Date: November 8 2024

Unified Architecture Framework (UAF) Domain Metamodel

Version 1.3

OMG Document Number: Normative dtc/2024-11-03

Standard document URL: http://www.omg.org/spec/UAF/1.3

Normative Machine Consumable File(s):

http://www.omg.org/spec/UAF/20241101/UAF.xmi

http://www.omg.org/spec/UAF/20241101/MeasurementsLibrary.xmi

Copyright © 2017-2021, IBM

Copyright © 2017-2021, KDM Analytics

Copyright © 2017-2021, Mega

Copyright © 2017-2021, Object Management Group, Inc.

Copyright © 2017-2021, No Magic Inc. a Dassault Systemes Company

Copyright © 2017-2021, PTC

Copyright © 2017-2021, Sparx Systems

USE OF SPECIFICATION - TERMS, CONDITIONS & NOTICES

The material in this document details an Object Management Group specification in accordance with the terms, conditions and notices set forth below. This document does not represent a commitment to implement any portion of this specification in any company's products. The information contained in this document is subject to change without notice.

LICENSES

The companies listed above have granted to the Object Management Group, Inc. (OMG) a nonexclusive, royalty-free, paid up, worldwide license to copy and distribute this document and to modify this document and distribute copies of the modified version. Each of the copyright holders listed above has agreed that no person shall be deemed to have infringed the copyright in the included material of any such copyright holder by reason of having used the specification set forth herein or having conformed any computer software to the specification.

Subject to all of the terms and conditions below, the owners of the copyright in this specification hereby grant you a fully-paid up, non-exclusive, nontransferable, perpetual, worldwide license (without the right to sublicense), to use this specification to create and distribute software and special purpose specifications that are based upon this specification, and to use, copy, and distribute this specification as provided under the Copyright Act; provided that: (1) both the copyright notice identified above and this permission notice appear on any copies of this specification; (2) the use of the specifications is for informational purposes and will not be copied or posted on any network computer or broadcast in any media and will not be otherwise resold or transferred for commercial purposes; and (3) no modifications are made to this specification. This limited permission automatically terminates without notice if you breach any of these terms or conditions. Upon termination, you will destroy immediately any copies of the specifications in your possession or control.

PATENTS

The attention of adopters is directed to the possibility that compliance with or adoption of OMG specifications may require use of an invention covered by patent rights. OMG shall not be responsible for identifying patents for which a license may be required by any OMG specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. OMG specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

GENERAL USE RESTRICTIONS

Any unauthorized use of this specification may violate copyright laws, trademark laws, and communications regulations and statutes. This document contains information which is protected by copyright. All Rights Reserved. No part of this work covered by copyright herein may be reproduced or used in any form or by any means--graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems--without permission of the copyright owner.

DISCLAIMER OF WARRANTY

WHILE THIS PUBLICATION IS BELIEVED TO BE ACCURATE, IT IS PROVIDED "AS IS" AND MAY CONTAIN ERRORS OR MISPRINTS. THE OBJECT MANAGEMENT GROUP AND THE COMPANIES LISTED ABOVE MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS PUBLICATION, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE OR USE. IN NO EVENT SHALL THE OBJECT MANAGEMENT GROUP OR ANY OF THE COMPANIES LISTED ABOVE BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, RELIANCE OR COVER DAMAGES, INCLUDING LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY ANY USER OR ANY THIRD PARTY IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The entire risk as to the quality and performance of software developed using this specification is borne by you. This disclaimer of warranty constitutes an essential part of the license granted to you to use this specification.

RESTRICTED RIGHTS LEGEND

Use, duplication or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c) (1) (ii) of The Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 or in subparagraph (c)(1) and (2) of the Commercial Computer Software - Restricted Rights clauses at 48 C.F.R. 52.227-19 or as specified in 48 C.F.R. 227-7202-2 of the DoD F.A.R. Supplement and its successors, or as specified in 48 C.F.R. 12.212 of the Federal Acquisition

Regulations and its successors, as applicable. The specification copyright owners are as indicated above and may be contacted through the Object Management Group, 109 Highland Avenue, Needham, MA 02494, U.S.A.

TRADEMARKS

CORBA®, CORBA logos®, FIBO®, Financial Industry Business Ontology®, FINANCIAL INSTRUMENT GLOBAL IDENTIFIER®, IIOP®, IMM®, Model Driven Architecture®, MDA®, Object Management Group®, OMG®, OMG Logo®, SoaML®, SOAML®, SysML®, UAF®, Unified Modeling Language®, UML®, UML Cube Logo®, VSIPL®, and XMI® are registered trademarks of the Object Management Group, Inc.

For a complete list of trademarks, see: http://www.omg.org/legal/tm_list.htm. All other products or company names mentioned are used for identification purposes only, and may be trademarks of their respective owners.

COMPLIANCE

The copyright holders listed above acknowledge that the Object Management Group (acting itself or through its designees) is and shall at all times be the sole entity that may authorize developers, suppliers and sellers of computer software to use certification marks, trademarks or other special designations to indicate compliance with these materials.

Software developed under the terms of this license may claim compliance or conformance with this specification if and only if the software compliance is of a nature fully matching the applicable compliance points as stated in the specification. Software developed only partially matching the applicable compliance points may claim only that the software was based on this specification, but may not claim compliance or conformance with this specification. In the event that testing suites are implemented or approved by Object Management Group, Inc., software developed using this specification may claim compliance or conformance with the specification only if the software satisfactorily completes the testing suites.

Table of Contents

PREFACE **15**

1. SCOPE **17**

1.1 INTRODUCTION 17
1.2 UAF BACKGROUND 17
1.3 INTENDED USAGE 18
1.4 RELATED DOCUMENTS 18

2. CONFORMANCE **19**

3. REFERENCES **20**

3.1 NORMATIVE REFERENCES 20
3.1.1 OMG Documents (Normative References) *20*
3.2 OTHER NORMATIVE REFERENCES 20
3.3 INFORMATIVE REFERENCES 21

4. TERMS AND DEFINITIONS **22**

5. ACRONYMS **23**

6. ADDITIONAL INFORMATION **25**

6.1 CHANGES TO ADOPTED OMG SPECIFICATIONS 25
6.2 LANGUAGE ARCHITECTURE 25
6.3 PHILOSOPHY 25
6.4 CORE PRINCIPLES 25

7. UAF GRID **27**

7.1 DESCRIPTIONS OF VIEWPOINTS AND ASPECTS 28
7.2 VIEWPOINT INTERRELATIONSHIPS 31
7.3 DOMAIN METAMODEL DIAGRAM LEGEND 31

8. DOMAIN METAMODEL (DMM) DIAGRAMS **34**

8.1 VIEW SPECIFICATIONS 34
8.1.1 View Specifications::Architecture Management *34*
8.1.2 View Specifications::Summary & Overview *40*
8.1.3 View Specifications::Strategic *41*
8.1.4 View Specifications::Operational *52*
8.1.5 View Specifications::Services *64*
8.1.6 View Specifications::Personnel *75*
8.1.7 View Specifications::Resources *89*
8.1.8 View Specifications::Security *104*
8.1.9 View Specifications::Projects *113*
8.1.10 View Specifications::Standards *119*
8.1.11 View Specifications::Actual Resources *123*
8.1.12 View Specifications::Motivation *125*
8.1.13 View Specifications::Information *126*
8.1.14 View Specifications::Parameters *128*
8.1.15 View Specifications::Other *133*

9. DOMAIN METAMODEL (DMM) ELEMENTS **137**

9.1 DOMAIN METAMODEL 137
9.1.1 Domain MetaModel::Architecture Management *137*
9.1.2 Domain MetaModel::Summary & Overview *158*
9.1.3 Domain MetaModel::Strategic *163*
9.1.4 Domain MetaModel::Operational *189*
9.1.5 Domain MetaModel::Services *204*
9.1.6 Domain MetaModel::Personnel *216*

9.1.7 Domain MetaModel::Resources *224*
9.1.8 Domain MetaModel::Security *243*
9.1.9 Domain MetaModel::Projects *257*
9.1.10 Domain MetaModel::Standards *264*
9.1.11 Domain MetaModel::Actual Resources *267*
9.1.12 Domain MetaModel::Parameters *275*

TABLE OF FIGURES

Figure 7:1- UAF Grid 27
Figure 7:2 – Viewpoint Interrelationships 31
Figure 7:3 - Legend of color codes for element types defined in UAF 32
Figure 8:1 - Architecture Principles 34
Figure 8:2 - Architecture Views 35
Figure 8:3 - Architecture References 36
Figure 8:4 - Architecture Development Method 36
Figure 8:5 - Architecture Status 37
Figure 8:6 - Dictionary 37
Figure 8:7 - Architecture Parameters 38
Figure 8:8 - Architecture Constraints 39
Figure 8:9 - Architecture Roadmap 39
Figure 8:10 - Architecture Traceability 40
Figure 8:11 - Summary & Overview 41
Figure 8:12 - Strategic Motivation 42
Figure 8:13 - Strategic Taxonomy 43
Figure 8:14 - Strategic Structure 43
Figure 8:15 - Strategic Connectivity 44
Figure 8:16 - Strategic Processes 45
Figure 8:17 - Strategic Processes: Mission Phases 46
Figure 8:18 - Strategic States 47
Figure 8:19 - Strategic Information 49
Figure 8:20 - Strategic Constraints 49
Figure 8:21 - Strategic Roadmap: Deployment 50
Figure 8:22 - Strategic Roadmap: Phasing 51
Figure 8:23 - Strategic Traceability 52
Figure 8:24 - Operational Taxonomy 53
Figure 8:25 - Operational Structure 54
Figure 8:26 - Operational Connectivity 55
Figure 8:27 - Operational Processes 56
Figure 8:28 - Operational Processes BPMN Semantics 58
Figure 8:29 - Operational Processes: Mission Threads 59
Figure 8:30 - Operational States 61
Figure 8:31 - Operational Sequences 62
Figure 8:32 - Operational Constraints 63

Figure 8:33 - Operational Traceability 64
Figure 8:34 - Services Taxonomy 65
Figure 8:35 - Services Structure 66
Figure 8:36 - Services Connectivity 67
Figure 8:37 - Services Processes 68
Figure 8:38 - Services Processes BPMN Semantics 69
Figure 8:39 - Services States 71
Figure 8:40 - Services Sequences 72
Figure 8:41 - Services Constraints 73
Figure 8:42 - Services Roadmap 74
Figure 8:43 - Services Traceability 75
Figure 8:44 - Personnel Taxonomy 76
Figure 8:45 - Personnel Structure 77
Figure 8:46 - Personnel Connectivity 78
Figure 8:47 - Personnel Processes 79
Figure 8:48 - Personnel States 80
Figure 8:49 - Personnel Sequences 81
Figure 8:50 - Personnel Constraints: Competence 82
Figure 8:51 - Personnel Constraints: Drivers 84
Figure 8:52 - Personnel Constraints: Performance 85
Figure 8:53 - Personnel Roadmap: Availability 86
Figure 8:54 - Personnel Roadmap: Evolution 87
Figure 8:55 - Personnel Roadmap: Forecast 88
Figure 8:56 - Personnel Traceability 89
Figure 8:57 - Resources Taxonomy 90
Figure 8:58 - Resources Structure 91
Figure 8:59 - Resources Connectivity 93
Figure 8:60 - Resources Processes 94
Figure 8:61 - Resources Processes BPMN Semantics 96
Figure 8:62 - Resources Processes: Mission Engineering Threads 97
Figure 8:63 - Resources States 99
Figure 8:64 - Resources Sequences 100
Figure 8:65 - Resources Constraints 101
Figure 8:66 - Resources Roadmap: Evolution 102
Figure 8:67 - Resources Roadmap: Forecast 103
Figure 8:68 - Resources Traceability 104
Figure 8:69 - Security Controls 105

Figure 8:70 - Security Taxonomy 106
Figure 8:71 - Security Structure 107
Figure 8:72 - Security Connectivity 109
Figure 8:73 - Security Processes 110
Figure 8:74 - Security Constraints 111
Figure 8:75 - Security Traceability 112
Figure 8:76 - Project Taxonomy 114
Figure 8:77 - Project Structure 115
Figure 8:78 - Project Connectivity 116
Figure 8:79 - Project Processes 117
Figure 8:80 - Project Roadmap 118
Figure 8:81 - Project Traceability 119
Figure 8:82 - Standards Taxonomy 120
Figure 8:83 - Standards Structure 121
Figure 8:84 - Standards Roadmap 122
Figure 8:85 - Standards Traceability 122
Figure 8:86 - Actual Resources Structure 123
Figure 8:87 - Actual Resources Connectivity 124
Figure 8:88 - Actual Resources Traceability 125
Figure 8:89 - Motivation: Requirements 126
Figure 8:90 - Information: Operational Information 127
Figure 8:91 - Information: Resources Information 128
Figure 8:92 - Parameters: Environment 129
Figure 8:93 - Parameters: Measurements 130
Figure 8:94 - Parameters: Mission Vignette and Scenario 131
Figure 8:95 - Parameters: Risk 132
Figure 8:96 - BPMN 134
Figure 8:97 - IEPPV 135
Figure 8:98 - NIEM 136
Figure 9:1 - ActualState 137
Figure 9:2 - DesignationKind 138
Figure 9:3 - InteractionScenarioGeneralization 138
Figure 9:4 - ISO8601DateTime 139
Figure 9:5 - OpposableElement 139
Figure 9:6 - ProcessGeneralization 140
Figure 9:7 - PropertySetGeneralization 140
Figure 9:8 - StateDescriptionGeneralization 141

Figure 9:9 - Exchange 141
Figure 9:10 - ExchangeItem 142
Figure 9:11 - Resource 143
Figure 9:12 - CapableElement 143
Figure 9:13 - IsCapableToPerform 144
Figure 9:14 - PerformsInContext 145
Figure 9:15 - Process 145
Figure 9:16 - ProcessEdge 146
Figure 9:17 - ProcessOperation 146
Figure 9:18 - ProcessParameter 147
Figure 9:19 - ProcessUsage 147
Figure 9:20 - StateDescription 148
Figure 9:21 - InteractionRole 148
Figure 9:22 - InteractionScenario 149
Figure 9:23 - Alias 149
Figure 9:24 - ArchitectureMetadata 150
Figure 9:25 - ConflictsWith 150
Figure 9:26 - Definition 151
Figure 9:27 - Information 151
Figure 9:28 - InformationModel 152
Figure 9:29 - Metadata 153
Figure 9:30 - Opposes 153
Figure 9:31 - SameAs 154
Figure 9:32 - Rule 155
Figure 9:33 - ArchitecturalReference 155
Figure 9:34 - ComparesTo 156
Figure 9:35 - Implements 157
Figure 9:36 - Sequence 157
Figure 9:37 - ArchitecturalDescription 158
Figure 9:38 - Architecture 159
Figure 9:39 - Concern 160
Figure 9:40 - Phases 160
Figure 9:41 - Stakeholder 161
Figure 9:42 - UAFElement 162
Figure 9:43 - View 163
Figure 9:44 - Viewpoint 163
Figure 9:45 - Challenge 164

Figure 9:46 - Driver 164
Figure 9:47 - Enables 165
Figure 9:48 - ImpactedBy 165
Figure 9:49 - MotivatedBy 166
Figure 9:50 - MotivationalElement 166
Figure 9:51 - Opportunity 167
Figure 9:52 - PresentedBy 167
Figure 9:53 - Capability 168
Figure 9:54 - CapabilityGeneralization 168
Figure 9:55 - EnterpriseGoal 169
Figure 9:56 - EnterpriseObjective 170
Figure 9:57 - EnterpriseVision 170
Figure 9:58 - Mission 171
Figure 9:59 - OwnsValue 171
Figure 9:60 - PhaseableElement 172
Figure 9:61 - StrategicAsset 172
Figure 9:62 - StrategicPhase 173
Figure 9:63 - ValueItem 173
Figure 9:64 - VisionStatement 174
Figure 9:65 - WholeLifeEnterprise 174
Figure 9:66 - CapabilityRole 175
Figure 9:67 - StructuralPart 175
Figure 9:68 - TemporalPart 176
Figure 9:69 - CapabilityDependency 176
Figure 9:70 - CapabilityRoleDependency 177
Figure 9:71 - StrategicExchange 177
Figure 9:72 - ActualEnduringTask 178
Figure 9:73 - ActualEnterprisePhase 178
Figure 9:74 - ActualMission 179
Figure 9:75 - ActualMissionPhase 179
Figure 9:76 - ActualStrategicPhase 180
Figure 9:77 - Creates 180
Figure 9:78 - Defines 181
Figure 9:79 - ValueStream 181
Figure 9:80 - Achiever 182
Figure 9:81 - Achieves 182
Figure 9:82 - ActualEffect 183

Figure 9:83 - ActualOutcome 183
Figure 9:84 - Desirer 184
Figure 9:85 - Desires 184
Figure 9:86 - Effect 185
Figure 9:87 - MapsToGoal 185
Figure 9:88 - StrategicConstraint 186
Figure 9:89 - SubjectOfStrategicConstraint 187
Figure 9:90 - EvokedBy 187
Figure 9:91 - Exhibits 188
Figure 9:92 - MapsToCapability 188
Figure 9:93 - OrganizationInPhase 189
Figure 9:94 - ArbitraryConnector 189
Figure 9:95 - ConceptItem 190
Figure 9:96 - HighLevelOperationalConcept 190
Figure 9:97 - KnownResource 191
Figure 9:98 - OperationalAgent 191
Figure 9:99 - OperationalArchitecture 192
Figure 9:100 - OperationalMethod 192
Figure 9:101 - OperationalParameter 193
Figure 9:102 - OperationalPerformer 193
Figure 9:103 - OperationalRole 194
Figure 9:104 - ProblemDomain 194
Figure 9:105 - OperationalConnector 195
Figure 9:106 - OperationalExchange 195
Figure 9:107 - OperationalExchangeItem 196
Figure 9:108 - OperationalInterface 197
Figure 9:109 - OperationalPort 197
Figure 9:110 - OperationalSignal 198
Figure 9:111 - MissionTask 198
Figure 9:112 - MissionTaskAction 199
Figure 9:113 - MissionThread 199
Figure 9:114 - OperationalActivity 200
Figure 9:115 - OperationalActivityAction 200
Figure 9:116 - OperationalActivityEdge 201
Figure 9:117 - StandardOperationalActivity 201
Figure 9:118 - OperationalStateDescription 202
Figure 9:119 - OperationalInteractionScenario 202

Figure 9:120 - OperationalMessage 203
Figure 9:121 - OperationalInformation 203
Figure 9:122 - OperationalConstraint 204
Figure 9:123 - SubjectOfOperationalConstraint 204
Figure 9:124 - Service 205
Figure 9:125 - ServiceArchitecture 205
Figure 9:126 - ServiceGeneralization 206
Figure 9:127 - ServiceMethod 207
Figure 9:128 - ServiceParameter 207
Figure 9:129 - ServiceRole 208
Figure 9:130 - ServiceConnector 208
Figure 9:131 - ServiceExchange 209
Figure 9:132 - ServiceExchangeItem 209
Figure 9:133 - ServiceInterface 210
Figure 9:134 - ServicePort 210
Figure 9:135 - ServiceSignal 211
Figure 9:136 - ServiceFunction 211
Figure 9:137 - ServiceFunctionAction 212
Figure 9:138 - ServiceFunctionEdge 212
Figure 9:139 - ServiceStateDescription 213
Figure 9:140 - ServiceInteractionScenario 213
Figure 9:141 - ServiceMessage 214
Figure 9:142 - ServiceContract 214
Figure 9:143 - ServicePolicy 215
Figure 9:144 - GovernedBy 215
Figure 9:145 - Supports 216
Figure 9:146 - Organization 217
Figure 9:147 - OrganizationalResource 217
Figure 9:148 - Person 218
Figure 9:149 - Post 218
Figure 9:150 - Responsibility 219
Figure 9:151 - PostRole 219
Figure 9:152 - SubOrganization 220
Figure 9:153 - Command 220
Figure 9:154 - Control 221
Figure 9:155 - ResourceInteractionScenario 221
Figure 9:156 - Competence 222

Figure 9:157 - CompetenceForRole 222
Figure 9:158 - RequiresCompetence 223
Figure 9:159 - FillsPost 223
Figure 9:160 - CompetenceToConduct 224
Figure 9:161 - CapabilityConfiguration 224
Figure 9:162 - NaturalResource 225
Figure 9:163 - PhysicalResource 225
Figure 9:164 - ResourceArchitecture 226
Figure 9:165 - ResourceArtifact 226
Figure 9:166 - ResourcePerformer 227
Figure 9:167 - ResourceService 228
Figure 9:168 - Software 228
Figure 9:169 - System 229
Figure 9:170 - ResourceMethod 229
Figure 9:171 - ResourceParameter 230
Figure 9:172 - ResourcePort 230
Figure 9:173 - ResourceRole 231
Figure 9:174 - ResourceConnector 232
Figure 9:175 - ResourceExchange 232
Figure 9:176 - ResourceExchangeItem 233
Figure 9:177 - ResourceInterface 234
Figure 9:178 - ResourceServiceInterface 234
Figure 9:179 - ResourceSignal 235
Figure 9:180 - Function 235
Figure 9:181 - FunctionAction 236
Figure 9:182 - FunctionEdge 236
Figure 9:183 - MissionEngineeringThread 237
Figure 9:184 - ResourceStateDescription 237
Figure 9:185 - ResourceMessage 238
Figure 9:186 - ResourceInformation 238
Figure 9:187 - ResourceConstraint 239
Figure 9:188 - SubjectOfResourceConstraint 239
Figure 9:189 - Forecast 240
Figure 9:190 - SubjectOfForecast 240
Figure 9:191 - Technology 241
Figure 9:192 - VersionedElement 241
Figure 9:193 - VersionOfConfiguration 242

Figure 9:194 - VersionSuccession 242
Figure 9:195 - WholeLifeConfiguration 243
Figure 9:196 - ProtocolImplementation 243
Figure 9:197 - EnhancedSecurityControl 244
Figure 9:198 - Enhances 244
Figure 9:199 - Protects 245
Figure 9:200 - ProtectsInContext 245
Figure 9:201 - SecurityControl 246
Figure 9:202 - SecurityControlFamily 247
Figure 9:203 - Asset 247
Figure 9:204 - OperationalAsset 248
Figure 9:205 - OperationalMitigation 248
Figure 9:206 - ResourceAsset 249
Figure 9:207 - ResourceMitigation 249
Figure 9:208 - SecurityEnclave 250
Figure 9:209 - AssetRole 250
Figure 9:210 - OperationalInformationRole 251
Figure 9:211 - ResourceInformationRole 251
Figure 9:212 - SecurityProcess 252
Figure 9:213 - SecurityProcessAction 252
Figure 9:214 - Caveat 253
Figure 9:215 - SecurityAvailability 253
Figure 9:216 - SecurityCategory 254
Figure 9:217 - SecurityClassification 254
Figure 9:218 - SecurityClassificationKind 255
Figure 9:219 - SecurityConstraint 255
Figure 9:220 - SecurityIntegrity 256
Figure 9:221 - SecurityMeasurement 256
Figure 9:222 - SecurityRisk 257
Figure 9:223 - SubjectOfSecurityConstraint 257
Figure 9:224 - Project 258
Figure 9:225 - ProjectMilestone 258
Figure 9:226 - ActualProjectMilestoneRole 259
Figure 9:227 - ProjectMilestoneRole 259
Figure 9:228 - ProjectStatus 260
Figure 9:229 - ProjectTheme 260
Figure 9:230 - StatusIndicators 261

Figure 9:231 - MilestoneDependency 261
Figure 9:232 - ProjectSequence 262
Figure 9:233 - ProjectActivity 262
Figure 9:234 - ProjectActivityAction 263
Figure 9:235 - ActualProject 263
Figure 9:236 - ActualProjectMilestone 264
Figure 9:237 - ResponsibleFor 264
Figure 9:238 - Doctrine 265
Figure 9:239 - Protocol 266
Figure 9:240 - ProtocolStack 266
Figure 9:241 - Standard 267
Figure 9:242 - ProtocolLayer 267
Figure 9:243 - ActualOrganization 268
Figure 9:244 - ActualOrganizationalResource 269
Figure 9:245 - ActualPerson 269
Figure 9:246 - ActualPost 270
Figure 9:247 - ActualResource 270
Figure 9:248 - ActualResourceRelationship 271
Figure 9:249 - ActualResponsibility 271
Figure 9:250 - ActualResponsibleResource 272
Figure 9:251 - FieldedCapability 272
Figure 9:252 - ActualService 273
Figure 9:253 - ProvidedServiceLevel 273
Figure 9:254 - ProvidesCompetence 274
Figure 9:255 - RequiredServiceLevel 274
Figure 9:256 - OwnsProcess 275
Figure 9:257 - ActualCondition 275
Figure 9:258 - ActualEnvironment 276
Figure 9:259 - ActualLocation 276
Figure 9:260 - ActualMeasurement 277
Figure 9:261 - ActualMeasurementSet 277
Figure 9:262 - ActualMissionScenario 278
Figure 9:263 - ActualMissionVignette 278
Figure 9:264 - ActualPropertySet 279
Figure 9:265 - ActualRisk 279
Figure 9:266 - AffectableElement 280
Figure 9:267 - Affects 280

Figure 9:268 - AffectsInContext 281
Figure 9:269 - Condition 281
Figure 9:270 - Environment 282
Figure 9:271 - EnvironmentProperty 282
Figure 9:272 - GeoPoliticalExtentType 283
Figure 9:273 - Location 284
Figure 9:274 - LocationHolder 284
Figure 9:275 - MeasurableElement 285
Figure 9:276 - Measurement 286
Figure 9:277 - MeasurementSet 286
Figure 9:278 - MissionScenario 287
Figure 9:279 - MissionVignette 287
Figure 9:280 - Mitigates 288
Figure 9:281 - OwnsRisk 288
Figure 9:282 - OwnsRiskInContext 289
Figure 9:283 - PropertySet 289
Figure 9:284 - Risk 290

TABLE OF TABLES

Table 1:1 - Table of Related Documents 18
Table 5:1 - Description of acronyms used in this specification 23
Table 7:1 - Definitions for the Viewpoints 28
Table 7:2 - Definitions of the Aspects 29

Preface

OMG

Founded in 1989, the Object Management Group, Inc. (OMG) is an open membership, not-for-profit computer industry standards consortium that produces and maintains computer industry specifications for interoperable, portable and reusable enterprise applications in distributed, heterogeneous environments. Membership includes Information Technology vendors, end users, government agencies and academia. OMG member companies write, adopt, and maintain its specifications following a mature, open process. OMG's specifications implement the Model Driven Architecture® (MDA®), maximizing ROI through a full-lifecycle approach to enterprise integration that covers multiple operating systems, programming languages, middleware and networking infrastructures, and software development environments. OMG's specifications include: UML® (Unified Modeling Language™); CORBA® (Common Object Request Broker Architecture); CWM™ (Common Warehouse Metamodel); and industry-specific standards for dozens of vertical markets. More information on the OMG is available at *http://www.omg.org/*.

OMG Specifications

As noted, OMG specifications address middleware, modeling and vertical domain frameworks. All OMG Specifications are available from this URL: *http://www.omg.org/spec*

Specifications are organized by the following categories:

Business Modeling Specifications

Middleware Specifications

- CORBA/IIOP
- Data Distribution Services
- Specialized CORBA IDL/Language Mapping Specifications

Modeling and Metadata Specifications

- UML, MOF, CWM, XMI
- UML Profile Specifications

Platform Independent Model (PIM) - Platform Specific Model (PSM) - Interface Specifications

- CORBAServices
- CORBAFacilities
- OMG Domain Specifications
- CORBA Embedded Intelligence Specifications
- CORBA Security Specifications

All of OMG's formal specifications may be downloaded without charge from our website. (Products implementing OMG specifications are available from individual suppliers.) Copies of specifications, available in PostScript and PDF format, may be obtained from the Specifications Catalog cited above or by contacting the Object Management Group, Inc. at: OMG Headquarters 109 Highland Avenue, Needham, MA 02494 USA Tel: +1- 781-444-0404 Fax: +1-781-444-0320 Email: *pubs@omg.org*

Certain OMG specifications are also available as ISO standards. Please consult *http://www.iso.org*

Typographical Conventions

The type styles shown below are used in this document to distinguish programming statements from ordinary English. However, these conventions are not used in tables or section headings where no distinction is necessary.

Times/Times New Roman - 10 pt.: Standard body text

Helvetica/Arial - 10 pt. Bold: OMG Interface Definition Language (OMG IDL) and syntax elements.

Courier - 10 pt. Bold: Programming language elements.

Helvetica/Arial - 10 pt: Exceptions

Note – Terms that appear in *italics* are defined in the glossary. Italic text also represents the name of a document, specification, or other publication.

1. Scope

1.1 Introduction

There are four parts to this specification, two are normative and two informative. The normative parts are:

1. The UAF Domain Metamodel (DMM) (this document) that provides the definition of concepts, relationships and viewpoints for the framework. The UAF DMM is the basis for any implementation of UAF including non-UML/SysML implementations.
2. The UAF Modeling Language (UAFML) (see document dtc/2024-11-01) is a UML/SysML implementation of the UAF DMM

The informative parts are:

3. The UAF Traceability, Annex A (see document dtc/2024-11-08), which details the mappings between the UAF and the various frameworks and languages that contribute to the UAF.
4. The UAF Example Model, Annex B (see document dtc/2024-11-09), which illustrates a practical usage of UAF.
5. The UAF Enterprise Architecture Guide (EAG), Annex C (see document dtc/2024-11-10), which provides guidance on how to use the UAFML to model an architecture.

1.2 UAF Background

UAF evolved from the Unified Profile for DoDAF and MODAF (UPDM), version 2.1. UAF extends the scope of UPDM and generalizes it to make it applicable to commercial as well as military architectures. The intent of UAF is to provide a standard representation for describing enterprise architectures using a Model Based Systems Engineering (MBSE) approach.

The core concepts in the UAF are based upon the DoDAF 2.0.2 Domain Metamodel (DM2) and the MODAF ontological data exchange mechanism (MODEM), Security Views from Canada's Department of National Defense Architecture Framework (DNDAF) and the North Atlantic Treaty Organization (NATO) Architecture Framework (NAF) v 4.

UAF models describe a system[1] from a set of stakeholders' concerns such as security or information through a set of predefined viewpoints. Developed models can also reflect custom viewpoints or users can develop more formal extensions for new viewpoints.

The UAFML can be used to develop architectures compliant with:

- Department of Defense Architecture Framework (DoDAF) version 2.02
- Ministry of Defence Architecture Framework (MODAF) version 1.3
- North Atlantic Treaty Organization (NATO) Architecture Framework (NAF) version 3.1
- North Atlantic Treaty Organization (NATO) Architecture Framework (NAF) version 4

UAF v 1.2 supports the capability to:

- model architectures for a broad range of complex systems, which may include hardware, software, data, personnel, and facility elements;

[1] The term system is used from: "Systems and software engineering -- Architecture description," http://www.iso.org/iso/catalogue_detail.htm?csnumber=50508

- model consistent architectures for system-of-systems (SoS) down to lower levels of design and implementation;
- support the analysis, specification, design, and verification of complex systems; and
- improve the ability to exchange architecture information among related tools that are SysML based.

1.3 Intended Usage

The UAF enables the modeling of strategic capabilities, operational scenarios, services, resources, personnel, security, projects, standards, measures and requirements; which supports best practices through, separation of concerns and abstractions. In addition, the UAF enables the modeling of related architecture concepts such as:

- System of Systems (SoS),
- information exchanges consistent with the National Information Exchange Model (NIEM),
- DoD's doctrine, organization, training material, leadership & education, personnel, and facilities (DOTMLPF)
- UK Ministry of Defence Lines of Development (DLOD) elements,
- Human Computer Interfaces (HCI).

Further, The UAF conforms to terms defined in the ISO/IEC/IEEE 42010 standard for architecture description, where the terms: architecture, architecture description (AD), architecture framework, architecture view, architecture viewpoint, concern, environment, model kind, stakeholder [ISO/IEC/IEEE 42010:2011] form correspondence rules specified as constraints on UAF.

1.4 Related Documents

The specification includes a metamodel and description as separate documents. Other appendices are also provided as separate documents. The table below provides a listing of these documents:

Table 1:1 - Table of Related Documents

dtc/2024-11-03	The UAF Domain MetaModel (DMM)
dtc/2024-11-01	The UAF Modeling Language (UAFML)
dtc/2024-11-08	Appendix A that contains a separate traceability subsection from UAFML to each of the frameworks listed in Section 1.2 of this specification
dtc/2024-11-09	Appendix B: An example of how the language can be used to represent a UAFML architecture
dtc/2024-11-10	Appendix C: An Enterprise Architecture Guide (EAG)
dtc/2024-11-05	UAF XMI file
dtc/2024-11-06	UAF XMI Measurements library
dtc/2024-11-11	Attachments

2. Conformance

UAF specifies four types of conformance.

Type 1 Conformance: - UAF View specification conformance. A tool demonstrating view specification conformance shall implement a version of all the view specifications defined in the UAF Grid, with the exception of the view specifications in the Architecture Management Viewpoint. Optionally the tool vendor can implement other donor framework viewpoints, for instance DoDAF, MODAF or NAF based upon the mapping between them and UAF provided in Appendix A (dtc/2024-11-08)

Type 2 Conformance: - UAF Conceptual Syntax Conformance. A tool demonstrating conceptual syntax conformance is consistent with the concepts, relationships and constraints defined in the UAF DMM (this document). UAF Conceptual Syntax Conformance implies Type 1 Conformance.

Type 3 Conformance: - UAF Formal Syntax Conformance. A tool demonstrating formal syntax conformance:

- enables instances of concrete UAFML stereotypes defined in the UAFML (dtc/2024-11-01)
- complies with the constraints defined in the UAFML (dtc/2024-11-01)
- complies with the SysML version 1.6 Concrete Syntax Conformance (formal/19-11-01)
- UAF Formal Syntax Conformance implies Type 2 Conformance.

Type 4 Conformance: - UAF Model interchange conformance. A tool demonstrating model interchange conformance can import and export conformant XMI for all valid UAFML models. Model interchange conformance implies Type 3 Conformance.

3. References

3.1 Normative References

The following normative documents contain provisions which, through reference in this text, constitute provisions of this specification. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply.

3.1.1 OMG Documents (Normative References)

- Unified Modeling Language (UML), 2.5.1, December 2017, http://www.omg.org/spec/UML
- Object Constraint Language (OCL), 2.4, February 2014, http://www.omg.org/spec/OCL
- System Modeling Language (SysML),1.6, November 2019, http://www.omg.org/spec/SysML
- Diagram Definition (DD), 1.1, June 2015, http://www.omg.org/spec/DD
- UML Profile for the National Information Exchange Model (NIEM UML), 3.0, April 2017, http://www.omg.org/spec/NIEM-UML
- UML Profile for BPMN Processes, 1.0, July 2014, http://www.omg.org/spec/BPMNProfile
- Information Exchange Packaging Policy Vocabulary (IEPPV) 1.0, May 2015, http://www.omg.org/spec/IEPPV
- XML Metadata Interchange, 2.5.1, June 2015, https://www.omg.org/spec/XMI/2.5.1/About-XMI/
- Business Motivation Model (BMM), Version 1.3, http://www.omg.org/spec/BMM/1.3/

3.2 Other Normative References

- Department of Defense Architecture Framework (DoDAF), Version 2.02, August 2010, http://dodcio.defense.gov/Library/DoDArchitectureFramework.aspx
- The DM2 (DoDAF Meta-Model) Conceptual Data Model, http://dodcio.defense.gov/Library/DoDArchitectureFramework/dodaf20_conceptual.aspx
- DM2 Logical Data Model, http://dodcio.defense.gov/Library/DoDArchitectureFramework/dodaf20_logical.aspx
- DM2 Formal Ontology. http://dodcio.defense.gov/Library/DoDArchitectureFramework/dodaf20_ontology1.aspx
- Department National Defence and Canadian Forces (DND/ CF) Architecture Framework (DNDAF), Version 1.8.1, 25 January 2013
- International Defence Enterprise Architecture Specification for Exchange (IDEAS) Group, https://en.wikipedia.org/wiki/IDEAS_Group
- ISO/IEC/IEEE 42010:2011, Systems and software engineering – Architecture Description, http://www.iso.org/iso/catalogue_detail.htm?csnumber=50508
- Ministry of Defence Architecture Framework (MODAF), https://www.gov.uk/mod-architecture-framework
- MODAF Ontological Data Exchange Mechanism (MODEM)
- https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/63980/20130117_MODAF_MODEM.pdf
- NATO Architecture Framework (NAF) Version 3, NATO C3 BOARD (AC/322-D(2007)0048), (no longer publicly available online as of 3 November 2015)
- NATO Architecture Framework v4.0, January 2021, https://www.nato.int/cps/en/natohq/topics_157575.htm

3.3 Informative References

- ISO 15704:2000, Industrial Automation Systems – "Requirements for Enterprise-Reference Architectures and Methodologies," http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=28777
- ISO 8601:2004 Data elements and interchange formats – Information interchange – Representation of dates and times, http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?ics1=01&ics2=140&ics3=30&csnumber=40874
- ISO/IEC 15288:2015, "Systems Engineering - Systems Life Cycle Processes," http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=63711
- Object Management Group (OMG), Metamodel Extension Facility, Initial submission, ad/12-02-01, http://www.omg.org/cgi-bin/doc?ad/12-02-01 (Requires OMG Member Access)
- OASIS SOA-RAF, Reference Architecture Foundation for Service Oriented Architecture Version 1.0, OASIS SOA Reference Model TC, 04 December 2012. http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/cs01/soa-ra-v1.0-cs01.pdf (Authoritative)
- Object Management Group (OMG), Semantics of Business Vocabulary and Business Rules (SBVR), Version 1.5, December 2019, http://www.omg.org/spec/SBVR
- International Council On Systems Engineering (INCOSE), Systems Engineering Handbook V4, 2015, http://www.incose.org/ProductsPublications/sehandbook
- Unified Profile for DoDAF and MODAF (UPDM), 2.1, August 2013, http://www.omg.org/spec/UPDM
- Ontology Definition Metamodel (ODM), 1.1, September 2014, http://www.omg.org/spec/ODM

4. Terms and Definitions

No new terms and definitions have been required to create this specification. All terms are available in the normative references or bibliographic citations for detailed explanation. The modeling concepts specified in this standard e.g., MetaModel Elements, Viewpoints, Aspects, View Specifications, etc. are defined in the appropriate section for that concept. Additional terms are defined in Appendix C: Enterprise Architecture Guide (EAG).

5. Acronyms

For the purposes of this specification, the following List of acronyms used in this specification.

Table 5:1 - Description of acronyms used in this specification

AcV-*[2]	Acquisition View
AD	Architecture Description
AV-*	All View
BMM	Business Motivation Model
BPMN	Business Process Modeling Notation
C4ISR	Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance
CaT	Capability Team
COI	Communities of Interest
CV-*	Capability View
DIV-*	Data and Information Views
DLOD	Defence Lines of Development
DM2	DoDAF Meta Model
DMM	Domain Meta Model
DNDAF	Department National Defence and Canadian Forces (DND/ CF) Architecture Framework
DoD	United States Department of Defense
DoDAF	Department of Defense Architecture Framework
DOTMLP	Doctrine, Organization, Training, Material, Leadership, Personnel, Facilities
EIE	Enterprise Information Environment
IDEAS	International Defense Enterprise Architecture Specification for Exchange
IDEF	Integrated DEFinition Methods
INCOSE	International Council Of Systems Engineering
JCIDS	Joint Capabilities Integration and Development System
MISIG	Model Interchange Special Interest Group
MOD	United Kingdom Ministry of Defence
MODAF	Ministry of Defence Architecture Framework
MODEM	MODAF Ontological Data Exchange Mechanism
NAF	NATO Architecture Framework
OASIS	Organization for the Advancement of Structured Information Standards
OSLC	**Open Services for Lifecycle Collaboration**
OV-*	Operational View
PES	DoDAF Physical Exchange Specification
POC	Proof of Concept
PV-*	Project View
RDF	Resource Description Framework
SoaML	Service orientated architecture Modeling Language

[2] * denotes a wildcard

SoS	System of Systems
SOV-*	Service Oriented View
StdV-*	Standards View in DoDAF 2.02 compare TV-* in UAF
STV-*	Strategic View
SV-*	System View
SvcV-*	Service View
TEPID OIL	Training, Equipment, Personnel, Information, Concepts and Doctrine, Organisation, Infrastructure, Logistics
TOGAF	The Open Group Architectural Framework©
TPPU	Task, Post, Process, and Use
TV-*	Technical View
UAF	Unified Architecture Framework
UAFML	Unified Architecture Framework Modeling Language
UPDM	Unified Profile for DoDAF/MODAF

6. Additional Information

6.1 Changes to Adopted OMG Specifications

This specification completely replaces Unified Architecture Framework (UAF), version 1.2
https://www.omg.org/spec/UAF/About-UAF/

6.2 Language Architecture

The UAF specification reuses a subset of UML 2.5.1 and SysML 1.6 and provides additional extensions needed to address requirements in the UPDM 3.0 RFP Mandatory Requirements. Those requirements form the basis for this specification. This specification documents the language architecture in terms of UML 2.5.1 and SysML 1.6 and specifies how to implement UAF. This clause explains design principles and how they are applied to define the UAF language architecture.

6.3 Philosophy

The UAF development uses a model-driven approach. A simple description of the work process is:

- A Domain Metamodel (DMM) uses UML Class models to represent individuals, types and tuples that maps the concepts defined in DoDAF, MODEM, NAF, and other frameworks.
- The aligned and renamed viewpoints from the various frameworks provide a common generic name for each viewpoint. It should be noted that the term viewpoint is in the context of ISO 42010 where a viewpoint is the specification of a view. The UAF viewpoints are mapped to the corresponding viewpoint in the relevant contributing framework. It is the viewpoints described in the DMM that provides the basis for the Unified Architecture Framework (UAF).
- The UAF provides an abstraction layer that separates the underlying UAF metamodel from the presentation layer. The results of this mapping are given in Appendix A (see document dtc/2021-11-07) and an overview of the viewpoints in a grid format are given in this document.
- The intent of the UAF is to provide a Domain MetaModel usable by non-UML/SysML tool vendors who may wish to implement the UAF within their own tool and metalanguage.
- The Unified Architecture Framework Modeling Language (UAFML) is the standard implementation of the UAF DMM. It was created by mapping the UAF concepts and relationships to corresponding stereotypes in the UAFML Profile.
- The UAFML analysis and refactoring reflects language architecture, tool implementation, and reuse considerations.
- The specification is generated from the UML model used to describe the UAF DMM and UAFML. This approach allows the team to concentrate on architecture issues rather than documentation production. The UML tool automatically maintains consistency. The UML tool improves maintenance and enables traceability between the UAF and the UAFML where every stereotype is linkable to the UAF element using UML Abstraction relationship.

6.4 Core Principles

The fundamental design principles for UAF DMM are:

- **Requirements-driven**: UAF is intended to satisfy the requirements of the UPDM 3.0 RFP Mandatory Requirements.
- **Influence from donor Frameworks:** The DMM was based upon an aggregation of concepts and relationships from the donor frameworks.
- **IDEAS Ontology driven:** The DMM was based upon a simplified version of the IDEAS ontology, see chapter 8.
- **DMM Notation:** The DMM was expressed using UML class diagram notation.

- **Reusability of UML Metamodel concepts:** The UAF DMM reuses a number of concepts from the UML Metamodel, such as Statemachines, Activities and Interactions. The explicit relationship to these concepts enables the UAF DMM to reuse UML semantics instead of reinventing its own semantics.
- **Reusability of BPMN concepts:** The UAF DMM reuses a number of concepts from BPMN, such as processes. The explicit relationship to these concepts enables the UAF DMM to reuse BPMN semantics instead of reinventing its own semantics.

7. UAF Grid

Due to the complexity of managing the multiple viewpoints with overlapping concerns and metamodels, the standard viewpoints are refactored as described in the donor frameworks into a more manageable format. This decision led to the development of the UAF grid which is described below.

The grid is a way of showing how the various *view specifications* (cells) correspond to *viewpoints (prev. known as domains)* (horizontal rows) and the *aspects (prev. known as model kinds)* (the columns) that describe the view specification. The intent of the grid is not to be complete, but to capture the information that is present in the frameworks that contributes to the UAF, consequently, some gaps are evident.

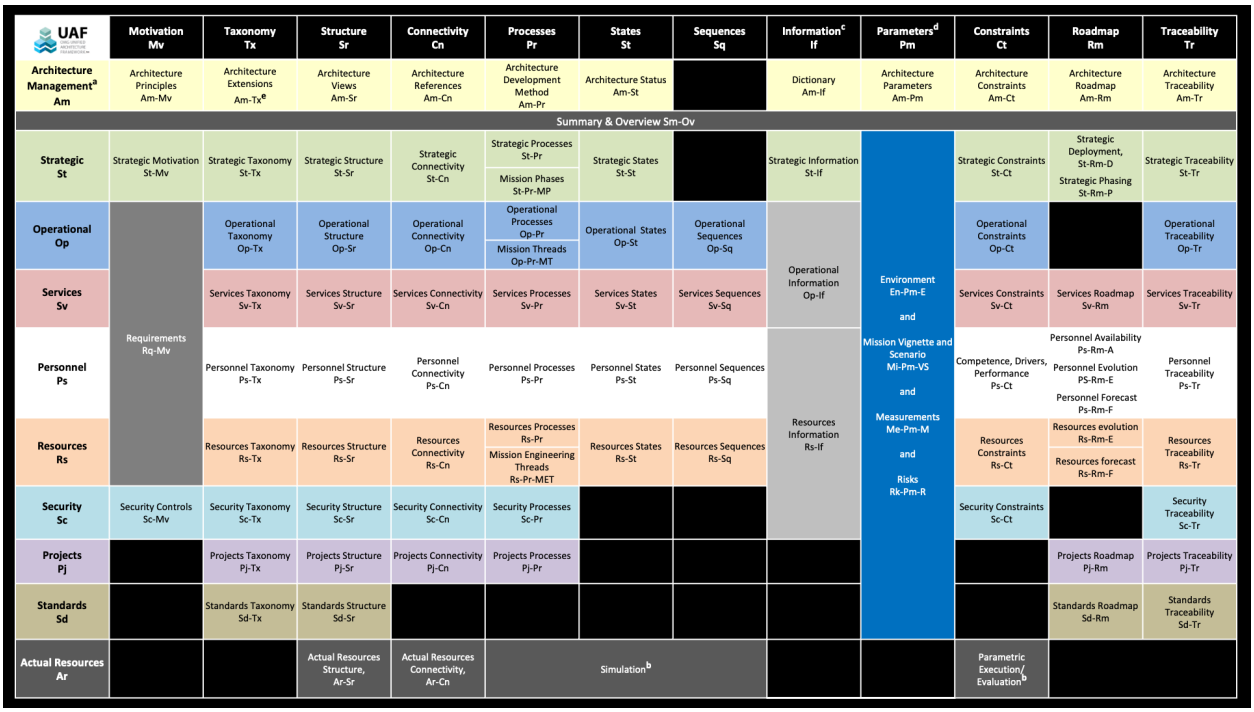

Figure 7:1- UAF Grid

Notes related to suffixes in the grid

a. The view specifications in the Architecture Management viewpoint are architectural artifacts that contribute to the success in defining and developing an architecture.

b. To be able to evaluate architecture behavior and constraints (i.e., non-functional requirements) it is necessary to define actual instances of the architectural elements. The expectation is that tool vendors intending to implement the UAF have capabilities native to their tools to enable behavioral simulation and the evaluation of measures and constraints through parametric diagrams or a proprietary equivalent.

c. The information model is an aspect across the domains and can be defined in any of its forms, i.e., Conceptual, Logical or Physical. The expectation is that most developers of the information model will use the Conceptual or Logical forms of the data model when using an abstract modeling tool.

d. The parameters column captures the measures, environments, and risks across the architecture in the different viewpoints.

e. The Architecture Extensions view specification provides a means to extend the framework to other domains.

The detailed mapping between the view specifications of the UAF shown in the grid and the viewpoints from the donor frameworks is described in dtc/2024-11-08. A definition for each view specification in the grid is described in the following chapters.

7.1 Descriptions of Viewpoints and Aspects

Table 7:1 - Definitions for the Viewpoints

Viewpoint	Acronym	Description
Architecture Management	Am	Identifies the metadata and views required to develop a suitable architecture that is fit for its purpose.
Strategic	St	Capability management process. Describes the capability taxonomy, composition, dependencies and evolution.
Operational	Op	Illustrates the Logical Architecture of the enterprise. Describes the requirements, operational behavior, structure, and exchanges required to support (exhibit) capabilities. Defines all operational elements in an implementation/solution independent manner.
Services	Sv	The Service-Orientated View (SOV) is a description of services needed to directly support the operational domain as described in the Operational View. A service within MODAF is understood in its broadest sense, as a unit of work through which a provider provides a useful result to a consumer. DoDAF: The Service Views within the Services Viewpoint describe the design for service-based solutions to support operational development processes (JCIDS) and Defense Acquisition System or capability development within the Joint Capability Areas.
Personnel	Ps	Defines and explores organizational resource types. Shows the taxonomy of types of organizational resources as well as connections, interaction and growth over time.
Resources	Rs	Captures a solution architecture consisting of resources, e.g., organizational, software, artifacts, capability configurations, and natural resources that implement the operational requirements. Further design of a resource is typically detailed in SysML or UML.
Security	Sc	Security assets and security enclaves. Defines the hierarchy of security assets and asset owners, security constraints (policy, laws, and guidance) and details where they are located (security enclaves).
Projects	Pj	Describes projects and project milestones, how those projects deliver capabilities, the organizations contributing to the projects and dependencies between projects.

Standards	Sd	MODAF: Technical Standards Views are extended from the core DoDAF views to include non-technical standards such as operational doctrine, industry process standards, etc. DoDAF: The Standards Views within the Standards Viewpoint are the set of rules governing the arrangement, interaction, and interdependence of solution parts or elements.
Actual Resources	Ar	The analysis, e.g., evaluation of different alternatives, what-if, trade-offs, V&V on the actual resource configurations. Illustrates the expected or achieved actual resource configurations.

Table 7:2 - Definitions of the Aspects

Aspect	Acronym	Description
Motivation	Mv	Captures motivational elements e.g., challenges, opportunities, and concerns, that pertain to enterprise transformation efforts, and different types of requirements, e.g., operational, services, personnel, resources, or security controls.
Taxonomy	Tx	Presents all the elements as a standalone structure. Presents all the elements as a specialization hierarchy, provides a text definition for each one and references the source of the element
Structure	Sr	Describes the breakdown of structural elements e.g., logical performers, systems, projects, etc. into their smaller parts
Connectivity	Cn	Describes the connections, relationships, and interactions between the different elements.
Processes	Pr	Captures activity based behavior and flows. It describes activities, their Inputs/Outputs, activity actions and flows between them.
States	St	Captures state-based behavior of an element. It is a graphical representation of states of a structural element and how it responds to various events and actions.
Sequences	Sq	Expresses a time ordered examination of the exchanges as a result of a particular scenario. Provides a time-ordered examination of the exchanges between participating elements as a result of a particular scenario.
Information	If	Address the information perspective on operational, service, and resource architectures. Allows analysis of an architecture's

Aspect	Acronym	Description
		information and data definition aspect, without consideration of implementation specific issues.
Constraints	Ct	Details the measurements that set performance requirements constraining capabilities. Also defines the rules governing behavior and structure.
Roadmap	Rm	Addresses how elements in the architecture change over time.
Traceability	Tr	Describes the mapping between elements in the architecture. This can be between different viewpoints within domains as well as between domains. It can also be between structure and behaviors.

7.2 Viewpoint Interrelationships

Although the grid is the primary means of expressing the relationship between the Viewpoints, Aspects and View Specifications, because of its two-dimensional nature it is not adequate to explain the abstract interrelationships that exist between the viewpoints. The following diagram is an indication of the how the viewpoints are interrelated.

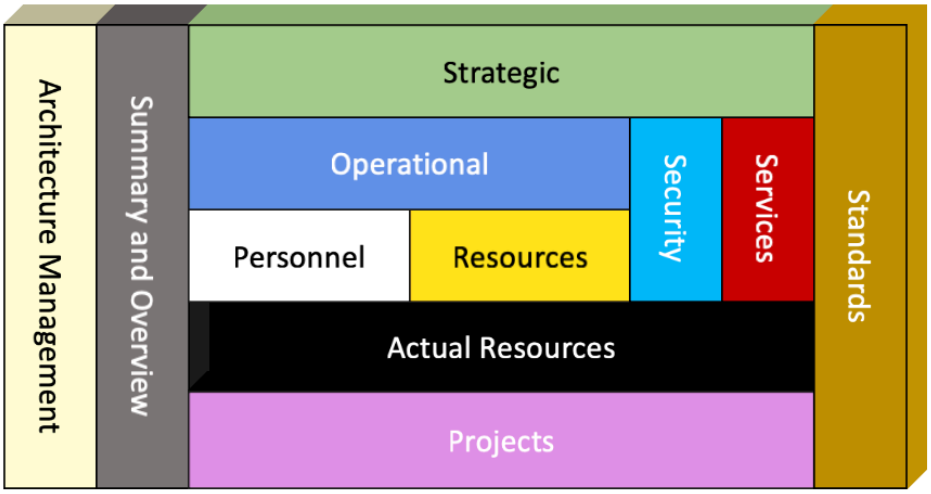

Figure 7:2 – Viewpoint Interrelationships

Where a Viewpoint is shown vertically the intent is to show that the Viewpoint is a cross-cutting concern that goes across the levels of abstraction in the architecture.

Where a Viewpoint is shown horizontally the intent is to show that the Viewpoint exists in a layer of abstraction between the Viewpoints above and below it and there is an interrelationship with the Viewpoints either side of it.

7.3 Domain Metamodel Diagram Legend

Note that the diagrams rely on color to aid the reader in understanding the model. Please refer to the legend below to understand the diagrams.

The following is the legend of element colors used in the DMM and what they denote.

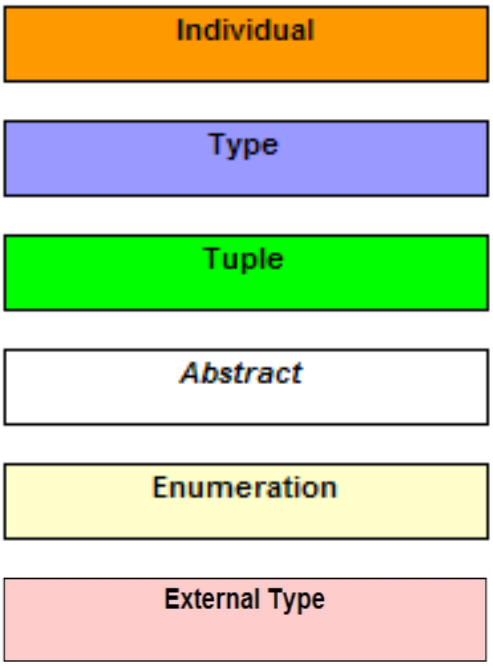

Figure 7:3 - Legend of color codes for element types defined in UAF

The meaning of the element types in the UAF are based upon concepts put forth in the International Defence Enterprise Architecture Specification (IDEAS).

- An Individual denotes a single instance of an element
- A Type denotes a set of Individuals
- A Tuple denotes a relationship that exists between elements
- An Abstract denotes that the element has no direct use but is a means of construction
- An Enumeration is a complete, ordered listing of all the items in a collection
- An External Type is an element that exists outside of the core DMM but is referenceable by elements in the DMM

8. Domain Metamodel (DMM) Diagrams

Note that the diagrams rely on color to aid the reader in understanding the model. Please refer to the legend in the various diagrams to understand the specific definitions.

8.1 View Specifications

This section documents each of the view specifications of UAF.

8.1.1 View Specifications::Architecture Management

Stakeholders: Enterprise Architects, people who want to discover the architecture, Technical Managers.

Concerns: Captures meta-data relevant to the entire architecture

Definition: Provide information pertinent to the entire architecture. Present supporting information rather than architectural models.

8.1.1.1 View Specifications::Architecture Management::Motivation

Contains the diagrams that document the Architecture Management Motivation View Specification.

8.1.1.1.1 View Specifications::Architecture Management::Motivation::Architecture Principles

Stakeholders: Enterprise Architects, Enterprise Systems Engineers, Model Managers, System Architects.

Concerns: alignment of architecture with architecture heuristics, guidelines and principles.

Definition: identifies relevant architectural principles and other guidelines to be used in architecture development and evaluation.

Recommended Implementation: SysML Block Diagram, tabular format

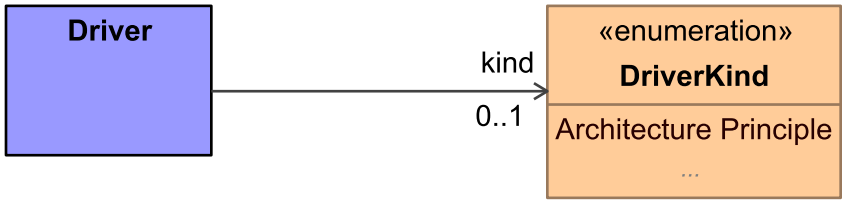

Figure 8:1 - Architecture Principles

Elements

- Driver

8.1.1.2 View Specifications::Architecture Management::Structure

Contains the diagrams that document the Architecture Management Structure View Specification.

8.1.1.2.1 View Specifications::Architecture Management::Structure::Architecture Views

Stakeholders: Enterprise Architects, Model Managers, Modelers, Technical Managers.

Concerns: domains, viewpoints, aspects, model kinds, and view specifications that are used to describe the architecture.

Definition: (i) lists predefined and custom domains, model kinds, viewpoints, aspects and view specifications (ii) and identify the key stakeholders and their perspectives and concerns.

Recommended Implementation: SysML Block Definition Diagram, SysML Package Diagram.

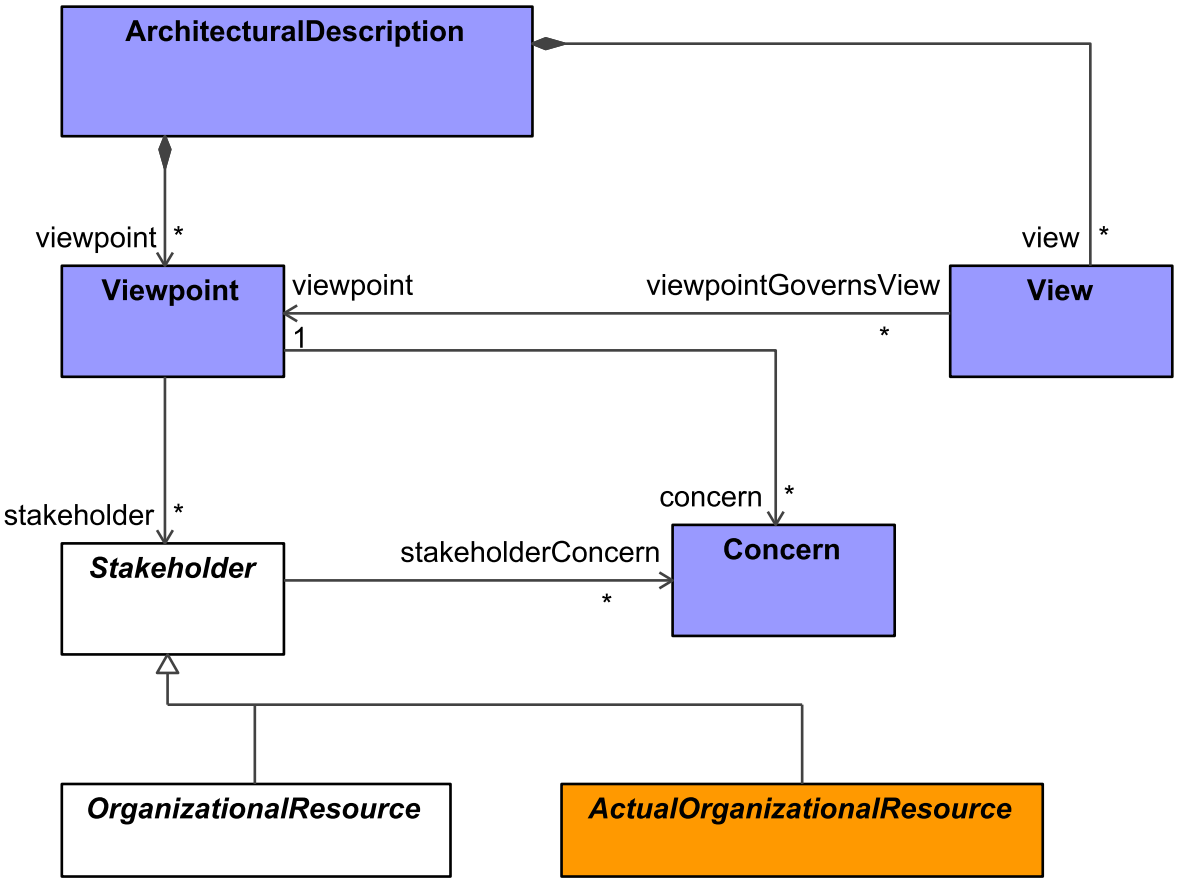

Figure 8:2 - Architecture Views

Elements

- ActualOrganizationalResource
- ArchitecturalDescription
- Concern
- OrganizationalResource
- Stakeholder
- View
- Viewpoint

8.1.1.3 View Specifications::Architecture Management::Connectivity

Contains the diagrams that document the Architecture Management Connectivity View Specification.

8.1.1.3.1 View Specifications::Architecture Management::Connectivity::Architecture References

Stakeholders: Enterprise Architects, people who want to understand relationships to related architectural descriptions, Technical Managers.

Concerns: high-level dependencies between architectural descriptions.

Definition: depicts and analyzes all relevant dependencies between architectural descriptions, e.g., reference architectures, as-is to to-be architectures, enterprise architecture to system architectures.

Recommended Implementation: SysML Block Definition Diagram, SysML Package Diagram, matrix format.

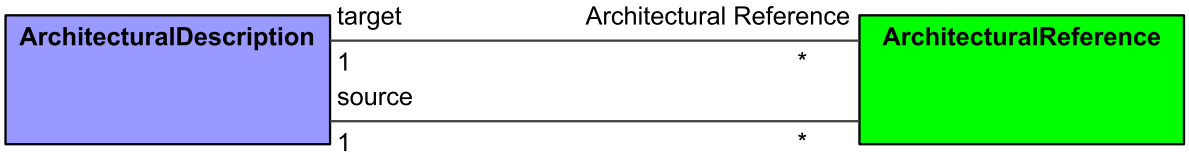

Figure 8:3 - Architecture References

Elements

- ArchitecturalDescription
- ArchitecturalReference

8.1.1.4 View Specifications::Architecture Management::Processes

Contains the diagrams that document the Architecture Management Processes View Specification.

8.1.1.4.1 View Specifications::Architecture Management::Processes::Architecture Development Method

Stakeholders: Enterprise Architects, Model Managers, Modelers, Enterprise Systems Engineers.

Concerns: development sequence of models and views and how they are related to each other.

Definition: defines workflow or process steps used in managing the architecture development.

Recommended Implementation: SysML Activity Diagram, text.

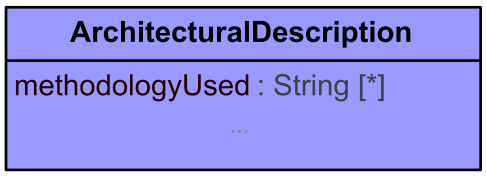

Figure 8:4 - Architecture Development Method

Elements

- ArchitecturalDescription

8.1.1.5 View Specifications::Architecture Management::States

Contains the diagrams that document the Architecture Management States View Specification.

8.1.1.5.1 View Specifications::Architecture Management::States::Architecture Status

Stakeholders: Enterprise Architects, people who want to understand the architecture governance, Technical Managers.

Concerns: architecture status.

Definition: captures version number and approval workflow of the architecture. Recommended Implementation: SysML State Machine Diagram, state table, text.

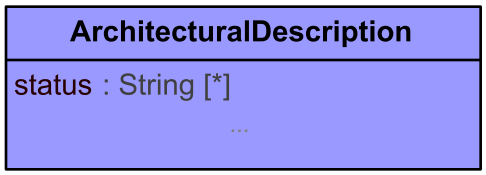

Figure 8:5 - Architecture Status

Elements

- ArchitecturalDescription

8.1.1.6 View Specifications::Architecture Management::Information

Contains the diagrams that document the Architecture Management Information View Specification.

8.1.1.6.1 View Specifications::Architecture Management::Information::Dictionary

Stakeholders: Solution Providers, Systems Engineers, Software Architects, Business Architects.

Concerns: provides a central reference for a given architecture's data and metadata. It enables the set of architecture description to stand alone, with minimal reference to outside resources.

Definition: contains definitions of terms used in the given architecture. It consists of textual definitions in the form of a glossary, their taxonomies, and their metadata (i.e., data about architecture data), including metadata for any custom-tailored views. Architects should use standard terms where possible (i.e., terms from existing, approved dictionaries, glossaries, and lexicons).

Recommended Implementation: text, table format.

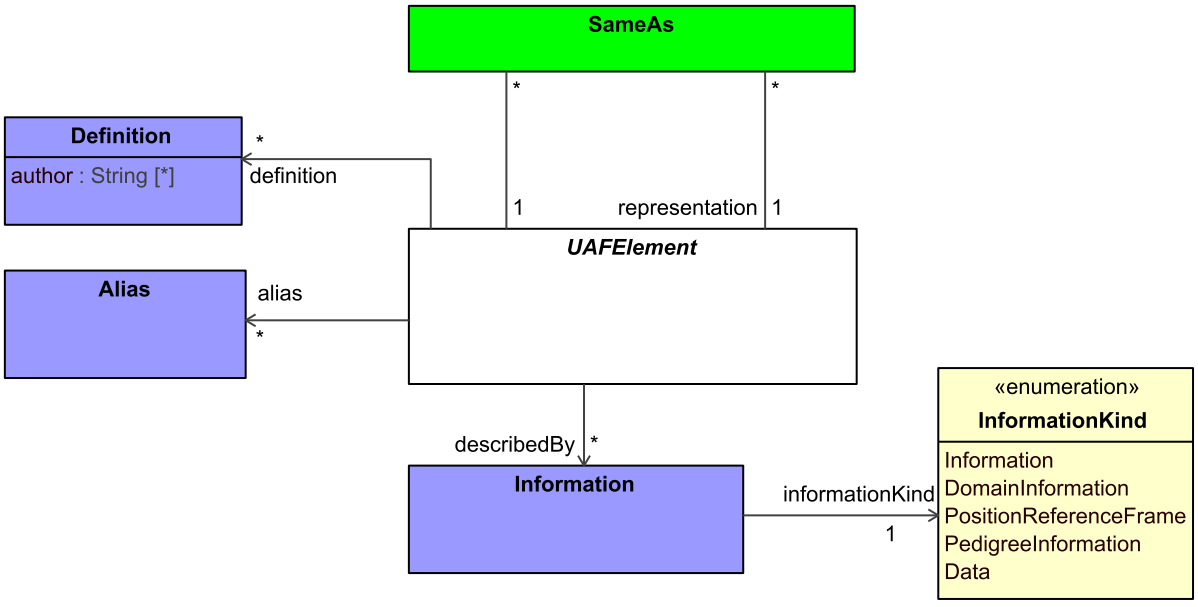

Figure 8:6 - Dictionary

Elements

- Alias
- Definition
- Information

- SameAs
- UAFElement

8.1.1.7 View Specifications::Architecture Management::Parameters

Contains the diagrams that document the Architecture Management Parameters View Specification.

8.1.1.7.1 View Specifications::Architecture Management::Parameters::Architecture Parameters

Stakeholders: Enterprise Architects, Enterprise Systems Engineers, Model Managers, System Architects.

Concerns: architecture parameters.

Definition: depicts and analyzes measures and measurements that are applicable to management of the architecture.

Recommended Implementation: SysML Block Definiton Diagram, tabular format.

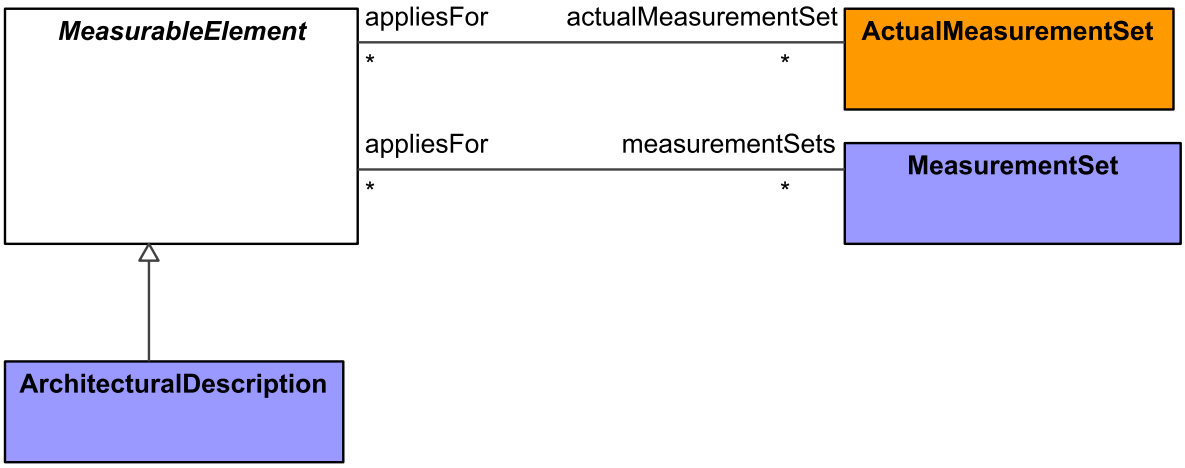

Figure 8:7 - Architecture Parameters

Elements

- ActualMeasurementSet
- ArchitecturalDescription
- MeasurableElement
- MeasurementSet

8.1.1.8 View Specifications::Architecture Management::Constraints

Contains the diagrams that document the Architecture Management Constraints View Specification.

8.1.1.8.1 View Specifications::Architecture Management::Constraints::Architecture Constraints

Stakeholders: Enterprise Architects, people who want to understand the architecture constraints, Technical Managers.

Concerns: architecture assumptions and constraints.

Definition: depicts and analyzes assumptions, constraints, rules, policy and guidance that are applicable to aspects of the architecture.

Recommended Implementation: SysML Package Diagram, tabular format.

Figure 8:8 - Architecture Constraints

Elements

- ArchitecturalDescription

8.1.1.9 View Specifications::Architecture Management::Roadmap

Contains the diagrams that document the Architecture Management Roadmap View Specification.

8.1.1.9.1 View Specifications::Architecture Management::Roadmap::Architecture Roadmap

Stakeholders: Enterprise Architects, people who want to understand the architecture development plan, Technical Managers.

Concerns: architecture release schedule.

Definition: captures project timeline for the architecture. Recommended Implementation: timeline, text.

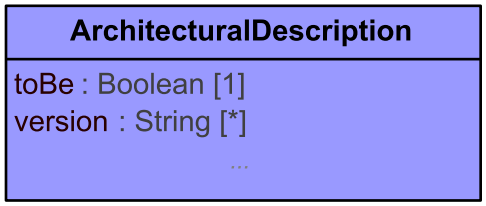

Figure 8:9 - Architecture Roadmap

Elements

- ArchitecturalDescription

8.1.1.10 View Specifications::Architecture Management::Traceability

Contains the diagrams that document the Architecture Management Traceability View Specification.

8.1.1.10.1 View Specifications::Architecture Management::Traceability::Architecture Traceability

Stakeholders: Enterprise Architects, people who want to understand impact of change across the architecture supporting assets, Technical Managers.

Concerns: reuse of architectures.

Definition: shows references to operational, services, and resource architectures, asset libraries, legacy architectures, and external sources, e.g., documents. Recommended Implementation: SysML Block Definition Diagram, SysML Package Diagram, tabular format

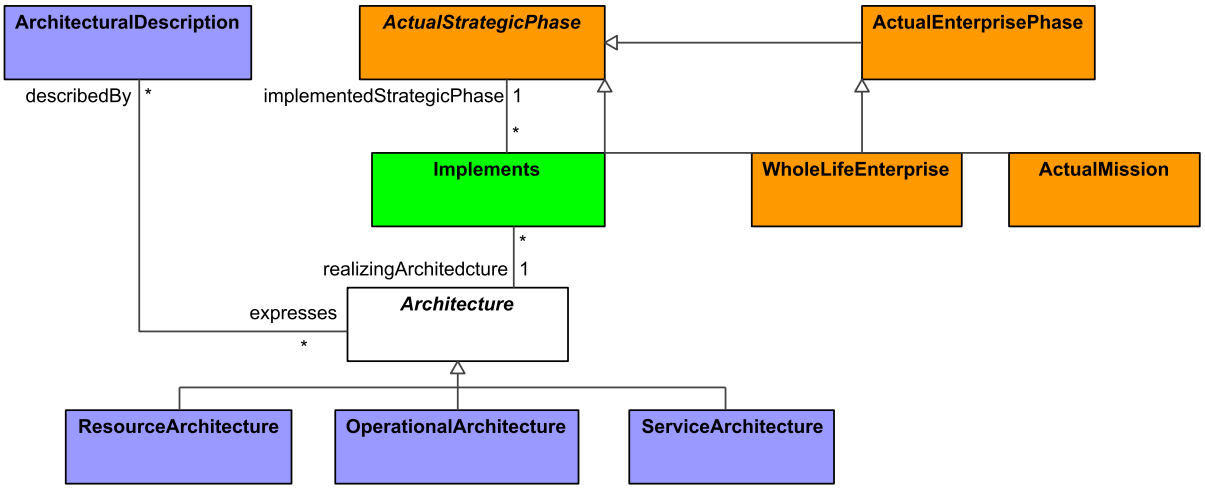

Figure 8:10 - Architecture Traceability

Elements

- ActualEnterprisePhase
- ActualMission
- ActualStrategicPhase
- ArchitecturalDescription
- Architecture
- Implements
- OperationalArchitecture
- ResourceArchitecture
- ServiceArchitecture
- WholeLifeEnterprise

8.1.2 View Specifications::Summary & Overview

Stakeholders: Executives, PMs, Enterprise Architects.

Concerns: executive-level summary information in a consistent form.

Definition: provides executive-level summary information in a consistent form that allows quick reference and comparison between architectural descriptions. Includes assumptions, constraints, and limitations that may affect high-level decisions relating to an architecture-based work programme.

8.1.2.1 View Specifications::Summary & Overview::Summary & Overview

Stakeholders: Decision makers, Solution Providers, Systems Engineers, Software Architects, Business Architects.

Concerns: quick overview of an architecture description and summary of analysis. In the initial phases of architecture development, it serves as a planning guide. Upon completion of an architecture, it provides a summary of findings, and any conducted analysis.

Definition: provides executive-level summary information in a consistent form that allows quick reference and comparison among architectures. The Summary and Overview includes assumptions, constraints, and limitations that may affect high-level decision processes involving the architecture.

Recommended Implementation: text, free form diagram, table format.

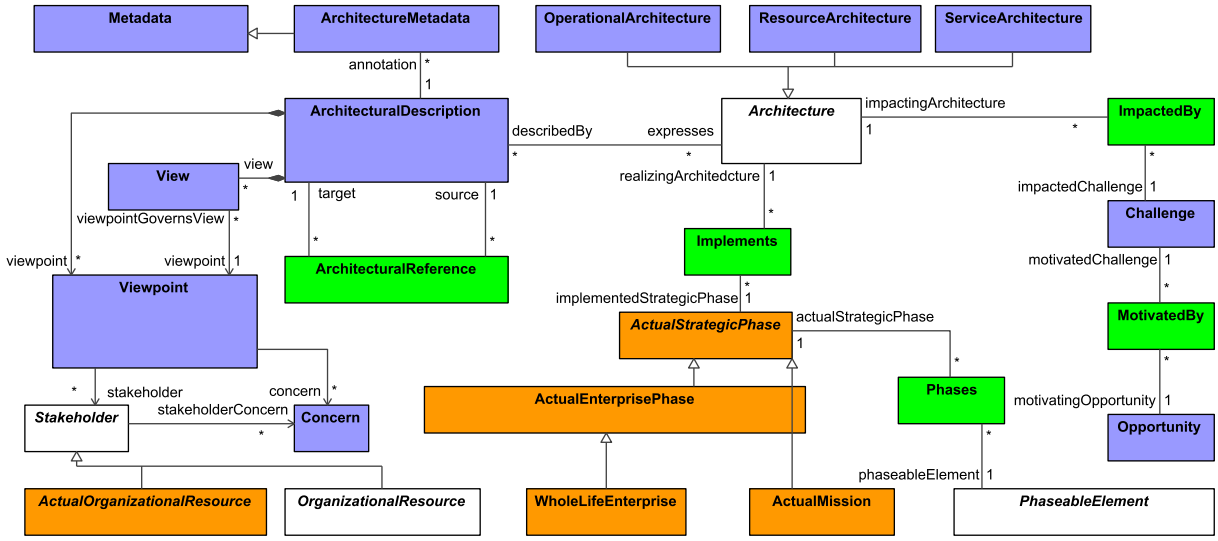

Figure 8:11 - Summary & Overview

Elements

- ActualEnterprisePhase
- ActualMission
- ActualOrganizationalResource
- ActualStrategicPhase
- ArchitecturalDescription
- ArchitecturalReference
- Architecture
- ArchitectureMetadata
- Challenge
- Concern
- ImpactedBy
- Implements
- Metadata
- MotivatedBy
- OperationalArchitecture
- Opportunity
- OrganizationalResource
- PhaseableElement
- Phases
- ResourceArchitecture
- ServiceArchitecture
- Stakeholder
- View
- Viewpoint
- WholeLifeEnterprise

8.1.3 View Specifications::Strategic

Stakeholders: Capability Portfolio Managers.

Concerns: capability management process.

Definition: describe capability taxonomy, composition, dependencies and evolution.

8.1.3.1 View Specifications::Strategic::Motivation
Contains the diagrams that document the Strategic Motivation View Specification.

8.1.3.1.1 View Specifications::Strategic::Motivation::Strategic Motivation
Stakeholders: Enterprise Architects, Portfolio Managers, Enterprise Systems Engineers, Program Managers.

Concerns: architecture drivers, challenges, opportunities, capabilities that address opportunities, phases and architectures that address challenges.

Definition: identifies and defines the drivers, challenges, and opportunities that are applicable to the architecture. defines the desired outcomes, goals and objectives that are motivated by the drivers, and the opportunities that enable the goals and objectives.

Recommended Implementation: SysML Block Definition Diagram, SysML Package Diagram, tabular format.

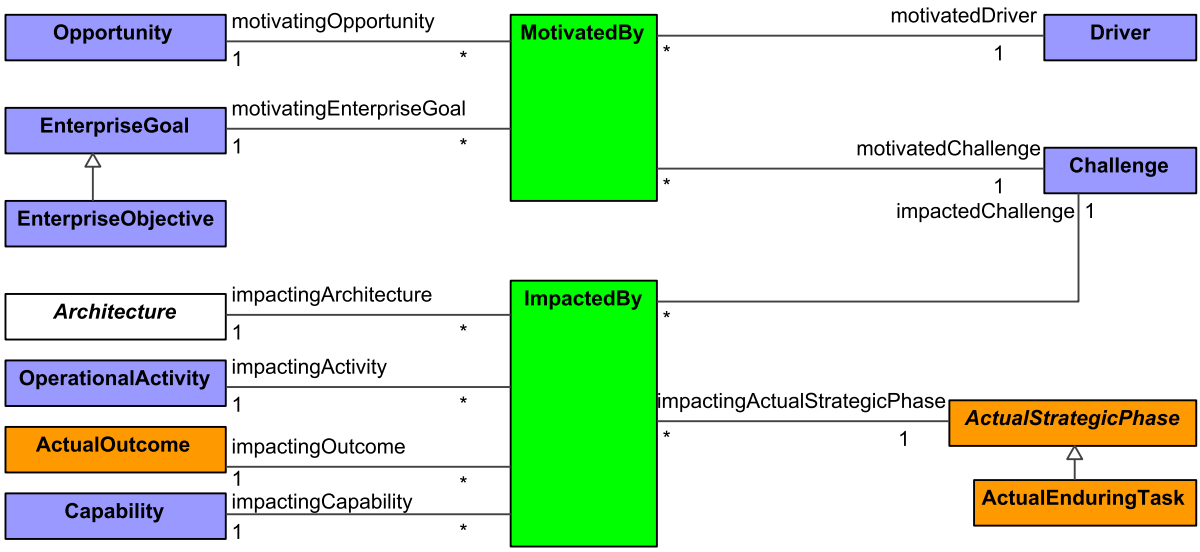

Figure 8:12 - Strategic Motivation

Elements

- ActualEnduringTask
- ActualOutcome
- ActualStrategicPhase
- Architecture
- Capability
- Challenge
- Driver
- EnterpriseGoal
- EnterpriseObjective
- ImpactedBy
- MotivatedBy
- OperationalActivity
- Opportunity

8.1.3.2 View Specifications::Strategic::Taxonomy
Contains the diagrams that document the Strategic Taxonomy View Specification.

8.1.3.2.1 View Specifications::Strategic::Taxonomy::Strategic Taxonomy
Stakeholders: PMs, Enterprise Architects, Executives.

Concerns: capability needs.

Definition: shows the taxonomy of capabilities.

Recommended Implementation: SysML Block Definition Diagram.

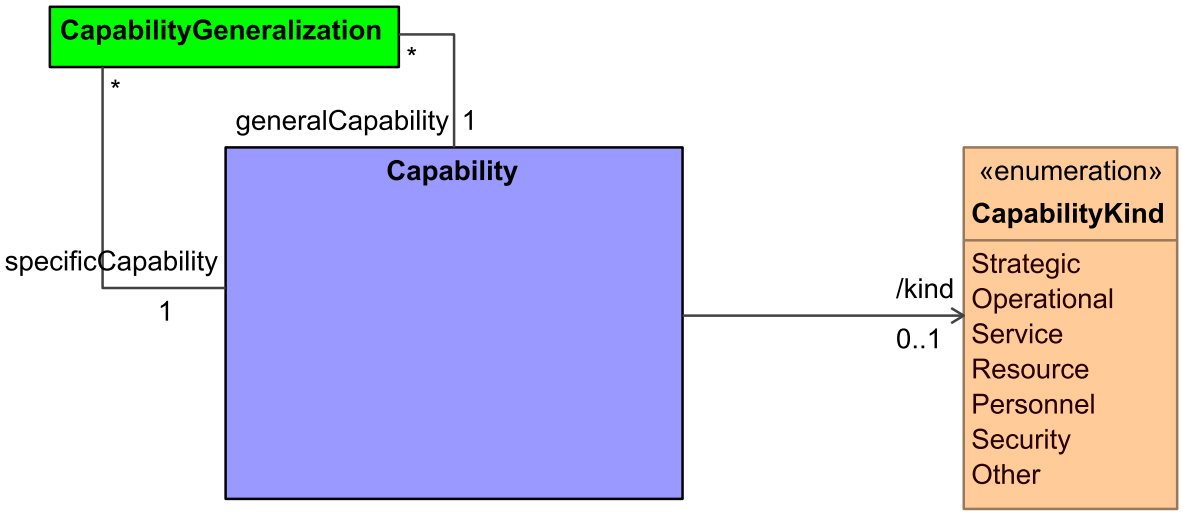

Figure 8:13 - Strategic Taxonomy

Elements

- Capability
- CapabilityGeneralization

8.1.3.3 View Specifications::Strategic::Structure

Contains the diagrams that document the Strategic Structure View Specification.

8.1.3.3.1 View Specifications::Strategic::Structure::Strategic Structure

Stakeholders: PMs, Enterprise Architects, Executives.

Concerns: capability needs.

Definition: shows the relationship between EnterprisePhases and the Capabilities that are intended to be developed during the enterprise phases, and the organizations involved in the enterprise.

Recommended Implementation: SysML Block Definition Diagram.

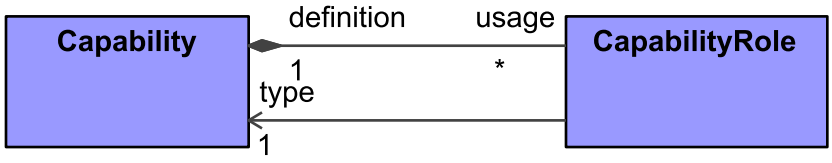

Figure 8:14 - Strategic Structure

Elements

- Capability
- CapabilityRole

8.1.3.4 View Specifications::Strategic::Connectivity
Contains the diagrams that document the Strategic Connectivity View Specification.

8.1.3.4.1 View Specifications::Strategic::Connectivity::Strategic Connectivity
Stakeholders: PMs, Executives, Enterprise Architects.

Concerns: capability dependencies.

Definition: describes the dependencies between planned capabilities.

Recommended Implementation: SysML Block Definition Diagram. SysML Internal Block Diagram.

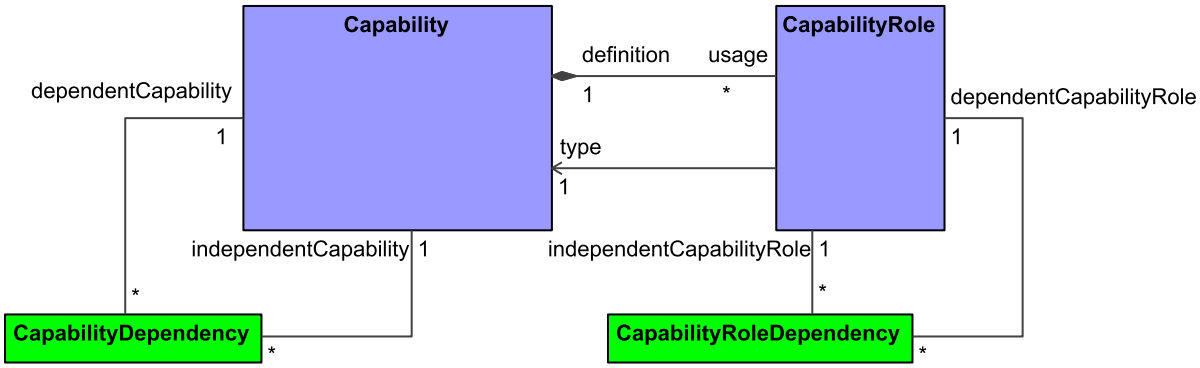

Figure 8:15 - Strategic Connectivity

Elements

- Capability
- CapabilityDependency
- CapabilityRole
- CapabilityRoleDependency

8.1.3.5 View Specifications::Strategic::Processes
Contains the diagrams that document the Strategic Processes View Specification.

8.1.3.5.1 View Specifications::Strategic::Processes::Strategic Processes
Stakeholders: Program/Project Managers, Portfolio Managers, Enterprise Architects, Executives.

Concerns: capability phasing.

Definition: shows the relationship between strategic phases and the Capabilities that are intended to be developed during the strategic phases, and the actual organizations involved.

Recommended Implementation: SysML Block Definition Diagram.

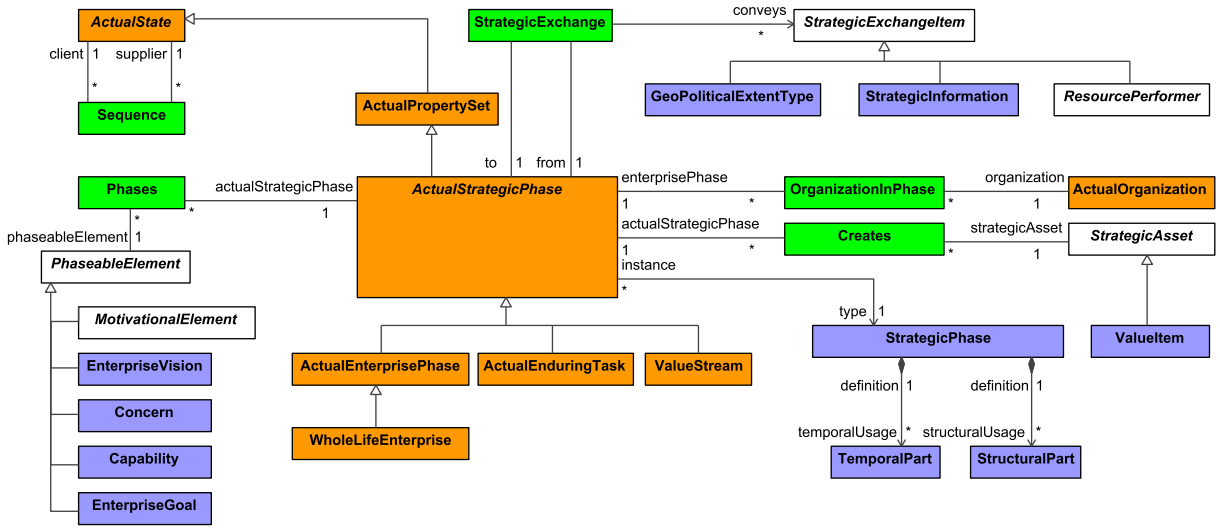

Figure 8:16 - Strategic Processes

Elements

- ActualEnduringTask
- ActualEnterprisePhase
- ActualOrganization
- ActualPropertySet
- ActualState
- ActualStrategicPhase
- Capability
- Concern
- Creates
- EnterpriseGoal
- EnterpriseVision
- GeoPoliticalExtentType
- MotivationalElement
- OrganizationInPhase
- PhaseableElement
- Phases
- ResourcePerformer
- Sequence
- StrategicAsset
- StrategicExchange
- StrategicExchangeItem
- StrategicInformation
- StrategicPhase
- StructuralPart
- TemporalPart
- ValueItem
- ValueStream
- WholeLifeEnterprise

8.1.3.5.2 View Specifications::Strategic::Processes::Strategic Processes: Mission Phases

Stakeholders: Program/Project Managers, Portfolio Managers, Enterprise Architects, Executives.

Concerns: capability phasing.

Definition: shows the relationship between strategic phases and the Capabilities that are intended to be developed during the strategic phases, and the actual organizations involved.

Recommended Implementation: SysML Block Definition Diagram.

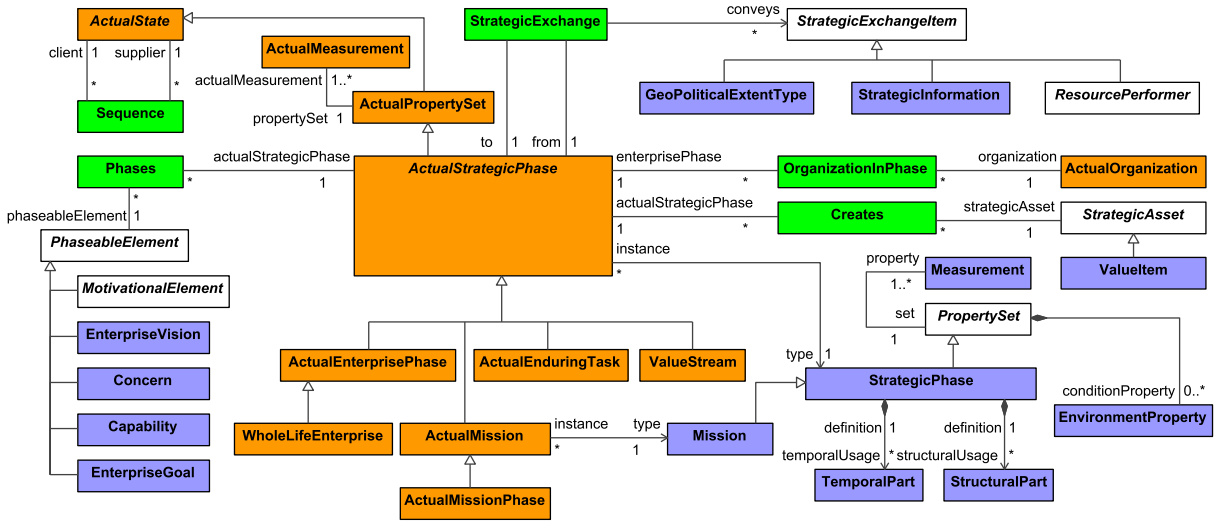

Figure 8:17 - Strategic Processes: Mission Phases

Elements

- ActualEnduringTask
- ActualEnterprisePhase
- ActualMeasurement
- ActualMission
- ActualMissionPhase
- ActualOrganization
- ActualPropertySet
- ActualState
- ActualStrategicPhase
- Capability
- Concern
- Creates
- EnterpriseGoal
- EnterpriseVision
- EnvironmentProperty
- GeoPoliticalExtentType
- Measurement
- Mission
- MotivationalElement
- OrganizationInPhase
- PhaseableElement
- Phases
- PropertySet
- ResourcePerformer
- Sequence
- StrategicAsset
- StrategicExchange
- StrategicExchangeItem
- StrategicInformation

- StrategicPhase
- StructuralPart
- TemporalPart
- ValueItem
- ValueStream
- WholeLifeEnterprise

8.1.3.6 View Specifications::Strategic::States

Contains the diagrams that document the Strategic States View Specification.

8.1.3.6.1 View Specifications::Strategic::States::Strategic States

Stakeholders: PMs, Enterprise Architects.

Concerns: effects that the implementation(s) of capabilities are expected to deliver.

Definition: captures the relationships between capability(ies) and desired effect(s) that implementation(s) of capability(ies) should achieve.

Recommended Implementation: SysML Block Definition Diagram.

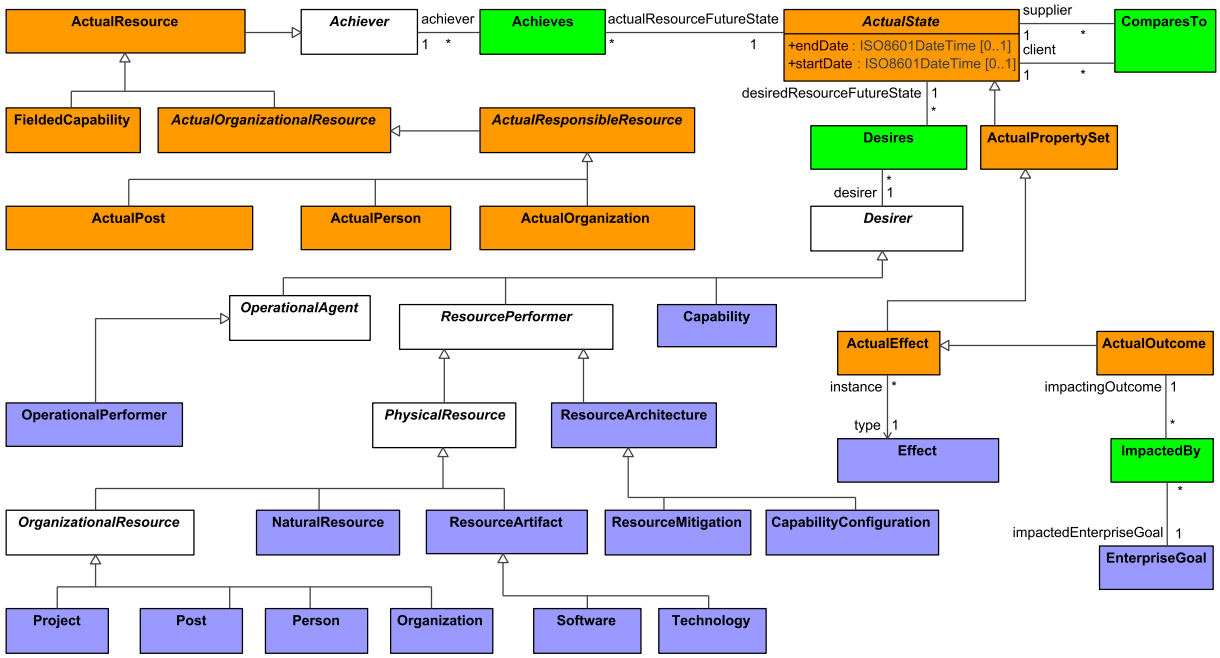

Figure 8:18 - Strategic States

Elements

- Achiever
- Achieves
- ActualEffect
- ActualOrganization
- ActualOrganizationalResource
- ActualOutcome
- ActualPerson
- ActualPost
- ActualPropertySet
- ActualResource
- ActualResponsibleResource

- ActualState
- Capability
- CapabilityConfiguration
- ComparesTo
- Desirer
- Desires
- Effect
- EnterpriseGoal
- FieldedCapability
- ImpactedBy
- NaturalResource
- OperationalAgent
- OperationalPerformer
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- Project
- ResourceArchitecture
- ResourceArtifact
- ResourceMitigation
- ResourcePerformer
- Software
- Technology

8.1.3.7 View Specifications::Strategic::Information

Contains the diagrams that document the Strategic Information View Specification.

8.1.3.7.1 View Specifications::Strategic::Information::Strategic Information

Stakeholders: Enterprise Architects, Portfolio Managers, Enterprise Systems Engineers, Business Managers.

Concerns: information that can be considered to be an enterprise strategic asset that can influence achievement of enterprise goals.

Definition: identifies and defines strategic information elements and their relationships that are applicable to the architecture.

Recommended Implementation: SysML Block Definition Diagram, tabular format.

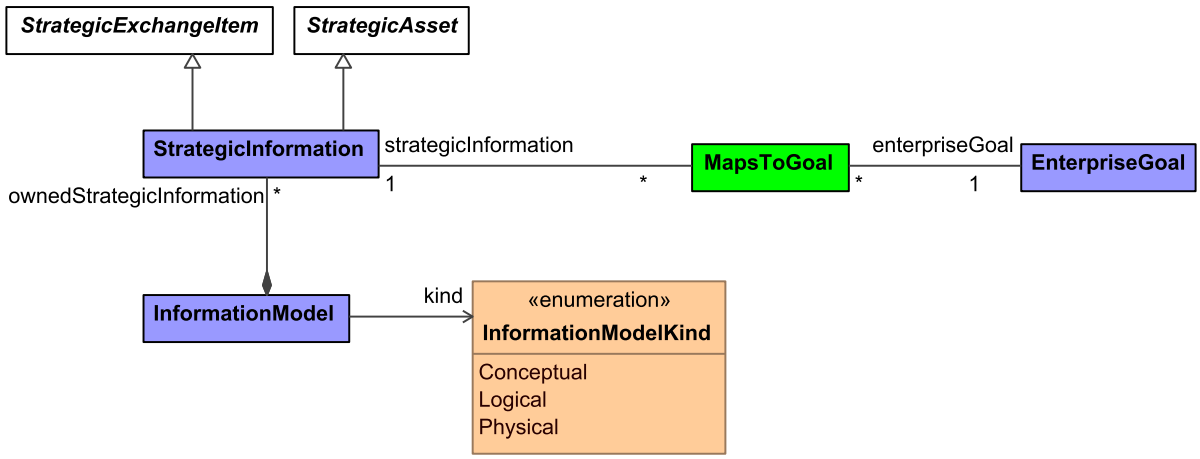

Figure 8:19 - Strategic Information

Elements

- EnterpriseGoal
- InformationModel
- MapsToGoal
- StrategicAsset
- StrategicExchangeItem
- StrategicInformation

8.1.3.8 View Specifications::Strategic::Constraints

Contains the diagrams that document the Strategic Constraints View Specification.

8.1.3.8.1 View Specifications::Strategic::Constraints::Strategic Constraints

Stakeholders: PMs, Enterprise Architects. Concerns: capability constraints. Definition: details the measurements that set performance requirements constraining capabilities. Recommended Implementation: tabular format, SysML Block Definition Diagram.

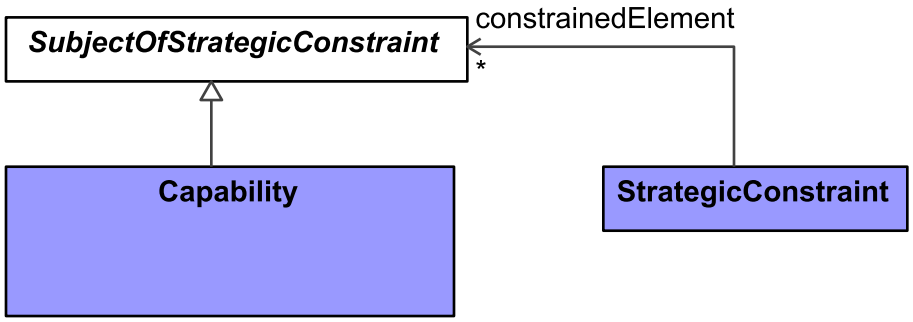

Figure 8:20 - Strategic Constraints

Elements

- Capability
- StrategicConstraint
- SubjectOfStrategicConstraint

8.1.3.9 View Specifications::Strategic::Roadmap

Contains the diagrams that document the Strategic Roadmap View Specification.

8.1.3.9.1 View Specifications::Strategic::Roadmap::Strategic Roadmap: Deployment

Stakeholders: PMs, Executives, Enterprise Architects.

Concerns: capability deployment to organizations over time.

Definition: addresses the deployment of capability(ies) to actual organizations over time.

Recommended Implementation: timeline, tabular format, SysML Block Definition Diagram.

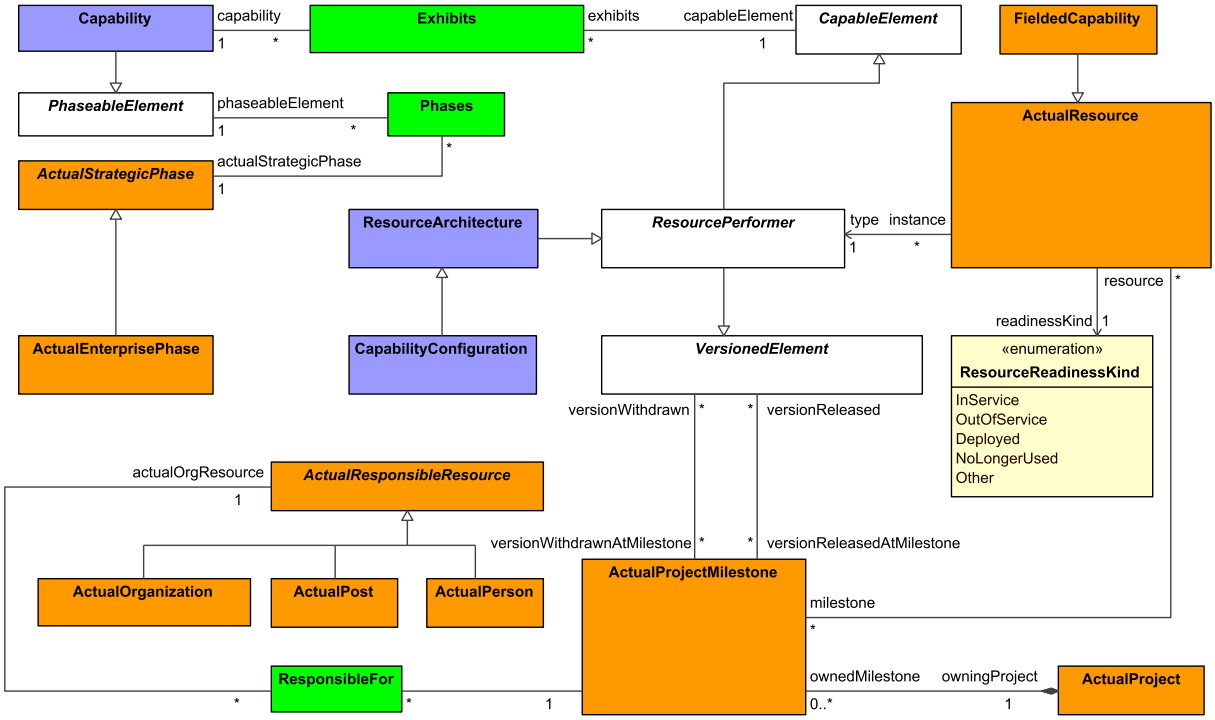

Figure 8:21 - Strategic Roadmap: Deployment

Elements

- ActualEnterprisePhase
- ActualOrganization
- ActualPerson
- ActualPost
- ActualProject
- ActualProjectMilestone
- ActualResource
- ActualResponsibleResource
- ActualStrategicPhase
- Capability
- CapabilityConfiguration
- CapableElement
- Exhibits
- FieldedCapability
- PhaseableElement
- Phases
- ResourceArchitecture
- ResourcePerformer
- ResponsibleFor

- VersionedElement

8.1.3.9.2 View Specifications::Strategic::Roadmap::Strategic Roadmap: Phasing

Stakeholders: PMs, Executives, Enterprise Architects.

Concerns: capability(ies) achievement over time.

Definition: the planned achievement of capability(ies) at different points in time or during specific periods of time.

Recommended Implementation: timeline, tabular format, SysML Block Definition Diagram.

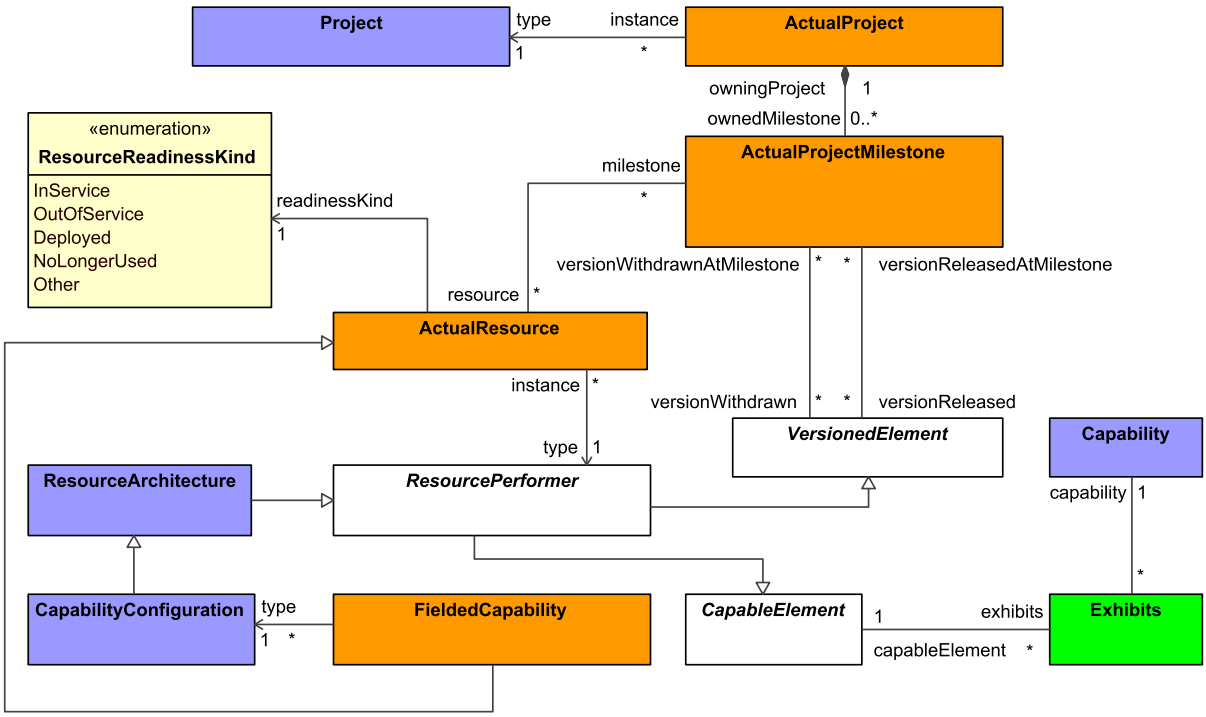

Figure 8:22 - Strategic Roadmap: Phasing

Elements

- ActualProject
- ActualProjectMilestone
- ActualResource
- Capability
- CapabilityConfiguration
- CapableElement
- Exhibits
- FieldedCapability
- Project
- ResourceArchitecture
- ResourcePerformer
- VersionedElement

8.1.3.10 View Specifications::Strategic::Traceability

Contains the diagrams that document the Strategic Traceability View Specification.

8.1.3.10.1 View Specifications::Strategic::Traceability::Strategic Traceability

Stakeholders: PMs, Enterprise Architects, Business Architects.

Concerns: traceability between capabilities and operational activities.

Definition: describes the mapping between the capabilities required by an Enterprise and the supporting operational activities.

Recommended Implementation: matrix format, SysML Block Definition Diagram.

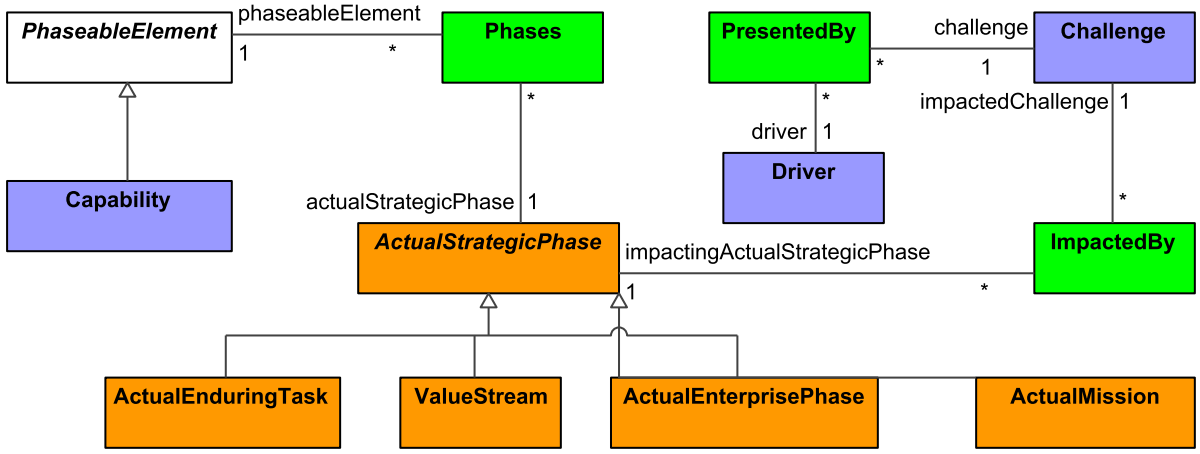

Figure 8:23 - Strategic Traceability

Elements

- ActualEnduringTask
- ActualEnterprisePhase
- ActualMission
- ActualStrategicPhase
- Capability
- Challenge
- Driver
- ImpactedBy
- PhaseableElement
- Phases
- PresentedBy
- ValueStream

8.1.4 View Specifications::Operational

Stakeholders: Business Architects, Executives.

Concerns: illustrate the Logical Architecture of the enterprise.

Definition: describe the requirements, operational behavior, structure, and exchanges required to support (exhibit) capabilities. Defines all operational elements in an implementation/solution independent manner.

8.1.4.1 View Specifications::Operational::Taxonomy

Contains the diagrams that document the Operational Taxonomy View Specification.

8.1.4.1.1 View Specifications::Operational::Taxonomy::Operational Taxonomy

Stakeholders: Business Architects, Systems Engineers, Enterprise Architects, Owners responsible for Operational Agents.

Concerns: OperationalAgent types.

Definition: shows the taxonomy of types of OperationalAgents.

Recommended Implementation: SysML Block Definition Diagram, SysML Internal Block Diagram.

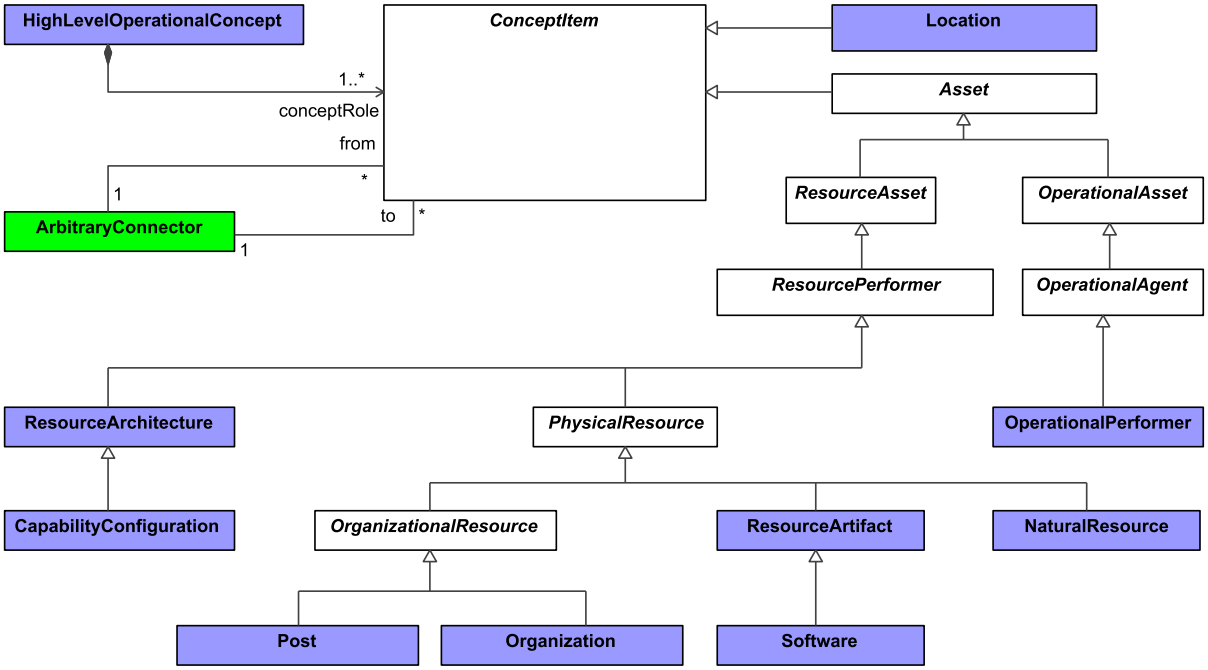

Figure 8:24 - Operational Taxonomy

Elements

- ArbitraryConnector
- Asset
- CapabilityConfiguration
- ConceptItem
- HighLevelOperationalConcept
- Location
- NaturalResource
- OperationalAgent
- OperationalAsset
- OperationalPerformer
- Organization
- OrganizationalResource
- PhysicalResource
- Post
- ResourceArchitecture
- ResourceArtifact
- ResourceAsset
- ResourcePerformer
- Software

8.1.4.2 View Specifications::Operational::Structure

Contains the diagrams that document the Operational Structure View Specification.

8.1.4.2.1 View Specifications::Operational::Structure::Operational Structure

Stakeholders: Business Architects, Systems Engineers, Enterprise Architects, Owners responsible for Operational Agents.

Concerns: identifies the operational exchange requirements between nodes.

Definition: defines operational architecture and exchange requirements necessary to support a specific set of Capability(ies).

Recommended Implementation: SysML Block Definition Diagram, SysML Internal Block Diagram.

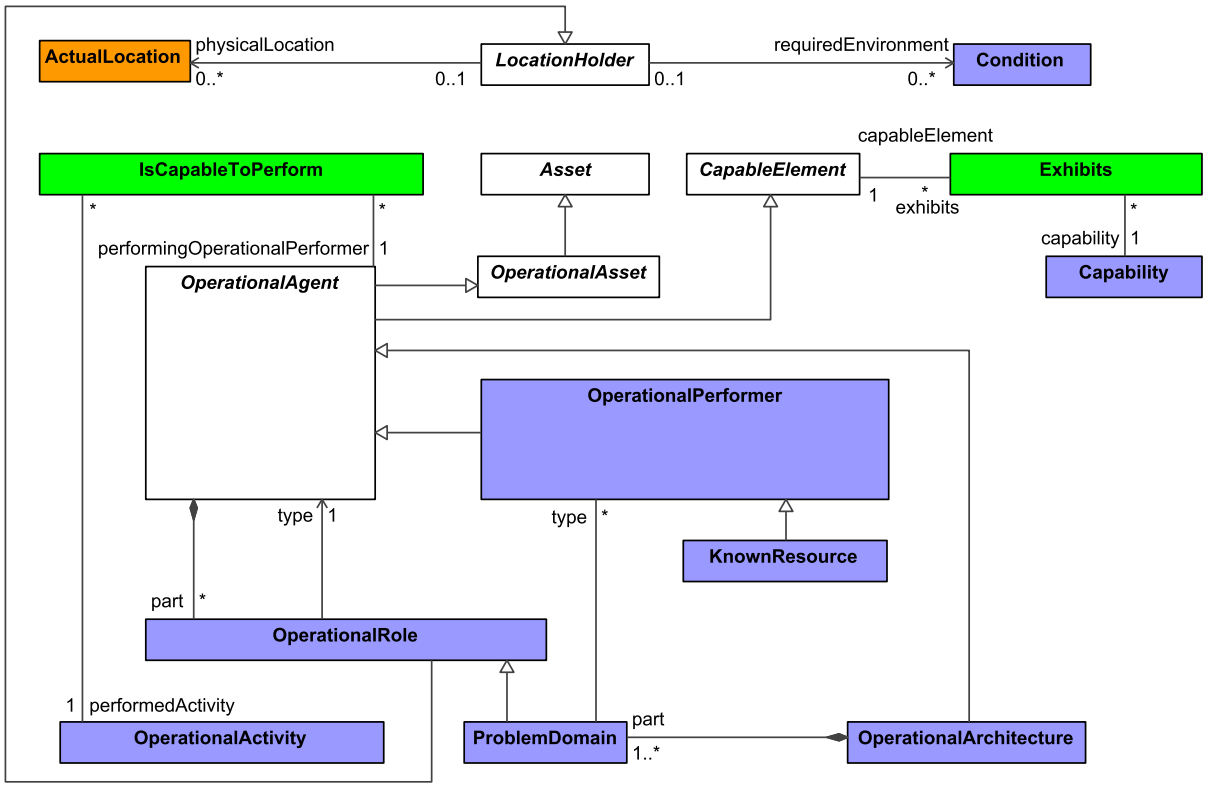

Figure 8:25 - Operational Structure

Elements

- ActualLocation
- Asset
- Capability
- CapableElement
- Condition
- Exhibits
- IsCapableToPerform
- KnownResource
- LocationHolder
- OperationalActivity
- OperationalAgent
- OperationalArchitecture
- OperationalAsset
- OperationalPerformer
- OperationalRole
- ProblemDomain

8.1.4.3 View Specifications::Operational::Connectivity

Contains the diagrams that document the Operational Connectivity View Specification.

8.1.4.3.1 View Specifications::Operational::Connectivity::Operational Connectivity

Stakeholders: Systems Engineers, Architects, Solution Providers.

Concerns: capture the interfaces between OperationalPerformers.

Definition: summarizes logical exchanges between OperationalPerformers of information, systems, personnel, energy etc. and the logical activities that produce and consume them. Measurements can optionally be included.

Recommended Implementation: SysML Internal Block Diagram, tabular format.

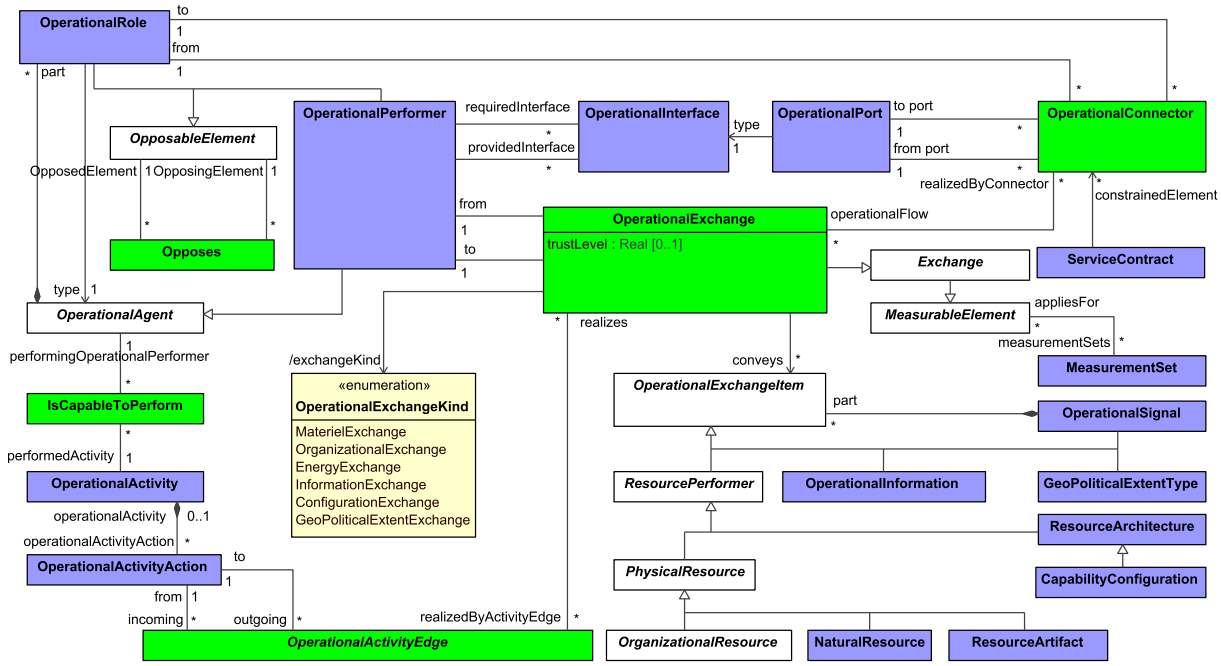

Figure 8:26 - Operational Connectivity

Elements

- CapabilityConfiguration
- Exchange
- GeoPoliticalExtentType
- IsCapableToPerform
- MeasurableElement
- MeasurementSet
- NaturalResource
- OperationalActivity
- OperationalActivityAction
- OperationalActivityEdge
- OperationalAgent
- OperationalConnector
- OperationalExchange
- OperationalExchangeItem
- OperationalInformation
- OperationalInterface
- OperationalPerformer
- OperationalPort

- OperationalRole
- OperationalSignal
- OpposableElement
- Opposes
- OrganizationalResource
- PhysicalResource
- ResourceArchitecture
- ResourceArtifact
- ResourcePerformer
- ServiceContract

8.1.4.4 View Specifications::Operational::Processes

Contains the diagrams that document the Operational Processes View Specification.

8.1.4.4.1 View Specifications::Operational::Processes::Operational Processes

Stakeholders: Business Architect, Systems Engineers, Enterprise Architects

Concerns: captures activity based behavior and flows.

Definition: describes the activities that are normally conducted in the course of achieving business goals that support a capability. It describes operational activities, their Inputs/Outputs, operational activity actions and flows between them.

Recommended Implementation: SysML Activity Diagram, SysML Block Definition Diagram.

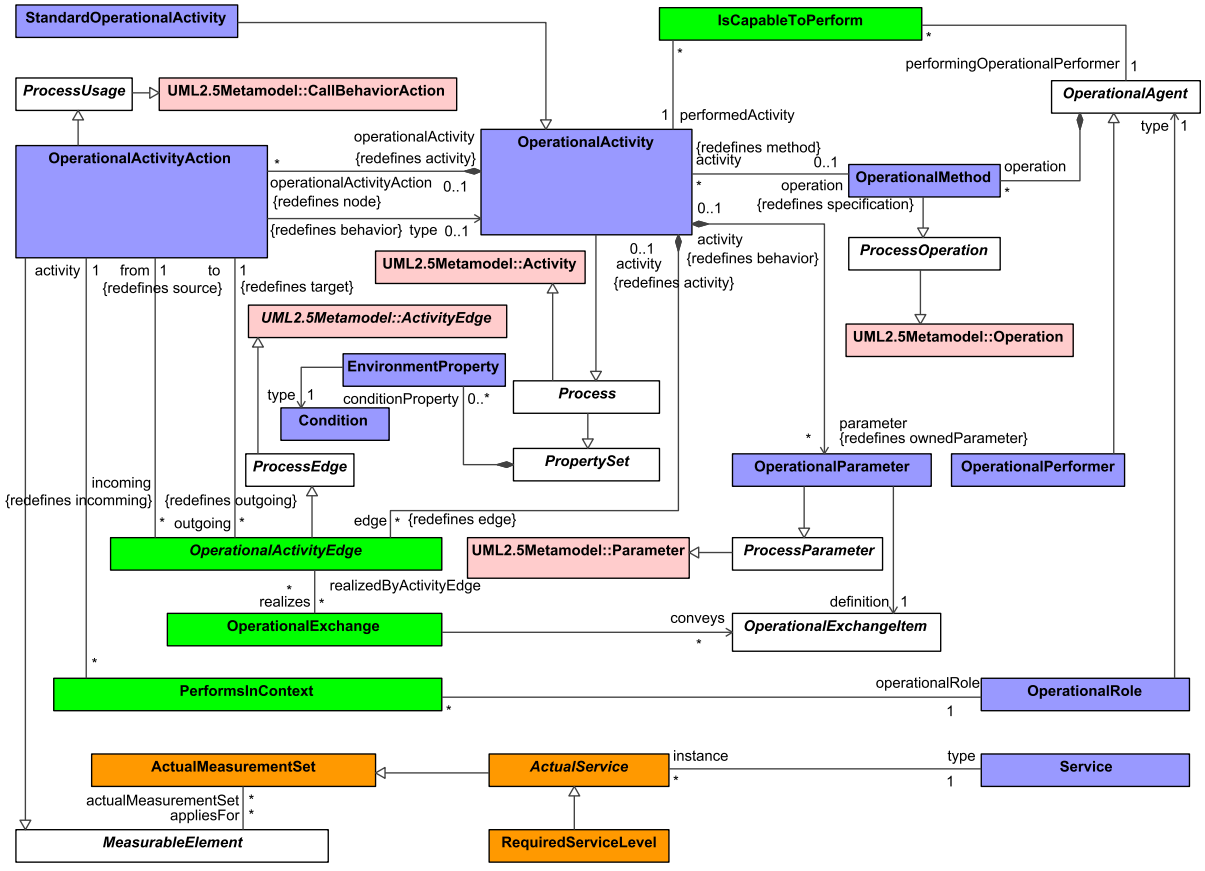

Figure 8:27 - Operational Processes

Elements

- ActualMeasurementSet
- ActualService
- Condition
- EnvironmentProperty
- IsCapableToPerform
- MeasurableElement
- OperationalActivity
- OperationalActivityAction
- OperationalActivityEdge
- OperationalAgent
- OperationalExchange
- OperationalExchangeItem
- OperationalMethod
- OperationalParameter
- OperationalPerformer
- OperationalRole
- PerformsInContext
- Process
- ProcessEdge
- ProcessOperation
- ProcessParameter
- ProcessUsage
- PropertySet
- RequiredServiceLevel
- Service
- StandardOperationalActivity
- UML2.5Metamodel::Activity
- UML2.5Metamodel::ActivityEdge
- UML2.5Metamodel::CallBehaviorAction
- UML2.5Metamodel::Operation
- UML2.5Metamodel::Parameter

8.1.4.4.2 View Specifications::Operational::Processes::Operational Processes BPMN Semantics

Stakeholders: Business Architect, Enterprise Architects.

Concerns: captures activity based behavior and flows using BPMN notation.

Definition: describes the BPMN processes that are normally conducted in the course of achieving business goals that support a capability. It describes operational activities, their Inputs/Outputs, operational activity actions and flows between them using BPMN notation.

Recommended Implementation: BPMN Process Diagram.

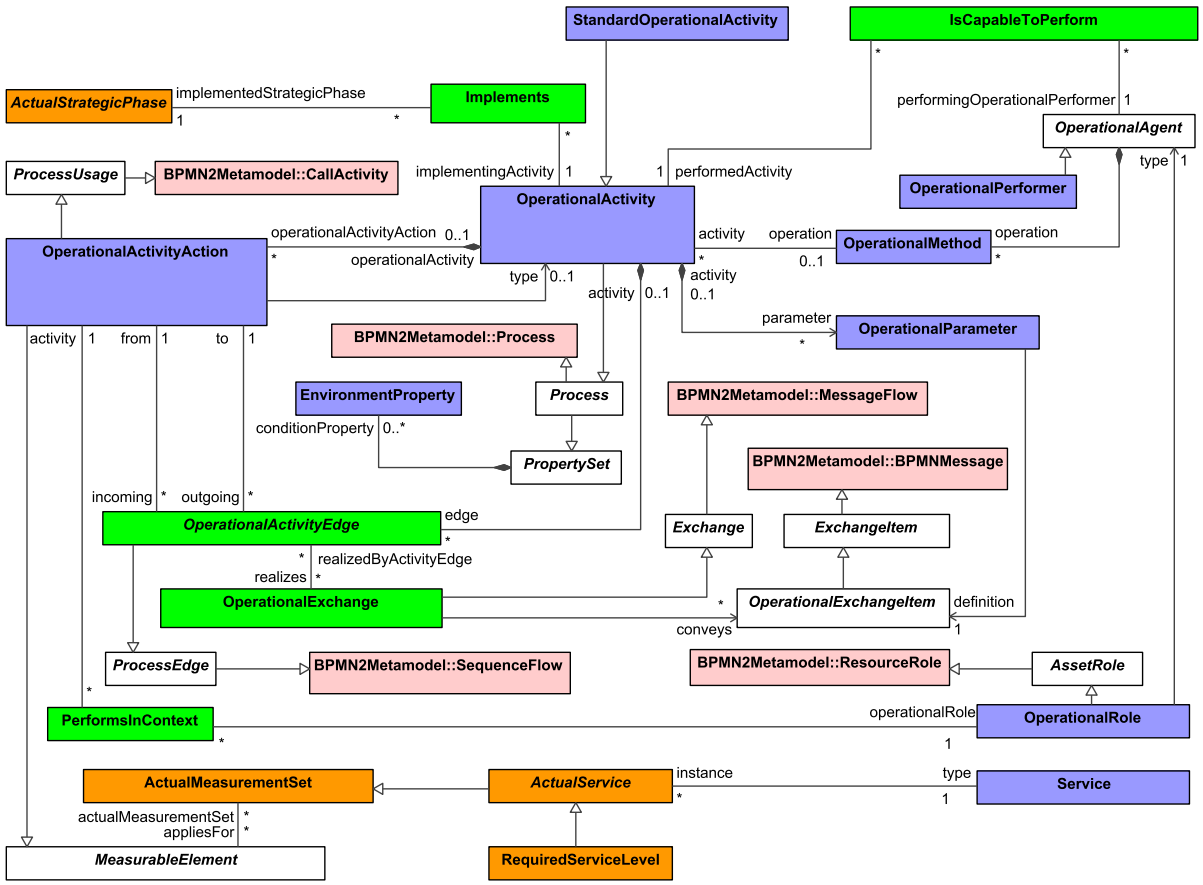

Figure 8:28 - Operational Processes BPMN Semantics

Elements

- ActualMeasurementSet
- ActualService
- ActualStrategicPhase
- AssetRole
- BPMN2Metamodel::BPMNMessage
- BPMN2Metamodel::CallActivity
- BPMN2Metamodel::MessageFlow
- BPMN2Metamodel::Process
- BPMN2Metamodel::ResourceRole
- BPMN2Metamodel::SequenceFlow
- EnvironmentProperty
- Exchange
- ExchangeItem
- Implements
- IsCapableToPerform
- MeasurableElement
- OperationalActivity
- OperationalActivityAction
- OperationalActivityEdge
- OperationalAgent
- OperationalExchange
- OperationalExchangeItem
- OperationalMethod

- OperationalParameter
- OperationalPerformer
- OperationalRole
- PerformsInContext
- Process
- ProcessEdge
- ProcessUsage
- PropertySet
- RequiredServiceLevel
- Service
- StandardOperationalActivity

8.1.4.4.3 View Specifications::Operational::Processes::Operational Processes: Mission Threads

Stakeholders: Business Architect, Systems Engineers, Enterprise Architects

Concerns: captures activity based behavior and flows.

Definition: describes the activities that are normally conducted in the course of achieving business goals that support a capability. It describes operational activities, their Inputs/Outputs, operational activity actions and flows between them.

Recommended Implementation: SysML Activity Diagram, SysML Block Definition Diagram.

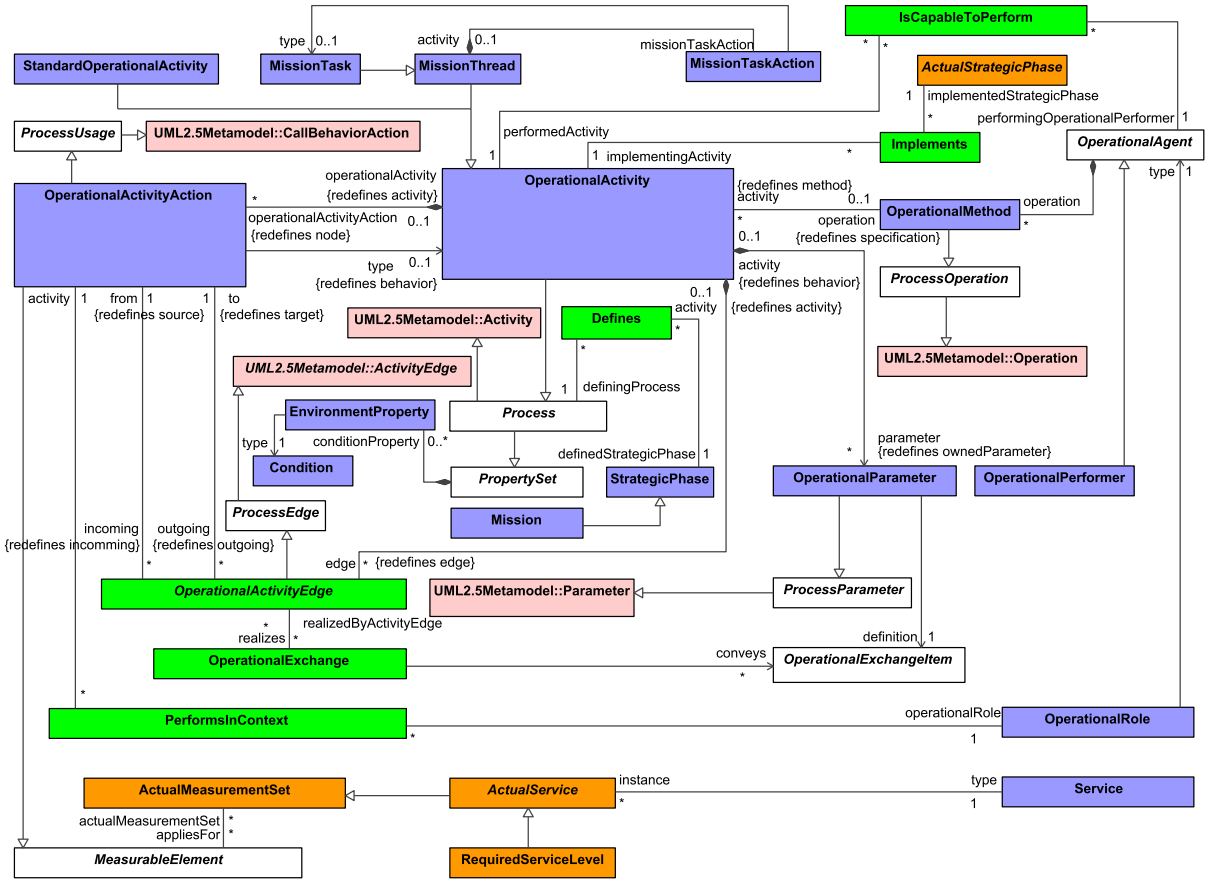

Figure 8:29 - Operational Processes: Mission Threads

Elements

- ActualMeasurementSet
- ActualService
- ActualStrategicPhase
- Condition
- Defines
- EnvironmentProperty
- Implements
- IsCapableToPerform
- MeasurableElement
- Mission
- MissionTask
- MissionTaskAction
- MissionThread
- OperationalActivity
- OperationalActivityAction
- OperationalActivityEdge
- OperationalAgent
- OperationalExchange
- OperationalExchangeItem
- OperationalMethod
- OperationalParameter
- OperationalPerformer
- OperationalRole
- PerformsInContext
- Process
- ProcessEdge
- ProcessOperation
- ProcessParameter
- ProcessUsage
- PropertySet
- RequiredServiceLevel
- Service
- StandardOperationalActivity
- StrategicPhase
- UML2.5Metamodel::Activity
- UML2.5Metamodel::ActivityEdge
- UML2.5Metamodel::CallBehaviorAction
- UML2.5Metamodel::Operation
- UML2.5Metamodel::Parameter

8.1.4.5 View Specifications::Operational::States

Contains the diagrams that document the Operational States View Specification.

8.1.4.5.1 View Specifications::Operational::States::Operational States

Stakeholders: Systems Engineers, Software Engineers.

Concerns: capture state-based behavior of an operational OperationalPerformer.

Definition: it is a graphical representation of states of an operational OperationalPerformer and how that operational OperationalPerformer responds to various events and actions.

Recommended Implementation: SysML State Machine Diagram.

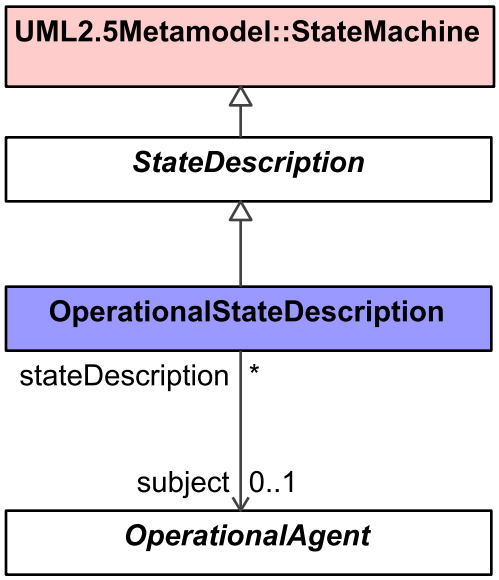

Figure 8:30 - Operational States

Elements

- OperationalAgent
- OperationalStateDescription
- StateDescription
- UML2.5Metamodel::StateMachine

8.1.4.6 View Specifications::Operational::Sequences

Contains the diagrams that document the Operational Sequences View Specification.

8.1.4.6.1 View Specifications::Operational::Sequences::Operational Sequences

Stakeholders: Systems Engineers, Business Architects.

Concerns: express a time ordered examination of the operational exchanges as a result of a particular operational scenario.

Definition: provides a time-ordered examination of the operational exchanges between participating nodes (OperationalPerformer roles) as a result of a particular operational scenario.

Recommended Implementation: SysML Sequence Diagram, BPMN Collaboration Diagram.

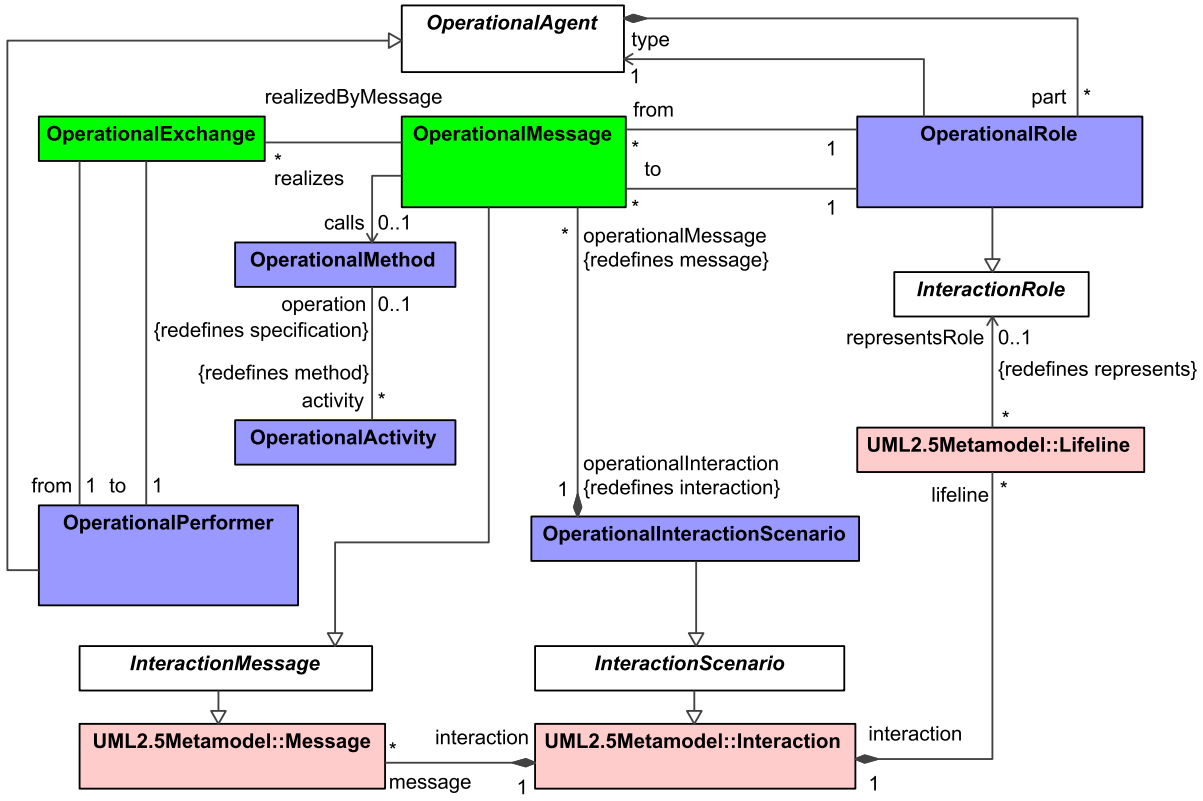

Figure 8:31 - Operational Sequences

Elements

- InteractionMessage
- InteractionRole
- InteractionScenario
- OperationalActivity
- OperationalAgent
- OperationalExchange
- OperationalInteractionScenario
- OperationalMessage
- OperationalMethod
- OperationalPerformer
- OperationalRole
- UML2.5Metamodel::Interaction
- UML2.5Metamodel::Lifeline
- UML2.5Metamodel::Message

8.1.4.7 View Specifications::Operational::Constraints

Contains the diagrams that document the Operational Constraints View Specification.

8.1.4.7.1 View Specifications::Operational::Constraints::Operational Constraints

Stakeholders: Systems Engineers, Architects, Program Sponsors

Concerns: define operational limitations, constraints and performance parameters for the enterprise.

Definition: specifies traditional textual operational or business rules that are constraints on the way that business is done in the enterprise. The addition of SysML parametrics provides a computational means of defining operational constraints across the enterprise or within a specific operational context.

Recommended Implementation: tabular format, SysML Block Definition Diagram, SysML Parametric Diagram.

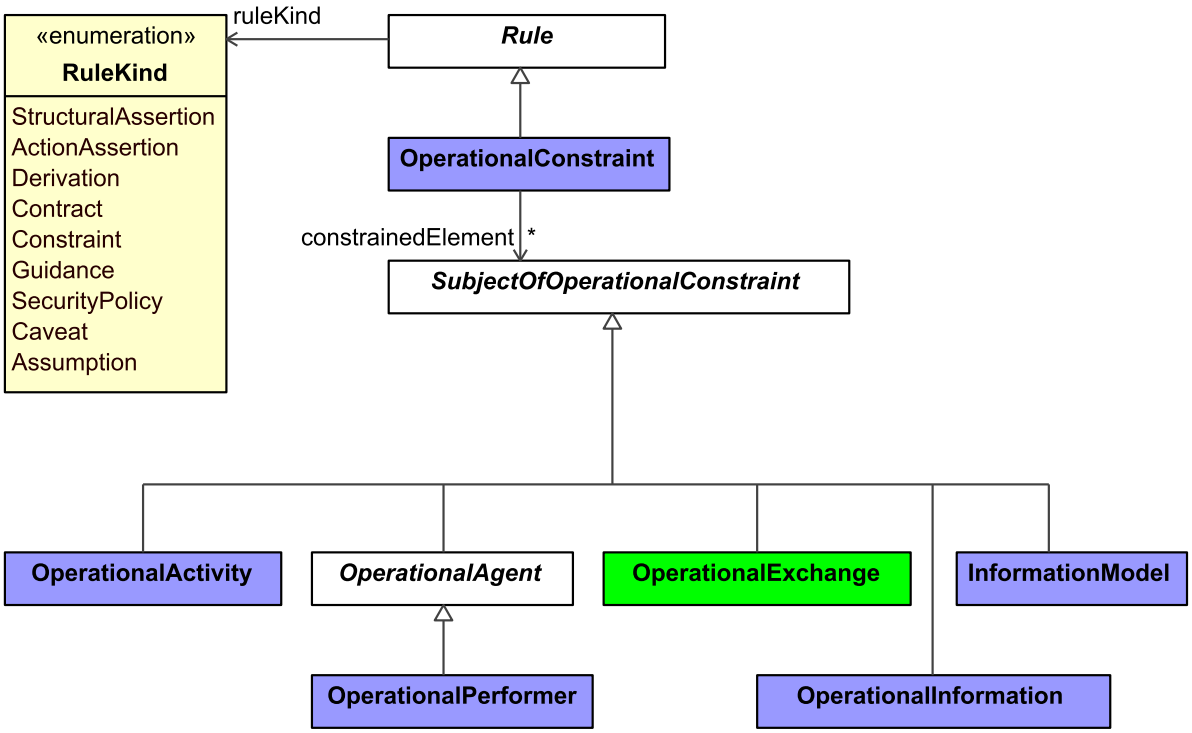

Figure 8:32 - Operational Constraints

Elements

- InformationModel
- OperationalActivity
- OperationalAgent
- OperationalConstraint
- OperationalExchange
- OperationalInformation
- OperationalPerformer
- Rule
- SubjectOfOperationalConstraint

8.1.4.8 View Specifications::Operational::Traceability

Contains the diagrams that document the Operational Traceability View Specification.

8.1.4.8.1 View Specifications::Operational::Traceability::Operational Traceability

Stakeholders: PMs, Enterprise Architects, Business Architects.

Concerns: traceability between capabilities and operational activities and capabilities and operational agents.

Definition: describes the mapping between the capabilities required by an Enterprise and the supporting operational activities and operational agents.

Recommended Implementation: matrix format, SysML Block Definition Diagram.

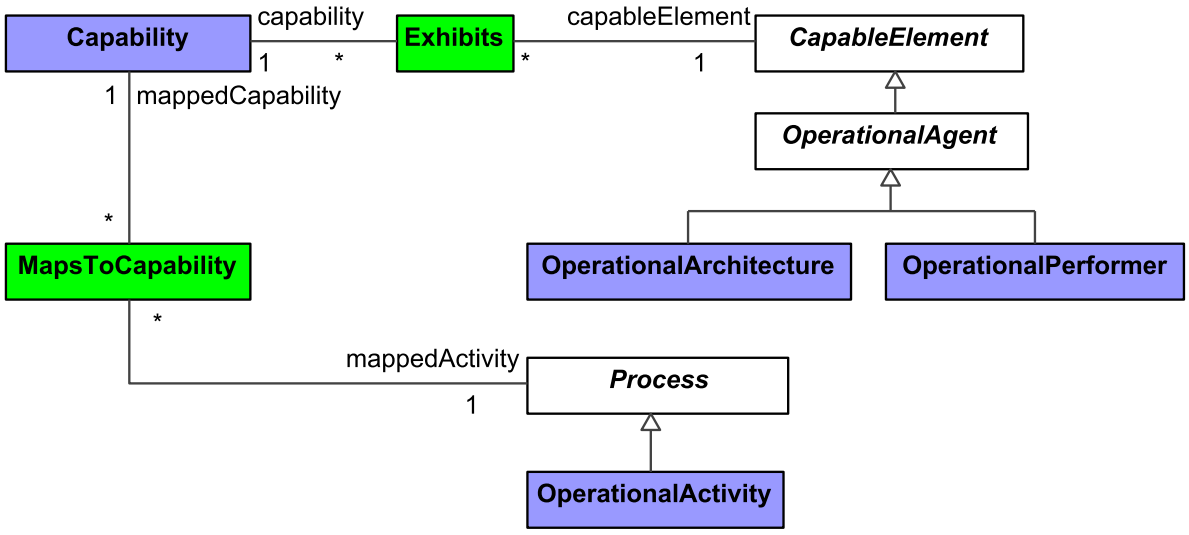

Figure 8:33 - Operational Traceability

Elements

- Capability
- CapableElement
- Exhibits
- MapsToCapability
- OperationalActivity
- OperationalAgent
- OperationalArchitecture
- OperationalPerformer
- Process

8.1.5 View Specifications::Services

Stakeholders: Enterprise Architects, Solution Providers, Systems Engineers, Software Architects, Business Architects..

Concerns: specifications of services required to exhibit a Capability.

Definition: shows Service specifications and required and provided service levels of these specifications required to exhibit a Capability or to support an Operational Activity.

8.1.5.1 View Specifications::Services::Taxonomy

Contains the diagrams that document the Services Taxonomy View Specification.

8.1.5.1.1 View Specifications::Services::Taxonomy::Services Taxonomy

Stakeholders: Enterprise Architects, Solution Providers, Systems Engineers, Software Architects, Business Architects.

Concerns: service specification types and required and provided service levels of these types.

Definition: shows the taxonomy of types of services and the level of service that they are expected to provide or are required to meet through the display of ActualMeasurements associated with the Provided and Required Service Level.

Recommended Implementation: SysML Block Definition Diagram.

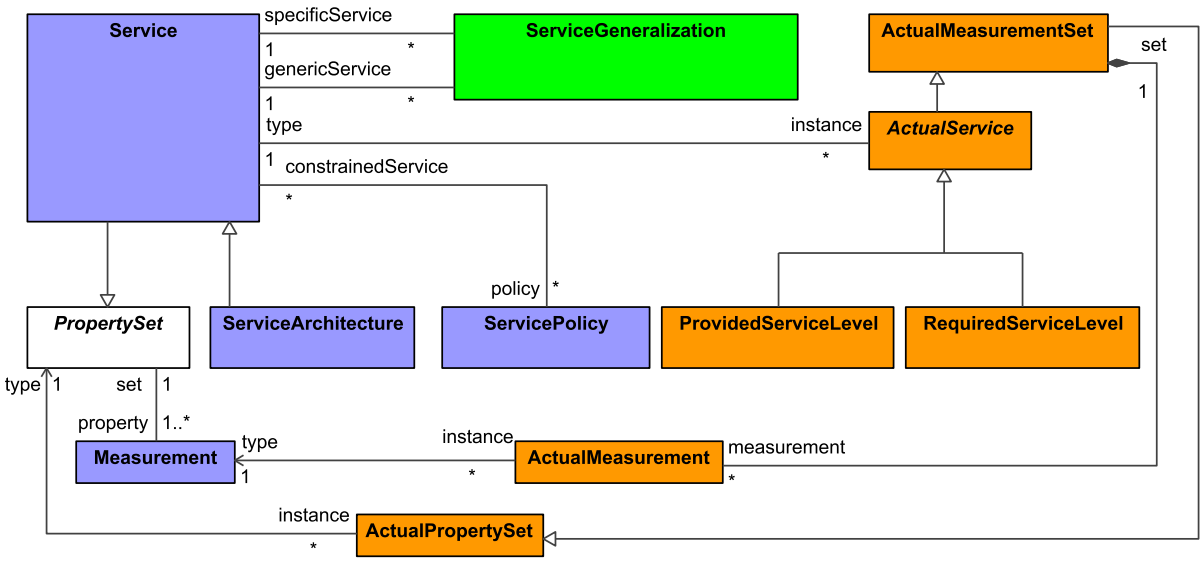

Figure 8:34 - Services Taxonomy

Elements

- ActualMeasurement
- ActualMeasurementSet
- ActualPropertySet
- ActualService
- Measurement
- PropertySet
- ProvidedServiceLevel
- RequiredServiceLevel
- Service
- ServiceArchitecture
- ServiceGeneralization
- ServicePolicy

8.1.5.2 View Specifications::Services::Structure

Contains the diagrams that document the Services Structure View Specification.

8.1.5.2.1 View Specifications::Services::Structure::Services Structure

Stakeholders: Solution Providers, Systems Engineers, Software Architects, Business Architects.

Concerns: combination of services required to exhibit a capability.

Definition: shows the composition of services and how services are combined into a higher level service required to exhibit a capability or support an operational activity.

Recommended Implementation: SysML Block Definition Diagram, SysML Internal Block Diagram.

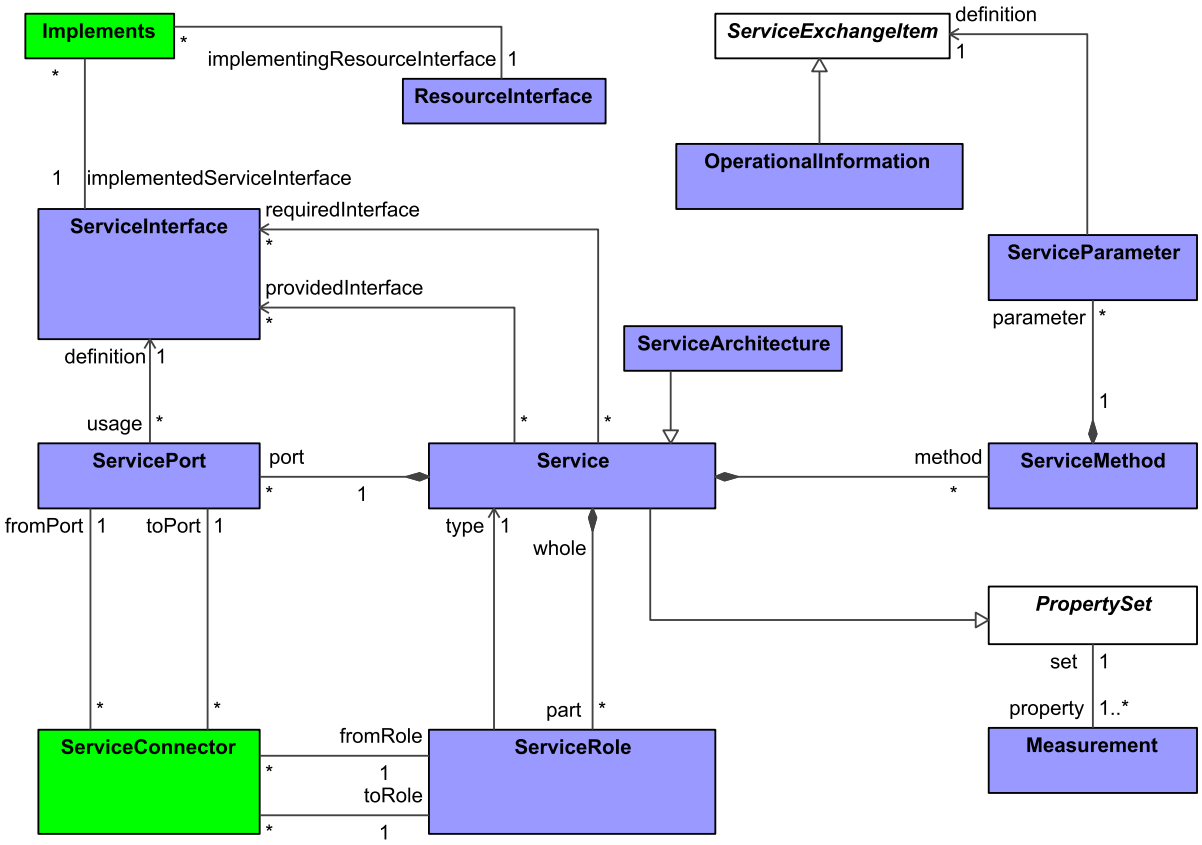

Figure 8:35 - Services Structure

Elements

- Implements
- Measurement
- OperationalInformation
- PropertySet
- ResourceInterface
- Service
- ServiceArchitecture
- ServiceConnector
- ServiceExchangeItem
- ServiceInterface
- ServiceMethod
- ServiceParameter
- ServicePort
- ServiceRole

8.1.5.3 View Specifications::Services::Connectivity

Contains the diagrams that document the Services Connectivity View Specification.

8.1.5.3.1 View Specifications::Services::Connectivity::Services Connectivity

Stakeholders: Solution Providers, Systems Engineers, Software Architects, Business Architects.

Concerns: interoperability among services

Definition: specifies service interfaces, e.g. provided and required service operations, to ensure compatibility and reusability of services.

Recommended Implementation: SysML Block Definition Diagram, SysML Internal Block Diagram, tabular format.

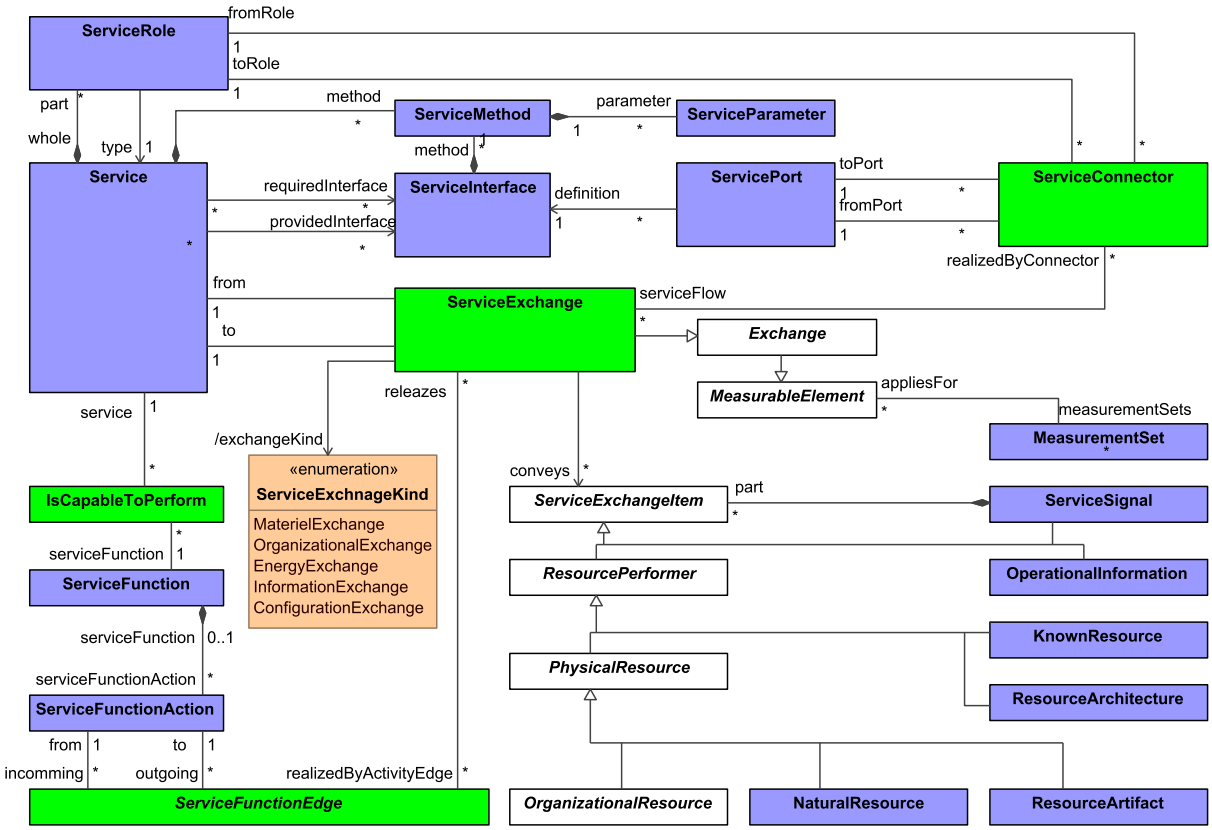

Figure 8:36 - Services Connectivity

Elements

- Exchange
- IsCapableToPerform
- KnownResource
- MeasurableElement
- MeasurementSet
- NaturalResource
- OperationalInformation
- OrganizationalResource
- PhysicalResource
- ResourceArchitecture
- ResourceArtifact
- ResourcePerformer
- Service
- ServiceConnector
- ServiceExchange
- ServiceExchangeItem
- ServiceFunction
- ServiceFunctionAction
- ServiceFunctionEdge
- ServiceInterface

- ServiceMethod
- ServiceParameter
- ServicePort
- ServiceRole
- ServiceSignal

8.1.5.4 View Specifications::Services::Processes

Contains the diagrams that document the Services Processes View Specification.

8.1.5.4.1 View Specifications::Services::Processes::Services Processes

Stakeholders: Solution Providers, Systems Engineers, Software Architects, Business Architects.

Concerns: the behavior of a service in terms of the operational activities it is expected to support.

Definition: provides detailed information regarding the allocation of service functions to service specifications, and data flows between service functions.

Recommended Implementation: SysML Activity Diagram, SysML Block Definition Diagram.

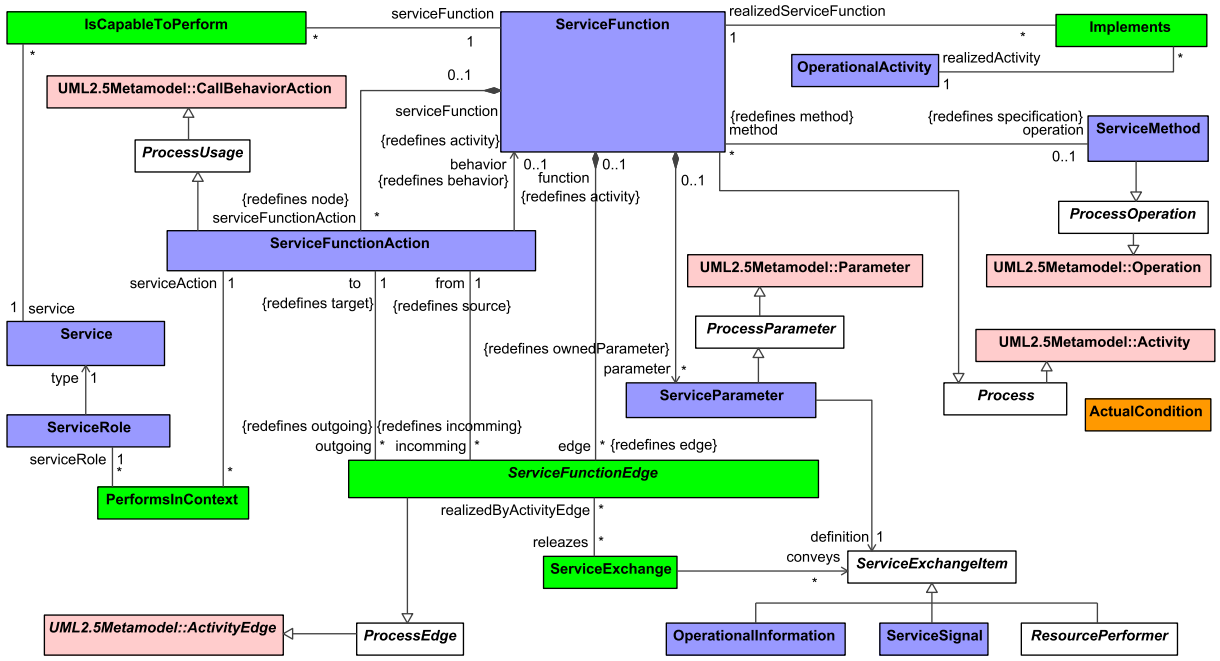

Figure 8:37 - Services Processes

Elements

- ActualCondition
- Implements
- IsCapableToPerform
- OperationalActivity
- OperationalInformation
- PerformsInContext
- Process
- ProcessEdge
- ProcessOperation
- ProcessParameter
- ProcessUsage
- ResourcePerformer

- Service
- ServiceExchange
- ServiceExchangeItem
- ServiceFunction
- ServiceFunctionAction
- ServiceFunctionEdge
- ServiceMethod
- ServiceParameter
- ServiceRole
- ServiceSignal
- UML2.5Metamodel::Activity
- UML2.5Metamodel::ActivityEdge
- UML2.5Metamodel::CallBehaviorAction
- UML2.5Metamodel::Operation
- UML2.5Metamodel::Parameter

8.1.5.4.2 View Specifications::Services::Processes::Services Processes BPMN Semantics

Stakeholders: Solution Providers, Software Architects, Business Architects.

Concerns: the behavior of a service in terms of the operational activities it is expected to support.

Definition: provides detailed information regarding the allocation of service functions to service specifications, and data flows between service functions using BPMN.

Recommended Implementation: BPMN Process Diagram, SysML Block Definition Diagram.

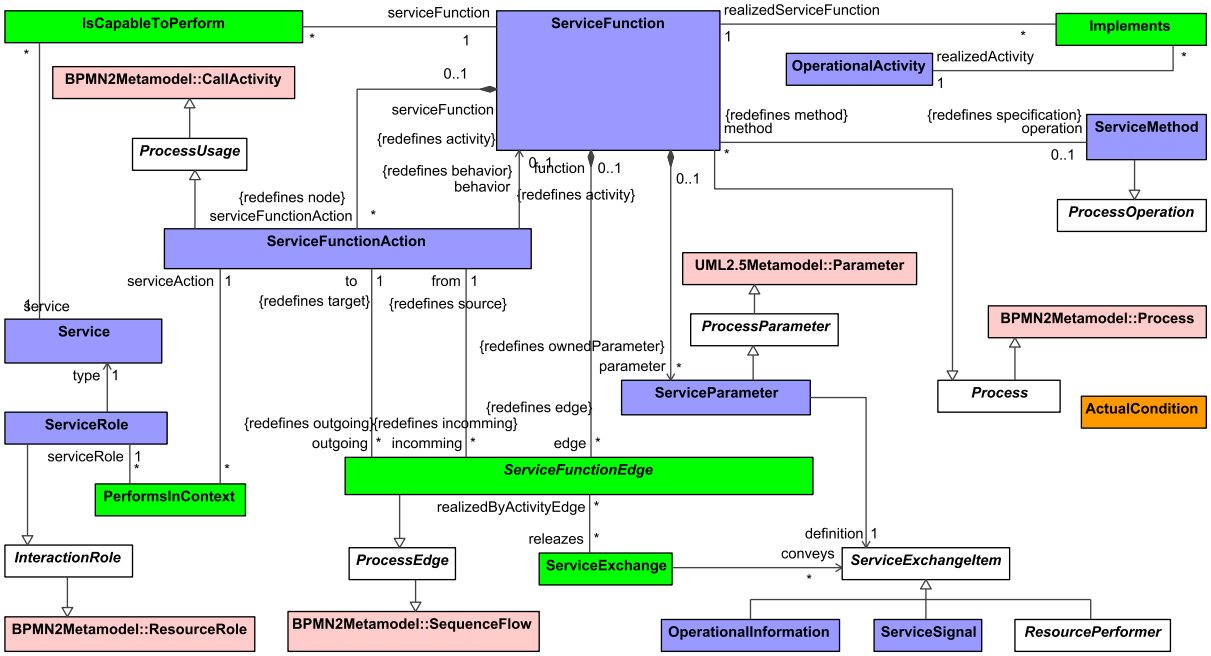

Figure 8:38 - Services Processes BPMN Semantics

Elements

- ActualCondition
- BPMN2Metamodel::CallActivity
- BPMN2Metamodel::Process
- BPMN2Metamodel::ResourceRole

- BPMN2Metamodel::SequenceFlow
- Implements
- InteractionRole
- IsCapableToPerform
- OperationalActivity
- OperationalInformation
- PerformsInContext
- Process
- ProcessEdge
- ProcessOperation
- ProcessParameter
- ProcessUsage
- ResourcePerformer
- Service
- ServiceExchange
- ServiceExchangeItem
- ServiceFunction
- ServiceFunctionAction
- ServiceFunctionEdge
- ServiceMethod
- ServiceParameter
- ServiceRole
- ServiceSignal
- UML2.5Metamodel::Parameter

8.1.5.5 View Specifications::Services::States

Contains the diagrams that document the Services States View Specification.

8.1.5.5.1 View Specifications::Services::States::Services States

Stakeholders: Solution Providers, Systems Engineers, Software Architects, Business Architects.

Concerns: the behavior of a service specification in terms of states and events causing transitions between states.

Definition: specifies the possible states a service specification may have, and the possible transitions between those states.

Recommended Implementation: SysML State Machine Diagram.

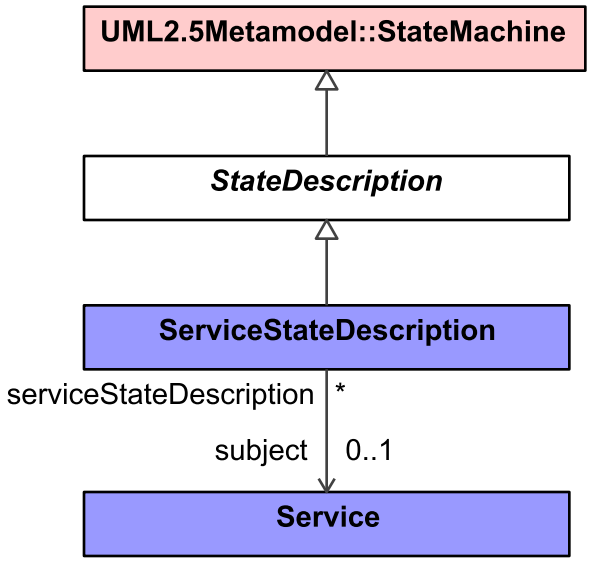

Figure 8:39 - Services States

Elements

- Service
- ServiceStateDescription
- StateDescription
- UML2.5Metamodel::StateMachine

8.1.5.6 View Specifications::Services::Sequences

Contains the diagrams that document the Services Sequences View Specification.

8.1.5.6.1 View Specifications::Services::Sequences::Services Sequences

Stakeholders: Solution Providers, Systems Engineers, Software Architects, Business Architects.

Concerns: the behavior of a service specification in terms of expected time-ordered examination of the interactions between service roles.

Definition: specifies how a service roles interact with each other, service providers and consumers, and the sequence and dependencies of those interactions.

Recommended Implementation: SysML Sequence Diagram.

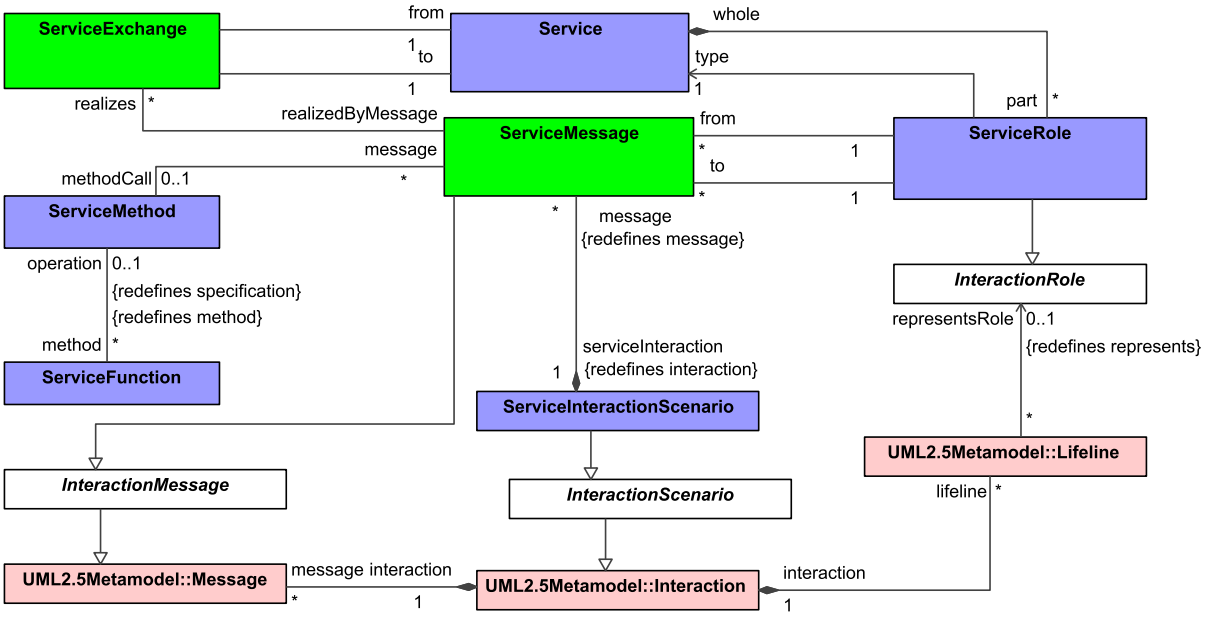

Figure 8:40 - Services Sequences

Elements

- InteractionMessage
- InteractionRole
- InteractionScenario
- Service
- ServiceExchange
- ServiceFunction
- ServiceInteractionScenario
- ServiceMessage
- ServiceMethod
- ServiceRole
- UML2.5Metamodel::Interaction
- UML2.5Metamodel::Lifeline
- UML2.5Metamodel::Message

8.1.5.7 View Specifications::Services::Constraints

Contains the diagrams that document the Services Constraints View Specification.

8.1.5.7.1 View Specifications::Services::Constraints::Services Constraints

Stakeholders: Solution Providers, Systems Engineers, Software Architects, Business Architects.

Concerns: service policies that apply to implementations of service specifications.

Definition: specifies traditional textual service policies that are constraints on the way that service specifications are implemented within resources. The addition of SysML parametrics provide a computational means of defining service policies across the enterprise or within a specific service configuration.

Recommended Implementation: tabular format, SysML Parametric Diagram.

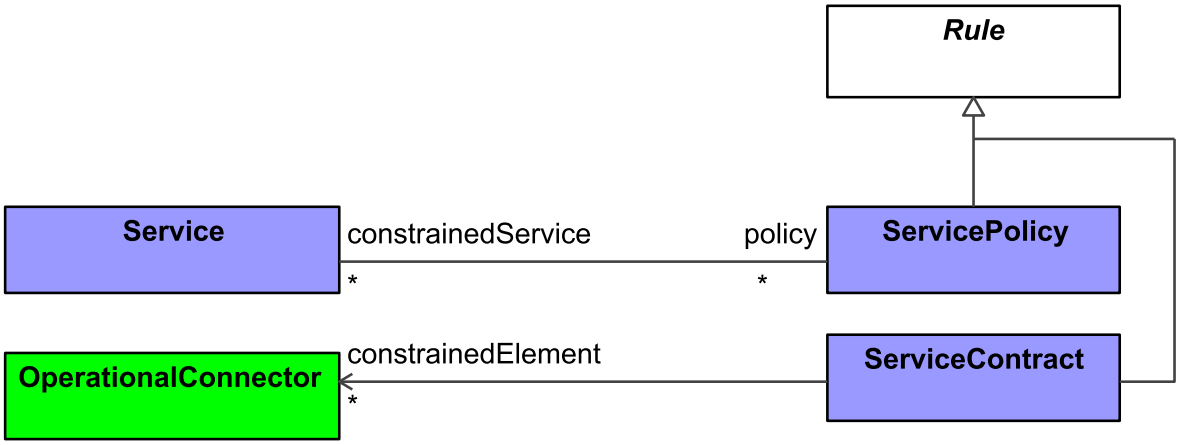

Figure 8:41 - Services Constraints

Elements

- OperationalConnector
- Rule
- Service
- ServiceContract
- ServicePolicy

8.1.5.8 View Specifications::Services::Roadmap

Contains the diagrams that document the Services Roadmap View Specification.

8.1.5.8.1 View Specifications::Services::Roadmap::Services Roadmap

Stakeholders: Solution Providers, Systems Engineers, Software Architects, Business Architects.

Concerns: service specification changes over time.

Definition: provides an overview of how a service specification changes over time. It shows the combination of several service specifications mapped against a timeline.

Recommended Implementation: timeline, SysML Block Definition Diagram, SysML Internal Block Diagram.

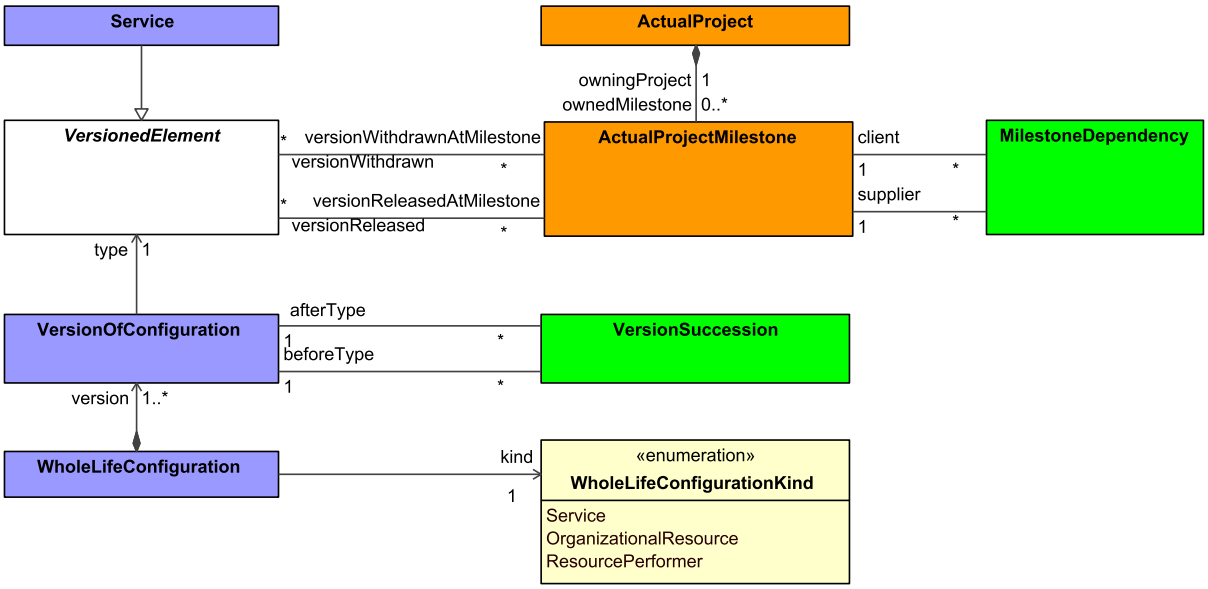

Figure 8:42 - Services Roadmap

Elements

- ActualProject
- ActualProjectMilestone
- MilestoneDependency
- Service
- VersionedElement
- VersionOfConfiguration
- VersionSuccession
- WholeLifeConfiguration

8.1.5.9 View Specifications::Services::Traceability

Contains the diagrams that document the Services Traceability View Specification.

8.1.5.9.1 View Specifications::Services::Traceability::Services Traceability

Stakeholders: Solution Providers, Systems Engineers, Software Architects, Business Architects.

Concerns: traceability between operational activities and service specifications that support them.

Definition: depicts the mapping of service specifications to operational activities and how service specifications contribute to the achievement of a capability.

Recommended Implementation: tabular or matrix format.

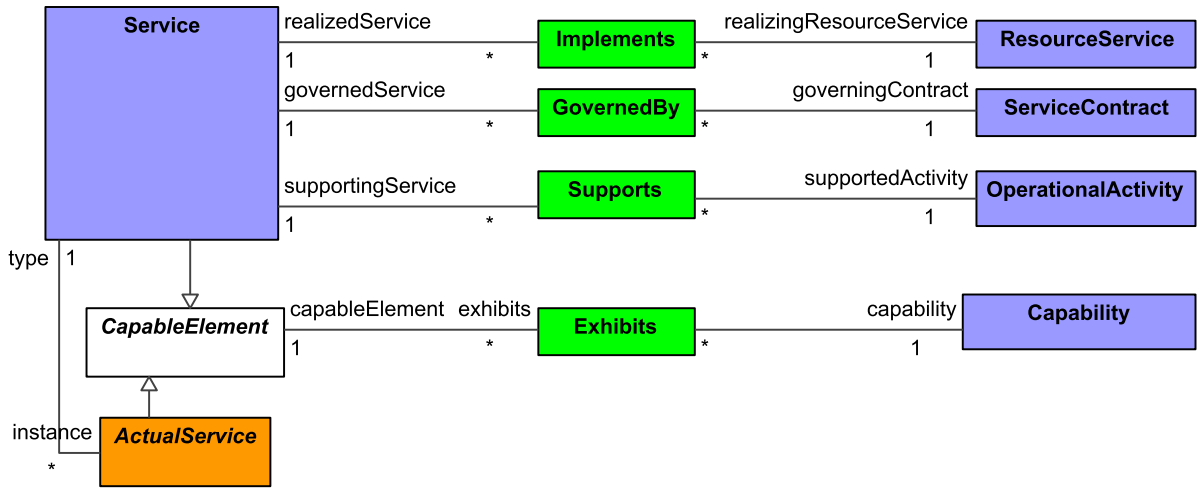

Figure 8:43 - Services Traceability

Elements

- ActualService
- Capability
- CapableElement
- Exhibits
- GovernedBy
- Implements
- OperationalActivity
- ResourceService
- Service
- ServiceContract
- Supports

8.1.6 View Specifications::Personnel

Stakeholders: Human resources, Solution Providers, PMs.

Concerns: human factors.

Definition: aims to clarify the role of Human Factors (HF) when creating architectures in order to facilitate both Human Factors Integration (HFI) and systems engineering (SE).

8.1.6.1 View Specifications::Personnel::Taxonomy

Contains the diagrams that document the Personnel Taxonomy View Specification.

8.1.6.1.1 View Specifications::Personnel::Taxonomy::Personnel Taxonomy

Stakeholders: Human resources, Solution Providers, PMs.

Concerns: organizational resource types.

Definition: shows the taxonomy of types of organizational resources.

Recommended Implementation: SysML Block Definition Diagram.

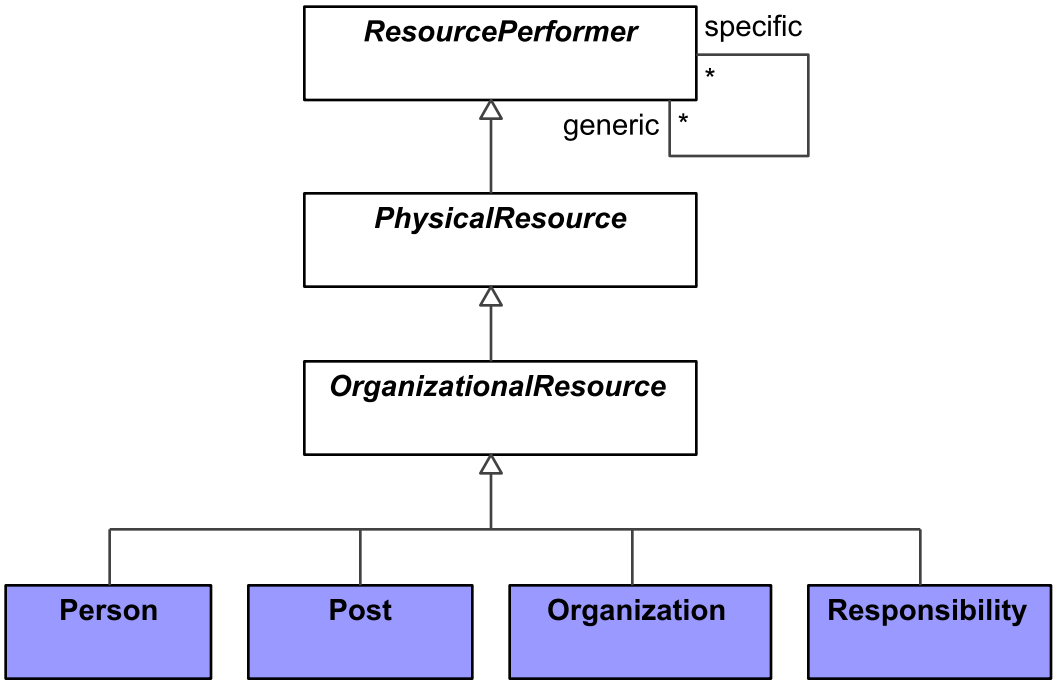

Figure 8:44 - Personnel Taxonomy

Elements

- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- ResourcePerformer
- Responsibility

8.1.6.2 View Specifications::Personnel::Structure

Contains the diagrams that document the Personnel Structure View Specification.

8.1.6.2.1 View Specifications::Personnel::Structure::Personnel Structure

Stakeholders: Human resources, Solution Providers, PMs.

Concerns: typical organizational structure used to support a capability(ies).

Definition: shows organizational structures and possible interactions between organizational resources.

Recommended Implementation: SysML Block Definition Diagram, SysML Internal Block Diagram.

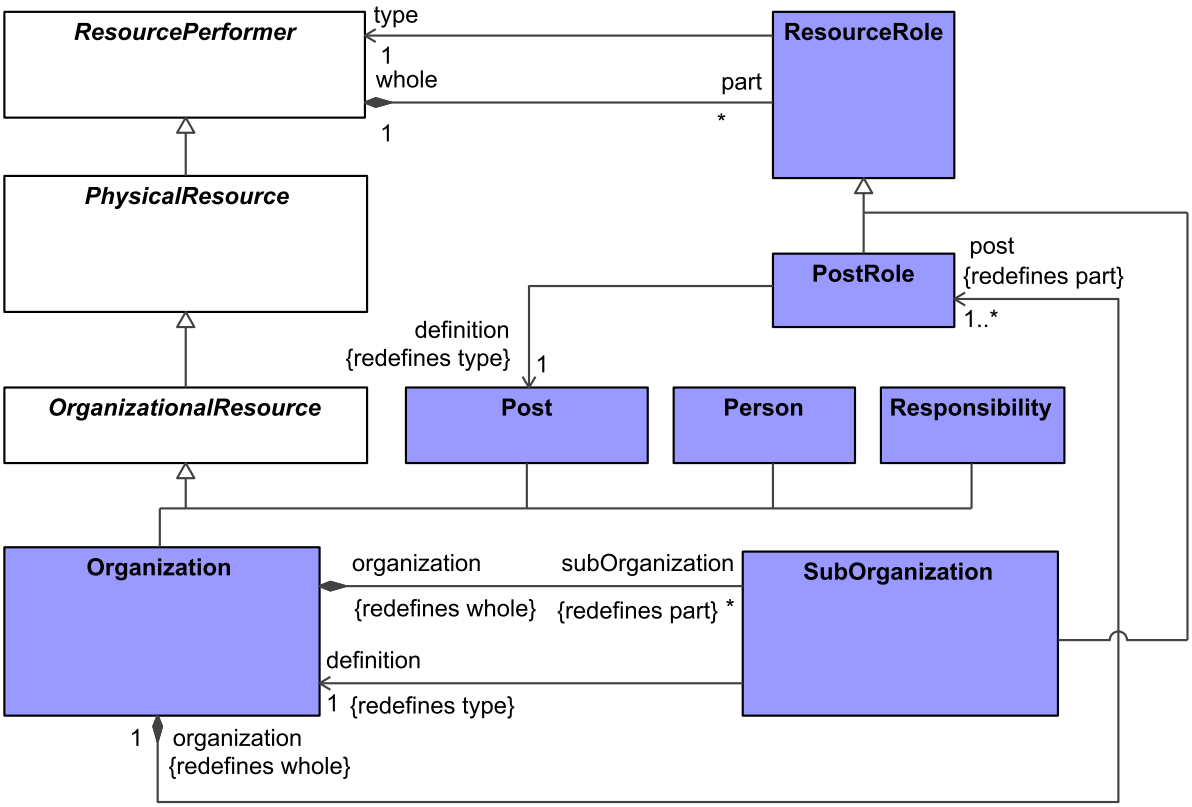

Figure 8:45 - Personnel Structure

Elements

- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- PostRole
- ResourcePerformer
- ResourceRole
- Responsibility
- SubOrganization

8.1.6.3 View Specifications::Personnel::Connectivity

Contains the diagrams that document the Personnel Connectivity View Specification.

8.1.6.3.1 View Specifications::Personnel::Connectivity::Personnel Connectivity

Stakeholders: Solution providers.

Concerns: interaction of organizational resources.

Definition: captures the possible interactions between organizational resources, including command and control relationships. Interactions typically illustrate the fundamental roles and management responsibilities.

Recommended Implementation: tabular format.

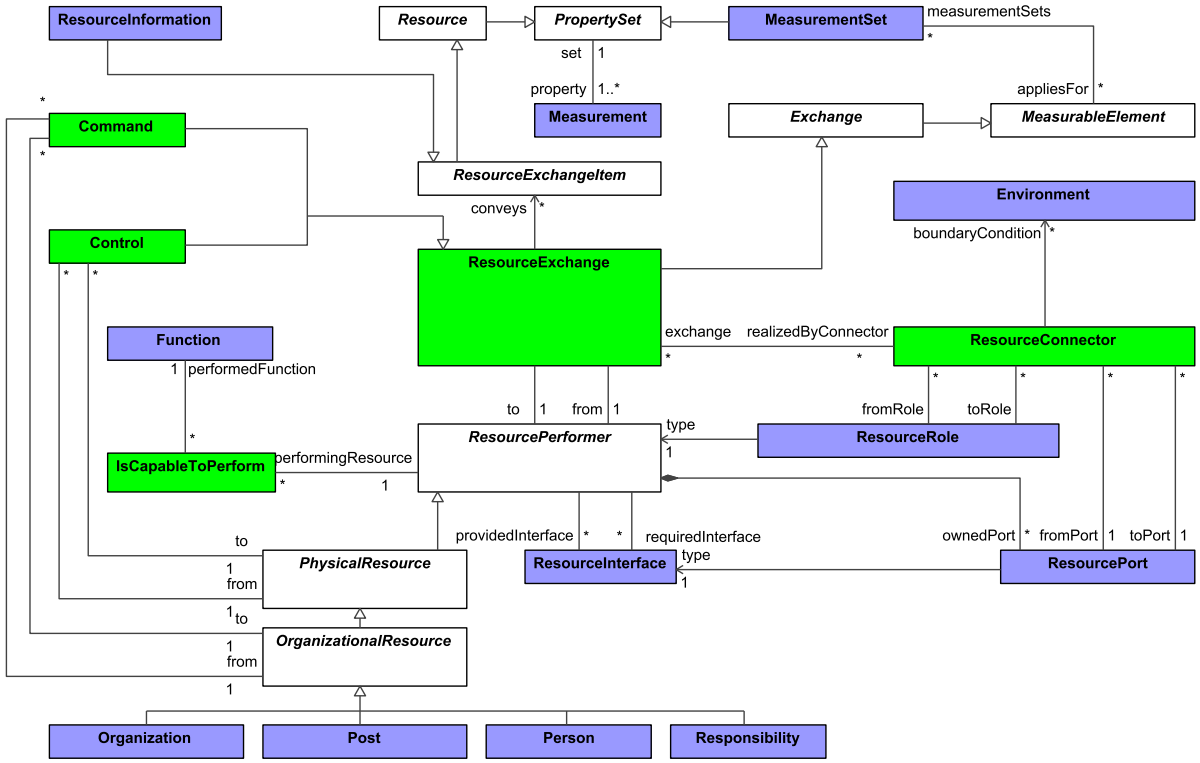

Figure 8:46 - Personnel Connectivity

Elements

- Command
- Control
- Environment
- Exchange
- Function
- IsCapableToPerform
- MeasurableElement
- Measurement
- MeasurementSet
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- PropertySet
- Resource
- ResourceConnector
- ResourceExchange
- ResourceExchangeItem
- ResourceInformation
- ResourceInterface
- ResourcePerformer
- ResourcePort
- ResourceRole
- Responsibility

8.1.6.4 View Specifications::Personnel::Processes

Contains the diagrams that document the Personnel Processes View Specification.

8.1.6.4.1 View Specifications::Personnel::Processes::Personnel Processes

Stakeholders: Systems engineers, Solution providers.

Concerns: functions that have to be carried out by organizational resources.

Definition: specifies organizational resource functions in relation to resource definitions.

Recommended Implementation: SysML Activity Diagram, SysML Block Definition Diagram, BPMN Process Diagram as described in the Resources Processes section.

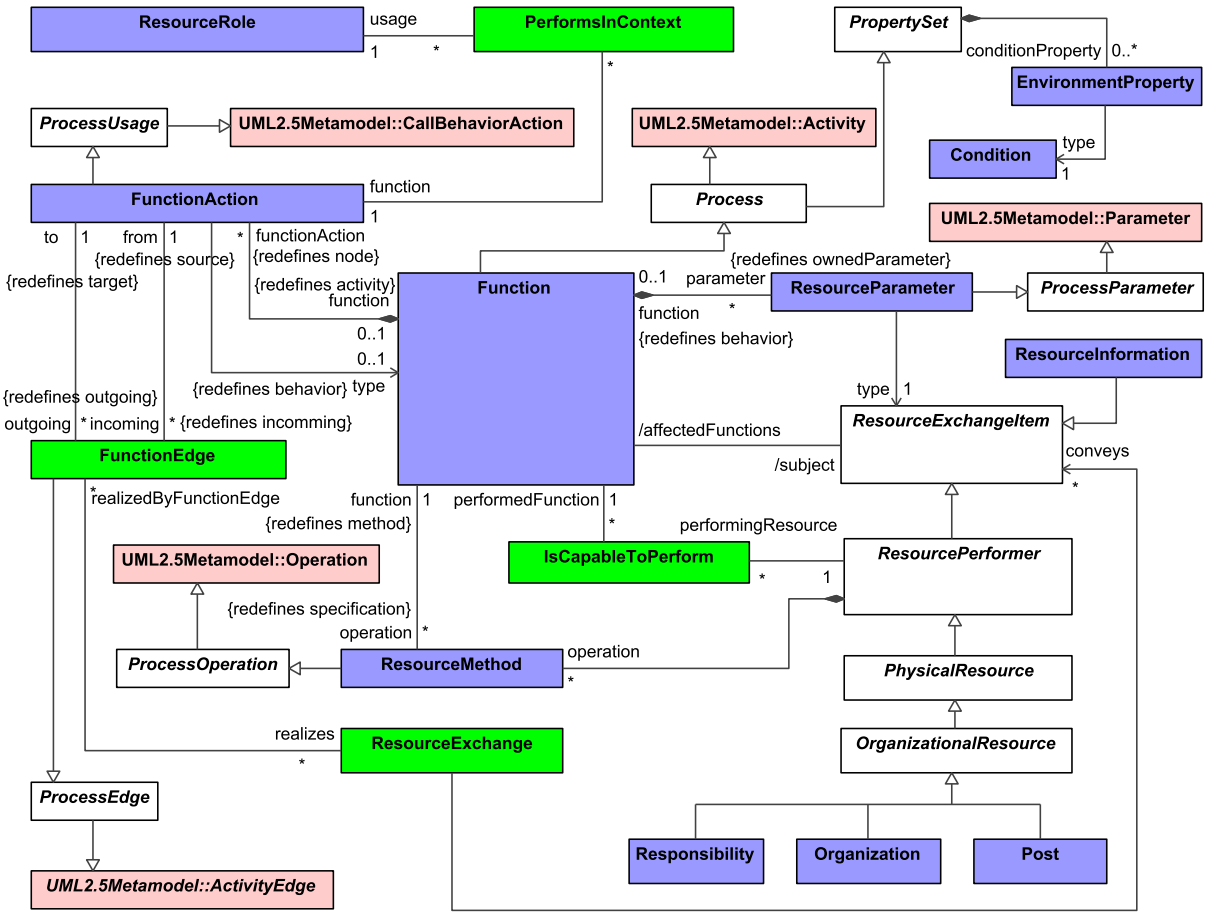

Figure 8:47 - Personnel Processes

Elements

- Condition
- EnvironmentProperty
- Function
- FunctionAction
- FunctionEdge
- IsCapableToPerform
- Organization
- OrganizationalResource
- PerformsInContext
- PhysicalResource

- Post
- Process
- ProcessEdge
- ProcessOperation
- ProcessParameter
- ProcessUsage
- PropertySet
- ResourceExchange
- ResourceExchangeItem
- ResourceInformation
- ResourceMethod
- ResourceParameter
- ResourcePerformer
- ResourceRole
- Responsibility
- UML2.5Metamodel::Activity
- UML2.5Metamodel::ActivityEdge
- UML2.5Metamodel::CallBehaviorAction
- UML2.5Metamodel::Operation
- UML2.5Metamodel::Parameter

8.1.6.5 View Specifications::Personnel::States

Contains the diagrams that document the Personnel States View Specification

8.1.6.5.1 View Specifications::Personnel::States::Personnel States

Stakeholders: Systems Engineers, Software Engineers.

Concerns: capture state-based behavior of an organizational resource.

Definition: it is a graphical representation of states of an organizational resource and how that organizational resource responds to various events and actions.

Recommended Implementation: SysML State Machine Diagram.

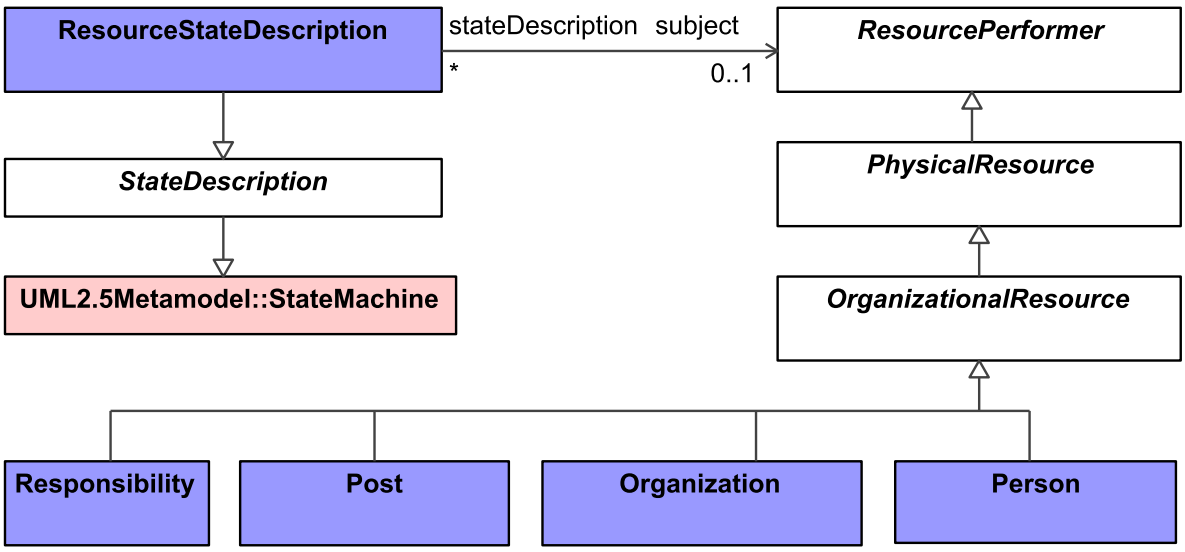

Figure 8:48 - Personnel States

Elements

- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- ResourcePerformer
- ResourceStateDescription
- Responsibility
- StateDescription
- UML2.5Metamodel::StateMachine

8.1.6.6 View Specifications::Personnel::Sequences

Contains the diagrams that document the Personnel Sequences View Specification.

8.1.6.6.1 View Specifications::Personnel::Sequences::Personnel Sequences

Stakeholders: Software Engineers, Systems Engineers.

Concerns: interactions between organizational resources (roles).

Definition: provides a time-ordered examination of the interactions between organizational resources.

Recommended Implementation: SysML Sequence Diagram, BPMN Collaboration Diagram.

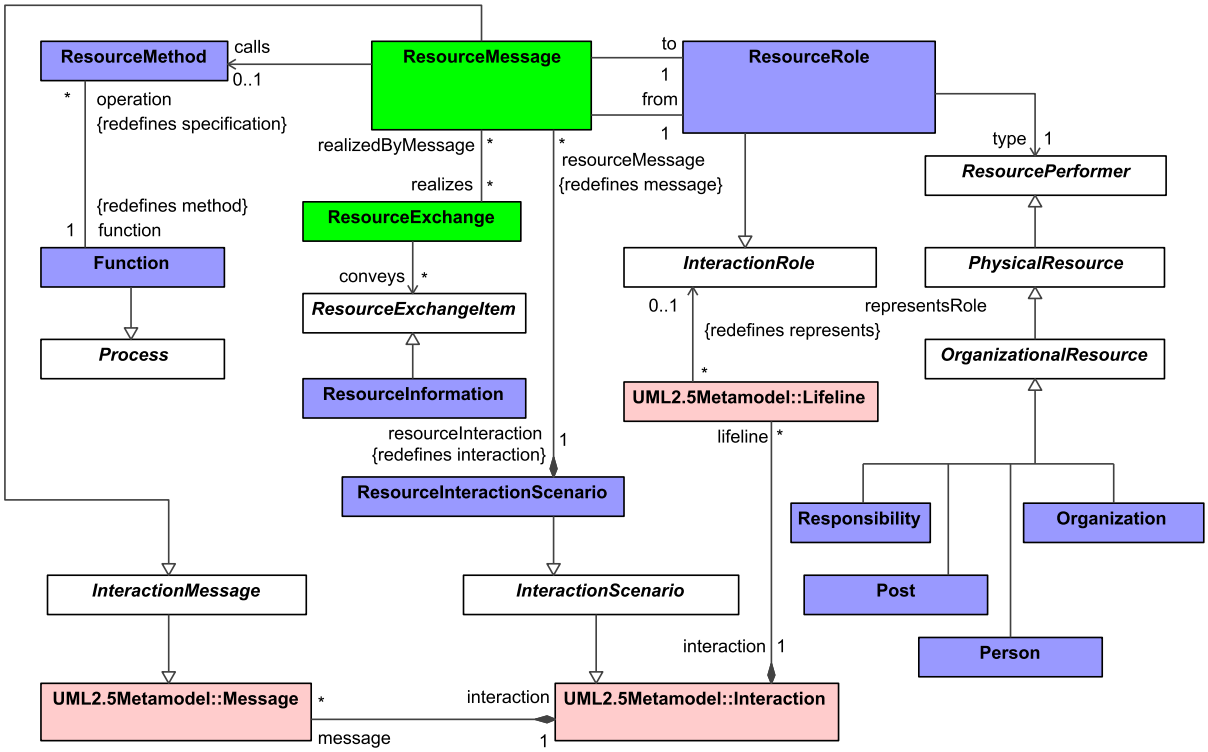

Figure 8:49 - Personnel Sequences

Elements

- Function
- InteractionMessage
- InteractionRole
- InteractionScenario
- Organization

- OrganizationalResource
- Person
- PhysicalResource
- Post
- Process
- ResourceExchange
- ResourceExchangeItem
- ResourceInformation
- ResourceInteractionScenario
- ResourceMessage
- ResourceMethod
- ResourcePerformer
- ResourceRole
- Responsibility
- UML2.5Metamodel::Interaction
- UML2.5Metamodel::Lifeline
- UML2.5Metamodel::Message

8.1.6.7 View Specifications::Personnel::Constraints

Contains the diagrams that document the Personnel Constraints View Specification.

8.1.6.7.1 View Specifications::Personnel::Constraints::Personnel Constraints: Competence

Stakeholders: Systems engineers, Solution providers.

Concerns: allocation of competencies to actual posts.

Definition: specifies requirements for actual organizational resources – by linking competencies and actual posts.

Recommended Implementation: SysML Block Definition Diagram.

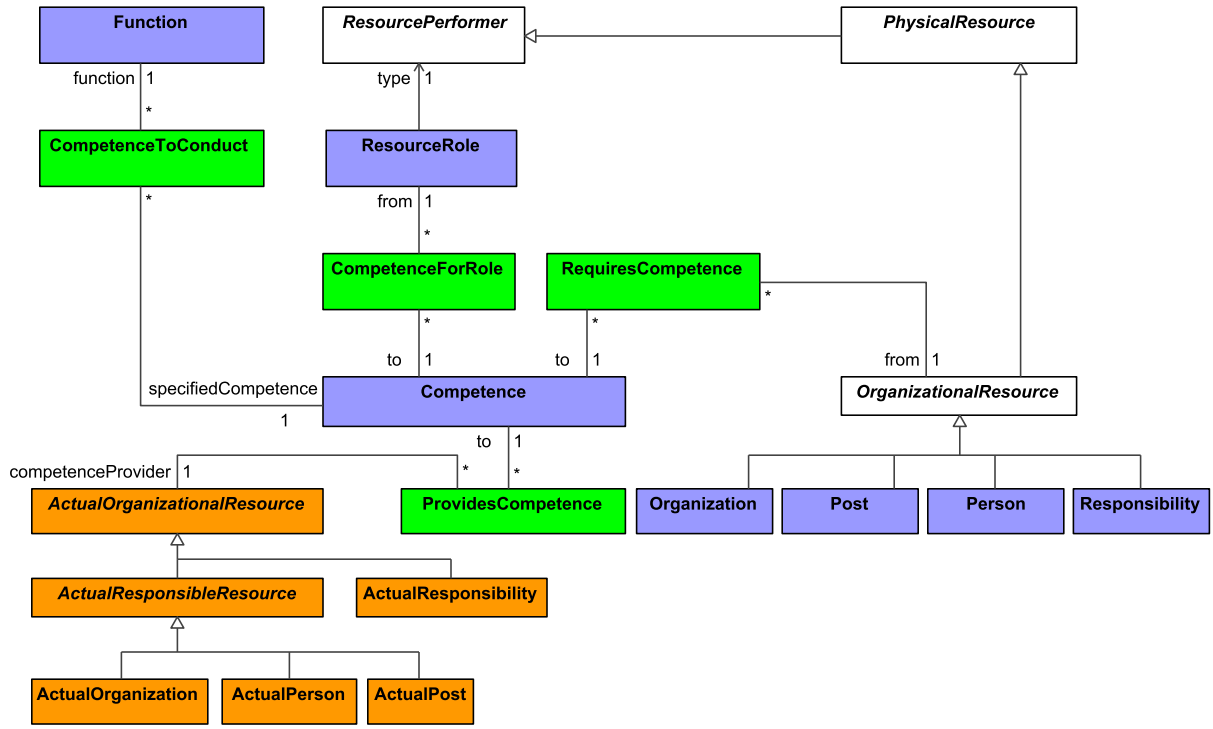

Figure 8:50 - Personnel Constraints: Competence

Elements

- ActualOrganization
- ActualOrganizationalResource
- ActualPerson
- ActualPost
- ActualResponsibility
- ActualResponsibleResource
- Competence
- CompetenceForRole
- CompetenceToConduct
- Function
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- ProvidesCompetence
- RequiresCompetence
- ResourcePerformer
- ResourceRole
- Responsibility

8.1.6.7.2 View Specifications::Personnel::Constraints::Personnel Constraints: Drivers

Stakeholders: Systems engineers, Solution providers, Human resources.

Concerns: optimization of organizational resource behavior.

Definition: captures the factors that affect, constrain and characterize organizational resource behavior as the basis for performance predictions at the level of actual persons and actual organizations. It creates a bridge between static architectural definitions and behavior predictions through executable models.

Recommended Implementation: tabular format, SysML Parametric Diagram, SysML Block Definition Diagram.

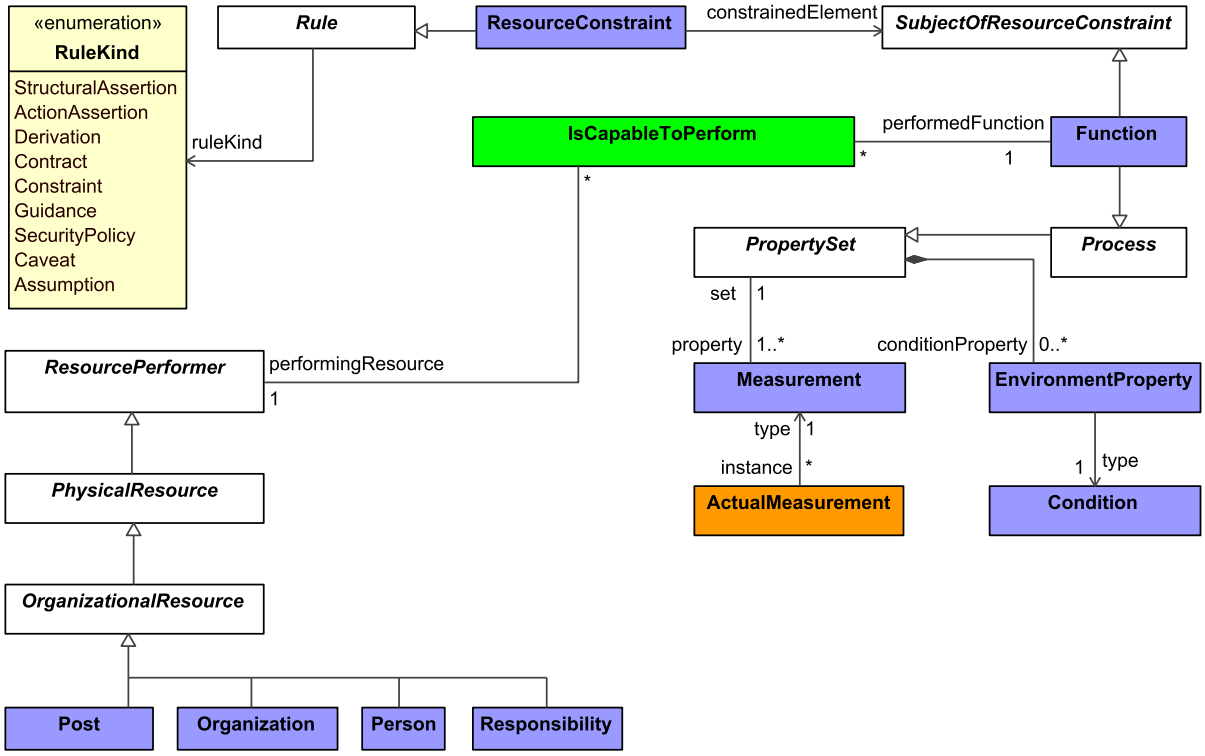

Figure 8:51 - Personnel Constraints: Drivers

Elements

- ActualMeasurement
- Condition
- EnvironmentProperty
- Function
- IsCapableToPerform
- Measurement
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- Process
- PropertySet
- ResourceConstraint
- ResourcePerformer
- Responsibility
- Rule
- SubjectOfResourceConstraint

8.1.6.7.3 View Specifications::Personnel::Constraints::Personnel Constraints: Performance

Stakeholders: Human resources, solution providers.

Concerns: how well an actual organizational resource matches the needs of the actual organization.

Definition: provides a repository for human-related measures (i.e. quality objectives and performance criteria (HFI values)), targets and competences.

Recommended Implementation: SysML Block Definition Diagram.

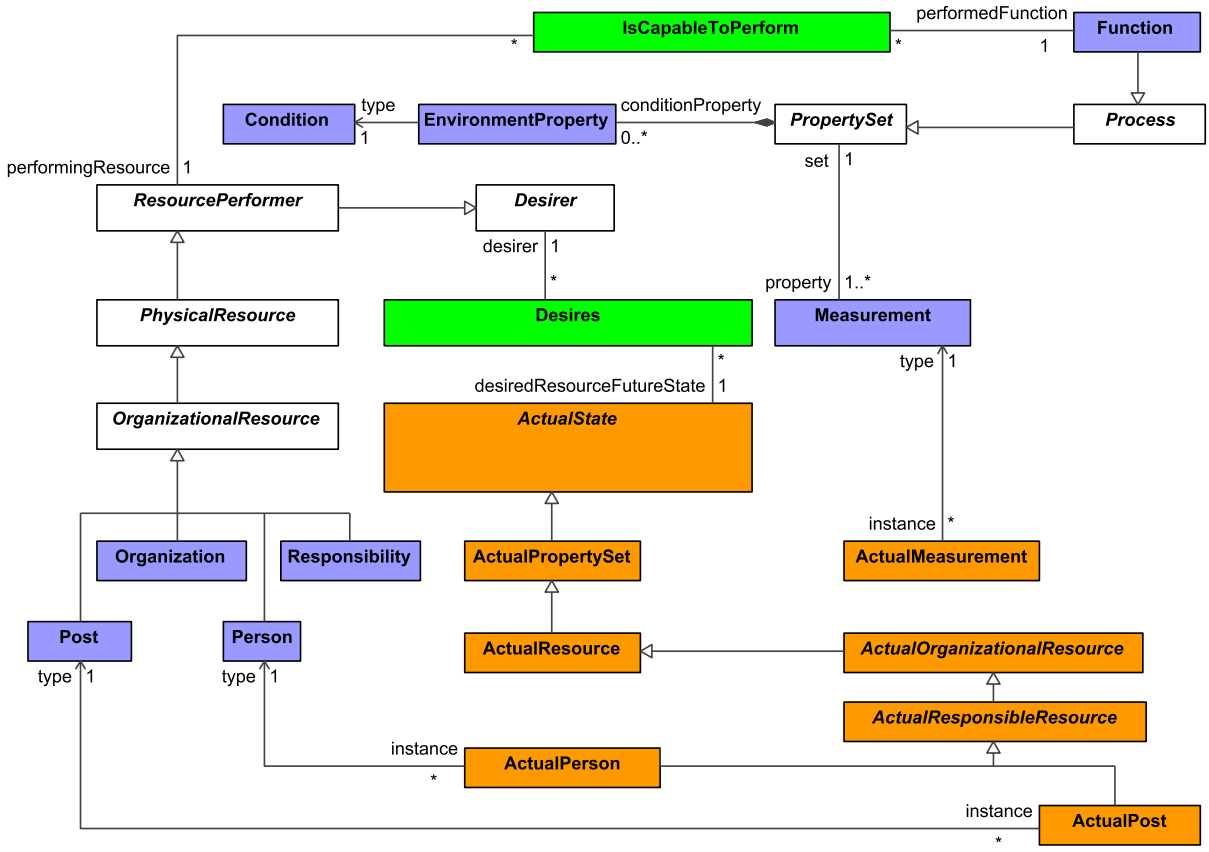

Figure 8:52 - Personnel Constraints: Performance

Elements

- ActualMeasurement
- ActualOrganizationalResource
- ActualPerson
- ActualPost
- ActualPropertySet
- ActualResource
- ActualResponsibleResource
- ActualState
- Condition
- Desirer
- Desires
- EnvironmentProperty
- Function
- IsCapableToPerform
- Measurement
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- Process
- PropertySet

- ResourcePerformer
- Responsibility

8.1.6.8 View Specifications::Personnel::Roadmap

Contains the diagrams that document the Personnel Roadmap View Specification.

8.1.6.8.1 View Specifications::Personnel::Roadmap::Personnel Roadmap: Availability

Stakeholders: Human Resources, Training, Logisticians, Solution Providers.

Concerns: the staffing and training of resources.

Definition: defines the requirements and functions to ensure that actual persons with the right competencies, and in the right numbers, are available to fulfill actual posts.

Recommended Implementation: Timeline, SysML Block Definition Diagram.

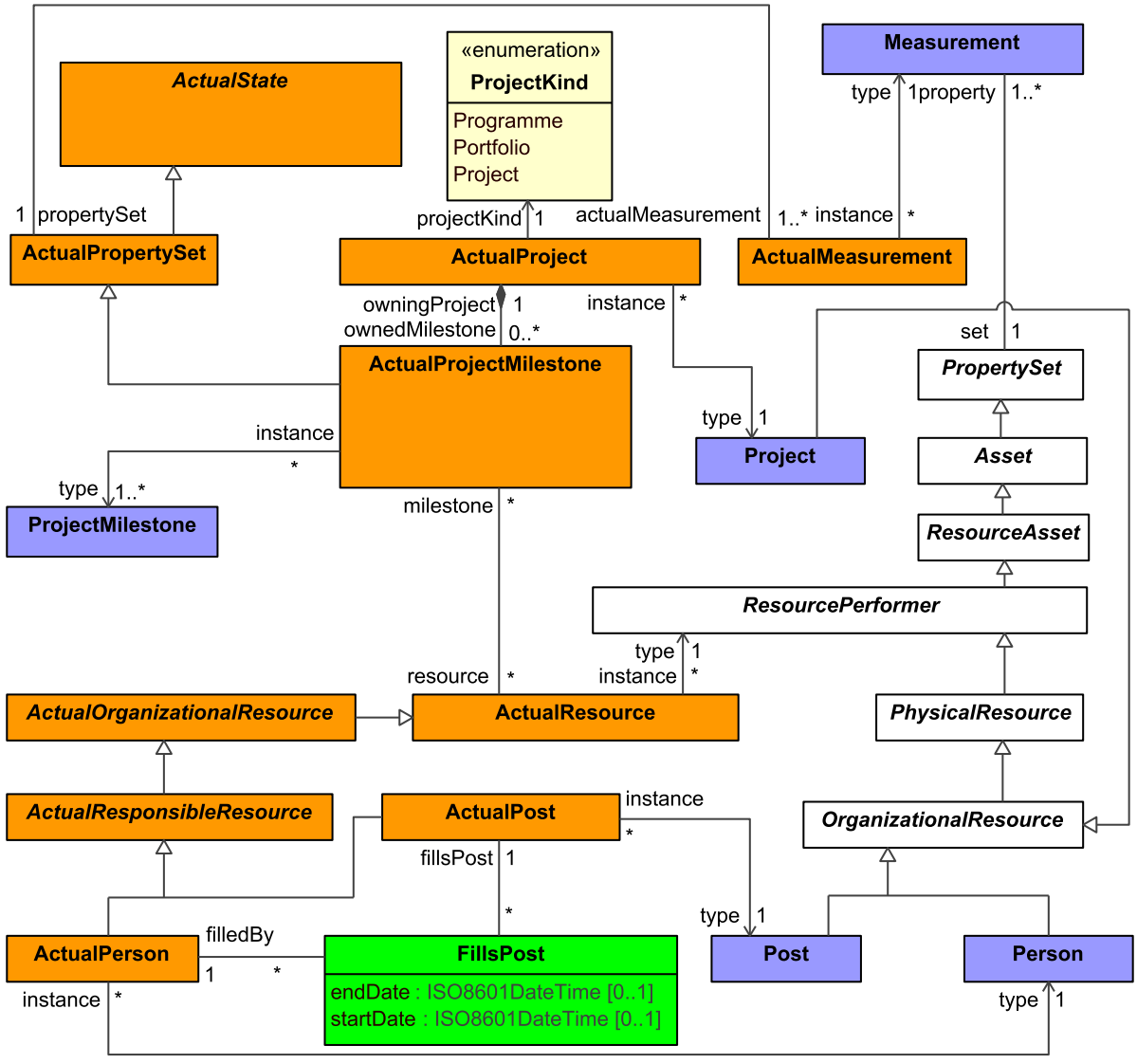

Figure 8:53 - Personnel Roadmap: Availability

Elements

- ActualMeasurement
- ActualOrganizationalResource
- ActualPerson
- ActualPost
- ActualProject
- ActualProjectMilestone
- ActualPropertySet
- ActualResource
- ActualResponsibleResource
- ActualState
- Asset
- FillsPost
- Measurement
- OrganizationalResource
- Person
- PhysicalResource
- Post
- Project
- ProjectMilestone
- PropertySet
- ResourceAsset
- ResourcePerformer

8.1.6.8.2 View Specifications::Personnel::Roadmap::Personnel Roadmap: Evolution

Stakeholders: Human resources, Solution Providers.

Concerns: organizational structure changes over time.

Definition: provides an overview of how an organizational structure changes over time. It shows the structure of several organizational structures mapped against a timeline.

Recommended Implementation: timeline, SysML Block Definition Diagram, SysML Internal Block Diagram.

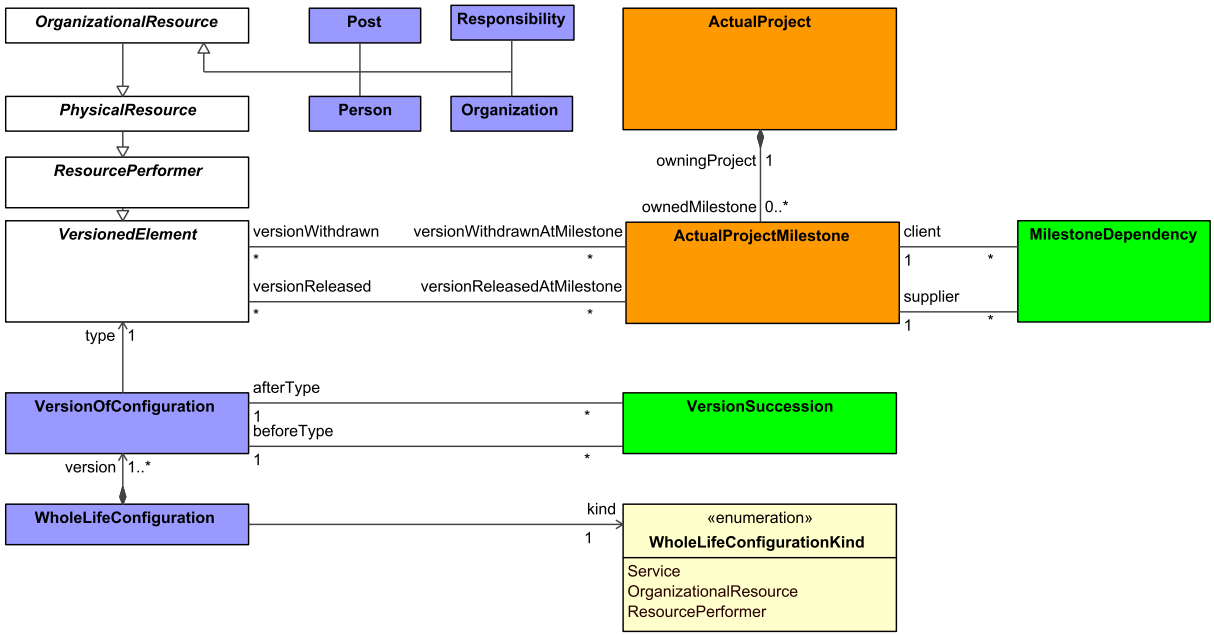

Figure 8:54 - Personnel Roadmap: Evolution

Elements

- ActualProject
- ActualProjectMilestone
- MilestoneDependency
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- ResourcePerformer
- Responsibility
- VersionedElement
- VersionOfConfiguration
- VersionSuccession
- WholeLifeConfiguration

8.1.6.8.3 View Specifications::Personnel::Roadmap::Personnel Roadmap: Forecast

Stakeholders: Human resources, Logisticians, Solution Providers.

Concerns: competencies and skills forecast.

Definition: defines the underlying current and expected supporting competencies and skills of organizational resources.

Recommended Implementation: timeline, tabular format, SysML Block Definition Diagram.

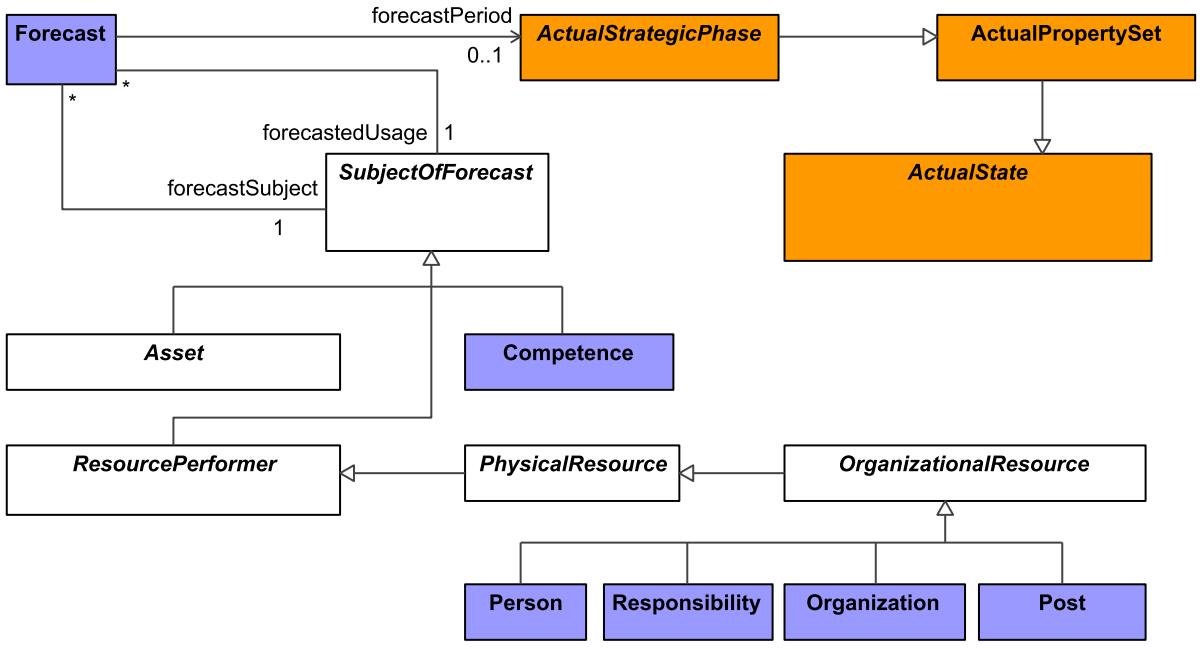

Figure 8:55 - Personnel Roadmap: Forecast

Elements

- ActualPropertySet
- ActualState
- ActualStrategicPhase

- Asset
- Competence
- Forecast
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- ResourcePerformer
- Responsibility
- SubjectOfForecast

8.1.6.9 View Specifications::Personnel::Traceability

Contains the diagrams that document the Personnel Traceability View Specification.

8.1.6.9.1 View Specifications::Personnel::Traceability::Personnel Traceability

Stakeholders: Systems Engineers, Enterprise Architects, Solution Providers, Business Architects.

Concerns: traceability between operational activities and functions that implements them.

Definition: depicts the mapping of functions (performed by organizational resources) to operational activities and thus identifies the transformation of an operational need into a purposeful function performed by an organizational resource or solution.

Recommended Implementation: Matrix format, SysML Block Definition Diagram.

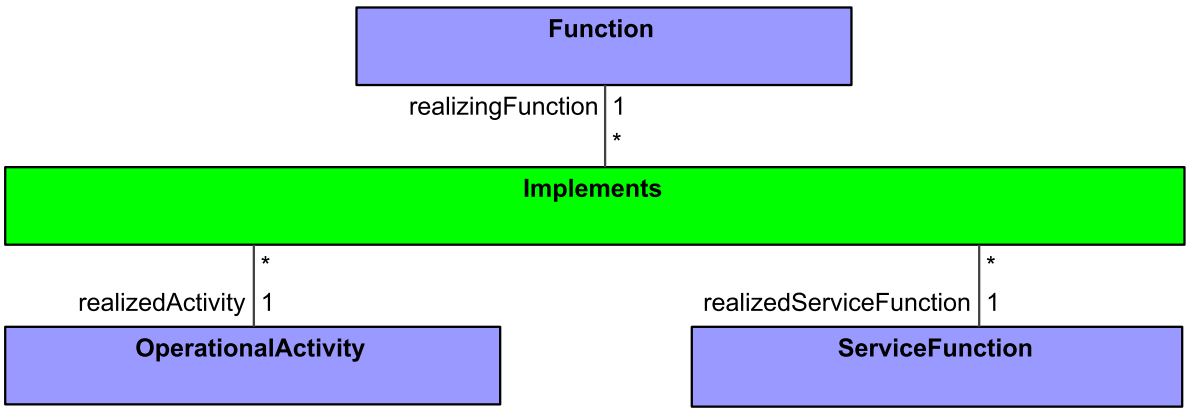

Figure 8:56 - Personnel Traceability

Elements

- Function
- Implements
- OperationalActivity
- ServiceFunction

8.1.7 View Specifications::Resources

Stakeholders: Systems Engineers, Resource Owners, Implementers, Solution Providers, IT Architects.

Concerns: definition of solution architectures to implement operational requirements.

Definition: captures a solution architecture consisting of resources, e.g. organizational, software, artifacts, capability configurations, natural resources that implement the operational requirements. Further design of a resource is typically detailed in SysML or UML.

8.1.7.1 View Specifications::Resources::Taxonomy

Contains the diagrams that document the Resources Taxonomy View Specification.

8.1.7.1.1 View Specifications::Resources::Taxonomy::Resources Taxonomy

Stakeholders: Solution Providers, Systems Engineers, IT Architects, Implementers.

Concerns: resource types.

Definition: shows the taxonomy of types of resources.

Recommended Implementation: SysML Block Definition Diagram.

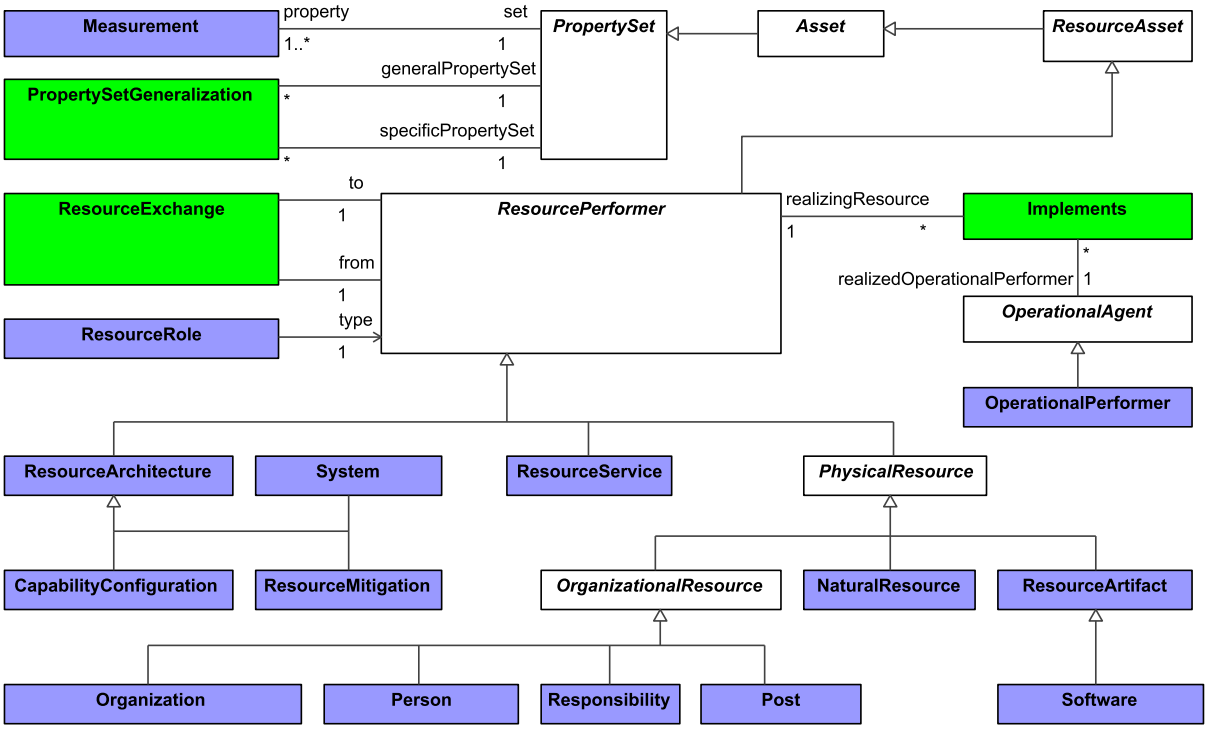

Figure 8:57 - Resources Taxonomy

Elements

- Asset
- CapabilityConfiguration
- Implements
- Measurement
- NaturalResource
- OperationalAgent
- OperationalPerformer
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- PropertySet

- PropertySetGeneralization
- ResourceArchitecture
- ResourceArtifact
- ResourceAsset
- ResourceExchange
- ResourceMitigation
- ResourcePerformer
- ResourceRole
- ResourceService
- Responsibility
- Software
- System

8.1.7.2 View Specifications::Resources::Structure

Contains the diagrams that document the Resources Structure View Specification.

8.1.7.2.1 View Specifications::Resources::Structure::Resources Structure

Stakeholders: Systems Engineers, Resource Owners, Implementers, Solution Providers.

Concerns: reference the resource structure, connectors and interfaces in a specific context.

Definition: defines the physical resources, e.g. capability configuration(s)/system(s) and interactions necessary to implement a specific set of OperationalPerformer(s). Can be used to represent communications networks and pathways that link communications resources and provides details regarding their configuration.

Recommended Implementation: SysML Internal Block Diagram, SysML Bock Definition Diagram.

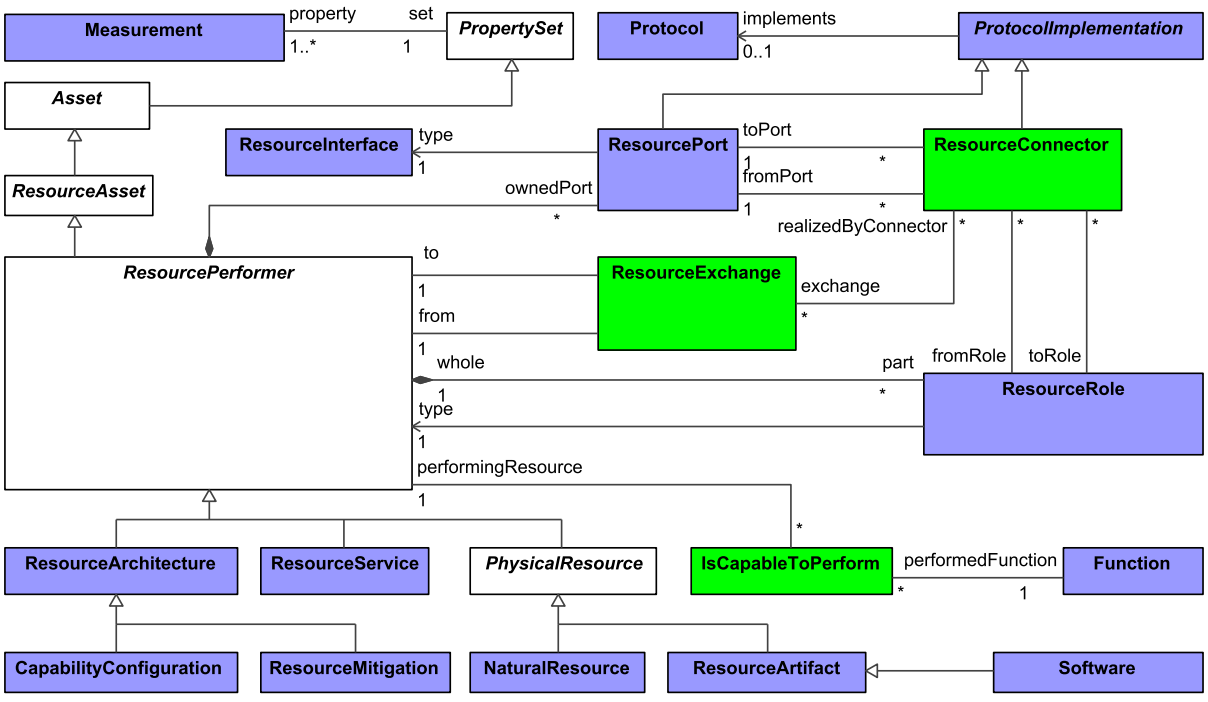

Figure 8:58 - Resources Structure

Elements

- Asset
- CapabilityConfiguration
- Function

- IsCapableToPerform
- Measurement
- NaturalResource
- PhysicalResource
- PropertySet
- Protocol
- ProtocolImplementation
- ResourceArchitecture
- ResourceArtifact
- ResourceAsset
- ResourceConnector
- ResourceExchange
- ResourceInterface
- ResourceMitigation
- ResourcePerformer
- ResourcePort
- ResourceRole
- ResourceService
- Software

8.1.7.3 View Specifications::Resources::Connectivity

Contains the diagrams that document the Resources Connectivity View Specification.

8.1.7.3.1 View Specifications::Resources::Connectivity::Resources Connectivity

Stakeholders: Systems Engineers, IT Architects, Solution Providers, Implementers.

Concerns: capture the interactions between resources.

Definition: summarizes interactions between resources of information, systems, personnel, natural resources etc. and the functions that produce and consume them. Measurements can optionally be included.

Recommended Implementation: SysML Internal Block Diagram, tabular format..

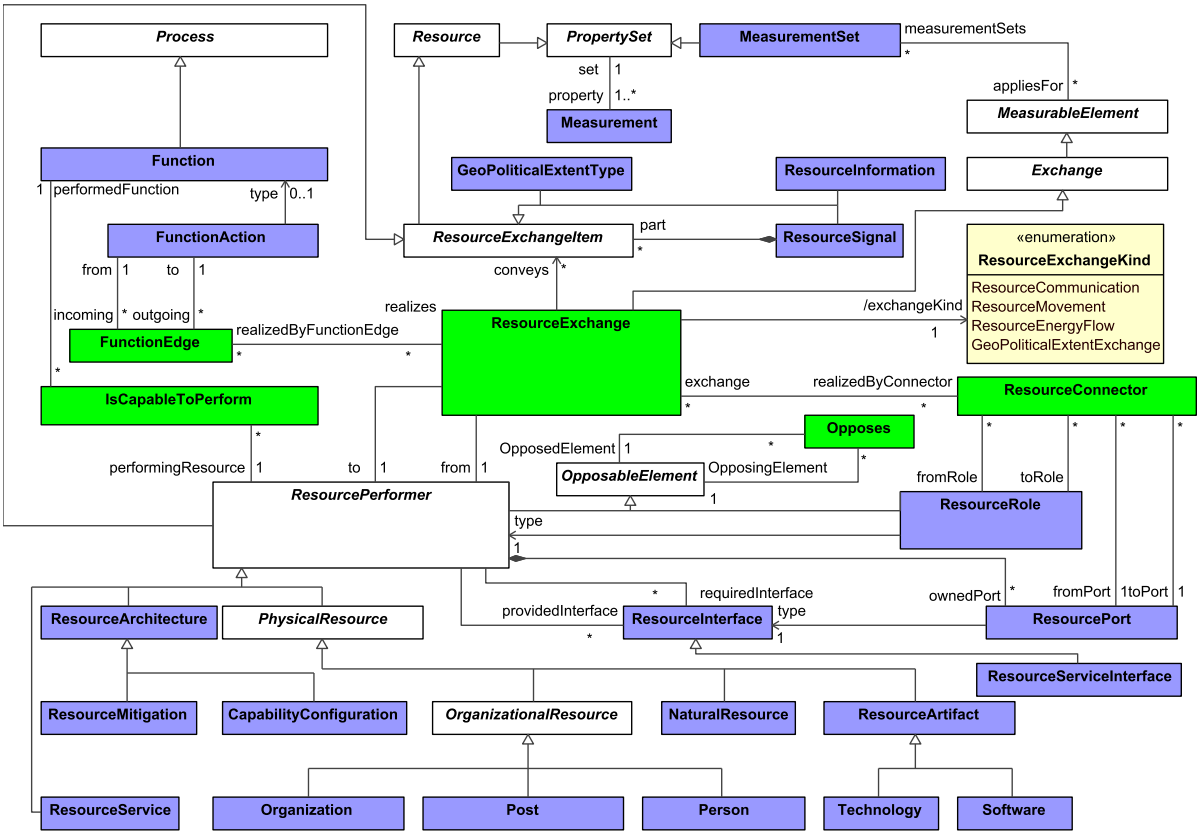

Figure 8:59 - Resources Connectivity

Elements

- CapabilityConfiguration
- Exchange
- Function
- FunctionAction
- FunctionEdge
- GeoPoliticalExtentType
- IsCapableToPerform
- MeasurableElement
- Measurement
- MeasurementSet
- NaturalResource
- OpposableElement
- Opposes
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- Process
- PropertySet
- Resource
- ResourceArchitecture
- ResourceArtifact
- ResourceConnector

- ResourceExchange
- ResourceExchangeItem
- ResourceInformation
- ResourceInterface
- ResourceMitigation
- ResourcePerformer
- ResourcePort
- ResourceRole
- ResourceService
- ResourceServiceInterface
- ResourceSignal
- Software
- Technology

8.1.7.4 View Specifications::Resources::Processes

Contains the diagrams that document the Resources Processes View Specification.

8.1.7.4.1 View Specifications::Resources::Processes::Resources Processes

Stakeholders: Solution Providers, Systems Engineers, IT Architects.

Concerns: captures activity based behavior and flows.

Definition: describes the functions that are normally conducted in the course of implementing operational activity(ies) in support of capability(ies). It describes the functions, their Inputs/Outputs, function actions and flows between them.

Recommended Implementation: SysML Activity Diagram, SysML Block Definition Diagram.

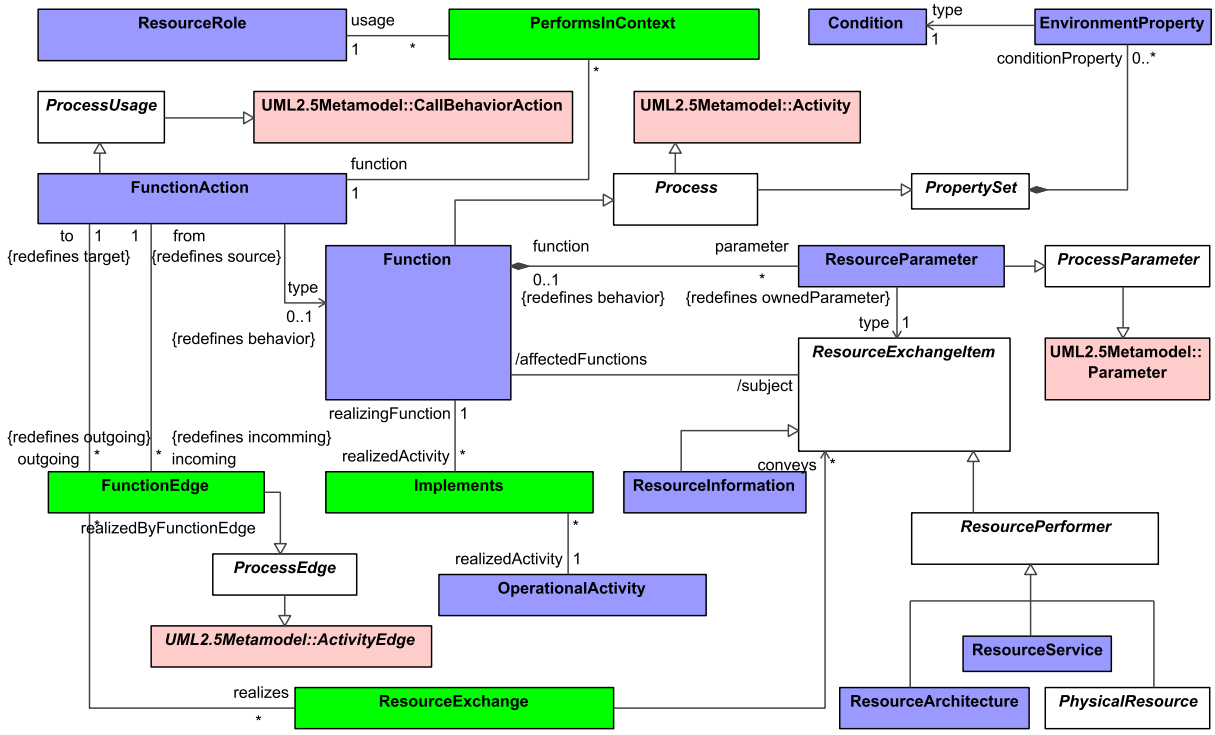

Figure 8:60 - Resources Processes

Elements

- Condition
- EnvironmentProperty
- Function
- FunctionAction
- FunctionEdge
- Implements
- OperationalActivity
- PerformsInContext
- PhysicalResource
- Process
- ProcessEdge
- ProcessParameter
- ProcessUsage
- PropertySet
- ResourceArchitecture
- ResourceExchange
- ResourceExchangeItem
- ResourceInformation
- ResourceParameter
- ResourcePerformer
- ResourceRole
- ResourceService
- UML2.5Metamodel::Activity
- UML2.5Metamodel::ActivityEdge
- UML2.5Metamodel::CallBehaviorAction
- UML2.5Metamodel::Parameter

8.1.7.4.2 View Specifications::Resources::Processes::Resources Processes BPMN Semantics

Stakeholders: Solution Providers, IT Architects.

Concerns: captures activity based behavior and flows using BPMN.

Definition: describes the functions that are normally conducted in the course of implementing operational activity(ies) in support of capability(ies). It describes the functions, their Inputs/Outputs, function actions and flows between them using BPMN.

Recommended Implementation: BPMN Process Diagram.

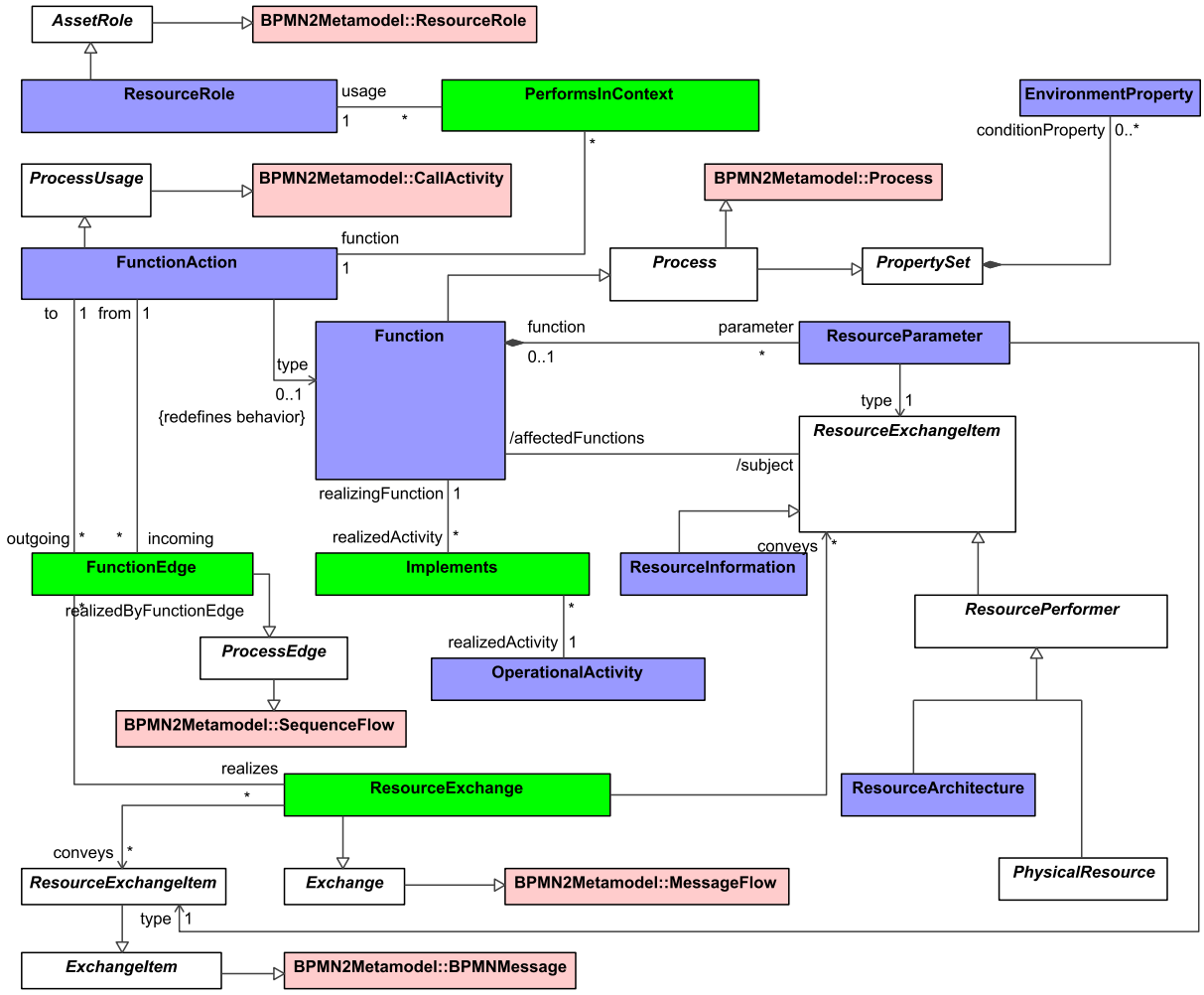

Figure 8:61 - Resources Processes BPMN Semantics

Elements

- AssetRole
- BPMN2Metamodel::BPMNMessage
- BPMN2Metamodel::CallActivity
- BPMN2Metamodel::MessageFlow
- BPMN2Metamodel::Process
- BPMN2Metamodel::ResourceRole
- BPMN2Metamodel::SequenceFlow
- EnvironmentProperty
- Exchange
- ExchangeItem
- Function
- FunctionAction
- FunctionEdge
- Implements
- OperationalActivity
- PerformsInContext
- PhysicalResource
- Process
- ProcessEdge

- ProcessUsage
- PropertySet
- ResourceArchitecture
- ResourceExchange
- ResourceExchangeItem
- ResourceInformation
- ResourceParameter
- ResourcePerformer
- ResourceRole

8.1.7.4.3 View Specifications::Resources::Processes::Resources Processes: Mission Engineering Threads

Stakeholders: Solution Providers, Systems Engineers, IT Architects.

Concerns: captures activity based behavior and flows.

Definition: describes the functions that are normally conducted in the course of implementing operational activity(ies) in support of capability(ies). It describes the functions, their Inputs/Outputs, function actions and flows between them.

Recommended Implementation: SysML Activity Diagram, SysML Block Definition Diagram.

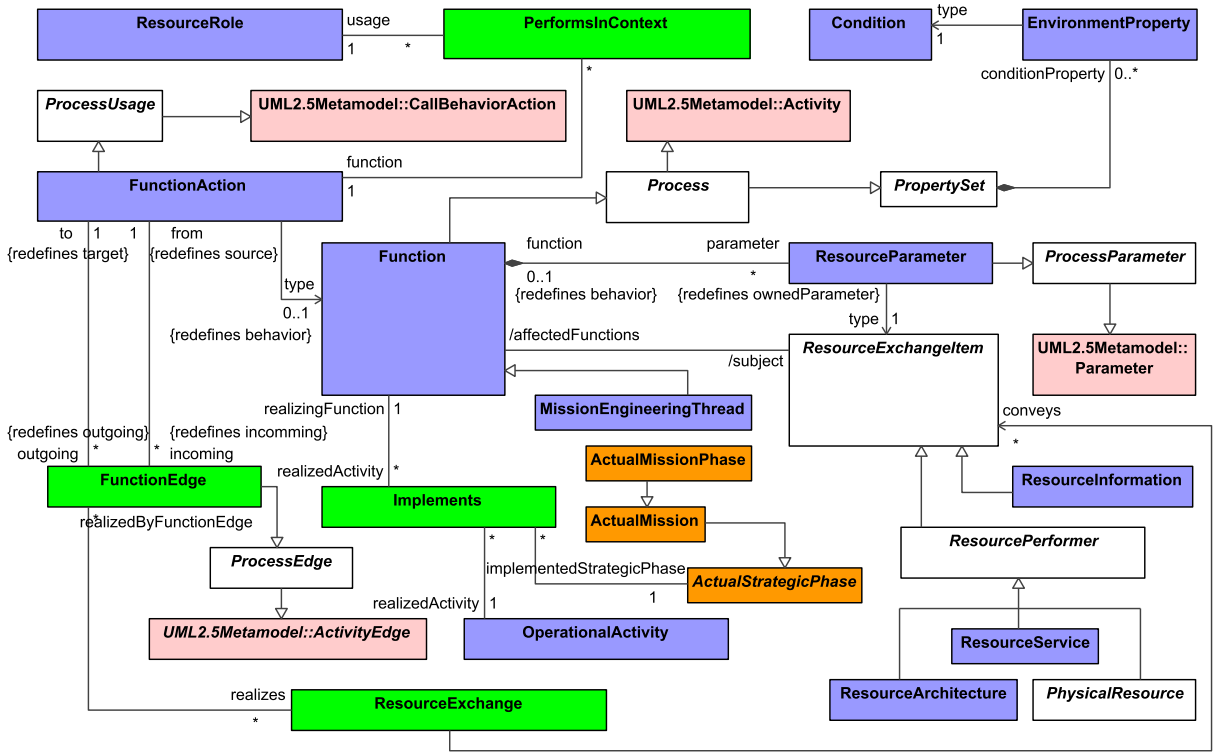

Figure 8:62 - Resources Processes: Mission Engineering Threads

Elements

- ActualMission
- ActualMissionPhase
- ActualStrategicPhase
- Condition
- EnvironmentProperty
- Function

- FunctionAction
- FunctionEdge
- Implements
- MissionEngineeringThread
- OperationalActivity
- PerformsInContext
- PhysicalResource
- Process
- ProcessEdge
- ProcessParameter
- ProcessUsage
- PropertySet
- ResourceArchitecture
- ResourceExchange
- ResourceExchangeItem
- ResourceInformation
- ResourceParameter
- ResourcePerformer
- ResourceRole
- ResourceService
- UML2.5Metamodel::Activity
- UML2.5Metamodel::ActivityEdge
- UML2.5Metamodel::CallBehaviorAction
- UML2.5Metamodel::Parameter

8.1.7.5 View Specifications::Resources::States

Contains the diagrams that document the Resources States View Specification.

8.1.7.5.1 View Specifications::Resources::States::Resources States

Stakeholders: Systems Engineers, Software Engineers.

Concerns: capture state-based behavior of a resource.

Definition: it is a graphical representation of states of a resource and how that resource responds to various events and actions.

Recommended Implementation: SysML State Machine Diagram.

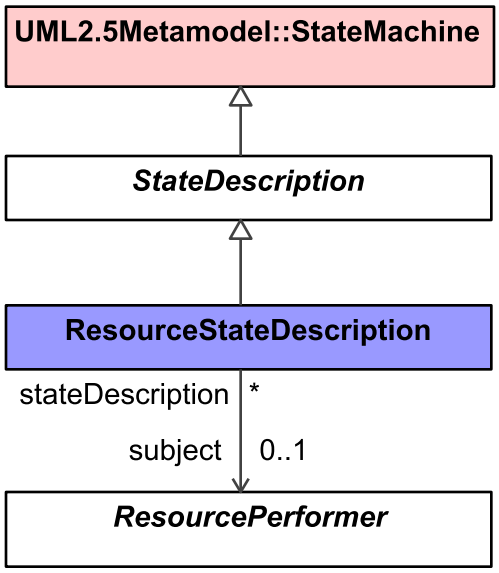

Figure 8:63 - Resources States

Elements

- ResourcePerformer
- ResourceStateDescription
- StateDescription
- UML2.5Metamodel::StateMachine

8.1.7.6 View Specifications::Resources::Sequences

Contains the diagrams that document the Resources Sequences View Specification.

8.1.7.6.1 View Specifications::Resources::Sequences::Resources Sequences

Stakeholders: Software Engineers, Systems Engineers.

Concerns: interactions between resources (roles).

Definition: provides a time-ordered examination of the interactions between resources.

Recommended Implementation: SysML Sequence Diagram.

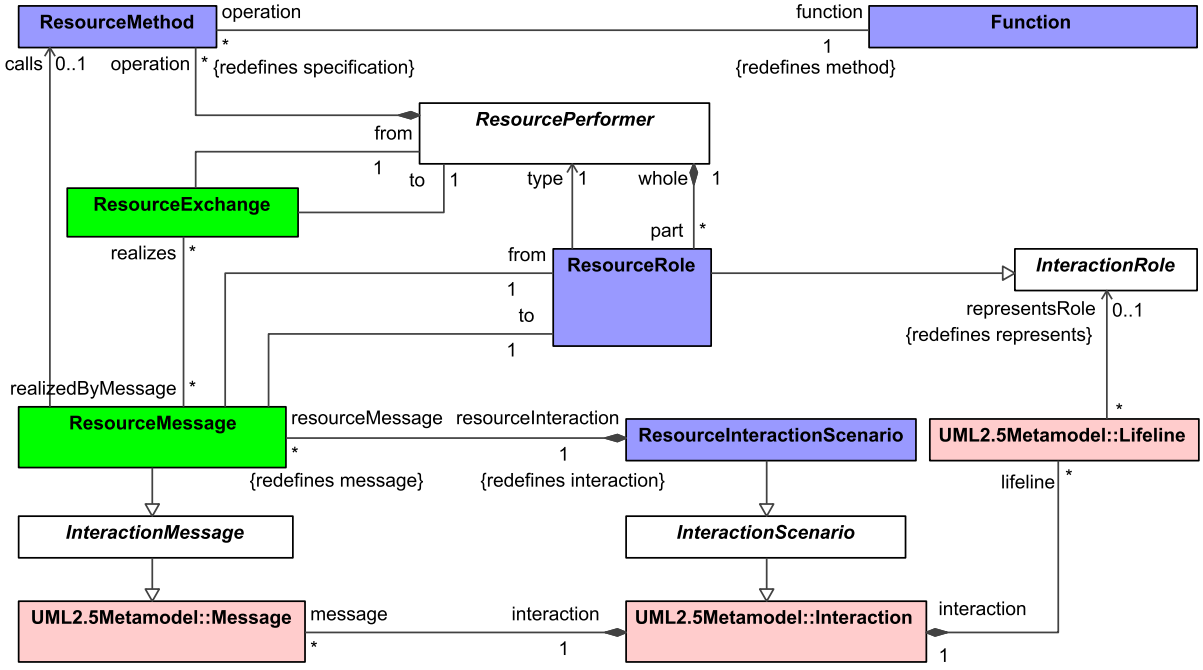

Figure 8:64 - Resources Sequences

Elements

- Function
- InteractionMessage
- InteractionRole
- InteractionScenario
- ResourceExchange
- ResourceInteractionScenario
- ResourceMessage
- ResourceMethod
- ResourcePerformer
- ResourceRole
- UML2.5Metamodel::Interaction
- UML2.5Metamodel::Lifeline
- UML2.5Metamodel::Message

8.1.7.7 View Specifications::Resources::Constraints

Contains the diagrams that document the Resources Constraints View Specification.

8.1.7.7.1 View Specifications::Resources::Constraints::Resources Constraints

Stakeholders: Systems Engineers, IT Architects, Solution Providers, Implementers.

Concerns: define limitations, constraints and performance parameters for resources, their interactions, performed functions, and data.

Definition: specifies traditional textual rules/non-functional requirements that are constraints on resources, their interactions, performed functions, and data. The addition of SysML parametrics provide a computational means of defining resource constraints within a specific context.

Recommended Implementation: tabular format, SysML Block Definition Diagram, SysML Parametric Diagram, OCL.

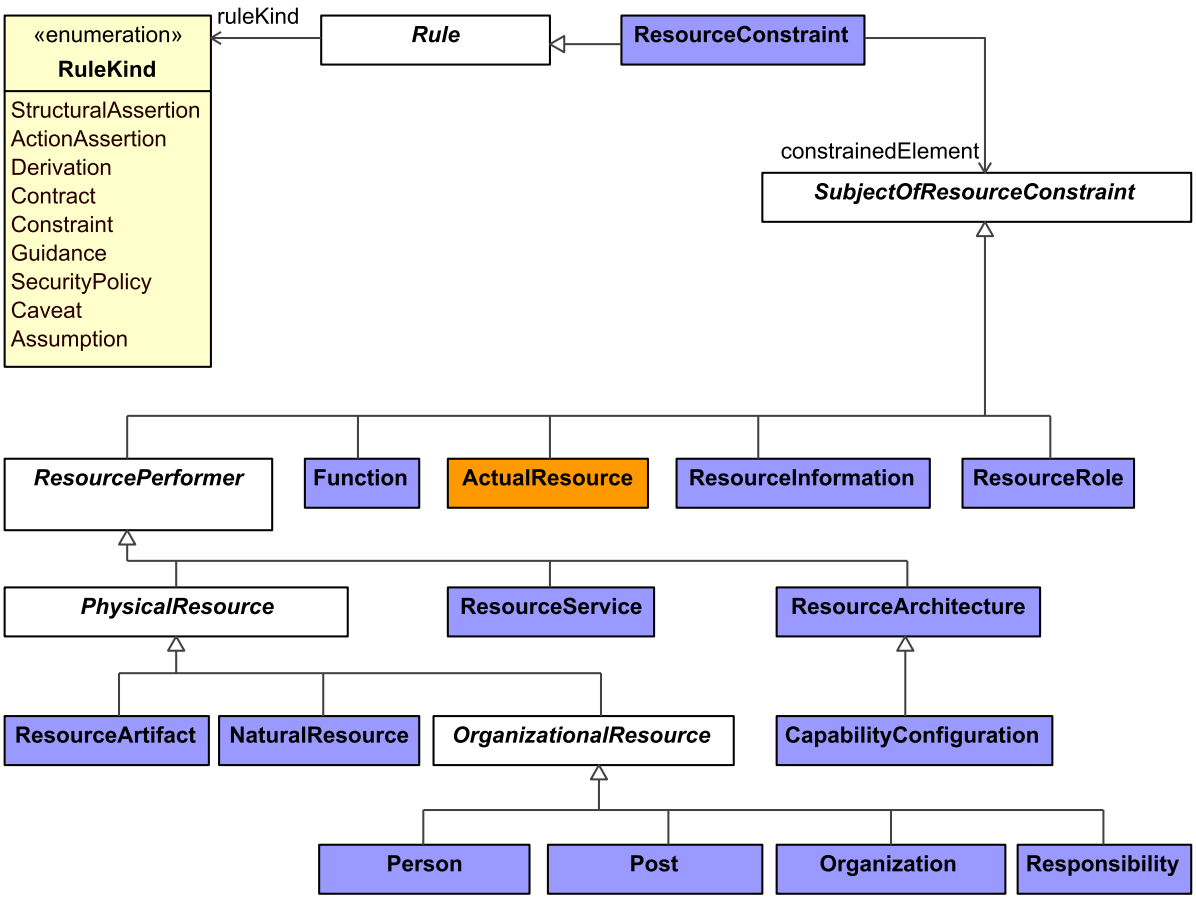

Figure 8:65 - Resources Constraints

Elements

- ActualResource
- CapabilityConfiguration
- Function
- NaturalResource
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- ResourceArchitecture
- ResourceArtifact
- ResourceConstraint
- ResourceInformation
- ResourcePerformer
- ResourceRole
- ResourceService
- Responsibility
- Rule
- SubjectOfResourceConstraint

8.1.7.8 View Specifications::Resources::Roadmap

Contains the diagrams that document the Resources Roadmap View Specification.

8.1.7.8.1 View Specifications::Resources::Roadmap::Resources Roadmap: Evolution

Stakeholders: Systems Engineers, IT Architects, Solution Providers, Implements.

Concerns: resource structure changes over time.

Definition: provides an overview of how a resource structure changes over time. It shows the structure of several resources mapped against a timeline.

Recommended Implementation: timeline, SysML Block Definition Diagram, SysML Internal Block Diagram.

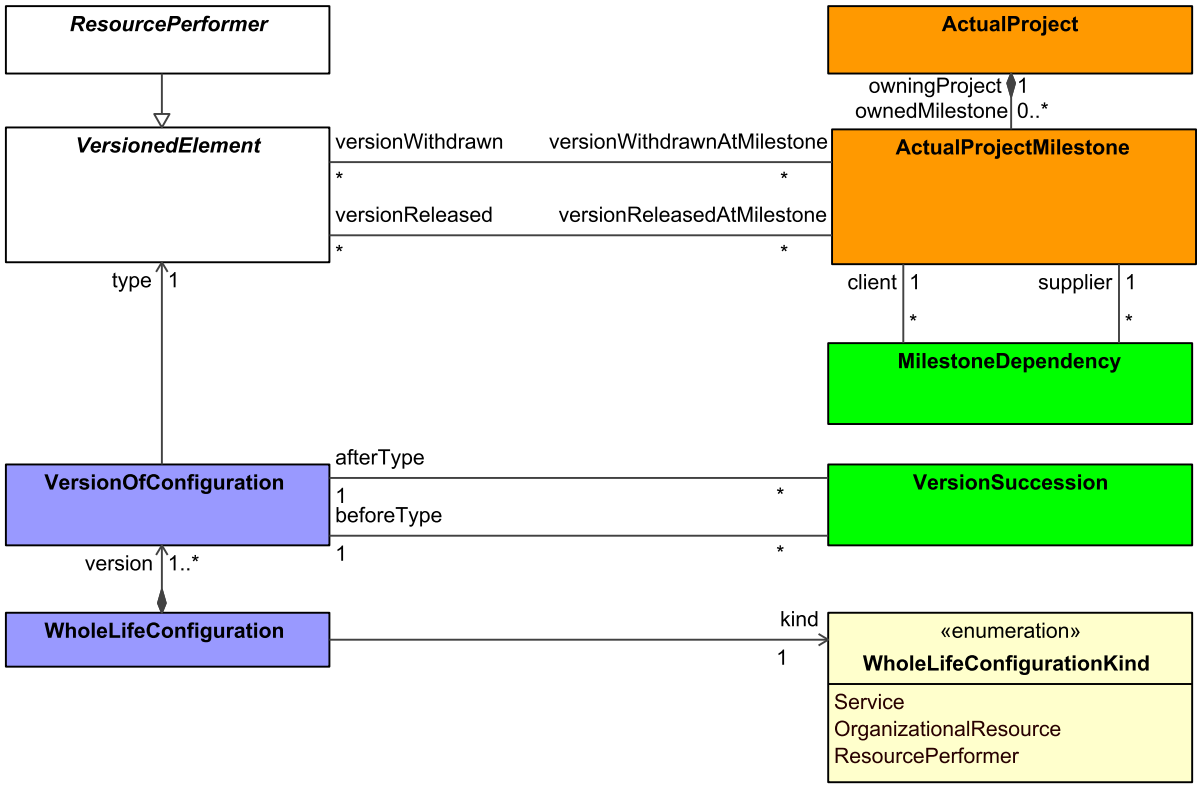

Figure 8:66 - Resources Roadmap: Evolution

Elements

- ActualProject
- ActualProjectMilestone
- MilestoneDependency
- ResourcePerformer
- VersionedElement
- VersionOfConfiguration
- VersionSuccession
- WholeLifeConfiguration

8.1.7.8.2 View Specifications::Resources::Roadmap::Resources Roadmap: Forecast

Stakeholders: Solution Providers, Systems Engineers, IT Architects.

Concerns: technology forecast.

Definition: defines the underlying current and expected supporting technologies. Expected supporting technologies are those that can be reasonably forecast given the current state of technology, and expected improvements / trends.

Recommended Implementation: timeline, tabular format, SysML Block Definition Diagram.

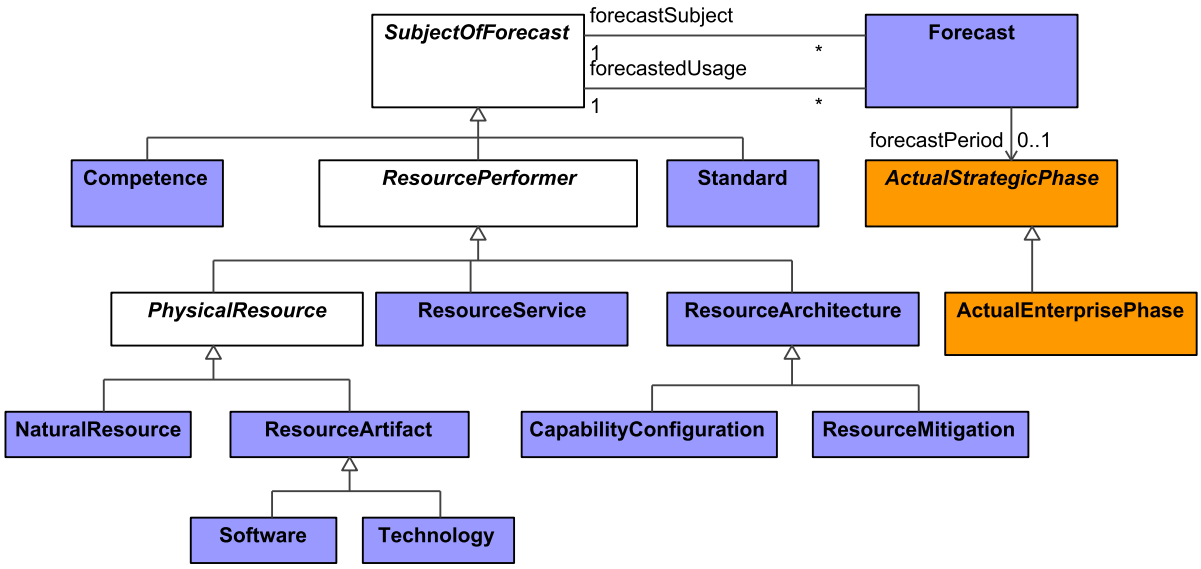

Figure 8:67 - Resources Roadmap: Forecast

Elements

- ActualEnterprisePhase
- ActualStrategicPhase
- CapabilityConfiguration
- Competence
- Forecast
- NaturalResource
- PhysicalResource
- ResourceArchitecture
- ResourceArtifact
- ResourceMitigation
- ResourcePerformer
- ResourceService
- Software
- Standard
- SubjectOfForecast
- Technology

8.1.7.9 View Specifications::Resources::Traceability

Contains the diagrams that document the Resources Traceability View Specification.

8.1.7.9.1 View Specifications::Resources::Traceability::Resources Traceability

Stakeholders: Systems Engineers, Enterprise Architects, Solution Providers, Business Architects.

Concerns: traceability between operational activities and functions that implements them.

Definition: depicts the mapping of functions to operational activities and thus identifies the transformation of an operational need into a purposeful function performed by a resource or solution.

Recommended Implementation: Matrix format, SysML Block Definition Diagram.

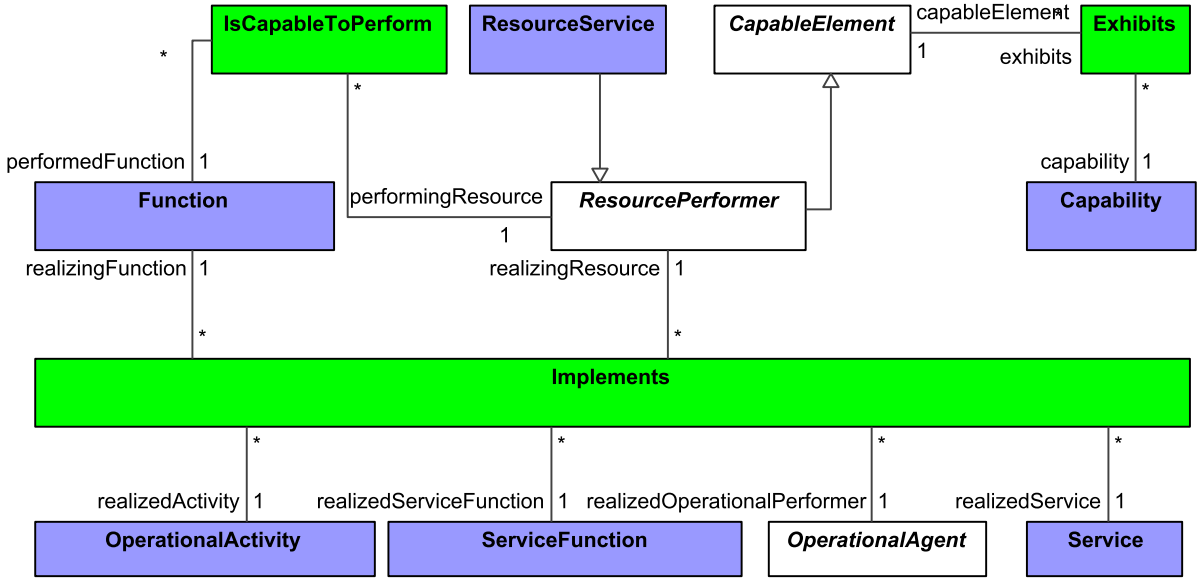

Figure 8:68 - Resources Traceability

Elements

- Capability
- CapableElement
- Exhibits
- Function
- Implements
- IsCapableToPerform
- OperationalActivity
- OperationalAgent
- ResourcePerformer
- ResourceService
- Service
- ServiceFunction

8.1.8 View Specifications::Security

Stakeholders: Security Architects, Security Engineers. Systems Engineers, Operational Architects.

Concerns: addresses the security constraints and information assurance attributes that exist on exchanges between resources and OperationalPerformers

Definition: illustrates the security assets, security constraints, security controls, families, and measures required to address specific security concerns.

8.1.8.1 View Specifications::Security::Motivation

Contains the diagrams that document the Security Motivation View Specification.

8.1.8.1.1 View Specifications::Security::Motivation::Security Controls

Stakeholders: Security Architects, Security Engineers, Risk Analysts.

Concerns: security controls, security control families, and overlays.

Definition: identifies security controls to mitigate against the security risks.

Recommended Implementation: tabular or Matrix format, SysML Block Definition Diagram.

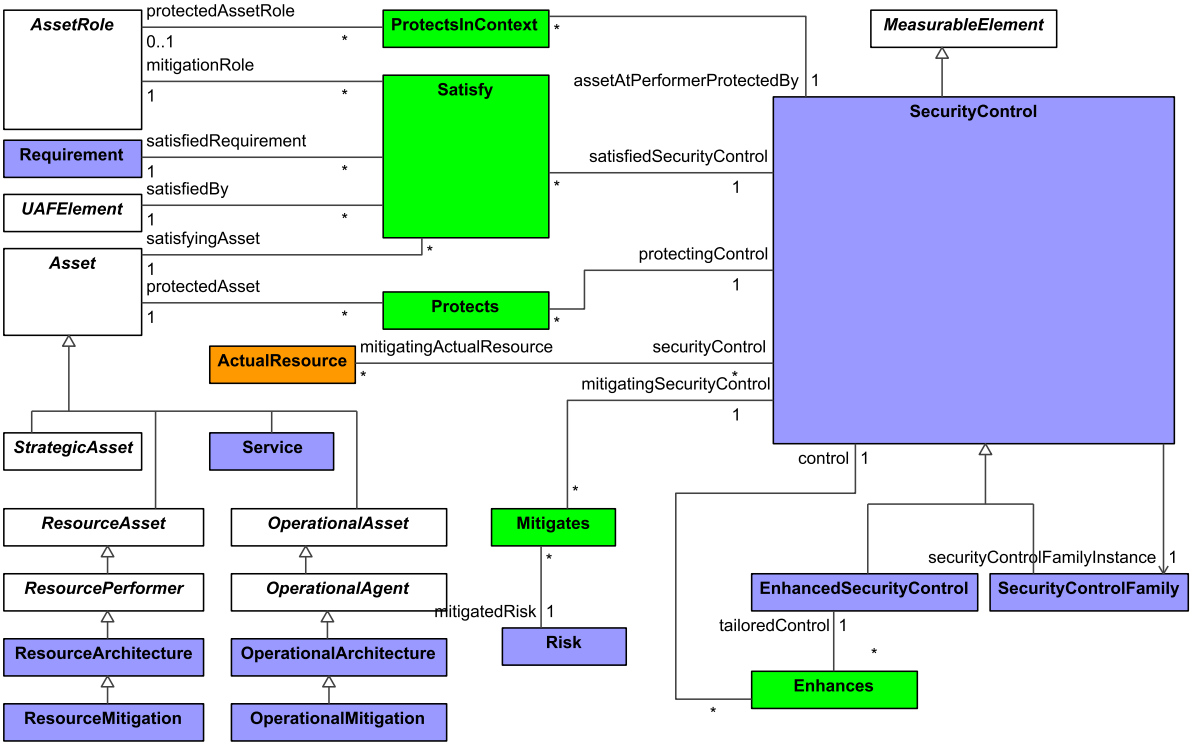

Figure 8:69 - Security Controls

Elements

- ActualResource
- Asset
- AssetRole
- EnhancedSecurityControl
- Enhances
- MeasurableElement
- Mitigates
- OperationalAgent
- OperationalArchitecture
- OperationalAsset
- OperationalMitigation
- Protects
- ProtectsInContext
- Requirement
- ResourceArchitecture
- ResourceAsset
- ResourceMitigation
- ResourcePerformer
- Risk
- Satisfy
- SecurityControl
- SecurityControlFamily
- Service
- StrategicAsset
- UAFElement

8.1.8.2 View Specifications::Security::Taxonomy

Contains the diagrams that document the Security Taxonomy View Specification.

8.1.8.2.1 View Specifications::Security::Taxonomy::Security Taxonomy

Stakeholders: Security Architects, Security Engineers.

Concerns: Security assets and security enclaves.

Definition: Defines the hierarchy of security assets and asset owners that are available to implement security, security constraints (policy, guidance, laws and regulations) and details where they are located (security enclaves).

Recommended Implementation: tabular format, SysML Block Definition Diagram.

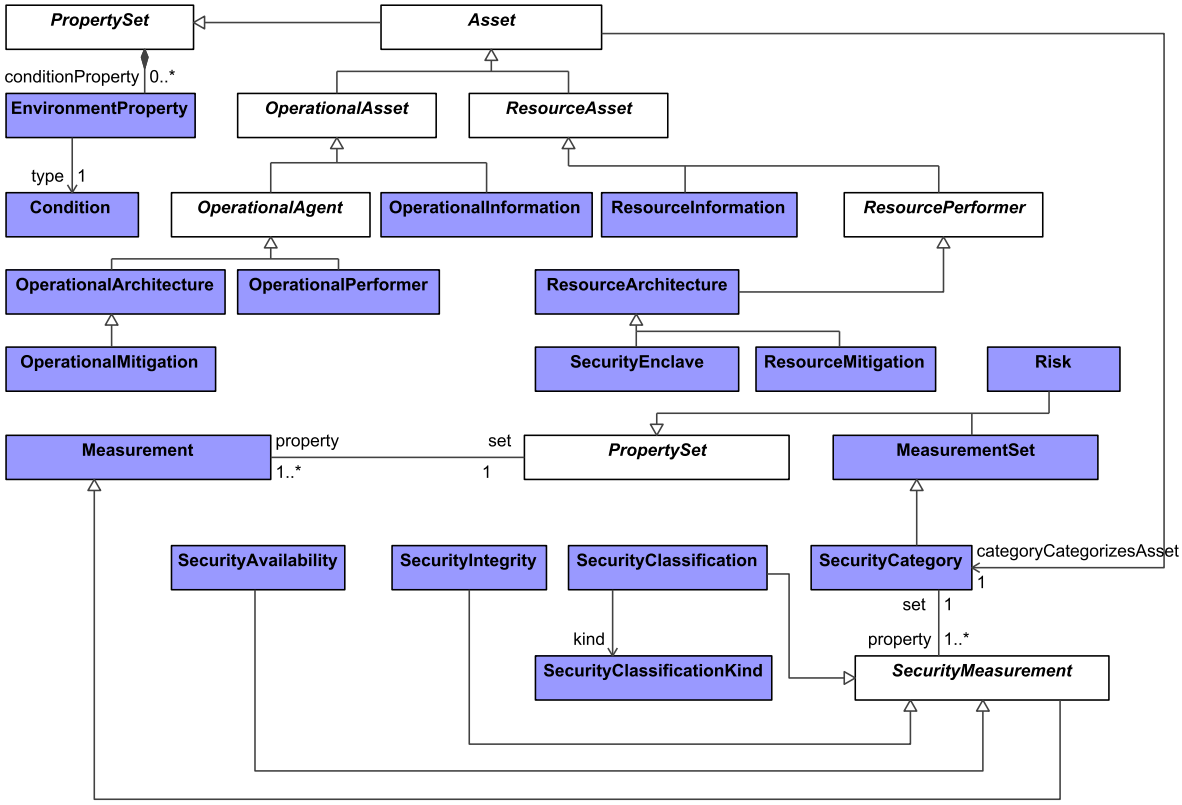

Figure 8:70 - Security Taxonomy

Elements

- Asset
- Condition
- EnvironmentProperty
- Measurement
- MeasurementSet
- OperationalAgent
- OperationalArchitecture
- OperationalAsset
- OperationalInformation
- OperationalMitigation
- OperationalPerformer
- PropertySet
- ResourceArchitecture

- ResourceAsset
- ResourceInformation
- ResourceMitigation
- ResourcePerformer
- Risk
- SecurityAvailability
- SecurityCategory
- SecurityClassification
- SecurityClassificationKind
- SecurityEnclave
- SecurityIntegrity
- SecurityMeasurement

8.1.8.3 View Specifications::Security::Structure

Contains the diagrams that document the Security Structure View Specification.

8.1.8.3.1 View Specifications::Security::Structure::Security Structure

Stakeholders: Security Architects, Security Engineers.

Concerns: The structure of security information and where it is used at the operational and resource level.

Definition: Captures the allocation of assets (operational and resource, information and data) across the security enclaves, shows applicable security controls necessary to protect organizations, systems and information during processing, while in storage (bdd), and during transmission (flows on an ibd). This view also captures Asset Aggregation and allocates the usage of the aggregated information at a location through the use of the SecurityProperty.

Recommended Implementation: SysML Internal Block Diagram, SysML Block Definition Diagram.

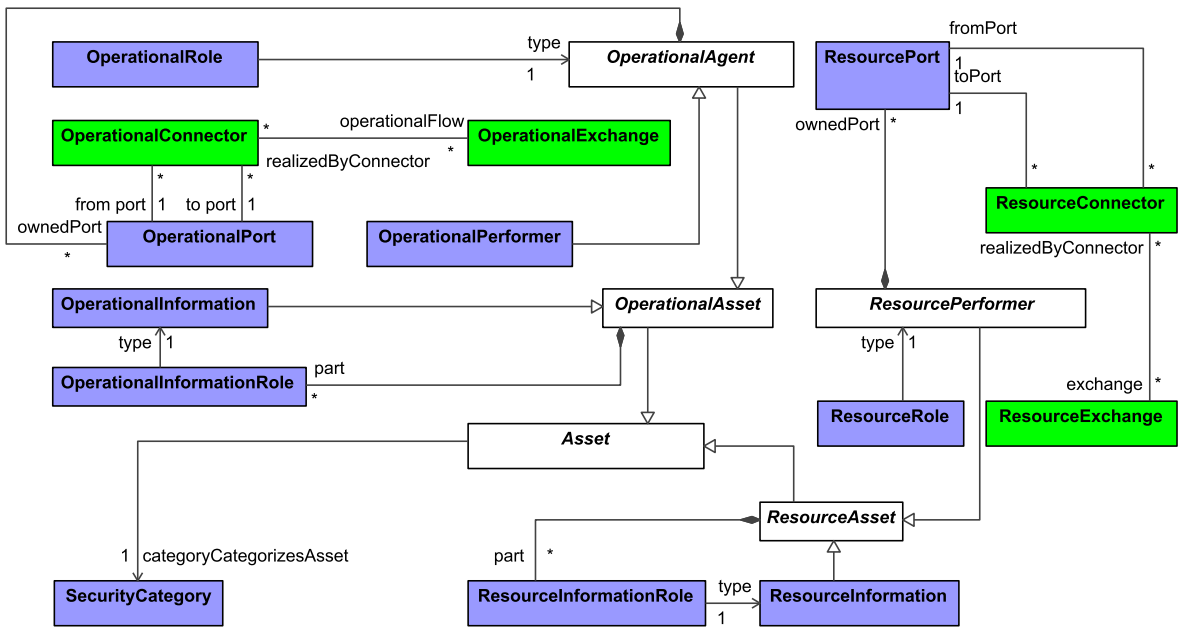

Figure 8:71 - Security Structure

Elements

- Asset
- OperationalAgent
- OperationalAsset

- OperationalConnector
- OperationalExchange
- OperationalInformation
- OperationalInformationRole
- OperationalPerformer
- OperationalPort
- OperationalRole
- ResourceAsset
- ResourceConnector
- ResourceExchange
- ResourceInformation
- ResourceInformationRole
- ResourcePerformer
- ResourcePort
- ResourceRole
- SecurityCategory

8.1.8.4 View Specifications::Security::Connectivity

Contains the diagrams that document the Security Connectivity View Specification.

8.1.8.4.1 View Specifications::Security::Connectivity::Security Connectivity

Stakeholders: Security Architects, Security Engineers.

Concerns: Addresses the security constraints and information assurance attributes that exist on exchanges across resources and across performers.

Definition: Lists security exchanges across security assets; the applicable security controls; and the security enclaves that house the producers and consumers of the exchanges. Measurements can optionally be included.

Recommended Implementation: SysML Internal Block Diagram, tabular format.

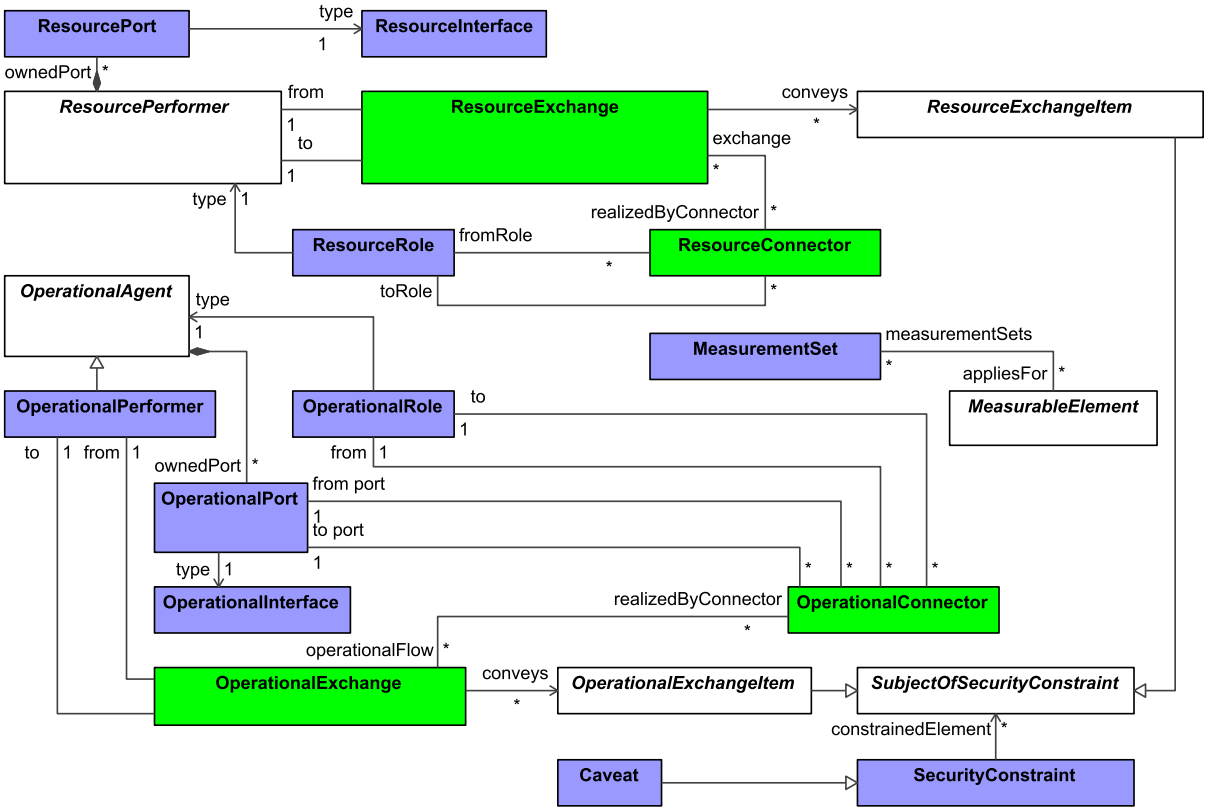

Figure 8:72 - Security Connectivity

Elements

- Caveat
- MeasurableElement
- MeasurementSet
- OperationalAgent
- OperationalConnector
- OperationalExchange
- OperationalExchangeItem
- OperationalInterface
- OperationalPerformer
- OperationalPort
- OperationalRole
- ResourceConnector
- ResourceExchange
- ResourceExchangeItem
- ResourceInterface
- ResourcePerformer
- ResourcePort
- ResourceRole
- SecurityConstraint
- SubjectOfSecurityConstraint

8.1.8.5 View Specifications::Security::Processes

Contains the diagrams that document the Security Processes View Specification.

8.1.8.5.1 View Specifications::Security::Processes::Security Processes

Stakeholders: Security Architects, Security Engineers.

Concerns: The specification of the Security Control families, security controls, and measures required to address a specific security baseline.

Definition: Provides a set of Security Controls and any possible enhancements as applicable to assets. The activity diagram describes operational or resource level processes that apply (operational level) or implement (resource level) security controls/enhancements to assets located in enclaves and across enclaves. This Security Process view can be instantiated either as a variant of an activity/flow diagram or as a hierarchical work breakdown structure.

Recommended Implementation: SysML Activity Diagram, SysML Block Definition Diagram, BPMN Process Diagram as described in Operational Processes and Resources Processes sections.

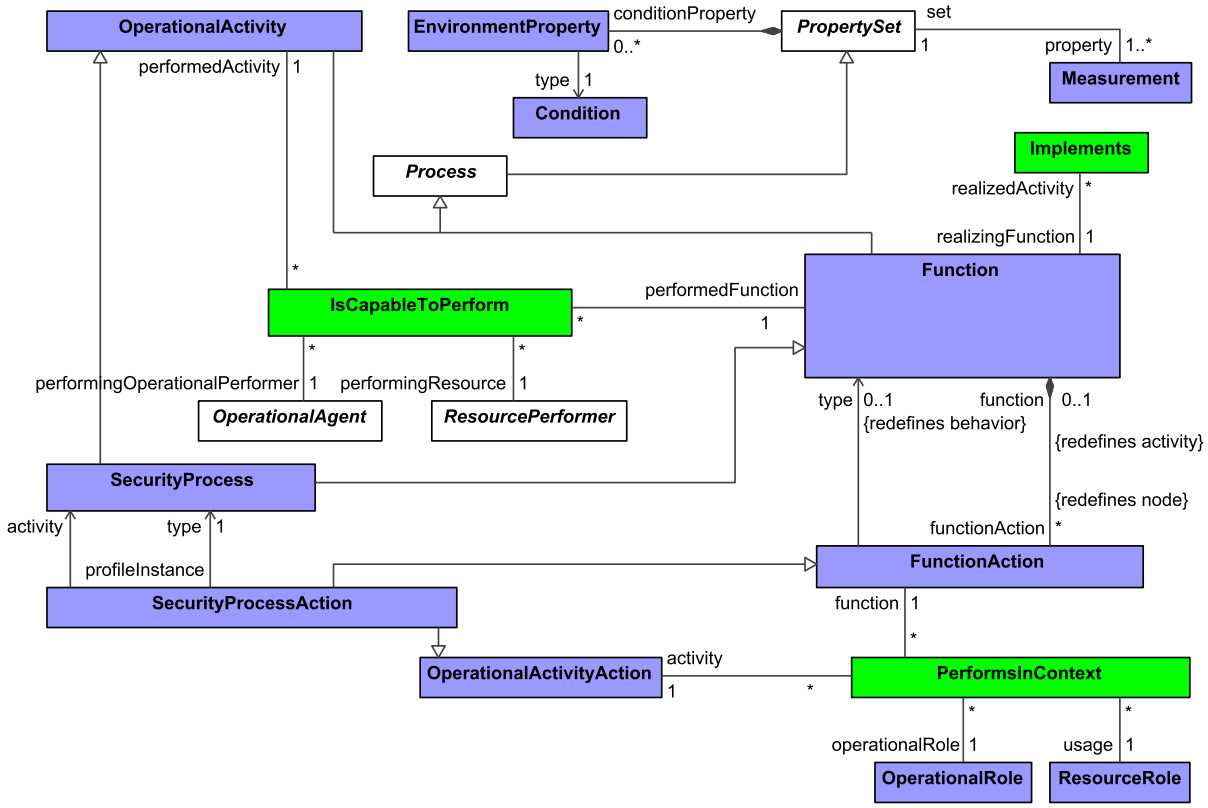

Figure 8:73 - Security Processes

Elements

- Condition
- EnvironmentProperty
- Function
- FunctionAction
- Implements
- IsCapableToPerform
- Measurement
- OperationalActivity
- OperationalActivityAction
- OperationalAgent
- OperationalRole
- PerformsInContext

- Process
- PropertySet
- ResourcePerformer
- ResourceRole
- SecurityProcess
- SecurityProcessAction

8.1.8.6 View Specifications::Security::Constraints

Contains the diagrams that document the Security Constraints View Specification.

8.1.8.6.1 View Specifications::Security::Constraints::Security Constraints

Stakeholders: Security Architects, Security Engineers, Risk Analysts.

Concerns: (i) Security-related policy, guidance, laws and regulations as applicable to assets, (ii) threats, vulnerabilities, and risk assessments as applicable to assets.

Definition: (i) Specifies textual rules/non-functional requirements that are security constraints on resources, information and data (e.g. security-related in the form of rules (e.g. access control policy). A common way of representing access control policy is through the use of XACML (eXtensible Access Control Markup Language), it is expected that implementations of UAF allow users to link security constraints to external files represented in XACML. (ii) Identifies risks, specifies risk likelihood, impact, asset criticality, other measurements and enables risk assessment.

Recommended Implementation: tabular or Matrix format, SysML Block Definition Diagram, SysML Parametric Diagram, or OCL.

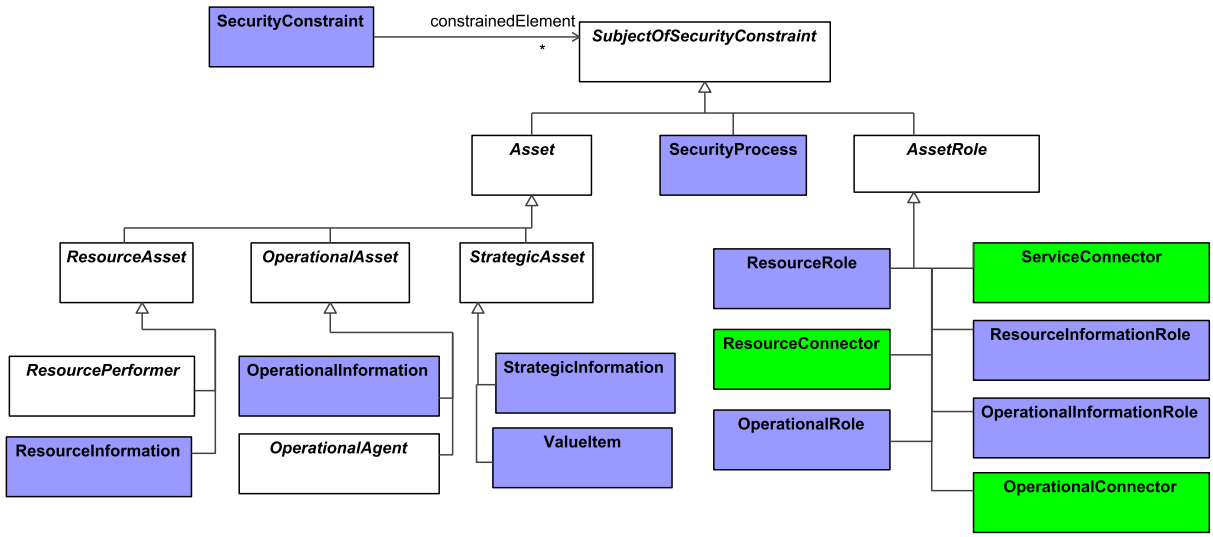

Figure 8:74 - Security Constraints

Elements

- Asset
- AssetRole
- OperationalAgent
- OperationalAsset
- OperationalConnector
- OperationalInformation
- OperationalInformationRole
- OperationalRole
- ResourceAsset

- ResourceConnector
- ResourceInformation
- ResourceInformationRole
- ResourcePerformer
- ResourceRole
- SecurityConstraint
- SecurityProcess
- ServiceConnector
- StrategicAsset
- StrategicInformation
- SubjectOfSecurityConstraint
- ValueItem

8.1.8.7 View Specifications::Security::Traceability

Contains the diagrams that document the Security Traceability View Specification.

8.1.8.7.1 View Specifications::Security::Traceability::Security Traceability

Stakeholders: Security Architects, Security Engineers, Risk Analysts.

Concerns: traceability between risk and risk owner, risk mitigations, and affected asset roles.

Definition: depicts the mapping of a risk to each of the following: risk owner, risk mitigations, and affected asset roles.

Recommended Implementation: Matrix format, SysML Block Definition Diagram.

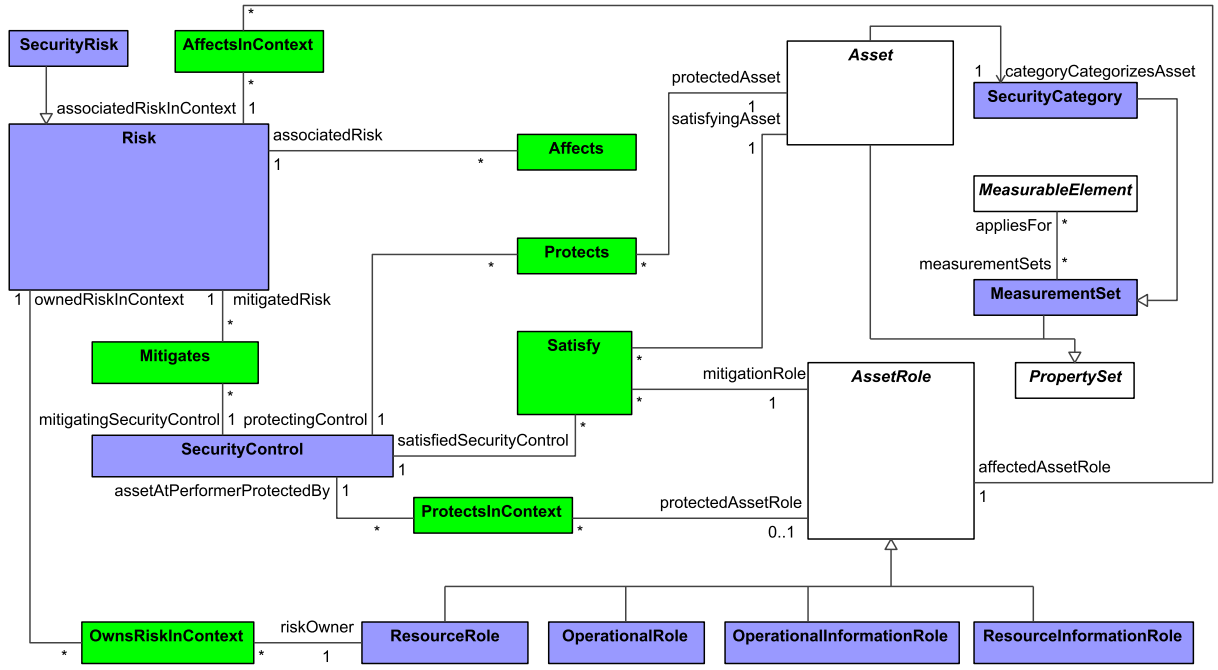

Figure 8:75 - Security Traceability

Elements

- Affects
- AffectsInContext
- Asset
- AssetRole
- MeasurableElement

- MeasurementSet
- Mitigates
- OperationalInformationRole
- OperationalRole
- OwnsRiskInContext
- PropertySet
- Protects
- ProtectsInContext
- ResourceInformationRole
- ResourceRole
- Risk
- Satisfy
- SecurityCategory
- SecurityControl
- SecurityRisk

8.1.9 View Specifications::Projects

Stakeholders: PMs, Project Portfolio Managers, Enterprise Architects.

Concerns: project portfolio, projects and project milestones.

Definition: describes projects and project milestones, how those projects deliver capabilities, the organizations contributing to the projects and dependencies between projects.

8.1.9.1 View Specifications::Projects::Taxonomy

Contains the diagrams that document the Project Taxonomy View Specification.

8.1.9.1.1 View Specifications::Projects::Taxonomy::Project Taxonomy

Stakeholders: PMs, Project Portfolio Managers, Enterprise Architects. Concerns: types of projects and project milestones. Definition: shows the taxonomy of types of projects and project milestones. Recommended Implementation: SysML Block Definition Diagram.

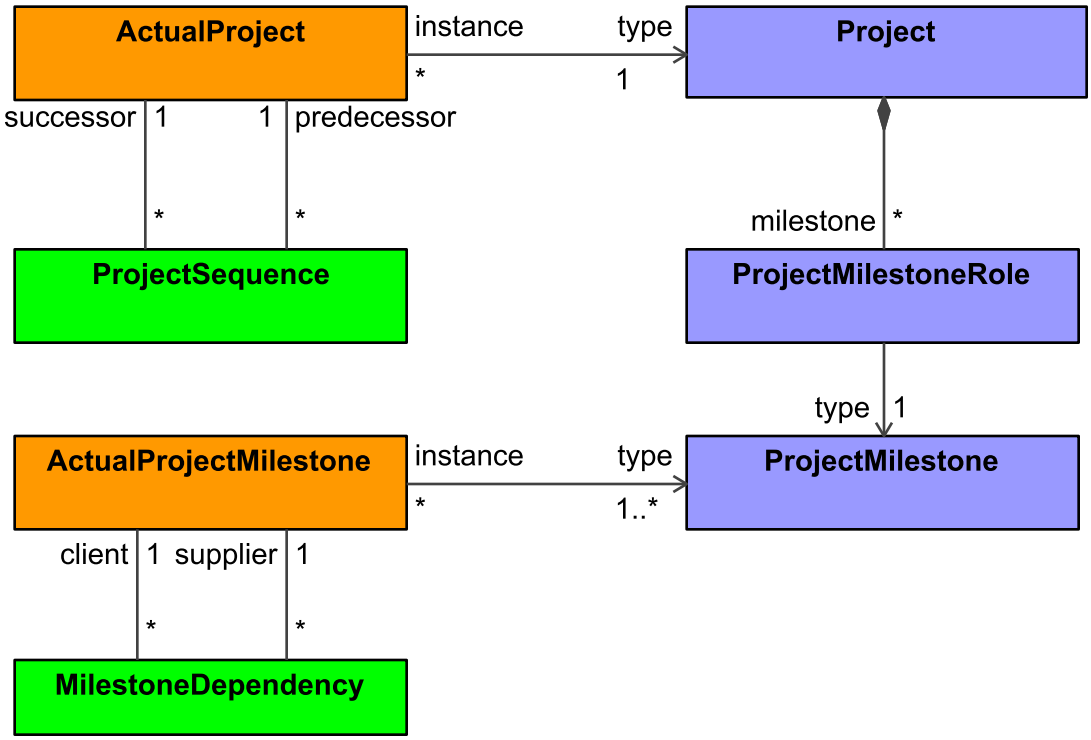

Figure 8:76 - Project Taxonomy

Elements

- ActualProject
- ActualProjectMilestone
- MilestoneDependency
- Project
- ProjectMilestone
- ProjectMilestoneRole
- ProjectSequence

8.1.9.2 View Specifications::Projects::Structure

Contains the diagrams that document the Project Structure View Specification.

8.1.9.2.1 View Specifications::Projects::Structure::Project Structure

Stakeholders: PMs.

Concerns: relationships between types of projects and project milestones.

Definition: provides a template for an actual project(s) road map(s) to be implemented.

Recommended Implementation: SysML Block Definition Diagram.

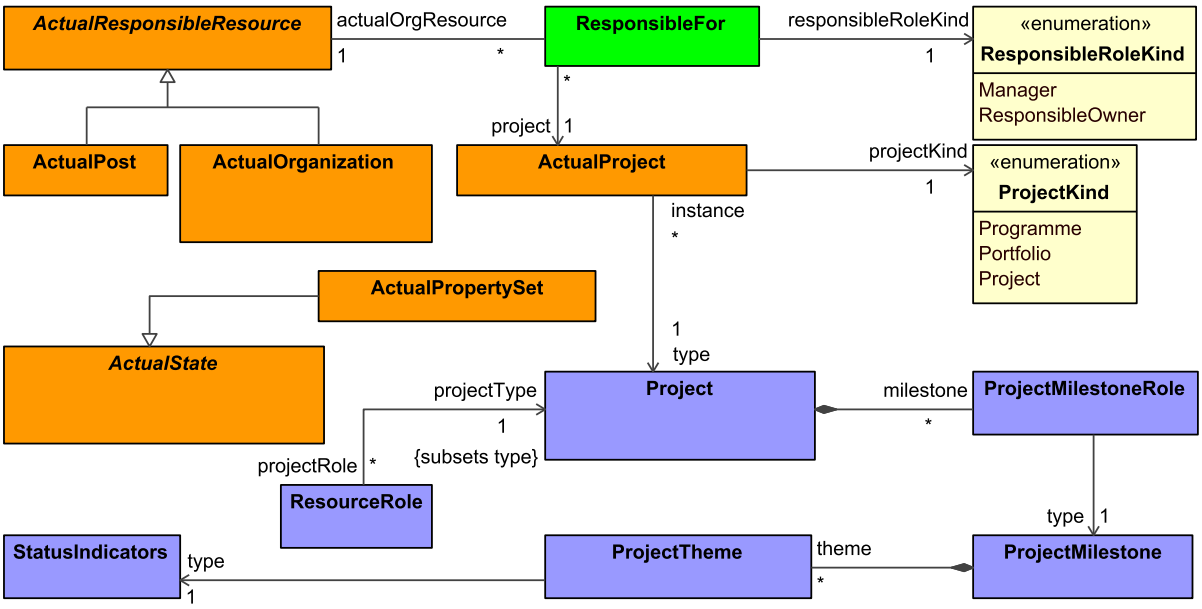

Figure 8:77 - Project Structure

Elements

- ActualOrganization
- ActualPost
- ActualProject
- ActualPropertySet
- ActualResponsibleResource
- ActualState
- Project
- ProjectMilestone
- ProjectMilestoneRole
- ProjectTheme
- ResourceRole
- ResponsibleFor
- StatusIndicators

8.1.9.3 View Specifications::Projects::Connectivity

Contains the diagrams that document the Project Connectivity View Specification.

8.1.9.3.1 View Specifications::Projects::Connectivity::Project Connectivity

Stakeholders: PMs.

Concerns: relationships between projects and project milestones.

Definition: shows how projects and project milestones are related in sequence.

Recommended Implementation: SysML Block Definition Diagram.

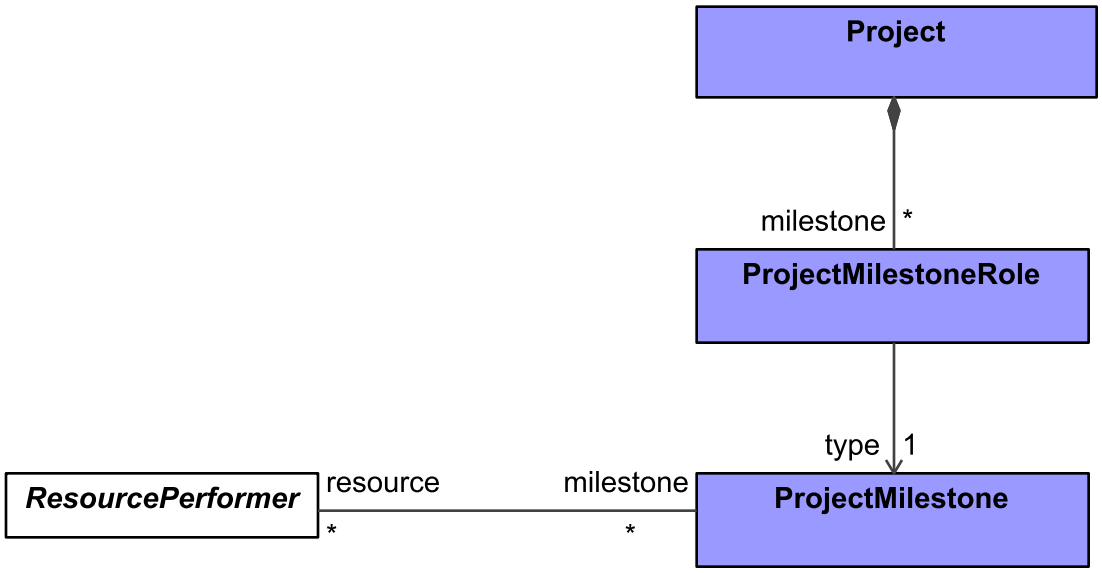

Figure 8:78 - Project Connectivity

Elements

- Project
- ProjectMilestone
- ProjectMilestoneRole
- ResourcePerformer

8.1.9.4 View Specifications::Projects::Processes

Contains the diagrams that document the Project Processes View Specification.

8.1.9.4.1 View Specifications::Projects::Processes::Project Processes

Stakeholders: PMs.

Concerns: captures project tasks (ProjectActivities) and flows between them.

Definition: describes the ProjectActivities that are normally conducted in the course of projects to support capability(ies) and implement resources. It describes the ProjectActivities, their Inputs/Outputs, ProjectActivityActions and flows between them.

Recommended Implementation: SysML Activity Diagram, SysML Block Definition Diagram, BPMN Process Diagram as described in Resources Processes section.

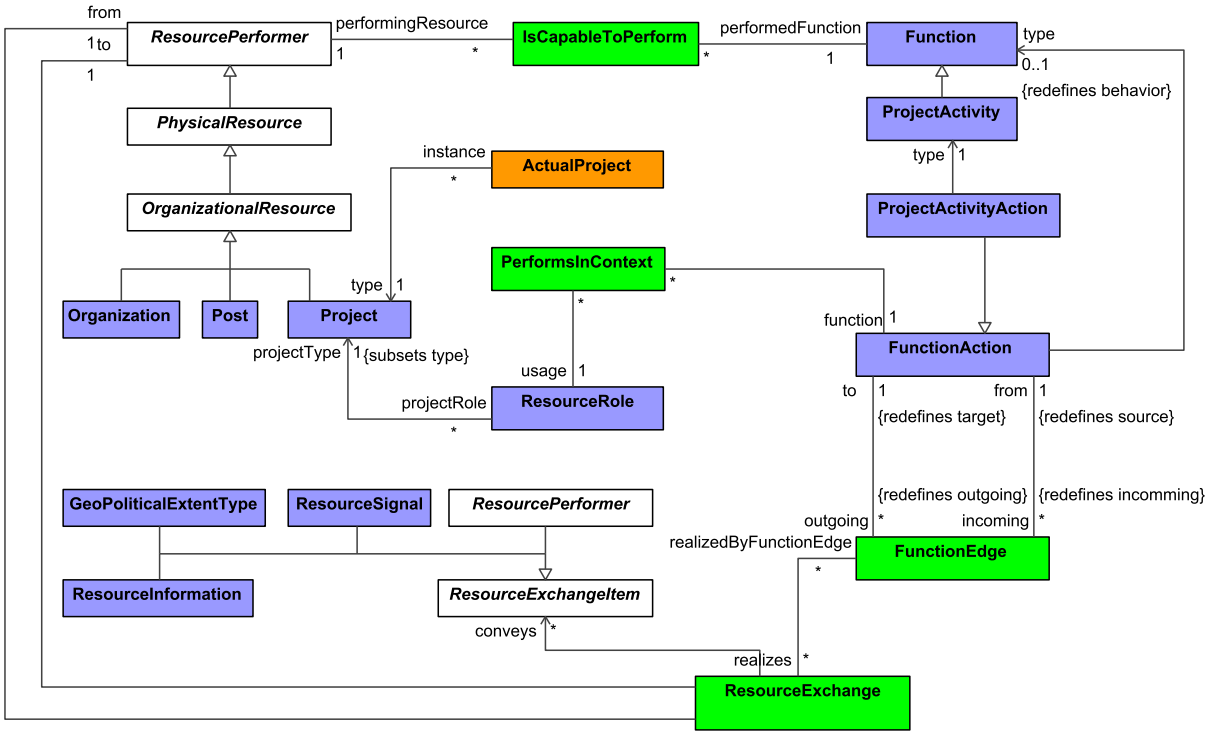

Figure 8:79 - Project Processes

Elements

- ActualProject
- Function
- FunctionAction
- FunctionEdge
- GeoPoliticalExtentType
- IsCapableToPerform
- Organization
- OrganizationalResource
- PerformsInContext
- PhysicalResource
- Post
- Project
- ProjectActivity
- ProjectActivityAction
- ResourceExchange
- ResourceExchangeItem
- ResourceInformation
- ResourcePerformer
- ResourceRole
- ResourceSignal

8.1.9.5 View Specifications::Projects::Roadmap

Contains the diagrams that document the Project Roadmap View Specification.

8.1.9.5.1 View Specifications::Projects::Roadmap::Project Roadmap

Stakeholders: PMs, Capability Owners, Solution Providers, Enterprise Architects.

Concerns: the product portfolio management; a planning of capability delivery.

Definition: provides a timeline perspective on programs or projects

Recommended Implementation: timeline, tabular format, SysML Block Definition Diagram.

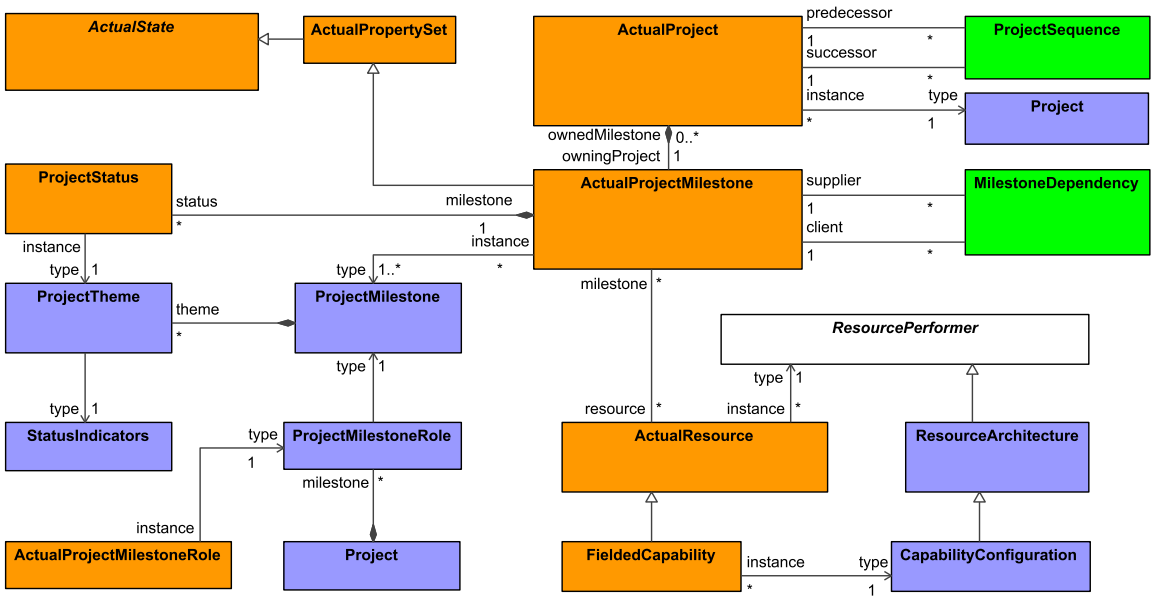

Figure 8:80 - Project Roadmap

Elements

- ActualProject
- ActualProjectMilestone
- ActualProjectMilestoneRole
- ActualPropertySet
- ActualResource
- ActualState
- CapabilityConfiguration
- FieldedCapability
- MilestoneDependency
- Project
- ProjectMilestone
- ProjectMilestoneRole
- ProjectSequence
- ProjectStatus
- ProjectTheme
- ResourceArchitecture
- ResourcePerformer
- StatusIndicators

8.1.9.6 View Specifications::Projects::Traceability

Contains the diagrams that document the Project Traceability View Specification.

8.1.9.6.1 View Specifications::Projects::Traceability::Project Traceability

Stakeholders: PMs, Project Portfolio Managers, Enterprise Architects.

Concerns: traceability between capabilities and projects that deliver them.

Definition: depicts the mapping of projects to capabilities and thus identifies the transformation of a capability(ies) into a purposeful implementation via projects.

Recommended Implementation: Matrix format, SysML Block Definition Diagram.

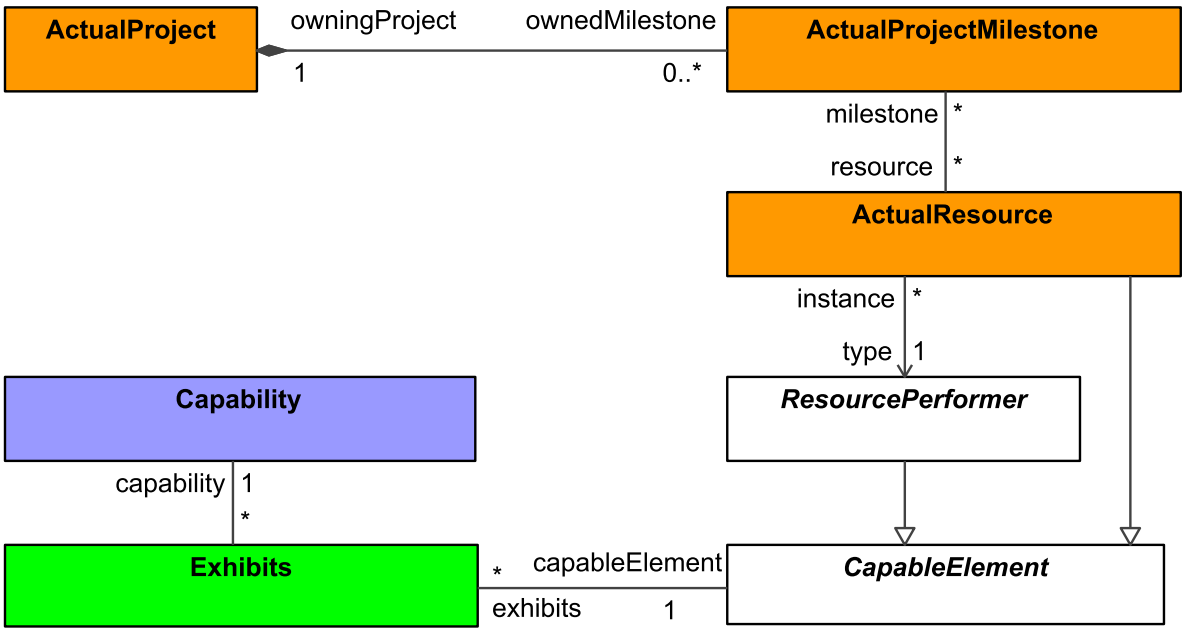

Figure 8:81 - Project Traceability

Elements

- ActualProject
- ActualProjectMilestone
- ActualResource
- Capability
- CapableElement
- Exhibits
- ResourcePerformer

8.1.10 View Specifications::Standards

Stakeholders: Solution Providers, Systems Engineers, Software Engineers, Systems Architects, Business Architects.

Concerns: technical and non-technical Standards applicable to the architecture.

Definition: shows the technical, operational, and business Standards applicable to the architecture. Defines the underlying current and expected Standards.

8.1.10.1 View Specifications::Standards::Taxonomy

Contains the diagrams that document the Standards Taxonomy View Specification.

8.1.10.1.1 View Specifications::Standards::Taxonomy::Standards Taxonomy

Stakeholders: Solution Providers, Systems Engineers, Software Engineers, Systems Architects, Business Architects.

Concerns: technical and non-technical standards, guidance and policy applicable to the architecture.

Definition: shows the taxonomy of types of technical, operational, and business standards, guidance and policy applicable to the architecture.

Recommended Implementation: SysML Block Definition Diagram.

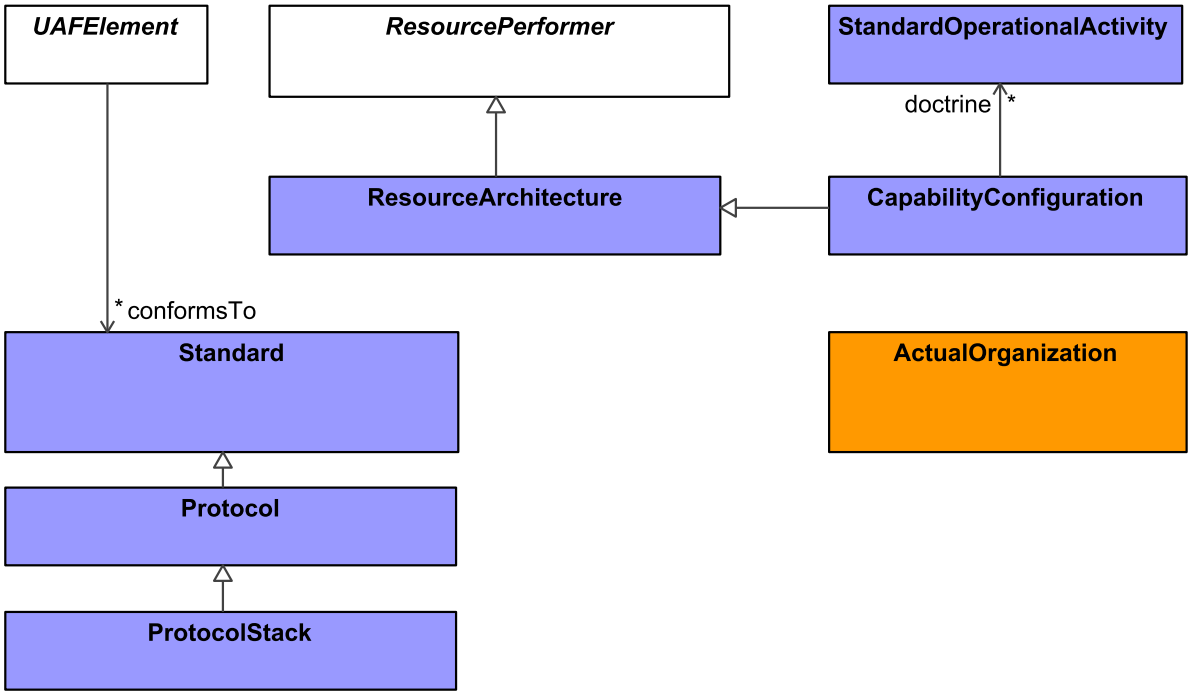

Figure 8:82 - Standards Taxonomy

Elements

- ActualOrganization
- CapabilityConfiguration
- Protocol
- ProtocolStack
- ResourceArchitecture
- ResourcePerformer
- Standard
- StandardOperationalActivity
- UAFElement

8.1.10.2 View Specifications::Standards::Structure

Contains the diagrams that document the Standards Structure View Specification.

8.1.10.2.1 View Specifications::Standards::Structure::Standards Structure

Stakeholders: Solution Providers, Systems Engineers, Software Engineers, Systems Architects.

Concerns: the specification of the protocol stack used in the architecture.

Definition: shows the composition of standards required to achieve the architecture's objectives.

Recommended Implementation: SysML Internal Block Diagram.

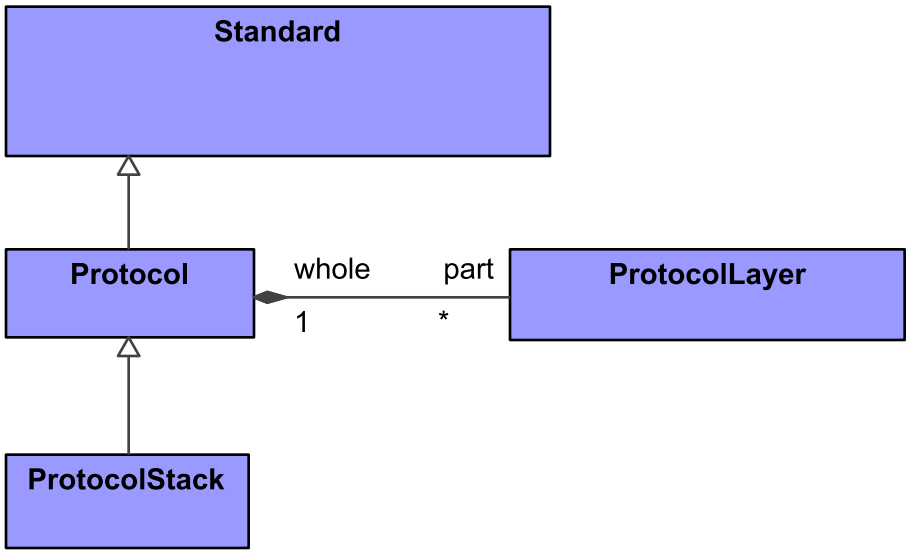

Figure 8:83 - Standards Structure

Elements

- Protocol
- ProtocolLayer
- ProtocolStack
- Standard

8.1.10.3 View Specifications::Standards::Roadmap

Contains the diagrams that document the Standards Roadmap View Specification.

8.1.10.3.1 View Specifications::Standards::Roadmap::Standards Roadmap

Stakeholders: Solution Providers, Systems Engineers, Systems Architects, Software Engineers, Business Architects.

Concerns: expected changes in technology-related standards and conventions, operational standards, or business standards and conventions.

Definition: defines the underlying current and expected standards. Expected standards are those that can be reasonably forecast given the current state of technology, and expected improvements / trends.

Recommended Implementation: timeline, tabular format, SysML Block Definition Diagram.

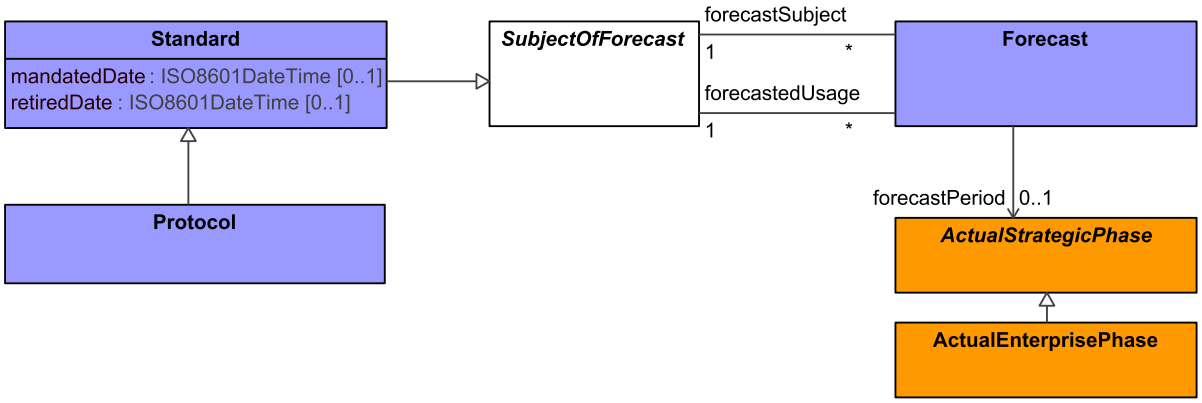

Figure 8:84 - Standards Roadmap

Elements

- ActualEnterprisePhase
- ActualStrategicPhase
- Forecast
- Protocol
- Standard
- SubjectOfForecast

8.1.10.4 View Specifications::Standards::Traceability

Contains the diagrams that document the Standards Traceability View Specification.

8.1.10.4.1 View Specifications::Standards::Traceability::Standards Traceability

Stakeholders: Solution Providers, Systems Engineers, Software Engineers, Systems Architects, Business Architects.

Concerns: standards that need to be taken in account to ensure the interoperability of the implementation of architectural elements.

Definition: shows the applicability of standards to specific elements in the architecture.

Recommended Implementation: tabular format, matrix format, SysML Block Definition Diagram.

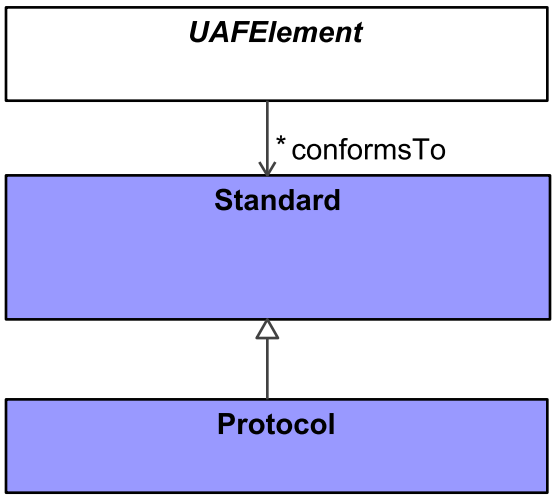

Figure 8:85 - Standards Traceability

Elements

- Protocol
- Standard
- UAFElement

8.1.11 View Specifications::Actual Resources

Stakeholders: Solution Providers, Systems Engineers, Business Architects, Human Resources.

Concerns: the analysis, e.g. evaluation of different alternatives, what-if, trade-offs, V&V on the actual resource configurations.

Definition: illustrates the expected or achieved actual resource configurations and actual relationships between them.

8.1.11.1 View Specifications::Actual Resources::Structure

Contains the diagrams that document the Actual Resources Structure View Specification.

8.1.11.1.1 View Specifications::Actual Resources::Structure::Actual Resources Structure

Stakeholders: Solution Providers, Systems Engineers, Business Architects.

Concerns: the analysis, e.g. evaluation of different alternatives, what-if, trade-offs, V&V on the actual resource configurations as it provides a means to capture different solution architectures. The detailed analysis (trade-off, what-if etc.) is carried out using the Resource Constraints view.

Definition: illustrates the expected or achieved actual resource configurations required to meet an operational need.

Recommended Implementation: SysML Block Definition Diagram.

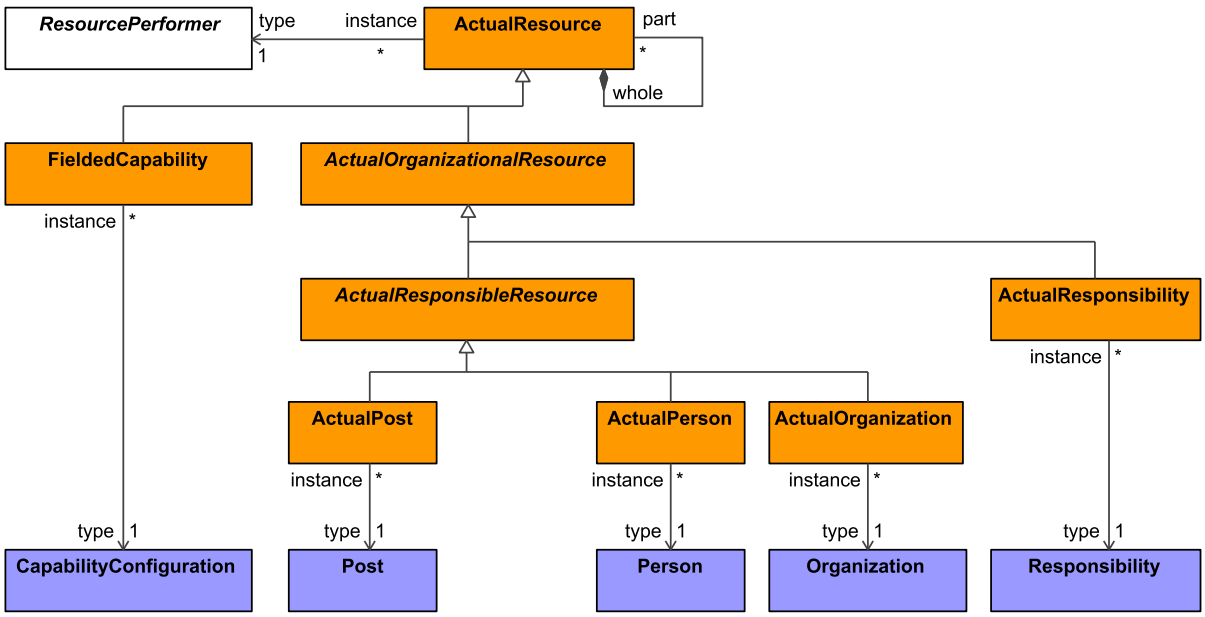

Figure 8:86 - Actual Resources Structure

Elements

- ActualOrganization
- ActualOrganizationalResource
- ActualPerson
- ActualPost

- ActualResource
- ActualResponsibility
- ActualResponsibleResource
- CapabilityConfiguration
- FieldedCapability
- Organization
- Person
- Post
- ResourcePerformer
- Responsibility

8.1.11.2 View Specifications::Actual Resources::Connectivity

Contains the diagrams that document the Actual Resources Connectivity View Specification.

8.1.11.2.1 View Specifications::Actual Resources::Connectivity::Actual Resources Connectivity

Stakeholders: Solution Providers, Systems Engineers, Business Architects.

Concerns: the communication of actual resource.

Definition: illustrates the actual resource configurations and actual relationships between them.

Recommended Implementation: tabular format, SysML Block Definition Diagram, SysML Internal Block Diagram, SysML Sequence Diagram.

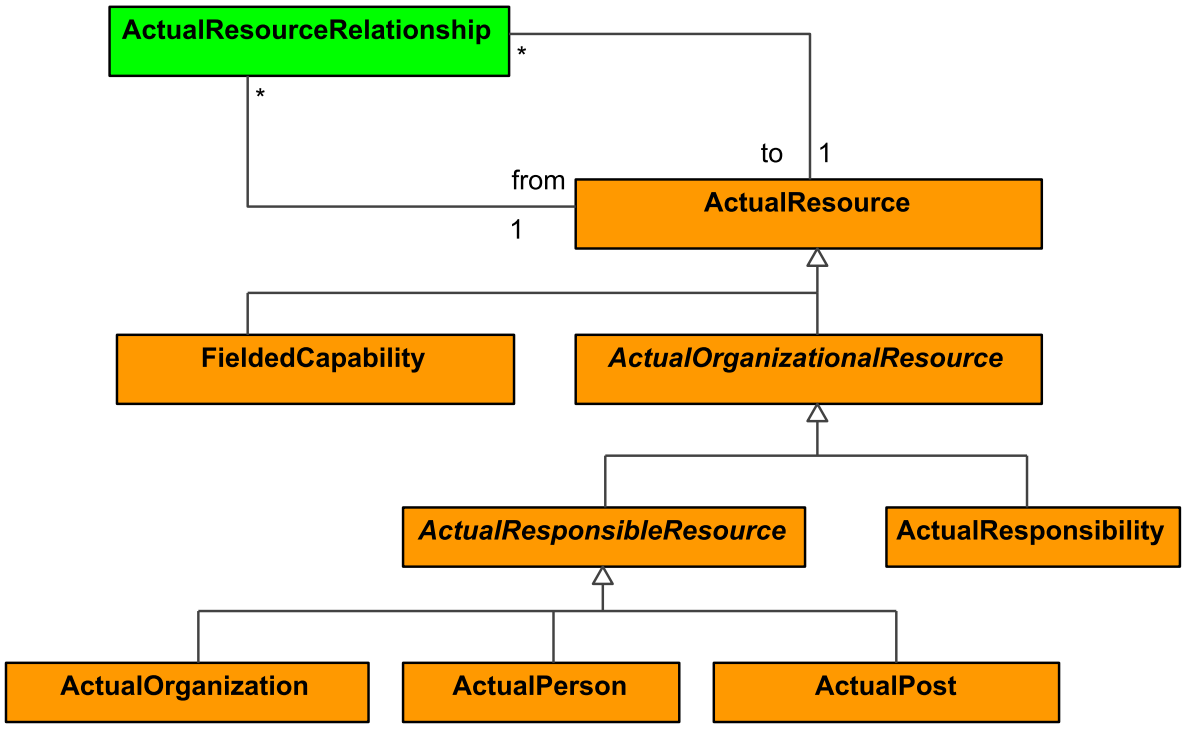

Figure 8:87 - Actual Resources Connectivity

Elements

- ActualOrganization
- ActualOrganizationalResource
- ActualPerson

- ActualPost
- ActualResource
- ActualResourceRelationship
- ActualResponsibility
- ActualResponsibleResource
- FieldedCapability

8.1.11.3 View Specifications::Actual Resources::Traceability

Contains the diagrams that document the Actual Resources Traceability View Specification.

8.1.11.3.1 View Specifications::Actual Resources::Traceability::Actual Resources Traceability

Stakeholders: Systems Engineers, Enterprise Architects, Solution Providers, Business Architects.

Concerns: traceability between operational activities and functions that implements them.

Definition: depicts the mapping of functions to operational activities and thus identifies the transformation of an operational need into a purposeful function performed by a resource or solution.

Recommended Implementation: Matrix format, SysML Block Definition Diagram.

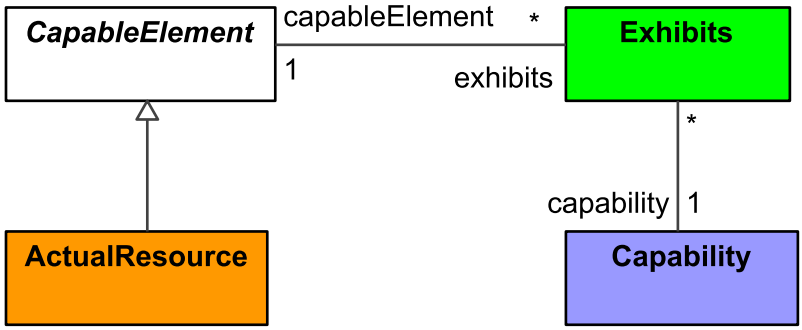

Figure 8:88 - Actual Resources Traceability

Elements

- ActualResource
- Capability
- CapableElement
- Exhibits

8.1.12 View Specifications::Motivation

Stakeholders: Enterprise Architects, Enterprise Systems Engineers, Model Managers, System Architects, Requirement Engineers, Solution Providers, Systems Engineers, Software Engineers, Business Architects, Portfolio Managers, Program Managers, Security Architects, Security Engineers, Risk Analysts.

Concerns: (i) architecture drivers, challenges, opportunities, capabilities that address opportunities, phases and architectures that address challenges; (ii) requirements, their relationship (via traceability) to more detailed requirements and the solution described by the architecture that will meet those requirements; (iii) security controls, security control families, and overlays.

Definition: Identifies and defines motivational elements e.g., challenges, opportunities, and concerns, that pertain to enterprise transformation efforts, and different types of requirements, e.g., operational, services, personnel, resources, or security controls.

Recommended Implementation: SysML Block Definition Diagram, SysML Package Diagram, tabular format.

8.1.12.1 View Specifications::Motivation::Motivation: Requirements

Stakeholders: Requirement Engineers, Solution Providers, Systems Engineers, Software Engineers, Systems Architects, Business Architects.

Concerns: provides a central reference for a set of stakeholder needs expressed as requirements, their relationship (via traceability) to more detailed requirements and the solution described by the architecture that will meet those requirements.

Definition: used to represent requirements, their properties, and relationships (trace, verify, satisfy, refine) between each other and to UAF architectural elements.

Recommended Implementation: SysML Requirement Diagram, tabular format, matrix format.

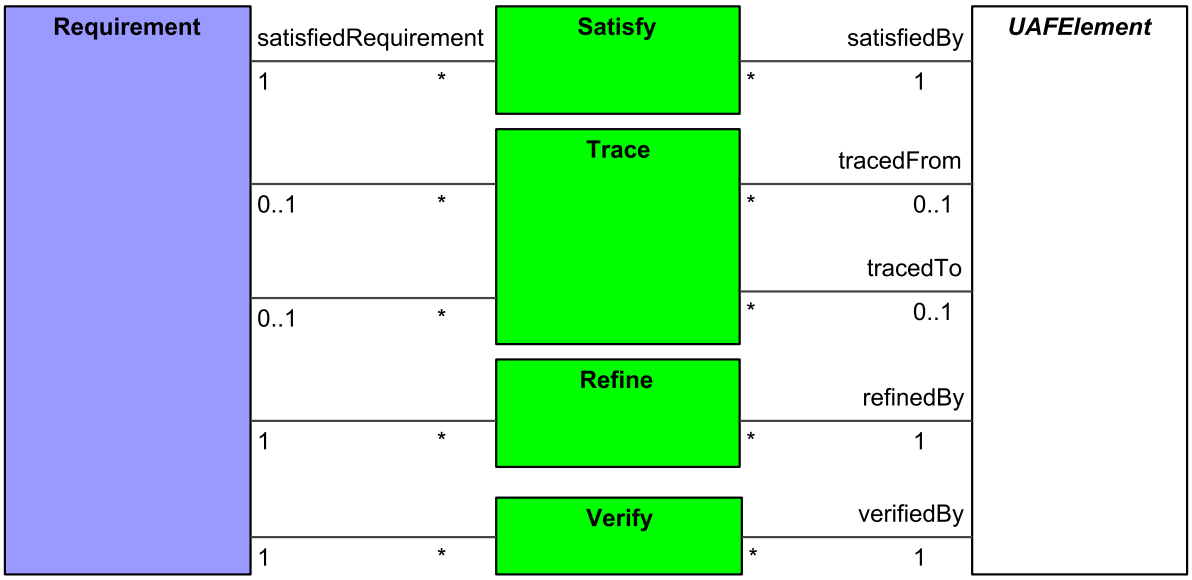

Figure 8:89 - Motivation: Requirements

Elements

- Refine
- Requirement
- Satisfy
- Trace
- UAFElement
- Verify

8.1.13 View Specifications::Information

Stakeholders: Data Modelers, Software Engineers, Systems Engineers

Concerns: address the information perspective on operational, service, and resource architectures.

Definition: allows analysis of an architecture's information and data definition aspect, without consideration of implementation specific issues.

Recommended Implementation: SysML Block Definition Diagram.

8.1.13.1 View Specifications::Information::Information: Operational Information

Stakeholders: Data Modelers, Software Engineers, Systems Engineers, Operators and Users, Service Managers and Providers

Concerns: address the information perspective on operational and service architectures.

Definition: allows analysis of an architecture's information and data definition aspect, without consideration of implementation specific issues.

Recommended Implementation: SysML Block Definition Diagram.

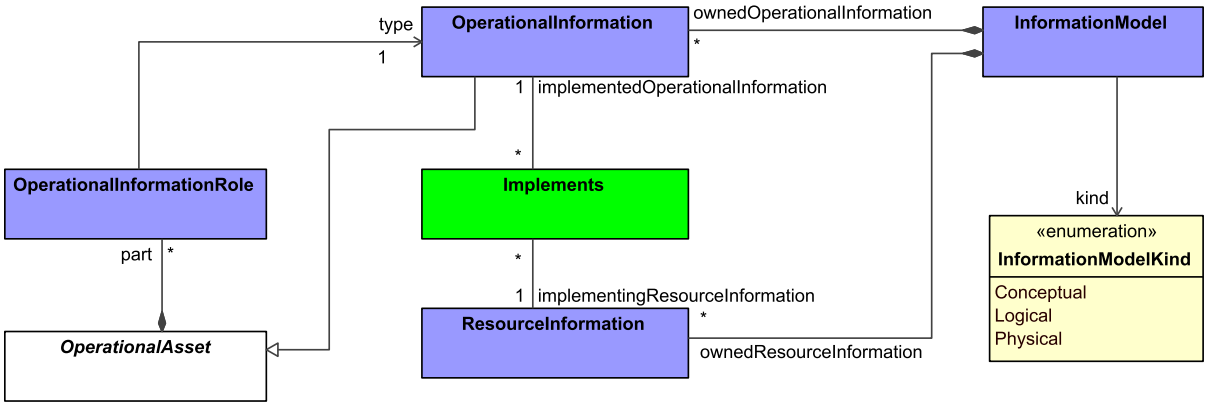

Figure 8:90 - Information: Operational Information

Elements

- Implements
- InformationModel
- OperationalAsset
- OperationalInformation
- OperationalInformationRole
- ResourceInformation

8.1.13.2 View Specifications::Information::Information: Resources Information

Stakeholders: Data Modelers, Software Engineers, Systems Engineers

Concerns: address the information perspective on operational, service, and resource architectures.

Definition: allows analysis of an architecture's information and data definition aspect, without consideration of implementation specific issues.

Recommended Implementation: SysML Block Definition Diagram.

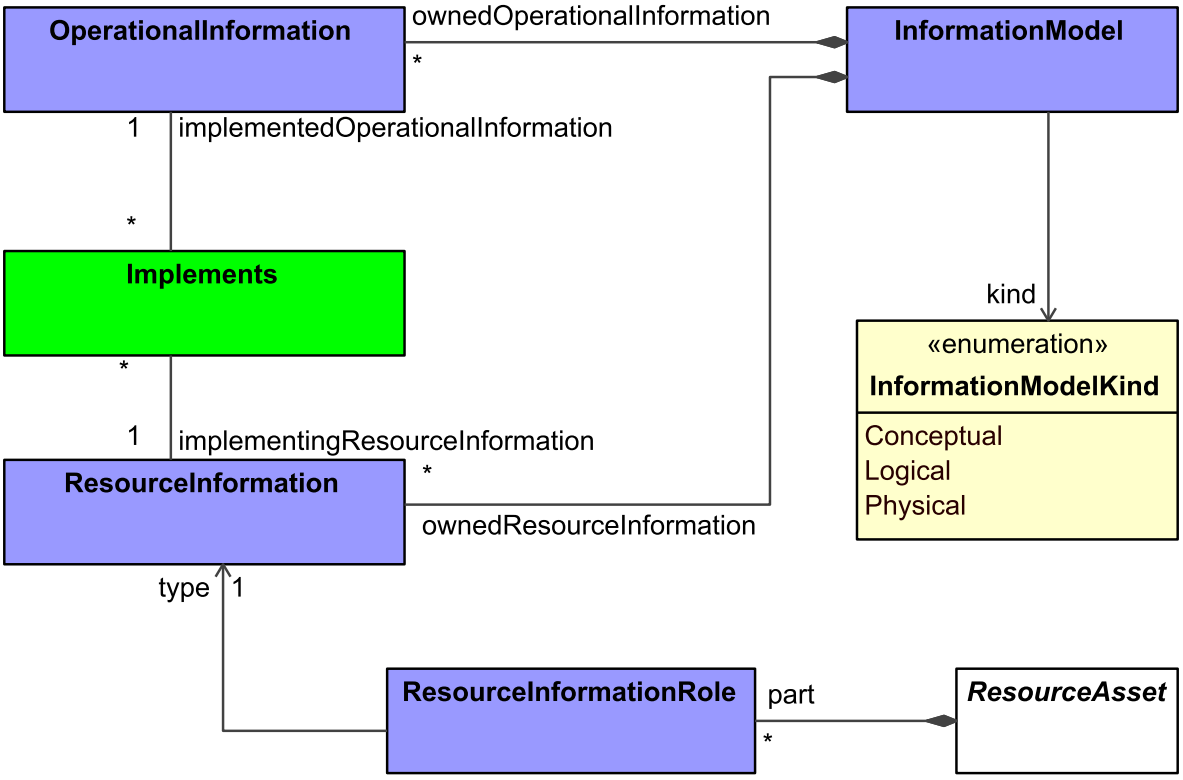

Figure 8:91 - Information: Resources Information

Elements

- Implements
- InformationModel
- OperationalInformation
- ResourceAsset
- ResourceInformation
- ResourceInformationRole

8.1.14 View Specifications::Parameters

Stakeholders: Capability owners, Systems Engineers, Solution Providers.

Concerns: identifies measurable properties that can be used to support engineering analysis and environment for the Capabilities

Definition: Shows the measurable properties of something in the physical world and elements and relationships that are involved in defining the environments applicable to capability, operational concept or set of systems.

8.1.14.1 View Specifications::Parameters::Parameters: Environment

Stakeholders: Capability owners, Systems Engineers, Solution Providers.

Concerns: defines the environment for the capabilities.

Definition: shows the elements and relationships that are involved in defining the environments applicable to capability, operational concept or set of systems.

Recommended Implementation: SysML Block Definition Diagram.

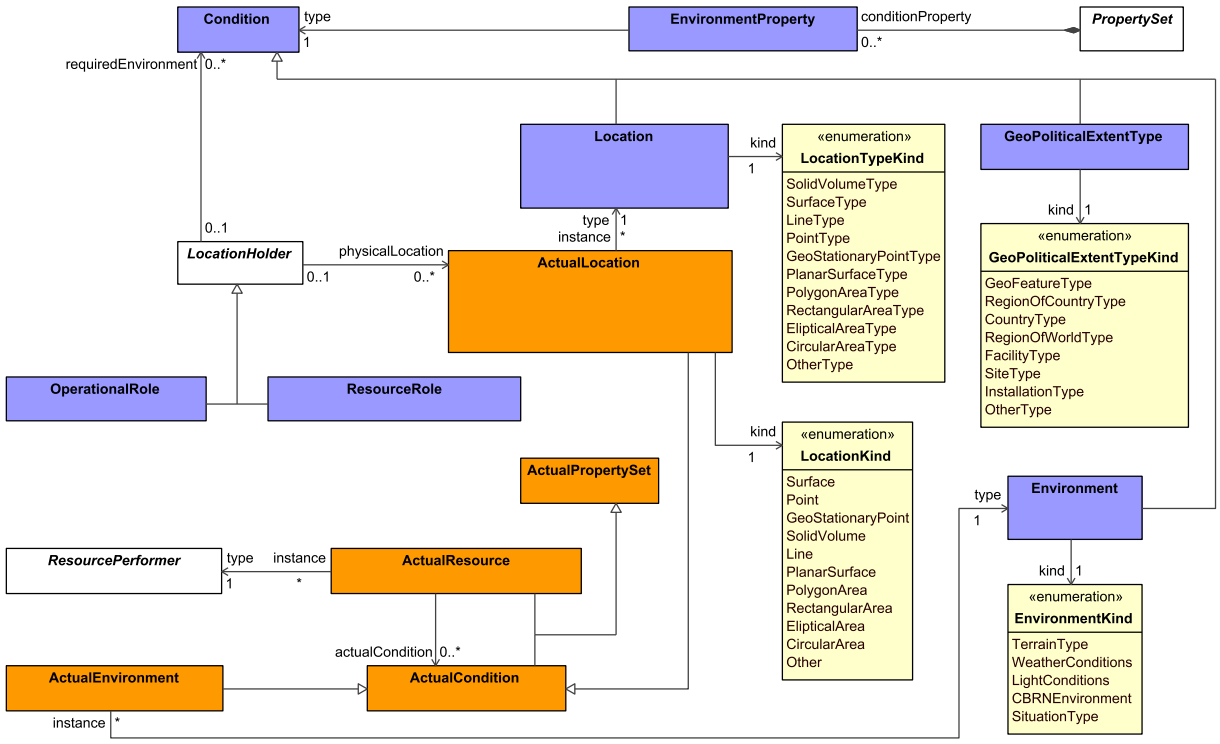

Figure 8:92 - Parameters: Environment

Elements

- ActualCondition
- ActualEnvironment
- ActualLocation
- ActualPropertySet
- ActualResource
- Condition
- Environment
- EnvironmentProperty
- GeoPoliticalExtentType
- Location
- LocationHolder
- OperationalRole
- PropertySet
- ResourcePerformer
- ResourceRole

8.1.14.2 View Specifications::Parameters::Parameters: Measurements

Stakeholders: Capability owners, Systems Engineers, Solution Providers.

Concerns: identifies measurable properties that can be used to support analysis such as KPIs, MoEs, TPIs etc.

Definition: Shows the measurable properties of something in the physical world, expressed in amounts of a unit of measure that can be associated with any element in the architecture.

Recommended Implementation: SysML Block Definition Diagram.

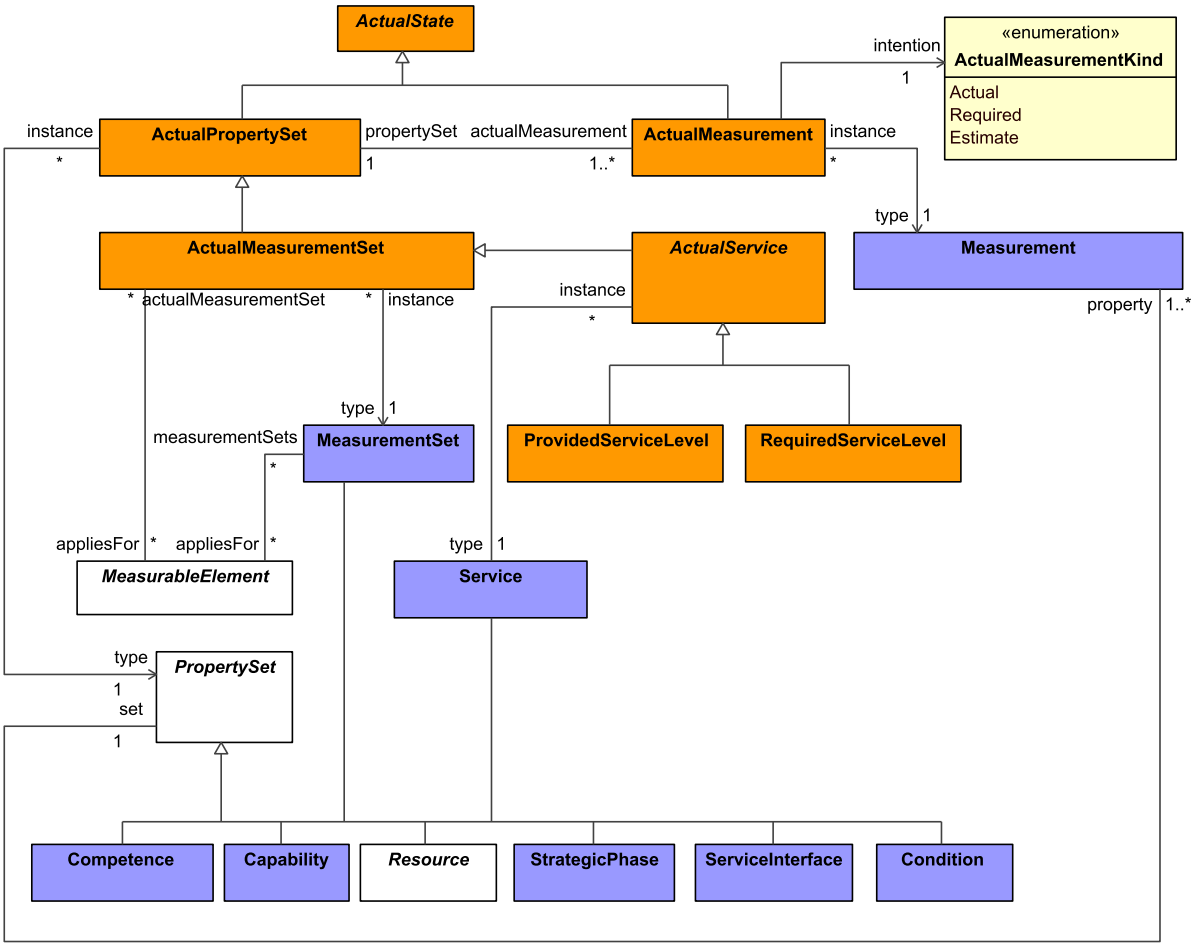

Figure 8:93 - Parameters: Measurements

Elements

- ActualMeasurement
- ActualMeasurementSet
- ActualPropertySet
- ActualService
- ActualState
- Capability
- Competence
- Condition
- MeasurableElement
- Measurement
- MeasurementSet
- PropertySet
- ProvidedServiceLevel
- RequiredServiceLevel
- Resource
- Service
- ServiceInterface
- StrategicPhase

8.1.14.3 View Specifications::Parameters::Parameters: Mission Vignette and Scenario

Stakeholders: Capability owners, Systems Engineers, Solution Providers.

Concerns: defines the environment for the capabilities.

Definition: shows the elements and relationships that are involved in defining the environments applicable to capability, operational concept or set of systems.

Recommended Implementation: SysML Block Definition Diagram.

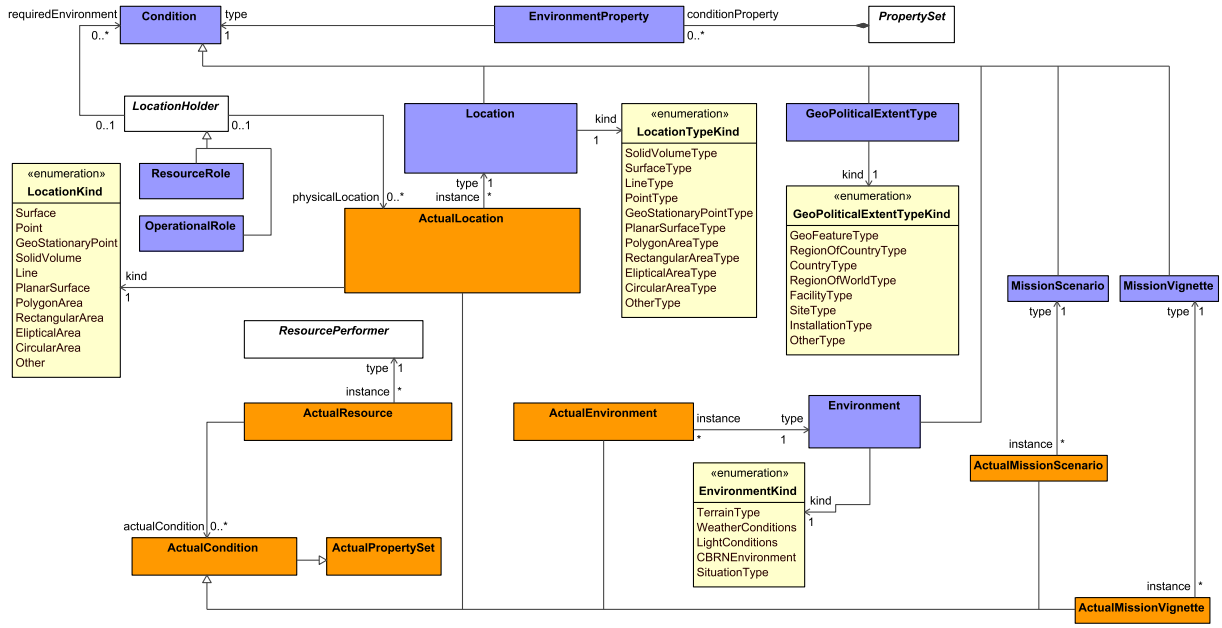

Figure 8:94 - Parameters: Mission Vignette and Scenario

Elements

- ActualCondition
- ActualEnvironment
- ActualLocation
- ActualMissionScenario
- ActualMissionVignette
- ActualPropertySet
- ActualResource
- Condition
- Environment
- EnvironmentProperty
- GeoPoliticalExtentType
- Location
- LocationHolder
- MissionScenario
- MissionVignette
- OperationalRole
- PropertySet
- ResourcePerformer
- ResourceRole

8.1.14.4 View Specifications::Parameters::Parameters: Risk

Stakeholders: Capability Owners, Systems Engineers, Solution Providers, Program Managers.

Concerns: identifies potential adverse conditions and situations that can inhibit achievement of goals.

Definition: Shows the relevant risks along with associated measures like likelihood of occurence and potential negative consequences.

Recommended Implementation: SysML Block Definition Diagram, matrix format.

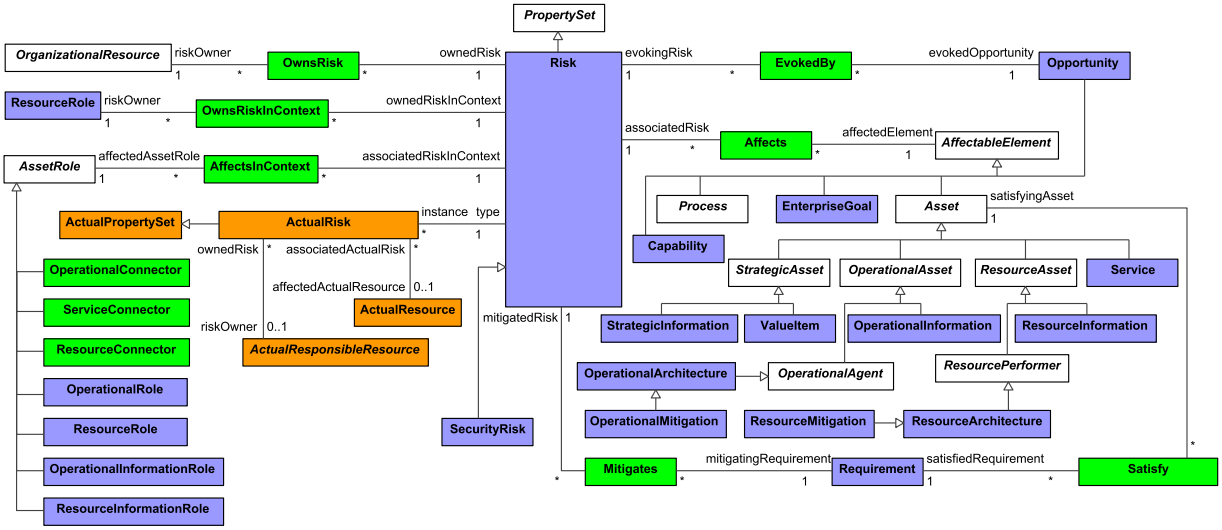

Figure 8:95 - Parameters: Risk

Elements

- ActualPropertySet
- ActualResource
- ActualResponsibleResource
- ActualRisk
- AffectableElement
- Affects
- AffectsInContext
- Asset
- AssetRole
- Capability
- EnterpriseGoal
- EvokedBy
- Mitigates
- OperationalAgent
- OperationalArchitecture
- OperationalAsset
- OperationalConnector
- OperationalInformation
- OperationalInformationRole
- OperationalMitigation
- OperationalRole
- Opportunity
- OrganizationalResource
- OwnsRisk
- OwnsRiskInContext

- Process
- PropertySet
- Requirement
- ResourceArchitecture
- ResourceAsset
- ResourceConnector
- ResourceInformation
- ResourceInformationRole
- ResourceMitigation
- ResourcePerformer
- ResourceRole
- Risk
- Satisfy
- SecurityRisk
- Service
- ServiceConnector
- StrategicAsset
- StrategicInformation
- ValueItem

8.1.15 View Specifications::Other

Contains the diagrams that document the use of BPMN, NIEM, IEPPV in the context of UAF.

8.1.15.1 View Specifications::Other::BPMN

Stakeholders: Business Architects, Enterprise Architects

Concerns: captures activity based behavior and flows.

Definition: describes the activities that are normally conducted in the course of achieving business goals that support a capability. It describes operational activities, their Inputs/Outputs, operational activity actions and flows between them using BPMN.

Recommended Implementation: BPMN Process Diagram.

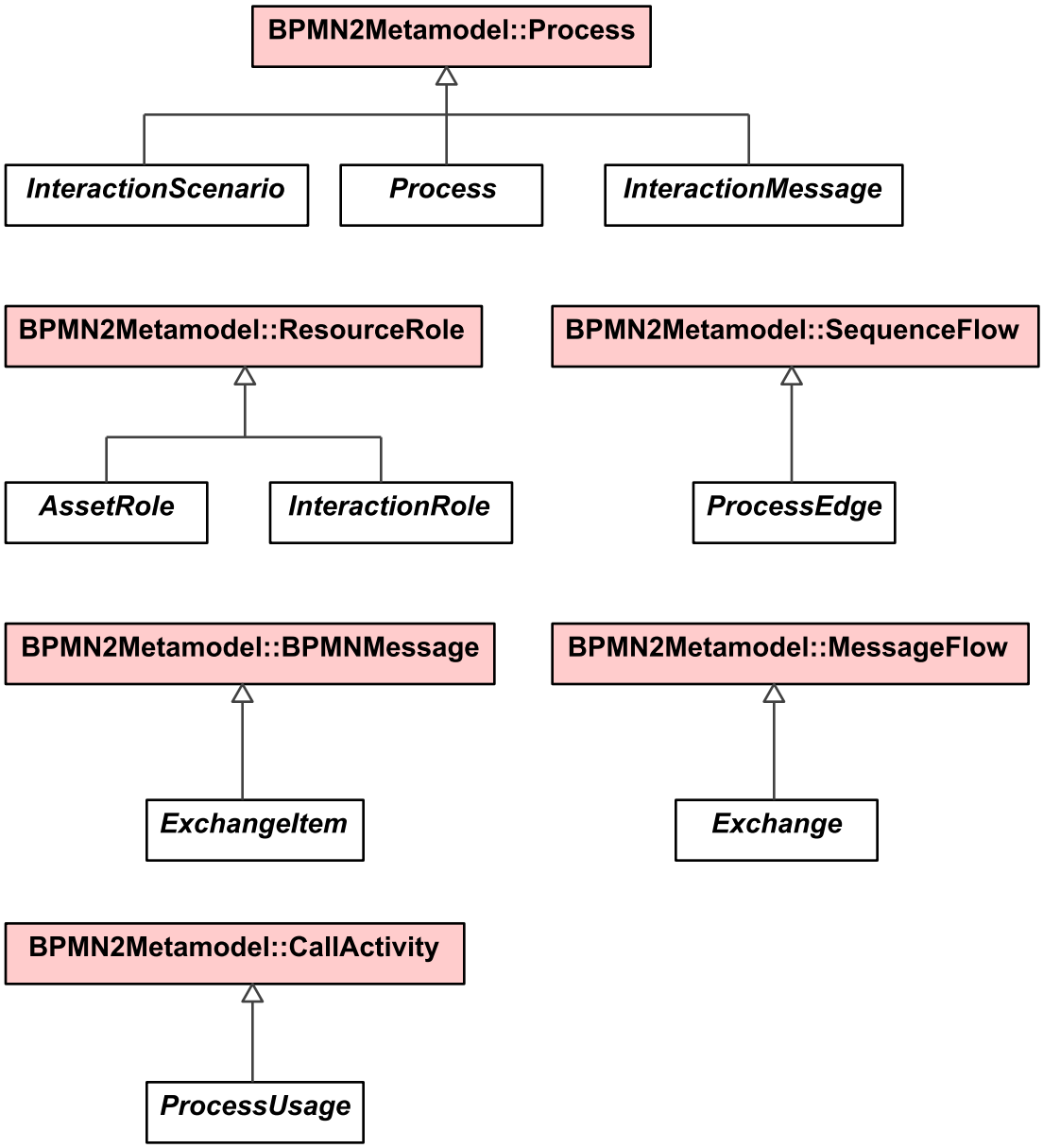

Figure 8:96 - BPMN

Elements

- AssetRole
- BPMN2Metamodel::BPMNMessage
- BPMN2Metamodel::CallActivity
- BPMN2Metamodel::MessageFlow
- BPMN2Metamodel::Process
- BPMN2Metamodel::ResourceRole
- BPMN2Metamodel::SequenceFlow
- Exchange
- ExchangeItem
- InteractionMessage
- InteractionRole
- InteractionScenario

- Process
- ProcessEdge
- ProcessUsage

8.1.15.2 View Specifications::Other::IEPPV

Stakeholders: Data Modelers, Solution Providers, Systems Engineers, Software Engineers, Systems Architects, Business Architects, information architects.

Concerns: information exchanges, information interfaces, information interoperability, information sharing and safeguarding.

Definition: UAFP supports information modeling and traceability to IEPPV model elements using the IEPPV-defined elements: Message, SemanticElement, and FilteredSemanticElement, used to represent data, properties/attributes, structure, format, and relationships. The IEPPV profile enables the specification of the policies, rules and constraints governing the packaging (assembly, transformation, marking, redaction) of data elements conforming to information sharing and safeguarding requirements. The IEPPV profile also governs the processing (parsing, transformation, and marshalling) received information and data element.

Recommended Implementation: UML Class Diagram, SysML Block Diagram.

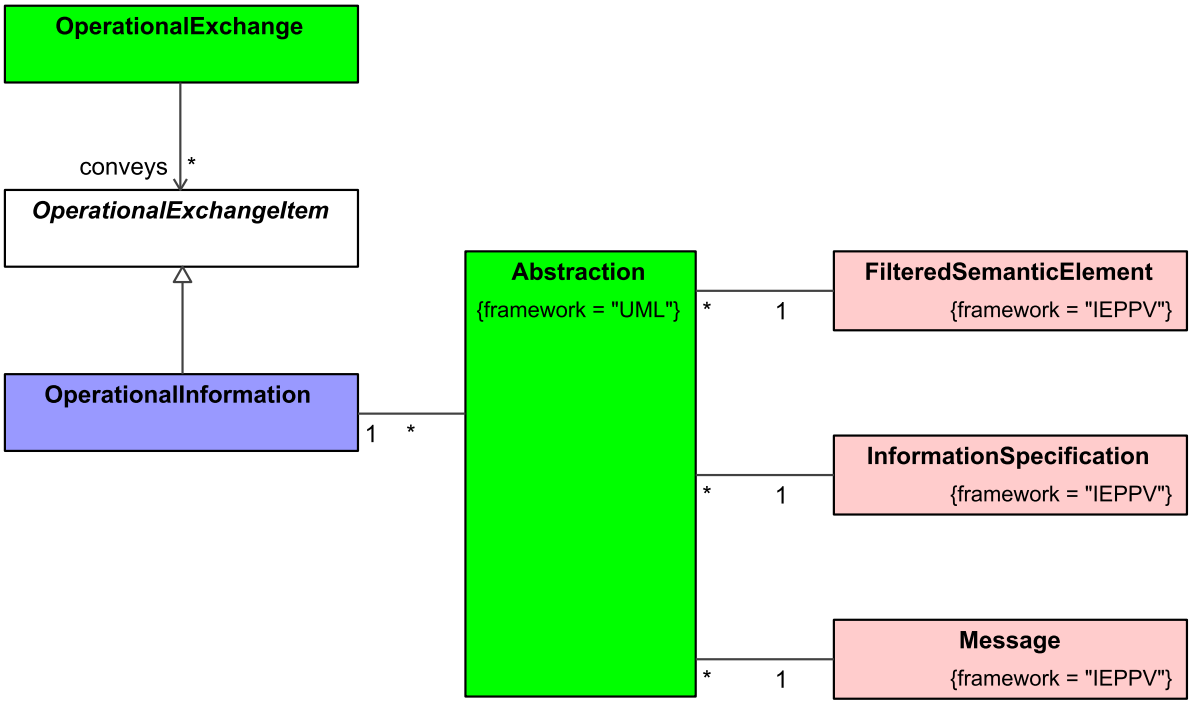

Figure 8:97 - IEPPV

Elements

- Abstraction
- FilteredSemanticElement
- InformationSpecification
- Message
- OperationalExchange
- OperationalExchangeItem
- OperationalInformation

8.1.15.3 View Specifications::Other::NIEM

Stakeholders: Data Modelers, Solution Providers, Systems Engineers, Software Engineers, Systems Architects, Business Architects.

Concerns: information exchanges, information interoperability, data schema.

Definition: A specification representing the structure, semantics, and relationships of data objects that satisfy an information exchange requirement. Used for organizing and packaging Model Package Descriptions (MPDs) and Information Exchange Package Documentation (IEPD) as defined by the National Information Exchange Model (NIEM). An IEPD is a type of MPD. The NIEM MPD defines an Enterprise Information Exchange Model (EIEM) as an MPD that contains NIEM-conforming schemas that define and declare data components to be consistently reused in the IEPDs of an enterprise. An EIEM is a collection of schemas organized into a collection of subset schemas and one or more extension schemas. An information sharing enterprise creates and maintains an EIEM.

Recommended Implementation: UML Class Diagram, SysML Block Diagram.

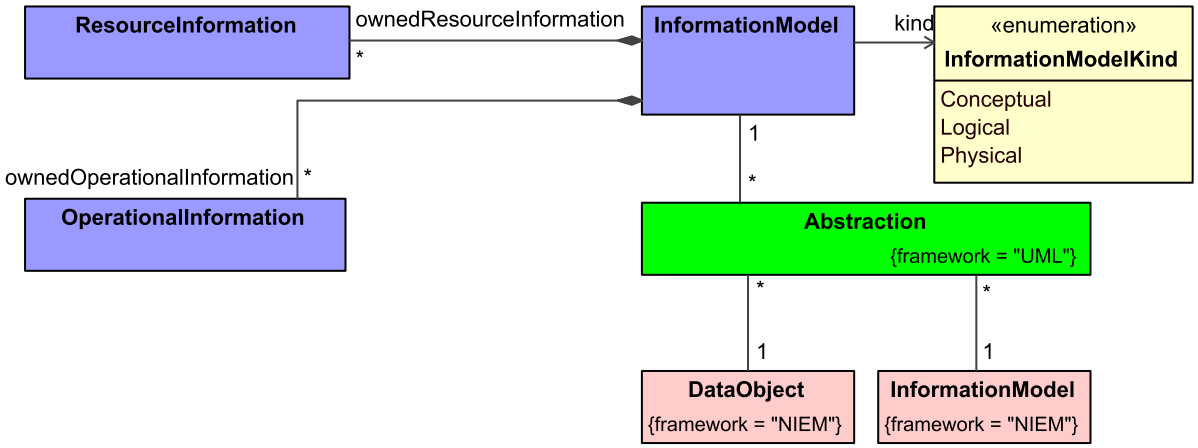

Figure 8:98 - NIEM

Elements

- Abstraction
- DataObject
- InformationModel
- InformationModel
- OperationalInformation
- ResourceInformation

9. Domain Metamodel (DMM) Elements

9.1 Domain MetaModel

This package contains the elements of the DMM.

9.1.1 Domain MetaModel::Architecture Management

Stakeholders: Enterprise Architects, people who want to discover the architecture, Technical Managers.

Concerns: Captures meta-data relevant to the entire architecture

Definition: Provide information pertinent to the entire architecture. Present supporting information rather than architectural models.

9.1.1.1 Domain MetaModel::Architecture Management::Taxonomy

ActualState

Package: Taxonomy
isAbstract: Yes
Generalization: UAFElement

Description

Abstract element that applies temporal extent to a set of elements realized as Instance Specifications.

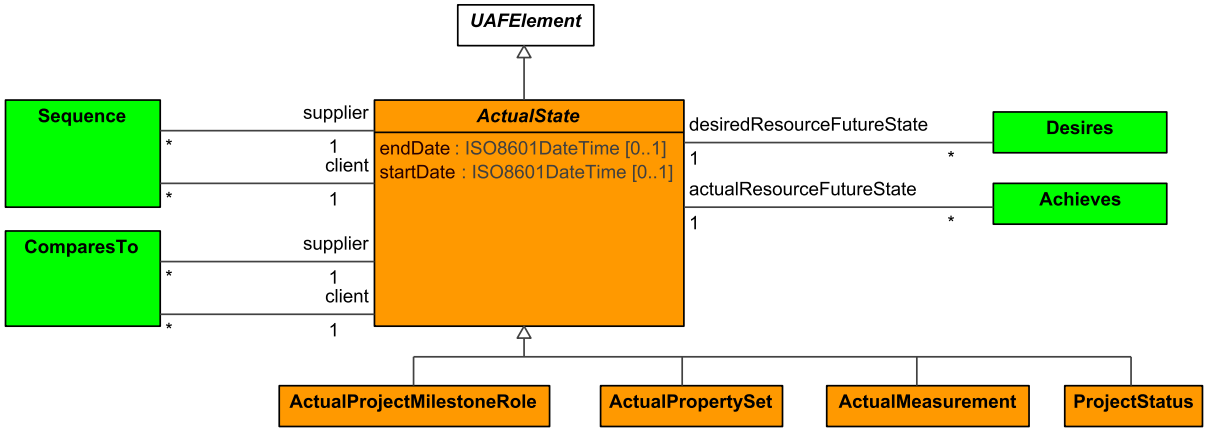

Figure 9:1 - ActualState

Attributes

endDate : ISO8601DateTime [0..1] End time for all individual elements.
startDate : ISO8601DateTime [0..1] Start time for all individual elements.

DesignationKind

Package: Taxonomy
isAbstract: No
Generalization: MeasurableElement

Description

An enumerated type that defines the kind of structural element with respect to purpose, intent and capabilities of that element.

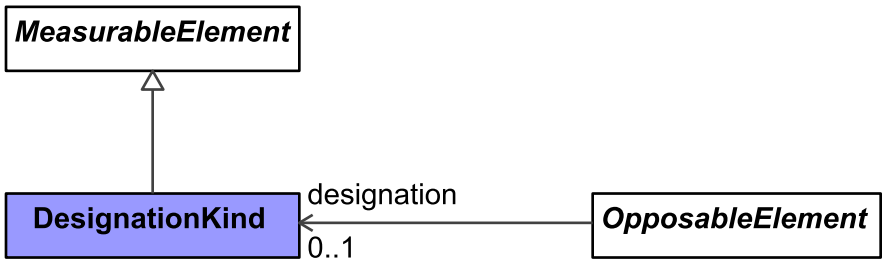

Figure 9:2 - DesignationKind

InteractionScenarioGeneralization

Package: Taxonomy
isAbstract: No
Generalization: UML2.5Metamodel::Generalization, MeasurableElement

Description

A InteractionScenarioGeneralization is a taxonomic relationship between a more general InteractionScenario and a more specific InteractionScenario.

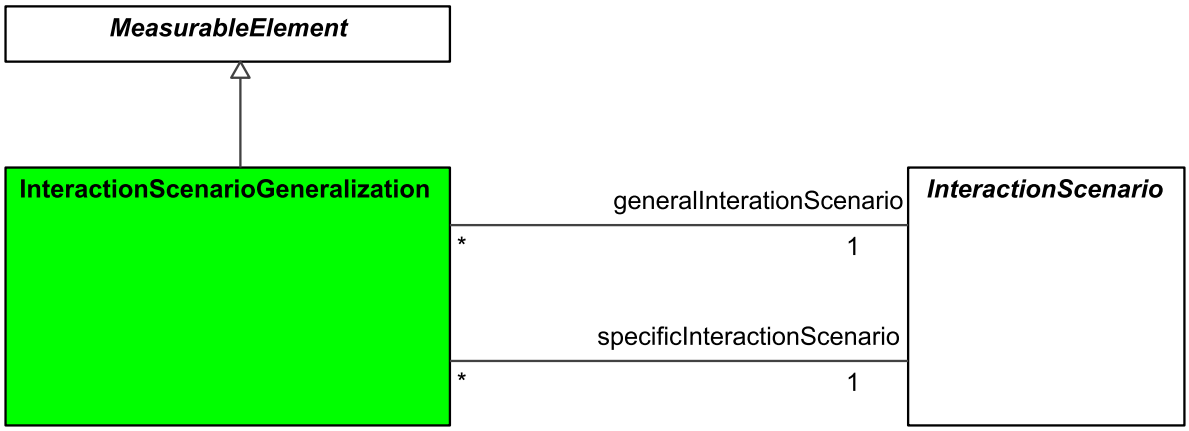

Figure 9:3 - InteractionScenarioGeneralization

ISO8601DateTime

Package: Taxonomy
isAbstract: No
Generalization: UAFElement

Description

A date and time specified in the ISO8601 date-time format including timezone designator (TZD): YYYY-MM-DDThh:mm:ssTZD.

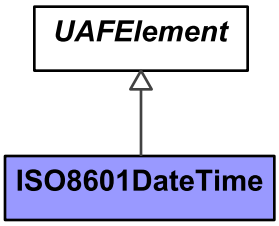

Figure 9:4 - ISO8601DateTime

OpposableElement

Package: Taxonomy
isAbstract: Yes
Generalization: UAFElement

Description

A type of element that is categorized to allow for it to have an Opposes relationship to some other opposable element.

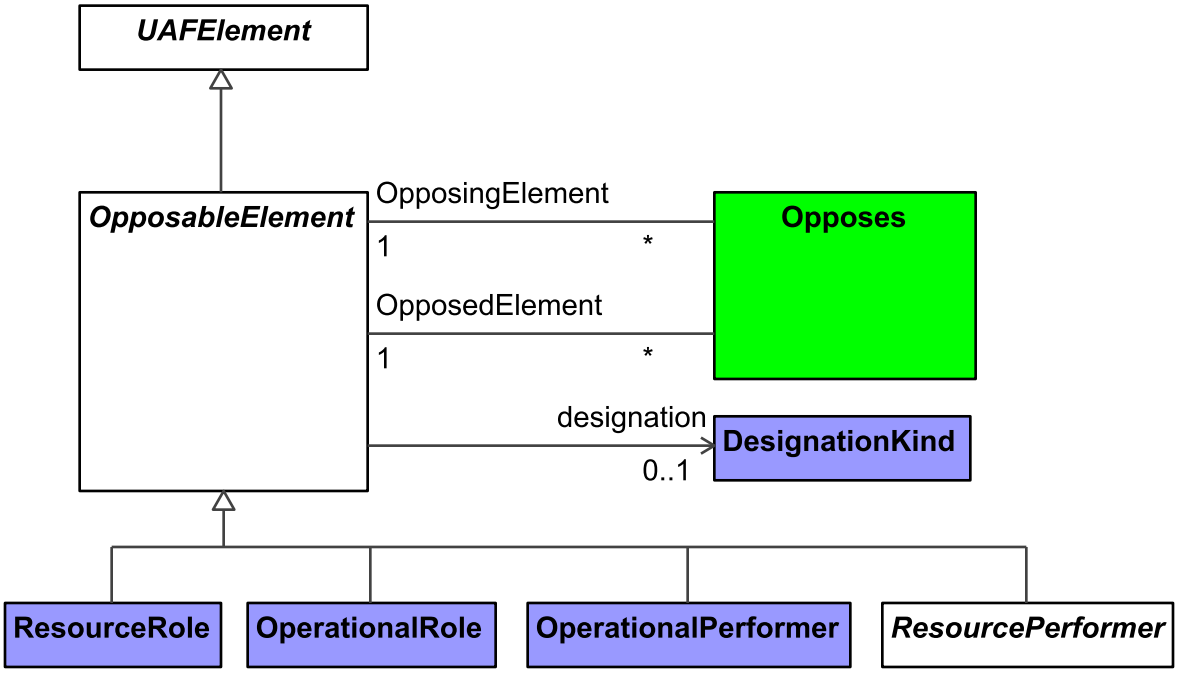

Figure 9:5 - OpposableElement

ProcessGeneralization

Package: Taxonomy
isAbstract: No
Generalization: UML2.5Metamodel::Generalization, MeasurableElement

Description

A ProcessGeneralization is a taxonomic relationship between a more general Process and a more specific Process.

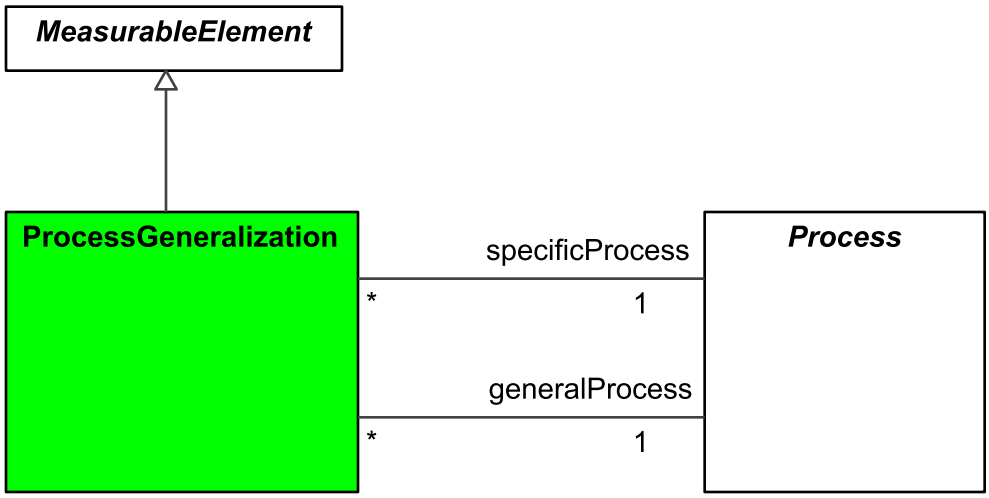

Figure 9:6 - ProcessGeneralization

PropertySetGeneralization

Package: Taxonomy
isAbstract: No
Generalization: UML2.5Metamodel::Generalization, MeasurableElement

Description

A PropertySetGeneralization is a taxonomic relationship between a more general PropertySet and a more specific PropertySet.

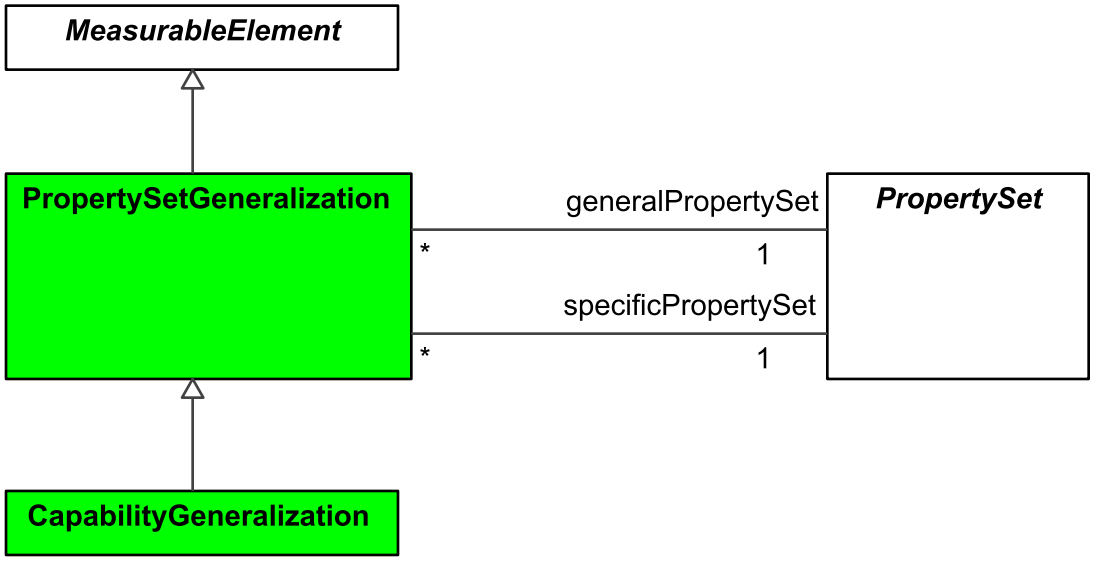

Figure 9:7 - PropertySetGeneralization

StateDescriptionGeneralization

Package: Taxonomy
isAbstract: No
Generalization: UML2.5Metamodel::Generalization, MeasurableElement

Description

A StateDescriptionGeneralization is a taxonomic relationship between a more general StateDescription and a more specific StateDescription.

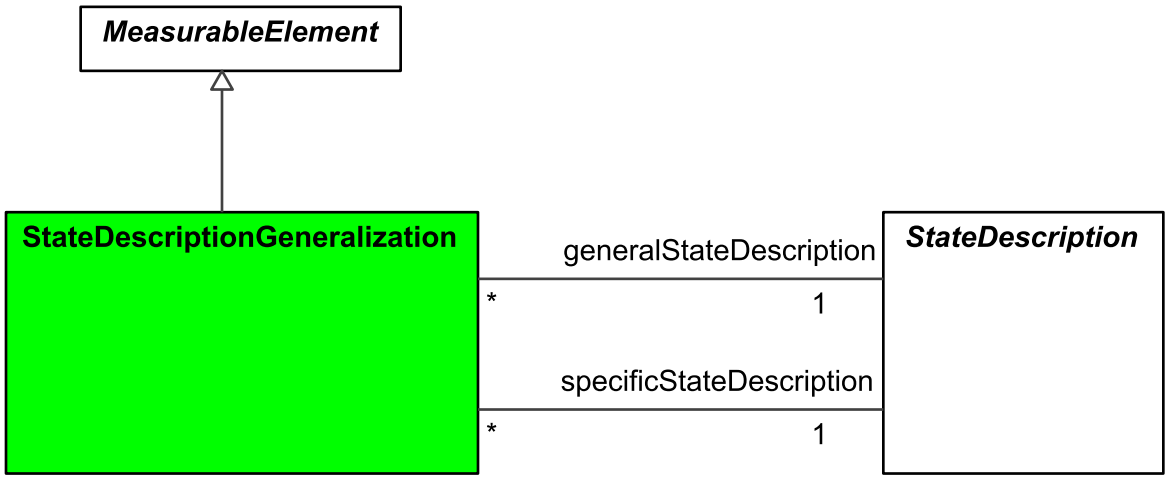

Figure 9:8 - StateDescriptionGeneralization

9.1.1.2 Domain MetaModel::Architecture Management::Connectivity

Exchange

Package: Connectivity
isAbstract: Yes
Generalization: MeasurableElement, BPMN2Metamodel::MessageFlow, SubjectOfSecurityConstraint

Description

Abstract tuple, grouping OperationalExchanges and ResourceExchanges that exchange Resources.

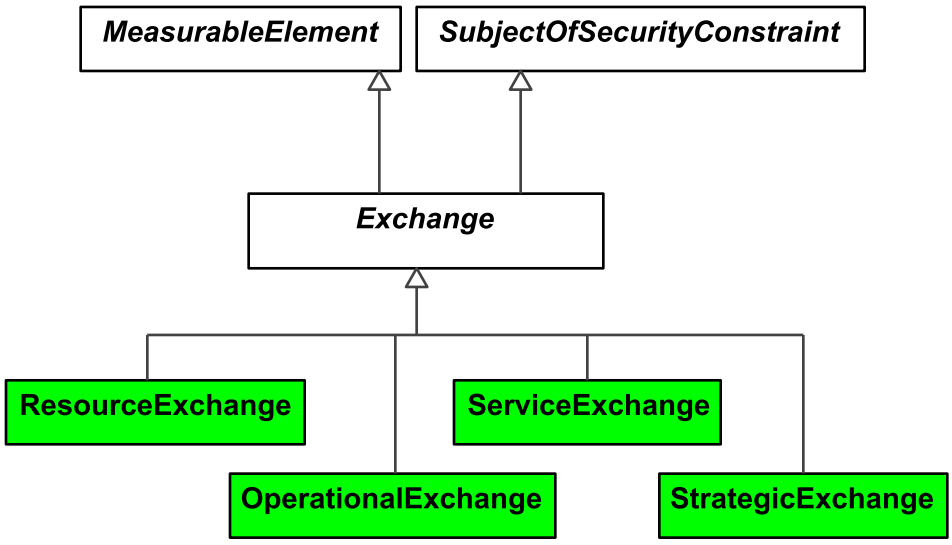

Figure 9:9 - Exchange

ExchangeItem

Package: Connectivity
isAbstract: Yes
Generalization: BPMN2Metamodel::BPMNMessage

Description

An abstract grouping for elements that defines the types of elements that can be exchanged between Assets and conveyed by an Exchange.

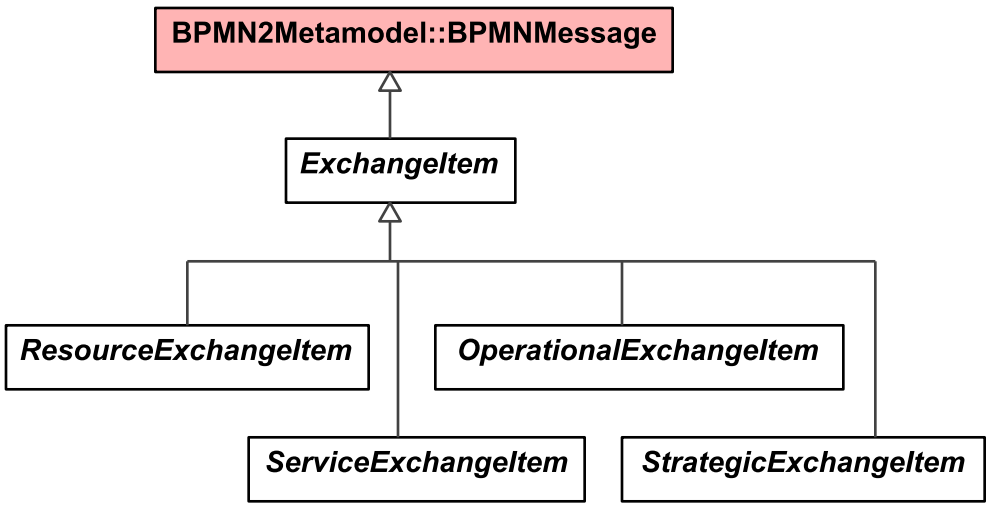

Figure 9:10 - ExchangeItem

Resource

Package: Connectivity
isAbstract: Yes
Generalization: PropertySet

Description

Abstract type grouping all elements that can be conveyed by an Exchange.

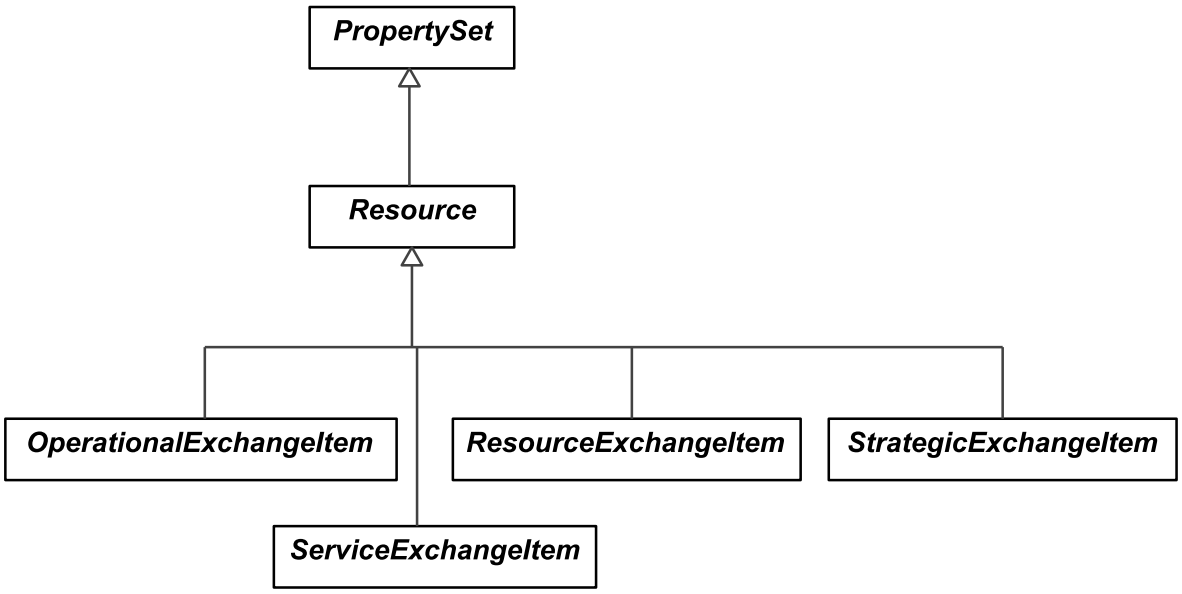

Figure 9:11 - Resource

9.1.1.3 Domain MetaModel::Architecture Management::Processes

CapableElement

Package: Processes
isAbstract: Yes
Generalization: UAFElement

Description

An abstract type that represents a structural element that can exhibit capabilities.

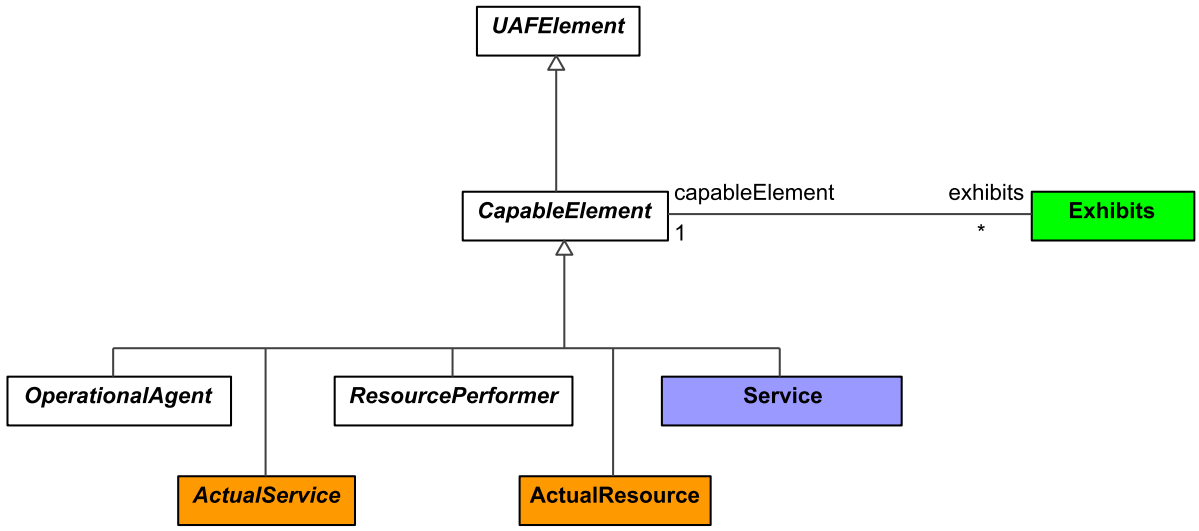

Figure 9:12 - CapableElement

IsCapableToPerform

Package: Processes
isAbstract: No
Generalization: MeasurableElement

Description

A tuple defining the traceability between the structural elements to the Activities that they can perform.

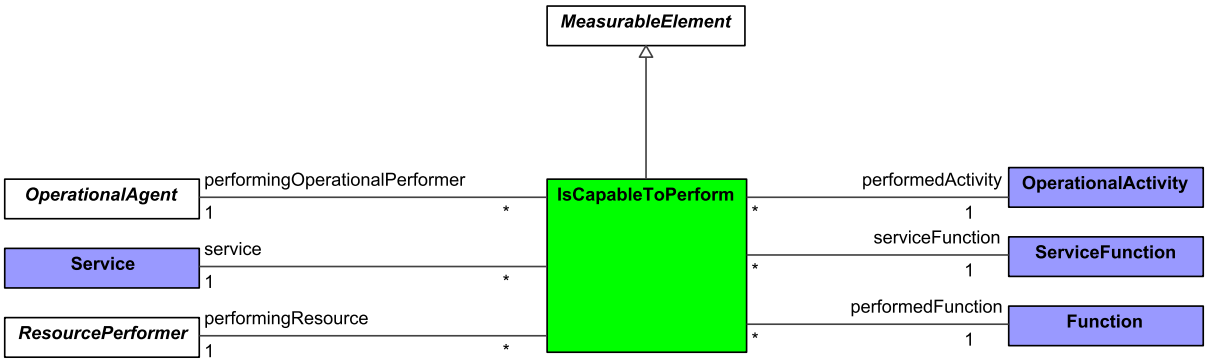

Figure 9:13 - IsCapableToPerform

PerformsInContext

Package: Processes
isAbstract: No
Generalization: MeasurableElement

Description

A tuple that relates an OperationalAction to a OperationalRole, or a FunctionAction to a ResourceRole. It indicates that the action can be carried out by the role when used in a specific context or configuration.

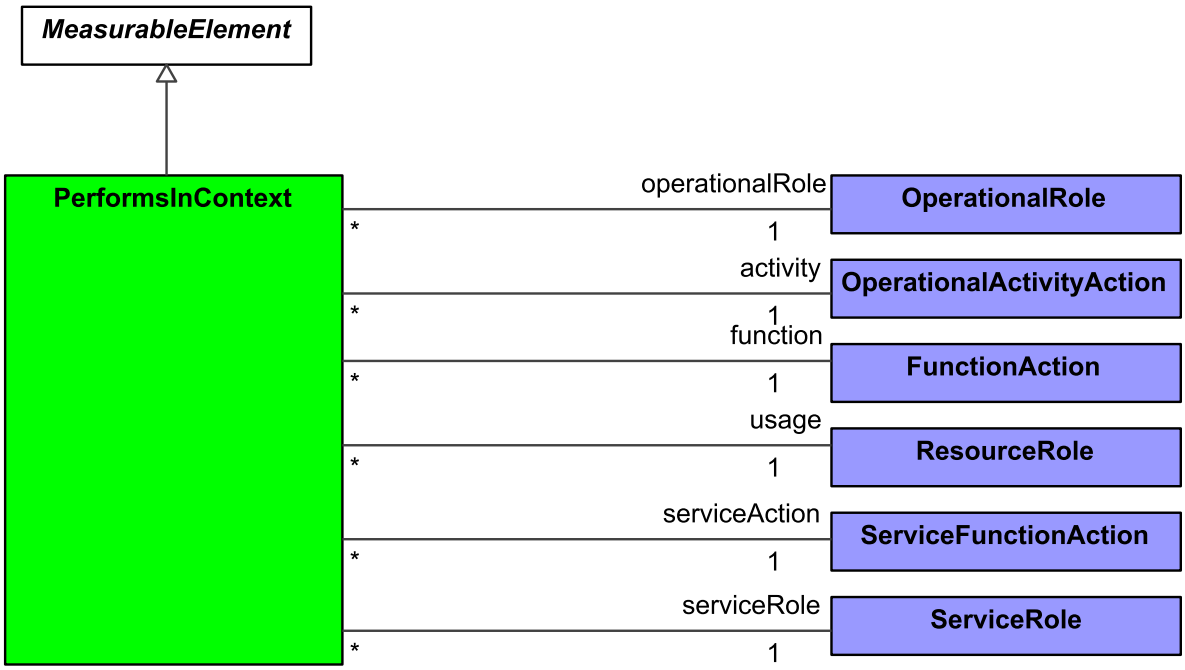

Figure 9:14 - PerformsInContext

Process

Package: Processes
isAbstract: Yes
Generalization: UML2.5Metamodel::Activity, BPMN2Metamodel::Process, AffectableElement, PropertySet

Description

An abstract type that represents a behavior or process (i.e. a Function or OperationalActivity) that can be performed by a Performer.

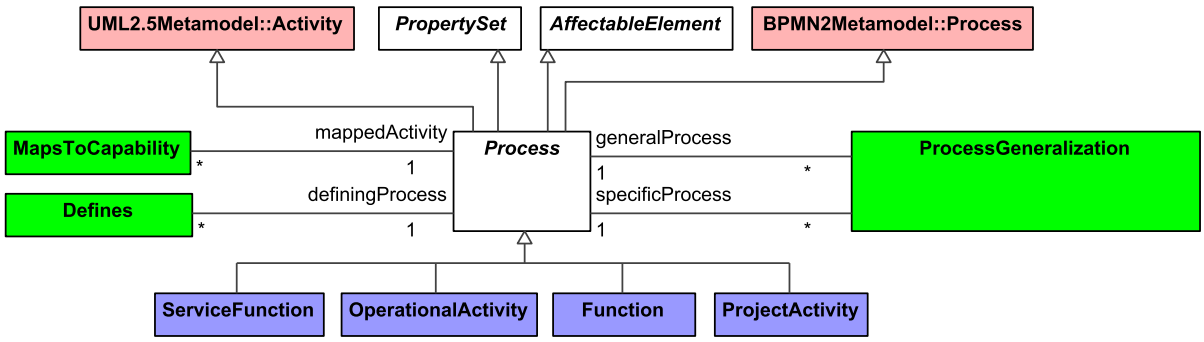

Figure 9:15 - Process

ProcessEdge

Package: Processes
isAbstract: Yes
Generalization: MeasurableElement, UML2.5Metamodel::Activity, UML2.5Metamodel::ActivityEdge, BPMN2Metamodel::SequenceFlow

Description

An abstract type that represents a behavior or process (i.e. a Function or OperationalActivity) that can be performed by a Performer.

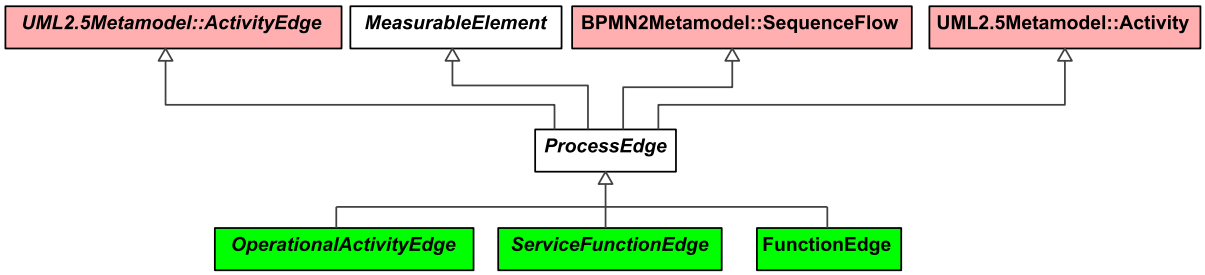

Figure 9:16 - ProcessEdge

ProcessOperation

Package: Processes
isAbstract: Yes
Generalization: MeasurableElement, UML2.5Metamodel::Activity, UML2.5Metamodel::Operation

Description

An abstract type that represents a behavior or process (i.e. a Function or OperationalActivity) that can be performed by a Performer.

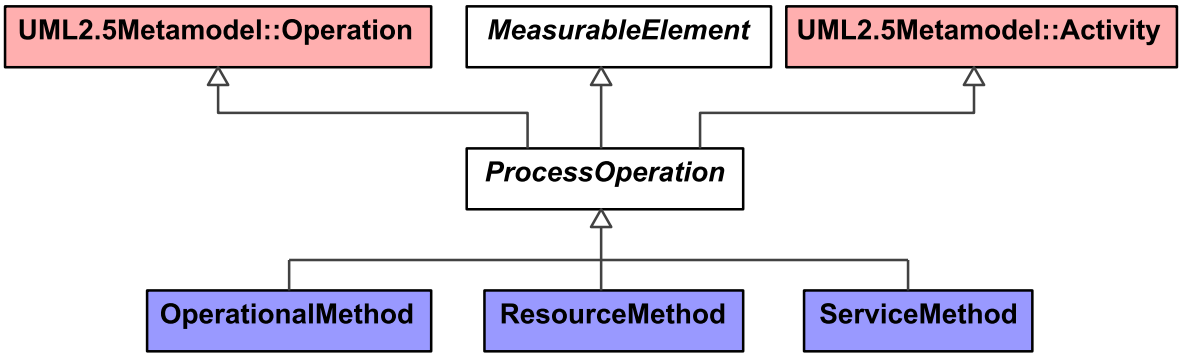

Figure 9:17 - ProcessOperation

ProcessParameter

Package: Processes
isAbstract: Yes
Generalization: MeasurableElement, UML2.5Metamodel::Activity, UML2.5Metamodel::CallBehaviorAction, UML2.5Metamodel::Parameter

Description

An abstract type that represents a behavior or process (i.e. a Function or OperationalActivity) that can be performed by a Performer.

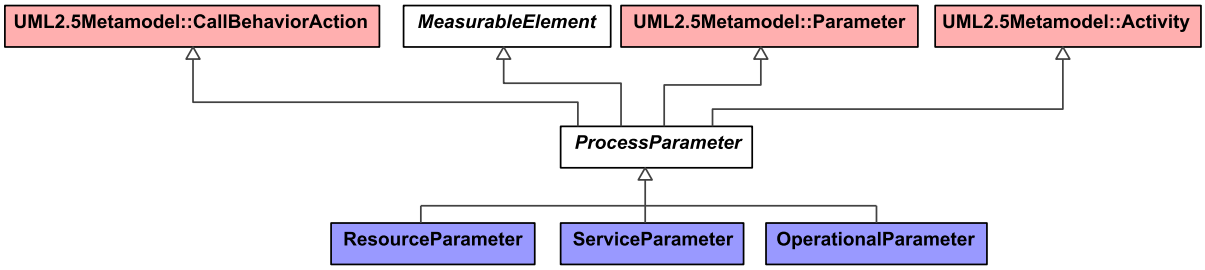

Figure 9:18 - ProcessParameter

ProcessUsage

Package: Processes
isAbstract: Yes
Generalization: MeasurableElement, UML2.5Metamodel::Activity, UML2.5Metamodel::CallBehaviorAction, BPMN2Metamodel::CallActivity

Description

An abstract type that represents a behavior or process (i.e. a Function or OperationalActivity) that can be performed by a Performer.

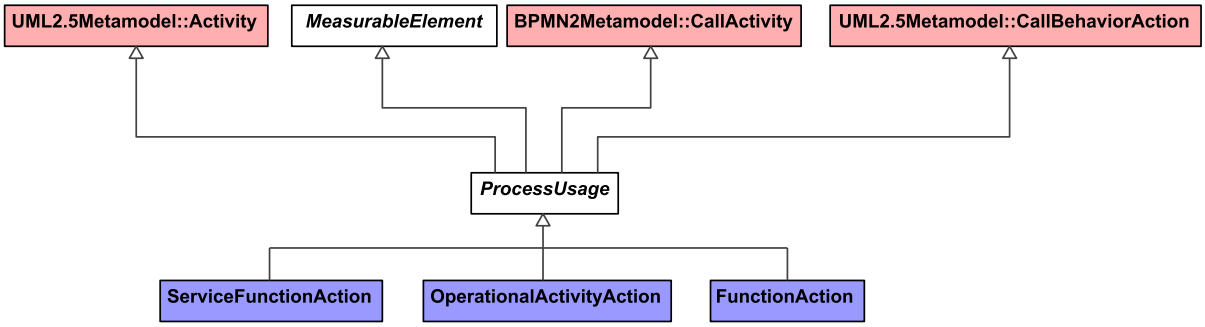

Figure 9:19 - ProcessUsage

9.1.1.4 Domain MetaModel::Architecture Management::States

StateDescription

Package: States
isAbstract: Yes
Generalization: UML2.5Metamodel::StateMachine

Description

An abstract type that represents a state machine (i.e. an OperationalStateDescription or ResourceStateDescription), depicting how the Asset responds to various events and the actions.

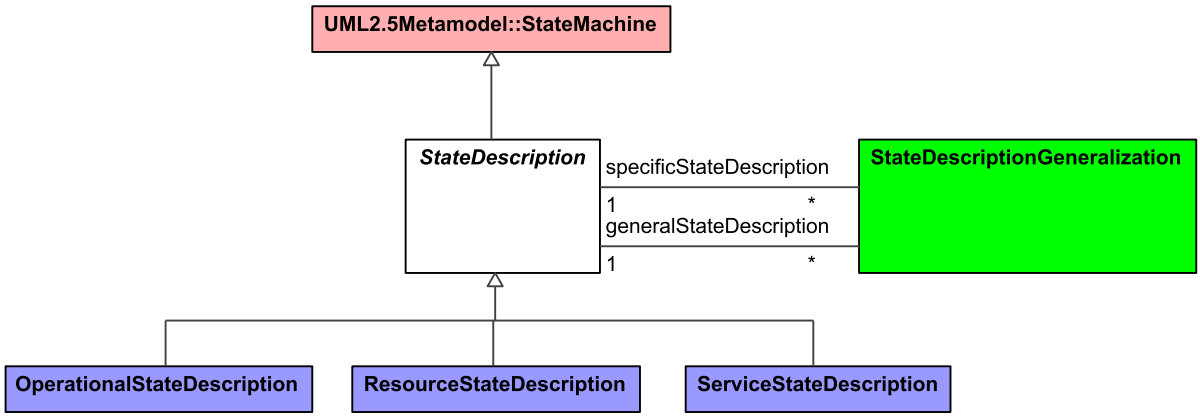

Figure 9:20 - StateDescription

9.1.1.5 Domain MetaModel::Architecture Management::Sequences

InteractionMessage

Package: Sequences
isAbstract: Yes
Generalization: MeasurableElement, UML2.5Metamodel::Activity, BPMN2Metamodel::Process, UML2.5Metamodel::Interaction, UML2.5Metamodel::Message

Description

An abstract type that groups several types of messages used in the InteractionScenario.

InteractionRole

Package: Sequences
isAbstract: Yes
Generalization: BPMN2Metamodel::ResourceRole

Description

An abstract type that represents an individual participant in the InteractionScenario.

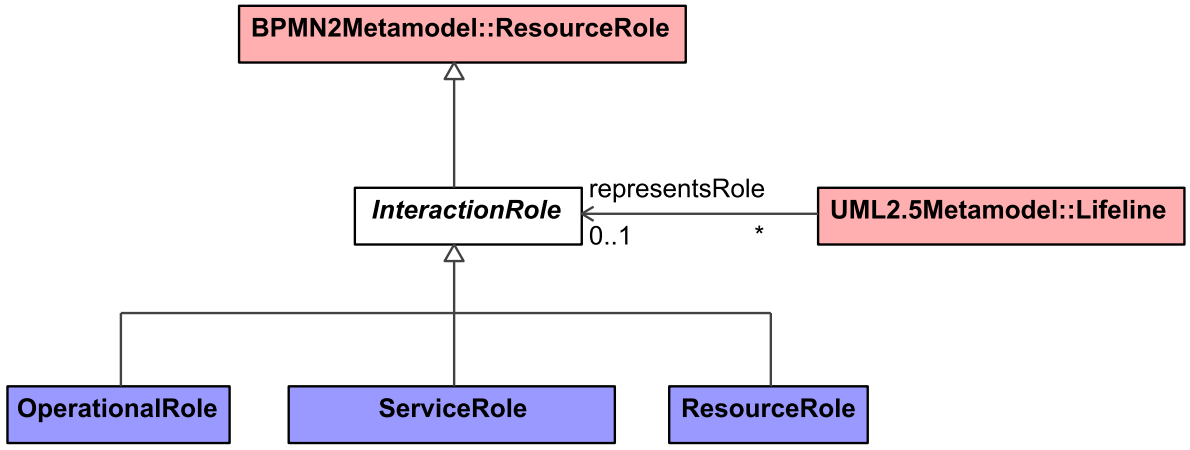

Figure 9:21 - InteractionRole

InteractionScenario

Package: Sequences
isAbstract: Yes
Generalization: MeasurableElement, UML2.5Metamodel::Activity, BPMN2Metamodel::Process, UML2.5Metamodel::Interaction

Description

An abstract type that specifies interactions between Assets, like ResourcePerformers, and Services.

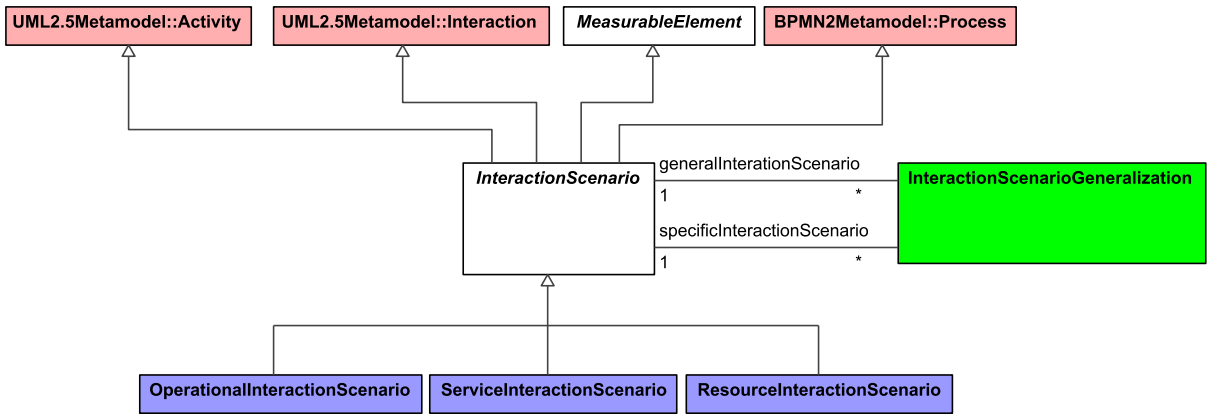

Figure 9:22 - InteractionScenario

9.1.1.6 Domain MetaModel::Architecture Management::Information

Alias

Package: Information
isAbstract: No
Generalization: MeasurableElement

Description

A metamodel Artifact used to define an alternative name for an element.

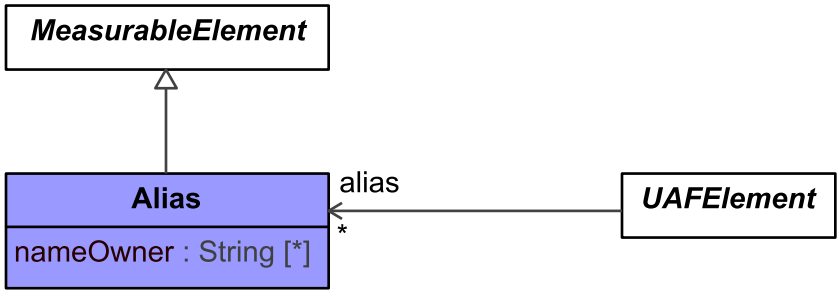

Figure 9:23 - Alias

Attributes

nameOwner : String [*] Someone or something that uses this alternative name.

ArchitectureMetadata

Package: Information
isAbstract: No
Generalization: Metadata

Description

Information associated with an ArchitecturalDescription, that supplements the standard set of tags used to summarize the Architecture. It states things like what methodology was used, notation, etc.

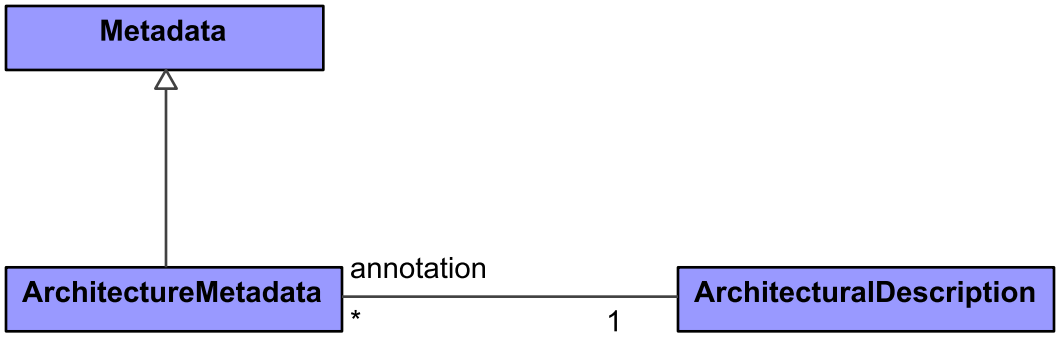

Figure 9:24 - ArchitectureMetadata

ConflictsWith

Package: Information
isAbstract: No
Generalization: MeasurableElement

Description

An annotation that indicates UAF elements acting against or in opposition to each other. This means that there are elements in the model that disagree about something important or are incompatible with each other in some way.

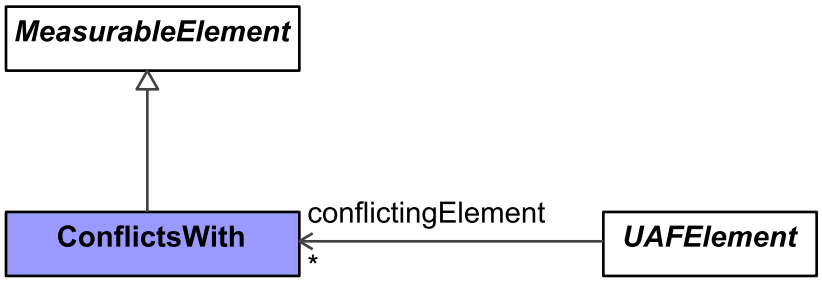

Figure 9:25 - ConflictsWith

Definition

Package: Information
isAbstract: No
Generalization: MeasurableElement

Description

A comment containing a description of an element in the architecture.

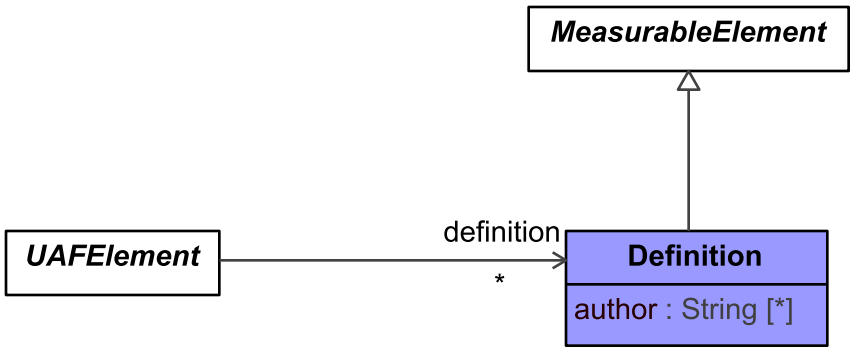

Figure 9:26 - Definition

Attributes

author : String [*] The original or current person (architect) responsible for the Definition.

Information

Package: Information
isAbstract: No
Generalization: MeasurableElement

Description

A comment that describes the state of an item of interest in any medium or form -- and is communicated or received.

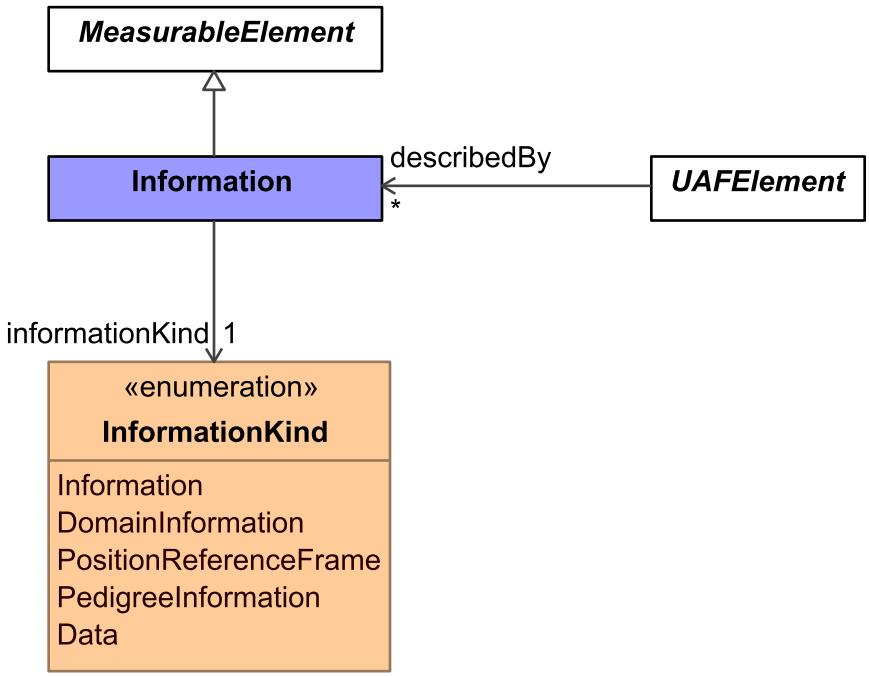

Figure 9:27 - Information

InformationModel

Package: Information
isAbstract: No
Generalization: SubjectOfOperationalConstraint, SubjectOfResourceConstraint

Description

A structural specification of data types, showing relationships between them. The type of information captured in the InformationModel is described using the enumeration InformationModelKind (Conceptual, Logical, and Physical).

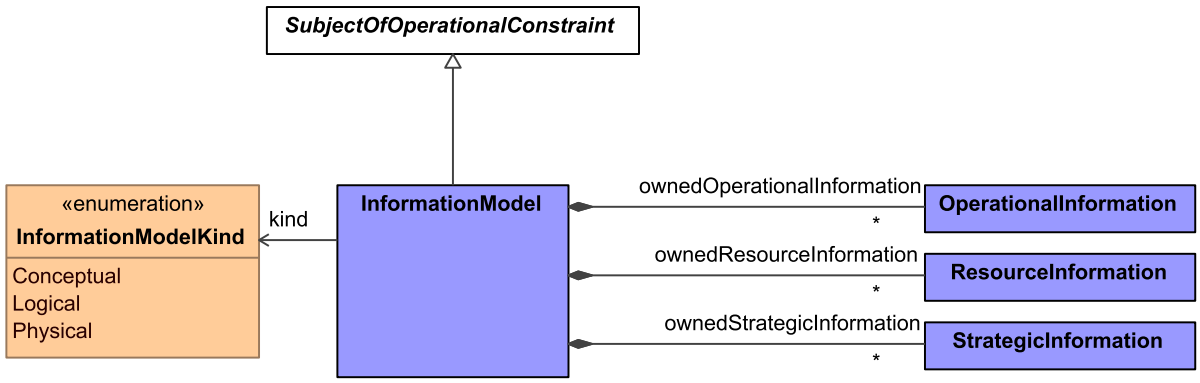

Figure 9:28 - InformationModel

Metadata

Package: Information
isAbstract: No
Generalization: MeasurableElement

Description

A comment that can be applied to any element in the architecture. The attributes associated with this element details the relationship between the element and its related dublinCoreElement, metaDataScheme, category and name. This allows the element to be referenced using the Semantic Web.

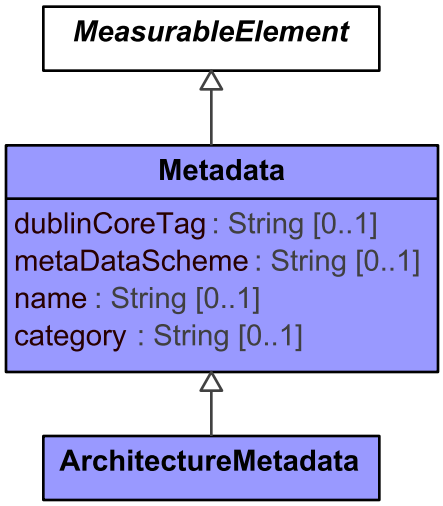

Figure 9:29 - Metadata

Attributes

category : String [0..1]	Defines the category of a Metadata element example: http://purl.org/dc/terms/abstract.
dublinCoreTag : String [0..1]	A metadata category that is a DublinCore tag.
metaDataScheme : String [0..1]	A representation scheme that defines a set of Metadata.
name : String [0..1]	The name of the Metadata.

Opposes

Package: Information
isAbstract: No
Generalization: MeasurableElement

Description

A tuple that indicates an open clash between two opposing groups (or individuals). This means that elements in the model are fighting against each other or are providing strong resistance to progress towards some goals. Note to entry: By individual this does not indicate instances in the model.

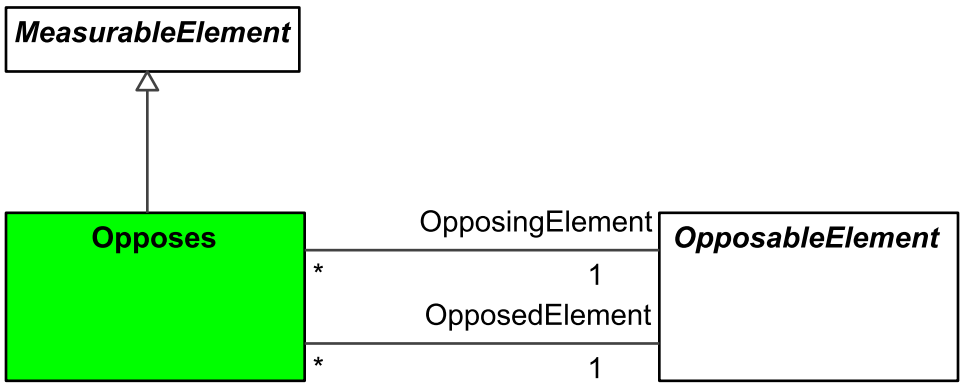

Figure 9:30 - Opposes

SameAs

Package: Information
isAbstract: No
Generalization: MeasurableElement

Description

A tuple that asserts that two elements refer to the same real-world thing.

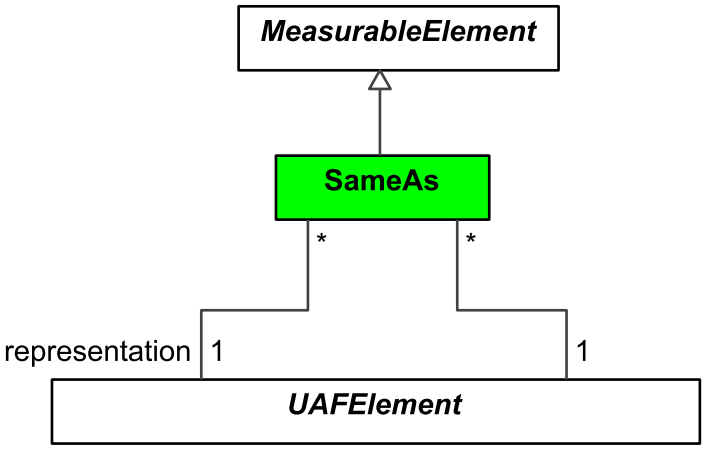

Figure 9:31 - SameAs

9.1.1.7 Domain MetaModel::Architecture Management::Constraints

Rule

Package: Constraints
isAbstract: Yes
Generalization: MeasurableElement

Description

An abstract type for all types of constraint (i.e. an OperationalConstraint could detail the rules of accountancy best practice).

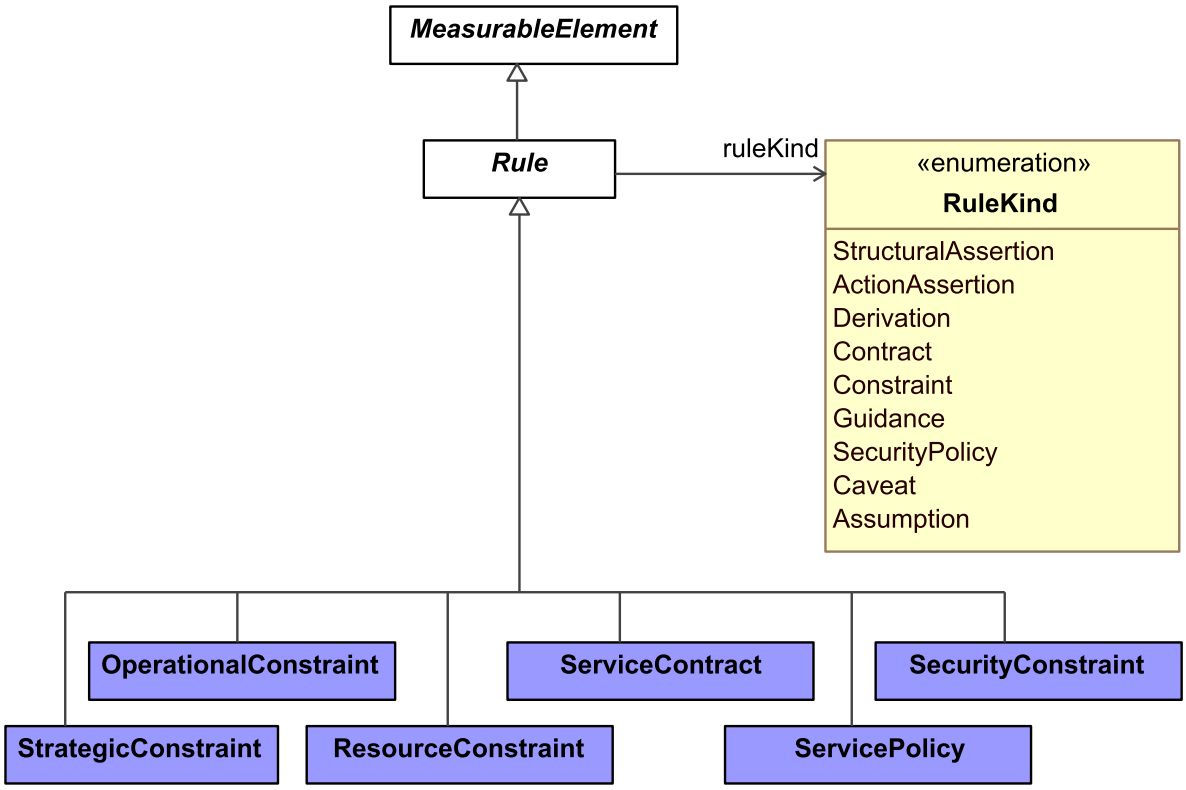

Figure 9:32 - Rule

9.1.1.8 Domain MetaModel::Architecture Management::Traceability

ArchitecturalReference

Package: Traceability
isAbstract: No
Generalization: MeasurableElement

Description

A tuple that specifies that one architectural description refers to another.

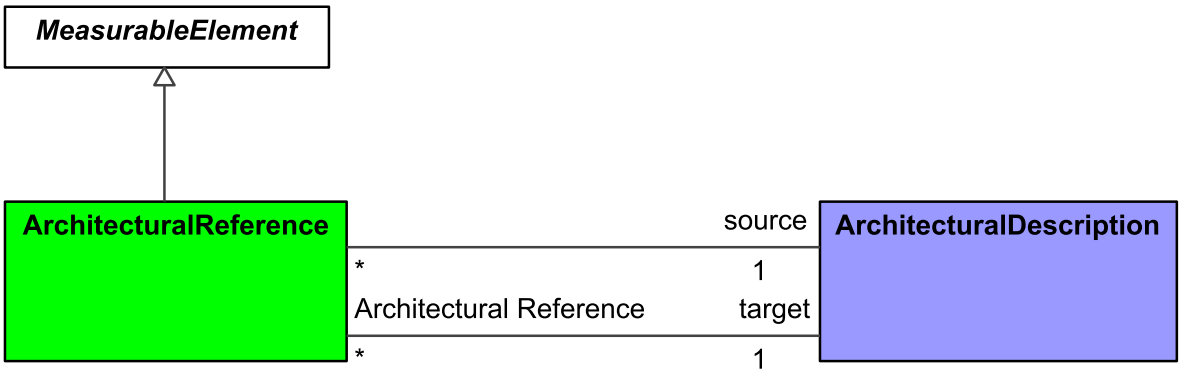

Figure 9:33 - ArchitecturalReference

ComparesTo

Package: Traceability
isAbstract: No
Generalization: MeasurableElement

Description

A tuple used to relate the effect that is achieved with the originally expected DesiredEffect. Providing a means of comparison, between the expectation of the desirer and the actual result.

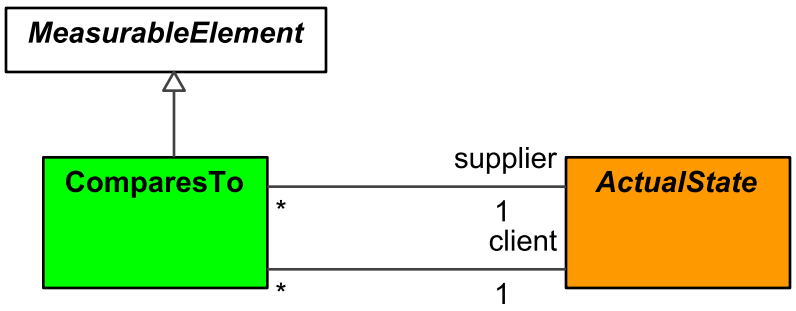

Figure 9:34 - ComparesTo

Implements

Package: Traceability
isAbstract: No
Generalization: MeasurableElement

Description

A tuple that defines how an element in the upper layer of abstraction is implemented by a semantically equivalent element (for example tracing the Functions to the OperationalActivities) in the lower level of abstraction.

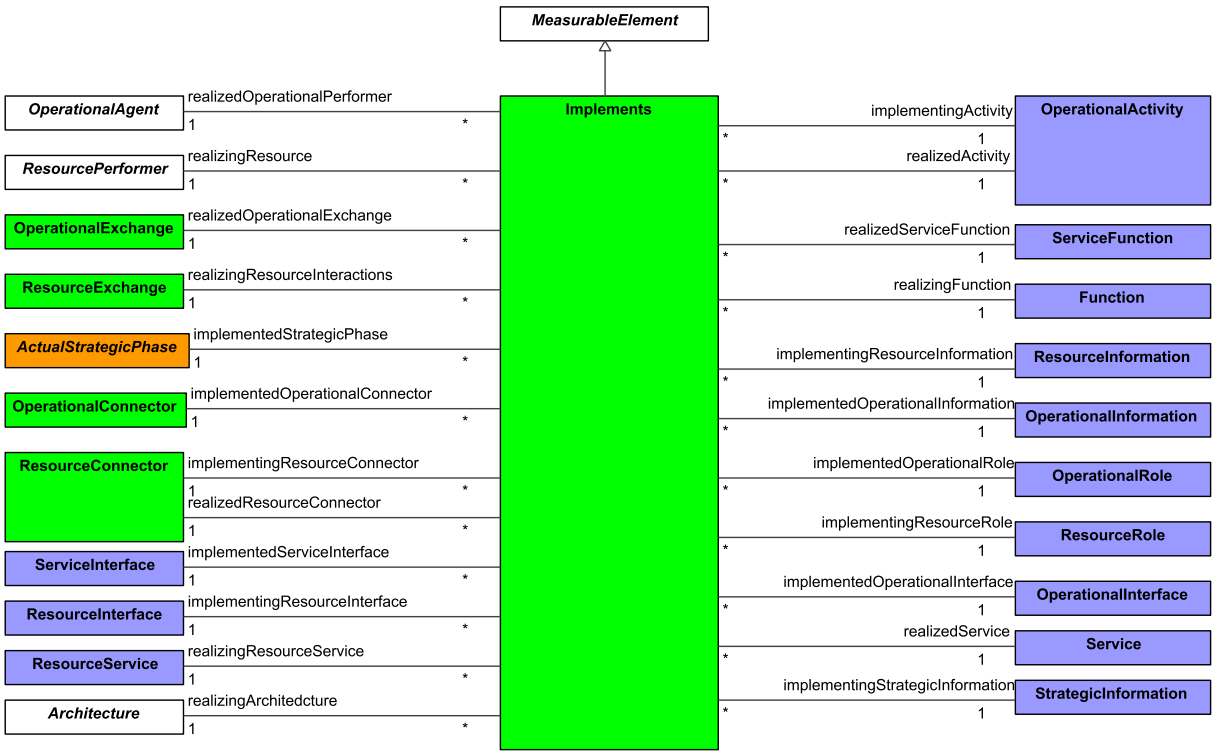

Figure 9:35 - Implements

Sequence

Package: Traceability
isAbstract: No
Generalization: MeasurableElement

Description

A tuple that asserts one Individual's temporal extent is completely before the temporal extent of another.

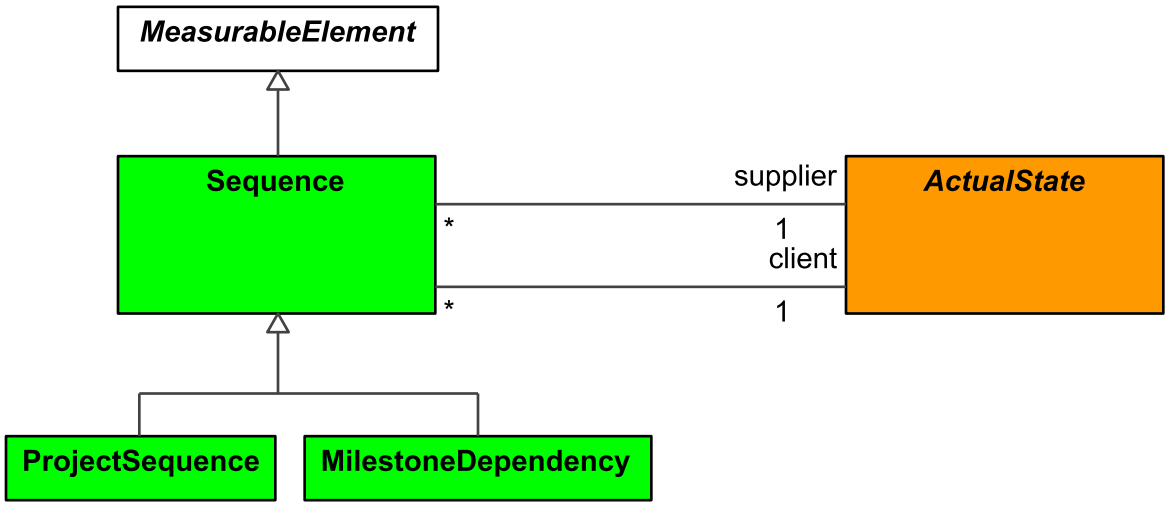

Figure 9:36 - Sequence

9.1.2 Domain MetaModel::Summary & Overview

ArchitecturalDescription

Package: Summary & Overview
isAbstract: No
Generalization: MeasurableElement

Description

An Architecture Description is a work product used to express the Architecture of some System Of Interest.

It provides executive-level summary information about the architecture description in a consistent form to allow quick reference and comparison between architecture descriptions -- It includes assumptions, constraints, and limitations that may affect high-level decisions relating to an architecture-based work program.

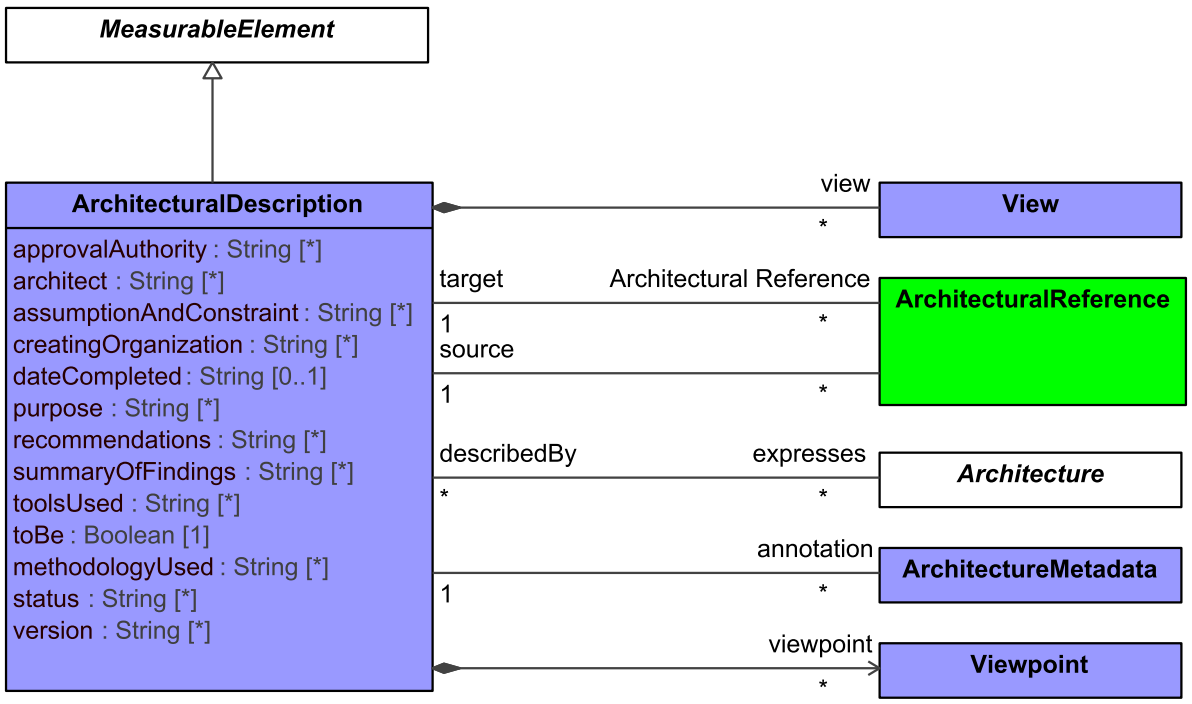

Figure 9:37 - ArchitecturalDescription

Attributes

approvalAuthority : String [*]	Someone or something that has the authority to approve the ArchitecturalDescription.
architect : String [*]	Someone responsible for the creation of ArchitecturalDescription.
assumptionAndConstraint : String [*]	Any assumptions, constraints, and limitations contained in the ArchitecturalDescription, including those affecting deployment, communications performance, information assurance environments, etc.
creatingOrganization : String [*]	The organization responsible for creating the ArchitecturalDescription.
dateCompleted : String [0..1]	Date that the ArchitecturalDescription was completed.
methodologyUsed : String [*]	Name of the documented methodology that will be or has been used in describing the architecture.
purpose : String [*]	Explains the need for the Architecture, what it will demonstrate, the types of analyses that will be applied to it, who is expected to perform the

	analyses, what decisions are expected to be made on the basis of each form of analysis, who is expected to make those decisions, and what actions are expected to result.
recommendations : String [*]	States the recommendations that have been developed based on the architecture effort. Examples include recommended system implementations, and opportunities for technology insertion.
status : String [*]	State of the architecture description in terms of its development, baselining, activity (e.g., active or inactive), or some other factor of importance.
summaryOfFindings : String [*]	Summarizes the findings that have been developed so far. This may be updated several times during the development of the ArchitecturalDescription.
toBe : Boolean [1]	Indicates whether the ArchitecturalDescription represents an Architecture that exists or will exist in the future.
toolsUsed : String [*]	Identifies any tools used to develop the ArchitecturalDescription as well as file names and formats if appropriate.
version : String [*]	

Architecture

Package: Summary & Overview
isAbstract: Yes
Generalization: UAFElement

Description

An abstract type that represents a generic architecture. Subtypes are OperationalArchitecture, Service Architecture, and ResourceArchitecture.

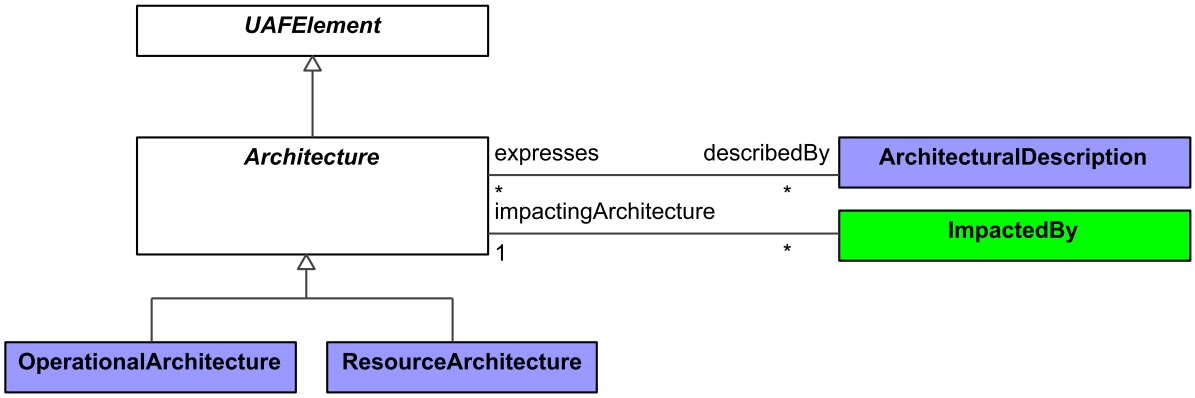

Figure 9:38 - Architecture

Concern

Package: Summary & Overview
isAbstract: No
Generalization: PropertySet, PhaseableElement

Description

A matter of relevance or importance to a stakeholder regarding an entity of interest.

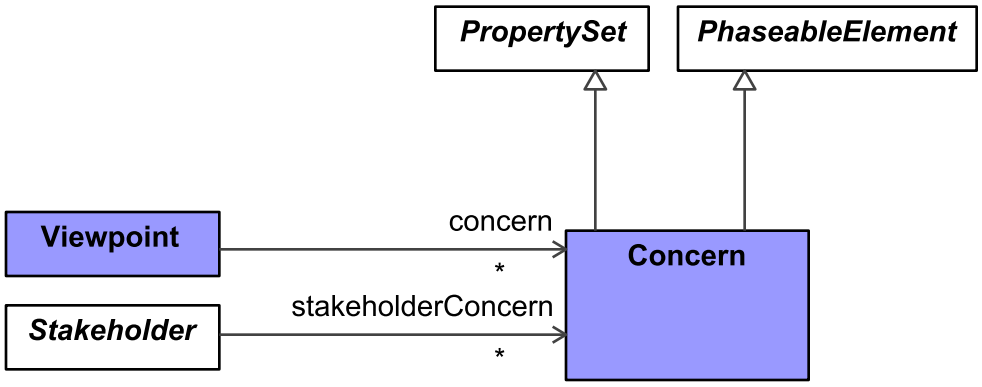

Figure 9:39 - Concern

Phases

Package: Summary & Overview
isAbstract: No
Generalization: MeasurableElement

Description

A tuple that exists between a PhaseableElement and an ActualStrategicPhase that it is assigned to.

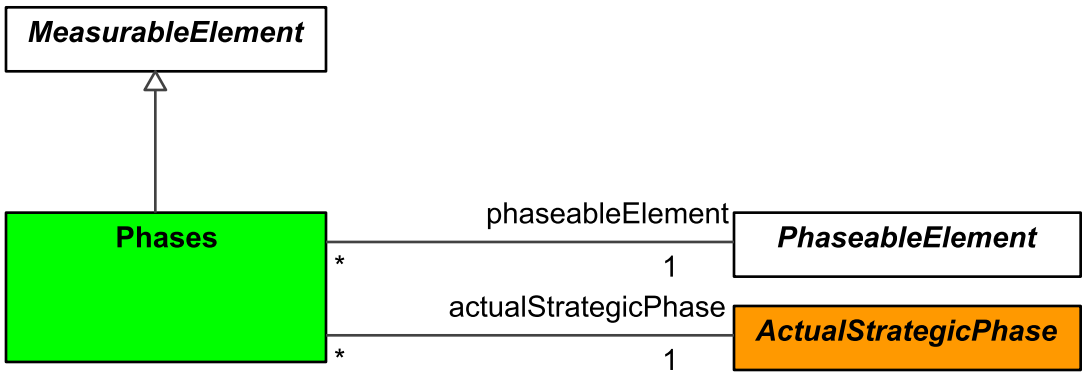

Figure 9:40 - Phases

Stakeholder

Package: Summary & Overview
isAbstract: Yes
Generalization: UAFElement

Description

An individual organizational resource, or a type of organizational resource (both internal and external to the enterprise) who has an interest in, or is affected by, outcomes or intermediate effects generated or influenced by the enterprise.

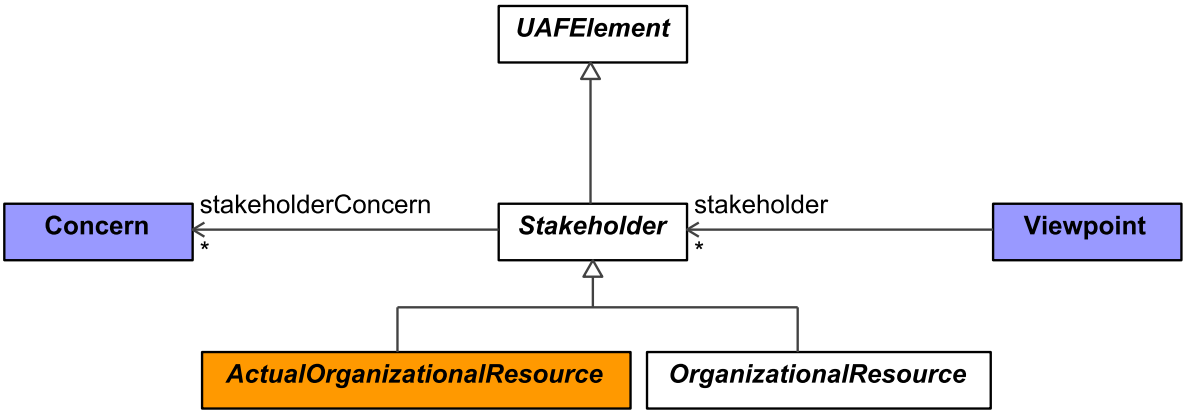

Figure 9:41 - Stakeholder

UAFElement

Package: Summary & Overview
isAbstract: Yes

Description

Abstract super type for all of the UAF elements. It provides a way for all of the UAF elements to have a common set of properties.

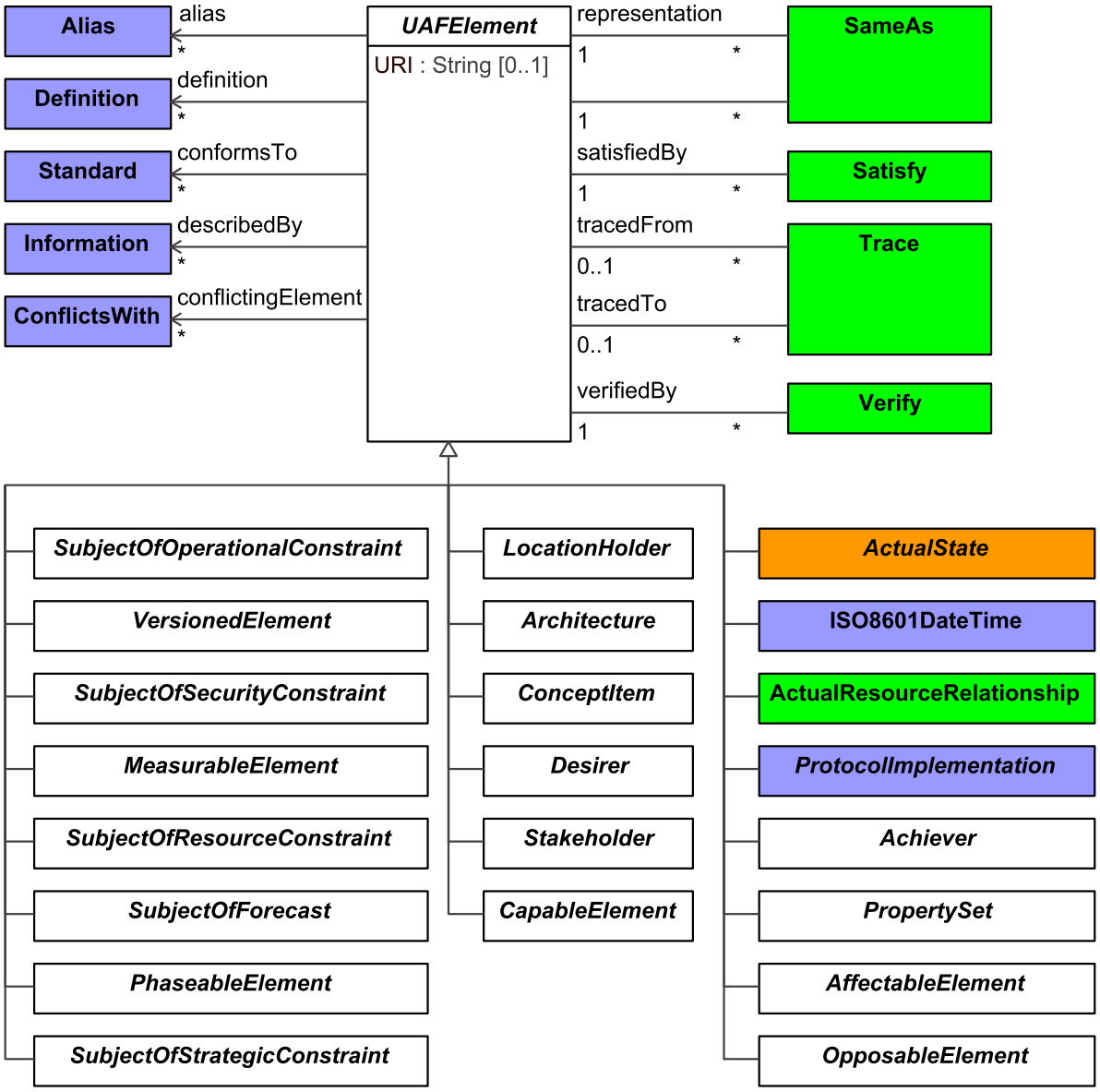

Figure 9:42 - UAFElement

Attributes

URI : String [0..1] Captures Unique identifier for the element.

View

Package: Summary & Overview
isAbstract: No
Generalization: PropertySet

Description

An information item, governed by an architecture viewpoint, comprising part of an architecture description that communicates some aspect of an architecture.

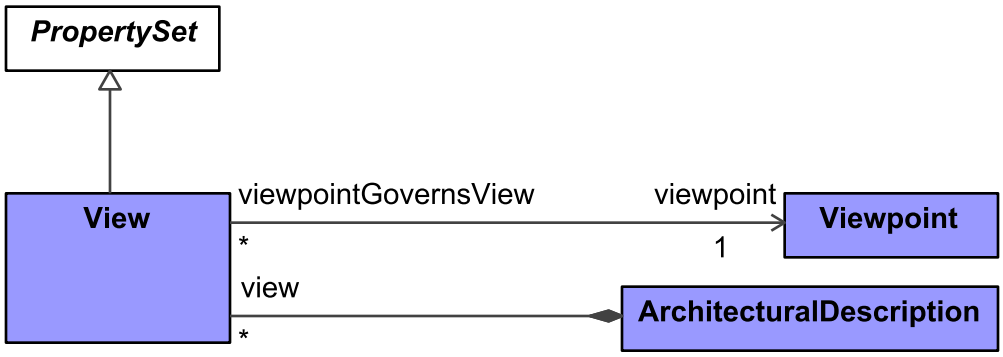

Figure 9:43 - View

Viewpoint

Package: Summary & Overview
isAbstract: No
Generalization: PropertySet

Description

Conventions for the creation, interpretation and use of an architecture view to frame one or more concerns that governs the creation of views.

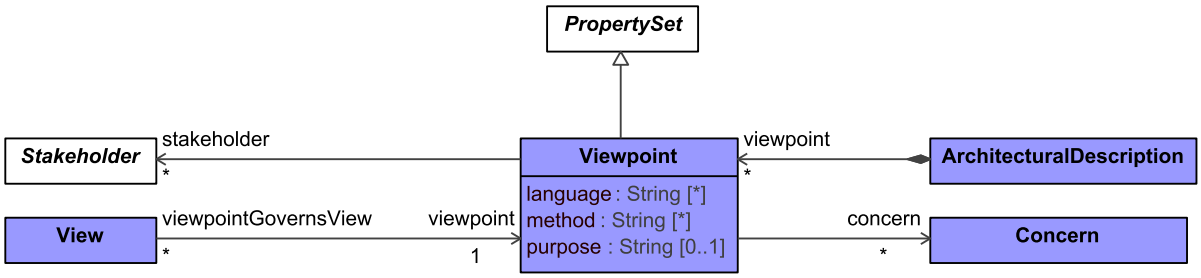

Figure 9:44 - Viewpoint

Attributes

language : String [*] The languages used to express the Viewpoint.
method : String [*] The methods employed in the development of the Viewpoint.
purpose : String [0..1] The purpose of the Viewpoint.

9.1.3 Domain MetaModel::Strategic

9.1.3.1 Domain MetaModel::Strategic::Motivation

Challenge

Package: Motivation
isAbstract: No
Generalization: MotivationalElement

Description

An existing or potential difficulty, circumstance, or obstacle which will require effort and determination from an enterprise to overcome in achieving its goals.

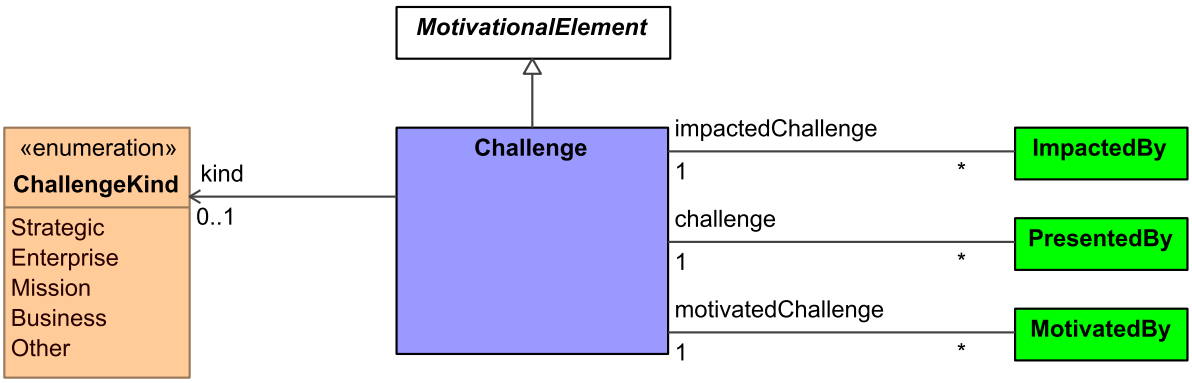

Figure 9:45 - Challenge

Driver

Package: Motivation
isAbstract: No
Generalization: MotivationalElement

Description

A factor which will have a significant impact on the activities, and goals of an enterprise

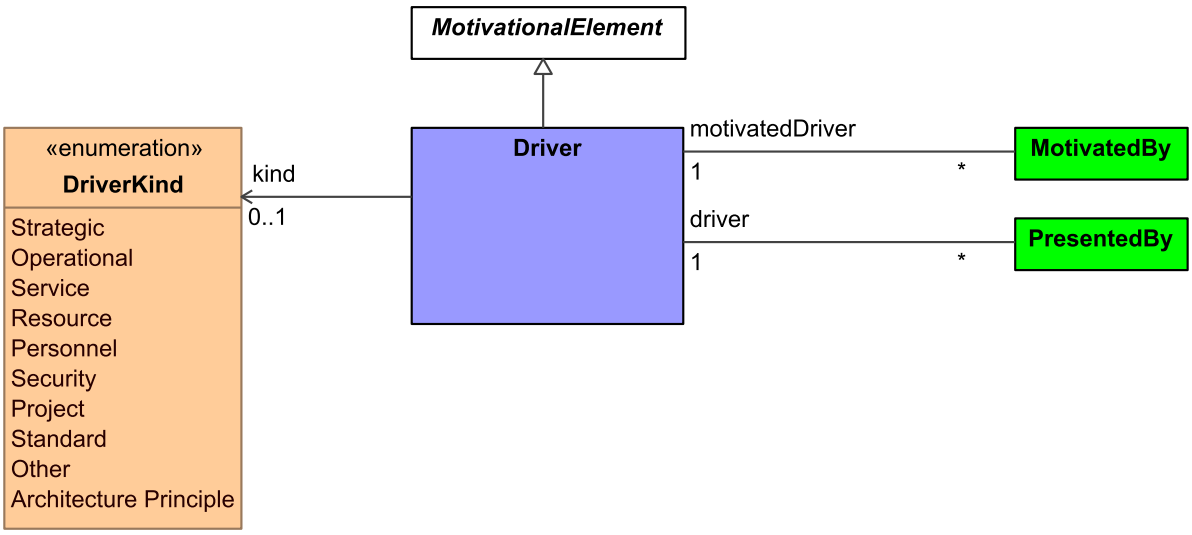

Figure 9:46 - Driver

Enables

Package: Motivation
isAbstract: No
Generalization: MeasurableElement

Description

A tuple used to denote that an Opportunity provides the means for achieving an EnterpriseGoal or Objective.

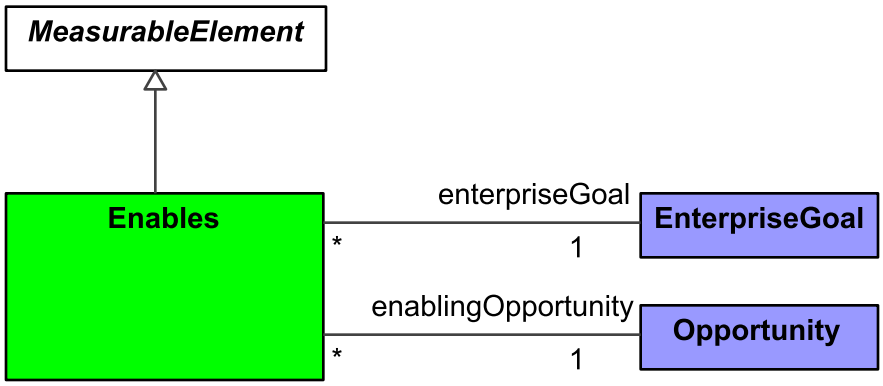

Figure 9:47 - Enables

ImpactedBy

Package: Motivation
isAbstract: No
Generalization: MeasurableElement

Description

A tuple used to denote that a Capability is affected by an Opportunity.

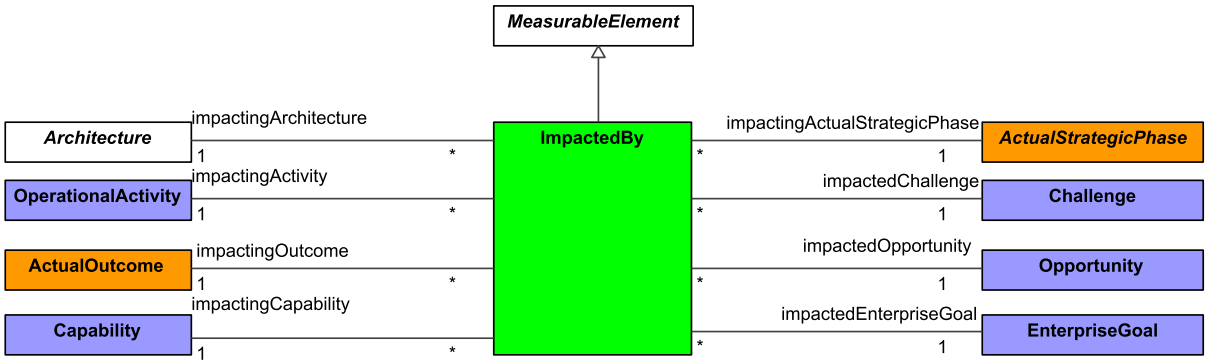

Figure 9:48 - ImpactedBy

MotivatedBy

Package: Motivation
isAbstract: No
Generalization: MeasurableElement

Description

A tuple denoting the reason or reasons one has for acting or behaving in a particular way

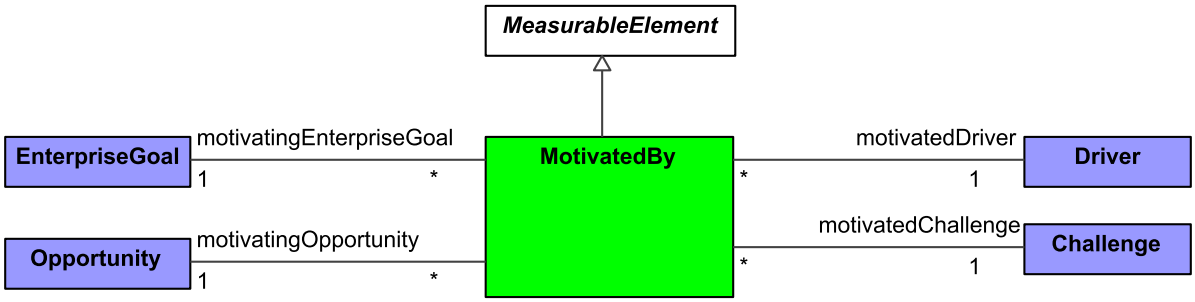

Figure 9:49 - MotivatedBy

MotivationalElement

Package: Motivation
isAbstract: Yes
Generalization: PropertySet, PhaseableElement

Description

An abstract kind of element in the model that provides the reason or reasons one has for acting or behaving in a particular way

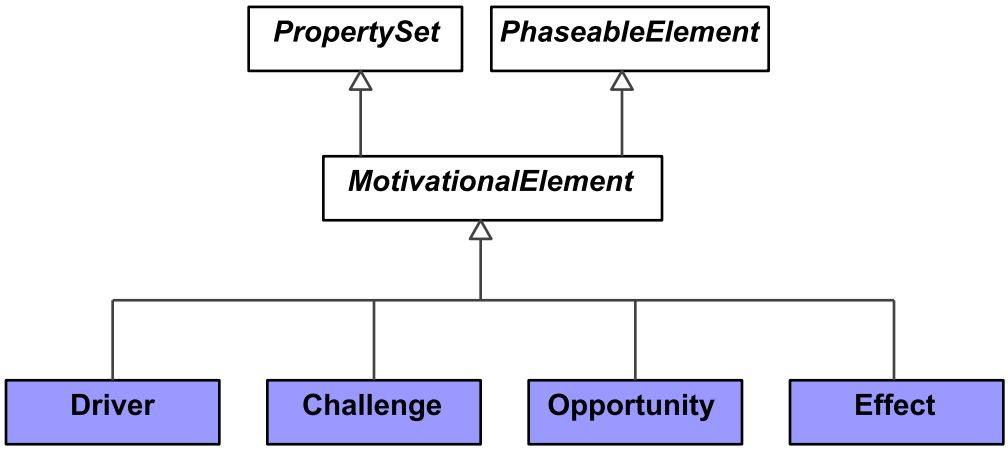

Figure 9:50 - MotivationalElement

Opportunity

Package: Motivation
isAbstract: No
Generalization: MotivationalElement, AffectableElement

Description

An existing or potential favorable circumstance or combination of circumstances which can be advantageous for addressing enterprise Challenges.

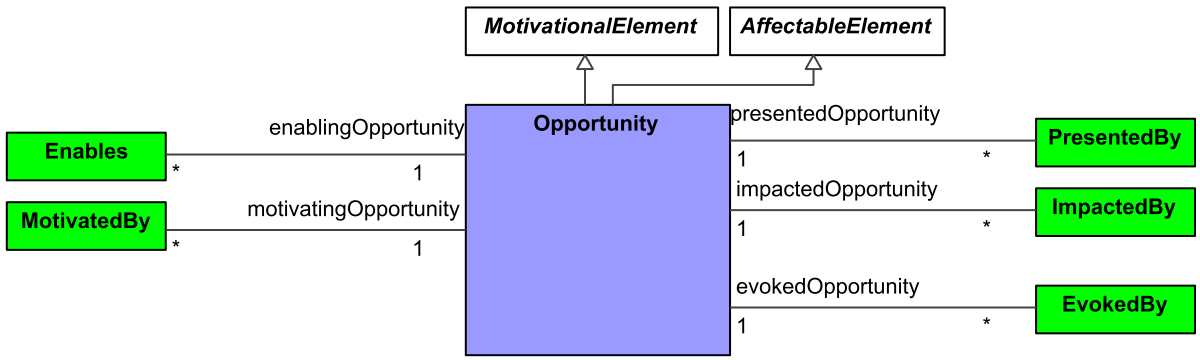

Figure 9:51 - Opportunity

PresentedBy

Package: Motivation
isAbstract: No
Generalization: MeasurableElement

Description

A tuple denoting that a Challenge must be overcome for addressing a Driver.

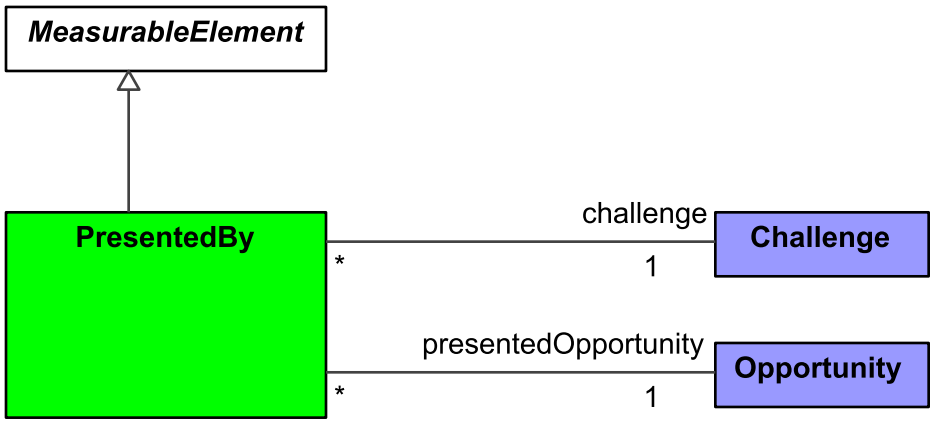

Figure 9:52 - PresentedBy

9.1.3.2 Domain MetaModel::Strategic::Taxonomy

Capability

Package: Taxonomy
isAbstract: No
Generalization: PropertySet, Desirer, PhaseableElement, AffectableElement, SubjectOfStrategicConstraint

Description

An enterprise's ability to Achieve a desired effect realized through a combination of ways and means (e.g. CapabilityConfigurations) along with specified measures.

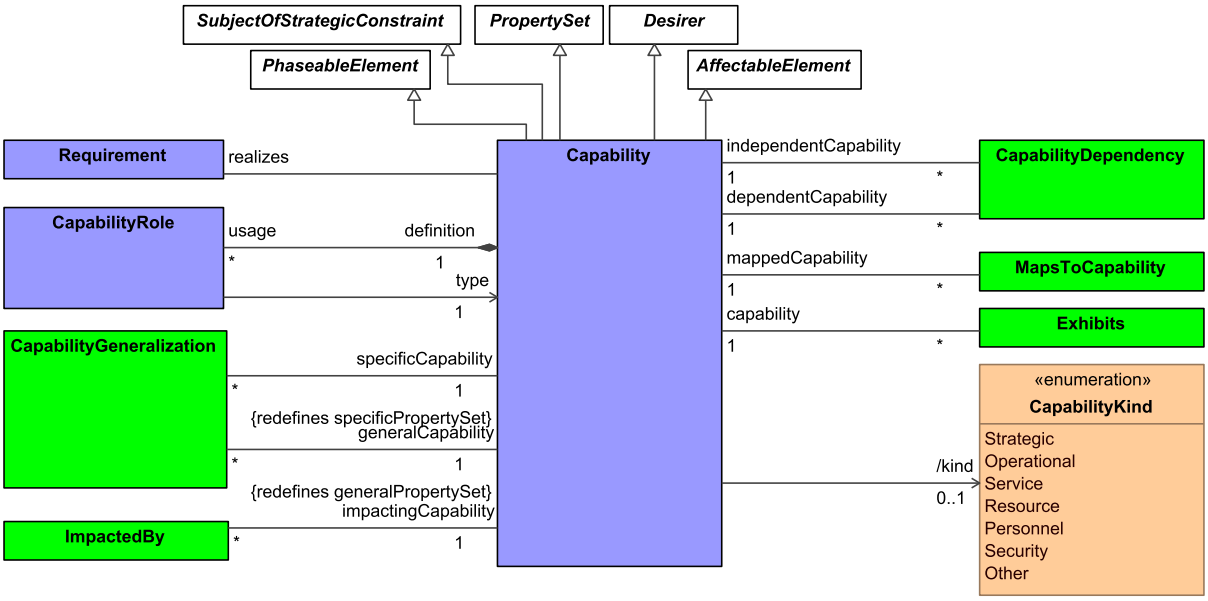

Figure 9:53 - Capability

CapabilityGeneralization

Package: Taxonomy
isAbstract: No
Generalization: PropertySetGeneralization

Description

A CapabilityGeneralization is a taxonomic relationship between a more general Capability and a more specific Capability.

Figure 9:54 - CapabilityGeneralization

EnterpriseGoal

Package: Taxonomy
isAbstract: No
Generalization: PropertySet, PhaseableElement, AffectableElement, Requirement

Description

A statement about a state or condition of the enterprise to be brought about or sustained through appropriate Means. An EnterpriseGoal amplifies an EnterpriseVision that is, it indicates what must be satisfied on a continuing basis to effectively attain the EnterpriseVision. BMM: OMG dtc-13-08-24.

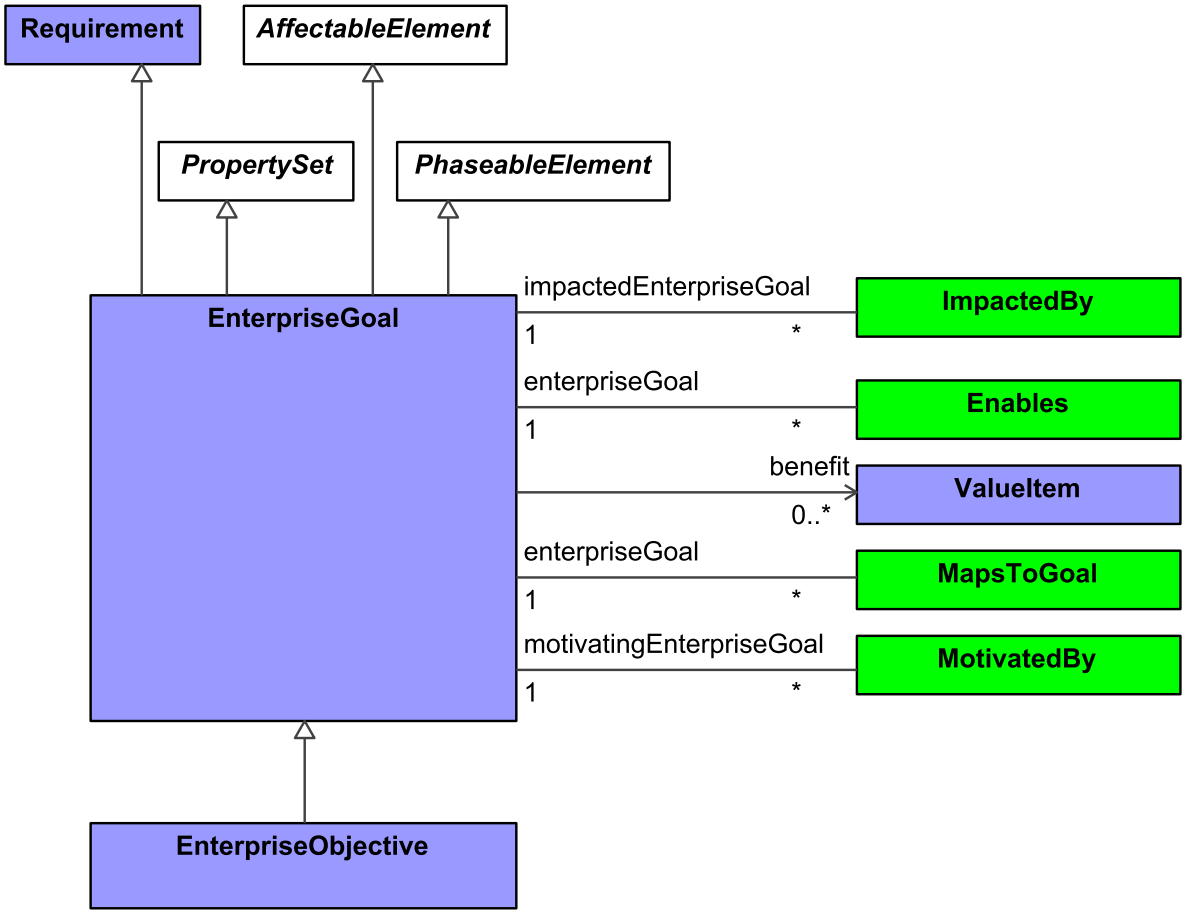

Figure 9:55 - EnterpriseGoal

EnterpriseObjective

Package: Taxonomy
isAbstract: No
Generalization: EnterpriseGoal

Description

A statement of an attainable, time-targeted, and measurable target that the enterprise seeks to meet in order to achieve its Goals. http://www.omg.org/spec/BMM/1.3/

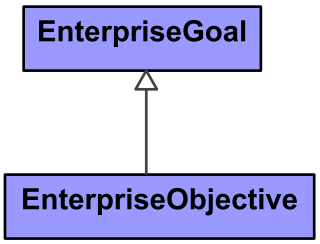

Figure 9:56 - EnterpriseObjective

EnterpriseVision

Package: Taxonomy
isAbstract: No
Generalization: PropertySet, PhaseableElement

Description

A Vision describes the future state of the enterprise, without regard to how it is to be achieved. BMM: OMG dtc-13-08-24.

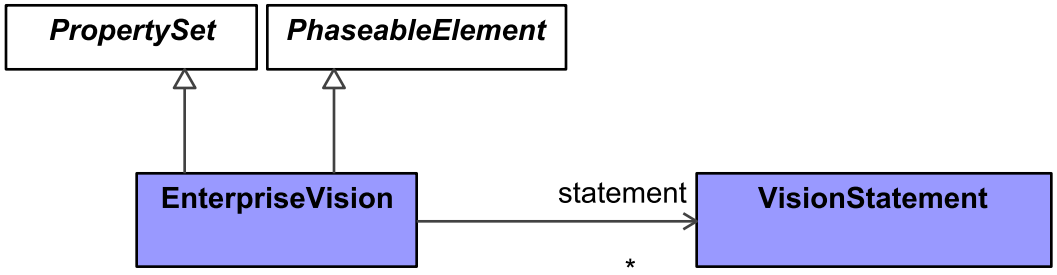

Figure 9:57 - EnterpriseVision

Mission

Package: Taxonomy
isAbstract: No
Generalization: StrategicPhase

Description

The task, together with the purpose, that clearly indicates the action to be taken and the reasoning behind the mission. (JP 1-02)

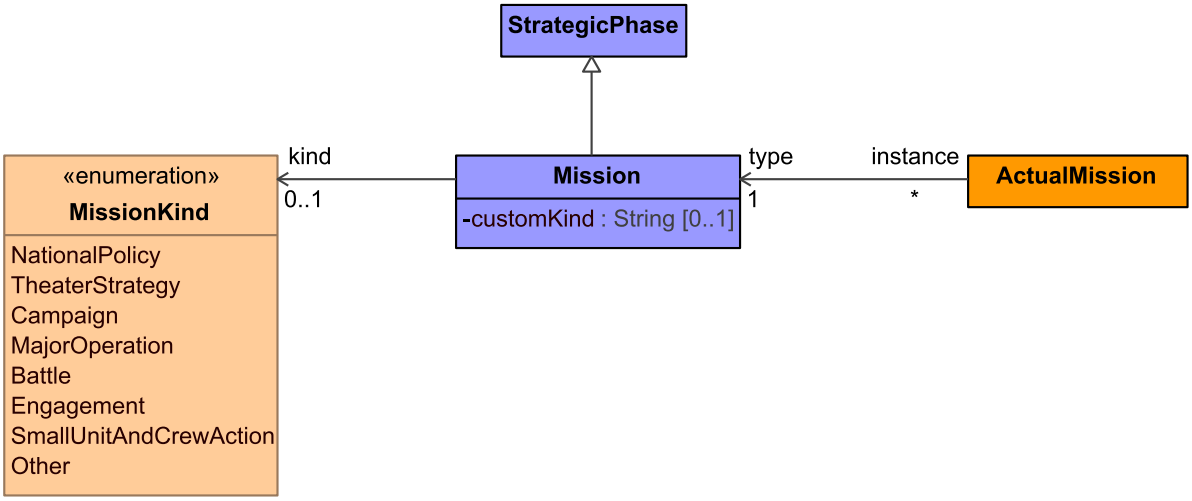

Figure 9:58 - Mission

Attributes

customKind : String [0..1] Identification of a special kind of Mission that is different from one of the predefined enumerated kinds.

OwnsValue

Package: Taxonomy
isAbstract: No
Generalization: MeasurableElement

Description

An tuple denoting that an ActualOrganizationalResource owns a ValueItem.

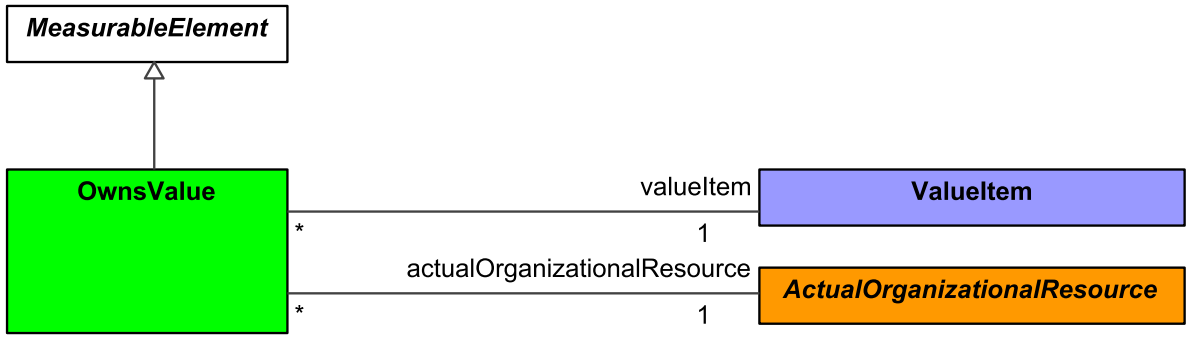

Figure 9:59 - OwnsValue

PhaseableElement

Package: Taxonomy
isAbstract: Yes
Generalization: UAFElement

Description

An abstract element that indicates the types of elements that can be assigned to a specific ActualStrategicPhase.

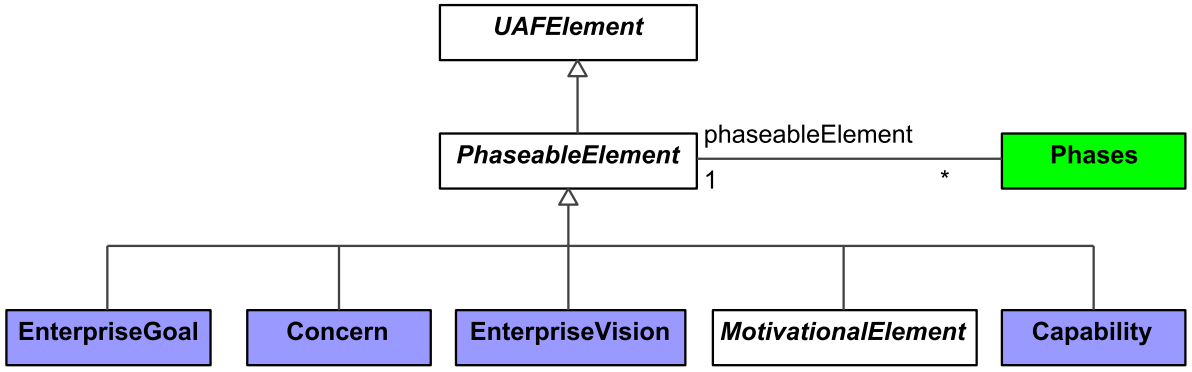

Figure 9:60 - PhaseableElement

StrategicAsset

Package: Taxonomy
isAbstract: Yes
Generalization: Asset

Description

An abstract element that indicates the types of strategic elements that can be affected by Risk.

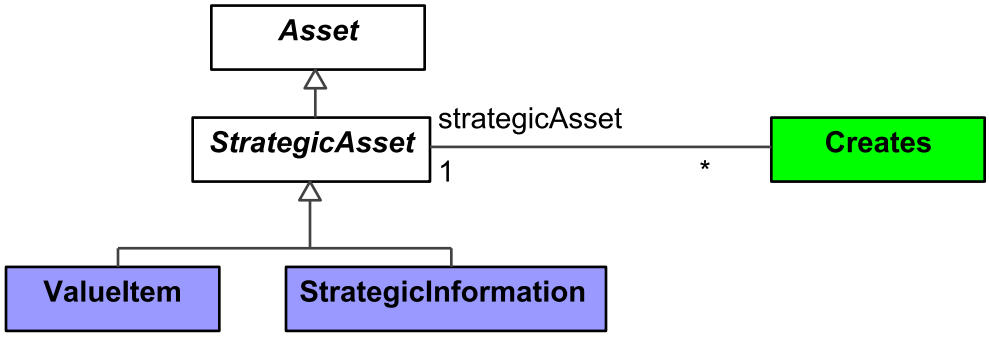

Figure 9:61 - StrategicAsset

StrategicPhase

Package: Taxonomy
isAbstract: No
Generalization: PropertySet

Description

A type of a current or future state of the enterprise, mission, ValueStream, or EnduringTask.

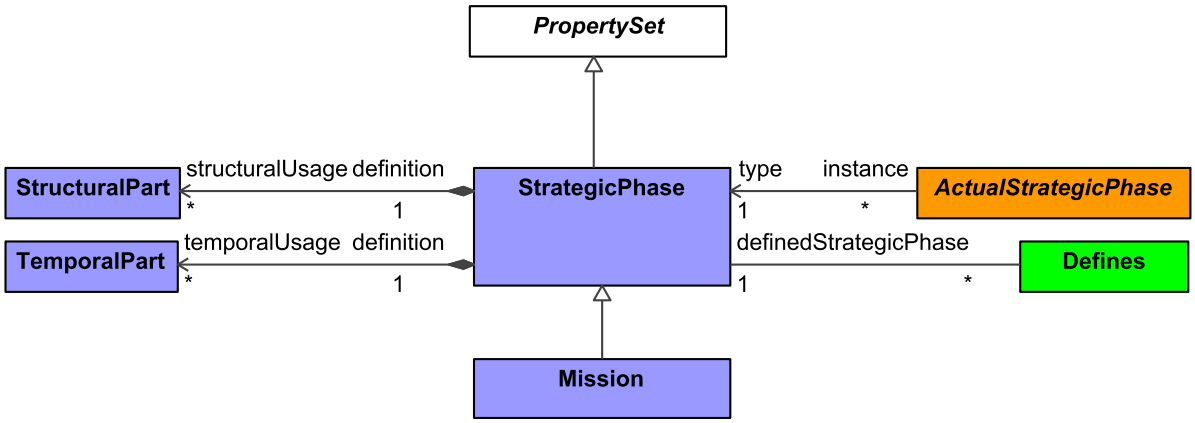

Figure 9:62 - StrategicPhase

ValueItem

Package: Taxonomy
isAbstract: No
Generalization: MeasurementSet, StrategicAsset

Description

An ideal, custom, or institution that an enterprise promotes or agrees with. It may be positive or negative, depending on point of view.

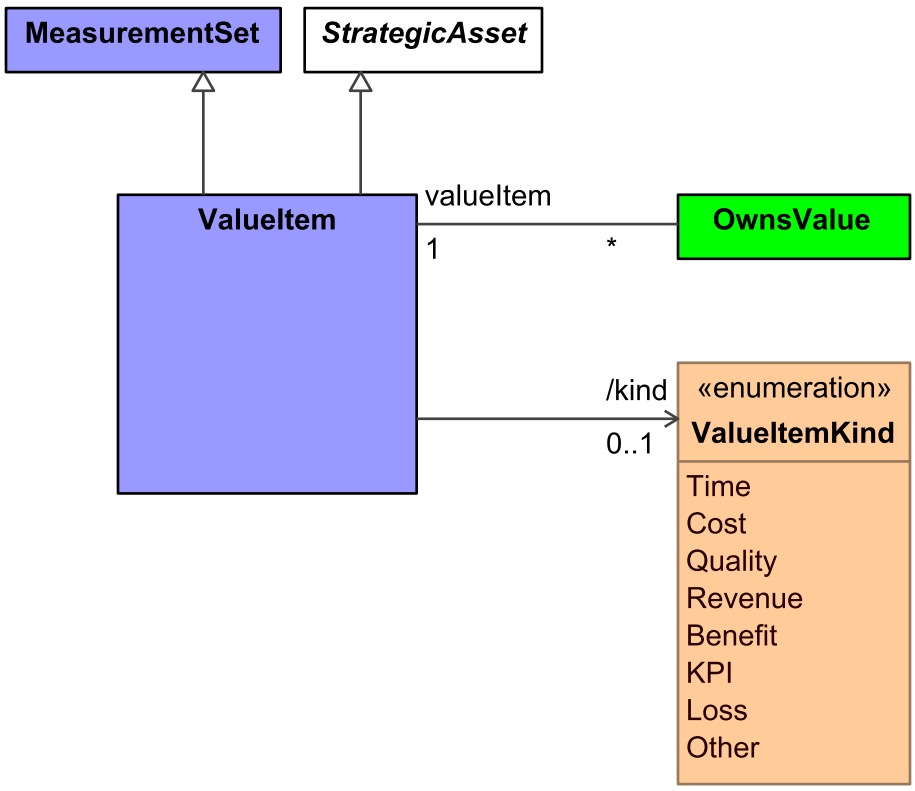

Figure 9:63 - ValueItem

VisionStatement

Package: Taxonomy
isAbstract: No
Generalization: MeasurableElement

Description

A type of comment that describes the future state of the enterprise, without regard to how it is to be achieved. BMM: OMG dtc-13-08-24.

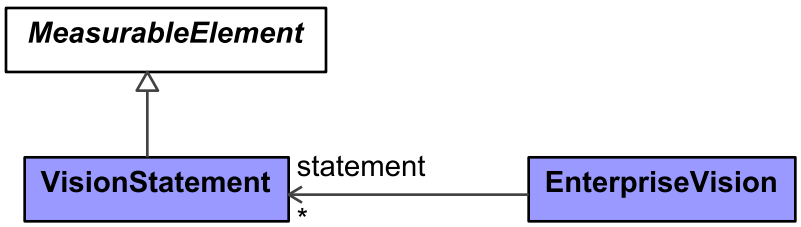

Figure 9:64 - VisionStatement

WholeLifeEnterprise

Package: Taxonomy
isAbstract: No
Generalization: ActualEnterprisePhase

Description

A WholeLifeEnterprise is a purposeful endeavor of any size involving people, organizations and supporting systems. It is made up of TemporalParts and StructuralParts.

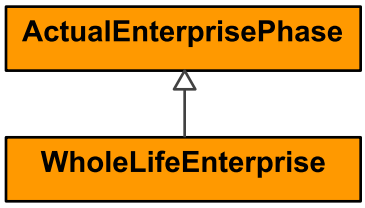

Figure 9:65 - WholeLifeEnterprise

9.1.3.3 Domain MetaModel::Strategic::Structure

CapabilityRole

Package: Structure
isAbstract: No
Generalization: PropertySet, Desirer, MeasurableElement

Description

A high level specification of the enterprise's ability to execute a specified course of action.

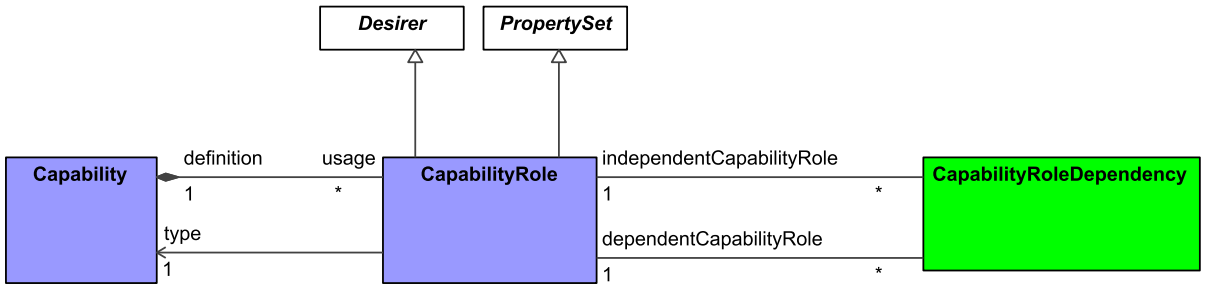

Figure 9:66 - CapabilityRole

StructuralPart

Package: Structure
isAbstract: No
Generalization: MeasurableElement

Description

Usage of a StrategicPhase in the context of another StrategicPhase. It asserts that one StrategicPhase is a spatial part of another. Creates a whole-part relationship that represents the structure of the StrategicPhase.

Figure 9:67 - StructuralPart

TemporalPart

Package: Structure
isAbstract: No
Generalization: MeasurableElement

Description

Usage of an StrategicPhase in the context of another StrategicPhase. It asserts that one StrategicPhase is a spatial part of another. Creates a whole-part relationship that represents the temporal structure of the StrategicPhase.

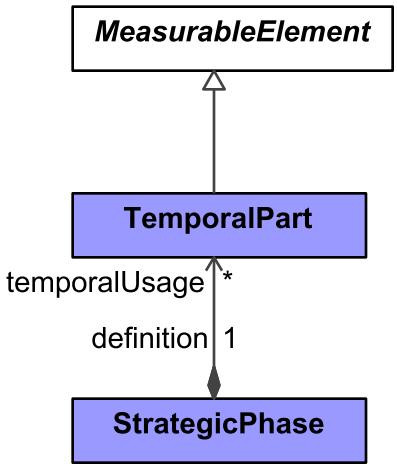

Figure 9:68 - TemporalPart

9.1.3.4 Domain MetaModel::Strategic::Connectivity

CapabilityDependency

Package: Connectivity
isAbstract: No
Generalization: MeasurableElement

Description

A tuple that asserts that one CapabilityDependency is dependent from another.

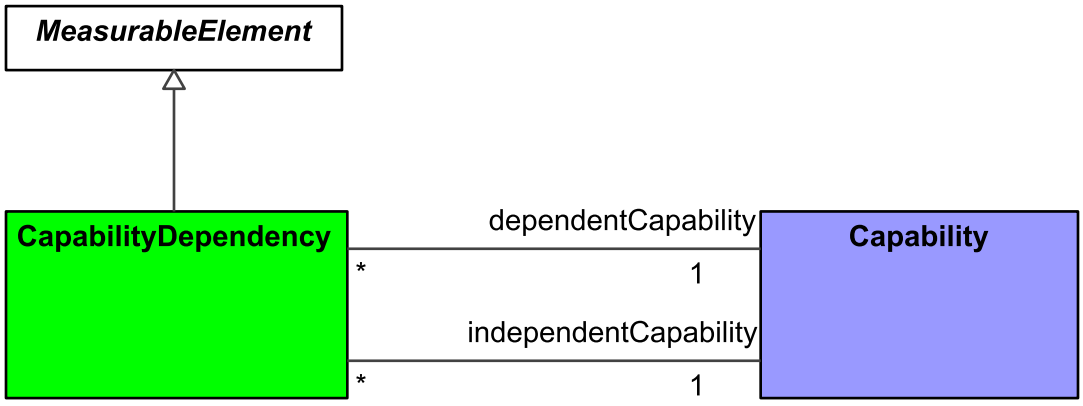

Figure 9:69 - CapabilityDependency

CapabilityRoleDependency

Package: Connectivity
isAbstract: No
Generalization: MeasurableElement

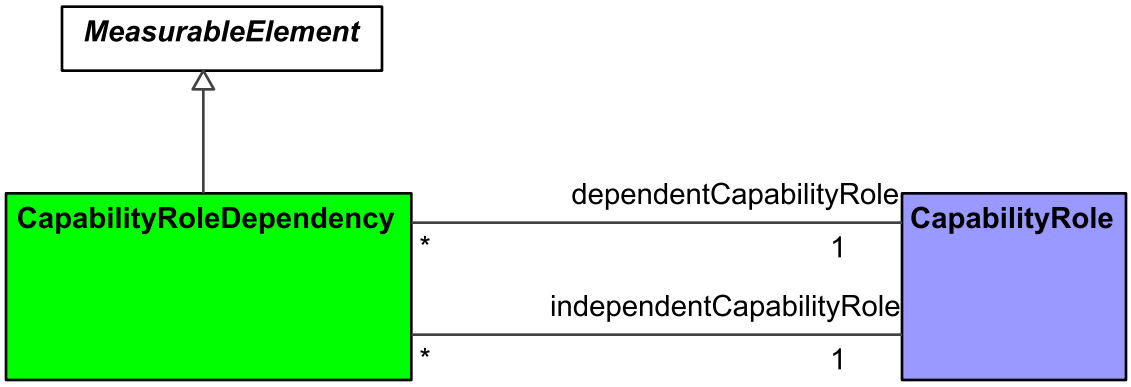

Figure 9:70 - CapabilityRoleDependency

StrategicExchange

Package: Connectivity
isAbstract: No
Generalization: Exchange

Description

Asserts that a flow can exist between ActualStrategicPhases (i.e. flows of information, people, materiel, or energy).

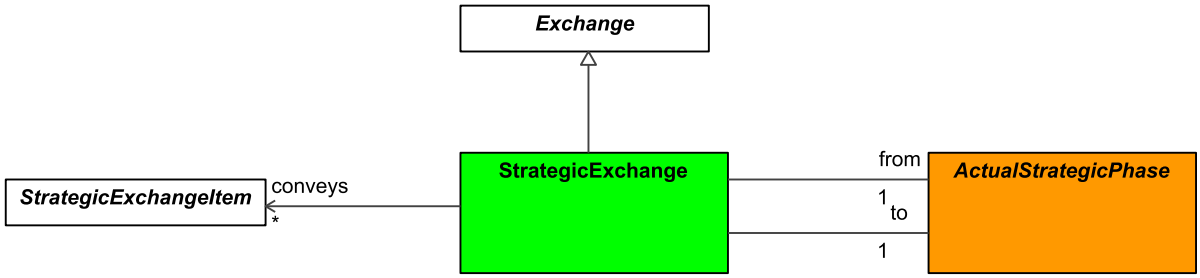

Figure 9:71 - StrategicExchange

StrategicExchangeItem

Package: Connectivity
isAbstract: Yes
Generalization: Resource, ExchangeItem

Description

An abstract grouping for elements that defines the types of elements that can be exchanged between ActualStrategicPhases and conveyed by a StrategicExchange.

9.1.3.5 Domain MetaModel::Strategic::Processes

ActualEnduringTask

Package: Processes
isAbstract: No
Generalization: ActualStrategicPhase

Description

An actual undertaking recognized by an enterprise as being essential to achieving its goals - i.e. a strategic specification of what the enterprise does.

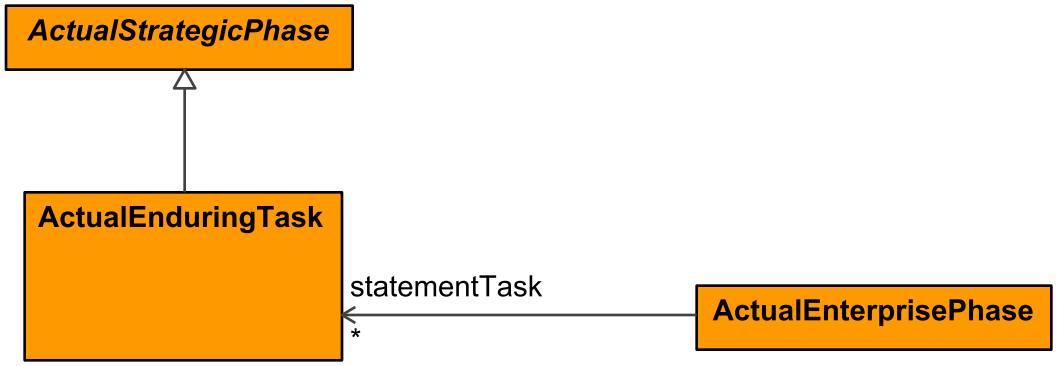

Figure 9:72 - ActualEnduringTask

ActualEnterprisePhase

Package: Processes
isAbstract: No
Generalization: ActualStrategicPhase

Description

A time period within which a set of Capabilities are deployed.

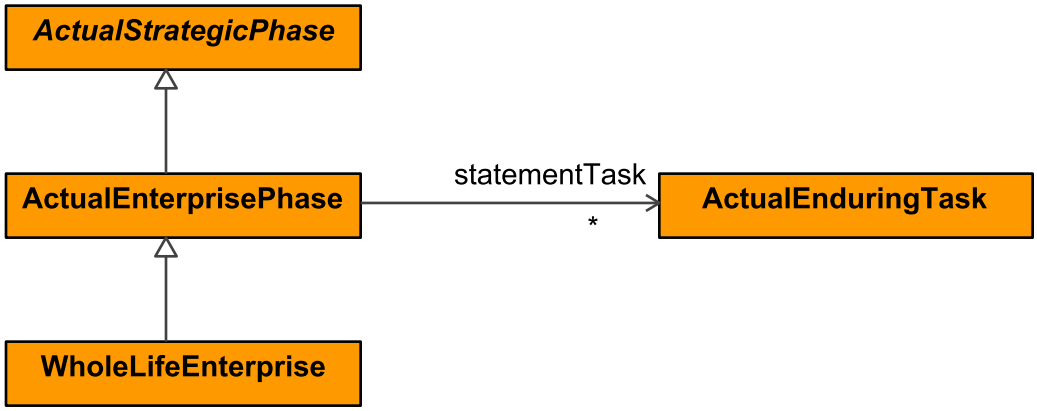

Figure 9:73 - ActualEnterprisePhase

ActualMission

Package: Processes
isAbstract: No
Generalization: ActualStrategicPhase

Description

A particular mission to accomplish assigned mission objectives

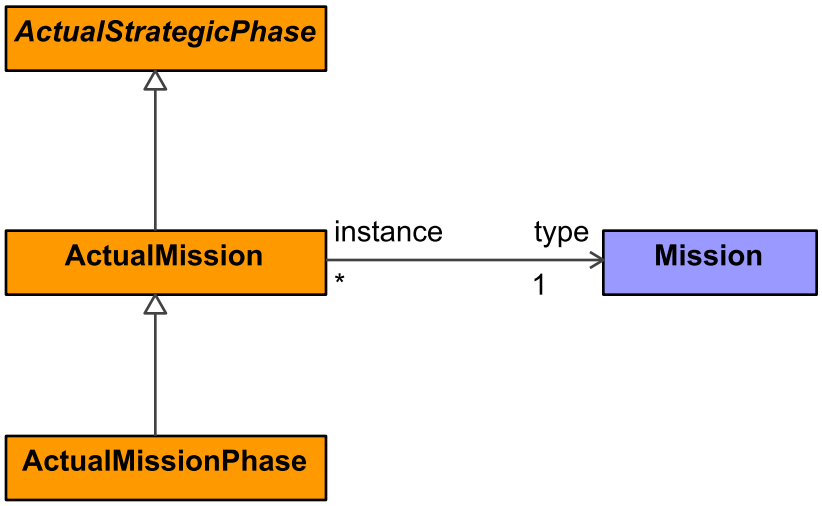

Figure 9:74 - ActualMission

ActualMissionPhase

Package: Processes
isAbstract: No
Generalization: ActualMission

Description

A particular phase in the accomplishment of an overall actual mission.

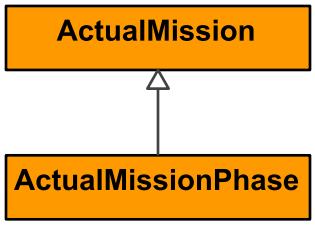

Figure 9:75 - ActualMissionPhase

ActualStrategicPhase

Package: Processes
isAbstract: Yes
Generalization: Achiever, ActualPropertySet

Description

A phase of an actual enterprise, mission, ValueStream or EnduringTask endeavor.

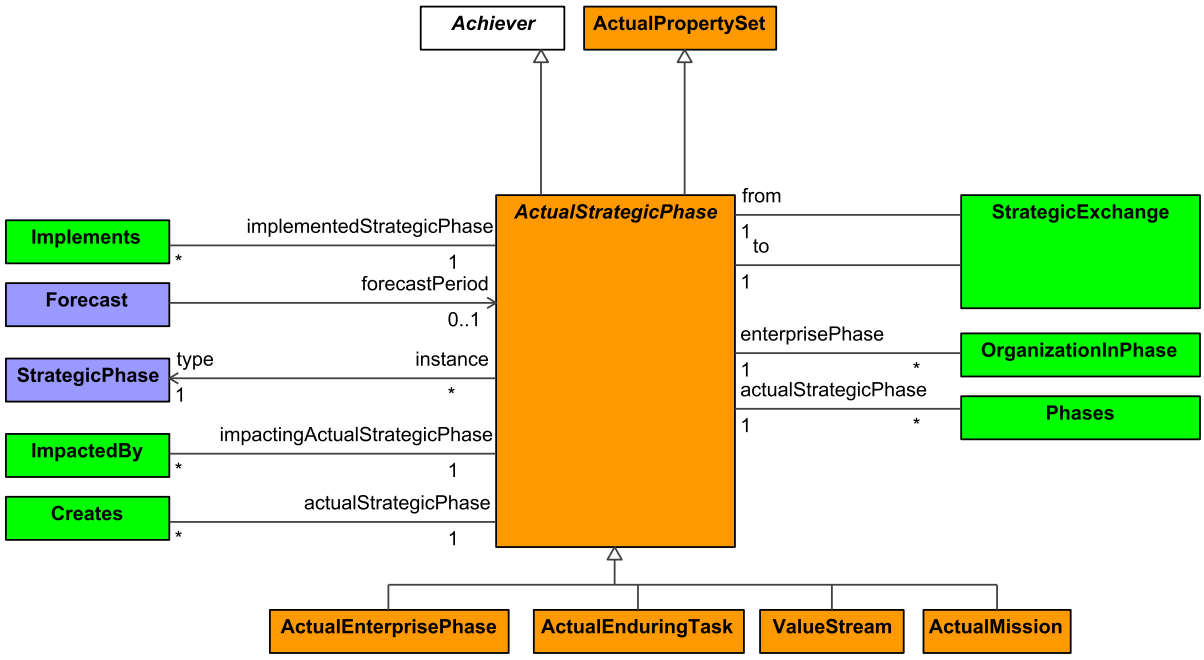

Figure 9:76 - ActualStrategicPhase

Creates

Package: Processes
isAbstract: No
Generalization: MeasurableElement

Description

A tuple used to denote that an ActualStrategicPhase brings into existence a StrategicAsset.

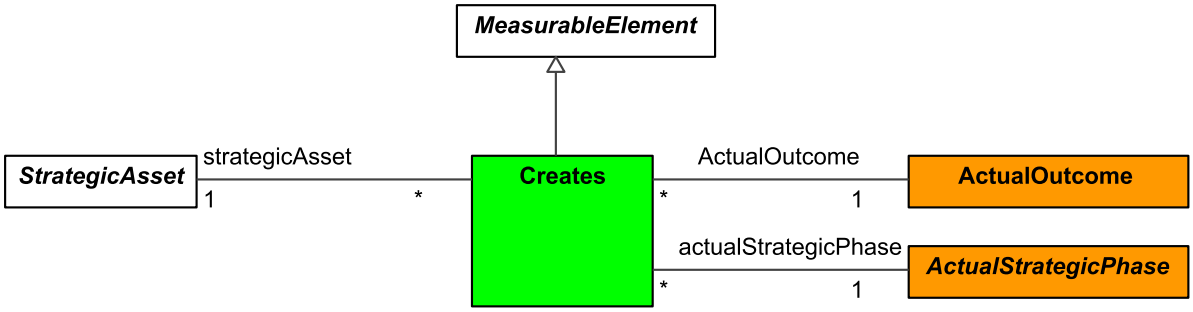

Figure 9:77 - Creates

Defines

Package: Processes
isAbstract: No
Generalization: MeasurableElement

Description

A tuple that links the StrategicPhase and the Activity that implements it.

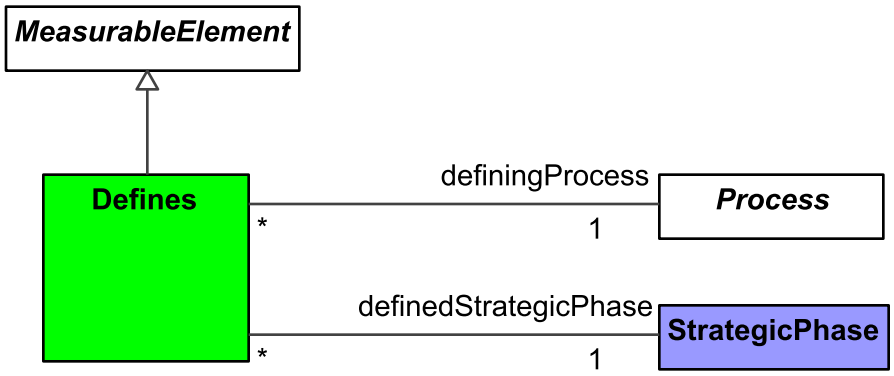

Figure 9:78 - Defines

ValueStream

Package: Processes
isAbstract: No
Generalization: ActualStrategicPhase

Description

An end-to-end collection of activities that create a result for a customer, who may be the ultimate customer or an internal end-user of the value stream. Value stream nested within another value stream may represent Value Stream Stage - a distinct, identifiable phase or step within a value stream [The Business Architecture Metamodel Guide, 2020]

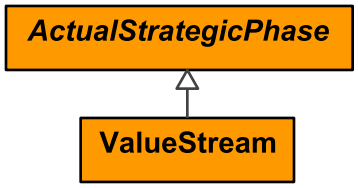

Figure 9:79 - ValueStream

9.1.3.6 Domain MetaModel::Strategic::States

Achiever

Package: States
isAbstract: Yes
Generalization: UAFElement

Description

An ActualResource, ActualProject or ActualStrategicPhase that can deliver a desired effect.

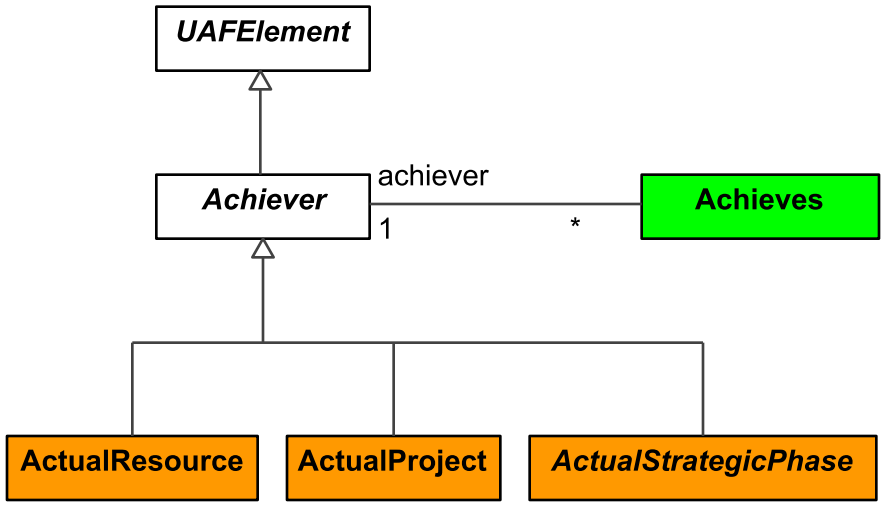

Figure 9:80 - Achiever

Achieves

Package: States
isAbstract: No
Generalization: MeasurableElement

Description

A tuple that exists between an ActualState (e.g., observed/measured during testing) of an element that attempts to achieve a desired effect and an Achiever.

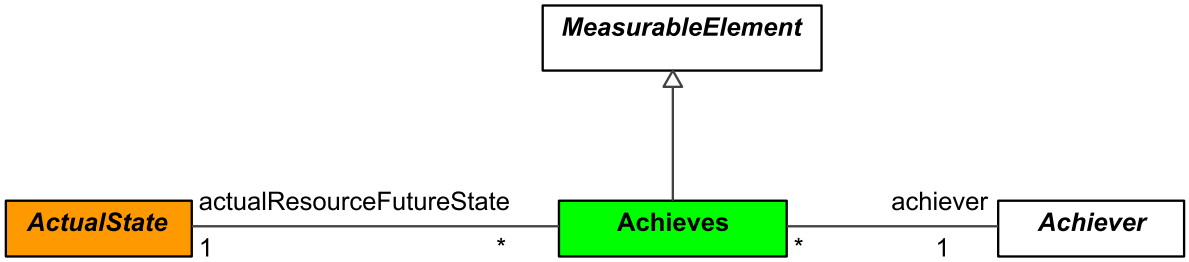

Figure 9:81 - Achieves

ActualEffect

Package: States
isAbstract: No
Generalization: ActualPropertySet

Description

A real world phenomenon that follows and is caused by some previous phenomenon.

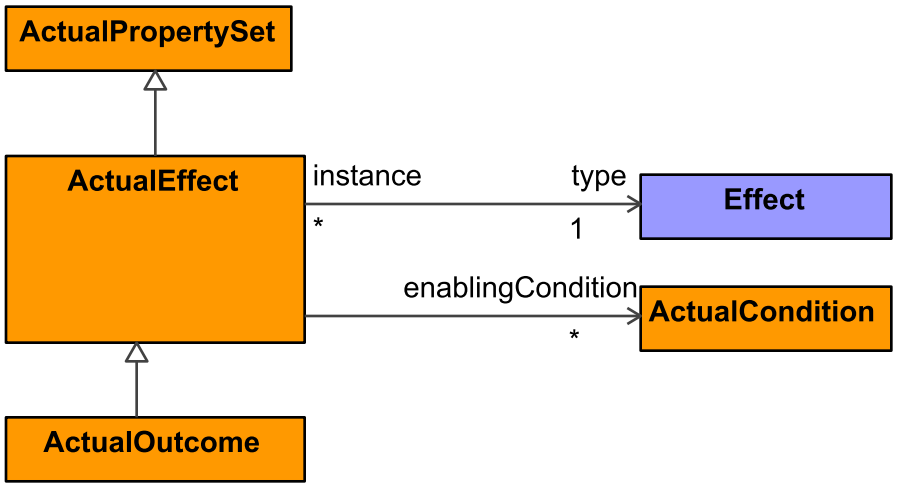

Figure 9:82 - ActualEffect

ActualOutcome

Package: States
isAbstract: No
Generalization: ActualEffect

Description

An individual that describes something that happens or is produced as the final consequence or product and is related to one of the goals for the business or enterprise. Outcome is a special kind of effect, one that is usually at the end of a chain of effects, i.e. an "end effect".

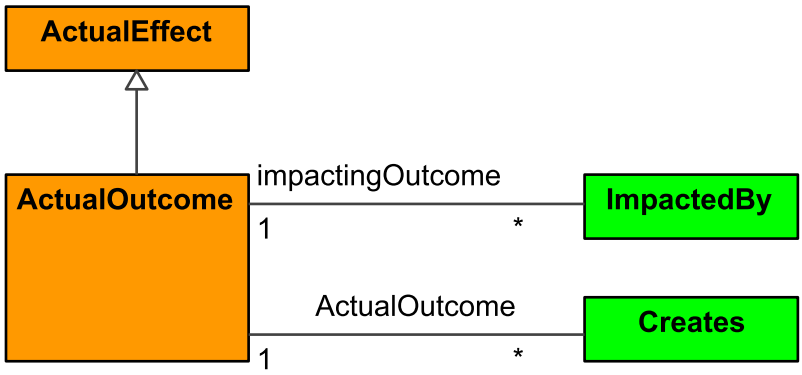

Figure 9:83 - ActualOutcome

Desirer

Package: States
isAbstract: Yes
Generalization: UAFElement

Description

Abstract type used to group architecture elements that might desire a particular effect.

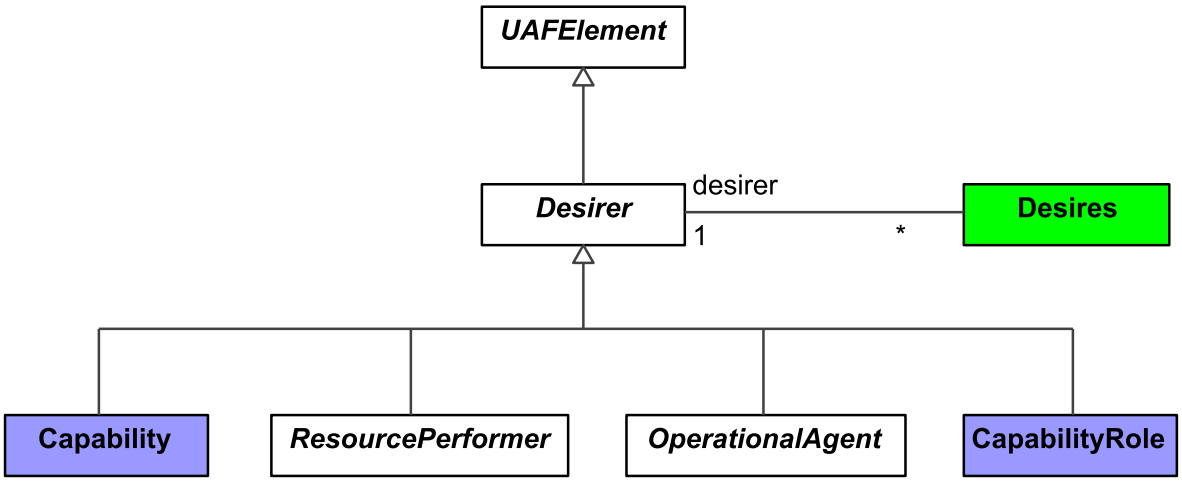

Figure 9:84 - Desirer

Desires

Package: States
isAbstract: No
Generalization: MeasurableElement

Description

A tuple relating the Desirer (a Capability or OrganizationalResource) to an ActualState.

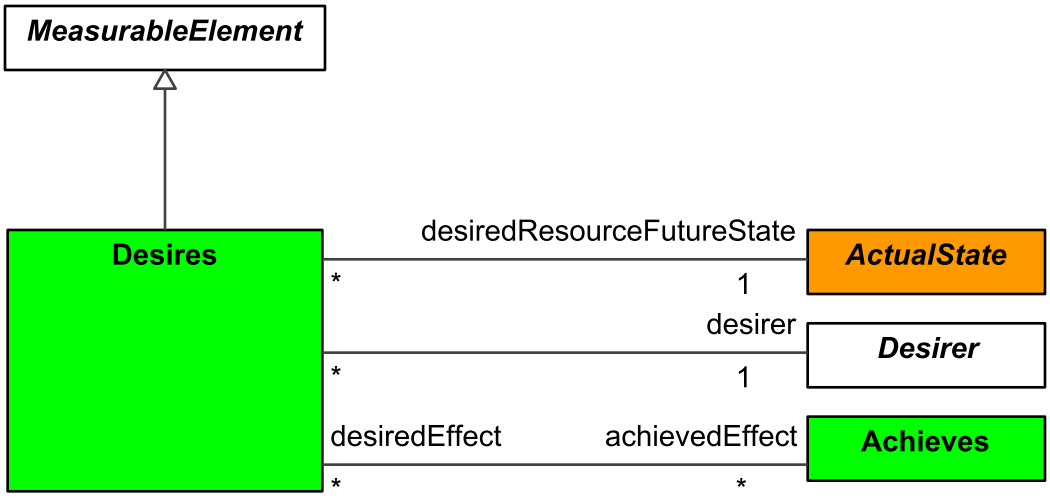

Figure 9:85 - Desires

Effect

Package: States
isAbstract: No
Generalization: MotivationalElement

Description

A kind of phenomenon that follows and is caused by some previous phenomenon that could lead to downstream effects or to one or more desired outcomes.

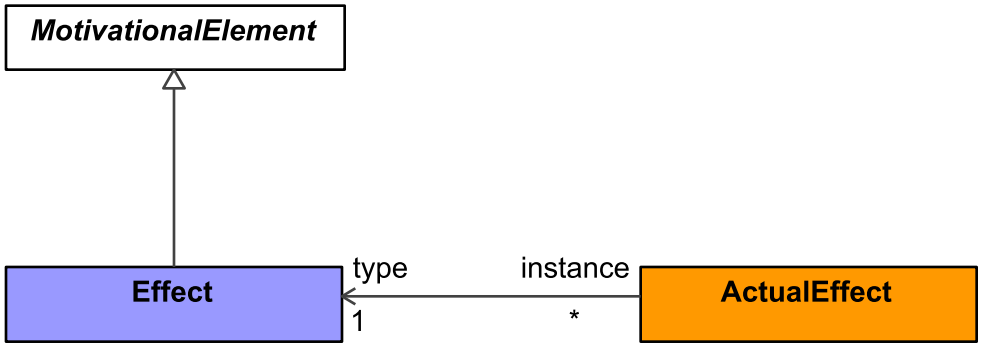

Figure 9:86 - Effect

9.1.3.7 Domain MetaModel::Strategic::Information

MapsToGoal

Package: Information
isAbstract: No
Generalization: MeasurableElement

Description

A tuple denoting that some StrategicInformation contributes to achieving an EnterpriseGoal or Objective.

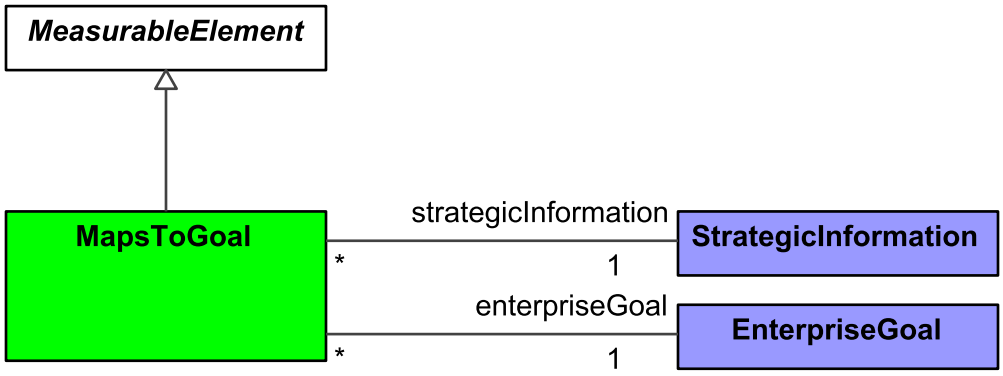

Figure 9:87 - MapsToGoal

StrategicInformation

Package: Information
isAbstract: No
Generalization: StrategicExchangeItem, StrategicAsset

Description

Knowledge communicated or received concerning a particular fact or circumstance that is strategic in nature that is important or essential in relation to a plan of action

9.1.3.8 Domain MetaModel::Strategic::Constraints

StrategicConstraint

Package: Constraints
isAbstract: No
Generalization: Rule

Description

A Rule governing a Capability.

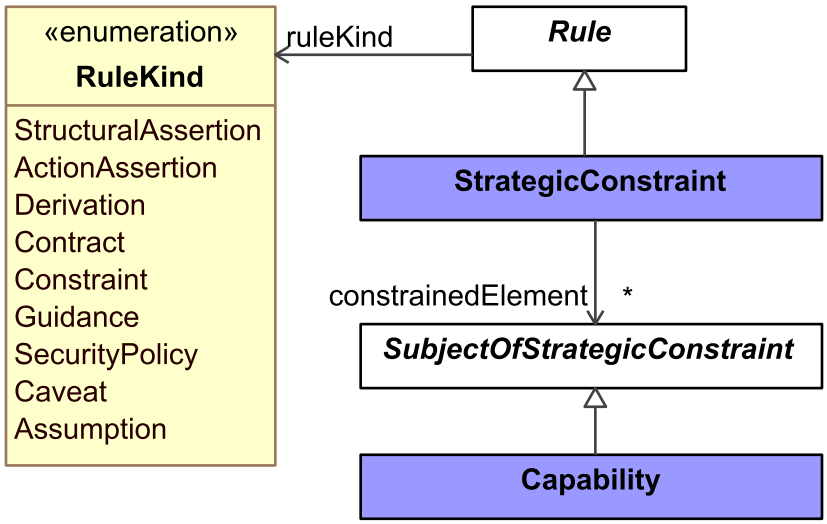

Figure 9:88 - StrategicConstraint

SubjectOfStrategicConstraint

Package: Constraints
isAbstract: Yes
Generalization: UAFElement

Description

An abstract grouping of elements that can be the subject of a StrategicConstraint.

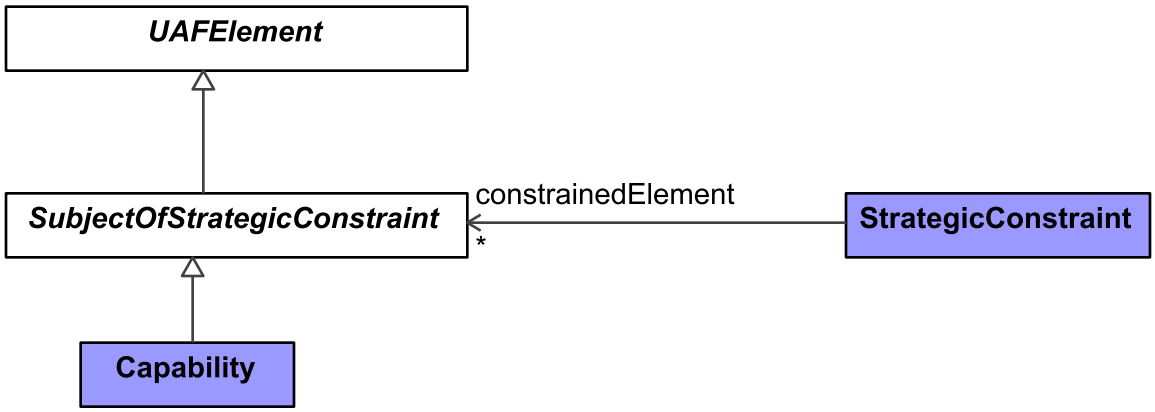

Figure 9:89 - SubjectOfStrategicConstraint

9.1.3.9 Domain MetaModel::Strategic::Traceability

EvokedBy

Package: Traceability
isAbstract: No
Generalization: MeasurableElement

Description

A tuple used to denote that a Risk is drawn out by an Opportunity.

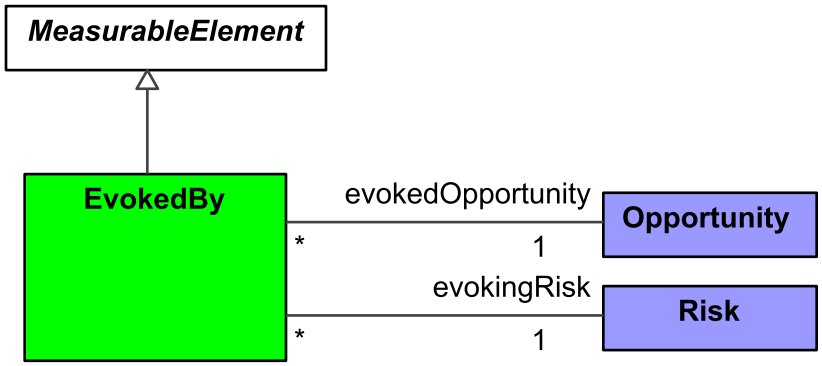

Figure 9:90 - EvokedBy

Exhibits

Package: Traceability
isAbstract: No
Generalization: MeasurableElement

Description

A tuple that exists between a CapableElement and a Capability that it meets under specific environmental conditions.

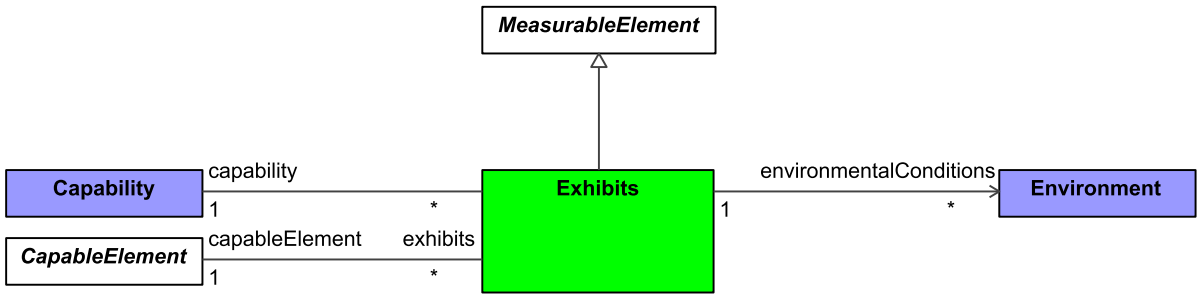

Figure 9:91 - Exhibits

MapsToCapability

Package: Traceability
isAbstract: No
Generalization: MeasurableElement

Description

A tuple denoting that an Activity contributes to providing a Capability.

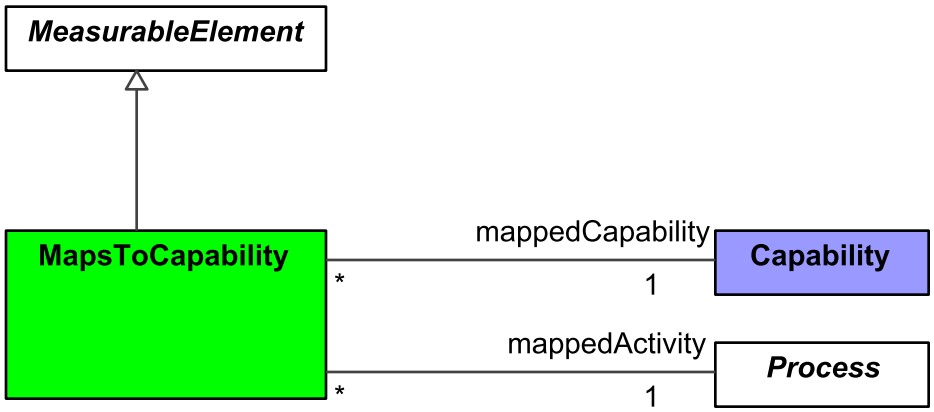

Figure 9:92 - MapsToCapability

OrganizationInPhase

Package: Traceability
isAbstract: No
Generalization: MeasurableElement

Description

An abstraction relationship relating an ActualOrganization to an ActualStrategicPhase to denote that the ActualOrganization plays a role or is a stakeholder in an ActualStrategicPhase.

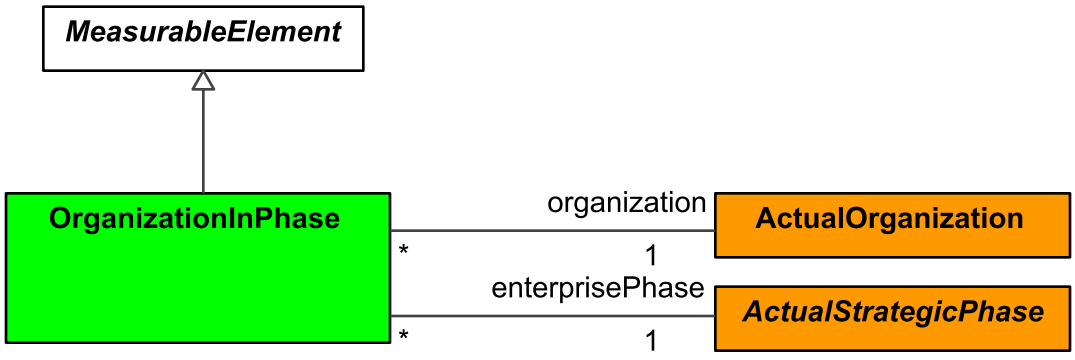

Figure 9:93 - OrganizationInPhase

9.1.4 Domain MetaModel::Operational

9.1.4.1 Domain MetaModel::Operational::Taxonomy

ArbitraryConnector

Package: Taxonomy
isAbstract: No
Generalization: MeasurableElement

Description

Represents a visual indication of a connection used in high level operational concept diagrams.

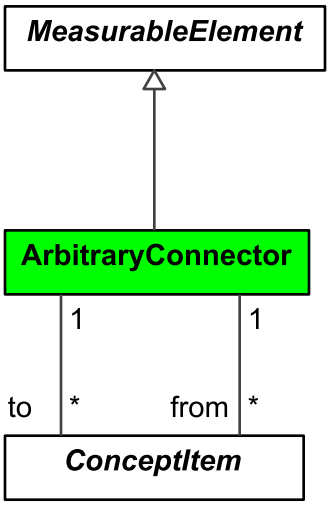

Figure 9:94 - ArbitraryConnector

ConceptItem

Package: Taxonomy
isAbstract: Yes
Generalization: UAFElement

Description

An abstract type which represents some part played by an asset or location in a HighLevelOperationalConcept.

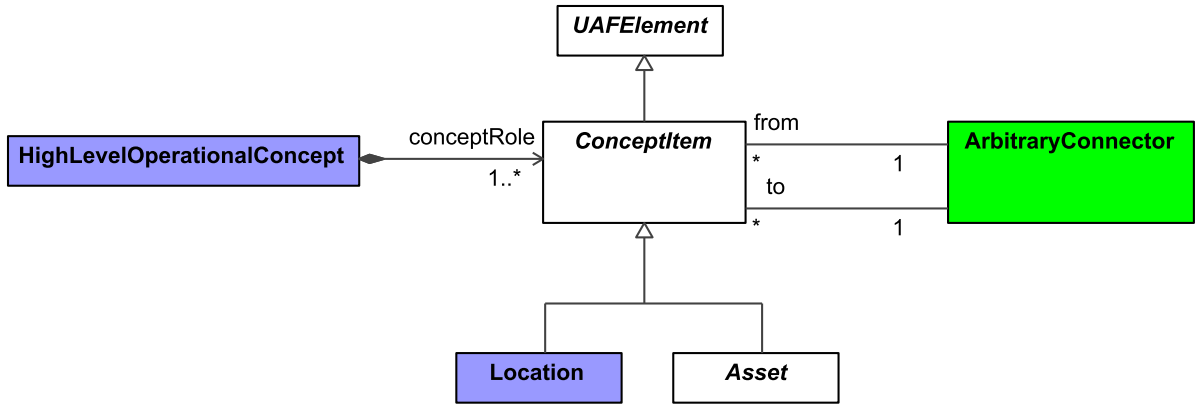

Figure 9:95 - ConceptItem

HighLevelOperationalConcept

Package: Taxonomy
isAbstract: No
Generalization: PropertySet

Description

Describes the Resources and Locations required to meet an operational scenario from an integrated systems point of view. It is used to communicate overall quantitative and qualitative system characteristics to stakeholders.

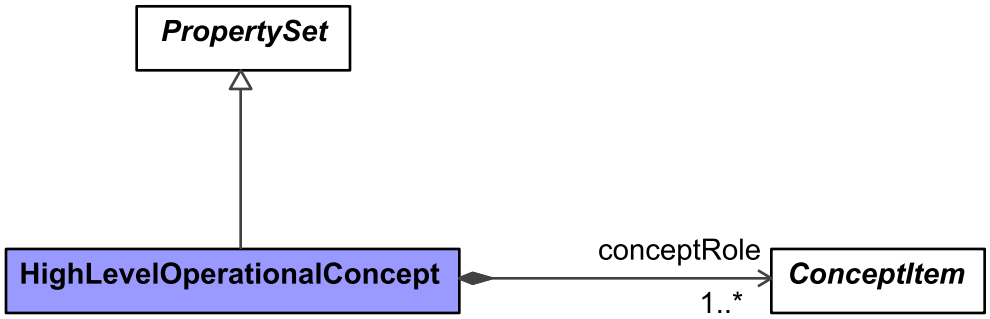

Figure 9:96 - HighLevelOperationalConcept

9.1.4.2 Domain MetaModel::Operational::Structure

KnownResource

Package: Structure
isAbstract: No
Generalization: OperationalPerformer, ResourcePerformer

Description

Asserts that a known ResourcePerformer constrains the implementation of the OperationalPerformer that plays the role in the OperationalArchitecture.

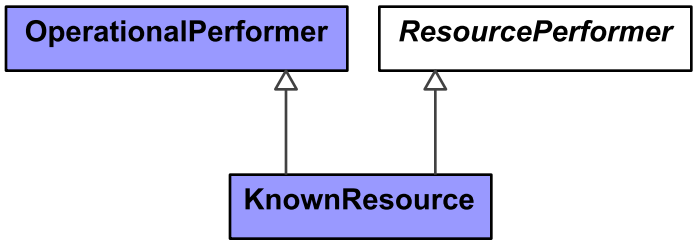

Figure 9:97 - KnownResource

OperationalAgent

Package: Structure
isAbstract: Yes
Generalization: SubjectOfOperationalConstraint, CapableElement, OperationalAsset, Desirer

Description

An abstract type grouping OperationalArchitecture and OperationalPerformer.

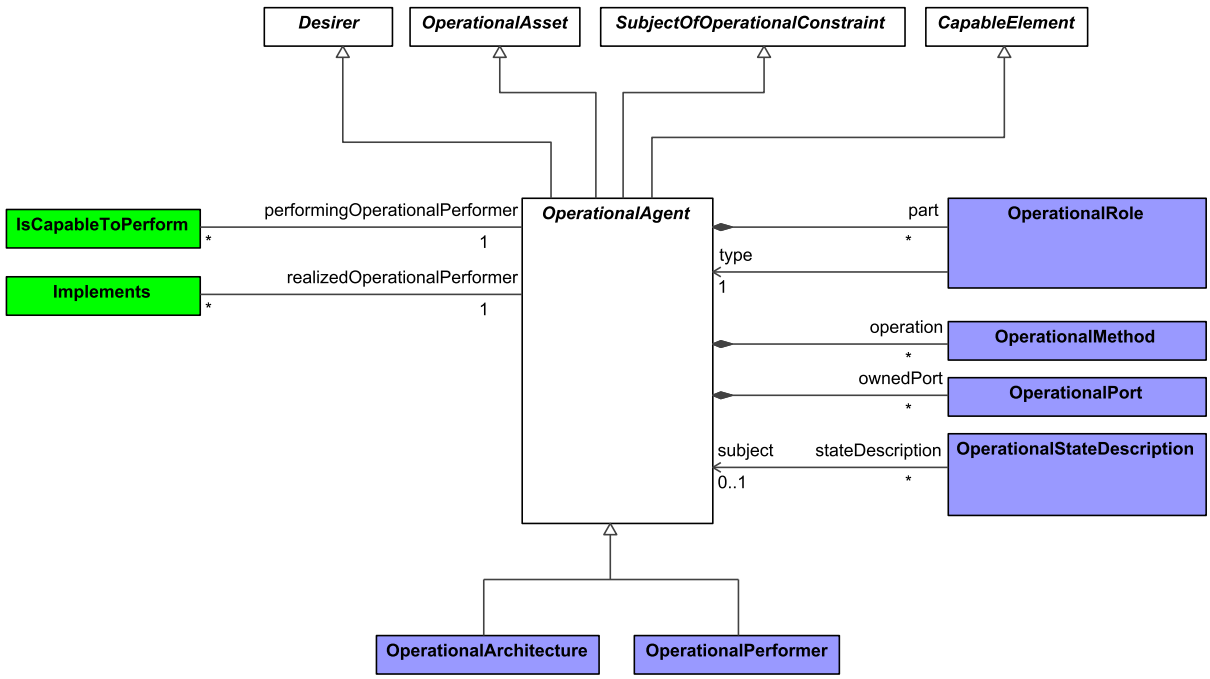

Figure 9:98 - OperationalAgent

OperationalArchitecture

Package: Structure
isAbstract: No
Generalization: OperationalAgent, Architecture

Description

A type used to denote a model of the Architecture, described from the Operational perspective.

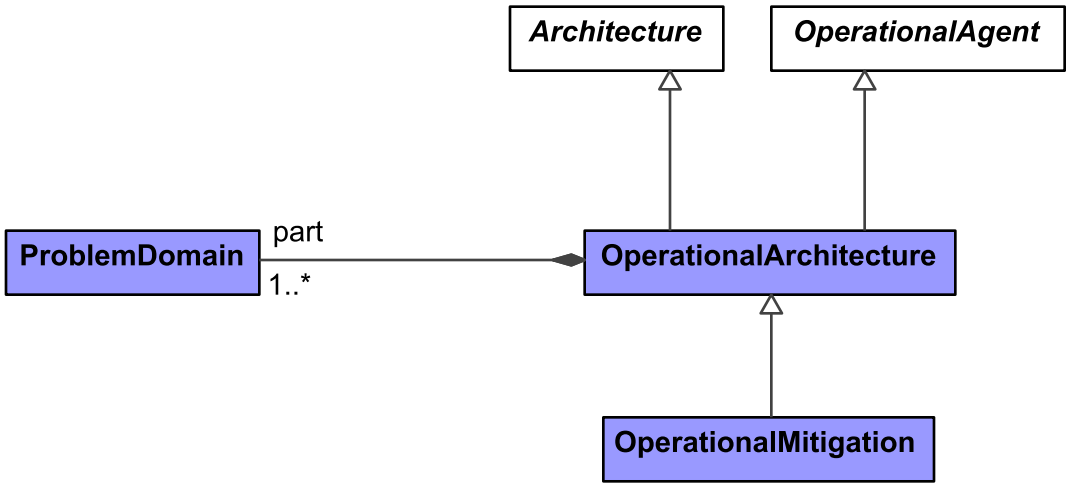

Figure 9:99 - OperationalArchitecture

OperationalMethod

Package: Structure
isAbstract: No
Generalization: ProcessOperation

Description

A behavioral feature of an OperationalAgent whose behavior is specified in an OperationalActivity.

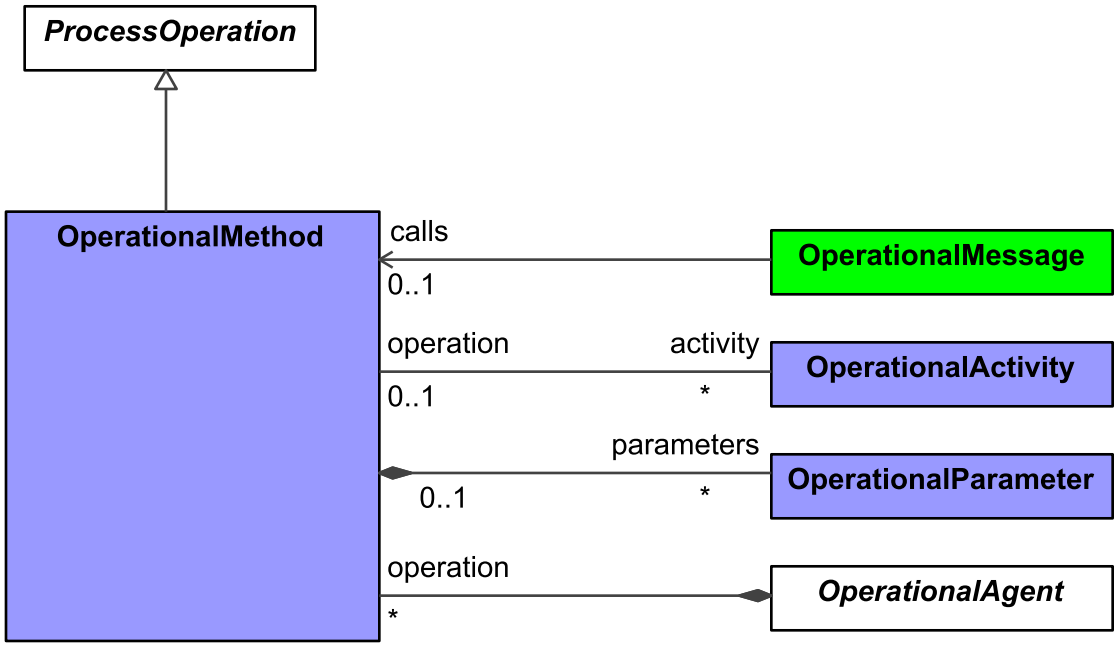

Figure 9:100 - OperationalMethod

OperationalParameter

Package: Structure

isAbstract: No
Generalization: ProcessParameter

Description

A type that represents inputs and outputs of an OperationalActivity. It is typed by an OperationalExchangeItem.

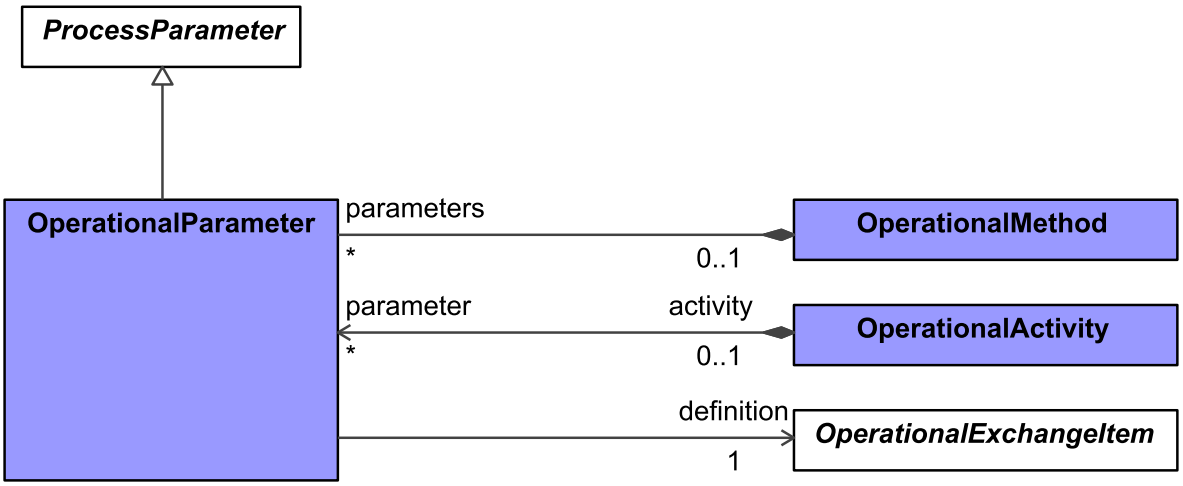

Figure 9:101 - OperationalParameter

OperationalPerformer

Package: Structure
isAbstract: No
Generalization: OperationalAgent, OpposableElement

Description

A logical entity that IsCapableToPerform OperationalActivities which produce, consume and process Resources.

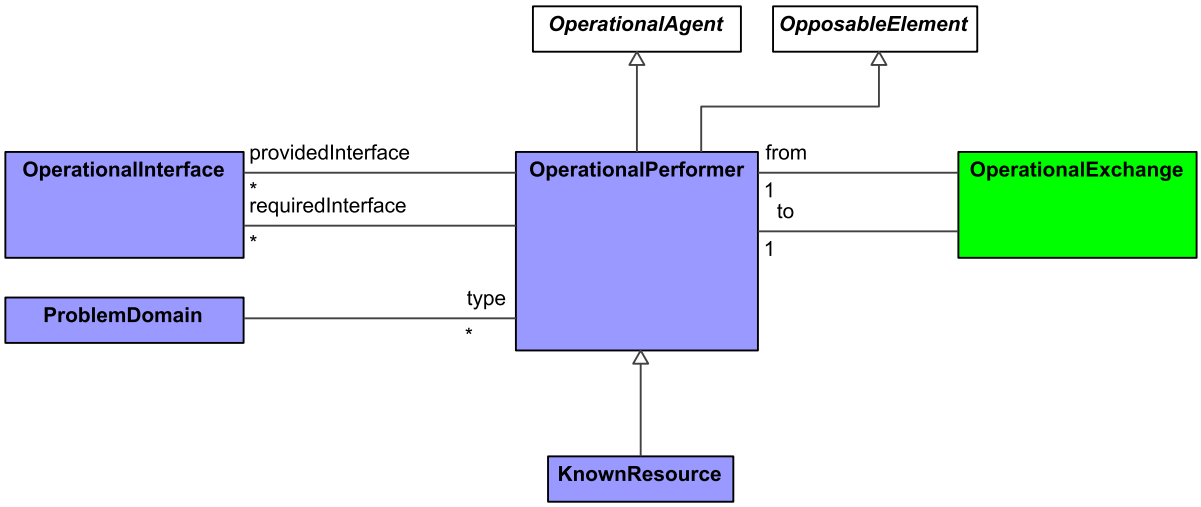

Figure 9:102 - OperationalPerformer

OperationalRole

Package: Structure
isAbstract: No
Generalization: LocationHolder, AssetRole, InteractionRole, OpposableElement

Description

Usage of a OperationalPerformer or OperationalArchitecture in the context of another OperationalPerformer or OperationalArchitecture. Creates a whole-part relationship.

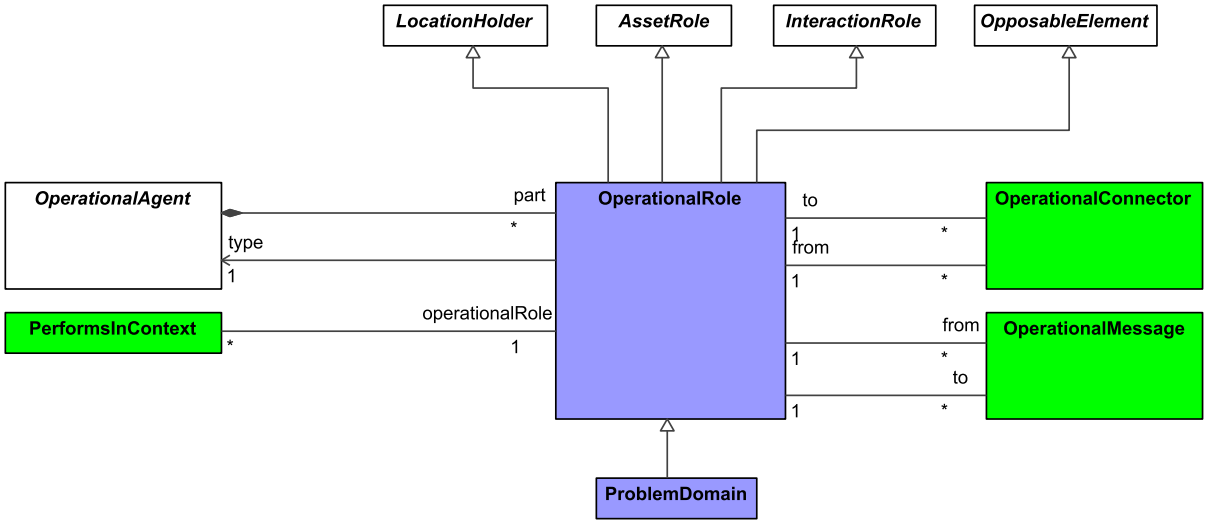

Figure 9:103 - OperationalRole

ProblemDomain

Package: Structure
isAbstract: No
Generalization: OperationalRole

Description

A property associated with an OperationalArchitecture, used to specify the scope of the problem.

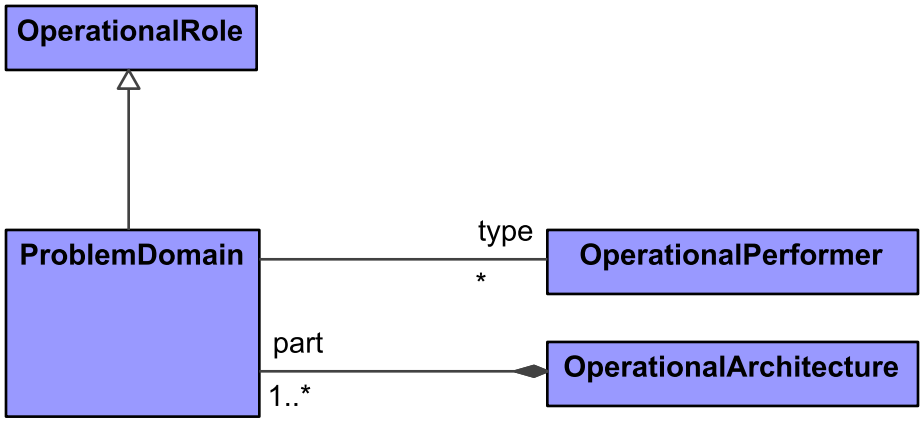

Figure 9:104 - ProblemDomain

9.1.4.3 Domain MetaModel::Operational::Connectivity

OperationalConnector

Package: Connectivity
isAbstract: No
Generalization: AssetRole

Description

A Connector that goes between OperationalRoles representing a need to exchange Resources. It can carry a number of OperationalExchanges.

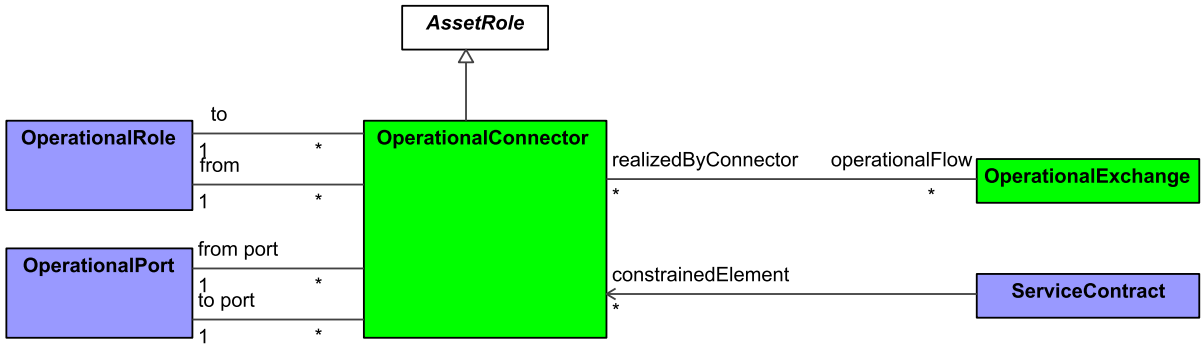

Figure 9:105 - OperationalConnector

OperationalExchange

Package: Connectivity
isAbstract: No
Generalization: Exchange, SubjectOfOperationalConstraint

Description

Asserts that a flow can exist between OperationalPerformers (i.e. flows of information, people, materiel, or energy).

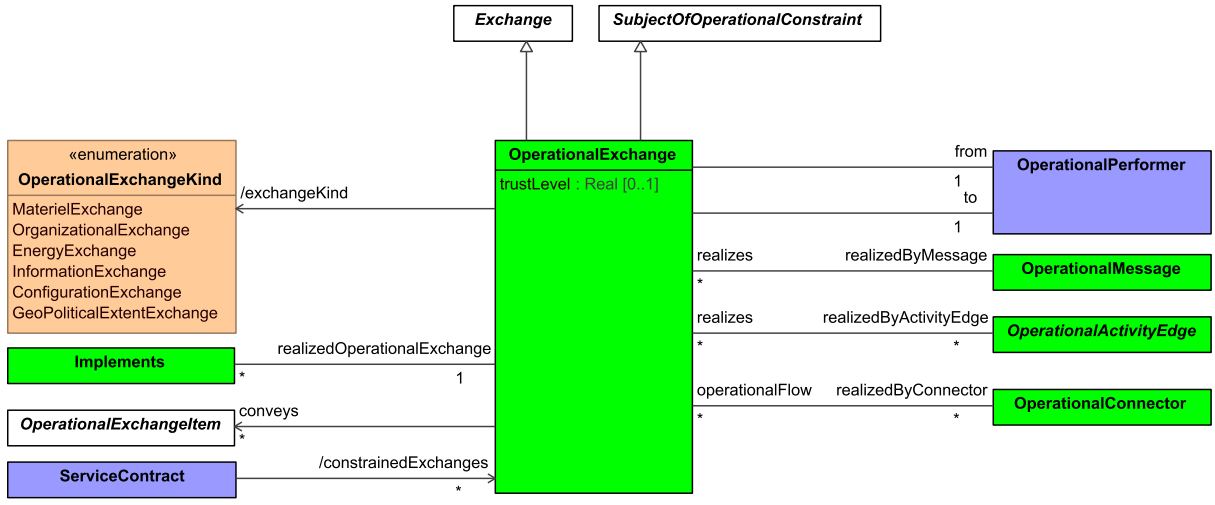

Figure 9:106 - OperationalExchange

Attributes

trustLevel : Real [0..1] Captures the directional arbitrary level of trust related to an OperationalExchange between two OperationalPerformers.

OperationalExchangeItem

Package: Connectivity
isAbstract: Yes
Generalization: Resource, SubjectOfSecurityConstraint, ExchangeItem

Description

An abstract grouping for elements that defines the types of elements that can be exchanged between OperationalPerformers and conveyed by an OperationalExchange.

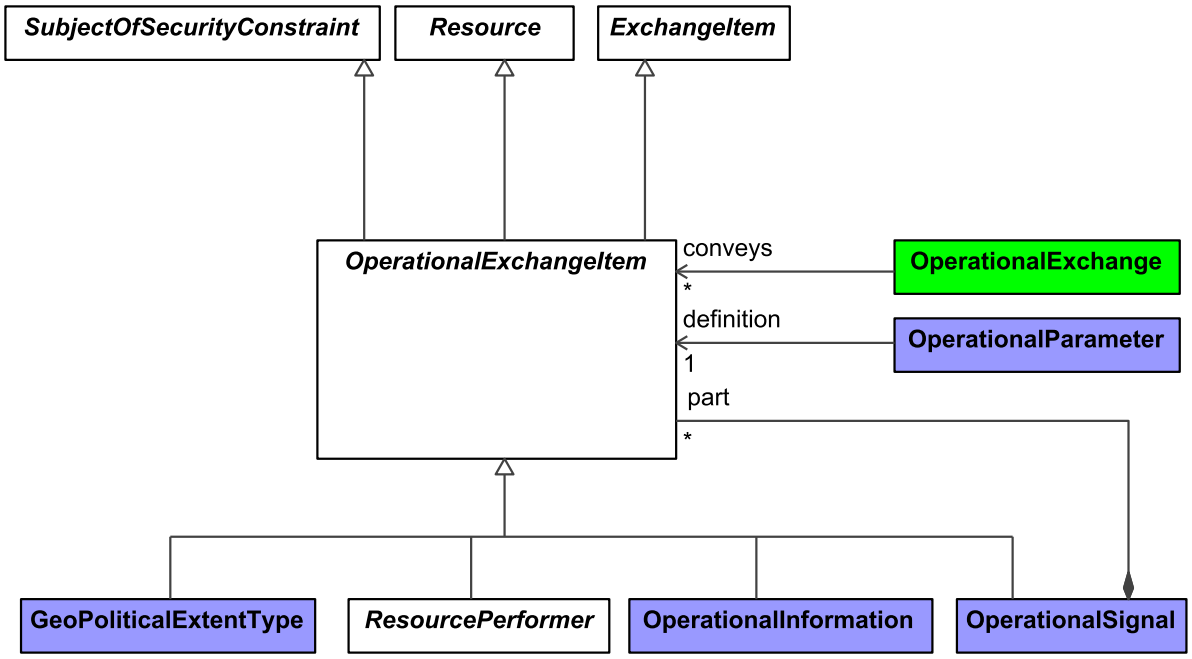

Figure 9:107 - OperationalExchangeItem

OperationalInterface

Package: Connectivity
isAbstract: No
Generalization: PropertySet

Description

A declaration that specifies a contract between the OperationalPerformer it is related to, and any other OperationalPerformers it can interact with.

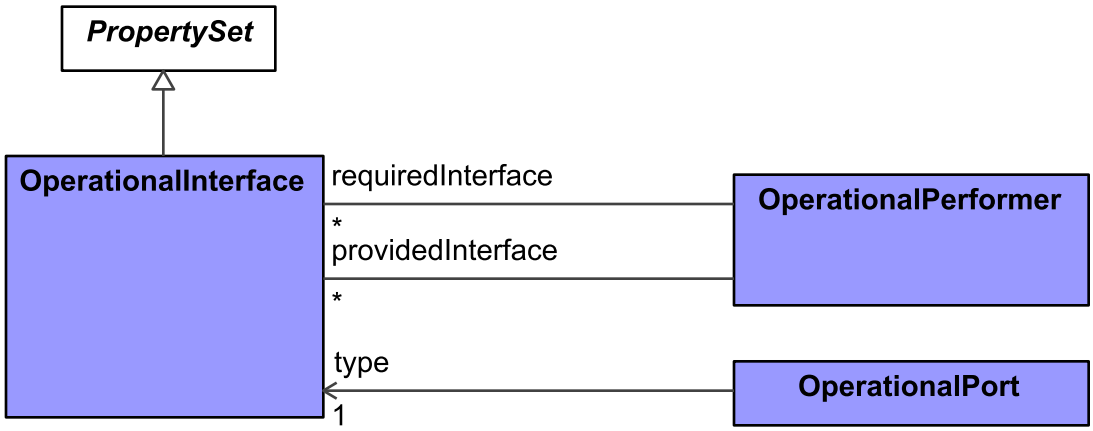

Figure 9:108 - OperationalInterface

OperationalPort

Package: Connectivity
isAbstract: No
Generalization: MeasurableElement

Description

An interaction point for an OperationalAgent through which it can interact with the outside environment and which is defined by an OperationalInterface.

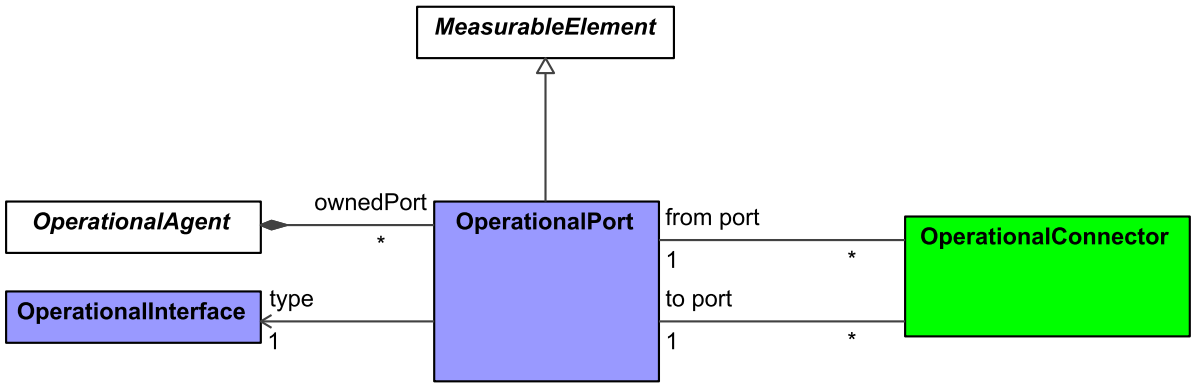

Figure 9:109 - OperationalPort

OperationalSignal

Package: Connectivity
isAbstract: No
Generalization: SubjectOfOperationalConstraint, OperationalExchangeItem

Description

An item of information that flows between OperationalPerformers and is produced and consumed by the OperationalActivities that the OperationalPerformers are capable of performing (see IsCapableToPerform).

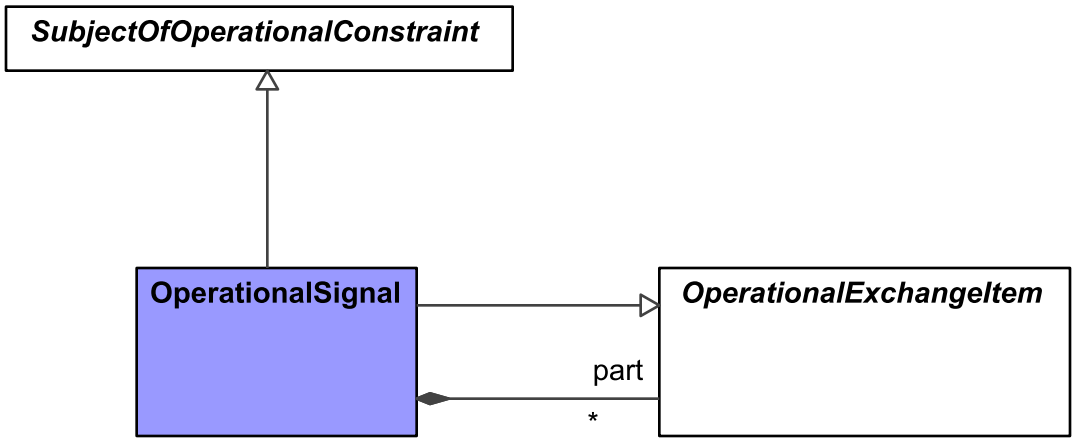

Figure 9:110 - OperationalSignal

9.1.4.4 Domain MetaModel::Operational::Processes

MissionTask

Package: Processes
isAbstract: No
Generalization: MissionThread

Description

A clearly defined activity specifically to be assigned to system, person, or organization that must be completed

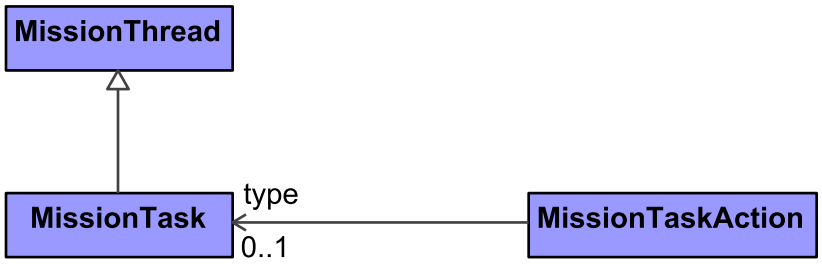

Figure 9:111 - MissionTask

MissionTaskAction

Package: Processes
isAbstract: No
Generalization: OperationalActivityAction

Description

The usage of a MissionTask within a MissionThread

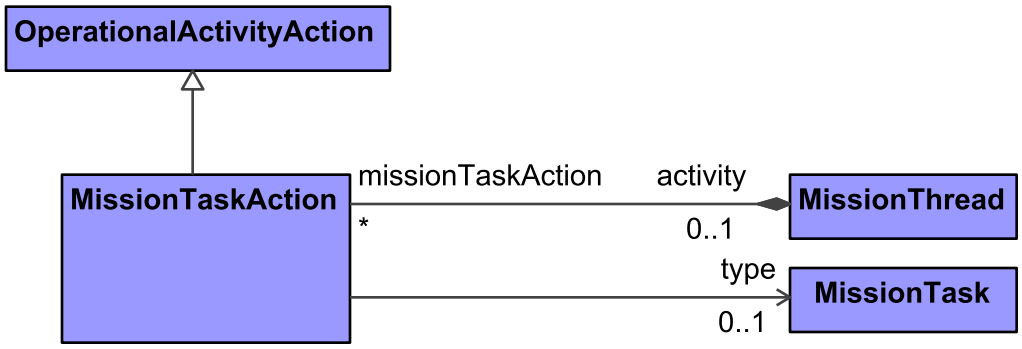

Figure 9:112 - MissionTaskAction

MissionThread

Package: Processes
isAbstract: No
Generalization: OperationalActivity

Description

A sequence of end-to-end mission tasks, activities, and events presented as a series of steps to achieve a mission. (OUSD(R&E))

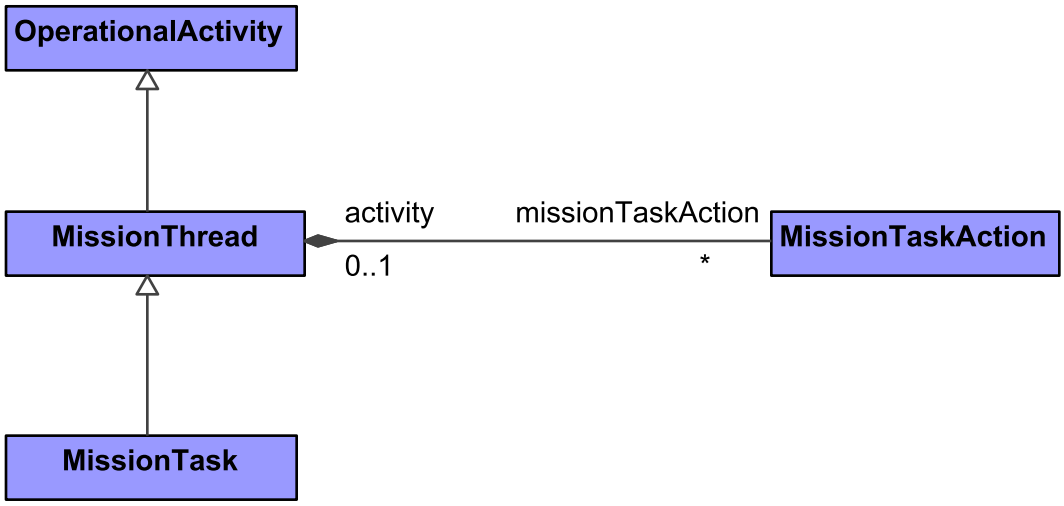

Figure 9:113 - MissionThread

OperationalActivity

Package: Processes
isAbstract: No
Generalization: SubjectOfOperationalConstraint, Process

Description

An Activity that captures a logical process, specified independently of how the process is carried out.

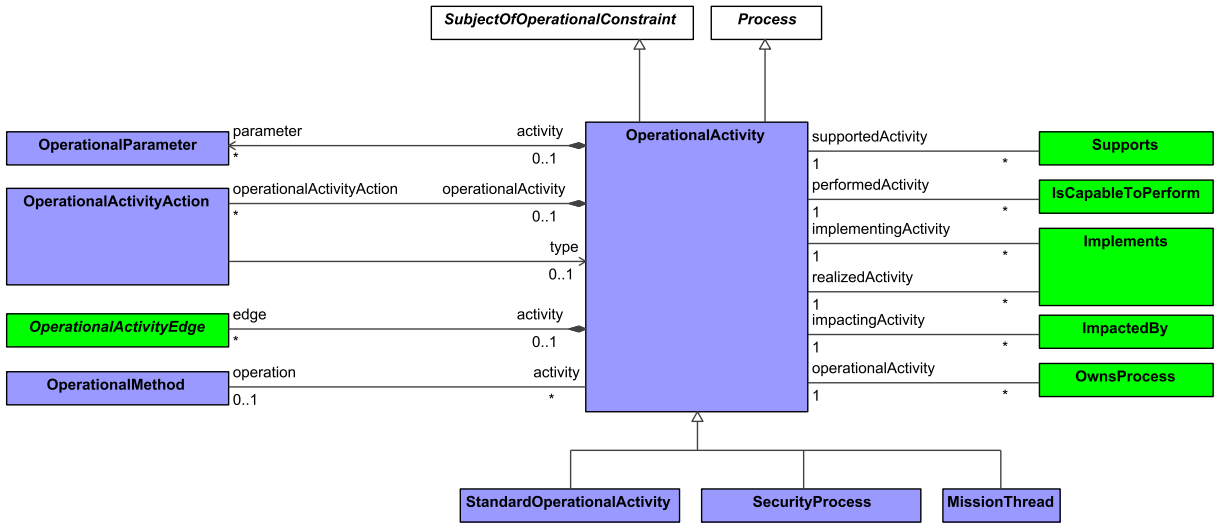

Figure 9:114 - OperationalActivity

OperationalActivityAction

Package: Processes
isAbstract: No
Generalization: ProcessUsage, MeasurableElement

Description

A call of an OperationalActivity in the context of another OperationalActivity.

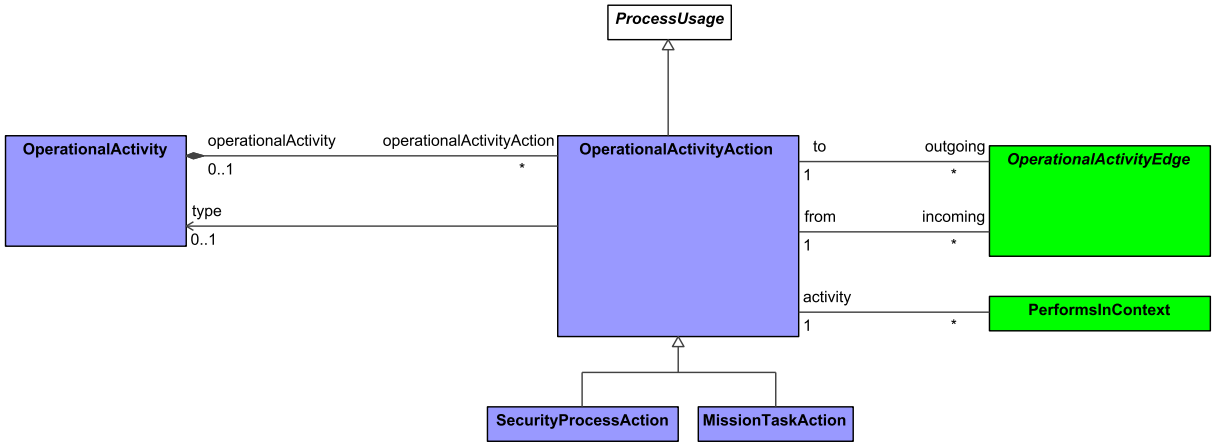

Figure 9:115 - OperationalActivityAction

OperationalActivityEdge

Package: Processes
isAbstract: Yes
Generalization: ProcessEdge

Description

A tuple that shows the flow of Resources (objects/information) between OperationalActivityActions.

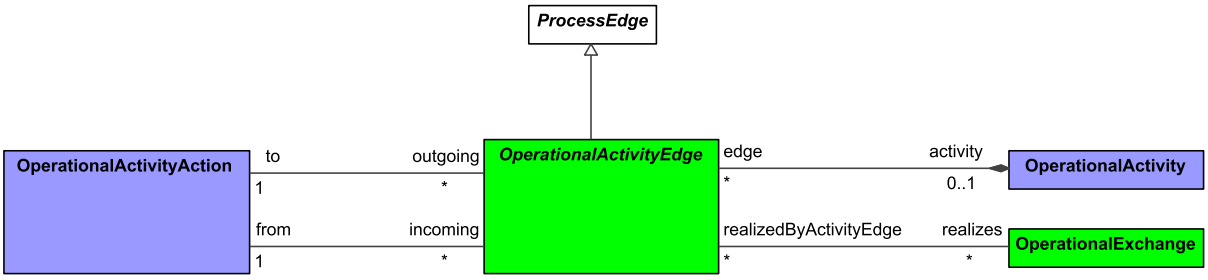

Figure 9:116 - OperationalActivityEdge

StandardOperationalActivity

Package: Processes
isAbstract: No
Generalization: OperationalActivity

Description

A sub-type of OperationalActivity that is a standard operating procedure.

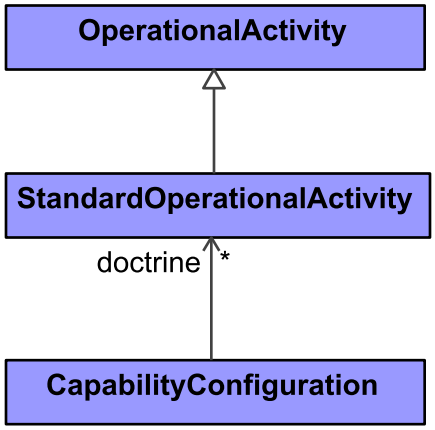

Figure 9:117 - StandardOperationalActivity

9.1.4.5 Domain MetaModel::Operational::States

OperationalStateDescription

Package: States
isAbstract: No
Generalization: MeasurableElement, StateDescription

Description

A state machine describing the behavior of a OperationalPerformer, depicting how the OperationalPerformer responds to various events and the actions.

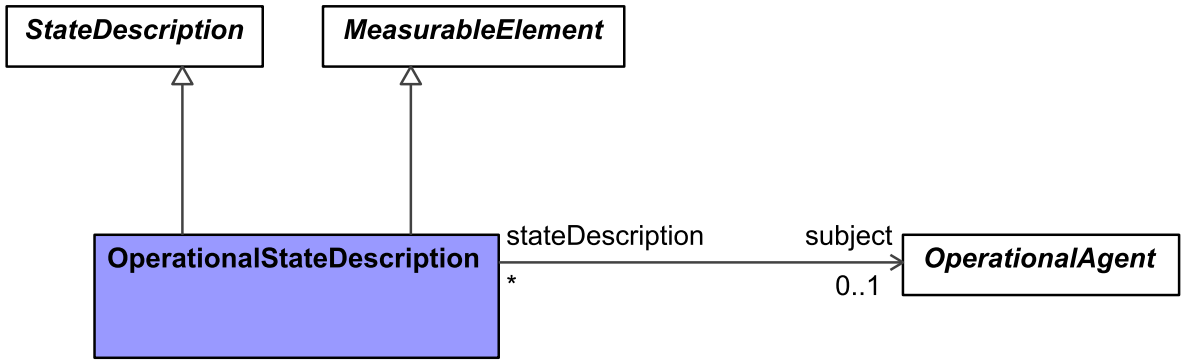

Figure 9:118 - OperationalStateDescription

9.1.4.6 Domain MetaModel::Operational::Sequences

OperationalInteractionScenario

Package: Sequences
isAbstract: No
Generalization: InteractionScenario

Description

A specification of the interactions between OperationalPerformers in an OperationalArchitecture.

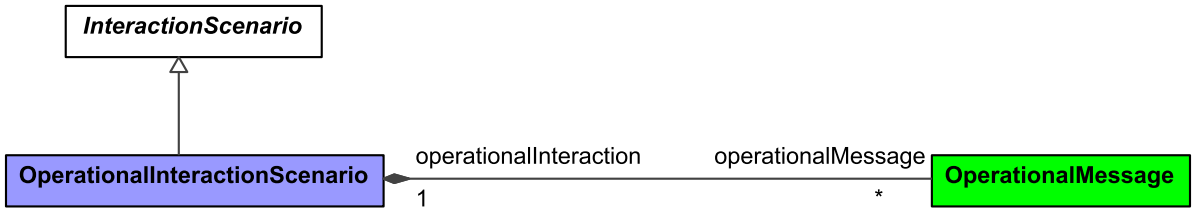

Figure 9:119 - OperationalInteractionScenario

OperationalMessage

Package: Sequences
isAbstract: No
Generalization: InteractionMessage

Description

Message for use in an OperationalInteractionScenario which carries any of the subtypes of OperationalExchange.

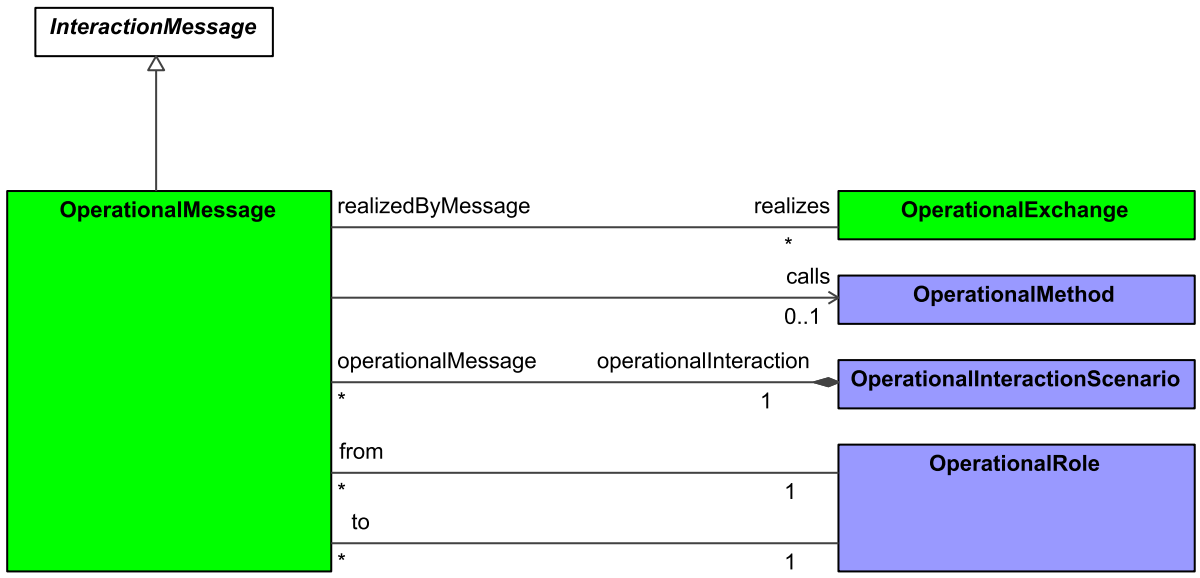

Figure 9:120 - OperationalMessage

9.1.4.7 Domain MetaModel::Operational::Information

OperationalInformation

Package: Information
isAbstract: No
Generalization: SubjectOfOperationalConstraint, OperationalAsset, OperationalExchangeItem, ServiceExchangeItem

Description

An item of information that flows between OperationalPerformers and is produced and consumed by the OperationalActivities that the OperationalPerformers are capable to perform (see IsCapableToPerform).

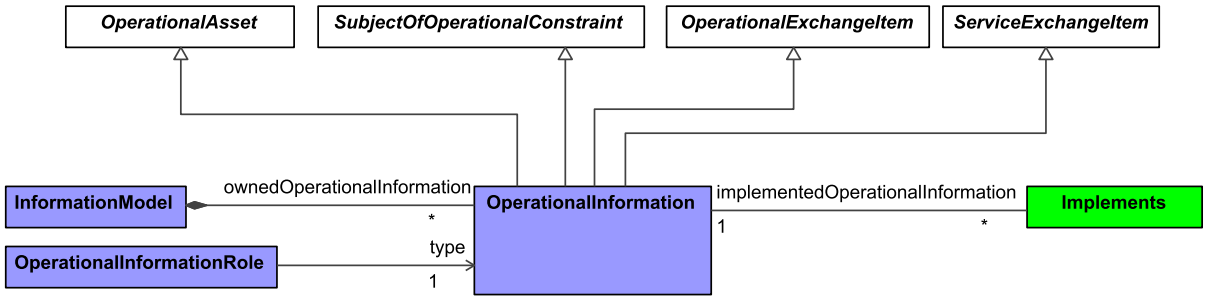

Figure 9:121 - OperationalInformation

9.1.4.8 Domain MetaModel::Operational::Constraints

OperationalConstraint

Package: Constraints
isAbstract: No
Generalization: Rule

Description

A Rule governing an operational architecture element i.e. OperationalPerformer, OperationalActivity, OperationalInformation etc.

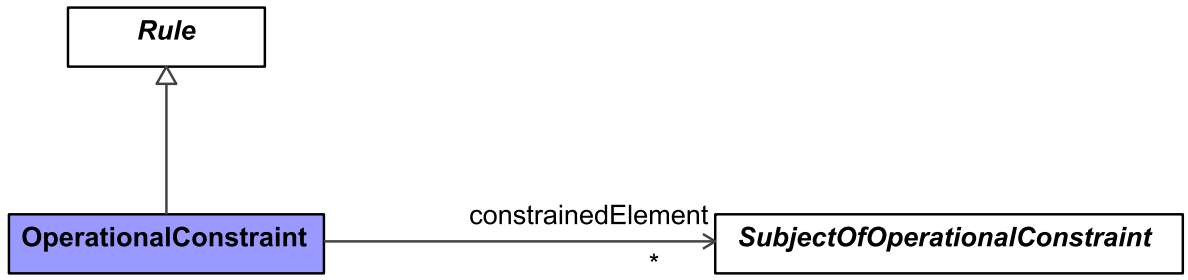

Figure 9:122 - OperationalConstraint

SubjectOfOperationalConstraint

Package: Constraints
isAbstract: Yes
Generalization: UAFElement

Description

An abstract type grouping elements that can be the subject of an OperationalConstraint.

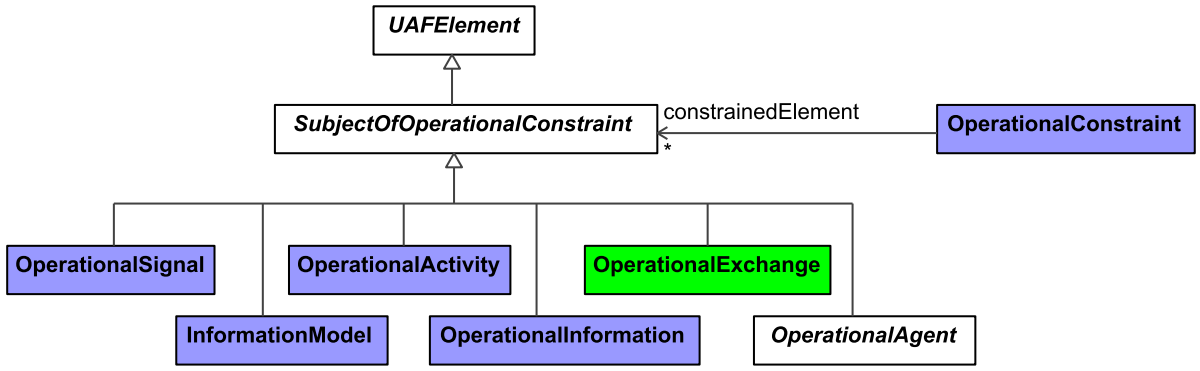

Figure 9:123 - SubjectOfOperationalConstraint

9.1.5 Domain MetaModel::Services

Stakeholders: Enterprise Architects, Solution Providers, Systems Engineers, Software Architects, Business Architects..

Concerns: specifications of services required to exhibit a Capability.

Definition: shows Services and required and provided service levels of these services needed to exhibit a Capability or to support an Operational Activity.

9.1.5.1 Domain MetaModel::Services::Taxonomy

Service

Package: Taxonomy

isAbstract: No
Generalization: PropertySet, VersionedElement, CapableElement, Asset

Description

A mechanism to enable access to one or more capabilities, where the access is provided using a prescribed service interface and is exercised consistent with service constraints and policies.

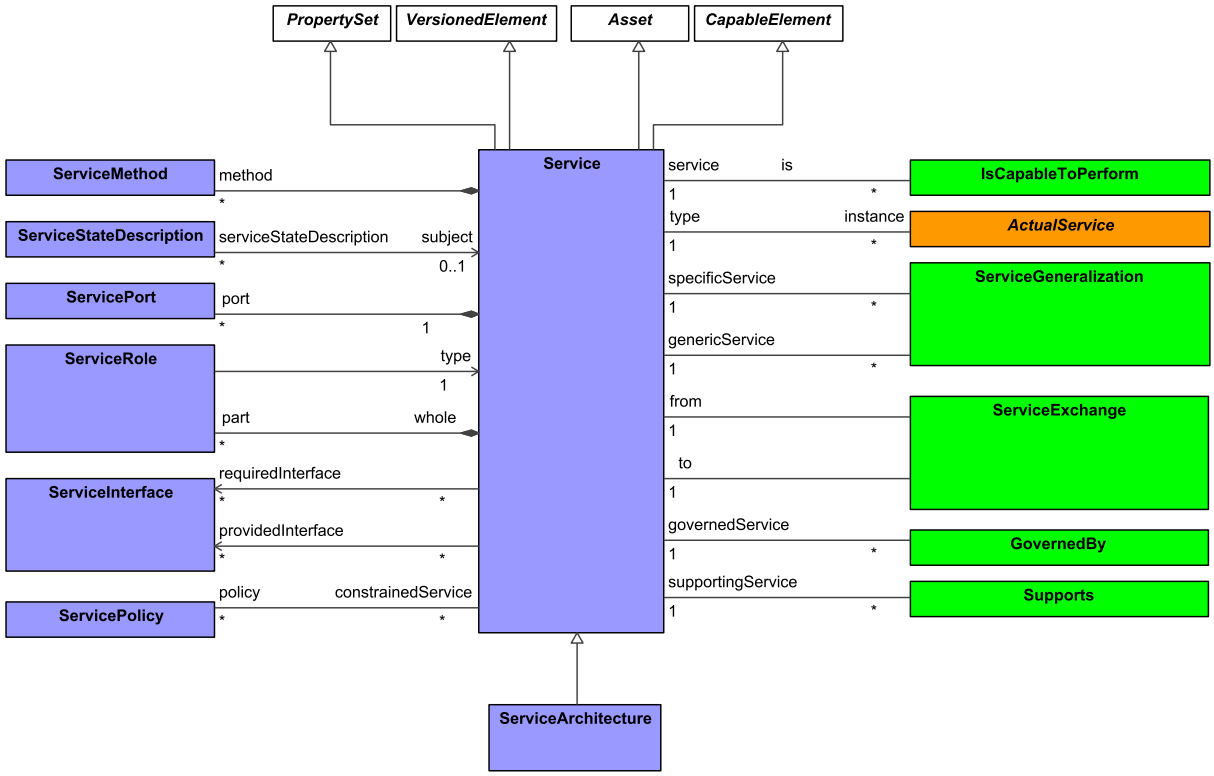

Figure 9:124 - Service

ServiceArchitecture

Package: Taxonomy
isAbstract: No
Generalization: Service, Architecture

Description

An element used to denote a model of the Architecture, described from the Services perspective.

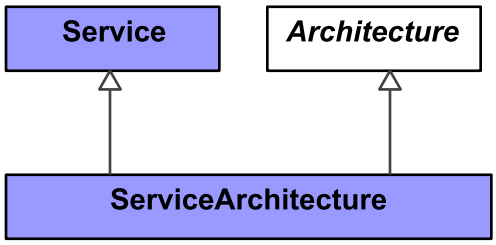

Figure 9:125 - ServiceArchitecture

ServiceGeneralization

Package: Taxonomy
isAbstract: No
Generalization: PropertySetGeneralization

Description

A ServiceGeneralization is a taxonomic relationship between a more general Service and a more specific Service.

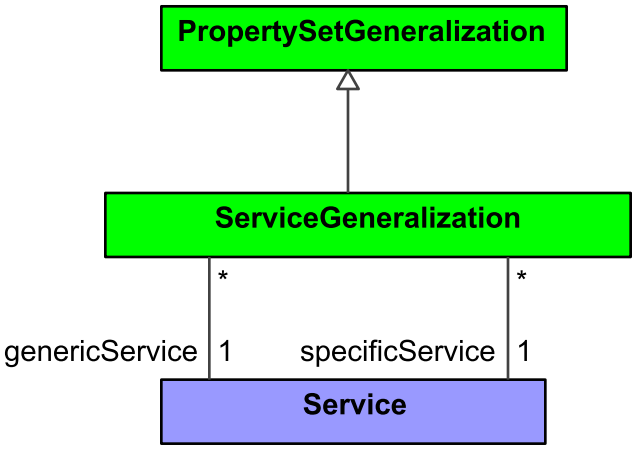

Figure 9:126 - ServiceGeneralization

9.1.5.2 Domain MetaModel::Services::Structure

ServiceMethod

Package: Structure
isAbstract: No
Generalization: ProcessOperation

Description

A behavioral feature of a Service whose behavior is specified in a ServiceFunction.

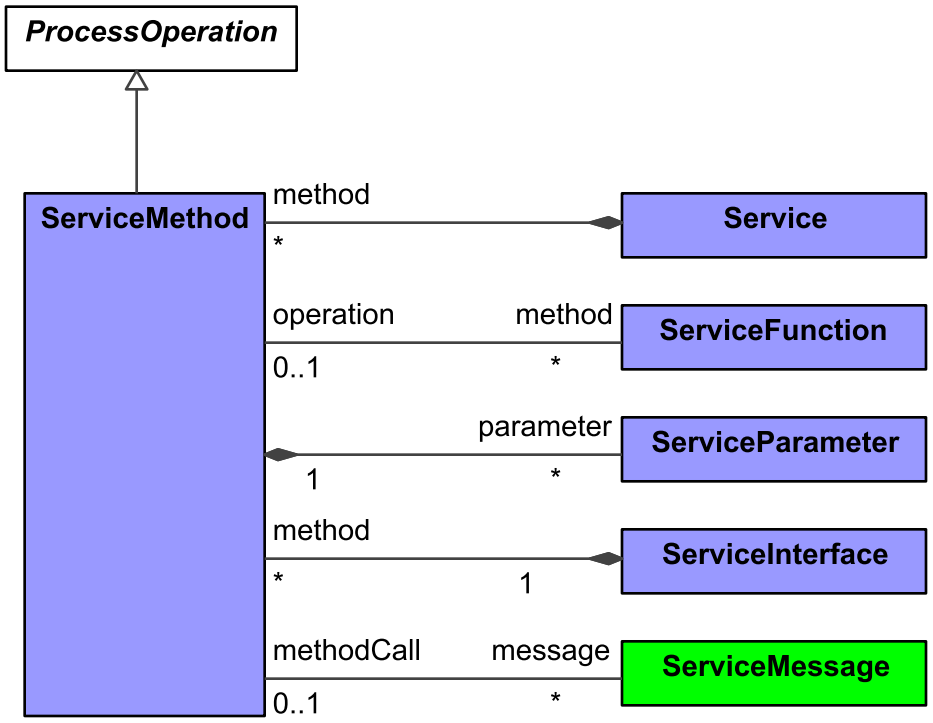

Figure 9:127 - ServiceMethod

ServiceParameter

Package: Structure
isAbstract: No
Generalization: ProcessParameter

Description

A type that represents inputs and outputs of a ServiceFunction, represents inputs and outputs of a Service.

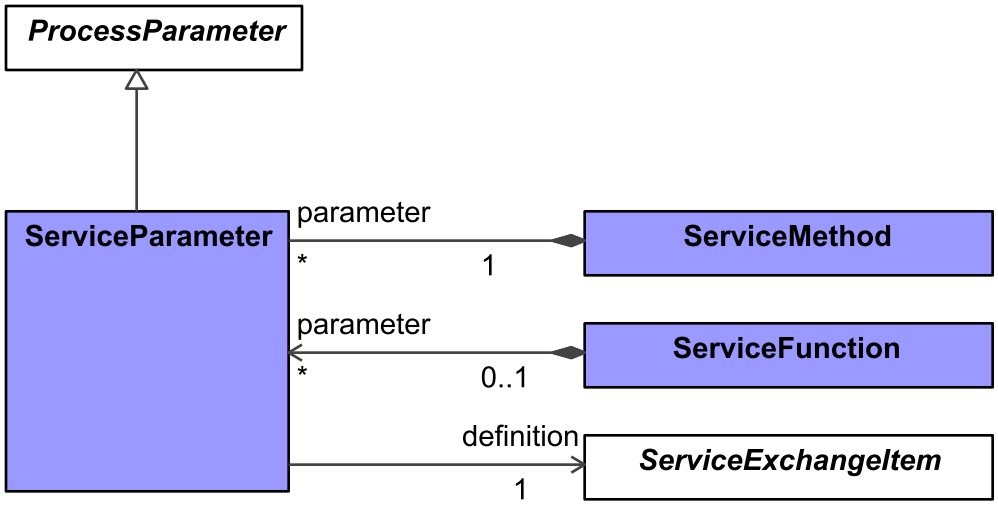

Figure 9:128 - ServiceParameter

ServiceRole

Package: Structure
isAbstract: No
Generalization: MeasurableElement, InteractionRole

Description

A behavioral feature of a Service whose behavior is specified in a ServiceFunction.

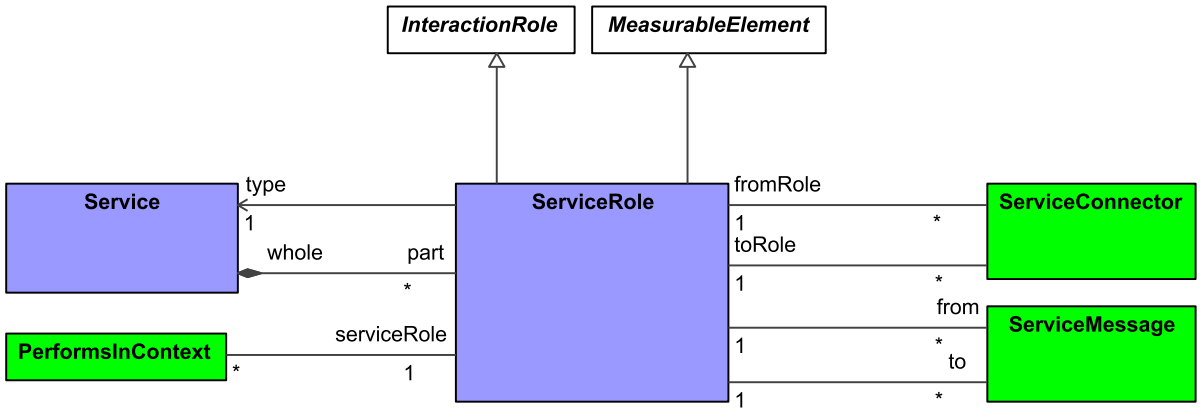

Figure 9:129 - ServiceRole

9.1.5.3 Domain MetaModel::Services::Connectivity

ServiceConnector

Package: Connectivity
isAbstract: No
Generalization: AssetRole

Description

A channel for exchange between two Service. Where one acts as the consumer of the other.

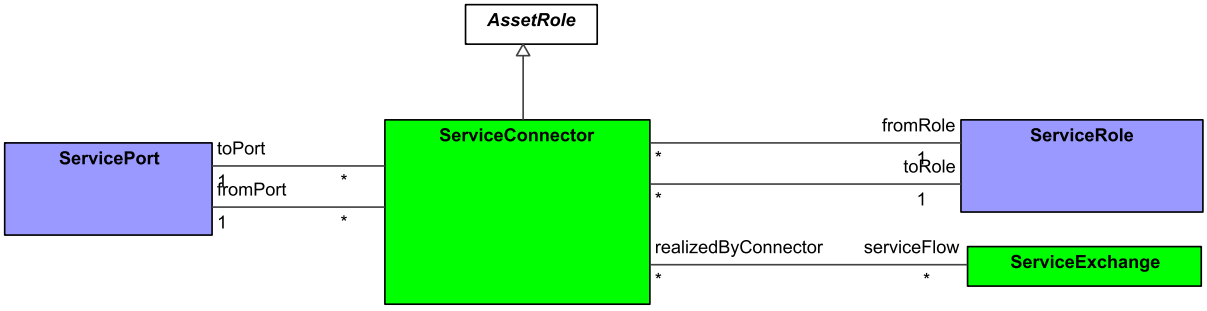

Figure 9:130 - ServiceConnector

ServiceExchange

Package: Connectivity
isAbstract: No
Generalization: Exchange

Description

Asserts that a flow can exist between Services (i.e. flows of information, people, materiel, or energy).

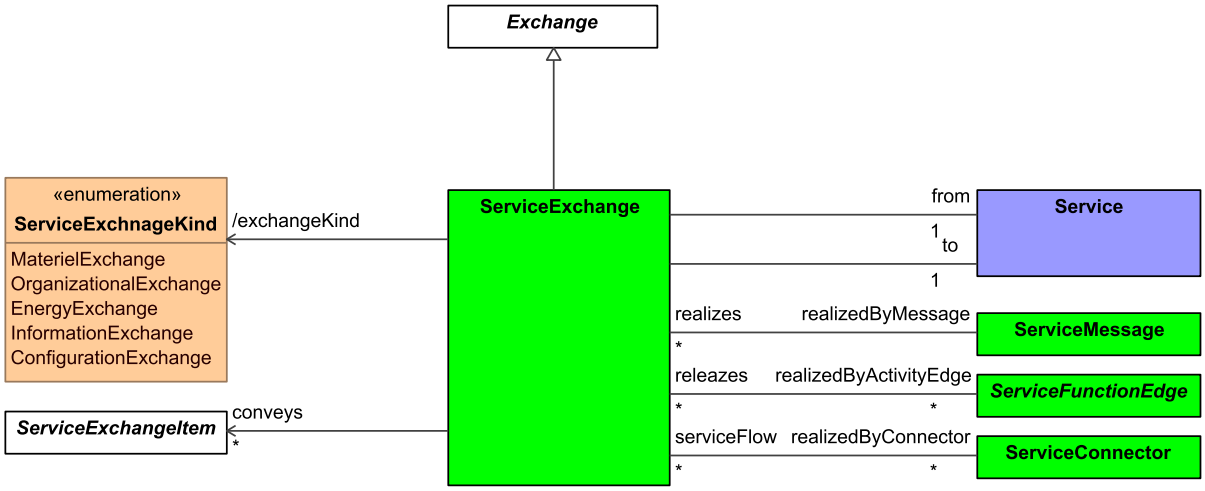

Figure 9:131 - ServiceExchange

ServiceExchangeItem

Package: Connectivity
isAbstract: Yes
Generalization: Resource, ExchangeItem

Description

An abstract grouping for elements that defines the types of elements that can be exchanged between Services and conveyed by a ServiceExchange.

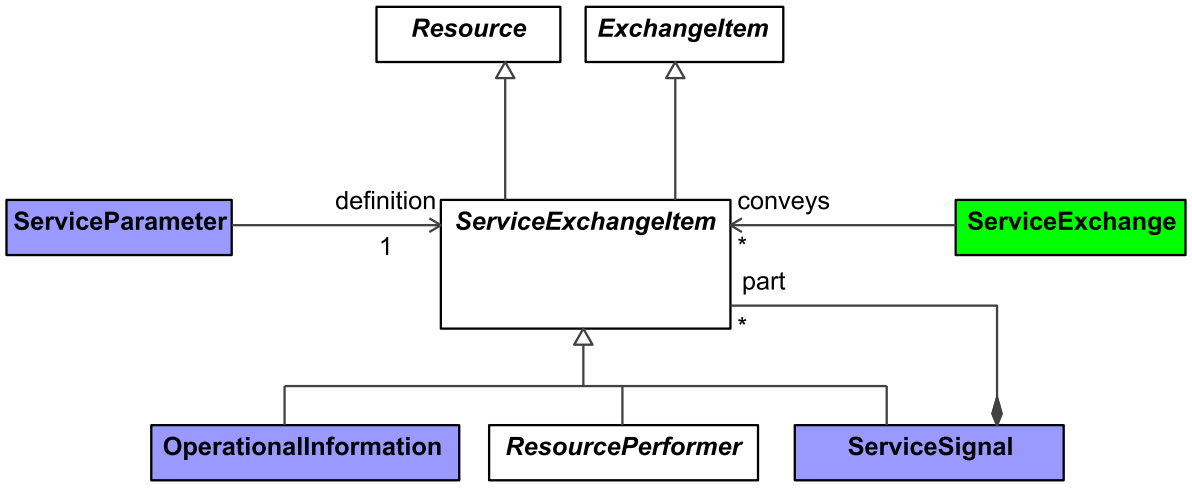

Figure 9:132 - ServiceExchangeItem

ServiceInterface

Package: Connectivity
isAbstract: No
Generalization: PropertySet

Description

A contract that defines the ServiceMethods and ServiceSignals that the Service realizes.

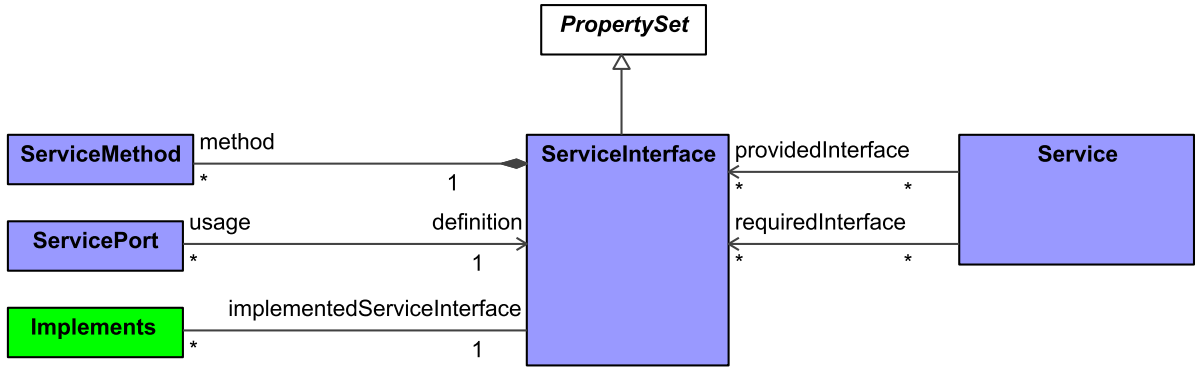

Figure 9:133 - ServiceInterface

ServicePort

Package: Connectivity
isAbstract: No
Generalization: MeasurableElement

Description

An interaction point for a Service through which it can interact with the outside environment and which is defined by a ServiceInterface.

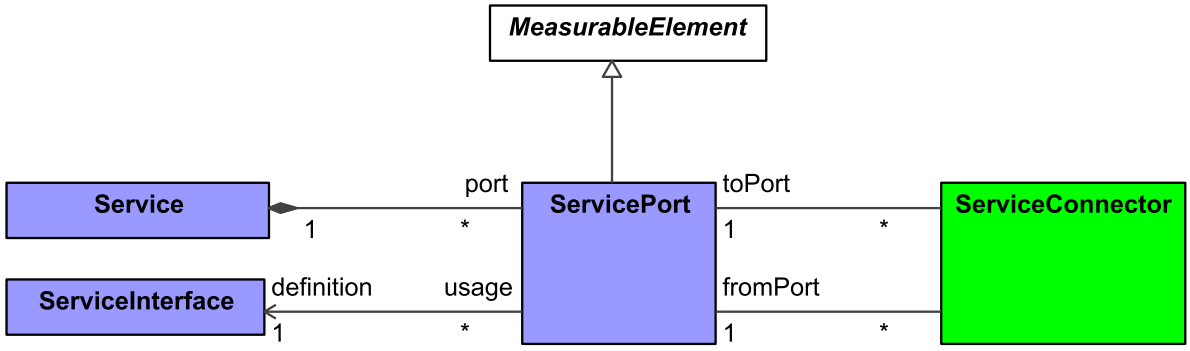

Figure 9:134 - ServicePort

ServiceSignal

Package: Connectivity
isAbstract: No
Generalization: ServiceExchangeItem

Description

A specification of a kind of communication between Services in which a reaction is asynchronously triggered in the receiver without a reply.

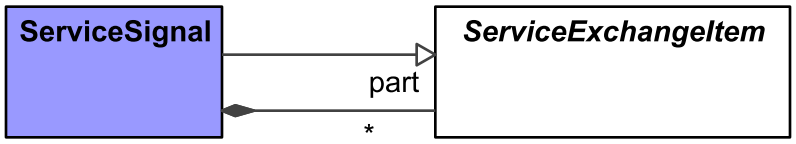

Figure 9:135 - ServiceSignal

ServiceSignalProperty

Package: Connectivity
isAbstract: No

Description

A property of a ServiceSignal typed by ServiceExchangeItem. It enables ServiceExchangeItem e.g. OperationalInformation to be passed as arguments of the ServiceSignal.

9.1.5.4 Domain MetaModel::Services::Processes

ServiceFunction

Package: Processes
isAbstract: No
Generalization: Process

Description

An Activity that describes the abstract behavior of Service, regardless of the actual implementation.

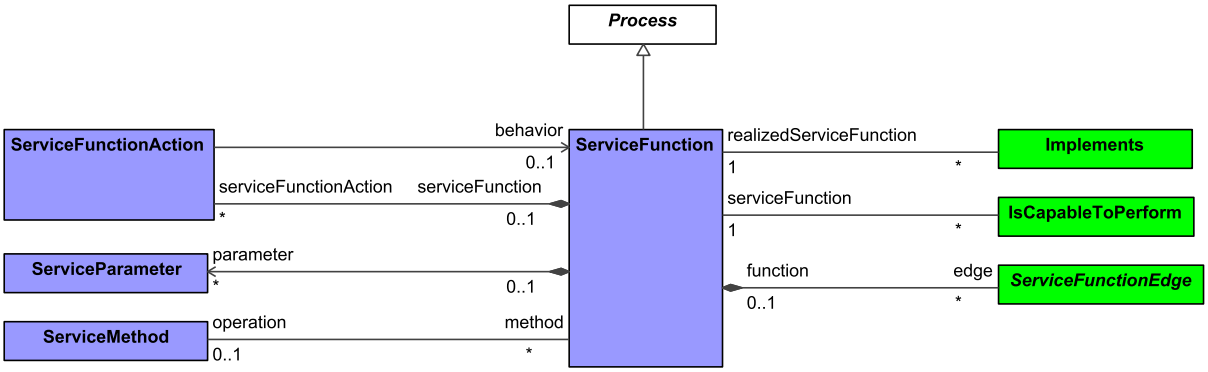

Figure 9:136 - ServiceFunction

ServiceFunctionAction

Package: Processes
isAbstract: No
Generalization: ProcessUsage

Description

A call of a ServiceFunction in the context of another ServiceFunction.

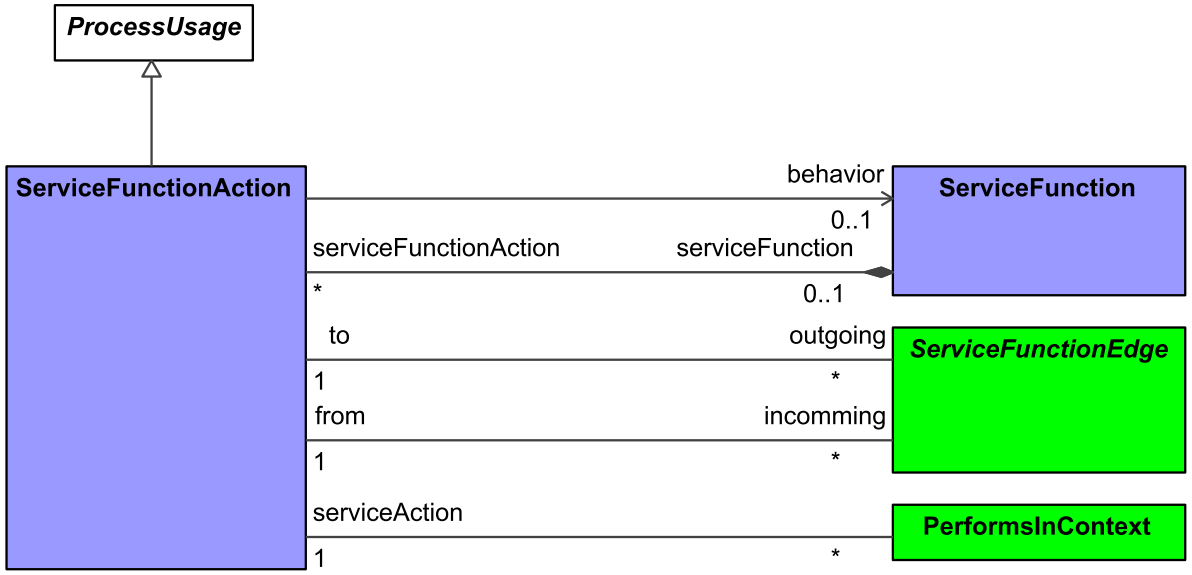

Figure 9:137 - ServiceFunctionAction

ServiceFunctionEdge

Package: Processes
isAbstract: Yes
Generalization: ProcessEdge

Description

A tuple that shows the flow of Resources (objects/information) between OperationalActivityActions.

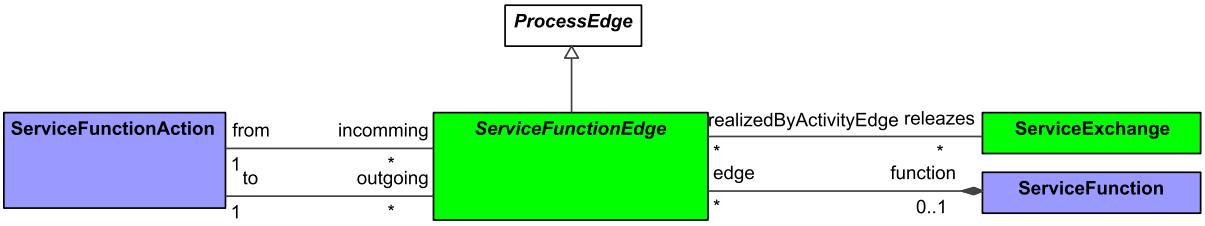

Figure 9:138 - ServiceFunctionEdge

9.1.5.5 Domain MetaModel::Services::States

ServiceStateDescription

Package: States
isAbstract: No
Generalization: MeasurableElement, StateDescription

Description

A state machine describing the behavior of a Service, depicting how the Service responds to various events and the actions.

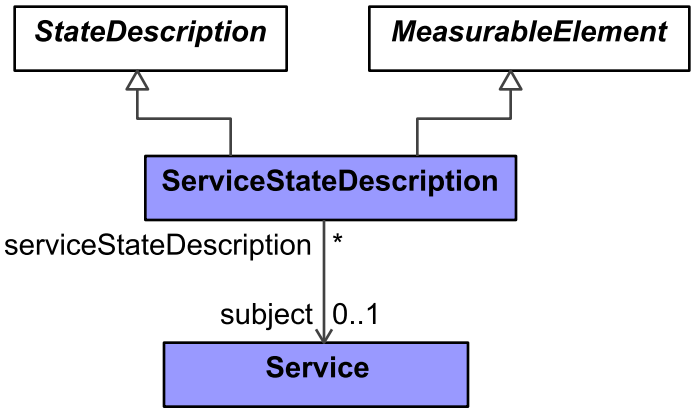

Figure 9:139 - ServiceStateDescription

9.1.5.6 Domain MetaModel::Services::Sequences

ServiceInteractionScenario

Package: Sequences
isAbstract: No
Generalization: InteractionScenario

Description

A specification of the interactions between Service.

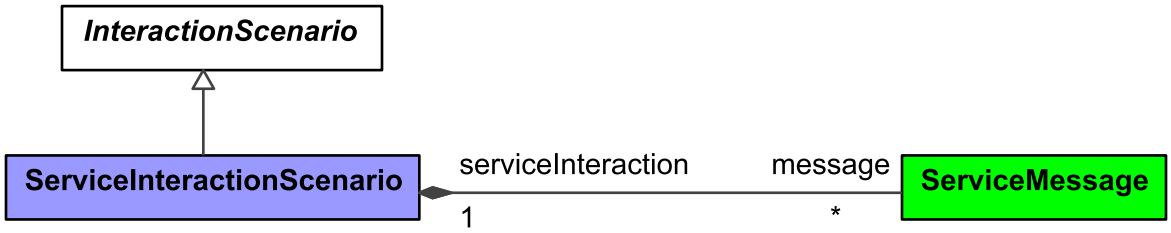

Figure 9:140 - ServiceInteractionScenario

ServiceMessage

Package: Sequences
isAbstract: No
Generalization: InteractionMessage

Description

Message for use in a services interaction scenario which carries any of the subtypes of ServiceExchange.

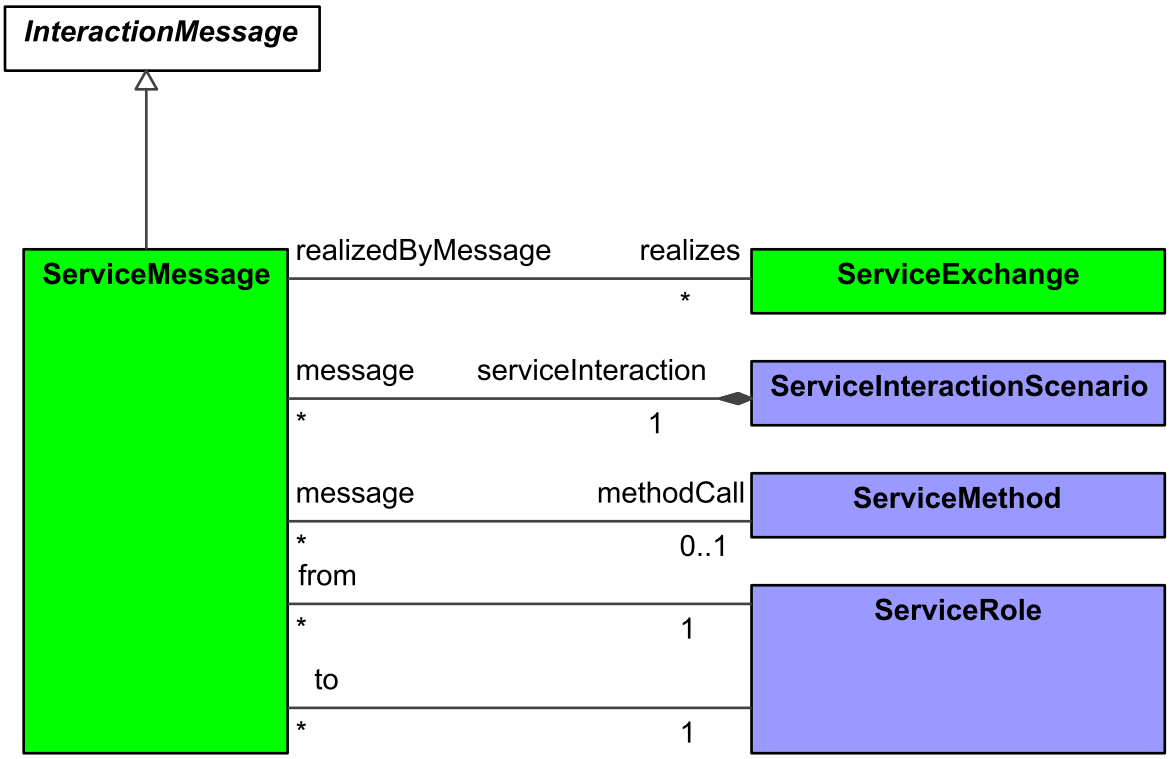

Figure 9:141 - ServiceMessage

9.1.5.7 Domain MetaModel::Services::Constraints

ServiceContract

Package: Constraints
isAbstract: No
Generalization: Rule

Description

A constraint governing the use of one or more Services.

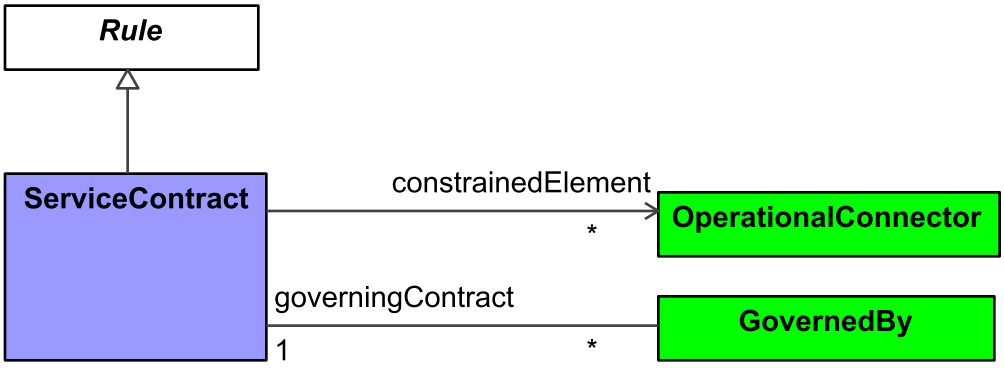

Figure 9:142 - ServiceContract

ServicePolicy

Package: Constraints
isAbstract: No
Generalization: Rule

Description

A constraint governing the use of one or more Service.

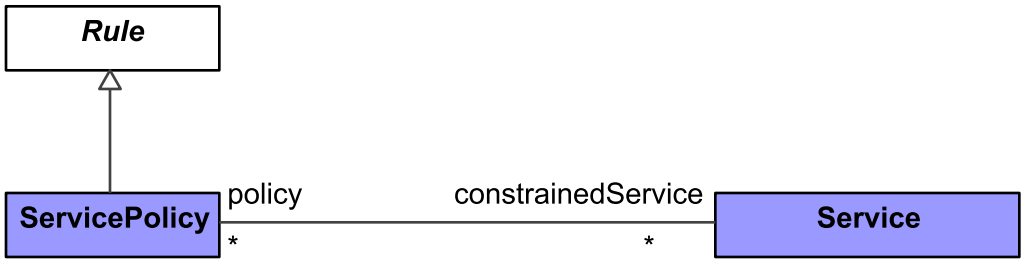

Figure 9:143 - ServicePolicy

9.1.5.8 Domain MetaModel::Services::Traceability

GovernedBy

Package: Traceability
isAbstract: No
Generalization: MeasurableElement

Description

A tuple that exists between the ServiceContract and the Service that it governs.

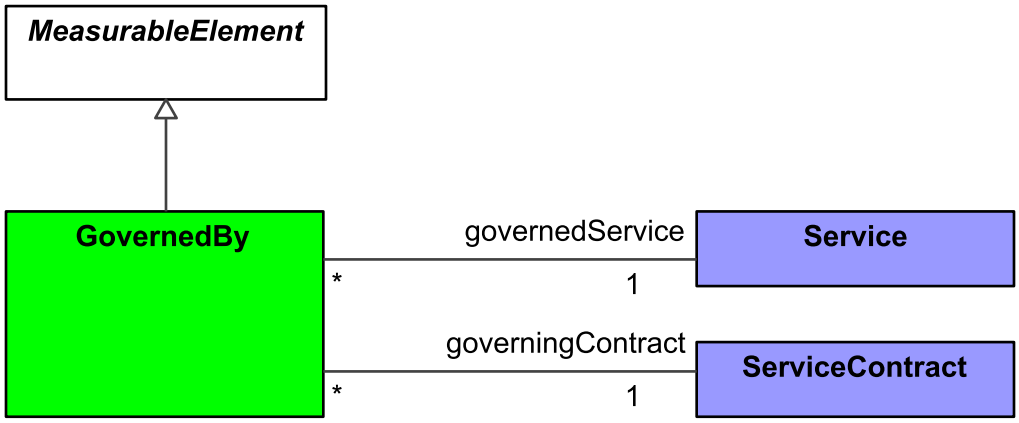

Figure 9:144 - GovernedBy

Supports

Package: Traceability
isAbstract: No
Generalization: MeasurableElement

Description

A tuple that asserts that asserts that a service in someway contributes or assists in the execution of an OperationalActivity.

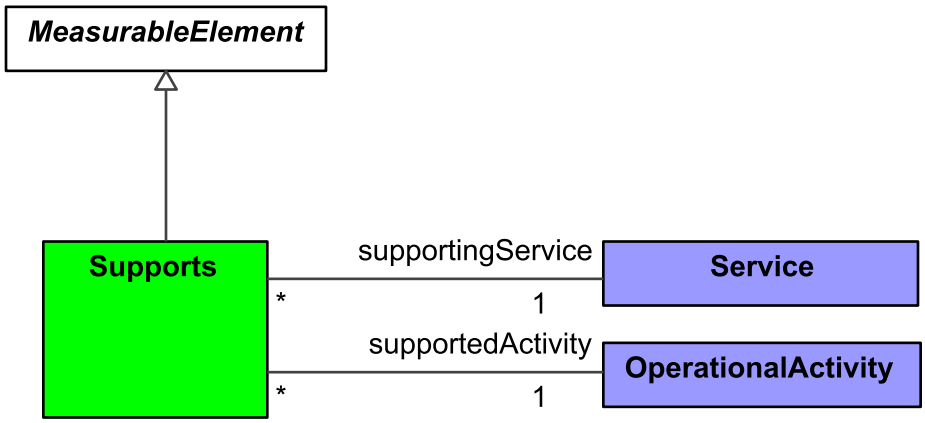

Figure 9:145 - Supports

9.1.6 Domain MetaModel::Personnel

Stakeholders: Human resources, Solution Providers, PMs.

Concerns: human factors.

Definition: aims to clarify the role of Human Factors (HF) when creating architectures in order to facilitate both Human Factors Integration (HFI) and systems engineering (SE).

9.1.6.1 Domain MetaModel::Personnel::Taxonomy

Organization

Package: Taxonomy
isAbstract: No
Generalization: OrganizationalResource

Description

A group of OrganizationalResources (Persons, Posts, Organizations and Responsibilities) associated for a particular purpose.

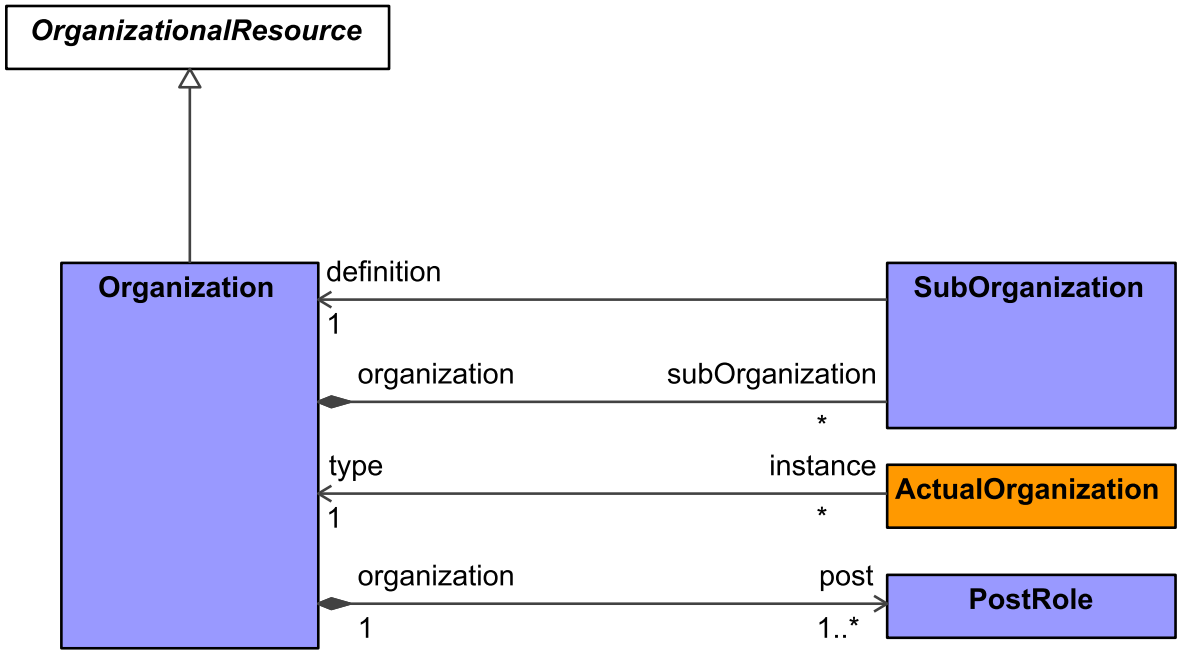

Figure 9:146 - Organization

OrganizationalResource

Package: Taxonomy
isAbstract: Yes
Generalization: PhysicalResource, Stakeholder

Description

An abstract type for Organization, Person, Post and Responsibility.

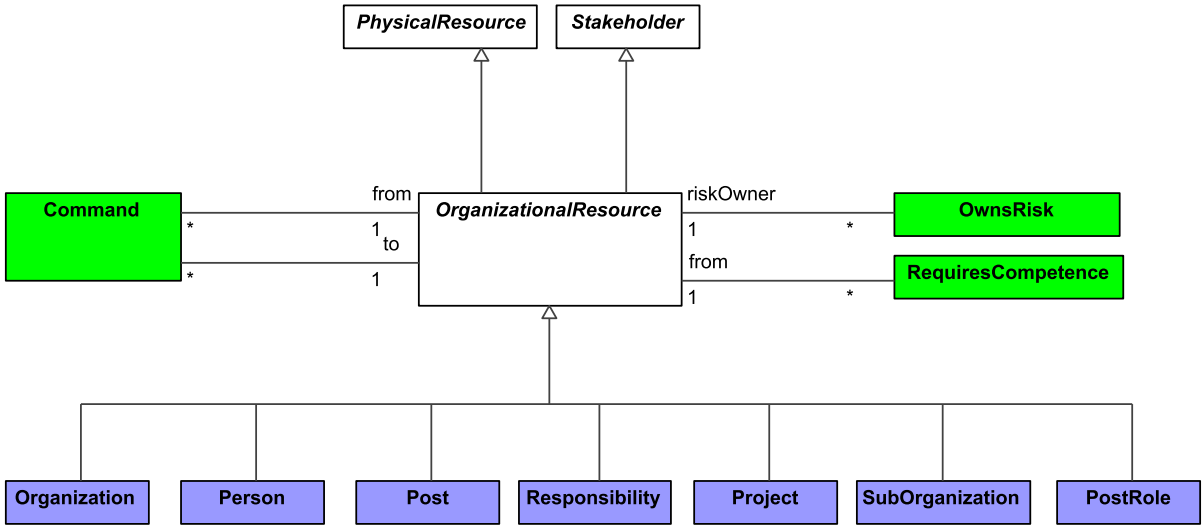

Figure 9:147 - OrganizationalResource

Person

Package: Taxonomy
isAbstract: No
Generalization: OrganizationalResource

Description

A type of a human being used to define the characteristics that need to be described for ActualPersons (e.g. properties such as address, telephone number, nationality, etc).

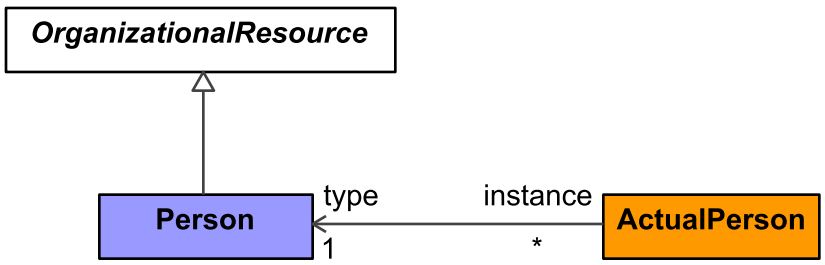

Figure 9:148 - Person

Post

Package: Taxonomy
isAbstract: No
Generalization: OrganizationalResource

Description

A type of job title or position that a person can fill (e.g. Lawyer, Solution Architect, Machine Operator or Chief Executive Officer).

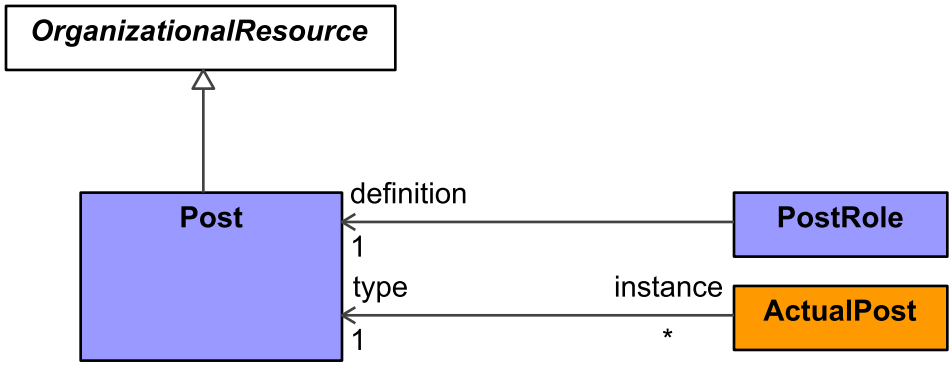

Figure 9:149 - Post

Responsibility

Package: Taxonomy
isAbstract: No
Generalization: OrganizationalResource

Description

The type of duty required of a Person or Organization.

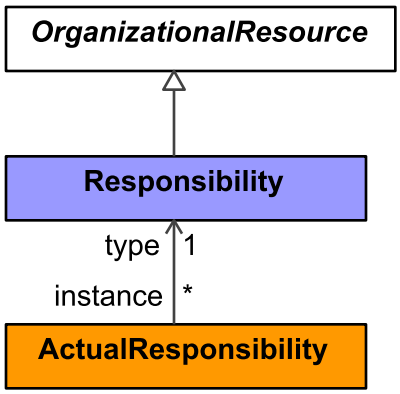

Figure 9:150 - Responsibility

9.1.6.2 Domain MetaModel::Personnel::Structure

PostRole

Package: Structure
isAbstract: No
Generalization: OrganizationalResource, ResourceRole

Description

A usage of a post in the context of another OrganizationalResource. Creates a whole-part relationship.

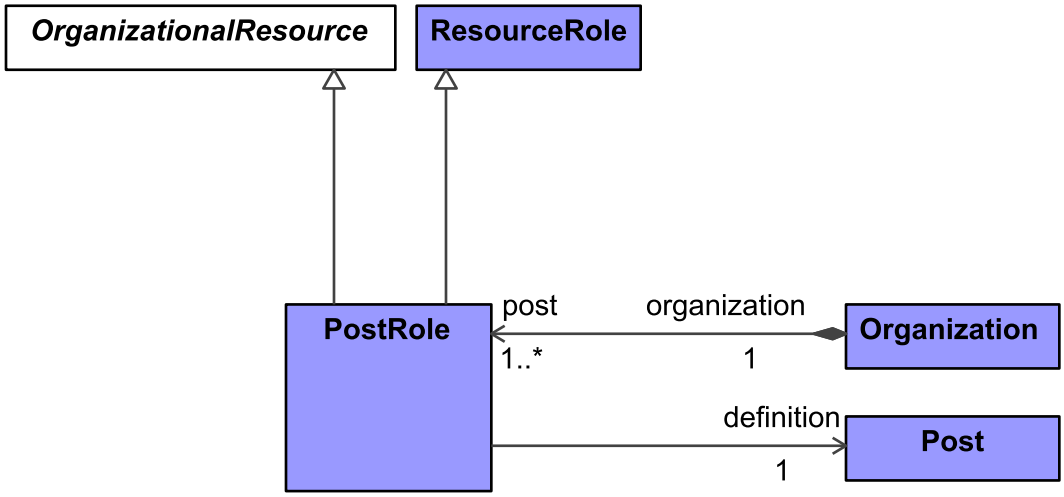

Figure 9:151 - PostRole

SubOrganization

Package: Structure
isAbstract: No
Generalization: OrganizationalResource, ResourceRole

Description

A type of a human being used to define the characteristics that need to be described for ActualPersons (e.g. properties such as address, telephone number, nationality, etc).

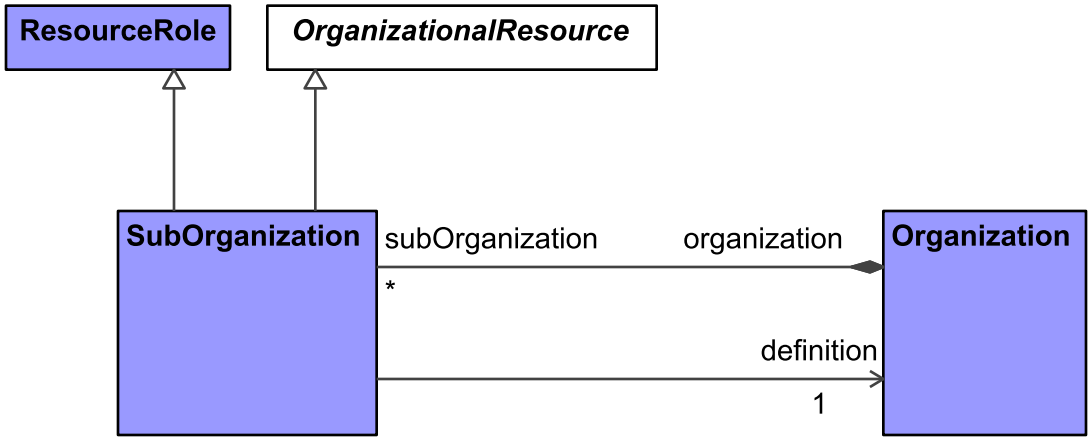

Figure 9:152 - SubOrganization

9.1.6.3 Domain MetaModel::Personnel::Connectivity

Command

Package: Connectivity
isAbstract: No
Generalization: ResourceExchange

Description

A type of ResourceExchange that asserts that one OrganizationalResource commands another.

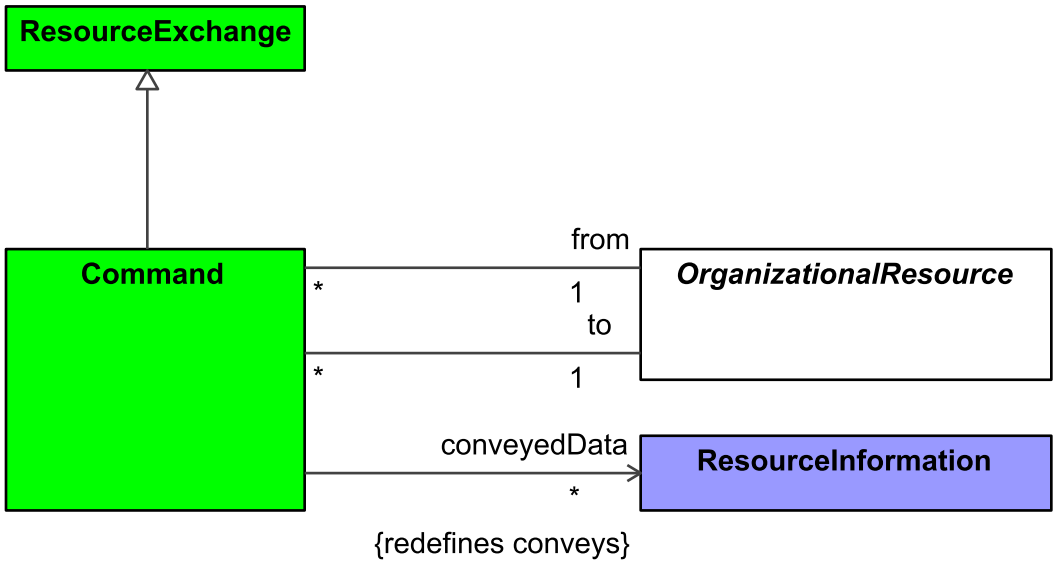

Figure 9:153 - Command

Control

Package: Connectivity
isAbstract: No
Generalization: ResourceExchange

Description

A type of ResourceExchange that asserts that one PhysicalResource controls another PhysicalResource (i.e. the driver of a vehicle controlling the vehicle speed or direction).

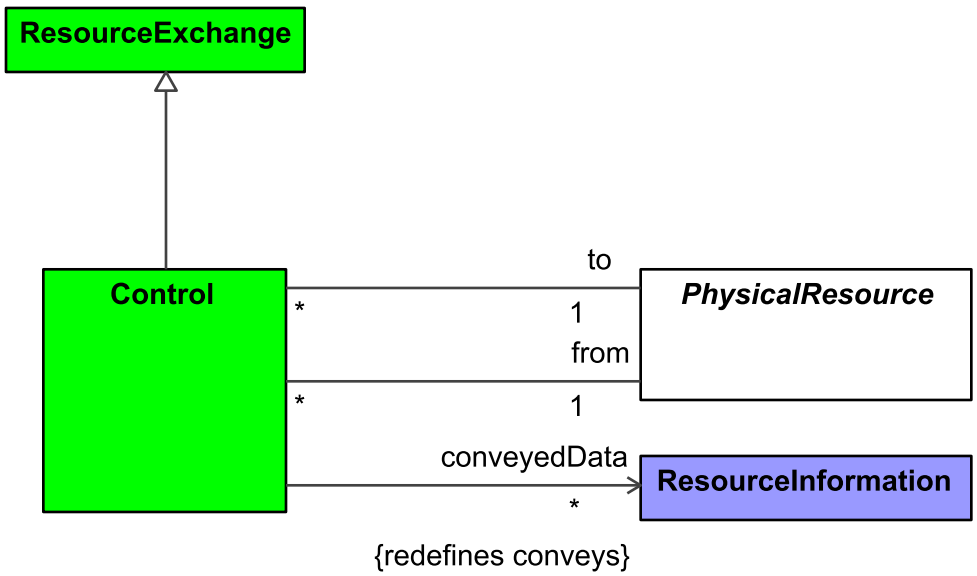

Figure 9:154 - Control

9.1.6.4 Domain MetaModel::Personnel::Sequences

ResourceInteractionScenario

Package: Sequences
isAbstract: No
Generalization: InteractionScenario

Description

A specification of the interactions between ResourcePerformers in a ResourceArchitecture.

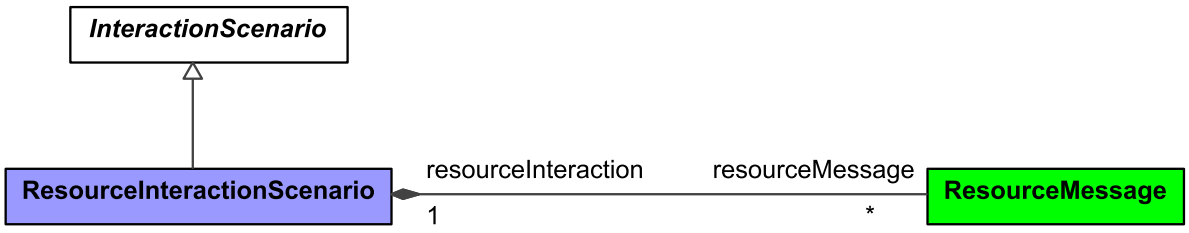

Figure 9:155 - ResourceInteractionScenario

9.1.6.5 Domain MetaModel::Personnel::Constraints

Competence

Package: Constraints
isAbstract: No
Generalization: PropertySet, SubjectOfForecast

Description

A specific set of abilities defined by knowledge, skills and aptitude.

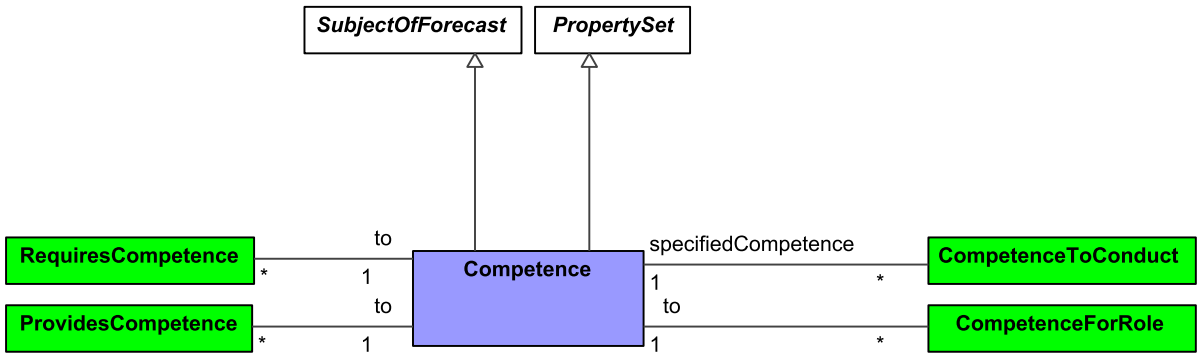

Figure 9:156 - Competence

CompetenceForRole

Package: Constraints
isAbstract: No
Generalization: MeasurableElement

Description

A tuple used to associate an organizational role with a specific set of required competencies.

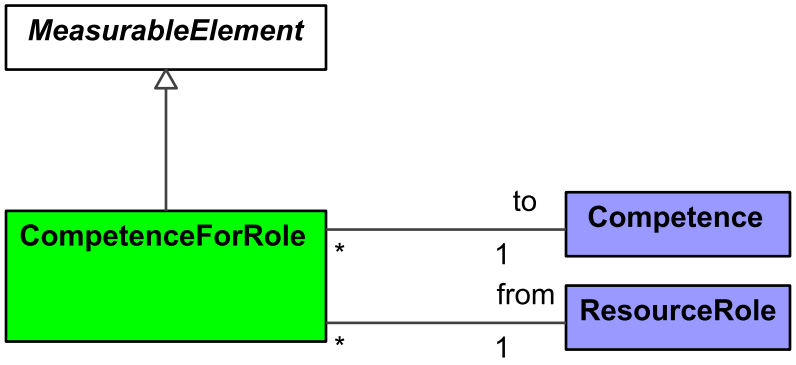

Figure 9:157 - CompetenceForRole

RequiresCompetence

Package: Constraints
isAbstract: No
Generalization: MeasurableElement

Description

A tuple that asserts that an ActualOrganizationalResource is required to have a specific set of Competencies.

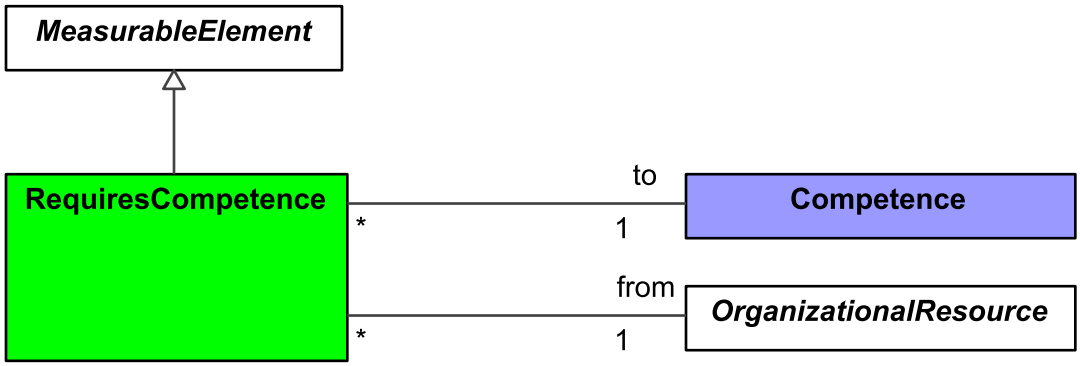

Figure 9:158 - RequiresCompetence

9.1.6.6 Domain MetaModel::Personnel::Roadmap

FillsPost

Package: Roadmap
isAbstract: No
Generalization: MeasurableElement

Description

A tuple that asserts that an ActualPerson fills an ActualPost.

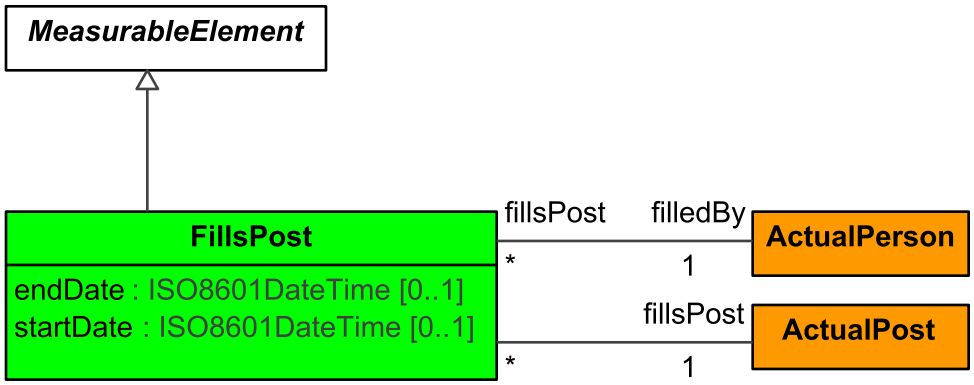

Figure 9:159 - FillsPost

Attributes

endDate : ISO8601DateTime [0..1] End time for all individual elements.
startDate : ISO8601DateTime [0..1] Start time for all individual elements.

9.1.6.7 Domain MetaModel::Personnel::Traceability

CompetenceToConduct

Package: Traceability

isAbstract: No
Generalization: MeasurableElement

Description

A tuple used to associate a Function with a specific set of Competencies needed to conduct the Function.

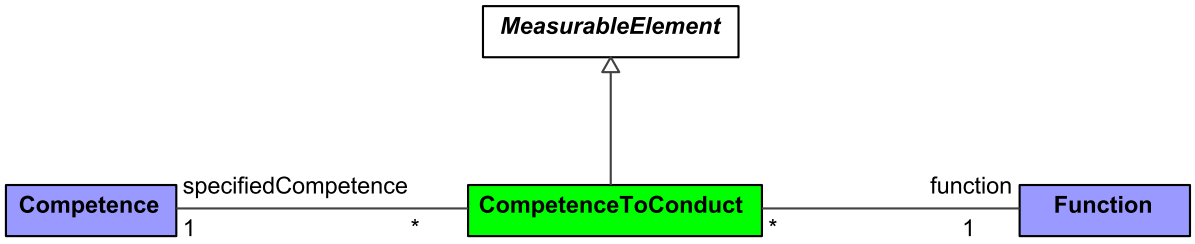

Figure 9:160 - CompetenceToConduct

9.1.7 Domain MetaModel::Resources

9.1.7.1 Domain MetaModel::Resources::Taxonomy

CapabilityConfiguration

Package: Taxonomy
isAbstract: No
Generalization: ResourceArchitecture

Description

A composite structure representing the physical and human resources (and their interactions) in an enterprise, assembled to meet a capability.

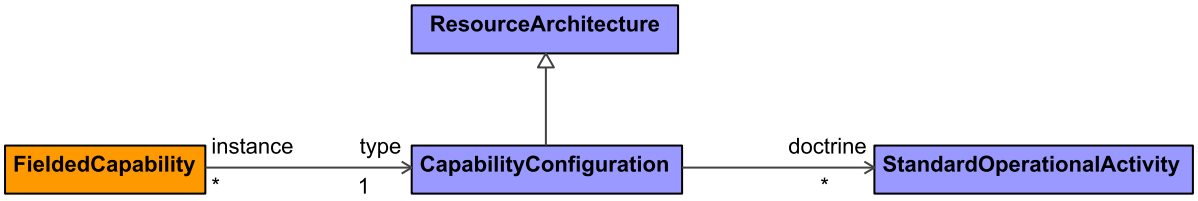

Figure 9:161 - CapabilityConfiguration

NaturalResource

Package: Taxonomy
isAbstract: No
Generalization: PhysicalResource

Description

Type of physical resource that occurs in nature such as oil, water, gas or coal.

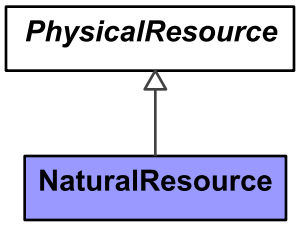

Figure 9:162 - NaturalResource

PhysicalResource

Package: Taxonomy
isAbstract: Yes
Generalization: ResourcePerformer

Description

An abstract type defining physical resources (i.e. OrganizationalResource, ResourceArtifact and NaturalResource).

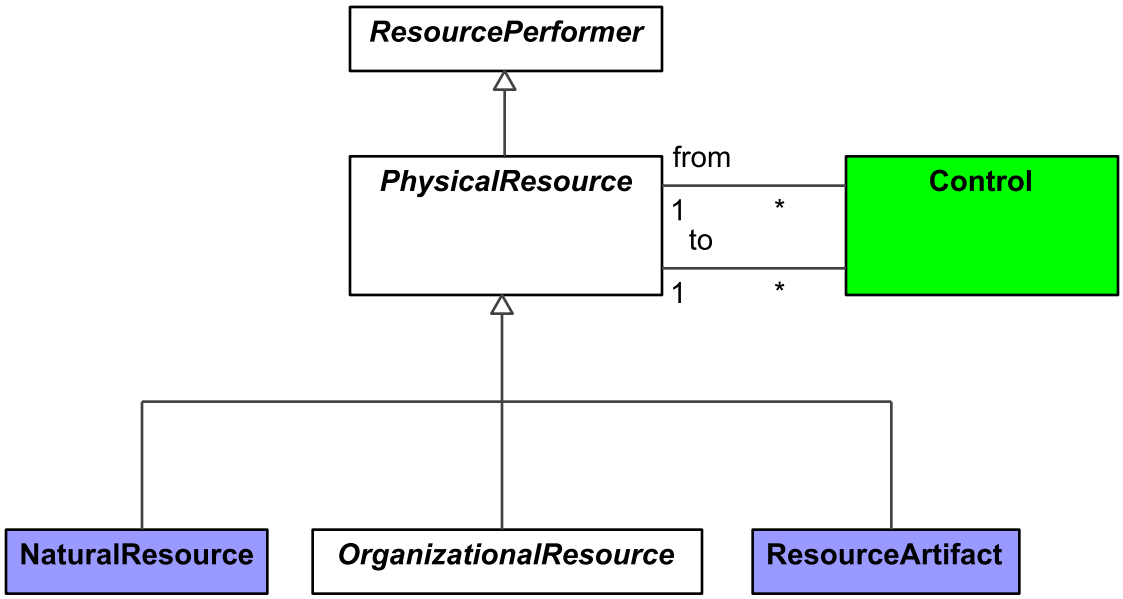

Figure 9:163 - PhysicalResource

ResourceArchitecture

Package: Taxonomy
isAbstract: No
Generalization: ResourcePerformer, Architecture

Description

A type used to denote a model of the Architecture, described from the ResourcePerformer perspective.

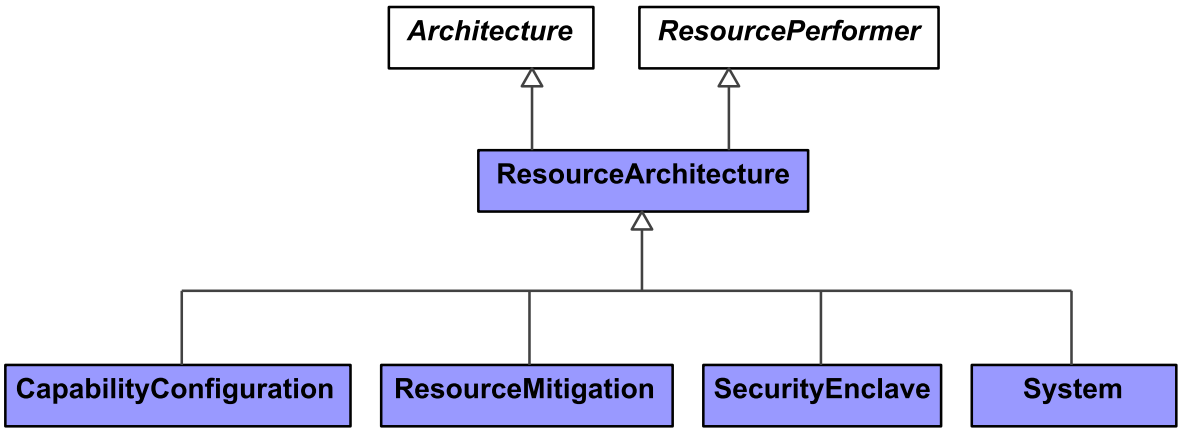

Figure 9:164 - ResourceArchitecture

ResourceArtifact

Package: Taxonomy
isAbstract: No
Generalization: PhysicalResource

Description

A type of man-made object that contains no human beings (i.e. satellite, radio, petrol, gasoline, etc.).

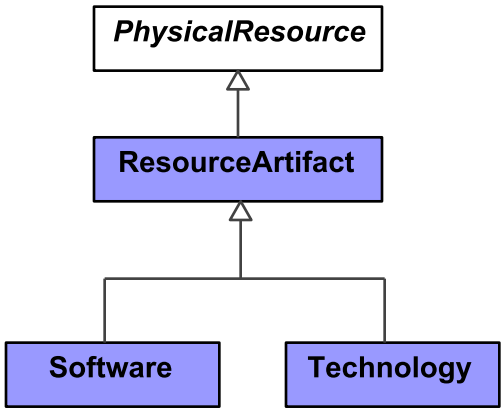

Figure 9:165 - ResourceArtifact

ResourcePerformer

Package: Taxonomy
isAbstract: Yes
Generalization: ResourceExchangeItem, SubjectOfResourceConstraint, OperationalExchangeItem, SubjectOfForecast, CapableElement, Desirer, VersionedElement, ResourceAsset, ServiceExchangeItem, StrategicExchangeItem, OpposableElement

Description

An abstract grouping of elements that can perform Functions.

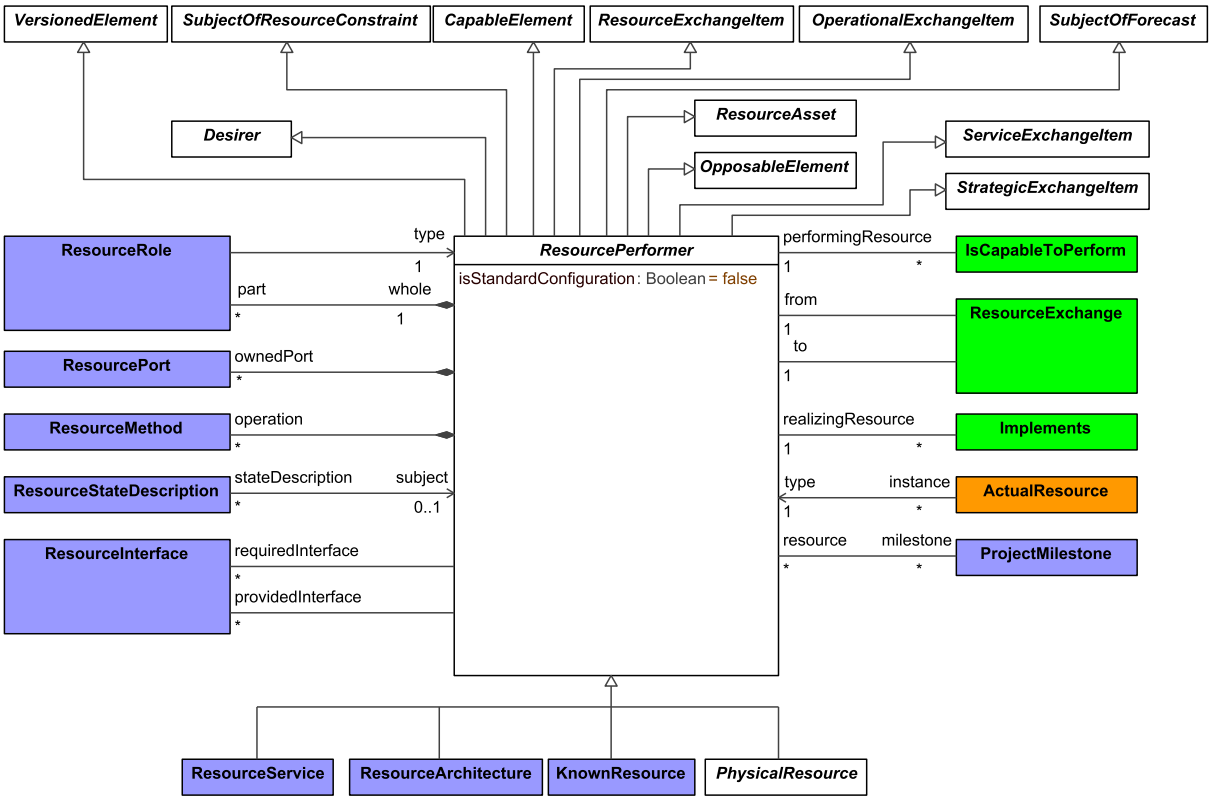

Figure 9:166 - ResourcePerformer

Attributes

isStandardConfiguration : Boolean [] Indicates if the ResourcePerformer is StandardConfiguration, default=false.

ResourceService

Package: Taxonomy
isAbstract: No
Generalization: ResourcePerformer

Description

A services that a ResourcePerformer provides to support higher level Services or OperationalActivities. Employee provisioning, backup and recovery, storage, self-service help desk are examples of ResourceServices.

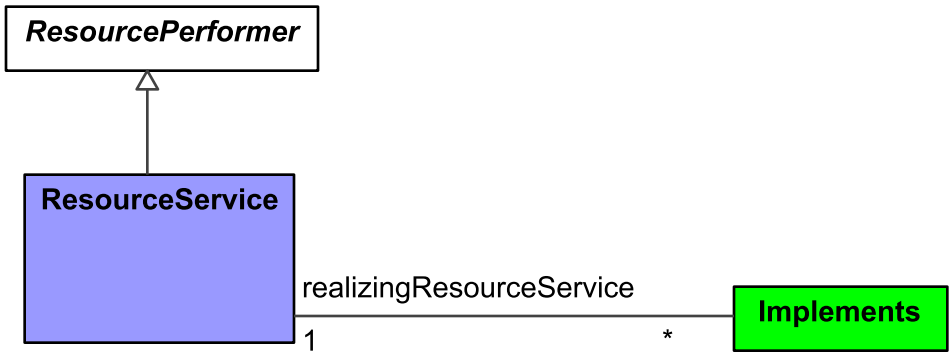

Figure 9:167 - ResourceService

Software

Package: Taxonomy
isAbstract: No
Generalization: ResourceArtifact

Description

A sub-type of ResourceArtifact that specifies an executable computer program.

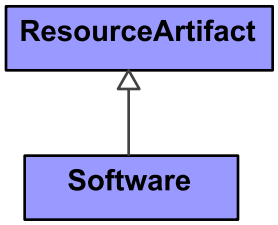

Figure 9:168 - Software

System

Package: Taxonomy
isAbstract: No
Generalization: ResourceArchitecture

Description

An integrated set of elements, subsystems, or assemblies that accomplish a defined objective. These elements include products (hardware, software, firmware), processes, people, information, techniques, facilities, services, and other support elements (INCOSE SE Handbook V4, 2015).

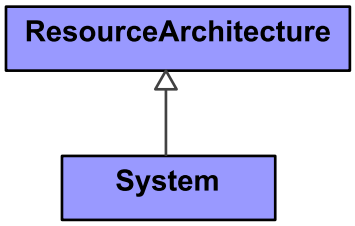

Figure 9:169 - System

9.1.7.2 Domain MetaModel::Resources::Structure

ResourceMethod

Package: Structure
isAbstract: No
Generalization: ProcessOperation

Description

A behavioral feature of a ResourcePerformer whose behavior is specified in a Function.

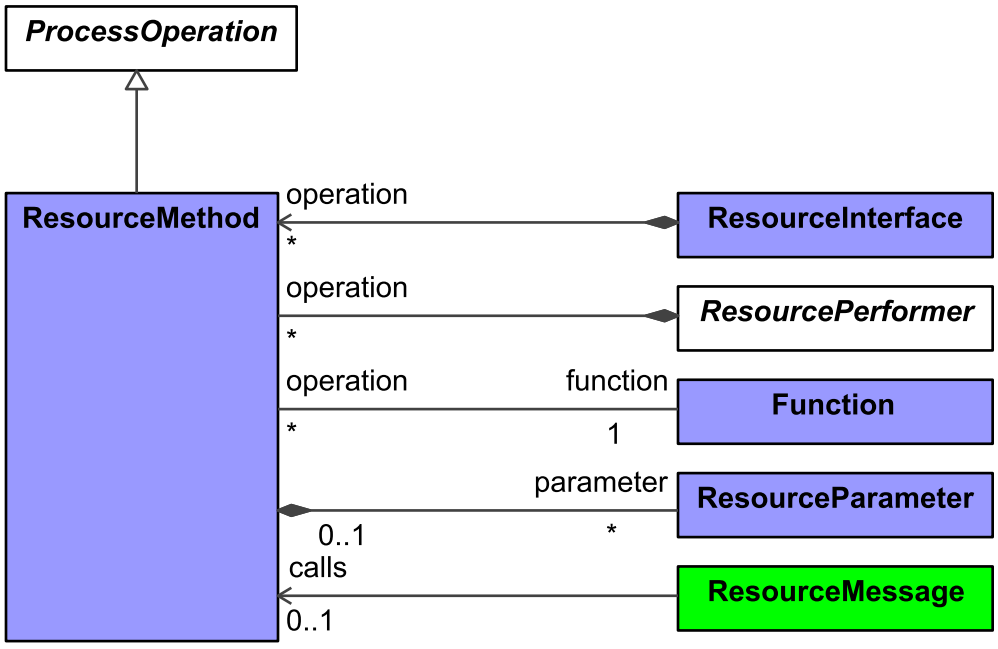

Figure 9:170 - ResourceMethod

ResourceParameter

Package: Structure
isAbstract: No
Generalization: ProcessParameter

Description

A type that represents inputs and outputs of a Function. It is typed by a ResourceInteractionItem.

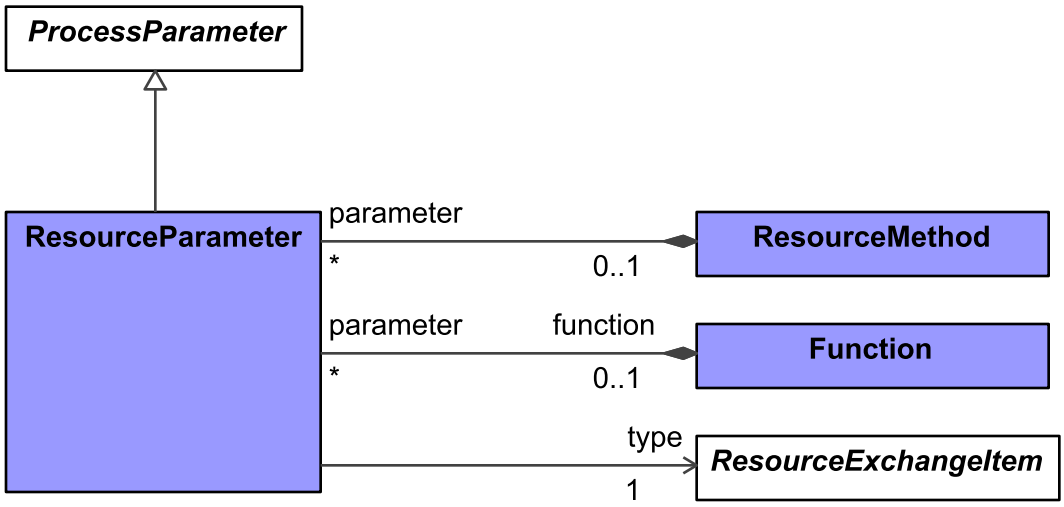

Figure 9:171 - ResourceParameter

ResourcePort

Package: Structure
isAbstract: No
Generalization: ProtocolImplementation, MeasurableElement

Description

An interaction point for a ResourcePerformer through which it can interact with the outside environment and which is defined by a ResourceInterface.

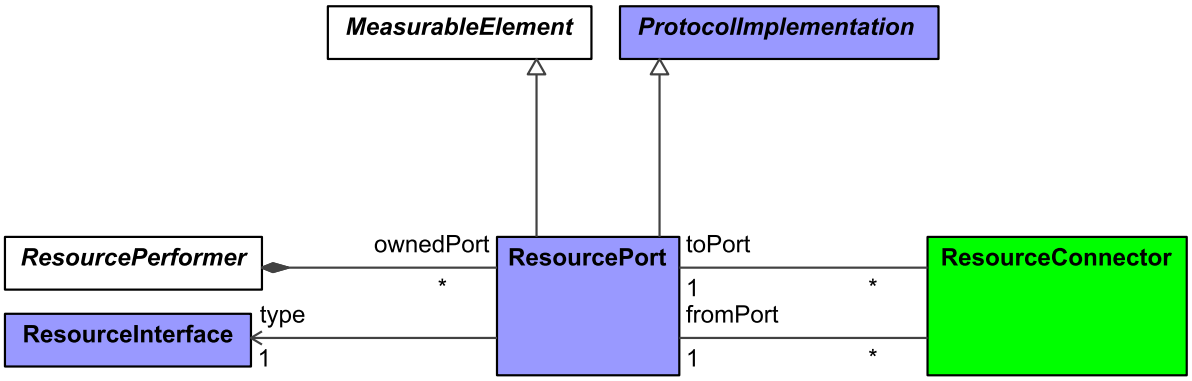

Figure 9:172 - ResourcePort

ResourceRole

Package: Structure
isAbstract: No
Generalization: SubjectOfResourceConstraint, LocationHolder, AssetRole, InteractionRole, OpposableElement

Description

Usage of a ResourcePerformer in the context of another ResourcePerformer. Creates a whole-part relationship.

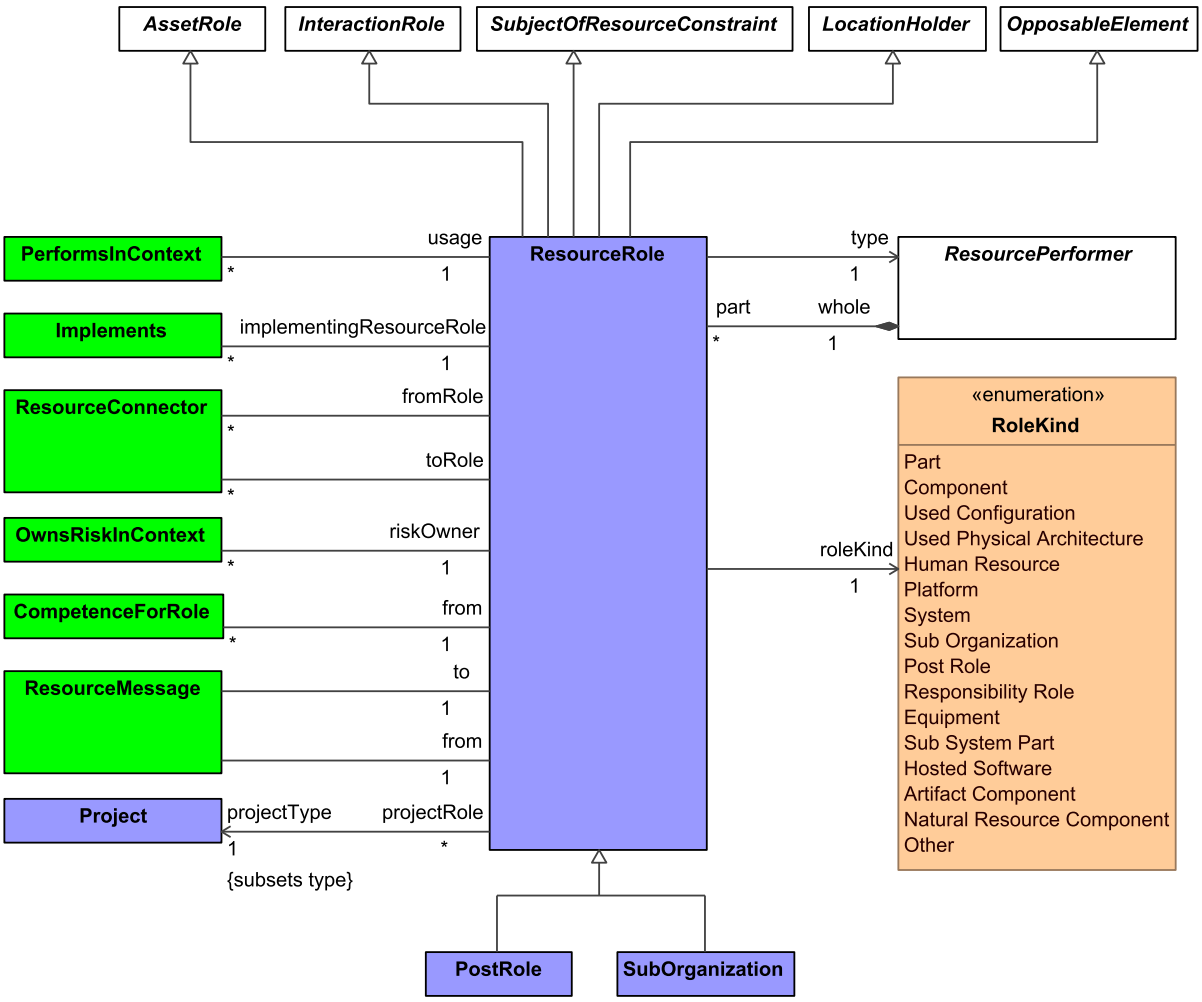

Figure 9:173 - ResourceRole

9.1.7.3 Domain MetaModel::Resources::Connectivity

ResourceConnector

Package: Connectivity
isAbstract: No
Generalization: ProtocolImplementation, AssetRole

Description

A channel for exchange between two ResourceRoles.

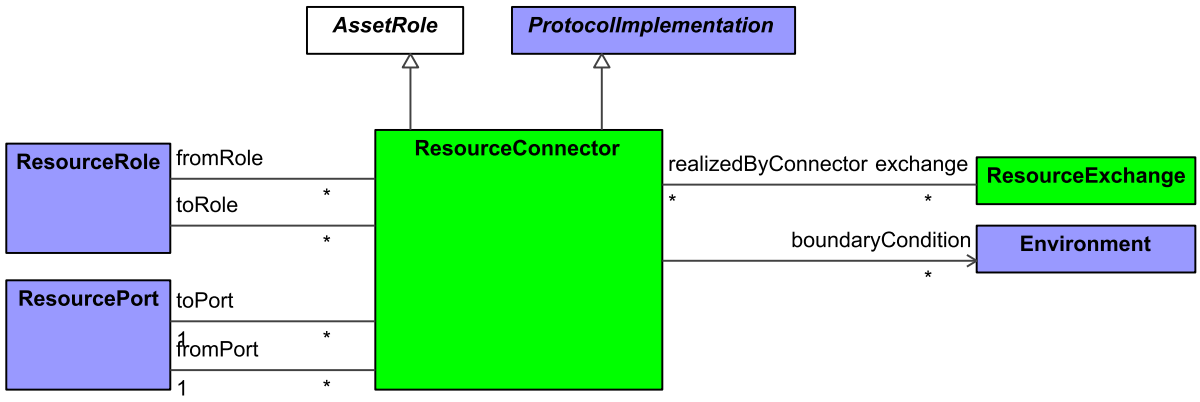

Figure 9:174 - ResourceConnector

ResourceExchange

Package: Connectivity
isAbstract: No
Generalization: Exchange

Description

Asserts that a flow can exist between ResourcePerformers (i.e. flows of data, people, material, or energy).

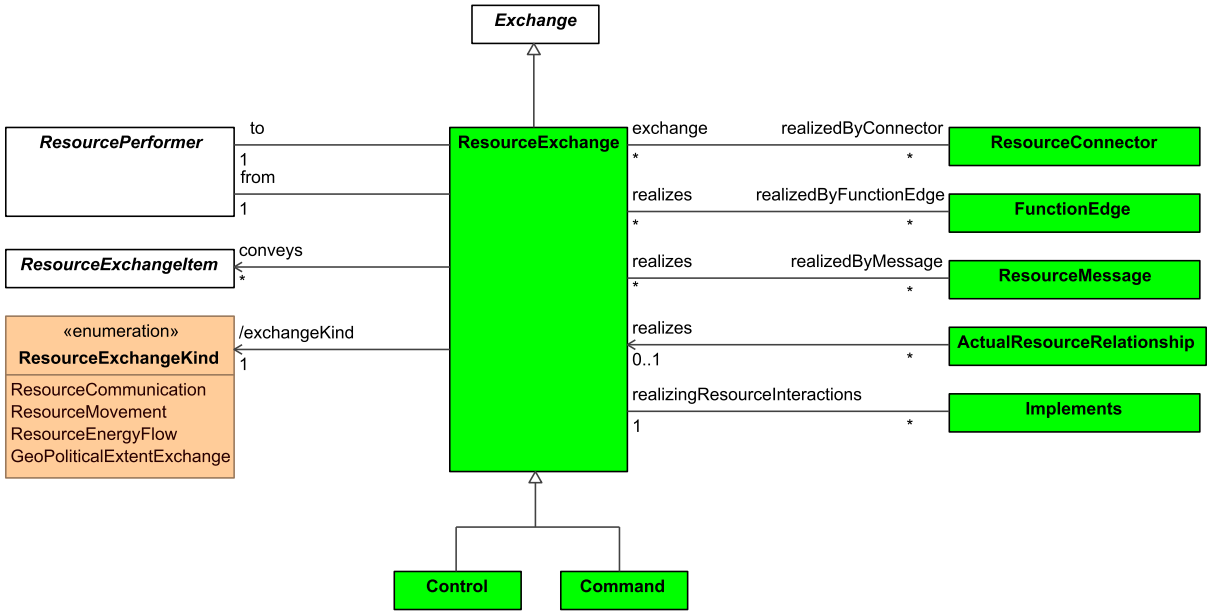

Figure 9:175 - ResourceExchange

ResourceExchangeItem

Package: Connectivity
isAbstract: Yes
Generalization: Resource, SubjectOfSecurityConstraint, ExchangeItem

Description

An abstract type grouping elements that defines the types of elements that can be exchanged between ResourcePerformers and conveyed by a ResourceExchange.

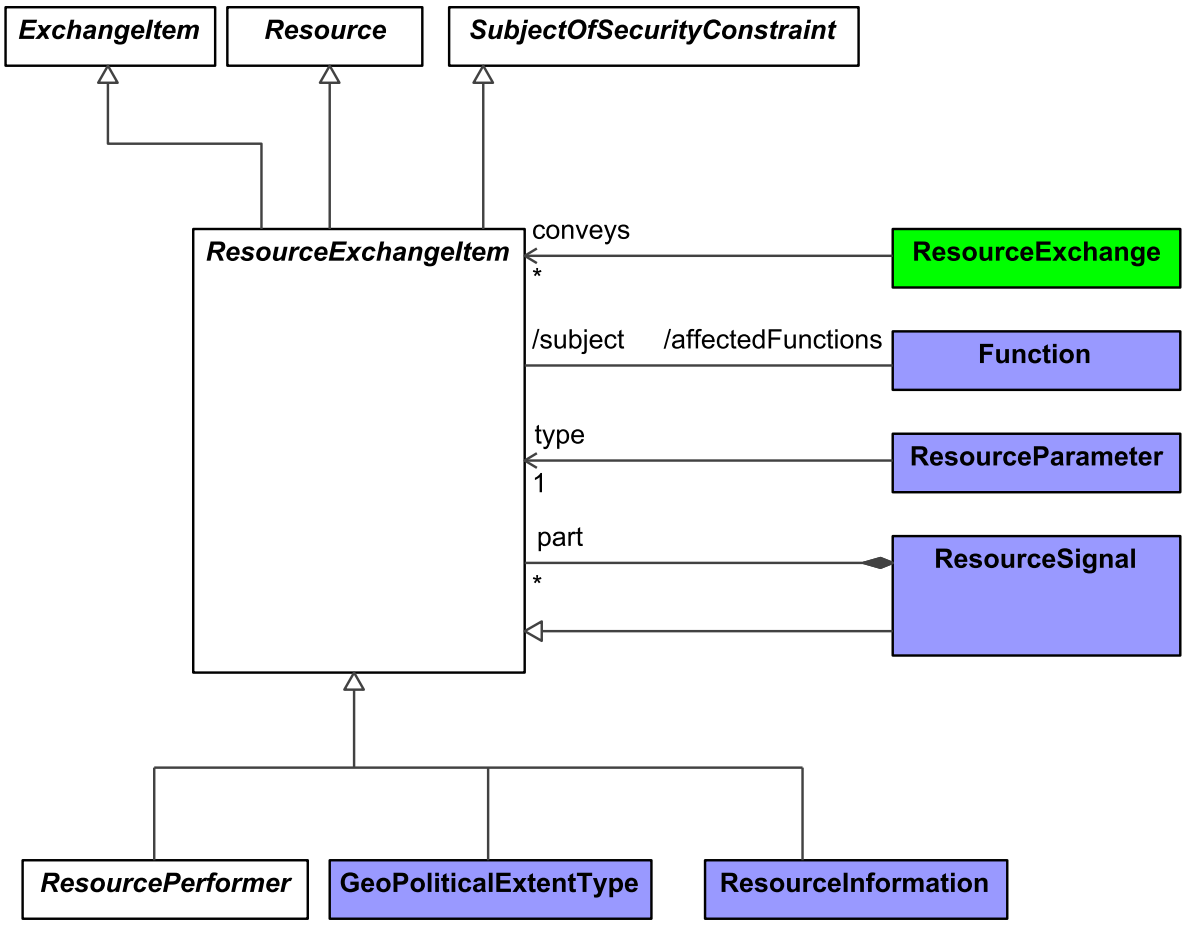

Figure 9:176 - ResourceExchangeItem

ResourceInterface

Package: Connectivity
isAbstract: No
Generalization: PropertySet

Description

A declaration that specifies a contract between the ResourcePerformers it is related to and any other ResourcePerformers it can interact with. It is also intended to be an implementation of a specification of an Interface in the Business and/or Service layer.

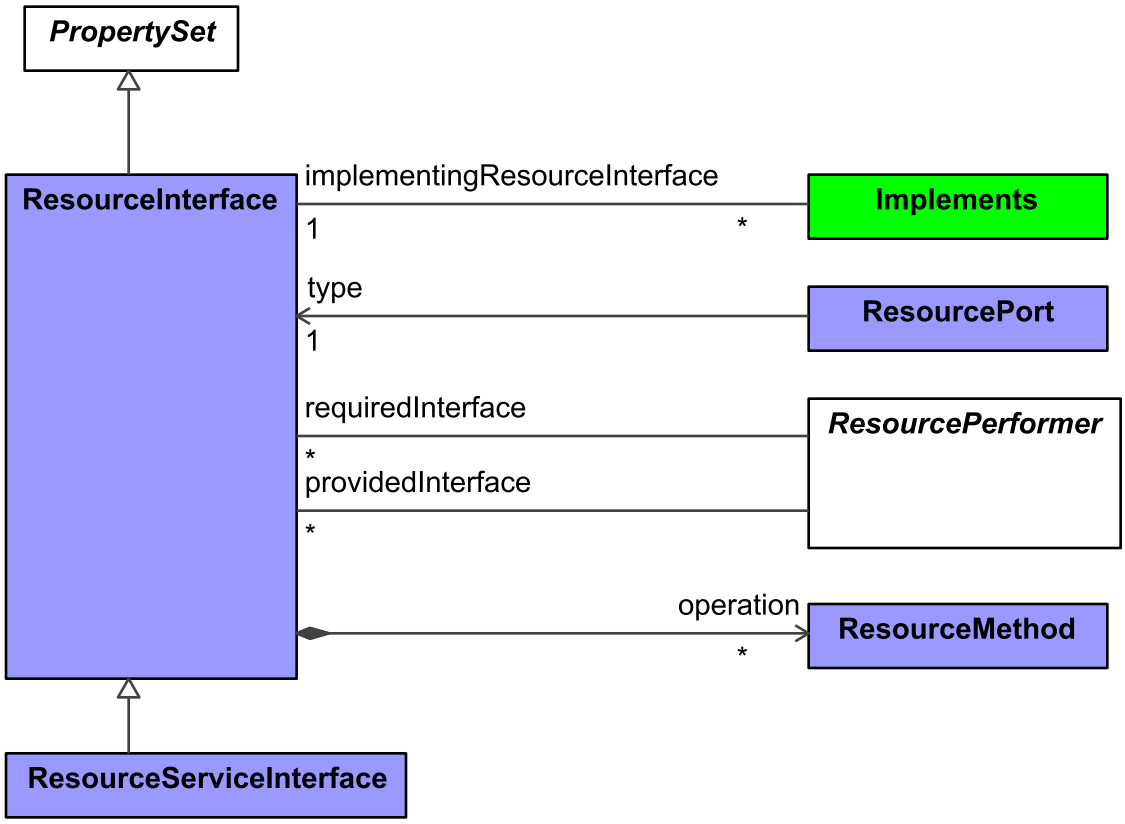

Figure 9:177 - ResourceInterface

ResourceServiceInterface

Package: Connectivity
isAbstract: No
Generalization: ResourceInterface

Description

A contract that defines the ResourceMethods and ResourceSignal receptions that the ResourceServices realize.

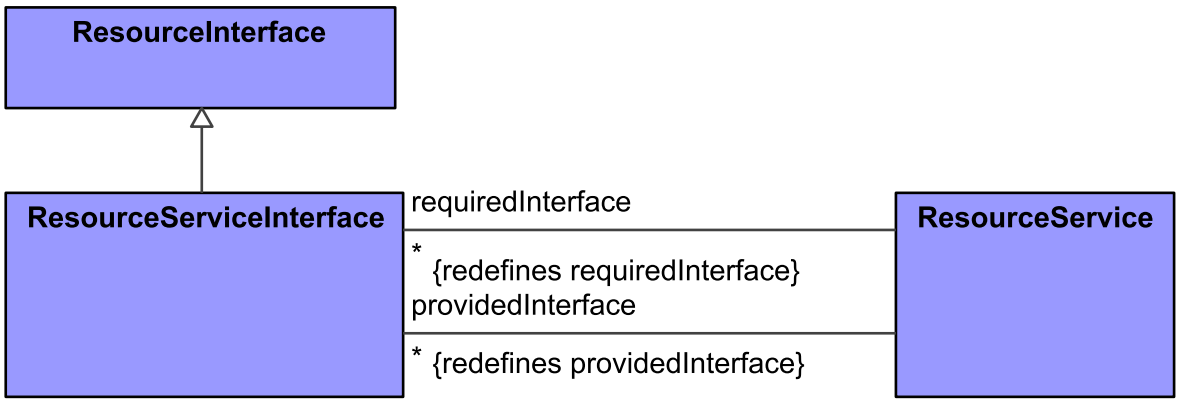

Figure 9:178 - ResourceServiceInterface

ResourceSignal

Package: Connectivity
isAbstract: No
Generalization: ResourceExchangeItem

Description

A property of an element representing something in the physical world, expressed in amounts of a unit of measure.

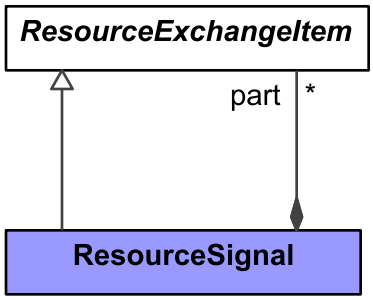

Figure 9:179 - ResourceSignal

9.1.7.4 Domain MetaModel::Resources::Processes

Function

Package: Processes
isAbstract: No
Generalization: SubjectOfResourceConstraint, Process

Description

An Activity which is specified in the context to the ResourcePerformer (human or machine) that IsCapableToPerform it.

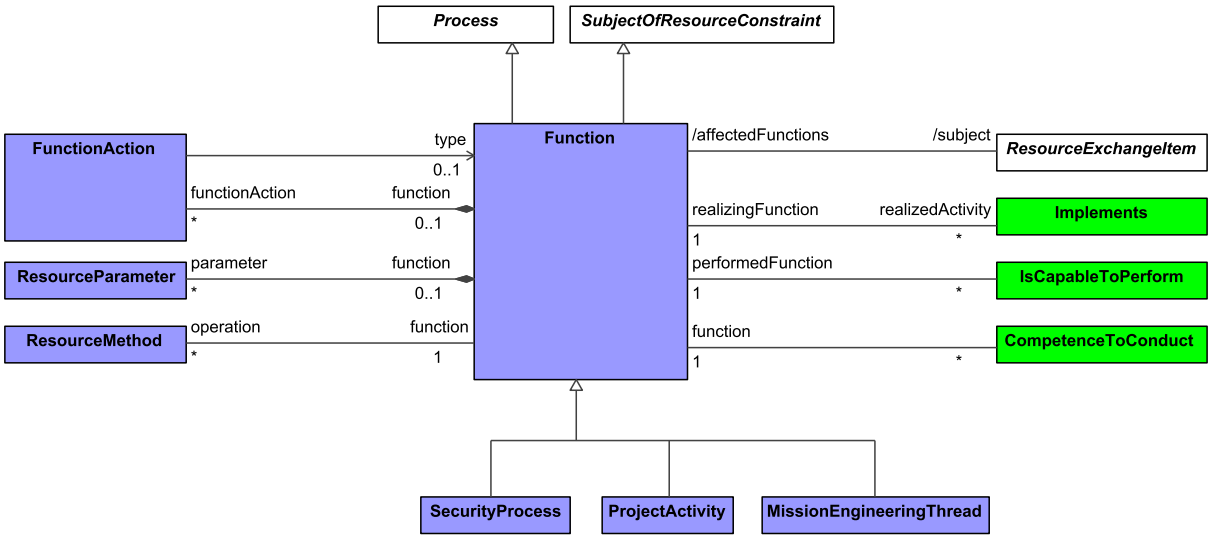

Figure 9:180 - Function

FunctionAction

Package: Processes
isAbstract: No
Generalization: ProcessUsage

Description

A call of a Function indicating that the Function is performed by a ResourceRole in a specific context.

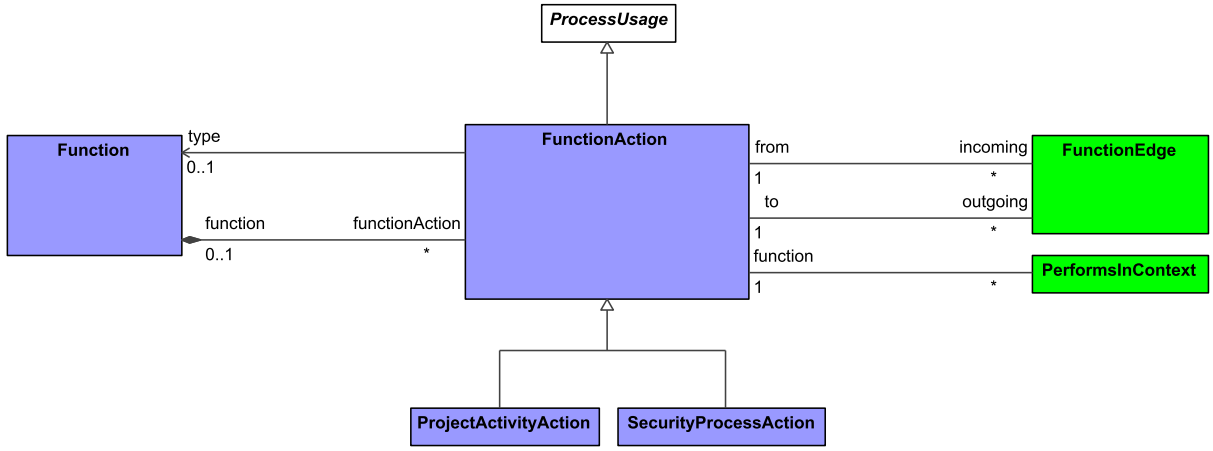

Figure 9:181 - FunctionAction

FunctionEdge

Package: Processes
isAbstract: No
Generalization: ProcessEdge

Description

A tuple that shows the flow of Resources (objects/data) between FunctionActions.

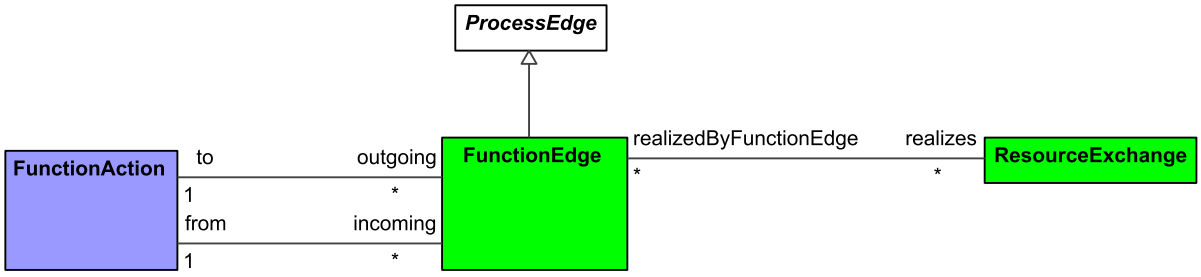

Figure 9:182 - FunctionEdge

MissionEngineeringThread

Package: Processes
isAbstract: No
Generalization: Function

Description

A sequence of end-to-end mission-related functions, actions, and events presented as a series of steps that implements a mission thread.

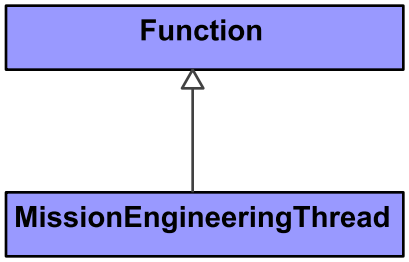

Figure 9:183 - MissionEngineeringThread

9.1.7.5 Domain MetaModel::Resources::States

ResourceStateDescription

Package: States
isAbstract: No
Generalization: MeasurableElement, StateDescription

Description

A state machine describing the behavior of a ResourcePerformer, depicting how the ResourcePerformer responds to various events and the actions.

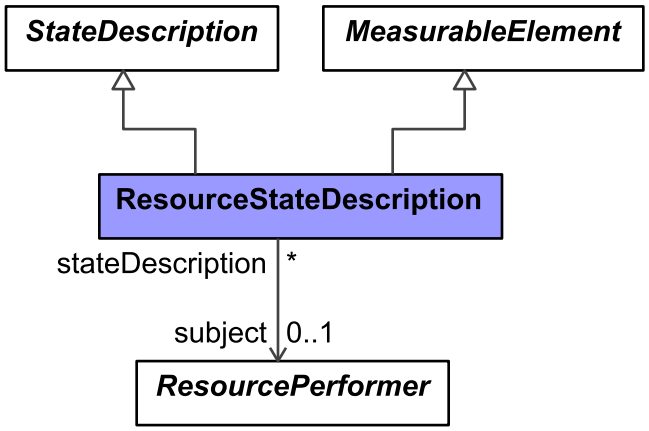

Figure 9:184 - ResourceStateDescription

9.1.7.6 Domain MetaModel::Resources::Sequences

ResourceMessage

Package: Sequences
isAbstract: No
Generalization: InteractionMessage

Description

Message for use in a Resource Event-Trace which carries any of the subtypes of ResourceExchange.

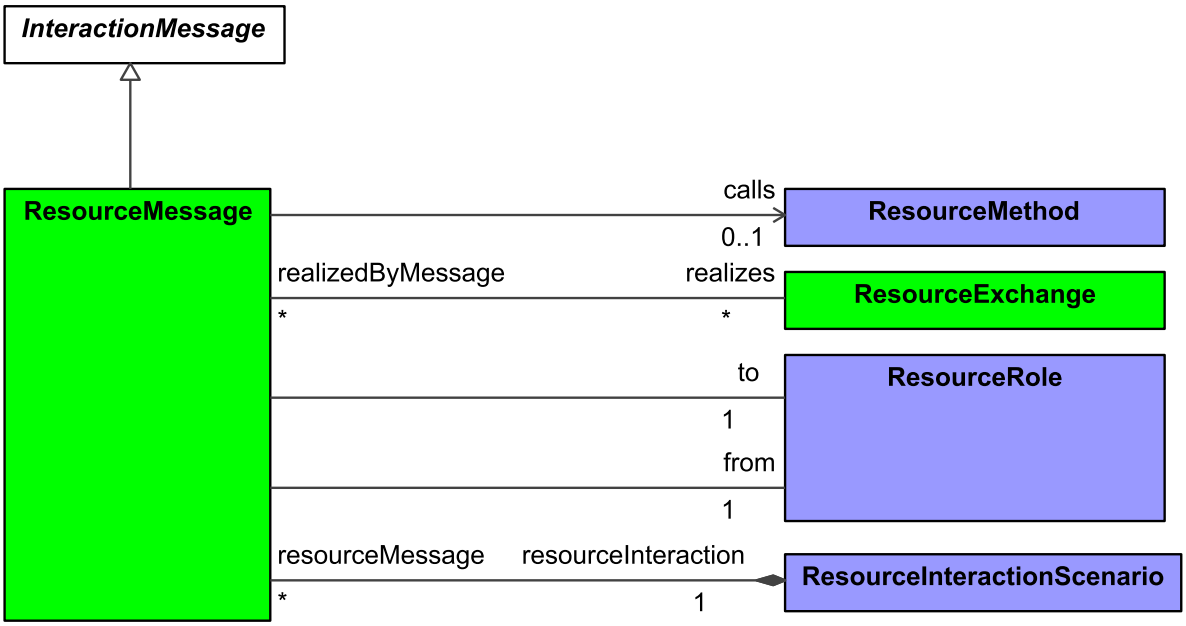

Figure 9:185 - ResourceMessage

9.1.7.7 Domain MetaModel::Resources::Information

ResourceInformation

Package: Information
isAbstract: No
Generalization: SubjectOfResourceConstraint, ResourceAsset, ResourceExchangeItem

Description

A formalized representation of information that is managed by or exchanged between systems.

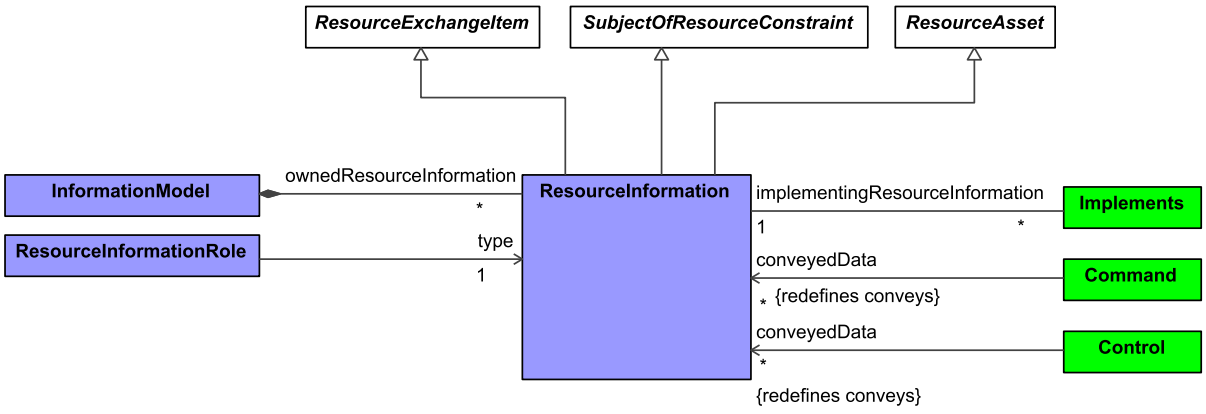

Figure 9:186 - ResourceInformation

9.1.7.8 Domain MetaModel::Resources::Constraints

ResourceConstraint

Package: Constraints

isAbstract: No
Generalization: Rule

Description

A rule governing the structural or functional aspects of an implementation.

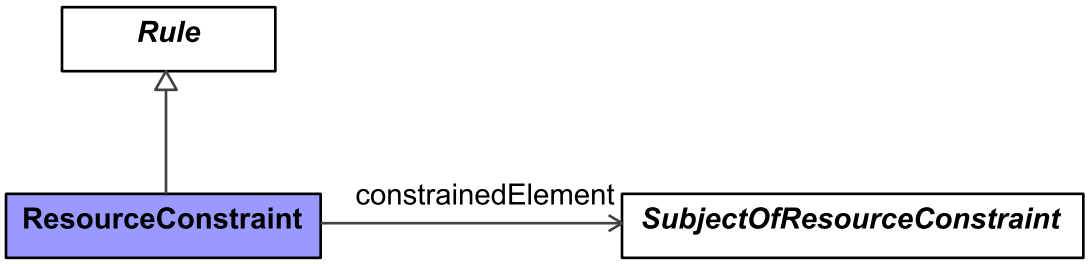

Figure 9:187 - ResourceConstraint

SubjectOfResourceConstraint

Package: Constraints
isAbstract: Yes
Generalization: UAFElement

Description

An abstract type grouping elements that can be the subject of a ResourceConstraint.

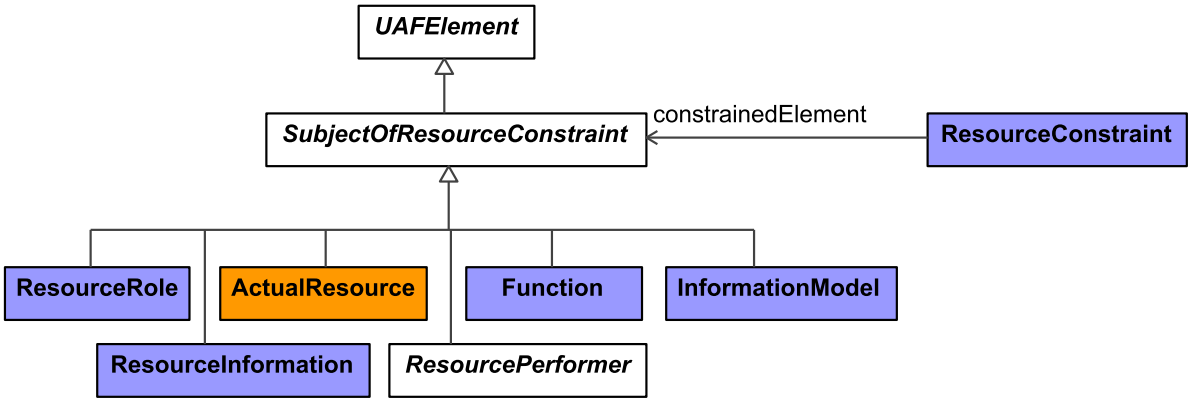

Figure 9:188 - SubjectOfResourceConstraint

9.1.7.9 Domain MetaModel::Resources::Roadmap

Forecast

Package: Roadmap
isAbstract: No
Generalization: MeasurableElement

Description

A dependency relationship that specifies a transition from one Resource Performer, Standard, Competence to another future one. It is related to an ActualStrategicPhase to give it a temporal context.

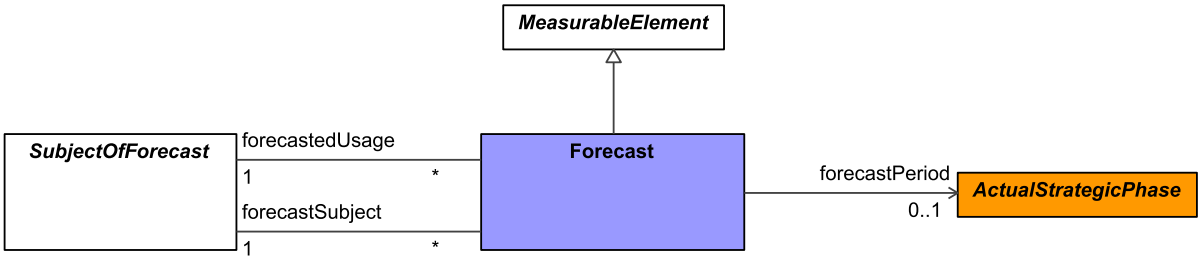

Figure 9:189 - Forecast

SubjectOfForecast

Package: Roadmap
isAbstract: Yes
Generalization: UAFElement

Description

An abstract type grouping elements that can be the subject of a Forecast.

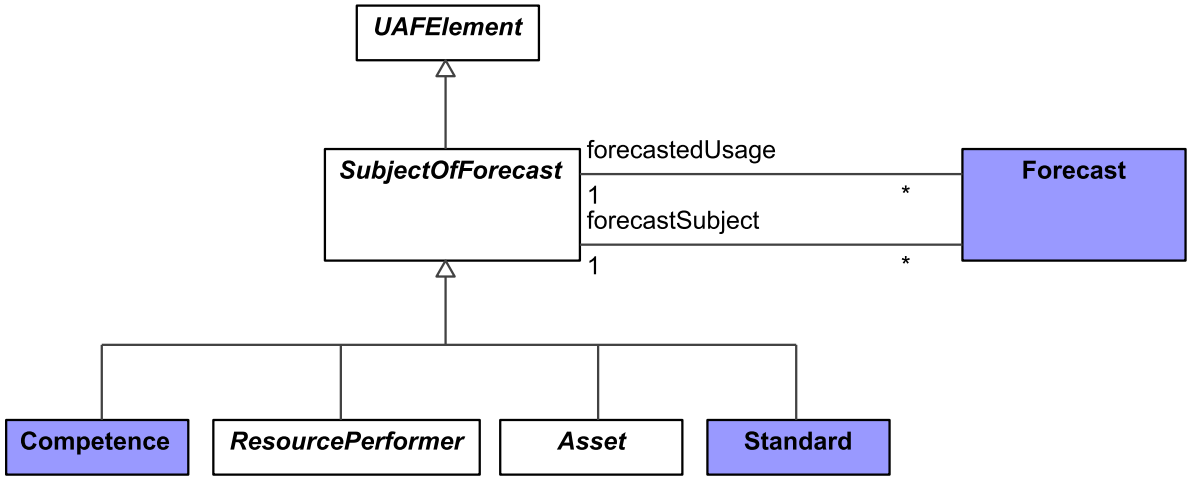

Figure 9:190 - SubjectOfForecast

Technology

Package: Roadmap
isAbstract: No
Generalization: ResourceArtifact

Description

A sub type of ResourceArtifact that indicates a technology domain, i.e. nuclear, mechanical, electronic, mobile telephony etc.

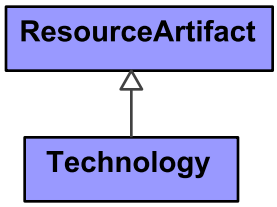

Figure 9:191 - Technology

VersionedElement

Package: Roadmap
isAbstract: Yes
Generalization: UAFElement

Description

An abstract type grouping ResourcePerformer and Service that allows VersionOfConfiguration to be related to ActualProjectMilestones.

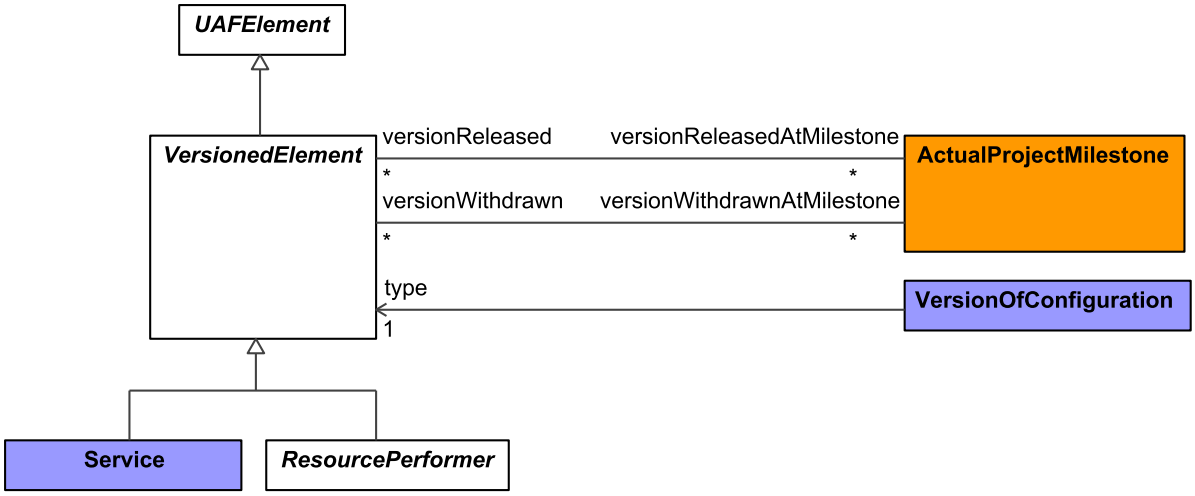

Figure 9:192 - VersionedElement

VersionOfConfiguration

Package: Roadmap
isAbstract: No
Generalization: MeasurableElement

Description

A property of a WholeLifeConfiguration, used in version control of a VersionedElement. It asserts that a VersionedElement is a version of a WholeLifeConfiguration.

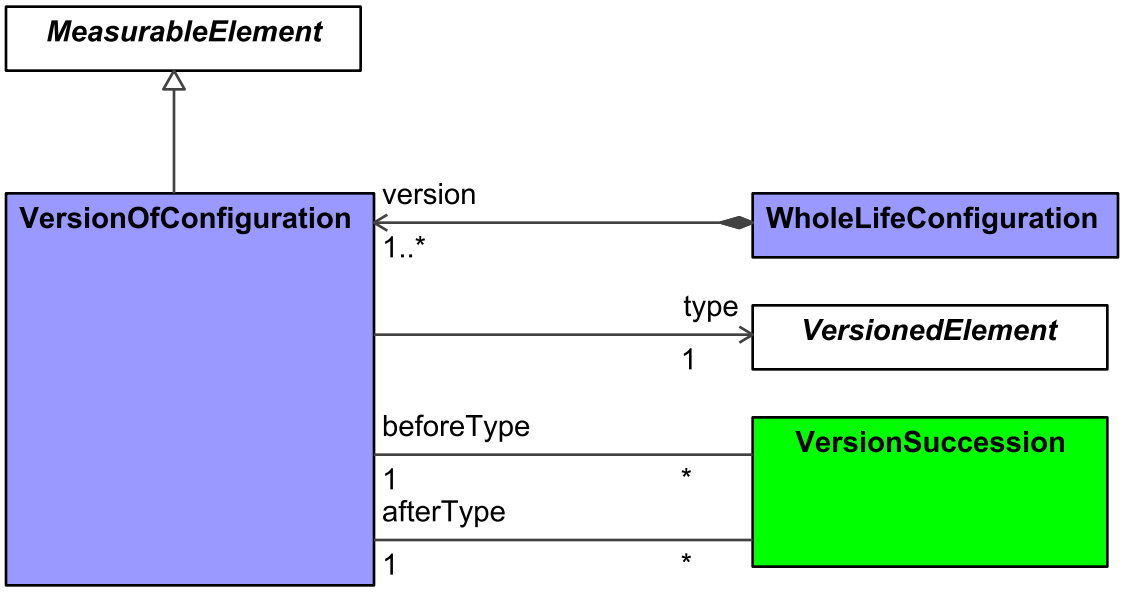

Figure 9:193 - VersionOfConfiguration

VersionSuccession

Package: Roadmap
isAbstract: No
Generalization: MeasurableElement

Description

A tuple between two VersionOfConfigurations that denotes that one VersionOfConfiguration follows from another.

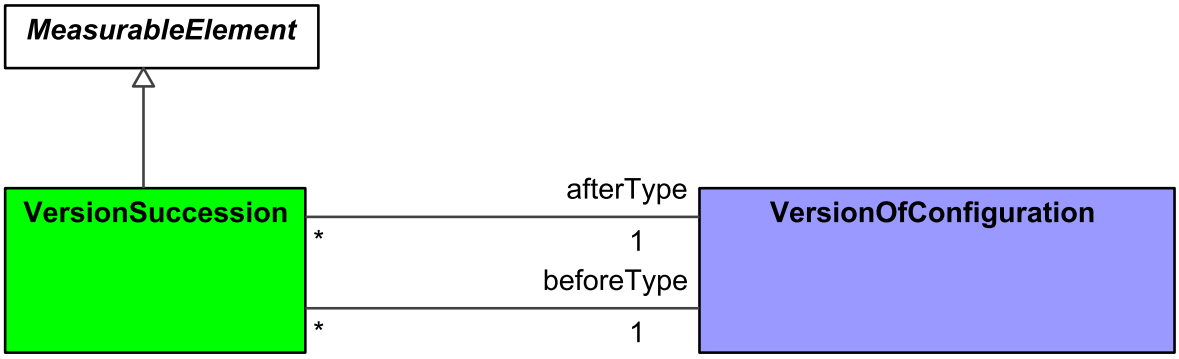

Figure 9:194 - VersionSuccession

WholeLifeConfiguration

Package: Roadmap
isAbstract: No
Generalization: PropertySet

Description

A set of VersionedElements, e.g. Services for a service provider or ResourcePerformers deployed for implementation.

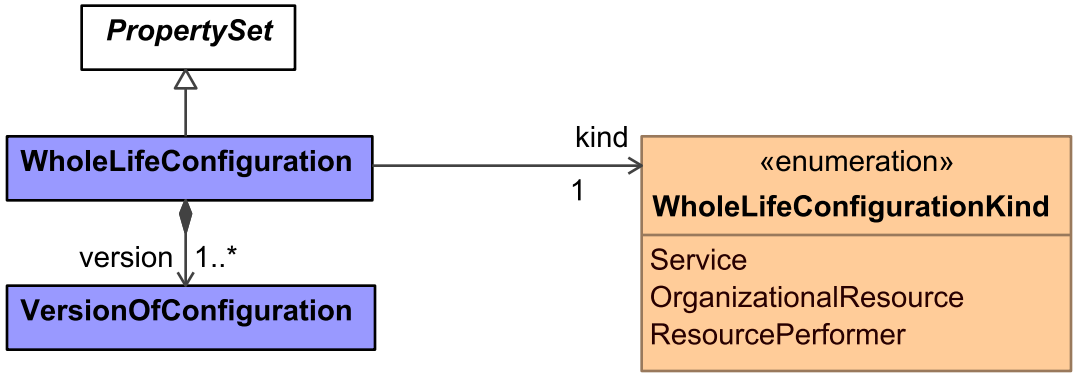

Figure 9:195 - WholeLifeConfiguration

9.1.7.10 Domain MetaModel::Resources::Traceability

ProtocolImplementation

Package: Traceability
isAbstract: Yes
Generalization: UAFElement

Description

An abstract type grouping architectural elements that can implement Protocols.

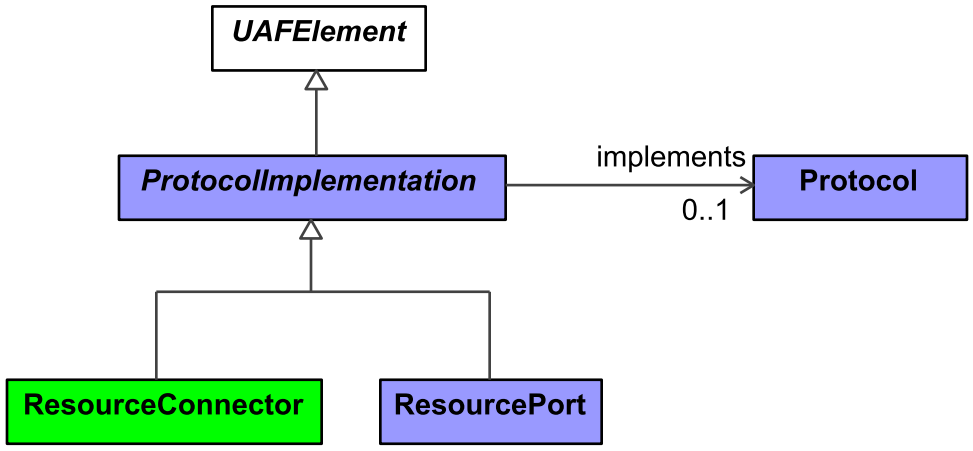

Figure 9:196 - ProtocolImplementation

9.1.8 Domain MetaModel::Security

Stakeholders: Security Architects, Security Engineers. Systems Engineers, Operational Architects.

Concerns: addresses the security constraints and information assurance attributes that exist on exchanges between resources and OperationalPerformers

Definition: illustrates the security assets, security constraints, security controls, families, and measures required to address specific security concerns.

9.1.8.1 Domain MetaModel::Security::Motivation

EnhancedSecurityControl

Package: Motivation
isAbstract: No
Generalization: SecurityControl

Description

Statement of security capability to: (i) build in additional but related, functionality to a basic control; and/or (ii)increase the strength of a basic control.

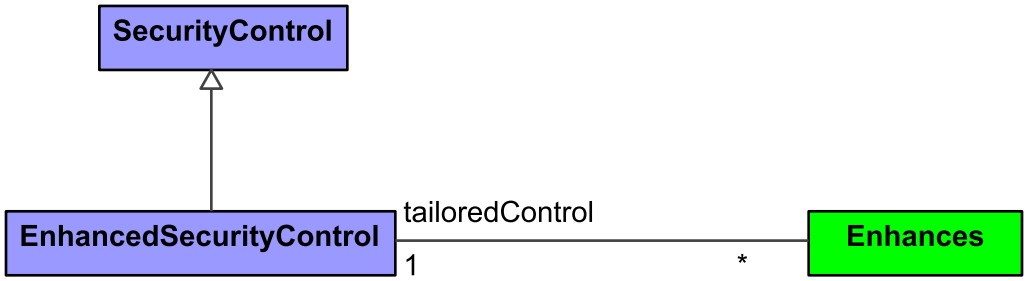

Figure 9:197 - EnhancedSecurityControl

Enhances

Package: Motivation
isAbstract: No
Generalization: MeasurableElement

Description

A tuple relating the EnhancedSecurityControl to a SecurityControl.

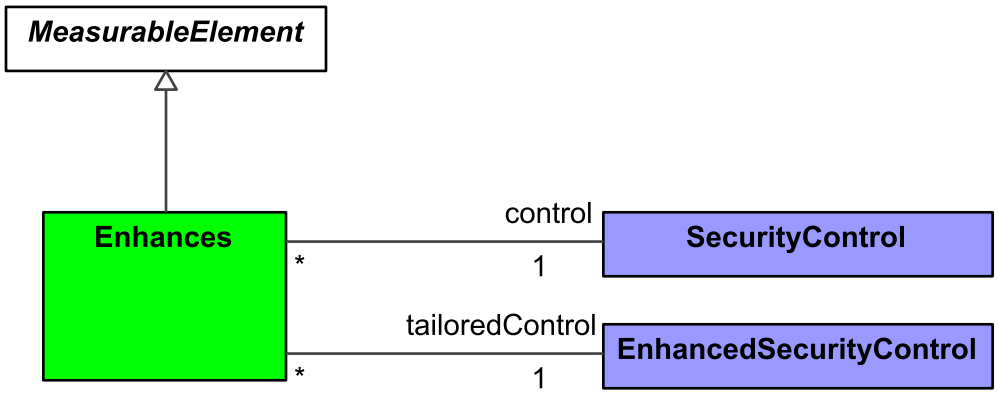

Figure 9:198 - Enhances

Protects

Package: Motivation
isAbstract: No
Generalization: MeasurableElement

Description

A tuple that asserts that a SecurityControl is required to protect an Asset.

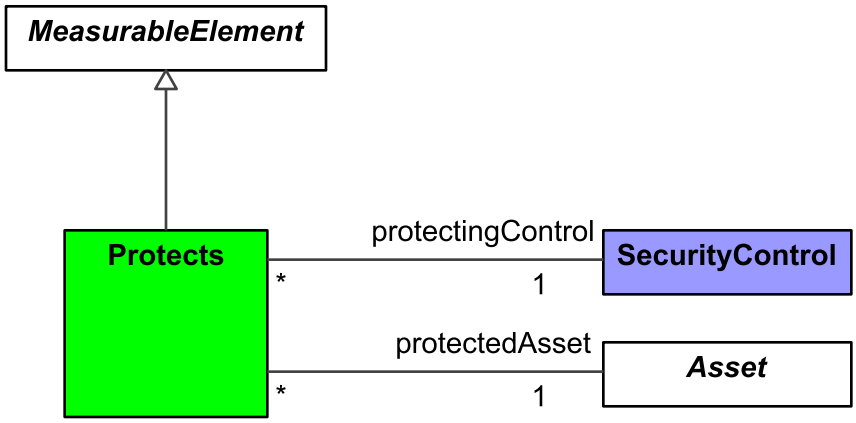

Figure 9:199 - Protects

ProtectsInContext

Package: Motivation
isAbstract: No
Generalization: MeasurableElement

Description

A tuple that relates a SecurityControlAction to a OperationalRole, or a ResourceRole. It indicates that SecurityControl is required to protect an Asset in a specific context or configuration.

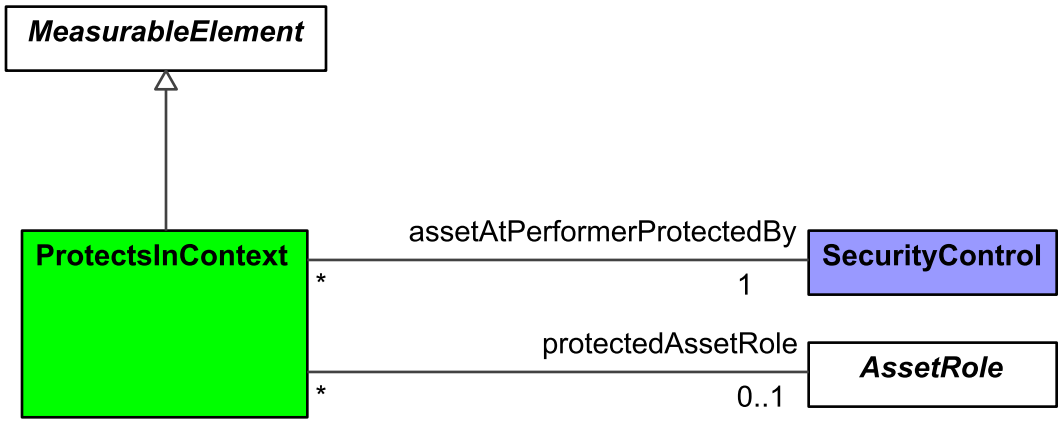

Figure 9:200 - ProtectsInContext

SecurityControl

Package: Motivation
isAbstract: No
Generalization: MeasurableElement

Description

The management, operational, and technical control (i.e., safeguard or countermeasure) prescribed for an information system to protect the confidentiality, integrity, and availability of the system and its information [NIST SP 800-53].

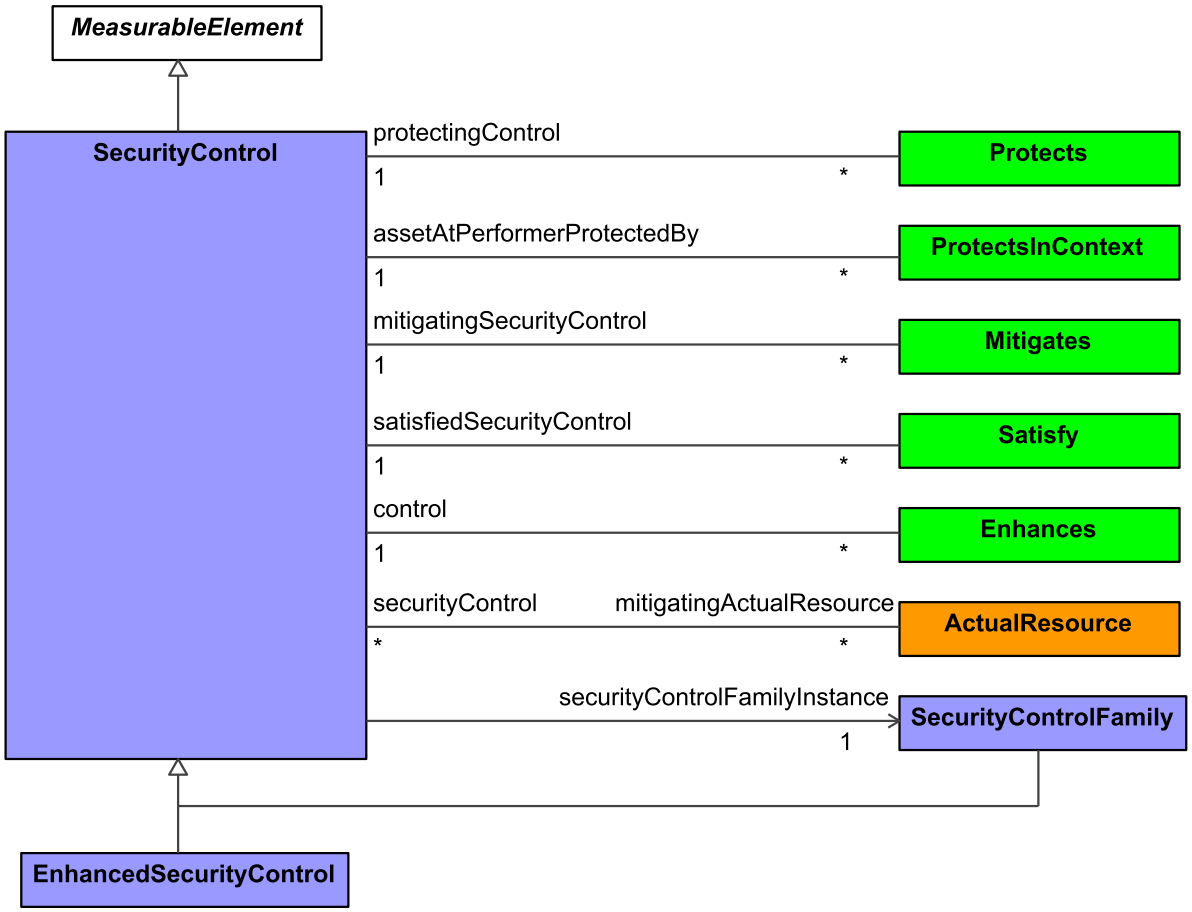

Figure 9:201 - SecurityControl

SecurityControlFamily

Package: Motivation
isAbstract: No
Generalization: SecurityControl

Description

An element that organizes security controls into a family. Each Security Control Family contains security controls related to the general security topic of the family.

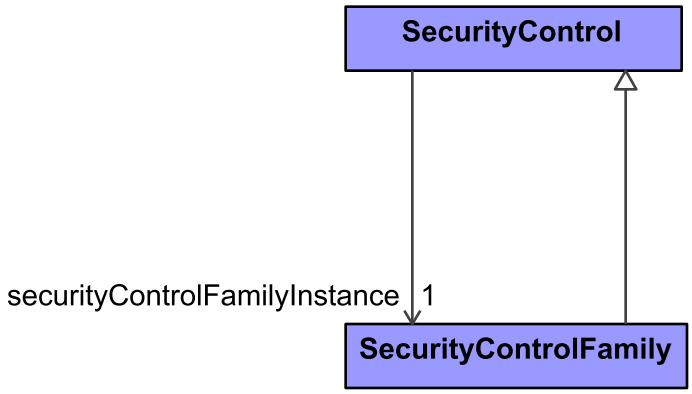

Figure 9:202 - SecurityControlFamily

9.1.8.2 Domain MetaModel::Security::Taxonomy

Asset

Package: Taxonomy
isAbstract: Yes
Generalization: SubjectOfForecast, ConceptItem, PropertySet, SubjectOfSecurityConstraint, AffectableElement

Description

An abstract element that indicates the types of elements that can be affected by Risk. Asset as applied to Security views is an abstract element that indicates the types of elements that can be considered as a subject for security analysis

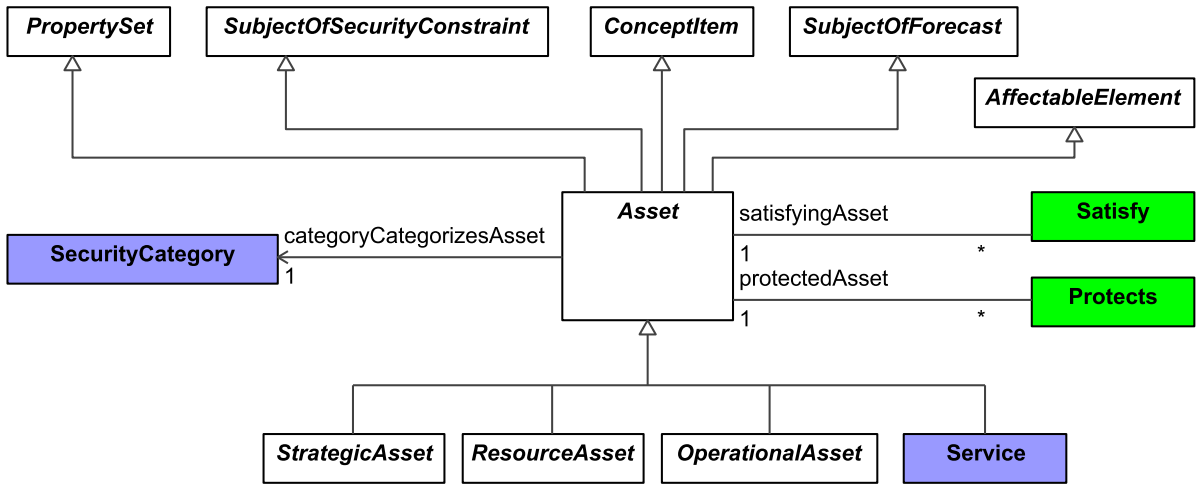

Figure 9:203 - Asset

OperationalAsset

Package: Taxonomy
isAbstract: Yes
Generalization: Asset

Description

An abstract element used to group the elements of OperationalAgent and OperationalInformation allowing them to own OperationalInformationRoles.

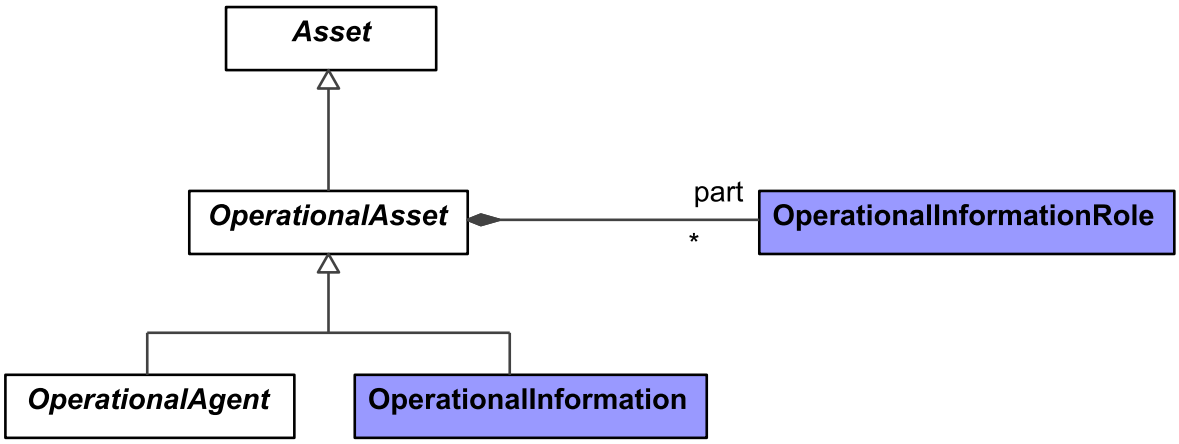

Figure 9:204 - OperationalAsset

OperationalMitigation

Package: Taxonomy
isAbstract: No
Generalization: OperationalArchitecture

Description

A set of OperationalPerformers intended to address against specific operational risks.

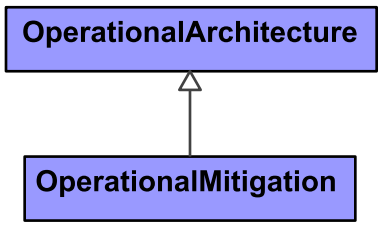

Figure 9:205 - OperationalMitigation

ResourceAsset

Package: Taxonomy
isAbstract: Yes
Generalization: Asset

Description

An abstract element used to group the elements of ResourcePerformer and ResourceInformation allowing them to own ResourceInformationRoles

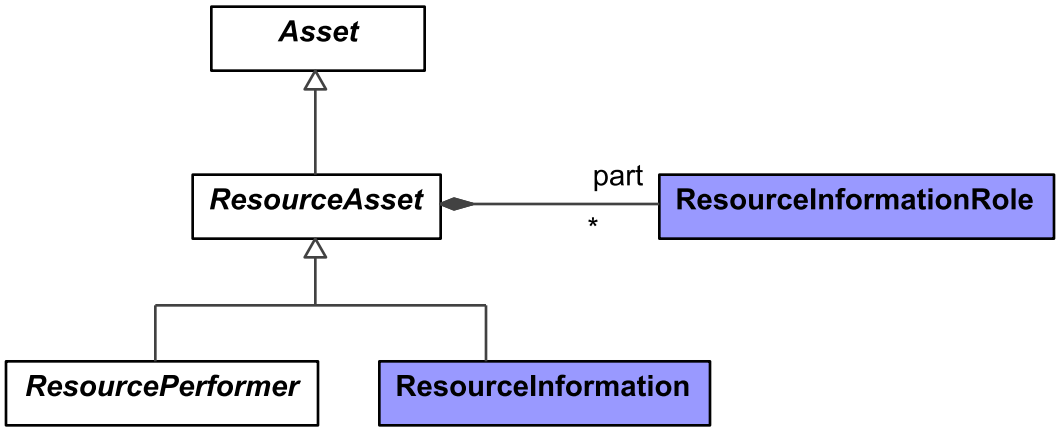

Figure 9:206 - ResourceAsset

ResourceMitigation

Package: Taxonomy
isAbstract: No
Generalization: ResourceArchitecture

Description

A set of ResourcePerformers intended to address against specific risks.

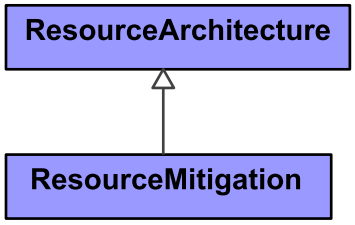

Figure 9:207 - ResourceMitigation

SecurityEnclave

Package: Taxonomy
isAbstract: No
Generalization: ResourceArchitecture

Description

Collection of information systems connected by one or more internal networks under the control of a single authority and security policy. The systems may be structured by physical proximity or by function, independent of location.

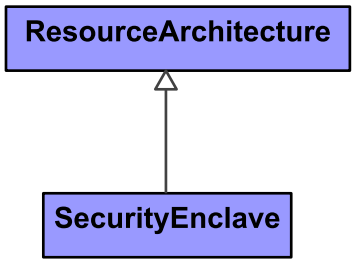

Figure 9:208 - SecurityEnclave

9.1.8.3 Domain MetaModel::Security::Structure

AssetRole

Package: Structure
isAbstract: Yes
Generalization: BPMN2Metamodel::ResourceRole, SubjectOfSecurityConstraint, MeasurableElement

Description

An abstract element that indicates the types of elements that can be affected by Risk in the particular context. AssetRole as applied to Security views, is an abstract element that indicates the type of elements that can be considered as a subject for security analysis in the particular context.

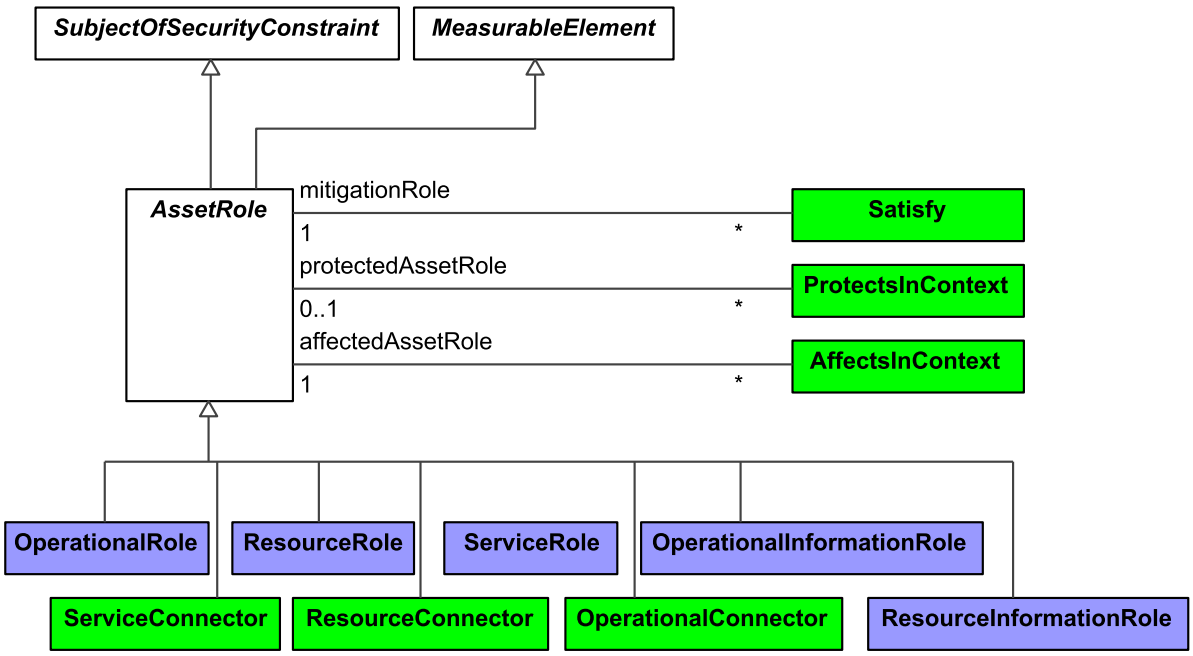

Figure 9:209 - AssetRole

OperationalInformationRole

Package: Structure
isAbstract: No
Generalization: AssetRole

Description

A usage of OperationalInformation that exists in the context of an OperationalAsset. It also allows the representation of the whole-part aggregation of OperationalInformation.

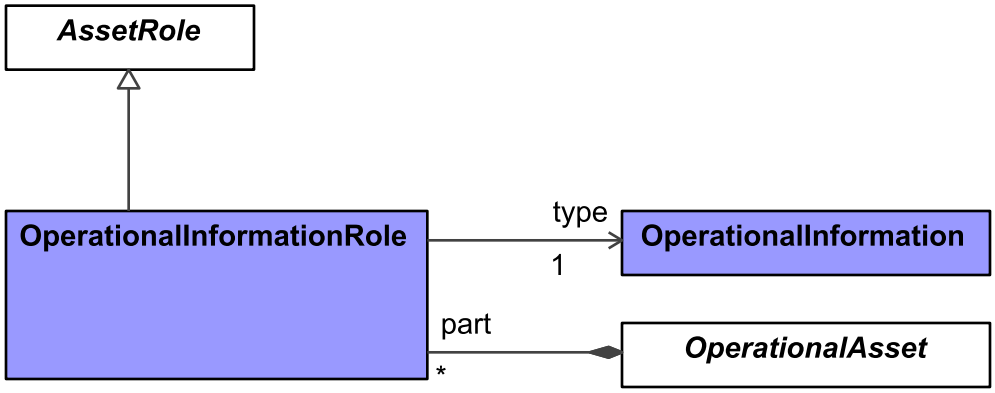

Figure 9:210 - OperationalInformationRole

ResourceInformationRole

Package: Structure
isAbstract: No
Generalization: AssetRole

Description

A usage of ResourceInformation that exists in the context of a ResourceAsset. It also allows the representation of the whole-part aggregation of ResourceInformation elements.

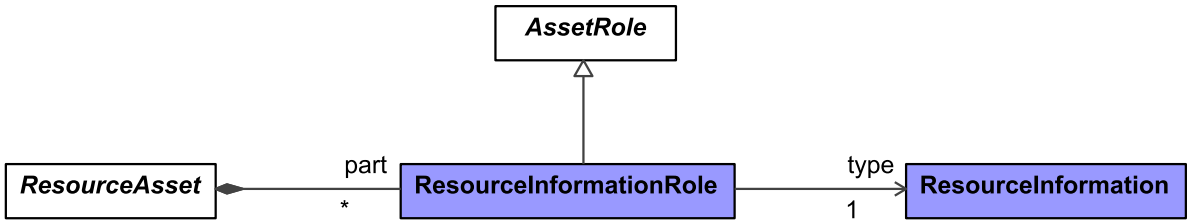

Figure 9:211 - ResourceInformationRole

9.1.8.4 Domain MetaModel::Security::Processes

SecurityProcess

Package: Processes
isAbstract: No
Generalization: OperationalActivity, Function, SubjectOfSecurityConstraint

Description

The security-related procedure that satisfies the security control requirement.

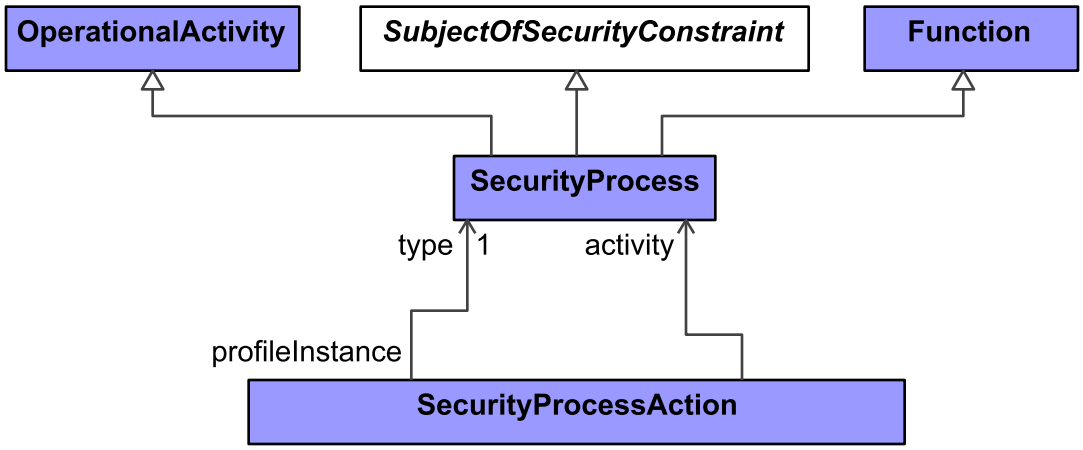

Figure 9:212 - SecurityProcess

SecurityProcessAction

Package: Processes
isAbstract: No
Generalization: OperationalActivityAction, FunctionAction

Description

A call of a SecurityProcess in the context of another SecurityProcess.

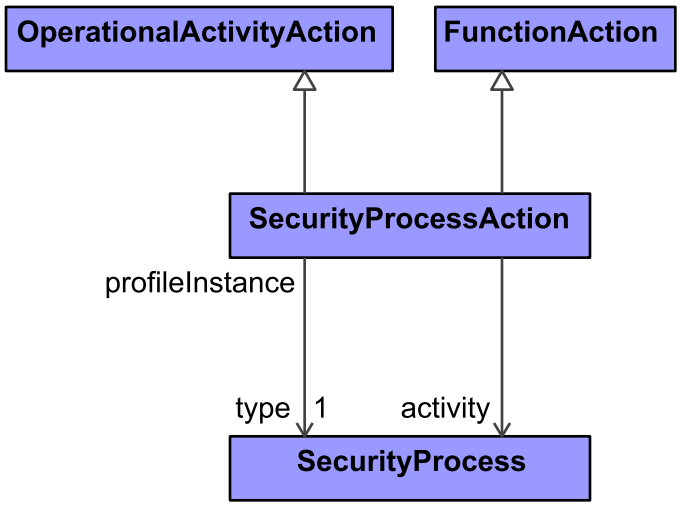

Figure 9:213 - SecurityProcessAction

9.1.8.5 Domain MetaModel::Security::Constraints

Caveat

Package: Constraints
isAbstract: No
Generalization: SecurityConstraint

Description

A statement that details alternate conditions under which the rule is not valid.

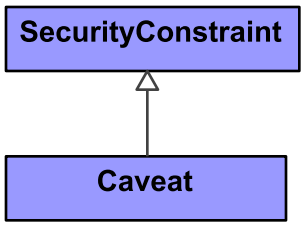

Figure 9:214 - Caveat

SecurityAvailability

Package: Constraints
isAbstract: No
Generalization: SecurityMeasurement

Description

Details the potential impact on organization or individuals if the information is not available to those who need to access it.

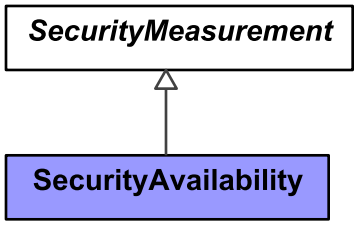

Figure 9:215 - SecurityAvailability

SecurityCategory

Package: Constraints
isAbstract: No
Generalization: MeasurementSet

Description

The security categories that have been determined for each type of information processed, stored, or transmitted by those information systems. The generalized format for expressing the security category (SC) of an information system is:

SC information system = {(confidentiality, impact), (integrity, impact), (availability, impact)}.

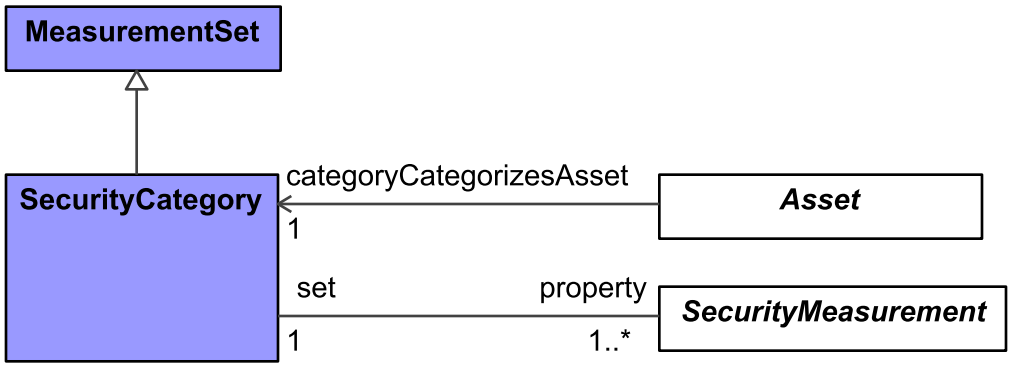

Figure 9:216 - SecurityCategory

SecurityClassification

Package: Constraints
isAbstract: No
Generalization: SecurityMeasurement

Description

Details a classification for the exchange.

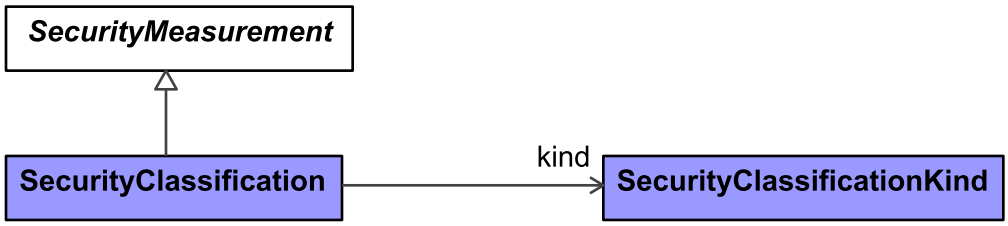

Figure 9:217 - SecurityClassification

SecurityClassificationKind

Package: Constraints
isAbstract: No
Generalization: MeasurableElement

Description

A type that defines acceptable values for the security category (SC) of an information system, where the acceptable values for potential impact are low, moderate, or high.

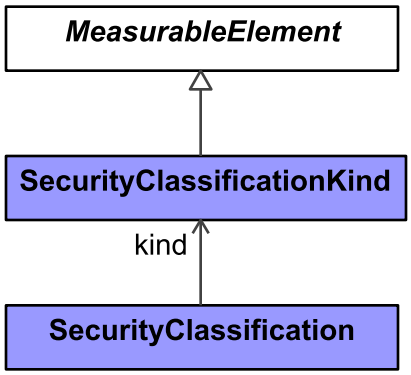

Figure 9:218 - SecurityClassificationKind

SecurityConstraint

Package: Constraints
isAbstract: No
Generalization: Rule

Description

A type of rule that captures a formal statement to define access control policy language.

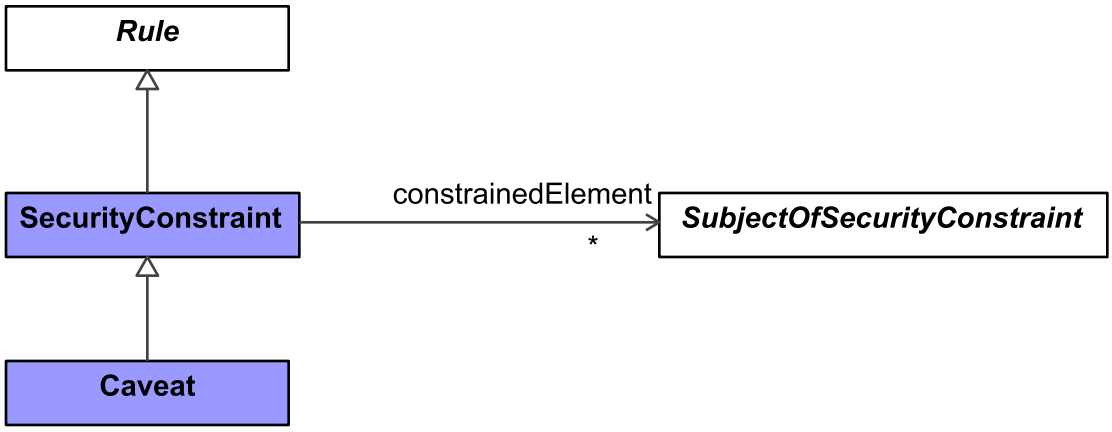

Figure 9:219 - SecurityConstraint

SecurityIntegrity

Package: Constraints
isAbstract: No
Generalization: SecurityMeasurement

Description

Details the potential impact on organization or individuals due to modification or destruction of information, and includes ensuring information non-repudiation and authenticity.

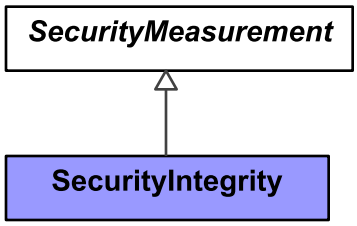

Figure 9:220 - SecurityIntegrity

SecurityMeasurement

Package: Constraints
isAbstract: Yes
Generalization: Measurement

Description

An abstract type grouping all types of security measurements (e.g. SecurityIntegrity, SecurityAvailability).

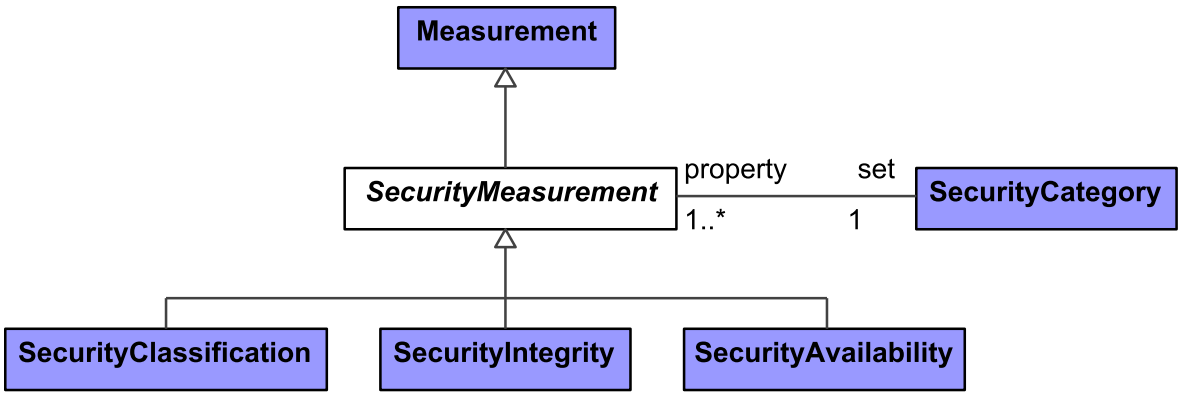

Figure 9:221 - SecurityMeasurement

SecurityRisk

Package: Constraints
isAbstract: No
Generalization: Risk

Description

The level of impact on enterprise operations, assets, or individuals resulting from the operation of an information system given the potential impact of a threat and the likelihood of that threat occurring [NIST SP 800-65].

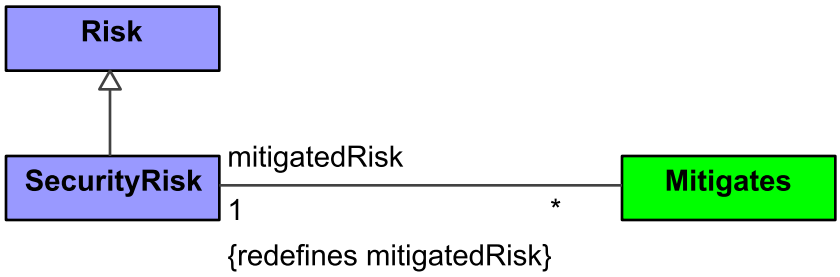

Figure 9:222 - SecurityRisk

SubjectOfSecurityConstraint

Package: Constraints
isAbstract: Yes
Generalization: UAFElement

Description

An abstract type grouping elements that can be the subject of a SecurityConstraint.

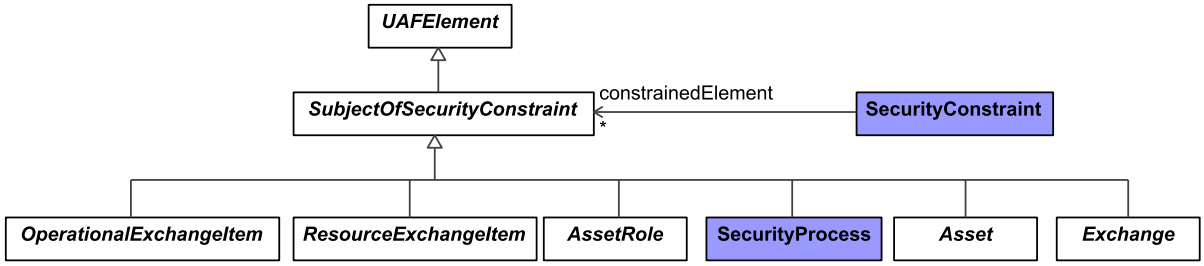

Figure 9:223 - SubjectOfSecurityConstraint

9.1.9 Domain MetaModel::Projects

9.1.9.1 Domain MetaModel::Projects::Taxonomy

Project

Package: Taxonomy
isAbstract: No
Generalization: OrganizationalResource

Description

A type that represents a planned endeavor executed by an ActualOrganization responsible for developing, deploying or decommissioning ResourcePerformers in accordance with ActualProjectMilestones.

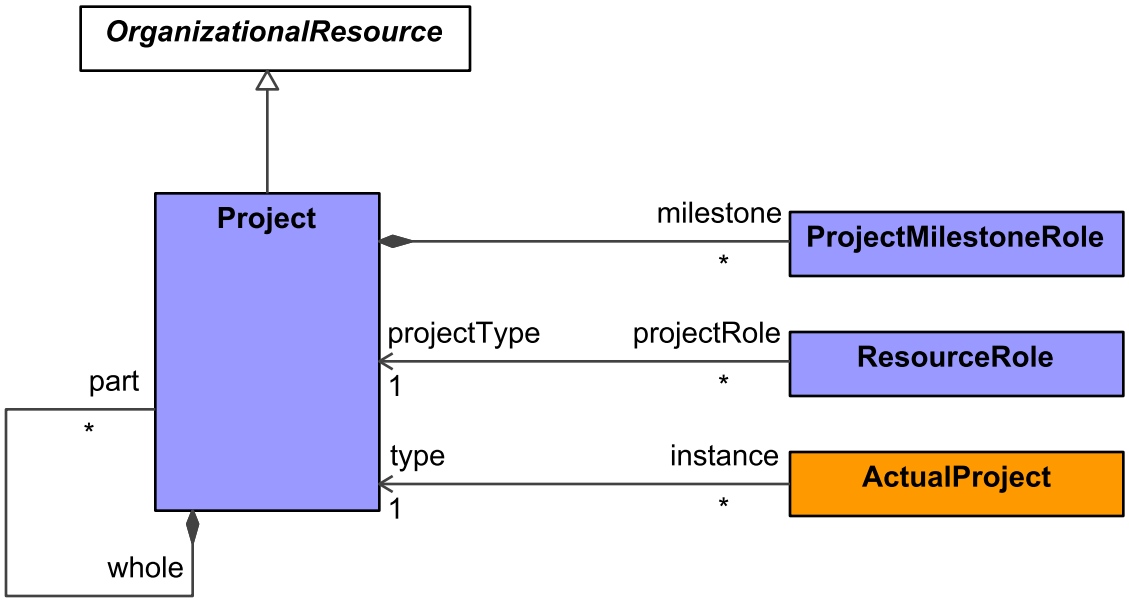

Figure 9:224 - Project

ProjectMilestone

Package: Taxonomy
isAbstract: No
Generalization: PropertySet

Description

A type of event in a Project by which progress is measured.

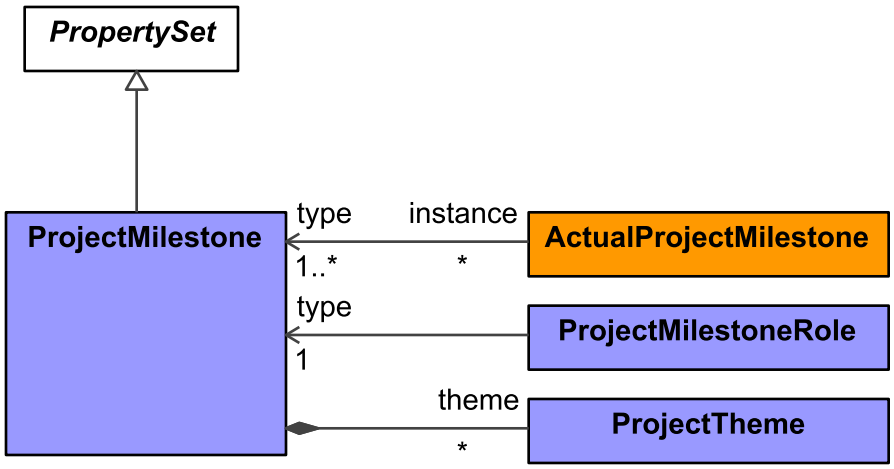

Figure 9:225 - ProjectMilestone

9.1.9.2 Domain MetaModel::Projects::Structure

ActualProjectMilestoneRole

Package: Structure

isAbstract: No
Generalization: ActualState

Description

An ActualProjectMilestone that is applied to a ProjectMilestoneRole.

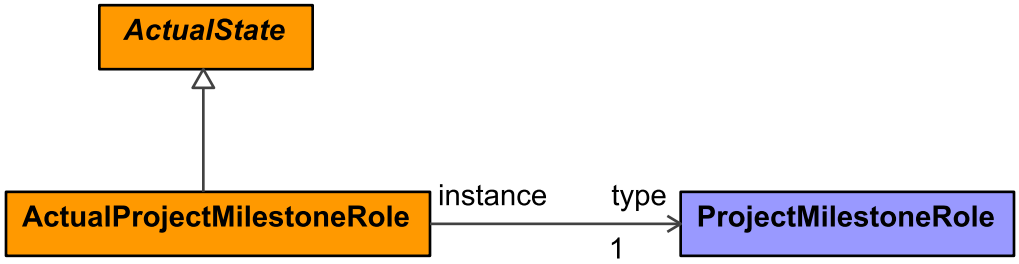

Figure 9:226 - ActualProjectMilestoneRole

ProjectMilestoneRole

Package: Structure
isAbstract: No
Generalization: MeasurableElement

Description

The role played by a ProjectMilestone in the context of a Project.

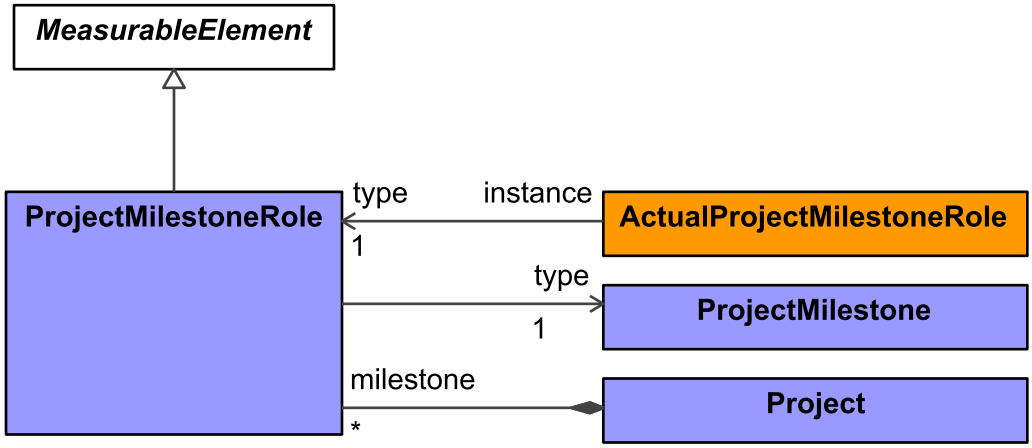

Figure 9:227 - ProjectMilestoneRole

ProjectStatus

Package: Structure
isAbstract: No
Generalization: ActualState

Description

The status (i.e. level of progress) of a ProjectTheme for an ActualProject at the time of the ActualProjectMilestone.

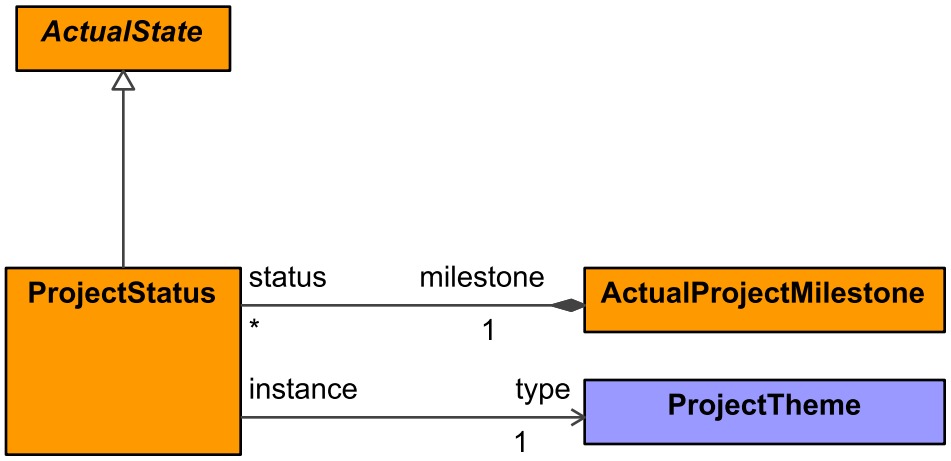

Figure 9:228 - ProjectStatus

ProjectTheme

Package: Structure
isAbstract: No
Generalization: MeasurableElement

Description

A property of a ProjectMilestone that captures an aspect by which the progress of ActualProjects may be measured.

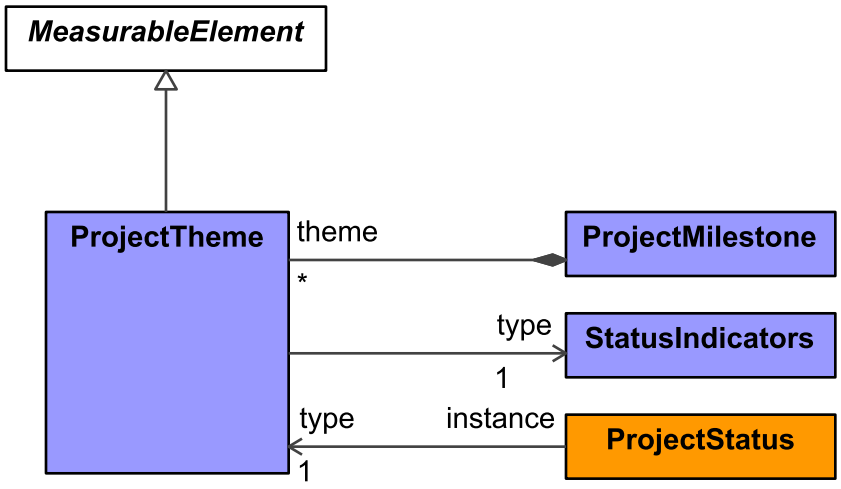

Figure 9:229 - ProjectTheme

StatusIndicators

Package: Structure
isAbstract: No
Generalization: MeasurableElement

Description

An enumerated type that specifies a status for a ProjectTheme.

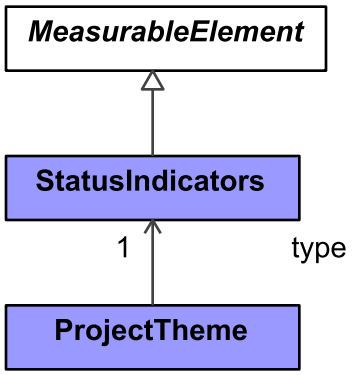

Figure 9:230 - StatusIndicators

9.1.9.3 Domain MetaModel::Projects::Connectivity

MilestoneDependency

Package: Connectivity
isAbstract: No
Generalization: Sequence

Description

A tuple between two ActualProjectMilestones that denotes one ActualProjectMilestone follows from another.

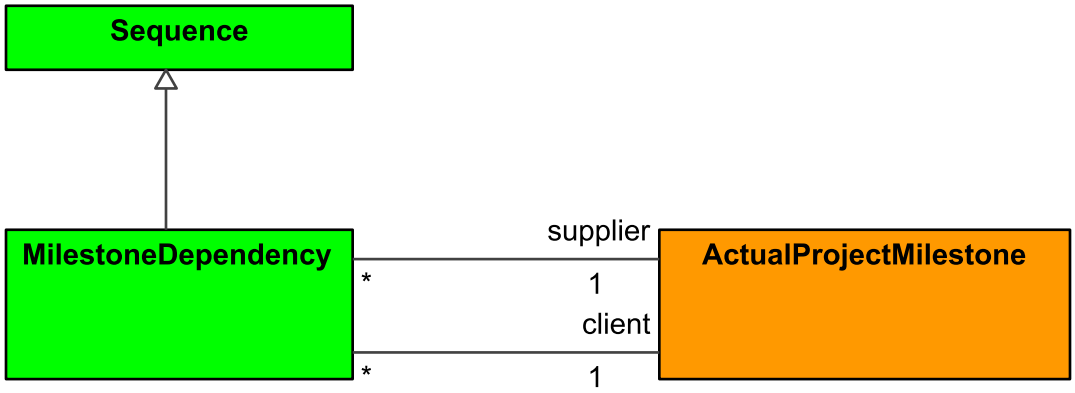

Figure 9:231 - MilestoneDependency

ProjectSequence

Package: Connectivity
isAbstract: No
Generalization: Sequence

Description

A tuple between two ActualProjects that denotes one ActualProject cannot start before the previous ActualProject is finished.

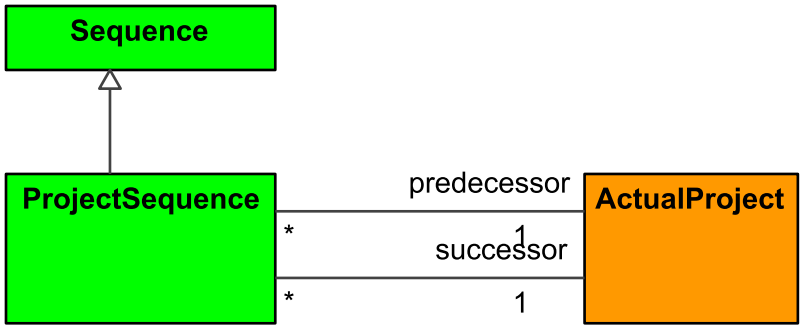

Figure 9:232 - ProjectSequence

9.1.9.4 Domain MetaModel::Projects::Processes

ProjectActivity

Package: Processes
isAbstract: No
Generalization: Function, Process

Description

An activity carried out during a project.

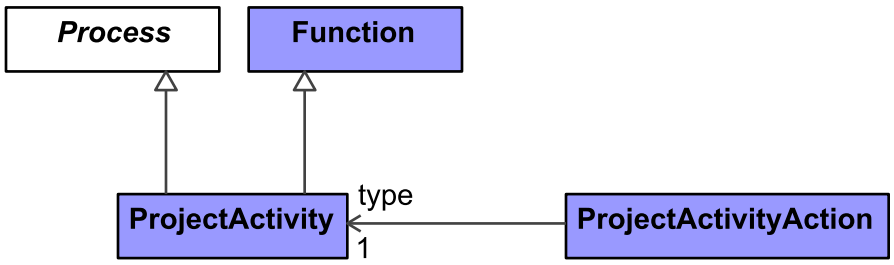

Figure 9:233 - ProjectActivity

ProjectActivityAction

Package: Processes
isAbstract: No
Generalization: FunctionAction

Description

The ProjectActivityAction is defined as a call behavior action that invokes the activity that needs to be preformed.

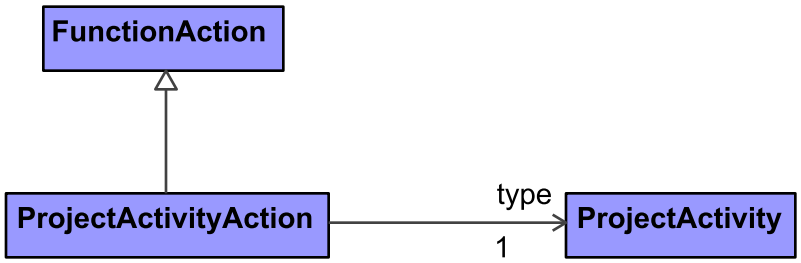

Figure 9:234 - ProjectActivityAction

9.1.9.5 Domain MetaModel::Projects::Roadmap

ActualProject

Package: Roadmap
isAbstract: No
Generalization: ActualOrganizationalResource, Achiever

Description

A time-limited planned endeavor executed by an ActualOrganization responsible for developing, deploying or decommissioning ResourcePerformers in accordance with ActualProjectMilestones.

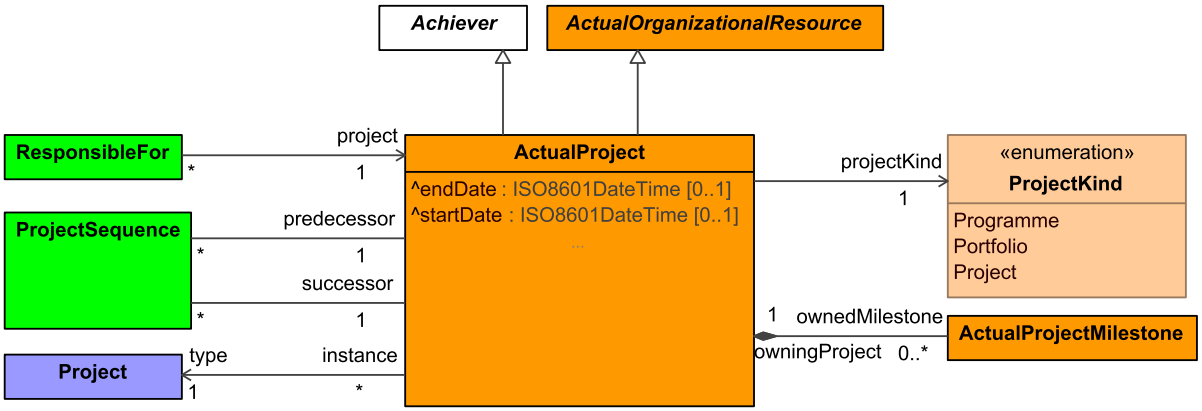

Figure 9:235 - ActualProject

ActualProjectMilestone

Package: Roadmap
isAbstract: No
Generalization: ActualPropertySet

Description

An event with a start date in a ActualProject from which progress is measured.

Figure 9:236 - ActualProjectMilestone

Constraints

[1] unnamed1 startTime=endTime

9.1.9.6 Domain MetaModel::Projects::Traceability

ResponsibleFor

Package: Traceability
isAbstract: No
Generalization: MeasurableElement

Description

A tuple between an ActualResponsibleResource and an ActualResponsibility or ActualProject. It defines the duties that the ActualResponsibleResource is ResponsibleFor.

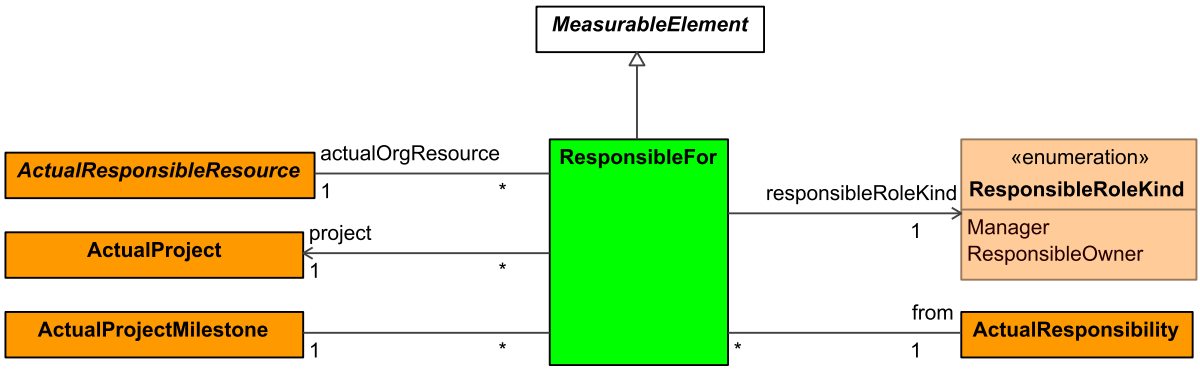

Figure 9:237 - ResponsibleFor

9.1.10 Domain MetaModel::Standards

Stakeholders: Solution Providers, Systems Engineers, Software Engineers, Systems Architects, Business Architects.

Concerns: technical and non-technical Standards applicable to the architecture.

Definition: shows the technical, operational, and business Standards applicable to the architecture. Defines the underlying current and expected Standards.

9.1.10.1 Domain MetaModel::Standards::Taxonomy

Doctrine

Package: Taxonomy
isAbstract: No
Generalization: Standard

Description

A formal expression of military knowledge and thought, accepted as being relevant at a given time, which covers the nature of conflict, the preparation of the military forces for conflict, and the method of engaging in conflict to achieve success. it is descriptive rather than prescriptive, requiring judgment in application.

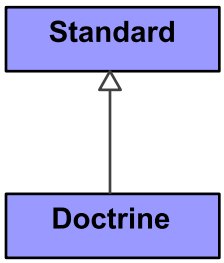

Figure 9:238 - Doctrine

Protocol

Package: Taxonomy
isAbstract: No
Generalization: Standard

Description

A Standard for communication over a network. Protocols may be composite, represented as a ProtocolStack made up of ProtocolLayers.

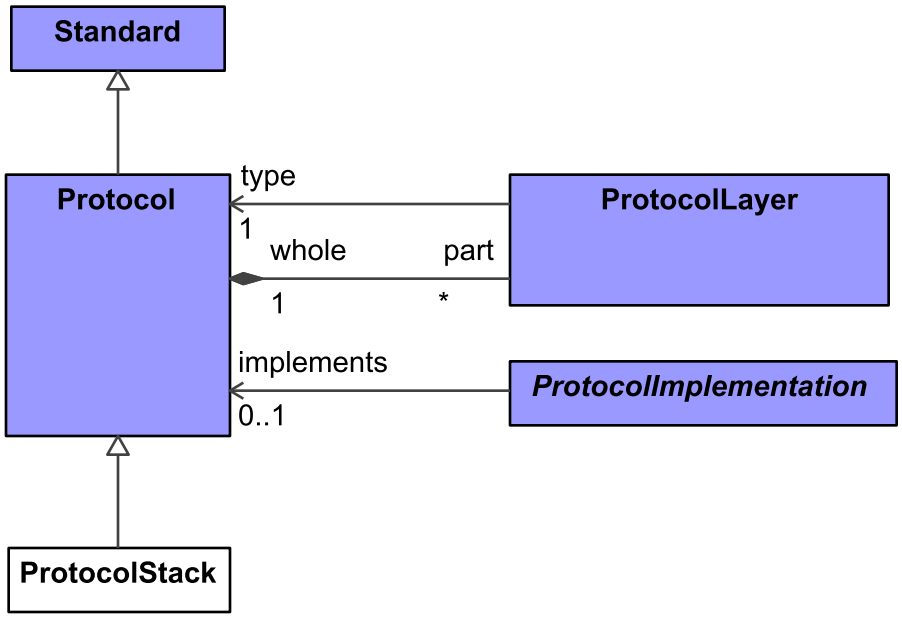

Figure 9:239 - Protocol

ProtocolStack

Package: Taxonomy
isAbstract: No
Generalization: Protocol

Description

A sub type of Protocol that contains the ProtocolLayers, defining a complete stack.

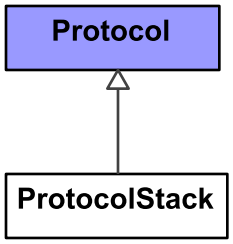

Figure 9:240 - ProtocolStack

Standard

Package: Taxonomy
isAbstract: No
Generalization: SubjectOfForecast, PropertySet

Description

A ratified and peer-reviewed specification that is used to guide or constrain the architecture. A Standard may be applied to any element in the architecture.

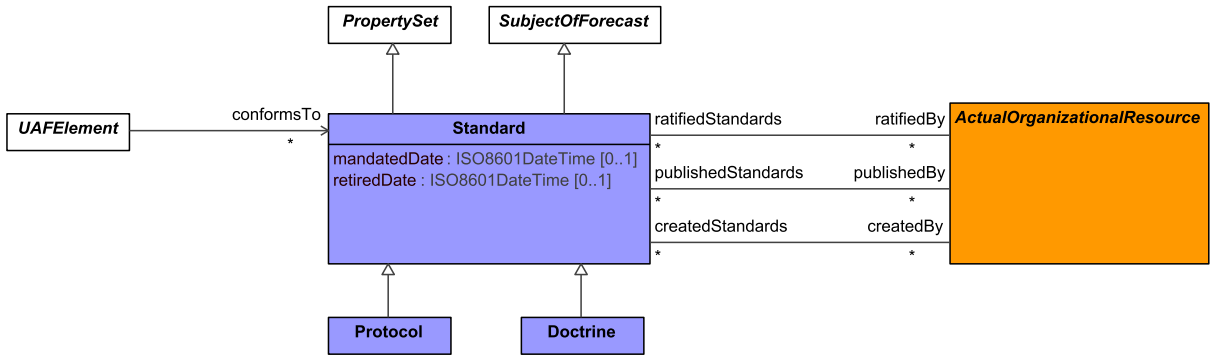

Figure 9:241 - Standard

Attributes

mandatedDate : ISO8601DateTime [0..1] The date when this version of the Standard was published.
retiredDate : ISO8601DateTime [0..1] The date when this version of the Standard was retired.

9.1.10.2 Domain MetaModel::Standards::Structure

ProtocolLayer

Package: Structure
isAbstract: No
Generalization: MeasurableElement

Description

Usage of a Protocol in the context of another Protocol. Creates a whole-part relationship.

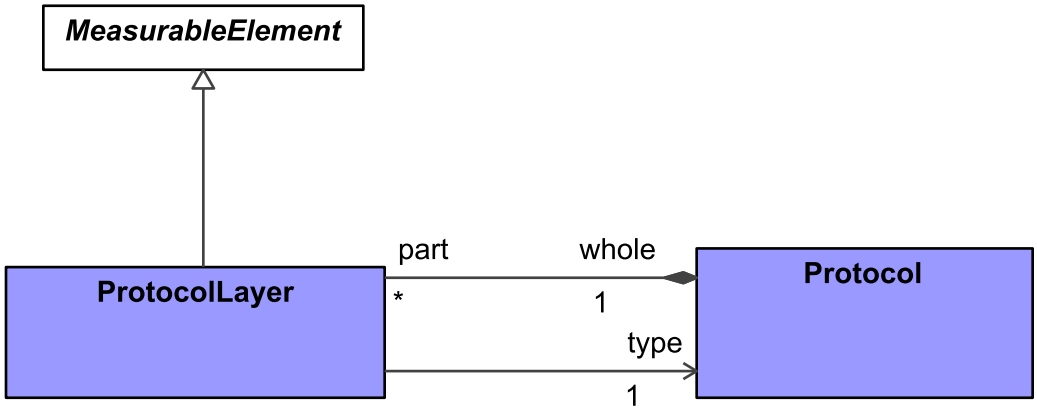

Figure 9:242 - ProtocolLayer

9.1.11 Domain MetaModel::Actual Resources

Stakeholders: Solution Providers, Systems Engineers, Business Architects, Human Resources.

Concerns: the analysis, e.g. evaluation of different alternatives, what-if, trade-offs, V&V on the actual resource configurations.

Definition: illustrates the expected or achieved actual resource configurations and actual relationships between them.

9.1.11.1 Domain MetaModel::Actual Resources::Taxonomy

ActualOrganization

Package: Taxonomy
isAbstract: No
Generalization: ActualResponsibleResource

Description

An actual formal or informal organizational unit, e.g. "Driving and Vehicle Licensing Agency", "UAF team Alpha".

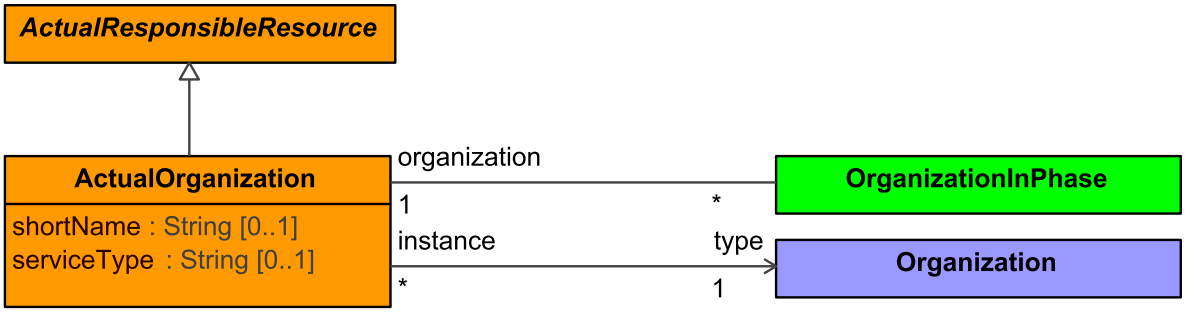

Figure 9:243 - ActualOrganization

Attributes

serviceType : String [0..1]	Service office code or symbol
shortName : String [0..1]	String providing a simplified means of identifying an ActualOrganization, i.e. SoftWareGroup could use SWG as the shortName.

ActualOrganizationalResource

Package: Taxonomy
isAbstract: Yes
Generalization: ActualResource, Stakeholder

Description

Abstract element for an ActualOrganization, ActualPerson or ActualPost.

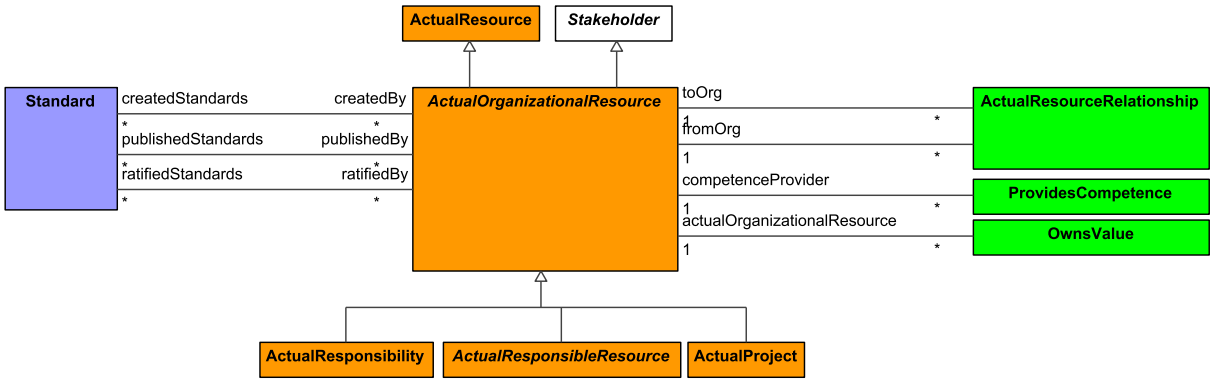

Figure 9:244 - ActualOrganizationalResource

ActualPerson

Package: Taxonomy
isAbstract: No
Generalization: ActualResponsibleResource

Description

An individual human being.

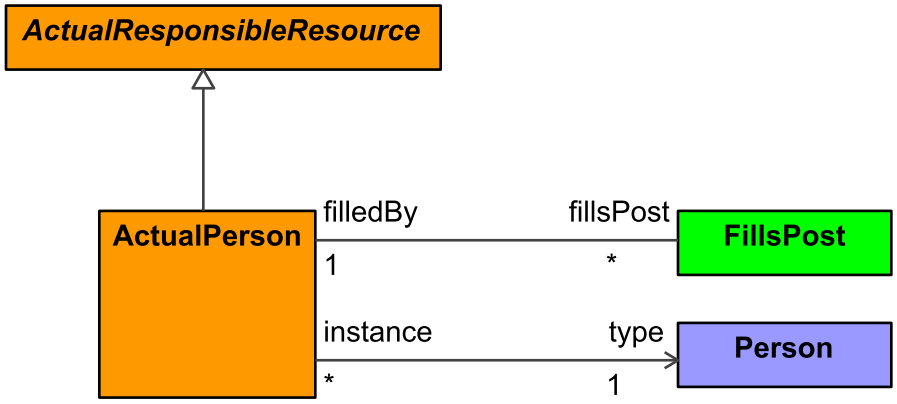

Figure 9:245 - ActualPerson

ActualPost

Package: Taxonomy
isAbstract: No
Generalization: ActualResponsibleResource

Description

An actual, specific post, an instance of a Post "type" - e.g., "President of the United States of America." where the Post would be president.

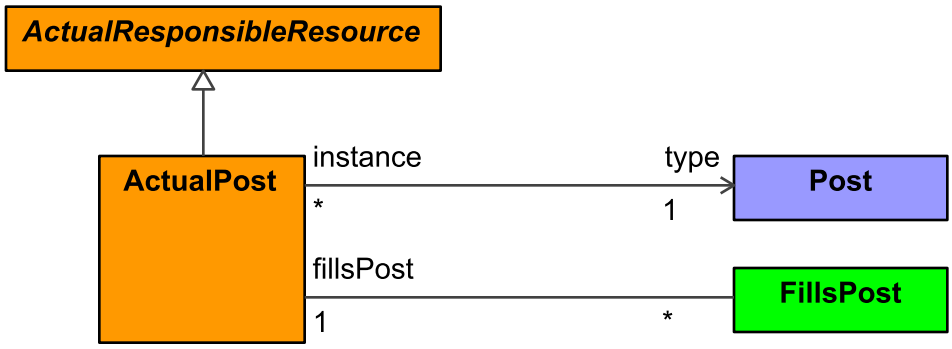

Figure 9:246 - ActualPost

ActualResource

Package: Taxonomy
isAbstract: No
Generalization: ActualPropertySet, SubjectOfResourceConstraint, Achiever, CapableElement

Description

An instance of a ResourcePerformer in the real world.

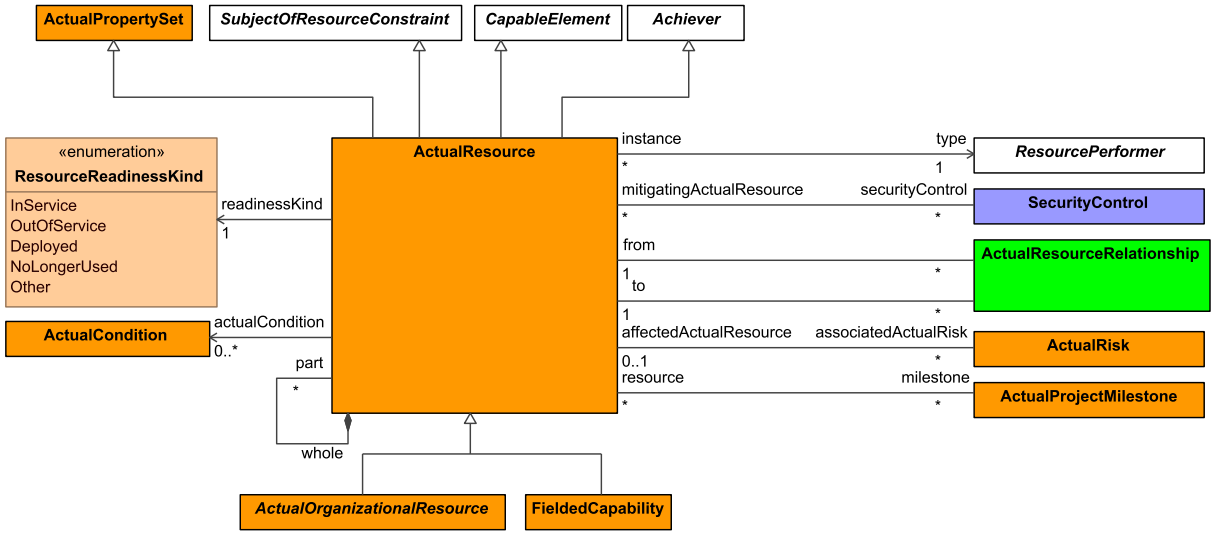

Figure 9:247 - ActualResource

ActualResourceRelationship

Package: Taxonomy
isAbstract: No
Generalization: UAFElement

Description

An abstract element that details the ActualOrganizationalResources that are able to carry out an ActualResponsibility.

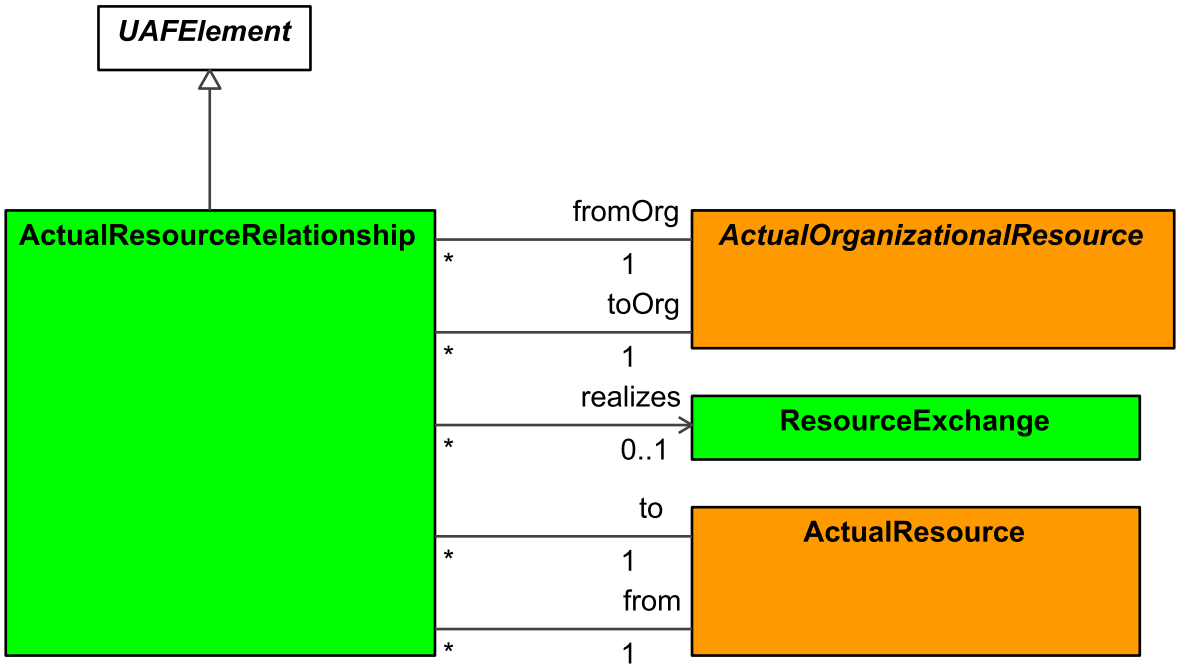

Figure 9:248 - ActualResourceRelationship

ActualResponsibility

Package: Taxonomy
isAbstract: No
Generalization: ActualOrganizationalResource

Description

An actual duty required of a Person or Organization.

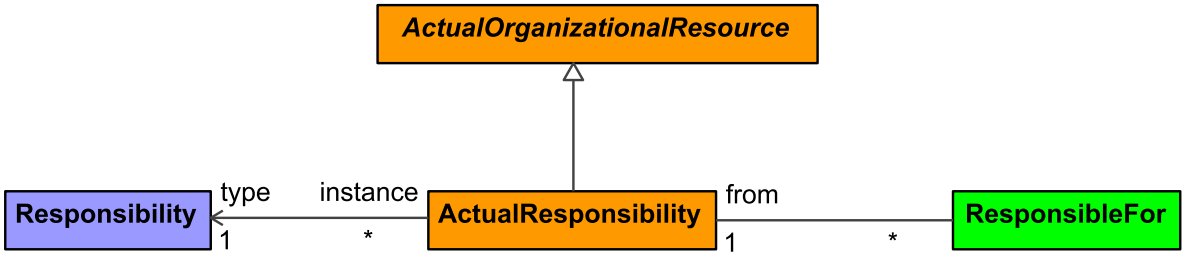

Figure 9:249 - ActualResponsibility

ActualResponsibleResource

Package: Taxonomy
isAbstract: Yes
Generalization: ActualOrganizationalResource

Description

An abstract type grouping responsible OrganizationalResources.

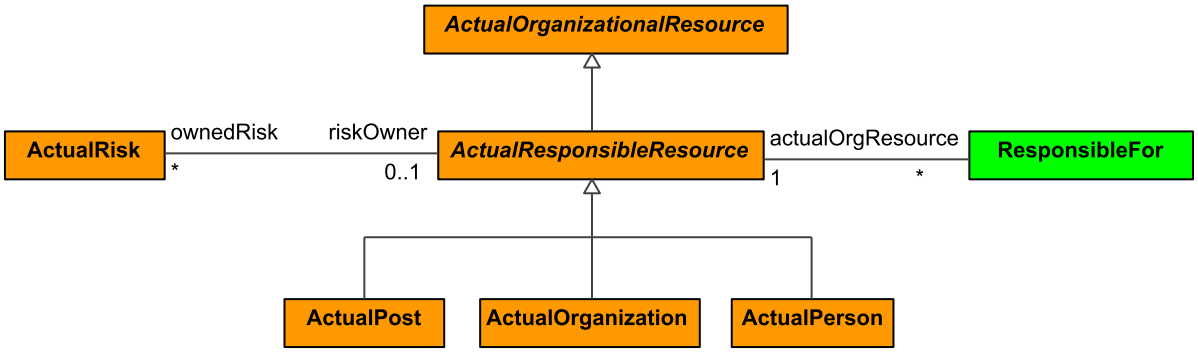

Figure 9:250 - ActualResponsibleResource

FieldedCapability

Package: Taxonomy
isAbstract: No
Generalization: ActualResource

Description

An individual, fully-realized capability.

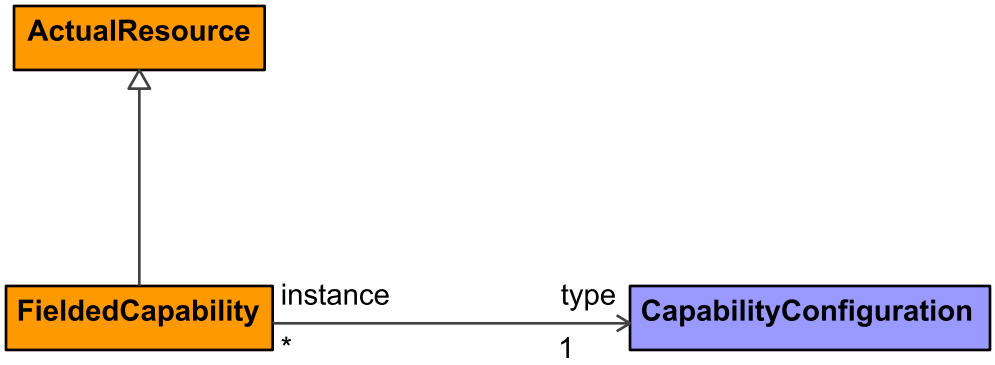

Figure 9:251 - FieldedCapability

9.1.11.2 Domain MetaModel::Actual Resources::Constraints

ActualService

Package: Constraints
isAbstract: Yes
Generalization: ActualMeasurementSet, CapableElement

Description

An individual Service.

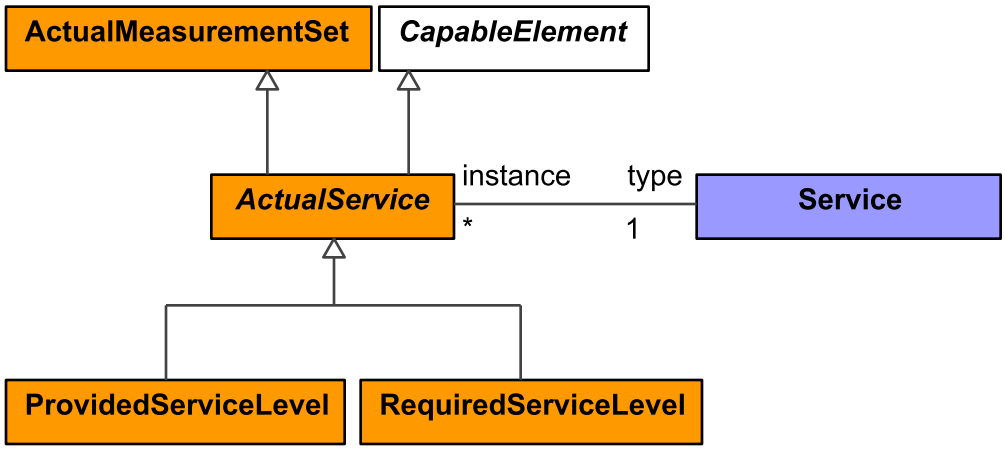

Figure 9:252 - ActualService

ProvidedServiceLevel

Package: Constraints
isAbstract: No
Generalization: ActualService

Description

A sub type of ActualService that details a specific service level delivered by the provider.

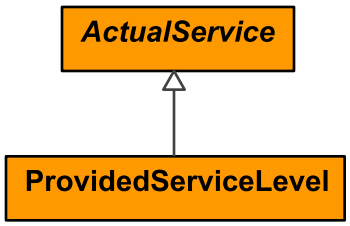

Figure 9:253 - ProvidedServiceLevel

ProvidesCompetence

Package: Constraints
isAbstract: No
Generalization: MeasurableElement

Description

A tuple that asserts that an ActualOrganizationalResource provides a specific set of Competencies.

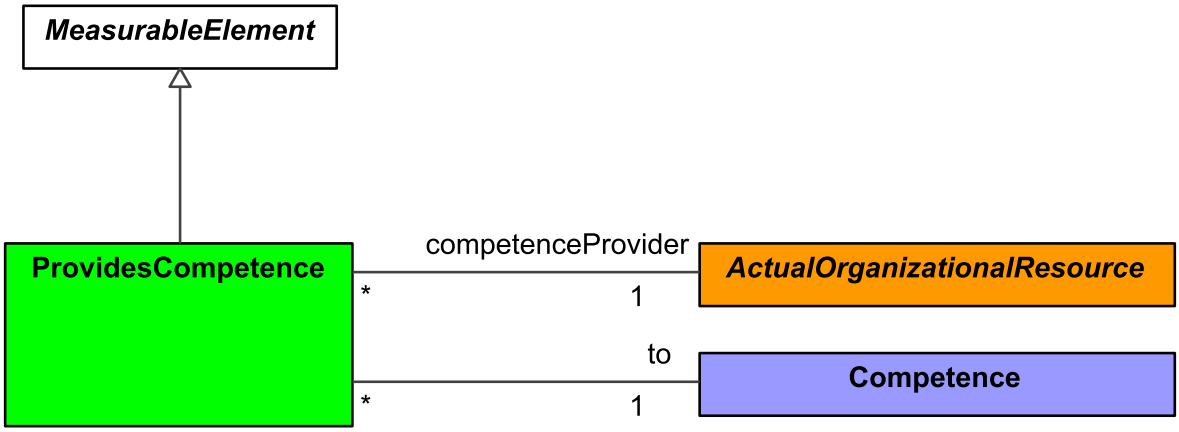

Figure 9:254 - ProvidesCompetence

RequiredServiceLevel

Package: Constraints
isAbstract: No
Generalization: ActualService

Description

A sub type of ActualService that details a specific service level required of the provider.

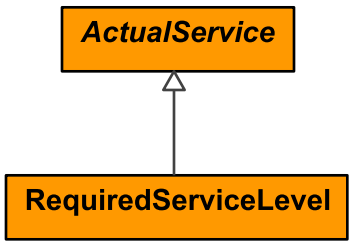

Figure 9:255 - RequiredServiceLevel

9.1.11.3 Domain MetaModel::Actual Resources::Traceability

OwnsProcess

Package: Traceability
isAbstract: No
Generalization: MeasurableElement

Description

A tuple denoting that an ActualOrganizationResource owns an OperationalActivity.

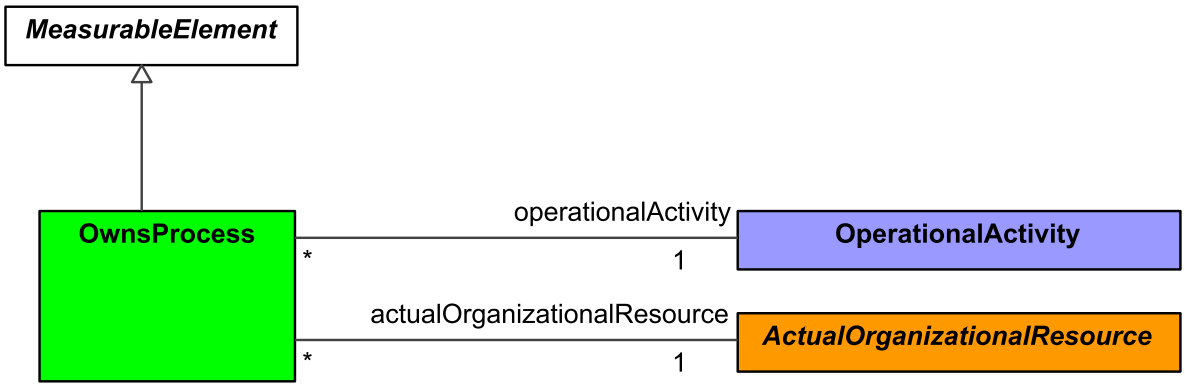

Figure 9:256 - OwnsProcess

9.1.12 Domain MetaModel::Parameters

ActualCondition

Package: Parameters
isAbstract: No
Generalization: ActualPropertySet

Description

An individual describing an actual situation with respect to circumstances under which an OperationalActivity, Function or ServiceFunction can be performed.

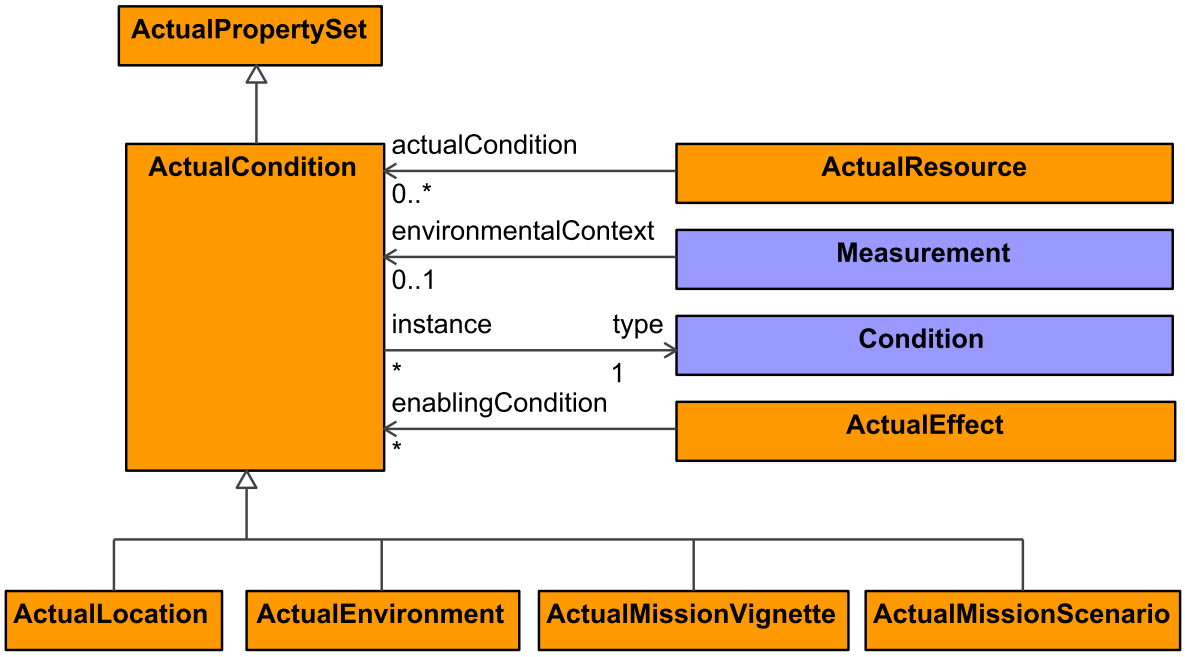

Figure 9:257 - ActualCondition

ActualEnvironment

Package: Parameters

isAbstract: No
Generalization: ActualCondition

Description

An individual that describes the circumstances of an Environment.

Figure 9:258 - ActualEnvironment

ActualLocation

Package: Parameters
isAbstract: No
Generalization: ActualCondition

Description

An individual that describes a physical location, for example using text to provide an address, Geo-coordinates, etc.

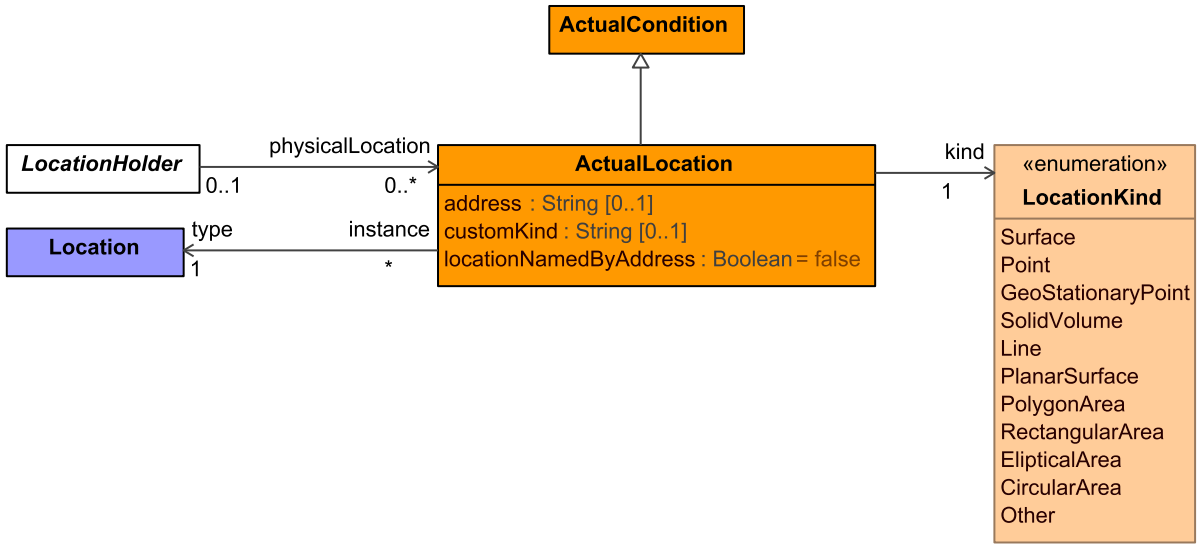

Figure 9:259 - ActualLocation

Attributes

address : String [0..1]	String describing the address of the ActualLocation, i.e. "1600 Pennsylvania avenue", "The White House"
customKind : String [0..1]	String describing a location kind that is not in the LocationKind enumerated list
locationNamedByAddress : Boolean []	Boolean that indicates if the ActualLocation address is embedded in the ActualLocation name. By default = false.

ActualMeasurement

Package: Parameters
isAbstract: No
Generalization: ActualState

Description

An actual value that is applied to a Measurement.

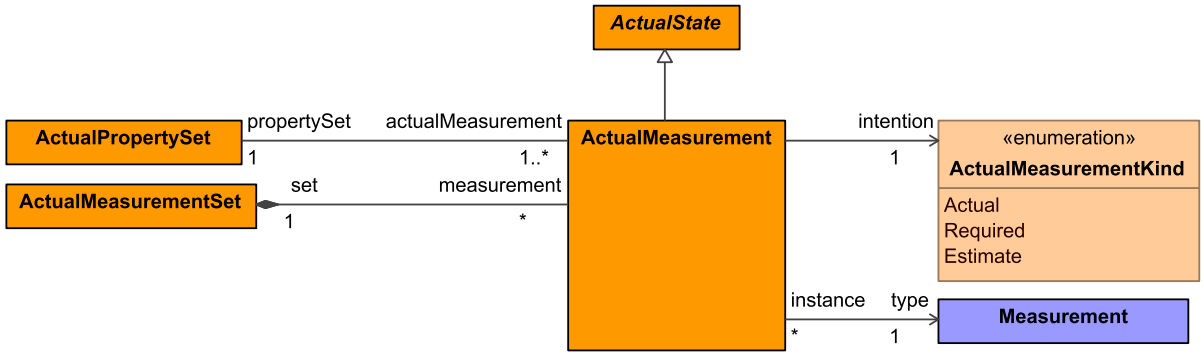

Figure 9:260 - ActualMeasurement

ActualMeasurementSet

Package: Parameters
isAbstract: No
Generalization: ActualPropertySet

Description

A set of ActualMeasurements.

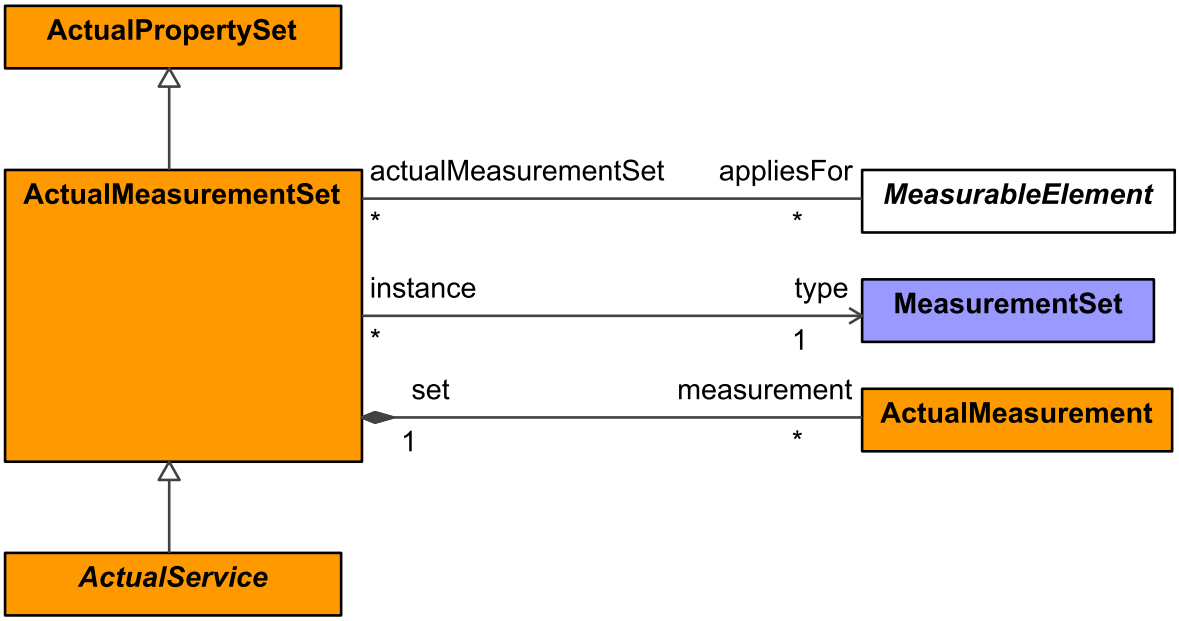

Figure 9:261 - ActualMeasurementSet

ActualMissionScenario

Package: Parameters
isAbstract: No
Generalization: ActualCondition

Description

A particular Mission Scenario within which an Actual Mission is to be accomplished.

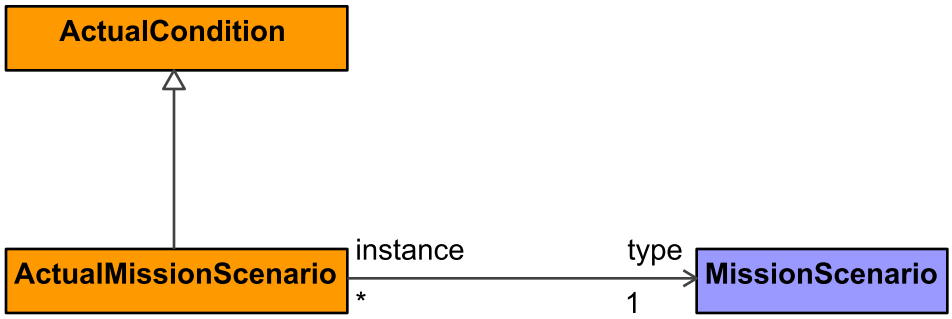

Figure 9:262 - ActualMissionScenario

ActualMissionVignette

Package: Parameters
isAbstract: No
Generalization: ActualCondition

Description

A particular Mission Vignette within which an Actual Mission is to be accomplished

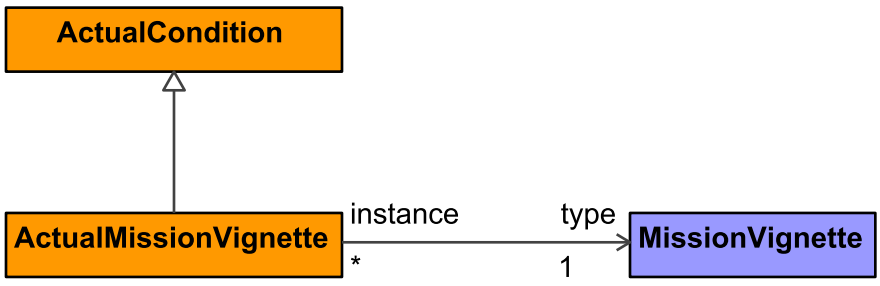

Figure 9:263 - ActualMissionVignette

ActualPropertySet

Package: Parameters
isAbstract: No
Generalization: ActualState

Description

A set or collection of Actual properties.

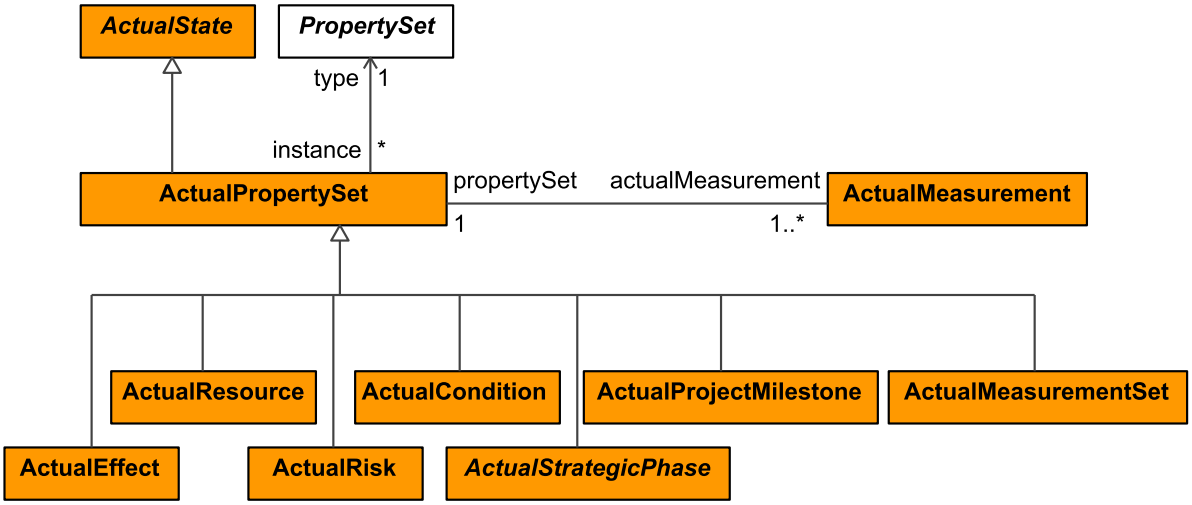

Figure 9:264 - ActualPropertySet

ActualRisk

Package: Parameters
isAbstract: No
Generalization: ActualPropertySet

Description

An instance of a Risk. A value holder for Risk Measurements.

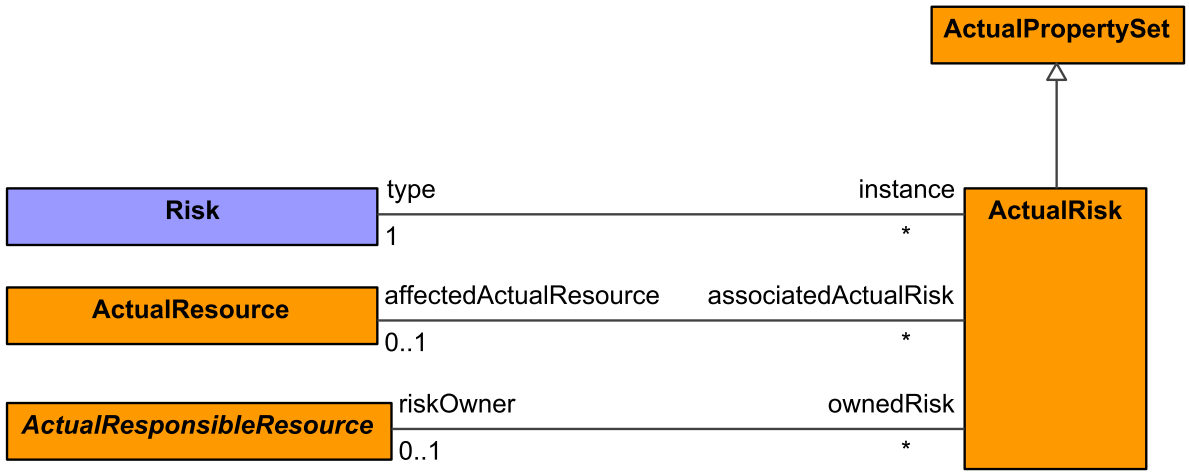

Figure 9:265 - ActualRisk

AffectableElement

Package: Parameters
isAbstract: Yes
Generalization: UAFElement

Description

An abstract grouping of elements that can be affected by Risk.

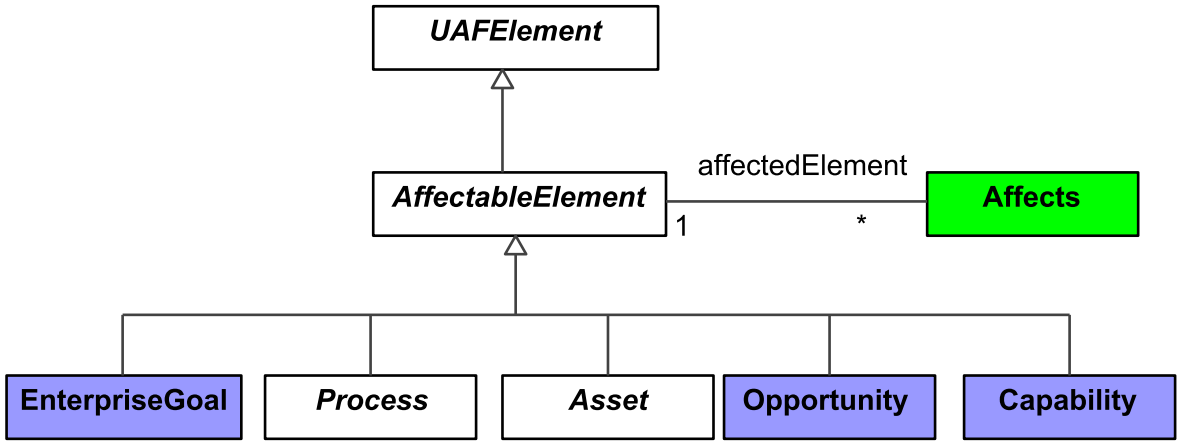

Figure 9:266 - AffectableElement

Affects

Package: Parameters
isAbstract: No
Generalization: MeasurableElement

Description

A tuple that asserts that a Risk is applicable to an Asset.

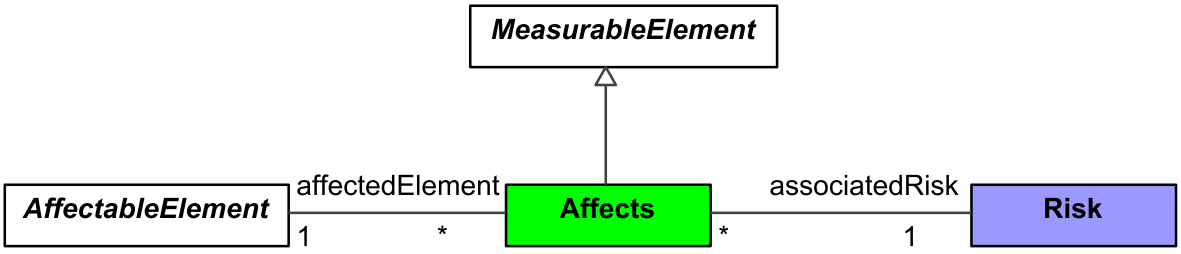

Figure 9:267 - Affects

AffectsInContext

Package: Parameters
isAbstract: No
Generalization: MeasurableElement

Description

A tuple that asserts that a Risk is applicable to an AssetRole in the specific context or configuration.

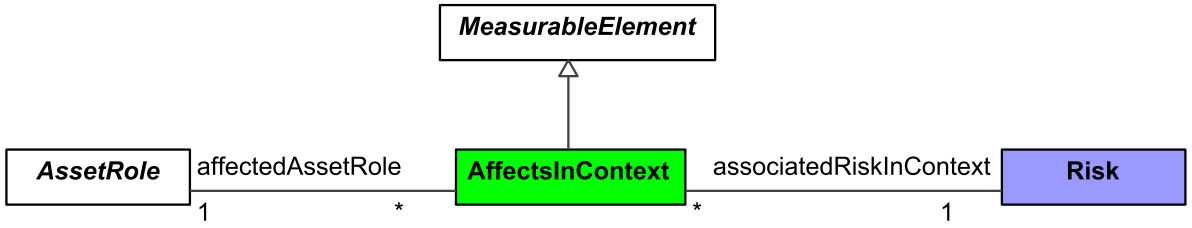

Figure 9:268 - AffectsInContext

Condition

Package: Parameters
isAbstract: No
Generalization: PropertySet

Description

A type that defines the Location, Environment and/or GeoPoliticalExtent.

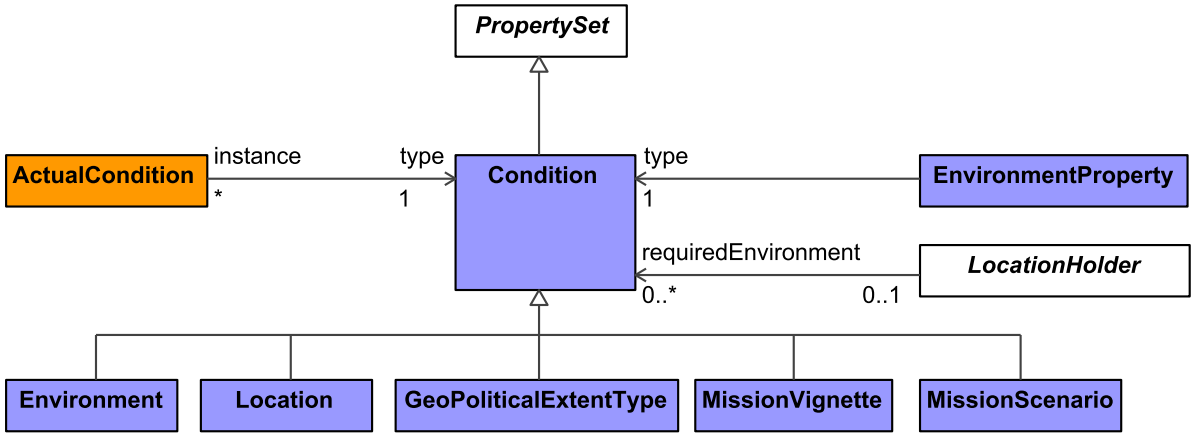

Figure 9:269 - Condition

Environment

Package: Parameters
isAbstract: No
Generalization: Condition

Description

A definition of the environmental factors in which something exists or functions. The definition of an Environment element can be further defined using EnvironmentKind.

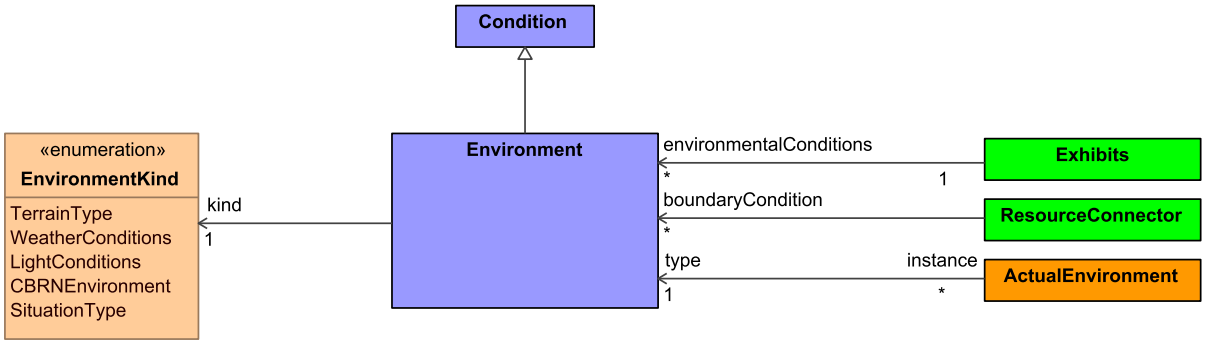

Figure 9:270 - Environment

EnvironmentProperty

Package: Parameters
isAbstract: No
Generalization: MeasurableElement

Description

A property of an Environment that is typed by a Condition. The kinds of Condition that can be represented are Location, GeoPoliticalExtentType and Environment.

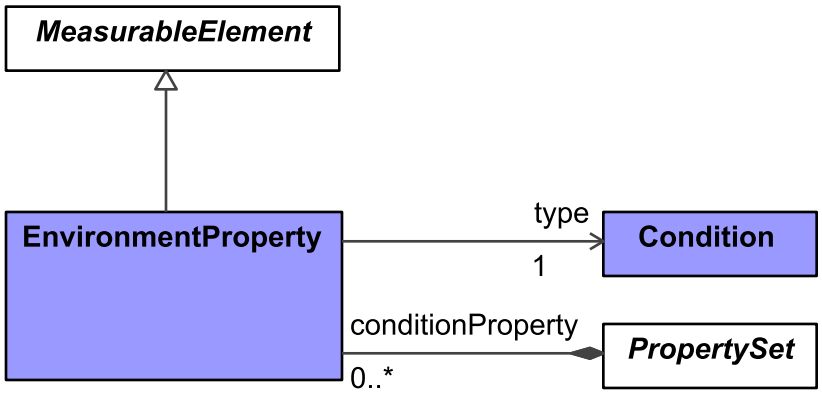

Figure 9:271 - EnvironmentProperty

GeoPoliticalExtentType

Package: Parameters
isAbstract: No
Generalization: Condition, OperationalExchangeItem, ResourceExchangeItem, StrategicExchangeItem

Description

A type of geospatial extent whose boundaries are defined by declaration or agreement by political parties.

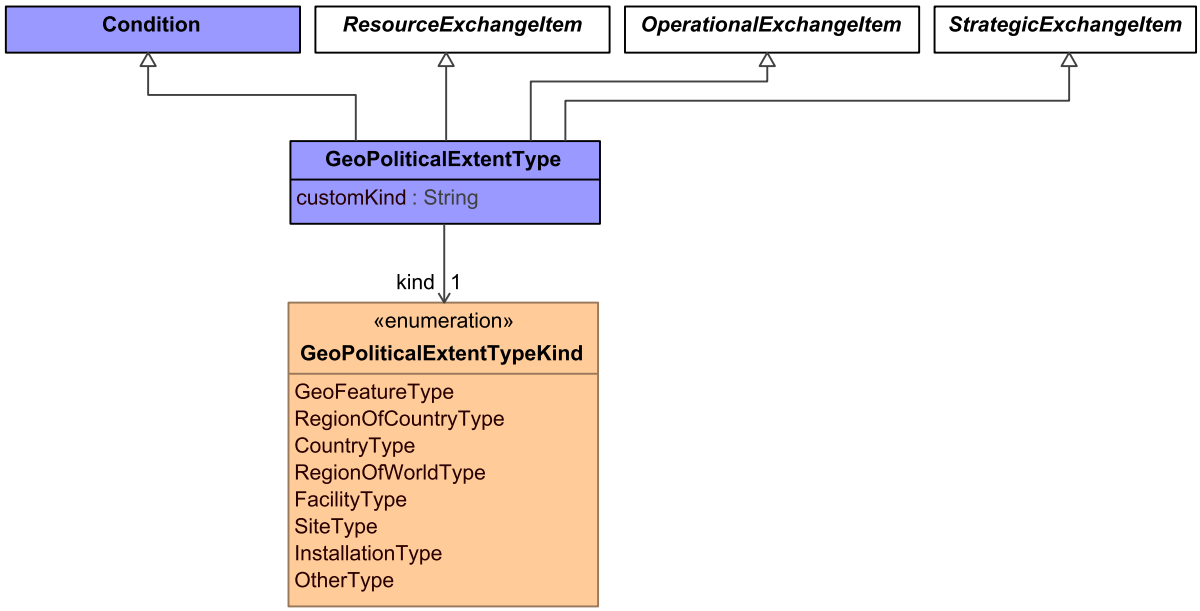

Figure 9:272 - GeoPoliticalExtentType

Attributes

customKind : String [] Captures the kind of GeopoliticalExtentType.

Location

Package: Parameters
isAbstract: No
Generalization: ConceptItem, Condition

Description

A specification of the generic area in which a LocationHolder is required to be located.

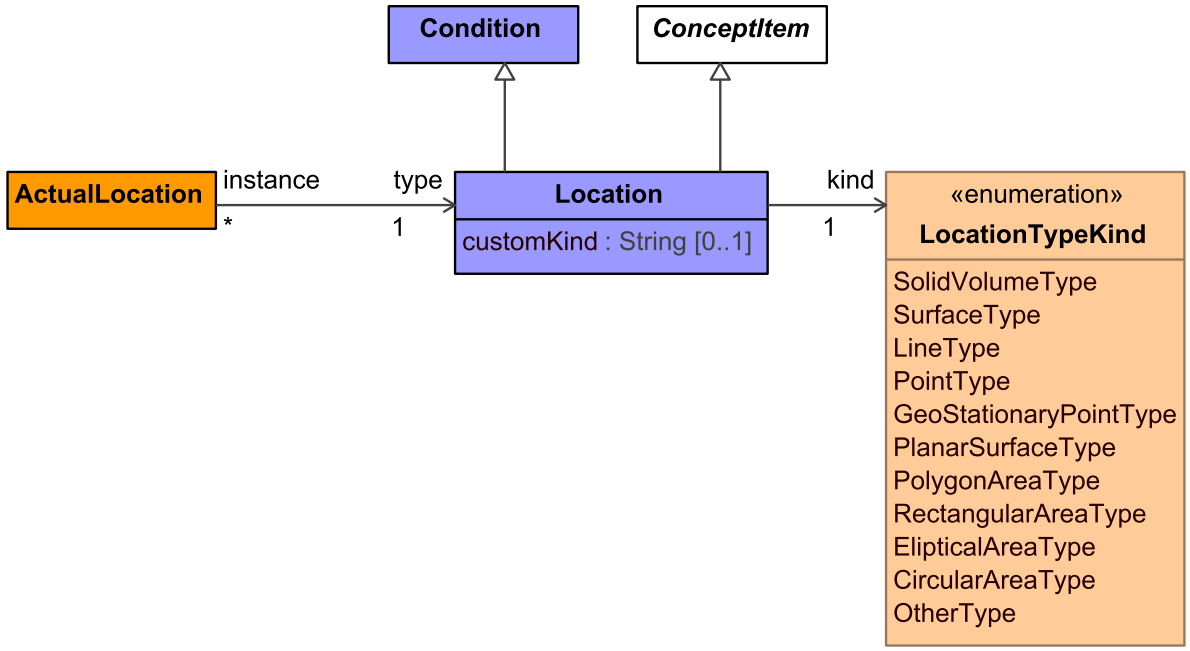

Figure 9:273 - Location

Attributes

customKind : String [0..1] Captures the kind of Location if the LocationTypeKind has been set to "OtherType".

LocationHolder

Package: Parameters
isAbstract: Yes
Generalization: UAFElement

Description

Abstract type, used to group elements that are allowed to be associated with a Location.

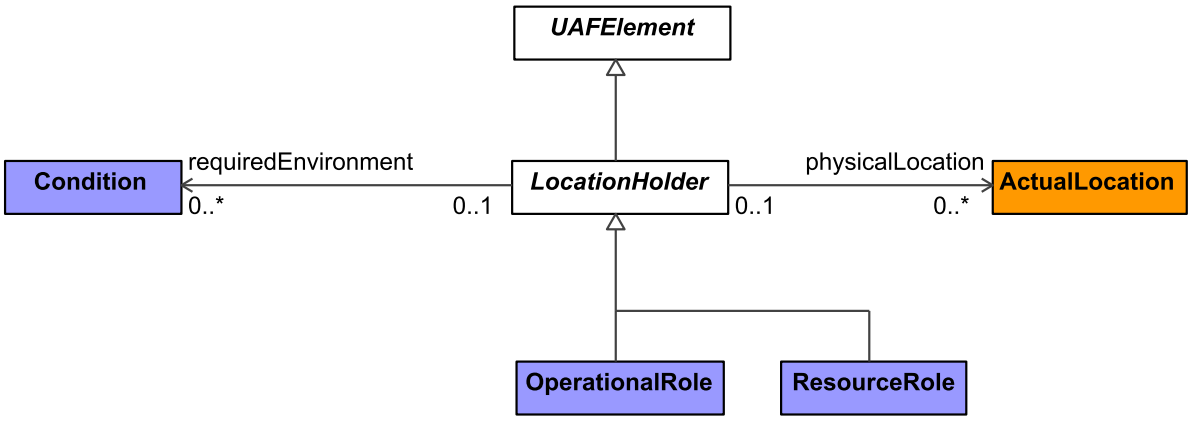

Figure 9:274 - LocationHolder

MeasurableElement

Package: Parameters
isAbstract: Yes
Generalization: UAFElement

Description

Abstract type, grouping elements that can be measured by applying MeasurementSets to them.

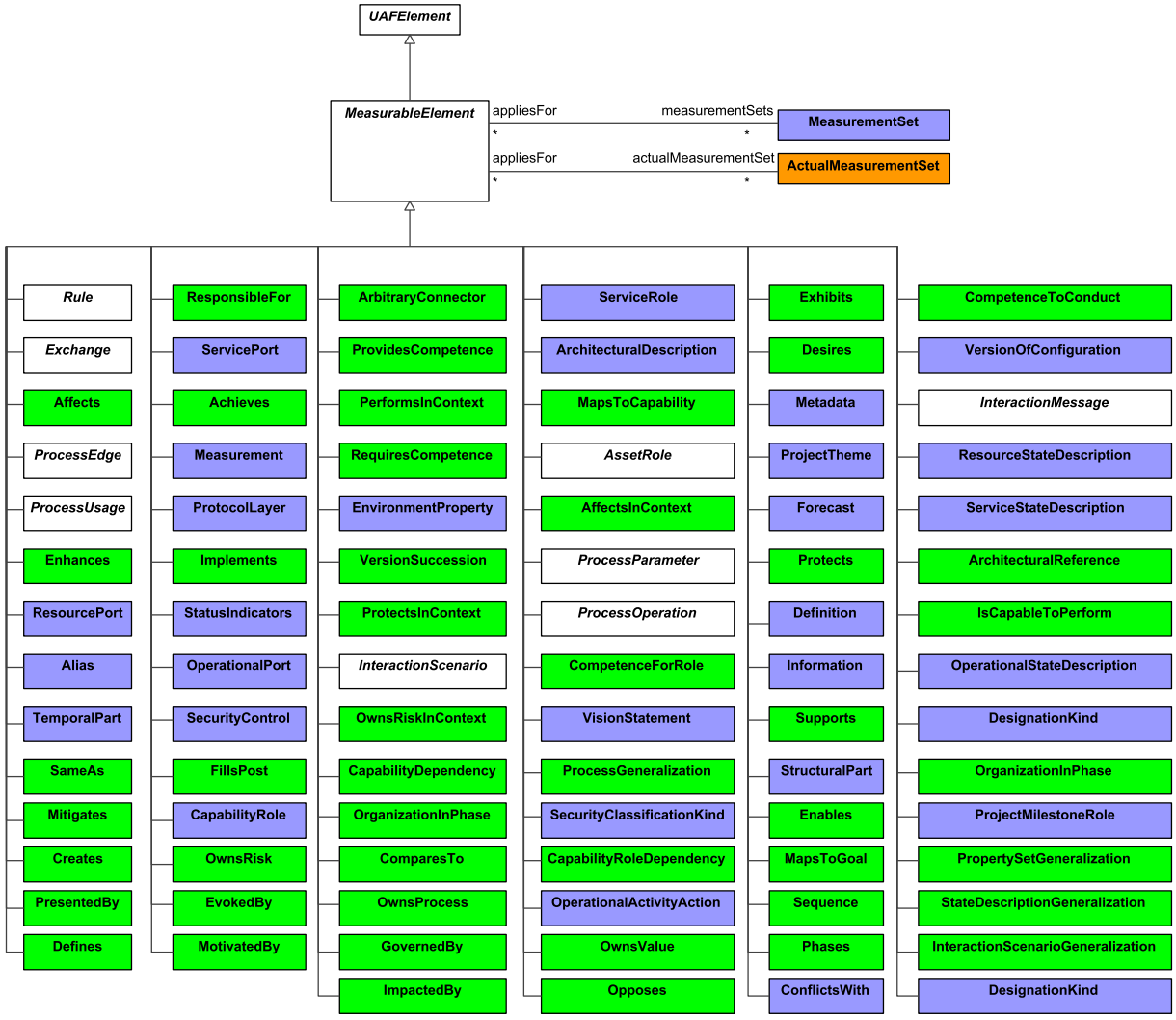

Figure 9:275 - MeasurableElement

Measurement

Package: Parameters
isAbstract: No
Generalization: MeasurableElement

Description

A property of an element representing something in the physical world, expressed in amounts of a unit of measure.

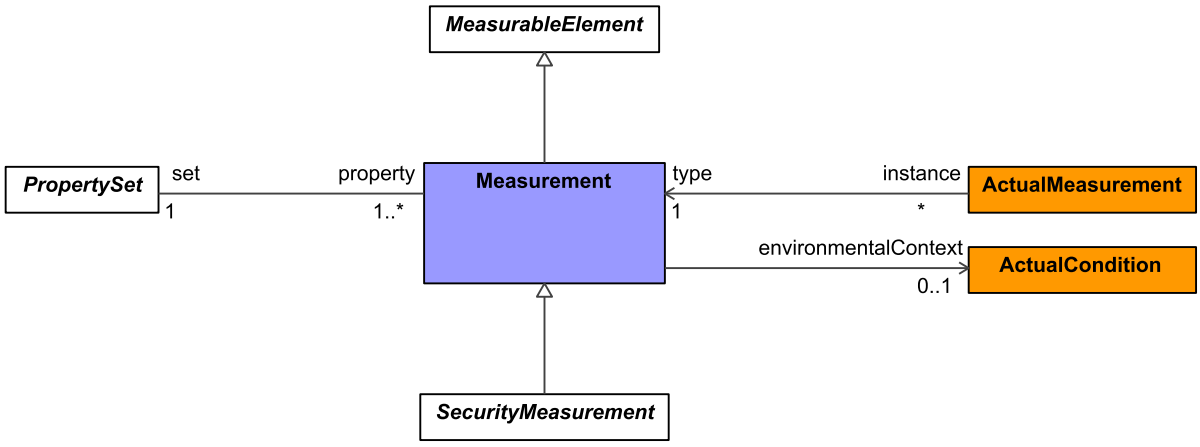

Figure 9:276 - Measurement

MeasurementSet

Package: Parameters
isAbstract: No
Generalization: PropertySet

Description

A collection of Measurements.

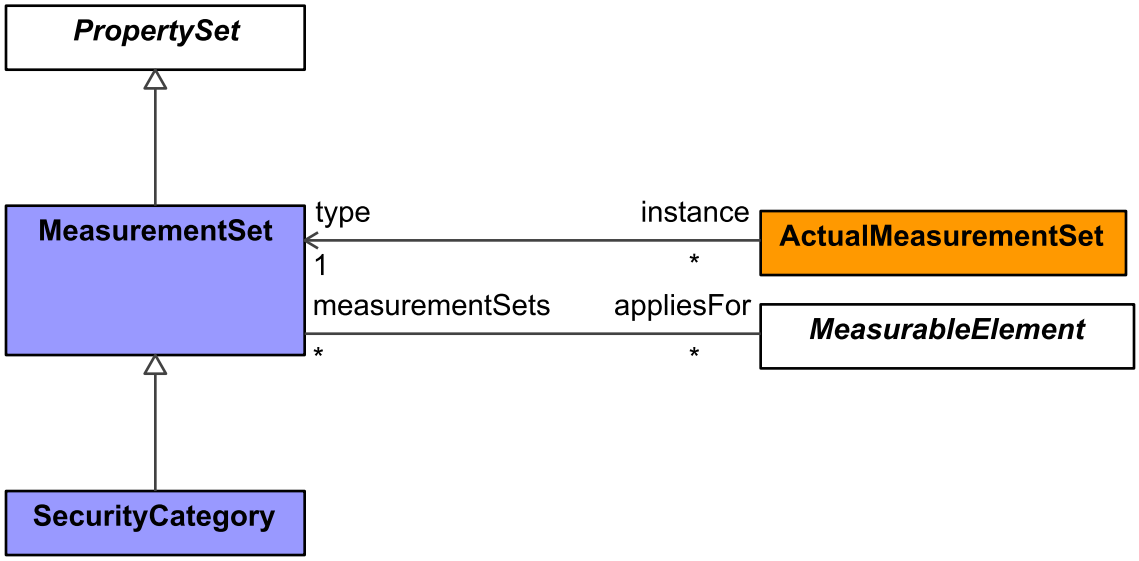

Figure 9:277 - MeasurementSet

MissionScenario

Package: Parameters
isAbstract: No
Generalization: Condition

Description

Description of the geographical location and time frame of the overall conflict. A scenario includes information such as threat and friendly politico-military contexts and backgrounds, assumptions, constraints, limitations, strategic objectives, and other planning considerations. (OUSD(R&E))

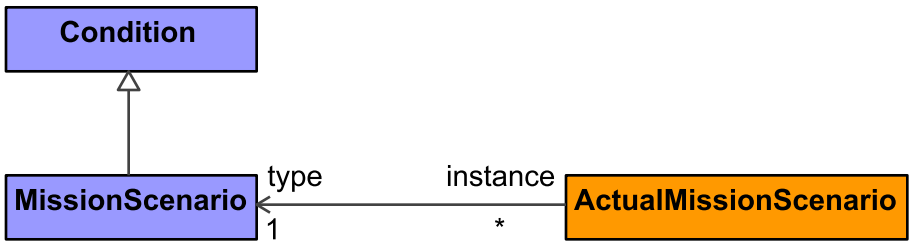

Figure 9:278 - MissionScenario

MissionVignette

Package: Parameters
isAbstract: No
Generalization: Condition

Description

Description of the geographical location and time frame of the overall conflict. A vignette includes information such as threat and friendly politico-military contexts and backgrounds, assumptions, constraints, limitations, strategic objectives, and other planning considerations. It can represent small, ideally self-contained parts of a scenario. (OUSD(R&E))

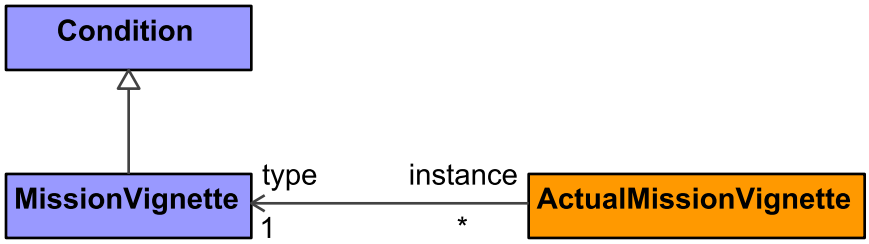

Figure 9:279 - MissionVignette

Mitigates

Package: Parameters
isAbstract: No
Generalization: MeasurableElement

Description

A tuple relating a Security Control to a Risk. Mitigation is established to manage risk and could be represented as an overall strategy or through techniques (mitigation configurations) and procedures (SecurityProcesses).

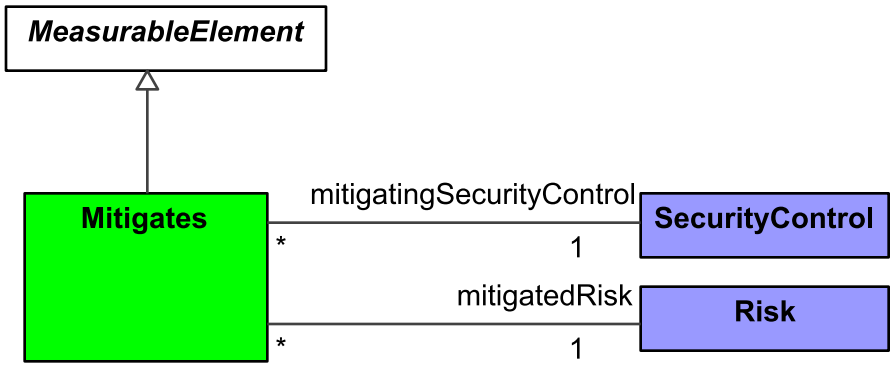

Figure 9:280 - Mitigates

OwnsRisk

Package: Parameters
isAbstract: No
Generalization: MeasurableElement

Description

A tuple relating a Risk to an organizational resource that is responsible for executing the risk mitigation.

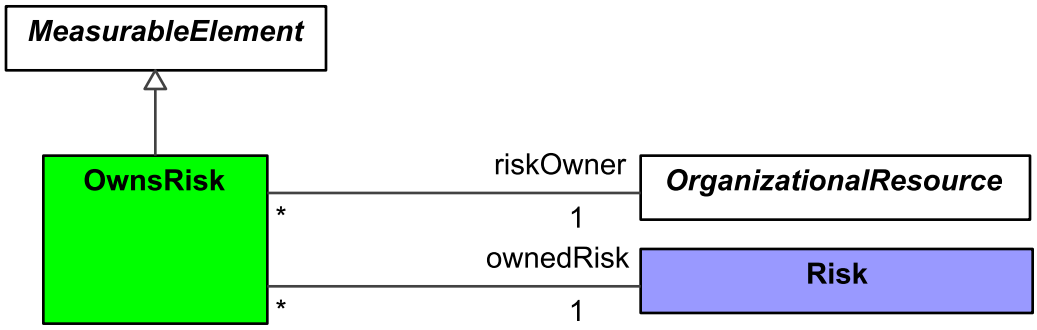

Figure 9:281 - OwnsRisk

OwnsRiskInContext

Package: Parameters
isAbstract: No
Generalization: MeasurableElement

Description

A tuple relating a Risk to an organizational role that is responsible for executing the risk mitigation in the specific context or configuration.

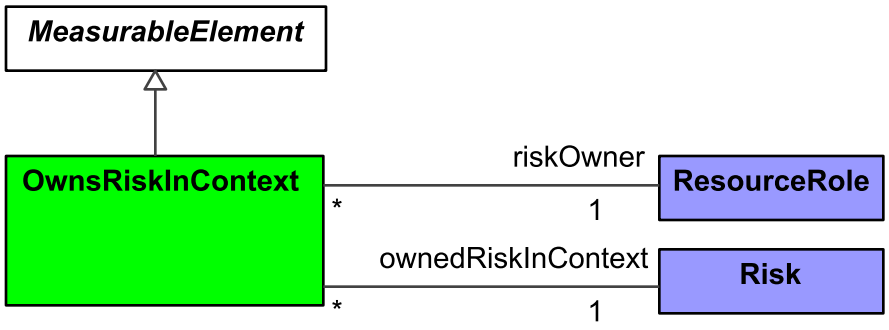

Figure 9:282 - OwnsRiskInContext

PropertySet

Package: Parameters
isAbstract: Yes
Generalization: UAFElement

Description

An abstract type grouping architectural elements that can own Measurements.

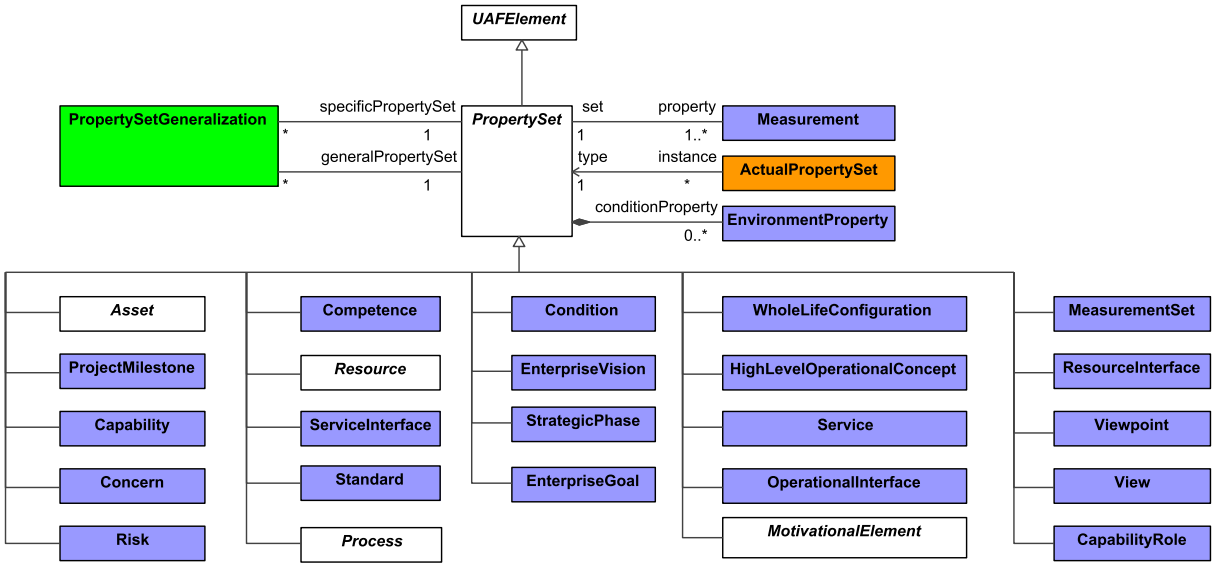

Figure 9:283 - PropertySet

Risk

Package: Parameters
isAbstract: No
Generalization: PropertySet

Description

A type that represents a situation involving exposure to danger of AffectableElements (e.g. Assets, Processes, Capabilities, Opportunities, or Enterprise Goals) where the effects of such exposure can be characterized in terms of the likelihood of occurrence of a given threat and the potential adverse consequences of that threat's occurrence.

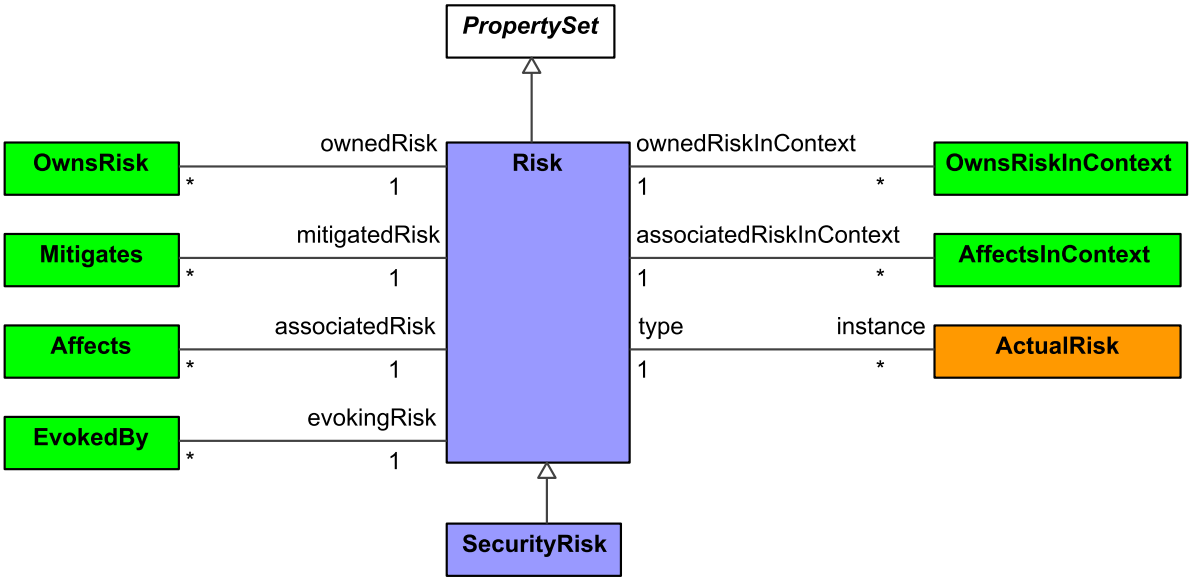

Figure 9:284 - Risk