INTERNATIONAL STANDARD

ISO/IEC 19540-1

First edition
2022-03

Information technology — Object Management Group Unified Architecture Framework (OMG UAF) —

Part 1: Domain Metamodel (DMM)

Reference number
ISO/IEC 19540-1:2022(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

FOREWORD ..... **xii**

PREFACE ..... **xiii**

1 SCOPE **1**

1.1 Introduction 1
1.2 UAF Background 1
1.3 Intended Usage 2
1.4 Related Documents 2

2 CONFORMANCE **3**

3 REFERENCES **4**

3.1 Normative References 4
3.2 OMG Documents (Normative References) 4
3.3 Other Normative References 4
3.4 Informative References 5

4 TERMS AND DEFINITIONS **6**

5 SYMBOLS **7**

6 ADDITIONAL INFORMATION **9**

6.1 Changes to Adopted OMG Specifications 9
6.2 Language Architecture 9
6.3 Philosophy 9
6.4 Core Principles 9

7 UAF GRID **11**

7.1 Descriptions of Domains and Model Kinds 12
7.2 Domain Interrelationships 13
7.3 Domain Metamodel Diagram Legend 14

8 DOMAIN METAMODEL DIAGRAMS **15**

8.1 View Specifications 15
8.1.1 View Specifications::Metadata 15
8.1.2 View Specifications::Strategic 16
8.1.3 View Specifications::Operational 24
8.1.4 View Specifications::Services 34
8.1.5 View Specifications::Personnel 43
8.1.6 View Specifications::Resources 56
8.1.7 View Specifications::Security 68
8.1.8 View Specifications::Projects 76
8.1.9 View Specifications::Standards 81
8.1.10 View Specifications::Actual Resources 84
8.1.11 View Specifications::Dictionary 87
8.1.12 View Specifications::Summary & Overview 88
8.1.13 View Specifications::Requirements 89
8.1.14 View Specifications::Information 90
8.1.15 View Specifications::Parameters 90
8.1.16 View Specifications::Other 93

9 DOMAIN METAMODEL (DMM) ELEMENTS 96

9.1 DOMAIN METAMODEL 96
9.1.1 Domain MetaModel::Metadata 96
9.1.2 Domain MetaModel::Strategic 109
9.1.3 Domain MetaModel::Operational 120
9.1.4 Domain MetaModel::Services 134
9.1.5 Domain MetaModel::Personnel 142
9.1.6 Domain MetaModel::Resources 149
9.1.7 Domain MetaModel::Security 166
9.1.8 Domain MetaModel::Projects 181
9.1.9 Domain MetaModel::Standards 188
9.1.10 Domain MetaModel::Actual Resources 190
9.1.11 Domain MetaModel::Dictionary 196
9.1.12 Domain MetaModel::Summary & Overview 198
9.1.13 Domain MetaModel::Information 202
9.1.14 Domain MetaModel::Parameters 203

TABLE OF FIGURES

Figure 7:1- UAF Grid 11
Figure 7:2 - Domain Interrelationships 13
Figure 7:3 - Legend of color codes for element types defined in UAF 14
Figure 8:2 - Strategic Structure 18
Figure 8:3 - Strategic Connectivity 19
Figure 8:4 - Strategic States 20
Figure 8:5 - Strategic Constraints 21
Figure 8:6 - Strategic Roadmap: Deployment 22
Figure 8:7 - Strategic Roadmap: Phasing 23
Figure 8:8 - Strategic Traceability 24
Figure 8:9 - Operational Taxonomy 25
Figure 8:10 - Operational Structure 26
Figure 8:11 - Operational Connectivity 27
Figure 8:12 - Operational Processes 28
Figure 8:13 - Operational Processes BPMN Semantics 29
Figure 8:14 - Operational States 31
Figure 8:15 - Operational Interaction Scenarios 32
Figure 8:16 - Operational Constraints 33
Figure 8:17 - Operational Traceability 34
Figure 8:18 - Services Taxonomy 35
Figure 8:19 - Services Structure 36
Figure 8:20 - Services Connectivity 37
Figure 8:21 - Services Processes 38
Figure 8:22 - Services Processes BPMN Semantics 39
Figure 8:23 - Services States 40
Figure 8:24 - Services Interaction Scenarios 41
Figure 8:25 - Services Constraints 42
Figure 8:26 - Services Roadmap 42
Figure 8:27 - Services Traceability 43
Figure 8:28 - Personnel Taxonomy 44
Figure 8:29 - Personnel Structure 45
Figure 8:30 - Personnel Connectivity 46
Figure 8:31 - Personnel Processes 47
Figure 8:32 - Personnel States 48
Figure 8:33 - Personnel Interaction Scenarios 49
Figure 8:34 - Personnel Constraints: Competence 50
Figure 8:35 - Personnel Constraints: Drivers 51
Figure 8:36 - Personnel Constraints: Performance 52
Figure 8:37 - Personnel Roadmap: Availability 53
Figure 8:38 - Personnel Roadmap: Evolution 54
Figure 8:39 - Personnel Roadmap: Forecast 55
Figure 8:40 - Personnel Traceability 56
Figure 8:41 - Resources Taxonomy 57
Figure 8:42 - Resources Structure 58
Figure 8:43 - Resources Connectivity 59
Figure 8:44 - Resources Processes 60
Figure 8:45 - Resources Processes BPMN Semantics 62
Figure 8:46 - Resources States 63
Figure 8:47 - Resources Interaction Scenarios 64
Figure 8:48 - Resources Constraints 65
Figure 8:49 - Resources Roadmap: Evolution 66
Figure 8:50 - Resources Roadmap: Forecast 67
Figure 8:51 - Resources Traceability 68
Figure 8:52 - Security Taxonomy 69
Figure 8:53 - Security Structure 70
Figure 8:54 - Security Connectivity 71
Figure 8:55 - Security Processes 72
Figure 8:56 - Security Constraints 74
Figure 8:57 - Security Traceability 75

Figure 8:58 - Project Taxonomy 76
Figure 8:59 - Project Structure 77
Figure 8:60 - Project Connectivity 78
Figure 8:61 - Project Processes 79
Figure 8:62 - Project Roadmap 80
Figure 8:63 - Project Traceability 81
Figure 8:64 - Standards Taxonomy 82
Figure 8:65 - Standards Structure 83
Figure 8:66 - Standards Roadmap 83
Figure 8:67 - Standards Traceability 84
Figure 8:68 - Actual Resources Structure 85
Figure 8:69 - Actual Resources Connectivity 86
Figure 8:70 - Actual Resources Traceability 86
Figure 8:71 - Dictionary 87
Figure 8:72 - Summary & Overview 88
Figure 8:73 - Requirements 89
Figure 8:74 - Information Model 90
Figure 8:75 - Parameters: Environment 91
Figure 8:76 - Parameters: Measurements 92
Figure 8:77 - BPMN 93
Figure 8:78 - IEPPV 94
Figure 8:79 - NIEM 95
Figure 9:1 - ArchitectureMetadata 96
Figure 9:2 - InteractionScenarioGeneralization 97
Figure 9:3 - Metadata 97
Figure 9:4 - ProcessGeneralization 98
Figure 9:5 - PropertySetGeneralization 98
Figure 9:6 - StateDescriptionGeneralization 99
Figure 9:7 - EnvironmentProperty 99
Figure 9:8 - Exchange 100
Figure 9:9 - ExchangeItem 100
Figure 9:10 - Resource 101
Figure 9:11 - ActivityPerformableUnderCondition 101
Figure 9:12 - IsCapableToPerform 102
Figure 9:13 - PerformsInContext 102
Figure 9:14 - Process 103
Figure 9:15 - Process 103
Figure 9:16 - ProcessEdge 104
Figure 9:17 - ProcessOperation 104
Figure 9:18 - ProcessParameter 104
Figure 9:19 - ProcessUsage 105
Figure 9:20 - StateDescription 105
Figure 9:21 - InteractionRole 106
Figure 9:22 - InteractionScenario 107
Figure 9:23 - Information 107
Figure 9:24 - Rule 108
Figure 9:25 - ArchitecturalReference 108
Figure 9:26 - Implements 109
Figure 9:27 - Capability 110
Figure 9:28 - CapabilityGeneralization 110
Figure 9:29 - ActualEnduringTask 111
Figure 9:30 - ActualEnterprisePhase 111
Figure 9:31 - CapabilityRole 112
Figure 9:32 - EnduringTask 112
Figure 9:33 - EnterpriseGoal 112
Figure 9:34 - EnterprisePhase 113
Figure 9:35 - EnterpriseVision 113
Figure 9:36 - StructuralPart 114
Figure 9:37 - TemporalPart 114
Figure 9:38 - VisionStatement 114
Figure 9:39 - WholeLifeEnterprise 115

Figure 9:40 - CapabilityDependency 115
Figure 9:41 - CapabilityRoleDependency 116
Figure 9:42 - AchievedEffect 116
Figure 9:43 - Achiever 117
Figure 9:44 - DesiredEffect 117
Figure 9:45 - Desirer 118
Figure 9:46 - CapabilityForTask 118
Figure 9:47 - CapableElement 119
Figure 9:48 - Exhibits 119
Figure 9:49 - MapsToCapability 120
Figure 9:50 - OrganizationInEnterprise 120
Figure 9:51 - ArbitraryConnector 121
Figure 9:52 - ConceptItem 121
Figure 9:53 - HighLevelOperationalConcept 122
Figure 9:54 - KnownResource 122
Figure 9:55 - OperationalAgent 123
Figure 9:56 - OperationalArchitecture 123
Figure 9:57 - OperationalMethod 124
Figure 9:58 - OperationalParameter 124
Figure 9:59 - OperationalPerformer 125
Figure 9:60 - OperationalRole 125
Figure 9:61 - ProblemDomain 126
Figure 9:62 - OperationalConnector 126
Figure 9:63 - OperationalExchange 127
Figure 9:64 - OperationalExchangeItem 127
Figure 9:65 - OperationalInterface 128
Figure 9:66 - OperationalPort 128
Figure 9:67 - OperationalSignal 129
Figure 9:68 - OperationalActivity 129
Figure 9:69 - OperationalActivityAction 130
Figure 9:70 - OperationalActivityEdge 130
Figure 9:71 - StandardOperationalActivity 131
Figure 9:72 - OperationalStateDescription 131
Figure 9:73 - OperationalInteractionScenario 132
Figure 9:74 - OperationalMessage 132
Figure 9:75 - InformationElement 133
Figure 9:76 - OperationalConstraint 133
Figure 9:77 - SubjectOfOperationalConstraint 134
Figure 9:78 - ServiceSpecification 134
Figure 9:79 - ServiceSpecificationGeneralization 135
Figure 9:80 - ServiceConnector 135
Figure 9:81 - ServiceMethod 136
Figure 9:82 - ServiceParameter 136
Figure 9:83 - ServiceSpecificationRole 137
Figure 9:84 - ServiceInterface 137
Figure 9:85 - ServicePort 138
Figure 9:86 - ServiceFunction 138
Figure 9:87 - ServiceFunctionAction 139
Figure 9:88 - ServiceFunctionEdge 139
Figure 9:89 - ServiceStateDescription 140
Figure 9:90 - ServiceInteractionScenario 140
Figure 9:91 - ServiceMessage 141
Figure 9:92 - ServicePolicy 141
Figure 9:93 - Consumes 142
Figure 9:94 - Organization 142
Figure 9:95 - OrganizationalResource 143
Figure 9:96 - Person 143
Figure 9:97 - Post 144
Figure 9:98 - Responsibility 144
Figure 9:99 - PostRole 145
Figure 9:100 - SubOrganization 145

Figure 9:101 - Command....146
Figure 9:102 - Control....146
Figure 9:103 - ResourceInteractionScenario....147
Figure 9:104 - Competence....147
Figure 9:105 - CompetenceForRole....148
Figure 9:106 - RequiresCompetence....148
Figure 9:107 - FillsPost....149
Figure 9:108 - CompetenceToConduct....149
Figure 9:109 - CapabilityConfiguration....150
Figure 9:110 - NaturalResource....150
Figure 9:111 - PhysicalResource....150
Figure 9:112 - ResourceArchitecture....151
Figure 9:113 - ResourceArtifact....151
Figure 9:114 - ResourcePerformer....152
Figure 9:115 - Software....152
Figure 9:116 - System....153
Figure 9:117 - ResourceMethod....153
Figure 9:118 - ResourceParameter....154
Figure 9:119 - ResourcePort....154
Figure 9:120 - ResourceRole....155
Figure 9:121 - ResourceConnector....156
Figure 9:122 - ResourceExchange....156
Figure 9:123 - ResourceExchangeItem....157
Figure 9:124 - ResourceInterface....158
Figure 9:125 - ResourceSignal....158
Figure 9:126 - Function....159
Figure 9:127 - FunctionAction....159
Figure 9:128 - FunctionEdge....160
Figure 9:129 - ResourceStateDescription....160
Figure 9:130 - ResourceMessage....161
Figure 9:131 - DataElement....161
Figure 9:132 - DataRole....162
Figure 9:133 - ResourceConstraint....162
Figure 9:134 - SubjectOfResourceConstraint....162
Figure 9:135 - Forecast....163
Figure 9:136 - SubjectOfForecast....163
Figure 9:137 - Technology....164
Figure 9:138 - VersionedElement....164
Figure 9:139 - VersionOfConfiguration....165
Figure 9:140 - VersionSuccession....165
Figure 9:141 - WholeLifeConfiguration....166
Figure 9:142 - ProtocolImplementation....166
Figure 9:143 - Asset....167
Figure 9:144 - OperationalAsset....167
Figure 9:145 - OperationalMitigation....168
Figure 9:146 - ResourceAsset....168
Figure 9:147 - ResourceMitigation....168
Figure 9:148 - SecurityEnclave....169
Figure 9:149 - AssetRole....169
Figure 9:150 - InformationRole....170
Figure 9:151 - EnhancedSecurityControl....170
Figure 9:152 - Enhances....171
Figure 9:153 - Protects....171
Figure 9:154 - ProtectsInContext....172
Figure 9:155 - SecurityProcess....172
Figure 9:156 - SecurityProcessAction....173
Figure 9:157 - ActualRisk....173
Figure 9:158 - Caveat....174
Figure 9:159 - Risk....174
Figure 9:160 - SecurityAvailability....174
Figure 9:161 - SecurityCategory....175

Figure 9:162 - SecurityClassification 175
Figure 9:163 - SecurityClassificationKind 176
Figure 9:164 - SecurityConstraint 176
Figure 9:165 - SecurityControl 177
Figure 9:166 - SecurityControlFamily 177
Figure 9:167 - SecurityIntegrity 178
Figure 9:168 - SecurityMeasurement 178
Figure 9:169 - SubjectOfSecurityConstraint 179
Figure 9:170 - Affects 179
Figure 9:171 - AffectsInContext 179
Figure 9:172 - Mitigates 180
Figure 9:173 - OwnsRisk 180
Figure 9:174 - OwnsRiskInContext 181
Figure 9:175 - Project 181
Figure 9:176 - ProjectMilestone 182
Figure 9:177 - ActualProjectMilestoneRole 182
Figure 9:178 - ProjectMilestoneRole 183
Figure 9:179 - ProjectStatus 183
Figure 9:180 - ProjectTheme 184
Figure 9:181 - StatusIndicators 184
Figure 9:182 - MilestoneDependency 185
Figure 9:183 - ProjectActivity 185
Figure 9:184 - ProjectActivityAction 185
Figure 9:185 - ProjectSequence 186
Figure 9:186 - ActualProject 186
Figure 9:187 - ActualProjectMilestone 187
Figure 9:188 - ResponsibleFor 187
Figure 9:189 - Protocol 188
Figure 9:190 - ProtocolStack 188
Figure 9:191 - Standard 189
Figure 9:192 - ProtocolLayer 189
Figure 9:193 - ActualOrganization 190
Figure 9:194 - ActualOrganizationalResource 191
Figure 9:195 - ActualPerson 191
Figure 9:196 - ActualPost 192
Figure 9:197 - ActualResource 192
Figure 9:198 - ActualResourceRelationship 193
Figure 9:199 - ActualResponsibility 193
Figure 9:200 - ActualResponsibleResource 194
Figure 9:201 - FieldedCapability 194
Figure 9:202 - ActualService 195
Figure 9:203 - ProvidedServiceLevel 195
Figure 9:204 - ProvidesCompetence 195
Figure 9:205 - RequiredServiceLevel 196
Figure 9:206 - Alias 196
Figure 9:207 - Definition 197
Figure 9:208 - SameAs 197
Figure 9:209 - ArchitecturalDescription 198
Figure 9:210 - Architecture 199
Figure 9:211 - Concern 199
Figure 9:212 - Stakeholder 200
Figure 9:213 - UAFElement 201
Figure 9:214 - View 201
Figure 9:215 - Viewpoint 202
Figure 9:216 - DataModel 202
Figure 9:217 - ActualCondition 203
Figure 9:218 - ActualEnvironment 203
Figure 9:219 - ActualLocation 204
Figure 9:220 - ActualMeasurement 205
Figure 9:221 - ActualMeasurementSet 205
Figure 9:222 - ActualPropertySet 206

Figure 9:223 - ActualState....206
Figure 9:224 - Condition....207
Figure 9:225 - Environment....207
Figure 9:226 - GeoPoliticalExtentType....208
Figure 9:227 - ISO8601DateTime....208
Figure 9:228 - Location....209
Figure 9:229 - LocationHolder....209
Figure 9:230 - MeasurableElement....210
Figure 9:231 - Measurement....211
Figure 9:232 - MeasurementSet....211
Figure 9:233 - PropertySet....212

TABLE OF TABLES

Table 1:1 - Table of Related Documents 2
Table 5:1 - Description of acronyms used in this specification 7
Table 7:1 - Definitions for the Domains 12
Table 7:2 - Definitions of the Model Kinds 13

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by the Object Management Group (OMG) (as Unified Architecture Framework [UAF] Domain Metamodel, Version 1.1) and drafted in accordance with its editorial rules. It was adopted, under the JTC 1 PAS procedure, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

A list of all parts in the ISO/IEC 19540 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

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 *https://www.omg.org/*.

OMG Specifications

As noted, OMG specifications address middleware, modeling and vertical domain frameworks. All OMG Specifications are available from this URL: *https://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*

Information technology — Object Management Group Unified Architecture Framework (OMG UAF) —

Part 1: Domain Metamodel (DMM)

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 Profile (UAFP) (see document dtc/19-06-15) is a UML/SysML implementation of the UAF DMM

The informative parts are:

3. The UAF Traceability, Annex A (see document dtc/19-06-17), 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/19-06-18), which illustrates a practical usage of UAF.

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.

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

The UAFP 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 1.1 supports the capability to:

- model architectures for a broad range of complex systems, which may include hardware, software, data, personnel, and facility elements,
- 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 metatmodel 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/19-06-16	The UAF Domain MetaModel (DMM)
dtc/19-06-15	The UAF Profile (UAFP)
dtc/19-06-17	Appendix A that contains a separate traceability subsection from UAFP to each of the frameworks listed in Section 1.2 of this specification
dtc/19-06-18	Appendix B: An example of how the language can be used to represent a UAFP architecture
dtc/19-06-19	UAF XMI file
dtc/19-06-20	UAF XMI Measurements library
dtc/19-05-14	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 Metadata Domain. 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/19-06-17)

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 UAFP stereotypes defined in the UAFP (dtc/19-06-15)
- complies with the constraints defined in the UAFP (dtc/19-06-15)
- complies with the SysML version 1.5 Concrete Syntax Conformance (formal/17-05-01)

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 UAFP 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.

3.2 OMG Documents (Normative References)

- Unified Modeling Language (UML), 2.5.1, December 2017, https://www.omg.org/spec/UML
- Object Constraint Language (OCL), 2.4, February 2014, https://www.omg.org/spec/OCL
- System Modeling Language (SysML) ,1.5, May 2017, https://www.omg.org/spec/SysML
- Diagram Definition (DD), 1.1, June 2015, https://www.omg.org/spec/DD
- UML Profile for the National Information Exchange Model (NIEM UML), 3.0, April 2017, https://www.omg.org/spec/NIEM-UML
- Unified Profile for DoDAF and MODAF (UPDM), 2.1, August 2013, https://www.omg.org/spec/UPDM
- UML Profile for BPMN Processes, 1.0, July 2014, https://www.omg.org/spec/BPMNProfile
- Ontology Definition Metamodel (ODM), 1.1, September 2014, https://www.omg.org/spec/ODM
- Information Exchange Packaging Policy Vocabulary (IEPPV) 1.0, May 2015, https://www.omg.org/spec/IEPPV

3.3 Other Normative References

- Department of Defense Architecture Framework (DoDAF), Version 2.02, August 2010, http://dodcio.defense.gov/Library/DoDArchitectureFramework.aspx
- DM2 - DoDAF Meta-Model,
- The DM2 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, http://www.ideasgroup.org/
- IDEAS Foundation, http://www.ideasgroup.org/foundation/
- IDEAS Foundation v1.0 as XMI File (zipped), http://www.ideasgroup.org/7Documents/
- 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), http://www.nhqc3s.nato.int/HomePage.asp (no longer publicly available online as of 3 November 2015)
- NATO Architecture Framework v4.0 Documentation.

3.4 Informative References

- Business Process Model & Notation (BPMN), Version 2.0.2, January 2014 https://www.omg.org/spec/BPMN
- 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, https://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 1.3, May 2015, https://www.omg.org/spec/SBVR
- Business Motivation Model (BMM), Version 1.3, http://www.omg.org/spec/BMM/1.3/
- International Council On Systems Engineering (INCOSE), Systems Engineering Handbook V4, 2015, http://www.incose.org/ProductsPublications/sehandbook

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.

5 Symbols

For the purposes of this specification, the following List of symbols/abbreviations apply.

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

[2] * denotes a wildcard

POC	Proof of Concept
PV-*	Project View
RDF	Resource Description Framework
SoaML	Service orientated architecture Modeling Language
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
UAFP	Unified Architecture Framework Profile
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 1.0
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 1.5 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 1.5 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 aggregate the concepts defined in DoDAF, MODEM, NAF, DNDAF 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/19-06-17 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 Profile (UAFP) is the standard implementation of the UAF DMM. It was created by mapping the UAF concepts and relationships to corresponding stereotypes in the UAFP.
- The UAFP 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 UAFP. 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 UAFP 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 viewpoints (known as *view specifications* in the rest of document) correspond to *domains* (horizontal rows) and the *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.

	Taxonomy Tx	Structure Sr	Connectivity Cn	Processes Pr	States St	Interaction Scenarios Is	Information[c] If	Parameters[d] Pm	Constraints Ct	Roadmap Rm	Traceability Tr
Metadata[a] Md	Metadata Taxonomy Md-Tx[f]	Metadata Structure Md-Sr	Metadata Connectivity Md-Cn	Metadata Processes Md-Pr	Metadata States Md-St	-			Metadata Constraints Md-Ct	Metadata Roadmap Md-Rm	Metadata Traceability Md-Tr
Strategic St	Strategic Taxonomy St-Tx	Strategic Structure St-Sr	Strategic Connectivity St-Cn	-	Strategic States St-St	-			Strategic Constraints St-Ct	Strategic Deployment, St-Rm Strategic Phasing St-Rm	Strategic Traceability St-Tr
Operational Op	Operational Taxonomy Op-Tx	Operational Structure Op-Sr	Operational Connectivity Op-Cn	Operational Processes Op-Pr	Operational States Op-St	Operational Interaction Scenarios Op-Is			Operational Constraints Op-Ct	-	Operational Traceability Op-Tr
Services Sv	Service Taxonomy Sv-Tx	Service Structure Sv-Sr	Service Connectivity Sv-Cn	Service Processes Sv-Pr	Service States Sv-St	Service Interaction Scenarios Sv-Is	Conceptual Data Model,	Environment Pm-En	Service Constraints Sv-Ct	Service Roadmap Sv-Rm	Service Traceability Sv-Tr
Personnel Pr	Personnel Taxonomy Pr-Tx	Personnel Structure Pr-Sr	Personnel Connectivity Pr-Cn	Personnel Processes Pr-Pr	Personnel States Pr-St	Personnel Interaction Scenarios Pr-Is	Logical Data Model,		Competence, Drivers, Performance Pr-Ct	Personnel Availability, Personnel Evolution, Personnel Forecast Pr-Rm	Personnel Traceability Pr-Tr
Resources Rs	Resource Taxonomy Rs-Tx	Resource Structure Rs-Sr	Resource Connectivity Rs-Cn	Resource Processes Rs-Pr	Resource States Rs-St	Resource Interaction Scenarios Rs-Is	Physical schema[e], real world results	Measurements Pm-Me	Resource Constraints Rs-Ct	Resource evolution, Resource forecast Rs-Rm	Resource Traceability Rs-Tr
Security Sc	Security Taxonomy Sc-Tx	Security Structure Sc-Sr	Security Connectivity Sc-Cn	Security Processes Sc-Pr	-	-			Security Constraints Sc-Ct	-	Security Traceability Sc-Tr
Projects Pj	Project Taxonomy Pj-Tx	Project Structure Pj-Sr	Project Connectivity Pj-Cn	Project Processes Pj-Pr	-	-			-	Project Roadmap Pj-Rm	Project Traceability Pj-Tr
Standards Sd	Standard Taxonomy Sd-Tx	Standards Structure Sd-Sr	-	-	-	-			-	Standards Roadmap Sd-Rm	Standards Traceability Sd-Tr
Actual Resources Ar		Actual Resources Structure Ar-Sr	Actual Resources Connectivity Ar-Cn	Simulation[b]					Parametric Execution/ Evaluation[b]	-	-
Dictionary Dc											
Summary & Overview Sm-Ov											
Requirements Req											

Figure 7:1- UAF Grid

Notes related to suffixes in the grid:

a. The view specifications in the Metadata Domain are not modeled as part of the UAF but 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 a column 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 and environments across the architecture in all the different domains.

e. The expectation is that the physical schema model would not be defined in the UAF. Any tool implementing the framework provides a means to import or link-to representations of the physical model.

f. The Metadata Taxonomy 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\2019-06-17. A definition for each view specification in the grid is described in the following chapters.

7.1 Descriptions of Domains and Model Kinds

Table 7:1 - Definitions for the Domains

Domain	Acronym	Description
Metadata	Md	Identifies the metadata 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	Pr	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 Model Kinds

Model Kind	Acronym	Description
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 definitions of the dependencies, connections, and relationships between the different elements.
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.
Interaction Scenarios	Is	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 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. Also, how at different points in time or different periods of 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 Domain Interrelationships

Although the gird is the primary means of expressing the relationship between the Domains, Model Kinds and View Specifications, because of it is two-dimensional nature it is not adequate to explain the abstract interrelationships that exist between the domains. The following diagram is an indication of how the domains are interrelated.

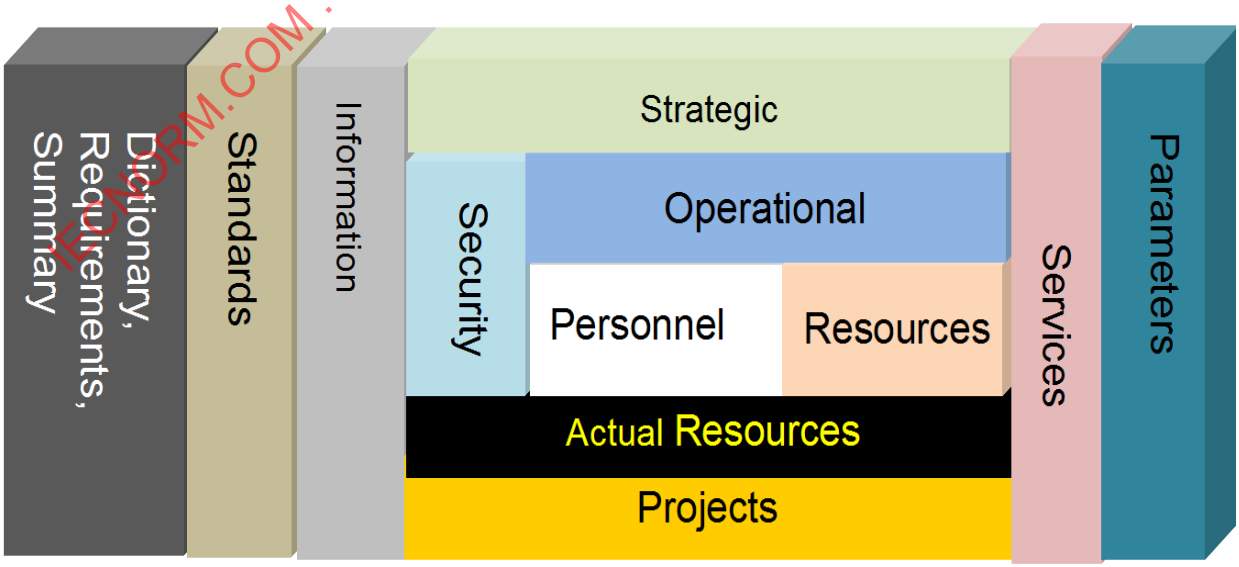

Figure 7:2 - Domain Interrelationships

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

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

7.3 Domain Metamodel Diagram Legend

This Annex comprises of various diagrams that document the Domain Metamodel (DMM) that document the MoDAF 1.5 and MoDAF 1.2 integrated model. This model was used as a basis for creating the UPDM profile.

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.

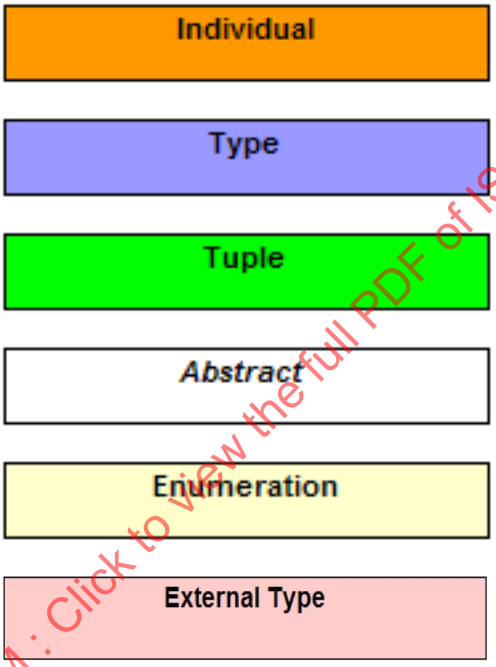

Figure 7-3 - Legend of color codes for element types defined in UAF

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 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::Metadata

Stakeholders: Enterprise Architects, Technical Managers.
Concerns: architecture development process, architecture traceability, metamodel and its extensions, architecture versioning.
Definition: Identifies the metadata required to develop a suitable architecture that is fit for its purpose.

View Specifications::Metadata::Taxonomy

Stakeholders: Enterprise Architects, Technical Managers.

Concerns: metamodel and its extensions.

Definition: captures user defined metamodel extensions

Recommended Implementation: UML Profile Diagram, SysML Block Definition Diagram

View Specifications::Metadata::Structure

Stakeholders: Enterprise Architects, Technical Managers.

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

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

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

View Specifications::Metadata::Connectivity

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.

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

View Specifications::Metadata::Processes

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

Concerns: methodology used.

Definition: methodology used in developing the architecture.

Recommended Implementation: SysML Activity Diagram, text.

View Specifications::Metadata::States

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.

View Specifications::Metadata::Constraints

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

Concerns: architectural constraints.

Definition: captures assumptions and constraints on the architecture.

Recommended Implementation: tabular format, text.

View Specifications::Metadata::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.

View Specifications::Metadata::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 asset libraries, legacy architectures, and external sources, e.g., documents.

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

8.1.2 View Specifications::Strategic

Stakeholders: Capability Portfolio Managers.
Concerns: capability management process.
Definition: describe capability taxonomy, composition, dependencies and evolution.

View Specifications::Strategic::Taxonomy

Contains the diagrams that document the Strategic Taxonomy Viewpoint.

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.

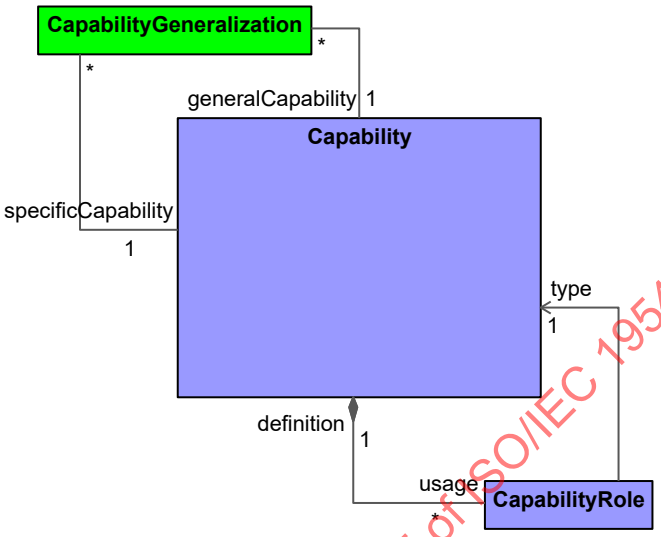

Figure 8:1 - Strategic Taxonomy

Elements

- Capability
- CapabilityGeneralization
- CapabilityRole

View Specifications::Strategic::Structure

Contains the diagrams that document the Strategic Structure Viewpoint.

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.

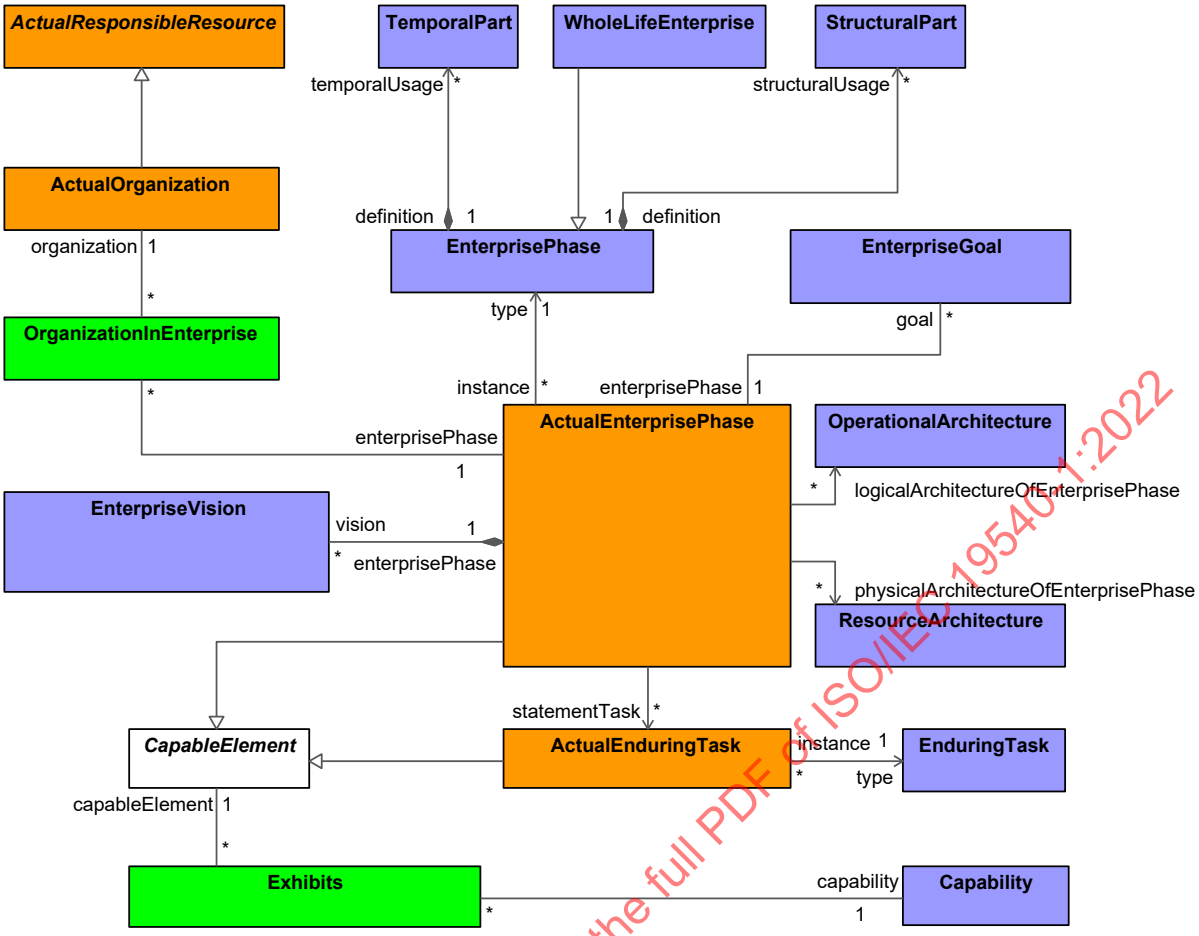

Figure 8:2 - Strategic Structure

Elements

- ActualEnduringTask
- ActualEnterprisePhase
- ActualOrganization
- ActualResponsibleResource
- Capability
- CapableElement
- EnduringTask
- EnterpriseGoal
- EnterprisePhase
- EnterpriseVision
- Exhibits
- OperationalArchitecture
- OrganizationInEnterprise
- ResourceArchitecture
- StructuralPart
- TemporalPart
- WholeLifeEnterprise

View Specifications::Strategic::Connectivity

Contains the diagrams that document the Strategic Connectivity Viewpoint.

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.

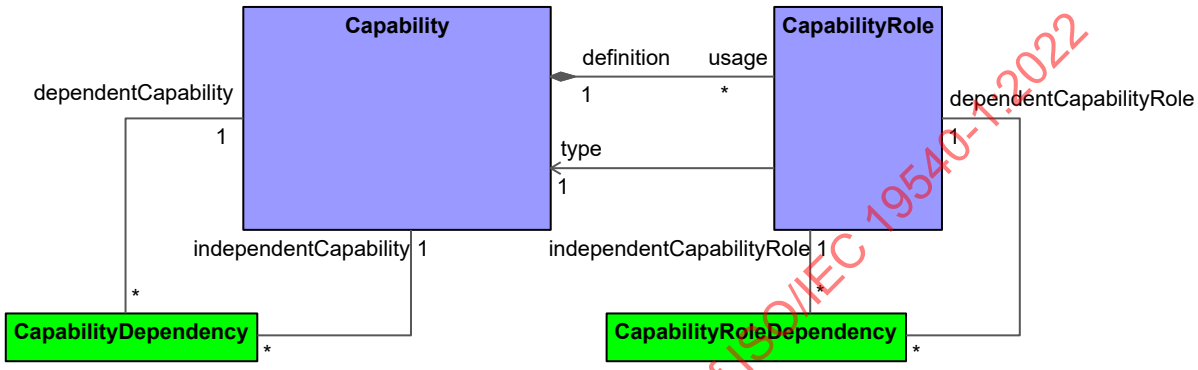

Figure 8:3 - Strategic Connectivity

Elements

- Capability
- CapabilityDependency
- CapabilityRole
- CapabilityRoleDependency

View Specifications::Strategic::States

Contains the diagrams that document the Strategic States Viewpoint.

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.

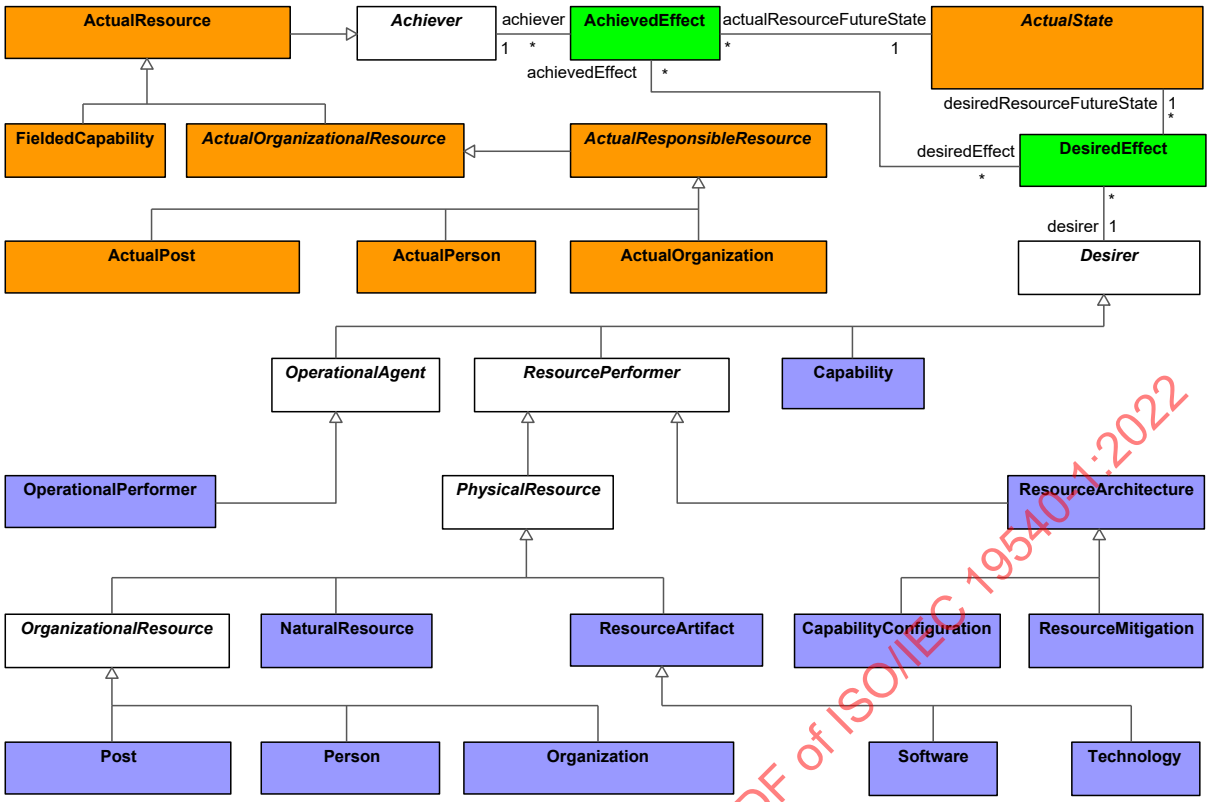

Figure 8:4 - Strategic States

Elements

- AchievedEffect
- Achiever
- ActualOrganization
- ActualOrganizationalResource
- ActualPerson
- ActualPost
- ActualResource
- ActualResponsibleResource
- ActualState
- Capability
- CapabilityConfiguration
- DesiredEffect
- Desirer
- FieldedCapability
- NaturalResource
- OperationalAgent
- OperationalPerformer
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- ResourceArchitecture
- ResourceArtifact
- ResourceMitigation
- ResourcePerformer
- Software
- Technology

View Specifications::Strategic::Constraints

Contains the diagrams that document the Strategic Constraints Viewpoint.

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.

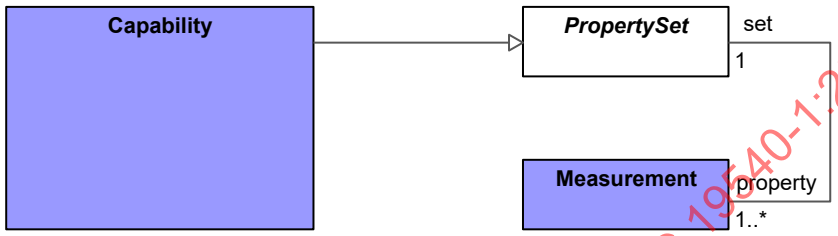

Figure 8:5 - Strategic Constraints

Elements

- Capability
- Measurement
- PropertySet

View Specifications::Strategic::Roadmap

Contains the diagrams that document the Strategic Roadmap Viewpoint.

View Specifications::Strategic::Roadmap::Deployment

View Specifications::Strategic::Roadmap::Deployment::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.

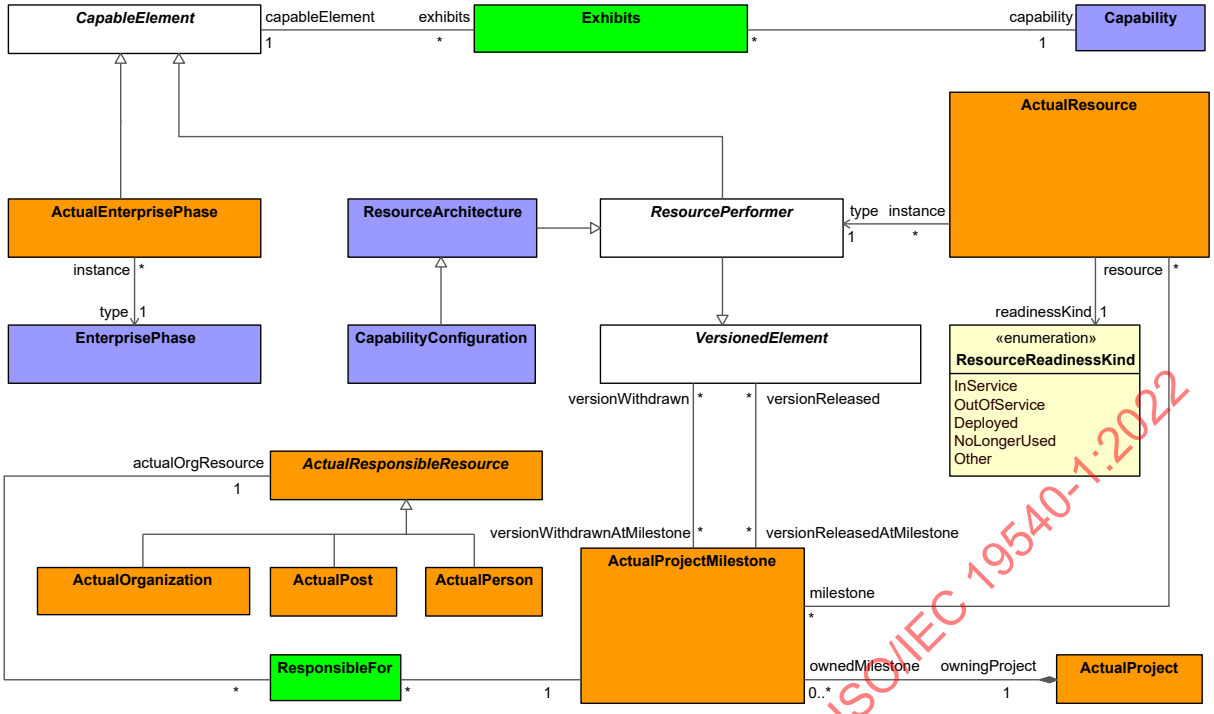

Figure 8:6 - Strategic Roadmap: Deployment

Elements

- ActualEnterprisePhase
- ActualOrganization
- ActualPerson
- ActualPost
- ActualProject
- ActualProjectMilestone
- ActualResource
- ActualResponsibleResource
- Capability
- CapabilityConfiguration
- CapableElement
- EnterprisePhase
- Exhibits
- ResourceArchitecture
- ResourcePerformer
- ResponsibleFor
- VersionedElement

View Specifications::Strategic::Roadmap::Phasing

View Specifications::Strategic::Roadmap::Phasing::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.

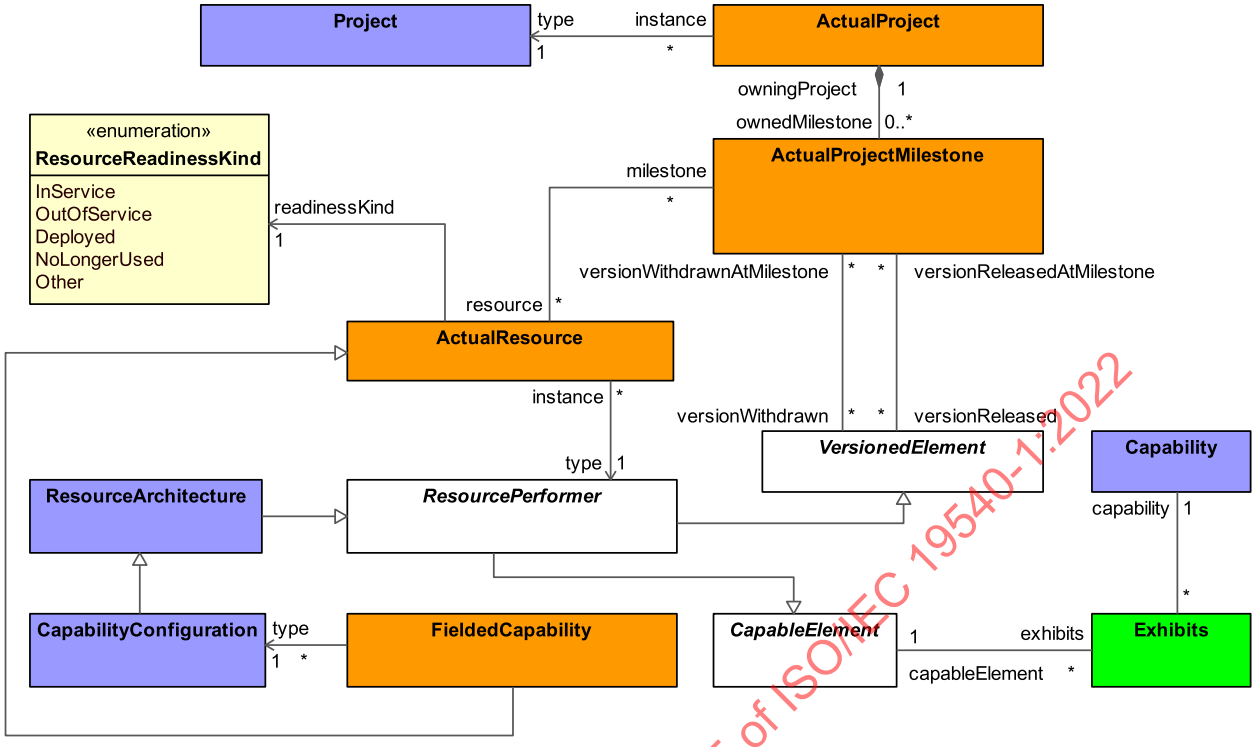

Figure 8:7 - Strategic Roadmap: Phasing

Elements

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

View Specifications::Strategic::Traceability

Contains the diagrams that document the Strategic Traceability Viewpoint.

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.

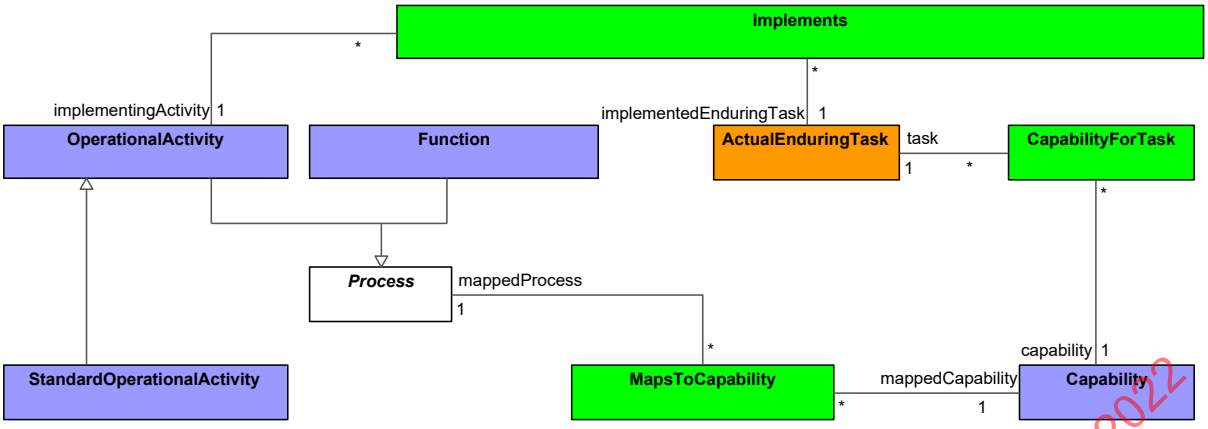

Figure 8:8 - Strategic Traceability

Elements

- ActualEnduringTask
- Capability
- CapabilityForTask
- Function
- Implements
- MapsToCapability
- OperationalActivity
- Process
- StandardOperationalActivity

8.1.3 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.

View Specifications::Operational::Taxonomy

Contains the diagrams that document the Operational Taxonomy Viewpoint.

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.

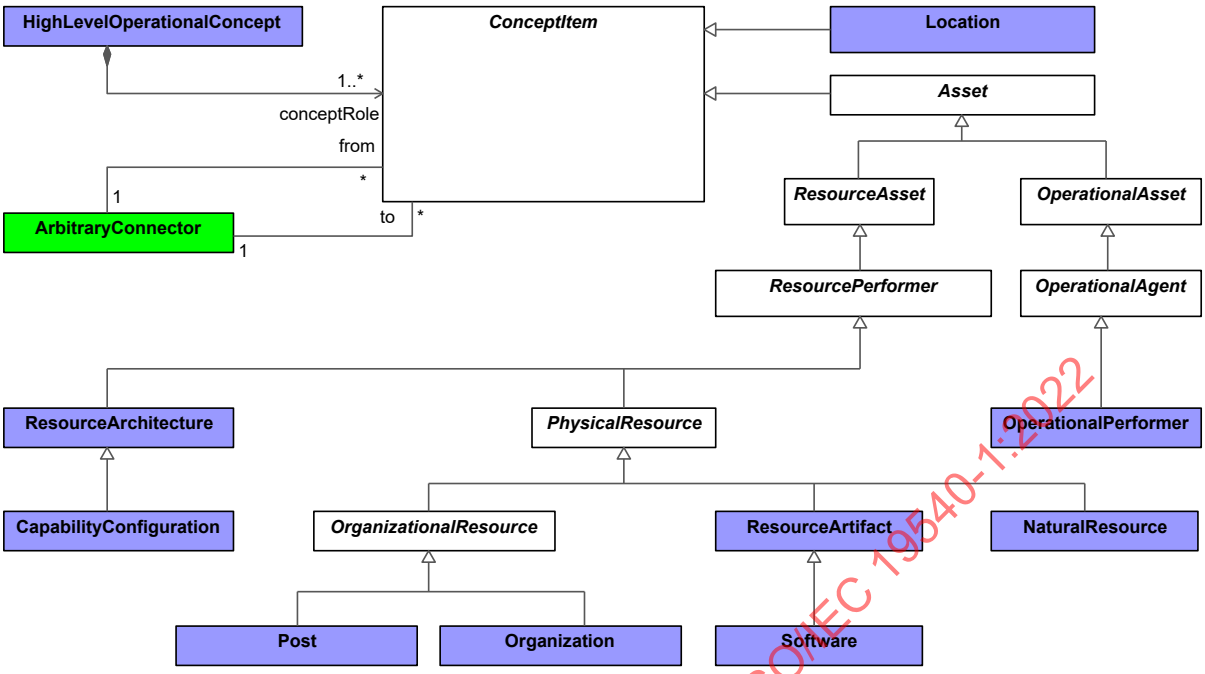

Figure 8:9 - Operational Taxonomy

Elements

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

View Specifications::Operational::Structure

Contains the diagrams that document the Operational Structure Viewpoint.

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.

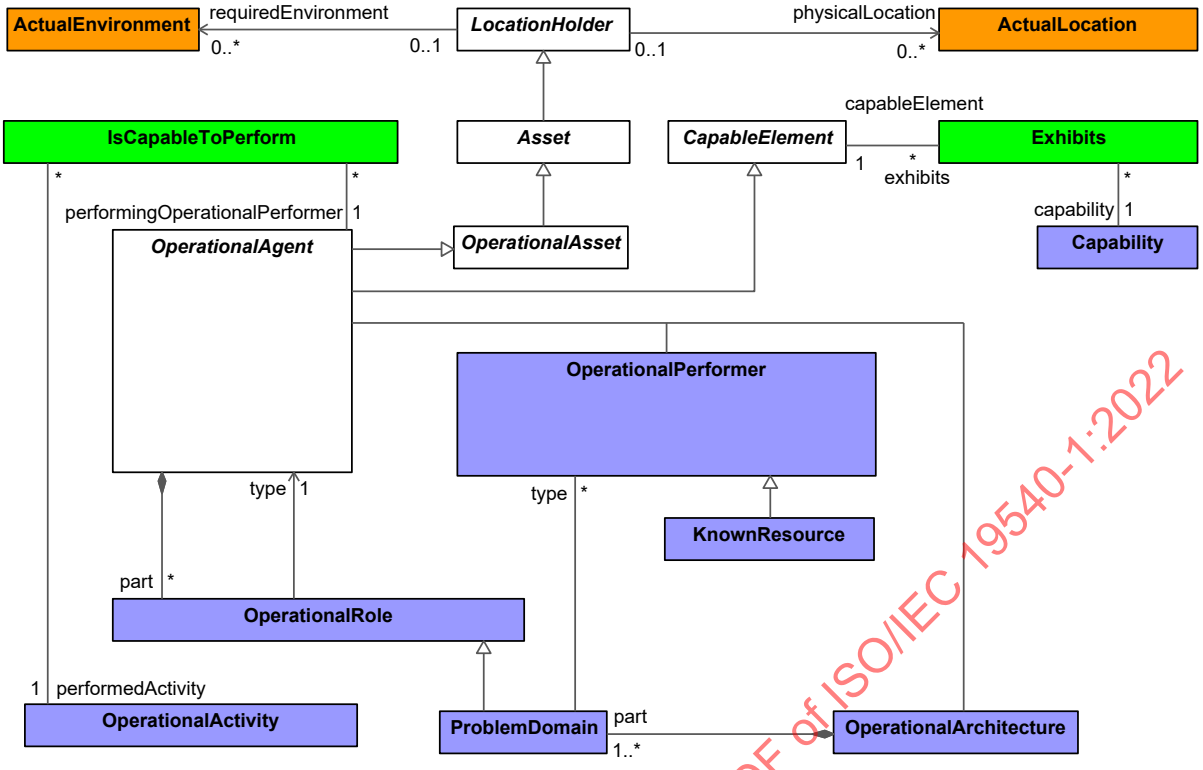

Figure 8:10 - Operational Structure

Elements

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

View Specifications::Operational::Connectivity

Contains the diagrams that document the Operational Connectivity Viewpoint.

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.

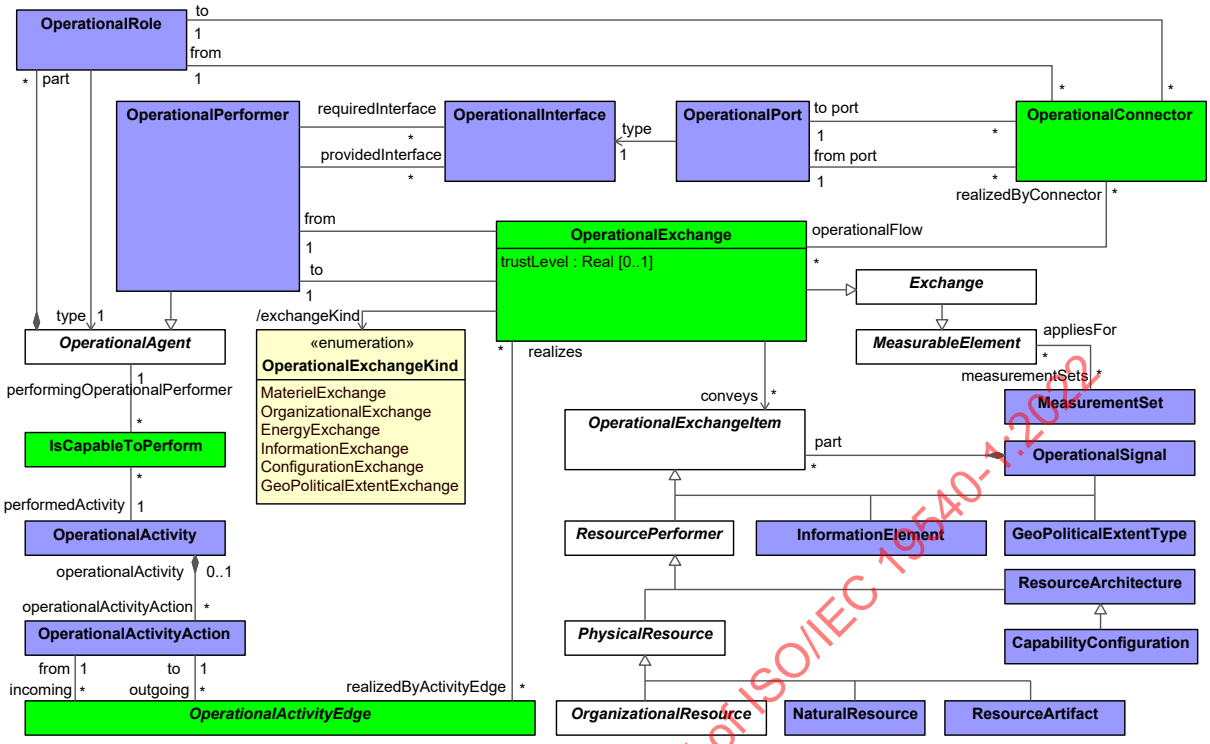

Figure 8:11 - Operational Connectivity

Elements

- CapabilityConfiguration
- Exchange
- GeoPoliticalExtentType
- InformationElement
- IsCapableToPerform
- MeasurableElement
- MeasurementSet
- NaturalResource
- OperationalActivity
- OperationalActivityAction
- OperationalActivityEdge
- OperationalAgent
- OperationalConnector
- OperationalExchange
- OperationalExchangeItem
- OperationalInterface
- OperationalPerformer
- OperationalPort
- OperationalRole
- OperationalSignal
- OrganizationalResource
- PhysicalResource
- ResourceArchitecture
- ResourceArtifact
- ResourcePerformer

View Specifications::Operational::Processes

Contains the diagrams that document the Operational Processes Viewpoint.

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.

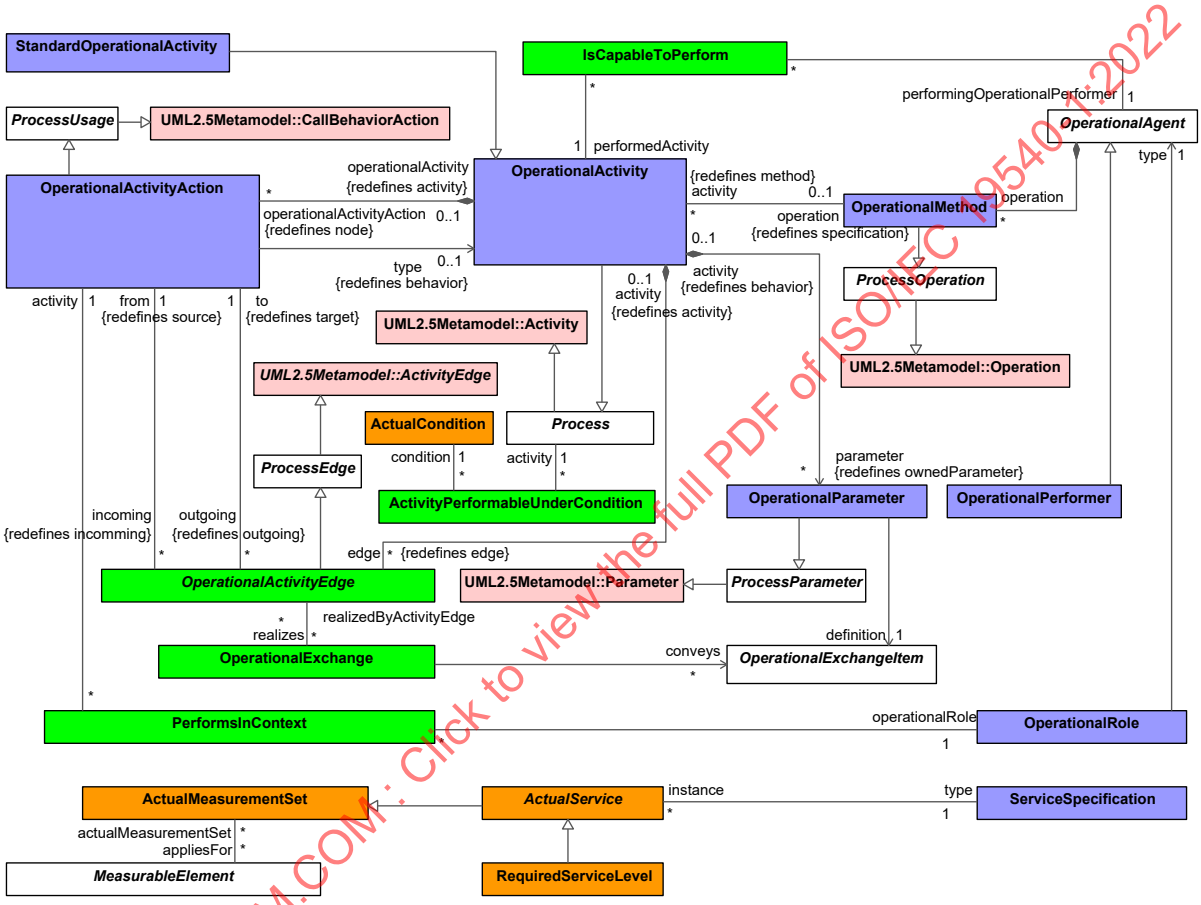

Figure 8:12 - Operational Processes

Elements

- ActivityPerformableUnderCondition
- ActualCondition
- ActualMeasurementSet
- ActualService
- IsCapableToPerform
- MeasurableElement
- OperationalActivity
- OperationalActivityAction
- OperationalActivityEdge
- OperationalAgent
- OperationalExchange
- OperationalExchangeItem
- OperationalMethod

- OperationalParameter
- OperationalPerformer
- OperationalRole
- PerformsInContext
- Process
- ProcessEdge
- ProcessOperation
- ProcessParameter
- ProcessUsage
- RequiredServiceLevel
- ServiceSpecification
- StandardOperationalActivity
- UML2.5Metamodel::Activity
- UML2.5Metamodel::ActivityEdge
- UML2.5Metamodel::CallBehaviorAction
- UML2.5Metamodel::Operation
- UML2.5Metamodel::Parameter

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.

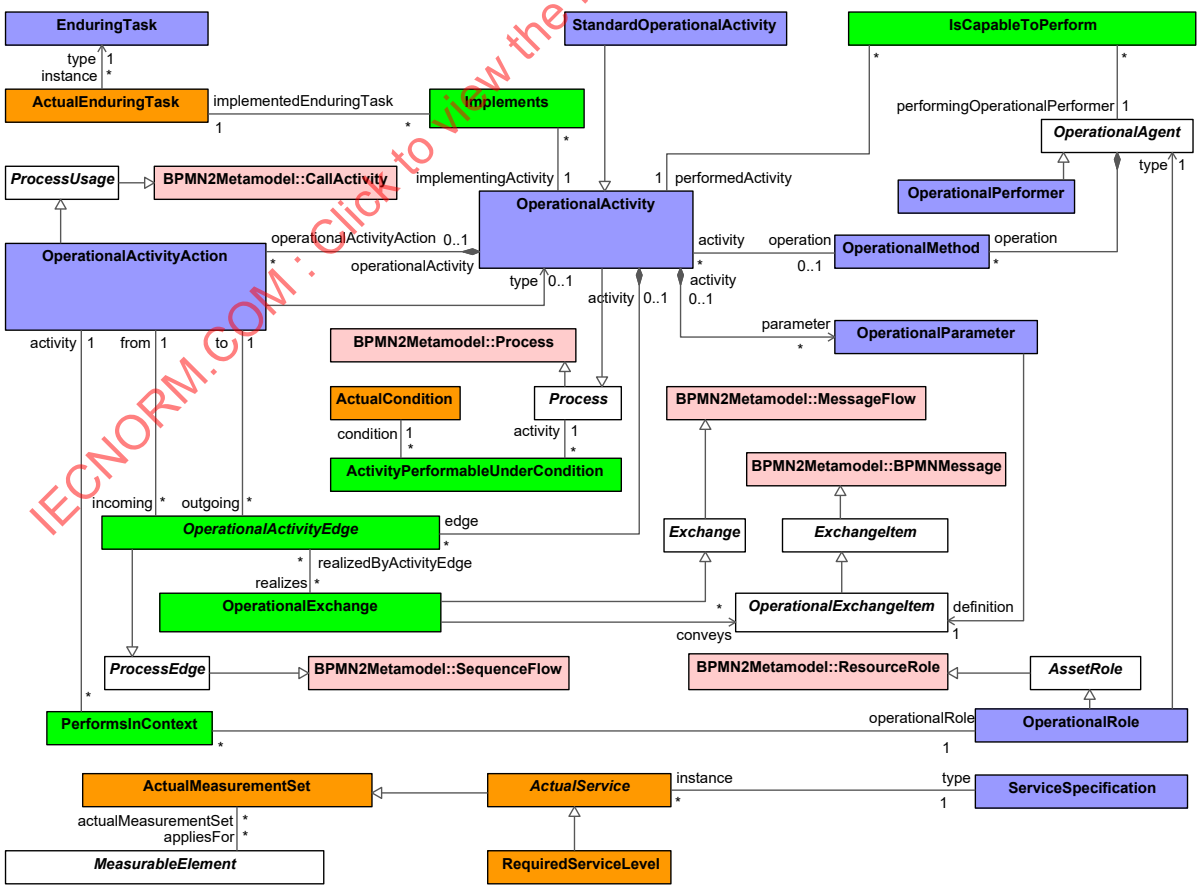

Figure 8:13 - Operational Processes BPMN Semantics

Elements

- ActivityPerformableUnderCondition
- ActualCondition
- ActualEnduringTask
- ActualMeasurementSet
- ActualService
- AssetRole
- BPMN2Metamodel::BPMNMessage
- BPMN2Metamodel::CallActivity
- BPMN2Metamodel::MessageFlow
- BPMN2Metamodel::Process
- BPMN2Metamodel::ResourceRole
- BPMN2Metamodel::SequenceFlow
- EnduringTask
- Exchange
- ExchangeItem
- Implements
- IsCapableToPerform
- MeasurableElement
- OperationalActivity
- OperationalActivityAction
- OperationalActivityEdge
- OperationalAgent
- OperationalExchange
- OperationalExchangeItem
- OperationalMethod
- OperationalParameter
- OperationalPerformer
- OperationalRole
- PerformsInContext
- Process
- ProcessEdge
- ProcessUsage
- RequiredServiceLevel
- ServiceSpecification
- StandardOperationalActivity

View Specifications::Operational::States

Contains the diagrams that document the Operational States Viewpoint.

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.

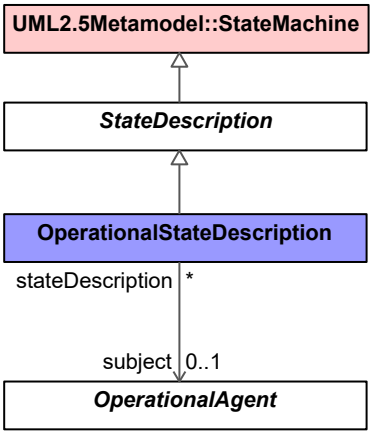

Figure 8:14 - Operational States

Elements

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

View Specifications::Operational::Interaction Scenarios

Contains the diagrams that document the Operational Interaction Scenarios Viewpoint.

View Specifications::Operational::Interaction Scenarios::Operational Interaction Scenarios

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.

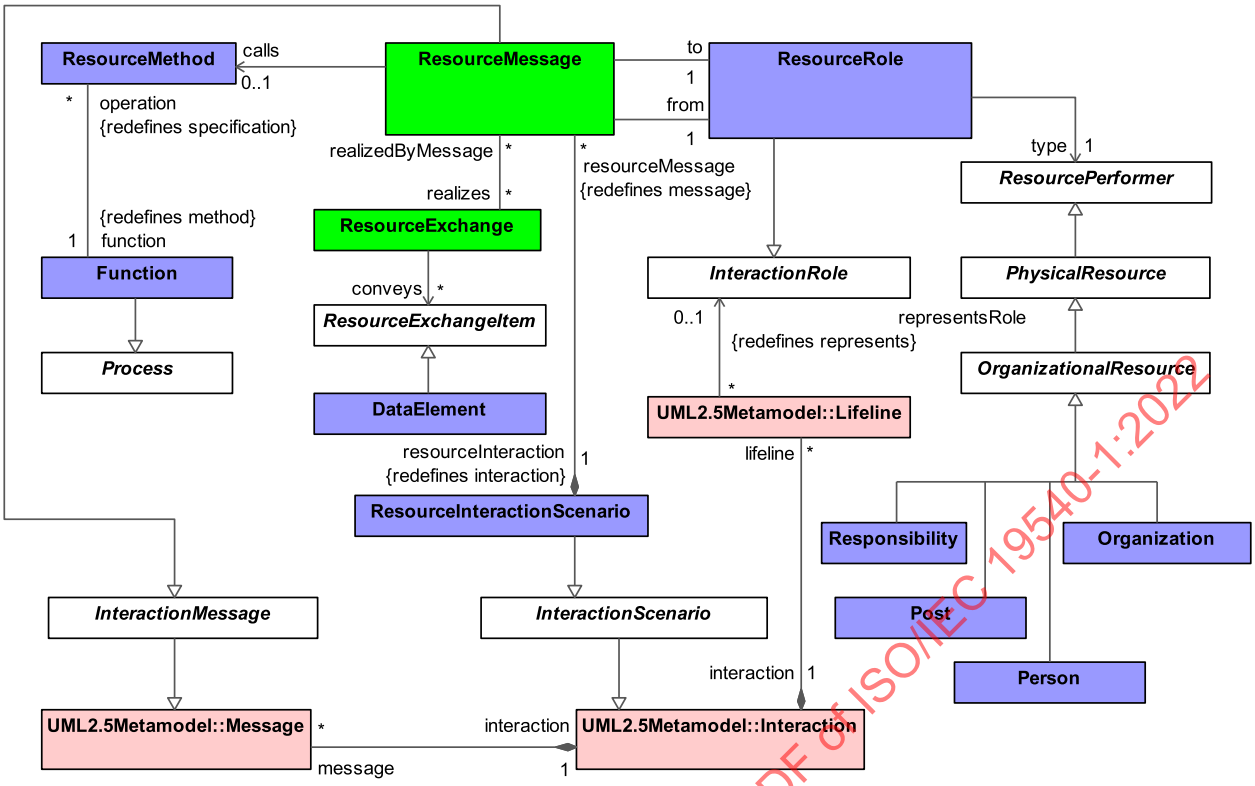

Figure 8:15 - Operational Interaction Scenarios

Elements

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

View Specifications::Operational::Constraints

Contains the diagrams that document the Operational Constraints Viewpoint.

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.

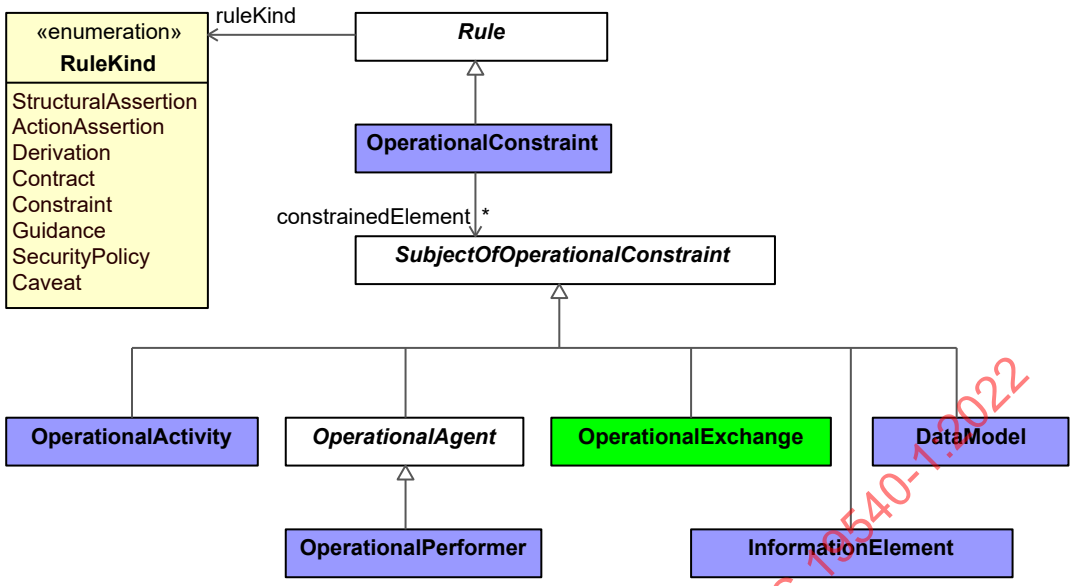

Figure 8:16 - Operational Constraints

Elements

- DataModel
- InformationElement
- OperationalActivity
- OperationalAgent
- OperationalConstraint
- OperationalExchange
- OperationalPerformer
- Rule
- SubjectOfOperationalConstraint

View Specifications::Operational::Traceability

Contains the diagrams that document the Operational Traceability Viewpoint.

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.

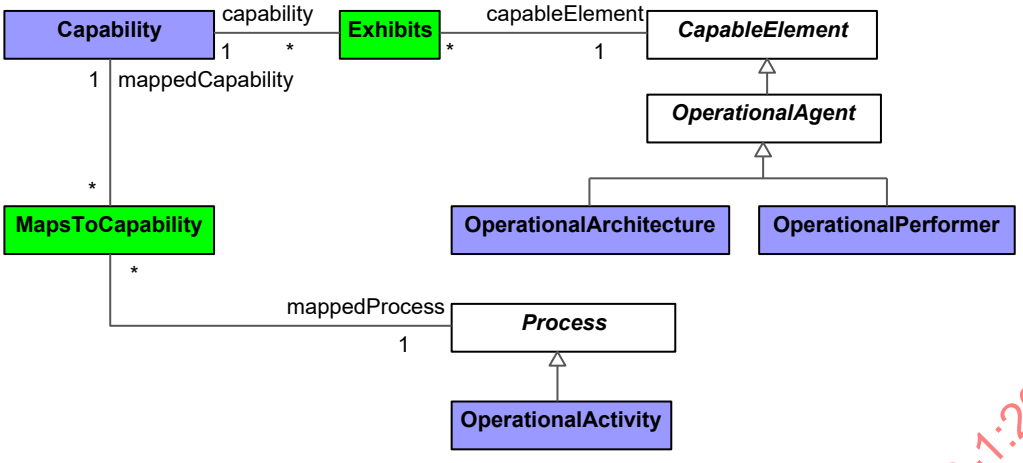

Figure 8:17 - Operational Traceability

Elements

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

8.1.4 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.

View Specifications::Services::Taxonomy

Contains the diagrams that document the Services Taxonomy Viewpoint.

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.

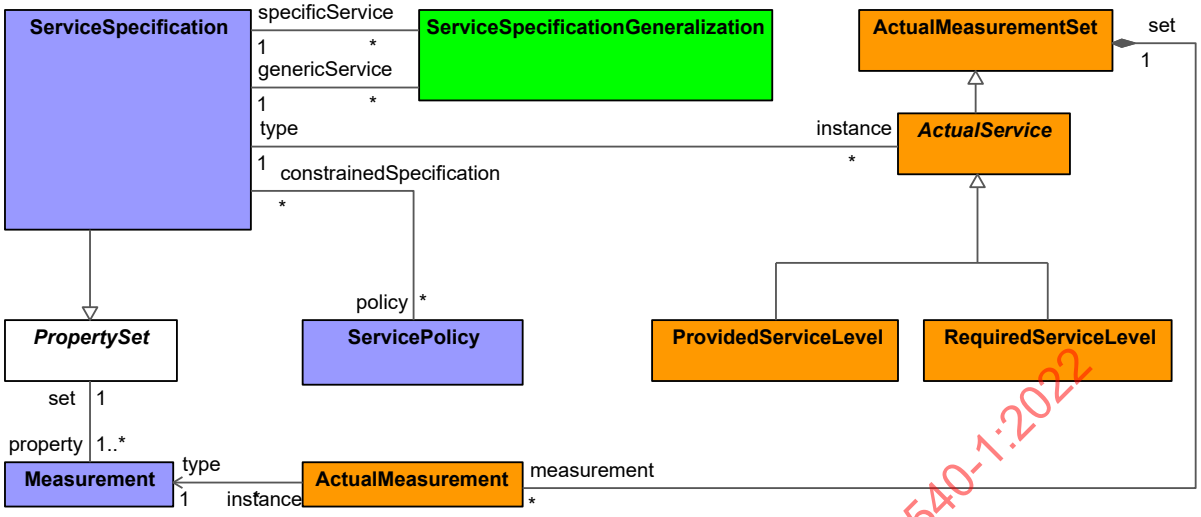

Figure 8:18 - Services Taxonomy

Elements

- ActualMeasurement
- ActualMeasurementSet
- ActualService
- Measurement
- PropertySet
- ProvidedServiceLevel
- RequiredServiceLevel
- ServicePolicy
- ServiceSpecification
- ServiceSpecificationGeneralization

View Specifications::Services::Structure

Contains the diagrams that document the Services Structure Viewpoint.

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.

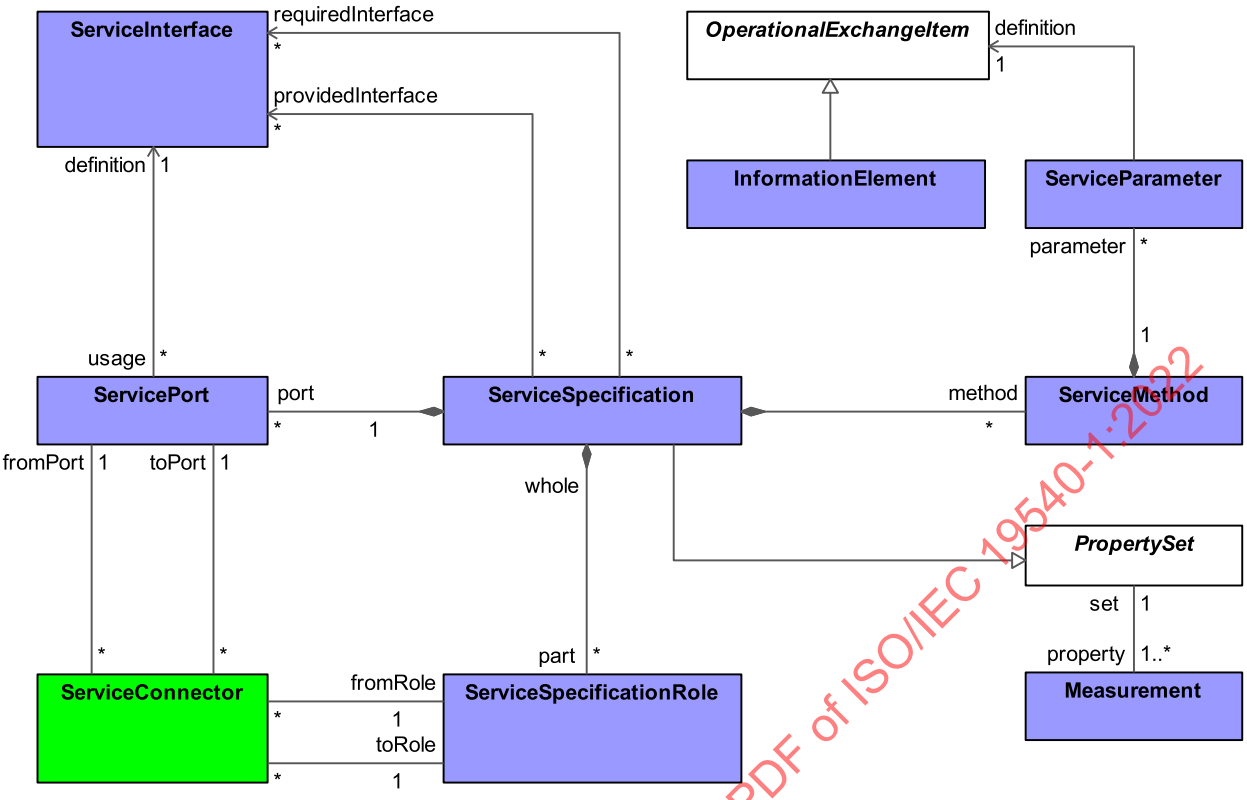

Figure 8:19 - Services Structure

Elements

- InformationElement
- Measurement
- OperationalExchangeItem
- PropertySet
- ServiceConnector
- ServiceInterface
- ServiceMethod
- ServiceParameter
- ServicePort
- ServiceSpecification
- ServiceSpecificationRole

View Specifications::Services::Connectivity

Contains the diagrams that document the Services Connectivity Viewpoint.

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.

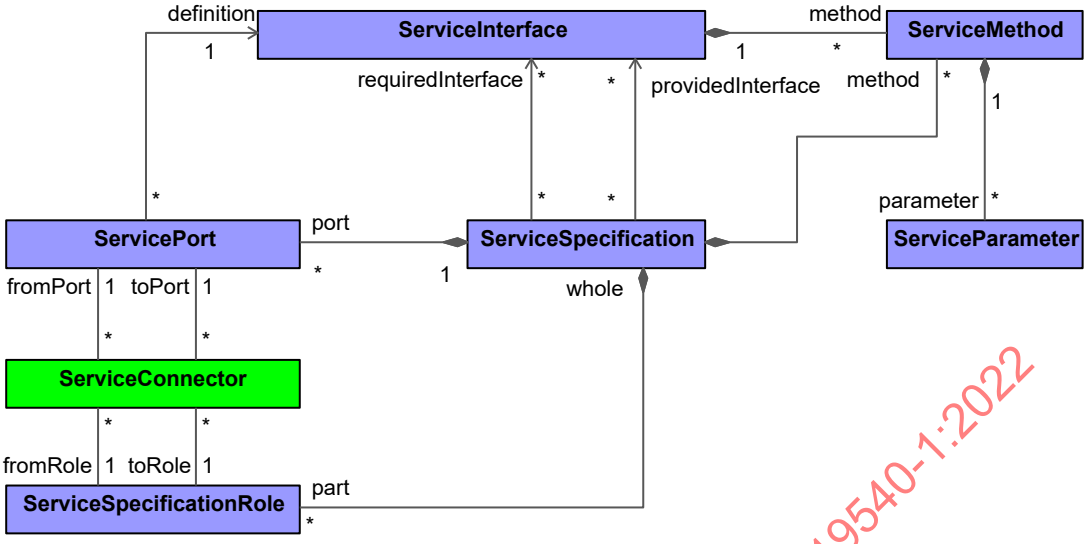

Figure 8:20 - Services Connectivity

Elements

- ServiceConnector
- ServiceInterface
- ServiceMethod
- ServiceParameter
- ServicePort
- ServiceSpecification
- ServiceSpecificationRole

View Specifications::Services::Processes

Contains the diagrams that document the Services Processes Viewpoint.

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.

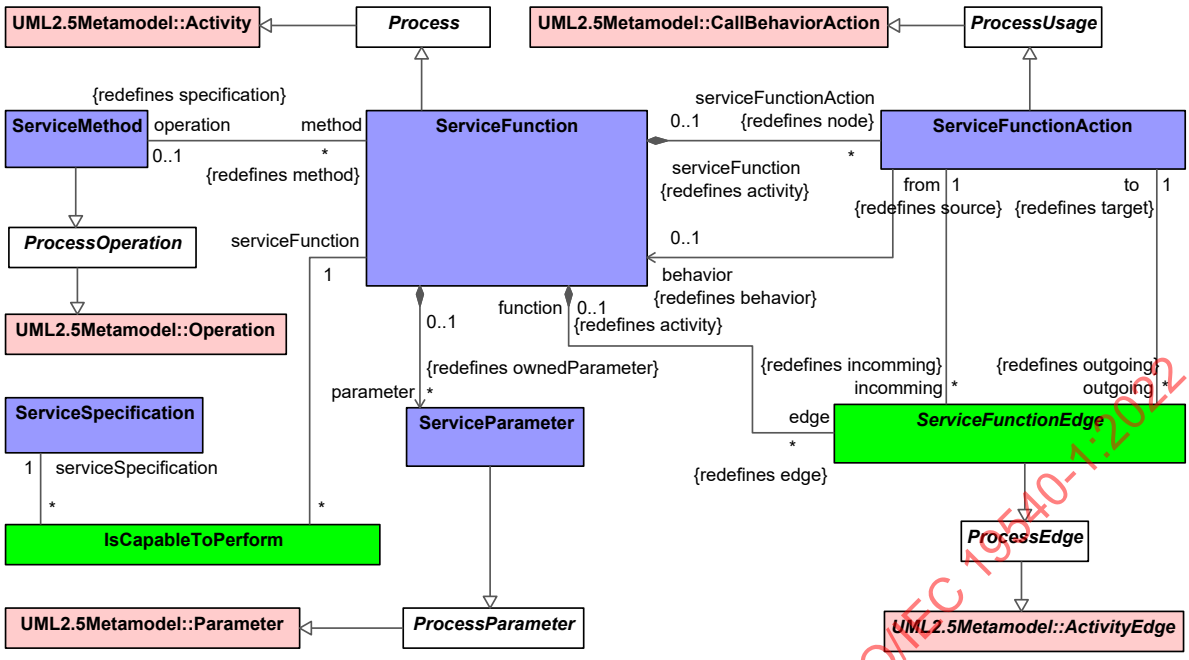

Figure 8:21 - Services Processes

Elements

- IsCapableToPerform
- Process
- ProcessEdge
- ProcessOperation
- ProcessParameter
- ProcessUsage
- ServiceFunction
- ServiceFunctionAction
- ServiceFunctionEdge
- ServiceMethod
- ServiceParameter
- ServiceSpecification
- UML2.5Metamodel::Activity
- UML2.5Metamodel::ActivityEdge
- UML2.5Metamodel::CallBehaviorAction
- UML2.5Metamodel::Operation
- UML2.5Metamodel::Parameter

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.

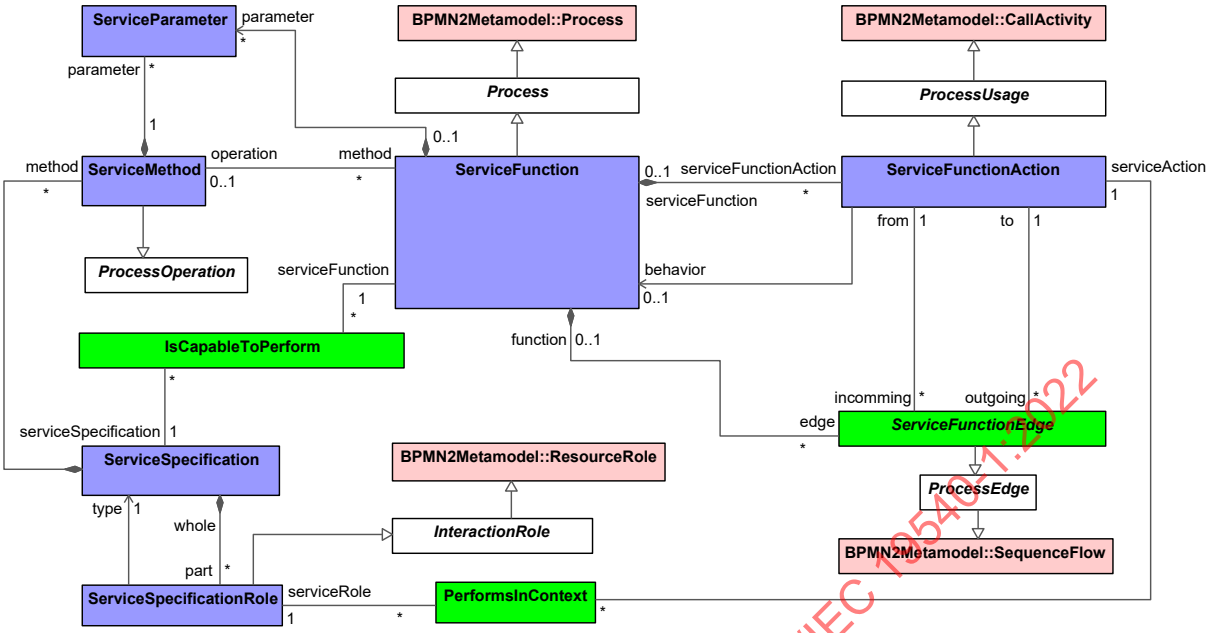

Figure 8:22 - Services Processes BPMN Semantics

Elements

- BPMN2Metamodel::CallActivity
- BPMN2Metamodel::Process
- BPMN2Metamodel::ResourceRole
- BPMN2Metamodel::SequenceFlow
- InteractionRole
- IsCapableToPerform
- PerformsInContext
- Process
- ProcessEdge
- ProcessOperation
- ProcessUsage
- ServiceFunction
- ServiceFunctionAction
- ServiceFunctionEdge
- ServiceMethod
- ServiceParameter
- ServiceSpecification
- ServiceSpecificationRole

View Specifications::Services::States

Contains the diagrams that document the Services States Viewpoint.

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.

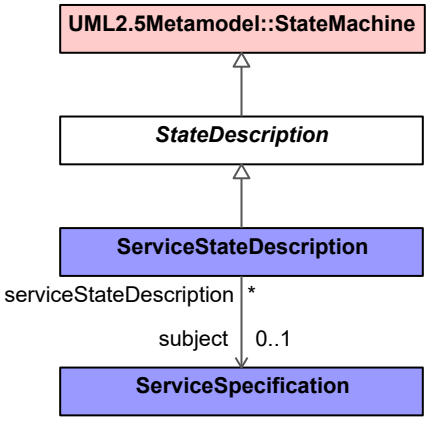

Figure 8:23 - Services States

Elements

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

View Specifications::Services::Interaction Scenarios

Contains the diagrams that document the Services Interaction Scenarios Viewpoint.

View Specifications::Services::Interaction Scenarios::Services Interaction Scenarios

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.

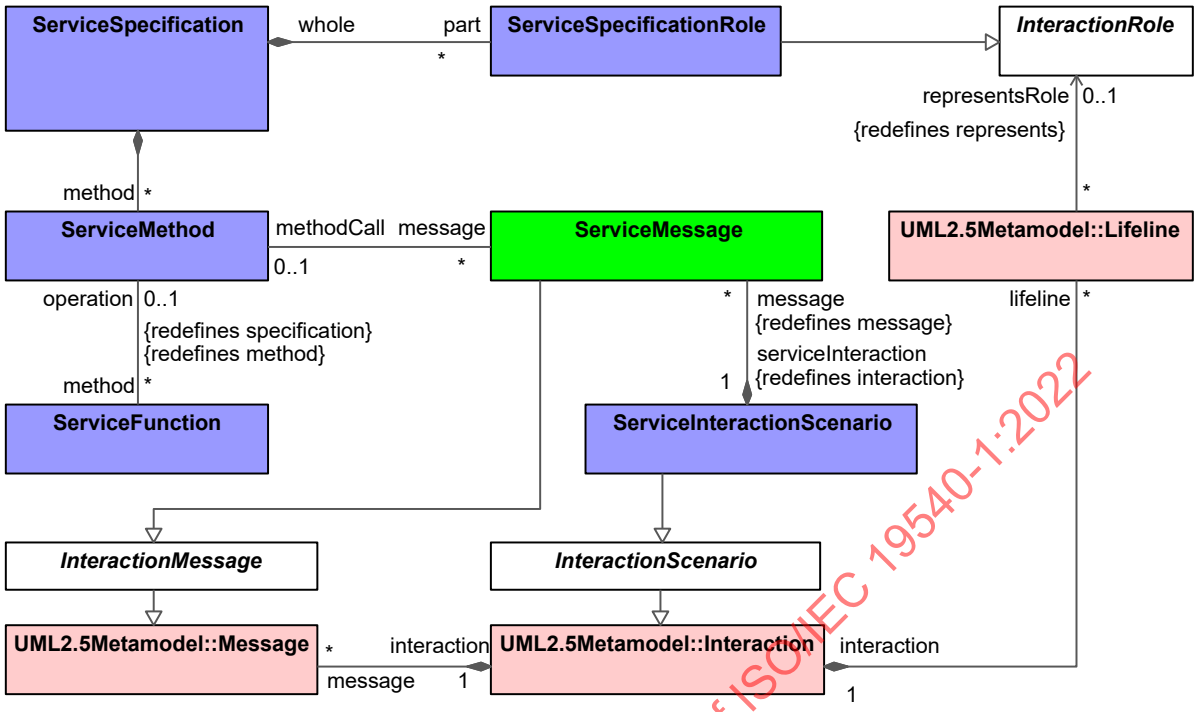

Figure 8:24 - Services Interaction Scenarios

Elements

- InteractionMessage
- InteractionRole
- InteractionScenario
- ServiceFunction
- ServiceInteractionScenario
- ServiceMessage
- ServiceMethod
- ServiceSpecification
- ServiceSpecificationRole
- UML2.5Metamodel::Interaction
- UML2.5Metamodel::Lifeline
- UML2.5Metamodel::Message

View Specifications::Services::Constraints

Contains the diagrams that document the Services Constraints Viewpoint.

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 8:25 - Services Constraints

Elements

- Rule
- ServicePolicy
- ServiceSpecification

View Specifications::Services::Roadmap

Contains the diagrams that document the Services Roadmap Viewpoint.

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.

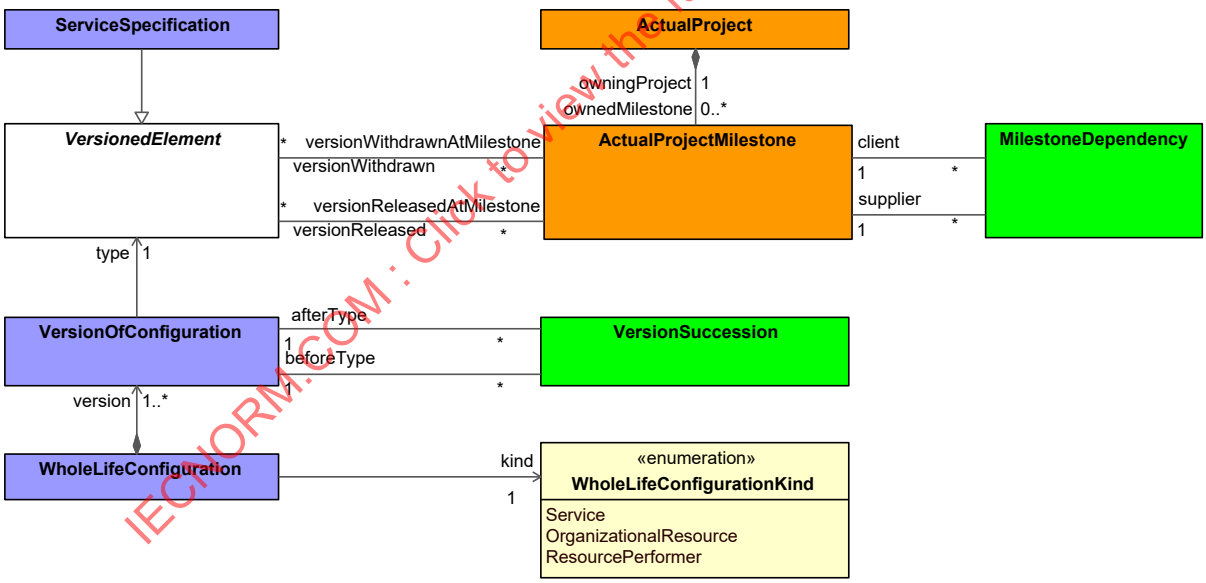

Figure 8:26 - Services Roadmap

Elements

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

View Specifications::Services::Traceability

Contains the diagrams that document the Services Traceability Viewpoint.

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.

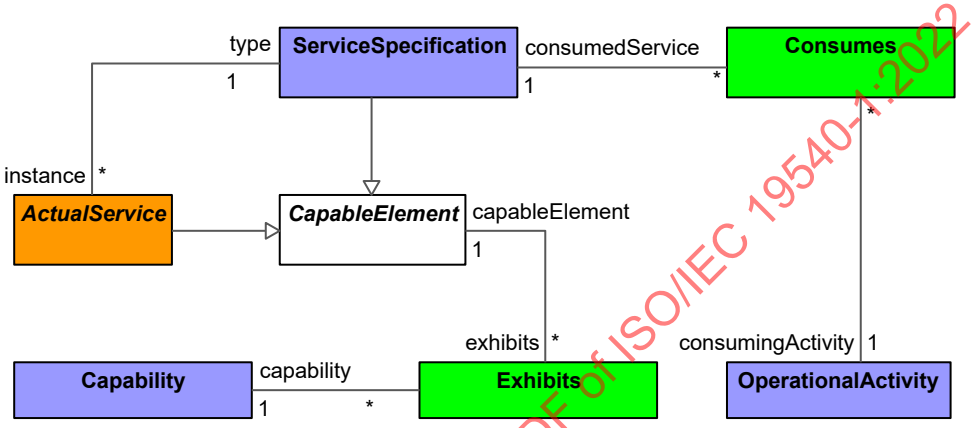

Figure 8:27 - Services Traceability

Elements

- ActualService
- Capability
- CapableElement
- Consumes
- Exhibits
- OperationalActivity
- ServiceSpecification

8.1.5 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).

View Specifications::Personnel::Taxonomy

Contains the diagrams that document the Personnel Taxonomy Viewpoint.

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.

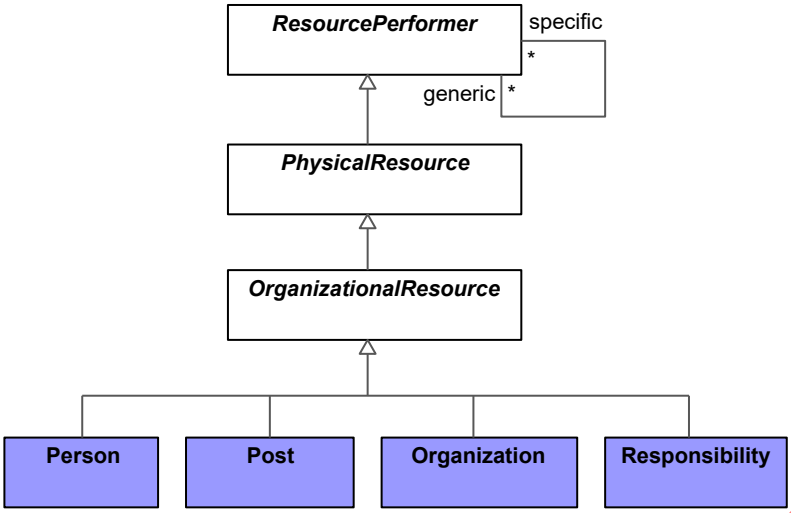

Figure 8:28 - Personnel Taxonomy

Elements

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

View Specifications::Personnel::Structure

Contains the diagrams that document the Personnel Structure Viewpoint.

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.

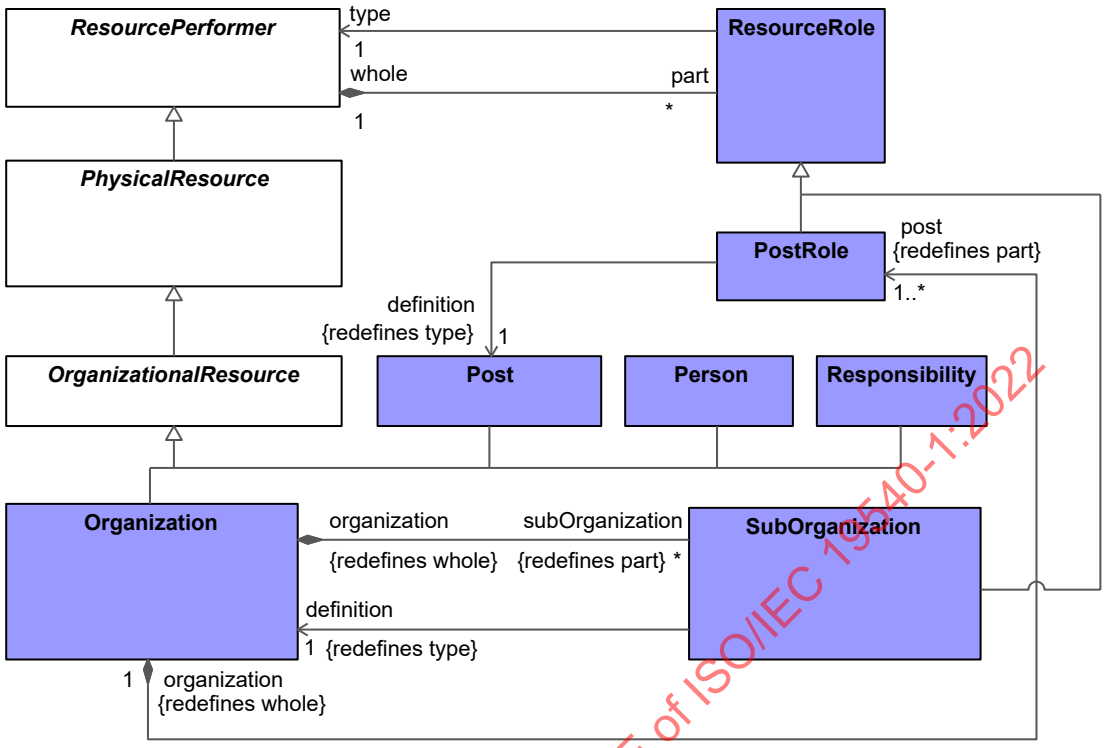

Figure 8:29 - Personnel Structure

Elements

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

View Specifications::Personnel::Connectivity

Contains the diagrams that document the Personnel Connectivity Viewpoint.

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.

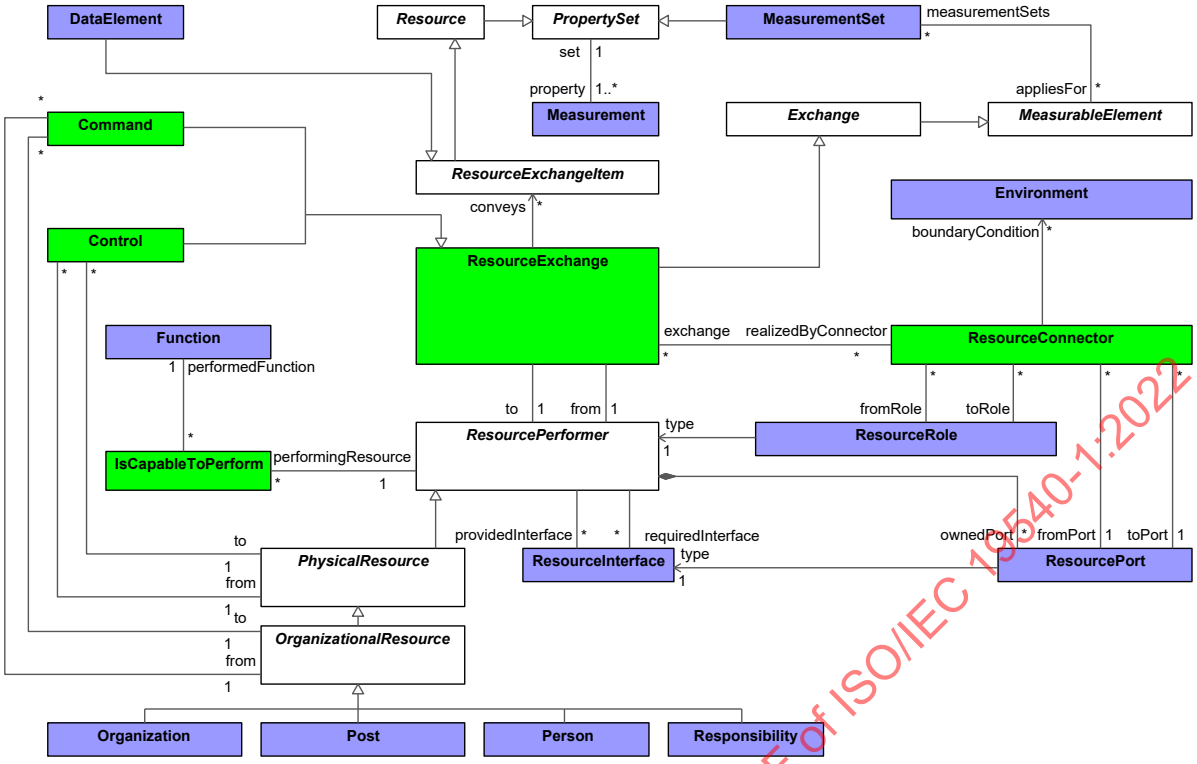

Figure 8:30 - Personnel Connectivity

Elements

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

View Specifications::Personnel::Processes

Contains the diagrams that document the Personnel Processes Viewpoint.

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.

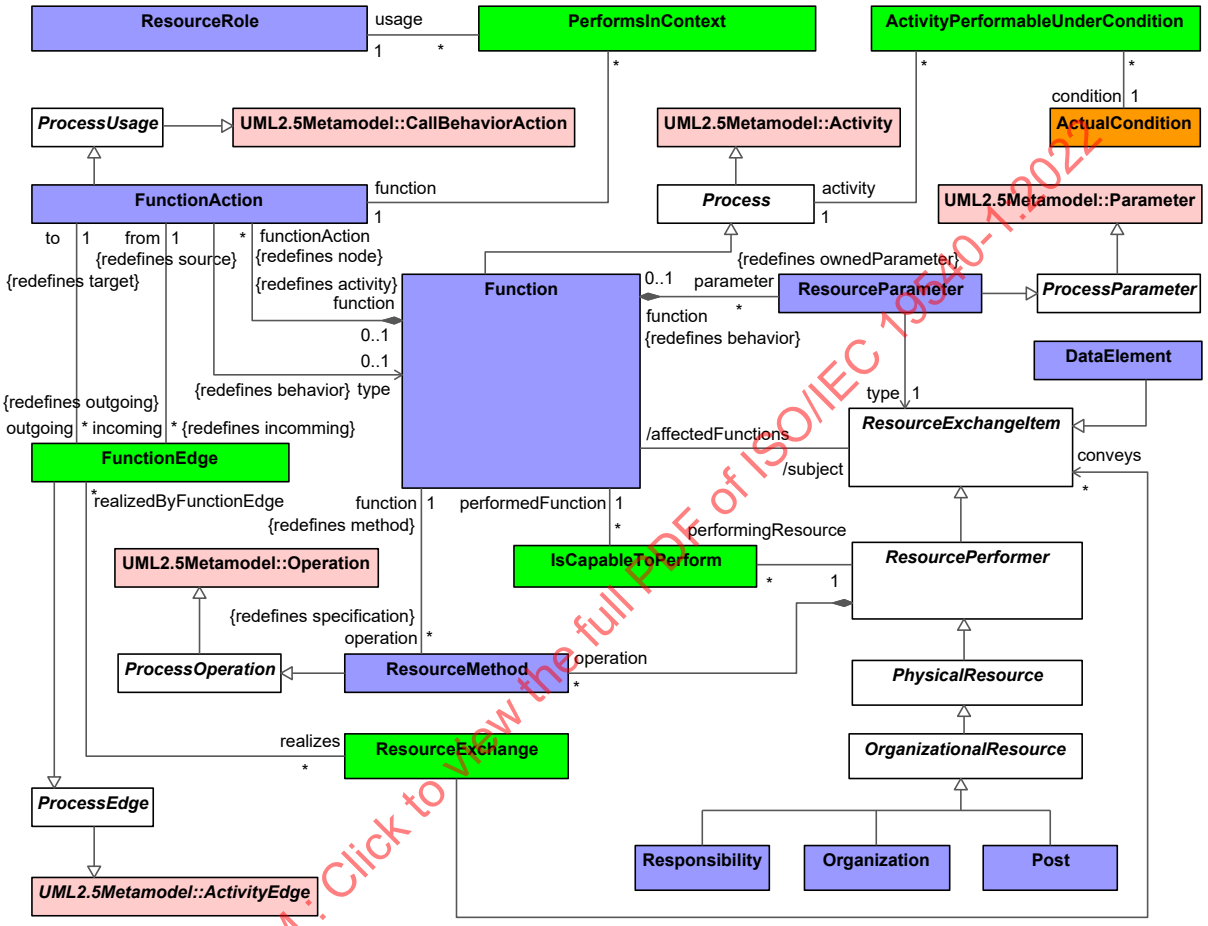

Figure 8:31 - Personnel Processes

Elements

- ActivityPerformableUnderCondition
- ActualCondition
- DataElement
- Function
- FunctionAction
- FunctionEdge
- IsCapableToPerform
- Organization
- OrganizationalResource
- PerformsInContext
- PhysicalResource
- Post
- Process
- ProcessEdge
- ProcessOperation
- ProcessParameter
- ProcessUsage

- ResourceExchange
- ResourceExchangeItem
- ResourceMethod
- ResourceParameter
- ResourcePerformer
- ResourceRole
- Responsibility
- UML2.5Metamodel::Activity
- UML2.5Metamodel::ActivityEdge
- UML2.5Metamodel::CallBehaviorAction
- UML2.5Metamodel::Operation
- UML2.5Metamodel::Parameter

View Specifications::Personnel::States

Contains the diagrams that document the Personnel States Viewpoint.

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.

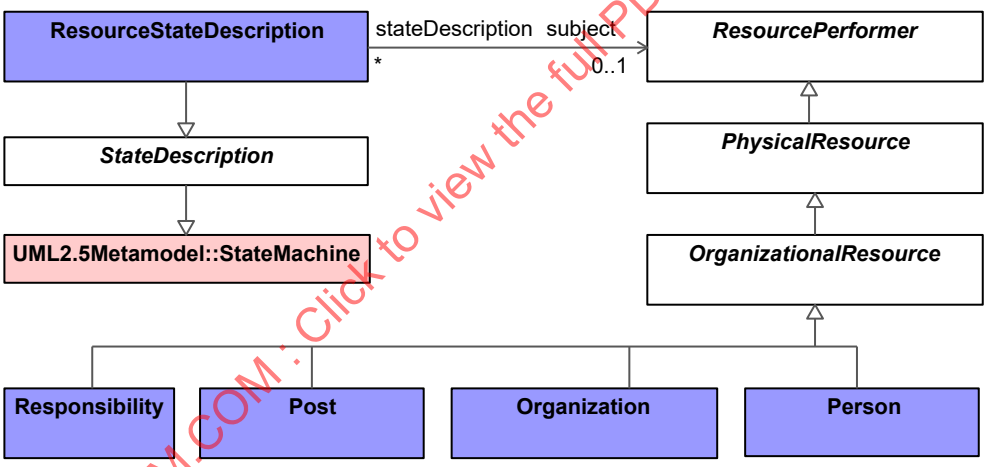

Figure 8:32 - Personnel States

Elements

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

View Specifications::Personnel::Interaction Scenarios

Contains the diagrams that document the Personnel Interaction Scenarios Viewpoint.

View Specifications::Personnel::Interaction Scenarios::Personnel Interaction Scenarios

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.

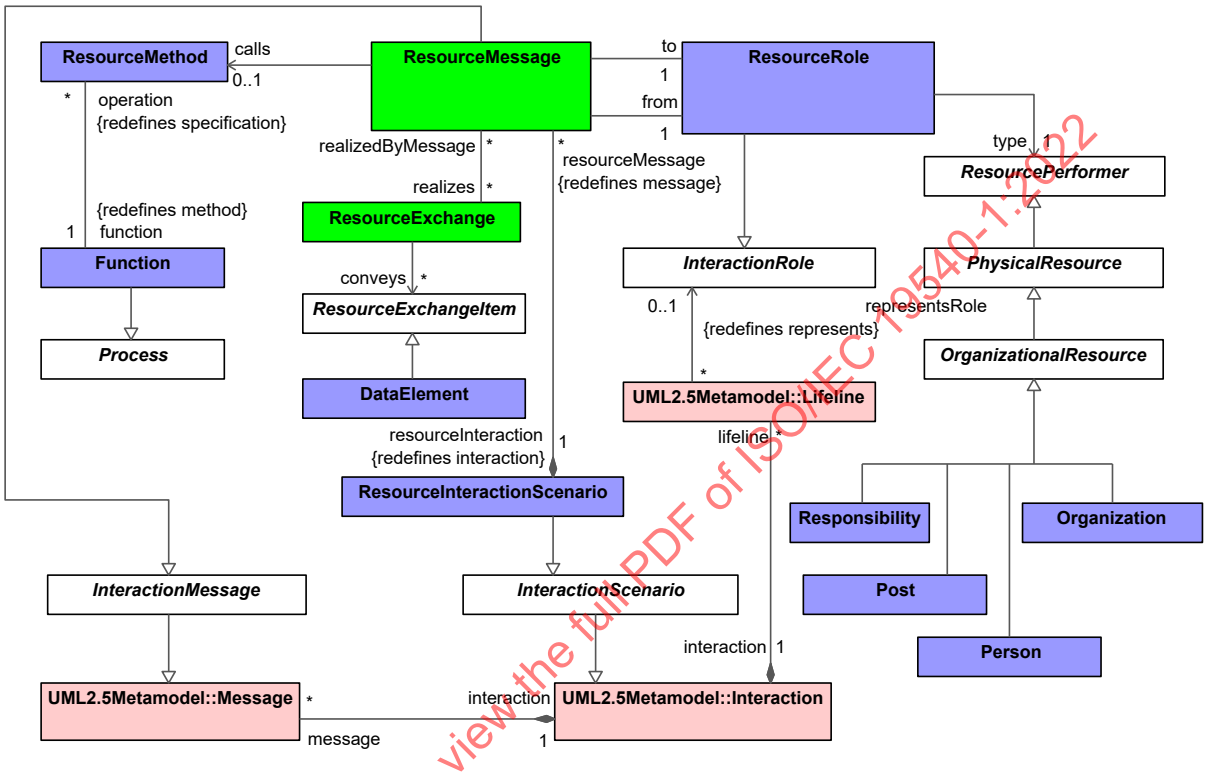

Figure 8:33 - Personnel Interaction Scenarios

Elements

- DataElement
- Function
- InteractionMessage
- InteractionRole
- InteractionScenario
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- Process
- ResourceExchange
- ResourceExchangeItem
- ResourceInteractionScenario
- ResourceMessage
- ResourceMethod
- ResourcePerformer
- ResourceRole
- Responsibility
- UML2.5Metamodel::Interaction
- UML2.5Metamodel::Lifeline
- UML2.5Metamodel::Message

View Specifications::Personnel::Constraints

Contains the diagrams that document the Personnel Constraints Viewpoint.

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.

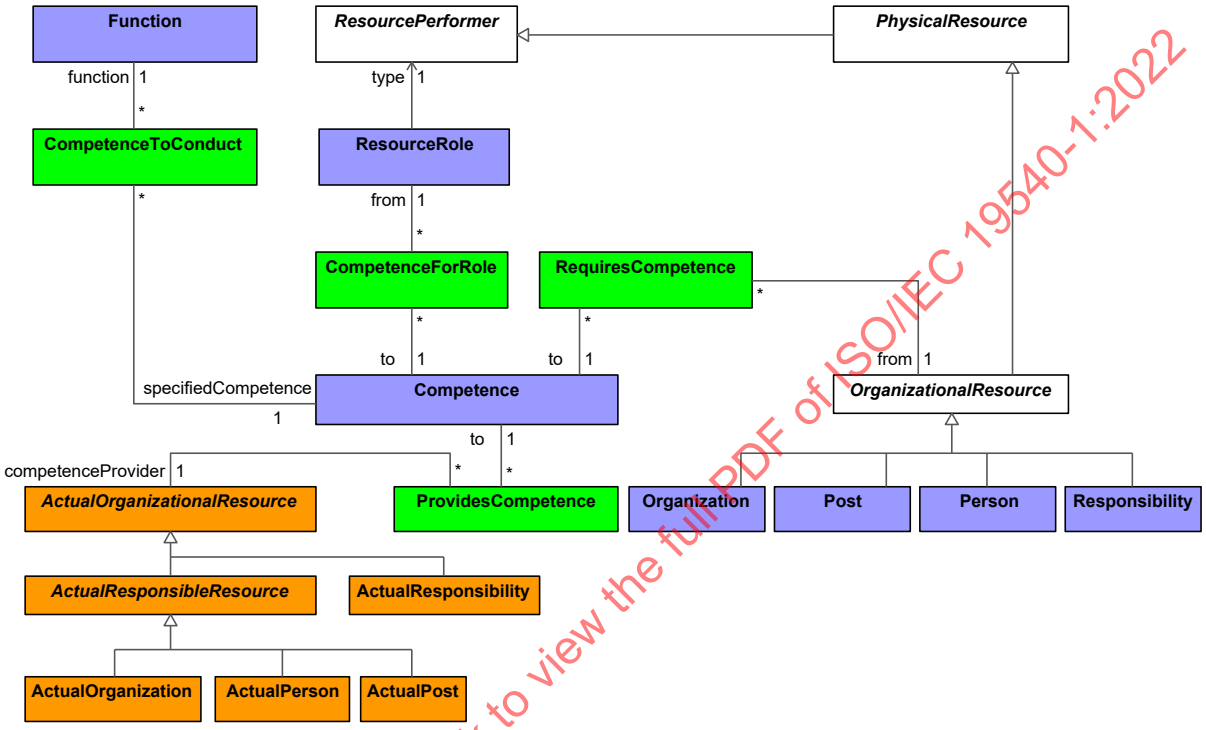

Figure 8:34 – Personnel Constraints: Competence

Elements

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

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.

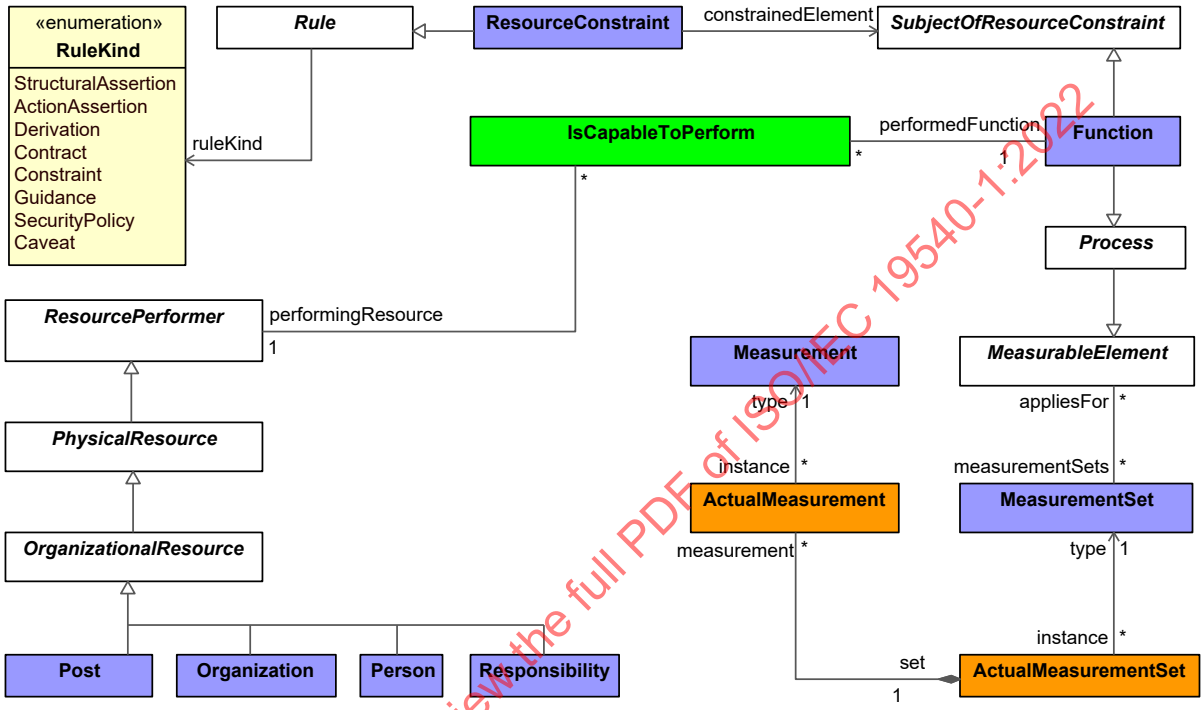

Figure 8:35 - Personnel Constraints: Drivers

Elements

- ActualMeasurement
- ActualMeasurementSet
- Function
- IsCapableToPerform
- MeasurableElement
- Measurement
- MeasurementSet
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- Process
- ResourceConstraint
- ResourcePerformer
- Responsibility
- Rule
- SubjectOfResourceConstraint

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.

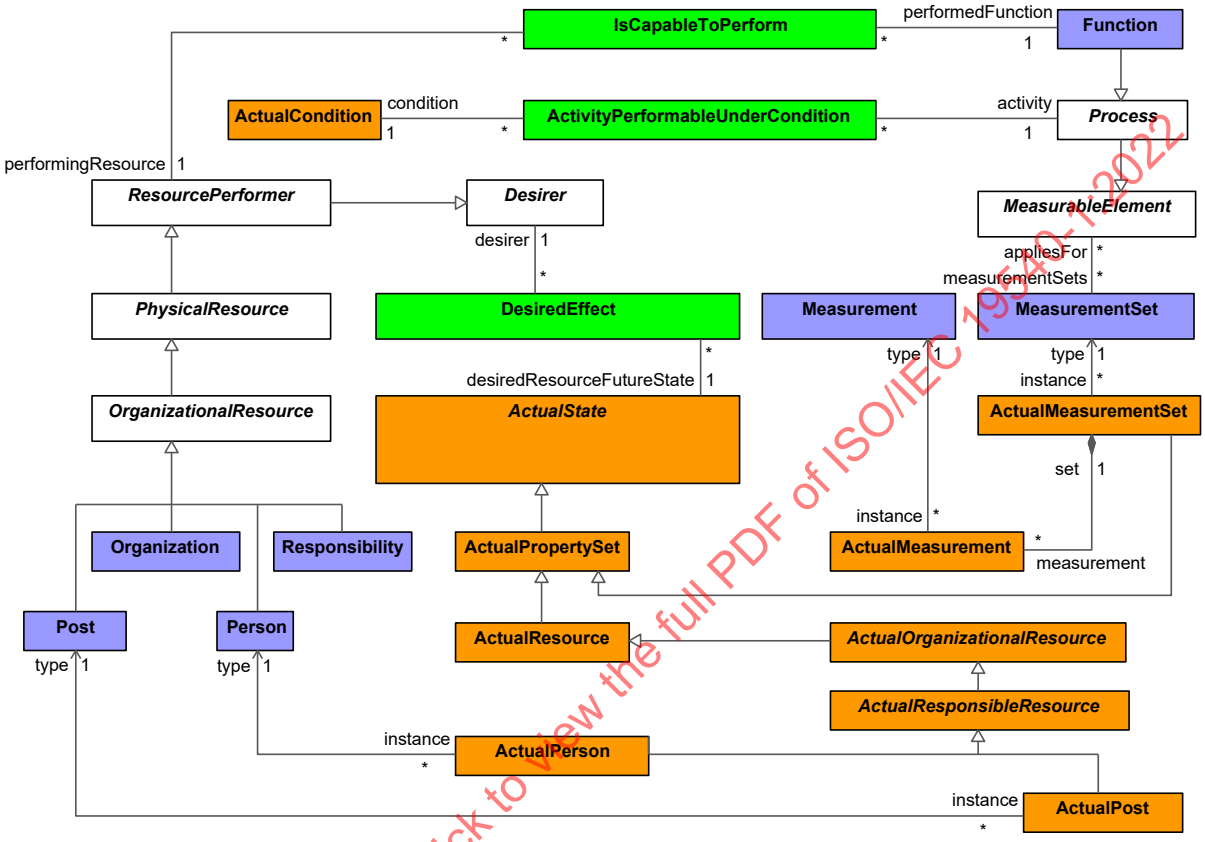

Figure 8.36 - Personnel Constraints: Performance

Elements

- ActivityPerformableUnderCondition
- ActualCondition
- ActualMeasurement
- ActualMeasurementSet
- ActualOrganizationalResource
- ActualPerson
- ActualPost
- ActualPropertySet
- ActualResource
- ActualResponsibleResource
- ActualState
- DesiredEffect
- Desirer
- Function
- IsCapableToPerform
- MeasurableElement
- Measurement
- MeasurementSet
- Organization

- OrganizationalResource
- Person
- PhysicalResource
  - Post
  - Process
  - ResourcePerformer
  - Responsibility

View Specifications::Personnel::Roadmap

Contains the diagrams that document the Personnel Roadmap Viewpoint.

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.

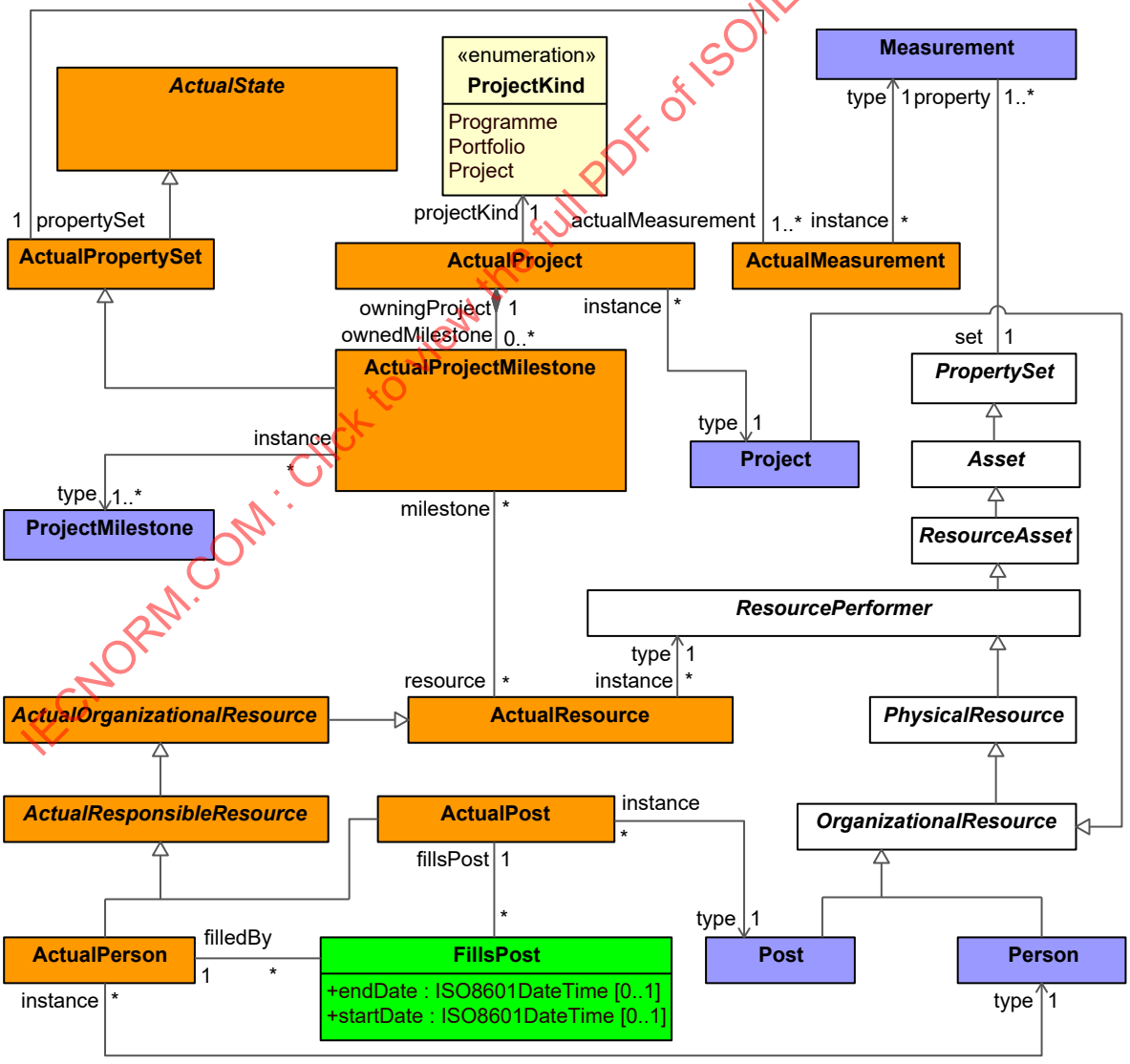

Figure 8:37 - Personnel Roadmap: Availability

Elements

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

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.

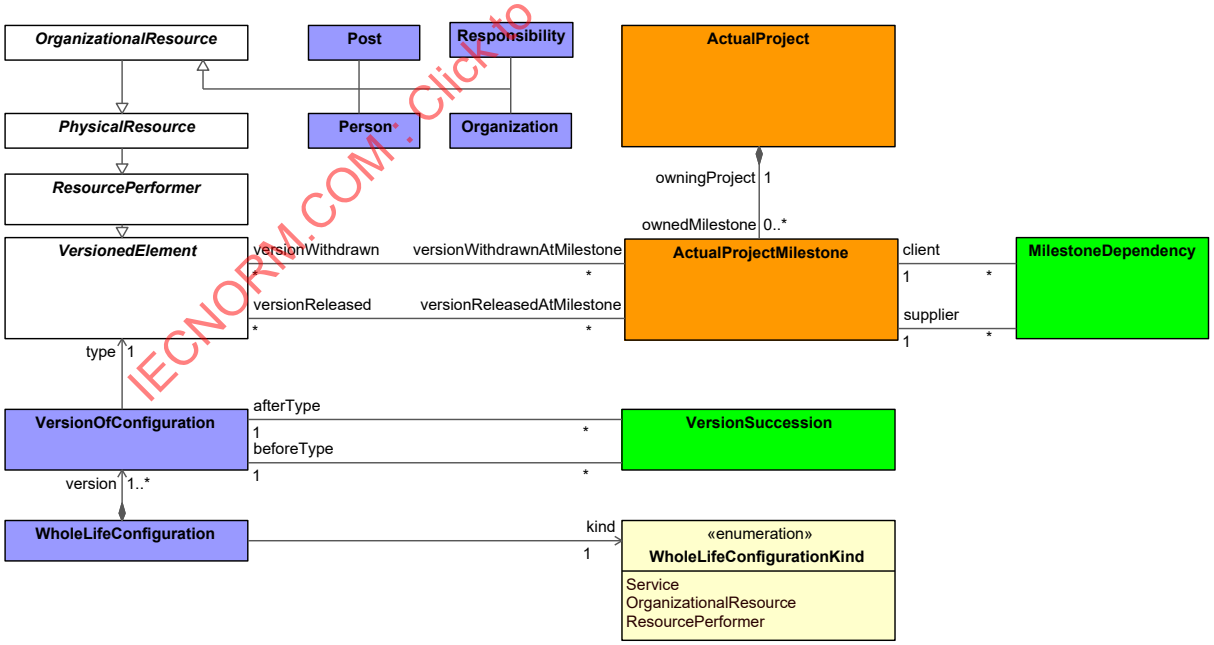

Figure 8:38 - Personnel Roadmap: Evolution

Elements

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

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.

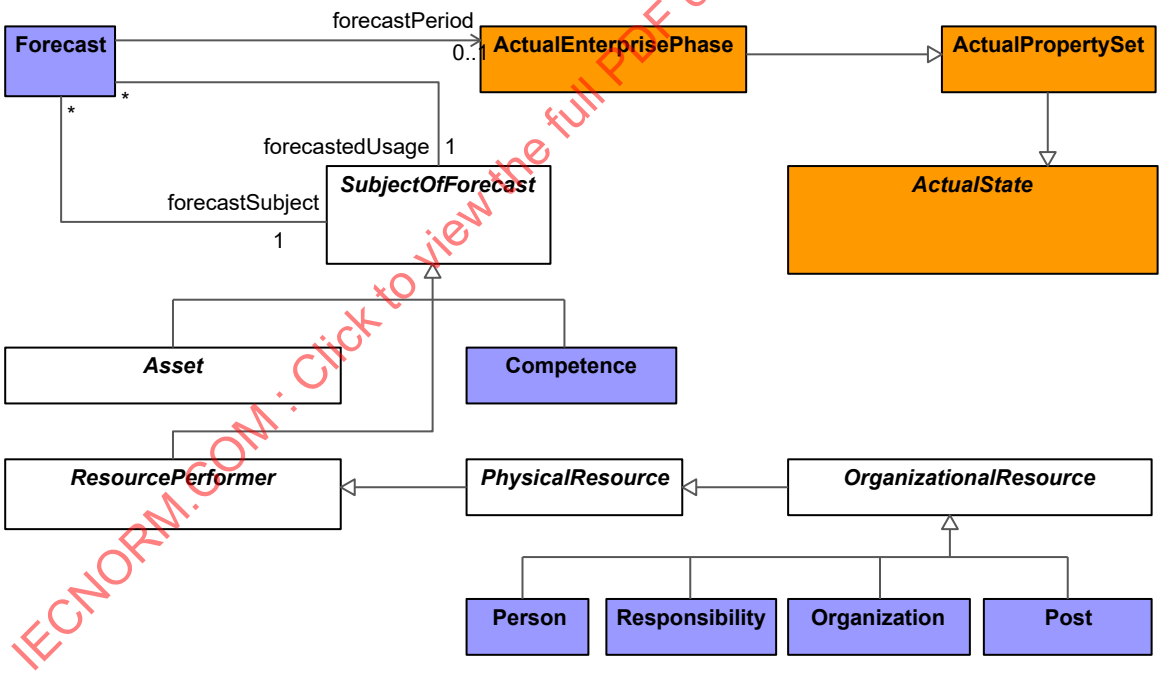

Figure 8:39 - Personnel Roadmap: Forecast

Elements

- ActualEnterprisePhase
- ActualPropertySet
- ActualState
- Asset
- Competence
- Forecast
- Organization
- OrganizationalResource
- Person

- PhysicalResource
- Post
- ResourcePerformer
- Responsibility
- SubjectOfForecast

View Specifications::Personnel::Traceability

Contains the diagrams that document the Personnel Traceability Viewpoint.

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.

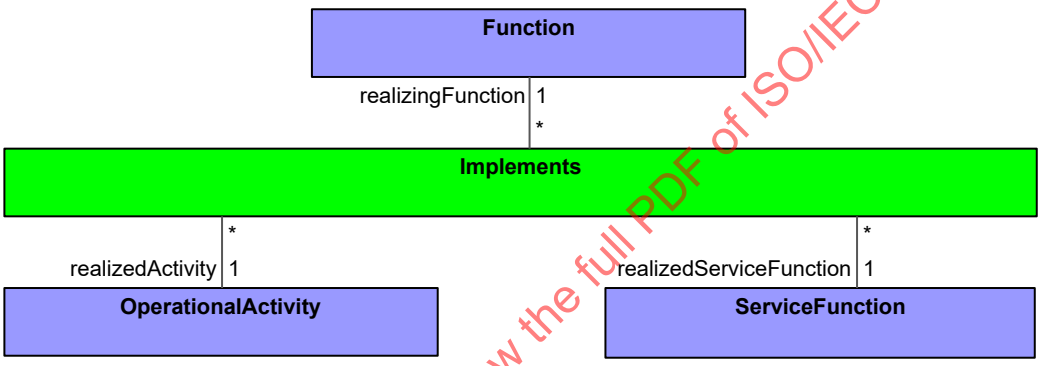

Figure 8:40 - Personnel Traceability

Elements

- Function
- Implements
- OperationalActivity
- ServiceFunction

8.1.6 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.

View Specifications::Resources::Taxonomy

Contains the diagrams that document the Resources Taxonomy Viewpoint.

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.

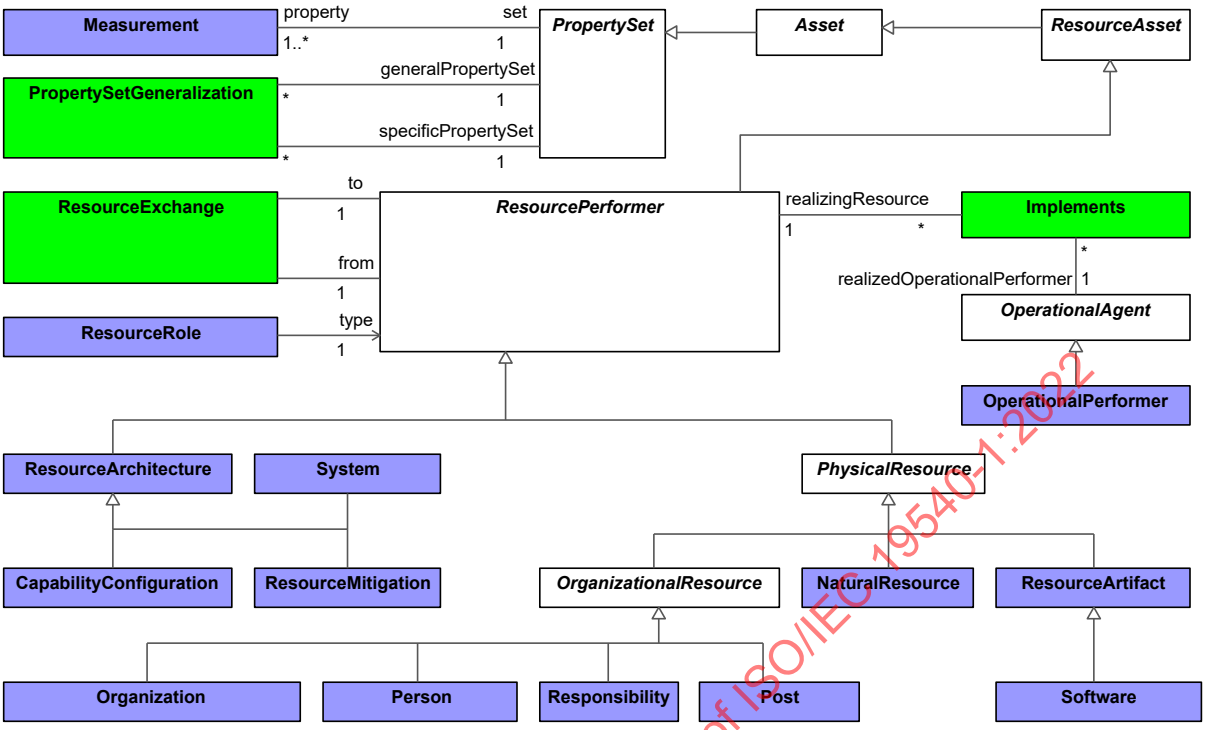

Figure 8:41 - Resources Taxonomy

Elements

- Asset
- CapabilityConfiguration
- Implements
- Measurement
- NaturalResource
- OperationalAgent
- OperationalPerformer
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- PropertySet
- PropertySetGeneralization
- ResourceArchitecture
- ResourceArtifact
- ResourceAsset
- ResourceExchange
- ResourceMitigation
- ResourcePerformer
- ResourceRole
- Responsibility
- Software
- System

View Specifications::Resources::Structure

Contains the diagrams that document the Resources Structure Viewpoint.

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.

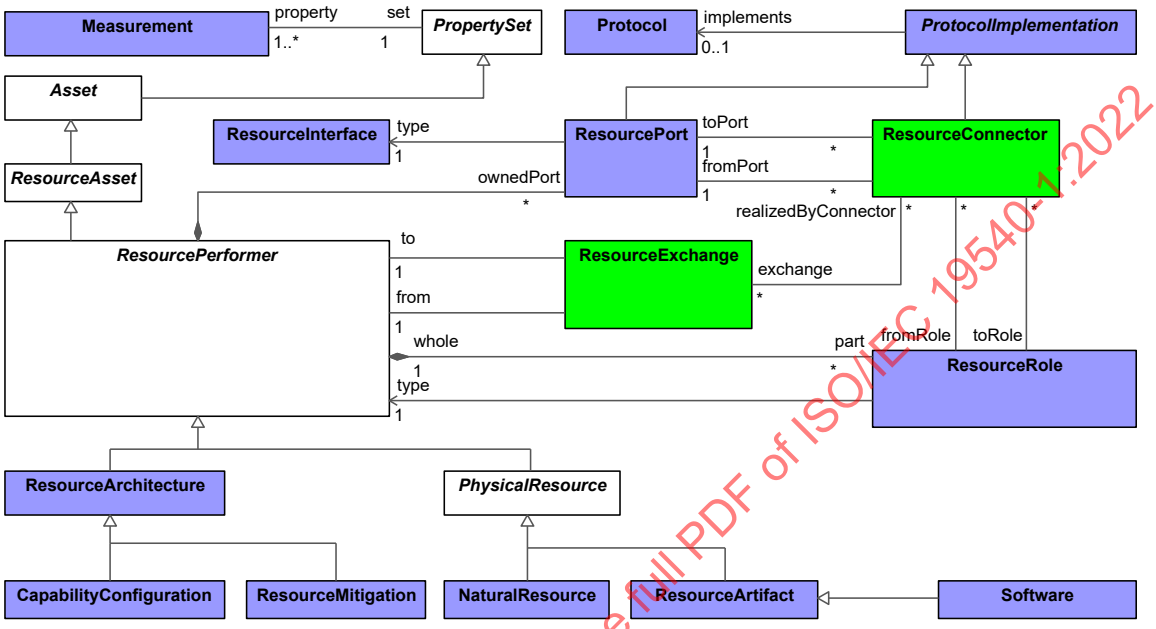

Figure 8:42 - Resources Structure

Elements

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

View Specifications::Resources::Connectivity

Contains the diagrams that document the Resources Connectivity Viewpoint.

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.

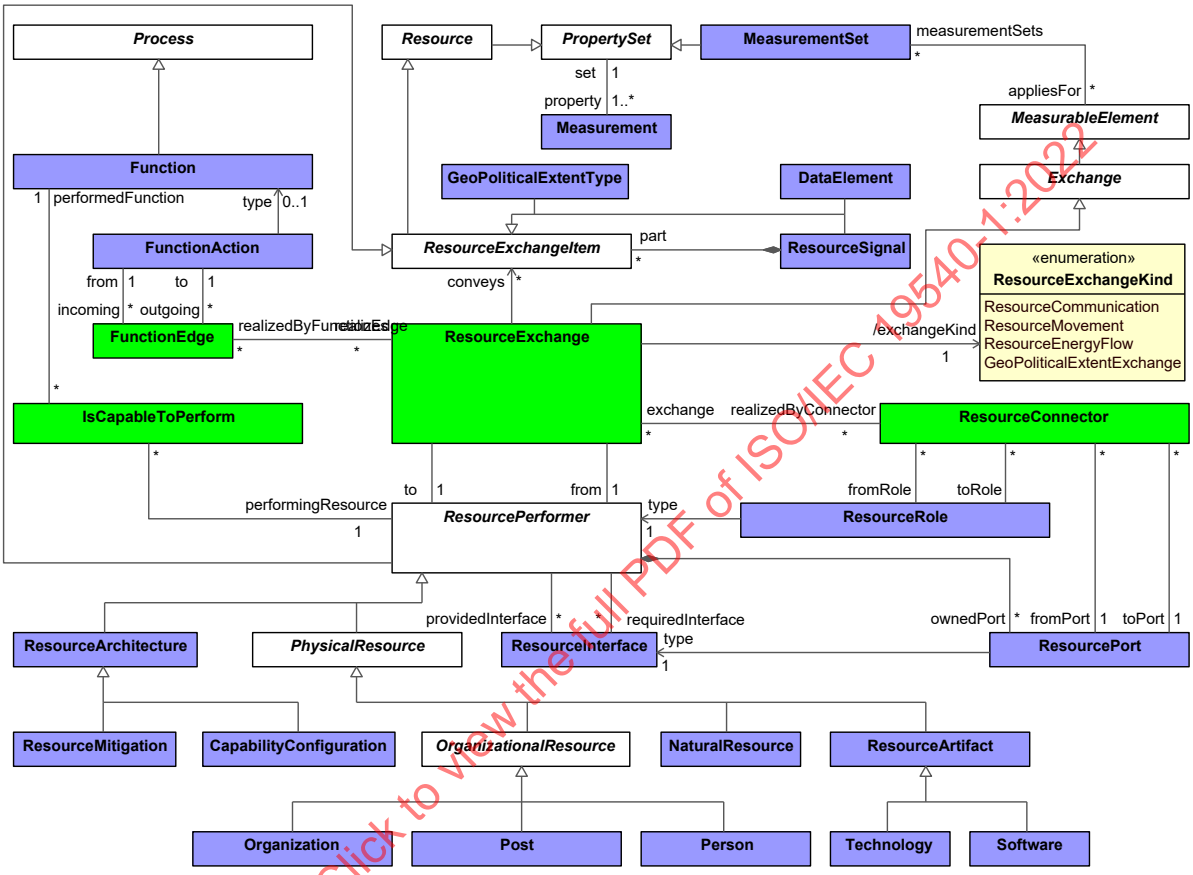

Figure 8:43 - Resources Connectivity

Elements

- CapabilityConfiguration
- DataElement
- Exchange
- Function
- FunctionAction
- FunctionEdge
- GeoPoliticalExtentType
- IsCapableToPerform
- MeasurableElement
- Measurement
- MeasurementSet
- NaturalResource
- Organization
- OrganizationalResource
- Person
- PhysicalResource
- Post
- Process

- PropertySet
- Resource
- ResourceArchitecture
- ResourceArtifact
- ResourceConnector
- ResourceExchange
- ResourceExchangeItem
- ResourceInterface
- ResourceMitigation
- ResourcePerformer
- ResourcePort
- ResourceRole
- ResourceSignal
- Software
- Technology

View Specifications::Resources::Processes

Contains the diagrams that document the Resources Processes Viewpoint.

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.

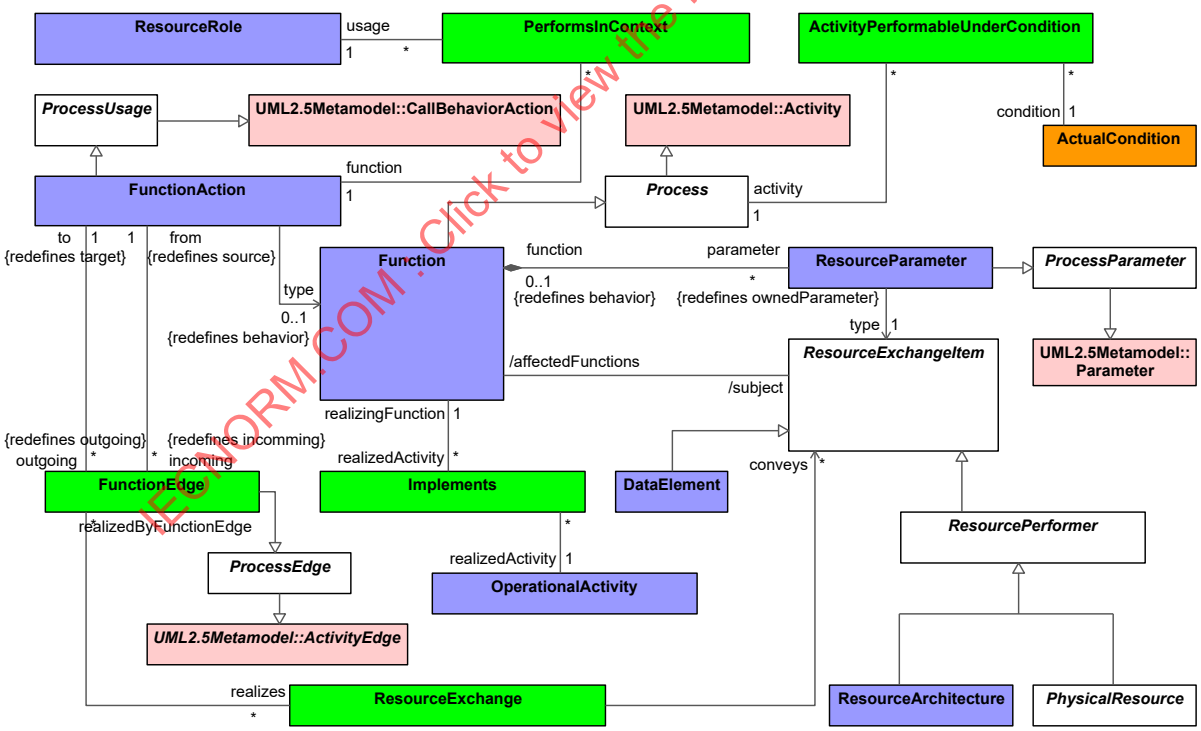

Figure 8:44 - Resources Processes

Elements

- ActivityPerformableUnderCondition
- ActualCondition
- DataElement
- Function
- FunctionAction
- FunctionEdge
- Implements
- OperationalActivity
- PerformsInContext
- PhysicalResource
- Process
- ProcessEdge
- ProcessParameter
- ProcessUsage
- ResourceArchitecture
- ResourceExchange
- ResourceExchangeItem
- ResourceParameter
- ResourcePerformer
- ResourceRole
- UML2.5Metamodel::Activity
- UML2.5Metamodel::ActivityEdge
- UML2.5Metamodel::CallBehaviorAction
- UML2.5Metamodel::Parameter

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.

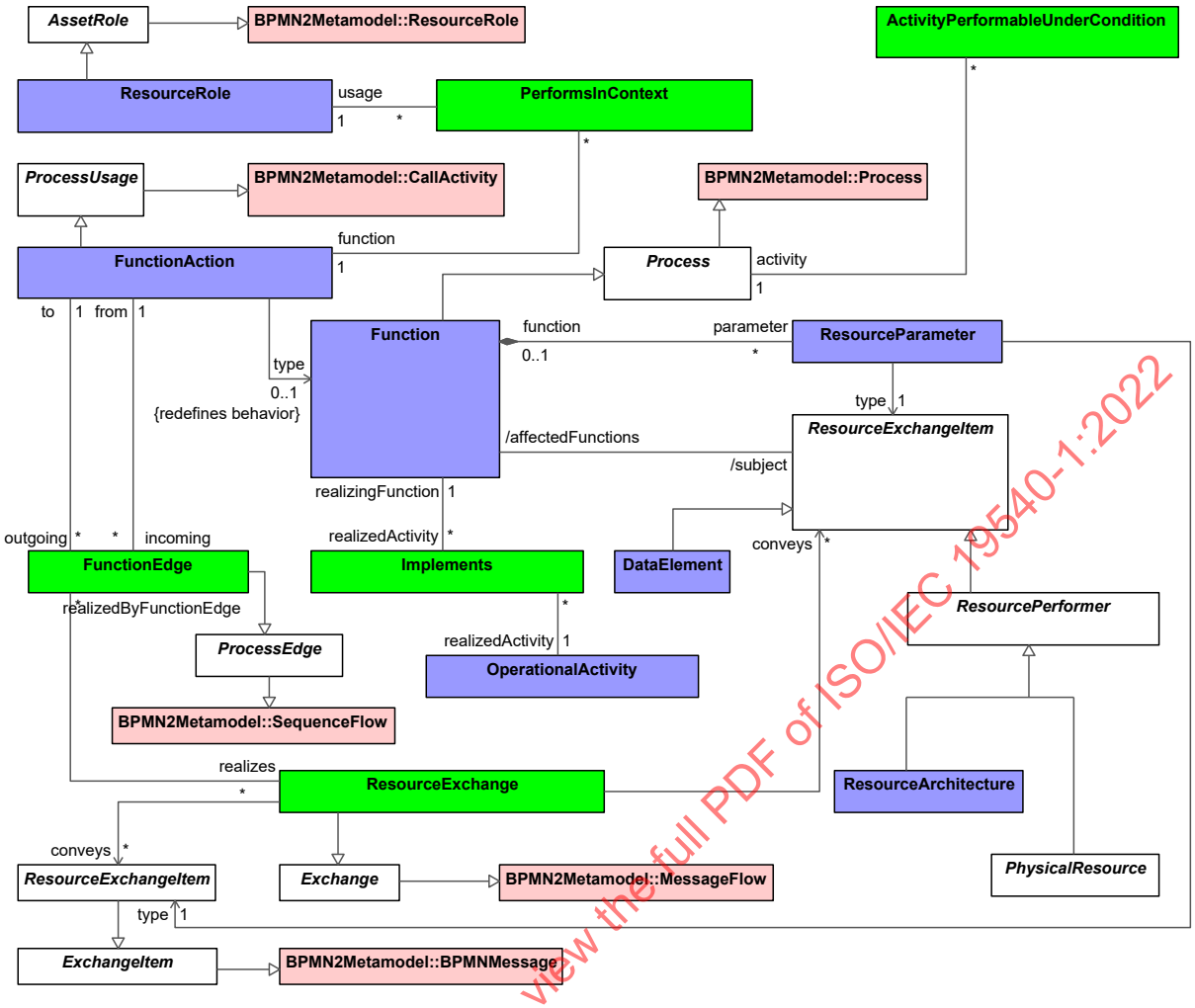

Figure 8:45 - Resources Processes BPMN Semantics

Elements

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

- ResourceExchange
- ResourceExchangeItem
- ResourceParameter
- ResourcePerformer
- ResourceRole

View Specifications::Resources::States

Contains the diagrams that document the Resources States Viewpoint.

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.

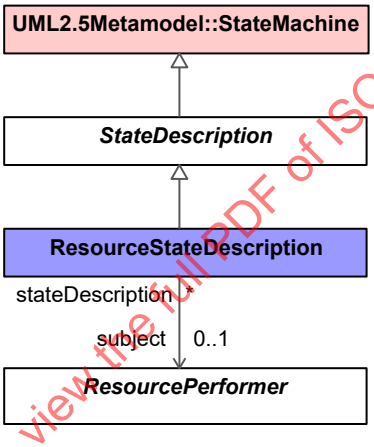

Figure 8:46 - Resources States

Elements

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

View Specifications::Resources::Interaction Scenarios

Contains the diagrams that document the Resources Interaction Scenarios Viewpoint.

View Specifications::Resources::Interaction Scenarios::Resources Interaction Scenarios

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.

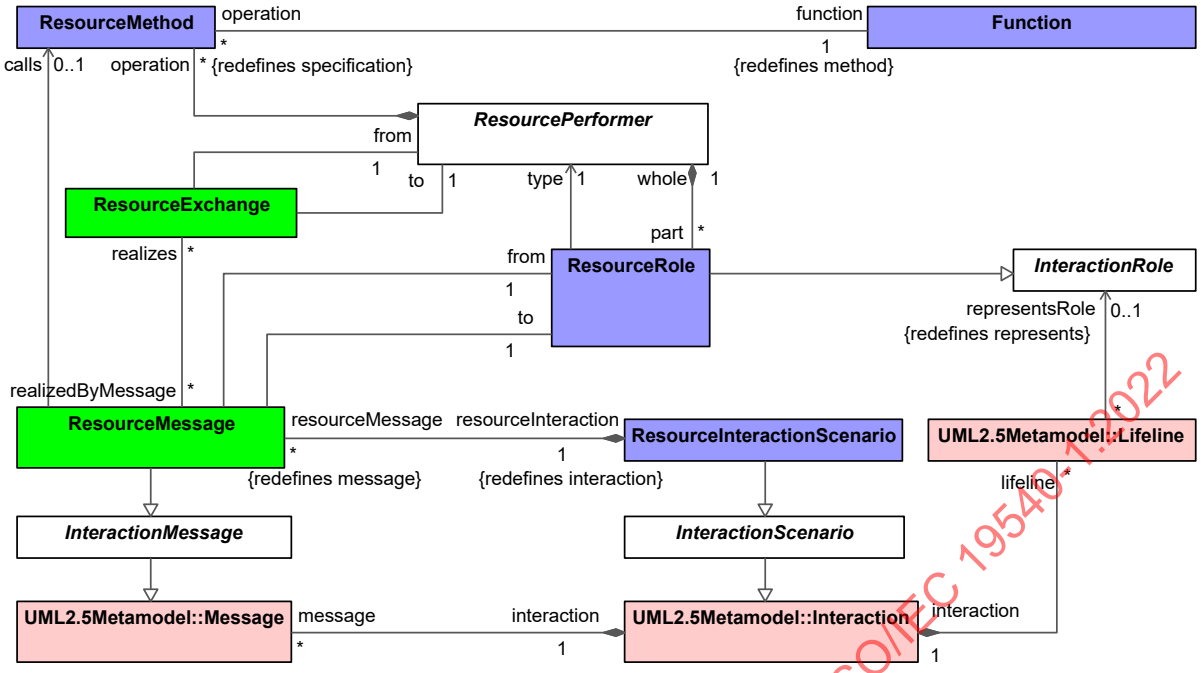

Figure 8:47 - Resources Interaction Scenarios

Elements

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

View Specifications::Resources::Constraints

Contains the diagrams that document the Resources Constraints Viewpoint.

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.

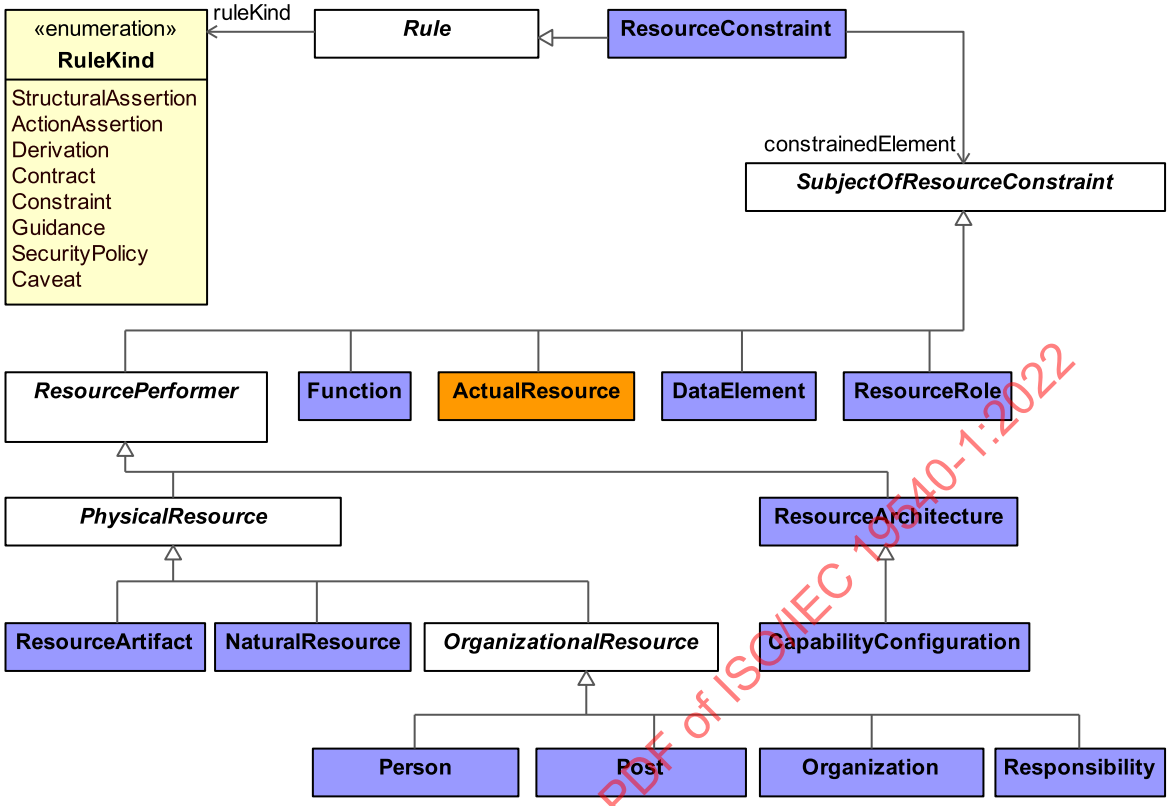

Figure 8:48 - Resources Constraints

Elements

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

View Specifications::Resources::Roadmap

Contains the diagrams that document the Resources Roadmap Viewpoint.

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.

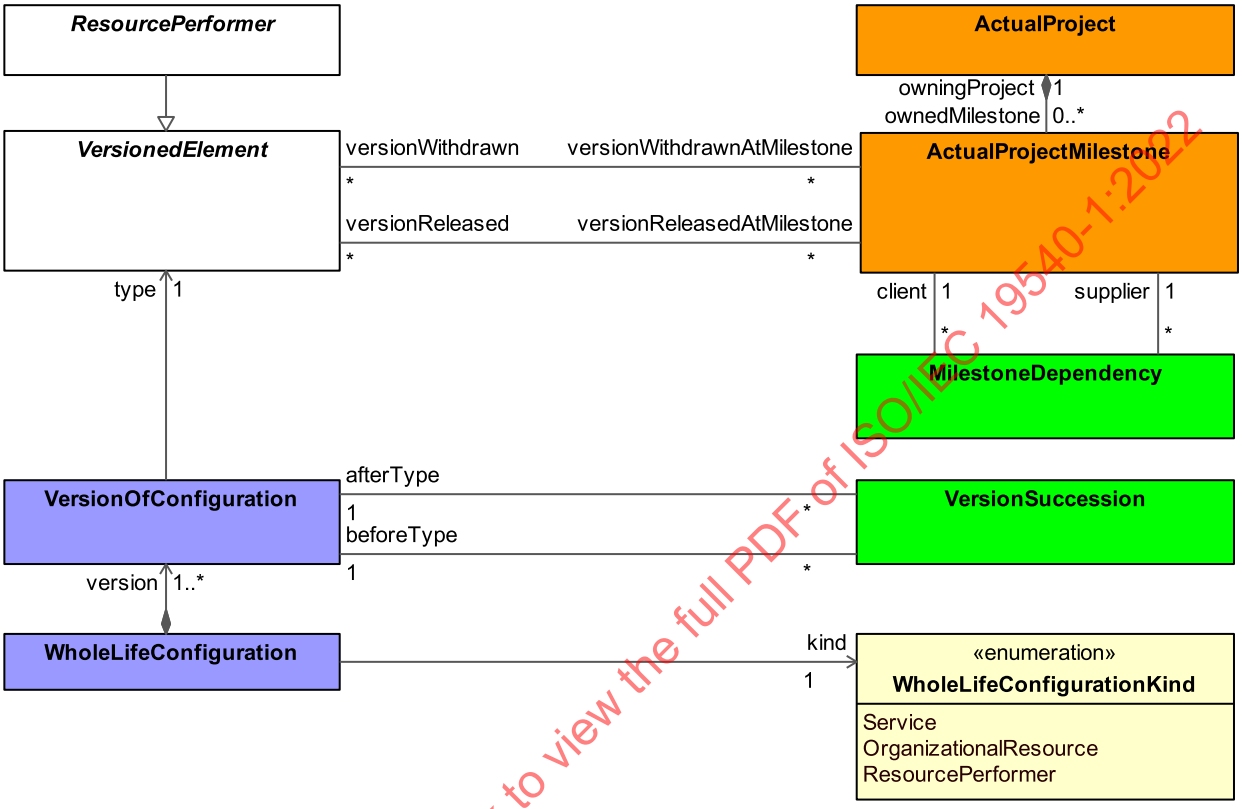

Figure 8.49 - Resources Roadmap: Evolution

Elements

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

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.

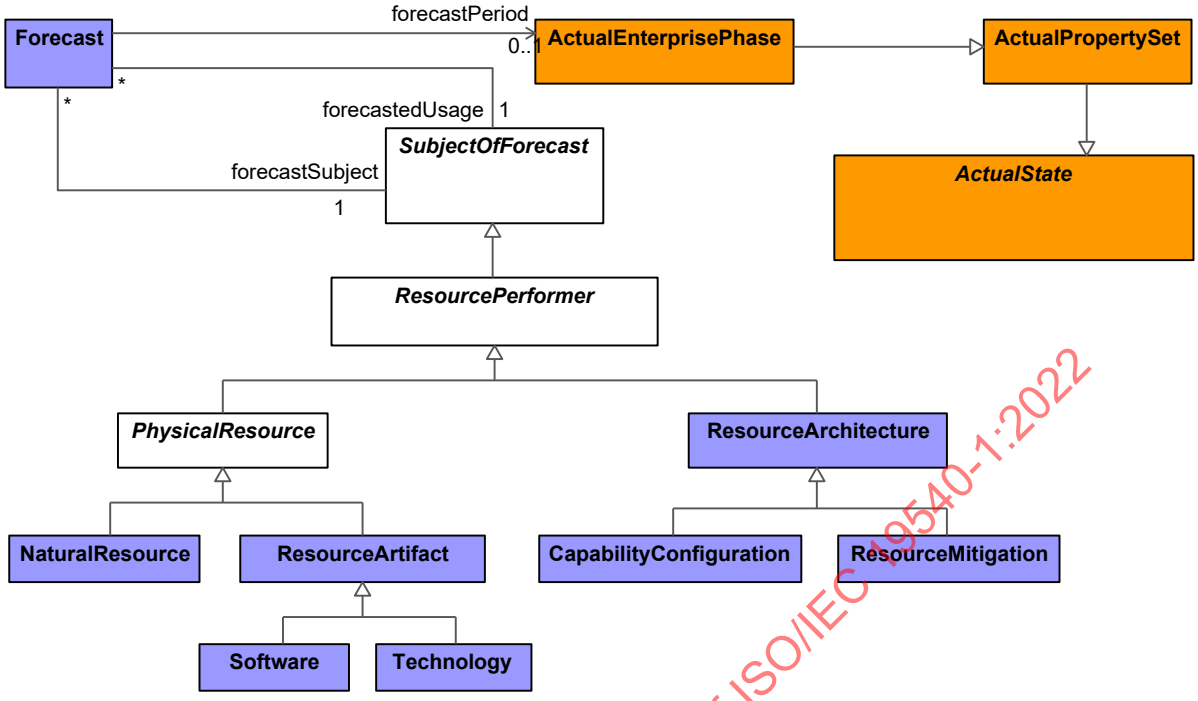

Figure 8:50 - Resources Roadmap: Forecast

Elements

- ActualEnterprisePhase
- ActualPropertySet
- ActualState
- CapabilityConfiguration
- Forecast
- NaturalResource
- PhysicalResource
- ResourceArchitecture
- ResourceArtifact
- ResourceMitigation
- ResourcePerformer
- Software
- SubjectOfForecast
- Technology

View Specifications::Resources::Traceability

Contains the diagrams that document the Resources Traceability Viewpoint.

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.

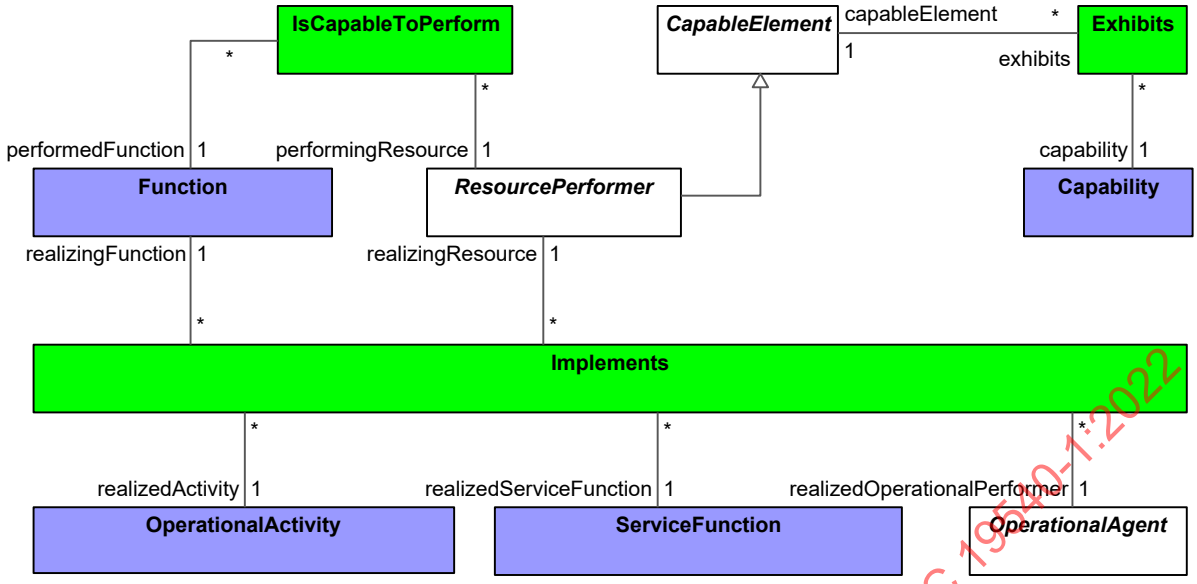

Figure 8:51 - Resources Traceability

Elements

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

8.1.7 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.

View Specifications::Security::Taxonomy

Contains the diagrams that document the Security Taxonomy Viewpoint.

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.

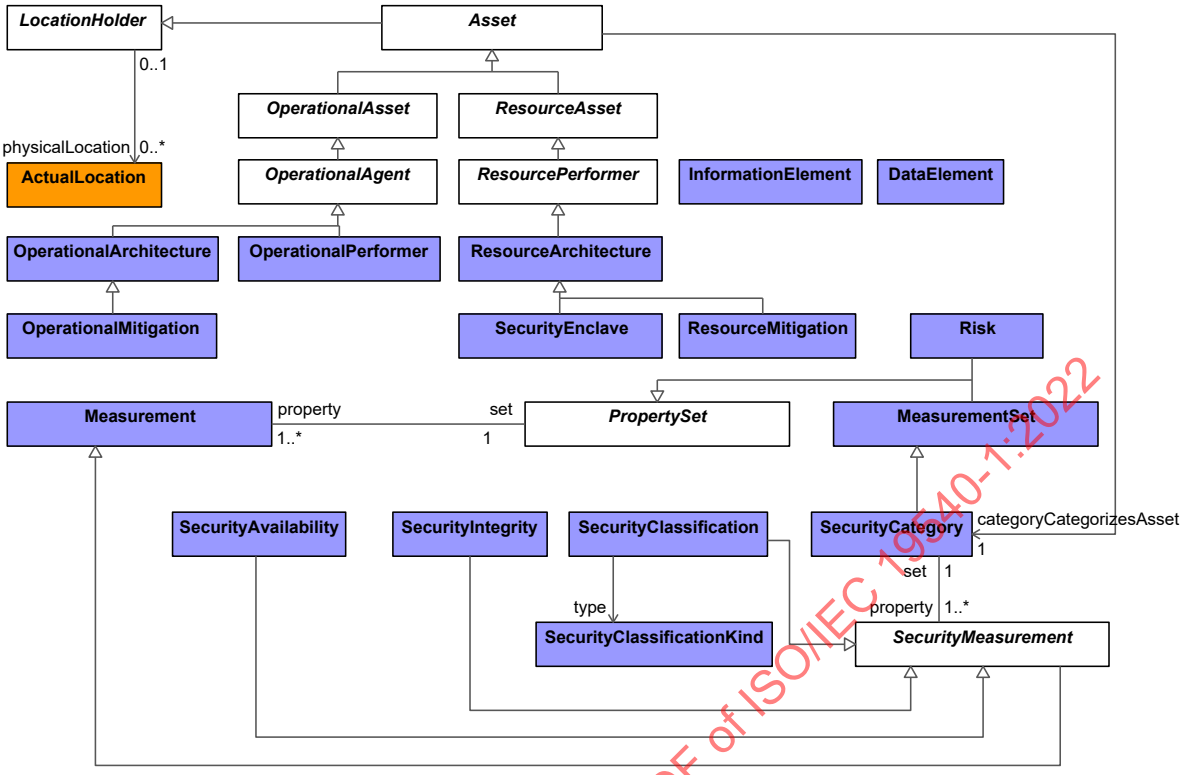

Figure 8:52 - Security Taxonomy

Elements

- ActualLocation
- Asset
- DataElement
- InformationElement
- LocationHolder
- Measurement
- MeasurementSet
- OperationalAgent
- OperationalArchitecture
- OperationalAsset
- OperationalMitigation
- OperationalPerformer
- PropertySet
- ResourceArchitecture
- ResourceAsset
- ResourceMitigation
- ResourcePerformer
- Risk
- SecurityAvailability
- SecurityCategory
- SecurityClassification
- SecurityClassificationKind
- SecurityEnclave
- SecurityIntegrity
- SecurityMeasurement

View Specifications::Security::Structure

Contains the diagrams that document the Security Structure Viewpoint.

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.

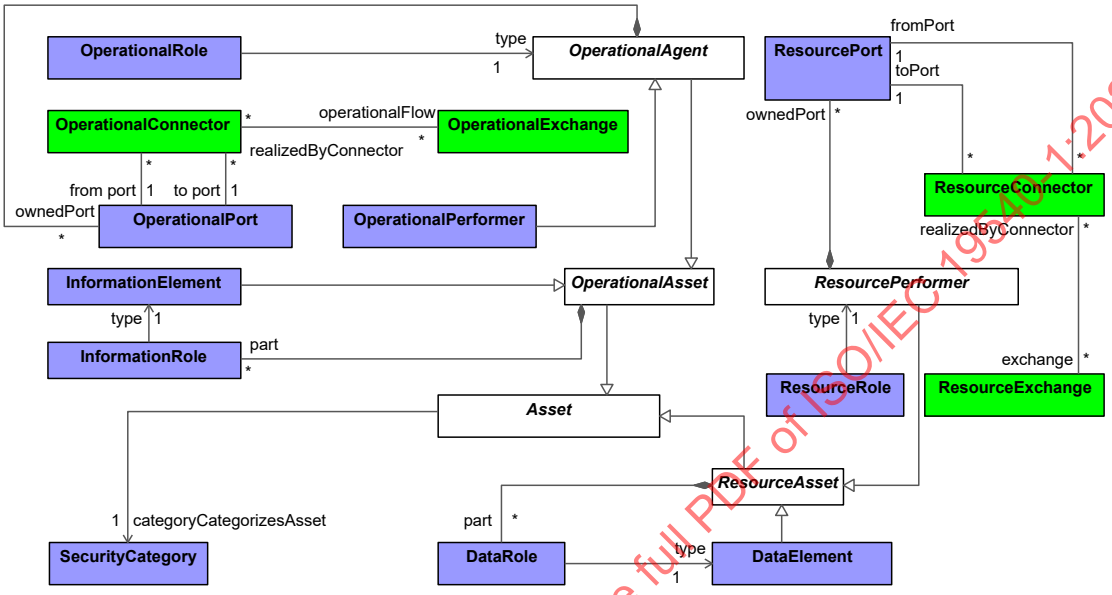

Figure 8:53 - Security Structure

Elements

- Asset
- DataElement
- DataRole
- InformationElement
- InformationRole
- OperationalAgent
- OperationalAsset
- OperationalConnector
- OperationalExchange
- OperationalPerformer
- OperationalPort
- OperationalRole
- ResourceAsset
- ResourceConnector
- ResourceExchange
- ResourcePerformer
- ResourcePort
- ResourceRole
- SecurityCategory

View Specifications::Security::Connectivity

Contains the diagrams that document the Security Connectivity Viewpoint.

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.

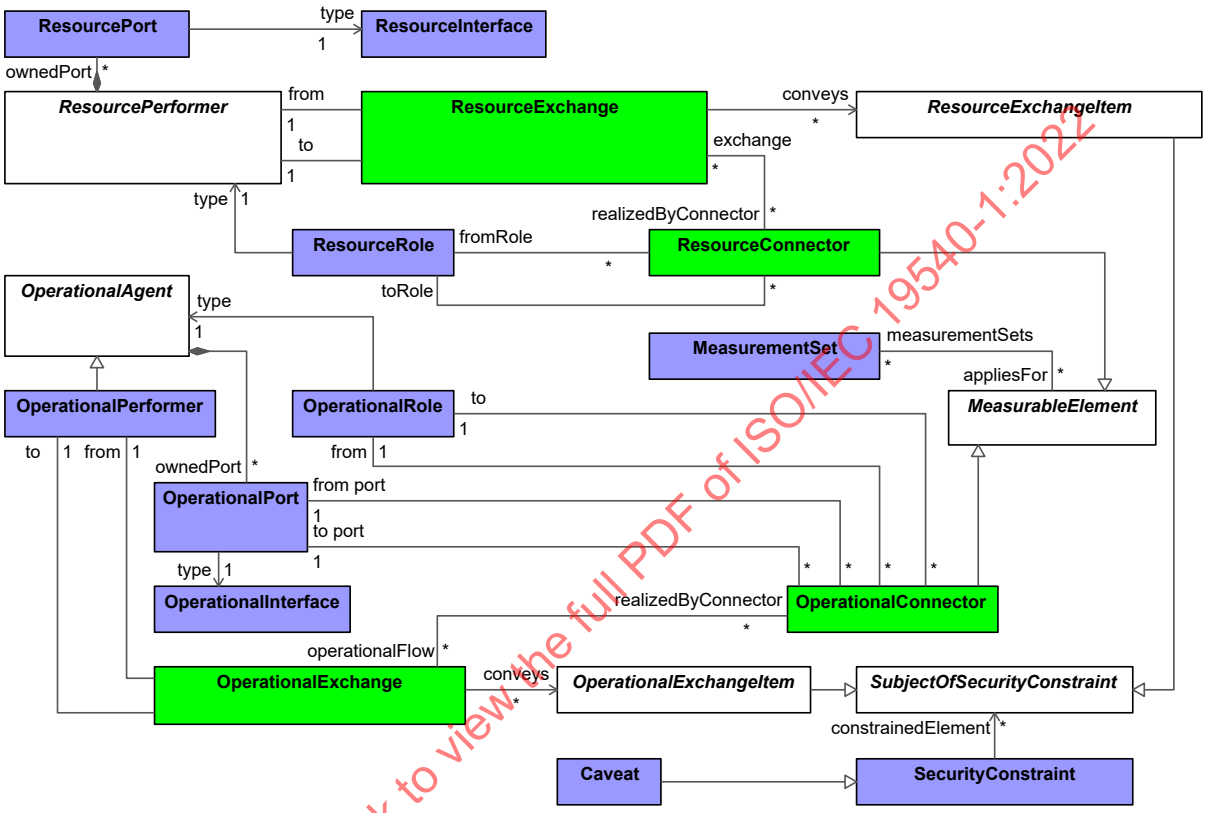

Figure 8:54 - Security Connectivity

Elements

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

View Specifications::Security::Processes

Contains the diagrams that document the Security Processes Viewpoint.

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.

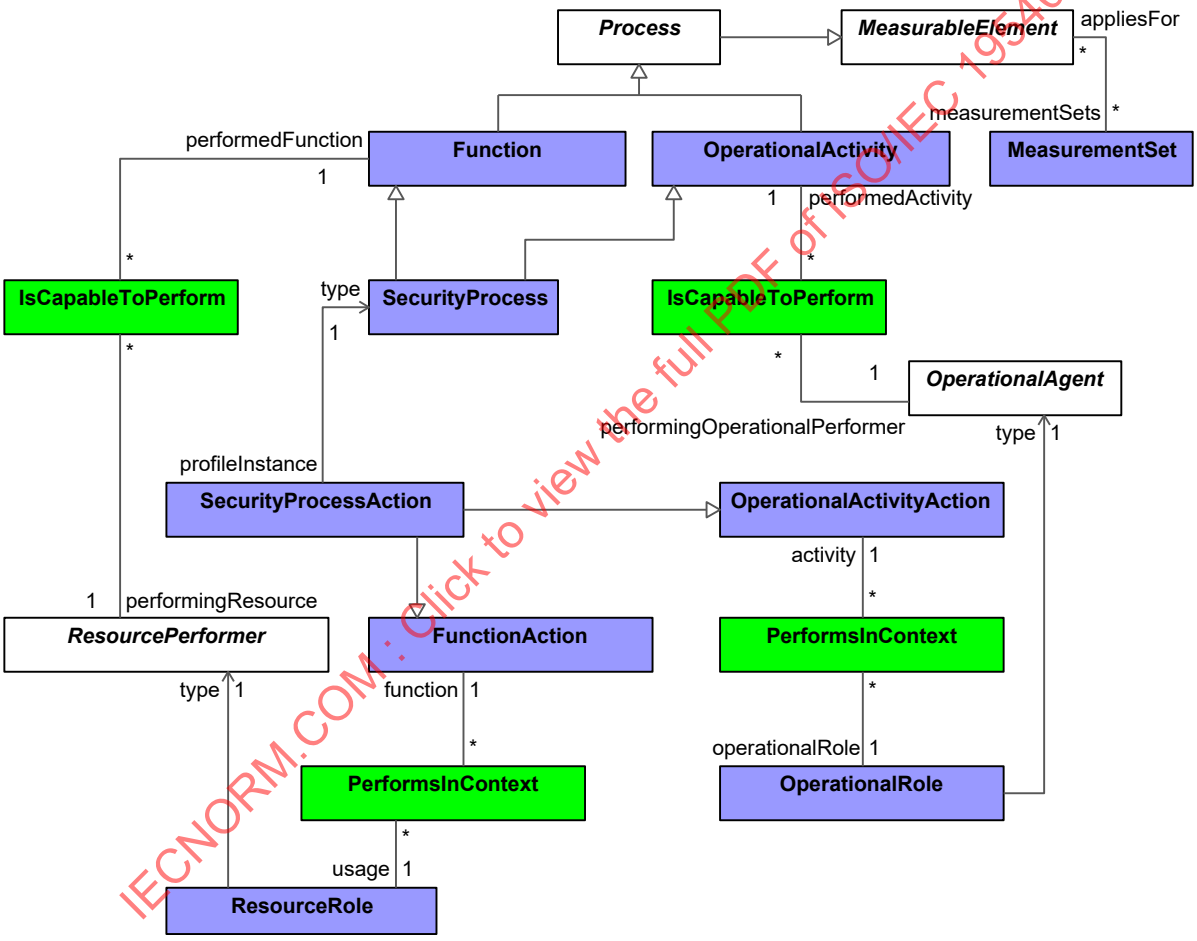

Figure 8:55 - Security Processes

Elements

- Function
- FunctionAction
- IsCapableToPerform
- MeasurableElement
- MeasurementSet
- OperationalActivity
- OperationalActivityAction
- OperationalAgent

- OperationalRole
- PerformsInContext
- Process
- ResourcePerformer
- ResourceRole
- SecurityProcess
- SecurityProcessAction

View Specifications::Security::Constraints

Contains the diagrams that document the Security Constraints Viewpoint.

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.

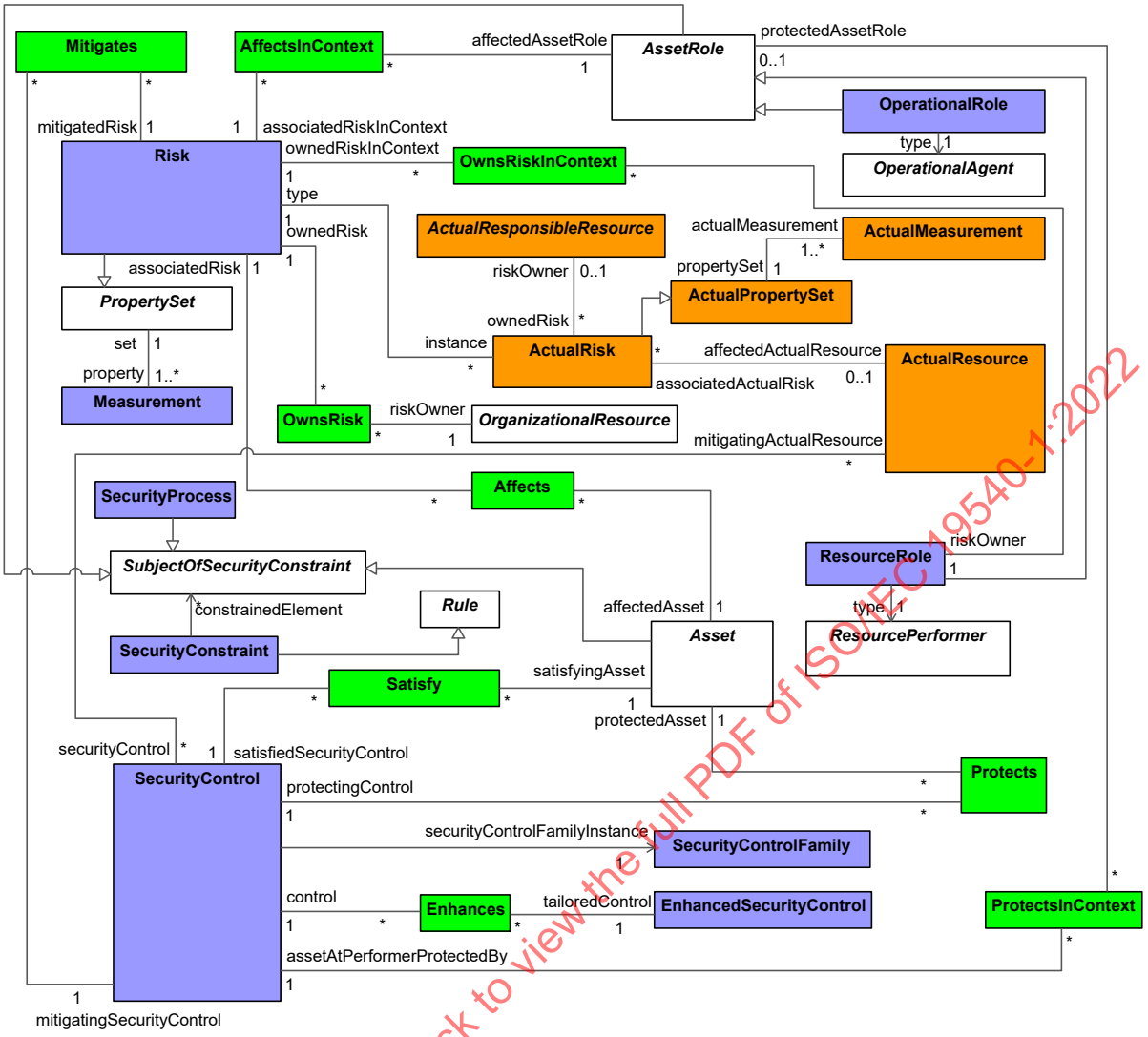

Figure 8:56 - Security Constraints

Elements

- ActualMeasurement
- ActualPropertySet
- ActualResource
- ActualResponsibleResource
- ActualRisk
- Affects
- AffectsInContext
- Asset
- AssetRole
- EnhancedSecurityControl
- Enhances
- Measurement
- Mitigates
- OperationalAgent
- OperationalRole
- OrganizationalResource
- OwnsRisk
- OwnsRiskInContext
- PropertySet

- Protects
- ProtectsInContext
- ResourcePerformer
- ResourceRole
- Risk
- Rule
- Satisfy
- SecurityConstraint
- SecurityControl
- SecurityControlFamily
- SecurityProcess
- SubjectOfSecurityConstraint

View Specifications::Security::Traceability

Contains the diagrams that document the Security Traceability Viewpoint.

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.

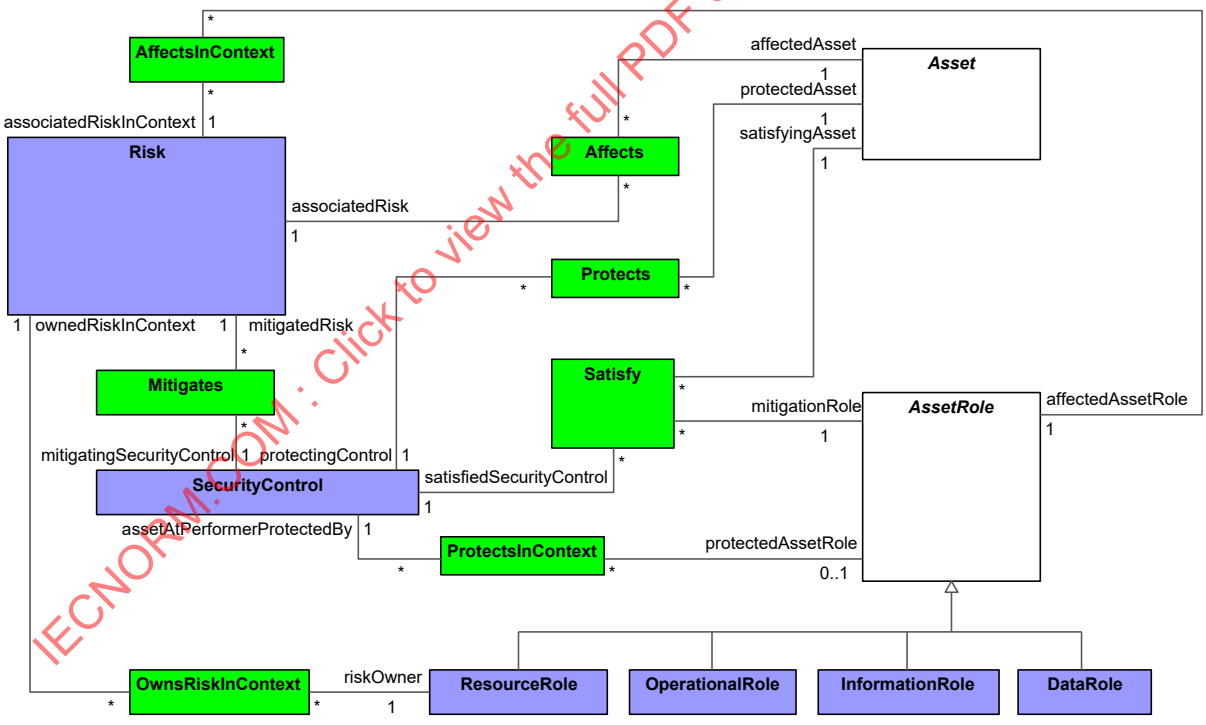

Figure 8:57 - Security Traceability

Elements

- Affects
- AffectsInContext
- Asset
- AssetRole
- DataRole
- InformationRole

- Mitigates
- OperationalRole
- OwnsRiskInContext
- Protects
- ProtectsInContext
- ResourceRole
- Risk
- Satisfy
- SecurityControl

8.1.8 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.

View Specifications::Projects::Taxonomy

Contains the diagrams that document the Project Taxonomy Viewpoint.

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.

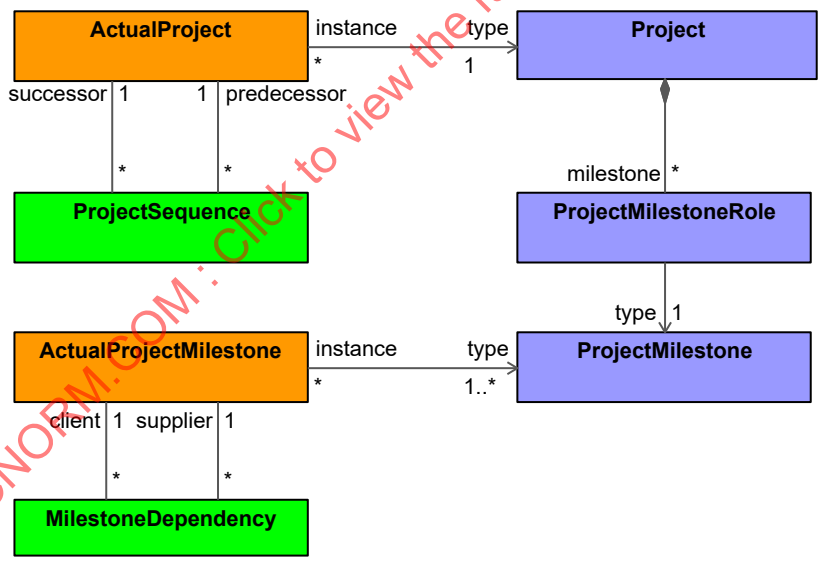

Figure 8:58 - Project Taxonomy

Elements

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

View Specifications::Projects::Structure

Contains the diagrams that document the Project Structure Viewpoint.

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.

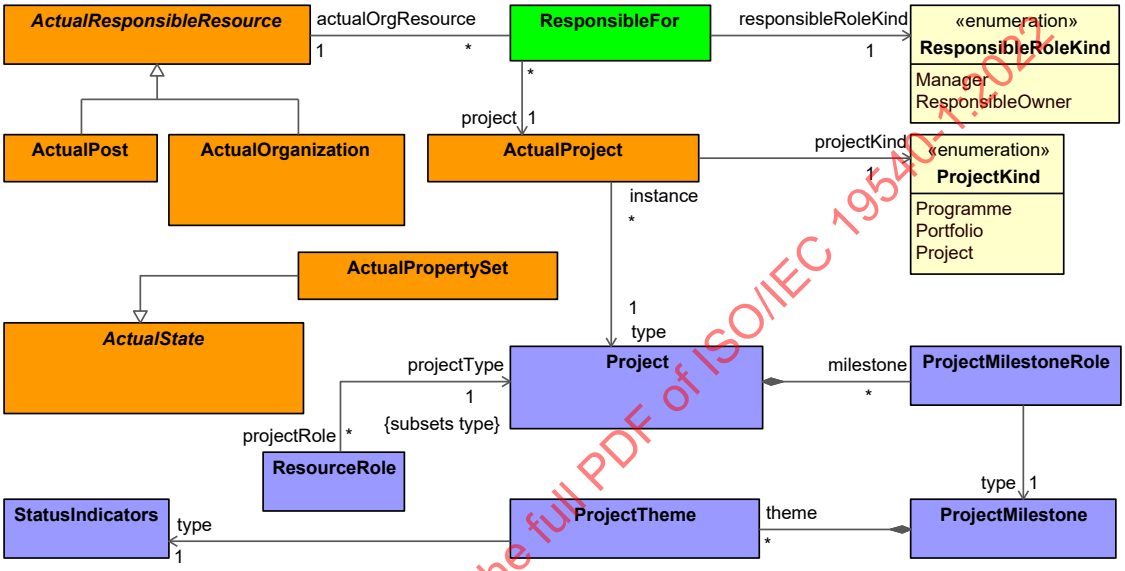

Figure 8:59 - Project Structure

Elements

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

View Specifications::Projects::Connectivity

Contains the diagrams that document the Project Connectivity Viewpoint.

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.

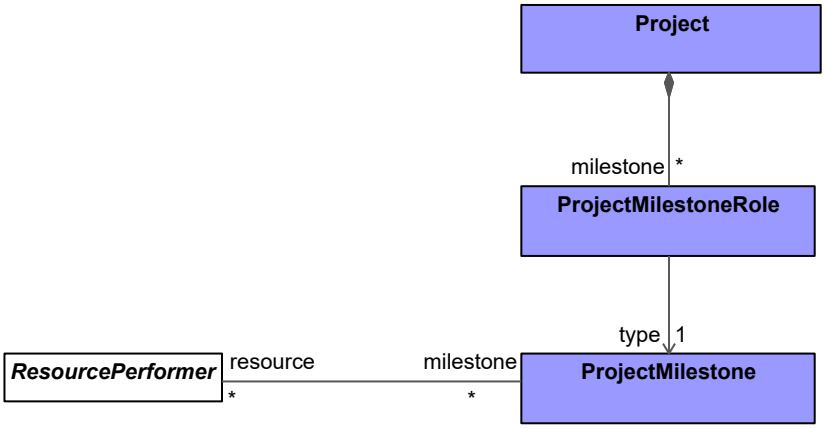

Figure 8:60 - Project Connectivity

Elements

- Project
- ProjectMilestone
- ProjectMilestoneRole
- ResourcePerformer

View Specifications::Projects::Processes

Contains the diagrams that document the Project Processes Viewpoint.

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.

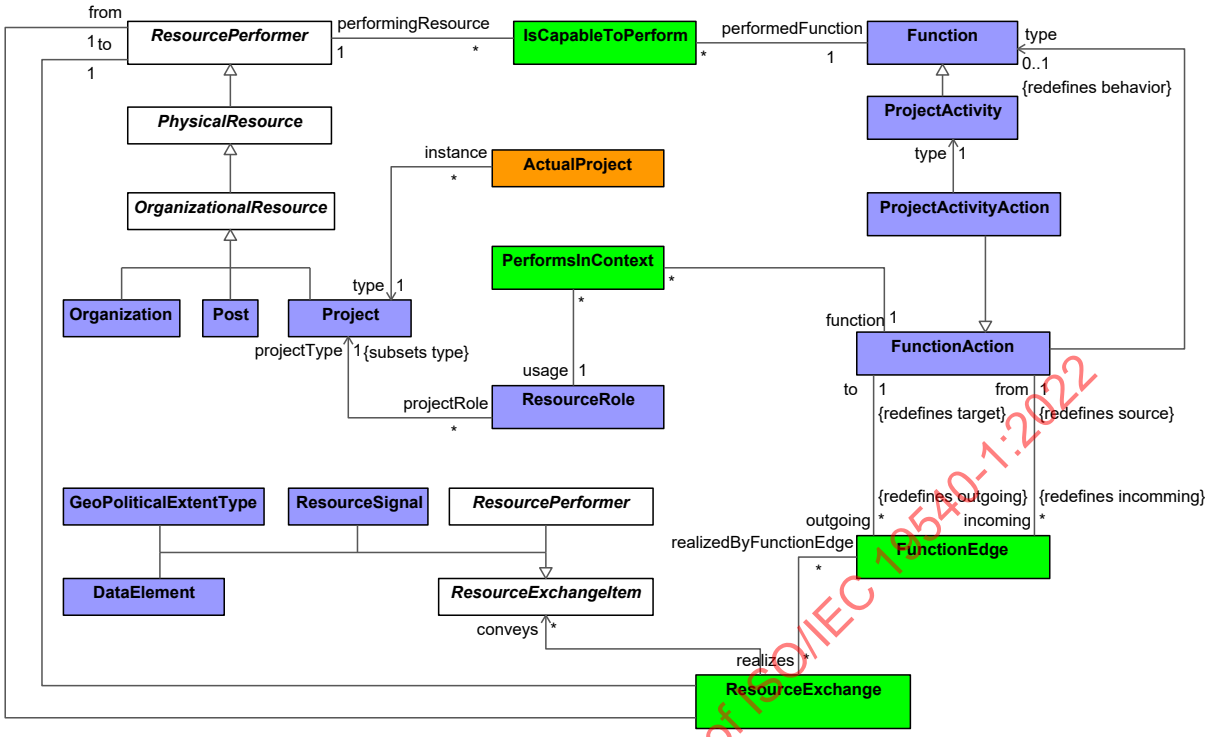

Figure 8:61 - Project Processes

Elements

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

View Specifications::Projects::Roadmap

Contains the diagrams that document the Project Roadmap Viewpoint.

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.

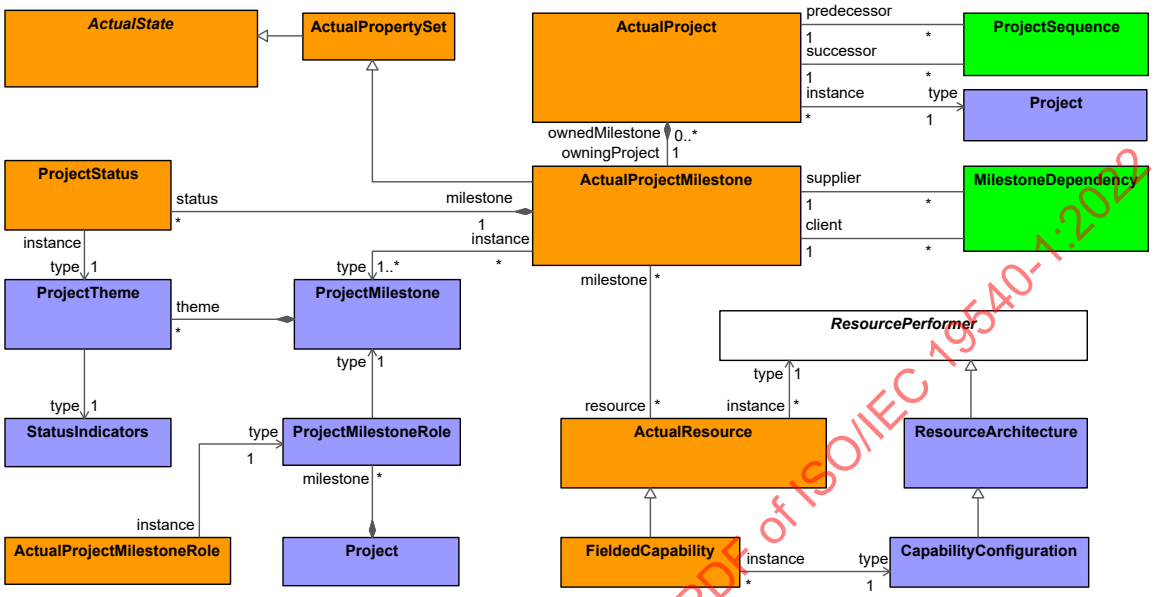

Figure 8:62 - Project Roadmap

Elements

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

View Specifications::Projects::Traceability

Contains the diagrams that document the Project Traceability Viewpoint.

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.

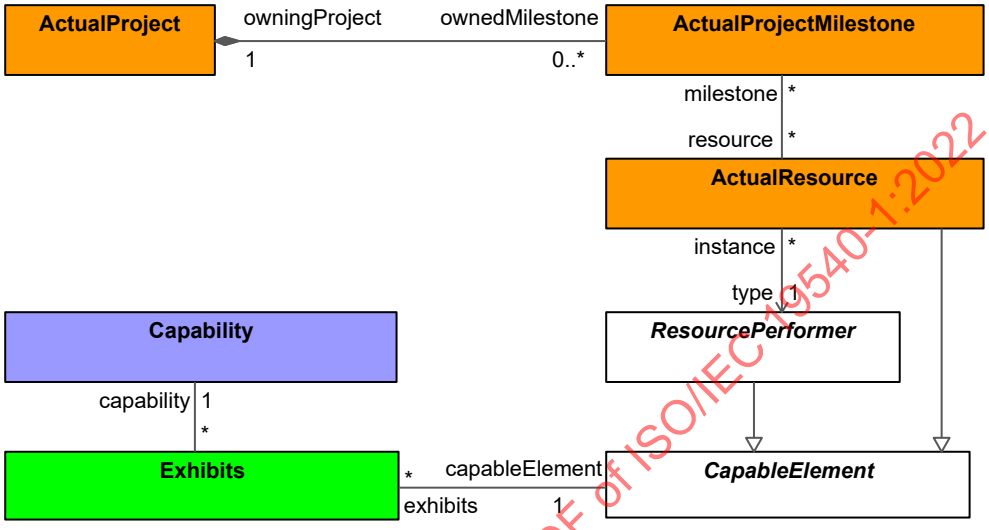

Figure 8:63 - Project Traceability

Elements

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

8.1.9 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.

View Specifications::Standards::Taxonomy

Contains the diagrams that document the Standards Taxonomy Viewpoint.

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.

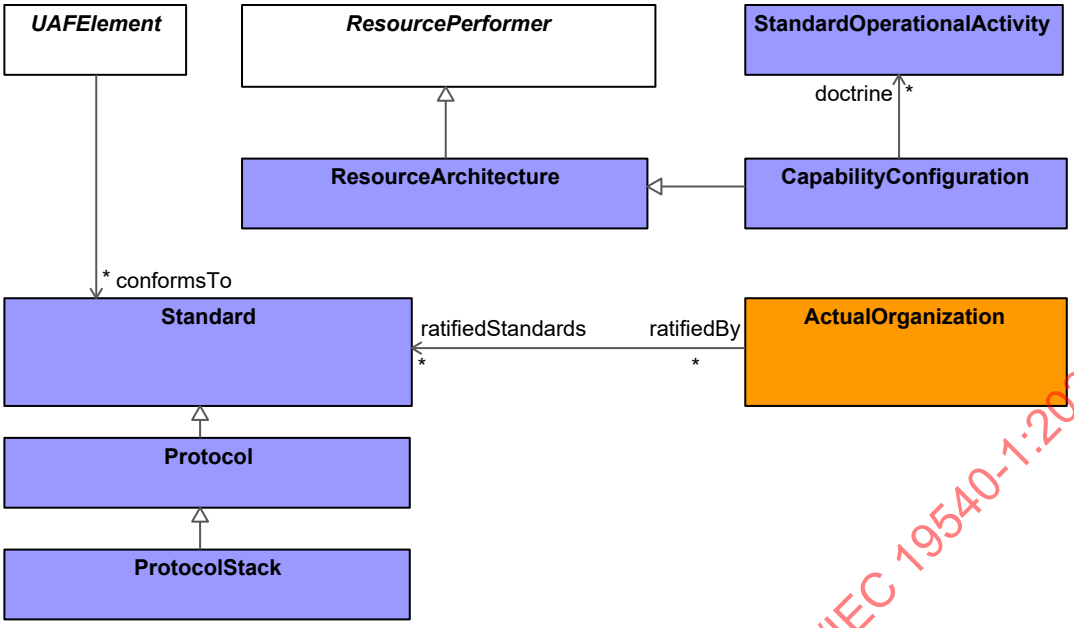

Figure 8:64 - Standards Taxonomy

Elements

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

View Specifications::Standards::Structure

Contains the diagrams that document the Standards Structure Viewpoint.

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.

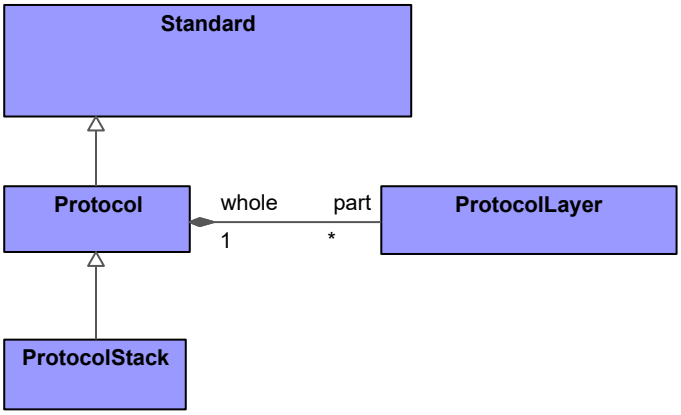

Figure 8:65 - Standards Structure

Elements

- Protocol
- ProtocolLayer
- ProtocolStack
- Standard

View Specifications::Standards::Roadmap

Contains the diagrams that document the Standards Roadmap Viewpoint.

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.

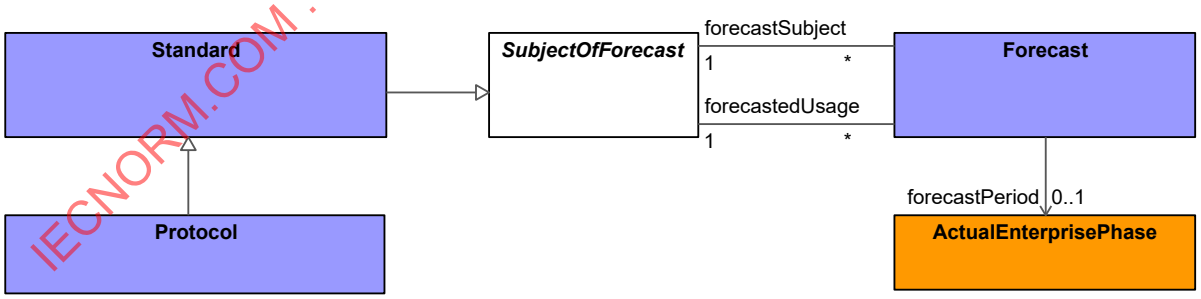

Figure 8:66 - Standards Roadmap

Elements

- ActualEnterprisePhase
- Forecast
- Protocol
- Standard
- SubjectOfForecast

View Specifications::Standards::Traceability

Contains the diagrams that document the Standards Traceability Viewpoint.

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.

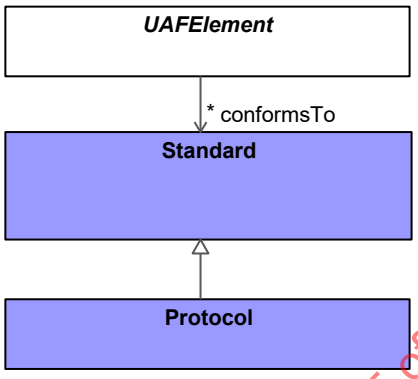

Figure 8:67 - Standards Traceability

Elements

- Protocol
- Standard
- UAFElement

8.1.10 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.

View Specifications::Actual Resources::Structure

Contains the diagrams that document the Actual Resources Structure Viewpoint.

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.

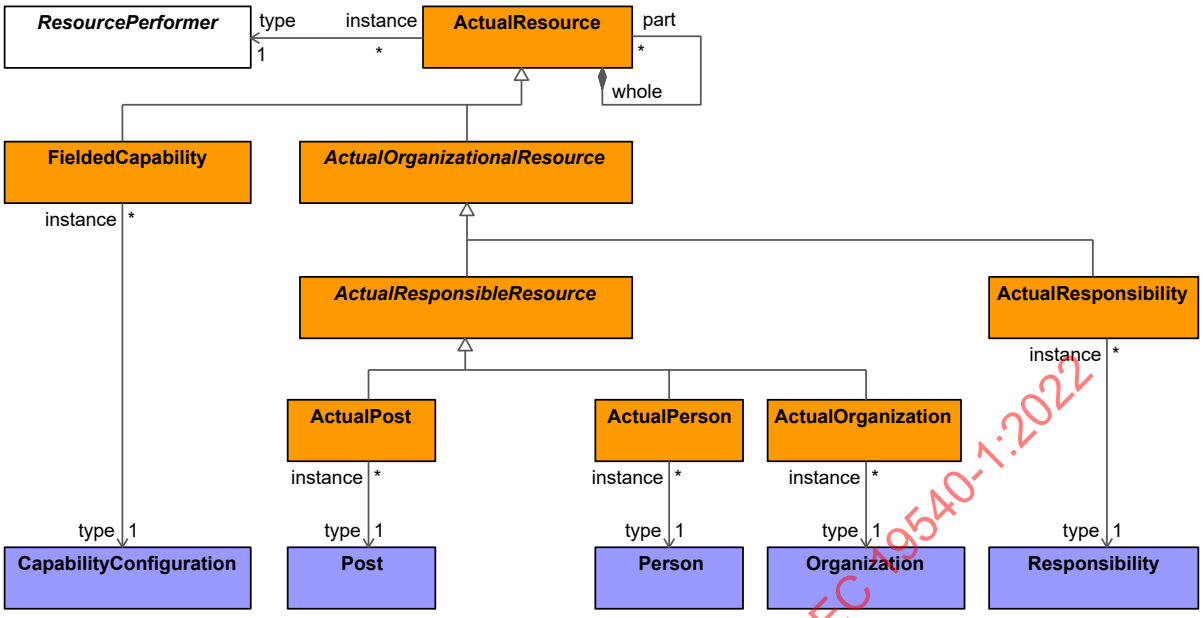

Figure 8:68 - Actual Resources Structure

Elements

- ActualOrganization
- ActualOrganizationalResource
- ActualPerson
- ActualPost
- ActualResource
- ActualResponsibility
- ActualResponsibleResource
- CapabilityConfiguration
- FieldedCapability
- Organization
- Person
- Post
- ResourcePerformer
- Responsibility

View Specifications::Actual Resources::Connectivity

Contains the diagrams that document the Actual Resources Connectivity Viewpoint.

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.

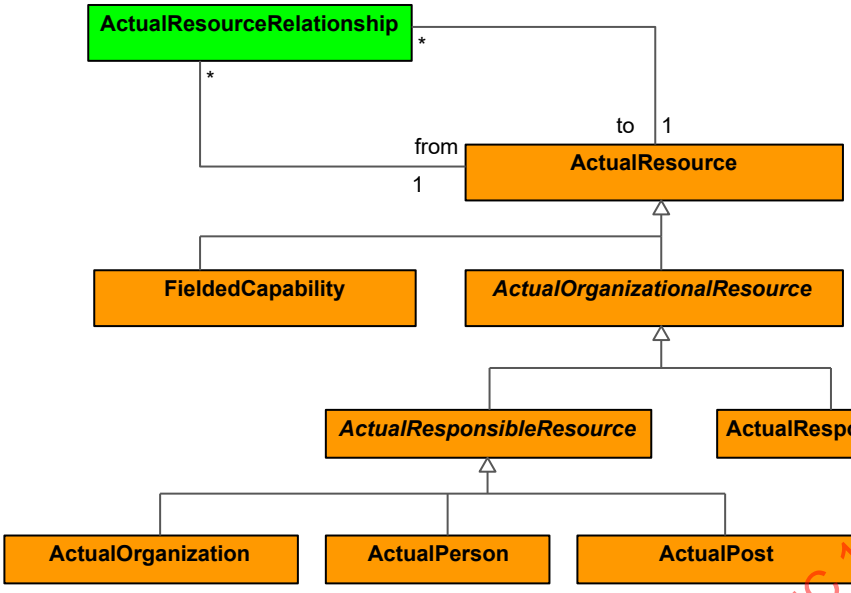

Figure 8:69 - Actual Resources Connectivity

Elements

- ActualOrganization
- ActualOrganizationalResource
- ActualPerson
- ActualPost
- ActualResource
- ActualResourceRelationship
- ActualResponsibility
- ActualResponsibleResource
- FieldedCapability

View Specifications::Actual Resources::Traceability

Contains the diagrams that document the Actual Resources Traceability Viewpoint.

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.

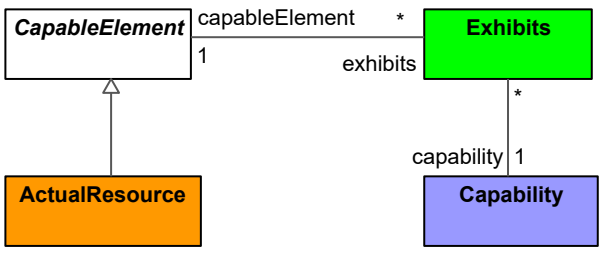

Figure 8:70 - Actual Resources Traceability

Elements

- ActualResource
- Capability
- CapableElement
- Exhibits

8.1.11 View Specifications::Dictionary

Stakeholders: Architects, users of the architecture, Capability Owners, Systems Engineers, Solution Providers.
Concerns: Definitions for all the elements in the architecture, libraries of environments and measurements.
Definition: Presents all the elements used in an architecture. Can be used specifically to capture:
a. Elements and relationships that are involved in defining the environments applicable to capability, operational concept or set of systems.
b. Measurable properties that can be used to support analysis such as KPIs, MoEs, TPIs etc.

View Specifications::Dictionary::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.

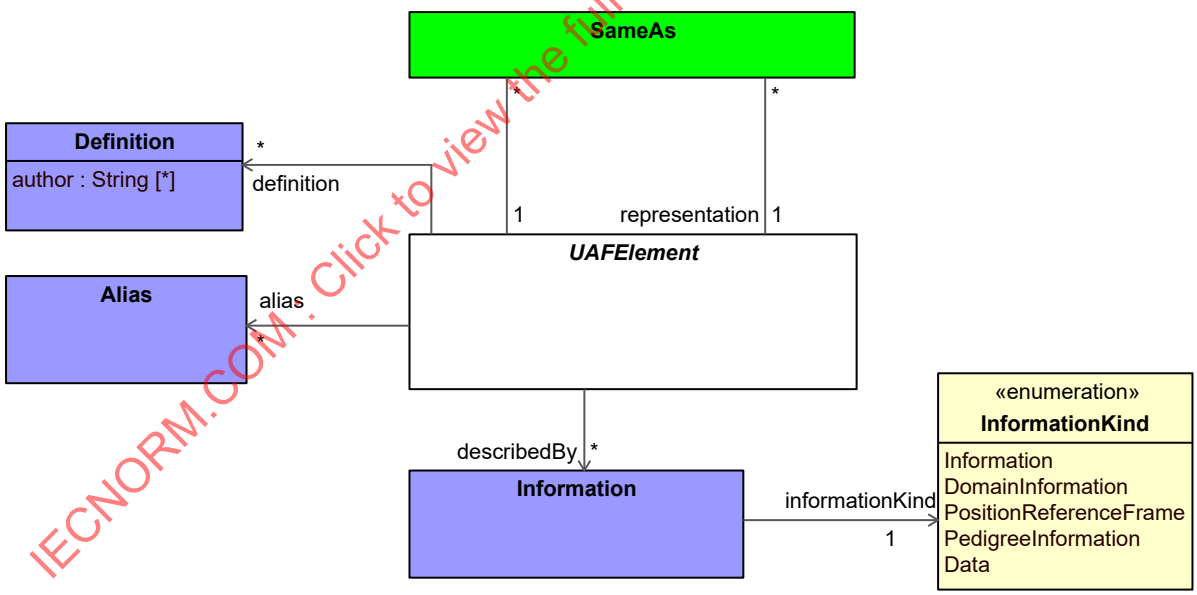

Figure 8:71 - Dictionary

Elements

- Alias
- Definition
- Information
- SameAs
- UAFElement

8.1.12 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 program.

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.

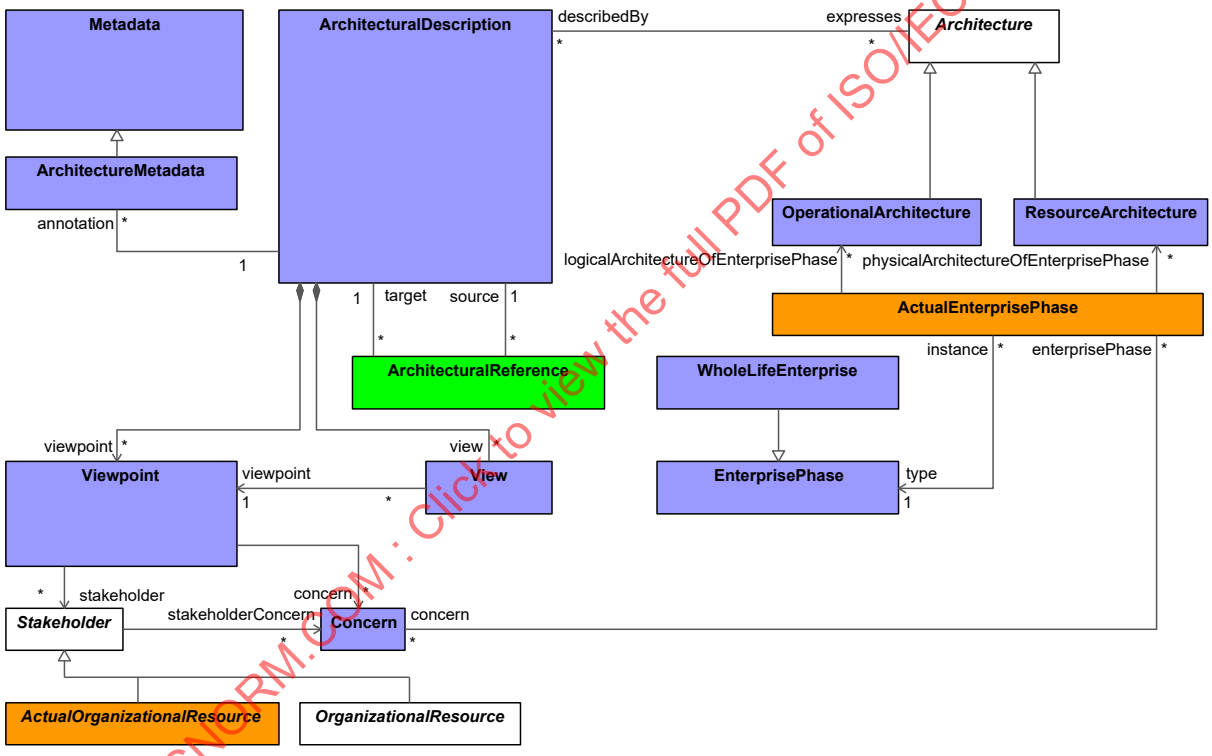

Figure 8:72 - Summary & Overview

Elements

- ActualEnterprisePhase
- ActualOrganizationalResource
- ArchitecturalDescription
- ArchitecturalReference
- Architecture
- ArchitectureMetadata
- Concern
- EnterprisePhase
- Metadata
- OperationalArchitecture
- OrganizationalResource

- ResourceArchitecture
- Stakeholder
- View
- Viewpoint
- WholeLifeEnterprise

8.1.13 View Specifications::Requirements

Stakeholders: Requirement Engineers, Solution Providers, Systems Engineers, Software Engineers, Systems Architects, Business Architects.
Concerns: requirements traceability.
Definition: used to represent requirements, their properties, and relationships (trace, verify, satisfy, refine) to UAF architectural elements.

View Specifications::Requirements::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.

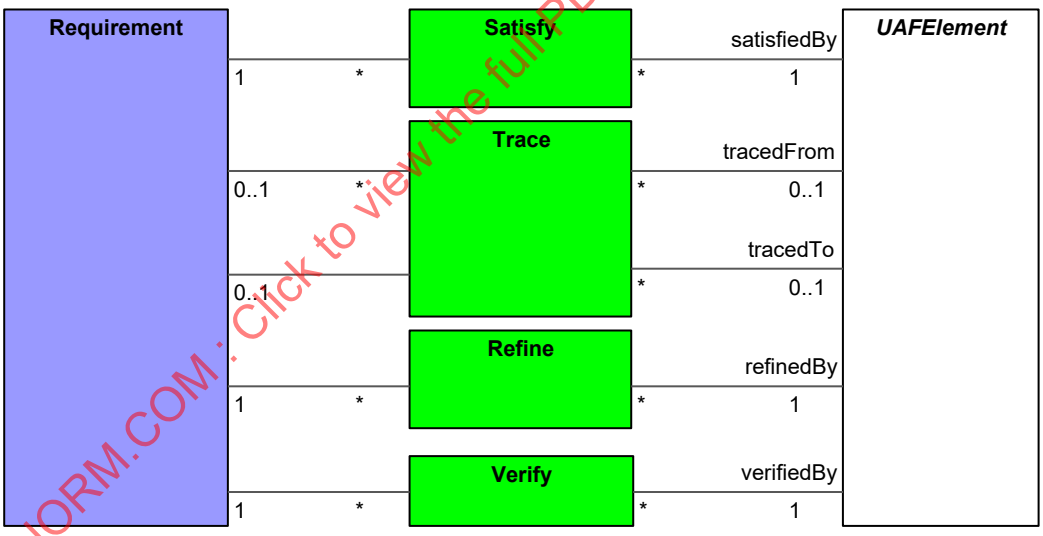

Figure 8:73 - Requirements

Elements

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

8.1.14 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.

View Specifications::Information::Information Model

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.

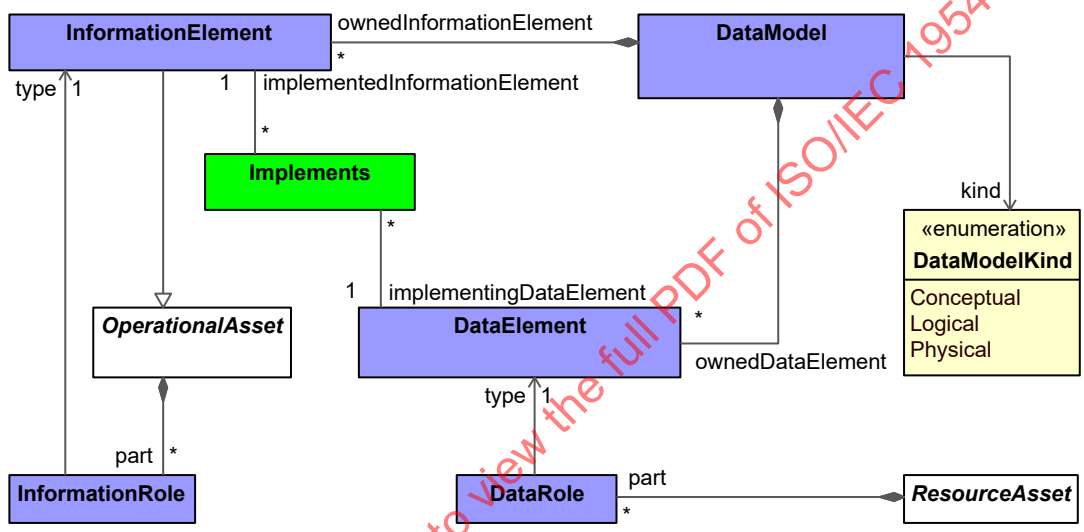

Figure 8:74 - Information Model

Elements

- DataElement
- DataModel
- DataRole
- Implements
- InformationElement
- InformationRole
- OperationalAsset
- ResourceAsset

8.1.15 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.

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.

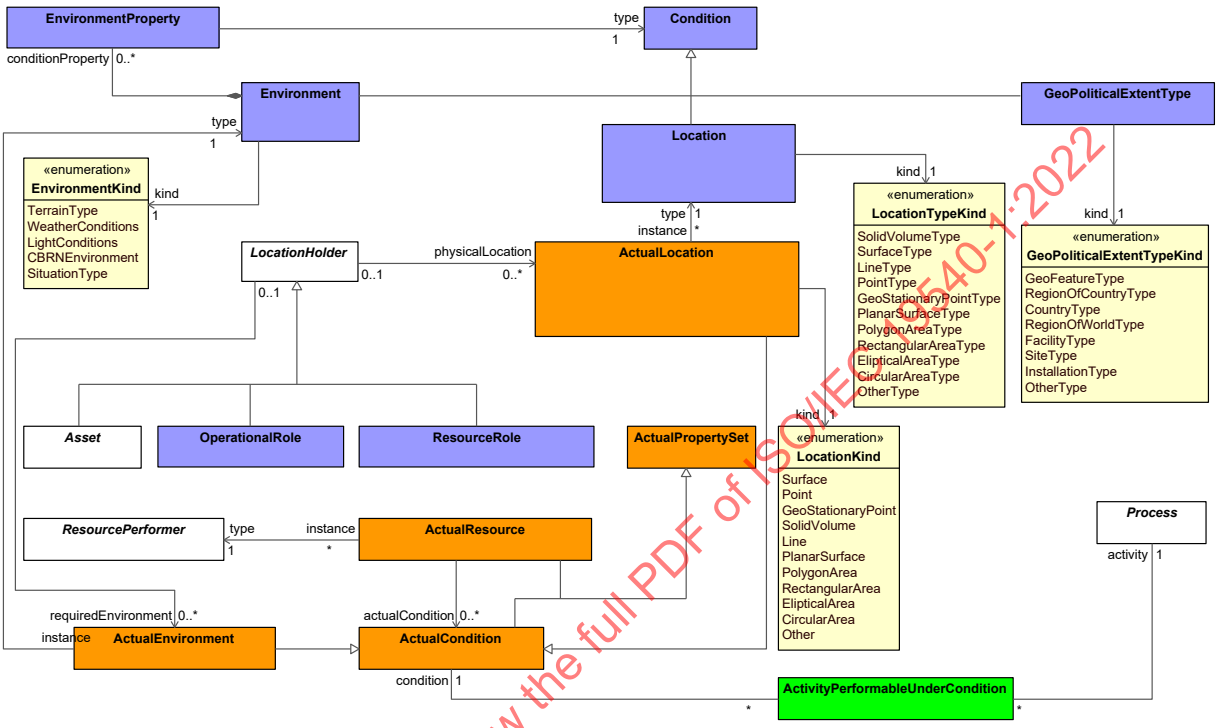

Figure 8:75 - Parameters: Environment

Elements

- ActivityPerformableUnderCondition
- ActualCondition
- ActualEnvironment
- ActualLocation
- ActualPropertySet
- ActualResource
- Asset
- Condition
- Environment
- EnvironmentProperty
- GeoPoliticalExtentType
- Location
- LocationHolder
- OperationalRole
- Process
- ResourcePerformer
- ResourceRole

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.

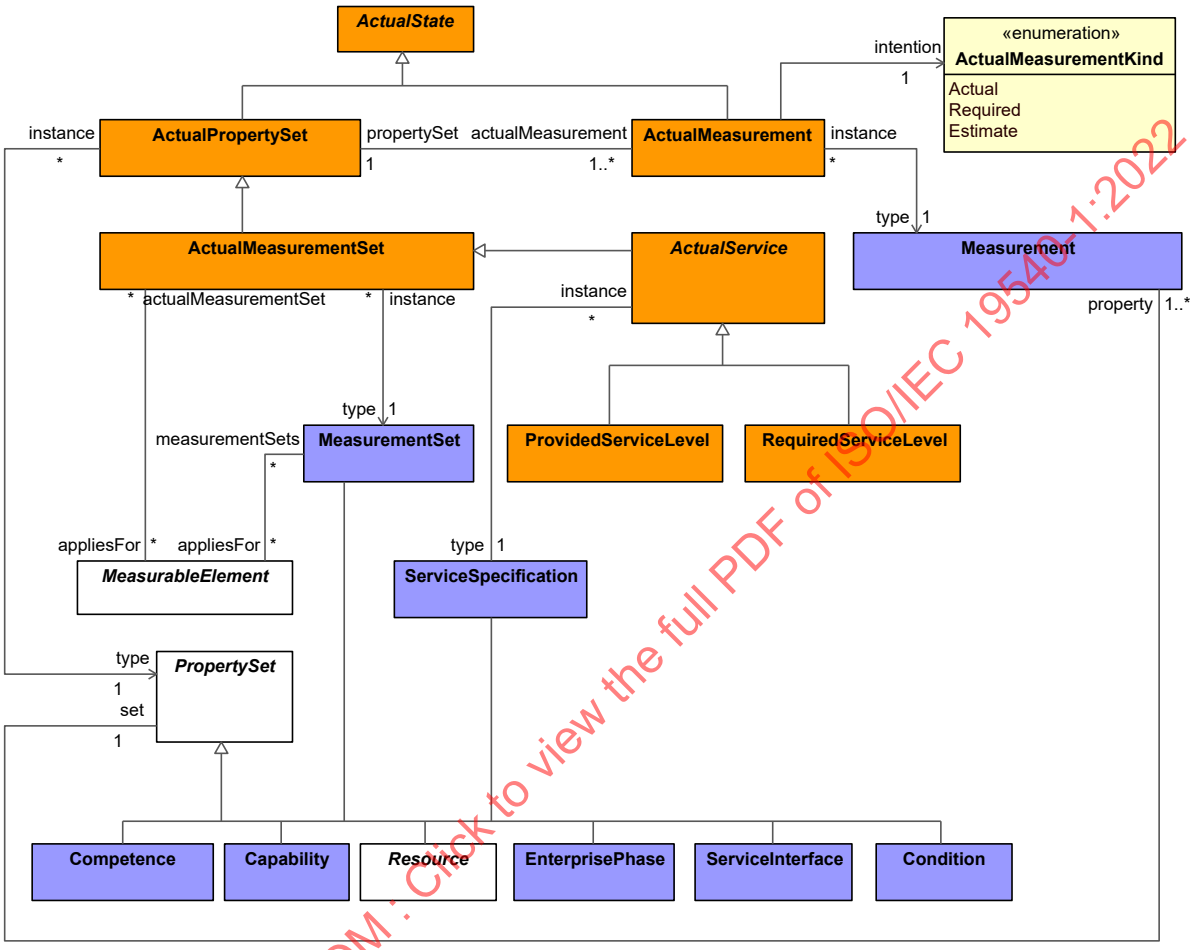

Figure 8:76 - Parameters: Measurements

Elements

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

- ServiceInterface
- ServiceSpecification

8.1.16 View Specifications::Other

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

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.

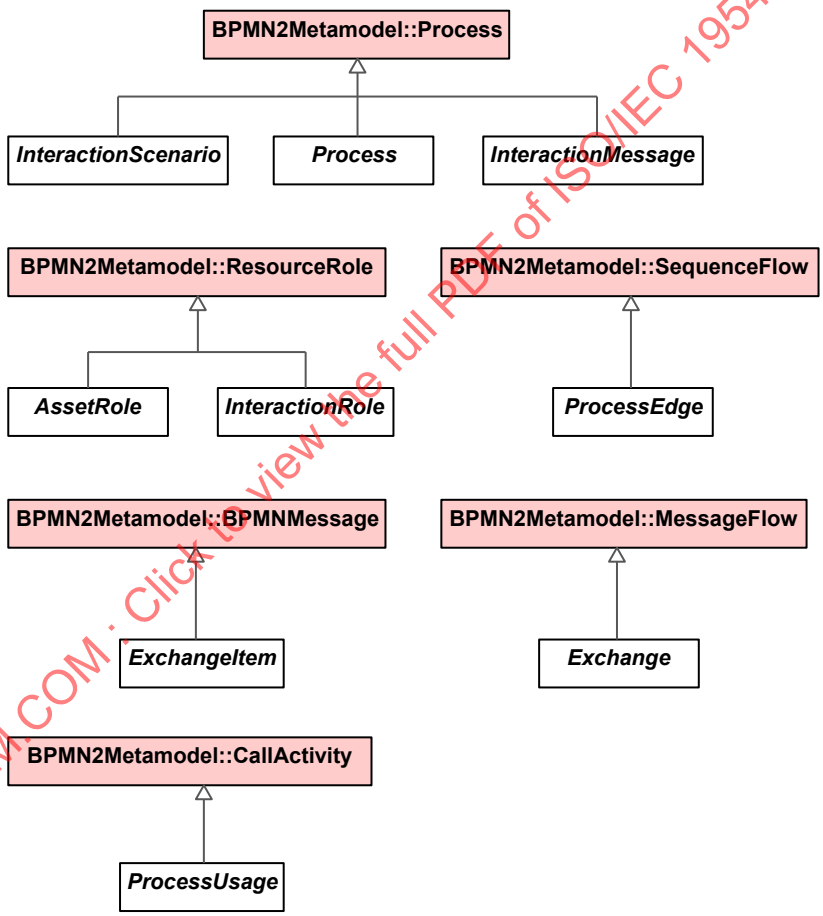

Figure 8:77 - BPMN

Elements

- AssetRole
- BPMN2Metamodel::BPMNMessage
- BPMN2Metamodel::CallActivity
- BPMN2Metamodel::MessageFlow
- BPMN2Metamodel::Process
- BPMN2Metamodel::ResourceRole
- BPMN2Metamodel::SequenceFlow
- Exchange

- ExchangeItem
- InteractionMessage
- InteractionRole
- InteractionScenario
- Process
- ProcessEdge
- ProcessUsage

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.

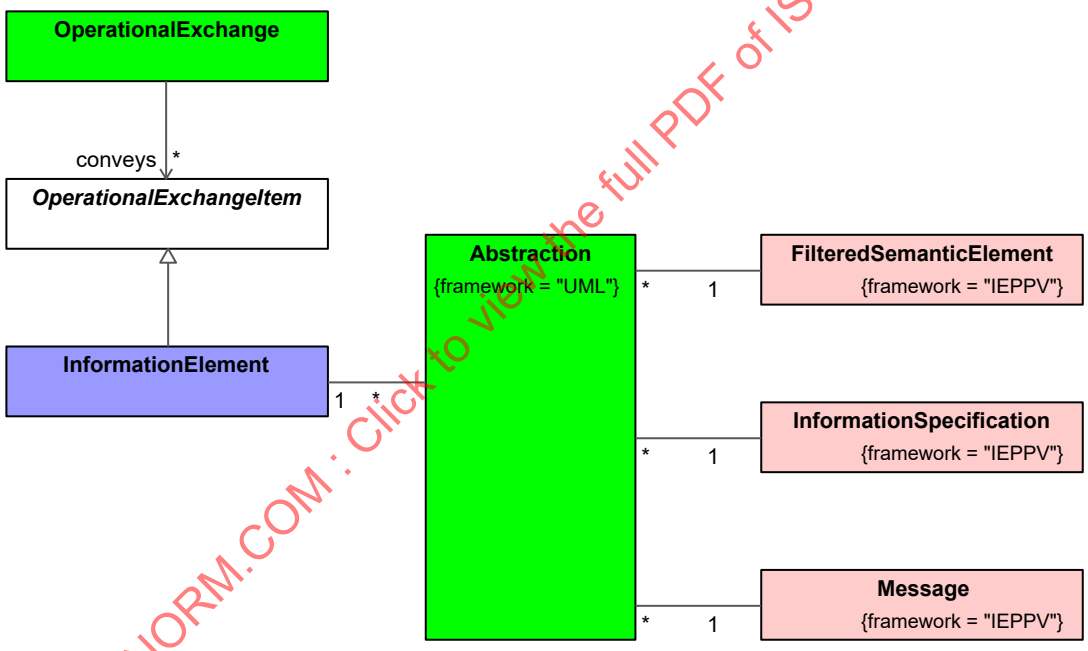

Figure 8:78 - IEPPV

Elements

- Abstraction
- FilteredSemanticElement
- InformationElement
- InformationSpecification
- Message
- OperationalExchange
- OperationalExchangeItem

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.

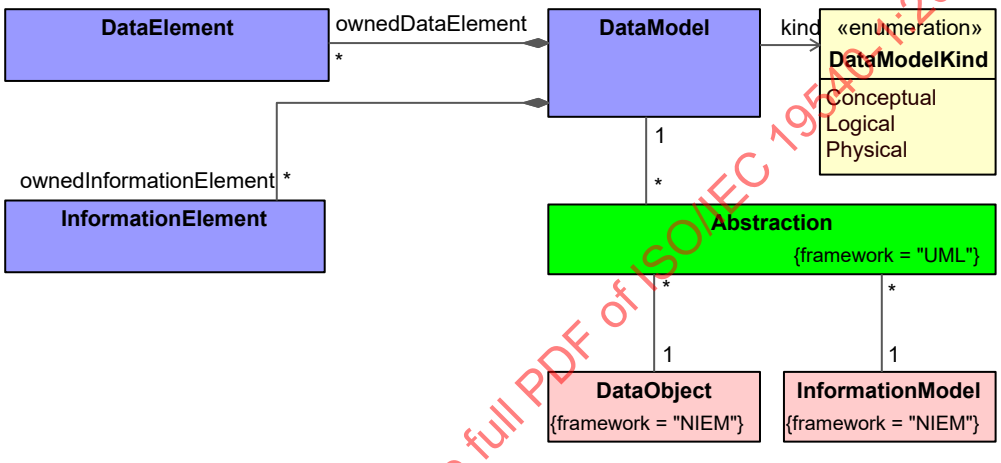

Figure 8:79 - NIEM

Elements

- Abstraction
- DataElement
- DataModel
- DataObject
- InformationElement
- InformationModel

9 Domain Metamodel (DMM) Elements

9.1 Domain MetaModel

This package contains the elements of the DMM.

9.1.1 Domain MetaModel::Metadata

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.

Domain MetaModel::Metadata::Taxonomy

ArchitectureMetadata

Package: Taxonomy
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.

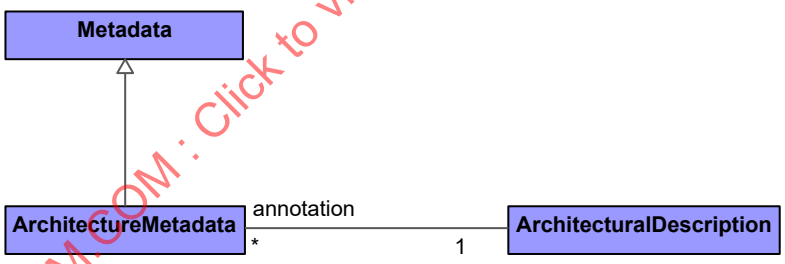

Figure 9:1 - ArchitectureMetadata

InteractionScenarioGeneralization

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

Description

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

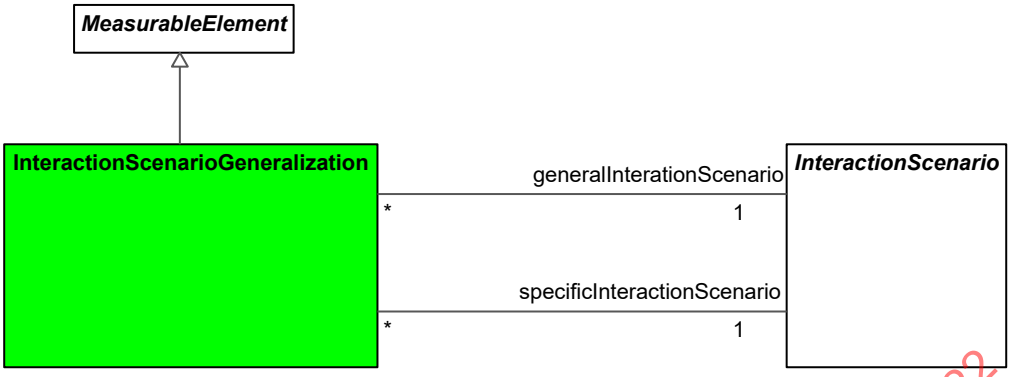

Figure 9:2 - InteractionScenarioGeneralization

Metadata

Package: Taxonomy

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.

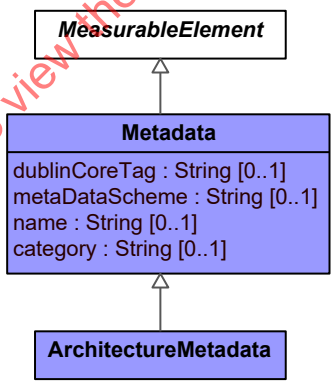

Figure 9:3 - Metadata

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.

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.

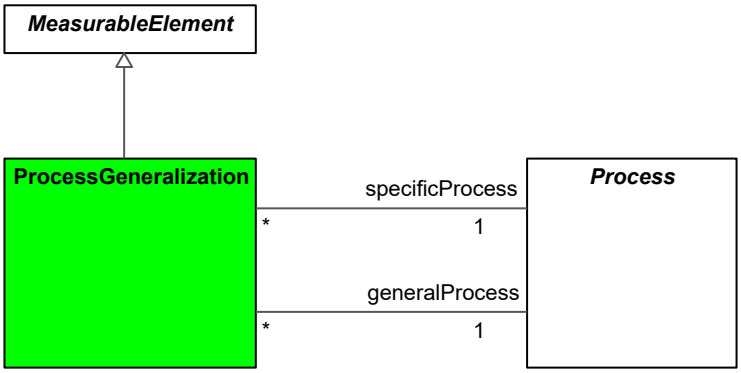

Figure 9:4 - ProcessGeneralization

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.

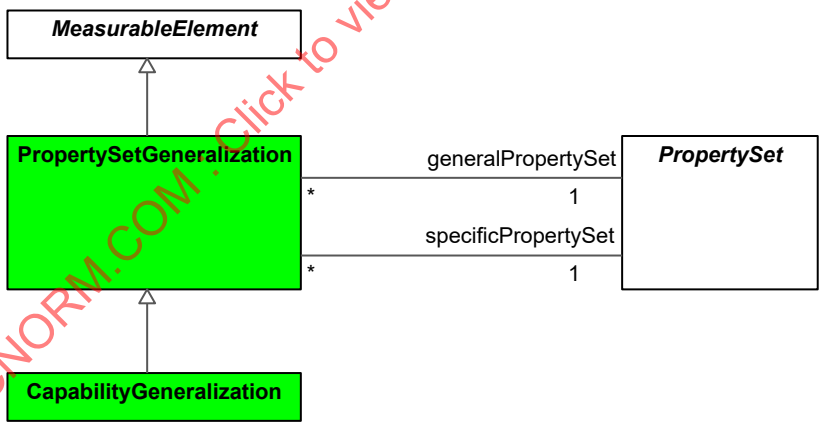

Figure 9:5 - PropertySetGeneralization

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.

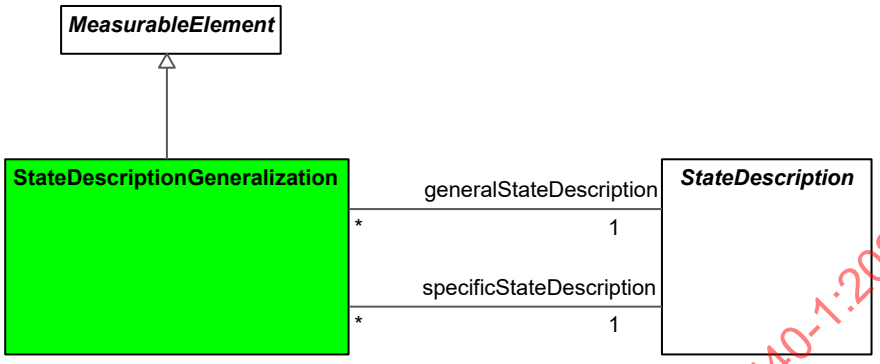

Figure 9:6 - StateDescriptionGeneralization

Domain MetaModel::Metadata::Structure

EnvironmentProperty

Package: Structure

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.

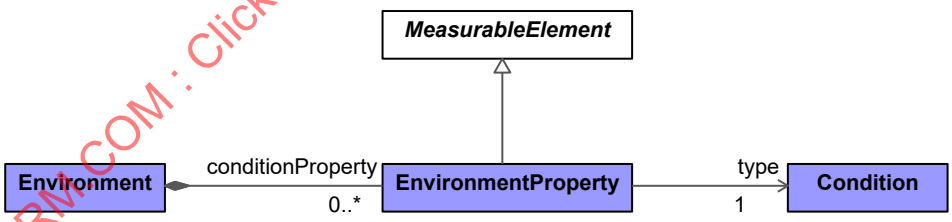

Figure 9:7 - EnvironmentProperty

Domain MetaModel::Metadata::Connectivity

Exchange

Package: Connectivity

isAbstract: Yes

Generalization: MeasurableElement, BPMN2Metamodel::MessageFlow, SubjectOfSecurityConstraint

Description

Abstract tuple, grouping OperationalExchanges and ResourceExchanges that exchange Resources.

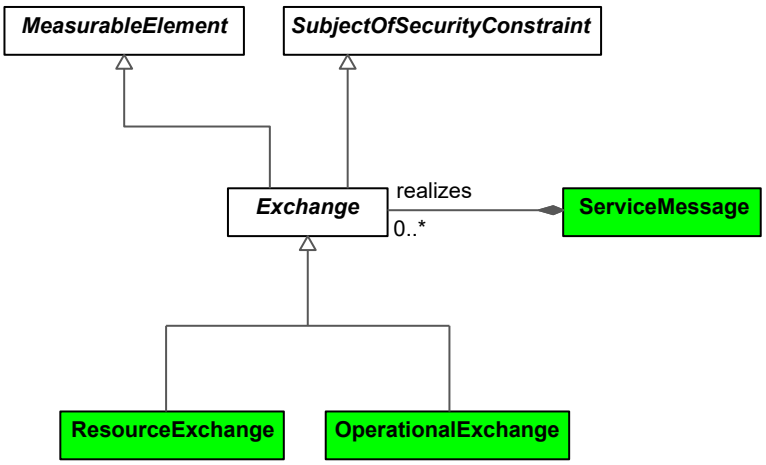

Figure 9:8 - Exchange

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.

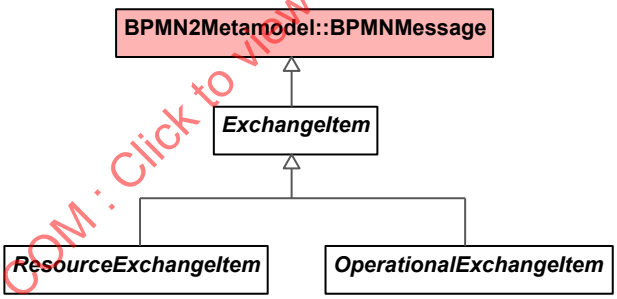

Figure 9:9 - ExchangeItem

Resource

Package: Connectivity
isAbstract: Yes
Generalization: PropertySet

Description

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

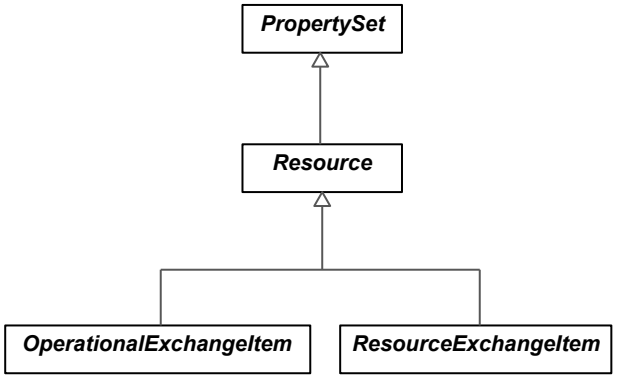

Figure 9:10 - Resource

Domain MetaModel::Metadata::Processes

ActivityPerformableUnderCondition

Package: Processes
isAbstract: No
Generalization: MeasurableElement

Description

The ActualCondition under which an Activity is performed.

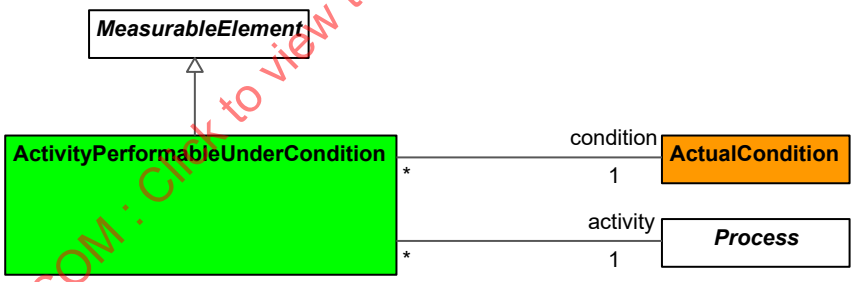

Figure 9:11 - ActivityPerformableUnderCondition

IsCapableToPerform

Package: Processes
isAbstract: No
Generalization: MeasurableElement

Description

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

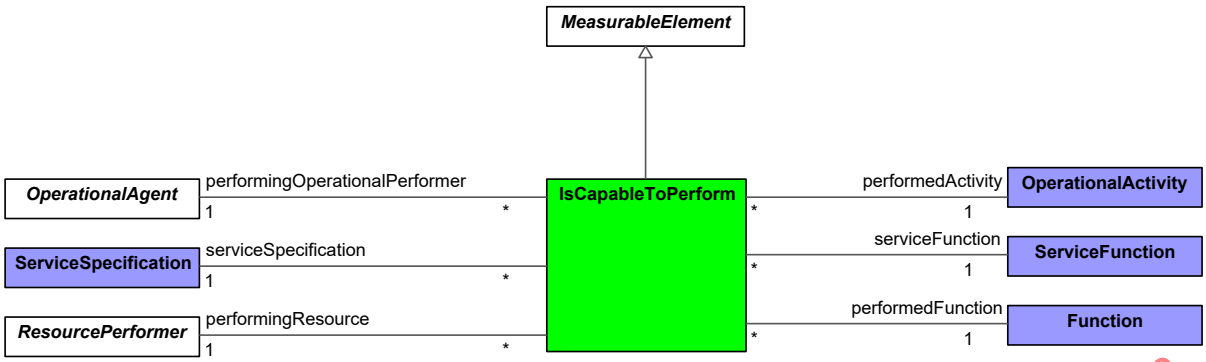

Figure 9:12 - IsCapableToPerform

PerformsInContext

Package: Processes

isAbstract: No

Generalization: MeasurableElement

Description

A tuple that relates an OperationalAction to an 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.

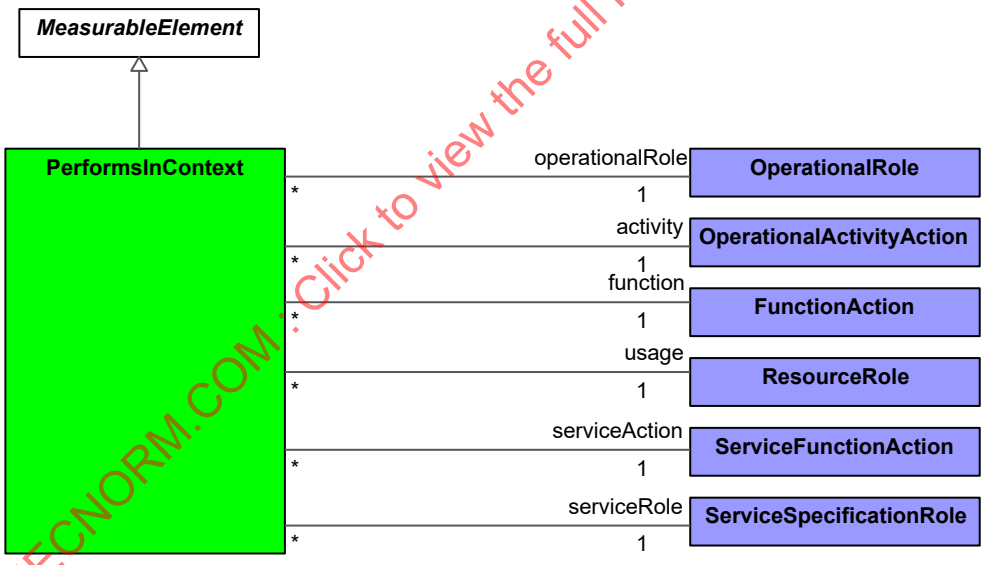

Figure 9:13 - PerformsInContext

Process

Package: Processes

isAbstract: Yes

Generalization: MeasurableElement, UML2.5Metamodel::Activity, BPMN2Metamodel::Process

Description

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

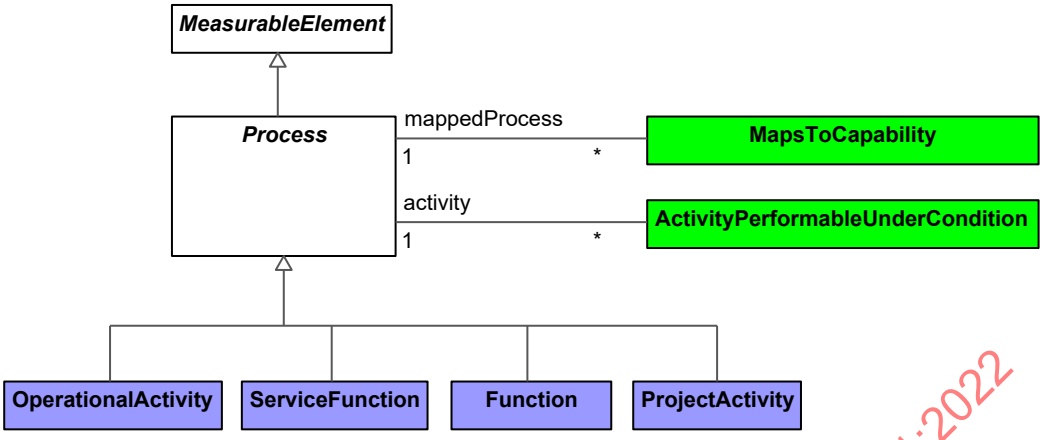

Figure 9:14 - Process

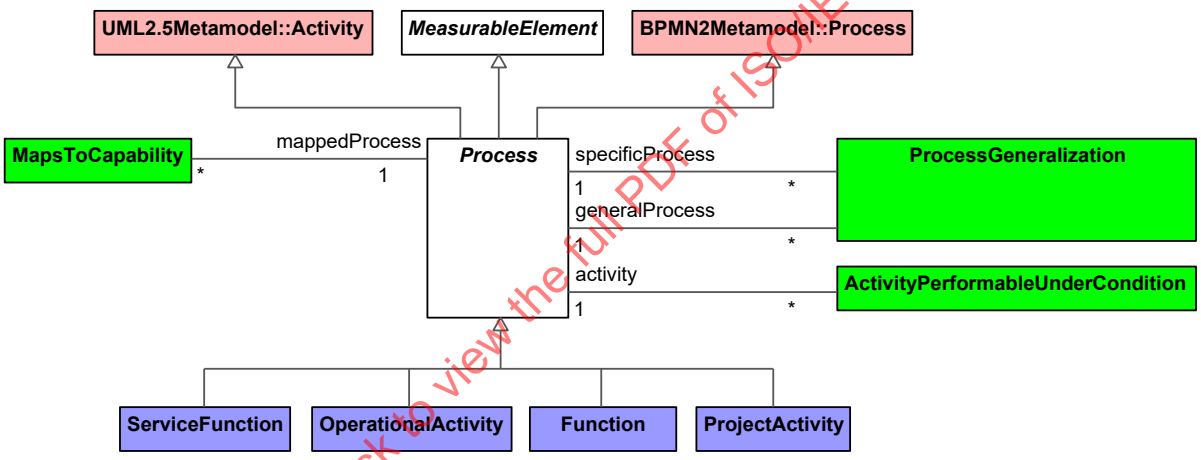

Figure 9:15 - Process

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.

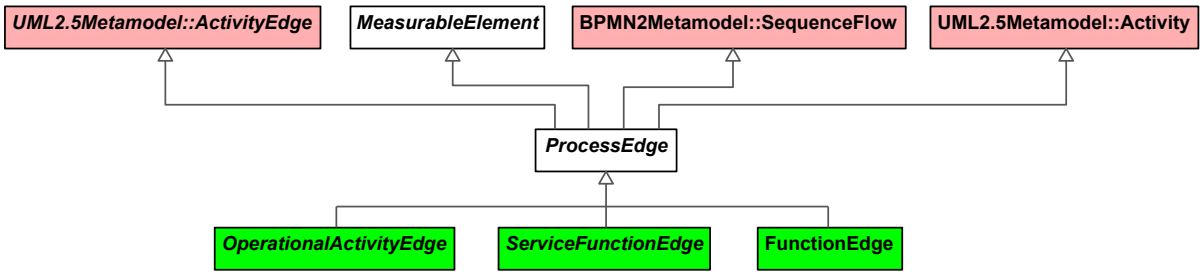

Figure 9:16 - ProcessEdge

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.

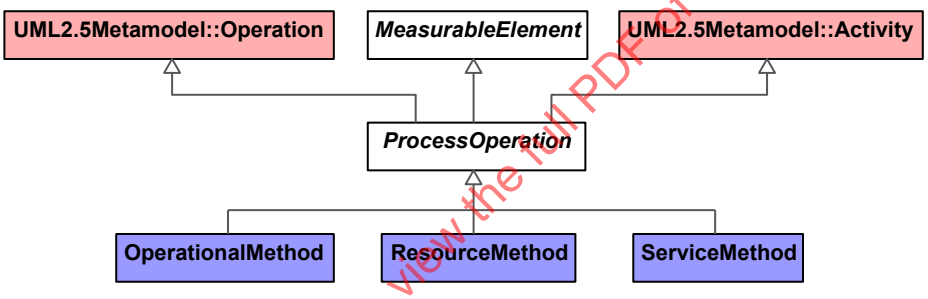

Figure 9:17 - ProcessOperation

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.

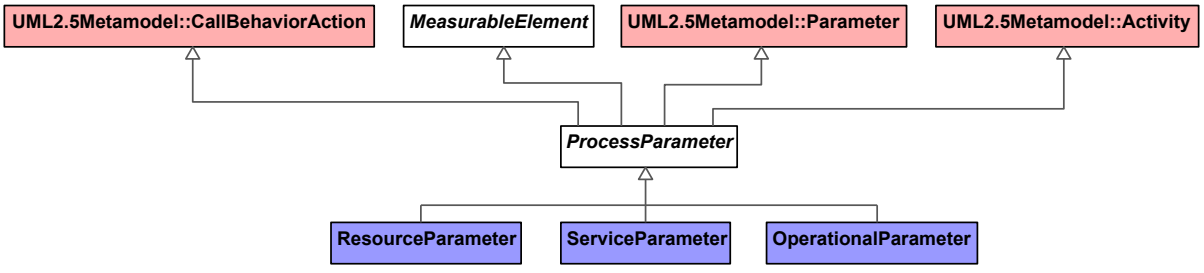

Figure 9:18 - ProcessParameter

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.

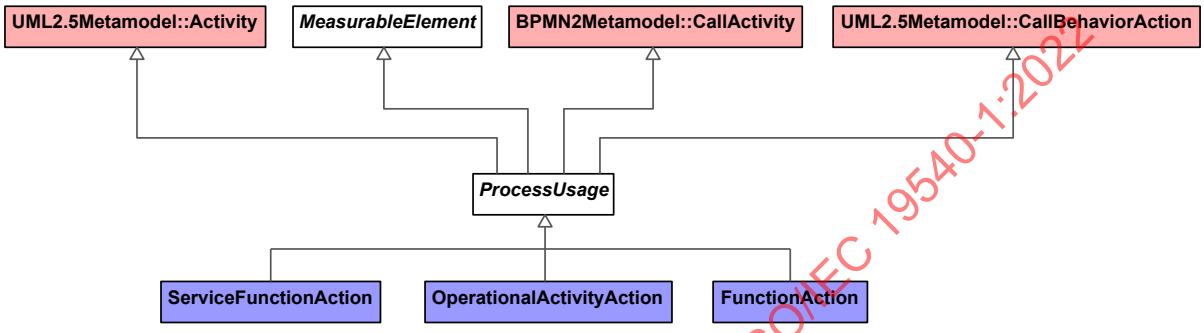

Figure 9:19 - ProcessUsage

Domain MetaModel::Metadata::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.

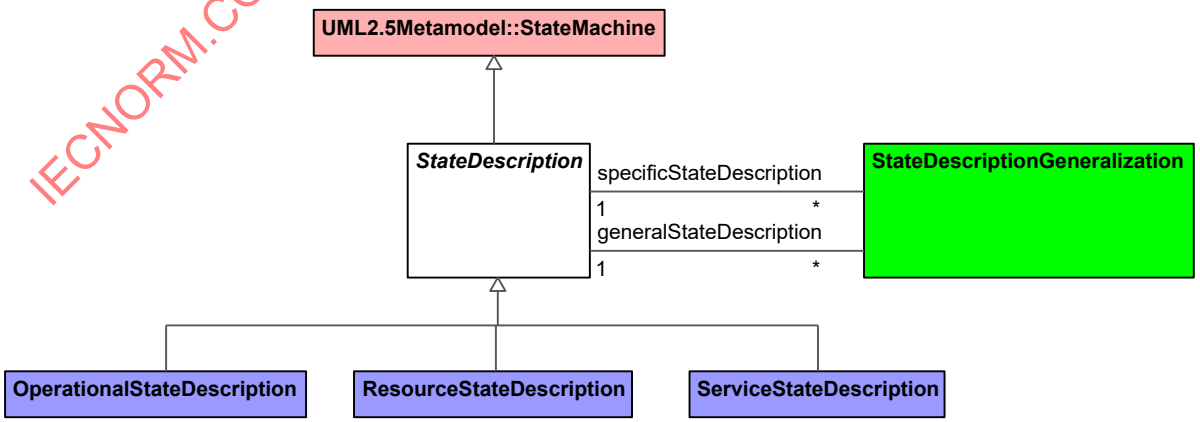

Figure 9:20 - StateDescription

Domain MetaModel::Metadata::Interaction Scenarios

InteractionMessage

Package: Interaction Scenarios

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: Interaction Scenarios

isAbstract: Yes

Generalization: BPMN2Metamodel::ResourceRole

Description

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

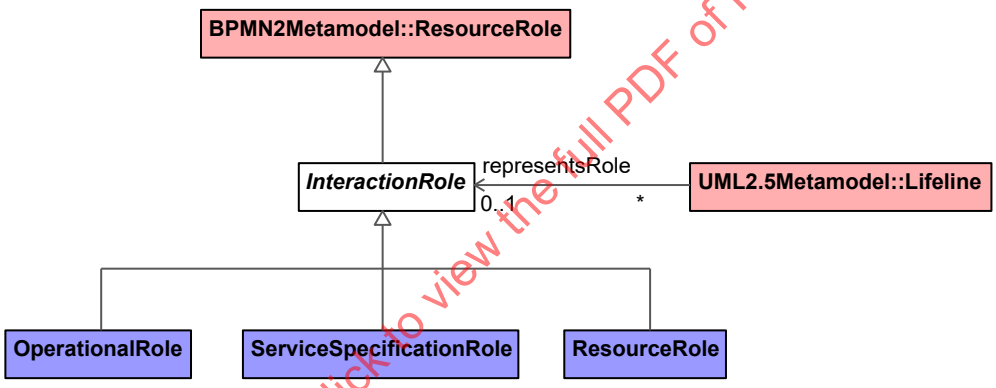

Figure 9:21 - InteractionRole

InteractionScenario

Package: Interaction Scenarios

isAbstract: Yes

Generalization: MeasurableElement, UML2.5Metamodel::Activity, BPMN2Metamodel::Process, UML2.5Metamodel::Interaction

Description

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

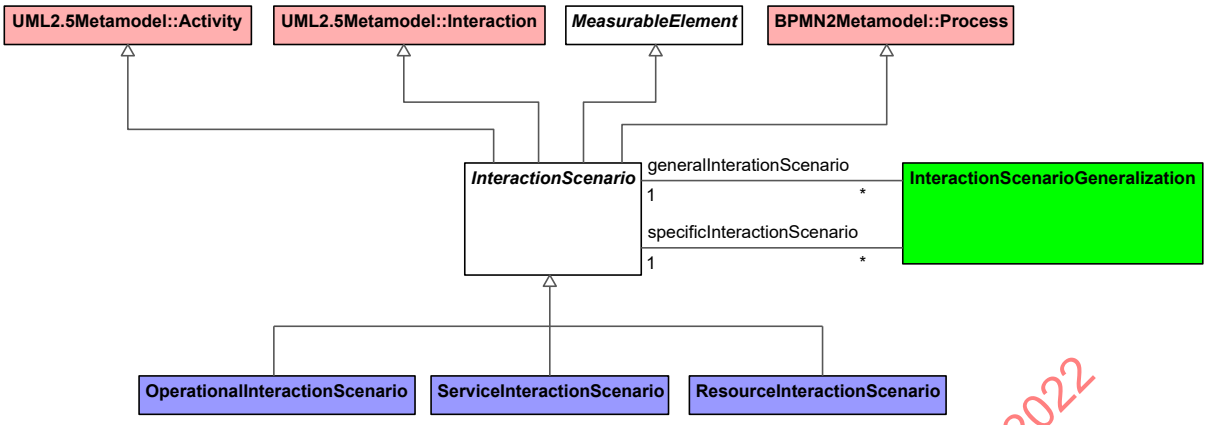

Figure 9:22 - InteractionScenario

Domain MetaModel::Metadata::Information

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.

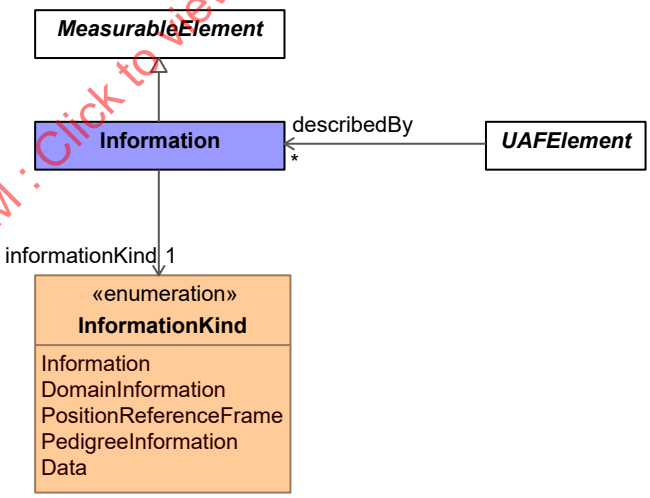

Figure 9:23 - Information

Domain MetaModel::Metadata::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).

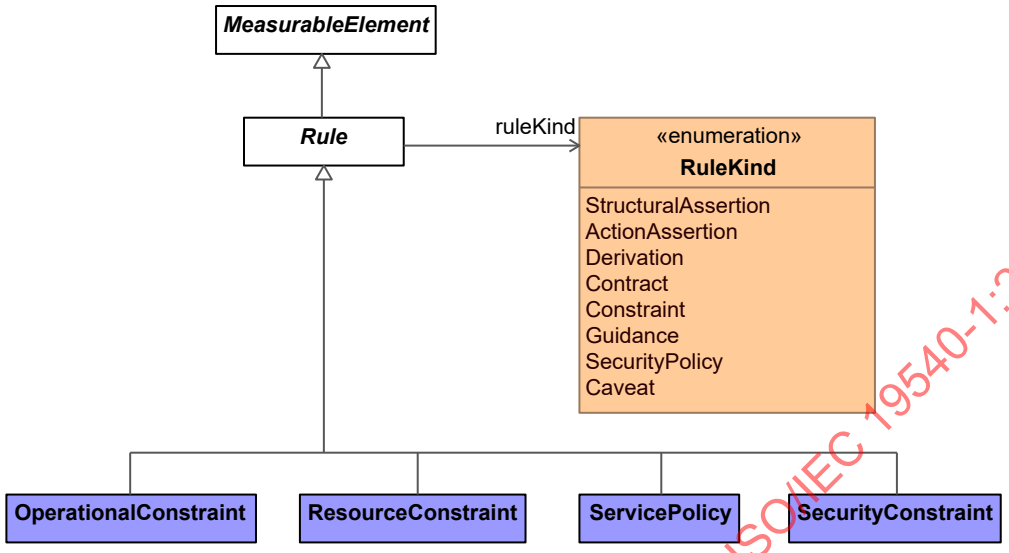

Figure 9:24 - Rule

Domain MetaModel::Metadata::Traceability

ArchitecturalReference

Package: Traceability

isAbstract: No

Generalization: MeasurableElement

Description

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

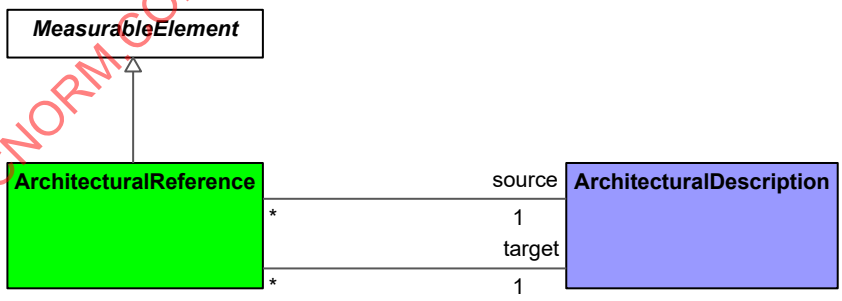

Figure 9:25 - ArchitecturalReference

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.

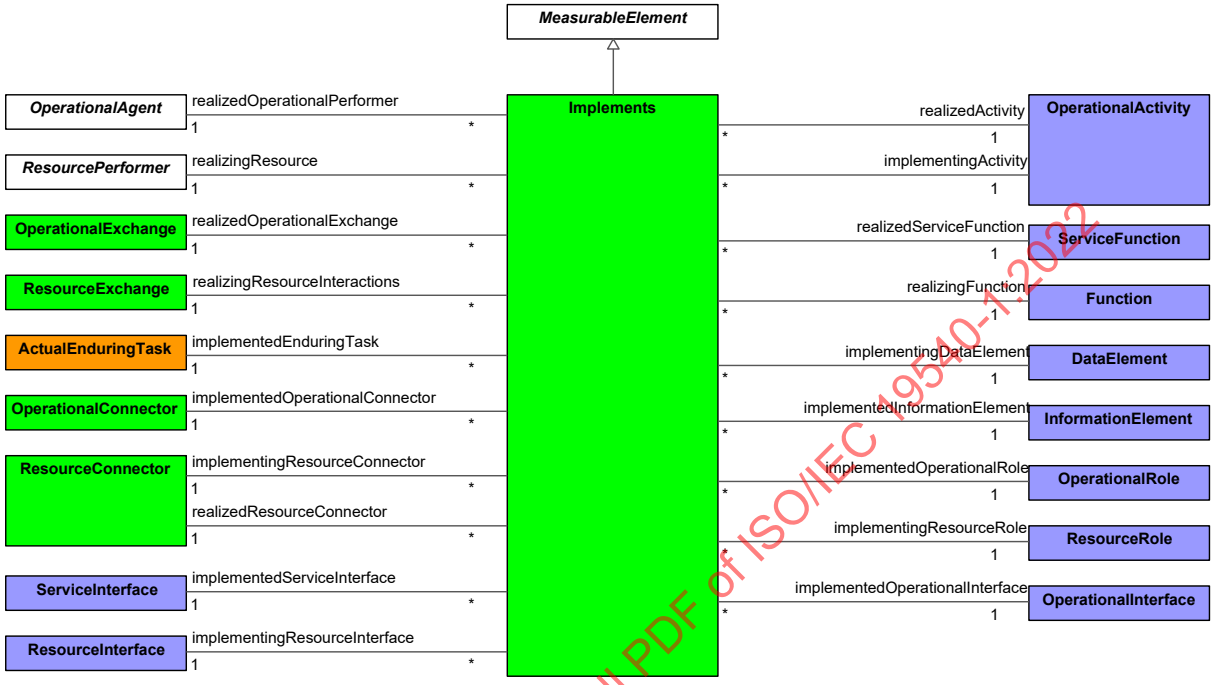

Figure 9-26 - Implements

9.1.2 Domain MetaModel::Strategic

Domain MetaModel::Strategic::Taxonomy

Capability

Package: Taxonomy

isAbstract: No

Generalization: PropertySet, Desirer

Description

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

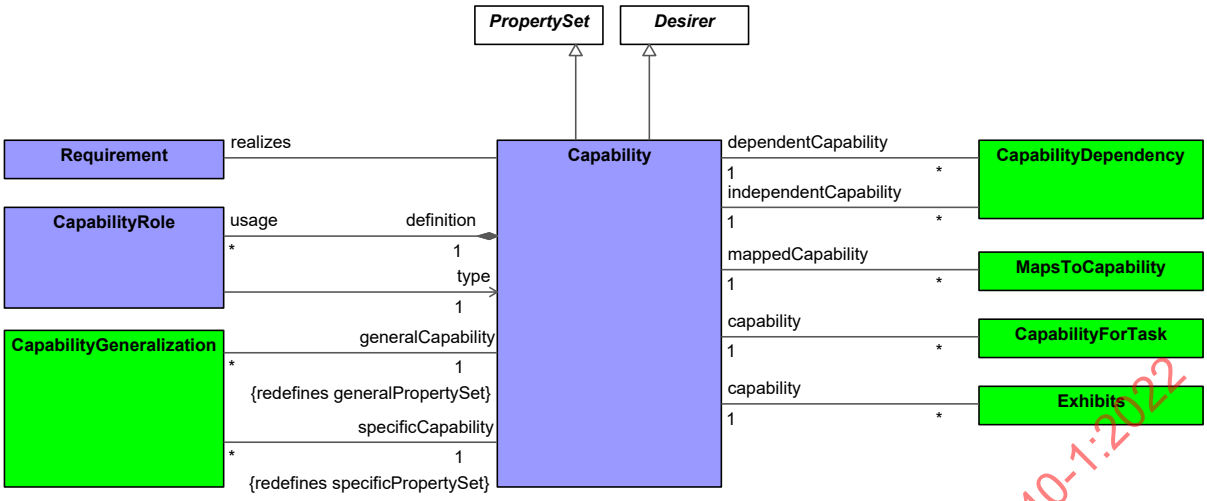

Figure 9:27 - Capability

CapabilityGeneralization

Package: Taxonomy

isAbstract: No

Generalization: PropertySetGeneralization

Description

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

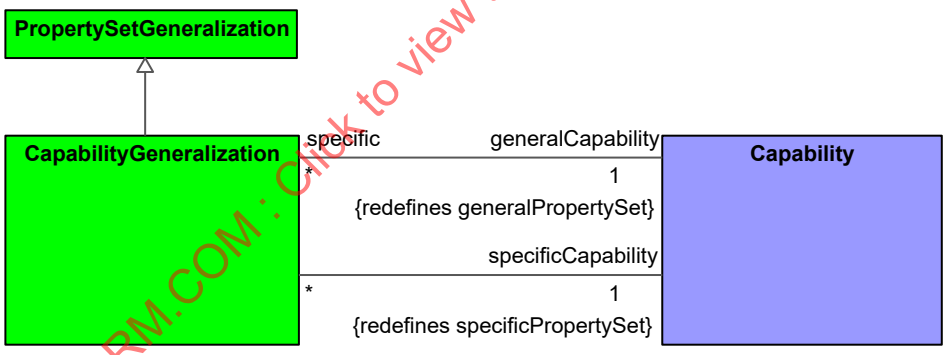

Figure 9:28 - CapabilityGeneralization

Domain MetaModel::Strategic::Structure

ActualEnduringTask

Package: Structure

isAbstract: No

Generalization: CapableElement, ActualPropertySet

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.

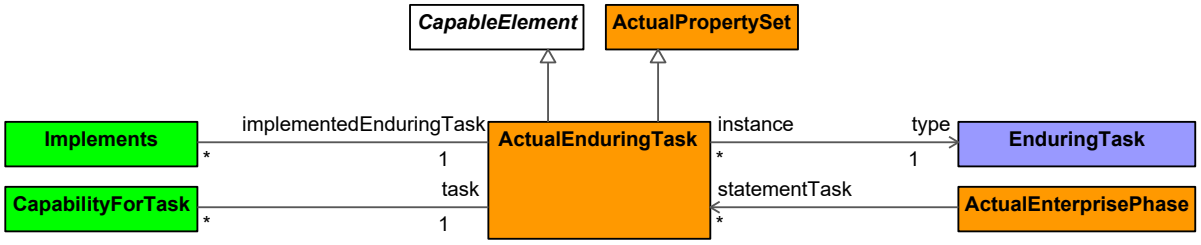

Figure 9:29 - ActualEnduringTask

ActualEnterprisePhase

Package: Structure

isAbstract: No

Generalization: CapableElement, ActualPropertySet, Achiever

Description

An individual that describes the phase of an actual enterprise endeavor.

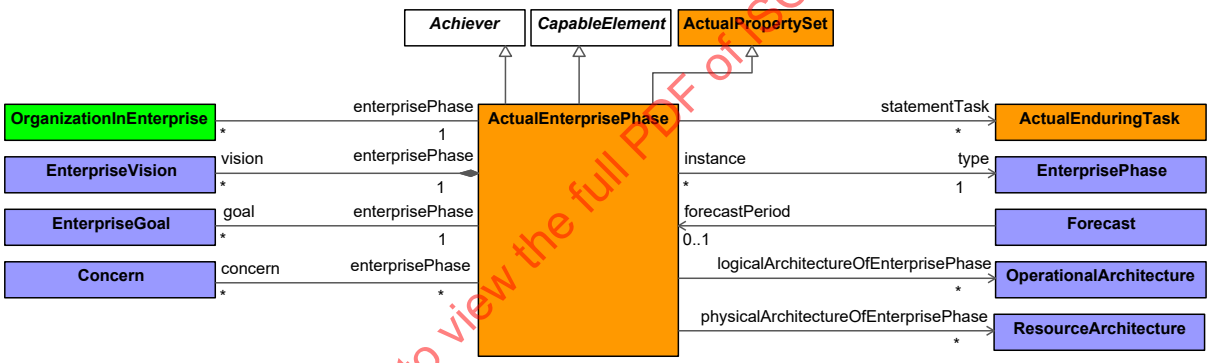

Figure 9:30 - ActualEnterprisePhase

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.

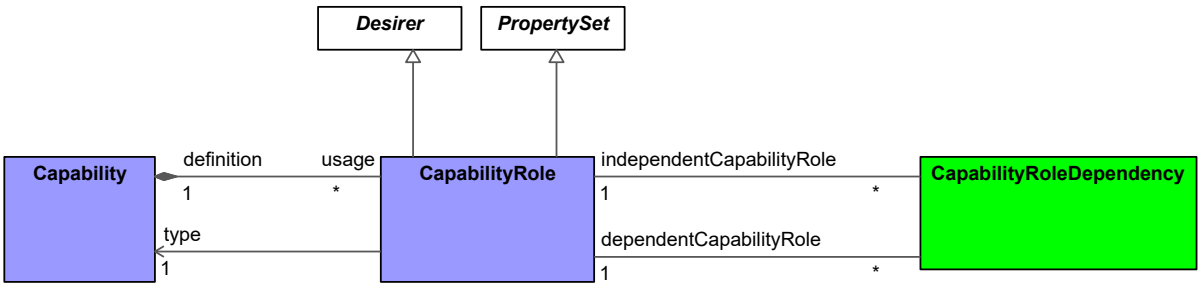

Figure 9:31 - CapabilityRole

EnduringTask

Package: Structure

isAbstract: No

Generalization: PropertySet

Description

A type of template behavior recognized by an enterprise as being essential to achieving its goals - i.e., a template for a strategic specification of what the enterprise does.

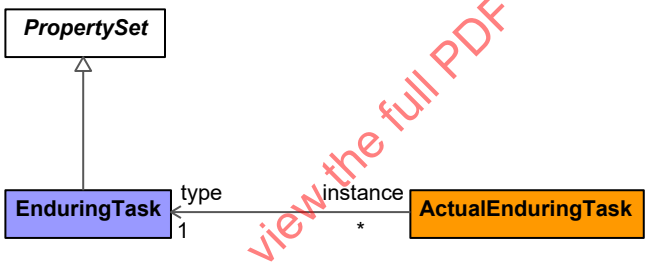

Figure 9:32 - EnduringTask

EnterpriseGoal

Package: Structure

isAbstract: No

Generalization: PropertySet

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.

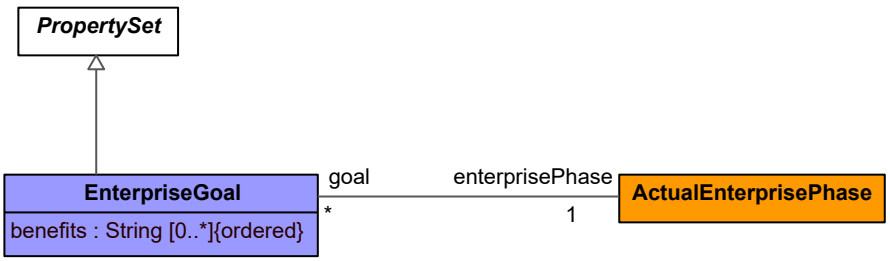

Figure 9:33 - EnterpriseGoal

Attributes

benefits :	String[0..*]	A description of the usefulness of the Goal in terms of why the state or condition of the Enterprise is worth attaining.

EnterprisePhase

Package: Structure

isAbstract: No

Generalization: PropertySet

Description

A type of a current or future state of the enterprise.

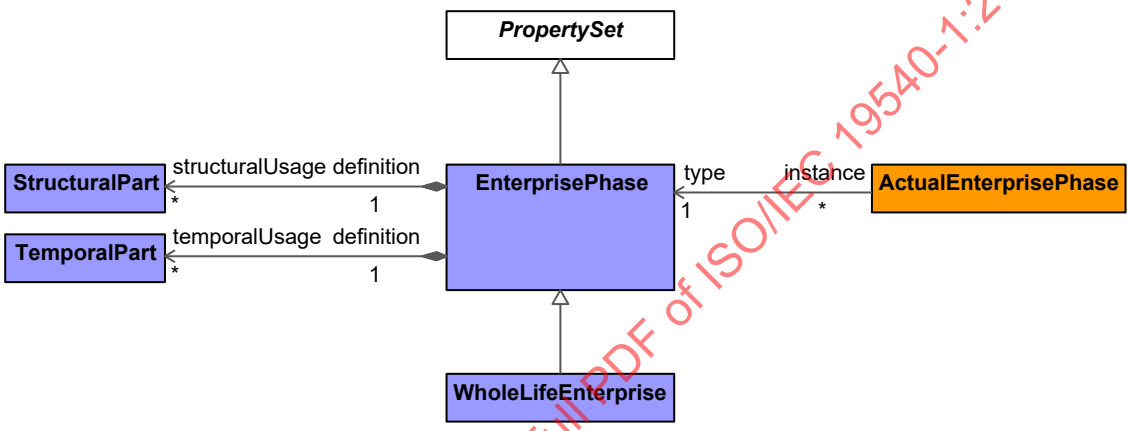

Figure 9:34 - EnterprisePhase

EnterpriseVision

Package: Structure

isAbstract: No

Generalization: PropertySet

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.

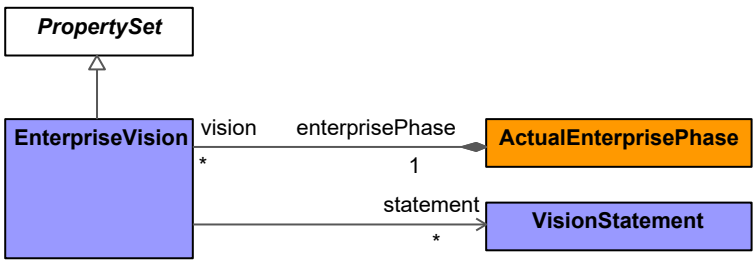

Figure 9:35 - EnterpriseVision

StructuralPart

Package: Structure

isAbstract: No

Generalization: MeasurableElement

Description

A current or future state of the wholeLifeEnterprise or another EnterprisePhase.

Figure 9:36 - StructuralPart

TemporalPart

Package: Structure
isAbstract: No
Generalization: MeasurableElement

Description

A current or future state of the wholeLifeEnterprise or another EnterprisePhase.

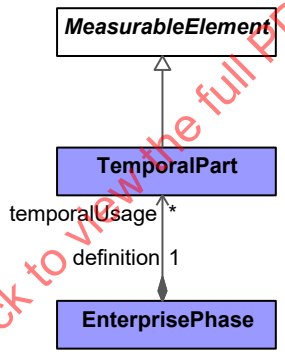

Figure 9:37 - TemporalPart

VisionStatement

Package: Structure
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.

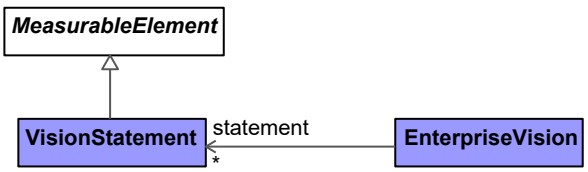

Figure 9:38 - VisionStatement

WholeLifeEnterprise

Package: Structure

isAbstract: No

Generalization: EnterprisePhase

Description

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

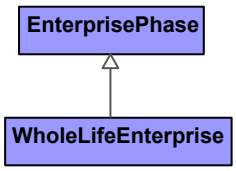

Figure 9:39 - WholeLifeEnterprise

Domain MetaModel::Strategic::Connectivity

CapabilityDependency

Package: Connectivity

isAbstract: No

Generalization: MeasurableElement

Description

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

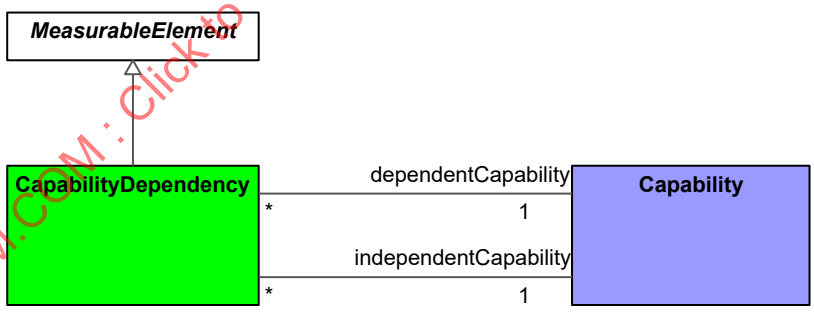

Figure 9:40 - CapabilityDependency

CapabilityRoleDependency

Package: Connectivity

isAbstract: No

Generalization: MeasurableElement

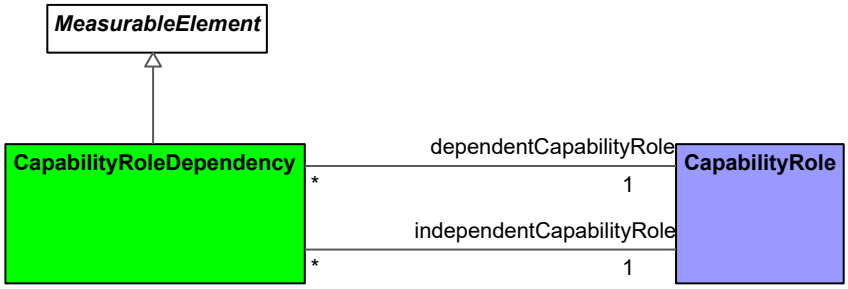

Figure 9:41 - CapabilityRoleDependency

Domain MetaModel::Strategic::States

AchievedEffect

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 DesiredEffect and an Achiever.

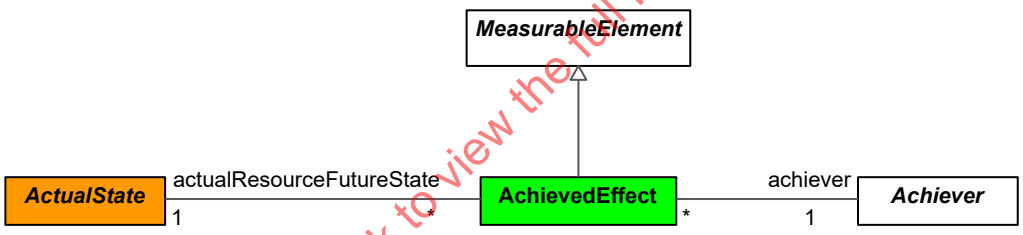

Figure 9:42 - AchievedEffect

Achiever

Package: States

isAbstract: Yes

Generalization: UAFElement

Description

An ActualResource, ActualProject or ActualEnterprisePhase that can deliver a DesiredEffect.

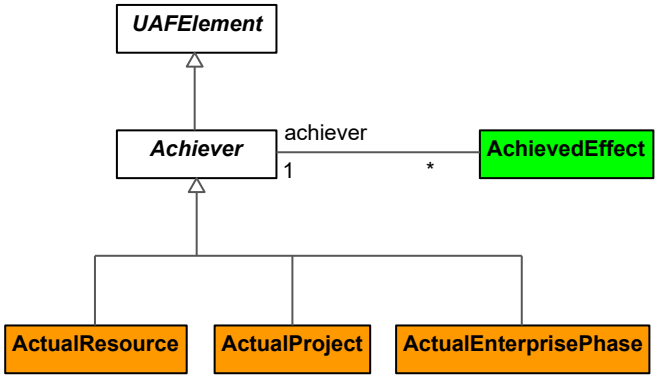

Figure 9:43 - Achiever

DesiredEffect

Package: States

isAbstract: No

Generalization: MeasurableElement

Description

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

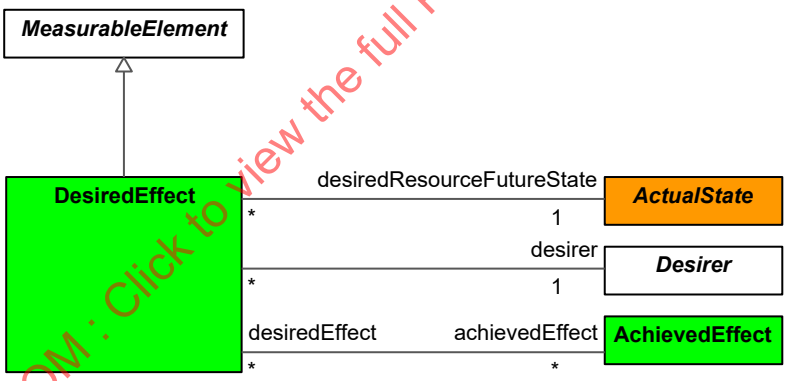

Figure 9:44 - DesiredEffect

Desirer

Package: States

isAbstract: Yes

Generalization: UAFElement

Description

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

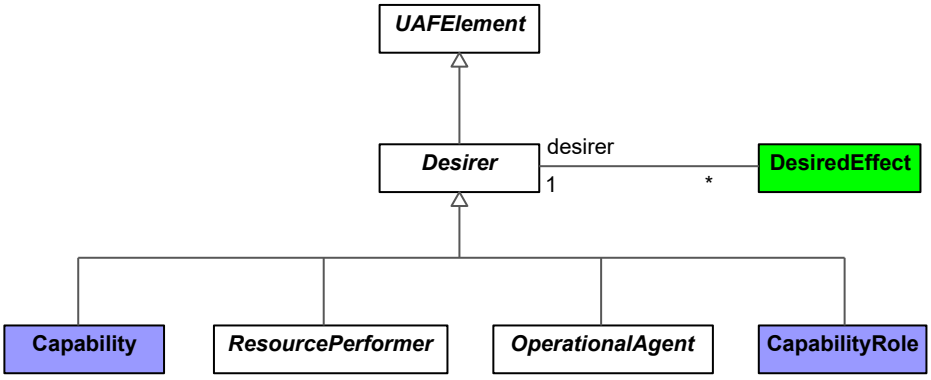

Figure 9:45 - Desirer

Domain MetaModel::Strategic::Traceability

CapabilityForTask

Package: Traceability

isAbstract: No

Generalization: MeasurableElement

Description

A tuple that asserts that a Capability is required in order for an Enterprise to conduct a phase of an EnduringTask.

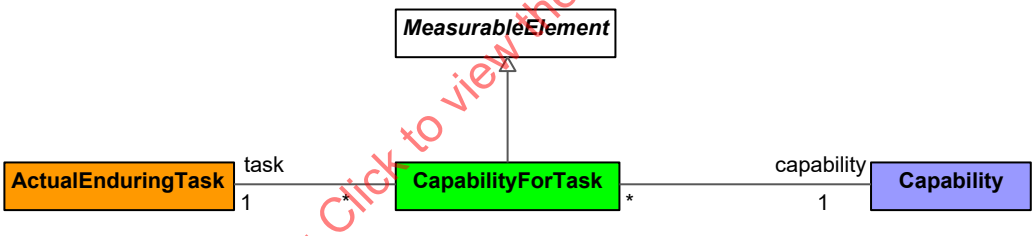

Figure 9:46 - CapabilityForTask

CapableElement

Package: Traceability

isAbstract: Yes

Generalization: UAFElement

Description

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

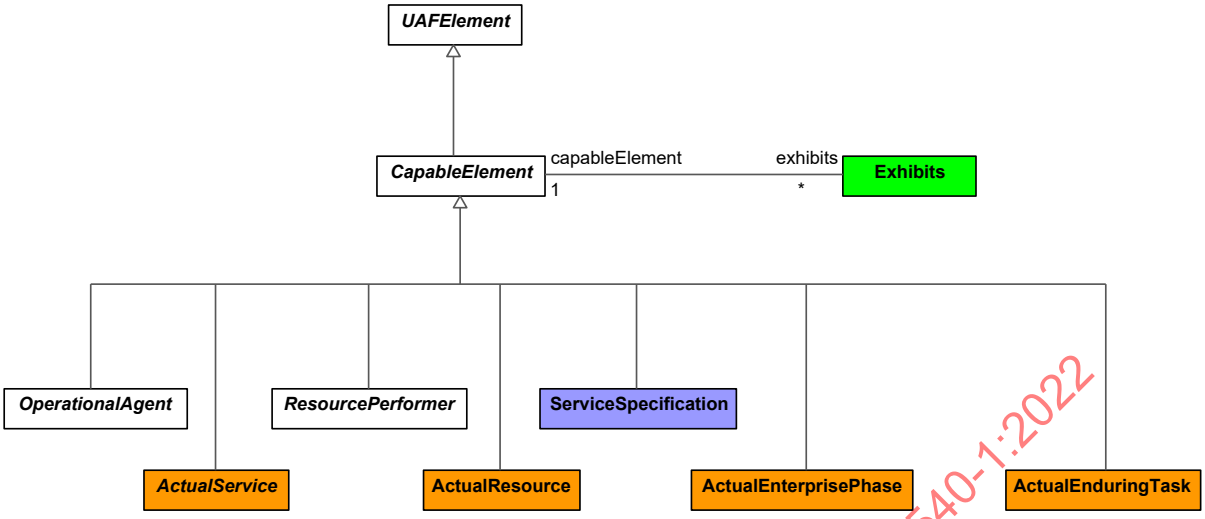

Figure 9:47 - CapableElement

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.

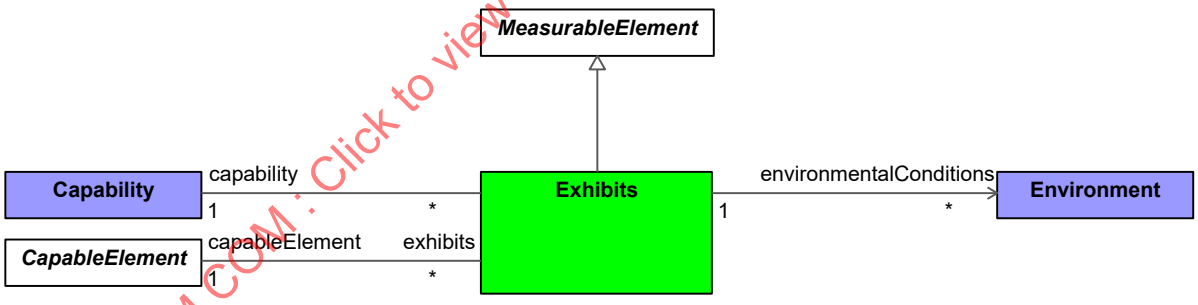

Figure 9:48 - Exhibits

MapsToCapability

Package: Traceability

isAbstract: No

Generalization: MeasurableElement

Description

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

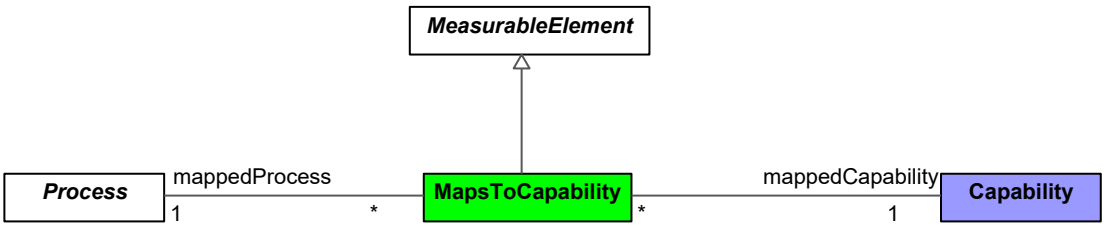

Figure 9:49 - MapsToCapability

OrganizationInEnterprise

Package: Traceability

isAbstract: No

Generalization: MeasurableElement

Description

A tuple relating an ActualOrganization to an ActualEnterprisePhase to denote that the ActualOrganization plays a role or is a stakeholder in an ActualEnterprisePhase.

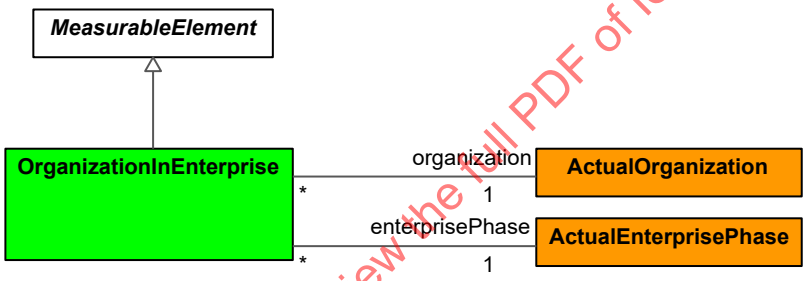

Figure 9:50 - OrganizationInEnterprise

9.1.3 Domain MetaModel::Operational

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.

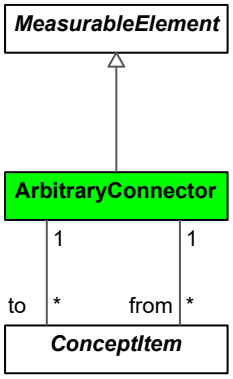

Figure 9:51 - ArbitraryConnector

ConceptItem

Package: Taxonomy

isAbstract: Yes

Generalization: UAFElement

Description

Abstract, an item which may feature in a HighLevelOperationalConcept.

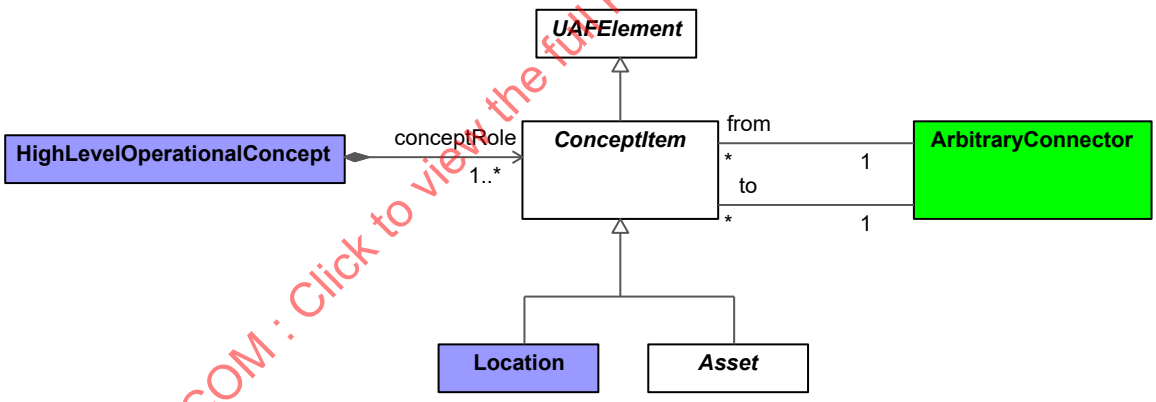

Figure 9:52 – ConceptItem

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.

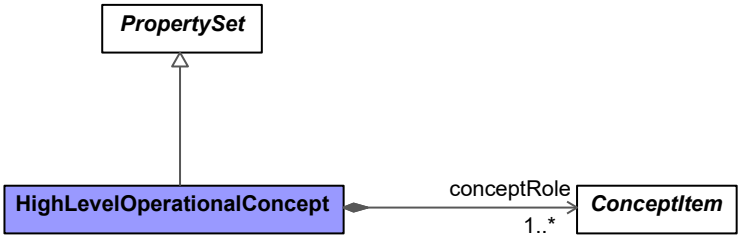

Figure 9:53 - HighLevelOperationalConcept

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.

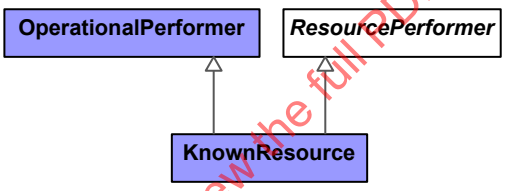

Figure 9:54 - KnownResource

OperationalAgent

Package: Structure

isAbstract: Yes

Generalization: SubjectOfOperationalConstraint, CapableElement, OperationalAsset, Desirer

Description

An abstract type grouping OperationalArchitecture and OperationalPerformer.

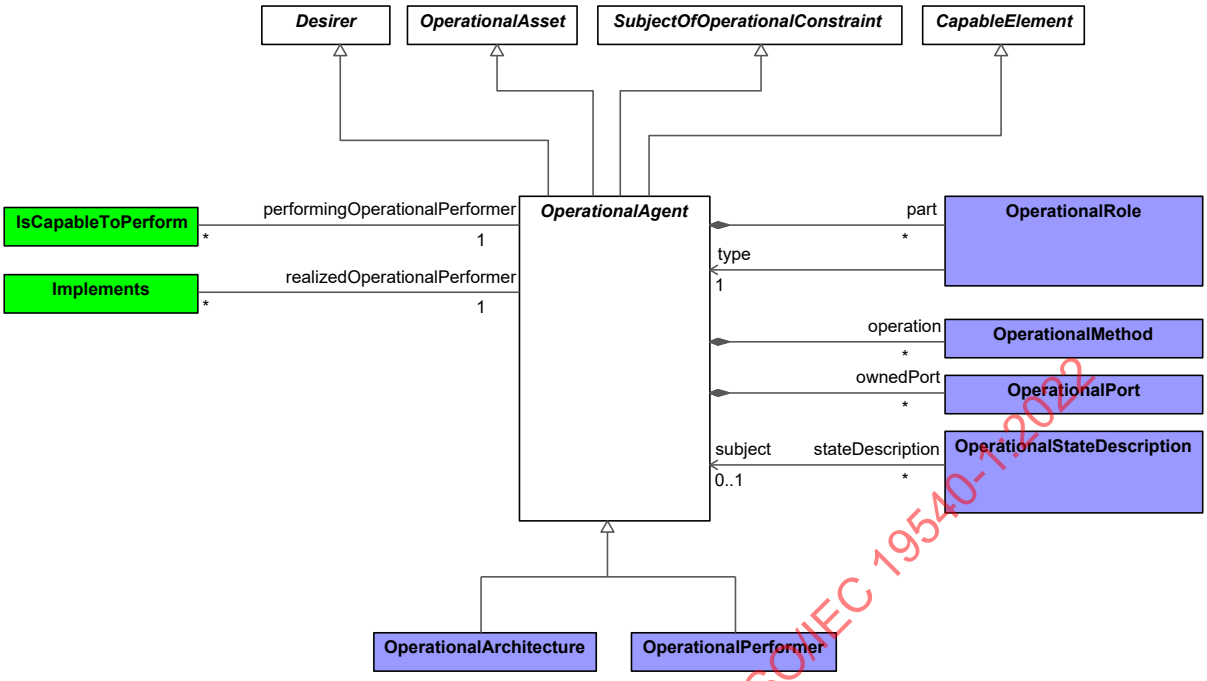

Figure 9:55 - OperationalAgent

OperationalArchitecture

Package: Structure

isAbstract: No

Generalization: OperationalAgent, Architecture

Description

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

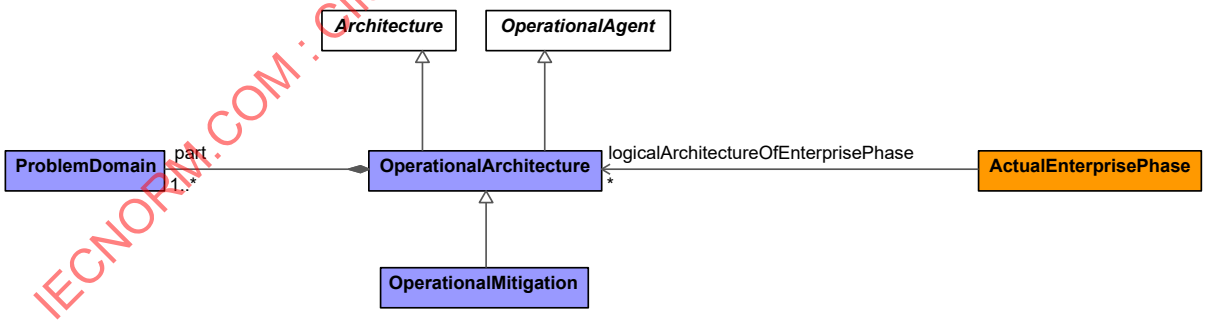

Figure 9:56 - OperationalArchitecture

OperationalMethod

Package: Structure

isAbstract: No

Generalization: ProcessOperation

Description

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

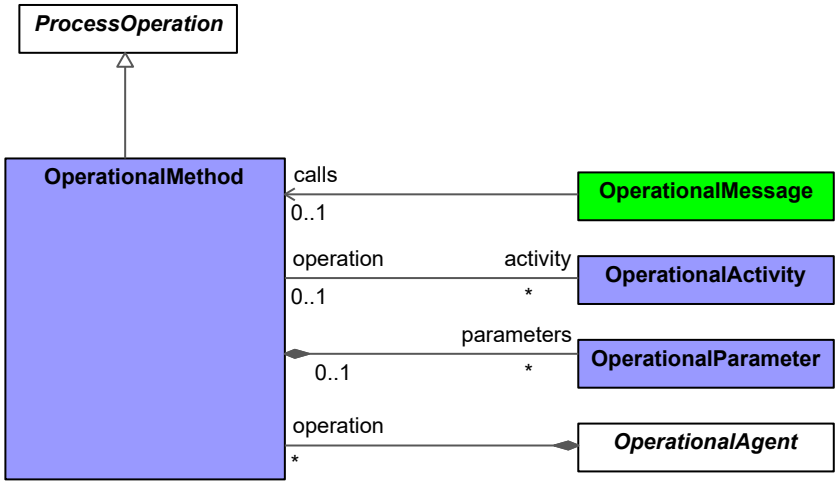

Figure 9:57 - OperationalMethod

OperationalParameter

Package: Structure

isAbstract: No

Generalization: ProcessParameter

Description

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

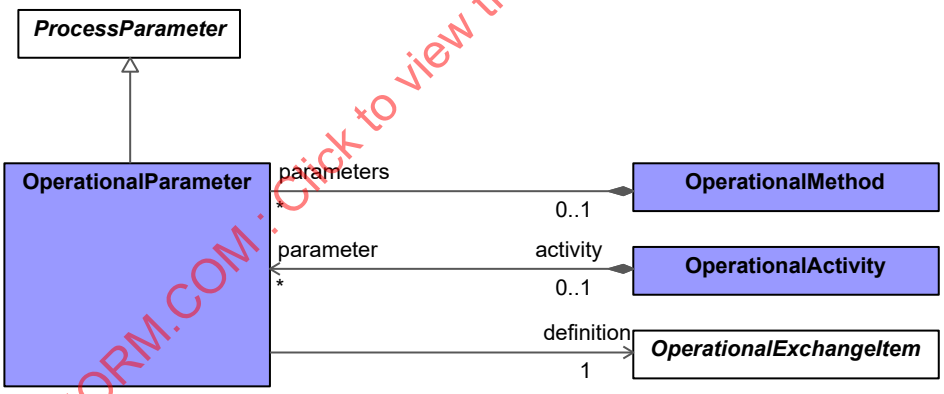

Figure 9:58 - OperationalParameter

OperationalPerformer

Package: Structure

isAbstract: No

Generalization: OperationalAgent

Description

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

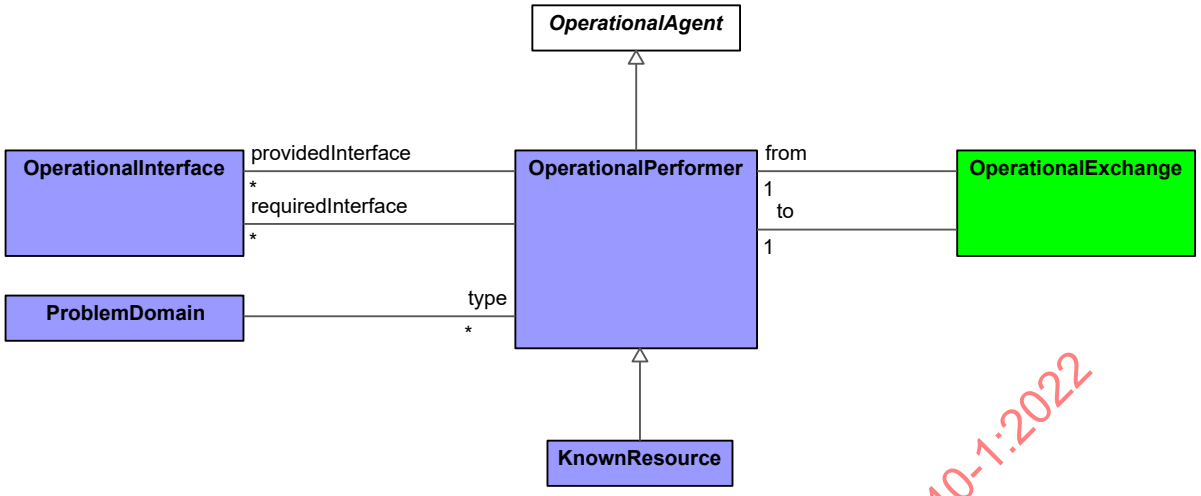

Figure 9:59 - OperationalPerformer

OperationalRole

Package: Structure

isAbstract: No

Generalization: LocationHolder, AssetRole, InteractionRole

Description

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

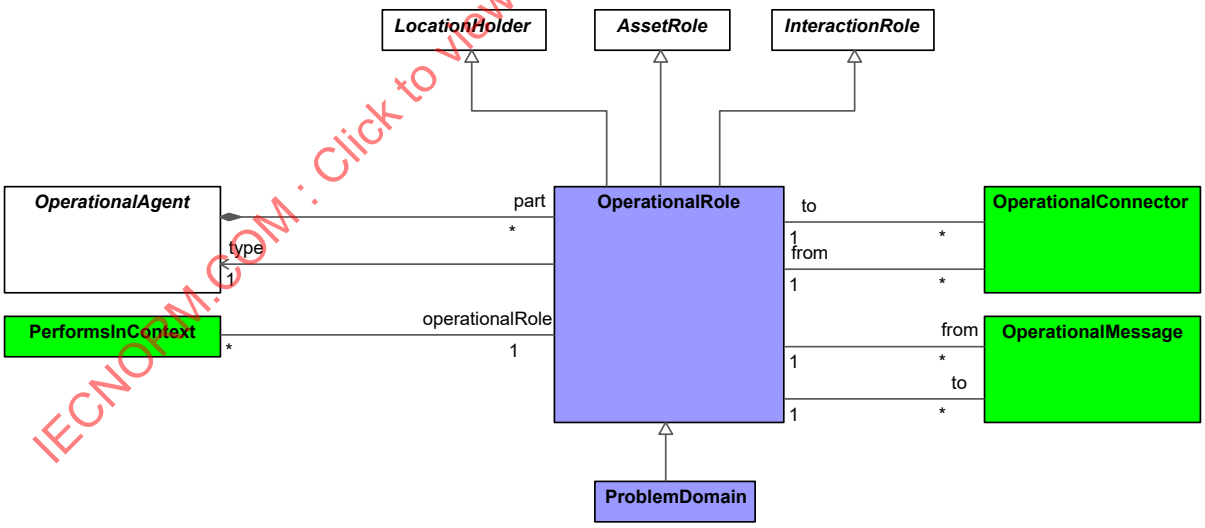

Figure 9:60 - OperationalRole

ProblemDomain

Package: Structure

isAbstract: No

Generalization: OperationalRole

Description

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

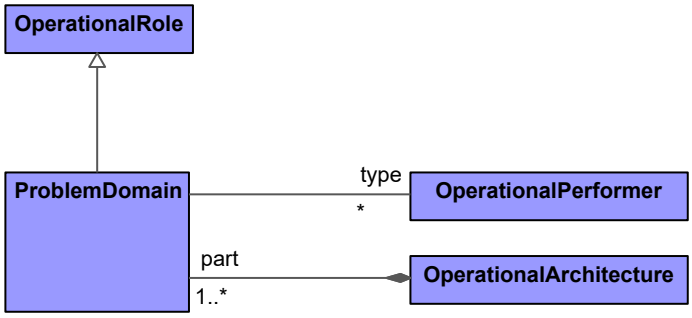

Figure 9:61 - ProblemDomain

Domain MetaModel::Operational::Connectivity

OperationalConnector

Package: Connectivity

isAbstract: No

Generalization: MeasurableElement

Description

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

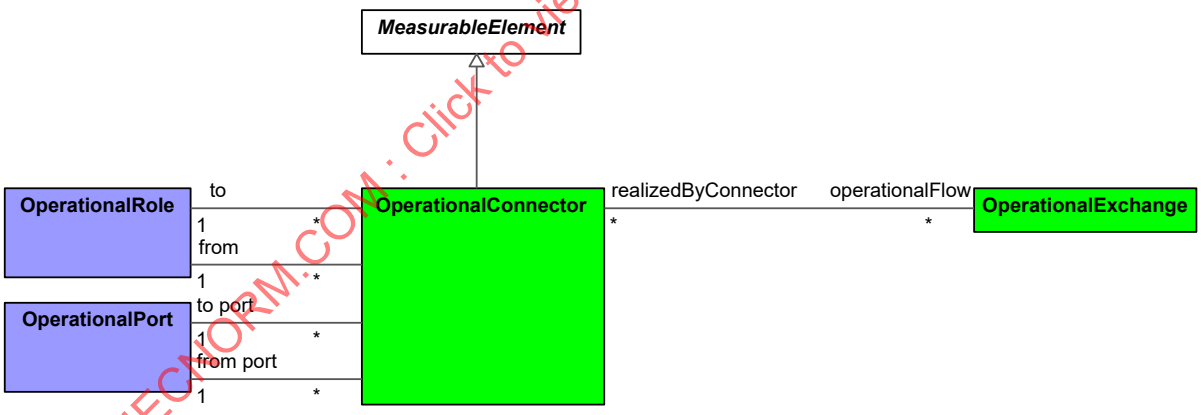

Figure 9:62 - OperationalConnector

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).

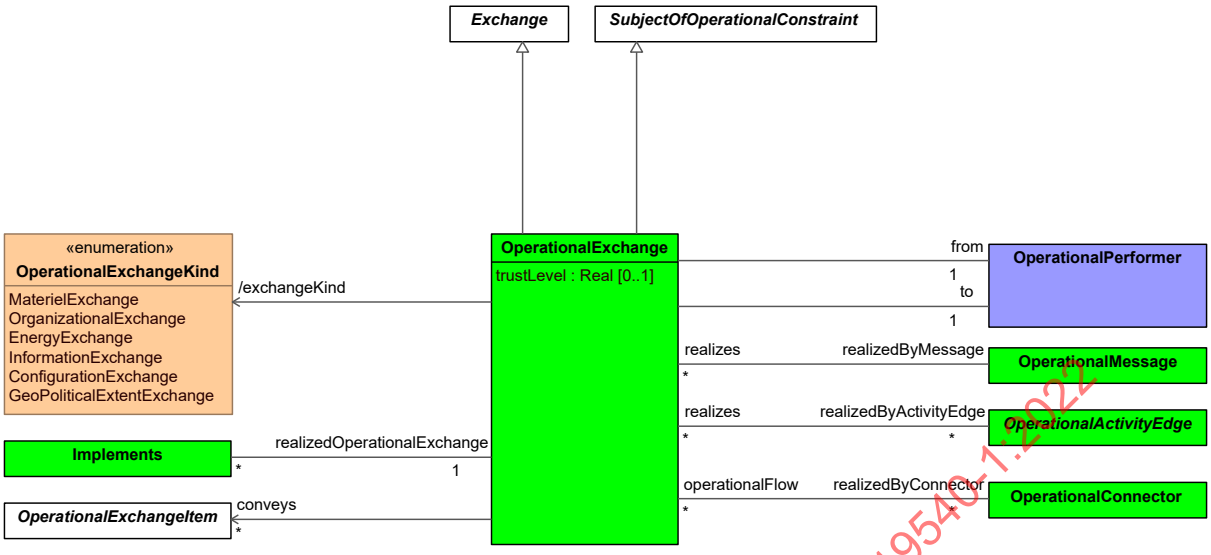

Figure 9:63 - OperationalExchange

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.

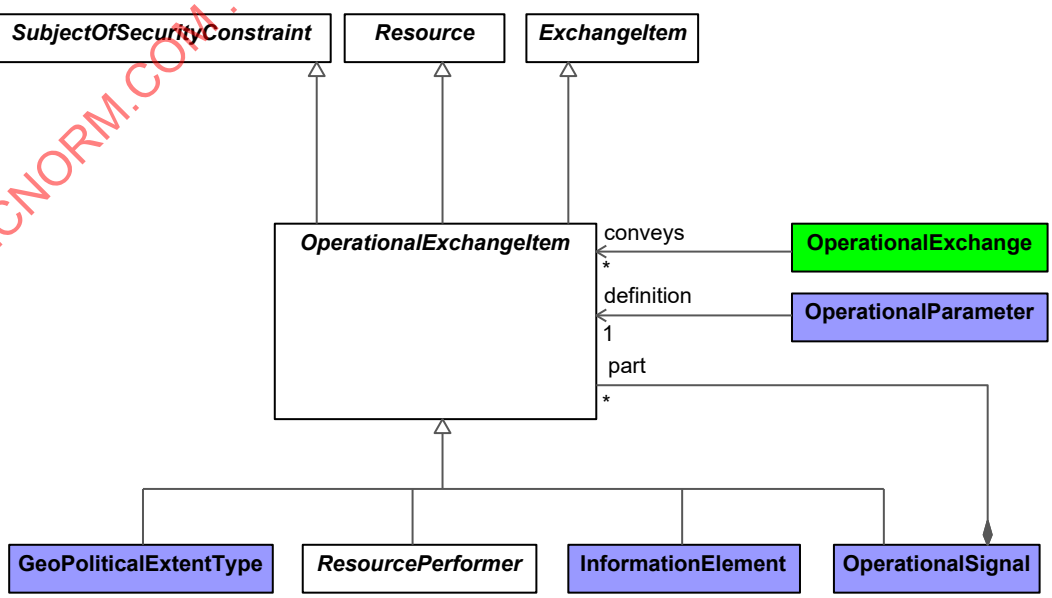

Figure 9:64 - OperationalExchangeItem

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.

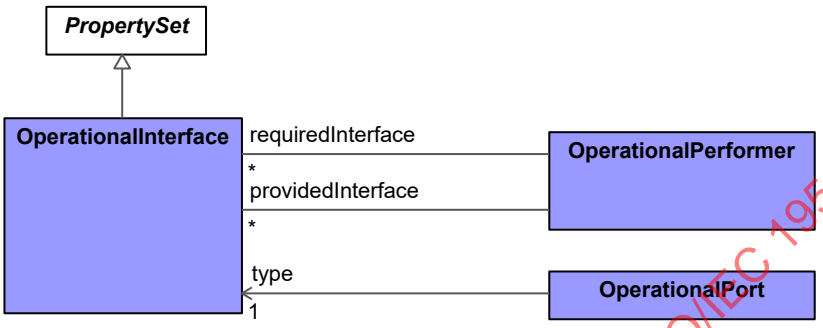

Figure 9:65 - OperationalInterface

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.

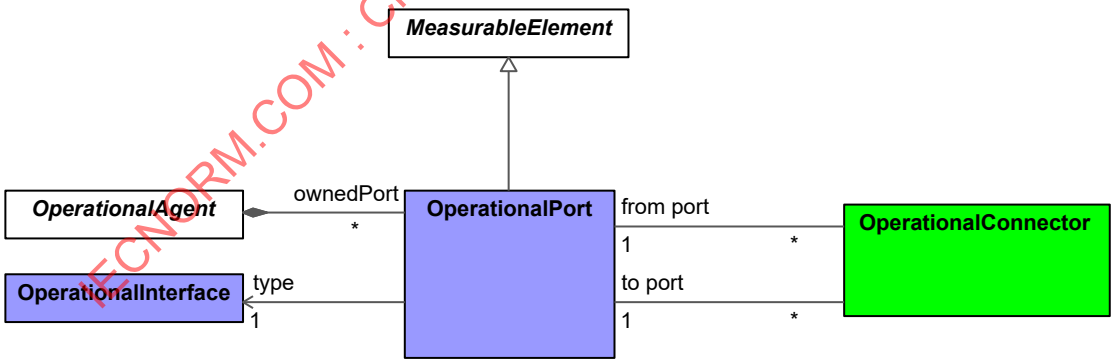

Figure 9:66 - OperationalPort

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).

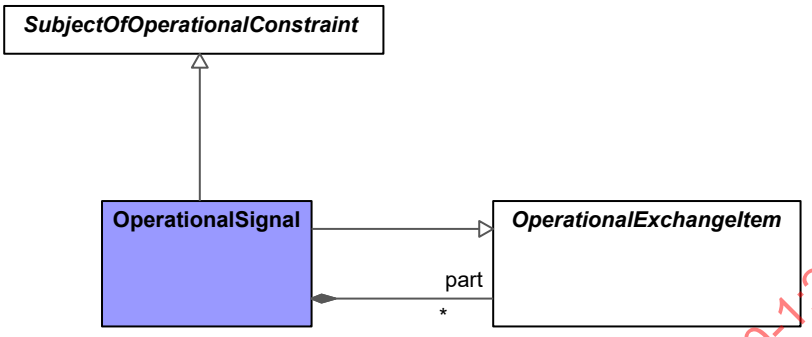

Figure 9:67 - OperationalSignal

Domain MetaModel::Operational::Processes

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.

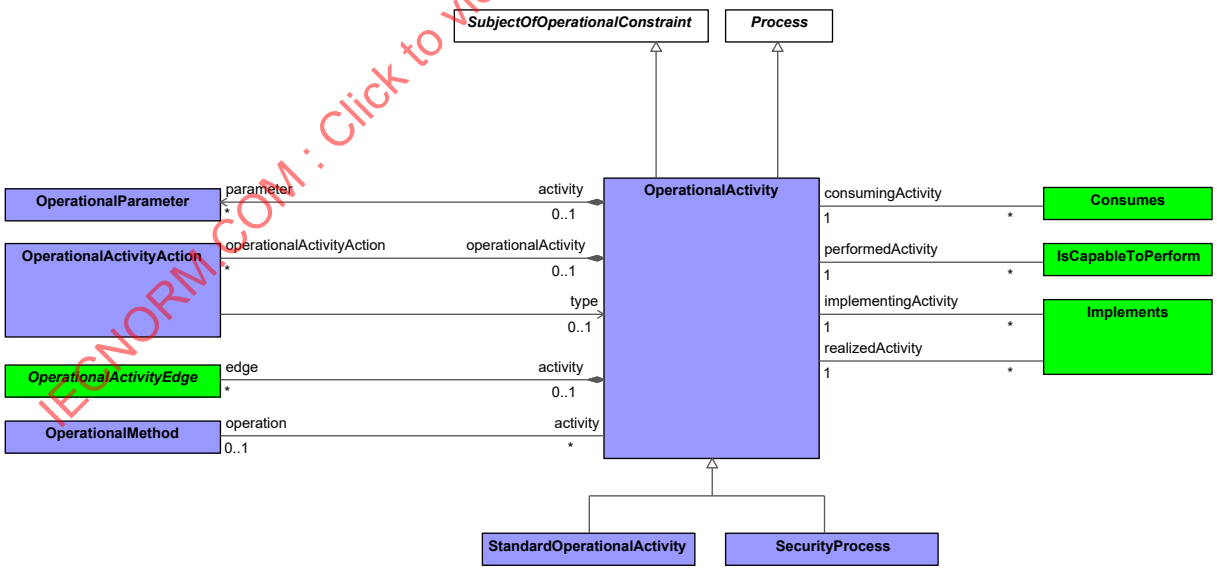

Figure 9:68 - OperationalActivity

OperationalActivityAction

Package: Processes

isAbstract: No

Generalization: ProcessUsage

Description

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

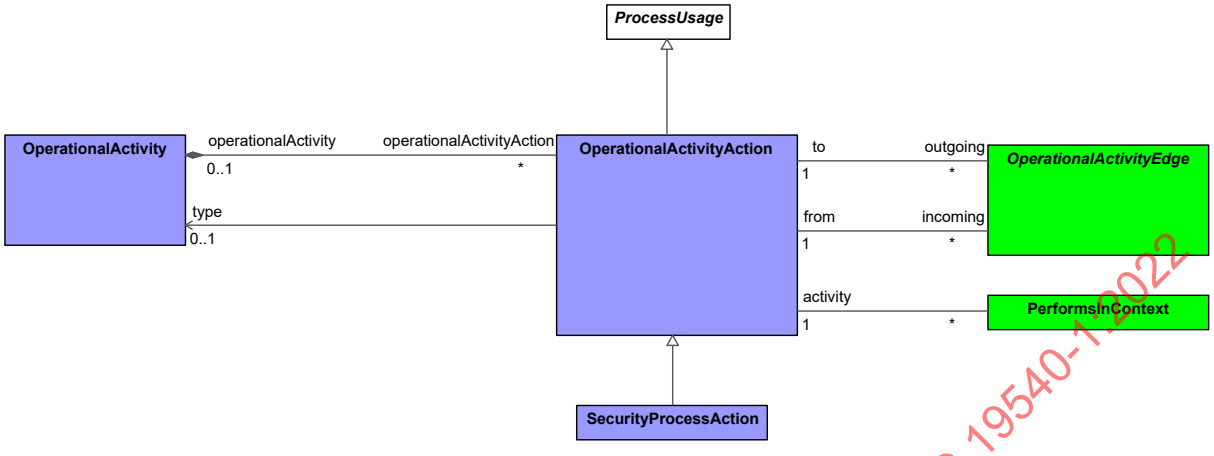

Figure 9:69 - OperationalActivityAction

OperationalActivityEdge

Package: Processes

isAbstract: Yes

Generalization: ProcessEdge

Description

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

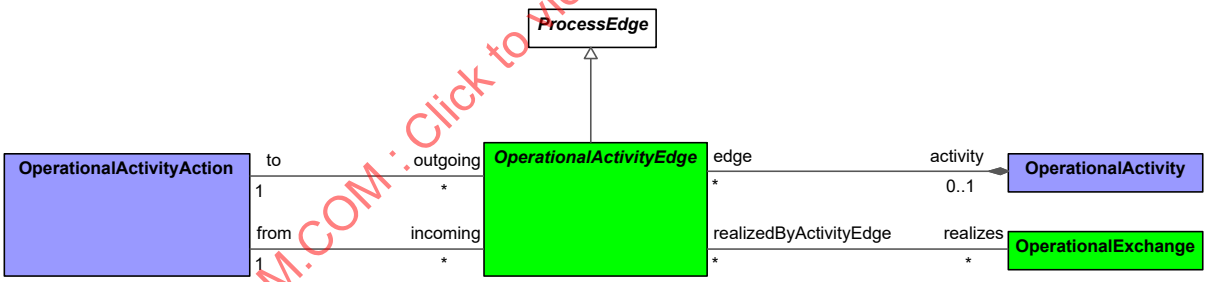

Figure 9:70 - OperationalActivityEdge

StandardOperationalActivity

Package: Processes

isAbstract: No

Generalization: OperationalActivity

Description

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

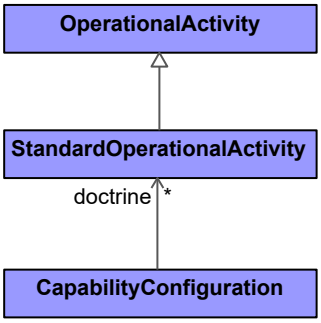

Figure 9:71 - StandardOperationalActivity

Domain MetaModel::Operational::States

OperationalStateDescription

Package: States

isAbstract: No

Generalization: MeasurableElement, StateDescription

Description

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

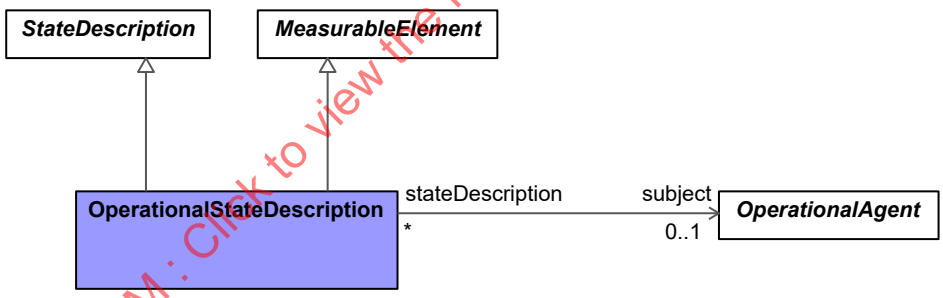

Figure 9:72 - OperationalStateDescription

Domain MetaModel::Operational::Interaction Scenarios

OperationalInteractionScenario

Package: Interaction Scenarios

isAbstract: No

Generalization: InteractionScenario

Description

A specification of the interactions between OperationalPerformers in an OperationalArchitecture.

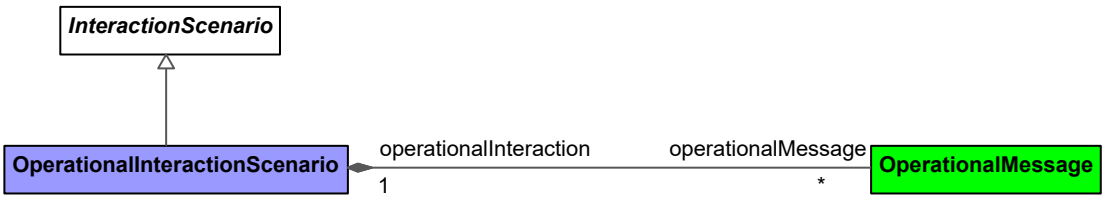

Figure 9:73 - OperationalInteractionScenario

OperationalMessage

Package: Interaction Scenarios

isAbstract: No

Generalization: InteractionMessage

Description

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

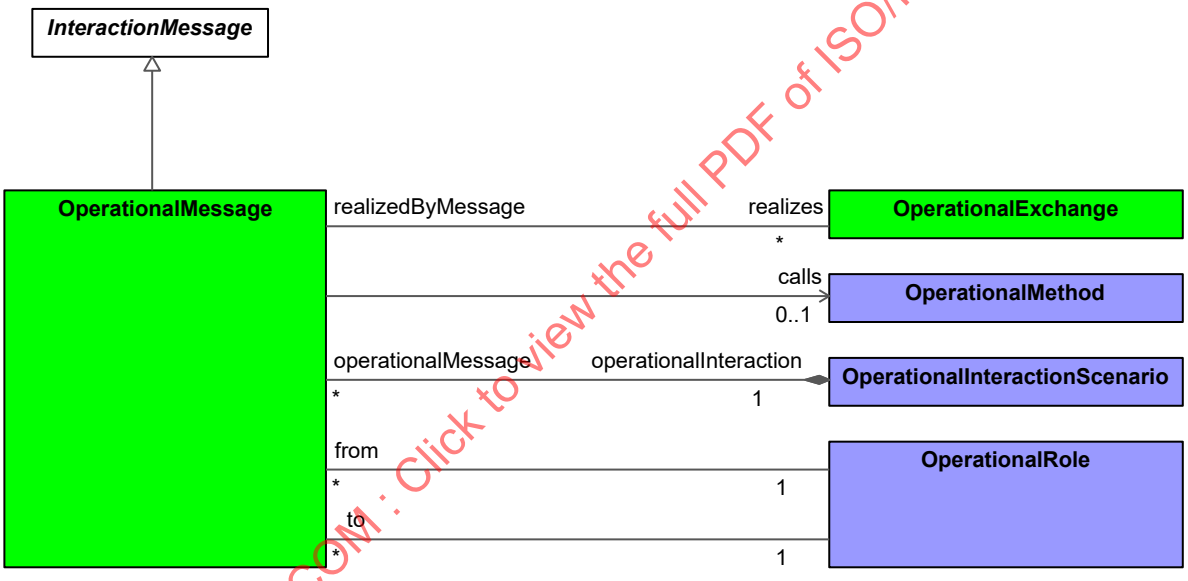

Figure 9:74 - OperationalMessage

Domain MetaModel::Operational::Information

InformationElement

Package: Information

isAbstract: No

Generalization: SubjectOfOperationalConstraint, OperationalAsset, OperationalExchangeItem

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).

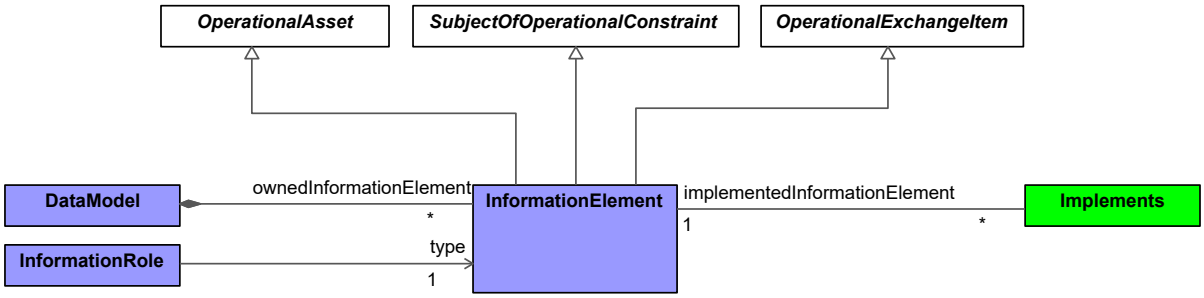

Figure 9:75 - InformationElement

Domain MetaModel::Operational::Constraints

OperationalConstraint

Package: Constraints

isAbstract: No

Generalization: Rule

Description

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

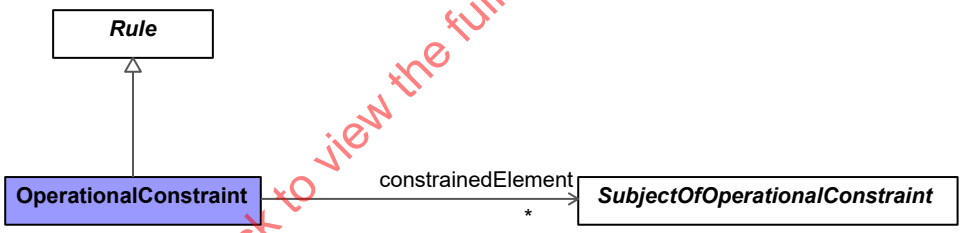

Figure 9:76 - OperationalConstraint

SubjectOfOperationalConstraint

Package: Constraints

isAbstract: Yes

Generalization: UAFElement

Description

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

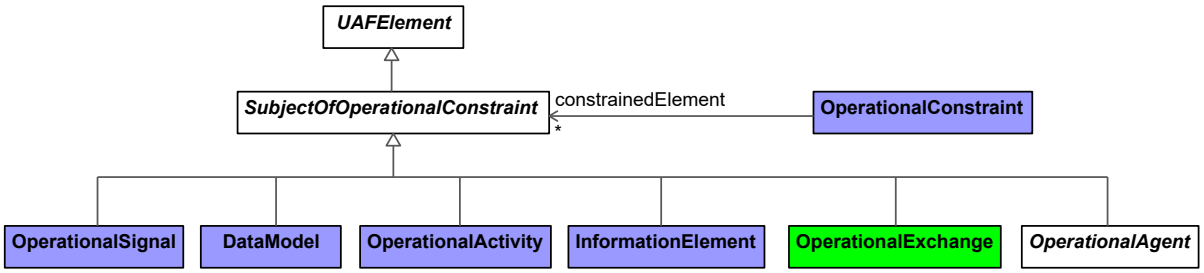

Figure 9:77 - SubjectOfOperationalConstraint

9.1.4 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 Service Specifications and required and provided service levels of these specifications required to exhibit a Capability or to support an Operational Activity.

Domain MetaModel::Services::Taxonomy

ServiceSpecification

Package: Taxonomy

isAbstract: No

Generalization: PropertySet, VersionedElement, CapableElement

Description

The specification of a set of functionalities provided by one element for the use of others.

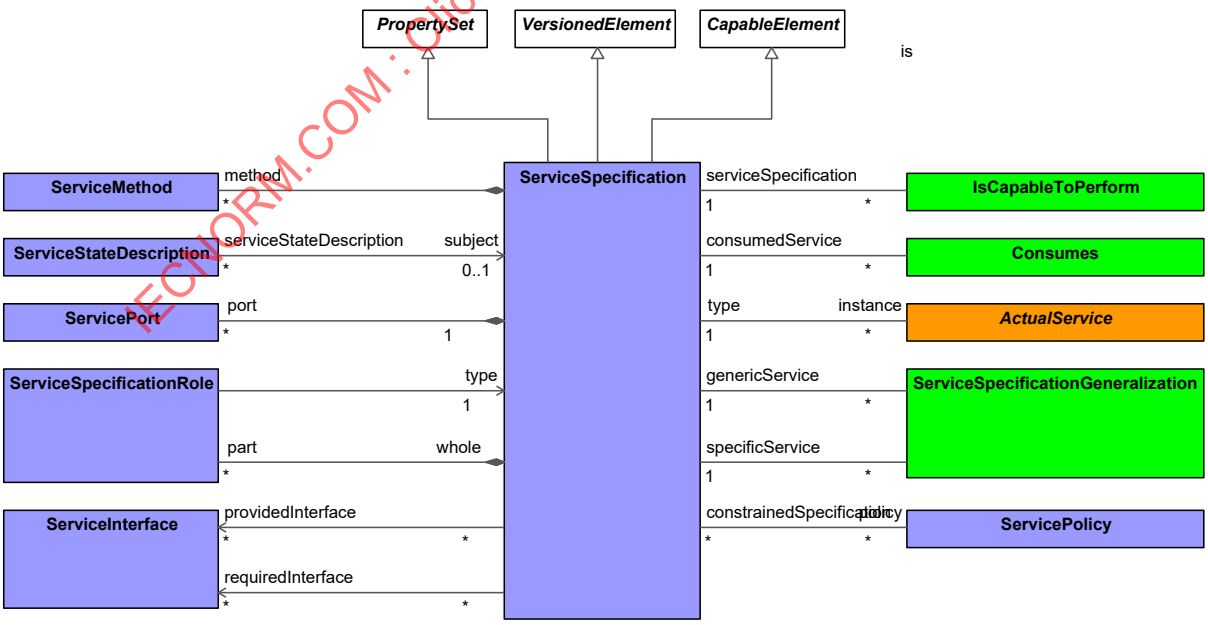

Figure 9:78 - ServiceSpecification

ServiceSpecificationGeneralization

Package: Taxonomy

isAbstract: No

Generalization: PropertySetGeneralization

Description

A ServiceSpecificationGeneralization is a taxonomic relationship between a more general ServiceSpecification and a more specific ServiceSpecification.

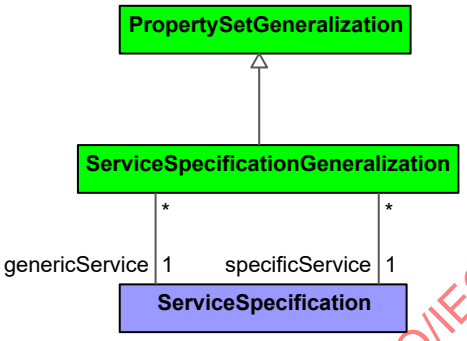

Figure 9:79 - ServiceSpecificationGeneralization

Domain MetaModel::Services::Structure

ServiceConnector

Package: Structure

isAbstract: No

Generalization: MeasurableElement

Description

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

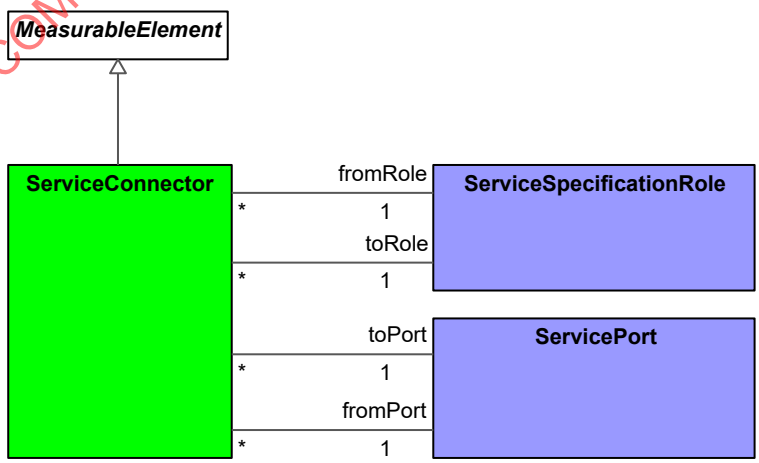

Figure 9:80 - ServiceConnector

ServiceMethod

Package: Structure
isAbstract: No
Generalization: ProcessOperation

Description

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

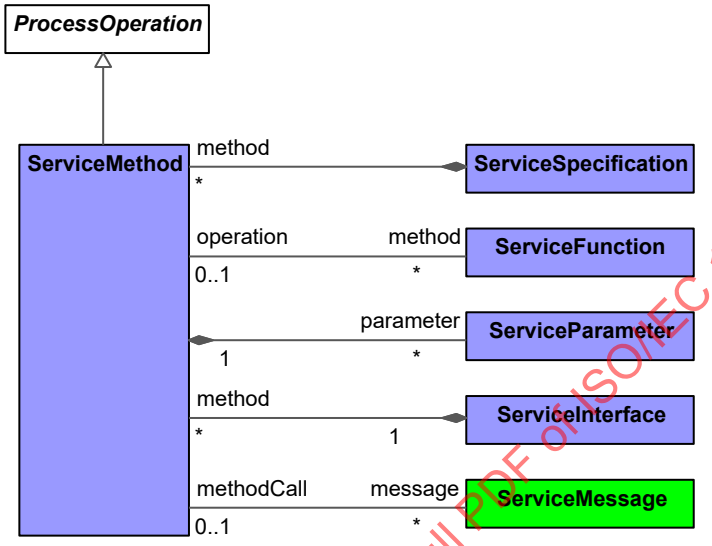

Figure 9:81 - ServiceMethod

ServiceParameter

Package: Structure
isAbstract: No
Generalization: ProcessParameter

Description

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

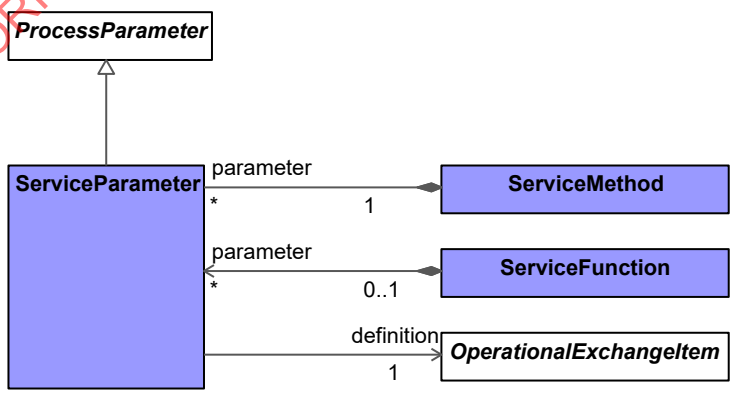

Figure 9:82 - ServiceParameter

ServiceSpecificationRole

Package: Structure

isAbstract: No

Generalization: MeasurableElement, InteractionRole

Description

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

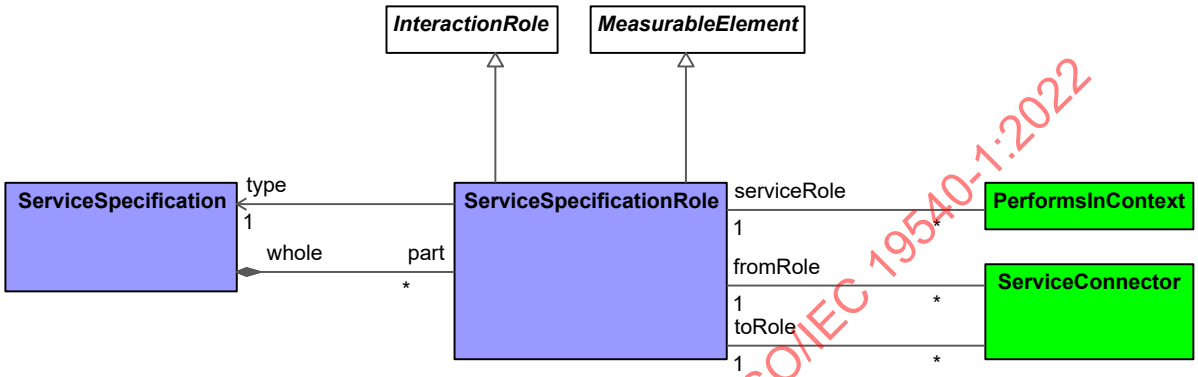

Figure 9:83 - ServiceSpecificationRole

Domain MetaModel::Services::Connectivity

ServiceInterface

Package: Connectivity

isAbstract: No

Generalization: PropertySet

Description

A contract that defines the ServiceMethods and ServiceMessageHandlers that the ServiceSpecification realizes.

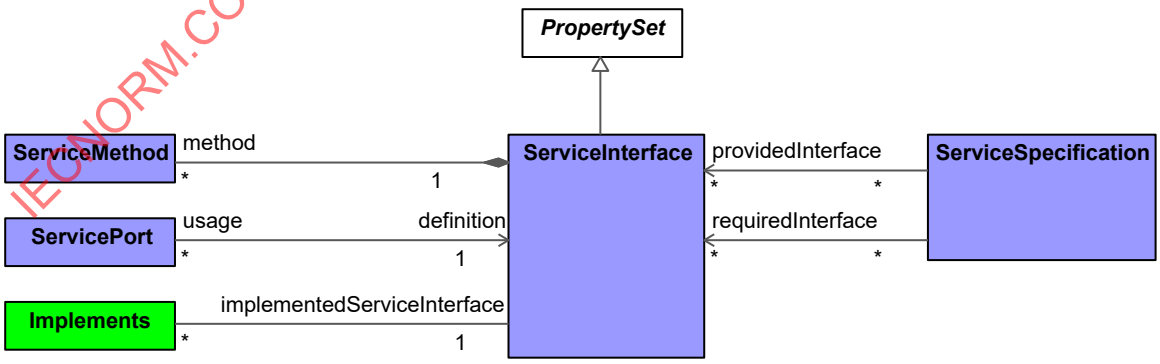

Figure 9:84 - ServiceInterface

ServicePort

Package: Connectivity

isAbstract: No

Generalization: MeasurableElement

Description

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

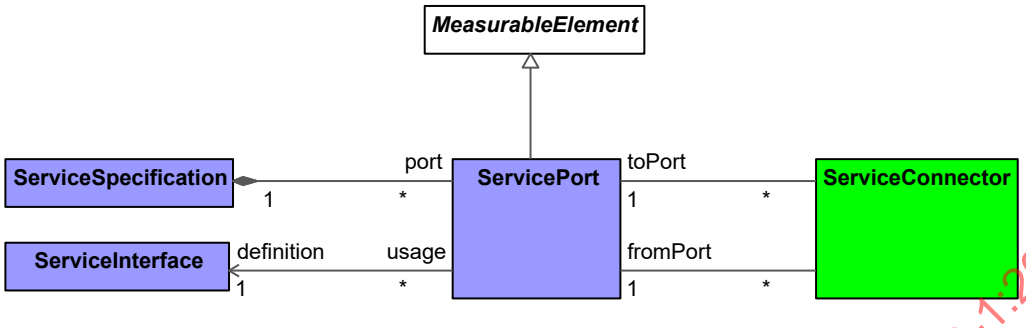

Figure 9:85 - ServicePort

Domain MetaModel::Services::Processes

ServiceFunction

Package: Processes

isAbstract: No

Generalization: Process

Description

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

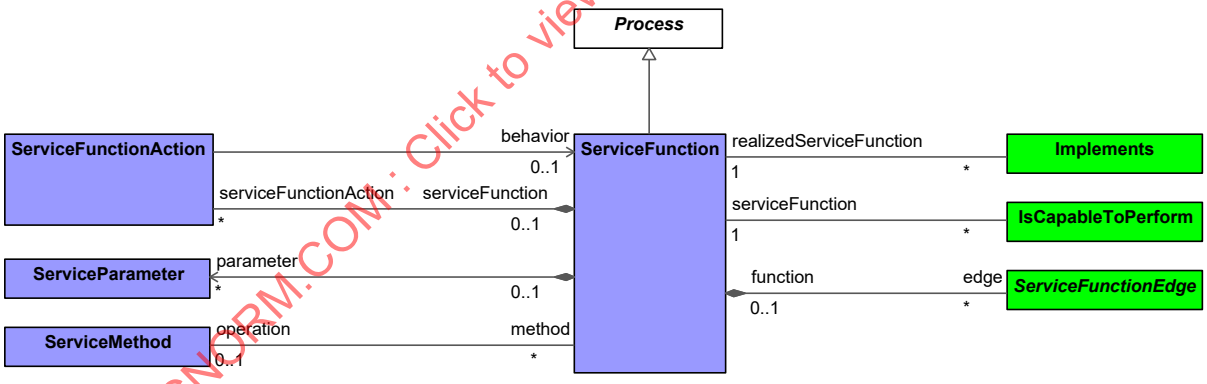

Figure 9:86 - ServiceFunction

ServiceFunctionAction

Package: Processes

isAbstract: No

Generalization: ProcessUsage

Description

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

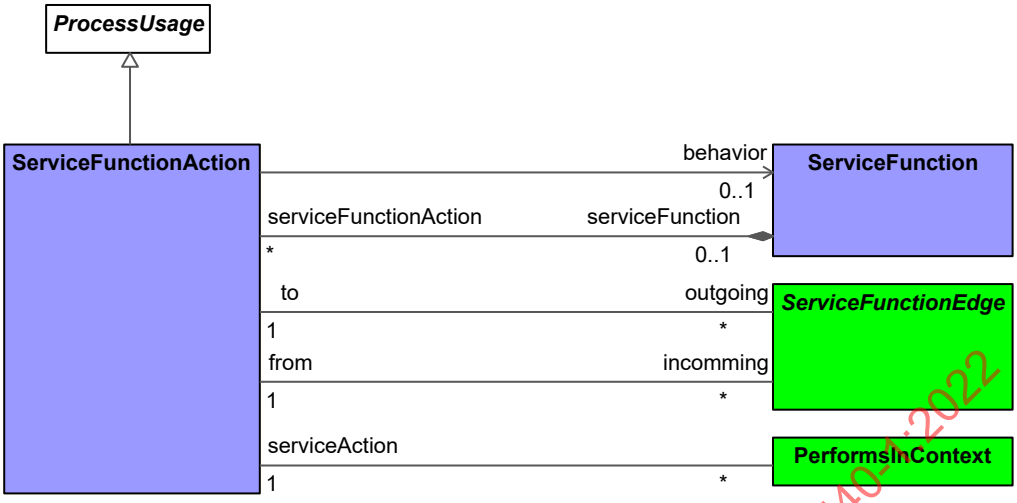

Figure 9:87 - ServiceFunctionAction

ServiceFunctionEdge

Package: Processes

isAbstract: Yes

Generalization: ProcessEdge

Description

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

Figure 9:88 - ServiceFunctionEdge

Domain MetaModel::Services::States

ServiceStateDescription

Package: States

isAbstract: No

Generalization: MeasurableElement, StateDescription

Description

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

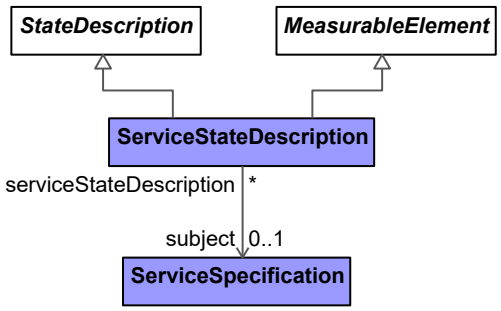

Figure 9:89 - ServiceStateDescription

Domain MetaModel::Services::Interaction Scenarios

ServiceInteractionScenario

Package: Interaction Scenarios
isAbstract: No
Generalization: InteractionScenario

Description

A specification of the interactions between ServiceSpecifications.

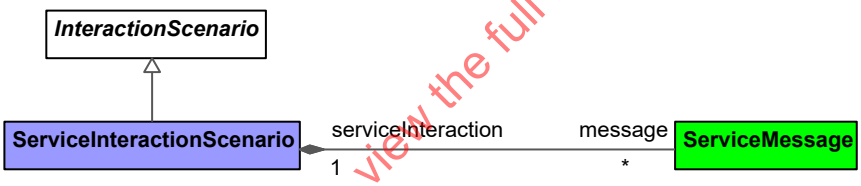

Figure 9:90 - ServiceInteractionScenario

ServiceMessage

Package: Interaction Scenarios
isAbstract: No
Generalization: InteractionMessage

Description

Message for use in a Service Event-Trace.

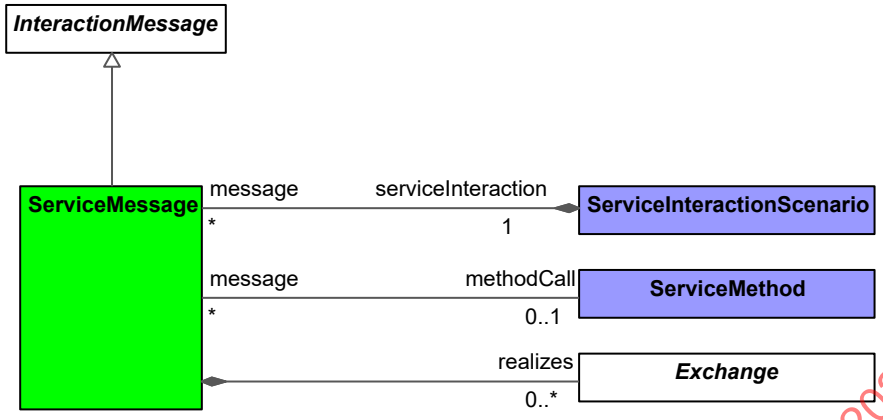

Figure 9:91 - ServiceMessage

Domain MetaModel::Services::Constraints

ServicePolicy

Package: Constraints

isAbstract: No

Generalization: Rule

Description

A constraint governing the use of one or more ServiceSpecifications.

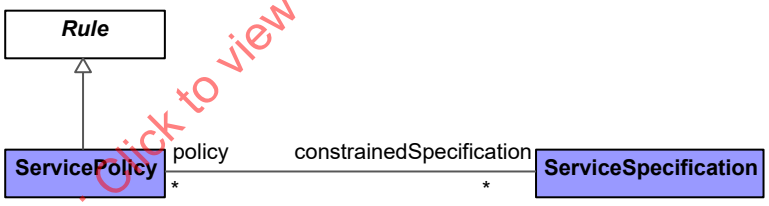

Figure 9:92 - ServicePolicy

Domain MetaModel::Services::Traceability

Consumes

Package: Traceability

isAbstract: No

Generalization: MeasurableElement

Description

A tuple that asserts that an OperationalActivity make use of a service.

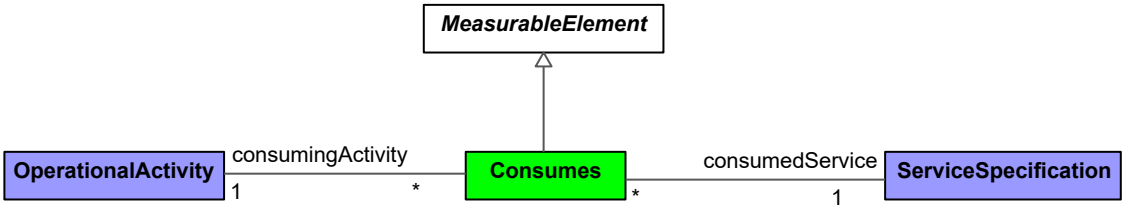

Figure 9:93 - Consumes

9.1.5 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).

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.

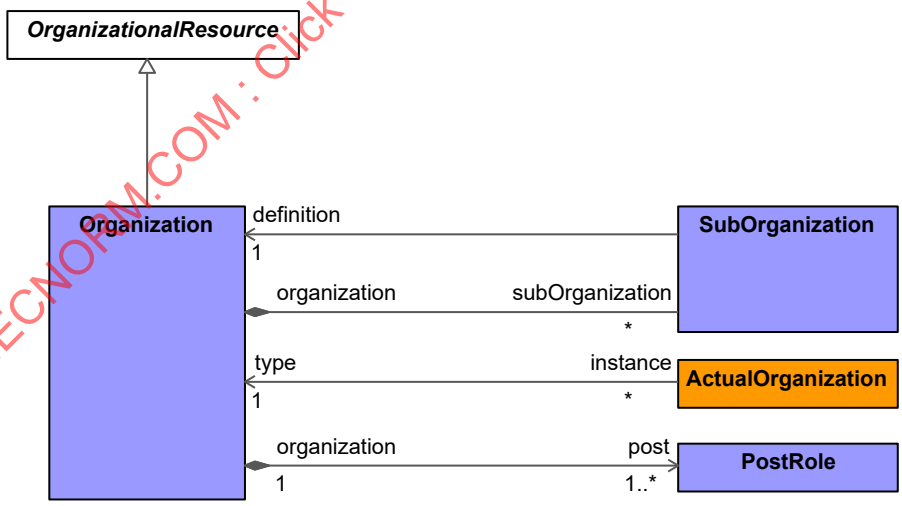

Figure 9:94 - Organization

OrganizationalResource

Package: Taxonomy

isAbstract: Yes

Generalization: PhysicalResource, Stakeholder

Description

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

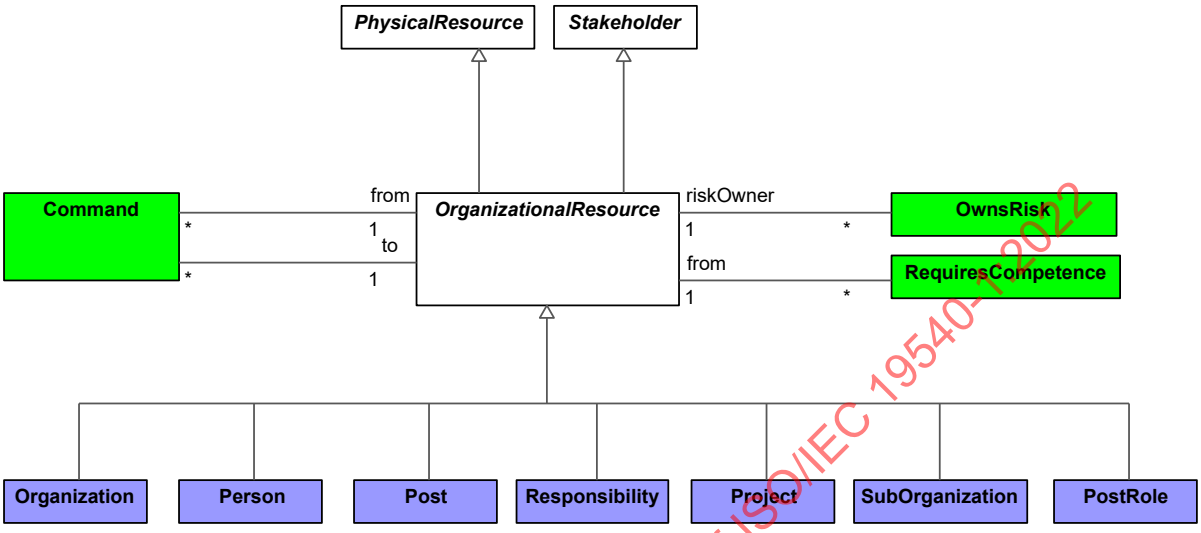

Figure 9:95 - OrganizationalResource

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.).

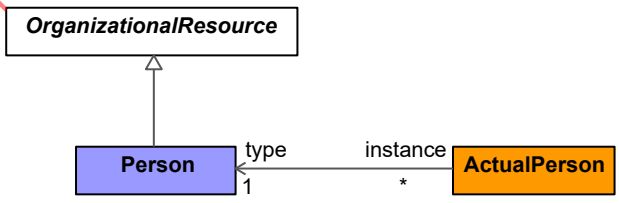

Figure 9:96 - Person

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).

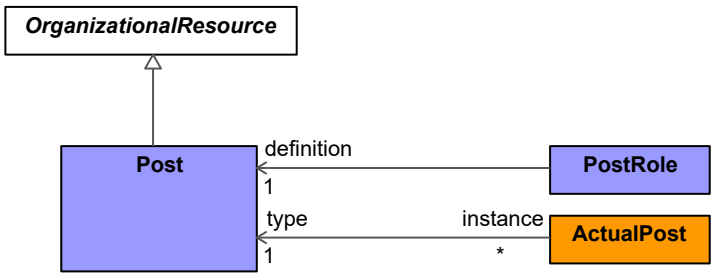

Figure 9:97 - Post

Responsibility

Package: Taxonomy
isAbstract: No
Generalization: OrganizationalResource

Description

The type of duty required of a Person or Organization.

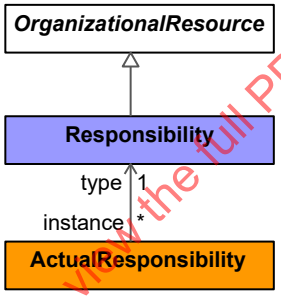

Figure 9:98 - Responsibility

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.

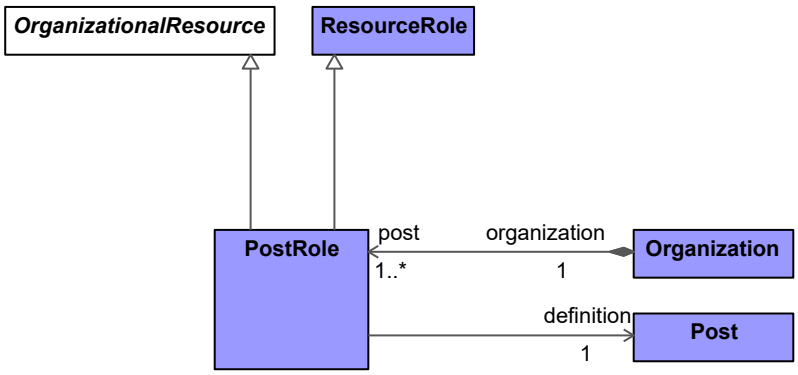

Figure 9:99 - PostRole

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.).

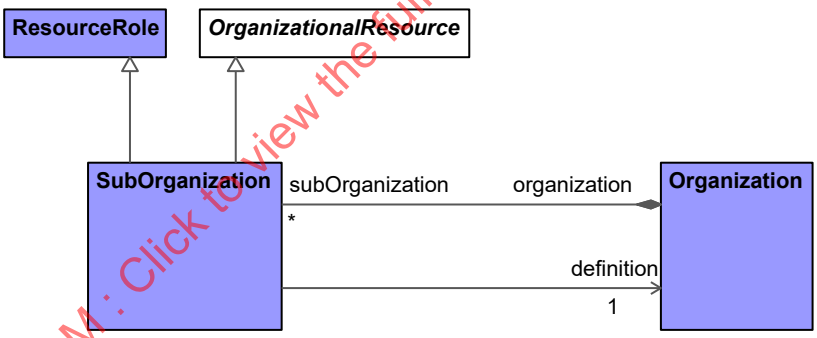

Figure 9:100 - SubOrganization

Domain MetaModel::Personnel::Connectivity

Command

Package: Connectivity

isAbstract: No

Generalization: ResourceExchange

Description

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

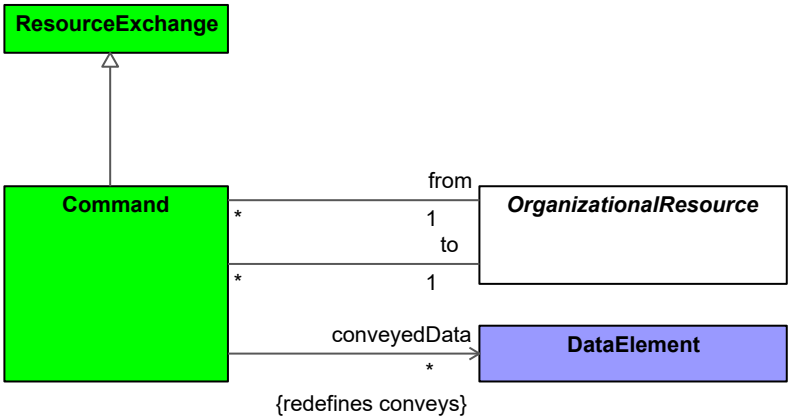

Figure 9:101 - Command

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).

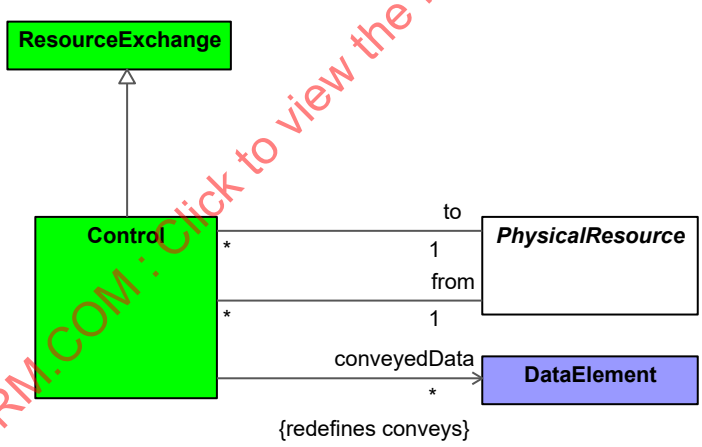

Figure 9:102 - Control

Domain MetaModel::Personnel::Interaction Scenarios

ResourceInteractionScenario

Package: Interaction Scenarios

isAbstract: No

Generalization: InteractionScenario

Description

A specification of the interactions between ResourcePerformers in a ResourceArchitecture.

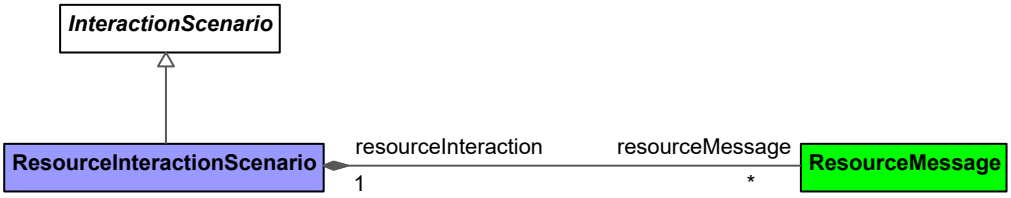

Figure 9:103 - ResourceInteractionScenario

Domain MetaModel::Personnel::Constraints

Competence

Package: Constraints

isAbstract: No

Generalization: PropertySet, SubjectOfForecast

Description

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

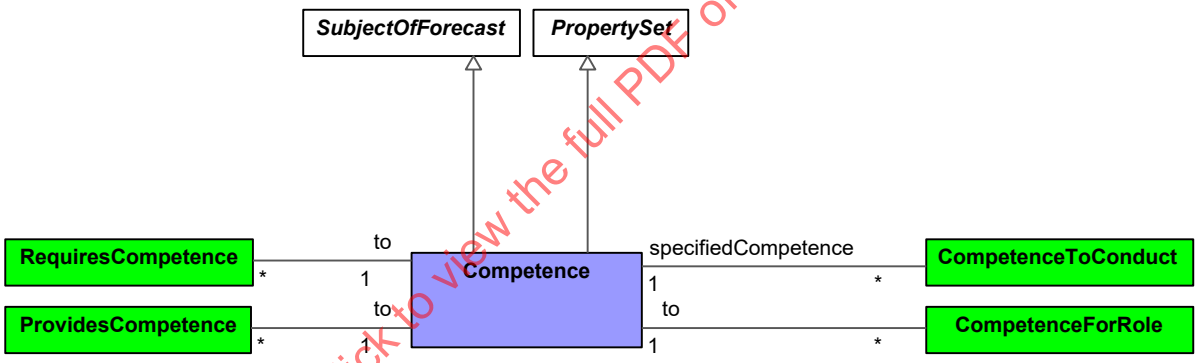

Figure 9:104 - Competence

CompetenceForRole

Package: Constraints

isAbstract: No

Generalization: MeasurableElement

Description

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

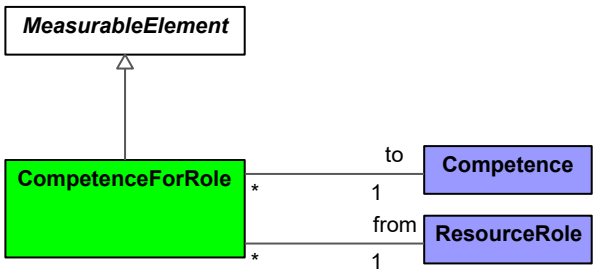

Figure 9:105 - CompetenceForRole

RequiresCompetence

Package: Constraints

isAbstract: No

Generalization: MeasurableElement

Description

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

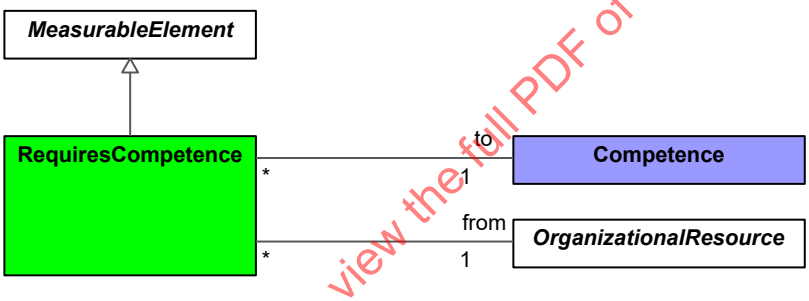

Figure 9:106 - RequiresCompetence

Domain MetaModel::Personnel::Roadmap

FillsPost

Package: Roadmap

isAbstract: No

Generalization: MeasurableElement

Description

A tuple that asserts that an ActualPerson fills an ActualPost.

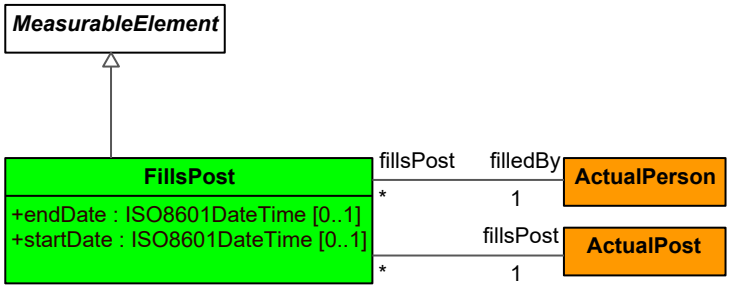

Figure 9:107 - FillsPost

Attributes

endDate : ISO8601DateTime[0..1]	End date of an ActualPerson filling an ActualPost.
startDate : ISO8601DateTime[0..1]	Start date of an ActualPerson filling an ActualPost.

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.

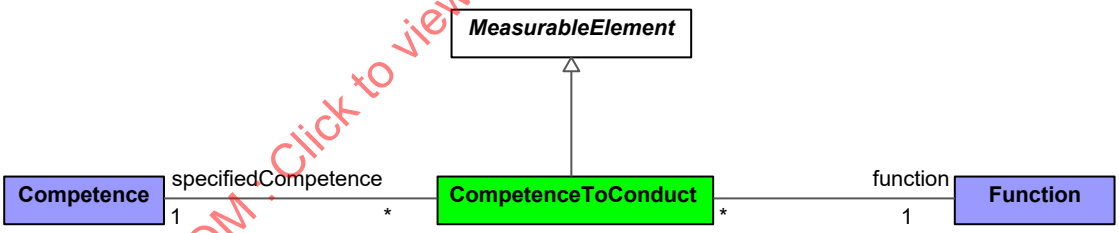

Figure 9:108 - CompetenceToConduct

9.1.6 Domain MetaModel::Resources

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.

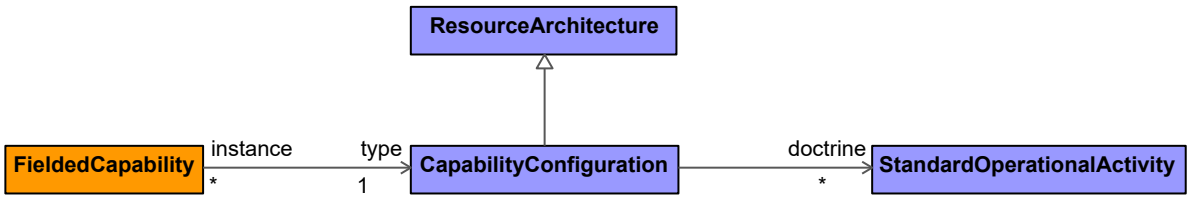

Figure 9:109 - CapabilityConfiguration

NaturalResource

Package: Taxonomy
isAbstract: No
Generalization: PhysicalResource

Description

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

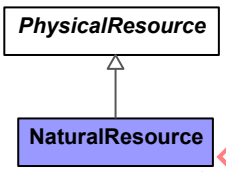

Figure 9:110 - NaturalResource

PhysicalResource

Package: Taxonomy
isAbstract: Yes
Generalization: ResourcePerformer

Description

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

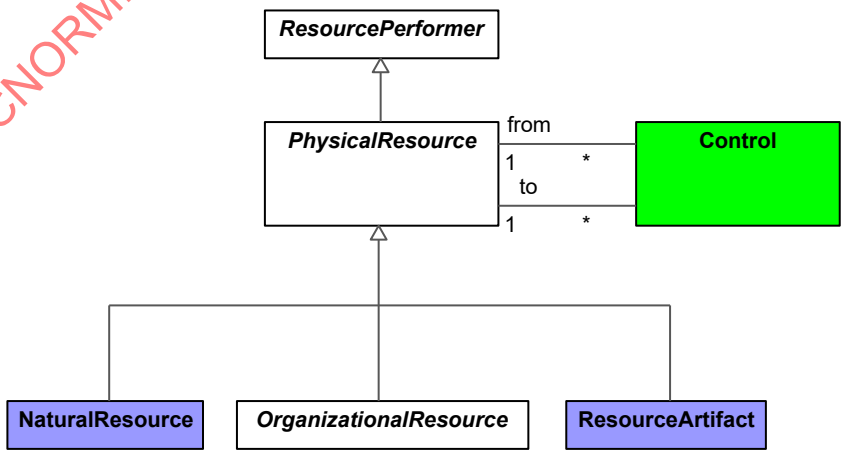

Figure 9:111 - PhysicalResource

ResourceArchitecture

Package: Taxonomy

isAbstract: No

Generalization: ResourcePerformer, Architecture

Description

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

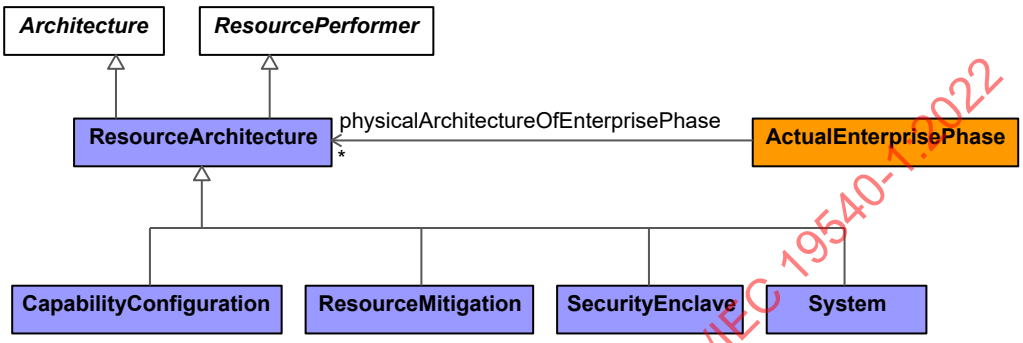

Figure 9:112 - ResourceArchitecture

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.).

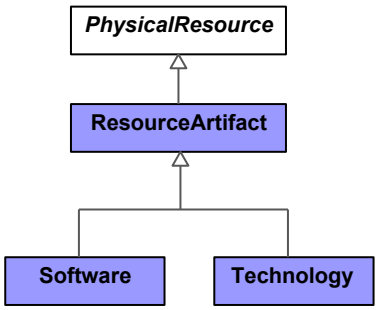

Figure 9:113 - ResourceArtifact

ResourcePerformer

Package: Taxonomy

isAbstract: Yes

Generalization: ResourceExchangeItem, SubjectOfResourceConstraint, OperationalExchangeItem, SubjectOfForecast, CapableElement, Desirer, VersionedElement, ResourceAsset

Description

An abstract grouping of elements that can perform Functions.

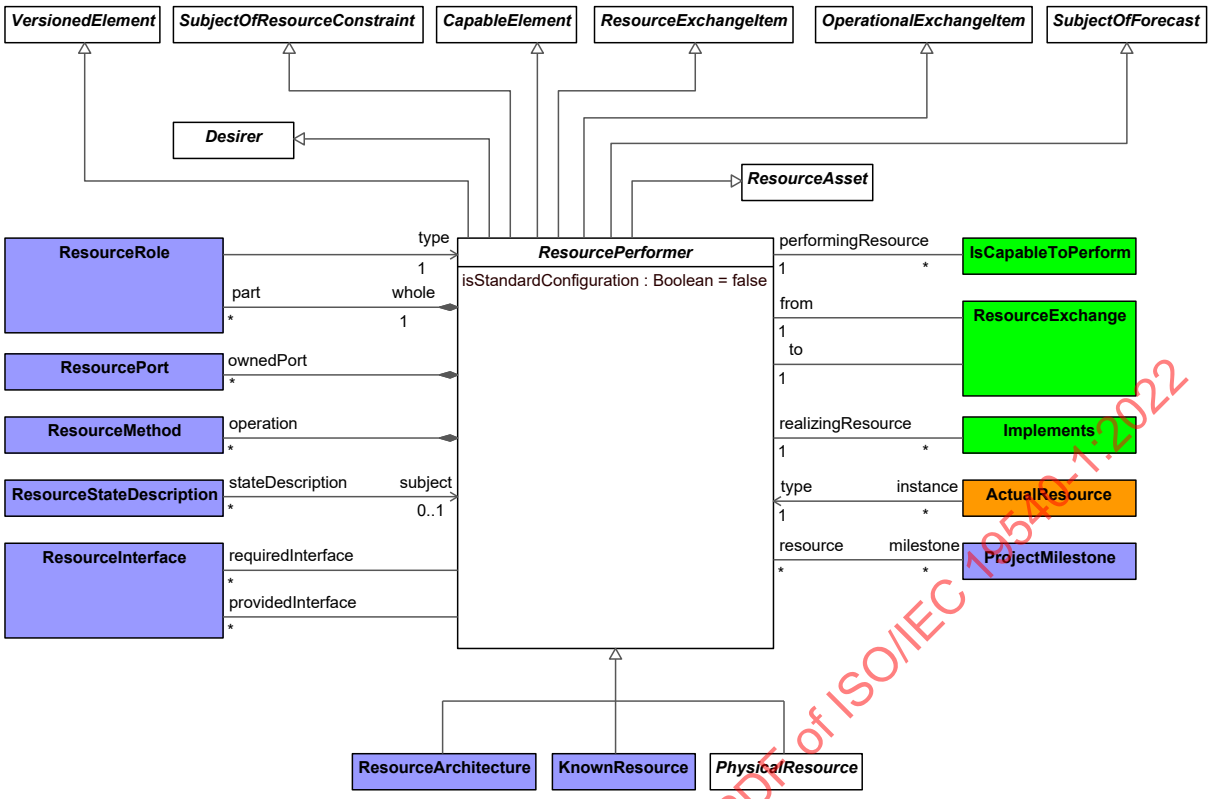

Figure 9:114 – ResourcePerformer

Attributes

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

Software

Package: Taxonomy
isAbstract: No
Generalization: ResourceArtifact

Description

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

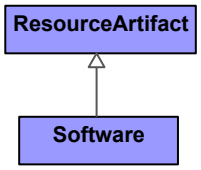

Figure 9:115 - Software

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).

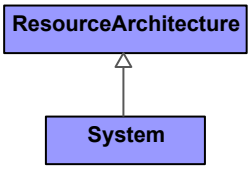

Figure 9:116 - System

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.

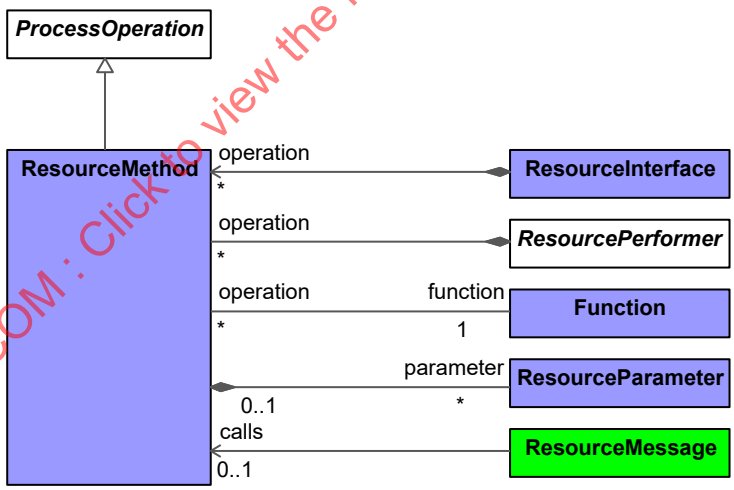

Figure 9:117 - ResourceMethod

ResourceParameter

Package: Structure

isAbstract: No

Generalization: ProcessParameter

Description

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

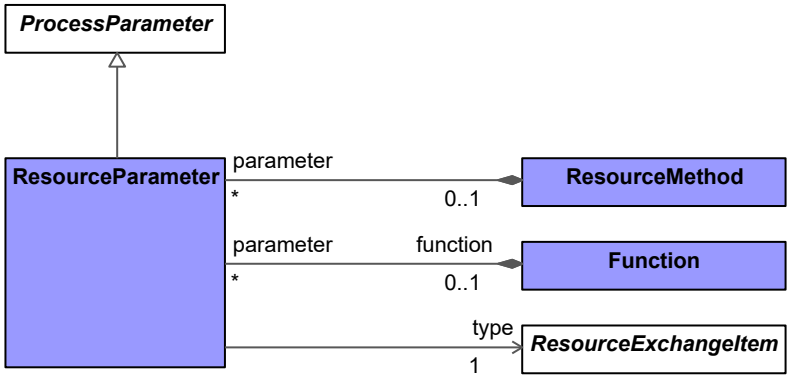

Figure 9:118 - ResourceParameter

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.

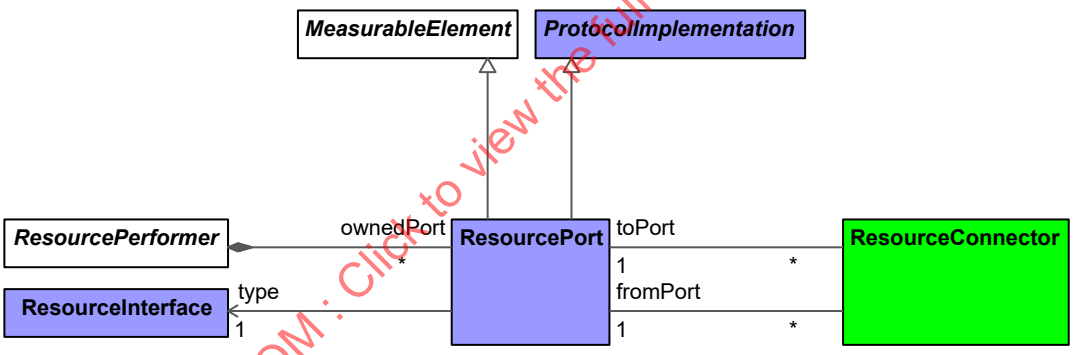

Figure 9:119 - ResourcePort

ResourceRole

Package: Structure

isAbstract: No

Generalization: SubjectOfResourceConstraint, LocationHolder, AssetRole, InteractionRole

Description

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

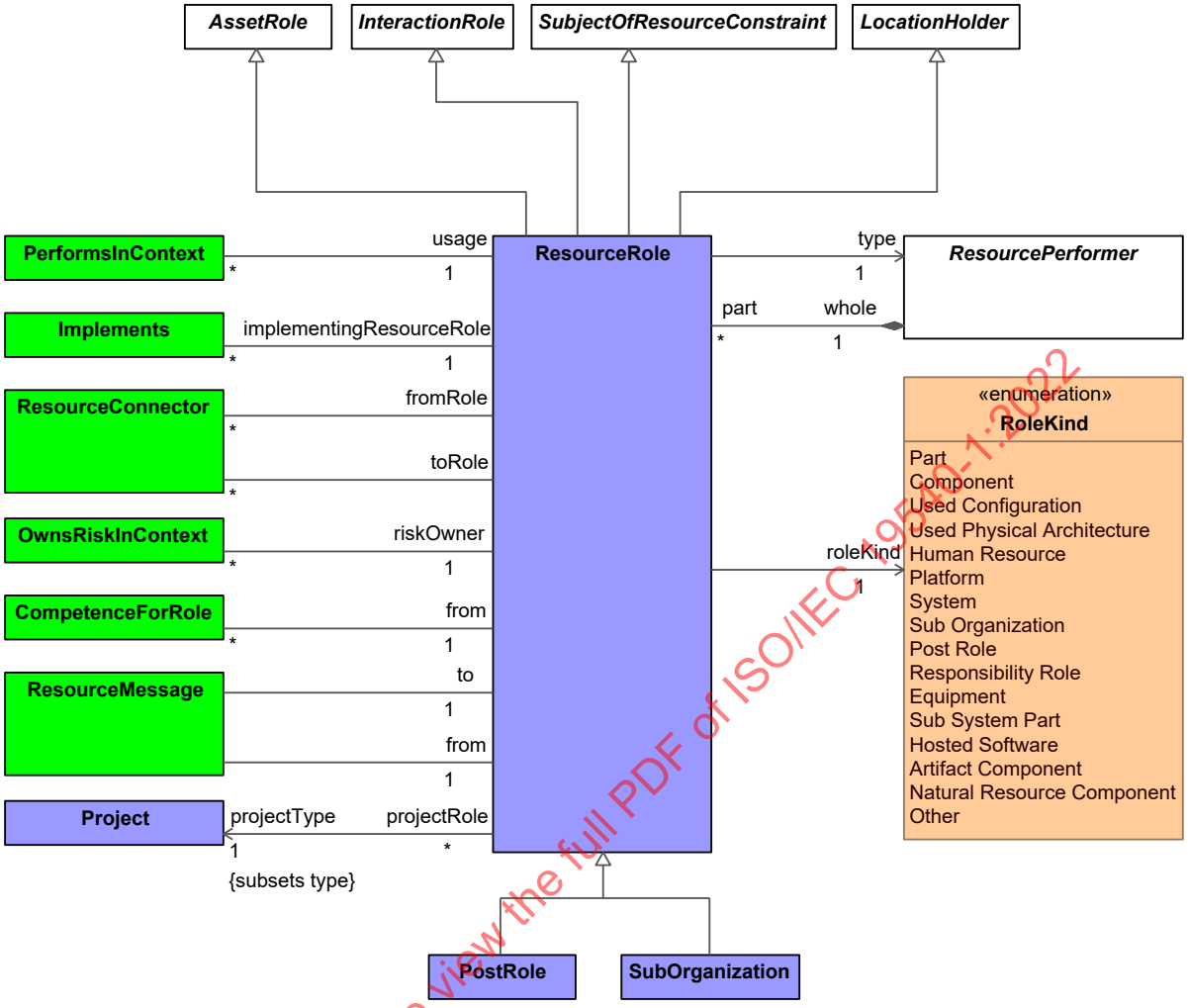

Figure 9:120 - ResourceRole

Domain MetaModel::Resources::Connectivity

ResourceConnector

Package: Connectivity

isAbstract: No

Generalization: ProtocolImplementation, MeasurableElement

Description

A channel for exchange between two ResourceRoles.

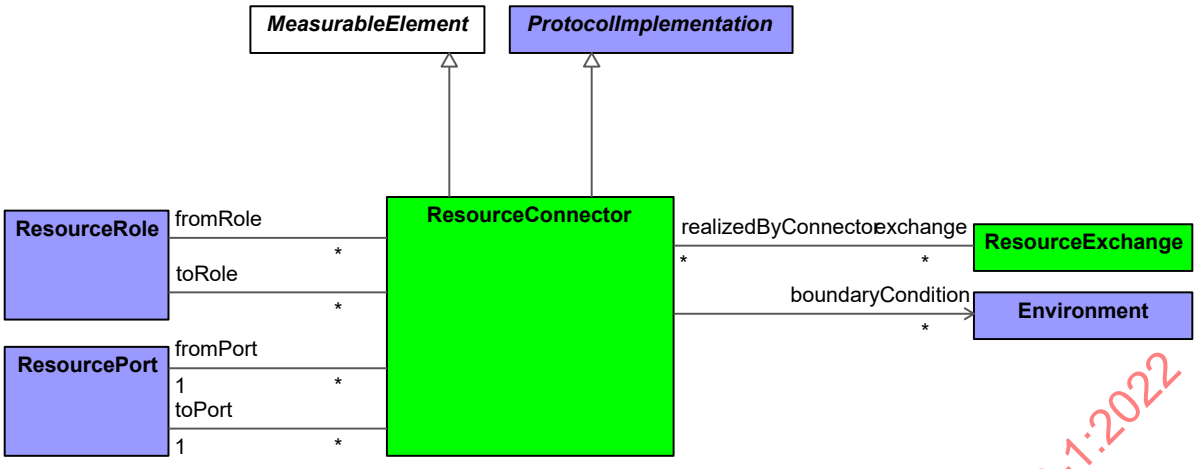

Figure 9:121 - ResourceConnector

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).

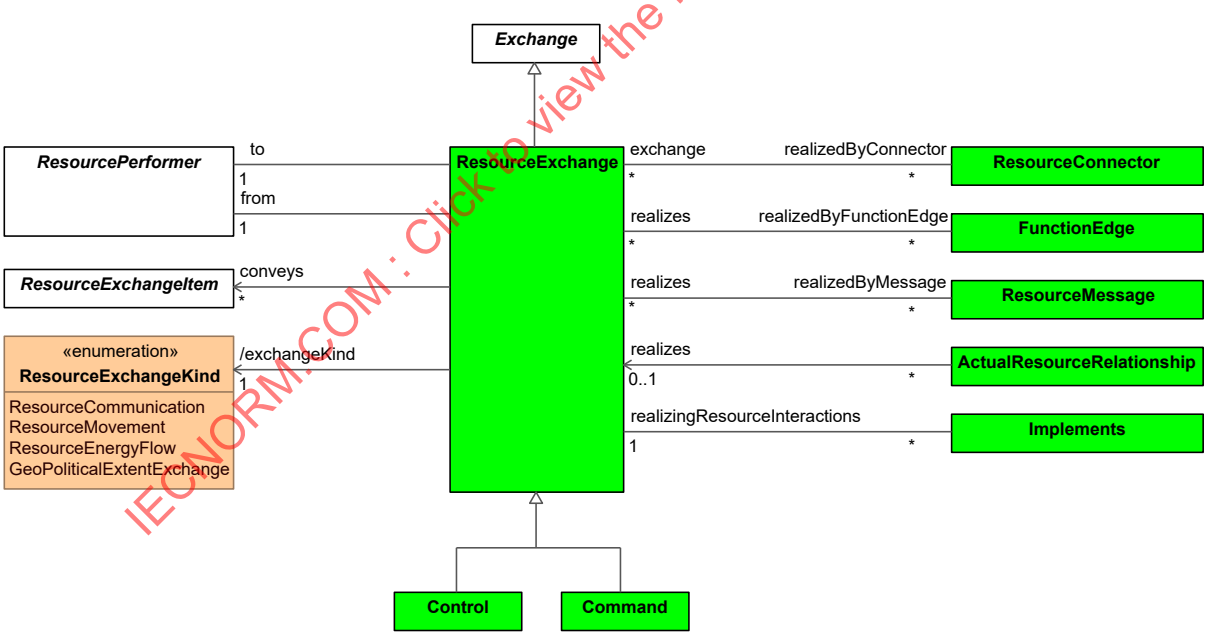

Figure 9:122 - ResourceExchange

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.

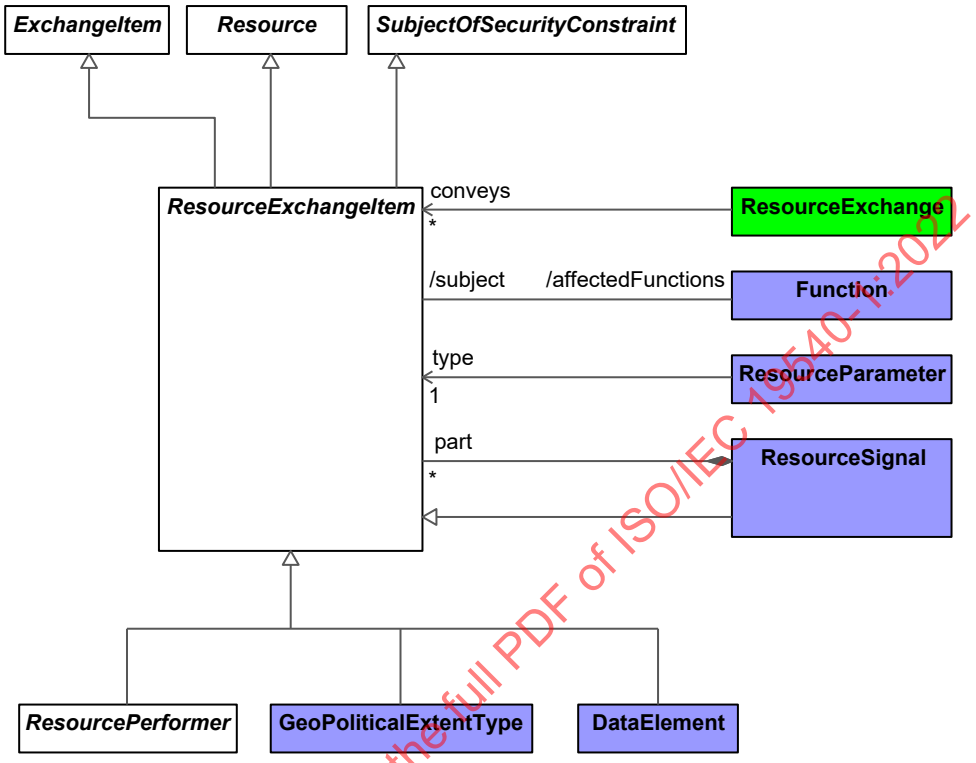

Figure 9:123 - ResourceExchangeItem

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.

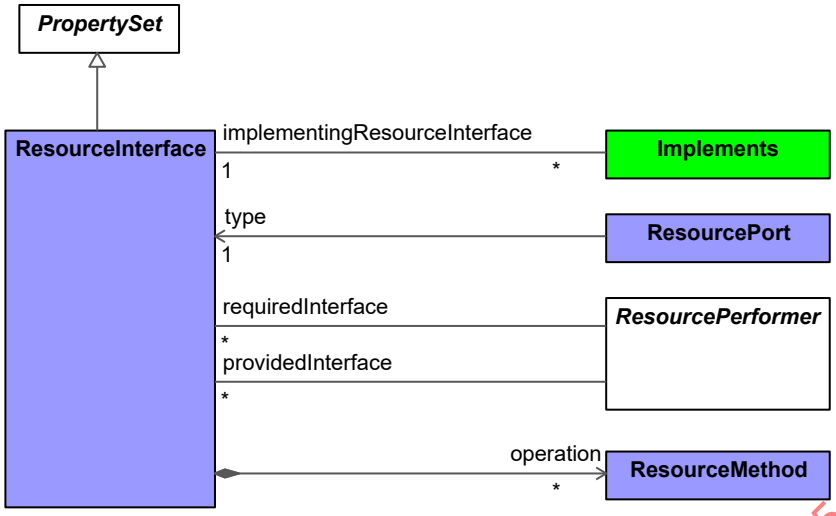

Figure 9:124 - ResourceInterface

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.

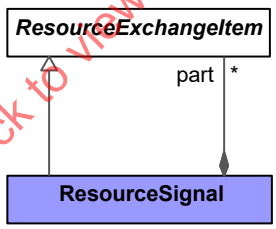

Figure 9:125 - ResourceSignal

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.

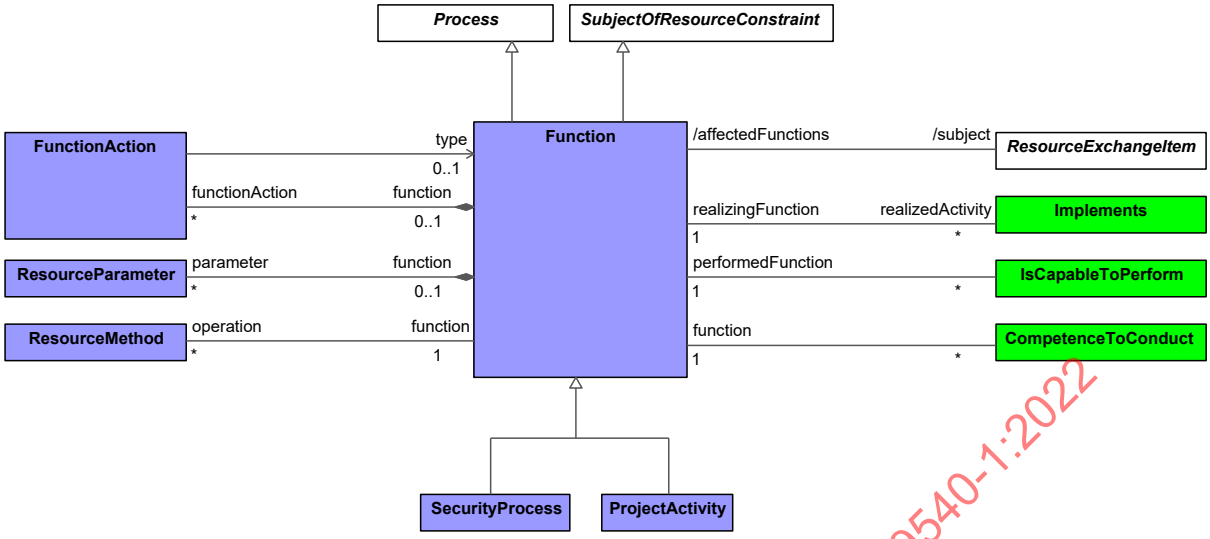

Figure 9:126 - Function

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.

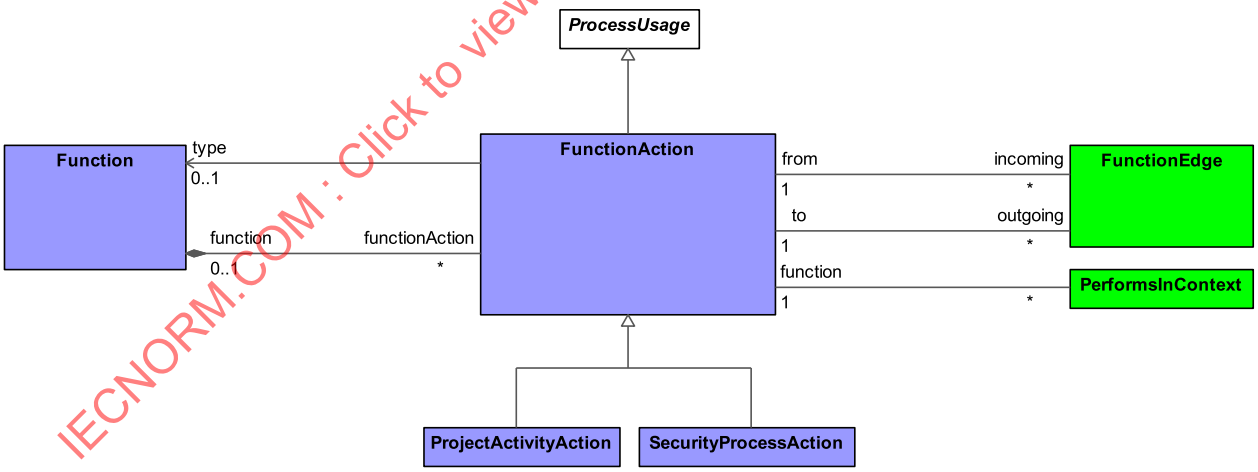

Figure 9:127 - FunctionAction

FunctionEdge

Package: Processes

isAbstract: No

Generalization: ProcessEdge

Description

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

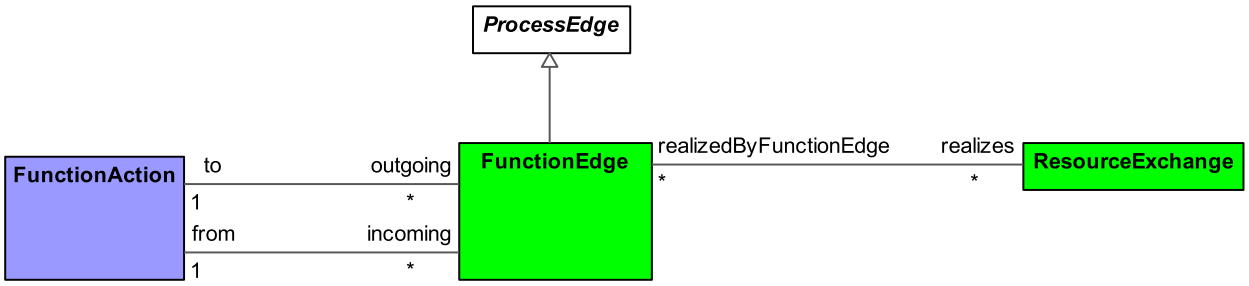

Figure 9:128 - FunctionEdge

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.

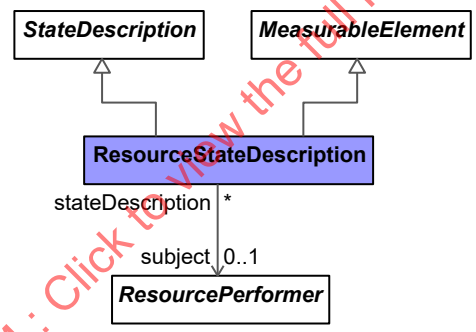

Figure 9:129 - ResourceStateDescription

Domain MetaModel::Resources::Interaction Scenarios

ResourceMessage

Package: Interaction Scenarios

isAbstract: No

Generalization: InteractionMessage

Description

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

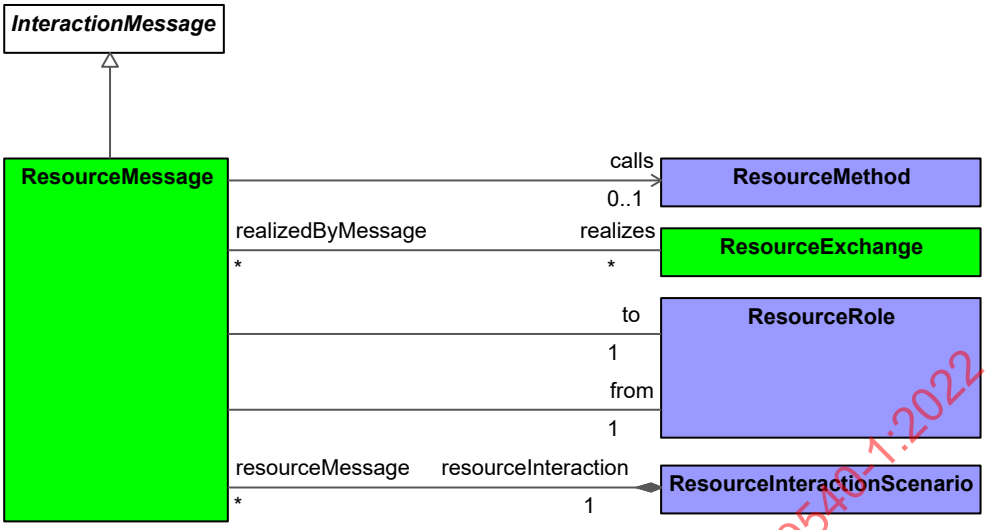

Figure 9:130 - ResourceMessage

Domain MetaModel::Resources::Information

DataElement

Package: Information

isAbstract: No

Generalization: SubjectOfResourceConstraint, ResourceAsset, ResourceExchangeItem

Description

A formalized representation of data that is managed by or exchanged between resources.

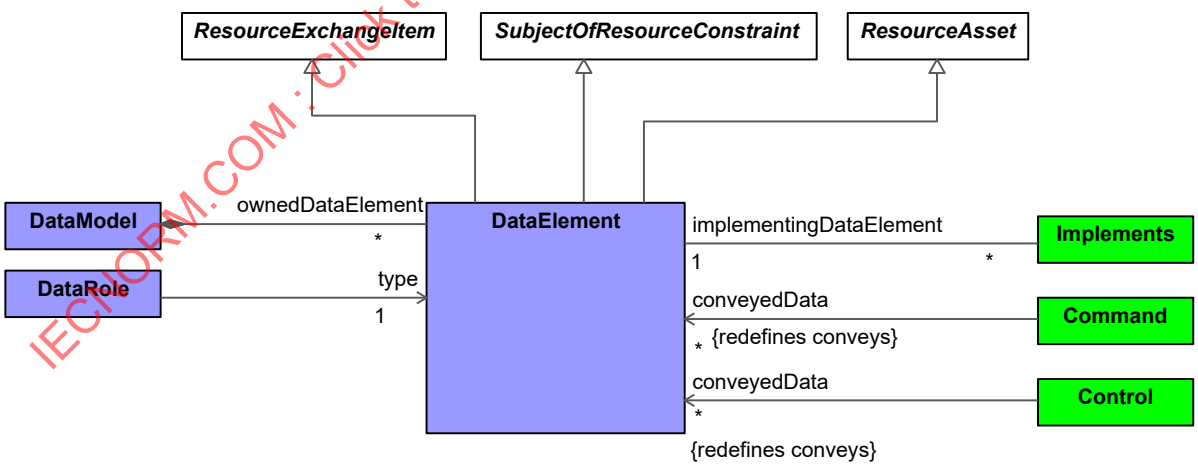

Figure 9:131 - DataElement

DataRole

Package: Information

isAbstract: No

Generalization: AssetRole

Description

A usage of DataElement that exists in the context of a ResourceAsset. It also allows the representation of the whole-part aggregation of DataElements.

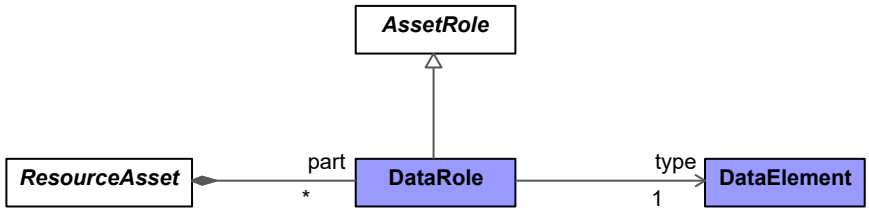

Figure 9:132 - DataRole

Domain MetaModel::Resources::Constraints

ResourceConstraint

Package: Constraints

isAbstract: No

Generalization: Rule

Description

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

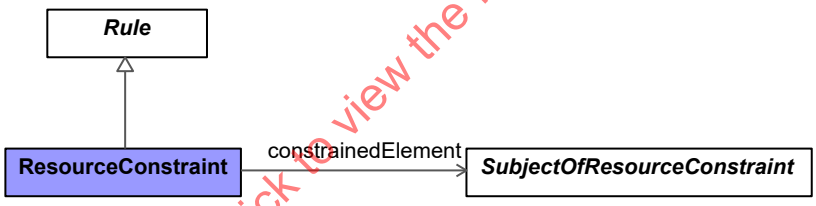

Figure 9:133 - ResourceConstraint

SubjectOfResourceConstraint

Package: Constraints

isAbstract: Yes

Generalization: UAFElement

Description

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

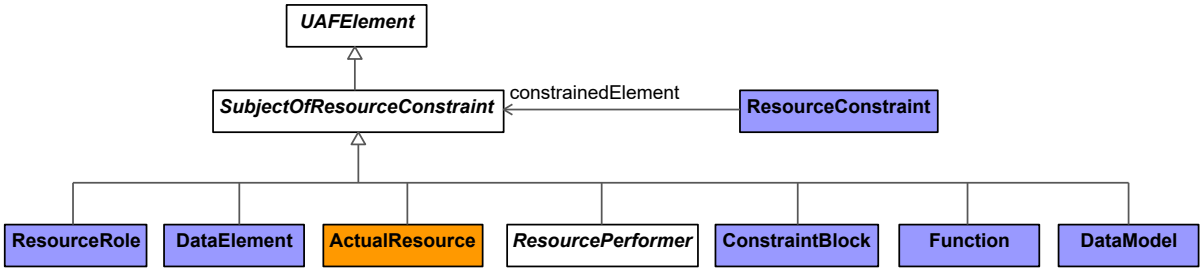

Figure 9:134 - SubjectOfResourceConstraint

Domain MetaModel::Resources::Roadmap

Forecast

Package: Roadmap

isAbstract: No

Generalization: MeasurableElement

Description

A tuple that specifies a transition from one Asset, Standard, Competence to another future one. It is related to an ActualEnterprisePhase to give it a temporal context.

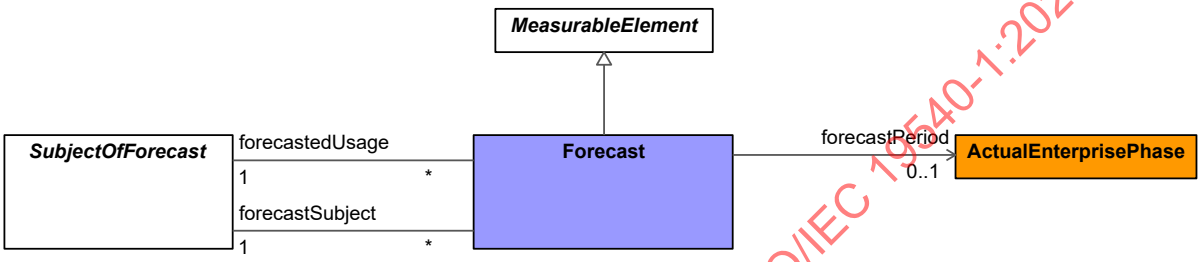

Figure 9:135 - Forecast

SubjectOfForecast

Package: Roadmap

isAbstract: Yes

Generalization: UAFElement

Description

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

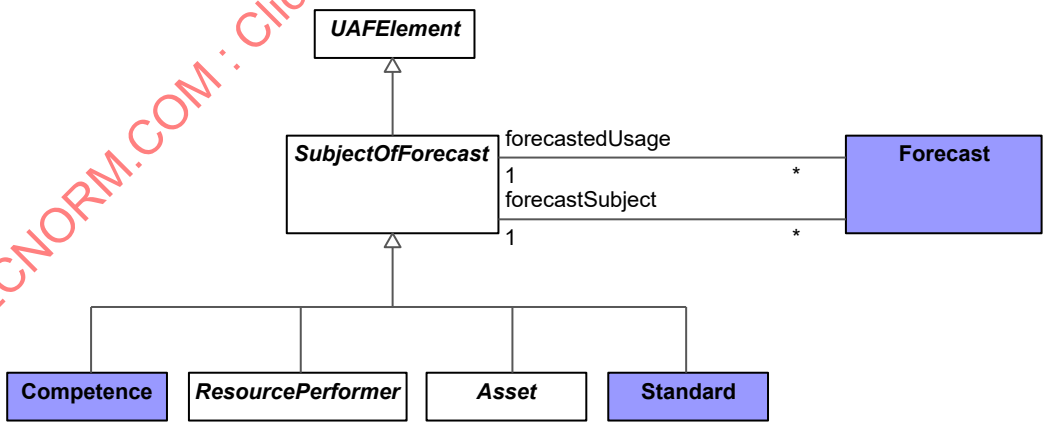

Figure 9:136 - SubjectOfForecast

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.

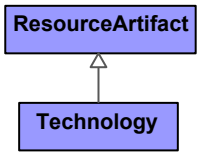

Figure 9:137 - Technology

VersionedElement

Package: Roadmap

isAbstract: Yes

Generalization: UAFElement

Description

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

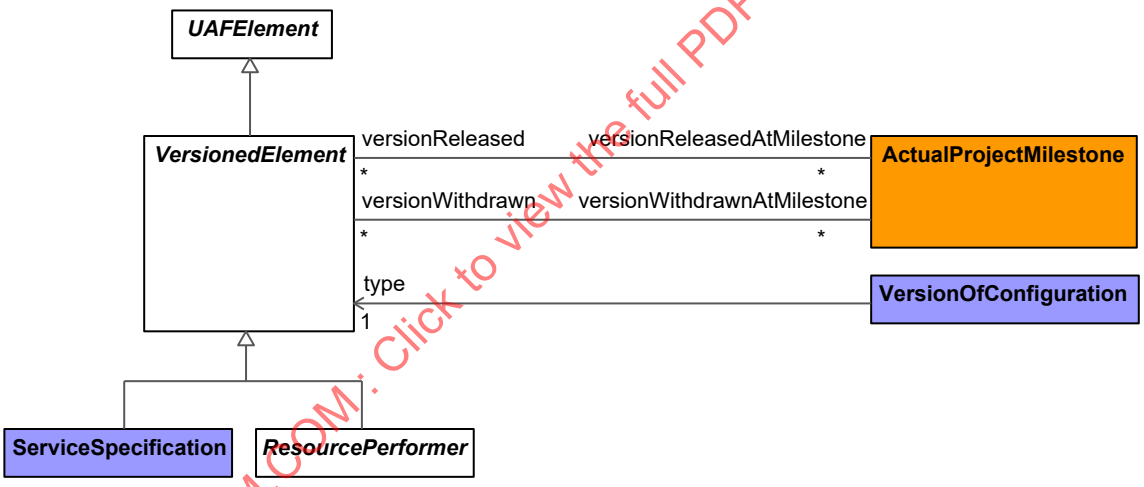

Figure 9:138 - VersionedElement

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.

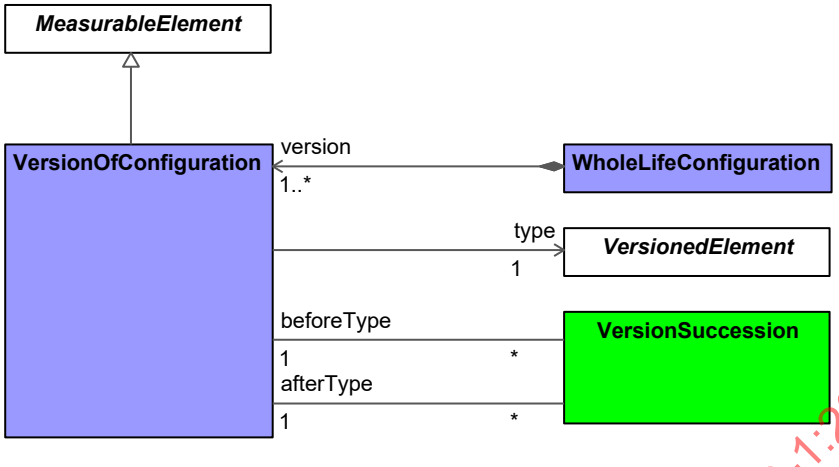

Figure 9:139 - VersionOfConfiguration

VersionSuccession

Package: Roadmap

isAbstract: No

Generalization: MeasurableElement

Description

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

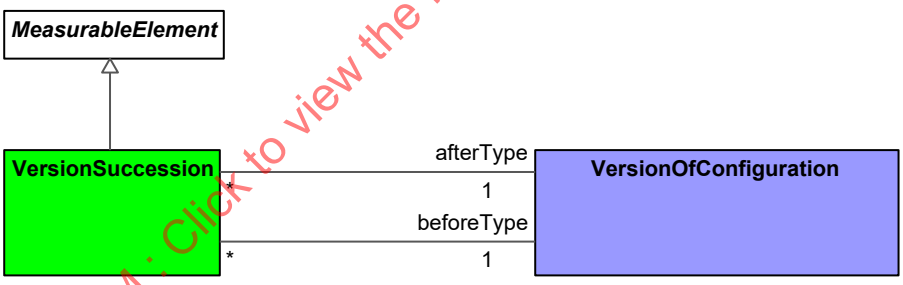

Figure 9:140 - VersionSuccession

WholeLifeConfiguration

Package: Roadmap

isAbstract: No

Generalization: PropertySet

Description

A set of VersionedElements.

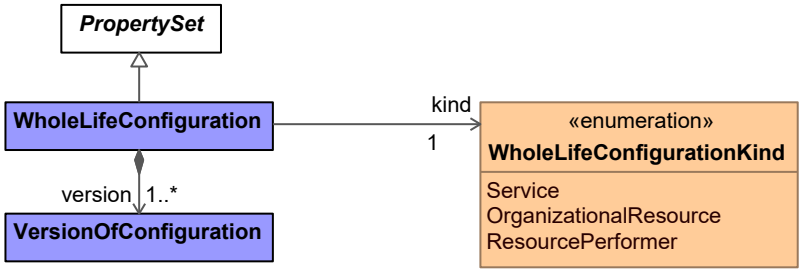

Figure 9:141 - WholeLifeConfiguration

Domain MetaModel::Resources::Traceability

ProtocolImplementation

Package: Traceability

isAbstract: Yes

Generalization: UAFElement

Description

An abstract type grouping architectural elements that can implement Protocols.

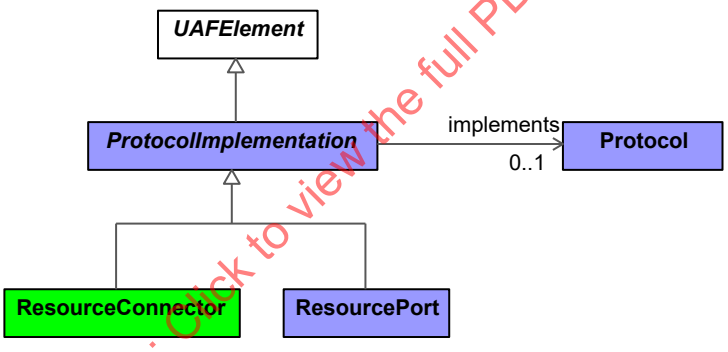

Figure 9:142 - ProtocolImplementation

9.1.7 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.

Domain MetaModel::Security::Taxonomy

Asset

Package: Taxonomy

isAbstract: Yes

Generalization: SubjectOfForecast, ConceptItem, LocationHolder, PropertySet, SubjectOfSecurityConstraint

Description

Asset as applied to Security views, an abstract type that indicates the types of elements that can be considered as a subject for security analysis.

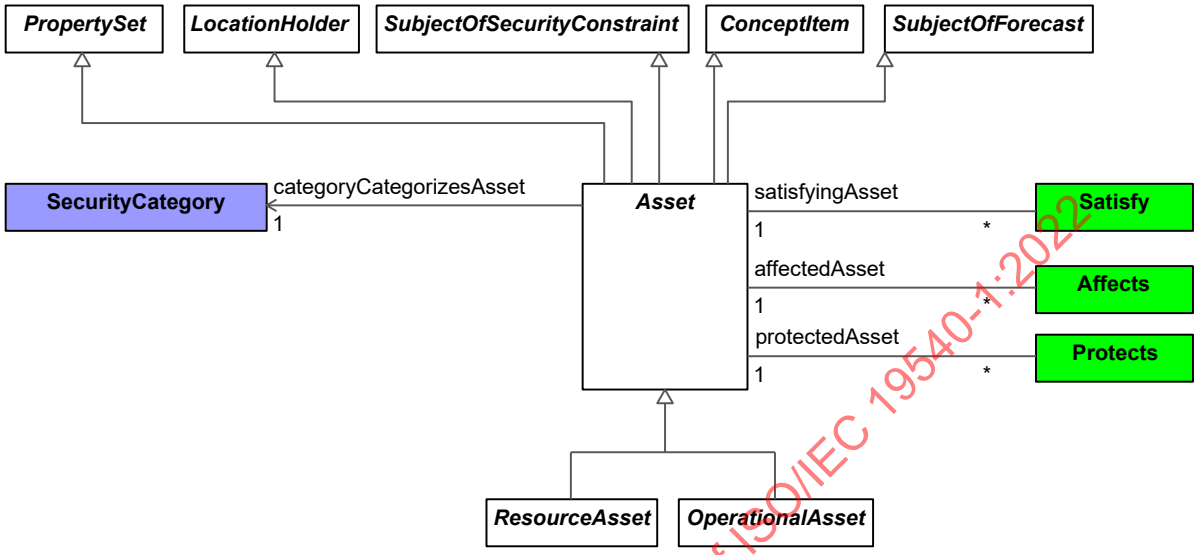

Figure 9:143 - Asset

OperationalAsset

Package: Taxonomy

isAbstract: Yes

Generalization: Asset

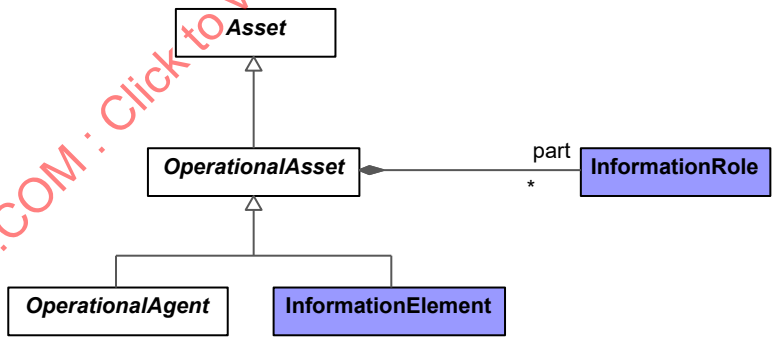

Figure 9:144 - OperationalAsset

OperationalMitigation

Package: Taxonomy

isAbstract: No

Generalization: OperationalArchitecture

Description

A set of OperationalPerformers intended to address against specific operational risks.

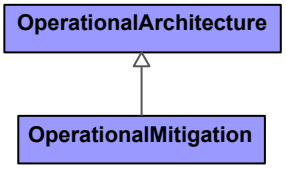

Figure 9:145 - OperationalMitigation

ResourceAsset

Package: Taxonomy
isAbstract: Yes
Generalization: Asset

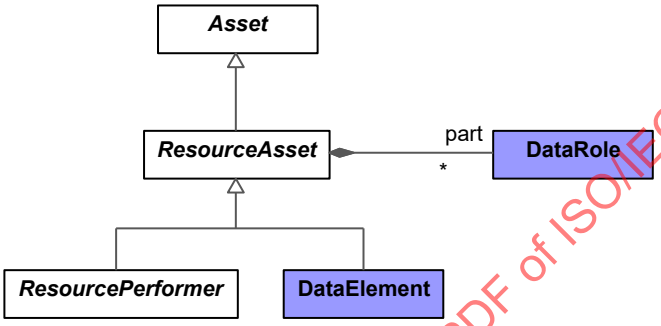

Figure 9:146 - ResourceAsset

ResourceMitigation

Package: Taxonomy
isAbstract: No
Generalization: ResourceArchitecture

Description

A set of ResourcePerformers intended to address against specific risks.

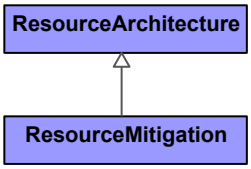

Figure 9:147 - ResourceMitigation

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.