension meta-model that is intended to annotate an existent Core PIM *ResourceModel*. With this annotation, the *ResourceModel* will embed the needed infrastructure in order to be searchable. Figure 5-8 demonstrates the *SearchableResourceModel* element of the *Database Searching* PIM extension meta-model and its relations.

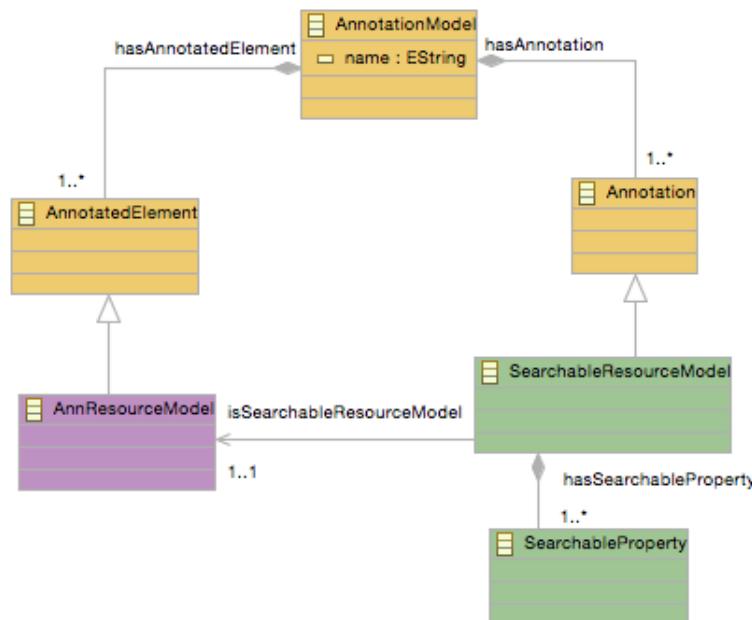


Figure 5-8 SearchableResourceModel

Properties

The *SearchableResourceModel* element of the *Database Searching* PIM extension meta-model does not have any property.

Relations

Table 5-21 SearchableResourceModel's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
AnnResourceModel	association	1	The <i>SearchableResourceModel</i> must be associated with exactly one <i>AnnResourceModel</i> , which references a Core PIM <i>ResourceModel</i> . With this annotation, the <i>ResourceModel</i> will embed the needed infrastructure in order to be searchable.
SearchableProperty	composition	1..*	The <i>SearchableResourceModel</i> must have at least one composition association with a <i>SearchableProperty</i> . This association models the fact that if a <i>ResourceModel</i> is annotated as <i>SearchableResourceModel</i> then it should have at least one property that is also annotated as <i>SearchableProperty</i> .

Behavioural Restrictions

The *SearchableResourceModel* element of the *Database Searching* PIM extension meta-model does not have any behavioural restriction.

5.3.1.2.13 SearchableProperty Element

Overview

The *SearchableProperty* element models an annotation of the *Database Searching* PIM extension meta-model that is intended to annotate an existent Core PIM *PIMComponentProperty*. With this annotation, the *PIMComponentProperty* will be implemented in such a way that it will be searchable. Figure 5-9 demonstrates the *SearchableProperty* element of the *Database Searching* PIM extension meta-model and its relations.

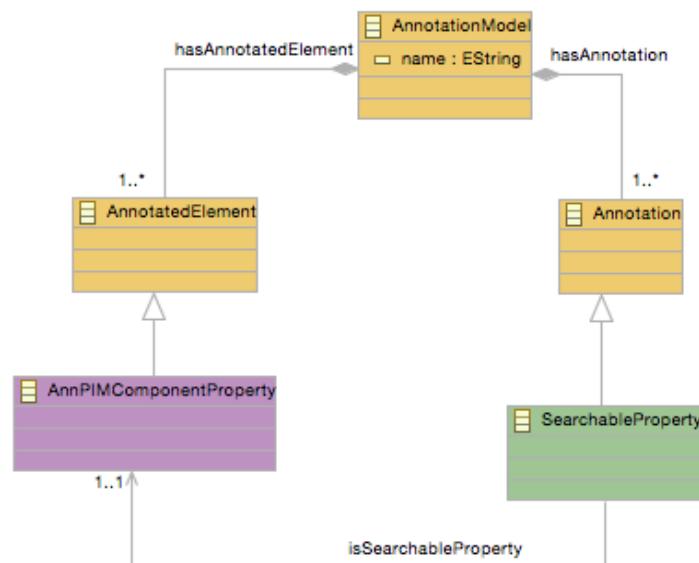


Figure 5-9 SearchableProperty annotation and its relations.

Properties

The *SearchableProperty* element of the *Database Searching* PIM extension meta-model does not have any property.

Relations

Table 5-22 SearchableProperty's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
AnnPIMComponentProperty	association	1	The <i>SearchableProperty</i> must be associated with exactly one <i>AnnPIMComponentProperty</i> , which references one <i>PIMComponentProperty</i> of the Core PIM model. With this annotation, the <i>PIMComponentProperty</i> will be implemented in such a way so that it will be searchable.

Behavioural Restrictions

The *SearchableProperty* element of the *Database Searching* PIM extension meta-model does not have any behavioural restriction.

5.4 PIM External Service Composition Ecore Meta-model Definition

5.4.1.1 Introduction

In order to fully define the *External Service Composition* Ecore PIM extension meta-model (Figure 5-10), its various aspects must be explained, demonstrated as well as documented. This is done in the following subsections following the format section 4.1.2.1 describes.

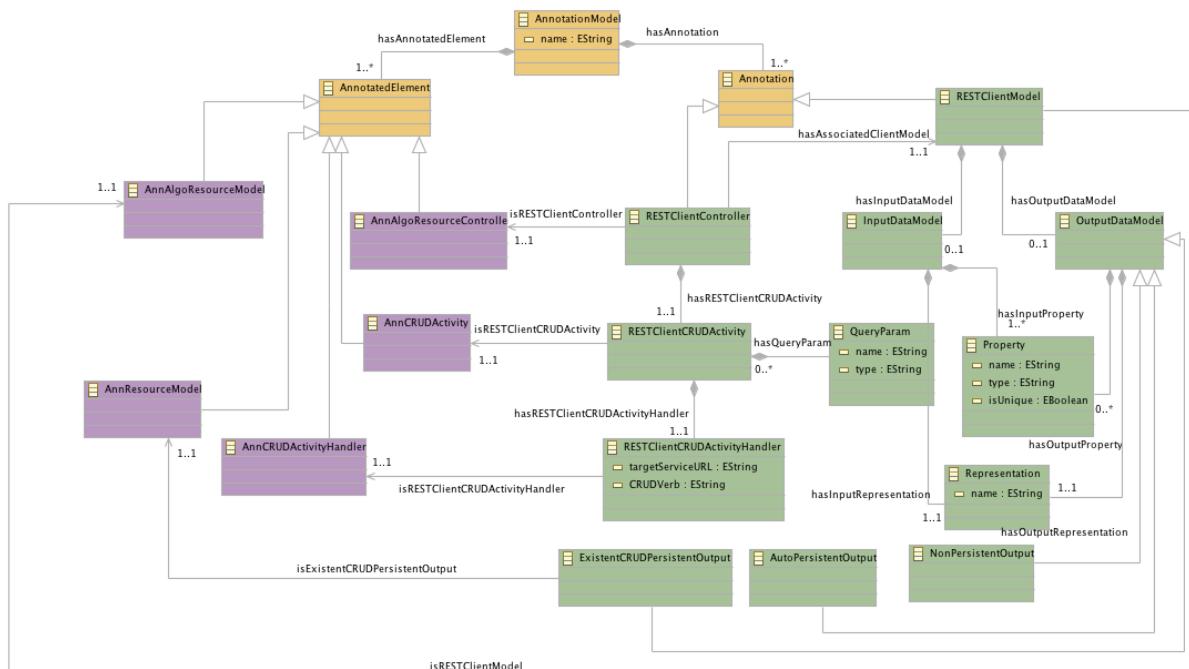


Figure 5-10 External Service Composition meta-model

5.4.1.2 PIM External Service Composition Ecore Meta-model Elements

5.4.1.2.1 AnnotationModel Element

Overview

The *AnnotationModel* element is the root element of the *External Service Composition* PIM extension meta-model and comprises of all the introduced *Annotations* and the existing Core PIM elements to which these annotations are attached in order to alter their semantic meaning, design and/or configuration.

Properties

Table 5-23 AnnotationModel's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the name of the <i>Annotation</i> model.

Relations

Table 5-24 AnnotationModel's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
AnnotatedElement	composition	1.. *	This composition association models the fact that every <i>AnnotationModel</i> element must have at least one <i>AnnotatedElement</i> .
Annotation	composition	1.. *	This composition association models the fact that every <i>AnnotationModel</i> element must have at least one <i>Annotation</i> .

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of an *AnnotationModel* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the *External Service Composition* PIM extension meta-model that can be found in appendix A.1.

- *atLeastOneAnnAlgoResourceControllerExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnAlgoResourceController*.
- *atLeastOneAnnCRUDActivityExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that are of type *AnnCRUDActivity*.
- *atLeastOneAnnCRUDActivityHandlerExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnCRUDActivityHandler*.
- *atLeastOneAnnAlgoResourceModelExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* element that is associated with this *AnnotationModel* that is of type *AnnAlgoResourceModel*.
- *atLeastOneRESTClientControllerExists*: This OCL constraint checks whether there exists at least one *Annotation* element that is associated with this *AnnotationModel* that is of type *RESTClientController*.
- *atLeastOneRESTClientModelExists*: This OCL constraint checks whether there exists at least one *Annotation* element that is associated with this *AnnotationModel* that is of type *RESTClientModel*.

5.4.1.2.2 AnnotatedElement Element

Overview

The *AnnotatedElement* element models any existent Core PIM meta-model element that can be annotated by the *External Service Composition* Core PIM extension in order to extend its semantic meaning and/or design. It is always further specialized to more specific categories of annotated elements.

Properties

The *AnnotatedElement* element of the *External Service Composition PIM* extension meta-model does not have any property.

Relations

The *AnnotatedElement* element of the *External Service Composition PIM* extension meta-model does not have any relation.

Behavioural Restrictions

The *AnnotatedElement* of the *External Service Composition PIM* extension meta-model does not have any behavioural restriction.

5.4.1.2.3 Annotation Element

Overview

The *Annotation* element models any new semantic meaning or design that is attached to a Core PIM meta-model element. It is always further specialized to more specific categories of annotations.

Properties

The *Annotation* element of the *External Service Composition PIM* extension meta-model does not have any property.

Relations

The *Annotation* element of the *External Service Composition PIM* extension meta-model does not have any relation.

Behavioural Restrictions

The *Annotation* element of the *External Service Composition PIM* extension meta-model does not have any behavioural restriction.

5.4.1.2.4 AnnAlgoResourceModel Element

Overview

The *AnnAlgoResourceModel* element models an existent *AlgoResourceModel* of the Core PIM model that can be annotated by the *External Service Composition PIM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnAlgoResourceModel* element of the *External Service Composition PIM* extension meta-model does not have any property.

Relations

Table 5-25 AnnAlgoResourceModel's Relations

Relation With PIM Element	Type	Multiplicity	Structural Constraints
AlgoResourceModel	reference	1	The <i>AnnAlgoResourceModel</i> must reference exactly one <i>AlgoResourceModel</i> of the Core PIM model.

Behavioural Restrictions

The *AnnAlgoResourceModel* element of the *External Service Composition PIM* extension meta-model does not have any behavioural restriction.

5.4.1.2.5 AnnResourceModel Element

Overview

The *AnnResourceModel* element models an existent *ResourceModel* of the Core PIM model that can be annotated by the *External Service Composition PIM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnResourceModel* element of the *External Service Composition PIM* extension meta-model does not have any property.

Relations

Table 5-26 AnnResourceModel's Relations

Relation With PIM Element	Type	Multiplicity	Structural Constraints
ResourceModel	reference	1	The <i>AnnResourceModel</i> must reference exactly one <i>ResourceModel</i> of the Core PIM model.

Behavioural Restrictions

The *AnnResourceModel* element of the *External Service Composition PIM* extension meta-model does not have any behavioural restriction.

5.4.1.2.6 AnnCRUDActivityHandler Element

Overview

The *AnnCRUDActivityHandler* element models an existent *CRUDActivityHandler* of the Core PIM model that can be annotated by the *External Service Composition PIM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnCRUDActivityHandler* element of the *External Service Composition PIM* extension meta-model does not have any property.

Relations

Table 5-27 AnnCRUDActivityHandler's Relations

Relation With PIM Element	Type	Multiplicity	Structural Constraints
AnnCRUDActivityHandler	reference	1	The <i>AnnCRUDActivityHandler</i> must reference exactly one <i>CRUDActivityHandler</i> of the Core PIM model.

Behavioural Restrictions

The *AnnCRUDActivityHandler* element of the *External Service Composition PIM* extension meta-model does not have any behavioural restriction.

5.4.1.2.7 AnnCRUDActivity Element

Overview

The *AnnCRUDActivity* element models an existent *CRUDActivity* of the Core PIM model that can be annotated by the *External Service Composition PIM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnCRUDActivity* element of the *External Service Composition PIM* extension meta-model does not have any property.

Relations

Table 5-28 AnnCRUDActivity's Relations

Relation With PIM Element	Type	Multiplicity	Structural Constraints
CRUDActivity	reference	1	The <i>AnnCRUDActivity</i> must reference exactly one <i>CRUDActivity</i> of the Core PIM model.

Behavioural Restrictions

The *AnnCRUDActivity* element of the *External Service Composition PIM* extension meta-model does not have any behavioural restriction.

5.4.1.2.8 AnnAlgoResourceController Element

Overview

The *AnnAlgoResourceController* element models an existent *AlgoResourceController* of the Core PIM model that can be annotated by the *External Service Composition PIM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnAlgoResourceController* element of the *External Service Composition PIM* extension meta-model does not have any property.

Relations

Table 5-29 AnnAlgoResourceController's Relations

Relation With PIM Element	Type	Multiplicity	Structural Constraints
AlgoResourceController	reference	1	The <i>AnnAlgoResourceController</i> must reference exactly one <i>AlgoResourceController</i> of the Core PIM model.

Behavioural Restrictions

The *AnnAlgoResourceController* element of the *External Service Composition PIM* extension meta-model does not have any behavioural restriction.

5.4.1.2.9 RESTClientController Element

Overview

The *RESTClientController* element models an annotation of the *External Service Composition PIM* extension meta-model that is intended to annotate an existent Core PIM *AlgoResourceController*. With this annotation, the *AlgoResourceController* will embed the needed infrastructure to accept incoming requests, the handling of which will be delegated to a composition of external services. Figure 5-11 demonstrates the *RESTClientController* element of the *External Service Composition PIM* extension meta-model and its relations.

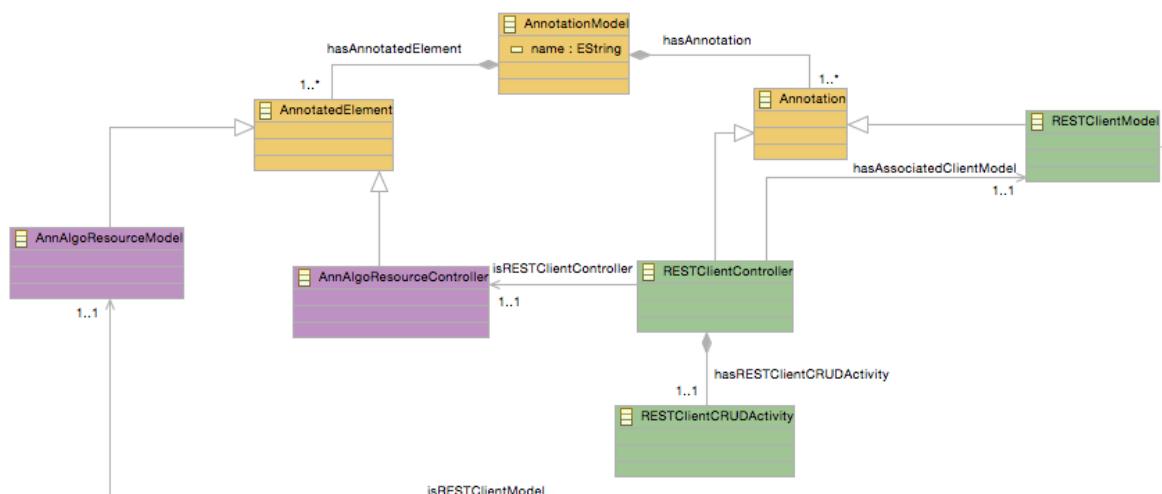


Figure 5-11 RESTClientController annotation and its relations.

Properties

The *RESTClientController* element of the *External Service Composition PIM* extension meta-model does not have any property.

Relations

Table 5-30 RESTClientController's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
AnnAlgoResourceController	association	1	The <i>RESTClientController</i> must be associated with exactly one <i>AnnAlgoResourceController</i> , which references one <i>AlgoResourceController</i> of the Core PIM model. With this annotation, the <i>AlgoResourceController</i> will embed the needed infrastructure to accept incoming requests that will be handled by a service composition of external services.
RESTClientModel	association	1	The <i>RESTClientController</i> must have an association with exactly one <i>RESTClientModel</i> . This association models the fact that every <i>RESTClientController</i> will have to handle specifically formatted input/output data. This format, is encapsulated in the composited <i>RESTClientModel</i> .
RESTClientCRUDActivity	composition	1	The <i>RESTClientController</i> must have a composition association with exactly one <i>RESTClientCRUDActivity</i> . This association models the fact that when an <i>AlgoResourceController</i> is annotated as <i>RESTClientController</i> , it must have its <i>CRUDActivity</i> annotated as <i>RESTClientCRUDActivity</i> as well.

Behavioural Restrictions

The *RESTClientController* element of the *External Service Composition PIM* extension meta-model does not have any behavioural restriction.

5.4.1.2.10 RESTClientCRUDActivity Element

Overview

The *RESTClientCRUDActivity* element models an annotation of the *External Service Composition PIM* extension meta-model that is intended to annotate an existent Core PIM *CRUDActivity*. With this annotation, the *CRUDActivity* will implement the needed infrastructure to handle incoming client's requests, which will be forwarded to a composition of external services. Figure 5-12 demonstrates

the *RESTClientCRUDActivity* element of the *External Service Composition PIM* extension meta-model and its relations.

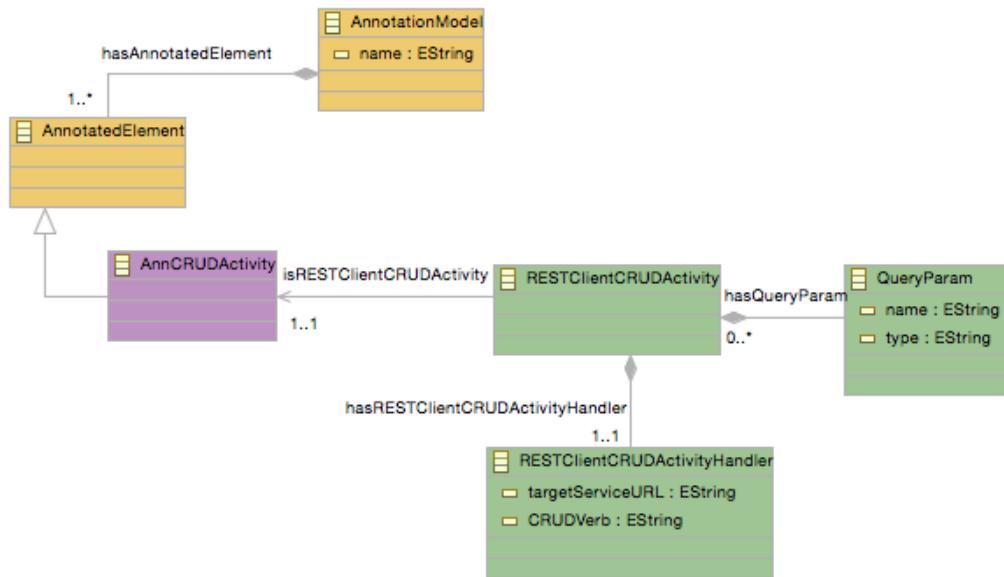


Figure 5-12 RESTClientCRUDActivity element and its annotations.

Properties

The *RESTClientCRUDActivity* element of the *External Service Composition PIM* extension meta-model does not have any property.

Relations

Table 5-31 RESTClientCRUDActivity's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
AnnCRUDActivity	association	1	The <i>RESTClientCRUDActivity</i> must be associated with exactly one <i>AnnCRUDActivity</i> , which references one <i>CRUDActivity</i> of the Core PIM model. With this annotation, the <i>CRUDActivity</i> will implement the needed infrastructure to forward incoming client's requests to a composition of external services.
QueryParam	composition	0..*	The <i>RESTClientCRUDActivity</i> can have a composition association with zero or more <i>QueryParams</i> . This association models the fact that the client's request to be forwarded to an external service composition for handling may require zero or more query parameters.
RESTClientCRUDActivityHandler	composition	1	The <i>RESTClientCRUDActivity</i> must have a composition association with exactly one <i>RESTClientCRUDActivityHandler</i> .

Behavioural Restrictions

The *RESTClientCRUDActivity* element of the *External Service Composition PIM* extension meta-model does not have any behavioural restriction.

5.4.1.2.11 RESTClientCRUDActivityHandler Element

Overview

The *RESTClientCRUDActivityHandler* element models an annotation of the *External Service Composition PIM* extension meta-model that is intended to annotate an existent Core PIM *CRUDActivityHandler*. With this annotation, the *CRUDActivityHandler* will reformat the incoming client's requests and then delegate them to a composition of external services. Figure 5-13 demonstrates the *RESTClientCRUDActivityHandler* element of the *External Service Composition PIM* extension meta-model and its relations.

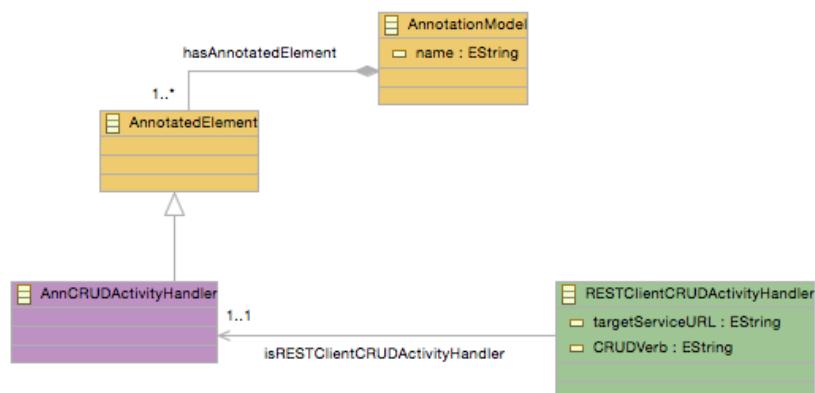


Figure 5-13 RESTClientCRUDActivityHandler

Properties

Table 5-32 RESTClientCRUDActivityHandler's Properties

Name	Type	Multiplicity	Explanation
targetServiceURL	EString	1	This is the <i>URL</i> of the external service composition to which the handling of the client's request will be delegated.
CRUDVerb	EString	1	This is the <i>CRUDVerb</i> that must be used by the envisioned system in order to interact with the composition of external services.

Relations

Table 5-33 RESTClientCRUDActivityHandler's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
AnnCRUDActivityHandler	association	1	The <i>RESTClientCRUDActivityHandler</i> must be associated with exactly one <i>AnnCRUDActivityHandler</i> , which references a <i>CRUDActivityHandler</i> of the Core PIM model. With this annotation, the <i>CRUDActivityHandler</i> will embed the needed functionality to reformat and forward incoming client's requests to an external services composition.

Behavioural Restrictions

The *RESTClientCRUDActivityHandler* element of the *External Service Composition PIM* extension meta-model does not have any behavioural restriction.

5.4.1.2.12 *QueryParam Element*

Overview

The *QueryParam* element models a query parameter that can be used when interacting with the external service composition.

Properties

Table 5-34 QueryParam's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the name of the <i>QueryParam</i> .
type	EString	1	This is the type of the <i>QueryParam</i> .

Relations

The *QueryParam* element of the *External Service Composition PIM* extension meta-model does not have any relation.

Behavioural Restrictions

The *QueryParam* element of the *External Service Composition PIM* extension meta-model does not have any behavioural restriction.

5.4.1.2.13 RESTClientModel Element

Overview

The *RESTClientModel* element models an annotation of the *External Service Composition* PIM extension meta-model that is intended to annotate an existent Core PIM *AlgoResourceModel*. With this annotation, the *AlgoResourceModel* will be expanded to embed the input data model of the external service composition as well. Figure 5-14 demonstrates the *RESTClientModel* element of the *External Service Composition* PIM extension meta-model and its relations.

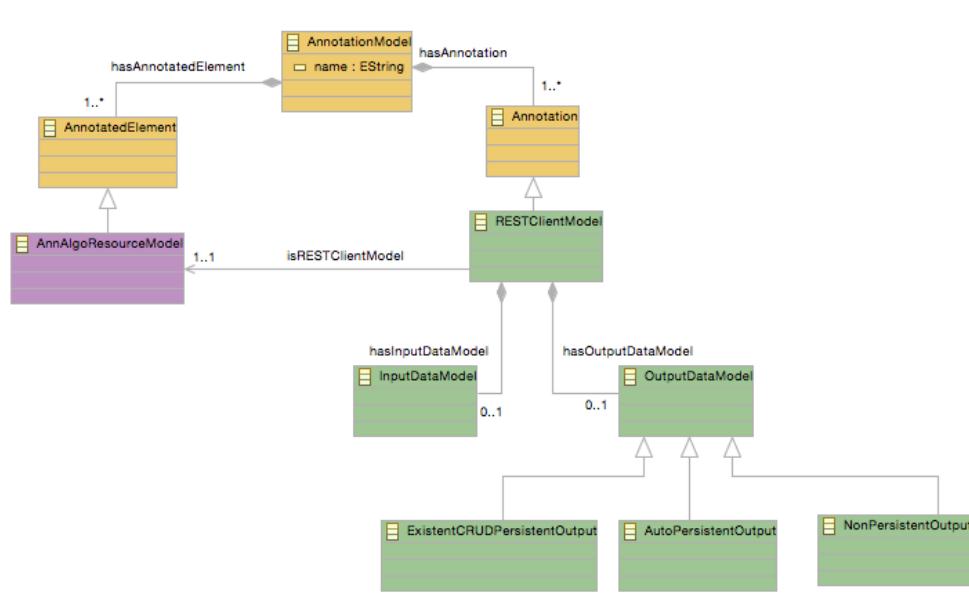


Figure 5-14 RESTClientModel annotation and its relations.

Properties

The *RESTClientModel* element of the *External Service Composition PIM* extension meta-model does not have any property.

Relations

Table 5-35 RESTClientModel's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
InputDataModel	composition	0..1	The <i>RESTClientModel</i> can have zero or one composition association with an <i>InputDataModel</i> . This association models the fact that the <i>RESTClientModel</i> may have or not to package some data using some media format and use it as input to the external service composition. This is the case when the external service composition requires some input data.
OutputDataModel	composition	0..1	The <i>RESTClientModel</i> can have zero or one composition association with an <i>OutputDataModel</i> . This association models the fact that the <i>RESTClientModel</i> may expect or not

			some data in some media format as a response from the external service composition. This is the case when the external service composition returns some output data as a response to a client's request.
AnnAlgoResourceModel	association	1	The <i>RESTClientModel</i> must be associated with exactly one <i>AnnAlgoResourceModel</i> , which references an <i>AlgoResourceModel</i> . With this annotation, the <i>AlgoResourceModel</i> will be expanded as needed, to accommodate any needed input data modelling.

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of a *RESTClientModel* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the external service composition PIM extension meta-model that can be found in appendix A.1.

- *hasOnlySpecializedOutputDataModels*: This OCL constraint checks whether any *OutputDataModel* that is associated with this *RESTClientModel* is always further specialized to either *ExistentsCRUDPersistentOutput*, *AutoPersistentOutput* or *NonPersistentOutput*.
- *onlyExistentsCRUDPersistentOutputHasZeroPropertiesAdded*: This OCL constraint checks whether an *OutputDataModel*, which is associated with this *RESTClientModel*, that is of type *ExistentsCRUDPersistentOutput* has exactly zero associated *Properties* and whether any other type of *OutputDataModel* has at least one associated *Property*.

5.4.1.2.14 InputDataModel Element

Overview

The *InputDataModel* element models the input data needed by the external service composition. Figure 5-15 demonstrates the *InputDataModel* element of the *External Service Composition* PIM extension meta-model and its relations.

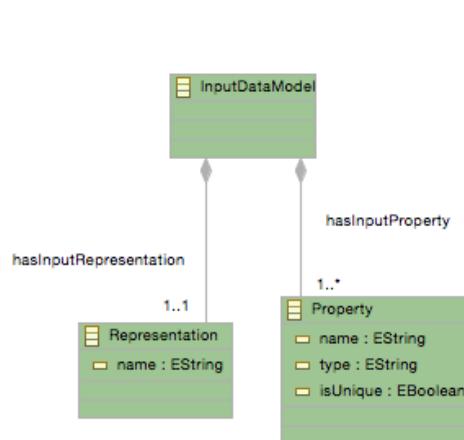


Figure 5-15 InputDataModel element and its relations.

Properties

The *InputdataModel* element of the *External Service Composition PIM* extension meta-model does not have any property.

Relations

Table 5-36 InputDataModel's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
Property	composition	1..*	The <i>InputDataModel</i> must have a composition association with at least one <i>Property</i> element. This association models the fact that should an <i>InputDataModel</i> of an external service composition exist, it should have at least one property in order to be meaningful.
Representation	composition	1	The <i>InputDataModel</i> must have a composition association with exactly one <i>Representation</i> element. This association models the fact that an <i>InputDataModel</i> of an external service composition must be formatted using a specific media format, such as application/JSON etc.

Behavioural Restrictions

The *InputDataModel* element of the *External Service Composition PIM* extension meta-model does not have any behavioural restriction.

5.4.1.2.15 OutputDataModel Element

Overview

The *OutputDataModel* element models the output data model of the external service composition. Figure 5-16 demonstrates the *OutputDataModel* element of the *External Service Composition PIM* extension meta-model and its relations.

Properties

The *OutputDataModel* element of the *External Service Composition PIM* extension meta-model does not have any property.

Relations

Table 5-37 OutputDataModel's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
Property	composition	0..*	The <i>OutputDataModel</i> must have a composition association with zero or more <i>Property</i> elements. This association models the fact that should an <i>OutputDataModel</i> of an external service composition exist, it may have some

Properties in order to be meaningful.			
Representation	composition	1	The <i>OutputDataModel</i> must have a composition association with exactly one <i>Representation</i> element. This association models the fact that an <i>OutputDataModel</i> of an external service composition must be formatted using a specific media format, such as application/JSON etc.

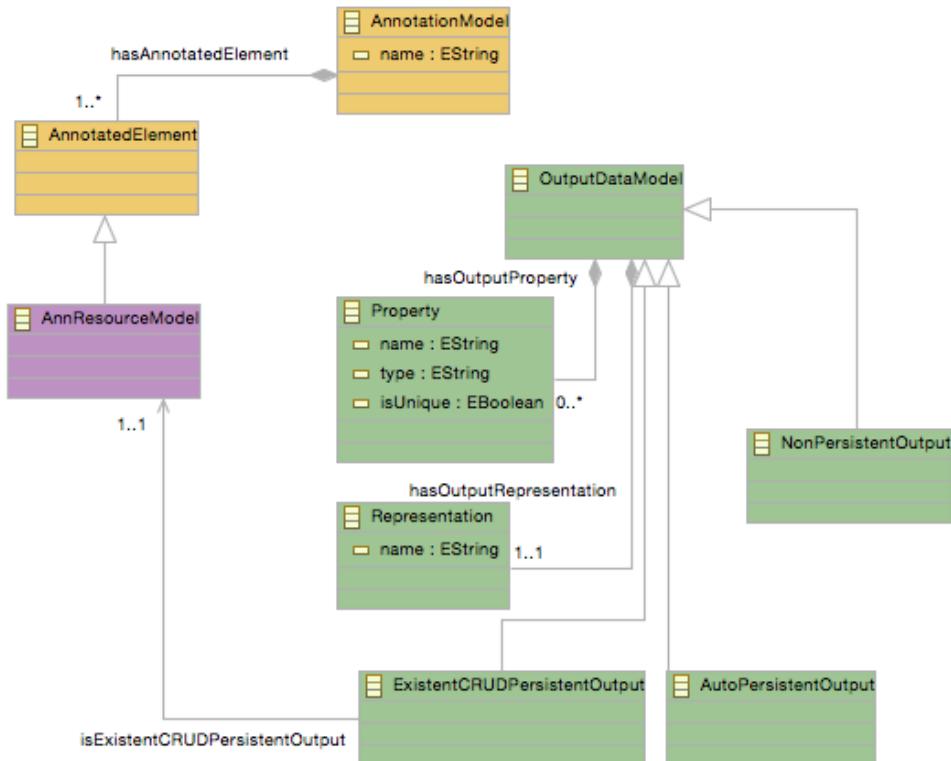


Figure 5-16 OutputDataModel element and its annotations.

Behavioural Restrictions

The *OutputDataModel* element of the *External Service Composition PIM* extension meta-model does not have any behavioural restriction.

5.4.1.2.16 Property Element

Overview

The *Property* element models a property of either an *InputDataModel* or an *OutputDataModel* of an external service composition.

Properties

Table 5-38 Property's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the name of the <i>Property</i> .

type	EString	1	This is the type of the <i>Property</i> .
isUnique	EBoolean	1	The <i>isUnique</i> property of the <i>Property</i> models the multiplicity of the <i>Property</i> . If it is set to true then it is of multiplicity one. Otherwise it is a multivalued array.

Relations

The *Property* element of the *External Service Composition PIM* extension meta-model does not have any relation.

Behavioural Restrictions

The *Property* element of the *External Service Composition PIM* extension meta-model does not have any behavioural restriction.

5.4.1.2.17 Representation Element

Overview

The *Representation* element models a representation media format with which either an *InputDataModel* or an *OutputDataModel* should be formatted when interacting with the external service composition.

Properties

Table 5-39 Representation's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the media type of the <i>Representation</i> e.g. application/JSON etc.

Relations

The *Representation* element of the *External Service Composition PIM* extension meta-model does not have any relation.

Behavioural Restrictions

The *Representation* element of the *External Service Composition PIM* extension meta-model does not have any behavioural restriction.

5.4.1.2.18 *NonPersistentOutput Element*

Overview

The *NonPersistentOutput* element models a specialization of the *OutputDataModel*, which embeds the meaning that the output data received from the external service composition should only be forwarded back to the envisioned system's client and not persisted in the local database as well.

Properties

The *NonPersistentOutput* element of the *External Service Composition PIM* extension meta-model does not have any property other than those inherited from the *OutputDataModel*.

Relations

The *NonPersistentOutput* element of the *External Service Composition PIM* extension meta-model does not have any relation other than those inherited from the *OutputDataModel*.

Behavioural Restrictions

The *NonPersistentOutput* element of the *External Service Composition PIM* extension meta-model does not have any behavioural restriction.

5.4.1.2.19 *AutoPersistentOutput Element*

Overview

The *AutoPersistentOutput* element models a specialization of the *OutputDataModel*, which embeds the meaning that the output data received from the external service composition should be forwarded back to the envisioned system's client and be stored in the local database using a data model that is automatically generated and is unrelated with the rest of the database relations.

Properties

The *AutoPersistentOutput* element of the *External Service Composition PIM* extension meta-model does not have any property other than those inherited from the *OutputDataModel*.

Relations

The *AutoPersistentOutput* element of the *External Service Composition PIM* extension meta-model does not have any relation other than those inherited from the *OutputDataModel*.

Behavioural Restrictions

The *AutoPersistentOutput* element of the *External Service Composition PIM* extension meta-model does not have any behavioural restrictions.

5.4.1.2.20 *ExistentCRUDPersistentOutput Element*

Overview

The *ExistentCRUDPersistentOutput* element models an *OutputDataModel* specialization, which embeds the meaning that the output data received from the external service composition should be forwarded back to the envisioned system's client and be stored in the local database using an existing *ResourceModel* of the Core PIM.

Properties

The *ExistentCRUDPersistentOutput* element of the *External Service Composition PIM* extension meta-model does not have any property other than those inherited from the *OutputDataModel*.

Relations

Table 5-40 ExistentCRUDPersistentOutput's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
AnnResourceModel	association	1	The <i>ExistentCRUDPersistentOutput</i> must be associated with exactly one <i>AnnResourceModel</i> , which references one <i>ResourceModel</i> of the Core PIM. With this annotation the <i>ResourceModel</i> , will be used as the persistence model of the output data received from the external service composition.

The *ExistentCRUDPersistentOutput* inherits as well, all the relations that its overlying *OutputDataModel* has.

Behavioural Restrictions

The *ExistentCRUDPersistentOutput* element of the *External Service Composition PIM* extension meta-model does not have any behavioural restriction.

5.5 PIM ABAC Authorization Ecore Meta-model Definition

5.5.1.1 Introduction

In order to fully define the *ABAC Authorization* Ecore PIM extension meta-model, its various aspects must be explained, demonstrated as well as documented. This is done in the following subsections following the format section 4.1.2.1 describes. Due to space limitations, the full *ABAC Authorization* meta-model can be found using the following link <https://github.com/s-case/mde> .

5.5.1.2 PIM ABAC Authorization Ecore Meta-model Elements

5.5.1.2.1 *AnnotationModel* Element

Overview

The *AnnotationModel* element is the root element of the *ABAC Authorization* PIM extension meta-model and comprises of all the introduced *Annotations* and the existing Core PIM elements to which these annotations are attached in order to alter their semantic meaning, design and/or configuration.

Properties

Table 5-41 *AnnotationModel*'s Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the name of the <i>Annotation</i> model.

Relations

Table 5-42 *AnnotationModel*'s Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
AnnotatedElement	composition	1.. *	This composition association models the fact that every <i>AnnotationModel</i> element must have at least one <i>AnnotatedElement</i> .
Annotation	composition	1.. *	This composition association models the fact that every <i>AnnotationModel</i> element must have at least one <i>Annotation</i> .

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of an *AnnotationModel* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the *ABAC Authorization* PIM extension meta-model that can be found in appendix A.1.

- *atLeastOneAnnResourceModelExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnResourceModel*.
- *atLeastOneAnnCRUDActivityHandlerExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnCRUDActivityHandler*.
- *exactlyOneAnnDatabaseControllerExists*: This OCL constraint checks whether there exists exactly one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnDatabaseController*.
- *atLeastOneAnnResourceControllerCRUDActivityExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnResourceControllerCRUDActivity*.
- *exactlyOneAuthorizationSubjectExists*: This OCL constraint checks whether there exists exactly one *Annotation* element that is associated with this *AnnotationModel* that is of type *AuthorizationSubject*.
- *atLeastOneAuthorizableResourceExists*: This OCL constraint checks whether there exist exactly two *Annotation* elements that are associated with this *AnnotationModel* that are of type *AuthorizableResource*.
- *exactlyOneAuthorizationDataHandlerExists*: This OCL constraint checks whether there exists exactly one *Annotation* element that is associated with this *AnnotationModel* that is of type *AuthorizationDataHandler*.

5.5.1.2.2 *AnnotatedElement* Element

Overview

The *AnnotatedElement* element models any existent Core PIM meta-model element that can be annotated by the *ABAC Authorization* Core PIM extension in order to extend its semantic meaning and/or design. It is always further specialized to more specific categories of annotated elements.

Properties

The *AnnotatedElement* element of the *ABAC Authorization PIM* extension meta-model does not have any property.

Relations

The *AnnotatedElement* element of the *ABAC Authorization PIM* extension meta-model does not have any relation.

Behavioural Restrictions

The *AnnotatedElement* of the *ABAC Authorization PIM* extension meta-model does not have any behavioural restriction.

5.5.1.2.3 *Annotation* Element

Overview

The *Annotation* element models any new semantic meaning or design that is attached to a Core PIM meta-model element. It is always further specialized to more specific categories of annotations.

Properties

The *Annotation* element of the *ABAC Authorization PIM* extension meta-model does not have any property.

Relations

The *Annotation* element of the *ABAC Authorization PIM* extension meta-model does not have any relation.

Behavioural Restrictions

The *Annotation* element of the *ABAC Authorization PIM* extension meta-model does not have any behavioural restriction.

5.5.1.2.4 *AnnResourceControllerCRUDActivity* Element

Overview

The *AnnResourceControllerCRUDActivity* element models an existent *ResourceControllerCRUDActivity* of the Core PIM model that can be annotated by the *ABAC Authorization PIM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnResourceControllerCRUDActivity* element of the *ABAC Authorization PIM* extension meta-model does not have any property.

Relations

Table 5-43 AnnResourceControllerCRUDActivity's Relations

Relation With PIM Element	Type	Multiplicity	Structural Constraints
ResourceControllerCRUDActivity	reference	1	The <i>AnnResourceControllerCRUDActivity</i> must reference exactly one <i>ResourceControllerCRUDActivity</i> of the Core PIM model.

Behavioural Restrictions

The *AnnResourceControllerCRUDActivity* element of the *ABAC Authorization PIM* extension meta-model does not have any behavioural restriction.

5.5.1.2.5 *AnnPIMComponentProperty* Element

Overview

The *AnnPIMComponentProperty* element models an existent *PIMComponentProperty* of the Core PIM model that can be annotated by the *ABAC Authorization PIM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnPIMComponentProperty* element of the *ABAC Authorization PIM* extension meta-model does not have any property

Relations

Table 5-44 AnnPIMComponentProperty's Relations

Relation With PIM Element	Type	Multiplicity	Structural Constraints
PIMComponentProperty	reference	1	The <i>AnnPIMComponentProperty</i> must reference exactly one <i>PIMComponentProperty</i> of the Core PIM model.

Behavioural Restrictions

The *AnnPIMComponentProperty* element of the *ABAC Authorization PIM* extension meta-model does not have any behavioural restriction.

5.5.1.2.6 AnnDatabaseController Element

Overview

The *AnnDatabaseController* element models an existent *DatabaseController* of the Core PIM model that can be annotated by the *ABAC Authorization PIM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnDatabaseController* element of the *ABAC Authorization PIM* extension meta-model does not have any property

Relations

Table 5-45 AnnDatabaseController's Relations

Relation With PIM Element	Type	Multiplicity	Structural Constraints
DatabaseController	reference	1	The <i>AnnDatabaseController</i> must reference exactly one <i>DatabaseController</i> element of the Core PIM model.

Behavioural Restrictions

The *AnnDatabaseController* element of the *ABAC Authorization PIM* extension meta-model does not have any behavioural restriction.

5.5.1.2.7 AnnCRUDActivityHandler Element

Overview

The *AnnCRUDActivityHandler* element models an existent *CRUDActivityHandler* of the Core PIM model that can be annotated by the *ABAC Authorization* PIM extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnCRUDActivityHandler* element of the *ABAC Authorization PIM* extension meta-model does not have any property.

Relations

Table 5-46 AnnCRUDActivityHandler's Relations

Relation With PIM Element	Type	Multiplicity	Structural Constraints
CRUDActivityHandler	reference	1	The <i>AnnCRUDActivityHandler</i> must reference exactly one <i>CRUDActivityHandler</i> element of the Core PIM model.

Behavioural Restrictions

The *AnnCRUDActivityHandler* element of the *ABAC Authorization PIM* extension meta-model does not have any behavioural restriction.

5.5.1.2.8 AnnResourceModelManager Element

Overview

The *AnnResourceModelManager* element models an existent *ResourceModelManager* of the Core PIM model that can be annotated by the *ABAC Authorization* PIM extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnResourceModelManager* element of the *ABAC Authorization PIM* extension meta-model does not have any property.

Relations

Table 5-47 AnnResourceModelManager's Relations

Relation With PIM Element	Type	Multiplicity	Structural Constraints
ResourceModelManager	reference	1	The <i>AnnResourceModelManager</i> must reference exactly one <i>ResourceModelManager</i> element of the Core PIM model.

Behavioural Restrictions

The *AnnResourceModelManager* element of the *ABAC Authorization PIM* extension meta-model does not have any behavioural restriction.

5.5.1.2.9 AnnAlgoResourceModel Element

Overview

The *AnnAlgoResourceModel* element models an existent *AlgoResourceModel* of the Core PIM model that can be annotated by the *ABAC Authorization PIM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnAlgoResourceModel* element of the *ABAC Authorization PIM* extension meta-model does not have any property

Relations

Table 5-48 AnnAlgoResourceModel's Relations

Relation With PIM Element	Type	Multiplicity	Structural Constraints
AlgoResourceModel	reference	1	The <i>AnnAlgoResourceModel</i> must reference exactly one <i>AlgoResourceModel</i> element of the Core PIM model.

Behavioural Restrictions

The *AnnAlgoResourceModel* element of the *ABAC Authorization PIM* extension meta-model does not have any behavioural restriction.

5.5.1.2.10 AnnResourceModel Element

Overview

The *AnnResourceModel* element models an existent *ResourceModel* of the Core PIM model that can be annotated by the *ABAC Authorization PIM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnResourceModel* element of the *ABAC Authorization PIM* extension meta-model does not have any property.

Relations

Table 5-49 AnnResourceModel's Relations

Relation With PIM Element	Type	Multiplicity	Structural Constraints
ResourceModel	reference	1	The <i>AnnResourceModel</i> must reference exactly one <i>ResourceModel</i> element of the Core PIM model.

Behavioural Restrictions

The *AnnResourceModel* element of the *ABAC Authorization PIM* extension meta-model does not have any behavioural restriction.

5.5.1.2.11 AuthorizationSubject Element

Overview

The *AuthorizationSubject* element models an annotation of the *ABAC Authorization PIM* extension meta-model that is intended to annotate an existent Core PIM *ResourceModel*. With this annotation, the *ResourceModel* will be used as the subject model of any access request on behalf of a client. That is, its properties will be used as attributes of the request issuer to be consumed by the resource access policy evaluator, in order to assess whether the he should be granted access or not. Figure 5-17 demonstrates the *AuthorizationSubject* element of the *ABAC Authorization PIM* extension meta-model and its relations.

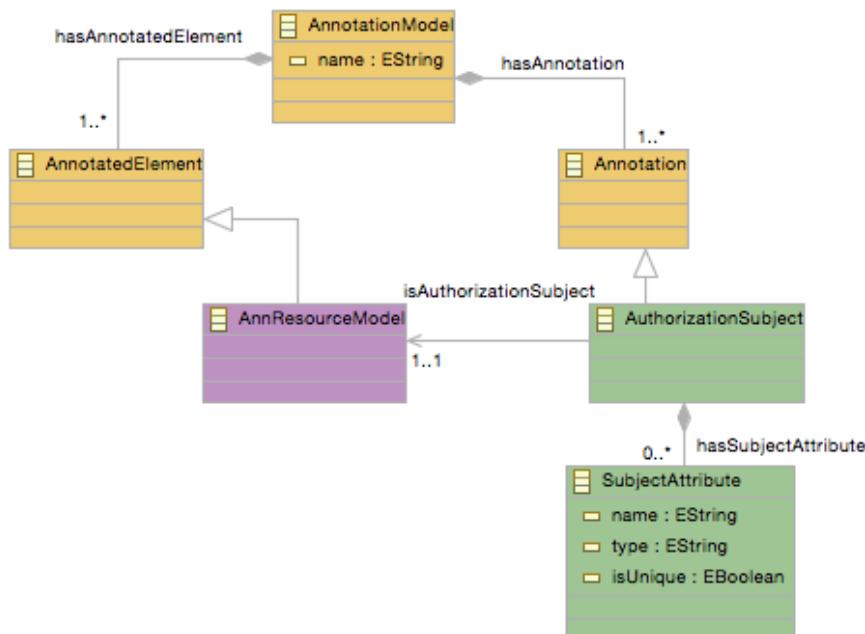


Figure 5-17 AuthorizationSubject annotation and its relations

Properties

The *AuthorizationSubject* element of the *ABAC Authorization PIM* extension meta-model does not have any property.

Relations

Table 5-50 AuthorizationSubject's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
SubjectAttribute	composition	0..*	The <i>AuthorizationSubject</i> can have a composition association with zero or more <i>SubjectAttributes</i> . This association models the fact that an <i>AuthorizationSubject</i> may have extra properties, other than those defined in the referenced Core PIM <i>ResourceModel</i> . This is needed, since the S-CASE developer designs the referenced <i>ResourceModel</i> 's properties having in mind the envisioned system's functionality rather than authorization issues.
AnnResourceModel	association	1	The <i>AuthorizationSubject</i> must have exactly one association with an <i>AnnResourceModel</i> , which references a Core PIM <i>ResourceModel</i> . With this annotation, the <i>ResourceModel</i> will be used as a model of the access request issuer. That is, its properties (such as username, groups etc.) will be used in the access evaluation process in order to conclude whether access should be granted to the issuer or not.

Behavioural Restrictions

The *AnnResourceModel* element of the *ABAC Authorization PIM* extension meta-model does not have any behavioural restriction.

5.5.1.2.12 *SubjectAttribute* Element

Overview

The *SubjectAttribute* element models an extra property that can be added to the referenced *AuthorizationModel*'s properties. This capability exists, since the S-CASE developer designs the Core PIM *ResourceModel*'s properties having in mind the envisioned system's functionality rather than authentication issues.

Properties

Table 5-51 SubjectAttribute's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the name of the <i>SubjectAttribute</i> .
type	EString	1	This is the type of the <i>SubjectAttribute</i> .

isUnique	EBoolean	1	This is the multiplicity of the <i>SubjectAttribute</i> . If the <i>EBoolean isUnique</i> is set to true, then the <i>SubjectAttribute</i> has multiplicity one. Otherwise, it is a multivalued array.
----------	----------	---	--

Relations

The *SubjectAttribute* element of the *ABAC Authorization PIM* extension meta-model does not have any relation.

Behavioural Restrictions

The *SubjectAttribute* element of the *ABAC Authorization PIM* extension meta-model does not have any behavioural restriction.

5.5.1.2.13 AuthorizableResource Element

Overview

The *AuthorizableResource* element models an annotation of the *ABAC Authorization PIM* extension meta-model that is intended to annotate an existent Core PIM *ResourceModel*, *ResourceModelManager* or *AlgoResourceModel*. With this annotation, access to either of them will be restricted using an *ABAC Authorization* resource access policy. Therefore, access to the allowed actions of these resources will only be granted if and only if the underlying authorization rules evaluation yields access permission. Figure 5-18 demonstrates the *AuthorizableResource* element of the *ABAC Authorization PIM* extension meta-model and its relations.

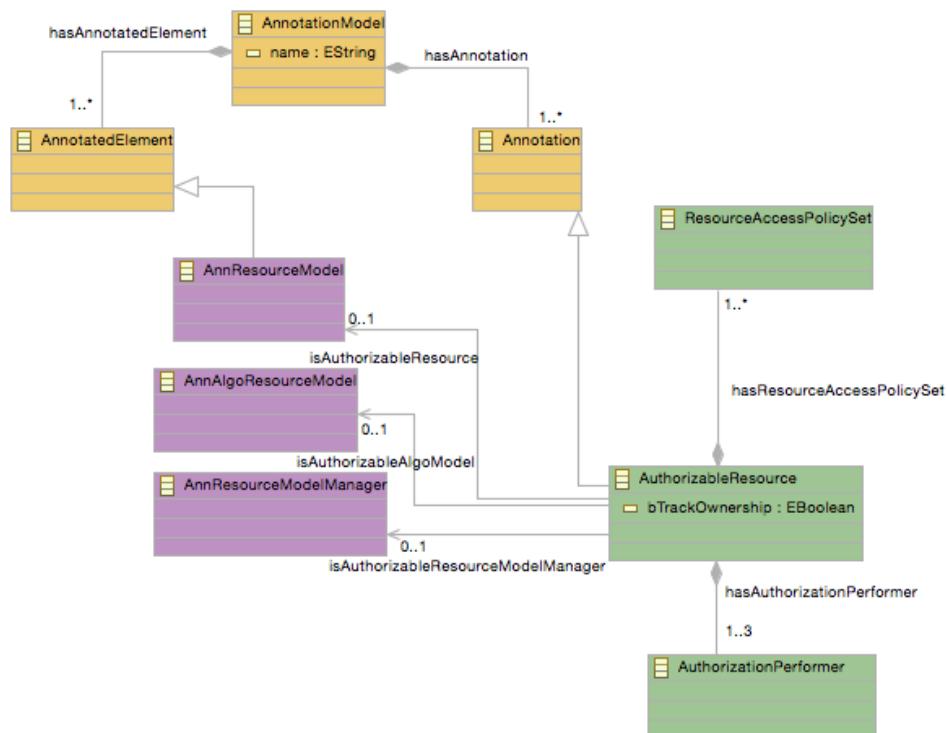


Figure 5-18 AuthorizableResource annotation and its relations

Properties

Table 5-52 AuthorizableResource's Properties

Name	Type	Multiplicity	Explanation
bTrackOwnership	EBoolean	1	This <i>EBoolean</i> models whether the ownership of an <i>AuthorizableResource</i> is tracked or not. If it is set to <i>true</i> then the ownership will be tracked. Otherwise it will not. This allows envisioned systems produced by S-CASE to be able to also implement several DAC authorization schemes (e.g. like Unix/Linux).

Relations

Table 5-53 AuthorizableResource's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
ResourceAccessPolicySet	composition	1..*	The <i>AuthorizableResource</i> must have a composition association with at least one <i>ResourceAccessPolicySet</i> . This association models the fact that every <i>AuthorizableResource</i> can have several authorization policy sets that must be evaluated every time a client makes an access request to it.
AuthorizationPerformer	composition	1..3	The <i>AuthorizableResource</i> must have at least one and at most three composition associations with <i>AuthorizationPerformers</i> . This association models the fact that every <i>AuthorizableResource</i> delegates the whole authorization process to distinct associated software artefacts that are annotated as <i>AuthorizationPerformer</i> .
AnnResourceModel	association	0..1	The <i>AuthorizableResource</i> can have zero or one association with an <i>AnnResourceModel</i> element, which references one <i>ResourceModel</i> of the Core PIM model. With this annotation, every access request an envisioned system's client makes to access functionality that this <i>ResourceModel</i> embeds will be firstly evaluated in regard to the resource access policy set. If and only if, this evaluation grants the client access permission, will it be allowed to interact with the resource.
AnnResourceModelManager	association	0..1	The <i>AuthorizableResource</i> can have zero or one association with an <i>AnnResourceModelManager</i> , which references one <i>ResourceModelManager</i> element of the Core PIM model. With this annotation, every access request will be evaluated in the way already described for the <i>ResourceModels' case</i>

AnnAlgoResourceModel	association	0..1	The <i>AuthorizableResource</i> can have zero or one association with an <i>AnnAlgoResourceModel</i> , which references one <i>AlgoResourceModel</i> element of the Core PIM model. with this annotation, every access requests will be evaluated in the way already described for the <i>ResourceModels</i> ' case above.
----------------------	-------------	------	--

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of a *AuthorizableResource* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the *ABAC Authorization* PIM extension meta-model that can be found in appendix A.1.

- annotesExactlyOnePIMArtifact: This OCL constraint checks whether an *AuthorizableResource* annotates exactly one *ResourceModel* and the *ResourceModelManager* of which it is related and has zero associated *AnnAlgoResourceModels*, or annotates exactly one *AlgoResourceModel* but has zero associated *ResourceModels* and zero associated *ResourceModelManagers*.

5.5.1.2.14 ResourceAccessPolicySet Element

Overview

The *ResourceAccessPolicySet* element models an ABAC authorization resource access policy set that comprises all the access policies a specific *AuthorizableResource* has. Figure 5-19 demonstrates the *ResourceAccessPolicySet* element of the *ABAC Authorization* PIM extension meta-model and its relations.

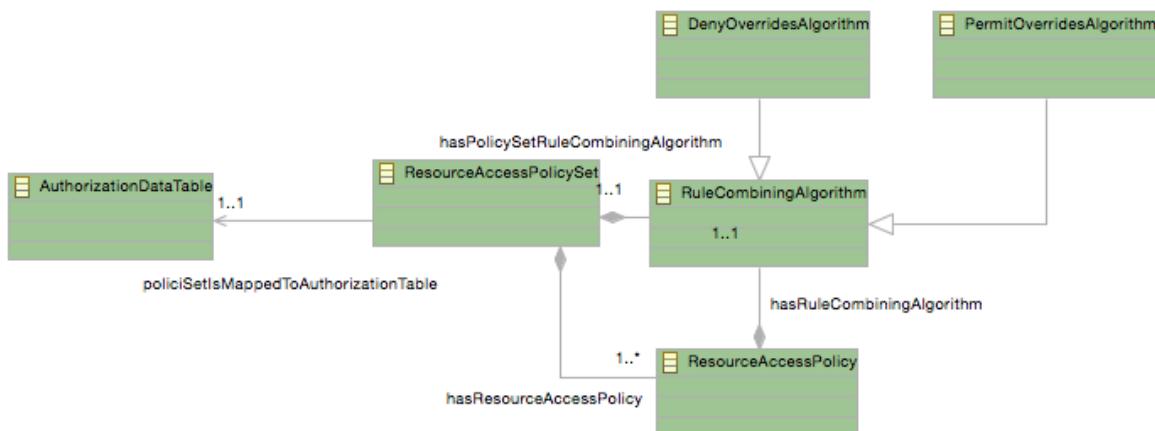


Figure 5-19 ResourceAccessPolicySet element and its relations

Properties

The *ResourceAccessPolicySet* element of the *ABAC Authorization PIM* extension meta-model does not have any property.

Relations

Table 5-54 ResourceAccessPolicySet's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
RuleCombiningAlgorithm	composition	1	The <i>ResourceAccessPolicySet</i> must have a composition association with exactly one <i>RuleCombiningAlgorithm</i> . This association models the fact that the evaluation of all the <i>ResourceAccessPolicies</i> of a resource is combined using the associated <i>RuleCombiningAlgorithm</i> in order to conclude the final response to a client's authorization request.
ResourceAccessPolicy	composition	1..*	The <i>ResourceAccessPolicySet</i> must have a composition association with at least one <i>ResourceAccessPolicy</i> . This association models the fact that every <i>ResourceAccessPolicySet</i> comprises of several <i>ResourceAccessPolicies</i> the evaluation of which conclude the final response to a client's authorization request.
AuthorizationDataTable	association	1	The <i>ResourceAccessPolicySet</i> must have exactly one association with an <i>AuthorizationTable</i> . This association models the fact that every <i>AuthorizableResource</i> 's <i>ResourceAccessPolicySets</i> are stored in the local envisioned system's database.

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of a *ResourceAccessPolicySet* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the *ABAC Authorization PIM* meta-model that can be found in appendix A.1.

- *hasSpecializedRuleCombiningAlgorithm*: This OCL constraint checks whether the *RuleCombiningAlgorithm* that is associated with this *ResourceAccessPolicySet* is further specialized to either *DenyOverridesAlgorithm* or *PermitOverridesAlgorithm*.

5.5.1.2.15 RuleCombiningAlgorithm Element

Overview

The *RuleCombiningAlgorithm* element models an authorization policy or rule, evaluation algorithm that is used when multiple rules/policies of an *AuthorizableResource* yield various permission

responses to a client's access requests. Every *RuleCombiningAlgorithm* is always further specialized as either *DenyOverridesAlgorithm* or *PermitOverridesAlgorithm*.

Properties

The *RuleCombiningAlgorithm* element of the *ABAC Authorization PIM* extension meta-model does not have any property.

Relations

The *RuleCombiningAlgorithm* element of the *ABAC Authorization PIM* extension meta-model does not have any relation.

Behavioural Restrictions

The *RuleCombiningAlgorithm* element of the *ABAC Authorization PIM* extension meta-model does not have any behavioural restriction.

5.5.1.2.16 DenyOverridesAlgorithm Element

Overview

The *DenyOverridesAlgorithm* element models a specialization of the *RuleCombiningAlgorithm*. If a *ResourceAccessPolicySet* or *ResourceAccessPolicy* is associated with a *DenyOverridesAlgorithm* access evaluation algorithm, then whenever one (even just one) *ResourceAccessPolicySet*'s *ResourceAccessPolicy* or *ResourceAccessPolicy*'s *ResourceAccessRule* yields permission denial then the envisioned system's client is not granted access to use the resource's functionality.

Properties

The *DenyOverridesAlgorithm* element of the *ABAC Authorization PIM* extension meta-model does not have any property other than those inherited from the *RuleCombiningAlgorithm*.

Relations

The *DenyOverridesAlgorithm* element of the *ABAC Authorization PIM* extension meta-model does not have any relation other than those inherited from the *RuleCombiningAlgorithm*.

Behavioural Restrictions

The *DenyOverridesAlgorithm* element of the *ABAC Authorization PIM* extension meta-model does not have any behavioural restrictions.

5.5.1.2.17 PermitOverridesAlgorithm Element

Overview

The *PermitOverridesAlgorithm* element models a *RuleCombiningAlgorithm* specialization. If a *ResourceAccessPolicySet* or a *ResourceAccessPolicy* is associated with a *PermitOverridesAlgorithm* then, whenever one (even just one) *ResourceAccessPolicySet*'s *ResourceAccessPolicy* or

ResourceAccessPolicy's ResourceAccessRule yields access permission, the envisioned system's client is granted access to use the underlying resource functionality.

Properties

The *PermitOverridesAlgorithm* element of the *ABAC Authorization PIM* extension meta-model does not have any property other than those inherited from the *RuleCombiningAlgorithm*.

Relations

The *PermitOverridesAlgorithm* element of the *ABAC Authorization PIM* extension meta-model does not have any relations other than those inherited from the *RuleCombiningAlgorithm*.

Behavioural Restrictions

The *PermitOverridesAlgorithm* element of the *ABAC Authorization PIM* extension meta-model does not have any behavioural restrictions.

5.5.1.2.18 *ResourceAccessPolicy* Element

Overview

The *ResourceAccessPolicy* element models an *AuthorizableResource*'s access policy, which comprises several *ResourceAccessRules*. Every such policy is evaluated whenever an envisioned system's client makes an access request. Figure 5-20 demonstrates the *ResourceAccessPolicy* element of the *ABAC Authorization PIM* extension meta-model and its relations.

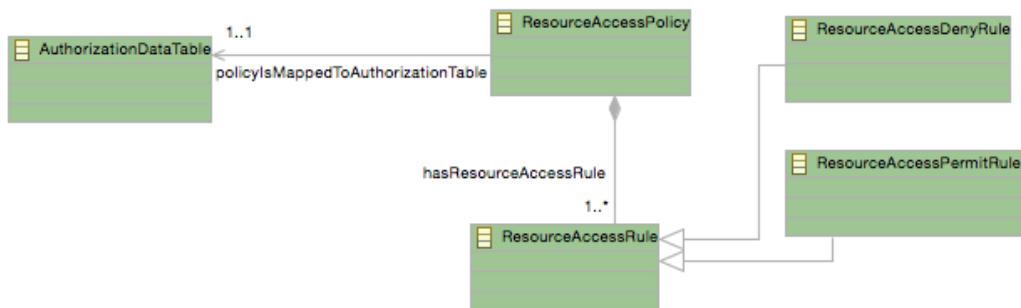


Figure 5-20 ResourceAccessPolicy element and its relations

Properties

The *ResourceAccessPolicy* element of the *ABAC Authorization PIM* extension meta-model does not have any property.

Relations

Table 5-55 ResourceAccessPolicy's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
RuleCombiningAlgorithm	composition	1	The <i>ResourceAccessPolicy</i> must have a

			composition association with exactly one <i>RuleCombiningAlgorithm</i> . This association models the fact that every <i>ResourceAccessPolicy</i> has one algorithm that is used to conclude whether permission should be granted to an envisioned system's client or not.
ResourceAccessRule	composition	1..*	The <i>ResourceAccessPolicy</i> must have at least one composition association with a <i>ResourceAccessRule</i> . This association models the fact that every <i>ResourceAccessPolicy</i> may have several access rules but at least one in order to be meaningful.
AuthorizationDataTable	association	1	The <i>ResourceAccessPolicy</i> must be associated with exactly one <i>AuthorizationDataTable</i> . This association models the fact that the <i>ResourceAccessPolicies</i> of a resource are stored in the envisioned system's local database.

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of a *ResourceAccessPolicy* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the *ABAC Authorization* PIM extension meta-model that can be found in appendix A.1.

- *hasSpecializedRuleCombiningAlgorithm*: This OCL constraint checks whether the associated *RuleCombiningAlgorithm* is always specialized to either *DenyOverridesAlgorithm* or *PermitOverridesAlgorithm*.
- *hasSpecializedAccessRules*: This OCL constraint checks whether every associated *ResourceAccessRule* is always specialized to either *ResourceAccessPermitRule* or *ResourceAccessDenyRule*.

5.5.1.2.19 ResourceAccessRule Element

Overview

The *ResourceAccessRule* element models an *ABAC* rule that is used to model the authorization scheme of a resource. Figure 5-21 demonstrates the *ResourceAccessRule* element of the *ABAC Authorization* PIM extension meta-model and its relations.

Properties

The *ResourceAccessRule* element of the *ABAC Authorization* PIM extension meta-model does not have any property.

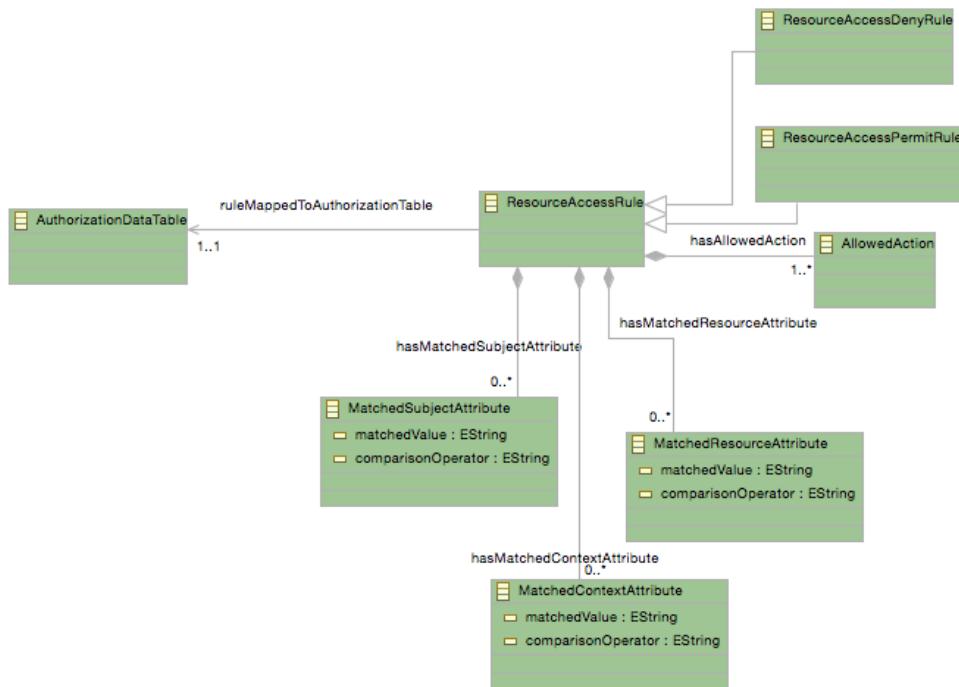


Figure 5-21 ResourceAccessRule element and its relations

Relations

Table 5-56 ResourceAccessRule's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
AuthorizationDataTable	association	1	The <i>ResourceAccessRule</i> must have exactly one association with an <i>AuthorizationDataTable</i> . This association models the fact that every <i>ResourceAccessRule</i> is stored in the envisioned system's local database.
AllowedAction	composition	1..*	The <i>ResourceAccessRule</i> must have at least one composition association with an <i>AllowedAction</i> . This association models the fact that every <i>ResourceAccessRule</i> may yield access to one or more of the resource's actions.
MatchedSubjectAttribute	composition	0..*	The <i>ResourceAccessRule</i> can have zero or more composition associations with <i>MatchedSubjectAttribute</i> elements. This association models the fact that some access rules may perform request issuer's attribute matching in order to evaluate an access request, whilst other not.
MatchedResourceAttribute	composition	0..*	The <i>ResourceAccessRule</i> can have zero or more composition associations with <i>MatchedResourceAttribute</i> elements. This association models the fact that some access rules may perform checks to the underlying

			resource's attribute prior yielding permission to the request issuer or not.
MatchedContextAttribute	composition	0..*	The <i>ResourceAccessRule</i> can have zero or more composition associations with <i>MatchedContextAttribute</i> elements. This association models the fact that some access rules may perform checks to several context attributes prior concluding whether to yield permission to the request issuer or not.

Behavioural Restrictions

The *ResourceAccessRule* element of the *ABAC Authorization PIM* extension meta-model does not have any behavioural restrictions.

5.5.1.2.20 *ResourceAccessDenyRule Element*

Overview

The *ResourceAccessDenyRule* element models a *ResourceAccessRule* specialization. In this case, every time an access request is evaluated positively by such a rule, the rule denies access to the issuer.

Properties

The *ResourceAccessDenyRule* element of the *ABAC Authorization PIM* extension meta-model does not have any property other than those inherited from the *ResourceAccessRule*.

Relations

The *ResourceAccessDenyRule* element of the *ABAC Authorization PIM* extension meta-model does not have any relation other than those inherited from the *ResourceAccessRule*.

Behavioural Restrictions

The *ResourceAccessDenyRule* element of the *ABAC Authorization PIM* extension meta-model does not have any behavioural restrictions.

5.5.1.2.21 *ResourceAccessPermitRule Element*

Overview

The *ResourceAccessPermitRule* element models a *ResourceAccessRule* specialization. In this case, whenever an access request is evaluated positively by this access rule, it yields permission to the request issuer.

Properties

The *ResourceAccessPermitRule* element of the *ABAC Authorization PIM* extension meta-model does not have any property other than those inherited from the *ResourceAccessRule*.

Relations

The *ResourceAccessPermitRule* element of the *ABAC Authorization PIM* extension meta-model does not have any relations other than those inherited from the *ResourceAccessRule*.

Behavioural Restrictions

The *ResourceAccessPermitRule* element of the *ABAC Authorization PIM* extension meta-model does not have any behavioural restrictions.

5.5.1.2.22 AllowedAction Element

Overview

The *AllowedAction* element models an annotation of the *ABAC Authorization PIM* extension meta-model that is intended to annotate an existent Core PIM *CRUDActivity* as an allowed action when an access rule yields permission to the issuer. With this annotation, whenever an access request is evaluated and permission is granted to the issuer, the issuer is authorized to use the underlying functionality of that *CRUDActivity*. Figure 5-22 demonstrates the *AllowedAction* element of the *ABAC Authorization PIM* extension meta-model and its relations.

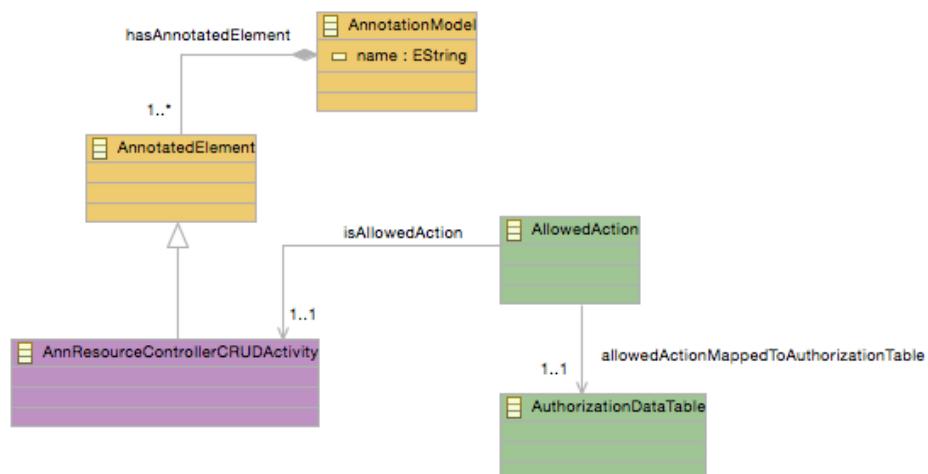


Figure 5-22 AllowedAction element and its relations

Properties

The *AllowedAction* element of the *ABAC Authorization PIM* extension meta-model does not have any property.

Relations

Table 5-57 AllowedAction's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
AuthorizationDataTable	association	1	The <i>AllowedAction</i> must be associated with exactly one <i>AuthorizationDataTable</i> .

			This association models the fact that every <i>AllowedAction</i> of a resource access rule is stored in the envisioned system's local database.
AnnResourceControllerCRUDActivity	association	1	The <i>AllowedAction</i> must be associated with exactly one <i>AnnResourceControllerCRUDActivity</i> , which references one <i>ResourceControllerCRUDActivity</i> element of the Core PIM model. With this annotation, whenever the overlying access rule yields access to the request issuer, the issuer will be able to use the underlying functionality of that <i>ResourceControllerCRUDActivity</i> .

Behavioural Restrictions

The *AllowedAction* element of the *ABAC Authorization PIM* extension meta-model does not have any behavioural restriction.

5.5.1.2.23 *MatchedResourceAttribute* Element

Overview

The *MatchedResourceAttribute* element models an annotation of the *ABAC Authorization PIM* extension meta-model that is intended to annotate an existent Core PIM *PIMComponentProperty*. With this annotation, the *PIMComponentProperty*'s runtime value of the overlying *AuthorizableResource*, will always be matched against the allowed values that are prescribed in any *ResourceAccessRule* of it, in order to conclude whether the request issuer should be granted access or not. Figure 5-23 demonstrates the *MatchedResourceAttribute* element of the *ABAC Authorization PIM* extension meta-model and its relations.

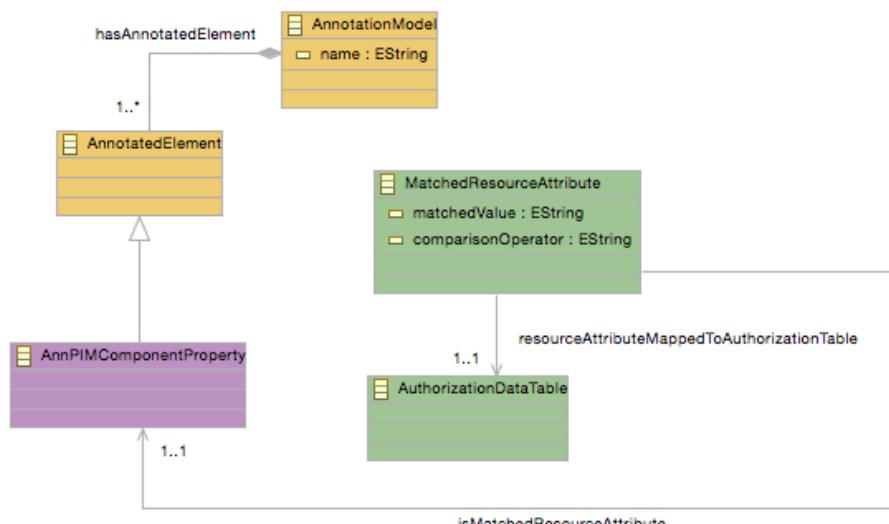


Figure 5-23 MatchedResourceAttribute element and its relations

Properties

Table 5-58 MatchedResourceAttribute's Properties

Name	Type	Multiplicity	Explanation
matchedValue	EString	1..*	This is a list of values with which the runtime value of the annotated <i>PIMComponentProperty</i> will always be compared prior concluding whether the request issuer should be granted access or not.
comparisonOperator	EString	1	This operator models the way the runtime values are compared with the prescribed <i>matchedValue</i> list of this rule e.g. "equal", "not equal", "greater" etc.

Relations

Table 5-59 MatchedResourceAttribute's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
AuthorizationDataTable	association	1	The <i>MatchedResourceAttribute</i> must be associated with exactly one <i>AuthorizationDataTable</i> . This association models the fact that the <i>MatchedResourceAttributes</i> of an <i>AuthorizableResource</i> are persisted in the envisioned system's local database.
AnnPIMComponentProperty	association	1	The <i>MatchedResourceAttribute</i> must be associated with exactly one <i>AnnPIMComponentProperty</i> , which references a <i>PIMComponentProperty</i> of the Core PIM model. With this annotation, the runtime value of that <i>PIMComponentProperty</i> will be used when the overlying <i>ResourceAccessRule</i> is evaluated in order to conclude whether the request issuer should be granted access or not.

Behavioural Restrictions

The *MatchedResourceAttribute* element of the *ABAC Authorization PIM* extension meta-model does not have any behavioural restrictions.

5.5.1.2.24 MatchedContextAttribute Element

Overview

The *MatchedContextAttribute* element models an annotation of the *ABAC Authorization PIM* extension meta-model that is intended to annotate an existent Core PIM *PIMComponentProperty*. With this annotation, the *PIMComponentProperty*'s runtime value of the overlying context resource, will always be matched against the allowed values that are prescribed in any *ResourceAccessRule* of it, in order to conclude whether the request issuer should be granted access or not. Figure 5-24

demonstrates the *MatchedContextAttribute* element of the *ABAC Authorization* PIM extension meta-model and its relations.

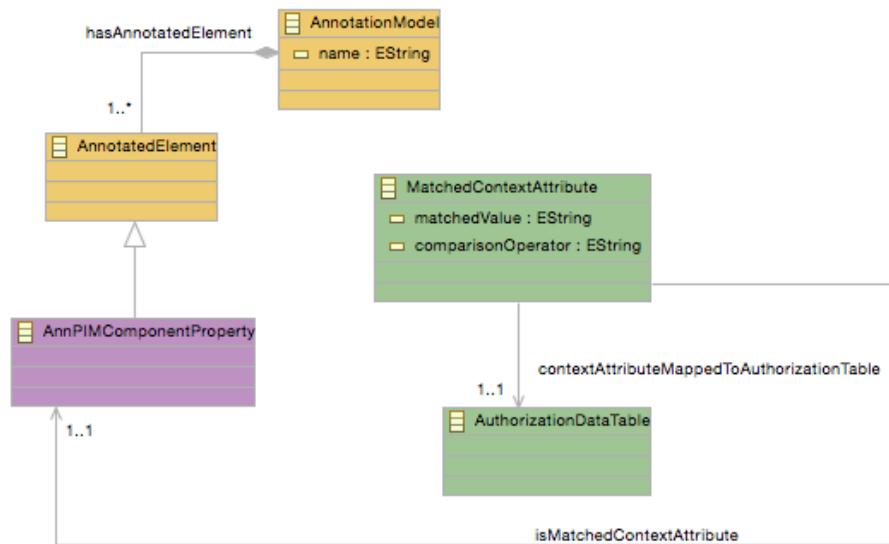


Figure 5-24 MatchedContextAttribute element and its relations

Properties

Table 5-60 MatchedContextAttribute's Properties

Name	Type	Multiplicity	Explanation
matchedValue	EString	1..*	This is a list of values with which the runtime value of the annotated <i>PIMComponentProperty</i> will always be compared prior concluding whether the request issuer should be granted access or not.
comparisonOperator	EString	1	This operator models the way the runtime values are compared with the prescribed <i>matchedValue</i> list of this rule e.g. "equal", "not equal", "greater" etc.

Relations

Table 5-61 MatchedContextAttribute's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
AuthorizationDataTable	association	1	The <i>MatchedContextAttribute</i> must be associated with exactly one <i>AuthorizationDataTable</i> . This association models the fact that the <i>MatchedContextAttributes</i> of a context resource are persisted in the envisioned system's local database.
AnnPIMComponentProperty	association	1	The <i>MatchedContextAttribute</i> must be associated with exactly one

			<p><i>AnnPIMComponentProperty</i>, which references a <i>PIMComponentProperty</i> of the Core PIM model. With this annotation, the runtime value of that <i>PIMComponentProperty</i> will be used when the overlying <i>ResourceAccessRule</i> is evaluated in order to conclude whether the request issuer should be granted access or not.</p>
--	--	--	--

Behavioural Restrictions

The *MatchedContextAttribute* element of the *ABAC Authorization PIM* extension meta-model does not have any behavioural restriction.

5.5.1.2.25 *MatchedSubjectAttribute* Element

Overview

The *MatchedSubjectAttribute* element models an annotation of the *ABAC Authorization PIM* extension meta-model that is intended to annotate an existent Core PIM *PIMComponentProperty*. With this annotation, the *PIMComponentProperty*'s runtime value of the overlying *AuthorizationSubject*, will always be matched against the allowed values that are prescribed in any *ResourceAccessRule* of it, in order to conclude whether the request issuer should be granted access or not. Figure 5-25 demonstrates the *MatchedSubjectAttribute* element of the *ABAC Authorization PIM* extension meta-model and its relations.

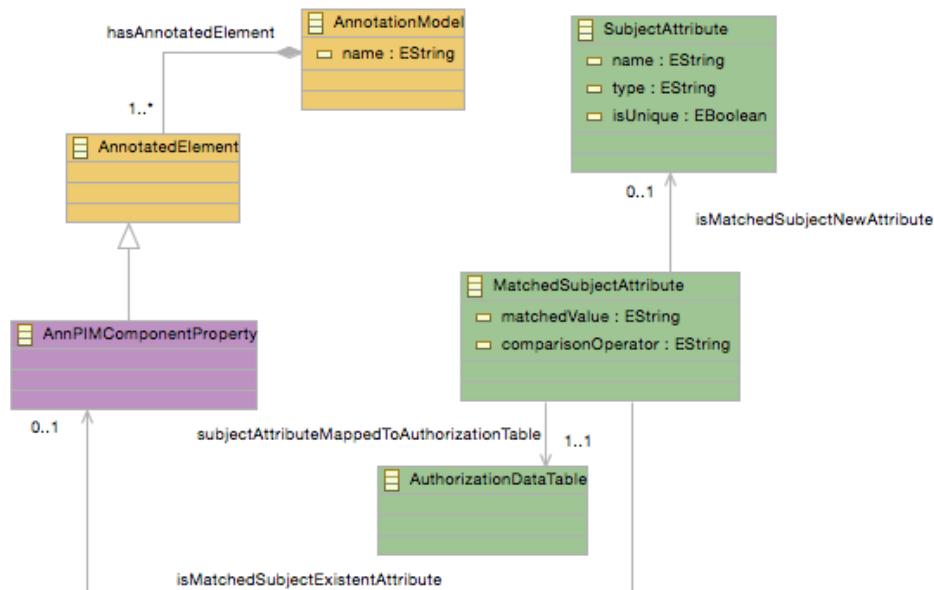


Figure 5-25 MatchedSubjectAttribute element and its relations.

Properties

Table 5-62 MatchedSubjectAttribute's Properties

Name	Type	Multiplicity	Explanation
matchedValues	EString	1	This is a list of values with which the runtime value of the annotated <i>PIMComponentProperty</i> will always be compared prior concluding whether the request issuer should be granted access or not.
comparisonOperator	EString	1	This operator models the way the runtime values are compared with the prescribed <i>matchedValue</i> list of this rule e.g. "equal", "not equal", "greater" etc.

Relations

Table 5-63 MatchedSubjectAttribute's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
AuthorizationDataTable	association	1	The <i>MatchedSubjectAttribute</i> must be associated with exactly one <i>AnnPIMComponentProperty</i> , which references a <i>PIMComponentProperty</i> of the Core PIM model. With this annotation, the runtime value of that <i>PIMComponentProperty</i> will be used when the overlying <i>ResourceAccessRule</i> is evaluated in order to conclude whether the request issuer should be granted access or not.
AnnPIMComponentProperty	association	1..*	The <i>MatchedSubjectAttribute</i> must be associated with exactly one <i>AnnPIMComponentProperty</i> , which references a <i>PIMComponentProperty</i> of the Core PIM model. With this annotation, the runtime value of that <i>PIMComponentProperty</i> will be used when the overlying <i>ResourceAccessRule</i> is evaluated in order to conclude whether the request issuer should be granted access or not.

Behavioural Restrictions

The *MatchedSubjectAttribute* element of the *ABAC Authorization PIM* extension meta-model does not have any behavioural restriction.

]

5.5.1.2.26 AuthorizationPerformer Element

Overview

The *AuthorizationPerformer* element models an annotation of the *ABAC Authorization* PIM extension meta-model that is intended to annotate an existent Core PIM *CRUDActivityHandler*. With this annotation, the *CRUDActivityHandler* will embed dedicated code in order to handle authorization requests and their responses. Figure 5-26 demonstrates the *AuthorizationPerformer* element of the *ABAC Authorization* PIM extension meta-model and its relations.

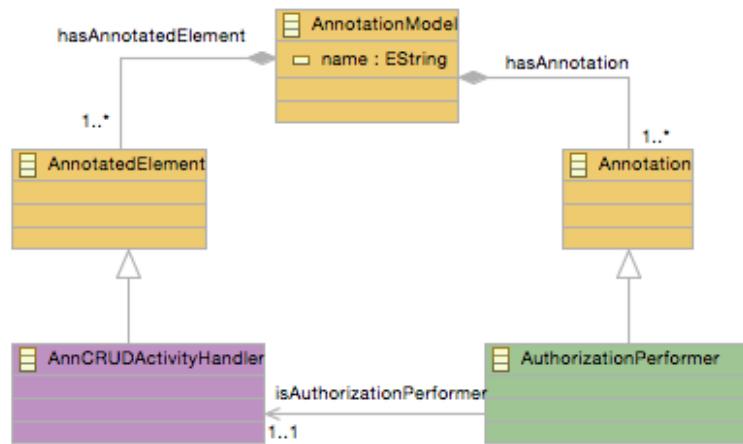


Figure 5-26 AuthorizationPerformer annotation and its relations.

Properties

The *AuthorizationPerformer* element of the *ABAC Authorization* PIM extension meta-model does not have any property.

Relations

Table 5-64 AuthorizationPerformer's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
AnnCRUDActivityHandler	association	1	The <i>AuthorizationPerformer</i> must be associated with exactly one <i>AnnCRUDActivityHandler</i> , which references one <i>CRUDActivityHandler</i> of the Core PIM model. With this annotation, the <i>CRUDActivityHandler</i> will embed any needed code to handle authorization requests and provide the appropriate response to the request issuer.

Behavioural Restrictions

The *AuthorizationPerformer* element of the *ABAC Authorization* PIM extension meta-model does not have any behavioural restriction.

5.5.1.2.27 AuthorizationPolicyEvaluator Element

Overview

The *AuthorizationPolicyEvaluator* element models the key component of every envisioned system that will be produced by S-CASE, which will evaluate any *ResourceAccessRule* of *ResourceAccessPolicySets* in order to conclude whether an incoming access request should be accepted or not.

Properties

The *AuthorizationPolicyEvaluator* element of the ABAC Authorization PIM extension meta-model does not have any property.

Relations

Table 5-65 AuthorizationPolicyEvaluator's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
AuthorizationDataHandler	association	1	The <i>AuthorizationPolicyEvaluator</i> must be associated with exactly one <i>AuthorizationDataHandler</i> . This association models the fact that the <i>AuthorizationPolicyEvaluator</i> uses the <i>AuthorizationDataHandler</i> in order to access authorization data that is stored in the envisioned system's local database.

Behavioural Restrictions

The *AuthorizationPolicyEvaluator* element of the ABAC Authorization PIM extension meta-model does not have any behavioural restriction.

5.5.1.2.28 AuthorizationDataHandler Element

Overview

The *AuthorizationDataHandler* element models an annotation of the ABAC Authorization PIM extension meta-model that is intended to annotate an existent Core PIM *DatabaseController*. With this annotation, the *DatabaseController* will embed the needed functionality by the rest of the envisioned system, in order to handle authorization data that is stored in its local database. Figure 5-27 demonstrates the *AuthorizationDataHandler* element of the ABAC Authorization PIM extension meta-model and its relations.

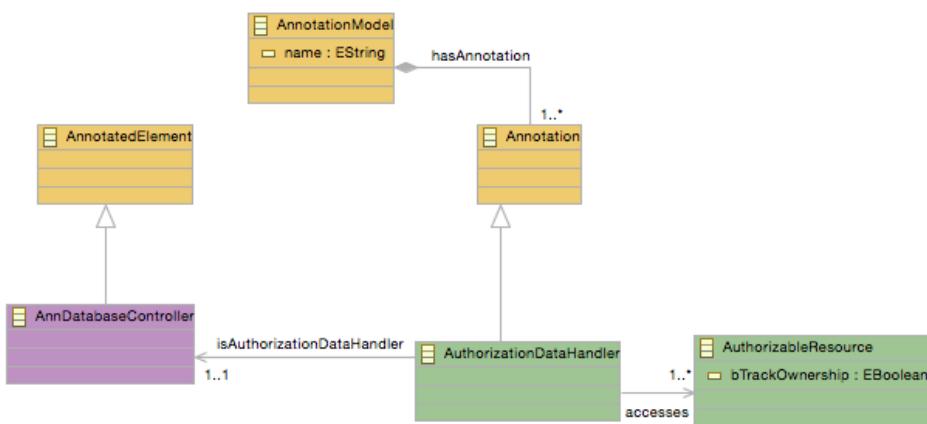


Figure 5-27 AuthorizationDataHandler annotation and its relations.

Properties

The *AuthorizationDataHandler* element of the *ABAC Authorization PIM* extension meta-model does not have any property.

Relations

Table 5-66 AuthorizationDataHandler's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
AuthorizableResource	association	1..*	The <i>AuthorizationDataHandler</i> must be associated with at least one <i>AuthorizableResource</i> . This association models the fact that the <i>AuthorizationDataHandler</i> embeds the needed functionality in order to be able to store and retrieve authorization data of any such <i>AuthorizableResource</i> .
AnnDatabaseController	association	1	The <i>AuthorizationDataHandler</i> must be associated with exactly one <i>AnnDatabaseController</i> , which references one <i>DatabaseController</i> element of the Core PIM model. With this annotation, the <i>DatabaseController</i> will also embed the needed functionality by the rest of the envisioned system, in order to be able to store and retrieve authorization data for any of the <i>AuthorizableResources</i> it has.
AuthorizationDataTable	composition	1..*	The <i>AuthorizationDataHandler</i> must have at least one composition association with an <i>AuthorizationDataTable</i> . This association models the fact that the <i>AuthorizationDataHandler</i> stores and retrieves authorization data to/from the envisioned system's local database.

Behavioural Restrictions

The *AuthorizationDataHandler* element of the *ABAC Authorization PIM* extension meta-model does not have any behavioural restriction.

5.5.1.2.29 AuthorizationDataTable Element

Overview

The *AuthorizationDataTable* element models an *RDBMS* table of the envisioned system's local database, which is used to store authorization data.

Properties

The *AuthorizationDataTable* element of the *ABAC Authorization PIM* extension meta-model does not have any property.

Relations

Table 5-67 AuthorizationDataTable's Relations

Relation With PIM Extension Element	Type	Multiplicity	Structural Constraints
AuthorizationDataTableColumn	composition	1..*	The <i>AuthorizationDataTable</i> must have at least one composition association with one <i>AuthorizationDataTableColumn</i> . This association models the fact that every <i>AuthorizationDataTable</i> comprises of several columns.

Behavioural Restrictions

The *AuthorizationDataTable* element of the *ABAC Authorization PIM* extension meta-model does not have any behavioural restriction.

6 Platform Specific Model UML Profile Extensions

6.1 PSM UML Profile Extension Goals

The following subsections describe the way the PIM abstract design of the extra functionality that is automatically embedded to the envisioned systems that S-CASE produces is specialized with concrete technologies to form the corresponding PSM. Following the meta-model extension mechanism that section 3.1 presents, the following Core PSM extensions allow the S-CASE MDE engine to embed to the systems it produces authentication and ABAC authorization mechanisms, external 3rd party service compositions as well as database searching capabilities.

6.2 PSM Authentication Ecore Meta-model Definition

6.2.1.1 Introduction

In order to fully define the Authentication Ecore PSM extension meta-model (Figure 6-1), its various aspects must be explained, demonstrated as well as documented. This is done in the following subsections following the format section 4.1.2.1 describes.

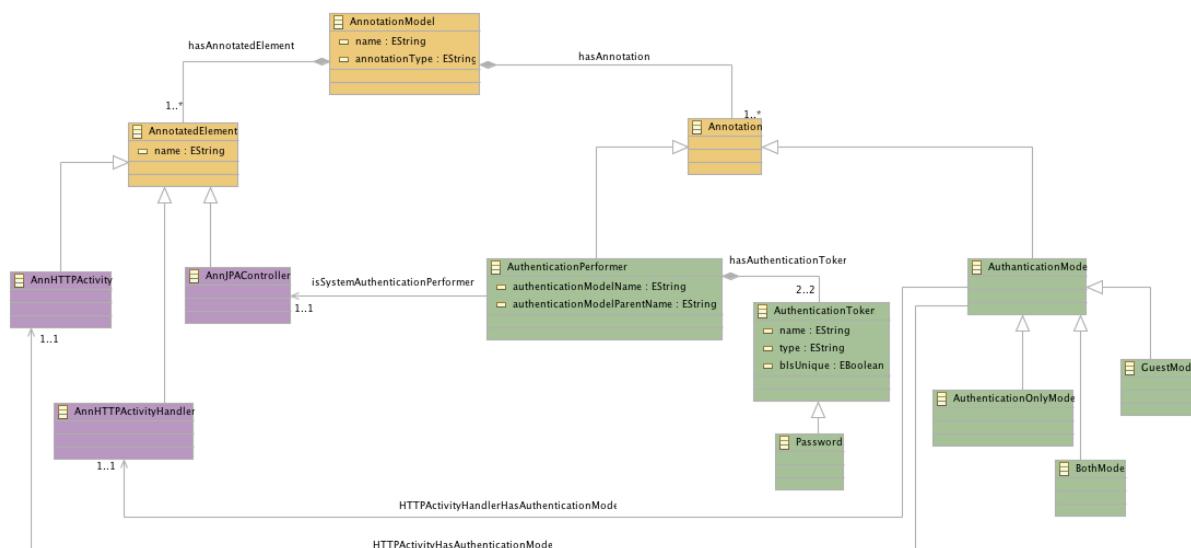


Figure 6-1 Authentication PSM extension meta-model

6.2.1.2 PSM Authentication Ecore Meta-model Elements

6.2.1.2.1 AnnotationModel Element

Overview

The *AnnotationModel* element is the root element of the *Authentication* PSM extension meta-model and comprises of all the introduced *Annotations* and the existing Core PSM elements to which these annotations are attached in order to alter their semantic meaning, design and/or configuration.

Properties

Table 6-1 AnnotationModel's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the name of the <i>Annotation</i> model.

Relations

Table 6-2 AnnotationModel's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
AnnotatedElement	composition	1..*	This composition association models the fact that every <i>AnnotationModel</i> element must have at least one <i>AnnotatedElement</i> .
Annotation	composition	1..*	This composition association models the fact that every <i>AnnotationModel</i> element must have at least one <i>Annotation</i> .

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of an *AnnotationModel* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the *Authentication* PSM extension meta-model that can be found in appendix A.1.

- *exactlyOneAnnJPAControllerExists*: This OCL constraint checks whether there exists exactly one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnJPAController*.
- *atLeastOneAnnHTTPActivityExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnHTTPActivity*
- *atLeastOneAnnHTTPActivityHandlerExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* and is of type *AnnHTTPActivityHandler*
- *sameNumberOfActivitiesAndHandlersExist*: This OCL constraint checks whether there exist the same number of *AnnotatedElements* that are associated with this *AnnotationModel* and are of type *AnnHTTPActivities* as the number of *AnnotatedElements* that are of type *AnnHTTPActivityHandlers*.
- *exactlyOneAuthenticationperformerExists*: This OCL constraint checks whether there exists exactly one *Annotation* element that is associated with this *AnnotationModel* that is of type *AuthenticationPerformer*.
- *twoAuthenticationTokensExist*: This OCL constraint checks whether there exist exactly two *Annotation* elements that are associated with this *AnnotationModel* that are of type *AuthenticationToken*.
- *exactlyOneAuthenticationTokenIsPassword*: This OCL constraint checks whether there exists exactly one *Annotation* element that is associated with this *AnnotationModel* that is of type *Password*.

- *atLeastOneAuthenticationModeAnnotationExists*: This OCL constraint checks whether there exists at least one *Annotation* element that is associated with this *AnnotationModel* that is of type *GuestMode*, or *AuthenticationOnlyMode*, or *BothMode*.

6.2.1.2.2 AnnotatedElement Element

Overview

The *AnnotatedElement* element models any existent Core PSM meta-model element that can be annotated by the *Authentication* Core PSM extension in order to extend its semantic meaning and/or design. It is always further specialized to more specific categories of annotated elements.

Properties

The *AnnotatedElement* element of the *Authentication PSM* extension meta-model does not have any property.

Relations

The *AnnotatedElement* element of the *Authentication PSM* extension meta-model does not have any relation

Behavioural Restrictions

The *AnnotatedElement* element of the *Authentication PSM* extension meta-model does not have any behavioural restriction.

6.2.1.2.3 Annotation Element

Overview

The *Annotation* element models any new semantic meaning or design that is attached to a Core PSM meta-model element. It is always further specialized to more specific categories of annotations.

Properties

The *Annotation* element of the *Authentication PSM* extension meta-model does not have any property.

Relations

The *Annotation* element of the *Authentication PSM* extension meta-model does not have any relation.

Behavioural Restrictions

The *Annotation* element of the *Authentication PSM* extension meta-model does not have any relation.

6.2.1.2.4 AnnHTTPActivity Element

Overview

The *AnnHTTPActivity* element models an existent *HTTPActivity* of the Core PSM model that can be annotated by the *Authentication PSM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnHTTPActivity* element of the *Authentication PSM* extension meta-model does not have any property.

Relations

Table 6-3 AnnHTTPActivity's Relations

Relation With PSM Element	Type	Multiplicity	Structural Constraints
HTTPActivity	reference	1	The <i>AnnHTTPActivity</i> must reference exactly one <i>HTTPActivity</i> element of the Core PSM model.

Behavioural Restrictions

The *AnnHTTPActivity* element of the *Authentication PSM* extension meta-model does not have any behavioural restriction.

6.2.1.2.5 AnnHTTPActivityHandler Element

Overview

The *AnnHTTPActivityHandler* element models an existent *HTTPActivityHandler* of the Core PSM model that can be annotated by the *Authentication PSM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnHTTPActivityHandler* element of the *Authentication PSM* extension meta-model does not have any property.

Relations

Table 6-4 AnnHTTPActivityHandler's Relations

Relation With PSM Element	Type	Multiplicity	Structural Constraints
HTTPActivityHandler	reference	1	The <i>AnnHTTPActivityHandler</i> must reference exactly one <i>HTTPActivityHandler</i> element of the Core PSM model.

Behavioural Restrictions

The *AnnHTTPActivityHandler* element of the *Authentication PSM* extension meta-model does not have any behavioural restriction.

6.2.1.2.6 AnnJPAController Element

Overview

The *AnnJPAController* element models an existent *JPAController* of the Core PSM model that can be annotated by the *Authentication PSM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnJPAController* element of the *Authentication PSM* extension meta-model does not have any property.

Relations

Table 6-5 AnnJPAController's Relations

Relation With PSM Element	Type	Multiplicity	Structural Constraints
JPAController	reference	1	The <i>AnnJPAController</i> must reference exactly one <i>JPAController</i> element of the Core PSM model.

Behavioural Restrictions

The *AnnJPAController* element of the *Authentication PSM* extension meta-model does not have any behavioural restriction.

6.2.1.2.7 AuthenticationPerformer Element

Overview

The *AuthenticationPerformer* element models an annotation of the *Authentication PSM* extension meta-model that is intended to annotate an existent Core PSM *JPAController*. With this annotation, the *JPAController* embeds the needed functionality to handle authentication data that are stored in the envisioned system's local database. Figure 6-2 demonstrates the *AuthenticationPerformer* element of the *Authentication PSM* extension meta-model and its relations.

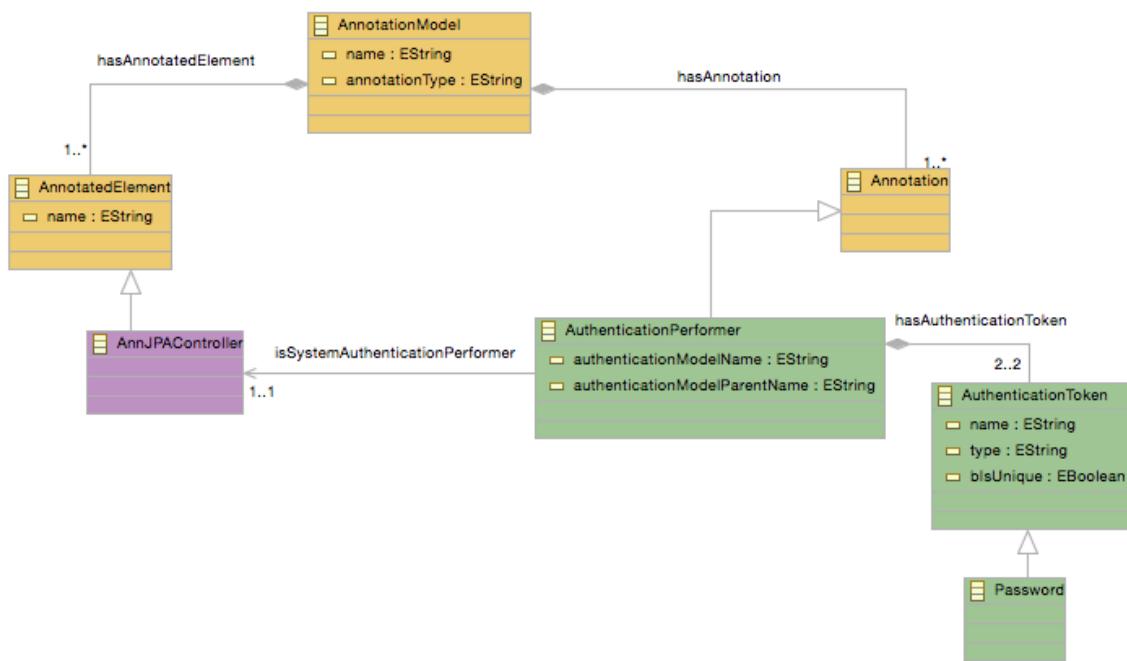


Figure 6-2 AuthenticationPerformer annotation and its relations

Properties

Table 6-6 AuthenticationPerformer's Properties

Name	Type	Multiplicity	Explanation
authenticationmodelName	EString	1	This is the name of the authentication model that is used for authentication.
authenticationModelParentName	EString	1	This is the name of the parent of the authentication model that is used for authentication.

Relations

Table 6-7 AuthenticationPerformer's Relations

Relation With PSM Extension Element	Type	Multiplicity	Structural Constraints
AnnJPAController	association	1	The AuthenticationPerformer must be associated with exactly one AnnJPAController , which references a JPAController of the Core PSM model. With this annotation, the JPAController will also embed the needed functionality to handle authentication data.
AuthenticationToken	composition	2	The AuthenticationPerformer must have a composition association with exactly two AuthenticationTokens . This association models the fact that every AuthenticationPerformer uses two AuthenticationTokens as credentials, namely a password and a username.

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of a *AuthenticationPerformer* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the Autnentication PSM extension meta-model that can be found in appendix A.1.

- *hasExactlyOnePasswordToken*: This OCL constraint checks whether this *AuthenticationPerformer* is associated with exactly one *Password AuthenticationToken*.

6.2.1.2.8 AuthenticationToken Element

Overview

The *AuthenticationToken* element models an authentication token, either a *Password* token or one used as username. These *AuthenticationTokens* are used as credentials within the envisioned systems that S-CASE produces.

Properties

Table 6-8 AuthenticationToken's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the name of the <i>AuthenticationToken</i> .
type	EString	1	This is the type of the <i>AuthenticationToken</i> .
blsUnique	EBoolean	1	This is the multiplicity of the <i>AuthenticationToken</i> . If this <i>EBoolean</i> is true, then the <i>AuthenticationToken</i> has multiplicity one, otherwise it is a multivalued array.

Relations

The *AuthenticationToken* element of the *Authentication PSM* extension meta-model does not have any relation.

Behavioural Restrictions

The *AuthenticationToken* element of the *Authentication PSM* extension meta-model does not have any behavioural restriction.

6.2.1.2.9 Password Element

Overview

The *Password* element models an *AuthenticationToken* specialization, which specifies that a specific *AuthenticationToken* should be handled as password.

Properties

The *Password* element of the *Authentication PSM* extension meta-model does not have any property other than those inherited from the *AuthenticationToken*.

Relations

The *Password* element of the *Authentication PSM* extension meta-model does not have any relation other than those inherited from the *AuthenticationToken*.

Behavioural Restrictions

The *Password* element of the *Authentication PSM* extension meta-model does not have any behavioural restriction.

6.2.1.2.10 AuthenticationMode Element

Overview

The *AuthenticationMode* element models an annotation of the *Authentication* PSM extension meta-model that is intended to annotate an existent Core PSM *HTTPActivity* or *HTTPActivityHandler*. With this annotation, the *HTTPActivity* or the *HTTPActivityHandler* will only be accessible to authenticated users, guest users or either, depending on the type of the *AuthenticationMode* specialization. The possible specializations are *AuthenticationOnlyMode*, *BothMode* or *GuestMode*. Figure 6-3 demonstrates the *AuthenticationMode* element of the *Authentication* PSM extension meta-model and its relations.

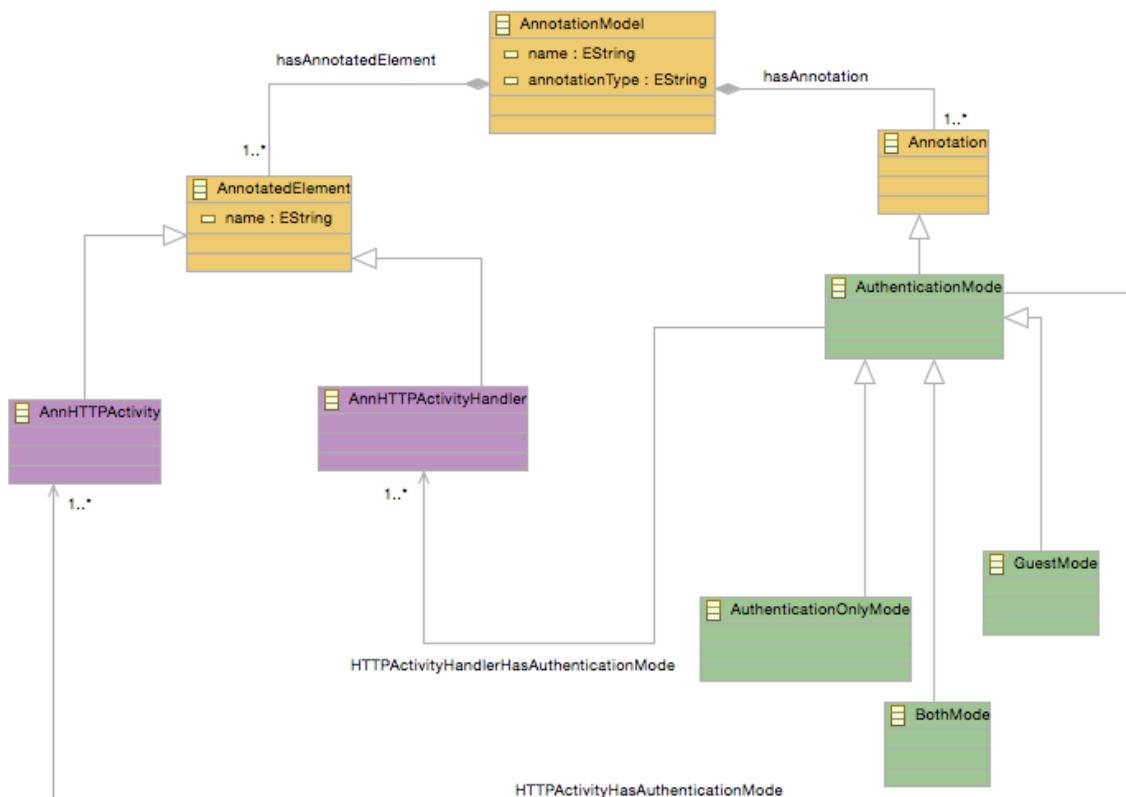


Figure 6-3 AuthenticationMode annotation and its relations

Properties

The *AuthenticationMode* element of the *Authentication PSM* extension meta-model does not have any property.

Relations

Table 6-9 AuthenticationMode's Relations

Relation With PSM Extension Element	Type	Multiplicity	Structural Constraints
AnnHTTPActivity	association	1	The <i>AuthenticationMode</i> must be associated with exactly one <i>AnnHTTPActivity</i> , which references one <i>HTTPActivity</i> element of the Core PSM model. With this annotation, the <i>HTTPActivity</i> will only be accessed if the user is authenticated, guest or either way, depending on the <i>AuthenticationMode</i> 's specialization.
AnnHTTPActivityHandler	association	1	The <i>AuthenticationMode</i> must be associated with exactly one <i>AnnHTTPActivityHandler</i> , which references one <i>HTTPActivityHandler</i> element of the Core PSM model. With this annotation, the <i>HTTPActivityHandler</i> will embed the needed code to as to expose the resource's functionality

Behavioural Restrictions

The *AuthenticationMode* element of the *Authentication PSM* extension meta-model does not have any behavioural restriction.

6.2.1.2.11 GuestMode Element

Overview

The *GuestMode* element models an *AuthenticationMode* specialization. With this specialization the annotated *HTTPActivity* or *HTTPActivityHandler* will expose their functionality to guest users.

Properties

The *GuestMode* element of the *Authentication PSM* extension meta-model does not have any property other than those inherited from the *AuthenticationMode*.

Relations

The *GuestMode* element of the *Authentication PSM* extension meta-model does not have any relation other than those inherited from the *AuthenticationMode*.

Behavioural Restrictions

The *GuestMode* element of the *Authentication PSM* extension meta-model does not have any behavioural restriction.

6.2.1.2.12 BothMode Element

Overview

The *BothMode* element models an *AuthenticationMode* specialization. The *HTTPActivity* or *HTTPActivityHandler* that is annotated with the *BothMode* annotation, exposes its functionality to both authenticated and guest users.

Properties

The *BothMode* element of the *Authentication PSM* extension meta-model does not have any property other than those inherited from the *AuthenticationMode*.

Relations

The *BothMode* element of the *Authentication PSM* extension meta-model does not have any relation other than those inherited from the *AuthenticationMode*.

Behavioural Restrictions

The *BothMode* element of the *Authentication PSM* extension meta-model does not have any behavioural restriction.

6.2.1.2.13 AuthenticationOnlyMode Element

Overview

The *AuthenticationOnlyMode* element models an *AuthenticationMode* specialization. The *HTTPActivity* or *HTTPActivityHandler* that is annotated with the *AuthenticationOnlyMode* annotation, will only expose its functionality to authenticated users.

Properties

The *AuthenticationOnlyMode* element of the *Authentication PSM* extension meta-model does not have any property other than those inherited from the *AuthenticationMode*.

Relations

The *AuthenticationOnlyMode* element of the *Authentication PSM* extension meta-model does not have any relation other than those inherited from the *AuthenticationMode*.

Behavioural Restrictions

The *AuthenticationOnlyMode* element of the *Authentication PSM* extension meta-model does not have any behavioural restriction.

6.3 PSM Database Searching Ecore Meta-model Definition

6.3.1.1 Introduction

In order to fully define the *Database Searching* Ecore PSM extension meta-model (Figure 6-4), its various aspects must be explained, demonstrated as well as documented. This is done in the following subsections following the format section 4.1.2.1 describes.

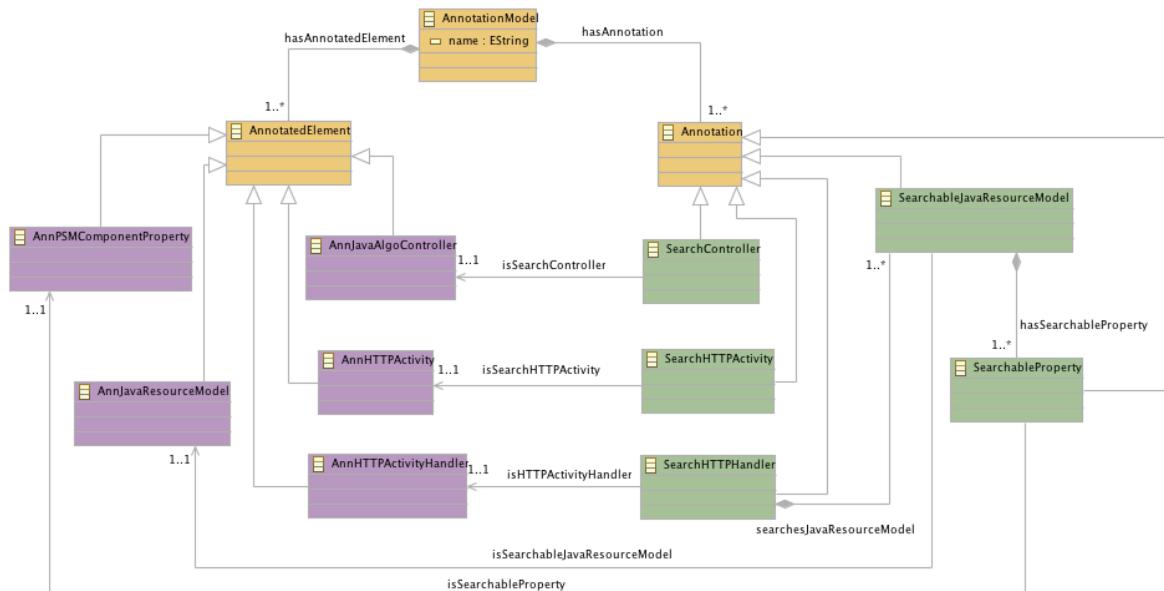


Figure 6-4 Database Searching PSM extension meta-model

6.3.1.2 PSM Database Searching Ecore Meta-model Elements

6.3.1.2.1 PSM Database Searching Custom Data Types Definition

6.3.1.2.2 AnnotationModel Element

Overview

The *AnnotationModel* element is the root element of the *Database Searching* PSM extension meta-model and comprises of all the introduced *Annotations* and the existing Core PSM elements to which these annotations are attached in order to alter their semantic meaning, design and/or configuration.

Properties

Table 6-10 AnnotationModel's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the name of the <i>Annotation</i> model.

Relations**Table 6-11 AnnotationModel's Relations**

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
AnnotatedElement	composition	1..*	This composition association models the fact that every <i>AnnotationModel</i> element must have at least one <i>AnnotatedElement</i> .
Annotation	composition	1..*	This composition association models the fact that every <i>AnnotationModel</i> element must have at least one <i>Annotation</i> .

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of an *AnnotationModel* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the *Database Searching* PSM extension meta-model that can be found in appendix A.1.

- *atLeastOneAnnJavaAlgoControllerExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnJavaAlgoController*.
- *atLeastOneAnnHTTPActivityExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnHTTPActivity*.
- *atLeastOneAnnHTTPActivityHandlerExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnHTTPActivity*.
- *atLeastOneAnnJavaResourceModelExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* and is of type *AnnJavaResourceModel*.
- *atLeastOneAnnPSMComponentPropertyExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* and is of type *AnnPSMComponentProperty*.
- *atLeastOneSearchControllerExists*: This OCL constraint checks whether there exists at least one *Annotation* element that is associated with this *AnnotationModel* that is of type *SearchController*.
- *atLeastOneSearchHTTPActivityExists*: This OCL constraint checks whether there exists at least one *Annotation* element that is associated with this *AnnotationModel* that is of type *SearchHTTPActivity*.

- *atLeastOneSearchHTTPActivityHandlerExists*: This OCL constraint checks whether there exists at least one *Annotation* element that is associated with this *AnnotationModel* that is of type *SearchHTTPActivityHandler*.

6.3.1.2.3 *AnnotatedElement* Element

Overview

The *AnnotatedElement* element models any existent Core PSM meta-model element that can be annotated by the *Database Searching* Core PSM extension in order to extend its semantic meaning and/or design. It is always further specialized to more specific categories of annotated elements.

Properties

The *AnnotatedElement* element of the *Database Searching PSM* extension meta-model does not have any property.

Relations

The *AnnotatedElement* element of the *Database Searching PSM* extension meta-model does not have any relation.

Behavioural Restrictions

The *AnnotatedElement* element of the *Database Searching PSM* extension meta-model does not have any behavioural restriction.

6.3.1.2.4 *Annotation* Element

Overview

The *Annotation* element models any new semantic meaning or design that is attached to a Core PSM meta-model element. It is always further specialized to more specific categories of annotations.

Properties

The *Annotation* element of the *Database Searching PSM* extension meta-model does not have any property.

Relations

The *Annotation* element of the *Database Searching PSM* extension meta-model does not have any relation.

Behavioural Restrictions

The *Annotation* element of the *Database Searching PSM* extension meta-model does not have any relation.

6.3.1.2.5 *AnnPSMComponentProperty* Element

Overview

The *AnnPSMComponentProperty* element models an existent *PSMComponentProperty* of the Core PSM model that can be annotated by the *Database Searching* PSM extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnPSMComponentProperty* element of the *Database Searching* PSM extension meta-model does not have any property.

Relations

Table 6-12 AnnPSMComponentProperty's Relations

Relation With PSM Element	Type	Multiplicity	Structural Constraints
PSMComponentProperty	reference	1	The <i>AnnPSMComponentProperty</i> must reference exactly one <i>PSMComponentProperty</i> element of the Core PSM model.

Behavioural Restrictions

The *AnnPSMComponentProperty* element of the *Database Searching* PSM extension meta-model does not have any behavioural restriction.

6.3.1.2.6 *AnnJavaResourceModel* Element

Overview

The *AnnJavaResourceModel* element models an existent *JavaResourceModel* of the Core PSM model that can be annotated by the *Database Searching* PSM extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnJavaResourceModel* element of the *Database Searching* PSM extension meta-model does not have any property.

Relations

Table 6-13 AnnJavaResourceModel's Relations

Relation With PSM Element	Type	Multiplicity	Structural Constraints
AnnJavaReourceModel	reference	1	The <i>AnnJavaResourceModel</i> must reference exactly one <i>JavaResourceModel</i> element of the Core PSM model.

Behavioural Restrictions

The *AnnJavaResourceModel* element of the *Database Searching PSM* extension meta-model does not have any behavioural restriction.

6.3.1.2.7 AnnHTTPActivityHandler Element

Overview

The *AnnHTTPActivityHandler* element models an existent *HTTPActivityHandler* of the Core PSM model that can be annotated by the *Database Searching PSM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnHTTPActivityHandler* element of the *Database Searching PSM* extension meta-model does not have any property.

Relations

Table 6-14 AnnHTTPActivityHandler's Relations

Relation With PSM Element	Type	Multiplicity	Structural Constraints
HTTPActivityHandler	reference	1	The <i>AnnHTTPActivityHandler</i> must reference exactly one <i>HTTPActivityHandler</i> element of the Core PSM model.

Behavioural Restrictions

The *AnnHTTPActivityHandler* element of the *Database Searching PSM* extension meta-model does not have any behavioural restriction.

6.3.1.2.8 AnnHTTPActivity Element

Overview

The *AnnHTTPActivity* element models an existent *HTTPActivity* of the Core PSM model that can be annotated by the *Database Searching PSM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnHTTPActivity* element of the *Database Searching PSM* extension meta-model does not have any property.

Relations

Table 6-15 AnnHTTPActivity's Relations

Relation With PSM Element	Type	Multiplicity	Structural Constraints
HTTPActivity	reference	1	The <i>AnnHTTPActivity</i> must reference exactly one <i>HTTPActivity</i> element of the Core PSM model.

Behavioural Restrictions

The *AnnHTTPActivity* element of the *Database Searching PSM* extension meta-model does not have any behavioural restriction.

6.3.1.2.9 AnnJavaAlgoController Element

Overview

The *AnnJavaAlgoController* element models an existent *Property* of the Core PSM model that can be annotated by the *Database Controller PSM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnJavaAlgoController* element of the *Database Searching PSM* extension meta-model does not have any property.

Relations

Table 6-16 AnnJavaAlgoController's Relations

Relation With PSM Element	Type	Multiplicity	Structural Constraints
JavaAlgoController	reference	1	The <i>AnnJavaAlgoController</i> must reference exactly one <i>JavaAlgoController</i> element of the Core PSM model.

Behavioural Restrictions

The *AnnJavaAlgoController* element of the *Database Searching PSM* extension meta-model does not have any behavioural restriction.

6.3.1.2.10 SearchController Element

Overview

The *SearchController* element models an annotation of the *Database Searching PSM* extension meta-model that is intended to annotate an existent Core PSM *JavaAlgoController*. With this annotation, the *JavaAlgoController* will embed the needed functionality to handle incoming requests for local

envisioned system's database searching. Figure 6-5 demonstrates the *SearchController* element of the *Database Searching PSM* extension meta-model and its relations.

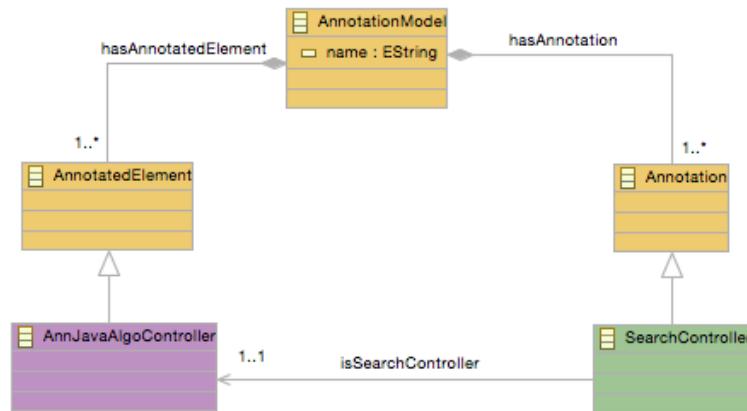


Figure 6-5 SearchController annotation and its relations

Properties

The *SearchController* element of the *Database Searching PSM* extension meta-model does not have any property.

Relations

Table 6-17 SearchController's Relations

Relation With PSM Extension Element	Type	Multiplicity	Structural Constraints
AnnJavaAlgoController	association	1	The <i>SearchController</i> must be associated with exactly one <i>AnnJavaAlgoController</i> , which references one <i>JavaAlgoController</i> element of the Core PSM model. With this annotation, the <i>JavaAlgoController</i> will embed the needed functionality to handle incoming database searching requests.

Behavioural Restrictions

The *SearchController* element of the *Database Searching PSM* extension meta-model does not have any behavioural restriction.

6.3.1.2.11 SearchHTTPActivity Element

Overview

The *SearchHTTPActivity* element models an annotation of the *Database Searching PSM* extension meta-model that is intended to annotate an existent Core PSM *HTTPActivity*. With this annotation, the *HTTPActivity* will embed the needed code to accept and forward incoming database searching

requests from envisioned system's clients. Figure 6-6 demonstrates the *SearchHTTPActivity* element of the *Database Searching PSM* extension meta-model and its relations.

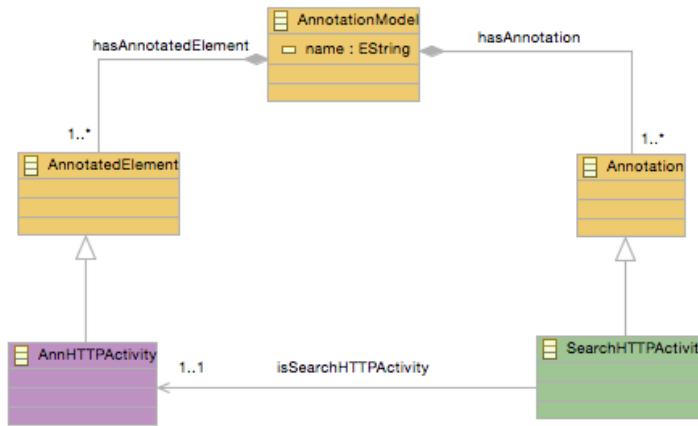


Figure 6-6 SearchHTTPActivity

Properties

The *SearchHTTPActivity* element of the *Database Searching PSM* extension meta-model does not have any property.

Relations

Table 6-18 SearchHTTPActivity's Relations

Relation With PSM Extension Element	Type	Multiplicity	Structural Constraints
AnnHTTPActivity	association	1	The <i>SearchHTTPActivity</i> must be associated with exactly one <i>AnnHTTPActivity</i> , which references one <i>HTTPActivity</i> of the Core PSM model. With this annotation, the <i>HTTPActivity</i> will embed the needed functionality to accept and forward incoming database searching requests.

Behavioural Restrictions

The *SearchHTTPActivity* element of the *Database Searching PSM* extension meta-model does not have any behavioural restriction.

6.3.1.2.12 SearchHTTPHandler Element

Overview

The *SearchHTTPHandler* element models an annotation of the *Database Searching PSM* extension meta-model that is intended to annotate an existent Core PSM *HTTPHandler*. With this annotation, the *HTTPHandler* will embed the needed functionality to perform the needed queries to the local

envisioned system's database in order to be able to respond to the client's search request. Figure 6-7 demonstrates the *SearchHTTPHandler* element of the *Database Searching PSM* extension meta-model and its relations.

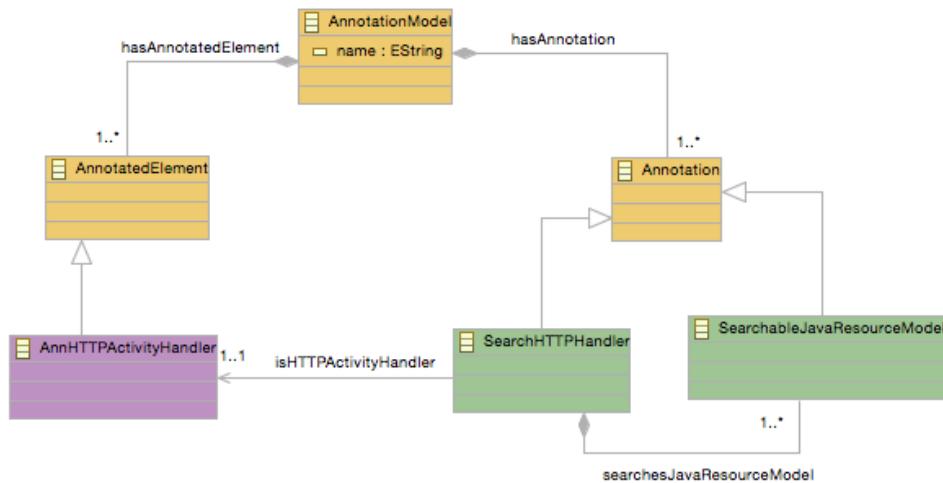


Figure 6-7 SearchHTTPHandler annotation and its relations.

Properties

The *SearchHTTPHandler* element of the *Database Searching PSM* extension meta-model does not have any property.

Relations

Table 6-19 SearchHTTPHandler's Relations

Relation With PSM Extension Element	Type	Multiplicity	Structural Constraints
SearchableJavaResourceModel	composition	1..*	The <i>SearchHTTPHandler</i> must have at least one composition association with a <i>SearchableJavaResourceModel</i> . This association models the fact that every <i>SearchHTTPHandler</i> should be have searching capabilities of at least one <i>JavaResourceModel</i> of the envisioned system in order to be meaningful.
AnnHTTPHandler	association	1	The <i>SearchHTTPHandler</i> must have exactly one association with an <i>AnnHTTPHandler</i> , which references one <i>HTTPHandler</i> of the Core PSM model. With this annotation, the <i>HTTPHandler</i> will embed the needed functionality to query the underlying envisioned system's local database in order to be able to answer client's search requests.

Behavioural Restrictions

The *SearchHTTPHandler* element of the *Database Searching PSM* extension meta-model does not have any behavioural restriction.

6.3.1.2.13 SearchableJavaResourceModel Element

Overview

The *SearchableJavaResourceModel* element models an annotation of the *Database Searching PSM* extension meta-model that is intended to annotate an existent Core PSM *JavaResourceModel*. With this annotation, the *JavaResourceModel* will be appropriately indexed with *Lucene* indexes so as to be searchable. Figure 6-8 demonstrates the *SearchableJavaResourceModel* element of the *Database Searching PSM* extension meta-model and its relations.

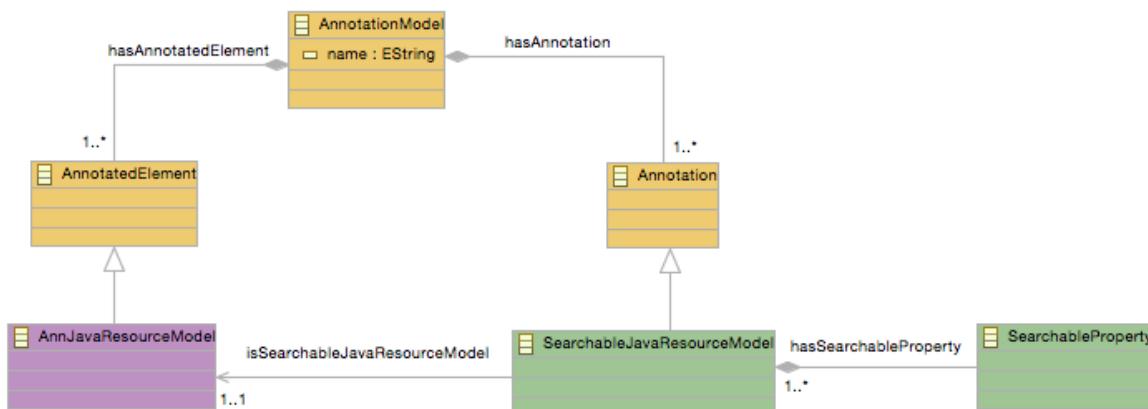


Figure 6-8 SearchableJavaResourceModel annotation and its relations.

Properties

The *SearchableJavaResourceModel* element of the *Database Searching PSM* extension meta-model does not have any property.

Relations

Table 6-20 SearchableJavaResourceModel's Relations

Relation With PSM Extension Element	Type	Multiplicity	Structural Constraints
SearchableProperty	composition	1..*	The <i>SearchableJavaResourceModel</i> must have at least one composition association with a <i>SearchableProperty</i> . This association models the fact that a <i>SearchableJavaResourceModel</i> must have at least one <i>SearchableProperty</i> in order to be meaningful.
AnnJavaResourceModel	association	1	The <i>SearchableJavaResourceModel</i> must be associated with exactly one <i>AnnJavaResourceModel</i> , which references one <i>JavaResourceModel</i> element of the Core PSM model. With this annotation, the

			<i>JavaResourceModel</i> will embed the needed indexing <i>Lucene</i> annotations in order to be searchable.
--	--	--	--

Behavioural Restrictions

The *SearchableJavaResourceModel* element of the *Database Searching PSM* extension meta-model does not have any behavioural restriction.

6.3.1.2.14 *SearchableProperty* Element

Overview

The *SearchableProperty* element models an annotation of the *Database Searching PSM* extension meta-model that is intended to annotate an existent Core PSM *PSMComponentProperty*. With this annotation, the *PSMComponentProperty* will be annotated with the appropriate *Lucene* annotations in order to be indexed. Figure 6-9 demonstrates the *SearchableProperty* element of the *Database Searching PSM* extension meta-model and its relations.

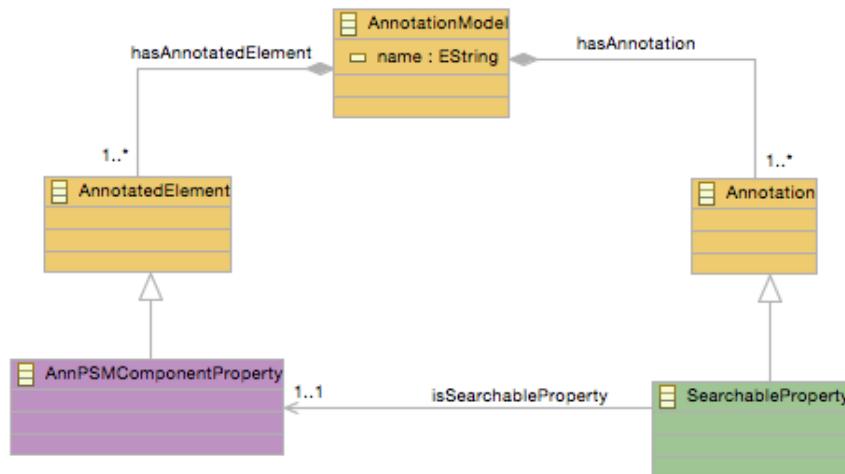


Figure 6-9 SearchableProperty

Properties

The *SearchableProperty* element of the *Database Searching PSM* extension meta-model does not have any property.

Relations

Table 6-21 SearchableProperty's Relations

Relation With PSM Extension Element	Type	Multiplicity	Structural Constraints
AnnPSMComponentProperty	association	1	The <i>SearchableProperty</i> must be associated with exactly one <i>AnnPSMComponentProperty</i> , which references one <i>PSMComponentProperty</i>

			element of the Core PSM model. With this annotation, the <i>PSMComponentProperty</i> will be annotated with the appropriate <i>Lucene</i> annotation in order to be indexed and therefore searchable.
--	--	--	---

Behavioural Restrictions

The *SearchableProperty* element of the *Database Searching PSM* extension meta-model does not have any behavioural restriction.

6.4 PSM External Service Composition Ecore Meta-model Definition

6.4.1.1 Introduction

In order to fully define the *External Service Composition* Ecore PSM extension meta-model (Figure 6-10), its various aspects must be explained, demonstrated as well as documented. This is done in the following subsections following the format section 4.1.2.1 describes.

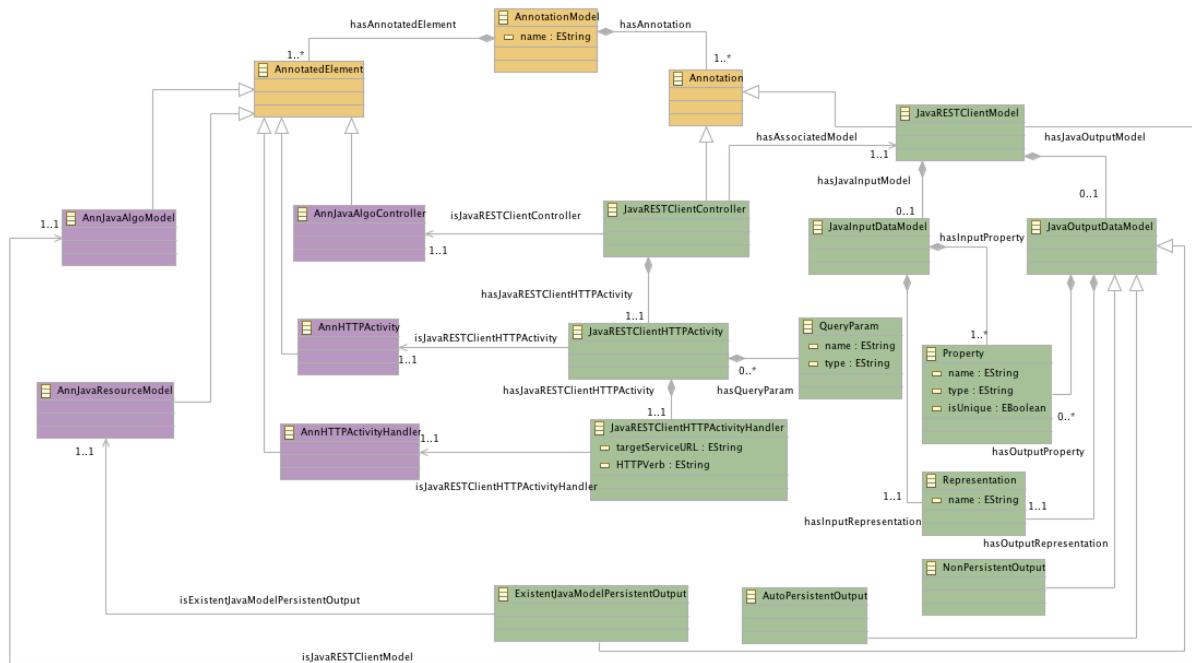


Figure 6-10 External Service Composition PSM extension meta-model

6.4.1.2 PSM External Service Composition Ecore Meta-model Elements

6.4.1.2.1 AnnotationModel Element

Overview

The *AnnotationModel* element is the root element of the *External Service Composition* PSM extension meta-model and comprises of all the introduced *Annotations* and the existing Core PSM elements to which these annotations are attached in order to alter their semantic meaning, design and/or configuration.

Properties

Table 6-22 AnnotationModel's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the name of the <i>Annotation</i> model.

Relations

Table 6-23 AnnotationModel's Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
AnnotatedElement	composition	1..*	This composition association models the fact that every <i>AnnotationModel</i> element must have at least one <i>AnnotatedElement</i> .
Annotation	composition	1..*	This composition association models the fact that every <i>AnnotationModel</i> element must have at least one <i>Annotation</i> .

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of an *AnnotationModel* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the *External Service Composition* PSM extension meta-model that can be found in appendix A.1.

- *atLeastOneAnnJavaAlgoControllerExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnJavaAlgoController*.
- *atLeastOneAnnHTTPActivityExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that are of type *AnnHTTPActivity*.
- *atLeastOneHTTPActivityHandlerExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnHTTPActivityHandler*.
- *sameNumberOfAnnJavaAlgoControllersAndAnnJavaAlgoModelsExist*: This OCL constraint checks whether the total number of the *AnnotatedElements* that are associated with this

AnnotationModel and are of type *AnnJavaAlgoModels*, is the same with the number of the *AnnJavaAlgoControllers*.

- *atLeastOneJavaRESTClientControllerExists*: This OCL constraint checks whether there exists at least one *Annotation* element that is associated with this *AnnotationModel* that is of type *JavaRESTClientController*.

6.4.1.2.2 AnnotatedElement Element

Overview

The *AnnotatedElement* element models any existent Core PSM meta-model element that can be annotated by the *External Service Composition* Core PSM extension in order to extend its semantic meaning and/or design. It is always further specialized to more specific categories of annotated elements.

Properties

The *AnnotatedElement* element of the *External Service Composition PSM* extension meta-model does not have any property.

Relations

The *AnnotatedElement* element of the *External Service Composition PSM* extension meta-model does not have any relation.

Behavioural Restrictions

The *AnnotatedElement* element of the *External Service Composition PSM* extension meta-model does not have any behavioural restriction.

6.4.1.2.3 Annotation Element

Overview

The *Annotation* element models any new semantic meaning or design that is attached to a Core PSM meta-model element. It is always further specialized to more specific categories of annotations.

Properties

The *Annotation* element of the *External Service Composition PSM* extension meta-model does not have any property.

Relations

The *Annotation* element of the *External Service Composition PSM* extension meta-model does not have any relation.

Behavioural Restrictions

The *Annotation* element of the *External Service Composition PSM* extension meta-model does not have any behavioural restriction.

6.4.1.2.4 AnnJavaAlgoModel Element

Overview

The *AnnJavaAlgoModel* element models an existent *JavaAlgoModel* of the Core PSM model that can be annotated by the *External Service Composition PSM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnJavaAlgoModel* element of the *External Service Composition PSM* extension meta-model does not have any property

Relations

Table 6-24 AnnJavaAlgoModel's Relations

Relation With PSM Element	Type	Multiplicity	Structural Constraints
JavaAlgoModel	reference	1	The <i>AnnJavaAlgoModel</i> must reference exactly one <i>JavaAlgoModel</i> element of the Core PSM model.

Behavioural Restrictions

The *AnnJavaAlgoModel* element of the *External Service Composition PSM* extension meta-model does not have any behavioural restriction.

6.4.1.2.5 AnnJavaResourceModel Element

Overview

The *AnnJavaResourceModel* element models an existent *JavaResourceModel* of the Core PSM model that can be annotated by the *External Service Composition PSM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnJavaResourceModel* element of the *External Service Composition PSM* extension meta-model does not have any property.

Relations

Table 6-25 AnnJavaResourceModel's Relations

Relation With PSM Element	Type	Multiplicity	Structural Constraints
JavaResourceModel	reference	1	The <i>AnnJavaResourceModel</i> must reference exactly one <i>JavaResourceModel</i> element of the Core PSM model.

Behavioural Restrictions

The *AnnJavaResourceModel* element of the *External Service Composition PSM* extension meta-model does not have any behavioural restriction.

6.4.1.2.6 AnnHTTPActivityHandler Element

Overview

The *AnnHTTPActivityHandler* element models an existent *HTTPActivityHandler* of the Core PSM model that can be annotated by the *External Service Composition PSM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnHTTPActivityHandler* element of the *External Service Composition PSM* extension meta-model does not have any property.

Relations

Table 6-26 AnnHTTPActivityHandler's Relations

Relation With PSM Element	Type	Multiplicity	Structural Constraints
HTTPActivityHandler	reference	1	The <i>AnnHTTPActivityHandler</i> must reference exactly one <i>HTTPActivityHandler</i> element of the Core PSM model.

Behavioural Restrictions

The *AnnHTTPActivityHandler* element of the *External Service Composition PSM* extension meta-model does not have any behavioural restriction.

6.4.1.2.7 AnnHTTPActivity Element

Overview

The *AnnHTTPActivity* element models an existent *HTTPActivity* of the Core PSM model that can be annotated by the *External Service Composition PSM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnHTTPActivity* element of the *External Service Composition PSM* extension meta-model does not have any property

Relations

Table 6-27 AnnHTTPActivity's Relations

Relation With PSM Element	Type	Multiplicity	Structural Constraints
HTTPActivity	reference	1	The <i>AnnHTTPActivity</i> must reference exactly one <i>HTTPActivity</i> element of the Core PSM model.

Behavioural Restrictions

The *AnnHTTPActivity* element of the *External Service Composition PSM* extension meta-model does not have any behavioural restriction.

6.4.1.2.8 AnnJavaAlgoController ElementOverview

The *AnnJavaAlgoController* element models an existent *JavaAlgoController* of the Core PSM model that can be annotated by the *External Service Composition PSM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnJavaAlgoController* element of the *External Service Composition PSM* extension meta-model does not have any property.

Relations**Table 6-28 AnnJavaAlgoController's Relations**

Relation With PSM Element	Type	Multiplicity	Structural Constraints
JavaAlgoController	reference	1	The <i>AnnJavaAlgoController</i> must reference exactly one <i>JavaAlgoController</i> element of the Core PSM model.

Behavioural Restrictions

The *AnnJavaAlgoController* element of the *External Service Composition PSM* extension meta-model does not have any behavioural restriction.

6.4.1.2.9 JavaRESTClientController ElementOverview

The *JavaRESTClientController* element models an annotation of the *External Service Composition PSM* extension meta-model that is intended to annotate an existent Core PSM *JavaAlgoController*. With this annotation, the *JavaAlgoController* will embed the needed functionality to accept incoming envisioned system clients' requests that will be forwarded for handling to an external service composition. Figure 6-11 demonstrates the *JavaRESTClientController* element of the *External Service Composition PSM* extension meta-model and its relations.

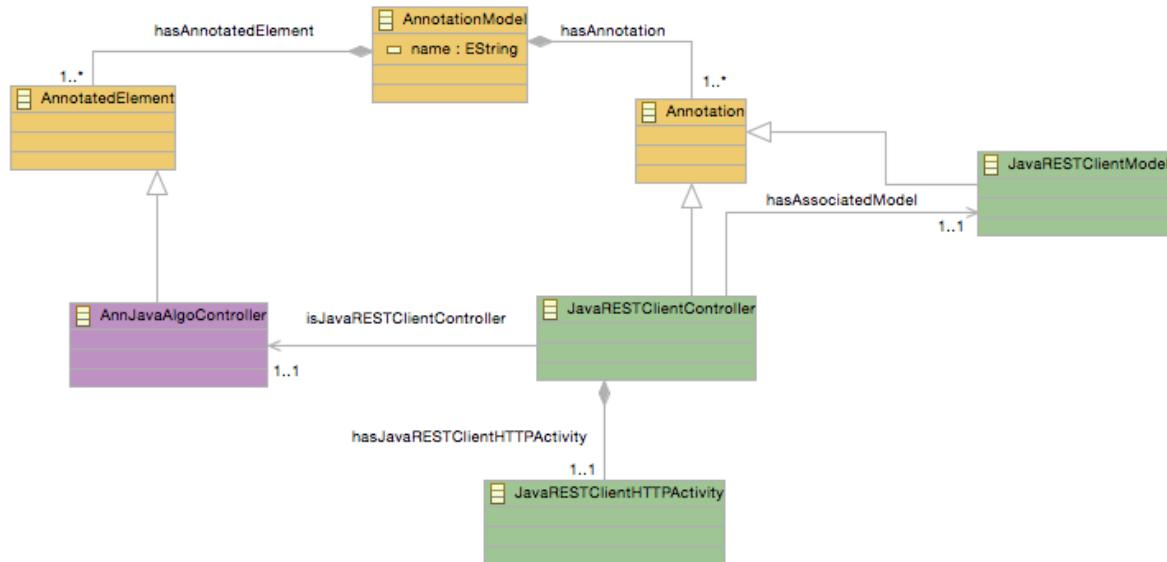


Figure 6-11 JavaRESTClientController annotation and its relations

Properties

The *JavaRESTClientController* element of the *External Service Composition PSM* extension meta-model does not have any property.

Relations

Table 6-29 JavaRESTClientController's Relations

Relation With PSM Extension Element	Type	Multiplicity	Structural Constraints
AnnJavaAlgoController	association	1	The <i>JavaRESTClientController</i> must be associated with exactly one <i>AnnJavaAlgoController</i> , which references a <i>JavaAlgoController</i> element of the Core PSM model. With this annotation, the <i>JavaAlgoController</i> will embed the needed functionality to accept client requests that will be forwarded to an external service composition for handling.
JavaRESTClientHTTPActivity	composition	1	The <i>JavaRESTClientController</i> must have a composition association with exactly one <i>JavaRESTClientHTTPActivity</i> . This association models the fact that every <i>JavaRESTClientController</i> has its API exposed by the means of a specialized <i>JavaRESTClientHTTPActivity</i> .

JavaRESTClientModel	association	1	The <i>JavaRESTClientController</i> must have an association with exactly one <i>JavaRESTClientModel</i> . This association models the fact that every <i>JavaRESTClientController</i> uses a specific input/output data format, which is described by the <i>JavaRESTClientModel</i> , in order to be able to communicate with external service compositions.
---------------------	-------------	---	--

Behavioural Restrictions

The *JavaRESTClientController* element of the *External Service Composition PSM* extension meta-model does not have any behavioural restriction.

6.4.1.2.10 JavaRESTClientHTTPActivity Element

Overview

The *JavaRESTClientHTTPActivity* element models an annotation of the *External Service Composition PSM* extension meta-model that is intended to annotate an existent Core PSM *HTTPActivity*. With this annotation, the *HTTPActivity* will embed the needed web API in order to be able to accept client requests that will be forwarded and handled by an external service composition. Figure 6-12 demonstrates the *JavaRESTClientHTTPActivity* element of the *External Service Composition PSM* extension meta-model and its relations.

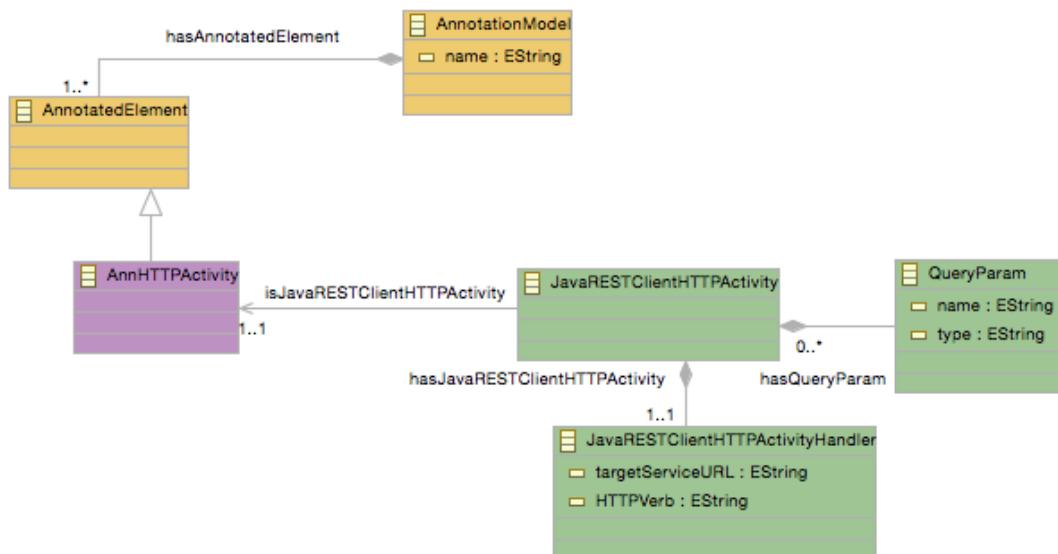


Figure 6-12 JavaRESTClientHTTPActivity element and its relations

Properties

The *JavaRESTClientHTTPActivity* element of the *External Service Composition PSM* extension meta-model does not have any property.

Relations

Table 6-30 JavaRESTClientHTTPActivity's Relations

Relation With PSM Extension Element	Type	Multiplicity	Structural Constraints
AnnHTTPActivity	association	1	The <i>JavaRESTClientHTTPActivity</i> must have exactly one association with an <i>AnnHTTPActivity</i> , which references one <i>HTTPActivity</i> element of the Core PSM model.
JavaRESTClientHTTPActivityHandler	composition	1	The <i>JavaRESTClientHTTPActivity</i> must have exactly one composition association with a <i>JavaRESTClientHTTPActivityHandler</i> . This association models the fact that every <i>JavaRESTClientHTTPActivity</i> delegates any needed interaction with an external service composition, which will handle the client's request, to one <i>JavaRESTClientHTTPActivityHandler</i> element.
QueryParam	composition	0..*	The <i>JavaRESTClientHTTPActivity</i> can have zero or more composition associations with <i>QueryParams</i> . This association models the fact that a <i>JavaRESTClientHTTPActivity</i> might need to forward a set of query parameters to the <i>JavaRESTClientHTTPActivityHandler</i> so as to be able to interact with the external web service composition.

Behavioural Restrictions

The *JavaRESTClientHTTPActivity* element of the *External Service Composition PSM* extension meta-model does not have any behavioural restriction.

6.4.1.2.11 *QueryParam* Element

Overview

The *QueryParam* element models a query parameter that might have to be used by the envisioned system when interacting with the external service composition.

Properties

Table 6-31 QueryParam's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the name of the <i>QueryParam</i> .

type	EString	1	This is the type of the <i>QueryParam</i> .
------	---------	---	---

Relations

The *QueryParam* element of the *External Service Composition PSM* extension meta-model does not have any relation.

Behavioural Restrictions

The *QueryParam* element of the *External Service Composition PSM* extension meta-model does not have any behavioural restriction.

6.4.1.2.12 JavaRESTClientHTTPActivityHandler Element

Overview

The *JavaRESTClientHTTPActivityHandler* element models an annotation of the *External Service Composition PSM* extension meta-model that is intended to annotate an existent Core PSM *HTTPActivityHandler*. With this annotation, the *HTTPActivityHandler* will embed the needed functionality to interact with an external service composition, to which an envisioned system client's request is forwarded and repackaging the response in an appropriate format for the client. Figure 6-13 demonstrates the *JavaRESTClientHTTPActivityHandler* element of the *External Service Composition PSM* extension meta-model and its relations.

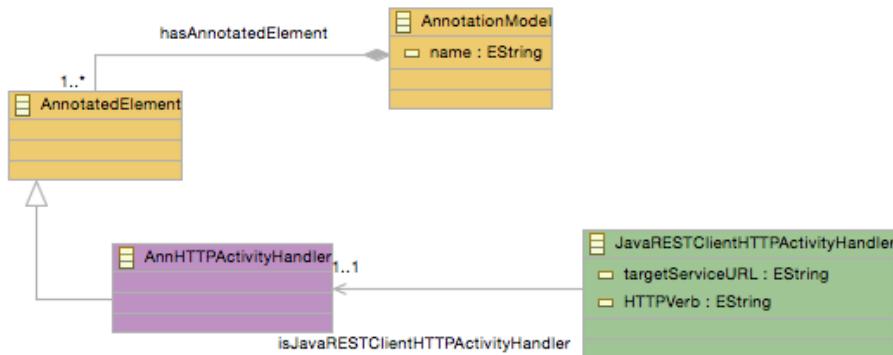


Figure 6-13 JavaRESTClientHTTPActivityHandler

Properties

Table 6-32 JavaRESTClientHTTPActivityHandler's Properties

Name	Type	Multiplicity	Explanation
targetServiceURL	EString	1	This is the <i>URL</i> that the envisioned system will use to reach the external service composition's functionality
HTTPVerb	EString	1	This is the <i>HTTP</i> verb that the envisioned system will use when it sends a request to the target service <i>URL</i> .

Relations

Table 6-33 JavaRESTClientHTTPActivityHandler's Relations

Relation With PSM Extension Element	Type	Multiplicity	Structural Constraints
AnnHTTPActivityHandler	association	1	The <i>JavaRESTClientHTTPActivityHandler</i> must have an association with exactly one <i>AnnHTTPActivityHandler</i> , which references an <i>HTTPActivityHandler</i> element of the Core PSM model.

Behavioural Restrictions

The *JavaRESTClientHTTPActivityHandler* element of the *External Service Composition PSM* extension meta-model does not have any behavioural restriction.

6.4.1.2.13 JavaRESTClientModel Element

Overview

The *JavaRESTClientModel* element models an annotation of the *External Service Composition PSM* extension meta-model that is intended to annotate an existent Core PSM *JavaAlgoModel*. With this annotation, the *JavaAlgoModel* will embed the needed code to model the input and output data format, which is needed when interacting with the external service composition. Figure 6-14 demonstrates the *JavaRESTClientModel* element of the *External Service Composition PSM* extension meta-model and its relations.

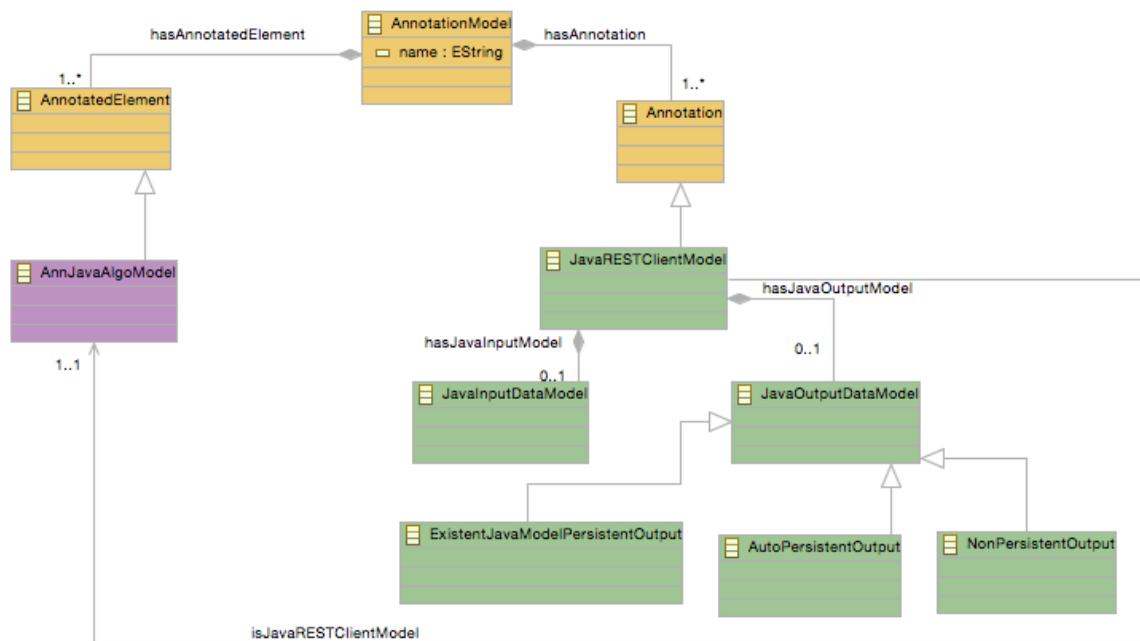


Figure 6-14 JavaRESTClientModel annotation and its relations.

Properties

The *JavaRESTClientModel* element of the *External Service Composition PSM* extension meta-model does not have any property.

Relations

Table 6-34 JavaRESTClientModel's Relations

Relation With PSM Extension Element	Type	Multiplicity	Structural Constraints
AnnJavaAlgoModel	association	1	The <i>JavaRESTClientModel</i> must have an association with exactly one <i>AnnJavaAlgoModel</i> , which references a <i>JavaAlgoModel</i> of the Core PSM model. With this annotation, the <i>JavaAlgoModel</i> will embed the needed code to model the input/output data format, which is needed to interact with external service composition.
JavaInputDataModel	composition	0..1	The <i>JavaRESTClientModel</i> can have either zero or one composition association with a <i>JavaInputDataModel</i> element. This association models the fact that a <i>JavaRESTClientModel</i> may embed an input data model to be used when sending requests to an external service composition. If there is no <i>JavaInputDataModel</i> , then the external service composition does not require any input.
JavaOutputDataModel	composition	0..1	The <i>JavaRESTClientModel</i> can have either zero or one composition association with a <i>JavaOutputDataModel</i> element. This association models the fact that a <i>JavaRESTClientModel</i> may embed an output data model to be used when receiving responses from the external service composition. If there is no <i>JavaOutputDataModel</i> , then the external service composition does not produce any output when it is invoked.

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of a *JavaRESTClientModel* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the *External Service Composition PIM* extension meta-model that can be found in appendix A.1.

- *hasSpecializedOutputDataModel*: This OCL constraint verifies that whenever a *JavaRESTClientModel* does have a *JavaOutputDataModel*, it is always specialized as either *NonPersistentOutput*, *AutoPersistentOutput* or *ExistentialJavaModelPersistentOutput*.

6.4.1.2.14 JavaInputDataModel Element

Overview

The *JavaInputDataModel* element models the input data that is required by an external service composition when interacting with it. Figure 6-15 demonstrates the *JavaInputDataModel* element of the *External Service Composition PSM* extension meta-model and its relations.

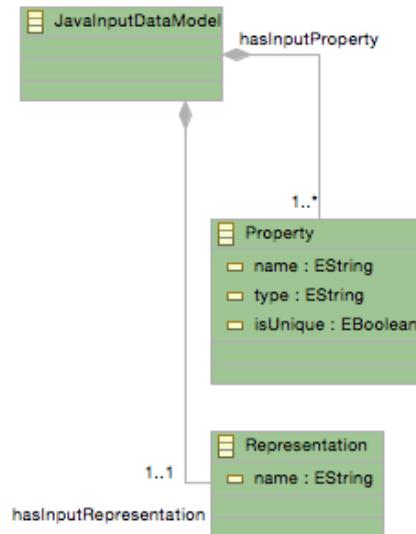


Figure 6-15 JavaInputDataModel element and its relations.

Properties

The *JavaInputDataModel* element of the *External Service Composition PSM* extension meta-model does not have any property.

Relations

Table 6-35 JavaInputDataModel's Relations

Relation With PSM Extension Element	Type	Multiplicity	Structural Constraints
Property	composition	1..*	The <i>JavaInputDataModel</i> must have a composition association with at least one <i>Property</i> . This association models the fact that any <i>JavaInputDataModel</i> must include at least one property in order to be meaningful.
Representation	composition	1	The <i>JavaInputDataModel</i> must have exactly one composition association with a <i>Representation</i> . This association models the fact that every <i>JavaInputDataModel</i> must be formatted in a specific way (e.g. application/JSON) so as it will be consumable by the external service composition.

Behavioural Restrictions

The *JavaOutputDataModel* element of the *External Service Composition PSM* extension meta-model does not have any behavioural restriction.

6.4.1.2.15 JavaOutputDataModel Element

Overview

The *JavaOutputDataModel* element models the output data format of an external service composition. It is always specialized as *NonpersistentOutput*, *AutopersistentOutput* or *ExistentalJavaModelPersistentOutput*. Figure 6-16 demonstrates the *JavaOutputDataModel* element of the *External Service Composition PSM* extension meta-model and its relations.

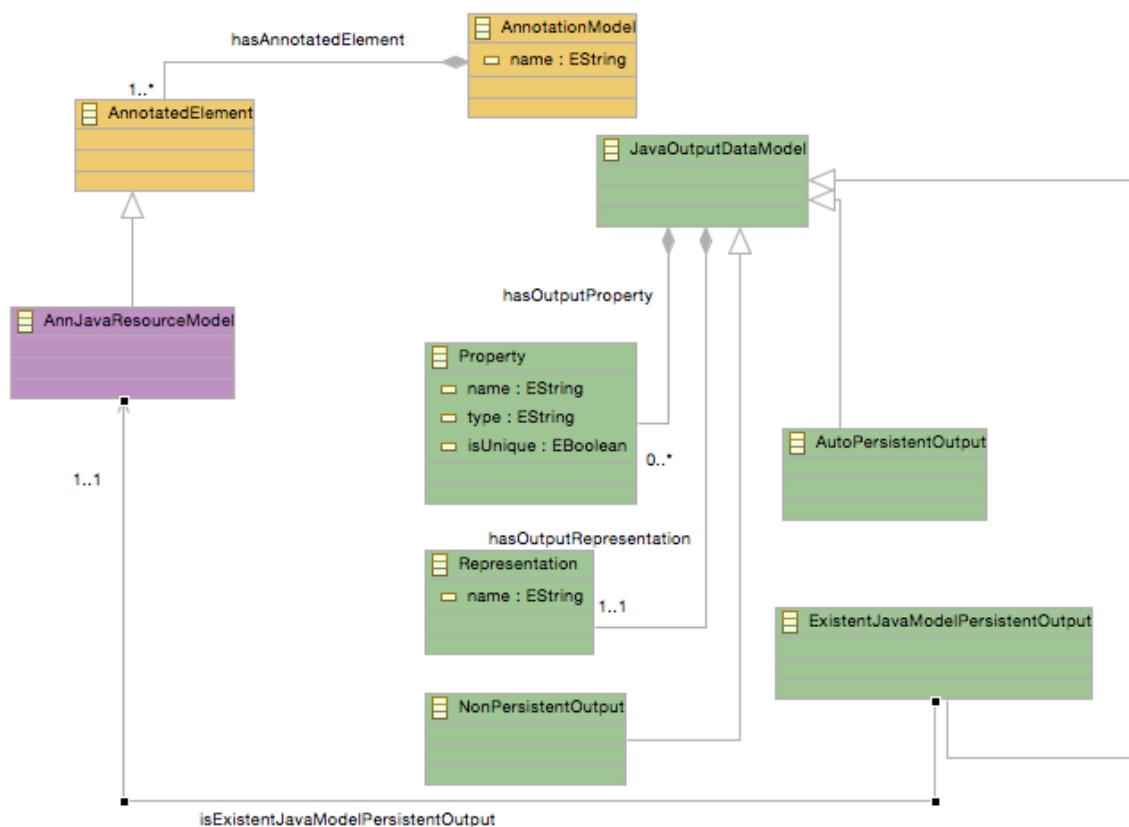


Figure 6-16 JavaOutputDataModel element and its relations.

Properties

The *JavaOutputDataModel* element of the *External Service Composition PSM* extension meta-model does not have any property.

Relations

Table 6-36 JavaOutputDataModel's Relations

Relation With PSM Extension Element	Type	Multiplicity	Structural Constraints
Property	composition	0..*	The <i>JavaOutputDataModel</i> must have at least one composition association with a <i>Property</i> . This association models the fact that every <i>JavaOutputDataModel</i> may have some properties. If a <i>JavaOutputDataModel</i> does not have any properties, then the response of the external service composition does not contain any data other than an HTTP status code.
Representation	composition	1	The <i>JavaOutputDataModel</i> must have exactly one composition association with a <i>Representation</i> . This association models the fact that every <i>JavaOutputDataModel</i> must expect responses of the external service composition that are formatted in a specific way, e.g. application/JSON.

Behavioural Restrictions

The *JavaOutputDataModel* element of the *External Service Composition PSM* extension meta-model does not have any behavioural restrictions.

6.4.1.2.16 Property Element

Overview

The *Property* element models a property of either a *JavaInputDataModel* or *JavaOutputDataModel*.

Properties

Table 6-37 Property's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the name of the <i>Property</i> .
type	EString	1	This is the type of the <i>Property</i> .
isUnique	EBoolean	1	This is the multiplicity of the <i>Property</i> . If it is set to <i>true</i> , then the <i>Property</i> has multiplicity one, otherwise it is a multivalued array.

Relations

The *Property* element of the *External Service Composition PSM* extension meta-model does not have any relation.

Behavioural Restrictions

The *Property* element of the *External Service Composition PSM* extension meta-model does not have any behavioural restriction.

6.4.1.2.17 Representation Element

Overview

The *Representation* element models the media type that should be used when packaging *JavaInputDataModels* or *JavaOutputDataModels* in order to interact with an external service composition.

Properties

Table 6-38 Representation's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the media type that should be used when packaging a <i>JavaInputDataModel</i> or a <i>JavaOutputDataModel</i> in order to successfully interact with an external service composition. Some representation examples are <i>application/JSON</i> and/or <i>application/XML</i> .

Relations

The *Representation* element of the *External Service Composition PSM* extension meta-model does not have any relation.

Behavioural Restrictions

The *Representation* element of the *External Service Composition PSM* extension meta-model does not have any behavioural restriction.

6.4.1.2.18 NonPersistentOutput Element

Overview

The *NonPersistentOutput* element models a *JavaOutputDataModel* specialization. With this specialization, data that is received from the external service composition data will be returned to the envisioned system's client without prior storing it to the local database.

Properties

The *NonPersistentOutput* element of the *External Service Composition PSM* extension meta-model does not have any property other than those inherited from the *JavaOutputDataModel*.

Relations

The *NonPersistentOutput* element of the *External Service Composition PSM* extension meta-model does not have any relation other than those inherited from the *JavaOutputDataModel*.

Behavioural Restrictions

The *NonPersistentOutput* element of the *External Service Composition PSM* extension meta-model does not have any behavioural restriction.

6.4.1.2.19 AutoPersistentOutput Element

Overview

The *AutoPersistentOutput* element models a *JavaOutputDataModel* specialization. With this specialization, data received from the external service composition will be stored in the envisioned system's local database prior sending it back to the client, using a non-user defined data model.

Properties

The *AutoPersistentOutput* element of the *External Service Composition PSM* extension meta-model does not have any property other than those inherited from the *JavaOutputDataModel*.

Relations

The *AutoPersistentOutput* element of the *External Service Composition PSM* extension meta-model does not have any relation other than those inherited from the *JavaOutputDataModel*.

Behavioural Restrictions

The *AutoPersistentOutput* element of the *External Service Composition PSM* extension meta-model does not have any behavioural restriction.

6.4.1.2.20 ExistentalJavaModelPersistentOutput Element

Overview

The *ExistentalJavaModelPersistentOutput* element models a *JavaOutputDataModel* specialization. With this specialization, data received from the external service composition will be stored in the envisioned system's local database prior sending it back to client, using a user defined *Resource* of the Core CIM. Figure 6-17 demonstrates the *ExistentalJavaModelPersistentOutput* element of the *External Service Composition PSM* extension meta-model and its relations.

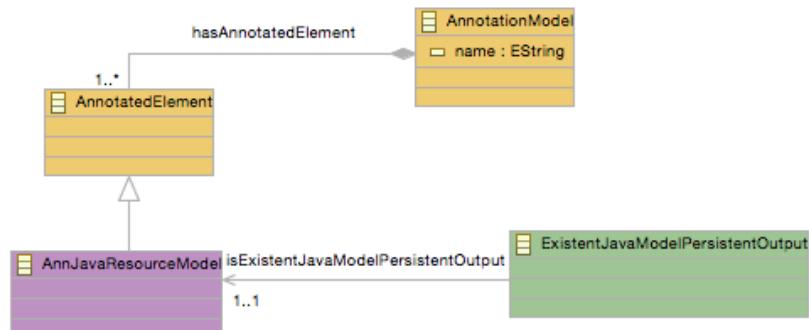


Figure 6-17 ExistentalJavaModelPersistentOutput annotation and its relations.

Properties

The *ExistenJavaModelPersistentOutput* element of the *External Service Composition PSM* extension meta-model does not have any property other than those inherited from the *JavaOutputDataModel*.

Relations

Table 6-39 ExistenJavaModelPersistentOutput's Relations

Relation With PSM Extension Element	Type	Multiplicity	Structural Constraints
AnnJavaResourceModel	association	1	The <i>ExistenJavaModelPersistentOutput</i> must have exactly one association with an <i>AnnJavaResourceModel</i> , which references a <i>JavaResourceModel</i> element of the Core PSM model. With this annotation, the <i>JavaResourceModel</i> will be used as an output data model for the data received from the external service composition, in order to be able to store it in the envisioned system's local database.

Behavioural Restrictions

The *ExistenJavaModelPersistentOutput* element of the *External Service Composition PSM* extension meta-model does not have any behavioural restriction.

6.5 PSM ABAC Authorization Ecore Meta-model Definition

6.5.1.1 Introduction

In order to fully define the *ABAC Authorization* Ecore PSM extension meta-model, its various aspects must be explained, demonstrated as well as documented. This is done in the following subsections following the format section 4.1.2.1 describes. Due to space limitations, the full *ABAC Authorization* meta-model can be found using the following link <https://github.com/s-case/mde> .

6.5.1.2 PSM ABAC Authorization Ecore Meta-model Elements

6.5.1.2.1 *AnnotationModel* Element

Overview

The *AnnotationModel* element is the root element of the *ABAC Authorization* PSM extension meta-model and comprises of all the introduced *Annotations* and the existing Core PSM elements to which these annotations are attached in order to alter their semantic meaning, design and/or configuration.

Properties

Table 6-40 *AnnotationModel*'s Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the name of the <i>Annotation</i> model.

Relations

Table 6-41 *AnnotationModel*'s Relations

Relation With CIM Extension Element	Type	Multiplicity	Structural Constraints
AnnotatedElement	composition	1.. *	This composition association models the fact that every <i>AnnotationModel</i> element must have at least one <i>AnnotatedElement</i> .
Annotation	composition	1.. *	This composition association models the fact that every <i>AnnotationModel</i> element must have at least one <i>Annotation</i> .

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of an *AnnotationModel* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the *ABAC Authorization* PSM extension meta-model that can be found in appendix A.1.

- *atLeastOneAnnJavaResourceModelExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnJavaResourceModel*.
- *atLeastOneAnnHTTPActivityHandlerExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnHTTPActivityHandler*.
- *atLeastOneAnnJPAControllerExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnJPAController*.
- *atLeastOneAnnHTTPActivityExists*: This OCL constraint checks whether there exists at least one *AnnotatedElement* that is associated with this *AnnotationModel* that is of type *AnnHTTPActivity*.
- *exactlyOneAuthorizationSubjectExists*: This OCL constraint checks whether there exists exactly one *Annotation* element that is associated with this *AnnotationModel* that is of type *AuthorizationSubject*.
- *atLeastOneAuthorizableResourceExists*: This OCL constraint checks whether there exists at least one *Annotation* elements that is associated with this *AnnotationModel* that is of type *AuthorizableResource*.
- *atLeastOneAuthorizationPerformerExists*: This OCL constraint checks whether there exists at least one *Annotation* element that is associated with this *AnnotationModel* that is of type *AuthorizationPerformer*.
- *atLeastOneAuthorizationDataHandler*: This OCL constraint checks whether there exists at least one *Annotation* element that is associated with this *AnnotationModel* that is of type *AuthorizationDataHandler*.

6.5.1.2.2 *AnnotatedElement* Element

Overview

The *AnnotatedElement* element models any existent Core PSM meta-model element that can be annotated by the *ABAC Authorization* Core PSM extension in order to extend its semantic meaning and/or design. It is always further specialized to more specific categories of annotated elements.

Properties

The *AnnotatedElement* element of the *ABAC Authorization PSM* extension meta-model does not have any property.

Relations

The *AnnotatedElement* element of the *ABAC Authorization PSM* extension meta-model does not have any relation.

Behavioural Restrictions

The *AnnotatedElement* element of the *ABAC Authorization PSM* extension meta-model does not have any behavioural restriction.

6.5.1.2.3 Annotation Element

Overview

The *Annotation* element models any new semantic meaning or design that is attached to a Core PSM meta-model element. It is always further specialized to more specific categories of annotations.

Properties

The *Annotation* element of the *ABAC Authorization PSM* extension meta-model does not have any property.

Relations

The *Annotation* element of the *ABAC Authorization PSM* extension meta-model does not have any relation.

Behavioural Restrictions

The *Annotation* element of the *ABAC Authorization PSM* extension meta-model does not have any relation.

6.5.1.2.4 AnnHTTPActivity Element

Overview

The *AnnHTTPActivity* element models an existent *HTTPActivity* of the Core PSM model that can be annotated by the *ABAC Authorization PSM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnHTTPActivity* element of the *ABAC Authorization PSM* extension meta-model does not have any property

Relations

Table 6-42 AnnHTTPActivity's Relations

Relation With PSM Element	Type	Multiplicity	Structural Constraints
HTTPActivity	reference	1	The <i>AnnHTTPActivity</i> must have exactly one reference with an <i>HTTPActivity</i> element of the Core PSM model.

Behavioural Restrictions

The *AnnHTTPActivity* element of the *ABAC Authorization PSM* extension meta-model does not have any behavioural restriction.

6.5.1.2.5 *AnnPSMComponentProperty* Element

Overview

The *AnnPSMComponentProperty* element models an existent *PSMComponentProperty* of the Core PSM model that can be annotated by the *ABAC Authorization PSM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnPSMComponentProperty* element of the *ABAC Authorization PSM* extension meta-model does not have any property.

Relations

Table 6-43 *AnnPSMComponentProperty*'s Relations

Relation With PSM Element	Type	Multiplicity	Structural Constraints
PSMComponentProperty	reference	1	The <i>AnnPSMComponentProperty</i> must reference exactly one <i>PSMComponentProperty</i> element of the Core PSM model.

Behavioural Restrictions

The *AnnPSMComponentProperty* element of the *ABAC Authorization PSM* extension meta-model does not have any behavioural restriction.

6.5.1.2.6 *AnnJPAController* Element

Overview

The *AnnJPAController* element models an existent *JPAController* of the Core PSM model that can be annotated by the *ABAC Authorization PSM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnJPAController* element of the *ABAC Authorization PSM* extension meta-model does not have any property.

Relations

Table 6-44 *AnnJPAController*'s Relations

Relation With PSM Element	Type	Multiplicity	Structural Constraints
JPAController	reference	1	The <i>AnnJPAController</i> must reference exactly

			one <i>JPAController</i> element of the Core PSM model.
--	--	--	---

Behavioural Restrictions

The *AnnJPAController* element of the *ABAC Authorization PSM* extension meta-model does not have any behavioural restriction.

6.5.1.2.7 AnnHTTPActivityHandler Element

Overview

The *AnnHTTPActivityHandler* element models an existent *HTTPActivityHandler* of the Core PSM model that can be annotated by the *ABAC Authorization PSM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnHTTPActivityHandler* element of the *ABAC Authorization PSM* extension meta-model does not have any property.

Relations

Table 6-45 AnnHTTPActivityHandler's Relations

Relation With PSM Element	Type	Multiplicity	Structural Constraints
HTTPActivityHandler	reference	1	The <i>AnnHTTPActivityHandler</i> must reference exactly one <i>HTTPActivityHandler</i> element of the Core PSM model.

Behavioural Restrictions

The *AnnHTTPActivityHandler* element of the *ABAC Authorization PSM* extension meta-model does not have any behavioural restriction.

6.5.1.2.8 AnnJavaResourceModelManager Element

Overview

The *AnnJavaResourceModelManager* element models an existent *JavaResourceModelManager* of the Core PSM model that can be annotated by the *ABAC Authorization PSM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnJavaResourceModelManager* element of the *ABAC Authorization PSM* extension meta-model does not have any property.

Relations

Table 6-46 AnnJavaResourceModelManager's Relations

Relation With PSM Element	Type	Multiplicity	Structural Constraints
JavaResourceModelManager	reference	1	The <i>AnnJavaResourceModelManager</i> must reference exactly one <i>JavaResourceModelManager</i> element of the Core PSM.

Behavioural Restrictions

The *AnnJavaResourceModelManager* element of the *ABAC Authorization PSM* extension meta-model does not have any behavioural restriction.

6.5.1.2.9 AnnJavaAlgoResourceModel Element

Overview

The *AnnJavaAlgoResourceModel* element models an existent *JavaAlgoResourceModel* of the Core PSM model that can be annotated by the *ABAC Authorization PSM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnJavaAlgoResourceModel* element of the *ABAC Authorization PSM* extension meta-model does not have any property.

Relations

Table 6-47 AnnJavaAlgoResourceModel's Relations

Relation With PSM Element	Type	Multiplicity	Structural Constraints
JavaAlgoResourceModel	reference	1	The <i>AnnJavaAlgoResourceModel</i> must reference exactly one <i>JavaAlgoResourceModel</i> element of the Core PSM.

Behavioural Restrictions

The *AnnJavaAlgoResourceModel* element of the *ABAC Authorization PSM* extension meta-model does not have any behavioural restriction.

6.5.1.2.10 AnnJavaResourceModel Element

Overview

The *AnnJavaResourceModel* element models an existent *JavaResourceModel* of the Core PSM model that can be annotated by the *ABAC Authorization PSM* extension meta-model, in order to alter its semantic meaning and/or design.

Properties

The *AnnJavaResourceModel* element of the *ABAC Authorization PSM* extension meta-model does not have any property.

Relations

Table 6-48 AnnJavaResourceModel's Relations

Relation With PSM Element	Type	Multiplicity	Structural Constraints
JavaResourceModel	reference	1	The <i>AnnJavaResourceModel</i> must reference exactly one <i>JavaResourceModel</i> element of the Core PSM model.

Behavioural Restrictions

The *AnnJavaResourceModel* element of the *ABAC Authorization PSM* extension meta-model does not have any behavioural restriction.

6.5.1.2.11 AuthorizationSubject Element

Overview

The *AuthorizationSubject* element models an annotation of the *ABAC Authorization PSM* extension meta-model that is intended to annotate an existent Core PSM *JavaResourceModel*. With this annotation, the *JavaResourceModel* will be used as an access request issuer model, some attributes of which should have certain runtime values in order to grant access to him. Figure 6-18 demonstrates the *AuthorizationSubject* element of the *ABAC Authorization PSM* extension meta-model and its relations.

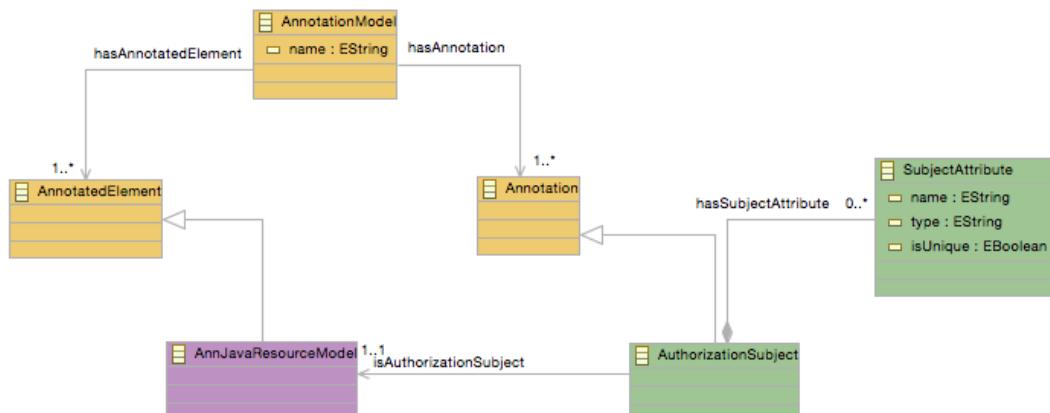


Figure 6-18 AuthorizationSubject annotation and its relations

Properties

The *AuthorizationSubject* element of the *ABAC Authorization PSM* extension meta-model does not have any property

Relations

Table 6-49 AuthorizationSubject's Relations

Relation With PSM Extension Element	Type	Multiplicity	Structural Constraints
AnnJavaResourceModel	association	1	The <i>AuthorizationSubject</i> must have exactly one association with an <i>AnnJavaResourceModel</i> , which references a <i>JavaResourceModel</i> of the Core PSM model. With this annotation, that <i>JavaResourceModel</i> will be used as an access request issuer model, some attributes of which must have certain runtime values in order to grant access to him.
SubjectAttribute	composition	0..*	The <i>AuthorizationSubject</i> can have zero or more composition associations with <i>SubjectAttributes</i> . This association models the fact that an <i>AuthorizationSubject</i> may have some additional attributes, other than those defined in the Core CIM Resource model, which are authorization oriented.

Behavioural Restrictions

The *AuthorizationSubject* element of the *ABAC Authorization PSM* extension meta-model does not have any behavioural restriction.

6.5.1.2.12 *SubjectAttribute* Element

Overview

The *SubjectAttribute* element models an extra property that can be added to the *JavaResourceModel* that is annotated as *AuthorizationSubject*, that can be authorization oriented.

Properties

Table 6-50 SubjectAttribute's Properties

Name	Type	Multiplicity	Explanation
name	EString	1	This is the name of the <i>SubjectAttribute</i> .
type	EString	1	This is the type of the <i>SubjectAttribute</i> .

isUnique	EString	1	This is the multiplicity of the <i>SubjectAttribute</i> . If it is set to <i>true</i> then it has multiplicity one. Otherwise, it is a multivalued array.
----------	---------	---	---

Relations

The *SubjectAttribute* element of the *ABAC Authorization PSM* extension meta-model does not have any relation.

Behavioural Restrictions

The *SubjectAttribute* element of the *ABAC Authorization PSM* extension meta-model does not have any behavioural restriction.

6.5.1.2.13 AuthorizableResource Element

Overview

The *AuthorizableResource* element models an annotation of the *ABAC Authorization PSM* extension meta-model that is intended to annotate either an existent Core PSM *JavaResourceModel/JavaResourceModelManager* pair or a *JavaAlgoResourceModel*. With this annotation, either of the *JavaResourceModel*, *JavaResourceModelManager* or *JavaAlgoResourceModel* will be accessible through their respective web API (controllers) if and only if their *ABAC Authorization* access rule set is evaluated and yields permission to the request issuer. Figure 6-19 demonstrates the *AuthorizableResource* element of the *ABAC Authorization PSM* extension meta-model and its relations.

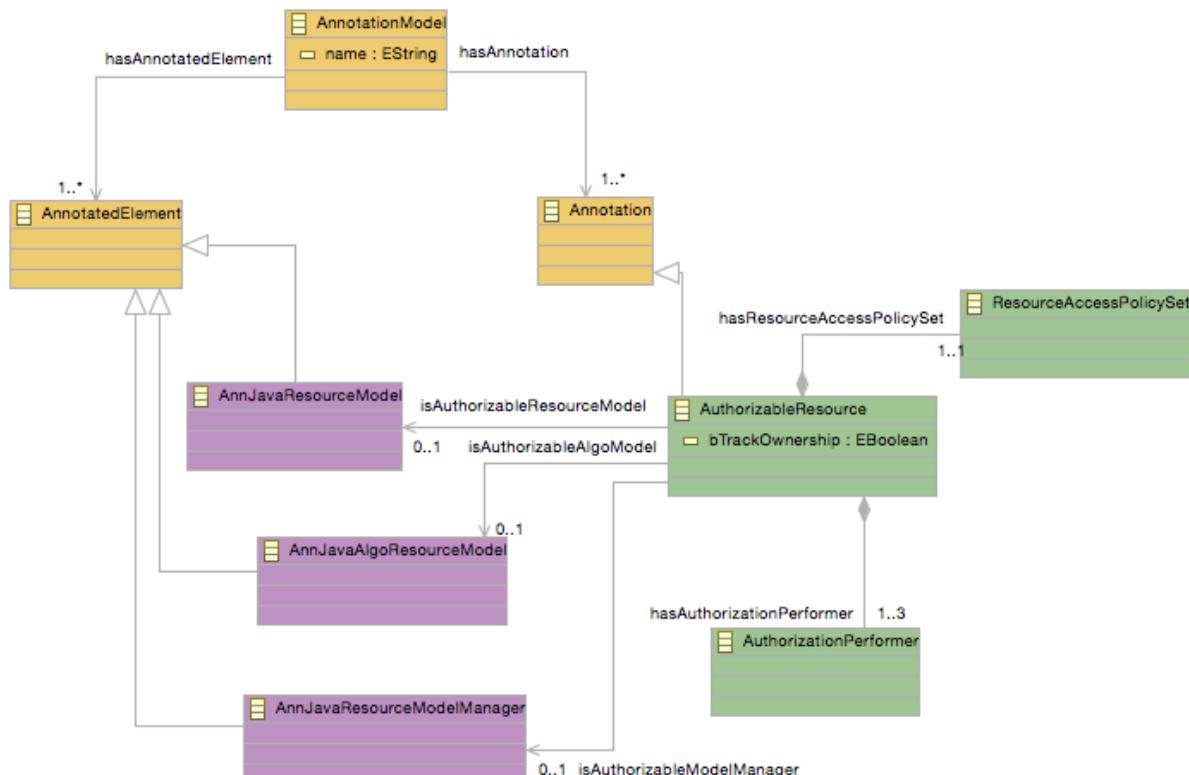


Figure 6-19 AuthorizableResource annotation and its relations

Properties

Table 6-51 AuthorizableResource's Properties

Name	Type	Multiplicity	Explanation
bTrackOwnership	EString	1	If this <i>EBoolean</i> is set to true, then the ownership of the referenced resource is tracked. This enables unix like authorization scheme implementation.

Relations

Table 6-52 AuthorizableResource's Relations

Relation With PSM Extension Element	Type	Multiplicity	Structural Constraints
AnnJavaResourceModel	association	0..1	The <i>AuthorizableResource</i> may have an association with one <i>AnnJavaResourceModel</i> , which references a <i>JavaResourceModel</i> of the Core PSM model. With this annotation, envisioned system's clients will only be granted access to that <i>JavaResourceModel</i> , if its <i>ABAC Authorization</i> rule set is evaluated and concludes likewise.
AnnJavaResourceModelManager	association	0..1	The <i>AuthorizableResource</i> may have an association with one <i>AnnJavaResourceModelManager</i> , which references a <i>JavaResourceModelManager</i> element of the Core PSM model. With this annotation, envisioned system's clients will only be granted access to that <i>JavaResourceModelManager</i> , if its <i>ABAC Authorization</i> rule set is evaluated and concludes likewise.
AnnJavaAlgoResourceModel	association	0..1	The <i>AuthorizableResource</i> may have an association with one <i>AnnJavaAlgoResourceModel</i> , which references a <i>JavaAlgoResourceModel</i> . With this annotation, envisioned system's clients will only be granted access to that <i>JavaAlgoResourceModel</i> , if its <i>ABAC Authorization</i> rule set is evaluated and concludes likewise.
AuthorizationPerformer	composition	1	The <i>AuthorizableResource</i> must have a composition association with exactly one <i>AuthorizationPerformer</i> . This association models the fact that every <i>AuthorizableResource</i> delegates the handling of authorization tasks to a dedicated part of it, the <i>AuthorizationPerformer</i> .
ResourceAccessPolicySet	composition	1	The <i>AuthorizableResource</i> must have a

			composition association with exactly one <i>ResourceAccessPolicySet</i> . This association models the fact that every <i>AuthorizableResource</i> has a set of access policies that together yield a conclusion on whether the request issuer should be granted access or not.
--	--	--	--

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of a *AuthorizableResource* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the *ABAC Authorization* PSM extension meta-model that can be found in appendix A.1.

- *annotatesExactlyOneResource*: This OCL constraint validates that any *AuthorizableResource* annotation, annotates either only a pair of *JavaResourceModel*/*JavaResourceModelManager* or a *JavaAlgoResourceModel* but never both, or none.

6.5.1.2.14 ResourceAccessPolicySet Element

Overview

The *ResourceAccessPolicySet* element models a set of resource access policies. These access policies include all the access rules that are evaluated together and yield a conclusion on whether the access request issuer should be granted permission or not. Figure 6-20 demonstrates the *ResourceAccessPolicySet* element of the *ABAC Authorization* PSM extension meta-model and its relations.



Figure 6-20 ResourceAccessPolicySet element and its relations

Properties

The *ResourceAccessPolicySet* element of the *ABAC Authorization PSM* extension meta-model does not have any property.

Relations

Table 6-53 ResourceAccessPolicySet's Relations

Relation With PSM Extension Element	Type	Multiplicity	Structural Constraints
JPAAnnotation	composition	1	The <i>ResourceAccessPolicySet</i> must have a composition association with exactly one <i>JPAAnnotation</i> . This association models the fact that the <i>ResourceAccessPolicySet</i> of every <i>AuthorizableResource</i> is stored in the envisioned system's local database using <i>JPA Annotations</i> .
ResourceAccessPolicy	composition	1..*	The <i>ResourceAccessPolicySet</i> must have a composition association with exactly one <i>ResourceAccessPolicy</i> . This association models the fact that every <i>ResourceAccessPolicySet</i> comprises at least one <i>ResourceAccessPolicy</i> .
RuleCombiningAlgorithm	composition	1	The <i>ResourceAccessPolicySet</i> must have a composition association with exactly one <i>RuleCombiningAlgorithm</i> . This association models the fact that every <i>ResourceAccessPolicySet</i> has exactly one policy evaluation algorithm that defines the way this evaluation is performed.

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of a *ResourceAccessPolicySet* element. The definition of each restriction begins by providing the unique OCL name of the implemented restriction within the *ABAC Authorization PIM* extension meta-model that can be found in appendix A.1.

- *hasSpecializedRuleCombiningAlgorithm*: This OCL constraint checks whether the *RuleCombiningAlgorithm* of any *ResourceAccessPolicySet* is always further specialized to as either *DenyOverridesAlgorithm* or *PermitOverridesAlgorithm*.

6.5.1.2.15 ResourceAccessPolicy Element

Overview

The *ResourceAccessPolicy* element models a resource access policy. That policy must be evaluated in order to conclude whether the access request issuer should be granted access or not. Figure 6-21 demonstrates the *ResourceAccessPolicy* element of the *ABAC Authorization PSM* extension meta-model and its relations.



Figure 6-21 ResourceAccessPolicy element and its relations

Properties

The *ResourceAccessPolicy* element of the *ABAC Authorization PSM* extension meta-model does not have any property.

Relations

Table 6-54 ResourceAccessPolicy's Relations

Relation With PSM Extension Element	Type	Multiplicity	Structural Constraints
JPAAnnotation	composition	1	The <i>ResourceAccessPolicy</i> must have a composition association with exactly one <i>JPAAnnotation</i> . This association models the fact that every <i>ResourceAccessPolicy</i> is stored in the envisioned system's local database.
ResourceAccessRule	composition	1..*	The <i>ResourceAccessPolicy</i> must have a composition association with at least one <i>ResourceAccessRule</i> . This association models the fact that every <i>ResourceAccessPolicy</i> must have at least one access rule in order to be meaningful.

Behavioural Restrictions

This subsection lists all the behavioural restrictions of the properties and relations of a *ResourceAccessPolicy* element. The definition of each restriction begins by providing the unique OCL

name of the implemented restriction within the *ABAC Authorization* PIM extension meta-model that can be found in appendix A.1.

- *hasSpecializedRuleCombiningAlgorithm*: This OCL constraint checks whether every *ResourceAccessPolicy* has a specialized *RuleCombiningAlgorithm* that can be either *DenyOverridesAlgorithm* or *PermitOverridesAlgorithm*.
- *hasSpecializedResourceAccessRule*: This OCL constraint checks whether every *ResourceAccessPolicy* has all its *ResourceAccessRules* specialized as either *ResourceAccessDenyRule* or *ResourceAccessPermitRule*.

6.5.1.2.16 *RuleCombiningAlgorithm* Element

Overview

The *RuleCombiningAlgorithm* element models an algorithm that defines the way a *ResourceAccessPolicySet* or *ResourceAccessPolicy* are evaluated in order to conclude whether the access request issuer should be granted access or not. It is always further specialized as either *DenyOverridesAlgorithm* or *PermitOverridesAlgorithm*.

Properties

The *RuleCombiningAlgorithm* element of the *ABAC Authorization PSM* extension meta-model does not have any property.

Relations

The *RuleCombiningAlgorithm* element of the *ABAC Authorization PSM* extension meta-model does not have any relation.

Behavioural Restrictions

The *RuleCombiningAlgorithm* element of the *ABAC Authorization PSM* extension meta-model does not have any behavioural restriction.

6.5.1.2.17 *DenyOverridesAlgorithm* Element

Overview

The *DenyOverridesAlgorithm* element models a *RuleCombiningAlgorithm* specialization. With this specialization, whenever even just one *ResourceAccessRule* of a *ResourceAccessPolicySet* or a *ResourceAccessPolicy* evaluates to deny, the access request issuer is not granted access.

Properties

The *DenyOverridesAlgorithm* element of the *ABAC Authorization PSM* extension meta-model does not have any property other than those inherited from the *RuleCombiningAlgorithm*.

Relations

The *DenyOverridesAlgorithm* element of the *ABAC Authorization PSM* extension meta-model does not have any relation other than those inherited from the *RuleCombiningAlgorithm*.

Behavioural Restrictions

The *DenyOverridesAlgorithm* element of the *ABAC Authorization PSM* extension meta-model does not have any behavioural restriction.

6.5.1.2.18 PermitOverridesAlgorithm Element

Overview

The *PermitOverridesAlgorithm* element models a *RuleCombiningAlgorithm* specialization. With this specialization, whenever even just one *ResourceAccessRule* of a *ResourceAccessPolicySet* or of a *ResourceAccessPolicy* evaluates to permit, the access request issuer is granted access.

Properties

The *PermitOverridesAlgorithm* element of the *ABAC Authorization PSM* extension meta-model does not have any property other than those inherited from the *RuleCombiningAlgorithm*.

Relations

The *PermitOverridesAlgorithm* element of the *ABAC Authorization PSM* extension meta-model does not have any relation other than those inherited from the *RuleCombiningAlgorithm*.

Behavioural Restrictions

The *PermitOverridesAlgorithm* element of the *ABAC Authorization PSM* extension meta-model does not have any behavioural restriction.

6.5.1.2.19 ResourceAccessRule Element

Overview

The *ResourceAccessRule* element models a resource access rule. Each such access rule is evaluated every time an access request is made and this evaluation either yields permission or not. It is always further specialized as either *ResourceAccessDenyRule* or *ResourceAccessPermitRule*. Figure 6-22 demonstrates the *ResourceAccessRule* element of the *ABAC Authorization PSM* extension meta-model and its relations.

Properties

The *ResourceAccessRule* element of the *ABAC Authorization PSM* extension meta-model does not have any property.

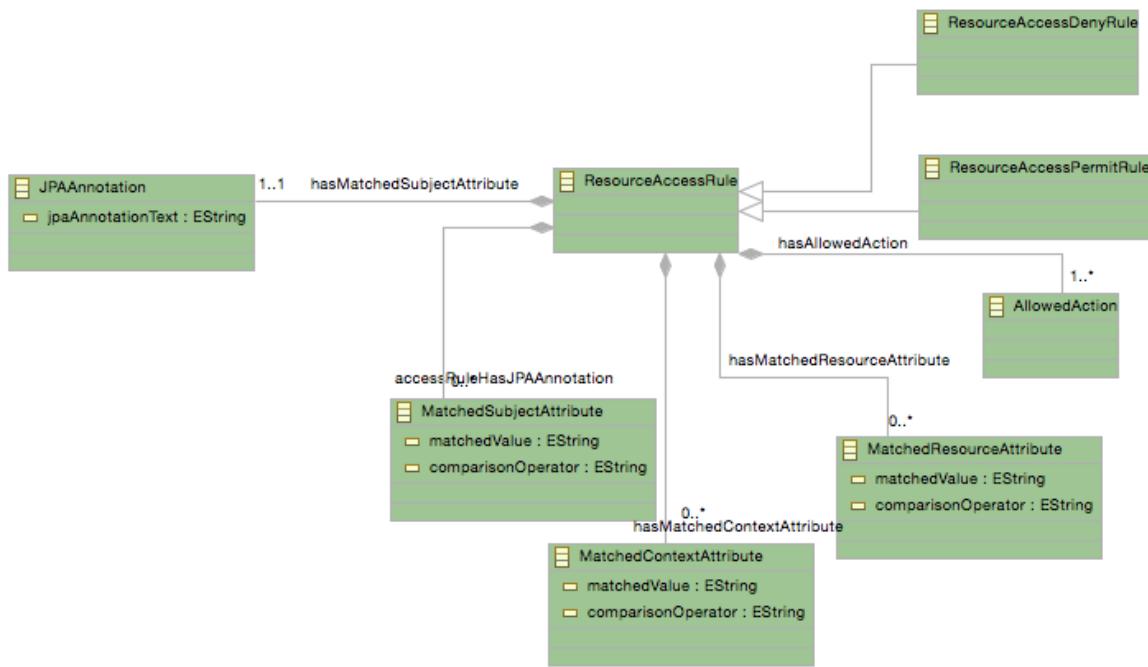


Figure 6-22 ResourceAccessRule element and its relations.

Relations

Table 6-55 ResourceAccessRule's Relations

Relation With PSM Extension Element	Type	Multiplicity	Structural Constraints
JPAAnnotation	composition	1	The <i>ResourceAccessRule</i> must have a composition association with exactly one <i>JPAAnnotation</i> . This association models the fact that every <i>ResourceAccessRule</i> is stored in the envisioned system's local database using <i>JPA Annotations</i> .
AllowedAction	composition	1..*	The <i>ResourceAccessRule</i> must have a composition association with at least one <i>AllowedAction</i> . This association models the fact that every <i>ResourceAccessRule</i> can have one or more allowed actions associated with it, if it is evaluated and yields permission to the access request issuer.
MatchedSubjectAttribute	composition	0..*	The <i>ResourceAccessRule</i> can have a composition association with zero or more <i>MatchedSubjectAttributes</i> . This association models the fact that any <i>ResourceAccessRule</i> 's evaluation may include the comparison of the runtime value of zero or more access request issuer model attribute with some predefined values, following the <i>ABAC Authorization</i> .

			scheme.
MatchedResourceAttribute	composition	0..*	The <i>ResourceAccessRule</i> can have a composition association with zero or more <i>MatchedResourceAttributes</i> . This association models the fact that any <i>ResourceAccessRule</i> 's evaluation may include the comparison of the runtime value of zero or more attributes of the underlying resource, to which access is requested, with some predefined values following the <i>ABAC Authorization</i> scheme.
MatchedContextAttribute	composition	0..*	The <i>ResourceAccessRule</i> can have a composition association with zero or more <i>MatchedContextAttributes</i> . This association models the fact that any <i>ResourceAccessRule</i> 's evaluation may include the comparison of the runtime value of zero or more context resources attributes with some predefined values, following the <i>ABAC Authorization</i> scheme.

Behavioural Restrictions

The *ResourceAccessRule* element of the *ABAC Authorization PSM* extension meta-model does not have any behavioural restriction.

6.5.1.2.20 *ResourceAccessDenyRule* Element

Overview

The *ResourceAccessDenyRule* element models a *ResourceAccessRule* specialization. With this specialization, whenever a *ResourceAccessRule* is evaluated positively (e.g. all the attribute comparisons succeed) it denies access to the access request issuer.

Properties

The *ResourceAccessDenyRule* element of the *ABAC Authorization PSM* extension meta-model does not have any property other than those inherited from the *ResourceAccessRule*.

Relations

The *ResourceAccessDenyRule* element of the *ABAC Authorization PSM* extension meta-model does not have any relation other than those inherited from the *ResourceAccessRule*.

Behavioural Restrictions

The *ResourceAccessDenyRule* element of the *ABAC Authorization PSM* extension meta-model does not have any behavioural restriction.

6.5.1.2.21 *ResourceAccessPermitRule Element*

Overview

The *ResourceAccessPermitRule* element models a *ResourceAccessRule* specialization. With this annotation, whenever a *ResourceAccessRule* is evaluated positively (e.g. all the attribute comparisons succeed) it yields access to the access request issuer.

Properties

The *ResourceAccessPermitRule* element of the *ABAC Authorization PSM* extension meta-model does not have any other than those inherited from the *ResourceAccessRule*.

Relations

The *ResourceAccessPermitRule* element of the *ABAC Authorization PSM* extension meta-model does not have any relation other than those inherited from the *ResourceAccessRule*.

Behavioural Restrictions

The *ResourceAccessPermitRule* element of the *ABAC Authorization PSM* extension meta-model does not have any behavioural restriction.

6.5.1.2.22 *AllowedAction Element*

Overview

The *AllowedAction* element models an annotation of the *ABAC Authorization Core PSM* extension meta-model that annotates *HTTPActivities* of the Core PSM meta-model. It is an allowed action that is associated with a *ResourceAccessRule*. Figure 6-23 demonstrates the *AllowedAction* element of the *ABAC Authorization PSM* extension meta-model and its relations.

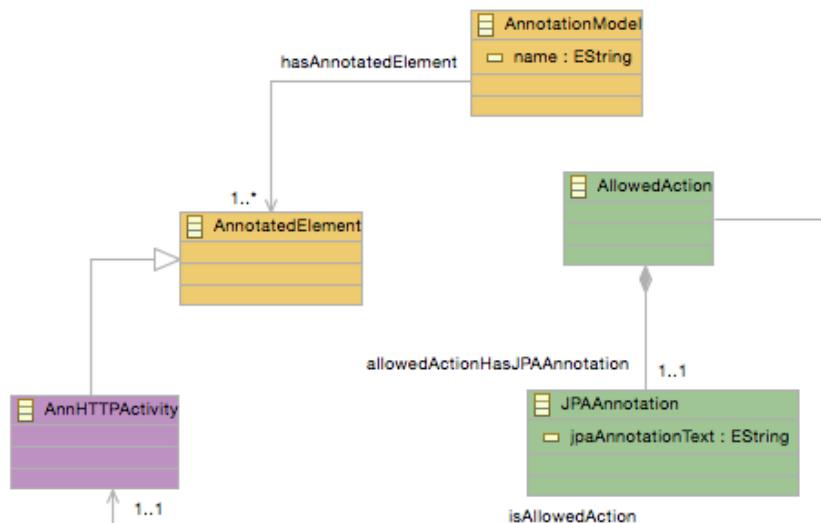


Figure 6-23 AllowedAction element and its relations.

Properties

The *AllowedAction* element of the *ABAC Authorization PSM* extension meta-model does not have any property.

Relations

Table 6-56 AllowedAction's Relations

Relation With PSM Extension Element	Type	Multiplicity	Structural Constraints
JPAAnnotation	composition	1	The <i>AllowedAction</i> must have a composition association with exactly one <i>JPAAnnotation</i> . This association models the fact that every allowed action of a <i>ResourceAccessRule</i> is stored in the envisioned system's local database.
AnnHTTPActivity	association	1	The <i>AllowedAction</i> must have an association with exactly one <i>AnnHTTPActivity</i> , which references an <i>HTTPActivity</i> element of the Core PSM model. With this annotation, the <i>HTTPActivity</i> will be accessible by an access request issuer, if and only if there is a <i>ResourceAccessPolicy</i> rule that can be satisfied and it has as allowed action that <i>HTTPActivity</i> and the <i>RuleCombiningRules</i> evaluate the <i>ResourceAccessPolicySet</i> accordingly.

Behavioural Restrictions

The *AllowedAction* element of the *ABAC Authorization PSM* extension meta-model does not have any behavioural restriction.

6.5.1.2.23 MatchedResourceAttribute Element

Overview

The *MatchedResourceAttribute* element models an annotation of the *ABAC Authorization PSM* extension meta-model that is intended to annotate an existent Core PSM *PSMComponentProperty*. With this annotation, the *PSMComponentProperty* runtime value will be compared with predefined values in any appropriate *ResourceAccessRule* so that a conclusion can be made on whether the access request issuer should be granted access or not. Figure 6-24 demonstrates the *MatchedResourceAttribute* element of the *ABAC Authorization PSM* extension meta-model and its relations.

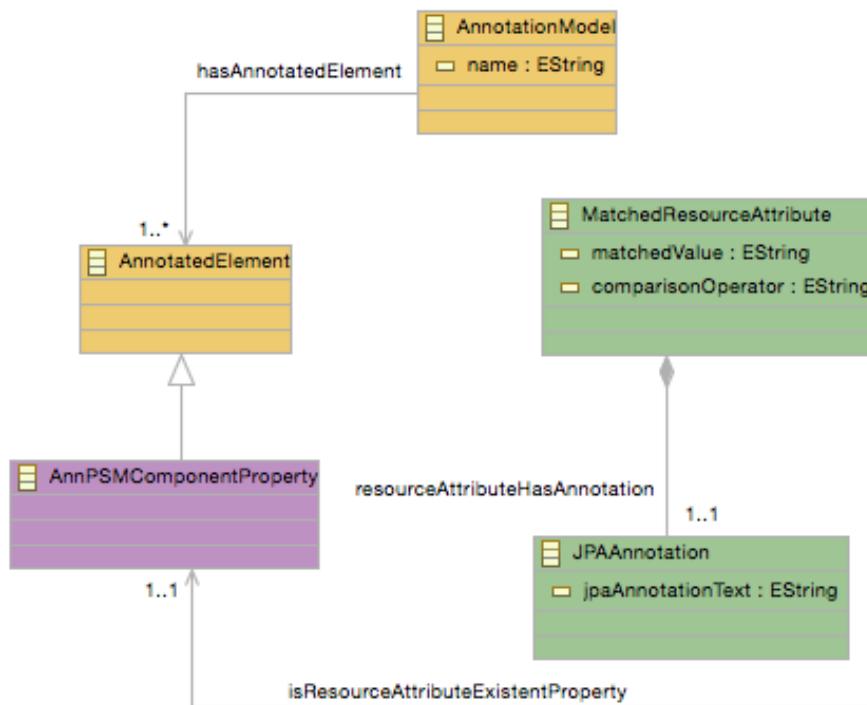


Figure 6-24 MatchedResourceAttribute element and its relations.

Properties

Table 6-57 MatchedResourceAttribute's Properties

Name	Type	Multiplicity	Explanation
matchedValue	EString	1..*	This is a list of values with which the runtime value of the referenced <i>PSMComponentProperty</i> will be compared whenever the overlying rule is evaluated.
comparisonOperator	EString	1	This is the comparison operator that will be used to perform the comparison of the runtime value of the referenced <i>PSMComponentProperty</i> with the predefined values in <i>matchedValue</i> . For example this can be “equals”, “greater than” etc.

Relations

Table 6-58 MatchedResourceAttribute's Relations

Relation With PSM Extension Element	Type	Multiplicity	Structural Constraints
JPAAnnotation	composition	1	The <i>MatchedResourceAttribute</i> must have a composition association with exactly one <i>JPAAnnotation</i> . This association models the fact that every <i>MatchedResourceAttribute</i> is stored in the envisioned system's local database using <i>JPA Annotations</i> .

AnnPSMComponentProperty	association	1	The <i>MatchedResourceAttribute</i> must have an association with exactly one <i>AnnPSMComponentProperty</i> , which references a <i>PSMComponentProperty</i> element of the Core PSM model. With this annotation, the <i>PSMComponentProperty</i> 's runtime value will be compared with the annotation's <i>matchedValue</i> whenever the overlying <i>ResourceAccessRule</i> is evaluated. Every <i>MatchedResourceAttribute</i> annotates a property of the overlying resource to which access is requested.
-------------------------	-------------	---	--

Behavioural Restrictions

The *MatchedResourceAttribute* element of the *ABAC Authorization PSM* extension meta-model does not have any behavioural restriction.

6.5.1.2.24 MatchedSubjectAttribute Element

Overview

The *MatchedSubjectAttribute* element models an annotation of the *ABAC Authorization PSM* extension meta-model that is intended to annotate an existent Core PSM *PSMComponentProperty*. With this annotation, the *PSMComponentProperty*'s runtime value will be compared with a set of predefined values of the overlying *ResourceAccessRule* whenever it is evaluated. Figure 6-25 demonstrates the *MatchedSubjectAttribute* element of the *ABAC Authorization PSM* extension meta-model and its relations.

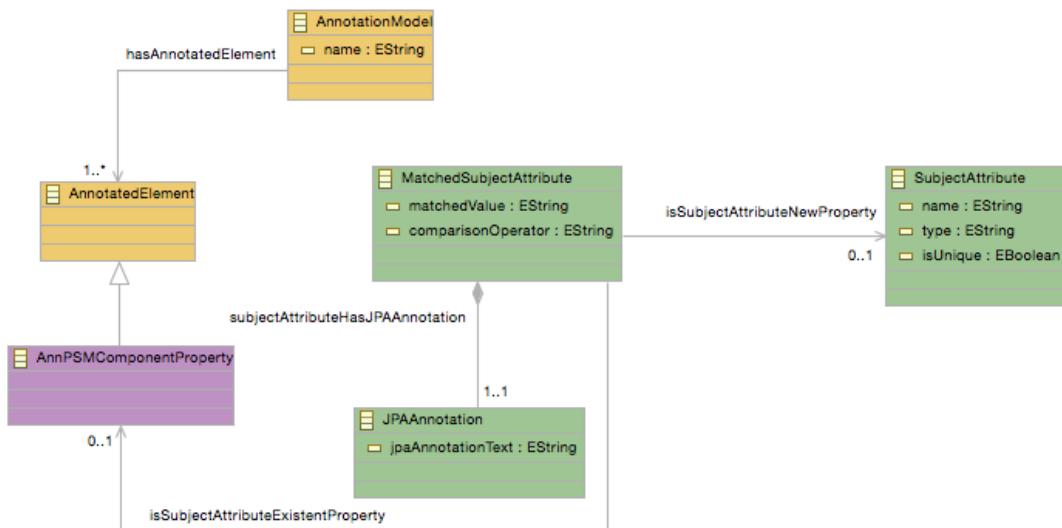


Figure 6-25 MatchedSubjectAttribute element and its relations.

Properties

Table 6-59 MatchedSubjectAttribute's Properties

Name	Type	Multiplicity	Explanation
matchedValue	EString	1	This is a list of values with which the runtime value of the referenced <i>PSMComponentProperty</i> will be compared whenever the overlying rule is evaluated.
comparisonOperator	EString	1	This is the comparison operator that will be used to perform the comparison of the runtime value of the referenced <i>PSMComponentProperty</i> with the predefined values in <i>matchedValue</i> . For example this can be “equals”, “greater than” etc.

Relations

Table 6-60 MatchedSubjectAttribute's Relations

Relation With PSM Extension Element	Type	Multiplicity	Structural Constraints
JPAAnnotation	composition	1	The <i>MatchedSubjectAttribute</i> must have a composition association with exactly one <i>JPAAnnotation</i> . This association models the fact that every <i>MatchedSubjectAttribute</i> is stored in the envisioned system's local database using <i>JPA Annotations</i> .
AnnPSMComponentProperty	association	1	The <i>MatchedSubjectAttribute</i> must have an association with exactly one <i>AnnPSMComponentProperty</i> , which references a <i>PSMComponentProperty</i> element of the Core PSM model. With this annotation, the <i>PSMComponentProperty</i> 's runtime value will be compared with the annotation's <i>matchedValue</i> whenever the overlying <i>ResourceAccessRule</i> is evaluated. Every <i>MatchedSubjectAttribute</i> annotates a property of the <i>JavaResourceModel</i> that is annotated as <i>AuthorizationSubject</i> (e.g. an access request issuer attribute).

Behavioural Restrictions

The *MatchedSubjectAttribute* element of the *ABAC Authorization PSM* extension meta-model does not have any behavioural restriction.

6.5.1.2.25 *MatchedContextAttribute* Element

Overview

The *MatchedContextAttribute* element models an annotation of the *ABAC Authorization* PSM extension meta-model that is intended to annotate an existent Core PSM *PSMComponentProperty*. With this annotation, the *PSMComponentProperty*'s runtime value will be compared with a set of predefined values whenever the overlying rule is evaluated. Figure 6-26 demonstrates the *MatchedContextAttribute* element of the *ABAC Authorization* PSM extension meta-model and its relations.

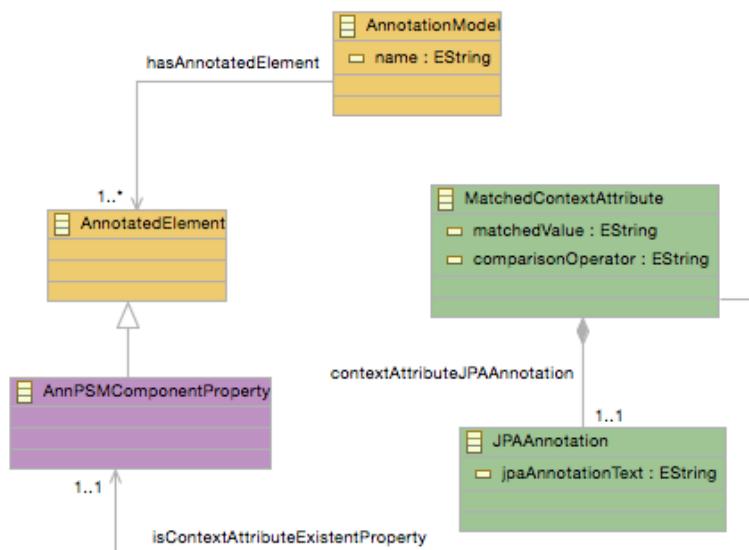


Figure 6-26 MatchedContextAttribute

Properties

Table 6-61 MatchedContextAttribute's Properties

Name	Type	Multiplicity	Explanation
matchedValue	EString	1	This is a list of values with which the runtime value of the referenced <i>PSMComponentProperty</i> will be compared whenever the overlying rule is evaluated.
comparisonOperator	EString	1	This is the comparison operator that will be used to perform the comparison of the runtime value of the referenced <i>PSMComponentProperty</i> with the predefined values in <i>matchedValue</i> . For example this can be “equals”, “greater than” etc.

Relations

Table 6-62 MatchedContextAttribute's Relations

Relation With PSM Extension Element	Type	Multiplicity	Structural Constraints
JPAAnnotation	composition	1	The <i>MatchedContextAttribute</i> must have a composition association with exactly one <i>JPAAnnotation</i> . This association models the fact that every <i>MatchedContextAttribute</i> is stored in

			the envisioned system's local database using <i>JPA Annotations</i> .
AnnPSMComponentProperty	association	1	The <i>MatchedContextAttribute</i> must have an association with exactly one <i>AnnPSMComponentProperty</i> , which references a <i>PSMComponentProperty</i> element of the Core PSM model. With this annotation, the <i>PSMComponentProperty</i> 's runtime value will be compared with the annotation's <i>matchedValue</i> whenever the overlying <i>ResourceAccessRule</i> is evaluated. Every <i>MatchedContextAttribute</i> annotates a property of the context of the overlying resource, to which access is requested.

Behavioural Restrictions

The *MatchedContextAttribute* element of the *ABAC Authorization PSM* extension meta-model does not have any behavioural restriction.

6.5.1.2.26 JPAAnnotation Element

Overview

The *JPAAnnotation* element models a *JPA* annotation that is used in order to automate the storage to the envisioned system's local database.

Properties

Table 6-63 JPAAnnotation's Properties

Name	Type	Multiplicity	Explanation
jpaAnnotationTEXT	EString	1	This is the text of the <i>JPA</i> annotation.

Relations

The *JPAAnnotation* element of the *ABAC Authorization PSM* extension meta-model does not have any property.

Behavioural Restrictions

The *JPAAnnotation* element of the *ABAC Authorization PSM* extension meta-model does not have any behavioural restriction.

6.5.1.2.27 AuthorizationPerformer Element

Overview

The *AuthorizationPerformer* element models an annotation of the *ABACAuthorization* PSM extension meta-model that is intended to annotate an existent Core PSM *HTTPActivityHandler*. With this annotation, the *HTTPActivityHandler* will handle any authorization actions that are needed for access requests of the overlying resource. Figure 6-27 demonstrates the *AuthorizationPerformer* element of the *ABAC Authorization* PSM extension meta-model and its relations.

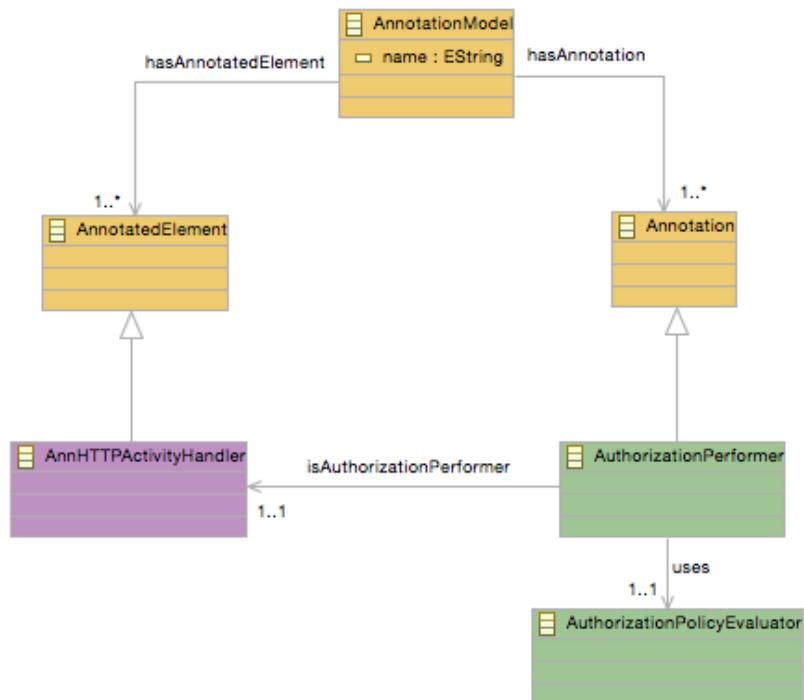


Figure 6-27 AuthorizationPerformer annotation and its relations.

Properties

The *AuthorizationPerformer* element of the *ABAC Authorization* PSM extension meta-model does not have any property.

Relations

Table 6-64 AuthorizationPerformer's Relations

Relation With PSM Extension Element	Type	Multiplicity	Structural Constraints
AnnHTTPActivityHandler	association	1	The <i>AuthorizationPerformer</i> must have an association with exactly one <i>AnnHTTPActivityHandler</i> , which references an <i>HTTPActivityHandler</i> element of the Core PSM model. With this annotation, the <i>HTTPActivityHandler</i> will perform any authorization actions needed whenever access requests are received for the overlying resource.

AuthorizationPolicyEvaluator	association	1	The <i>AuthorizationPerformer</i> must have an association with exactly one <i>AuthorizationPolicyEvaluator</i> . This association models the fact that every <i>AuthorizationPerformer</i> delegates the <i>ResourceAccessPolicySet</i> evaluation to a dedicated envisioned system's component name <i>AuthorizationPolicyEvaluator</i> .
------------------------------	-------------	---	---

Behavioural Restrictions

The *AuthorizationPerformer* element of the *ABAC Authorization PSM* extension meta-model does not have any behavioural restriction.

6.5.1.2.28 AuthorizationDataHandler Element

Overview

The *AuthorizationDataHandler* element models an annotation of the *ABAC Authorization PSM* extension meta-model that is intended to annotate an existent Core PSM *JPAController*. With this annotation, the *JPAController* will embed the needed functionality to store and retrieve to the envisioned system's local database authorization data. Figure 6-28 demonstrates the *AuthorizationDataHandler* element of the *ABAC Authorization PSM* extension meta-model and its relations.

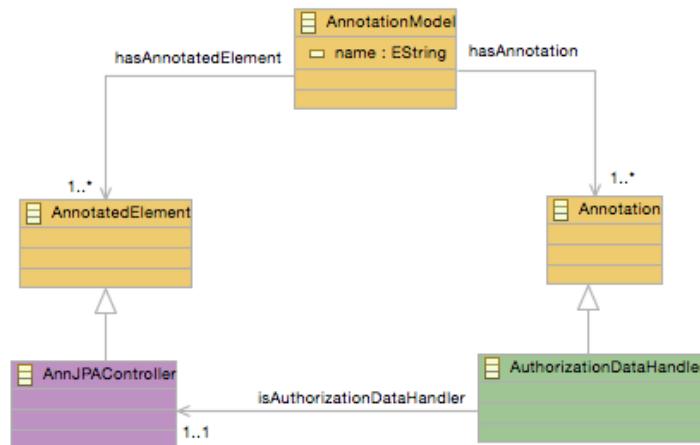


Figure 6-28 AuthorizationDataHandler

Properties

The *AuthorizationDataHandler* element of the *ABAC Authorization PSM* extension meta-model does not have any property.

Relations

Table 6-65 AuthorizationDataHandler's Relations

Relation With PSM Extension Element	Type	Multiplicity	Structural Constraints
AnnJPAController	association	1	The <i>AuthorizationDataHandler</i> must have an association with exactly one <i>AnnJPAController</i> , which references a <i>JPAController</i> element of the Core PSM model. With this annotation, the <i>JPAController</i> will embed the needed functionality to store and retrieve authorization data to/from the envisioned system's local database.

Behavioural Restrictions

The *AuthorizationDataHandler* element of the *ABAC Authorization PSM* extension meta-model does not have any behavioural restriction.

6.5.1.2.29 AuthorizationPolicyEvaluator Element

Overview

The *AuthorizationPolicyEvaluator* models a dedicated envisioned system's component that evaluates *ResourceAccessPolicySets*.

Properties

The *AuthorizationPolicyEvaluator* element of the *ABAC Authorization PSM* extension meta-model does not have any property.

Relations

The *AuthorizationPolicyEvaluator* element of the *ABAC Authorization PSM* extension meta-model does not have any relation.

Behavioural Restrictions

The *AuthorizationPolicyEvaluator* element of the *ABAC Authorization PSM* extension meta-model does not have any behavioural restriction.

7 Ontology Software Artefacts to MOF models transformation

7.1 Overview

As illustrated in tasks T3.1, T3.2, and T3.3 of WP3, the aggregated S-CASE ontology of software projects can be instantiated using input from different requirements representations, involving both the static (functional requirements, use case diagrams) and the dynamic (storyboards, activity diagrams) aspects of software projects. Upon instantiating the ontology (which is described in deliverable D3.2.2), the first transformation that takes place is the transformation from the ontology to the MOF models, and specifically the CIM.

Since the CIM may be overwhelming for the end-user, we first design an intermediate representation in YAML [11], which shall effectively describe the conceptual elements of the CIM and provide the user with the ability to modify (fine-grain) the model. YAML was selected as the representation since it is intuitive, human-readable (i.e. in comparison to XML) and is supported by several tools, even plugins of the Eclipse IDE [12].

Thus, in the following sections, we define the YAML representation, and determine how it can be extracted from the ontology of software artefacts. After that, the transformation from the YAML representation to the CIM is presented in detail.

7.2 Intermediary YAML Representation

YAML supports several well-known data structures, since it has been designed to be easily mapped to programming languages [11]. In our case, we use lists and associative arrays (i.e. key-value structures) in order to create a structure for resources, their properties and the different types of information that has to be stored for each resource. The schema of our representation is shown in Figure 7-1.

```

- !!cim.Resource
  Name: String
  IsAlgorithmic: Boolean
  CRUDActivities: List of Create, Read, Update, and/or Delete
  InputRepresentation: String/null
  OutputRepresentation: String/null
  Properties:
    - Name: String
      Type: Integer/Float/String/Boolean/null
      Unique: Boolean
      NamingProperty: Boolean
    - ...
  RelatedResources: List of String
- ...

```

Figure 7-1 Schema of the Intermediary YAML Representation

As shown in Figure 7-1, the main element of the YAML representation is the resource (`cim.Resource`). In fact, a project consists of a list of resources. Several fields are defined for each resource, each with its own data type and allowed values.

At first, each resource must have a name, which also has to be unique. Additionally, every resource may be either algorithmic (i.e. requiring some code to be written or some external service to be

called in order to be called) or non-algorithmic. This is represented using the `IsAlgorithmic` Boolean field.

There are four types of activities that can be applied to resources, in compliance with the CRUD actions (Create, Read, Update, Delete). Each resource may support one or more of these actions, i.e. any combination of them, represented as a list (`CRUDActivities`).

In the case that a resource is algorithmic, it may have an input and an output representation, stored in fields `InputRepresentation` and `OutputRepresentation` respectively. These representations hold the information required for calling external web services. For example, given a resource “wordmap” that is provided by an external web service, e.g. “Wordmap Unlimited”, the input representation would include information about calling the service, i.e. what type of verb must be used (e.g. GET, POST etc.) and what parameters may be required, while the output representation would explain the response of the service (e.g. JSON or XML). Since flexibility may be required for these representations, their type is alphanumerical (`String`).

Resources also have properties, which are defined as a list of objects. Each property has a `Name`, which is alphanumerical, as well as a `Type`, which corresponds to the common data types of programming languages, i.e. integers, float, strings, and booleans. Furthermore, each property has two boolean fields: `Unique` and `NamingProperty`. The former denotes whether the property has a unique value for each instance of the resource, while the latter denotes whether the resource is named after the value of this property. For example, a resource “user” could have the properties “username” and “email account”. In this case, the “username” would possibly be unique as could also be the “email account” (it could of course not be unique, allowing the user to have more than one email accounts). Any instance of “user”, however, should also be uniquely identified in the system. Thus, if we do not allow two users to have the same username, we could declare “username” as a naming property, and then we could refer to a user given his/her username.

Finally, each resource may have related resources. The field `RelatedResources` is a list of alphanumerical values corresponding to the names of other resources.

7.3 Ontology to YAML Representation Transformation Steps

7.3.1 Ontology to YAML Transformation

Extracting information from the aggregated ontology and creating the corresponding YAML file is a straightforward procedure. The aggregated ontology is, in fact, defined using RESTful-compliant terms. As noted in Deliverable D3.2.2, the elements of a project (subclasses of OWL class `Element`) in the ontology are the following:

- `Resource`
- `Activity`
- `Condition`
- `Property`
- `Representation`, which includes the subclasses:
 - `InputRepresentation`
 - `OutputRepresentation`

The ontology contains also information about the requirements/UML diagrams instances that these elements were extracted from (see more about the OWL classes and properties of the aggregated ontology in Deliverable D3.2.2), however these are not used for creating the YAML representation.

At first, instances of the OWL class `Resource` can directly be mapped to YAML objects of type `cim.Resource`. Each resource is initially considered non-algorithmic. The flow of activities and conditions is used to find the types of verbs that are used on any resource as well as the related

resources. Thus, for example, given an activity “Add bookmark” followed by an activity “Add tag”, one may identify two resources, “bookmark” and “tag”, where “tag” is also a related resource for “bookmark”. Additionally, both “bookmark” and “tag” must have the “Create” CRUD activity enabled, since the verb “add” implies creating a new instance. The type for each verb is recognized using a lexicon.

Whenever an action verb cannot be classified as any of the four CRUD types, a new algorithmic resource is created. Thus, for example, an activity “Search user” would spawn the new algorithmic resource “user_search”, and connecting it as a related resource of the resource “user”. Upon applying the transformations of the MDE engine, it would then be possible to manually write code for the function of this algorithmic resource.

Finally, the properties of the resources are mapped to the Properties list field, while any representations, input and output, are mapped to the fields InputRepresentation and OutputRepresentation respectively.

The implementation of this transformation is quite straightforward using the ontology API given in section 5.3 of Deliverable D3.2.2. The implementation involves iterating over all resources, extracting the activities of each resource and also checking whether the activities that follow may provide related resources. Finally, the properties of each resource are directly extracted from the ontology.

7.3.2 Example Transformation

In this subsection, we illustrate the transformation from the software artefacts that are stored in the aggregated ontology to the YAML representation using an example software project. We use project Restmarks [13] as an example for this transformation.

Restmarks is a service that can be seen as a social network where users can share their internet bookmarks. The users may also add tags to bookmarks, create, modify, or delete their bookmarks and search for their private bookmarks and/or public bookmarks of other users. In the context of the ontology to YAML file transformation, Restmarks shall provide an informative example since an instantiation of the aggregated ontology for this project is also present in deliverable D3.2.2 (see Figure 5.4 in D3.2.2).

The Restmarks project includes three resources: “account”, “bookmark”, and “tag”. Each of these resources has certain properties (e.g. “bookmark” has a name, “tag” has a description etc.), while there are also connections between these resources, so they are related. The service has no need of external web services, thus the input and output representation fields shall be null.

Upon using the transformation mechanism defined in the previous subsection, the YAML representation for project Restmarks is shown in Figure 7-2.

```

- !!cim.Resource
  Name: account
  IsAlgorithmic: false
  CRUDActivities: [Create, Read, Update, Delete]
  InputRepresentation: null
  OutputRepresentation: null
  Properties:
    - Name: username
      Type: null
      Unique: false
      NamingProperty: false
    - Name: password
      Type: null
      Unique: false
      NamingProperty: false
  RelatedResources: [bookmark]

```

```

- !!cim.Resource
  Name: tag_search
  IsAlgorithmic: true
  CRUDActivities: []
  InputRepresentation: null
  OutputRepresentation: null
  Properties: []
  RelatedResources: []

- !!cim.Resource
  Name: tag
  IsAlgorithmic: false
  CRUDActivities: [Create, Read, Update, Delete]
  InputRepresentation: null
  OutputRepresentation: null
  Properties:
    - Name: description
      Type: null
      Unique: false
      NamingProperty: false
  RelatedResources: [tag_search]

- !!cim.Resource
  Name: bookmark
  IsAlgorithmic: false
  CRUDActivities: [Create, Read, Update, Delete]
  InputRepresentation: null
  OutputRepresentation: null
  Properties:
    - Name: url
      Type: null
      Unique: false
      NamingProperty: false
    - Name: public
      Type: null
      Unique: false
      NamingProperty: false
    - Name: private
      Type: null
      Unique: false
      NamingProperty: false
  RelatedResources: [tag]

```

Figure 7-2 Example YAML file for Project Restmarks

As shown in Figure 7-2, the YAML representation involves the three aforementioned resources of the project, as well as a newly derived resource “tag_search”. Following the naming convention that was defined in subsection 7.3.1, we presume that this algorithmic resource was spawned by an activity “search” performed on resource “tag”.

Concerning the properties of each resource, the YAML representation of Figure 7-2 includes all recognized properties, i.e. “username” and “password” for “account”, “url” and “public” and “private” for “bookmark”, and “description” for “tag”. Note that “public” and “private” could be replaced by a “scope” property, however this information is not contained in the ontology. Also, one may note that the types of these properties as well as the information about their uniqueness and whether they are naming properties are not automatically derived from the ontology. This is expected since information about types, cardinality, etc. is not given at the requirements specification phase, but instead at the software design phase of software development. Although this

does not restrain the MDE engine from producing a functional service (by selecting certain defaults), having a fully user-compliant service may involve the developer filling in these fields.

Our mechanism also successfully extracted the actions for each resource. As shown in Figure 7-2, “account”, “bookmark”, and “tag” all allow performing “Create”, “Read”, “Update”, and “Delete” actions on them.

Additionally, the `RelatedResources` field is effectively instantiated using the dynamic elements of the ontology representation (actions and conditions). As shown in Figure 7-2, “account” has a related resource “bookmark”, and “bookmark” has a related resource “tag”. These relations are actually quite reasonable since a user of Restmarks should first have an account and then add bookmarks to his/her account. Additionally, bookmarks can have one or more tags. In the case of the algorithmic resource “tag_search”, we can see that it is a related resource of “tag”. This is also well defined since searching for a tag is an option that depends on the existence of tags.

Finally, as already mentioned, any missing information is usually not included in the requirements specification level of a project. Since the ontology contains artefacts extracted from requirements, design and implementation details such as the types of the properties or their uniqueness have to be added manually. In any case, manually editing any missing information is actually not a difficult task for the developer. In the case of Restmarks, for instance, the developer would only have to declare all types as alphanumerical (String). Additionally, any adjustments as to the names of the properties could be made, for example by replacing the “public” and “private” Boolean properties of “bookmark” with a property “scope” that would contain the corresponding information, or by adding the property “name” to “tag”.

After that, the developer would have to select one naming property per resource. The selection is quite intuitive, setting “username” as the naming property of “account”, “url” as the naming property of “bookmark”, and the newly added “name” as the naming property of “tag”. Obviously any naming property has also to be unique. After that, we may check whether any of the remaining types are also unique (none in this case). Finally, since no external web services are required, the value of the input and output representation fields should remain null. Thus, after these modifications, the final YAML representation of Restmarks is shown in Figure 7-3. Comparing this Figure to Figure 7-2, one can conclude that the modifications required are limited, while most of them are quite intuitive.

```

- !!cim.Resource
  Name: account
  IsAlgorithmic: false
  CRUActivities: [Create, Read, Update, Delete]
  InputRepresentation: null
  OutputRepresentation: null
  Properties:
    - Name: username
      Type: String
      Unique: true
      NamingProperty: true
    - Name: password
      Type: String
      Unique: false
      NamingProperty: false
  RelatedResources: [bookmark]

- !!cim.Resource
  Name: tag_search
  IsAlgorithmic: true
  CRUActivities: []
  InputRepresentation: null
  OutputRepresentation: null

```

```

Properties: []
RelatedResources: []

- !!cim.Resource
  Name: tag
  IsAlgorithmic: false
  CRUDActivities: [Create, Read, Update, Delete]
  InputRepresentation: null
  OutputRepresentation: null
  Properties:
    - Name: name
      Type: String
      Unique: true
      NamingProperty: true
    - Name: description
      Type: String
      Unique: false
      NamingProperty: false
  RelatedResources: [tag_search]

- !!cim.Resource
  Name: bookmark
  IsAlgorithmic: false
  CRUDActivities: [Create, Read, Update, Delete]
  InputRepresentation: null
  OutputRepresentation: null
  Properties:
    - Name: url
      Type: String
      Unique: true
      NamingProperty: true
    - Name: scope
      Type: String
      Unique: false
      NamingProperty: false
  RelatedResources: [tag]

```

Figure 7-3 User-modified YAML file for Project Restmarks

7.4 Intermediary YAML to CIM Transformation Steps

Once the aforementioned YAML file is populated with the parsed software artefacts from the S-CASE ontology, it is used as input to the MDE engine in order to produce the envisioned systems CIM model by following the next steps:

- Initially the MDE engines parses the YAML file and populates the envisioned system's CIM model.
- Then the S-CASE developer is prompted to make any modifications he wishes and fill in any incomplete artefacts.
- Once the S-CASE developer has finished any modifications, the input is validated against the Core CIM as well as the Core CIM extensions structural and behavioural constraints in order to check whether they are valid CIM models.
- If they are, the MDE engine begins the transformation process in order to produce the envisioned system as it is described in the following sections 8, 9 and 10. Otherwise, the S-CASE developer is prompted to fix the inconsistencies he introduced by making modifications.

8 S-CASE MDE Engine Transformation Definitions

8.1 Core CIM to Core PIM Transformation Steps

As section 3 already explained, one of the principal activities in Model Driven Engineering is the transformation of one model that conforms to a specific meta-model to another model which conforms to a different meta-model. This section defines the transformation of an envisioned CIM model of a service that conforms to the CIM meta-model, which section 4 defines, to the corresponding PIM model that conforms to the PIM meta-model that section 5 defines. Figure 8-1 revisits as indicative example the transformation process of the Core PIM to Core PSM meta-model as it has been already presented in D.2.2.

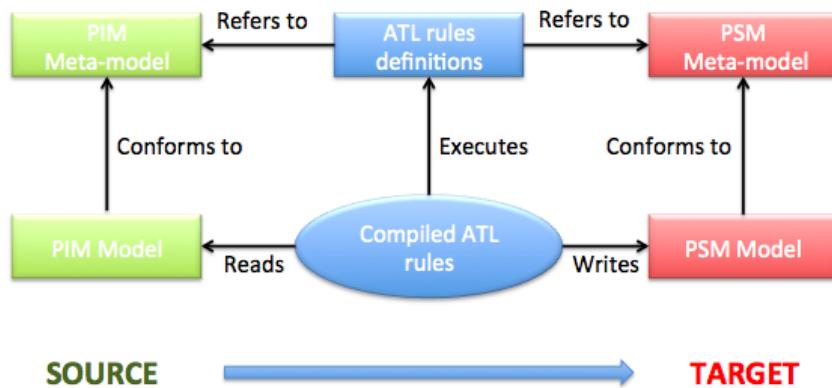


Figure 8-1 Indicative S-CASE transformation example introduced in D2.2

Every rule that the next section defines performs one out of two possible actions. It either reads an element of the input CIM model and transforms it to the corresponding output PIM model or further refines already transformed PIM elements by applying extra design patterns or relating them to PIM meta-model elements that have no CIM counterparts. Since ATL is a hybrid declarative/imperative programming language, these rules can be implemented either way. However, since according to Amstel [9] the declarative rules are more efficient, they are always preferred whenever possible.

Therefore, in the first case the MDE engine always uses ATL declarative rules. Moreover, the declarative rules increase the visibility and traceability of the transformation process since every declarative rule simply declares for every CIM meta-model element (along with its properties and relations) its PIM counterpart without defining the step-by-step transformation (imperative programming). Thus, every rule transparently links source with target elements. In the second case, since there are no counterparts between CIM and PIM meta-model elements, the ATL imperative rules are used instead.

Of course, the same transformation structure is used for all the Model-to-Model transformations of the S-CASE MDE engine, namely:

- Core CIM to Core PIM
- Authentication CIM extension to Authentication PIM extension
- Database Searching CIM extension to Database Searching PIM extension
- External Service Composition CIM extention to External Service Composition PIM extension
- ABAC Authorization CIM extension to ABAC Authorization PIM extension
- Core PIM to Core PSM

- Authentication PIM extension to Authentication PSM extension
- Database Searching PIM extension to Database Searching PSM extension
- External Service Composition PIM extension to External Service Composition PSM extension
- ABAC Authorization PIM extension to ABAC Authorization PSM extension

8.1.1 RESTfulServiceCIM element transformation

The *CIMToPIMService* ATL rule transforms *RESTfulServiceCIM* elements of the Core CIM meta-model, to *RESTfulServicePIM* ones of the Core PIM meta-model. Therefore, the *RESTfulServicePIM* embeds the abstract envisioned system's design of the conceptual service part modelled with the *RESTfulServiceCIM* element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-1 CIMToPIMService ATL rule

RESTfulServiceCIM property/relation	RESTfulServicePIM property/relation	Explanation
name	name	The <i>name</i> of the <i>RESTfulServiceCIM</i> remains unchanged after this transformation
serviceOutputPath	serviceOutputPath	The <i>serviceOutputPath</i> of the <i>RESTfulServiceCIM</i> remains unchanged after this transformation.
serviceDatabaseIP	serviceDatabaseIP	The <i>serviceDatabaseIP</i> of the <i>RESTfulServiceCIM</i> remains unchanged after this transformation.
serviceDatabasePort	serviceDatabasePort	The <i>serviceDatabasePort</i> of the <i>RESTfulServiceCIM</i> remains unchanged after this transformation.
serviceDatabaseUsername	serviceDatabaseUsername	The <i>serviceDatabaseUsername</i> of the <i>RESTfulServiceCIM</i> remains unchanged after this transformation.
serviceDatabasePassword	serviceDatabasePassword	The <i>serviceDatabasePassword</i> of the <i>RESTfulServiceCIM</i> remains unchanged after this transformation
-	DatabaseController	Once this transformation is done, one <i>DatabaseController</i> element of the Core PIM meta-model is added to the <i>RESTfulServicePIM</i> one, by calling the <i>createDatabaseController</i> ATL rule.
-	AlgoResourceModel	Once this transformation is done, one <i>AlgoResourceModel</i> element of the Core PIM meta-model is added to the <i>RESTfulServicePIM</i> one, by calling the <i>createAlgoResourceModel</i> ATL rule, for every CIM Resource that has its <i>isAlgorithmic</i> property set to true.
-	AlgoResourceController	Once this transformation is done, one <i>AlgoResourceController</i> element of the Core PIM meta-model is added to the <i>RESTfulServicePIM</i> one, by calling the <i>createAlgoResourceController</i>

		ATL rule, for every CIM <i>Resource</i> that has its <i>isAlgorithmic</i> property set to true.
-	ResourceModel	Once this transformation is done, one <i>ResourceModel</i> element of the Core PIM meta-model is added to the <i>RESTfulServicePIM</i> one, by calling the <i>createCRUDModel</i> ATL rule, for every CIM <i>Resource</i> that has its <i>isAlgorithmic</i> property set to false.
-	ResourceModelManager	Once this transformation is done, one <i>ResourceModelManager</i> element of the Core PIM meta-model is added to the <i>RESTfulServicePIM</i> one, by calling the <i>createCRUDRMManager</i> ATL rule, for every CIM <i>Resource</i> that has its <i>isAlgorithmic</i> property set to false.
-	ResourceController	Once this transformation is done, one <i>ResourceController</i> element of the Core PIM meta-model is added to the <i>RESTfulServicePIM</i> one, by calling the <i>createRController</i> ATL rule, for every CIM <i>Resource</i> that has its <i>isAlgorithmic</i> property set to false.
-	ResourceControllerManager	Once this transformation is done, one <i>ResourceControllerManager</i> element of the Core PIM meta-model is added to the <i>RESTfulServicePIM</i> one, by calling the <i>createRCManager</i> ATL rule, for every CIM <i>Resource</i> that has its <i>isAlgorithmic</i> property set to false.
-	RDBMSTable	Once this transformation is done, one <i>RDBMSTable</i> element of the Core PIM meta-model is added to the <i>RESTfulServicePIM</i> one, by calling the <i>createRDBMSTable</i> ATL rule, for every CIM <i>Resource</i> that has its <i>isAlgorithmic</i> property set to false.

8.1.2 AlgoResourceModel element introduction

The *createAlgoResourceModel* ATL rule introduces *AlgoResourceModel* elements of the Core PIM meta-model, by taking as input information from a *Resource* element of the Core CIM meta-model, which has its *isAlgorithmic* property set to true. That is, *AlgoResourceModels* embed one part of the abstract design in the envisioned system's PIM for the corresponding concepts of a CIM *Resource*. The properties and relations of the rule are defined in the table below:

Table 8-2 createAlgoResourceModel ATL rule

AlgoResourceModel property/relation	Explanation
parentName	The <i>parentName</i> property of the <i>AlgoResourceModel</i> is the name of the input Core CIM <i>Resource</i> .

name	The <i>name</i> property of the <i>AlgoResourceModel</i> equals to “Algo” + the name of the input Core CIM Resource + “Model”
ResourceInputRepresentation	Once this transformation is done, one <i>ResourceInputRepresentation</i> element of the Core PIM meta-model is added to the <i>AlgoResourceModel</i> one, by calling the <i>addAlgoModelRepresentation</i> ATL rule, for every CIM <i>InputRepresentation</i> of the input <i>Resource</i> .
ResourceOutputRepresentation	Once this transformation is done, one <i>ResourceOutputRepresentation</i> element of the Core PIM meta-model is added to the <i>AlgoResourceModel</i> one, by calling the <i>addAlgoModelORepresentation</i> ATL rule, for every CIM <i>OutputRepresentation</i> of the input <i>Resource</i> .
PIMComponentProperty	Once this transformation is done, one <i>PIMComponentProperty</i> element of the Core PIM meta-model is added to the <i>AlgoResourceModel</i> one, by calling the <i>createLinkListProperty</i> ATL rule in order to create a link list property to package hypermedia links in runtime.
SetterFunction	Once this transformation is done, one <i>SetterFunction</i> element of the Core PIM meta-model is added to the <i>AlgoResourceModel</i> one, by calling the <i>createLinkListPropertySetter</i> ATL rule, which adds a setter function for the link list property already added.
GetterFunction	Once this transformation is done, one <i>GetterFunction</i> element of the Core PIM meta-model is added to the <i>AlgoResourceModel</i> one, by calling the <i>createLinkListPropertyGetter</i> ATL rule, which adds a getter function for the link list property already added.
AlgoResourceModel	Once this transformation is done, one <i>AlgoResourceModel</i> element of the Core PIM meta-model is added to the <i>AlgoResourceModel</i> one, by calling the <i>createAlgoResourceModel</i> ATL rule, for every related CIM <i>Resource</i> of the input <i>Resource</i> one.
AlgoResourceModel	Once this transformation is done, one <i>AlgoResourceModel</i> element of the Core PIM meta-model is added to the <i>AlgoResourceModel</i> one, by calling the <i>createAlgoResourceModel</i> ATL rule, for every CIM <i>Resource</i> that has as related resource the input <i>Resource</i> of this rule.
RepresentationParseFunction	Once this transformation is done, one <i>RepresentationParseFunction</i> element of the Core PIM meta-model is added to the <i>AlgoResourceModel</i> one, by calling the <i>createParseFunction</i> ATL rule, for every <i>InputRepresentation</i> of the Core CIM input

	<i>Resource</i> of this rule.
RepresentationMarshallingFunction	Once this transformation is done, one <i>RepresentationMarshallingFunction</i> element of the Core PIM meta-model is added to the <i>AlgoResourceModel</i> one, by calling the <i>createMarshalFunction</i> ATL rule, for every <i>OutputRepresentation</i> of the Core CIM input <i>Resource</i> of this rule.

8.1.3 AlgoResourceController element introduction

The *createAlgoResourceController* ATL rule introduces *AlgoResourceController* elements of the Core PIM meta-model, by taking as input a *Resource* element of the Core CIM meta-model. That is, *AlgoResourceControllers* embed one part of the abstract design in the envisioned system's PIM for the corresponding concepts of a CIM *Resource*. The properties and relations of the rule are defined in the table below:

Table 8-3 createAlgoResourceController ATL rule

AlgoResourceController property/relation	Explanation
parentName	The <i>parentName</i> property of the <i>AlgoResourceController</i> is the name of the input Core CIM <i>Resource</i> .
name	The <i>name</i> property of the <i>AlgoResourceController</i> equals to “Algo” + the name of the input Core CIM <i>Resource</i> + “Controller”
controllerURI	The <i>controllerURI</i> property of the <i>AlgoResourceController</i> equals to “/Algo” + the name of the input Core CIM <i>Resource</i> .
AlgoResourceModel	Once this transformation is done, one <i>AlgoResourceModel</i> element of the Core PIM meta-model is added to the <i>AlgoResourceController</i> one, by calling the <i>createAlgoResourceModel</i> ATL rule, in order to associate the <i>AlgoResourceController</i> with the <i>AlgoResourceModel</i> that both originate from the same Core CIM <i>Resource</i> .
ResourceControllerCRUDActivity	Once this transformation is done, one <i>ResourceControllerCRUDActivity</i> element of the Core PIM meta-model is added to the <i>AlgoResourceController</i> one, by calling the <i>createAlgoControllerCRUDActivity</i> ATL rule.

8.1.4 ResourceModel element introduction

The *createCRUDModel* ATL rule introduces *ResourceModel* elements of the Core PIM meta-model, by taking as input a *Resource* element of the Core CIM meta-model. That is, *ResourceModels* embed one

part of the abstract design in the envisioned system's PIM for the corresponding concepts of a CIM *Resource* that has its *isAlgorithmic* property set to false. The properties and relations of the rule are defined in the table below:

Table 8-4 createCRUDModel ATL rule

ResourceModel property/relation	Explanation
parentName	The <i>parentName</i> property of the <i>ResourceModel</i> is the name of the input Core CIM <i>Resource</i> .
name	The <i>name</i> property of the <i>ResourceModel</i> equals to the name of the input Core CIM <i>Resource</i> + "Model"
ResourceModelManager	Once this transformation is done, one <i>ResourceModelManager</i> element of the Core PIM meta-model is added to the <i>ResourceModel</i> one, by calling the <i>createCRUDRMManager</i> ATL rule, in order to associate the <i>ResourceModel</i> with the <i>ResourceModelManagers</i> that originates from the from Core CIM <i>Resources</i> that are related of the Core CIM input <i>Resource</i> of this rule.
AlgoResourceModel	Once this transformation is done, one <i>AlgoResourceModel</i> element of the Core PIM meta-model is added to the <i>ResourceModel</i> one, by calling the <i>createAlgoResourceModel</i> ATL rule, in order to associate the <i>ResourceModel</i> with the <i>AlgoResourceModel</i> that originates from the Core CIM <i>Resources</i> that are related of the Core CIM input <i>Resource</i> of this rule.
ResourceInputRepresentation	Once this transformation is done, one <i>ResourceInputRepresentation</i> element of the Core PIM meta-model is added to the <i>ResourceModel</i> one, by calling the <i>addrModelRepresentation</i> ATL rule, for every <i>InputRepresentation</i> the Core CIM input <i>Resource</i> has.
ResourceOutputRepresentation	Once this transformation is done, one <i>ResourceOutputRepresentation</i> element of the Core PIM meta-model is added to the <i>ResourceModel</i> one, by calling the <i>addrModelOutputRepresentation</i> ATL rule, for every <i>OutputRepresentation</i> the Core CIM input <i>Resource</i> has.
PIMComponentProperty	Once this transformation is done, one <i>PIMComponentProperty</i> element of the Core PIM meta-model is added to the <i>ResourceModel</i> one, by calling the <i>addrModelProperties</i> ATL rule, for every <i>Property</i> the Core CIM input <i>Resource</i> has. It is also added one <i>PIMComponentProperty</i> element by calling the

	<i>createRModelPrimaryIdentifier</i> ATL rule in order to add a primary identifier property as well as one more <i>PIMComponentProperty</i> element by calling the <i>createLinkListProperty</i> ATL rule in order to add a link list property to the <i>ResourceModel</i> .
SetterFuntion	Once this transformation is done, one <i>SetterFunction</i> element of the Core PIM meta-model is added to the <i>ResourceModel</i> one, by calling the <i>addSetterFunction</i> ATL rule, for every <i>Property</i> the Core CIM input <i>Resource</i> has. It is also added one more <i>SetterFunction</i> element by calling the <i>createRModelPrimaryIdSetter</i> ATL rule in order to add a <i>SetterFunction</i> for the primary id property as well as one more <i>SetterFunction</i> by calling the <i>createLinkListPropertySetter</i> , in order to add a <i>SetterFunction</i> for the link list property to the <i>ResourceModel</i> .
GetterFunction	Once this transformation is done, one <i>GetterFunction</i> element of the Core PIM meta-model is added to the <i>ResourceModel</i> one, by calling the <i>addGetterFunction</i> ATL rule, for every <i>Property</i> the Core CIM input <i>Resource</i> has. It is also added one more <i>GetterFunction</i> element by calling the <i>createRModelPrimaryIdGetter</i> ATL rule in order to add a <i>GetterFunction</i> for the primary id property as well as one more <i>GetterFunction</i> by calling the <i>createLinkListPropertyGetter</i> , in order to add a <i>GetterFunction</i> for the link list property to the <i>ResourceModel</i> .
RDBMSTable	Once this transformation is done, one <i>RDBMSTable</i> element of the Core PIM meta-model is added to the <i>ResourceModel</i> one, by calling the <i>createRDBMSTable</i> ATL rule, in order to associate the <i>ResourceModel</i> with the corresponding database table in which its data will be stored.
RepresentationParseFunction	Once this transformation is done, one <i>RepresentationParseFunction</i> element of the Core PIM meta-model is added to the <i>ResourceModel</i> one, by calling the <i>createParseFunction</i> ATL rule, for every <i>InputRepresentation</i> the Core CIM input <i>Resource</i> has.
RepresentationMarshallingFunction	Once this transformation is done, one <i>RepresentationMarshallingFunction</i> element of the Core PIM meta-model is added to the <i>ResourceModel</i> one, by calling the <i>createMarshalFunction</i> ATL rule, for every <i>OutputRepresentation</i> the Core CIM input <i>Resource</i> has.

8.1.5 ResourceModelManager element introduction

The *createCRUDRMMManager* ATL rule introduces *ResourceModelManager* elements of the Core PIM meta-model, by taking as input a *Resource* element of the Core CIM meta-model. That is, *ResourceModelManagers* embed one part of the abstract design in the envisioned system's PIM for the corresponding concepts of a CIM *Resource* that has its *isAlgorithmic* property set to false. The properties and relations of the rule are defined in the table below:

Table 8-5 *createCRUDRMMManager* ATL rule

ResourceModelManager property/relation	Explanation
parentName	The <i>parentName</i> property of the <i>ResourceModelManager</i> is the name of the input Core CIM <i>Resource</i> .
name	The <i>name</i> property of the <i>ResourceModelManager</i> equals to the name of the input Core CIM <i>Resource</i> + "ModelManager"
ResourceInputRepresentation	Once this transformation is done, one <i>ResourceInputRepresentation</i> element of the Core PIM meta-model is added to the <i>ResourceModelManager</i> one, by calling the <i>addRMMManagerRepresentation</i> ATL rule, for every <i>InputRepresentation</i> the Core CIM input <i>Resource</i> has.
ResourceOutputRepresentation	Once this transformation is done, one <i>ResourceOutputRepresentation</i> element of the Core PIM meta-model is added to the <i>ResourceModelManager</i> one, by calling the <i>addRMMManagerOutputRepresentation</i> ATL rule, for every <i>OutputRepresentation</i> the Core CIM input <i>Resource</i> has.
ResourceModel	Once this transformation is done, one <i>ResourceModel</i> element of the Core PIM meta-model is added to the <i>ResourceModelManager</i> one, by calling the <i>createCRUDModel</i> ATL rule, in order to associate the <i>ResourceModelManager</i> with the <i>ResourceModels</i> that originate from from Core CIM <i>Resources</i> that are related of the Core CIM input <i>Resource</i> of this rule.
PIMComponentProperty	Once this transformation is done, one <i>PIMComponentProperty</i> element of the Core PIM meta-model is added to the <i>ResourceModel</i> one, by calling the <i>createLinkListProperty</i> ATL rule in order to add a link list property.
SetterFunction	Once this transformation is done, one <i>SetterFunction</i> element of the Core PIM meta-model is added to the <i>ResourceModel</i> one, by calling the <i>createLinkListPropertySetter</i> , in order to add a <i>SetterFunction</i> for the link list property.
GetterFunction	Once this transformation is done, one

	<i>GetterFunction</i> element of the Core PIM meta-model is added to the <i>ResourceModel</i> one, by calling the <i>createLinkListPropertyGetter</i> , in order to add a <i>GetterFunction</i> for the link list property to the <i>ResourceModel</i> .
RepresentationParseFunction	Once this transformation is done, one <i>RepresentationParseFunction</i> element of the Core PIM meta-model is added to the <i>ResourceModelManager</i> one, by calling the <i>createParseFunction</i> ATL rule, for every <i>InputRepresentation</i> the Core CIM input <i>Resource</i> has.
RepresentationMarshallingFunction	Once this transformation is done, one <i>RepresentationMarshallingFunction</i> element of the Core PIM meta-model is added to the <i>ResourceModelManager</i> one, by calling the <i>createMarshalFunction</i> ATL rule, for every <i>OutputRepresentation</i> the Core CIM input <i>Resource</i> has.

8.1.6 ResourceController element introduction

The *createRController* ATL rule introduces *ResourceController* elements of the Core PIM meta-model, by taking as input a *Resource* element of the Core CIM meta-model. That is, *ResourceControllers* embed one part of the abstract design in the envisioned system's PIM for the corresponding concepts of a CIM *Resource* that has its *isAlgorithmic* property set to false. The properties and relations of the rule are defined in the table below:

Table 8-6 createRController ATL rule

ResourceController property/relation	Explanation
parentName	The <i>parentName</i> property of the <i>ResourceController</i> is the name of the input Core CIM <i>Resource</i> .
name	The <i>name</i> property of the <i>ResourceController</i> equals to the name of the input Core CIM <i>Resource</i> + "Controller"
controllerURI	The <i>controllerURI</i> is calculated by calling the helper function <i>createRControllerURI</i>
ResourceModel	Once this transformation is done, one <i>ResourceModel</i> element of the Core PIM meta-model is added to the <i>ResourceController</i> one, by calling the <i>createCRUDRModel</i> ATL rule, in order to associate the <i>ResourceController</i> with the <i>ResourceModel</i> that both originate from the same Core CIM <i>Resource</i> .
ResourceControllerCRUDActivity	Once this transformation is done, one <i>ResourceControllerCRUDActivity</i> element of the Core PIM meta-model is added to the

	<i>ResourceController</i> one, by calling the <i>addControllerCRUDActivity</i> ATL rule, for every <i>CRUDActivity</i> of the Core CIM input <i>Resource</i> and that are of type “ <i>UPDATE</i> ”, “ <i>READ</i> ” or “ <i>DELETE</i> ”.
--	--

8.1.7 ResourceControllerManager element introduction

The *createRCManager* ATL rule introduces *ResourceControllerManager* elements of the Core PIM meta-model, by taking as input a *Resource* element of the Core CIM meta-model. That is, *ResourceControllerManagers* embed one part of the abstract design in the envisioned system’s PIM for the corresponding concepts of a CIM *Resource* that has its *isAlgorithmic* property set to false. The properties and relations of the rule are defined in the table below:

Table 8-7 CreateRCManager ATL rule

ResourceControllerManager property/relation	Explanation
parentName	The <i>parentName</i> property of the <i>ResourceControllerManager</i> is the name of the input Core CIM <i>Resource</i> .
name	The <i>name</i> property of the <i>ResourceControllerManager</i> equals to the name of the input Core CIM <i>Resource</i> + “ControllerManager”
controllerURI	The <i>controllerURI</i> is calculated by calling the helper function <i>createRCManagerURI</i>
ResourceModelManager	Once this transformation is done, one <i>ResourceModelManager</i> element of the Core PIM meta-model is added to the <i>ResourceControllerManager</i> one, by calling the <i>createCRUDRMMManager</i> ATL rule, in order to associate the <i>ResourceControllerManager</i> with the <i>ResourceModelManager</i> that both originate from the same Core CIM <i>Resource</i> .
ResourceControllerCRUDActivity	Once this transformation is done, one <i>ResourceControllerCRUDActivity</i> element of the Core PIM meta-model is added to the <i>ResourceControllerManager</i> one, by calling the <i>addControllerCRUDActivity</i> ATL rule, for every <i>CRUDActivity</i> of the Core CIM input <i>Resource</i> and that are of type “ <i>CREATE</i> ” or “ <i>READ</i> ”.

8.1.8 InputRepresentation element transformation (case 1)

The *addResourceModelRepresentation* ATL rule transforms *InputRepresentation* elements of the Core CIM meta-model, to *ResourceInputRepresentation* ones of the Core PIM meta-model. Therefore, the *ResourceInputRepresentation* embeds the abstract envisioned system’s design of the *InputRepresentation* CIM concept, which is modelled with the *InputRepresentation* element. This first case refers to *InputRepresentations* that are associated with Core CIM Resources, which have their

isAlgorithmic property set to false. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-8 addResourceModelRepresentation ATL rule

InputRepresentation property/relation	ResourceInputRepresentation property/relation	Explanation
inputMediaType	inputType	The <i>inputType</i> of the <i>ResourceInputRepresentation</i> is assigned the value of the <i>inputMediaType</i> of the source <i>InputRepresentation</i> element.

8.1.9 OutputRepresentation element transformation (case 1)

The *addResourceModelORespresentation* ATL rule transforms *OutputRepresentation* elements of the Core CIM meta-model, to *ResourceOutputRepresentation* ones of the Core PIM meta-model. Therefore, the *ResourceOutputRepresentation* embeds the abstract envisioned system's design of the *OutputRepresentation* CIM concept, which is modelled with the *OutputRepresentation* element. This first case refers to *OutputRepresentations* that are associated with Core CIM Resources, which have their *isAlgorithmic* property set to false. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-9 addRModelOutputRepresentation ATL rule

OutputRepresentation property/relation	ResourceOutputRepresentation property/relation	Explanation
outputMediaType	outputType	The <i>outputType</i> of the <i>ResourceOutputRepresentation</i> is assigned the value of the <i>outputMediaType</i> of the source <i>OutputRepresentation</i> element.

8.1.10 InputRepresentation element transformation (case 2)

The *addAlgoModelRepresentation* ATL rule transforms *InputRepresentation* elements of the Core CIM meta-model, to *ResourceInputRepresentation* ones of the Core PIM meta-model. Therefore, the *ResourceInputRepresentation* embeds the abstract envisioned system's design of the *InputRepresentation* CIM concept, which is modelled with the *InputRepresentation* element. This second case refers to *InputRepresentations* that are associated with Core CIM Resources, which have their *isAlgorithmic* property set to true. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-10 addAlgoModelRepresentation ATL rule

InputRepresentation property/relation	ResourceInputRepresentation property/relation	Explanation
inputMediaType	inputType	The <i>inputType</i> of the <i>ResourceInputRepresentation</i> is assigned the

		value of the <i>inputMediaType</i> of the source <i>InputRepresentation</i> element.
--	--	--

8.1.11 OutputRepresentation element transformation (case 2)

The *addAlgoModelORepresentation* ATL rule transforms *OutputRepresentation* elements of the Core CIM meta-model, to *ResourceOutputRepresentation* ones of the Core PIM meta-model. Therefore, the *ResourceOutputRepresentation* embeds the abstract envisioned system's design of the *OutputRepresentation* CIM concept, which is modelled with the *OutputRepresentation* element. This second case refers to *OutputRepresentations* that are associated with Core CIM Resources, which have their *isAlgorithmic* property set to true. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-11 *addAlgoModelORepresentation* ATL rule

OutputRepresentation property/relation	ResourceOutputRepresentation property/relation	Explanation
<i>outputMediaType</i>	<i>outputType</i>	The <i>outputType</i> of the <i>ResourceOutputRepresentation</i> is assigned the value of the <i>outputMediaType</i> of the source <i>OutputRepresentation</i> element.

8.1.12 InputRepresentation element transformation (case 3)

The *addAlgoModelIRepresentation* ATL rule transforms *InputRepresentation* elements of the Core CIM meta-model, to *ResourceInputRepresentation* ones of the Core PIM meta-model. Therefore, the *ResourceInputRepresentation* embeds the abstract envisioned system's design of the *InputRepresentation* CIM concept, which is modelled with the *InputRepresentation* element. This second case refers to *InputRepresentations* that are associated with Core CIM Resources, which have their *isAlgorithmic* property set to false, that will be associated with *ResourceModelManager* elements of the Core PIM meta-model. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-12 *addRMManagerIRepresentation* ATL rule

InputRepresentation property/relation	ResourceInputRepresentation property/relation	Explanation
<i>inputMediaType</i>	<i>inputType</i>	The <i>inputType</i> of the <i>ResourceInputRepresentation</i> is assigned the value of the <i>inputMediaType</i> of the source <i>InputRepresentation</i> element.

8.1.13 OutputRepresentation element transformation (case 3)

The *addAlgoModelORepresentation* ATL rule transforms *OutputRepresentation* elements of the Core CIM meta-model, to *ResourceOutputRepresentation* ones of the Core PIM meta-model. Therefore, the *ResourceOutputRepresentation* embeds the abstract envisioned system's design of the

OutputRepresentation CIM concept, which is modelled with the *OutputRepresentation* element. This second case refers to *OutputRepresentations* that are associated with Core CIM Resources, which have their *isAlgorithmic* property set to false, that will be associated with *ResourceModelManagers*. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-13 addRMManagerIRepresentation ATL rule

InputRepresentation property/relation	ResourceInputRepresentation property/relation	Explanation
outputMediaType	outputType	The <i>outputType</i> of the <i>ResourceOutputRepresentation</i> is assigned the value of the <i>outputMediaType</i> of the source <i>OutputRepresentation</i> element.

8.1.14 Property element transformation

The *addRModelProperties* ATL rule transforms *Property* elements of the Core CIM meta-model, to *PIMComponentProperty* ones of the Core PIM meta-model. Therefore, the *PIMComponentProperty* embeds the abstract envisioned system's design of the conceptual property modelled with the *Property* element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-14 addRModelProperties ATL rule

Property property/relation	PIMComponentProperty property/relation	Explanation
name	name	The <i>name</i> of the <i>PIMComponentProperty</i> equals the one of the <i>Property</i> element.
type	type	The <i>type</i> of the <i>PIMComponentProperty</i> remains the same with the one of the <i>Property</i> element.
isUnique	isUnique	The <i>isUnique</i> of the <i>PIMComponentProperty</i> remains the same with the one of the <i>Property</i> element.
isNamingProperty	isNamingProperty	The <i>isNamingProperty</i> of the <i>PIMComponentProperty</i> remains the same with the one of the <i>Property</i> element.
-	isPrimaryIdentifier	The <i>isPrimaryIdentifier</i> of the <i>PIMComponentProperty</i> is set to false for all the <i>PIMComponentProperties</i> that originate to a Core CIM <i>Property</i> source element.
-	RDBMSTableColumn	Once this transformation is done, one <i>RDBMSTableColumn</i> element of the Core PIM meta-model is added to the <i>PIMComponentProperty</i> one, by calling the <i>createRDBMSTableColumn</i> ATL rule, in order to associate the <i>PIMComponentProperty</i> with the <i>RDBMSTableColumn</i> that will store its data in the

		envisioned system's local database.
--	--	-------------------------------------

8.1.15 SetterFunction element introduction

The *addSetterFunction* ATL rule introduces *SetterFunction* elements of the Core PIM meta-model, by taking as input a *Property* element of the Core CIM meta-model. That is, *SetterFunctions* do not have Core CIM meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-15 addSetterFunction ATL rule

SetterFunction property/relation	Explanation
name	The name of the <i>SetterFunction</i> equals to "set" + the name of the Core CIM input <i>Property</i> of this rule.
PIMComponentProperty	Once this transformation is done, one <i>PIMComponentProperty</i> element of the Core PIM meta-model is added to the <i>SetterFunction</i> one, by calling the <i>addRModelProperties</i> ATL rule, in order to associate the <i>SetterFunction</i> with the <i>PIMComponentProperty</i> the runtime value of which it sets.
FunctionParameter	Once this transformation is done, one <i>FunctionParameter</i> element of the Core PIM meta-model is added to the <i>SetterFunction</i> one, by calling the <i>createFunctionParameter</i> ATL rule, in order to associate the <i>SetterFunction</i> with the <i>FunctionParameter</i> , which has the runtime value to which the <i>PIMComponentProperty</i> this <i>SetterFunction</i> updates, must be set.

8.1.16 FunctionParameter element introduction

The *createFunctionParameter* ATL rule introduces *FunctionParameter* elements of the Core PIM meta-model, by taking as input a *Property* element of the Core CIM meta-model. That is, *FunctionParameters* do not have Core CIM meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-16 createFunctionParameter ATL rule

FunctionParameter property/relation	Explanation
name	The <i>name</i> of the <i>FunctionParameter</i> equals the one of the <i>Property</i> element.
type	The <i>type</i> of the <i>Functionparameter</i> remains the same with the one of the <i>Property</i> element.

isUnique	The <i>isUnique</i> of the <i>FunctionParameter</i> remains the same with the one of the <i>Property</i> element.
bIsReturnType	The <i>bIsReturnType</i> is set to false, since <i>SetterFunctions</i> do not have a return type (void).

8.1.17 GetterFunction element introduction

The *addGetterFunction* ATL rule introduces *GetterFunction* elements of the Core PIM meta-model, by taking as input a *Property* element of the Core CIM meta-model. That is, *GetterFunctions* do not have Core CIM meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-17 addGetterFunction ATL rule

GetterFunction property/relation	Explanation
name	The name of the <i>GetterFunction</i> equals to “get” + the name of the Core CIM input <i>Property</i> of this rule.
PIMComponentProperty	Once this transformation is done, one <i>PIMComponentProperty</i> element of the Core PIM meta-model is added to the <i>GetterFunction</i> one, by calling the <i>addRModelProperties</i> ATL rule, in order to associate the <i>GetterFunction</i> with the <i>PIMComponentProperty</i> the runtime value of which it gets.
FunctionParameter	Once this transformation is done, one <i>FunctionParameter</i> element of the Core PIM meta-model is added to the <i>GetterFunction</i> one, by calling the <i>createFunctionReturnParameter</i> ATL rule, in order to associate the <i>GetterFunction</i> with the <i>FunctionParameter</i> , which has the runtime value to which the <i>PIMComponentProperty</i> this <i>GetterFunction</i> retrieves, is set.

8.1.18 RDBMSTable element introduction

The *createRDBMSTable* ATL rule introduces *RDBMSTable* elements of the Core PIM meta-model, by taking as input a *Resource* element of the Core CIM meta-model. That is, *RDBMSTables* embeds the abstract envisioned system’s design of the conceptual property modelled with the *Resource* element. The properties and relations of the rule are defined in the table below:

Table 8-18 createRDBMSTable ATL rule

RDBMSTable property/relation	Explanation
name	The <i>name</i> of the <i>RDBMSTable</i> is set equal to the name of the Core CIM input <i>Resource</i> of this rule.

RDBMSTableColumn	Once this transformation is done, one <i>RDBMSTableColumn</i> element of the Core PIM meta-model is added to the <i>RDBMSTable</i> one, by calling the <i>createRDBMSTableColumn</i> ATL rule, for every <i>Property</i> that the Core CIM input <i>Resource</i> has. One <i>RDBMSTableColumn</i> is also added by calling the <i>createRDBMSTablePrimaryKey</i> ATL rule in order to add to the <i>RDBMSTable</i> the needed primary key as well as one <i>RDBMSTableColumn</i> by calling the <i>createRDBMSTableForeignKey</i> ATL rule to add a foreign key for every related Core CIM <i>Resource</i> of the Core CIM input <i>Resource</i> of this rule.
------------------	--

8.1.19 RDBMSTableColumn element introduction (case 1)

The *createRDBMSTableColumn* ATL rule introduces *RDBMSTableColumn* elements of the Core PIM meta-model, by taking as input a *Property* element of the Core CIM meta-model. That is, *RDBMSTableColumns* do not have Core CIM meta-model counterparts. This first case refers to *RDBMSTableColumns* that store data of Core CIM *Properties*. The properties and relations of the rule are defined in the table below:

Table 8-19 *createRDBMSTableColumn* ATL rule

RDBMSTableColumn property/relation	Explanation
name	The <i>name</i> of the <i>RDBMSTableColumn</i> remains the same as the one of the <i>Property</i> .
type	The <i>type</i> of the <i>RDBMSTableColumn</i> remains the same as the one of the <i>Property</i> .
isForeignKey	The <i>isForeignKey</i> of the <i>RDBMSTableColumn</i> of this 1 st case is always set to false, since in this case columns cannot be foreign keys.
isPrimaryKey	The <i>isPrimaryKey</i> of the <i>RDBMSTableColumn</i> of this 1 nd case is always set to false, since in this case columns cannot be primary keys.

8.1.20 RDBMSTableColumn element transformation (case 2)

The *createRDBMSTablePrimaryKey* ATL rule introduces *RDBMSTableColumn* elements of the Core PIM meta-model, by taking as input a *Resource* element of the Core CIM meta-model. That is, *RDBMSTableColumns* do not have Core CIM meta-model counterparts. This second case refers to *RDBMSTableColumns* that will be primary keys. The properties and relations of the rule are defined in the table below:

Table 8-20 *createRDBMSTablePrimaryKey* ATL rule

RDBMSTableColumn	Explanation
------------------	-------------

property/relation	
name	The <i>name</i> of the <i>RDBMSTableColumn</i> equals to one of the Core CIM Resource + “Id”.
type	The <i>type</i> of the <i>RDBMSTableColumn</i> in this case is always set to “int”, since the produced database of the S-CASE envisioned systems have integers as primary keys.
isForeignKey	The <i>isForeignKey</i> of the <i>RDBMSTableColumn</i> of this 2 st case is always set to false, since in this case columns cannot be foreign keys.
isPrimaryKey	The <i>isPrimaryKey</i> of the <i>RDBMSTableColumn</i> of this 2 nd case is always set to true, since in this case columns are always primary keys.

8.1.21 RDBMSTableColumn element introduction (case 3)

The *createRDBMSTablePrimaryKey* ATL rule introduces *RDBMSTableColumn* elements of the Core PIM meta-model, by taking as input a *Resource* element of the Core CIM meta-model. That is, *RDBMSTableColumns* do not have Core CIM meta-model counterparts. This third case refers to *RDBMSTableColumns* that will be foreign keys. The properties and relations of the rule are defined in the table below:

Table 8-21 *createRDBMSTableForeignKey* ATL rule

RDBMSTableColumn property/relation	Explanation
name	The <i>name</i> of the <i>RDBMSTableColumn</i> equals to one of the Core CIM Resource + “Id”.
type	The <i>type</i> of the <i>RDBMSTableColumn</i> in this case is always set to “int”, since the produced database of the S-CASE envisioned systems have integers as primary keys and therefore as foreign keys as well.
isForeignKey	The <i>isForeignKey</i> of the <i>RDBMSTableColumn</i> of this 3 rd case is always set to true, since in this case columns are always foreign keys.
isPrimaryKey	The <i>isPrimaryKey</i> of the <i>RDBMSTableColumn</i> of this 3 rd case is always set to false, since in this case columns are never primary keys.

8.1.22 RepresentationParseFunction element introduction

The *createParseFunction* ATL rule introduces *RepresentationParseFunction* elements of the Core PIM meta-model, by taking as input an *InputRepresentation* element of the Core CIM meta-model. That is, *RepresentationParseFunctions* do not have Core CIM meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-22 createParseFunction ATL rule

RepresentationParseFunction property/relation	Explanation
ResourceInputRepresentation	Once this transformation is done, one <i>ResourceInputRepresentation</i> element of the Core PIM meta-model is added to the <i>RepresentationParseFunction</i> one. That <i>ResourceInputRepresentation</i> is to be parsed by this <i>RepresentationParseFunction</i> .

8.1.23 RepresentationMarshallingFunction element introduction

The *createMarshalFunction* ATL rule introduces *RepresentationMarshallingFunction* elements of the Core PIM meta-model, by taking as input a *ResourceOutputRepresentation* element of the Core CIM meta-model. That is, *RepresentationMarshallingFunctions* do not have Core CIM meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-23 createMarshalFunction ATL rule

RepresentationMarshallingFunction property/relation	Explanation
ResourceOutputRepresentation	Once this transformation is done, one <i>ResourceOutputRepresentation</i> element of the Core PIM meta-model is added to the <i>RepresentationMarshallingFunction</i> one. That <i>ResourceOutputRepresentation</i> is to be marshalled by this <i>RepresentationMarshallingFunction</i> .

8.1.24 PIMComponentProperty element introduction (case 2)

The *createRModelPrimaryIdentifier* ATL rule introduces *PIMComponentProperty* elements of the Core PIM meta-model, by taking as input a *Resource* element of the Core CIM meta-model. In this 2nd case, *PIMComponentProperties* do not have Core CIM meta-model counterparts, since they model a property that does not originate from any CIM *Property*. The properties and relations of the rule are defined in the table below:

Table 8-24 createRModelPrimaryIdentifier ATL rule

PIMComponentProperty property/relation	Explanation
name	The <i>name</i> of the this case's <i>PIMComponentProperty</i> equals to the name of the CIM input <i>Resource</i> + "Id"
type	The <i>type</i> of the this case's <i>PIMComponentProperty</i> is always set to "int"

isUnique	The multiplicity of this case's <i>PIMComponentProperty</i> is always set to one, therefore the <i>isUnique</i> property is set to true.
isNamingProperty	The <i>isNamingProperty</i> property of this case's <i>PIMComponentProperty</i> is always set to false.
isPrimaryIdentifier	The <i>isPrimaryIdentifier</i> property of this case's <i>PIMComponentProperty</i> is always set to true.

8.1.25 SetterFunction element introduction (case 2)

The *createRModelPrimaryIdSetter* ATL rule introduces *SetterFunction* elements of the Core PIM meta-model, by taking as input a *Resource* element of the Core CIM meta-model. In this case, the *SetterFunction* does not update the value of a CIM *Property* but the value of the primary identifier *PIMComponentProperty* that does not have *CIM* meta-model counterpart. The properties and relations of the rule are defined in the table below:

Table 8-25 *createRModelPrimaryIdSetter* ATL rule

SetterFunction property/relation	Explanation
name	The <i>name</i> of the <i>SetterFunction</i> in this case is set to equal "set" + the name of the CIM input <i>Resource</i> + "Id"
PIMComponentProperty	Once this transformation is done, one <i>PIMComponentProperty</i> element of the Core PIM meta-model is added to the <i>SetterFunction</i> one, in order to associate the <i>SetterFunction</i> with the primary identifier <i>PIMComponentProperty</i> it sets.

8.1.26 GetterFunction element introduction (case 2)

The *createRModelPrimaryIdGetter* ATL rule introduces *GetterFunction* elements of the Core PIM meta-model, by taking as input a *Resource* element of the Core CIM meta-model. In this case, the *GetterFunction* does not retrieve the value of a CIM *Property* but the value of the primary identifier *PIMComponentProperty* that does not have *CIM* meta-model counterpart. The properties and relations of the rule are defined in the table below:

Table 8-26 *createRModelPrimaryIdGetter* ATL rule

GetterFunction property/relation	Explanation
name	The <i>name</i> of the <i>GetterFunction</i> in this case is set to equal "get" + the name of the CIM input <i>Resource</i> + "Id"
PIMComponentProperty	Once this transformation is done, one <i>PIMComponentProperty</i> element of the Core PIM meta-model is added to the <i>GetterFunction</i> one,

	in order to associate the <i>GetterFunction</i> with the primary identifier <i>PIMComponentProperty</i> it retrieves.
--	---

8.1.27 PIMComponentProperty element introduction (case 3)

The *createLinkListProperty* ATL rule introduces *PIMComponentProperty* elements of the Core PIM meta-model. In this 3rd case the *PIMComponentProperties* model link lists of hypermedia links. That is, in this case *PIMComponentProperties* do not have Core CIM meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-27 createLinkListProperty ATL rule

PIMComponentProperty property/relation	Explanation
name	In this case the <i>name</i> of the <i>PIMComponentProperty</i> is always set to “linklist”
type	In this case the <i>type</i> of the <i>PIMComponentProperty</i> is always set to <i>HypermediaLink</i> .
isUnique	In this case the <i>isUnique</i> property of the <i>PIMComponentProperty</i> is always set to false, since this <i>PIMComponentProperty</i> is intended to hold many hypermedia links.

8.1.28 SetterFunction element introduction (case 3)

The *createLinkListPropertySetter* ATL rule introduces *SetterFunction* elements of the Core PIM meta-model. In this case the *SetterFunction* sets the value of the *linklist PIMComponentProperty*. That is, in this case the *SetterFunctions* do not have Core CIM meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-28 createLinkListPropertySetter ATL rule

SetterFunction property/relation	Explanation
name	In this case the <i>name</i> of the <i>SetterFunction</i> is set to “setlinklist”
PIMComponentProperty	Once this transformation is done, one <i>PIMComponentProperty</i> element of the Core PIM meta-model is added to the <i>SetterFunction</i> one, in order to associate the <i>SetterFunction</i> with the <i>linklist PIMComponentProperty</i> it sets.
FunctionParameter	Once this transformation is done, one <i>FunctionParameter</i> element of the Core PIM meta-model is added to the <i>SetterFunction</i> one, in order to associate the <i>SetterFunction</i> with its

	input function parameter that holds the value of the <i>PIMComponentProperty</i> to be updated.
--	---

8.1.29 GetterFunction element introduction (case 3)

The *createLinkListPropertyGetter* ATL rule introduces *GetterFunction* elements of the Core PIM meta-model. In this case the *GetterFunction* retrieves the value of the *linklist PIMComponentProperty*. That is, in this case the *GetterFunctions* do not have Core CIM meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-29 *createLinkListPropertyGetter* ATL rule

GetterFunction property/relation	Explanation
name	In this case the <i>name</i> of the <i>GetterFunction</i> is set to “getlinklist”
PIMComponentProperty	Once this transformation is done, one <i>PIMComponentProperty</i> element of the Core PIM meta-model is added to the <i>GetterFunction</i> one, in order to associate the <i>GetterFunction</i> with the <i>linklist PIMComponentProperty</i> it retrieves.
FunctionParameter	Once this transformation is done, one <i>FunctionParameter</i> element of the Core PIM meta-model is added to the <i>GetterFunction</i> one, in order to associate the <i>GetterFunction</i> with its input function parameter that holds the value of the <i>PIMComponentProperty</i> to be retrieved.

8.1.30 ResourceControllerCRUDActivity element introduction (case 1)

The *createAlgoControllerCRUDActivity* ATL rule introduces *ResourceControllerCRUDActivity* elements of the Core PIM meta-model, by taking as input a *Resource* element of the Core CIM meta-model. Therefore, the *ResourceControllerCRUDActivity* embeds the abstract envisioned system’s design of the *CRUDActivity* CIM concept. In this case the *ResourceControllerCRUDActivities* introduced will be associated with *AlgoResourceControllers* of the Core PIM meta-model and therefore they are the counterparts of the unique Core CIM input *Resources’ CRUDActivities* that have their *isAlgorithmic* property set to true. The properties and relations of the rule are defined in the table below:

Table 8-30 *createAlgoControllerCRUDActivity* ATL rule

ResourceControllerCRUDActivity property/relation	Explanation
name	The <i>name</i> of the <i>ResourceControllerCRUDActivity</i> is set to equal the <i>crudVerb</i> of it + the name of the CIM input <i>Resource</i> .
activityURI	In this case the relative <i>activityURI</i> is set always to “/”

crudVerb	The <i>crudVerb</i> of the <i>ResourceControllerCRUDActivity</i> is the same with the <i>crudVerb</i> with the one of the unique <i>CRUDActivity</i> the Core CIM input <i>Resource</i> has.
CRUDActivityHandler	Once this transformation is done, one <i>CRUDActivityHandler</i> element of the Core PIM meta-model is added to the <i>ResourceControllerCRUDActivity</i> one, in order to associate the <i>ResourceControllerCRUDActivity</i> with the <i>CRUDActivityHandler</i> to which the client request will be forwarded.

8.1.31 ResourceControllerCRUDActivity element introduction (case 2)

The *addRControllerCRUDActivity* ATL rule introduces *ResourceControllerCRUDActivity* elements of the Core PIM meta-model, by taking as input data of a *Resource CRUDActivity* element of the Core CIM meta-model. Therefore, the *ResourceControllerCRUDActivity* embeds the abstract envisioned system's design of the *CRUDActivity* CIM concept. In this case the *ResourceControllerCRUDActivities* introduced will be associated with *ResourceControllers* of the Core PIM meta-model and therefore they are the counterparts of the Core CIM input *Resources' CRUDActivities* that have their *isAlgorithmic* property set to false. The properties and relations of the rule are defined in the table below:

Table 8-31 addRControllerCRUDActivity ATL rule

ResourceControllerCRUDActivity property/relation	Explanation
name	The <i>name</i> of the <i>ResourceControllerCRUDActivity</i> is set to equal the <i>crudVerb</i> of it + the name of the CIM input <i>Resource</i> if that input <i>Resource</i> is related of at most one other <i>Resources</i> . Otherwise, it is set to equal the <i>crudVerb</i> of it + the name of a resource of which it is related + the name of the CIM input <i>Resource</i> .
activityURI	In this case the relative <i>activityURI</i> is set usually to “/{" + the name of the CIM input <i>Resource</i> + “Id}”.
crudVerb	The <i>crudVerb</i> of the <i>ResourceControllerCRUDActivity</i> is the same with the <i>crudVerb</i> with the one of the unique <i>CRUDActivity</i> the Core CIM input <i>Resource</i> has.
CRUDActivityHandler	Once this transformation is done, one <i>CRUDActivityHandler</i> element of the Core PIM meta-model is added to the <i>ResourceControllerCRUDActivity</i> one, in order to associate the <i>ResourceControllerCRUDActivity</i> with the <i>CRUDActivityHandler</i> to which the client request will be forwarded.

8.1.32 ResourceControllerCRUDActivity element introduction (case 3)

The *addRCManagerCRUDActivity* ATL rule introduces *ResourceControllerCRUDActivity* elements of the Core PIM meta-model, by taking as input data of a *Resource CRUDActivity* element of the Core CIM meta-model. Therefore, the *ResourceControllerCRUDActivity* embeds the abstract envisioned system's design of the *CRUDActivity* CIM concept. In this case the *ResourceControllerCRUDActivities* introduced will be associated with *ResourceControllerManagers* of the Core PIM meta-model and therefore they are the counterparts of the Core CIM input *Resources' CRUDActivities* that have their *isAlgorithmic* property set to false. The properties and relations of the rule are defined in the table below:

Table 8-32 addRCManagerCRUDActivity ATL rule

ResourceControllerCRUDActivity property/relation	Explanation
name	The <i>name</i> of the <i>ResourceControllerCRUDActivity</i> is set to equal the <i>crudVerb</i> of it + the name of the CIM input <i>Resource</i> if that input <i>Resource</i> is related of at most one other <i>Resources</i> . Otherwise, it is set to equal the <i>crudVerb</i> of it + the name of a resource of which it is related + the name of the CIM input <i>Resource</i> .
activityURI	In this case the relative <i>activityURI</i> is set always to "/".
crudVerb	The <i>crudVerb</i> of the <i>ResourceControllerCRUDActivity</i> is the same with the <i>crudVerb</i> with the one of the unique <i>CRUDActivity</i> the Core CIM input <i>Resource</i> has.
CRUDActivityHandler	Once this transformation is done, one <i>CRUDActivityHandler</i> element of the Core PIM meta-model is added to the <i>ResourceControllerCRUDActivity</i> one, in order to associate the <i>ResourceControllerCRUDActivity</i> with the <i>CRUDActivityHandler</i> to which the client request will be forwarded.

8.1.33 CRUDActivityHandler element introduction (case 1)

The *createCRUDActivityHandler* ATL rule introduces *CRUDActivityHandler* elements of the Core PIM meta-model, by taking as input data of a *CRUDActivity* element of the Core CIM meta-model. Therefore, the *ResourceCRUDActivityHandler* embeds a part of the abstract envisioned system's design of the *CRUDActivity* CIM concept. In this case the *ResourceControllerCRUDActivityHandlers* introduced will be associated with *ResourceControllerCRUDActivities* of either *AlgoResourceControllers* or *ResourceControllers*. The properties and relations of the rule are defined in the table below:

Table 8-33 createCRUDActivityHandler ATL rule

CRUDActivityHandler	Explanation
---------------------	-------------

property/relation	
name	The <i>name</i> of the <i>CRUDActivityHandler</i> is set to equal the <i>crudVerb</i> of it + the name of the CIM input <i>Resource</i> if that input <i>Resource</i> is related of at most one other <i>Resources</i> + “Handler”. Otherwise, it is set to equal the <i>crudVerb</i> of it + the name of a resource of which it is related + the name of the CIM input <i>Resource</i> + “Handler”.
crudVerb	The <i>crudVerb</i> of the <i>CRUDActivityHandler</i> is the same with the <i>crudVerb</i> with the corresponding <i>CRUDActivity</i> the Core CIM input <i>CRUDActivity</i> has.
DatabaseController	Once this transformation is done, one <i>DatabaseController</i> element of the Core PIM meta-model is added to the <i>CRUDActivityHandler</i> one, by calling the <i>createDatabaseController</i> ATL rule, in order to associate the <i>CRUDActivityHandler</i> with the <i>DatabaseController</i> to which the service data is stored.
CreateHypermediaPIMFunction	Once this transformation is done, one <i>CreateHypermediaPIMFunction</i> element of the Core PIM meta-model is added to the <i>CRUDActivityHandler</i> one, in order to associate the <i>CRUDActivityHandler</i> with the <i>CreateHypermediaPIMFunction</i> , which will generate any hypermedia links to be sent to the client.

8.1.34 **CRUDActivityHandler element introduction (case 2)**

The *createRCManagerCRUDActivityHandler* ATL rule introduces *CRUDActivityHandler* elements of the Core PIM meta-model, by taking as input data of a *CRUDActivity* element of the Core CIM meta-model. Therefore, the *ResourceCRUDActivityHandler* embeds a part of the abstract envisioned system’s design of the *CRUDActivity* CIM concept. In this case the *ResourceControllerCRUDActivityHandlers* introduced will be associated with *ResourceControllerCRUDActivities*, which are associated with *ResourceControllerManagers*. The properties and relations of the rule are defined in the table below:

Table 8-34 createRCManagerCRUDActivityHandler ATL rule

CRUDActivityHandler property/relation	Explanation
name	The <i>name</i> of the <i>CRUDActivityHandler</i> is set to equal the <i>crudVerb</i> of it + the name of the CIM input <i>Resource</i> if that input <i>Resource</i> is related of at most one other <i>Resources</i> + “Handler”. Otherwise, it is set to equal the <i>crudVerb</i> of it + the name of a resource of which it is related + the name of the CIM input <i>Resource</i> + “Handler”.

crudVerb	The <i>crudVerb</i> of the <i>CRUDActivityHandler</i> is the same with the <i>crudVerb</i> with the corresponding <i>CRUDActivity</i> the Core CIM input <i>CRUDActivity</i> has.
DatabaseController	Once this transformation is done, one <i>DatabaseController</i> element of the Core PIM meta-model is added to the <i>CRUDActivityHandler</i> one, by calling the <i>createDatabaseController</i> ATL rule, in order to associate the <i>CRUDActivityHandler</i> with the <i>DatabaseController</i> to which the service data is stored.
CreateHypermediaPIMFunction	Once this transformation is done, one <i>CreateHypermediaPIMFunction</i> element of the Core PIM meta-model is added to the <i>CRUDActivityHandler</i> one, in order to associate the <i>CRUDActivityHandler</i> with the <i>CreateHypermediaPIMFunction</i> , which will generate any hypermedia links to be sent to the client.

8.1.35 DatabaseController element introduction

The *createDatabaseController* ATL rule introduces *DatabaseController* elements of the Core PIM meta-model, by taking as input a *RESTfulServiceCIM* element of the Core CIM meta-model. Therefore, the *DatabaseController* embeds a part of the abstract envisioned system's design of the *RESTfulServiceCIM* CIM concept. The properties and relations of the rule are defined in the table below:

Table 8-35 *createDatabaseController* ATL rule

DatabaseController property/relation	Explanation
name	The <i>name</i> of the <i>DatabaseController</i> is always set to "DatabaseController"
RDBMSActivity	Once this transformation is done, one <i>RDBMSActivity</i> element of the Core PIM meta-model is added to the <i>DatabaseController</i> one, in order to model all the possible input/output interactions it will have with the envisioned system's local database.

8.1.36 RDBMSActivity element introduction

The *createRDBMSActivity* ATL rule introduces *RDBMSActivity* elements of the Core PIM meta-model. The *RDBMSActivities* do not have Core CIM meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-36 createRDBMSActivity ATL rule

RDBMSActivity property/relation	Explanation
name	The <i>name</i> of the <i>RDBMSActivity</i> equals the <i>name</i> of the overlying <i>ResourceControllerCRUDActivity</i> .
rbmsVerb	If the overlying <i>ResourceControllerCRUDActivity</i> 's <i>crudVerb</i> is <i>CREATE</i> then the <i>rbmsVerb</i> is <i>INSERT</i> . If it is <i>READ</i> then the <i>rbmsVerb</i> is <i>SELECT</i> . If it is <i>UPDATE</i> then the <i>rbmsVerb</i> is <i>UPDATE</i> and finally if the <i>crudVerb</i> is <i>DELETE</i> the <i>rbmsVerb</i> is <i>DELETE</i> .
RDBMSTable	Once this transformation is done, one <i>RDBMSTable</i> element of the Core PIM meta-model is added to the <i>RDBMSActivity</i> one, in order to associate it with the <i>RDBMSTable</i> that it will access.

8.1.37 CreateHypermediaPIMFunction element introduction

The *createAlgoResourceHypermediaFunction* ATL rule introduces *CreateHypermediaPIMFunction* elements of the Core PIM meta-model, by taking as input a *Resource* element of the Core CIM meta-model. That is, *AlgoResourceModels* do not have Core CIM meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-37 createAlgoResourceHypermediaFunction ATL rule

CreateHypermediaPIMFunction property/relation	Explanation
HypermediaLink	Once this transformation is done, several <i>HypermediaLink</i> elements of the Core PIM meta-model are added to the <i>CreateHypermediaPIMFunction</i> one, in order to associate it with the <i>HypermediaLinks</i> that the envision system will send to its client.

8.1.38 HypermediaLink element introduction

The *createHypermediaLink* ATL rule introduces *HypermediaLink* elements of the Core PIM meta-model. The *HypermediaLinks* do not have Core CIM meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-38 createHypermediaLink ATL rule

HypermediaLink property/relation	Explanation
linkCRUDVerb	The <i>linkCRUDVerb</i> is set according to the <i>crudVerb</i> of the target <i>ResourceControllerCRUDActivity</i> to which this <i>HypermediaLink</i> links.
linkType	The <i>linkType</i> is set according to the target <i>ResourceControllerCRUDActivity</i> to which this <i>HypermediaLink</i> links.
AlgoController / ResourceController / ResourceControllerManager	Once this transformation is done, one <i>AlgoController</i> or one <i>ResourceController</i> or one <i>ResourceControllerManager</i> element of the Core PIM meta-model is added to the <i>HypermediaLink</i> according to the case, in order to associate it with the <i>Controller</i> to which this <i>HypermediaLink</i> links.

8.2 Authentication Extension CIM to PIM ATL Rules

8.2.1 AnnotationModel element transformation

The *CIMToPIMAAuthentication* ATL rule transforms *AnnotationModel* elements of the *Authentication* CIM extension meta-model, to *AnnotationModel* ones of the *Authentication* PIM extension meta-model. Therefore, the *AnnotationModel* of the *Authentication* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *AnnotationModel* of the *Authentication* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-39 CIMtoPIMAAuthentication ATL rule

AnnotationModel property/relation	AnnotationModel property/relation	Explanation
name	name	The <i>name</i> of the <i>AnnotationModel</i> of the <i>Authentication</i> PIM extension equals the name of the <i>AnnotationModel</i> of the <i>Authentication</i> CIM extension.
AnnotatedElement	AnnotatedElement	Every <i>AnnotatedElement</i> of the <i>Authentication</i> CIM extension meta-model is transformed to its <i>Authentication</i> PIM extension meta-model counterpart by calling once the <i>createPIMAnnCRUDActivity</i> ATL rule for every <i>Authentication</i> CIM extension <i>AnnCRUDActivity</i> , once the <i>createPIMAnnCRUDActivityHandler</i> ATL rule for every <i>Authentication</i> CIM extension <i>AnnCRUDActivityHandler</i> and once the <i>createAnnDatabaseController</i> ATL rule.

Annotation	Annotation	Every <i>Annotation</i> of the <i>Authentication</i> CIM extension meta-model is transformed to its <i>Authentication</i> PIM extension meta-model counterpart by calling once the <i>createGuestMode</i> ATL rule for every <i>Authentication</i> CIM extension <i>GuestMode</i> , once the <i>createAuthenticationOnlyMode</i> ATL rule for every <i>Authentication</i> CIM extension <i>AuthenticationMode</i> , once the <i>createBothMode</i> ATL rule for every <i>Authentication</i> CIM extension <i>BothMode</i> and once the <i>createAuthenticationPerformer</i> ATL rule.
------------	------------	---

8.2.2 AnnCRUDActivity element transformation

The *createPIMAnnCRUDActivity* ATL rule transforms *AnnCRUDActivity* elements of the *Authentication* CIM extension meta-model, to *AnnCRUDActivity* ones of the *Authentication* PIM extension meta-model. Therefore, the *AnnCRUDActivity* of the *Authentication* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *AnnCRUDActivity* of the *Authentication* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-40 *createPIMAnnCRUDActivity* ATL rule

AnnCRUDActivity property/relation	AnnCRUDActivity property/relation	Explanation
name	name	The <i>name</i> property of the <i>Authentication</i> PIM extension <i>AnnCRUDActivity</i> equals to the name of the <i>Authentication</i> CIM extension <i>AnnCRUDActivity</i> + the name of the Core CIM <i>Resource</i> it belongs to.
CRUDActivity	ResourceControllerCRUDActivity	This ATL rule transforms any references of the <i>Authentication</i> CIM extension <i>AnnCRUDActivities</i> to Core CIM <i>CRUDActivities</i> to <i>Authentication</i> PIM extension <i>AnnCRUDActivities</i> that reference the corresponding Core PIM <i>ResourceControllerCRUDActivities</i> .

8.2.3 AnnCRUDActivityhandler element introduction

The *createPIMAnnCRUDActivityHandler* ATL rule introduces *AnnCRUDActivityHandler* elements of the *Authentication* PIM extension meta-model, by taking as input an *AnnCRUDActivity* element of the *Authentication* CIM extension meta-model. That is, *AnnCRUDActivityHandlers* do not have Core CIM meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-41 createPIMAnnCRUDActivityHandler ATL rule

AnnCRUDActivityHandler property/relation	Explanation
name	The <i>name</i> of the <i>AnnCRUDActivityHandler</i> is set equal to the one of the <i>Authentication</i> CIM input <i>AnnCRUDActivity</i> + “Handler”
CRUDActivityHandler	Once this transformation is done, one reference to the annotated <i>CRUDActivityHandler</i> element of the Core PIM meta-model is added to the <i>AnnCRUDActivityHandler</i> .

8.2.4 AnnDatabaseController element introduction

The *createAnnDatabaseController* ATL rule introduces *AnnDatabaseController* elements of the *Authentication* PIM extension meta-model. *AnnDatabaseControllers* do not have *Authentication* CIM extension meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-42 createAnnDatabaseController ATL rule

AnnDatabaseController property/relation	Explanation
DatabaseController	Once this transformation is done, one reference to the annotated <i>DatabaseController</i> element of the Core PIM meta-model is added to the <i>AnnDatabaseController</i> .

8.2.5 AuthenticationPerformer element introduction

The *createAuthenticationPerformer* ATL rule introduces *AuthenticationPerformer* elements of the *Authentication* PIM extension meta-model, by taking as input an *AuthenticationModel* element of the *Authentication* CIM extension meta-model. *AuthenticationPerformers* do not have Core CIM meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-43 createAuthenticationPerformer ATL rule

AuthenticationPerformer property/relation	Explanation
authenticationmodelName	The <i>authenticationmodelName</i> of the <i>AuthenticationPerformer</i> is set equal to the name of the annotated Core PIM <i>ResourceModel</i> that is used as authentication model.
authenticationModelParentName	The <i>authenticationmodelName</i> of the <i>authenticationperformer</i> is set equal to the name of the Core CIM <i>Resource</i> that is used as authentication model.

AnnDatabaseController	Once this transformation is done, one association to the <i>AnnDatabaseController</i> element of the <i>Authentication</i> PIM extension meta-model is added to the <i>AuthenticationPerformer</i> , by calling the <i>createAnnDatabaseController</i> ATL rule.
AuthenticationToken	Once this transformation is done, two <i>AuthenticationToken</i> elements of the Core PIM meta-model are added to the <i>AuthenticationPerformer</i> , by calling once the <i>createPIMAAuthenticationToken</i> and once the <i>createPIMAAuthenticationPasswordToken</i> ATL rule.

8.2.6 AuthenticationToken element transformation

The *createPIMAAuthenticationToken* ATL rule transforms *AuthenticationToken* elements of the *Authentication* CIM extension meta-model, to *AuthenticationToken* ones of the *Authentication* PIM extension meta-model. Therefore, the *AuthenticationToken* of the *Authentication* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *AuthenticationToken* of the *Authentication* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-44 createPIMAAuthenticationToken ATL rule

AuthenticationToken property/relation	AuthenticationToken property/relation	Explanation
name	name	The <i>name</i> of the <i>AuthenticationToken</i> of the <i>Authentication</i> PIM extension meta-model is set equal to the <i>name</i> of the <i>Authentication</i> CIM extension meta-model <i>AuthenticationToken</i> .
type	type	The <i>type</i> of the <i>AuthenticationToken</i> of the <i>Authentication</i> PIM extension meta-model is set equal to the <i>type</i> of the <i>Authentication</i> CIM extension meta-model <i>AuthenticationToken</i> .
isUnique	isUnique	The <i>isUnique</i> property of the <i>AuthenticationToken</i> of the <i>Authentication</i> PIM extension meta-model is set equal to the <i>isUnique</i> property of the <i>Authentication</i> CIM extension meta-model <i>AuthenticationToken</i> .

8.2.7 Password element transformation

The *createPIMAAuthenticationPasswordToken* ATL rule transforms *Password* elements of the *Authentication* CIM extension meta-model, to *Password* ones of the *Authentication* PIM extension meta-model. Therefore, the *Password* of the *Authentication* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *Password* of the *Authentication* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-45 createPIMAuthenticationPasswordToken ATL rule

Password property/relation	Password property/relation	Explanation
name	name	The <i>name</i> of the <i>Password</i> of the <i>Authentication</i> PIM extension meta-model is set equal to the <i>name</i> of the <i>Authentication</i> CIM extension meta-model <i>Password</i> .
type	type	The <i>type</i> of the <i>Password</i> of the <i>Authentication</i> PIM extension meta-model is set equal to the <i>type</i> of the <i>Authentication</i> CIM extension meta-model <i>Password</i> .
isUnique	isUnique	The <i>isUnique</i> property of the <i>Password</i> of the <i>Authentication</i> PIM extension meta-model is set equal to the <i>isUnique</i> property of the <i>Authentication</i> CIM extension meta-model <i>Password</i> .

8.2.8 GuestMode element transformation

The *createGuestMode* ATL rule transforms *GuestMode* elements of the *Authentication* CIM extension meta-model, to *GuestMode* ones of the *Authentication* PIM extension meta-model. Therefore, the *GuestMode* of the *Authentication* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *GuestMode* of the *Authentication* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-46 createGuestMode ATL rule

GuestMode property/relation	GuestMode property/relation	Explanation
AnnCRUDActivity	AnnResourceCRUDActivity	Once this transformation is done, one <i>AnnResourceCRUDActivity</i> element of the Core PIM meta-model is added to the <i>GuestMode</i> , by calling once the <i>createPIMAnnCRUDActivity</i> for every <i>Authentication</i> CIM extension meta-model input <i>AnnCRUDActivity</i> .
AnnCRUDActivity	AnnCRUDActivityHandler	Once this transformation is done, one <i>AnnCRUDActivityHandler</i> element of the Core PIM meta-model is added to the <i>GuestMode</i> , by calling once the <i>createPIMAnnCRUDActivityHandler</i> for every <i>Authentication</i> CIM extension meta-model input <i>AnnCRUDActivity</i> .

8.2.9 AuthenticationOnlyMode element transformation

The *createAuthenticationOnlyMode* ATL rule transforms *AuthenticationOnlyMode* elements of the *Authentication* CIM extension meta-model, to *AuthenticationOnlyMode* ones of the *Authentication* PIM extension meta-model. Therefore, the *AuthenticationOnlyMode* of the *Authentication* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *AuthenticationOnlyMode* of the *Authentication* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-47 *createAuthenticationOnlyMode* ATL rule

AuthenticationOnlyMode property/relation	AuthenticationOnlyMode property/relation	Explanation
AnnCRUDActivity	AnnResourceCRUDActivity	Once this transformation is done, one <i>AnnResourceCRUDActivity</i> element of the Core PIM meta-model is added to the <i>AuthenticationOnlyMode</i> , by calling once the <i>createPIMAnnCRUDActivity</i> for every <i>Authentication</i> CIM extension meta-model input <i>AnnCRUDActivity</i> .
AnnCRUDActivity	AnnCRUDActivityHandler	Once this transformation is done, one <i>AnnCRUDActivityHandler</i> element of the Core PIM meta-model is added to the <i>AuthenticationOnlyMode</i> , by calling once the <i>createPIMAnnCRUDActivityHandler</i> for every <i>Authentication</i> CIM extension meta-model input <i>AnnCRUDActivity</i> .

8.2.10 BothMode element transformation

The *createBothMode* ATL rule transforms *BothMode* elements of the *Authentication* CIM extension meta-model, to *BothMode* ones of the *Authentication* PIM extension meta-model. Therefore, the *BothMode* of the *Authentication* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *BothMode* of the *Authentication* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-48 *createBothMode* ATL rule

BothMode property/relation	BothMode property/relation	Explanation
AnnCRUDActivity	AnnResourceCRUDActivity	Once this transformation is done, one <i>AnnResourceCRUDActivity</i> element of the Core PIM meta-model is added to the <i>BothMode</i> , by calling once the <i>createPIMAnnCRUDActivity</i> for every <i>Authentication</i> CIM extension meta-model input <i>AnnCRUDActivity</i> .
AnnCRUDActivity	AnnCRUDActivityHandler	Once this transformation is done, one <i>AnnCRUDActivityHandler</i> element of the Core

		PIM meta-model is added to the <i>BothMode</i> , by calling once the <i>createPIMAnnCRUDActivityHandler</i> for every <i>Authentication</i> CIM extension meta-model input <i>AnnCRUDActivity</i> .
--	--	---

8.3 Database Searching Extension CIM to PIM ATL Rules

8.3.1 AnnotationModel element transformation

The *SearchLayerCIMToPIM* ATL rule transforms *AnnotationModel* elements of the *Database Searching* CIM extension meta-model, to *AnnotationModel* ones of the *Database Searching* PIM extension meta-model. Therefore, the *AnnotationModel* of the *Database Searching* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *AnnotationModel* of the *Database Searching* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-49 SearchLayerCIMToPIM ATL rule

AnnotationModel property/relation	AnnotationModel property/relation	Explanation
name	name	The <i>name</i> of the <i>AnnotationModel</i> of the <i>Database Searching</i> PIM extension is the same with the name of the <i>Database Searching</i> CIM extension <i>AnnotationModel</i> .
AnnotatedElement	AnnotatedElement	Every <i>AnnotatedElement</i> of the <i>Database Searching</i> CIM extension meta-model is transformed to its <i>Database Searching</i> PIM extension meta-model counterpart by calling either the <i>createAnnPIMComponentProperty</i> ATL rule for every <i>Database Searching</i> CIM extension <i>AnnProperty</i> , or the <i>createAnnAlgoResourceController</i> ATL rule for every <i>Database Searching</i> CIM extension <i>AnnAlgoResource</i> , or the <i>createAnnCRUDActivity</i> and <i>createAnnCRUDActivityHandler</i> one for every <i>AnnAlgoResource</i> and the <i>createAnnResourceModel</i> ATL rule for every <i>Database Searching</i> CIM extension <i>AnnCRUDResource</i> .
Annotation	Annotation	Every <i>Annotation</i> of the <i>Database Searching</i> CIM extension meta-model is transformed to its <i>Database Searching</i> PIM extension meta-model counterpart by calling once the <i>createPIMSearchController</i> ATL rule, the <i>createPIMSearchCRUDActivity</i> as well as the <i>createPIMSearchCRUDActivityHandler</i> one for every <i>Database Searching</i> CIM extension <i>SearchResource</i> .

8.3.2 AnnPIMComponentProperty element transformation

The *createAnnPIMComponentProperty* ATL rule transforms *AnnProperty* elements of the *Database Searching* CIM extension meta-model, to *AnnPIMComponentProperty* ones of the *Database Searching* PIM extension meta-model. Therefore, the *AnnPIMComponentProperty* of the *Database Searching* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *AnnProperty* of the *Database Searching* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-50 createAnnPIMComponentProperty ATL rule

AnnProperty property/relation	AnnPIMComponentProperty property/relation	Explanation
Property	PIMComponentProperty	This ATL rule transforms any references of the <i>Database Searching</i> CIM extension <i>AnnProperties</i> to Core CIM <i>Properties</i> to <i>Database Searching</i> PIM extension <i>AnnPIMComponentProperties</i> that reference the corresponding Core PIM <i>PIMComponentProperties</i> .

8.3.3 AnnAlgoResourceController element transformation

The *createAnnAlgoResourceController* ATL rule transforms *AnnAlgoResource* elements of the *Database Searching* CIM extension meta-model, to *AnnAlgoResourceController* ones of the *Database Searching* PIM extension meta-model. Therefore, the *AnnAlgoResourceController* of the *Database Searching* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *AnnAlgoResource* of the *Database Searching* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-51 createAnnAlgoResourceController ATL rule

AnnAlgoResource property/relation	AnnAlgoResourceController property/relation	Explanation
Resource	AlgoResourceController	This ATL rule transforms any references of the <i>Database Searching</i> CIM extension <i>AnnAlgoResources</i> to Core CIM <i>Resources</i> to <i>Database Searching</i> PIM extension <i>AnnAlgoResourceControllers</i> that reference the corresponding Core PIM <i>AlgoResourceControllers</i> .

8.3.4 AnnCRUDActivity element introduction

The *createAnnCRUDActivity* ATL rule introduces *AnnCRUDActivity* elements of the *Database Searching* PIM extension meta-model, by taking as input an *AnnAlgoResource* element of the *Database Searching* CIM extension meta-model. *AnnCRUDActivities* do not have Database Searching

CIM extension meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-52 createAnnCRUDActivity ATL rule

AnnCRUDActivity property/relation	Explanation
ResourceControllerCRUDActivity	This ATL rule creates <i>AnnCRUDActivity</i> elements that reference the corresponding, unique Core PIM <i>ResourceControllerCRUDActivity</i> of the overlying <i>AlgoResourceController</i> that is annotated as <i>SearchController</i> .

8.3.5 AnnCRUDActivityHandler element introduction

The *createAnnCRUDActivityHandler* ATL rule introduces *AnnCRUDActivityHandler* elements of the *Database Searching* PIM extension meta-model, by taking as input an *AnnAlgoResource* element of the *Database Searching* CIM extension meta-model. *AnnCRUDActivityHandler* do not have Database Searching CIM extension meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-53 createAnnCRUDActivityHandler ATL rule

AnnCRUDActivityHandler property/relation	Explanation
CRUDActivityHandler	This ATL rule creates <i>AnnCRUDActivityHandler</i> elements that reference the corresponding, unique Core PIM <i>CRUDActivityHandler</i> of the overlying <i>AlgoResourceController</i> that is annotated as <i>SearchController</i> .

8.3.6 AnnResourceModel element transformation

The *createAnnResourceModel* ATL rule transforms *AnnCRUDResource* elements of the *Database Searching* CIM extension meta-model, to *AnnResourceModel* ones of the *Database Searching* PIM extension meta-model. Therefore, the *AnnResourceModel* of the *Database Searching* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *AnnCRUDResource* of the *Database Searching* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-54 createAnnResourceModel ATL rule

AnnCRUDResource property/relation	AnnResourceModel property/relation	Explanation
Resource	ResourceModel	This ATL rule transforms any references of the <i>Database Searching</i> CIM extension <i>AnnCRUDResources</i> to Core CIM <i>Resources</i> to

		<i>Database Searching</i> PIM extension <i>AnnResourceModels</i> that reference the corresponding Core PIM <i>ResourceModels</i> .
--	--	--

8.3.7 SearchController element transformation

The *createPIMSearchController* ATL rule transforms *SearchResource* elements of the *Database Searching* CIM extension meta-model, to *SearchController* ones of the *Database Searching* PIM extension meta-model. Therefore, the *SearchController* of the *Database Searching* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *SearchResource* of the *Database Searching* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-55 createPIMSearchController ATL rule

SearchResource property/relation	SearchController property/relation	Explanation
AnnAlgoResource	AnnAlgoResourceController	This ATL rule transforms any associations of the <i>Database Searching</i> CIM extension <i>SearchResources</i> to <i>AnnAlgoResources</i> , to <i>Database Searching</i> PIM extension <i>SearchControllers</i> that have association with the corresponding <i>AnnAlgoResourceController</i> .

8.3.8 SearchCRUDActivity element introduction

The *createPIMSearchCRUDActivity* ATL rule introduces *SearchCRUDActivity* elements of the *Database Searching* PIM extension meta-model, by taking as input a *SearchResource* element of the *Database Searching* CIM extension meta-model. *SearchCRUDActivities* do not have Core CIM meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-56 createPIMSearchCRUDActivity ATL rule

SearchCRUDActivity property/relation	Explanation
AnnCRUDActivity	This ATL rule introduces any associations of the <i>Database Searching</i> CIM extension <i>SearchCRUDActivities</i> to <i>AnnCRUDActivities</i> by calling the <i>createAnnCRUDActivity</i> ATL rule.

8.3.9 SearchCRUDActivityHandler element introduction

The *createPIMSearchCRUDActivityHandler* ATL rule introduces *SearchCRUDActivityHandler* elements of the *Database Searching* PIM extension meta-model, by taking as input a *SearchResource* element of the *Database Searching* CIM extension meta-model. *SearchCRUDActivityHandlers* do not have

Core CIM meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-57 createPIMSearchCRUDActivityHandler ATL rule

SearchCRUDActivityHandler property/relation	Explanation
AnnCRUDActivityHandler	This ATL rule introduces any associations of the <i>Database Searching</i> CIM extension <i>SearchCRUDActivityHandlers</i> to <i>AnnCRUDActivityHandlers</i> by calling the <i>createAnnCRUDActivityHandler</i> ATL rule.
SearchableResourceModel	This ATL rule transforms any composition associations of the <i>Database Searching</i> CIM extension <i>SearchableResources</i> to <i>SearchableProperties</i> , to <i>Database Searching</i> PIM extension <i>SearchableResourceModels</i> that have composition association with the corresponding <i>Database Searching</i> PIM extension <i>SearchableProperties</i> .

8.3.10 SearchableResourceModel element transformation

The *createSearchableResourceModel* ATL rule transforms *SearchableResource* elements of the *Database Searching* CIM extension meta-model, to *SearchableResourceModel* ones of the *Database Searching* PIM extension meta-model. Therefore, the *SearchableResourceModel* of the *Database Searching* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *SearchableResource* of the *Database Searching* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-58 createSearchableResourceModel ATL rule

SearchableResource property/relation	SearchableResourceModel property/relation	Explanation
AnnCRUDResource	AnnResourceModel	This ATL rule transforms any associations of the <i>Database Searching</i> CIM extension <i>SearchableResources</i> to <i>AnnCRUDResources</i> , to <i>Database Searching</i> PIM extension <i>SearchableResourceModels</i> that have associations with the corresponding <i>AnnResourceModels</i> .
SearchableProperty	SearchableProperty	This ATL rule transforms any composition associations of the <i>Database Searching</i> CIM extension <i>SearchableResources</i> to <i>SearchableProperties</i> , to <i>Database Searching</i> PIM extension <i>SearchableResourceModels</i> that have composition association with the corresponding <i>Database Searching</i> PIM extension <i>SearchableProperties</i> .

8.3.11 SearchableProperty element transformation

The *createPIMSearchableProperty* ATL rule transforms *SearchableProperty* elements of the *Database Searching* CIM extension meta-model, to *SearchableProperty* ones of the *Database Searching* PIM extension meta-model. Therefore, the *SearchableProperty* of the *Database Searching* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *SearchableProperty* of the *Database Searching* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-59 createPIMSearchableProperty ATL rule

SearchableProperty property/relation	SearchableProperty property/relation	Explanation
AnnProperty	AnnPIMComponentProperty	This ATL rule transforms any associations of the <i>Database Searching</i> CIM extension <i>SearchableProperties</i> to <i>AnnProperties</i> , to <i>Database Searching</i> PIM extension <i>SearchableProperties</i> that have associations with the corresponding <i>AnnPIMComponentProperties</i> .

8.4 External Service Composition Extension CIM to PIM ATL Rules

8.4.1 AnnotationModel element transformation

The *ExternalServiceCIMtoPIM* ATL rule transforms *AnnotationModel* elements of the *External Service Composition* CIM extension meta-model, to *AnnotationModel* ones of the *External Service Composition* PIM extension meta-model. Therefore, the *AnnotationModel* of the *External Service Composition* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *AnnotationModel* of the *External Service Composition* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-60 ExternalServiceCIMToPIM ATL rule

AnnotationModel property/relation	AnnotationModel property/relation	Explanation
name	name	The <i>name</i> of the <i>AnnotationModel</i> of the <i>External Service Composition</i> PIM extension is the same with the name of the <i>External Service Composition</i> CIM extension <i>AnnotationModel</i> .
AnnotatedElement	AnnotatedElement	Every <i>AnnotatedElement</i> of the <i>External Service Composition</i> CIM extension meta-model is transformed to its <i>External Service Composition</i> PIM extension meta-model counterpart by calling either the <i>createAnnResourceModel</i> ATL rule for every <i>External Service Composition</i> CIM extension <i>AnnCRUDResource</i> , or the

		<i>createAnnAlgoResourceController</i> and the <i>createAnnCRUDActivity</i> and the <i>createAnnCRUDActivityHandler</i> and the <i>createAnnAlgoResourceModel</i> one for every <i>External Service Composition</i> CIM extension <i>AnnAlgoResource</i> .
Annotation	Annotation	Every <i>Annotation</i> of the <i>External Service Composition</i> CIM extension meta-model is transformed to its <i>External Service Composition</i> PIM extension meta-model counterpart by calling once the <i>createRESTClientController</i> ATL rule as well as the <i>createRESTClientModel</i> one for every <i>External Service Composition</i> CIM extension <i>RESTClientResource</i> .

8.4.2 AnnAlgoResource element transformation

The *createAnnAlgoResourceController* ATL rule transforms *AnnAlgoResource* elements of the *External Service Composition* CIM extension meta-model, to *AnnAlgoResourceController* ones of the *External Service Composition* PIM extension meta-model. Therefore, the *AnnAlgoResourceController* of the *External Service Composition* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *AnnAlgoResource* of the *External Service Composition* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-61 *createAnnAlgoResourceController* ATL rule

AnnAlgoResource property/relation	AnnAlgoResourceController property/relation	Explanation
Resource	AlgoResourceController	This ATL rule transforms any references of the <i>External Service Composition</i> CIM extension <i>AnnAlgoResources</i> to Core CIM <i>Resources</i> to <i>External Service Composition</i> PIM extension <i>AnnAlgoResourceControllers</i> that reference the corresponding Core PIM <i>AlgoResourceControllers</i> .

8.4.3 AnnCRUDResource element transformation

The *createAnnCRUDResource* ATL rule transforms *AnnCRUDResource* elements of the *External Service Composition* CIM extension meta-model, to *AnnResourceModel* ones of the *External Service Composition* PIM extension meta-model. Therefore, the *AnnResourceModel* of the *External Service Composition* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *AnnCRUDResource* of the *External Service Composition* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-62 createAnnCRUDResource ATL rule

AnnCRUDResource property/relation	AnnResourceModel property/relation	Explanation
Resource	ResourceModel	This ATL rule transforms any references of the <i>External Service Composition</i> CIM extension <i>AnnCRUDResources</i> to Core CIM <i>Resources</i> , to <i>External Service Composition</i> PIM extension <i>AnnResourceModels</i> that reference the corresponding Core PIM <i>ResourceModels</i> .

8.4.4 AnnCRUDActivity element introduction

The *createAnnCRUDActivity* ATL rule introduces *AnnCRUDActivity* elements of the *External Service Composition* PIM extension meta-model, by taking as input an *AnnAlgoResource* element of the *External Service Composition* CIM extension meta-model. *AnnCRUDActivities* do not have *External Service Composition* CIM extension meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-63 createAnnCRUDActivity ATL rule

AnnCRUDActivity property/relation	Explanation
ResourceControllerCRUDActivity	This ATL rule creates <i>AnnCRUDActivity</i> elements that reference the corresponding, unique Core PIM <i>ResourceControllerCRUDActivity</i> of the overlying <i>AlgoResourceController</i> that is annotated as <i>RESTClientController</i> .

8.4.5 AnnCRUDActivityHandler element introduction

The *createAnnCRUDActivityHandler* ATL rule introduces *AnnCRUDActivityHandler* elements of the *External Service Composition* PIM extension meta-model, by taking as input an *AnnAlgoResource* element of the *External Service Composition* CIM extension meta-model. *AnnCRUDActivityHandlers* do not have *External Service Composition* CIM extension meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-64 createAnnCRUDActivityHandler ATL rule

AnnCRUDActivityHandler property/relation	Explanation
CRUDActivityHandler	This ATL rule creates <i>AnnCRUDActivityHandler</i> elements that reference the corresponding, unique Core PIM <i>CRUDActivityHandler</i> of the overlying <i>AlgoResourceController</i> that is annotated as <i>RESTClientController</i> .

8.4.6 AnnAlgoResourceModel element introduction

The *createAnnAlgoResourceModel* ATL rule introduces *AnnAlgoResourceModel* elements of the *External Service Composition* PIM extension meta-model, by taking as input an *AnnAlgoResource* element of the *External Service Composition* CIM extension meta-model. *AnnAlgoResourceModels* do not have *External Service Composition* CIM extension meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-65 createAnnAlgoResourceModel ATL rule

AnnAlgoResourceModel property/relation	Explanation
AlgoResourceModel	This ATL rule creates <i>AnnAlgoResourceModel</i> elements that reference the corresponding, unique Core PIM <i>AlgoResourceModel</i> of the associated <i>AlgoResourceController</i> that is annotated as <i>RESTClientController</i> .

8.4.7 RESTClientResource element transformation

The *createRESTClientController* ATL rule transforms *RESTClientResource* elements of the *External Service Composition* CIM extension meta-model, to *RESTClientController* ones of the *External Service Composition* PIM extension meta-model. Therefore, the *RESTClientController* of the *External Service Composition* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *RESTClientResource* of the *External Service Composition* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-66 createRESTClientController ATL rule

RESTClientResource property/relation	RESTClientController property/relation	Explanation
AnnAlgoResource	AnnAlgoResourceController	This ATL rule transforms any associations of the <i>External Service Composition</i> CIM extension <i>RESTClientResources</i> to <i>AnnAlgoResources</i> , to <i>External Service Composition</i> PIM extension <i>RESTClientControllers</i> that have associations with the corresponding <i>AnnAlgoResourceControllers</i> .
-	RESTClientModel	This ATL rule introduces associations of the <i>External Service Composition</i> CIM extension <i>RESTClientControllers</i> to <i>RESTClientModels</i> , by calling the <i>createRESTClientModel</i> ATL rule.
-	RESTClientCRUDActivity	This ATL rule associates the <i>External Service Composition</i> CIM extension <i>RESTClientControllers</i> to underlying <i>RESTClientCRUDActivities</i> , by calling the <i>createRESTClientCRUDActivity</i> ATL rule.

8.4.8 RESTClientCRUDActivity element introduction

The *createRESTClientCRUDActivity* ATL rule introduces *RESTClientCRUDActivity* elements of the *External Service Composition* PIM extension meta-model, by taking as input a *RESTClientResource* element of the *External Service Composition* CIM extension meta-model. *RESTClientCRUDActivities* do not have *External Service Composition* CIM extension meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-67 createRESTClientCRUDActivity ATL rule

RESTClientCRUDActivity property/relation	Explanation
AnnCRUDActivity	This ATL rule creates <i>RESTClientCRUDActivity</i> elements that reference the corresponding, unique Core PIM <i>ResourceControllerCRUDActivity</i> of the associated <i>AlgoResourceController</i> that is annotated by the overlying <i>RESTClientController</i> .
RESTClientCRUDActivityHandler	This ATL rule introduces associations of the <i>External Service Composition</i> CIM extension <i>RESTClientCRUDActivities</i> to <i>RESTClientCRUDActivityHandlers</i> , by calling the <i>createRESTClientCRUDActivityHandler</i> ATL rule.
QueryParam	This ATL rule introduces associations of the <i>External Service Composition</i> CIM extension <i>RESTClientCRUDActivities</i> to <i>QueryParams</i> , by calling the <i>createQueryParam</i> ATL rule.

8.4.9 RESTClientModel element introduction

The *createRESTClientModel* ATL rule introduces *RESTClientModel* elements of the *External Service Composition* PIM extension meta-model, by taking as input a *RESTClientResource* element of the *External Service Composition* CIM extension meta-model. *RESTClientModels* do not have *External Service Composition* CIM extension meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-68 createRESTClientModel ATL rule

RESTClientModel property/relation	Explanation
AnnAlgoResourceModel	This ATL rule creates <i>AnnAlgoResourceModel</i> elements that reference the corresponding, unique Core PIM <i>AlgoResourceModel</i> of the associated <i>AlgoResourceController</i> that is annotated by the overlying <i>RESTClientController</i> .
InputDataManager	This ATL rule introduces associations of the <i>External Service Composition</i> CIM extension <i>RESTClientModel</i> to <i>InputDataModels</i> , by calling the <i>createInputDataManager</i> ATL rule, if the corresponding <i>External Service Composition</i> CIM

	extension <i>TargetRestService</i> has an associated <i>InputDataModel</i> .
OutputDataModel	This ATL rule introduces associations of the <i>External Service Composition</i> CIM extension <i>RESTClientModel</i> to <i>OutputDataModels</i> , by calling the <i>createNonPersistentOutput</i> , or the <i>createAutoPersistentOutput</i> , or the <i>createExistenCRUDPersistentOutput</i> ATL rule, if the corresponding <i>External Service Composition</i> CIM extension <i>TargetRestService</i> has an associated <i>NonPersistentOutput</i> , <i>AutoPersistentOutput</i> or <i>ExistenCRUDPersistentOutput</i> accordingly.

8.4.10 TargetRESTService element transformation

The *createRESTClientCRUDActivityHandler* ATL rule transforms *TargetRESTService* elements of the *External Service Composition* CIM extension meta-model, to *RESTClientCRUDActivityHandler* ones of the *External Service Composition* PIM extension meta-model. Therefore, the *RESTClientCRUDActivityHandler* of the *External Service Composition* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *TargetRESTService* of the *External Service Composition* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-69 *createRESTClientCRUDActivityHandler* ATL rule

TargetRESTService property/relation	RESTClientCRUDActivityHandler property/relation	Explanation
targetURL	targetServiceURL	The <i>targetServiceURL</i> of the <i>External Service Composition</i> PIM extension <i>RESTClientCRUDActivityHandler</i> is set equal to the <i>targetURL</i> of the <i>External Service Composition</i> CIM extension <i>TargetRESTService</i> .
CRUDVerb	CRUDVerb	The <i>CRUDVerb</i> of the <i>External Service Composition</i> PIM extension <i>RESTClientCRUDActivityHandler</i> is set equal to the <i>CRUDVerb</i> of the <i>External Service Composition</i> CIM extension <i>TargetRESTService</i> .
-	AnnCRUDActivityHandler	This ATL rule introduces associations of the <i>External Service Composition</i> CIM extension <i>RESTClientCRUDActivityHandlers</i> to <i>AnnCRUDActivityHandlers</i> , by calling the <i>createAnnCRUDActivityHandler</i> ATL rule.

8.4.11 InputDataModel element transformation

The *createInputDataModel* ATL rule transforms *InputDataModel* elements of the *External Service Composition* CIM extension meta-model, to *InputDataModel* ones of the *External Service*

Composition PIM extension meta-model. Therefore, the *InputDataModel* of the *External Service Composition* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *InputDataModel* of the *External Service Composition* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-70 createInputDataModel ATL rule

InputDataModel property/relation	InputDataModel property/relation	Explanation
Property	Property	This ATL rule transforms any composition associations of the <i>External Service Composition</i> CIM extension <i>InputDataModels</i> to <i>Properties</i> , to <i>External Service Composition</i> PIM extension <i>InputDataModels</i> that have composition association with the corresponding <i>External Service Composition</i> PIM extension <i>Properties</i> .
Representation	Representation	This ATL rule transforms any composition associations of the <i>External Service Composition</i> CIM extension <i>InputDataModels</i> to <i>Representations</i> , to <i>External Service Composition</i> PIM extension <i>InputDataModels</i> that have composition association with the corresponding <i>External Service Composition</i> PIM extension <i>Representations</i> .

8.4.12 Representation element transformation

The *createRepresentation* ATL rule transforms *Representation* elements of the *External Service Composition* CIM extension meta-model, to *Representation* ones of the *External Service Composition* PIM extension meta-model. Therefore, the *Representation* of the *External Service Composition* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *Representation* of the *External Service Composition* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-71 createRepresentation ATL rule

Representation property/relation	Representation property/relation	Explanation
name	name	The <i>name</i> of the <i>External Service Composition</i> PIM extension <i>Representation</i> is set equal to the <i>name</i> of the <i>External Service Composition</i> CIM extension <i>Representation</i> .

8.4.13 Property element transformation

The *createProperty* ATL rule transforms *Property* elements of the *External Service Composition* CIM extension meta-model, to *Property* ones of the *External Service Composition* PIM extension meta-model. Therefore, the *Property* of the *External Service Composition* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *Property* of the *External Service Composition* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-72 *createProperty* ATL rule

Property property/relation	Property property/relation	Explanation
name	name	The <i>name</i> of the <i>External Service Composition</i> PIM extension <i>Property</i> is set equal to the <i>name</i> of the <i>External Service Composition</i> CIM extension <i>Property</i> .
type	type	The <i>type</i> of the <i>External Service Composition</i> PIM extension <i>Property</i> is set equal to the <i>type</i> of the <i>External Service Composition</i> CIM extension <i>Property</i> .
isUnique	isUnique	The <i>isUnique</i> property of the <i>External Service Composition</i> PIM extension <i>Property</i> is set equal to the <i>isUnique</i> property of the <i>External Service Composition</i> CIM extension <i>Property</i> .

8.4.14 QueryParam element transformation

The *createQueryParam* ATL rule transforms *QueryParam* elements of the *External Service Composition* CIM extension meta-model, to *QueryParam* ones of the *External Service Composition* PIM extension meta-model. Therefore, the *QueryParam* of the *External Service Composition* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *QueryParam* of the *External Service Composition* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-73 *createQueryParam* ATL rule

QueryParam property/relation	QueryParam property/relation	Explanation
name	name	The <i>name</i> of the <i>External Service Composition</i> PIM extension <i>QueryParam</i> is set equal to the <i>name</i> of the <i>External Service Composition</i> CIM extension <i>QueryParam</i> .
type	type	The <i>type</i> of the <i>External Service Composition</i> PIM extension <i>QueryParam</i> is set equal to the <i>type</i> of

		the <i>External Service Composition</i> CIM extension <i>QueryParam</i> .
--	--	---

8.4.15 NonPersistentOutput element transformation

The *createNonPersistentOutput* ATL rule transforms *NonPersistentOutput* elements of the *External Service Composition* CIM extension meta-model, to *NonPersistentOutput* ones of the *External Service Composition* PIM extension meta-model. Therefore, the *NonPersistentOutput* of the *External Service Composition* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *NonPersistentOutput* of the *External Service Composition* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-74 *createNonPersistentOutput* ATL rule

NonPersistentOutput property/relation	NonPersistentOutput property/relation	Explanation
Property	Property	This ATL rule transforms any composition associations of the <i>External Service Composition</i> CIM extension <i>NonPersistentOutputs</i> to <i>Properties</i> , to <i>External Service Composition</i> PIM extension <i>NonPersistentOutputs</i> that have composition association with the corresponding <i>External Service Composition</i> PIM extension <i>Properties</i> .
Representation	Representation	This ATL rule transforms any composition associations of the <i>External Service Composition</i> CIM extension <i>NonPersistentOutputs</i> to <i>Representations</i> , to <i>External Service Composition</i> PIM extension <i>NonPersistentOutputs</i> that have composition association with the corresponding <i>External Service Composition</i> PIM extension <i>Representations</i> .

8.4.16 AutoPersistentOutput element transformation

The *ExternalServiceCIMtoPIM* ATL rule transforms *AutoPersistentOutput* elements of the *External Service Composition* CIM extension meta-model, to *AutoPersistentOutput* ones of the *External Service Composition* PIM extension meta-model. Therefore, the *AutoPersistentOutput* of the *External Service Composition* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *AutoPersistentOutput* of the *External Service Composition* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-75 createAutoPersistentOutput ATL rule

AutoPersistentOutput property/relation	AutoPersistentOutput property/relation	Explanation
Property	Property	This ATL rule transforms any composition associations of the <i>External Service Composition</i> CIM extension <i>AutoPersistentOutputs</i> to <i>Properties</i> , to <i>External Service Composition</i> PIM extension <i>AutoPersistentOutputs</i> that have composition association with the corresponding <i>External Service Composition</i> PIM extension <i>Properties</i> .
Representation	Representation	This ATL rule transforms any composition associations of the <i>External Service Composition</i> CIM extension <i>AutoPersistentOutputs</i> to <i>Representations</i> , to <i>External Service Composition</i> PIM extension <i>AutoPersistentOutputs</i> that have composition association with the corresponding <i>External Service Composition</i> PIM extension <i>Representations</i> .

8.4.17 ExistentsCRUDPersistentOutput element transformation

The *ExternalServiceCIMtoPIM* ATL rule transforms *ExistentsCRUDPersistentOutput* elements of the *External Service Composition* CIM extension meta-model, to *ExistentsCRUDPersistentOutput* ones of the *External Service Composition* PIM extension meta-model. Therefore, the *ExistentsCRUDPersistentOutput* of the *External Service Composition* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *ExistentsCRUDPersistentOutput* of the *External Service Composition* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-76 createExistentsCRUDPersistentOutput ATL rule

ExistentsCRUDPersistentOutput property/relation	ExistentsCRUDPersistentOutput property/relation	Explanation
Property	Property	This ATL rule transforms any composition associations of the <i>External Service Composition</i> CIM extension <i>ExistentsCRUDPersistentOutputs</i> to <i>Properties</i> , to <i>External Service Composition</i> PIM extension <i>ExistentsCRUDPersistentOutputs</i> that have composition association with the corresponding <i>External Service Composition</i> PIM extension <i>Properties</i> .
Representation	Representation	This ATL rule transforms any composition associations of the <i>External Service Composition</i> CIM extension <i>ExistentsCRUDPersistentOutputs</i> to <i>Representations</i> , to <i>External Service Composition</i> PIM extension <i>ExistentsCRUDPersistentOutputs</i> that have composition association with the corresponding <i>External Service Composition</i> PIM extension <i>Representations</i> .

		<p><i>Composition CIM extension</i> <i>ExistentCRUDPersistentOutputs to Representations, to External Service</i> <i>Composition PIM extension</i> <i>ExistentCRUDPersistentOutputs that have composition association with the corresponding External Service Composition PIM extension Representations.</i></p>
AnnCRUDResource	AnnResourceModel	<p>This ATL rule transforms any associations of the <i>External Service Composition CIM extension</i> <i>ExistentCRUDPersistentOutputs to AnnCRUDResources, to External Service</i> <i>Composition PIM extension</i> <i>ExistentCRUDPersistentOutputs that have associations with the corresponding AnnResourceModels.</i></p>

8.5 ABAC Authorization Extension CIM to PIM ATL Rules

8.5.1 AnnotationModel element transformation

The *AuthorizationCIMtoPIM* ATL rule transforms *AnnotationModel* elements of the *ABAC Authorization CIM extension* meta-model, to *AnnotationModel* ones of the *ABAC Authorization PIM extension* meta-model. Therefore, the *AnnotationModel* of the *ABAC Authorization PIM extension* meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *AnnotationModel* of the *ABAC Authorization CIM extension* meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-77 AuthorizationCIMToPIM ATL rule

AnnotationModel property/relation	AnnotationModel property/relation	Explanation
name	name	The <i>name</i> of the <i>AnnotationModel</i> of the <i>ABAC Authorization PIM extension</i> equals the name of the <i>AnnotationModel</i> of the <i>ABAC Authorization CIM extension</i> .
AnnotatedElement	AnnotatedElement	Every <i>AnnotatedElement</i> of the <i>ABAC Authorization CIM extension</i> meta-model is transformed to its <i>ABAC Authorization PIM extension</i> meta-model counterpart by calling once the <i>createAnnResourceModel</i> ATL rule, or the <i>createAnnAlgoResourceModel</i> , or the <i>createAnnResourceModelManager</i> one, depending on the case, for every <i>ABAC Authorization CIM extension</i> <i>AnnResource</i> . Moreover, the <i>createAnnDatabaseController</i> ATL rule introduces <i>AnnDatabaseController</i> elements,

		the <i>createAnnCRUDActivityHandler</i> introduces <i>AnnCRUDActivityHandlers</i> and the <i>createAnnPIMComponentProperty</i> ATL rule or the <i>createAnnResourceControllerCRUDActivity</i> for every ABAC Authorization CIM extension <i>AnnProperty</i> or <i>AnnCRUDActivity</i> respectively.
Annotation	Annotation	Every <i>Annotation</i> of the ABAC Authorization CIM extension meta-model is transformed to its ABAC Authorization PIM extension meta-model counterpart by calling once the <i>createAuthorizationSubject</i> ATL rule for every ABAC Authorization CIM extension <i>AuthorizationSubject</i> and the <i>createAuthorizableResource</i> ATL rule for every ABAC Authorization CIM extension <i>AuthorizableResource</i> . Additionally, it introduces <i>AuthorizationPerformer</i> annotations by calling the <i>createAuthorizationperformer</i> ATL rule.

8.5.2 AnnResource element transformation (case 1)

The *createAnnResourceModel* ATL rule transforms *AnnResource* elements of the ABAC Authorization CIM extension meta-model, to *AnnResourceModel* ones of the ABAC Authorization PIM extension meta-model. Therefore, the *AnnResourceModel* of the ABAC Authorization PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *AnnResource* of the ABAC Authorization CIM extension meta-model element. In this case, only the *AnnResource* that is annotated as *AuthorizationSubject* in the ABAC Authorization CIM extension meta-model is transformed. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-78 createAnnResourceModel ATL rule

AnnResource property/relation	AnnResourceModel property/relation	Explanation
Resource	ResourceModel	This ATL rule transforms any references of the ABAC Authorization CIM extension <i>AnnResources</i> to Core CIM <i>Resources</i> , to ABAC Authorization PIM extension <i>AnnResourceModels</i> that reference the corresponding Core PIM <i>ResourceModels</i> .

8.5.3 AnnResource element transformation (case 2)

The *createAnnResourceModel* and *createAnnResourceModelManager* ATL rules transform *AnnResource* elements of the ABAC Authorization CIM extension meta-model, to *AnnResourceModel* and *AnnResourceModelManager* ones of the ABAC Authorization PIM extension meta-model. Therefore, the pair of *AnnResourceModel* and *AnnResourceModelManager* of the ABAC

Authorization PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *AnnResource* of the *ABAC Authorization* CIM extension meta-model element. In this case, only *AnnResources* that are not annotated as *AuthorizationSubject* and that reference Core CIM *Resources*, which have their *isAlgorithmic* property set to false, are transformed. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-79 createAnnResourceModel ATL rule

AnnResource property/relation	AnnResourceModel property/relation	Explanation
Resource	ResourceModel	This ATL rule transforms any references of the <i>ABAC Authorization</i> CIM extension <i>AnnResources</i> to Core CIM <i>Resources</i> , to <i>ABAC Authorization</i> PIM extension <i>AnnResourceModels</i> that reference the corresponding Core PIM <i>ResourceModels</i> .

Table 8-80 createAnnResourceModelManager ATL rule

AnnResource property/relation	AnnResourceModelManager property/relation	Explanation
Resource	ResourceModelManager	This ATL rule transforms any references of the <i>ABAC Authorization</i> CIM extension <i>AnnResources</i> to Core CIM <i>Resources</i> , to <i>ABAC Authorization</i> PIM extension <i>AnnResourceModelManagers</i> that reference the corresponding Core PIM <i>ResourceModelManagers</i> .

8.5.4 AnnResource element transformation (case 3)

The *createAnnAlgoResourceModel* ATL rule transforms *AnnResource* elements of the *ABAC Authorization* CIM extension meta-model, to *AnnAlgoResourceModel* ones of the *ABAC Authorization* PIM extension meta-model. Therefore, the *AnnAlgoResourceModel* of the *ABAC Authorization* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *AnnResource* of the *ABAC Authorization* CIM extension meta-model element. In this case, only *AnnResources* that are not annotated as *AuthorizationSubject* and that reference Core CIM *Resources*, which have their *isAlgorithmic* property set to true, are transformed. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-81 createAnnAlgoResourceModel ATL rule

AnnResource property/relation	AnnAlgoResourceModel property/relation	Explanation
Resource	AlgoResourceModel	This ATL rule transforms any references of the <i>ABAC Authorization</i> CIM extension <i>AnnResources</i> to Core CIM <i>Resources</i> , to <i>ABAC Authorization</i>

		PIM extension <i>AnnAlgoResourceModels</i> that reference the corresponding Core PIM <i>AlgoResourceModels</i> .
--	--	--

8.5.5 AnnProperty element transformation

The *createAnnPIMComponentProperty* ATL rule transforms *AnnProperty* elements of the *ABAC Authorization* CIM extension meta-model, to *AnnPIMComponentProperty* ones of the *ABAC Authorization* PIM extension meta-model. Therefore, the *AnnPIMComponentProperty* of the *ABAC Authorization* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *AnnProperty* of the *ABAC Authorization* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-82 createAnnPIMComponentProperty ATL rule

AnnProperty property/relation	AnnPIMComponentProperty property/relation	Explanation
Property	PIMComponent	This ATL rule transforms any references of the <i>ABAC Authorization</i> CIM extension <i>AnnProperties</i> to Core CIM <i>Properties</i> , to <i>ABAC Authorization</i> PIM extension <i>AnnPIMComponentProperties</i> that reference the corresponding Core PIM <i>PIMComponentProperties</i> .

8.5.6 AnnCRUDActivity element transformation

The *createAnnCRUDActivity* ATL rule transforms *AnnCRUDActivity* elements of the *ABAC Authorization* CIM extension meta-model, to *AnnResourceControllerCRUDActivity* ones of the *ABAC Authorization* PIM extension meta-model. Therefore, the *AnnResourceControllerCRUDActivity* of the *ABAC Authorization* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *AnnCRUDActivity* of the *ABAC Authorization* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-83 createAnnResourceControllerCRUDActivity ATL rule

AnnCRUDActivity property/relation	AnnResourceControllerCRUDActivity property/relation	Explanation
CRUDActivity	ResourceControllerCRUDActivity	This ATL rule transforms any references of the <i>ABAC Authorization</i> CIM extension <i>AnnCRUDActivities</i> to Core CIM <i>CRUDActivities</i> , to <i>ABAC Authorization</i> PIM extension <i>AnnResourceControllerCRUDActivities</i> that reference the corresponding Core PIM <i>ResourceControllerCRUDActivities</i> .

8.5.7 AnnCRUDActivityHandler element introduction

The *createAnnCRUDActivityHandler* ATL rule introduces *AnnCRUDActivityHandler* elements of the *ABAC Authorization* PIM extension meta-model, by taking as input an *AnnResource* element of the *ABAC Authorization* CIM extension meta-model, that is annotated as *AuthorizableResource*. *AnnCRUDActivities* do not have *ABAC Authorization* CIM extension meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-84 createAnnCRUDActivityHandler ATL rule

AnnCRUDActivityHandler property/relation	Explanation
CRUDActivityHandler	This ATL rule creates <i>AnnCRUDActivityHandler</i> elements that reference the corresponding Core PIM <i>CRUDActivityHandler</i> of the overlying <i>ResourceModel</i> or <i>ResourceModelManager</i> that are annotated as <i>AuthorizableResources</i> .

8.5.8 AnnDatabaseController element introduction

The *createAnnDatabaseController* ATL rule introduces *AnnDatabaseController* elements of the *ABAC Authorization* PIM extension meta-model. *AnnDatabaseControllers* do not have *ABAC Authorization* CIM extension meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-85 createAnnDatabaseController ATL rule

AnnDatabaseController property/relation	Explanation
DatabaseController	This ATL rule creates <i>AnnDatabaseController</i> elements that reference the corresponding, unique Core PIM <i>DatabaseController</i> .

8.5.9 AuthorizationSubject element transformation

The *createAuthorizationSubject* ATL rule transforms *AuthorizationSubject* elements of the *ABAC Authorization* CIM extension meta-model, to *AuthorizationSubject* ones of the *ABAC Authorization* PIM extension meta-model. Therefore, the *AuthorizationSubject* of the *ABAC Authorization* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *AuthorizationSubject* of the *ABAC Authorization* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-86 createAuthorizationSubject ATL rule

AuthorizationSubject property/relation	AuthorizationSubject property/relation	Explanation
AnnResource	AnnResourceModel	This ATL rule transforms any associations of the <i>ABAC Authorization</i> CIM extension <i>AuthorizationSubjects</i> to <i>AnnResources</i> , to <i>ABAC Authorization</i> PIM extension <i>AuthorizationSubjects</i> that have associations with the corresponding <i>AnnResourceModels</i> .
SubjectAttribute	SubjectAttribute	This ATL rule transforms any composition associations of the <i>ABAC Authorization</i> CIM extension <i>AuthorizationSubjects</i> to <i>SubjectAttributes</i> , to <i>ABAC Authorization</i> PIM extension <i>AuthorizationSubjects</i> that have composition association with the corresponding <i>ABAC Authorization</i> PIM extension <i>SubjectAttributes</i> .

8.5.10 SubjectAttribute element transformation

The *createSubjectAttribute* ATL rule transforms *SubjectAttribute* elements of the *ABAC Authorization* CIM extension meta-model, to *SubjectAttribute* ones of the *ABAC Authorization* PIM extension meta-model. Therefore, the *SubjectAttribute* of the *ABAC Authorization* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *SubjectAttribute* of the *ABAC Authorization* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-87 createSubjectAttribute ATL rule

SubjectAttribute property/relation	SubjectAttribute property/relation	Explanation
name	name	The <i>name</i> of the <i>ABAC Authorization</i> PIM extension <i>SubjectAttribute</i> is set equal to the <i>name</i> of the <i>ABAC Authorization</i> CIM extension <i>SubjectAttribute</i> .
type	type	The <i>type</i> of the <i>ABAC Authorization</i> PIM extension <i>SubjectAttribute</i> is set equal to the <i>type</i> of the <i>ABAC Authorization</i> CIM extension <i>SubjectAttribute</i> .
isUnique	isUnique	The <i>isUnique</i> property of the <i>ABAC Authorization</i> PIM extension <i>SubjectAttribute</i> is set equal to the <i>isUnique</i> property of the <i>ABAC Authorization</i> CIM extension <i>SubjectAttribute</i> .

8.5.11 AuthorizableResource element transformation

The *createAuthorizableResource* ATL rule transforms *AuthorizableResource* elements of the *ABAC Authorization* CIM extension meta-model, to *AuthorizableResource* ones of the *ABAC Authorization*

PIM extension meta-model. Therefore, the *AuthorizableResource* of the *ABAC Authorization* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *AuthorizableResource* of the *ABAC Authorization* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-88 createAuthorizableResource ATL rule

AuthorizableResource property/relation	AuthorizableResource property/relation	Explanation
AnnResource	AnnResourceModel/AnnResourceModelManager - AnnAlgoResourceModel	This ATL rule transforms any associations of the <i>ABAC Authorization</i> CIM extension <i>AuthorizableResource</i> to <i>AnnResources</i> , to <i>ABAC Authorization</i> PIM extension <i>AuthorizableResources</i> that have associations with the corresponding pair of <i>AnnResourceModel/AnnResourceModelManager</i> or the corresponding <i>AnnAlgoResourceModel</i> depending on the case.
ResourceAccessPolicySet	ResourceAccessPolicySet	This ATL rule transforms any composition associations of the <i>ABAC Authorization</i> CIM extension <i>AuthorizableResources</i> to <i>ResourceAccessPolicySets</i> , to <i>ABAC Authorization</i> PIM extension <i>AuthorizableResources</i> that have composition association with the corresponding <i>ABAC Authorization</i> PIM extension <i>ResourcePolicySets</i> .
-	AuthorizationPerformer	This ATL rule introduces associations of the <i>ABAC Authorization</i> PIM extension <i>AuthorizableResources</i> to <i>AuthorizationPerformers</i> , by calling the <i>createAuthorizationperformer</i> ATL rule.

8.5.12 AuthorizationPerformer element introduction

The *createAuthorizationPerformer* ATL rule introduces *AuthorizationPerformer* elements of the *ABAC Authorization* PIM extension meta-model, by taking as input an *AnnResource* element of the *ABAC Authorization* CIM extension meta-model. *AuthorizationPerformer* do not have *ABAC Authorization* CIM extension meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-89 createAuthorizationPerformer ATL rule

AuthorizationPerformer property/relation	Explanation
AnnCRUDActivityHandler	This ATL rule introduces associations of the <i>ABAC Authorization</i> PIM extension <i>AuthorizationPerformers</i> to

	<i>AnnCRUDActivityHandlers</i> , by calling the <i>createAnnCRUDActivityHandlers</i> ATL rule.
AuthorizationPolicyEvaluator	This ATL rule introduces associations of the <i>ABAC Authorization</i> PIM extension <i>AuthorizationPerformer</i> to <i>AuthorizationPolicyEvaluator</i> , by calling the <i>createAuthorizationPolicyEvaluator</i> ATL rule.

8.5.13 AuthorizationPolicyEvaluator element introduction

The *createAuthorizationPolicyEvaluator* ATL rule introduces *AuthorizationPolicyEvaluator* elements of the *ABAC Authorization* PIM extension meta-model. *AuthorizationPolicyEvaluators* do not have *ABAC Authorization* CIM extension meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-90 *createAuthorizationPolicyEvaluator* ATL rule

AuthorizationPolicyEvaluator property/relation	Explanation
AuthorizationDataHandler	This ATL rule introduces associations of the <i>ABAC Authorization</i> PIM extension <i>AuthorizationPolicyEvaluator</i> to <i>AuthorizationDataHandler</i> , by calling the <i>createAuthorizationDataHandler</i> ATL rule.

8.5.14 ResourceAccessPolicySet element transformation

The *createResourceAccessPolicySet* ATL rule transforms *ResourceAccessPolicySet* elements of the *ABAC Authorization* CIM extension meta-model, to *ResourceAccessPolicySet* ones of the *ABAC Authorization* PIM extension meta-model. Therefore, the *ResourceAccessPolicySet* of the *ABAC Authorization* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *ResourceAccessPolicySet* of the *ABAC Authorization* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-91 *createResourceAccessPolicySet* ATL rule

ResourceAccessPolicySet property/relation	ResourceAccessPolicySet property/relation	Explanation
RuleCombiningAlgorithm	RuleCombiningAlgorithm	This ATL rule transforms any composition associations of the <i>ABAC Authorization</i> CIM extension <i>ResourceAccessPolicySets</i> to <i>RuleCombiningAlgorithms</i> , to <i>ABAC Authorization</i> PIM extension <i>ResourceAccessPolicySets</i> that have composition association with the corresponding <i>ABAC Authorization</i> PIM extension <i>RuleCombiningAlgorithms</i> .
ResourceAccessPolicy	ResourceAccessPolicy	This ATL rule transforms any composition

		associations of the <i>ABAC Authorization</i> CIM extension <i>ResourceAccessPolicySets</i> to <i>ResourceAccessPolicies</i> , to <i>ABAC Authorization</i> PIM extension <i>ResourceAccessPolicySets</i> that have composition association with the corresponding <i>ABAC Authorization</i> PIM extension <i>ResourceAccessPolicies</i> .
-	AuthorizationDataTable	This ATL rule introduces associations of the <i>ABAC Authorization</i> PIM extension <i>ResourceAccessPolicySets</i> to <i>AuthorizationDataTables</i> , by calling the <i>createAuthorizationDataTable</i> ATL rule.

8.5.15 ResourceAccessPolicy element transformation

The *createResourceAccessPolicy* ATL rule transforms *ResourceAccessPolicy* elements of the *ABAC Authorization* CIM extension meta-model, to *ResourceAccessPolicy* ones of the *ABAC Authorization* PIM extension meta-model. Therefore, the *ResourceAccessPolicy* of the *ABAC Authorization* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *ResourceAccessPolicy* of the *ABAC Authorization* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-92 *createResourceAccessPolicy* ATL rule

ResourceAccessPolicy property/relation	ResourceAccessPolicy property/relation	Explanation
RuleCombiningAlgorithm	RuleCombiningAlgorithm	This ATL rule transforms any composition associations of the <i>ABAC Authorization</i> CIM extension <i>ResourceAccessPolicies</i> to <i>RuleCombiningAlgorithms</i> , to <i>ABAC Authorization</i> PIM extension <i>ResourceAccessPolicies</i> that have composition association with the corresponding <i>ABAC Authorization</i> PIM extension <i>RuleCombiningAlgorithms</i> .
ResourceAccessRule	ResourceAccessRule	This ATL rule transforms any composition associations of the <i>ABAC Authorization</i> CIM extension <i>ResourceAccessPolicies</i> to <i>ResourceAccessRules</i> , to <i>ABAC Authorization</i> PIM extension <i>ResourceAccessPolicies</i> that have composition association with the corresponding <i>ABAC Authorization</i> PIM extension <i>ResourceAccessRules</i> .
-	AuthorizationDataTable	This ATL rule introduces associations of the <i>ABAC Authorization</i> PIM extension <i>ResourceAccessPolicies</i> to <i>AuthorizationDataTables</i> , by calling the <i>createAuthorizationDataTable</i> ATL rule.

8.5.16 DenyOverridesAlgorithm element transformation

The *createDenyOverridesAlgorithm* ATL rule transforms *DenyOverridesAlgorithm* elements of the *ABAC Authorization* CIM extension meta-model, to *DenyOverridesAlgorithm* ones of the *ABAC Authorization* PIM extension meta-model. Therefore, the *DenyOverridesAlgorithm* of the *ABAC Authorization* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *DenyOverridesAlgorithm* of the *ABAC Authorization* CIM extension meta-model element. Since *DenyOverridesAlgorithm* elements do not have any associations or properties, this ATL rule does not perform any association/property mappings.

8.5.17 PermitOverridesAlgorithm element transformation

The *createPermitOverridesAlgorithm* ATL rule transforms *PermitOverridesAlgorithm* elements of the *ABAC Authorization* CIM extension meta-model, to *PermitOverridesAlgorithm* ones of the *ABAC Authorization* PIM extension meta-model. Therefore, the *PermitOverridesAlgorithm* of the *ABAC Authorization* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *PermitOverridesAlgorithm* of the *ABAC Authorization* CIM extension meta-model element. Since the *PermitOverridesAlgorithm* elements do not have any associations/properties, this rule does not perform any association/property mappings.

8.5.18 ResourceAccessPermitRule element transformation

The *createResourceAccessPermitRule* ATL rule transforms *ResourceAccessPermitRule* elements of the *ABAC Authorization* CIM extension meta-model, to *ResourceAccessPermitRule* ones of the *ABAC Authorization* PIM extension meta-model. Therefore, the *ResourceAccessPermitRule* of the *ABAC Authorization* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *ResourceAccessPermitRule* of the *ABAC Authorization* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-93 *createResourceAccessPermitRule* ATL rule

ResourceAccessPermitRule property/relation	ResourceAccessPermitRule property/relation	Explanation
AllowedAction	AllowedAction	This ATL rule transforms any composition associations of the <i>ABAC Authorization</i> CIM extension <i>ResourceAccessPermitRules</i> to <i>AllowedActions</i> , to <i>ABAC Authorization</i> PIM extension <i>ResourceAccessPermitRules</i> that have composition association with the corresponding <i>ABAC Authorization</i> PIM extension <i>AllowedActions</i> .
MatchedResourceAttribute	MatchedResourceAttribute	This ATL rule transforms any composition associations of the <i>ABAC Authorization</i> CIM extension <i>ResourceAccessPermitRules</i> to <i>MatchedResourceAttributes</i> , to <i>ABAC Authorization</i> PIM extension <i>ResourceAccessPermitRules</i> that have composition association with the corresponding

		<i>ABAC Authorization PIM extension MatchedResourceAttributes.</i>
MatchedContextAttribute	MatchedContextAttribute	This ATL rule transforms any composition associations of the <i>ABAC Authorization CIM extension ResourceAccessPermitRules</i> to <i>MatchedContextAttributes</i> , to <i>ABAC Authorization PIM extension ResourceAccessPermitRules</i> that have composition association with the corresponding <i>ABAC Authorization PIM extension MatchedContextAttributes</i> .
MatchedSubjectAttribute	MatchedSubjectAttribute	This ATL rule transforms any composition associations of the <i>ABAC Authorization CIM extension ResourceAccessPermitRules</i> to <i>MatchedSubjectAttributes</i> , to <i>ABAC Authorization PIM extension ResourceAccessPermitRules</i> that have composition association with the corresponding <i>ABAC Authorization PIM extension MatchedSubjectAttribute</i> .
-	AuthorizationDataTable	This ATL rule introduces associations of the <i>ABAC Authorization PIM extension ResourceAccessPermitRules</i> to <i>AuthorizationDataTables</i> , by calling the <i>createAuthorizationDataTable</i> ATL rule.

8.5.19 ResourceAccessDenyRule element transformation

The *createResourceAccessRule* ATL rule transforms *ResourceAccessDenyRule* elements of the *ABAC Authorization CIM extension* meta-model, to *ResourceAccessDenyRule* ones of the *ABAC Authorization PIM extension* meta-model. Therefore, the *ResourceAccessDenyRule* of the *ABAC Authorization PIM extension* meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *ResourceAccessDenyRule* of the *ABAC Authorization CIM extension* meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-94 *createResourceAccessDenyRule* ATL rule

ResourceAccessDenyRule property/relation	ResourceAccessDenyRule property/relation	Explanation
AllowedAction	AllowedAction	This ATL rule transforms any composition associations of the <i>ABAC Authorization CIM extension ResourceAccessDenyRules</i> to <i>AllowedActions</i> , to <i>ABAC Authorization PIM extension ResourceAccessDenyRules</i> that have composition association with the corresponding <i>ABAC Authorization PIM extension AllowedActions</i> .
MatchedResourceAttribute	MatchedResourceAttribute	This ATL rule transforms any composition associations of the <i>ABAC Authorization CIM extension ResourceAccessDenyRules</i> to <i>MatchedResourceAttributes</i> , to <i>ABAC</i>

		<p><i>Authorization</i> PIM extension <i>ResourceAccessDenyRules</i> that have composition association with the corresponding <i>ABAC Authorization</i> PIM extension <i>MatchedResourceAttributes</i>.</p>
MatchedContextAttribute	MatchedContextAttribute	<p>This ATL rule transforms any composition associations of the <i>ABAC Authorization</i> CIM extension <i>ResourceAccessDenyRules</i> to <i>MatchedContextAttributes</i>, to <i>ABAC Authorization</i> PIM extension <i>ResourceAccessDenyRules</i> that have composition association with the corresponding <i>ABAC Authorization</i> PIM extension <i>MatchedContextAttributes</i>.</p>
MatchedSubjectAttribute	MatchedSubjectAttribute	<p>This ATL rule transforms any composition associations of the <i>ABAC Authorization</i> CIM extension <i>ResourceAccessDenyRules</i> to <i>MatchedSubjectAttributes</i>, to <i>ABAC Authorization</i> PIM extension <i>ResourceAccessDenyRules</i> that have composition association with the corresponding <i>ABAC Authorization</i> PIM extension <i>MatchedSubjectAttribute</i>.</p>
-	AuthorizationDataTable	<p>This ATL rule introduces associations of the <i>ABAC Authorization</i> PIM extension <i>ResourceAccessDenyRules</i> to <i>AuthorizationDataTables</i>, by calling the <i>createAuthorizationDataTable</i> ATL rule.</p>

8.5.20 AllowedAction element transformation

The *createAllowedAction* ATL rule transforms *AllowedAction* elements of the *ABAC Authorization* CIM extension meta-model, to *AllowedAction* ones of the *ABAC Authorization* PIM extension meta-model. Therefore, the *AllowedAction* of the *ABAC Authorization* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *AllowedAction* of the *ABAC Authorization* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-95 *createAllowedAction* ATL rule

AllowedAction property/relation	AllowedAction property/relation	Explanation
AnnCRUDActivity	AnnResourceControllerCRUDActivity	<p>This ATL rule transforms any associations of the <i>ABAC Authorization</i> CIM extension <i>AllowedActions</i> to <i>AnnCRUDActivities</i>, to <i>ABAC Authorization</i> PIM extension <i>AllowedActions</i> that have association with the corresponding <i>ABAC Authorization</i> PIM extension <i>AnnResourceControllerCRUDActivity</i>.</p>
-	AuthorizationDataTable	<p>This ATL rule introduces associations of the <i>ABAC Authorization</i> PIM extension <i>AllowedActions</i> to <i>AuthorizationDataTables</i>, by calling the</p>

		<i>createAuthorizationDataTable</i> ATL rule.
--	--	---

8.5.21 MatchedResourceAttribute element transformation

The *createMatchedResourceAttribute* ATL rule transforms *MatchedResourceAttribute* elements of the *ABAC Authorization* CIM extension meta-model, to *MatchedResourceAttribute* ones of the *ABAC Authorization* PIM extension meta-model. Therefore, the *MatchedResourceAttribute* of the *ABAC Authorization* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *MatchedResourceAttribute* of the *ABAC Authorization* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-96 *createMatchedResourceAttribute* ATL rule

MatchedResourceAttribute property/relation	MatchedResourceAttribute property/relation	Explanation
matchedValue	matchedValue	The <i>matchedValue</i> of the <i>MatchedResourceAttributes</i> of the <i>ABAC Authorization</i> PIM extension equals the <i>matchedValue</i> of the <i>MatchedResourceAttributes</i> of the <i>ABAC Authorization</i> CIM extension.
comparisonOperator	comparisonOperator	The <i>comparisonOperator</i> of the <i>MatchedResourceAttributes</i> of the <i>ABAC Authorization</i> PIM extension equals the <i>comparisonOperator</i> of the <i>MatchedResourceAttributes</i> of the <i>ABAC Authorization</i> CIM extension.
-	AuthorizationDataTable	This ATL rule introduces associations of the <i>ABAC Authorization</i> PIM extension <i>MatchedResourceAttributes</i> to <i>AuthorizationDataTables</i> , by calling the <i>createAuthorizationDataTable</i> ATL rule.
AnnProperty	AnnPIMComponentProperty	This ATL rule transforms any associations of the <i>ABAC Authorization</i> CIM extension <i>MatchedResourceAttributes</i> to <i>AnnProperties</i> , to <i>ABAC Authorization</i> PIM extension <i>MatchedResourceAttributes</i> that have association with the corresponding <i>ABAC Authorization</i> PIM extension <i>AnnPIMComponentProperties</i> .

8.5.22 MatchedContextAttribute element transformation

The *createMatchedContextAttribute* ATL rule transforms *MatchedContextAttribute* elements of the *ABAC Authorization* CIM extension meta-model, to *MatchedContextAttribute* ones of the *ABAC Authorization* PIM extension meta-model. Therefore, the *MatchedContextAttribute* of the *ABAC Authorization* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *MatchedContextAttribute* of the *ABAC Authorization* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-97 createMatchedContextAttribute ATL rule

MatchedContextAttribute property/relation	MatchedContextAttribute property/relation	Explanation
matchedValue	matchedValue	The <i>matchedValue</i> of the <i>MatchedContextAttributes</i> of the <i>ABAC Authorization</i> PIM extension equals the <i>matchedValue</i> of the <i>MatchedContextAttributes</i> of the <i>ABAC Authorization</i> CIM extension.
comparisonOperator	comparisonOperator	The <i>comparisonOperator</i> of the <i>MatchedContextAttributes</i> of the <i>ABAC Authorization</i> PIM extension equals the <i>comparisonOperator</i> of the <i>MatchedContextAttributes</i> of the <i>ABAC Authorization</i> CIM extension.
-	AuthorizationDataTable	This ATL rule introduces associations of the <i>ABAC Authorization</i> PIM extension <i>MatchedContextAttributes</i> to <i>AuthorizationDataTables</i> , by calling the <i>createAuthorizationDataTable</i> ATL rule.
AnnProperty	AnnPIMComponentProperty	This ATL rule transforms any associations of the <i>ABAC Authorization</i> CIM extension <i>MatchedContextAttributes</i> to <i>AnnProperties</i> , to <i>ABAC Authorization</i> PIM extension <i>MatchedContextAttributes</i> that have association with the corresponding <i>ABAC Authorization</i> PIM extension <i>AnnPIMComponentProperties</i> .

8.5.23 **MatchedSubjectAttribute** element transformation

The *createMatchedSubjectAttribute* ATL rule transforms *MatchedSubjectAttribute* elements of the *ABAC Authorization* CIM extension meta-model, to *MatchedSubjectAttribute* ones of the *ABAC Authorization* PIM extension meta-model. Therefore, the *MatchedSubjectAttribute* of the *ABAC Authorization* PIM extension meta-model embeds the abstract envisioned system's design of the conceptual service part modelled with the *MatchedSubjectAttribute* of the *ABAC Authorization* CIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 8-98 createMatchedSubjectAttribute ATL rule

MatchedSubjectAttribute property/relation	MatchedSubjectAttribute property/relation	Explanation
matchedValue	matchedValue	The <i>matchedValue</i> of the <i>MatchedSubjectAttributes</i> of the <i>ABAC Authorization</i> PIM extension equals the <i>matchedValue</i> of the <i>MatchedSubjectAttributes</i> of the <i>ABAC Authorization</i> CIM extension.

comparisonOperator	comparisonOperator	The <i>comparisonOperator</i> of the <i>MatchedSubjectAttributes</i> of the ABAC Authorization PIM extension equals the <i>comparisonOperator</i> of the <i>MatchedSubjectAttributes</i> of the ABAC Authorization CIM extension.
-	AuthorizationDataTable	This ATL rule introduces associations of the ABAC Authorization PIM extension <i>MatchedSubjectAttributes</i> to <i>AuthorizationDataTables</i> , by calling the <i>createAuthorizationDataTable</i> ATL rule.
AnnProperty	AnnPIMComponentProperty	This ATL rule transforms any associations of the ABAC Authorization CIM extension <i>MatchedSubjectAttributes</i> to <i>AnnProperties</i> , to ABAC Authorization PIM extension <i>MatchedSubjectAttributes</i> that have association with the corresponding ABAC Authorization PIM extension <i>AnnPIMComponentProperties</i> .

8.5.24 AuthorizationDataHandler element introduction

The *createAuthorizationDataHandler* ATL rule introduces *AuthorizationDataHandler* elements of the ABAC Authorization PIM extension meta-model. *AuthorizationDataHandlers* do not have ABAC Authorization CIM extension meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-99 *createAuthorizationDataHandler* ATL rule

AuthorizationDataHandler property/relation	Explanation
AnnDatabaseController	This ATL rule introduces associations of the ABAC Authorization PIM extension <i>AuthorizationDataHandlers</i> to <i>AnnDatabaseControllers</i> , by calling the <i>createAnnDatabaseController</i> ATL rule.
AuthorizationDataTable	This ATL rule introduces associations of the ABAC Authorization PIM extension <i>AuthorizationDataHandlers</i> to <i>AuthorizationDataTables</i> , by calling the <i>createAuthorizationDataTable</i> ATL rule.

8.5.25 AuthorizationDataTable element introduction

The *createAuthorizationDataTable* ATL rule introduces *AuthorizationDataTable* elements of the ABAC Authorization PIM extension meta-model. *AuthorizationDataTables* do not have ABAC Authorization

CIM extension meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-100 createAuthorizationDataTable ATL rule

AuthorizationDataTable property/relation	Explanation
AuthorizationDataTableColumn	This ATL rule introduces associations of the <i>ABAC Authorization</i> PIM extension <i>AuthorizationDataTables</i> to <i>AuthorizationDataTableColumns</i> , by calling the <i>createAuthorizationDataTableColumn</i> ATL rule.

8.5.26 AuthorizationDataTableColumn element introduction

The *createAuthorizationDataTableColumn* ATL rule introduces *AuthorizationDataTableColumn* elements of the *ABAC Authorization* PIM extension meta-model. *AuthorizationDataTableColumns* do not have *ABAC Authorization* CIM extension meta-model counterparts. The properties and relations of the rule are defined in the table below:

Table 8-101 createAuthorizationDataTableColumn ATL rule

AuthorizationDataTableColumn property/relation	Explanation
name	The <i>name</i> property of the <i>AuthorizationDataTableColumn</i> element is set equal to the name of the <i>MatchedContextAttribute</i> , <i>MatchedSubjectAttribute</i> or <i>MatchedResourceAttribute</i> depending on the case.
type	The <i>type</i> property of the <i>AuthorizationDataTableColumn</i> element is set equal to the type of the <i>MatchedContextAttribute</i> , <i>MatchedSubjectAttribute</i> or <i>MatchedResourceAttribute</i> depending on the case.
isPrimaryKey	Depending on the case, an <i>AuthorizationDataTableColumn</i> may be set, by this ATL rule, to be a primary key by setting the <i>isPrimaryKey</i> property to be true.
isForeignKey	Depending on the case, an <i>AuthorizationDataTableColumn</i> may be set, by this ATL rule, to be a foreign key by setting the <i>isForeignKey</i> property to be true.

9 PIM extensions to PSM extensions Transformation Definition

9.1 PIM extensions to PSM extensions Transformation Steps

The conceptual steps to perform these ATL transformations are the same with the ones described in section 8.1.

9.2 Authentication Extension PIM to PSM ATL Rules

9.2.1 AnnotationModel element transformation

The *PIMToPSMAuthentication* ATL rule transforms *AnnotationModel* elements of the *Authentication* PIM extension meta-model, to *AnnotationModel* ones of the *Authentication* PSM extension meta-model. Therefore, the *AnnotationModel* of the *Authentication* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AnnotationModel* of the *Authentication* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-1 PIMToPSMAuthentication ATL rule

AnnotationModel property/relation	AnnotationModel property/relation	Explanation
name	name	The <i>name</i> of the <i>AnnotationModel</i> of the <i>Authentication</i> PSM extension equals the name of the <i>AnnotationModel</i> of the <i>Authentication</i> PIM extension.
AnnotatedElement	AnnotatedElement	Every <i>AnnotatedElement</i> of the <i>Authentication</i> PIM extension meta-model is transformed to its <i>Authentication</i> PSM extension meta-model counterpart by calling once the <i>createAnnHTTPActivityHandler</i> ATL rule for every <i>Authentication</i> PIM extension <i>AnnCRUDActivityHandler</i> , once the <i>createAnnJPACController</i> ATL rule for every <i>Authentication</i> PIM extension <i>AnnDatabaseController</i> , and once the <i>createAnnHTTPActivity</i> ATL rule for every <i>Authentication</i> PIM extension <i>AnnCRUDActivity</i> .
Annotation	Annotation	Every <i>Annotation</i> of the <i>Authentication</i> PIM extension meta-model is transformed to its <i>Authentication</i> PSM extension meta-model counterpart by calling once the <i>createGuestMode</i> ATL rule for every <i>Authentication</i> PIM extension <i>GuestMode</i> , once the <i>createAuthenticationOnlyMode</i> ATL rule for every <i>Authentication</i> PIM extension <i>AuthenticationMode</i> , once the <i>createBothMode</i> ATL rule for every <i>Authentication</i> PIM extension <i>BothMode</i> and once the <i>createAuthenticationPerformer</i> ATL rule.

9.2.2 AnnDatabaseController element transformation

The *createAnnJPAController* ATL rule transforms *AnnDatabaseController* elements of the *Authentication* PIM extension meta-model, to *AnnJPAController* ones of the *Authentication* PSM extension meta-model. Therefore, the *AnnJPAController* of the *Authentication* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AnnDatabaseController* of the *Authentication* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-2 createAnnJPAController ATL rule

AnnDatabaseController property/relation	AnnJPAController property/relation	Explanation
DatabaseController	JPAController	This ATL rule transforms any references of the <i>Authentication</i> PIM extension <i>AnnDatabaseControllers</i> to Core PIM <i>DatabaseControllers</i> to <i>Authentication</i> PSM extension <i>AnnJPAControllers</i> that reference the corresponding Core PSM <i>JPAControllers</i> .

9.2.3 AnnCRUDActivity element transformation

The *createAnnHTTPActivity* ATL rule transforms *AnnCRUDActivity* elements of the *Authentication* PIM extension meta-model, to *AnnHTTPActivity* ones of the *Authentication* PSM extension meta-model. Therefore, the *AnnHTTPActivity* of the *Authentication* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AnnCRUDActivity* of the *Authentication* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-3 createAnnHTTPActivity ATL rule

AnnCRUDActivity property/relation	AnnHTTPActivity property/relation	Explanation
CRUDActivity	HTTPActivity	This ATL rule transforms any references of the <i>Authentication</i> PIM extension <i>AnnCRUDActivities</i> to Core PIM <i>CRUDActivities</i> to <i>Authentication</i> PSM extension <i>AnnHTTPActivities</i> that reference the corresponding Core PSM <i>HTTPActivities</i> .

9.2.4 AnnCRUDActivityHandler element transformation

The *createAnnHTTPActivityHandler* ATL rule transforms *AnnCRUDActivityHandler* elements of the *Authentication* PIM extension meta-model, to *AnnHTTPActivityHandler* ones of the *Authentication* PSM extension meta-model. Therefore, the *AnnHTTPActivityHandler* of the *Authentication* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's

design part modelled with the *AnnCRUDActivityHandler* of the *Authentication* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-4 createAnnHTTPActivityHandler ATL rule

AnnCRUDActivityHandler property/relation	AnnHTTPActivityHandler property/relation	Explanation
CRUDActivityHandler	HTTPActivityHandler	This ATL rule transforms any references of the <i>Authentication</i> PIM extension <i>AnnCRUDActivityHandlers</i> to Core PIM <i>CRUDActivityHandlers</i> to <i>Authentication</i> PSM extension <i>AnnHTTPActivityHandlers</i> that reference the corresponding Core PSM <i>HTTPActivityHandlers</i> .

9.2.5 AuthenticationPerformer element transformation

The *createAuthenticationPerformer* ATL rule transforms *AuthenticationPerformer* elements of the *Authentication* PIM extension meta-model, to *AuthenticationPerformer* ones of the *Authentication* PSM extension meta-model. Therefore, the *AuthenticationPerformer* of the *Authentication* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AuthenticationPerformer* of the *Authentication* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-5 createAuthenticationPerformer ATL rule

AuthenticationPerformer property/relation	AuthenticationPerformer property/relation	Explanation
authentication modelName	authentication modelName	The <i>authentication modelName</i> of the <i>AnnotationModel</i> of the <i>Authentication</i> PSM extension equals the <i>authentication modelName</i> of the <i>AnnotationModel</i> of the <i>Authentication</i> PIM extension.
authentication model ParentName	authentication model ParentName	The <i>authentication model ParentName</i> of the <i>AnnotationModel</i> of the <i>Authentication</i> PSM extension equals the <i>authentication model ParentName</i> of the <i>AnnotationModel</i> of the <i>Authentication</i> PIM extension.
AnnDatabaseController	AnnJPAController	This ATL rule transforms any associations of the <i>Authentication</i> PIM extension <i>AuthenticationPerformers</i> to <i>AnnDatabaseControllers</i> , to <i>Authentication</i> PSM extension <i>AnnJPAControllers</i> , by calling the <i>createAnnJPAController</i> ATL rule.
AuthenticationToken	AuthenticationToken	This ATL rule transforms any associations of the <i>Authentication</i> PIM extension <i>AuthenticationPerformers</i> to

		<i>AuthenticationTokens</i> , to <i>Authentication</i> PSM extension <i>AnnJPACollectors</i> , by calling the <i>createAuthenticationToken</i> or <i>createPassword</i> ATL rule depending on the case.
--	--	---

9.2.6 AuthenticationToken element transformation

The *createAuthenticationToken* ATL rule transforms *AuthenticationToken* elements of the *Authentication* PIM extension meta-model, to *AuthenticationToken* ones of the *Authentication* PSM extension meta-model. Therefore, the *AuthenticationToken* of the *Authentication* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AuthenticationToken* of the *Authentication* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-6 *createAuthenticationToken* ATL rule

AuthenticationToken property/relation	AuthenticationToken property/relation	Explanation
name	name	The <i>name</i> of the <i>AuthenticationToken</i> of the <i>Authentication</i> PSM extension equals the name of the <i>AuthenticationToken</i> of the <i>Authentication</i> PIM extension.
type	type	The <i>type</i> of the <i>AuthenticationToken</i> of the <i>Authentication</i> PSM extension equals the type of the <i>AuthenticationToken</i> of the <i>Authentication</i> PIM extension.
isUnique	isUnique	The <i>isUnique</i> property of the <i>AuthenticationToken</i> of the <i>Authentication</i> PSM extension equals the <i>isUnique</i> property of the <i>AuthenticationToken</i> of the <i>Authentication</i> PIM extension.

9.2.7 Password element transformation

The *createPassword* ATL rule transforms *Password* elements of the *Authentication* PIM extension meta-model, to *Password* ones of the *Authentication* PSM extension meta-model. Therefore, the *Password* of the *Authentication* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *Password* of the *Authentication* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-7 *createPassword* ATL rule

Password property/relation	Password property/relation	Explanation
name	name	The <i>name</i> of the <i>Password</i> of the <i>Authentication</i> PSM extension equals the name of the <i>Password</i>

		of the <i>Authentication</i> PIM extension.
type	type	The <i>type</i> of the <i>Password</i> of the <i>Authentication</i> PSM extension equals the <i>type</i> of the <i>Password</i> of the <i>Authentication</i> PIM extension.
isUnique	isUnique	The <i>isUnique</i> property of the <i>Password</i> of the <i>Authentication</i> PSM extension equals the <i>isUnique</i> property of the <i>Password</i> of the <i>Authentication</i> PIM extension.

9.2.8 GuestMode element transformation

The *createGuestMode* ATL rule transforms *GuestMode* elements of the *Authentication* PIM extension meta-model, to *GuestMode* ones of the *Authentication* PSM extension meta-model. Therefore, the *GuestMode* of the *Authentication* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *GuestMode* of the *Authentication* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-8 *createGuestMode* ATL rule

GuestMode property/relation	GuestMode property/relation	Explanation
AnnCRUDActivity	AnnHTTPActivity	This ATL rule transforms any associations of the <i>Authentication</i> PIM extension <i>GuestModes</i> to <i>AnnCRUDActivities</i> , to <i>Authentication</i> PSM extension <i>AnnHTTPActivities</i> of <i>GuestModes</i> , by calling the <i>createAnnHTTPActivity</i> ATL rule.
AnnCRUDActivityHandlers	AnnHTTPActivityHandlers	This ATL rule transforms any associations of the <i>Authentication</i> PIM extension <i>GuestModes</i> to <i>AnnCRUDActivityHandlers</i> , to <i>Authentication</i> PSM extension <i>AnnHTTPActivityHandlers</i> of <i>GuestModes</i> , by calling the <i>createAnnHTTPActivityHandler</i> ATL rule.

9.2.9 AuthenticationOnlyMode element transformation

The *createAuthenticationOnlyMode* ATL rule transforms *AuthenticationOnlyMode* elements of the *Authentication* PIM extension meta-model, to *AuthenticationOnlyMode* ones of the *Authentication* PSM extension meta-model. Therefore, the *AuthenticationOnlyMode* of the *Authentication* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AuthenticationOnlyMode* of the *Authentication* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-9 *createAuthenticationOnlyMode* ATL rule

AuthenticationOnlyMode property/relation	AuthenticationOnlyMode property/relation	Explanation
--	--	-------------

AnnCRUDActivity	AnnHTTPActivity	This ATL rule transforms any associations of the <i>Authentication</i> PIM extension <i>AuthenticationOnlyModes</i> to <i>AnnCRUDActivities</i> , to <i>Authentication</i> PSM extension <i>AnnHTTPActivities</i> of <i>AuthenticationOnlyModes</i> , by calling the <i>createAnnHTTPActivity</i> ATL rule.
AnnCRUDActivityHandlers	AnnHTTPActivityHandlers	This ATL rule transforms any associations of the <i>Authentication</i> PIM extension <i>AuthenticationOnlyModes</i> to <i>AnnCRUDActivityHandlers</i> , to <i>Authentication</i> PSM extension <i>AnnHTTPActivityHandlers</i> of <i>AuthenticationOnlyModes</i> , by calling the <i>createAnnHTTPActivityHandler</i> ATL rule.

9.2.10 BothMode element transformation

The *createBothMode* ATL rule transforms *BothMode* elements of the *Authentication* PIM extension meta-model, to *BothMode* ones of the *Authentication* PSM extension meta-model. Therefore, the *BothMode* of the *Authentication* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *BothMode* of the *Authentication* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-10 createBothMode ATL rule

BothMode property/relation	BothMode property/relation	Explanation
AnnCRUDActivity	AnnHTTPActivity	This ATL rule transforms any associations of the <i>Authentication</i> PIM extension <i>BothModes</i> to <i>AnnCRUDActivities</i> , to <i>Authentication</i> PSM extension <i>AnnHTTPActivities</i> of <i>BothModes</i> , by calling the <i>createAnnHTTPActivity</i> ATL rule.
AnnCRUDActivityHandlers	AnnHTTPActivityHandlers	This ATL rule transforms any associations of the <i>Authentication</i> PIM extension <i>BothModes</i> to <i>AnnCRUDActivityHandlers</i> , to <i>Authentication</i> PSM extension <i>AnnHTTPActivityHandlers</i> of <i>BothModes</i> , by calling the <i>createAnnHTTPActivityHandler</i> ATL rule.

9.3 Database Searching Extension PIM to PSM ATL Rules

9.3.1 AnnotationModel element transformation

The *SearchLayerPIMToPSM* ATL rule transforms *AnnotationModel* elements of the *Database Searching* PIM extension meta-model, to *AnnotationModel* ones of the *Database Searching* PSM extension meta-model. Therefore, the *AnnotationModel* of the *Database Searching* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AnnotationModel* of the *Database Searching* PIM extension meta-model

element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-11 SearchLayerPIMToPSM ATL rule

AnnotationModel property/relation	AnnotationModel property/relation	Explanation
name	name	The <i>name</i> of the <i>AnnotationModel</i> of the <i>Database Searching PSM</i> extension is the same with the name of the <i>Database Searching PIM</i> extension <i>AnnotationModel</i> .
AnnotatedElement	AnnotatedElement	Every <i>AnnotatedElement</i> of the <i>Database Searching PIM</i> extension meta-model is transformed to its <i>Database Searching PSM</i> extension meta-model counterpart by calling either the <i>createAnnPSMComponentProperty</i> ATL rule for every <i>Database Searching PIM</i> extension <i>AnnPIMComponentProperty</i> , or the <i>createAnnJavaAlgoController</i> ATL rule for every <i>Database Searching PIM</i> extension <i>AnnAlgoResourceController</i> , or the <i>createAnnHTTPActivity</i> and <i>createAnnHTTPActivityHandler</i> one for every <i>Database Searching PIM</i> extension <i>AnnCRUDactivity</i> or <i>AnnCRUDActivityHandler</i> respectively. Finally, by calling the <i>createAnnJavaResourceModel</i> ATL rule for every <i>Database Searching PIM</i> extension <i>AnnResourceModel</i> .
Annotation	Annotation	Every <i>Annotation</i> of the <i>Database Searching PIM</i> extension meta-model is transformed to its <i>Database Searching PSM</i> extension meta-model counterpart by calling once the <i>createPSMSearchController</i> ATL rule, the <i>createSearchHTTPActivity</i> as well as the <i>createSearchHTTPActivityHandler</i> one for every <i>Database Searching PIM</i> extension <i>SearchController</i> , <i>SearchCRUDActivity</i> and <i>SearchCRUDActivityHandler</i> respectively.

9.3.2 AnnPIMComponentProperty element transformation

The *createPSMAnnComponentProperty* ATL rule transforms *AnnPIMComponentProperty* elements of the *Database Searching PIM* extension meta-model, to *AnnPSMComponentProperty* ones of the *Database Searching PSM* extension meta-model. Therefore, the *AnnPSMComponentProperty* of the *Database Searching PSM* extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AnnPIMComponentProperty* of the *Database Searching PIM* extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-12 createAnnPSMComponentProperty ATL rule

AnnPIMComponentProperty property/relation	AnnPSMComponentProperty property/relation	Explanation
PIMComponentProperty	PSMComponentProperty	This ATL rule transforms any references of the <i>Database Searching</i> PIM extension <i>AnnPIMComponentProperties</i> to Core PIM <i>PIMComponentProperties</i> to <i>Database Searching</i> PSM extension <i>AnnPSMComponentProperties</i> that reference the corresponding Core PSM <i>PSMComponentProperties</i> .

9.3.3 AnnAlgoResourceController element transformation

The *createAnnJavaAlgoController* ATL rule transforms *AnnAlgoResourceController* elements of the *Database Searching* PIM extension meta-model, to *AnnJavaAlgoResourceController* ones of the *Database Searching* PSM extension meta-model. Therefore, the *AnnJavaAlgoResourceController* of the *Database Searching* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AnnAlgoResourceController* of the *Database Searching* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-13 createAnnJavaAlgoController ATL rule

AnnAlgoResourceController property/relation	AnnJavaAlgoResourceController property/relation	Explanation
AlgoResourceController	JavaAlgoResourceController	This ATL rule transforms any references of the <i>Database Searching</i> PIM extension <i>AnnAlgoResourceController</i> to Core PIM <i>AlgoResourceControllers</i> to <i>Database Searching</i> PSM extension <i>AnnJavaAlgoResourceControllers</i> that reference the corresponding Core PSM <i>JavaAlgoResourceController</i> .

9.3.4 AnnCRUDActivity element transformation

The *createAnnHTTPActivity* ATL rule transforms *AnnCRUDActivity* elements of the *Database Searching* PIM extension meta-model, to *AnnHTTPActivity* ones of the *Database Searching* PSM extension meta-model. Therefore, the *AnnHTTPActivity* of the *Database Searching* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AnnCRUDActivity* of the *Database Searching* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-14 createAnnHTTPActivity ATL rule

AnnCRUDActivity property/relation	AnnHTTPActivity property/relation	Explanation

CRUDActivity	HTTPActivity	This ATL rule transforms any references of the <i>Database Searching</i> PIM extension <i>AnnCRUDActivities</i> to Core PIM <i>CRUDActivities</i> to <i>Database Searching</i> PSM extension <i>AnnHTTPActivities</i> that reference the corresponding Core PSM <i>HTTPActivities</i> .
--------------	--------------	---

9.3.5 AnnCRUDActivityHandler element transformation

The *createAnnHTTPActivityHandler* ATL rule transforms *AnnCRUDActivityHandler* elements of the *Database Searching* PIM extension meta-model, to *AnnHTTPActivityHandler* ones of the *Database Searching* PSM extension meta-model. Therefore, the *AnnHTTPActivityHandler* of the *Database Searching* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AnnCRUDActivityHandler* of the *Database Searching* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-15 *createAnnHTTPActivityHandler* ATL rule

AnnCRUDActivityHandler property/relation	AnnHTTPActivityHandler property/relation	Explanation
CRUDActivityHandler	HTTPActivityHandler	This ATL rule transforms any references of the <i>Database Searching</i> PIM extension <i>AnnCRUDActivityHandlers</i> to Core PIM <i>CRUDActivityHandlers</i> to <i>Database Searching</i> PSM extension <i>AnnHTTPActivityHandlers</i> that reference the corresponding Core PSM <i>HTTPActivityHandlers</i> .

9.3.6 AnnResourceModel element transformation

The *createAnnJavaResourceModel* ATL rule transforms *AnnResourceModel* elements of the *Database Searching* PIM extension meta-model, to *AnnJavaResourceModel* ones of the *Database Searching* PSM extension meta-model. Therefore, the *AnnJavaResourceModel* of the *Database Searching* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AnnResourceModel* of the *Database Searching* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-16 *createAnnJavaResourceModel* ATL rule

AnnResourceModel property/relation	AnnJavaResourceModel property/relation	Explanation
ResourceModel	JavaResourceModel	This ATL rule transforms any references of the <i>Database Searching</i> PIM extension <i>AnnResourceModels</i> to Core PIM <i>ResourceModels</i> to <i>Database Searching</i> PSM extension <i>AnnJavaResourceModels</i> that reference the

		corresponding Core PSM JavaResourceModels.
--	--	--

9.3.7 SearchController element transformation

The *createPSMSearchController* ATL rule transforms *SearchController* elements of the *Database Searching* PIM extension meta-model, to *SearchController* ones of the *Database Searching* PSM extension meta-model. Therefore, the *SearchController* of the *Database Searching* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *SearchController* of the *Database Searching* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-17 createPSMSearchController ATL rule

SearchController property/relation	SearchController property/relation	Explanation
AnnAlgoResourceController	AnnJavaAlgoResourceController	This ATL rule transforms any associations of the <i>Database Searching</i> PIM extension <i>SearchControllers</i> to <i>AnnAlgoResourceControllers</i> to <i>Database Searching</i> PSM extension <i>SearchControllers</i> that have associations with the corresponding <i>AnnJavaAlgoResourceControllers</i> .

9.3.8 SearchCRUDActivity element transformation

The *createSearchHTTPActivity* ATL rule transforms *SearchCRUDActivity* elements of the *Database Searching* PIM extension meta-model, to *SearchHTTPActivity* ones of the *Database Searching* PSM extension meta-model. Therefore, the *SearchHTTPActivity* of the *Database Searching* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *SearchCRUDActivity* of the *Database Searching* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-18 createSearchHTTPActivity ATL rule

SearchCRUDActivity property/relation	SearchHTTPActivity property/relation	Explanation
AnnCRUDActivity	AnnHTTPActivity	This ATL rule transforms any associations of the <i>Database Searching</i> PIM extension <i>SearchCRUDActivities</i> to <i>AnnCRUDActivities</i> , to <i>Database Searching</i> PSM extension <i>SearchHTTPActivities</i> that have associations with the corresponding <i>AnnHTTPActivities</i> .

9.3.9 SearchCRUDActivityHandler element transformation

The *createSearchHTTPActivityhandler* ATL rule transforms *SearchCRUDActivityHandler* elements of the *Database Searching* PIM extension meta-model, to *SearchHTTPHandler* ones of the *Database Searching* PSM extension meta-model. Therefore, the *SearchHTTPHandler* of the *Database Searching* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *SearchCRUDActivityHandler* of the *Database Searching* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-19 *createSearchHTTPActivityHandler* ATL rule

SearchCRUDActivityHandler property/relation	SearchHTTPHandler property/relation	Explanation
AnnCRUDActivityHandler	AnnHTTPActivityHandler	This ATL rule transforms any associations of the <i>Database Searching</i> PIM extension <i>SearchCRUDActivityHandlers</i> to <i>AnnCRUDActivityHandlers</i> , to <i>Database Searching</i> PSM extension <i>SearchHTTPHandlers</i> that have associations with the corresponding <i>AnnHTTPActivityHandlers</i> .
SearchableResourceModel	SearchableJavaResourceModel	This ATL rule transforms any composition associations of the <i>Database Searching</i> PIM extension <i>SearchCRUDActivityhandlers</i> to <i>SearchableResourceModels</i> , to <i>Database Searching</i> PSM extension <i>SearchHTTPHandlers</i> that have composition associations with the corresponding <i>SearchableJavaResourceModels</i> .

9.3.10 SearchableResourceModel element transformation

The *createPSMSelectableResource* ATL rule transforms *SearchableResourceModel* elements of the *Database Searching* PIM extension meta-model, to *SearchableJavaResourceModel* ones of the *Database Searching* PSM extension meta-model. Therefore, the *SearchableJavaResourceModel* of the *Database Searching* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *SearchableResourceModel* of the *Database Searching* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-20 *createPSMSelectableResource* ATL rule

SearchableResourceModel property/relation	SearchableJavaResourceModel property/relation	Explanation
AnnResourceModel	AnnJavaResourceModel	This ATL rule transforms any associations of the <i>Database Searching</i> PIM extension <i>SearchableResourceModels</i> to <i>AnnResourceModels</i> , to <i>Database Searching</i> PSM extension <i>SearchableJavaResourceModels</i> that have associations with the corresponding <i>AnnJavaResourceModels</i> .

SearchableProperty	SearchableProperty	This ATL rule transforms any composition associations of the <i>Database Searching</i> PIM extension <i>SearchableResourceModels</i> to <i>SearchableProperties</i> , to <i>Database Searching</i> PSM extension <i>SearchableJavaResourceModels</i> that have composition associations with the corresponding <i>SearchableProperties</i> .
--------------------	--------------------	--

9.3.11 SearchableProperty element transformation

The *createPSMSearchableProperty* ATL rule transforms *SearchableProperty* elements of the *Database Searching* PIM extension meta-model, to *SearchableProperty* ones of the *Database Searching* PSM extension meta-model. Therefore, the *SearchableProperty* of the *Database Searching* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *SearchableProperty* of the *Database Searching* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-21 *createPSMSearchableProperty* ATL rule

SearchableProperty property/relation	SearchableProperty property/relation	Explanation
AnnPIMComponentProperty	AnnPSMComponentProperty	This ATL rule transforms any associations of the <i>Database Searching</i> PIM extension <i>SearchableProperties</i> to <i>AnnPIMComponentProperties</i> , to <i>Database Searching</i> PSM extension <i>SearchableProperties</i> that have associations with the corresponding <i>AnnPSMComponentProperties</i> .

9.4 External Service Composition Extension PIM to PSM ATL Rules

9.4.1 AnnotationModel element transformation

The *ExternalServicePIMToPSM* ATL rule transforms *AnnotationModel* elements of the *External Service Composition* PIM extension meta-model, to *AnnotationModel* ones of the *External Service Composition* PSM extension meta-model. Therefore, the *AnnotationModel* of the *External Service Composition* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AnnotationModel* of the *External Service Composition* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-22 *ExternalServicePIMToPSM* ATL rule

AnnotationModel property/relation	AnnotationModel property/relation	Explanation
-----------------------------------	-----------------------------------	-------------

name	name	The <i>name</i> of the <i>AnnotationModel</i> of the <i>External Service Composition PSM</i> extension is the same with the <i>name</i> of the <i>External Service Composition PIM</i> extension <i>AnnotationModel</i> .
AnnotatedElement	AnnotatedElement	Every <i>AnnotatedElement</i> of the <i>External Service Composition PIM</i> extension meta-model is transformed to its <i>External Service Composition PSM</i> extension meta-model counterpart by calling either the <i>createAnnJavaAlgoModel</i> ATL rule for every <i>External Service Composition PIM</i> extension <i>AnnAlgoResourceModel</i> , the <i>createAnnJavaResourceModel</i> , the <i>createAnnJavaAlgoController</i> , the <i>createAnnHTTPActivity</i> and the <i>createAnnHTTPActivityHandler</i> one for every <i>External Service Composition PIM</i> extension <i>AnnResourceModel</i> , <i>AnnAlgoResourceController</i> , <i>AnnCRUDActivity</i> and <i>AnnCRUDActivityHandler</i> respectively.
Annotation	Annotation	Every <i>Annotation</i> of the <i>External Service Composition PIM</i> extension meta-model is transformed to its <i>External Service Composition PSM</i> extension meta-model counterpart by calling once the <i>createJavaRESTClientController</i> ATL rule as well as the <i>createJavaRESTClientModel</i> one for every <i>External Service Composition PIM</i> extension <i>RESTClientController</i> or <i>RESTClientModel</i> respectively.

9.4.2 AnnAlgoResourceModel element transformation

The *createAnnJavaAlgoModel* ATL rule transforms *AnnAlgoResourceModel* elements of the *External Service Composition PIM* extension meta-model, to *AnnJavaAlgoModel* ones of the *External Service Composition PSM* extension meta-model. Therefore, the *AnnJavaAlgoModel* of the *External Service Composition PSM* extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AnnAlgoResourceModel* of the *External Service Composition PIM* extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-23 *createAnnJavaAlgoModel* ATL rule

AnnAlgoResourceModel property/relation	AnnJavaAlgoModel property/relation	Explanation
AlgoResourceModel	JavaAlgoResourceModel	This ATL rule transforms any references of the <i>External Service Composition PIM</i> extension <i>AnnAlgoResourceModels</i> to Core PIM <i>AlgoResourceModels</i> to <i>External Service Composition PSM</i> extension <i>AnnJavaAlgoModels</i> that reference the corresponding Core PSM <i>JavaAlgoResourceModels</i> .

9.4.3 AnnResourceModel element transformation

The *createAnnJavaResourceModel* ATL rule transforms *AnnResourceModel* elements of the *External Service Composition* PIM extension meta-model, to *AnnJavaResourceModel* ones of the *External Service Composition* PSM extension meta-model. Therefore, the *AnnJavaResourceModel* of the *External Service Composition* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AnnResourceModel* of the *External Service Composition* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-24 *createAnnJavaResourceModel* ATL rule

AnnResourceModel property/relation	AnnJavaResourceModel property/relation	Explanation
ResourceModel	JavaResourceModel	This ATL rule transforms any references of the <i>External Service Composition</i> PIM extension <i>AnnResourceModels</i> to Core PIM <i>ResourceModels</i> to <i>External Service Composition</i> PSM extension <i>AnnJavaResourceModels</i> that reference the corresponding Core PSM <i>JavaResourceModel</i> .

9.4.4 AnnAlgoResourceController element transformation

The *createAnnJavaAlgoController* ATL rule transforms *AnnAlgoResourceController* elements of the *External Service Composition* PIM extension meta-model, to *AnnJavaAlgoController* ones of the *External Service Composition* PSM extension meta-model. Therefore, the *AnnJavaAlgoController* of the *External Service Composition* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AnnAlgoResourceController* of the *External Service Composition* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-25 *createAnnJavaAlgoController* ATL rule

AnnAlgoResourceController property/relation	AnnJavaAlgoController property/relation	Explanation
AlgoResourceController	JavaAlgoResourceController	This ATL rule transforms any references of the <i>External Service Composition</i> PIM extension <i>AnnAlgoResourceController</i> s to Core PIM <i>AlgoResourceController</i> s to <i>External Service Composition</i> PSM extension <i>AnnJavaAlgoController</i> s that reference the corresponding Core PSM <i>JavaAlgoResourceController</i> s.

9.4.5 AnnCRUDActivity element transformation

The *createAnnHTTPActivity* ATL rule transforms *AnnCRUDActivity* elements of the *External Service Composition* PIM extension meta-model, to *AnnHTTPActivity* ones of the *External Service Composition* PSM extension meta-model. Therefore, the *AnnHTTPActivity* of the *External Service Composition* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AnnCRUDActivity* of the *External Service Composition* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-26 createAnnHTTPActivity ATL rule

AnnCRUDActivity property/relation	AnnHTTPActivity property/relation	Explanation
ResourceControllerCRUDActivity	HTTPActivity	This ATL rule transforms any references of the <i>External Service Composition</i> PIM extension <i>AnnCRUDActivities</i> to Core PIM <i>ResourceControllerCRUDActivities</i> to <i>External Service Composition</i> PSM extension <i>AnnHTTPActivities</i> that reference the corresponding Core PSM <i>HTTPActivities</i> .

9.4.6 AnnCRUDActivityHandler element transformation

The *createAnnHTTPActivityHandler* ATL rule transforms *AnnCRUDActivityHandler* elements of the *External Service Composition* PIM extension meta-model, to *AnnHTTPActivityHandler* ones of the *External Service Composition* PSM extension meta-model. Therefore, the *AnnHTTPActivityHandler* of the *External Service Composition* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AnnCRUDActivityHandler* of the *External Service Composition* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-27 createAnnHTTPActivityHandler ATL rule

AnnCRUDActivityHandler property/relation	AnnHTTPActivityHandler property/relation	Explanation
CRUDActivityHandler	HTTPActivityHandler	This ATL rule transforms any references of the <i>External Service Composition</i> PIM extension <i>AnnCRUDActivityHandlers</i> to Core PIM <i>CRUDActivityHandlers</i> to <i>External Service Composition</i> PSM extension <i>AnnHTTPActivityHandlers</i> that reference the corresponding Core PSM <i>HTTPActivityHandlers</i> .

9.4.7 RESTClientController element transformation

The *createJavaRESTClientController* ATL rule transforms *RESTClientController* elements of the *External Service Composition* PIM extension meta-model, to *JavaRESTClientController* ones of the

External Service Composition PSM extension meta-model. Therefore, the *JavaRESTClientController* of the *External Service Composition* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *RESTClientController* of the *External Service Composition* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-28 createJavaRESTClientController ATL rule

RESTClientController property/relation	JavaRESTClientController property/relation	Explanation
AnnAlgoResourceController	AnnJavaAlgoController	This ATL rule transforms any associations of the <i>External Service Composition</i> PIM extension <i>RESTClientControllers</i> to <i>AnnAlgoResourceControllers</i> , to <i>External Service Composition</i> PSM extension <i>JavaRESTClientController</i> that have associations with the corresponding <i>AnnJavaAlgoController</i> .
RESTClientCRUDActivity	JavaRESTClientHTTPActivity	This ATL rule transforms any composition associations of the <i>External Service Composition</i> PIM extension <i>RESTClientControllers</i> to <i>RESTClientCRUDActivities</i> , to <i>External Service Composition</i> PSM extension <i>JavaRESTClientControllers</i> that have composition associations with the corresponding <i>JavaRESTClientHTTPActivities</i> .
RESTClientModel	JavaRESTClientModel	This ATL rule transforms any associations of the <i>External Service Composition</i> PIM extension <i>RESTClientControllers</i> to <i>RESTClientModels</i> , to <i>External Service Composition</i> PSM extension <i>JavaRESTClientControllers</i> that have associations with the corresponding <i>JavaRESTClientModels</i> .

9.4.8 RESTClientCRUDActivity element transformation

The *createJavaRESTClientHTTPActivity* ATL rule transforms *RESTClientCRUDActivity* elements of the *External Service Composition* PIM extension meta-model, to *JavaRESTClientHTTPActivity* ones of the *External Service Composition* PSM extension meta-model. Therefore, the *JavaRESTClientHTTPActivity* of the *External Service Composition* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *RESTClientCRUDActivity* of the *External Service Composition* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-29 createJavaRESTClientHTTPActivity ATL rule

RESTClientCRUDActivity property/relation	JavaRESTClientHTTPActivity property/relation	Explanation
AnnCRUDActivity	AnnHTTPActivity	This ATL rule transforms any associations of the <i>External Service Composition</i> PIM extension <i>RESTClientCRUDActivities</i> to <i>AnnCRUDActivities</i> ,

		to <i>External Service Composition</i> PSM extension <i>JavaRESTClientHTTPActivities</i> that have associations with the corresponding <i>AnnHTTPActivities</i> .
RESTClientCRUDActivityHandler	JavaRESTClientCRUDActivityHandler	This ATL rule transforms any composition associations of the <i>External Service Composition</i> PIM extension <i>RESTClientCRUDActivities</i> to <i>RESTClientCRUDActivityHandlers</i> , to <i>External Service Composition</i> PSM extension <i>JavaRESTClientHTTPActivities</i> that have composition associations with the corresponding <i>JavaRESTClientCRUDActivityHandlers</i> .
QueryParam	QueryParam	This ATL rule transforms any associations of the <i>External Service Composition</i> PIM extension <i>RESTClientCRUDActivities</i> to <i>QueryParams</i> , to <i>External Service Composition</i> PSM extension <i>JavaRESTClientHTTPActivities</i> that have associations with the corresponding <i>QueryParams</i> .

9.4.9 RESTClientCRUDActivityHandler element transformation

The *createJavaRESTClientHTTPActivityHandler* ATL rule transforms *RESTClientCRUDActivityHandlers* elements of the *External Service Composition* PIM extension meta-model, to *JavaRESTClientHTTPActivityHandlers* ones of the *External Service Composition* PSM extension meta-model. Therefore, the *JavaRESTClientHTTPActivityHandlers* of the *External Service Composition* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *RESTClientCRUDActivityHandlers* of the *External Service Composition* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-30 *createJavaRESTClientHTTPActivityHandler* ATL rule

RESTClientCRUDActivityHandler property/relation	JavaRESTClientHTTPActivityHandler property/relation	Explanation
AnnCRUDActivityHandler	AnnHTTPActivityHandler	This ATL rule transforms any associations of the <i>External Service Composition</i> PIM extension <i>RESTClientCRUDActivityHandlers</i> to <i>AnnCRUDActivityHandlers</i> , to <i>External Service Composition</i> PSM extension <i>JavaRESTClientHTTPActivityHandlers</i> that have associations with the corresponding <i>AnnHTTPActivityHandlers</i> .

9.4.10 QueryParam element transformation

The *createQueryParam* ATL rule transforms *QueryParam* elements of the *External Service Composition* PIM extension meta-model, to *QueryParam* ones of the *External Service Composition* PSM extension meta-model. Therefore, the *QueryParam* of the *External Service Composition* PSM

extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *QueryParam* of the *External Service Composition* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-31 createQueryParam ATL rule

QueryParam property/relation	QueryParam property/relation	Explanation
name	name	The <i>name</i> of the <i>QueryParam</i> of the <i>External Service Composition</i> PSM extension equals the <i>name</i> of the <i>QueryParam</i> of the <i>External Service Composition</i> PIM extension.
type	type	The <i>type</i> of the <i>QueryParam</i> of the <i>External Service Composition</i> PSM extension equals the <i>type</i> of the <i>QueryParam</i> of the <i>External Service Composition</i> PIM extension.

9.4.11 RESTClientModel element transformation

The *createJavaRESTClientModel* ATL rule transforms *RESTClientModel* elements of the *External Service Composition* PIM extension meta-model, to *JavaRESTClientModel* ones of the *External Service Composition* PSM extension meta-model. Therefore, the *JavaRESTClientModel* of the *External Service Composition* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *RESTClientModel* of the *External Service Composition* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-32 createJavaRESTClientModel ATL rule

RESTClientModel property/relation	JavaRESTClientModel property/relation	Explanation
AnnAlgoResourceModel	AnnJavaAlgoModel	This ATL rule transforms any associations of the <i>External Service Composition</i> PIM extension <i>RESTClientModel</i> to <i>AnnAlgoResourceModels</i> , to <i>External Service Composition</i> PSM extension <i>JavaRESTClientModels</i> that have associations with the corresponding <i>AnnJavaAlgoModels</i> .
InputDataModel	InputDataModel	This ATL rule transforms any composition associations of the <i>External Service Composition</i> PIM extension <i>RESTClientModel</i> to <i>InputDataModels</i> , to <i>External Service Composition</i> PSM extension <i>JavaRESTClientModels</i> that have composition associations with the corresponding <i>InputDataModels</i> .
OutputDataModel	OutputDataModel	This ATL rule transforms any associations of the <i>External Service Composition</i> PIM extension <i>RESTClientModel</i> to <i>OutputDataModels</i> , to <i>External Service Composition</i> PSM extension <i>JavaRESTClientModels</i> that have associations with

		the corresponding <i>OutputModels</i> , by calling either the <i>createNonPersistentOutput</i> , or <i>createAutoPersistentOutput</i> , or <i>createExistenJavaModelPersitentOutput</i> one ATL rule according to the type of the input <i>OutputDataModel</i> .
--	--	--

9.4.12 InputDataModel element transformation

The *createJavaInputDataModel* ATL rule transforms *InputDataModel* elements of the *External Service Composition* PIM extension meta-model, to *InputDataModel* ones of the *External Service Composition* PSM extension meta-model. Therefore, the *InputDataModel* of the *External Service Composition* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *InputDataModel* of the *External Service Composition* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-33 *createJavaInputDataModel* ATL rule

InputDataModel property/relation	JavaInputDataModel property/relation	Explanation
Property	Property	This ATL rule transforms any composition associations of the <i>External Service Composition</i> PIM extension <i>InputDataModels</i> to <i>Properties</i> , to <i>External Service Composition</i> PSM extension <i>InputDataModels</i> that have composition associations with the corresponding <i>Properties</i> .
Representation	Representation	This ATL rule transforms any composition associations of the <i>External Service Composition</i> PIM extension <i>InputDataModels</i> to <i>Representations</i> , to <i>External Service Composition</i> PSM extension <i>InputDataModels</i> that have composition associations with the corresponding <i>Representations</i> .

9.4.13 Property element transformation

The *createProeprty* ATL rule transforms *Property* elements of the *External Service Composition* PIM extension meta-model, to *Property* ones of the *External Service Composition* PSM extension meta-model. Therefore, the *Property* of the *External Service Composition* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *Property* of the *External Service Composition* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-34 *createProperty* ATL rule

Property property/relation	Property property/relation	Explanation
name	name	The <i>name</i> of the <i>Proeprty</i> of the <i>External Service Composition</i> PSM extension equals the name of

		the <i>Property of the External Service Composition</i> PIM extension.
type	type	The <i>type</i> of the <i>Property of the External Service Composition</i> PSM extension equals the <i>type</i> of the <i>Property of the External Service Composition</i> PIM extension.
isUnique	isUnique	The <i>isUnique</i> property of the <i>Property of the External Service Composition</i> PSM extension equals the <i>isUnique</i> property of the <i>Property of the External Service Composition</i> PIM extension.

9.4.14 Representation element transformation

The *createRepresentation* ATL rule transforms *Representation* elements of the *External Service Composition* PIM extension meta-model, to *Representation* ones of the *External Service Composition* PSM extension meta-model. Therefore, the *Representation* of the *External Service Composition* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *Representation* of the *External Service Composition* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-35 *createRepresentation* ATL rule

Representation property/relation	Representation property/relation	Explanation
name	name	The <i>name</i> of the <i>Representation</i> of the <i>External Service Composition</i> PSM extension equals the <i>name</i> of the <i>Representation</i> of the <i>External Service Composition</i> PIM extension.

9.4.15 NonPersistentOutput element transformation

The *createNonPersistentOutput* ATL rule transforms *NonPersistentOutput* elements of the *External Service Composition* PIM extension meta-model, to *NonPersistentOutput* ones of the *External Service Composition* PSM extension meta-model. Therefore, the *NonPersistentOutput* of the *External Service Composition* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *NonPersistentOutput* of the *External Service Composition* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-36 *createNonPersistentOutput* ATL rule

NonPersistentOutput property/relation	NonPersistentOutput property/relation	Explanation
Property	Property	This ATL rule transforms any composition associations of the <i>External Service Composition</i> PIM extension <i>NonPersistentOutputs</i> to <i>Properties</i> , to <i>External Service Composition</i> PSM

		extension <i>NonPersistentOutputs</i> that have composition associations with the corresponding <i>Properties</i> .
Representation	Representation	This ATL rule transforms any composition associations of the <i>External Service Composition</i> PIM extension <i>NonPersistentOutputs</i> to <i>Representations</i> , to <i>External Service Composition</i> PSM extension <i>NonPersistentOutputs</i> that have composition associations with the corresponding <i>Representations</i> .

9.4.16 AutoPersistentOutput element transformation

The *createAutoPersistentOutput* ATL rule transforms *AutoPersistentOutput* elements of the *External Service Composition* PIM extension meta-model, to *AutoPersistentOutput* ones of the *External Service Composition* PSM extension meta-model. Therefore, the *AutoPersistentOutput* of the *External Service Composition* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AutoPersistentOutput* of the *External Service Composition* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-37 *createAutoPersistentOutput* ATL rule

AutoPersistentOutput property/relation	AutoPersistentOutput property/relation	Explanation
Property	Property	This ATL rule transforms any composition associations of the <i>External Service Composition</i> PIM extension <i>AutoPersistentOutputs</i> to <i>Properties</i> , to <i>External Service Composition</i> PSM extension <i>AutoPersistentOutputs</i> that have composition associations with the corresponding <i>Properties</i> .
Representation	Representation	This ATL rule transforms any composition associations of the <i>External Service Composition</i> PIM extension <i>AutoPersistentOutputs</i> to <i>Representations</i> , to <i>External Service Composition</i> PSM extension <i>AutoPersistentOutputs</i> that have composition associations with the corresponding <i>Representations</i> .

9.4.17 ExistentsCRUDPersistentOutput element transformation

The *createExistentsJavaModelPersistentOutput* ATL rule transforms *ExistentsCRUDPersistentOutput* elements of the *External Service Composition* PIM extension meta-model, to *ExistentsJavaModelPersistentOutput* ones of the *External Service Composition* PSM extension meta-model. Therefore, the *ExistentsJavaModelPersistentOutput* of the *External Service Composition* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *ExistentsCRUDPersistentOutput* of the *External Service Composition*

PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-38 createExistentJavaModelPersistentOutput ATL rule

ExistenCRUDPersistentOutput property/relation	ExistenJavaModelPersistentOutput property/relation	Explanation
Property	Property	This ATL rule transforms any composition associations of the <i>External Service Composition</i> PIM extension <i>ExistenCRUDPersistentOutputs</i> to <i>Properties</i> , to <i>External Service Composition</i> PSM extension <i>ExistenJavaModelPersistentOutput</i> that have composition associations with the corresponding <i>Properties</i> .
Representation	Representation	This ATL rule transforms any composition associations of the <i>External Service Composition</i> PIM extension <i>ExistenCRUDPersistentOutputs</i> to <i>Representations</i> , to <i>External Service Composition</i> PSM extension <i>ExistenJavaModelPersistentOutput</i> that have composition associations with the corresponding <i>Representations</i> .
AnnResourceModel	AnnJavaResourceModel	This ATL rule transforms any associations of the <i>External Service Composition</i> PIM extension <i>ExistenCRUDPersistentOutputs</i> to <i>AnnResourceModels</i> , to <i>External Service Composition</i> PSM extension <i>ExistenJavaModelPersistentOutput</i> that have associations with the corresponding <i>AnnJavaResourceModels</i> .

9.5 ABAC Authorization Extension PIM to PSM ATL Rules

9.5.1 AnnotationModel element transformation

The *AuthorizationPIMToPSM* ATL rule transforms *AnnotationModel* elements of the *ABAC Authorization* PIM extension meta-model, to *AnnotationModel* ones of the *ABAC Authorization* PSM extension meta-model. Therefore, the *AnnotationModel* of the *ABAC Authorization* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AnnotationModel* of the *ABAC Authorization* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-39 AuthorizationPIMToPSM ATL rule

AnnotationModel property/relation	AnnotationModel property/relation	Explanation
name	name	The <i>name</i> of the <i>AnnotationModel</i> of the <i>ABAC Authorization PSM</i> extension equals the name of the <i>AnnotationModel</i> of the <i>ABAC Authorization PIM</i> extension.
AnnotatedElement	AnnotatedElement	Every <i>AnnotatedElement</i> of the <i>ABAC Authorization PIM</i> extension meta-model is transformed to its <i>ABAC Authorization PSM</i> extension meta-model counterpart by calling once the <i>createAnnJavaResourceModel</i> ATL rule, or the <i>createAnnJavaAlgoResourceModel</i> , or the <i>createAnnJavaResourceModelManager</i> one, depending on the case, for every <i>ABAC Authorization PIM</i> extension <i>AnnResourceModel</i> , <i>AnnAlgoResourceModel</i> or <i>AnnResourceModelManager</i> respectively. Moreover, the <i>createAnnJPAController</i> ATL rule introduces <i>AnnJPAController</i> elements, the <i>createAnnHTTPActivityHandler</i> introduces <i>AnnHTTPActivityHandlers</i> and the <i>createAnnPSMComponentProperty</i> ATL rule or the <i>createAnnResourceControllerHTTPActivity</i> for every <i>ABAC Authorization PIM</i> extension <i>AnnCRUDActivityHandler</i> or <i>AnnResourceControllerCRUDActivity</i> respectively.
Annotation	Annotation	Every <i>Annotation</i> of the <i>ABAC Authorization PIM</i> extension meta-model is transformed to its <i>ABAC Authorization PSM</i> extension meta-model counterpart by calling once the <i>createAuthorizationSubject</i> ATL rule for every <i>ABAC Authorization PIM</i> extension <i>AuthorizationSubject</i> and the <i>createAuthorizableResource</i> ATL rule for every <i>ABAC Authorization PIM</i> extension <i>AuthorizableResource</i> . Additionally, it transforms <i>AuthorizationPerformers</i> of the <i>ABAC Authorization PIM</i> extension meta-model annotations by calling the <i>createAuthorizationperformer</i> ATL rule.

9.5.2 AnnResourceModel element transformation

The *createAnnJavaResourceModel* ATL rule transforms *AnnResourceModel* elements of the *ABAC Authorization PIM* extension meta-model, to *AnnJavaResourceModel* ones of the *ABAC Authorization PSM* extension meta-model. Therefore, the *AnnJavaResourceModel* of the *ABAC Authorization PSM* extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AnnResourceModel* of the *ABAC Authorization PIM* extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-40 createAnnJavaResourceModel ATL rule

AnnResourceModel property/relation	AnnJavaResourceModel property/relation	Explanation
ResourceModel	JavaResourceModel	This ATL rule transforms any references of the <i>ABAC Authorization</i> PIM extension <i>AnnResourceModels</i> to Core PIM <i>ResourceModels</i> , to <i>ABAC Authorization</i> PSM extension <i>AnnJavaResourceModels</i> that reference the corresponding Core PSM <i>JavaResourceModels</i> .

9.5.3 AnnAlgoResourceModel element transformation

The *createAnnJavaAlgoResourceModel* ATL rule transforms *AnnAlgoResourceModel* elements of the *ABAC Authorization* PIM extension meta-model, to *AnnJavaAlgoResourceModel* ones of the *ABAC Authorization* PSM extension meta-model. Therefore, the *AnnJavaAlgoResourceModel* of the *ABAC Authorization* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AnnAlgoResourceModel* of the *ABAC Authorization* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-41 createAnnJavaAlgoResourceModel ATL rule

AnnAlgoResourceModel property/relation	AnnJavaAlgoResourceModel property/relation	Explanation
AlgoResourceModel	JavaAlgoResourceModel	This ATL rule transforms any references of the <i>ABAC Authorization</i> PIM extension <i>AnnResourceModels</i> to Core PIM <i>ResourceModels</i> , to <i>ABAC Authorization</i> PSM extension <i>AnnJavaResourceModels</i> that reference the corresponding Core PSM <i>JavaResourceModels</i> .

9.5.4 AnnResourceModelManager element transformation

The *createAnnJavaResourceModelManager* ATL rule transforms *AnnResourceModelManager* elements of the *ABAC Authorization* PIM extension meta-model, to *AnnJavaResourceModelManager* ones of the *ABAC Authorization* PSM extension meta-model. Therefore, the *AnnJavaResourceModelManager* of the *ABAC Authorization* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AnnResourceModelManager* of the *ABAC Authorization* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-42 createAnnJavaResourceModelManager ATL rule

AnnResourceModelManager property/relation	AnnJavaResourceModelManager property/relation	Explanation
ResourceModelManager	JavaResourceModelManager	This ATL rule transforms any references of the <i>ABAC Authorization</i> PIM extension <i>AnnResourceModelManagers</i> to Core PIM <i>ResourceModelManagers</i> , to <i>ABAC Authorization</i> PSM extension <i>AnnJavaResourceModelManagers</i> that reference the corresponding Core PSM <i>JavaResourceModelManagers</i> .

9.5.5 AnnCRUDActivityHandler element transformation

The *createAnnHTTPActivityHandler* ATL rule transforms *AnnCRUDActivityHandler* elements of the *ABAC Authorization* PIM extension meta-model, to *AnnHTTPActivityHandler* ones of the *ABAC Authorization* PSM extension meta-model. Therefore, the *AnnHTTPActivityHandler* of the *ABAC Authorization* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AnnCRUDActivityHandler* of the *ABAC Authorization* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-43 createAnnHTTPActivityHandler ATL rule

AnnCRUDActivityHandler property/relation	AnnHTTPActivityHandler property/relation	Explanation
CRUDActivityHandler	HTTPActivityHandler	This ATL rule transforms any references of the <i>ABAC Authorization</i> PIM extension <i>AnnCRUDActivityHandlers</i> to Core PIM <i>CRUDActivityHandlers</i> , to <i>ABAC Authorization</i> PSM extension <i>AnnHTTPActivityHandlers</i> that reference the corresponding Core PSM <i>HTTPActivityHandlers</i> .

9.5.6 AnnDatabaseController element transformation

The *createAnnJPAController* ATL rule transforms *AnnDatabaseController* elements of the *ABAC Authorization* PIM extension meta-model, to *AnnJPAController* ones of the *ABAC Authorization* PSM extension meta-model. Therefore, the *AnnJPAController* of the *ABAC Authorization* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AnnDatabaseController* of the *ABAC Authorization* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-44 createAnnJPAController ATL rule

AnnDatabaseController property/relation	AnnJPAController property/relation	Explanation

DatabaseController	JPAController	This ATL rule transforms any references of the <i>ABAC Authorization</i> PIM extension <i>AnnDatabaseControllers</i> to Core PIM <i>DatabaseControllers</i> , to <i>ABAC Authorization</i> PSM extension <i>AnnJPAControllers</i> that reference the corresponding Core PSM <i>JPAControllers</i> .
--------------------	---------------	---

9.5.7 AnnPIMComponentProperty element transformation

The *createAnnPSMComponentProperty* ATL rule transforms *AnnPIMComponentProperty* elements of the *ABAC Authorization* PIM extension meta-model, to *AnnPSMComponentProperty* ones of the *ABAC Authorization* PSM extension meta-model. Therefore, the *AnnPSMComponentProperty* of the *ABAC Authorization* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AnnPIMComponentProperty* of the *ABAC Authorization* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-45 *createAnnPSMComponentProperty* ATL rule

AnnPIMComponentProperty property/relation	AnnPSMComponentProperty property/relation	Explanation
PIMComponentProperty	PSMComponentProperty	This ATL rule transforms any references of the <i>ABAC Authorization</i> PIM extension <i>AnnPIMComponentProperties</i> to Core PIM <i>PIMComponentProperties</i> , to <i>ABAC Authorization</i> PSM extension <i>AnnPSMComponentProperties</i> that reference the corresponding Core PSM <i>PSMComponentProperties</i> .

9.5.8 AnnResourceControllerCRUDActivity element transformation

The *createAnnHTTPActivity* ATL rule transforms *AnnResourceControllerCRUDActivity* elements of the *ABAC Authorization* PIM extension meta-model, to *AnnHTTPActivity* ones of the *ABAC Authorization* PSM extension meta-model. Therefore, the *AnnHTTPActivity* of the *ABAC Authorization* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AnnResourceControllerCRUDActivity* of the *ABAC Authorization* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-46 *createAnnHTTPActivity* ATL rule

AnnResourceControllerCRUDActivity property/relation	AnnHTTPActivity property/relation	Explanation
ResourceControllerCRUDActivity	HTTPActivity	This ATL rule transforms any references of the <i>ABAC Authorization</i> PIM extension <i>AnnResourceControllerCRUDActivities</i> to Core PIM <i>ResourceControllerCRUDActivities</i> , to <i>ABAC Authorization</i> PSM extension <i>AnnHTTPActivities</i> that reference the corresponding Core PSM

		HTTPActivities.
--	--	-----------------

9.5.9 AuthorizationSubject element transformation

The *createAuthorizationSubject* ATL rule transforms *AuthorizationSubject* elements of the *ABAC Authorization* PIM extension meta-model, to *AuthorizationSubject* ones of the *ABAC Authorization* PSM extension meta-model. Therefore, the *AuthorizationSubject* of the *ABAC Authorization* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AuthorizationSubject* of the *ABAC Authorization* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-47 *createAuthorizationSubject* ATL rule

AuthorizationSubject property/relation	AuthorizationSubject property/relation	Explanation
AnnResourceModel	AnnJavaResourceModel	This ATL rule transforms any associations of the <i>ABAC Authorization</i> PIM extension <i>AuthorizationSubjects</i> to <i>AnnResourceModels</i> , to <i>ABAC Authorization</i> PSM extension <i>AuthorizationSubjects</i> that have associations with the corresponding <i>AnnJavaResourceModels</i> .
SubjectAttribute	SubjectAttribute	This ATL rule transforms any composition associations of the <i>ABAC Authorization</i> PIM extension <i>AuthorizationSubjects</i> to <i>SubjectAttributes</i> , to <i>ABAC Authorization</i> PSM extension <i>AuthorizationSubjects</i> that have composition association with the corresponding <i>ABAC Authorization</i> PSM extension <i>SubjectAttributes</i> .

9.5.10 SubjectAttribute element transformation

The *createSubjectAttribute* ATL rule transforms *SubjectAttribute* elements of the *ABAC Authorization* PIM extension meta-model, to *SubjectAttribute* ones of the *ABAC Authorization* PSM extension meta-model. Therefore, the *SubjectAttribute* of the *ABAC Authorization* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *SubjectAttribute* of the *ABAC Authorization* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-48 *createSubjectAttribute* ATL rule

SubjectAttribute property/relation	SubjectAttribute property/relation	Explanation
name	name	The <i>name</i> of the <i>SubjectAttribute</i> of the <i>ABAC Authorization</i> PSM extension equals the name of the <i>SubjectAttribute</i> of the <i>ABAC Authorization</i> PIM extension.

type	type	The <i>type</i> of the <i>SubjectAttribute</i> of the <i>ABAC Authorization</i> PSM extension equals the <i>type</i> of the <i>SubjectAttribute</i> of the <i>ABAC Authorization</i> PIM extension.
isUnique	isUnique	The <i>isUnique</i> property of the <i>SubjectAttribute</i> of the <i>ABAC Authorization</i> PSM extension equals the <i>isUnique</i> property of the <i>SubjectAttribute</i> of the <i>ABAC Authorization</i> PIM extension.

9.5.11 AuthorizableResource element transformation

The *createAuthorizableResource* ATL rule transforms *AuthorizableResource* elements of the *ABAC Authorization* PIM extension meta-model, to *AuthorizableResource* ones of the *ABAC Authorization* PSM extension meta-model. Therefore, the *AuthorizableResource* of the *ABAC Authorization* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AuthorizableResource* of the *ABAC Authorization* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-49 *createAuthorizableResource* ATL rule

AuthorizableResource property/relation	AuthorizableResource property/relation	Explanation
AnnResourceModel	AnnJavaResourceModel	This ATL rule transforms any associations of the <i>ABAC Authorization</i> PIM extension <i>AuthorizableResources</i> to <i>AnnResourceModels</i> , to <i>ABAC Authorization</i> PSM extension <i>AuthorizableResources</i> that have associations with the corresponding <i>AnnJavaResourceModels</i> .
AnnAlgoResourceModel	AnnJavaAlgoResourceModel	This ATL rule transforms any associations of the <i>ABAC Authorization</i> PIM extension <i>AuthorizableResources</i> to <i>AnnAlgoResourceModels</i> , to <i>ABAC Authorization</i> PSM extension <i>AuthorizableResources</i> that have associations with the corresponding <i>AnnJavaAlgoResourceModels</i> .
AnnResourceModelManager	AnnJavaResourceModelManager	This ATL rule transforms any associations of the <i>ABAC Authorization</i> PIM extension <i>AuthorizableResources</i> to <i>AnnResourceModelManagers</i> , to <i>ABAC Authorization</i> PSM extension <i>AuthorizableResources</i> that have associations with the corresponding <i>AnnJavaResourceModelManagers</i> .
AuthorizationPerformer	AuthorizationPerformer	This ATL rule transforms any associations of the <i>ABAC Authorization</i> PIM extension <i>AuthorizableResources</i> to <i>AuthorizationPerformers</i> , to <i>ABAC Authorization</i> PSM extension <i>AuthorizableResources</i> that have composition association with the corresponding <i>ABAC Authorization</i> PSM extension

		<i>AuthorizationPerformers.</i>
ResourceAccessPolicySet	ResourceAccessPolicySet	This ATL rule transforms any composition associations of the <i>ABAC Authorization</i> PIM extension <i>AuthorizableResources</i> to <i>ResourceAccessPolicySets</i> , to <i>ABAC Authorization</i> PSM extension <i>AuthorizableResources</i> that have composition association with the corresponding <i>ABAC Authorization</i> PSM extension <i>ResourceAccessPolicySets</i> .

9.5.12 AuthorizationPerformer element transformation

The *createAuthorizationPerformer* ATL rule transforms *AuthorizationPerformer* elements of the *ABAC Authorization* PIM extension meta-model, to *AuthorizationPerformer* ones of the *ABAC Authorization* PSM extension meta-model. Therefore, the *AuthorizationPerformer* of the *ABAC Authorization* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AuthorizationPerformer* of the *ABAC Authorization* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-50 createAuthorizationPerformer ATL rule

AuthorizationPerformer property/relation	AuthorizationPerformer property/relation	Explanation
AnnCRUDActivityHandler	AnnHTTPActivityHandler	This ATL rule transforms any associations of the <i>ABAC Authorization</i> PIM extension <i>AuthorizationPerformers</i> to <i>AnnCRUDActivityHandlers</i> , to <i>ABAC Authorization</i> PSM extension <i>AuthorizationPerformers</i> that have associations with the corresponding <i>AnnHTTPActivityHandlers</i> .
AuthorizationPolicyEvaluator	AuthorizationPolicyEvaluator	This ATL rule transforms any associations of the <i>ABAC Authorization</i> PIM extension <i>AuthorizationPerformers</i> to <i>AuthorizationPolicyEvaluators</i> , to <i>ABAC Authorization</i> PSM extension <i>AuthorizationPerformers</i> that have association with the corresponding <i>ABAC Authorization</i> PSM extension <i>AuthorizationPolicyEvaluator</i> .

9.5.13 AuthorizationPolicyEvaluator element transformation

The *createAuthorizationPolicyEvaluator* ATL rule transforms *AuthorizationPolicyEvaluator* elements of the *ABAC Authorization* PIM extension meta-model, to *AuthorizationPolicyEvaluator* ones of the *ABAC Authorization* PSM extension meta-model. Therefore, the *AuthorizationPolicyEvaluator* of the *ABAC Authorization* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AuthorizationPolicyEvaluator* of the *ABAC Authorization* PIM extension meta-model element. Since *ABAC Authorization* PSM extension

AuthorizationPolicyEvaluators do not have any associations/properties, this ATL rule does not perform any association/property mappings.

9.5.14 ResourceAccessPolicySet element transformation

The *createResourceAccessPolicySet* ATL rule transforms *ResourceAccessPolicySet* elements of the *ABAC Authorization* PIM extension meta-model, to *ResourceAccessPolicySet* ones of the *ABAC Authorization* PSM extension meta-model. Therefore, the *ResourceAccessPolicySet* of the *ABAC Authorization* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *ResourceAccessPolicySet* of the *ABAC Authorization* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-51 *createResourceAccessPolicySet* ATL rule

ResourceAccessPolicySet property/relation	ResourceAccessPolicySet property/relation	Explanation
RuleCombiningAlgorithm	RuleCombiningAlgorithm	This ATL rule transforms any associations of the <i>ABAC Authorization</i> PIM extension <i>ResourceAccessPolicySets</i> to <i>RuleCombiningAlgorithms</i> , to <i>ABAC Authorization</i> PSM extension <i>ResourceAccessPolicySets</i> that have association with the corresponding <i>ABAC Authorization</i> PSM extension <i>RuleCombiningAlgorithms</i> .
ResourceAccessPolicy	ResourceAccessPolicy	This ATL rule transforms any associations of the <i>ABAC Authorization</i> PIM extension <i>ResourceAccessPolicySets</i> to <i>ResourceAccessPolicies</i> , to <i>ABAC Authorization</i> PSM extension <i>ResourceAccessPolicySets</i> that have association with the corresponding <i>ABAC Authorization</i> PSM extension <i>ResourceAccessPolicies</i> .
AuthorizationDataTable	JPAAnnotation	This ATL rule transforms any associations of the <i>ABAC Authorization</i> PIM extension <i>ResourceAccessPolicySets</i> to <i>AuthorizationDataTables</i> , to <i>ABAC Authorization</i> PSM extension <i>ResourceAccessPolicySets</i> that have association with the corresponding <i>ABAC Authorization</i> PSM extension <i>JPAAnnotations</i> .

9.5.15 ResourceAccessPolicy element transformation

The *createResourceAccessPolicy* ATL rule transforms *ResourceAccessPolicy* elements of the *ABAC Authorization* PIM extension meta-model, to *ResourceAccessPolicy* ones of the *ABAC Authorization* PSM extension meta-model. Therefore, the *ResourceAccessPolicy* of the *ABAC Authorization* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *ResourceAccessPolicy* of the *ABAC Authorization* PIM extension meta-

model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-52 createResourceAccessPolicy ATL rule

ResourceAccessPolicy property/relation	ResourceAccessPolicy property/relation	Explanation
RuleCombiningAlgorithm	RuleCombiningAlgorithm	This ATL rule transforms any associations of the <i>ABAC Authorization</i> PIM extension <i>ResourceAccessPolicies</i> to <i>RuleCombiningAlgorithms</i> , to <i>ABAC Authorization</i> PSM extension <i>ResourceAccessPolicies</i> that have association with the corresponding <i>ABAC Authorization</i> PSM extension <i>RuleCombiningAlgorithms</i> .
ResourceAccessRule	ResourceAccessRule	This ATL rule transforms any associations of the <i>ABAC Authorization</i> PIM extension <i>ResourceAccessPolicies</i> to <i>ResourceAccessRules</i> , to <i>ABAC Authorization</i> PSM extension <i>ResourceAccessPolicies</i> that have association with the corresponding <i>ABAC Authorization</i> PSM extension <i>ResourceAccessRules</i> .
AuthorizationDataTable	JPAAnnotation	This ATL rule transforms any associations of the <i>ABAC Authorization</i> PIM extension <i>ResourceAccessPolicies</i> to <i>AuthorizationDataTables</i> , to <i>ABAC Authorization</i> PSM extension <i>ResourceAccessPolicies</i> that have association with the corresponding <i>ABAC Authorization</i> PSM extension <i>JPAAnnotations</i> .

9.5.16 DenyOverridesAlgorithm element transformation

The *createDenyOverridesAlgorithm* ATL rule transforms *DenyOverridesAlgorithm* elements of the *ABAC Authorization* PIM extension meta-model, to *DenyOverridesAlgorithm* ones of the *ABAC Authorization* PSM extension meta-model. Therefore, the *DenyOverridesAlgorithm* of the *ABAC Authorization* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *DenyOverridesAlgorithm* of the *ABAC Authorization* PIM extension meta-model element. Since the *DenyOverridesAlgorithm* elements do not have any associations/properties, this ATL rule does not perform any association/property mappings.

9.5.17 PermitOverridesAlgorithm element transformation

The *createPermitOverridesAlgorithm* ATL rule transforms *PermitOverridesAlgorithm* elements of the *ABAC Authorization* PIM extension meta-model, to *PermitOverridesAlgorithm* ones of the *ABAC Authorization* PSM extension meta-model. Therefore, the *PermitOverridesAlgorithm* of the *ABAC Authorization* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *PermitOverridesAlgorithm* of the *ABAC Authorization* PIM extension meta-model element. Since the *PermitOverridesAlgorithm* elements do

not have any associations/properties, this ATL rule does not perform any association/property mappings.

9.5.18 ResourceAccessDenyRule element transformation

The *createResourceAccessDenyRule* ATL rule transforms *ResourceAccessDenyRule* elements of the *ABAC Authorization* PIM extension meta-model, to *ResourceAccessDenyRule* ones of the *ABAC Authorization* PSM extension meta-model. Therefore, the *ResourceAccessDenyRule* of the *ABAC Authorization* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *ResourceAccessDenyRule* of the *ABAC Authorization* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-53 *createResourceAccessDenyRule* ATL rule

ResourceAccessDenyRule property/relation	ResourceAccessDenyRule property/relation	Explanation
AllowedAction	AllowedAction	This ATL rule transforms any associations of the <i>ABAC Authorization</i> PIM extension <i>ResourceAccessDenyRule</i> to <i>AllowedActions</i> , to <i>ABAC Authorization</i> PSM extension <i>ResourceAccessDenyRules</i> that have association with the corresponding <i>ABAC Authorization</i> PSM extension <i>AllowedActions</i> .
MatchedResourceAttribute	MatchedResourceAttribute	This ATL rule transforms any associations of the <i>ABAC Authorization</i> PIM extension <i>ResourceAccessDenyRules</i> to <i>MatchedResourceAttributes</i> , to <i>ABAC Authorization</i> PSM extension <i>ResourceAccessDenyRules</i> that have association with the corresponding <i>ABAC Authorization</i> PSM extension <i>MatchedResourceAttributes</i> .
MatchedSubjectAttribute	MatchedSubjectAttribute	This ATL rule transforms any associations of the <i>ABAC Authorization</i> PIM extension <i>ResourceAccessDenyRules</i> to <i>MatchedSubjectAttributes</i> , to <i>ABAC Authorization</i> PSM extension <i>ResourceAccessDenyRules</i> that have association with the corresponding <i>ABAC Authorization</i> PSM extension <i>MatchedSubjectAttributes</i> .
MatchedContextAttribute	MatchedContextAttribute	This ATL rule transforms any associations of the <i>ABAC Authorization</i> PIM extension <i>ResourceAccessDenyRules</i> to <i>MatchedContextAttributes</i> , to <i>ABAC Authorization</i> PSM extension <i>ResourceAccessDenyRules</i> that have association with the corresponding <i>ABAC Authorization</i> PSM extension <i>MatchedContextAttributes</i> .
AuthorizationDataTable	JPAAnnotation	This ATL rule transforms any associations of the <i>ABAC Authorization</i> PIM extension <i>ResourceAccessDenyRules</i> to <i>AuthorizationDataTables</i> , to <i>ABAC Authorization</i>

		PSM extension <i>ResourceAccessDenyRules</i> that have association with the corresponding <i>ABAC Authorization</i> PSM extension <i>JPAAnnotations</i> .
--	--	---

9.5.19 ResourceAccessPermitRule element transformation

The *createResourceAccessPermitRule* ATL rule transforms *ResourceAccessPermitRule* elements of the *ABAC Authorization* PIM extension meta-model, to *ResourceAccessPermitRule* ones of the *ABAC Authorization* PSM extension meta-model. Therefore, the *ResourceAccessPermitRule* of the *ABAC Authorization* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *ResourceAccessPermitRule* of the *ABAC Authorization* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-54 *createResourceAccessPermitRule* ATL rule

ResourceAccessPermitRule property/relation	ResourceAccessPermitRule property/relation	Explanation
AllowedAction	AllowedAction	This ATL rule transforms any associations of the <i>ABAC Authorization</i> PIM extension <i>ResourceAccessPermitRules</i> to <i>AllowedActions</i> , to <i>ABAC Authorization</i> PSM extension <i>ResourceAccessPermitRules</i> that have association with the corresponding <i>ABAC Authorization</i> PSM extension <i>AllowedActions</i> .
MatchedResourceAttribute	MatchedResourceAttribute	This ATL rule transforms any associations of the <i>ABAC Authorization</i> PIM extension <i>ResourceAccessPermitRules</i> to <i>MatchedResourceAttributes</i> , to <i>ABAC Authorization</i> PSM extension <i>ResourceAccessPermitRules</i> that have association with the corresponding <i>ABAC Authorization</i> PSM extension <i>MatchedResourceAttributes</i> .
MatchedSubjectAttribute	MatchedSubjectAttribute	This ATL rule transforms any associations of the <i>ABAC Authorization</i> PIM extension <i>ResourceAccessPermitRules</i> to <i>MatchedSubjectAttributes</i> , to <i>ABAC Authorization</i> PSM extension <i>ResourceAccessPermitRules</i> that have association with the corresponding <i>ABAC Authorization</i> PSM extension <i>MatchedSubjectAttributes</i> .
MatchedContextAttribute	MatchedContextAttribute	This ATL rule transforms any associations of the <i>ABAC Authorization</i> PIM extension <i>ResourceAccessPermitRules</i> to <i>MatchedContextAttributes</i> , to <i>ABAC Authorization</i> PSM extension <i>ResourceAccessPermitRules</i> that have association with the corresponding <i>ABAC Authorization</i> PSM extension <i>MatchedContextAttributes</i> .
AuthorizationDataTable	JPAAnnotation	This ATL rule transforms any associations of the <i>ABAC Authorization</i> PIM extension

		<i>ResourceAccessPermitRules</i> to <i>AuthorizationDataTables</i> , to ABAC Authorization PSM extension <i>ResourceAccessPermitRules</i> that have association with the corresponding ABAC Authorization PSM extension <i>JPAAnnotations</i> .
--	--	---

9.5.20 AllowedAction element transformation

The *createAllowedAction* ATL rule transforms *AllowedAction* elements of the ABAC Authorization PIM extension meta-model, to *AllowedAction* ones of the ABAC Authorization PSM extension meta-model. Therefore, the *AllowedAction* of the ABAC Authorization PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AllowedAction* of the ABAC Authorization PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-55 createAllowedAction ATL rule

AllowedAction property/relation	AllowedAction property/relation	Explanation
AnnResourceControllerCRUDActivity	AnnHTTPActivity	This ATL rule transforms any associations of the ABAC Authorization PIM extension <i>AllowedActions</i> to <i>AnnResourceControllerCRUDActivities</i> , to ABAC Authorization PSM extension <i>AllowedActions</i> that have association with the corresponding ABAC Authorization PSM extension <i>AnnHTTPActivities</i> .
AuthorizationDataTable	JPAAnnotation	This ATL rule transforms any associations of the ABAC Authorization PIM extension <i>AllowedActions</i> to <i>AuthorizationDataTables</i> , to ABAC Authorization PSM extension <i>AllowedActions</i> that have association with the corresponding ABAC Authorization PSM extension <i>JPAAnnotations</i> .

9.5.21 MatchedResourceAttribute element transformation

The *createMatchedResourceAttribute* ATL rule transforms *MatchedResourceAttribute* elements of the ABAC Authorization PIM extension meta-model, to *MatchedResourceAttribute* ones of the ABAC Authorization PSM extension meta-model. Therefore, the *MatchedResourceAttribute* of the ABAC Authorization PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *MatchedResourceAttribute* of the ABAC Authorization PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-56 createMatchedResourceAttribute ATL rule

MatchedResourceAttribute property/relation	MatchedResourceAttribute property/relation	Explanation
--	--	-------------

matchedValue	matchedValue	The <i>matchedValue</i> of the <i>MatchedResourceAttributes</i> of the ABAC Authorization PSM extension equals the <i>matchedValue</i> of the <i>MatchedResourceAttributes</i> of the ABAC Authorization PIM extension.
comparisonOperator	comparisonOperator	The <i>comparisonOperator</i> of the <i>MatchedResourceAttributes</i> of the ABAC Authorization PSM extension equals the <i>comparisonOperator</i> of the <i>MatchedResourceAttributes</i> of the ABAC Authorization PIM extension.
AuthorizationDataTable	JPAAnnotation	This ATL rule transforms any associations of the ABAC Authorization PIM extension <i>MatchedResourceAttributes</i> to <i>AuthorizationDataTables</i> , to ABAC Authorization PSM extension <i>MatchedResourceAttributes</i> that have association with the corresponding ABAC Authorization PSM extension <i>JPAAnnotations</i> .
AnnPIMComponentProperty	AnnPSMComponentProperty	This ATL rule transforms any associations of the ABAC Authorization PIM extension <i>MatchedResourceAttributes</i> to <i>AnnPIMComponentProperties</i> , to ABAC Authorization PSM extension <i>MatchedResourceAttributes</i> that have association with the corresponding ABAC Authorization PSM extension <i>AnnPSMComponentProperties</i> .

9.5.22 MatchedContextAttribute element transformation

The *createMatchedContextAttribute* ATL rule transforms *MatchedContextAttribute* elements of the ABAC Authorization PIM extension meta-model, to *MatchedContextAttribute* ones of the ABAC Authorization PSM extension meta-model. Therefore, the *MatchedContextAttribute* of the ABAC Authorization PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *MatchedContextAttribute* of the ABAC Authorization PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-57 *createMatchedContextAttribute* ATL rule

MatchedContextAttribute property/relation	MatchedContextAttribute property/relation	Explanation
matchedValue	matchedValue	The <i>matchedValue</i> of the <i>MatchedContextAttributes</i> of the ABAC Authorization PSM extension equals the <i>matchedValue</i> of the <i>MatchedContextAttributes</i> of the ABAC Authorization PIM extension.
comparisonOperator	comparisonOperator	The <i>comparisonOperator</i> of the <i>MatchedContextAttributes</i> of the ABAC Authorization PSM extension equals the <i>comparisonOperator</i> of the <i>MatchedContextAttributes</i> of the ABAC Authorization PIM extension.

		<i>Authorization PIM extension.</i>
AuthorizationDataTable	JPAAnnotation	This ATL rule transforms any associations of the <i>ABAC Authorization PIM extension MatchedContextAttributes</i> to <i>AuthorizationDataTables</i> , to <i>ABAC Authorization PSM extension MatchedContextAttributes</i> that have association with the corresponding <i>ABAC Authorization PSM extension JPAAnnotations</i> .
AnnPIMComponentProperty	AnnPSMComponentProperty	This ATL rule transforms any associations of the <i>ABAC Authorization PIM extension MatchedContextAttributes</i> to <i>AnnPIMComponentProperties</i> , to <i>ABAC Authorization PSM extension MatchedContextAttributes</i> that have association with the corresponding <i>ABAC Authorization PSM extension AnnPSMComponentProperties</i> .

9.5.23 MatchedSubjectAttribute element transformation

The *createMatchedSubjectAttribute* ATL rule transforms *MatchedSubjectAttribute* elements of the *ABAC Authorization PIM extension* meta-model, to *MatchedSubjectAttribute* ones of the *ABAC Authorization PSM extension* meta-model. Therefore, the *MatchedSubjectAttribute* of the *ABAC Authorization PSM extension* meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *MatchedSubjectAttribute* of the *ABAC Authorization PIM extension* meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-58 *createMatchedSubjectAttribute* ATL rule

MatchedSubjectAttribute property/relation	MatchedSubjectAttribute property/relation	Explanation
matchedValue	matchedValue	The <i>matchedValue</i> of the <i>MatchedSubjectAttributes</i> of the <i>ABAC Authorization PSM extension</i> equals the <i>matchedValue</i> of the <i>MatchedSubjectAttributes</i> of the <i>ABAC Authorization PIM extension</i> .
comparisonOperator	comparisonOperator	The <i>comparisonOperator</i> of the <i>MatchedSubjectAttributes</i> of the <i>ABAC Authorization PSM extension</i> equals the <i>comparisonOperator</i> of the <i>MatchedSubjectAttributes</i> of the <i>ABAC Authorization PIM extension</i> .
AuthorizationDataTable	JPAAnnotation	This ATL rule transforms any associations of the <i>ABAC Authorization PIM extension MatchedSubjectAttributes</i> to <i>AuthorizationDataTables</i> , to <i>ABAC Authorization PSM extension MatchedSubjectAttributes</i> that have association with the corresponding <i>ABAC Authorization PSM extension JPAAnnotations</i> .
AnnPIMComponentProperty	AnnPSMComponentProperty	This ATL rule transforms any associations of the <i>ABAC Authorization PIM extension MatchedSubjectAttributes</i> to <i>AnnPIMComponentProperties</i> , to <i>ABAC Authorization PSM extension MatchedSubjectAttributes</i> that have association with the corresponding <i>ABAC Authorization PSM extension AnnPSMComponentProperties</i> .

		<p><i>ABAC Authorization</i> PIM extension <i>MatchedSubjectAttributes</i> to <i>AnnPIMComponentProperties</i>, to <i>ABAC Authorization</i> PSM extension <i>MatchedSubjectAttributes</i> that have association with the corresponding <i>ABAC Authorization</i> PSM extension <i>AnnPSMComponentProperties</i>.</p>
--	--	--

9.5.24 AuthorizationDataHandler element transformation

The *createAuthorizationDataHandler* ATL rule transforms *AuthorizationDataHandler* elements of the *ABAC Authorization* PIM extension meta-model, to *AuthorizationDataHandler* ones of the *ABAC Authorization* PSM extension meta-model. Therefore, the *AuthorizationDataHandler* of the *ABAC Authorization* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AuthorizationDataHandler* of the *ABAC Authorization* PIM extension meta-model element. The mapping of properties and relations between the two meta-model elements is defined in the table below:

Table 9-59 *createAuthorizationDataHandler* ATL rule

AuthorizationDataHandler property/relation	AuthorizationDataHandler property/relation	Explanation
AnnDatabaseController	AnnJPAController	This ATL rule transforms any associations of the <i>ABAC Authorization</i> PIM extension <i>AuthorizationDataHandlers</i> to <i>AnnDatabaseControllers</i> , to <i>ABAC Authorization</i> PSM extension <i>AuthorizationDataHandlers</i> that have association with the corresponding <i>ABAC Authorization</i> PSM extension <i>AnnJPAControllers</i> .

9.5.25 AuthorizationTable element transformation

The *createJPATableAnnotation* ATL rule transforms *AuthorizationTable* elements of the *ABAC Authorization* PIM extension meta-model, to *JPAAnnotation* ones of the *ABAC Authorization* PSM extension meta-model. Therefore, the *JPAAnnotation* of the *ABAC Authorization* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AuthorizationTable* of the *ABAC Authorization* PIM extension meta-model element. Since *ABAC Authorization* PSM extension *JPAAnnotation* elements do not have any associations/properties, this ATL rule does not perform any association/property mappings.

9.5.26 AuthorizationDataTableColumn element transformation

The *createJPAColumnAnnotation* ATL rule transforms *AuthorizationDataTableColumn* elements of the *ABAC Authorization* PIM extension meta-model, to *JPAAnnotation* ones of the *ABAC Authorization* PSM extension meta-model. Therefore, the *JPAAnnotation* of the *ABAC Authorization* PSM extension meta-model is a realization with concrete technologies of the abstract envisioned system's design part modelled with the *AuthorizationDataTableColumn* of the *ABAC Authorization* PIM extension meta-model element. Since the *ABAC Authorization* PSM extension *JPAAnnotation*

elements do not have any associations/properties, this ATL rule does not perform any association/property mappings.

10 PSM to Code Transformation Definition

10.1 PSM to Code Transformation

The final step of a Model Driven Engineering process is a Model-To-Text transformation in order to produce code. Section 3.2.5 has already introduced the *Acceleo* as the M2T transformation definition language that S-CASE MDE uses in order to produce the envisioned system's code. The acceleo language allows the developer to write code templates in a structured and natural way, which in turn are filled in by meta-data embedded in input meta-models.

In S-CASE case, the Acceleo templates take as input depending on case the following produced models by the ATL transformations:

Always:

- Core PSM model of the envisioned system

Optionally:

- Authentication PSM extension model of the envisioned system, if the S-CASE developer wishes to embed authentication functionality to his system.
- Database Searching PSM extension model of the envisioned system, if the S-CASE developer wishes to embed searching functionality to his system.
- External Service Composition PSM extension model of the envisioned system, if the S-CASE developer wishes to embed to his system the needed functionality so as it will be able to interact with compositions of external services.
- ABAC Authorization PSM extension model of the envisioned system, if the S-CASE developer wishes to include an ABAC authorization mechanism to his system.

Once the M2T component of the S-CASE MDE engine parses the aforementioned input envisioned system models, it produces the envision system code, accompanied with a set of project configuration files and a maven pom.xml file to compile and package as a *WAR* the envisioned system. The structure of the output project, complies with the maven *Standard Directory Layout* [20] as Figure 10-1 demonstrates in the Eclipse project explorer environment. Every Acceleo template produces one type of the demonstrated files/artefacts. The next section documents these templates, their input and their output artefacts.

10.2 PSM to Code Acceleo Templates

This section documents all the Acceleo templates, which are used by the S-CASE MDE engine in order to perform the final code generation step. The entry point of this generation step is the *generate.mtl* main Acceleo template the details of which are concluded in the following table. This main template triggers any needed calls to all the rest acceleo templates that together form the M2T transformation bundle. The rest of this section documents this template chain.

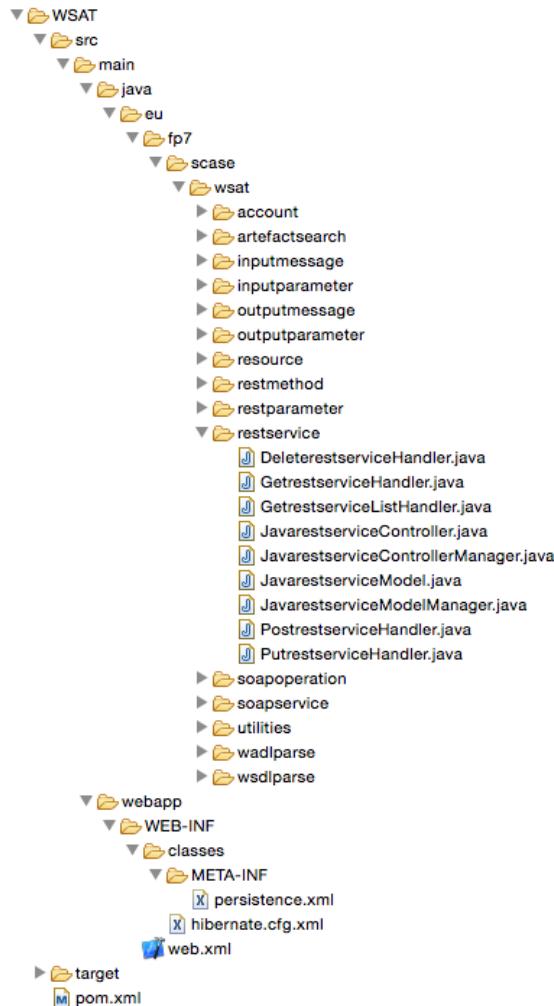


Figure 10-1 Maven Standard Layout Directory of WSAT system

Table 10-1 generate.mtl input/output

Acceleo Template Name:	generate.mtl
Uses PSM elements:	The whole <i>Annotation Stack</i> , namely the Core PSM and any available PSM extensions.
Produces code files:	none
Calls other Aceleo Templates:	hibernateConfigurationFile.mtl javaJAXRSPublisher.mtl javaAlgoResourceController.mtl javaAlgoResourceModel.mtl javaHibernateController.mtl javaHibernateUtil.mtl javaHTTPActivityHandler.mtl javaHypermediaLink.mtl javaResourceController.mtl

	javaResourceControllerManager.mtl javaResourceModel.mtl javaResourceModelManager.mtl mavenConfigurationFile.mtl webXMLConfigurationFile.mtl javaPersistentUtil.mtl (if there exists Database Searching PSM model) javaLuceneStringSetBridge.mtl (if there exists Database Searching PSM model) persistence.mtl (if there exists Database Searching PSM model) javaOutputDataModel.mtl (if there exists External Service Composition PSM model) javaAuthorizationPolicyEvaluator.mtl (if there exists ABAC Authorization PSM model) javaResourceAccessPolicySet.mtl (if there exists ABAC Authorization PSM model) javaResourceAccessPolicy.mtl (if there exists ABAC Authorization PSM model) javaResourceAccessRule.mtl (if there exists ABAC Authorization PSM model) javaAllowedAction.mtl (if there exists ABAC Authorization PSM model) javaMatchedResourceAttribute.mtl (if there exists ABAC Authorization PSM model) javaMatchedSubjectAttribute.mtl (if there exists ABAC Authorization PSM model) javaMatchedContextAttribute.mtl (if there exists ABAC Authorization PSM model)
--	--

Table 10-2 hibernateConfigurationFile.mtl input/output

Acceleo Template Name:	hibernateConfigurationFile.mtl
Uses PSM elements:	RESTfulServicePSM
Produces code files:	hibernate.cfg.xml
Calls other Acceleo Templates:	-

Table 10-3 javaJAXRSPublisher.mtl input/output

Acceleo Template Name:	javaJAXRSPublisher.mtl
Uses PSM elements:	RESTfulServicePSM, javaResourceControllers, javaResourceControllerManagers, javaAlgoResourceControllers
Produces code files:	JAXRSPublisher.java
Calls other Acceleo Templates:	-

Table 10-4 javaAlgoResourceController.mtl input/output

Acceleo Template Name:	javaAlgoResourceController.mtl
Uses PSM elements:	RESTfulServicePSM, javaAlgoResourceController, SearchableProperties, SearchableResources, AuthenticationMode, QueryParams, JavaInputDataModel, JavaOutputDataModel, HTTPActivity
Produces code files:	javaAlgoResourceController's name + ".java"
Calls other Acceleo Templates:	-

Table 10-5 javaAlgoResourceModel.mtl input/output

Acceleo Template Name:	javaAlgoResourceModel.mtl
Uses PSM elements:	RESTfulServicePSM, javaAlgoResourceModel, PSMComponentProperties, JPAAnnotations, JAXBAnnotations, JavaInputDataModel, JavaOutputDataModel
Produces code files:	javaAlgoResourceModel's name + ".java"
Calls other Acceleo Templates:	-

Table 10-6 javaHibernateController.mtl input/output

Acceleo Template Name:	javaHibernateController.mtl
Uses PSM elements:	RESTfulServicePSM, authenticationModel, authenticationperformer, AuthenticationToken, JavaResourceModels, JavaResourceControllers, JavaResourceControllerManagers, JavaOutputDataModel, ResourceAccessPolicySet, ResourceAccessPolicy, ResourceAccessRule, MatchedResourceAttribute, MatchedContextAttribute, MatchedSubjectAttribute, AllowedAction

Produces code files:	JPAController.java
Calls other Aceleo Templates:	-

Table 10-7 javaHibernateUtil.mtl input/output

Acceleo Template Name:	javaHibernateUtil.mtl
Uses PSM elements:	RESTfulServicePSM, JavaResourceModels,
Produces code files:	HibernateUtil.java
Calls other Aceleo Templates:	-

Table 10-8 javaHTTPActivityHandler.mtl input/output

Acceleo Template Name:	javaHTTPActivityHandler.mtl
Uses PSM elements:	javaResourceControllers, javaResourceModels, HTTPActivities, javaResourceControllerManager, javaAlgoResourceControllers, SearchController, JavaRESTClientController,
Produces code files:	-
Calls other Aceleo Templates:	getResourceHandlerMultipleParents.mtl, putResourceHandlerMultipleParents.mtl, deleteResourceHandlerMultipleParents.mtl, getResourceHandlerSingleParent.mtl, putResourceHandlerSingleParent.mtl, deleteResourceHandlerSingleParent.mtl, getResourceHandlerNoParents.mtl, putResourceHandlerNoParents.mtl, deleteResourceHandlerNoParents.mtl, postResourceListHandlerMultipleParents.mtl, getResourceListHandlerMultipleParents.mtl, postResourceHandlerSingleParent.mtl, getResourceHandlerSingleParent.mtl, postResourceHandlerNoParents.mtl, getResourceListHandlerNoParents.mtl, generateSearchHTTPHandler.mtl, generateJavaRESTClientHTTPHandler.mtl

Table 10-9 javaHypermediaLink.mtl input/output

Acceleo Template Name:	javaHypermediaLink.mtl
Uses PSM elements:	RESTfulServicePSM
Produces code files:	HypermediaLink.java
Calls other Aceleo Templates:	-

Table 10-10 javaResourceController.mtl input/output

Acceleo Template Name:	javaResourceController.mtl
Uses PSM elements:	RESTfulServicePSM, javaResourceControllers, HTTPActivity, JAXRSAnnotation, JavaResourceModels, HTTPActivityHandlers
Produces code files:	javaResourceController's name + ".java"
Calls other Aceleo Templates:	-

Table 10-11 javaResourceControllerManager.mtl input/output

Acceleo Template Name:	javaResourceControllerManager.mtl
Uses PSM elements:	RESTfulServicePSM, JavaResourceControllerManager, HTTPActivityFunctionParameters, HTTPActivities, JAXRSAnnotations, JavaResourceModel, GuestMode, AuthenticationOnlyMode, BothMode
Produces code files:	javaResourceControllersManager's name + ".java"
Calls other Aceleo Templates:	-

Table 10-12 javaResourceModel.mtl input/output

Acceleo Template Name:	javaResourceModel.mtl
Uses PSM elements:	RESTfulServicePSM, javaResourceModels, PSMComponentProperties, JavaResourceModelManagers, JPAAnnotations, SearchableResourceModel, JavaGetterFunction, JavaSetterFunction, FunctionParameters, SearchableProperty, SubjectAttribute
Produces code files:	javaResourceModel's name + ".java"
Calls other Aceleo Templates:	-

Table 10-13 javaResourceModelManager.mtl input/output

Acceleo Template Name:	javaResourceModelManager.mtl
Uses PSM elements:	RESTfulServicePSM, javaResourceModelManager,
Produces code files:	javaResourceModelManager's name + ".java"
Calls other Acceleo Templates:	-

Table 10-14 mavenConfigurationFile.mtl input/output

Acceleo Template Name:	mavenConfigurationFile.mtl
Uses PSM elements:	RESTfulServicePSM
Produces code files:	pom.xml
Calls other Acceleo Templates:	-

Table 10-15 webXMLConfigurationFile.mtl input/output

Acceleo Template Name:	webXMLConfigurationFile.mtl
Uses PSM elements:	RESTfulServicePSM
Produces code files:	web.xml
Calls other Acceleo Templates:	-

Table 10-16 javaPersistentUtil.mtl input/output

Acceleo Template Name:	javaPersistentUtil.mtl
Uses PSM elements:	RESTfulServicePSM,
Produces code files:	PersistenceUtil.java
Calls other Acceleo Templates:	-

Table 10-17 javaLuceneStringSetBridge.mtl input/output

Acceleo Template Name:	javaLuceneStringSetBridge.mtl
Uses PSM elements:	RESTfulServicePSM
Produces code files:	SetStringFieldBridge.java
Calls other Acceleo Templates:	-

Table 10-18 persistenceXML.mtl input/output

Acceleo Template Name:	persistenceXML.mtl
Uses PSM elements:	RESTfulServicePSM
Produces code files:	persistence.xml
Calls other Acceleo Templates:	-

Table 10-19 JavaOutputModel.mtl input/output

Acceleo Template Name:	javaOutpuModel.mtl
Uses PSM elements:	RESTfulServicePSM, javaOutputModel, Properties
Produces code files:	JavaOutputModel's name (depending on case) + ".java"
Calls other Acceleo Templates:	-

Table 10-20 AuthorizationPolicyEvaluator.mtl input/output

Acceleo Template Name:	authorizationPolicyEvaluator.mtl
Uses PSM elements:	RESTfulServicePSM, ResourceAccessPolicySet, ResourceAccessPolicy, ResourceAccessRule, MatchedResourceAttribute, MatchedContextAttribute, MatchedSubjectAttribute, AllowedAction,
Produces code files:	JavaAuthorizationPolicyEvaluator.mtl
Calls other Acceleo Templates:	-

Table 10-21 putResourceHandlerSingleParent.mtl input/output

Acceleo Template Name:	putResourceHandlerSingleParent.mtl
Uses PSM elements:	RESTfulServicePSM, JavaResourceController, GuestModel, AuthenticationOnlyModel, BothModel, AuthenticationPerformer, AuthenticationToken, AuthorizationPolicyEvaluator, ResourceAccessPolicySet
Produces code files:	putResourceHandlerSingleParent's name + ".java"

Calls other Aceleo Templates:	-
--------------------------------------	---

The following Aceleo templates have the same input/output as the aforementioned `putResourceHandlerSingleParent.mtl` template, however, depending on the type and case of the HTTP handler the code differs:

- `deleteResourceHandlerMultipleParents`,
- `deleteResourceHandlerNoParents`,
- `deleteResourceHandlerSingleParent`,
- `getResourceHandlerMultipleParents`,
- `getResourceHandlerNoParents`,
- `getResourceHandlerSingleParent`,
- `getResourceListHandlerMultipleParents`,
- `getResourceListHandlerNoParents`,
- `getResourceListHandlerSingleParent`,
- `postResourceHandlerNoParents`,
- `postResourceHandlerSingleParent`,
- `postResourceHandlerMultipleParents`,
- `putResourceHandlerMultipleParents`,
- `putResourceHandlerNoParents`

Table 10-22 generateSearchHTTPHandler.mtl input/output

Acceleo Template Name:	generateSearchHTTPHandler.mtl
Uses PSM elements:	RESTfulServicePSM, javaAlgoResourceController, GuestMode, BothMode, AuthenticationOnlyMode, SearchableJavaResourceModel, SearchableProperty, AuthenticationPerformer, AuthenticationToken, SearchHTTPHandler, JavaResourceModel, AccessPolicyEvaluator, ResourceAccessPolicySet
Produces code files:	SearchHTTPHandler's name + ".java"
Calls other Aceleo Templates:	-

Table 10-23 javaRESTClientHTTPActivityHandler.mtl input/output

Acceleo Template Name:	javaRESTClientHTTPActivityHandler.mtl
Uses PSM elements:	RESTfulServicePSM, javaAlgoResourceController, GuestMode, BothMode, AuthenticationOnlyMode, AuthenticationPerformer, AuthenticationToken, JavaResourceModel, AccessPolicyEvaluator, ResourceAccessPolicySet, javaRESTClientModel, javaInputDataModel, javaOutputDataModel, QueryParams, Properties, Representations

Produces code files:	javaRESTClientHTTPActivityHandler's name + ".java"
Calls other Aceleo Templates:	-

Table 10-24 ResourceAccessPolicySet.mtl input/output

Acceleo Template Name:	javaResourceAccessPolicySet.mtl
Uses PSM elements:	javaResourceAccessPolicySet, javaResourceAccessPolicy, javaRuleCombiningAlgorithm, JPAAnnotation
Produces code files:	ResourceAccessPolicySet.java
Calls other Aceleo Templates:	-

Table 10-25 javaResourceAccessPolicy.mtl input/output

Acceleo Template Name:	javaResourceAccessPolicy.mtl
Uses PSM elements:	ResourceAccessPolicy, ResourceAccessRule, JPAAnnotation
Produces code files:	ResourceAccessPolicy.java
Calls other Aceleo Templates:	-

Table 10-26 javaResourceAccessRule.mtl input/output

Acceleo Template Name:	javaResourceAccessRule.mtl
Uses PSM elements:	ResourceAccessRule, JPAAnnotation, AllowedAction, MatchedResourceAttribute, MatchedContextAttribute, MatchedSubjectAttribute
Produces code files:	ResourceAccessRule.java
Calls other Aceleo Templates:	-

Table 10-27 javaAllowedAction.mtl input/output

Acceleo Template Name:	javaAllowedAction.mtl
Uses PSM elements:	AllowedAction, HTTPActivity, JPAAnnotation
Produces code files:	AllowedAction.java
Calls other Aceleo Templates:	-

Table 10-28 javaMatchedResourceAttribute.mtl input/output

Acceleo Template Name:	javaMatchedResourceAttribute.mtl
Uses PSM elements:	MatchedResourceAttribute, PSMComponentProperty, JPAAnnotation
Produces code files:	MatchedResourceAttribute.java
Calls other Acceleo Templates:	-

The *javaMatchedSubjectAttribute.mtl* as well as the *javaMatchedContextAttribute.mtl* have the same input/output as the *javaMatchedResourceAttribute.mtl* Acceleo template.

References

- [1] R.T. Fielding. (2000). Architectural Styles and the Design of Network-based Software Architecture", PhD Dissertation, Department of Information and Computer Science, University of California, Irvine
- [2] <http://rubyonrails.org>
- [3] <http://emf-rest.com>
- [4] <http://www.djangoproject-rest-framework.org>
- [5] <http://www.omg.org/mda/>
- [6] <http://eclipse.org/modeling/emf/>
- [7] <http://www.omg.org/spec/OCL/>
- [8] <https://eclipse.org/atl/>
- [9] Marcel van Amstel, Steven Boems, Ivan Kurtev and Luis Ferreira Pires. Performance in Model Transformations: Experiments with ATL and QVT. ICMT 2011, pp. 198-212
- [10] <http://martinfowler.com/articles/richardsonMaturityModel.html>
- [11] YAML, YAML Ain't Markup Language, The Official YAML Web Site, available online: <http://yaml.org/>
- [12] YEdit, Eclipse editor for editing YAML files, Eclipse marketplace, available online: <http://marketplace.eclipse.org/content/yedit>
- [13] Project Restmarks, RESTAPPS, S-CASE Consortium, 2014.
- [14] Cloudfier, a platform for rapid development and deployment of business apps
- [15] Thegrid.io, AI websites that design themselves.
- [16] Acceleo, a pragmatic implementation of the OMG MOF M2T standard. <http://www.eclipse.org/acceleo>
- [17] Basic Authentication, https://en.wikipedia.org/wiki/Basic_access_authentication
- [18] ABAC Authorization, https://en.wikipedia.org/wiki/Attribute-based_access_control
- [19] XACML, https://www.oasis-open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html
- [20] Maven Standard Directory Layout, <https://maven.apache.org/guides/introduction/introduction-to-the-standard-directory-layout.html>
- [21] J. Hutchinson, M. Rouncefield, and J. Whittle. Model-driven Engineering Practices in Industry. In 33rd ICSE, 2011, pages 633-642. IEEE, May 2011

A. Annex A – OCL Constraints Full List

Due to space limitations, the Ecore Core CIM, Authentication CIM extension, Database Searching CIM extension, ABAC Authorization CIM extension, Authentication PIM extension, Database Searching PIM extension, ABAC Authorization PIM extension, Athentication PSM extension, Database Searching PSM extension, ABAC Authorization PSM extension alongside their full OCL contraints can be found at <https://github.com/s-case/mde>. They should be opened with the OCLinEclipse editor.

B. Annex B – ATL/Acceleo Transformations Full List

Due to space limitations, the Core CIM to Core PIM, Authentication CIM extension to Authentication PIM extension, Database Searching CIM extension to Database Searching PIM extension, External Service Composition CIM extension to External Service Composition PIM extension, ABAC Authorization CIM extension to ABAC Authorization PIM extension, Authentication PIM extension to Authentication PSM extension, Database Searching PIM extension to Database Searching PSM extension, External Service Composition PIM extension to External Service Composition PSM extension and ABAC Authorization PIM to ABAC Authorization PSM ATL rules alongside the full Acceleo templates bundle can be found at <https://github.com/s-case/mde>.