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

**Part 2:
Unified Architecture Framework
Profile (UAFP)**

iTeh STANDARDS REVIEW
(standards.itih.ai)

ISO/IEC 19540-2:2022

<https://standards.itih.ai/catalog/standards/sist/e770e245-ae45-4055-bd58-0e12a20ab0a4/iso-iec-19540-2-2022>



iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC 19540-2:2022

<https://standards.iteh.ai/catalog/standards/sist/e770e245-ae45-4055-bd58-0e12a20ab0a4/iso-iec-19540-2-2022>



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2022

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

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	X
PREFACE	XI
1 SCOPE	1
1.1 INTRODUCTION	1
1.1.1 Overview	1
2 ADDITIONAL INFORMATION	1
2.1 LANGUAGE ARCHITECTURE	1
2.2 CORE PRINCIPLES	1
2.3 REPRESENTING STEREOTYPE CONSTRAINTS	2
2.3.1 <i>Metaconstraint dependency</i>	2
2.3.2 <i>Metarelationship dependency</i>	3
2.3.3 <i>Stereotyped relationship dependency</i>	4
3 UAF STEREOTYPES	5
3.1 UAF	5
3.1.1 <i>UAF::Dictionary</i>	5
3.1.2 <i>UAF::Parameters</i>	6
3.1.3 <i>UAF::Metadata</i>	19
3.1.4 <i>UAF::Strategic</i>	31
3.1.5 <i>UAF::Operational</i>	43
3.1.6 <i>UAF::Services</i>	61
3.1.7 <i>UAF::Personnel</i>	69
3.1.8 <i>UAF::Resources</i>	76
3.1.9 <i>UAF::Security</i>	99
3.1.10 <i>UAF::Projects</i>	113
3.1.11 <i>UAF::Standards</i>	122
3.1.12 <i>UAF::Actual Resources</i>	125
3.1.13 <i>UAF::Summary and Overview</i>	134
4 UAF VIEW SPECIFICATIONS	141
4.1 VIEW SPECIFICATIONS	141
4.1.1 <i>View Specifications::Strategic</i>	141
4.1.2 <i>View Specifications::Operational</i>	147
4.1.3 <i>View Specifications::Services</i>	155
4.1.4 <i>View Specifications::Personnel</i>	162
4.1.5 <i>View Specifications::Resources</i>	174
4.1.6 <i>View Specifications::Security</i>	182
4.1.7 <i>View Specifications::Projects</i>	189
4.1.8 <i>View Specifications::Standards</i>	194
4.1.9 <i>View Specifications::Actual Resources</i>	197
4.1.10 <i>View Specifications::Dictionary</i>	199
4.1.11 <i>View Specifications::Requirements</i>	200
4.1.12 <i>View Specifications::Summary & Overview</i>	201
4.1.13 <i>View Specifications::Information</i>	202
4.1.14 <i>View Specifications::Parameters</i>	203
5 MEASUREMENT LIBRARY	207

TABLE OF FIGURES

Figure 2:1 – MapsToCapability Stereotype.....	2
Figure 2:2 – Connector Extension.....	2
Figure 2:3 – Capabilities Generalization.....	3
Figure 2:4 – Visualizing «metarelationship».....	3
Figure 2:5 – Use of the AchievedEffect «stereotyped relationship» dependency.....	4
Figure 3:1 - Alias.....	5
Figure 3:2 - Definition.....	6
Figure 3:3 - SameAs.....	6
Figure 3:4 - ActualCondition.....	7
Figure 3:5 - ActualEnvironment.....	7
Figure 3:6 - ActualLocation.....	8
Figure 3:7 - ActualMeasurement.....	9
Figure 3:8 - ActualMeasurementSet.....	10
Figure 3:9 - ActualPropertySet.....	10
Figure 3:10 - Condition.....	11
Figure 3:11 - Environment.....	11
Figure 3:12 - EnvironmentProperty.....	12
Figure 3:13 - GeoPoliticalExtentType.....	13
Figure 3:14 - Location.....	14
Figure 3:15 - LocationHolder.....	15
Figure 3:16 - MeasurableElement.....	17
Figure 3:17 - Measurement.....	18
Figure 3:18 - MeasurementSet.....	18
Figure 3:19 - PropertySet.....	19
Figure 3:20 - ActualState.....	20
Figure 3:21 - ISO8601DateTime.....	20
Figure 3:22 - Exchange.....	21
Figure 3:23 - Resource.....	21
Figure 3:24 - Activity.....	22
Figure 3:25 - CapableElement.....	22
Figure 3:26 - IsCapableToPerform.....	23
Figure 3:27 - PerformsInContext.....	24
Figure 3:28 - ArchitectureMetadata.....	25
Figure 3:29 - DataModel.....	25
Figure 3:30 - Information.....	26
Figure 3:31 - Metadata.....	27
Figure 3:32 - Rule.....	28
Figure 3:33 - ArchitecturalReference.....	29
Figure 3:34 - Implements.....	30
Figure 3:35 - ActualEnterprisePhase.....	32
Figure 3:36 - Capability.....	33
Figure 3:37 - EnterpriseGoal.....	33
Figure 3:38 - EnterprisePhase.....	34
Figure 3:39 - EnterpriseVision.....	35
Figure 3:40 - VisionStatement.....	35
Figure 3:41 - WholeLifeEnterprise.....	36
Figure 3:42 - CapabilityRole.....	36
Figure 3:43 - StructuralPart.....	37
Figure 3:44 - TemporalPart.....	37

Figure 3:45 - ActualEnduringTask.....	38
Figure 3:46 - CapabilityForTask.....	38
Figure 3:47 - EnduringTask.....	39
Figure 3:48 - AchievedEffect.....	39
Figure 3:49 - Achiever.....	40
Figure 3:50 - DesiredEffect.....	40
Figure 3:51 - Desirer.....	41
Figure 3:52 - Exhibits.....	41
Figure 3:53 - MapsToCapability.....	42
Figure 3:54 - OrganizationInEnterprise.....	42
Figure 3:55 - ArbitraryConnector.....	43
Figure 3:56 - ConceptItem.....	44
Figure 3:57 - ConceptRole.....	44
Figure 3:58 - HighLevelOperationalConcept.....	45
Figure 3:59 - KnownResource.....	45
Figure 3:60 - OperationalAgent.....	46
Figure 3:61 - OperationalArchitecture.....	47
Figure 3:62 - OperationalMethod.....	47
Figure 3:63 - OperationalParameter.....	48
Figure 3:64 - OperationalPerformer.....	48
Figure 3:65 - OperationalPort.....	49
Figure 3:66 - OperationalRole.....	50
Figure 3:67 - ProblemDomain.....	50
Figure 3:68 - OperationalConnector.....	51
Figure 3:69 - OperationalExchange.....	52
Figure 3:70 - OperationalExchangeItem.....	53
Figure 3:71 - OperationalInterface.....	54
Figure 3:72 - OperationalSignal.....	54
Figure 3:73 - OperationalSignalProperty.....	55
Figure 3:74 - OperationalActivity.....	55
Figure 3:75 - OperationalActivityAction.....	56
Figure 3:76 - OperationalActivityEdge.....	56
Figure 3:77 - OperationalControlFlow.....	57
Figure 3:78 - OperationalObjectFlow.....	57
Figure 3:79 - StandardOperationalActivity.....	58
Figure 3:80 - OperationalStateDescription.....	58
Figure 3:81 - OperationalMessage.....	59
Figure 3:82 - InformationElement.....	59
Figure 3:83 - OperationalConstraint.....	60
Figure 3:84 - SubjectOfOperationalConstraint.....	60
Figure 3:85 - ServiceSpecification.....	61
Figure 3:86 - ServiceMethod.....	62
Figure 3:87 - ServiceParameter.....	62
Figure 3:88 - ServicePort.....	63
Figure 3:89 - ServiceSpecificationRole.....	64
Figure 3:90 - ServiceConnector.....	65
Figure 3:91 - ServiceInterface.....	65
Figure 3:92 - ServiceFunction.....	66
Figure 3:93 - ServiceFunctionAction.....	66
Figure 3:94 - ServiceStateDescription.....	67

Figure 3:95 - ServiceMessage	67
Figure 3:96 - ServicePolicy	68
Figure 3:97 - Consumes.....	68
Figure 3:98 - Organization	69
Figure 3:99 - OrganizationalResource.....	70
Figure 3:100 - Person	70
Figure 3:101 - Post	71
Figure 3:102 - Responsibility	71
Figure 3:103 - Command.....	72
Figure 3:104 - Control.....	72
Figure 3:105 - CompetenceToConduct	73
Figure 3:106 - Competence	74
Figure 3:107 - CompetenceForRole	74
Figure 3:108 - RequiresCompetence	75
Figure 3:109 - ResponsibleFor	76
Figure 3:110 - CapabilityConfiguration	77
Figure 3:111 - NaturalResource	77
Figure 3:112 - PhysicalResource.....	78
Figure 3:113 - ResourceArchitecture	78
Figure 3:114 - ResourceArtifact.....	79
Figure 3:115 - ResourcePerformer	79
Figure 3:116 - Software.....	80
Figure 3:117 - System.....	80
Figure 3:118 - ResourceMethod.....	81
Figure 3:119 - ResourceParameter	81
Figure 3:120 - ResourcePort.....	82
Figure 3:121 - ResourceRole.....	83
Figure 3:122 - ResourceConnector.....	85
Figure 3:123 - ResourceExchange.....	86
Figure 3:124 - ResourceExchangeItem	87
Figure 3:125 - ResourceInterface	88
Figure 3:126 - ResourceSignal	88
Figure 3:127 - ResourceSignalProperty	89
Figure 3:128 - Function	89
Figure 3:129 - FunctionAction	90
Figure 3:130 - FunctionControlFlow.....	91
Figure 3:131 - FunctionEdge.....	91
Figure 3:132 - FunctionObjectFlow	92
Figure 3:133 - ResourceStateDescription.....	92
Figure 3:134 - ResourceMessage	93
Figure 3:135 - DataElement	93
Figure 3:136 - ResourceConstraint.....	94
Figure 3:137 - SubjectOfResourceConstraint.....	94
Figure 3:138 - Forecast.....	95
Figure 3:139 - SubjectOfForecast.....	95
Figure 3:140 - Technology	96
Figure 3:141 - VersionedElement.....	96
Figure 3:142 - VersionOfConfiguration	97
Figure 3:143 - VersionSuccession.....	97
Figure 3:144 - WholeLifeConfiguration.....	98

Figure 3:145 - ProtocolImplementation	99
Figure 3:146 - Asset	100
Figure 3:147 - OperationalAsset	100
Figure 3:148 - OperationalMitigation	101
Figure 3:149 - ResourceAsset	101
Figure 3:150 - ResourceMitigation	102
Figure 3:151 - SecurityEnclave	102
Figure 3:152 - AssetRole	103
Figure 3:153 - DataRole	103
Figure 3:154 - InformationRole	104
Figure 3:155 - EnhancedSecurityControl	104
Figure 3:156 - Enhances	105
Figure 3:157 - Protects	105
Figure 3:158 - ProtectsInContext	106
Figure 3:159 - SecurityProcess	106
Figure 3:160 - SecurityProcessAction	107
Figure 3:161 - ActualRisk	107
Figure 3:162 - Risk	108
Figure 3:163 - SecurityConstraint	108
Figure 3:164 - SecurityControl	109
Figure 3:165 - SecurityControlFamily	110
Figure 3:166 - SubjectOfSecurityConstraint	110
Figure 3:167 - Affects	111
Figure 3:168 - AffectsInContext	111
Figure 3:169 - Mitigates	112
Figure 3:170 - OwnsRisk	112
Figure 3:171 - OwnsRiskInContext	113
Figure 3:172 - Project	114
Figure 3:173 - ProjectMilestone	115
Figure 3:174 - ProjectMilestoneRole	115
Figure 3:175 - ProjectRole	116
Figure 3:176 - ProjectStatus	116
Figure 3:177 - ProjectTheme	117
Figure 3:178 - StatusIndicators	117
Figure 3:179 - MilestoneDependency	118
Figure 3:180 - ProjectSequence	118
Figure 3:181 - ProjectActivity	119
Figure 3:182 - ProjectActivityAction	119
Figure 3:183 - ActualProject	120
Figure 3:184 - ActualProjectMilestone	121
Figure 3:185 - ActualProjectMilestoneRole	121
Figure 3:186 - ActualProjectRole	122
Figure 3:187 - Protocol	123
Figure 3:188 - ProtocolStack	123
Figure 3:189 - Standard	124
Figure 3:190 - ProtocolLayer	124
Figure 3:191 - ActualOrganization	125
Figure 3:192 - ActualOrganizationalResource	126
Figure 3:193 - ActualPerson	126
Figure 3:194 - ActualPost	127

Figure 3:195 - ActualResource.....	127
Figure 3:196 - ActualResponsibility.....	128
Figure 3:197 - ActualResponsibleResource	129
Figure 3:198 - FieldedCapability.....	129
Figure 3:199 - ActualOrganizationRole	130
Figure 3:200 - ActualResourceRole	130
Figure 3:201 - ActualResourceRelationship.....	131
Figure 3:202 - FillsPost	131
Figure 3:203 - ActualService.....	132
Figure 3:204 - ProvidedServiceLevel.....	133
Figure 3:205 - ProvidesCompetence	133
Figure 3:206 - RequiredServiceLevel.....	133
Figure 3:207 - OwnsProcess.....	134
Figure 3:208 - ArchitecturalDescription.....	135
Figure 3:209 - Architecture	136
Figure 3:210 - Concern.....	136
Figure 3:211 - Stakeholder	137
Figure 3:212 - UAFEElement	138
Figure 3:213 - View.....	139
Figure 3:214 - Viewpoint	139
Figure 4:1 - Strategic Taxonomy.....	141
Figure 4:2 - Strategic Structure	142
Figure 4:3 - Strategic Connectivity	142
Figure 4:4 - Strategic States.....	143
Figure 4:5 - Strategic Constraints.....	144
Figure 4:6 - Strategic Roadmap: Deployment.....	145
Figure 4:7 - Strategic Roadmap: Phasing	146
Figure 4:8 - Strategic Traceability.....	147
Figure 4:9 - Operational Taxonomy	148
Figure 4:10 - Operational Structure.....	149
Figure 4:11 - Operational Connectivity.....	150
Figure 4:12 - Operational Processes.....	151
Figure 4:13 - Operational States	152
Figure 4:14 - Operational Interaction Scenarios.....	153
Figure 4:15 - Operational Constraints	154
Figure 4:16 - Operational Traceability	155
Figure 4:17 - Services Taxonomy	156
Figure 4:18 - Services Structure	157
Figure 4:19 - Services Connectivity	158
Figure 4:20 - Services Processes	159
Figure 4:21 - Services States	159
Figure 4:22 - Services Interaction Scenarios.....	160
Figure 4:23 - Services Constraints.....	160
Figure 4:24 - Services Roadmap	161
Figure 4:25 - Services Traceability	162
Figure 4:26 - Personnel Taxonomy	162
Figure 4:27 - Personnel Sructure	163
Figure 4:28 - Personnel Connectivity.....	164
Figure 4:29 - Personnel Processes.....	165
Figure 4:30 - Personnel States	166

Figure 4:31 - Personnel Interaction Scenarios	167
Figure 4:32 - Personnel Constraints: Competence	168
Figure 4:33 - Personnel Constraints: Drivers	169
Figure 4:34 - Personnel Constraints: Performance	170
Figure 4:35 - Personnel Roadmap: Availability	171
Figure 4:36 - Personnel Roadmap: Evolution	172
Figure 4:37 - Personnel Roadmap: Forecast	173
Figure 4:38 - Personnel Traceability	173
Figure 4:39 - Resources Taxonomy	174
Figure 4:40 - Resources Structure	175
Figure 4:41 - Resources Connectivity	176
Figure 4:42 - Resources Processes	177
Figure 4:43 - Resources States	178
Figure 4:44 - Resources Interaction Scenarios	178
Figure 4:45 - Resources Constraints	179
Figure 4:46 - Resources Roadmap: Evolution	180
Figure 4:47 - Resources Roadmap: Forecast	181
Figure 4:48 - Resources Traceability	182
Figure 4:49 - Security Taxonomy	183
Figure 4:50 - Security Structure	184
Figure 4:51 - Security Connectivity	185
Figure 4:52 - Security Processes	186
Figure 4:53 - Security Constraints	187
Figure 4:54 - Security Traceability	188
Figure 4:55 - Project Taxonomy	189
Figure 4:56 - Project Structure	190
Figure 4:57 - Project Connectivity	191
Figure 4:58 - Project Processes	192
Figure 4:59 - Project Roadmap	193
Figure 4:60 - Project Traceability	194
Figure 4:61 - Standards Taxonomy	195
Figure 4:62 - Standards Structure	196
Figure 4:63 - Standards Roadmap	196
Figure 4:64 - Standards Traceability	197
Figure 4:65 - Actual Resources Structure	198
Figure 4:66 - Actual Resources Connectivity	199
Figure 4:67 - Actual Resources Traceability	199
Figure 4:68 - Dictionary	200
Figure 4:69 - Requirements	201
Figure 4:70 - Summary & Overview	201
Figure 4:71 - Information Model	202
Figure 4:72 - Parameters: Environment	203
Figure 4:73 - Parameters: Measurements	204

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^[1]

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 2: Unified Architecture Framework Profile (UAFP)

1 Scope

1.1 Introduction

1.1.1 Overview

This document is a normative supplement to the UAF DMM document (c4i/19-06-16).

This document specifies a UAF profile to enable practitioners to express architectural model elements and organize them in a set of domains, model kinds, and view specification (specified in the UAF DMM) that support the specific needs of end users in defense and commercial industry.

UAFP 1.1 defines a set of stereotypes and model elements and relationships to satisfy the requirements of the UPDM 3.0 RFP and the UAF DMM. The profile specification documents the language architecture in terms of UML profiling mechanism.

A number of UAFP stereotypes inherit from SysML stereotypes where reuse of SysML semantics is necessary. The reusable portions of the SysML specification are not included directly in the specification but are made explicit through the stereotype inheritance.

2 Additional Information

2.1 Language Architecture

The UAFP 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 document. This document describes the language architecture in terms of the UML 2.5.1 and SysML 1.5 parts that are reused and the defined UML 2.5.1 extensions; and specifies how to implement UAFP. This clause explains design principles and how they are applied to define the UAFP language architecture.

2.2 Core Principles

The fundamental design principles for UAFP are:

- **Requirements-driven:** UAFP is intended to satisfy the requirements of the UPDM 3.0 RFP Mandatory Requirements.
- **UAF Domain Metamodel (DMM) driven:** The DMM served as a foundation for profile development.
- **Reuse of existing specifications:** UAFP reuses UML/SysML wherever practical to satisfy the requirements of the UAFP 3.0 RFP and leverage features from both UML and SysML to provide a robust modeling capability. Consequently, UAFP is intended to be relatively easy to implement for vendors who support UML 2.x and SysML 1.x.
- **Compliance levels:** UAFP has a single compliance level based upon a combination of the reuse of UML and SysML elements. It is expected that the views that are created as result of this profile have frames that reflect the underlying SysML diagram type that is used as the basis for the view. It also expected that the graphical notation used to display elements within those views correspond to the standard SysML graphical notation of the SysML/UML metaclass that the stereotype extends.
- **Interoperability:** UAFP inherits the XMI interchange capability from UML. The UAFP specification reuses a subset of UML 2.5.1 and provides additional extensions needed to address requirements in the UPDM 3.0 RFP Mandatory Requirements.

2.3 Representing Stereotype Constraints

The UAF Profile uses an enhanced standard notation to represent metaconstraints graphically in the UAF profile diagrams to improve readability of the UAF Profile specification and overcome limitations of being unable to visualize constraints diagrammatically in UML.

The metaconstraints appears in the UAFP specification diagrams for visualization purposes only, however the representation in the XMI is as a UML constraint, specified in structured English. These constraints are implementable in a tool, by OCL for example.

A simple UML profile defines these metaconstraints.

The following subsections detail the metaconstraint profile definition within the UAF profile.

2.3.1 Metaconstraint dependency

«metaconstraint» is a stereotype that extends the Dependency metaclass. It is used to specify constrained elements within the profile.

A sample of the «metaconstraint» dependency is a diagram for stereotype extending the Dependency metaclass.

MapsToCapability is a UAFP stereotype that extends Abstraction (a type of Dependency in UML). The constraint on this stereotype is that its client end must be stereotyped by an Activity (which is abstract) and its supplier end must be stereotyped by a Capability. But as it is not possible to show this constraint graphically the diagram does not communicate the needed information. We then use the "metaconstraint" dependency to visualize the constraint.

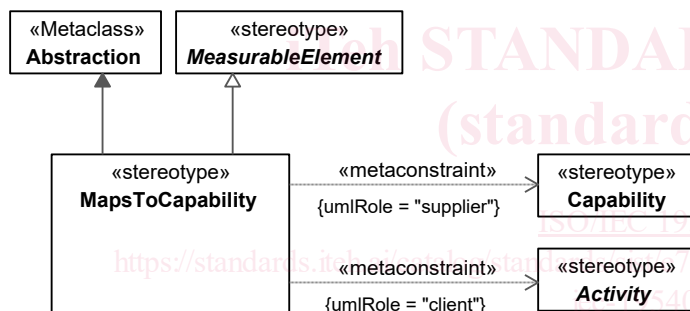


Figure 2:1 – MapsToCapability Stereotype

With the metaconstraint dependency added to the diagram (see Figure) which shows that MapsToCapability is a stereotype extending the Abstraction metaclass, that inherits the properties of a MeasurableElement and is used for modeling a relationship between an Activity (or its specializations) and a Capability (or its specializations). A Dependency stereotyped MapsToCapability must have its values for the client property stereotyped as an Activity, and its values for the supplier property must be stereotyped Capability.

Note – When stereotype extends Connector, the stereotype property umlRole has values "end[0].role" and "end[1].role." For example:

This is done because Connector has no direct "linkage" to the connected element; it links to the Connector Ends, which references the linked element. So, end[n] gives the reference to the ConnectorEnd, and role gives the reference to the linked element.

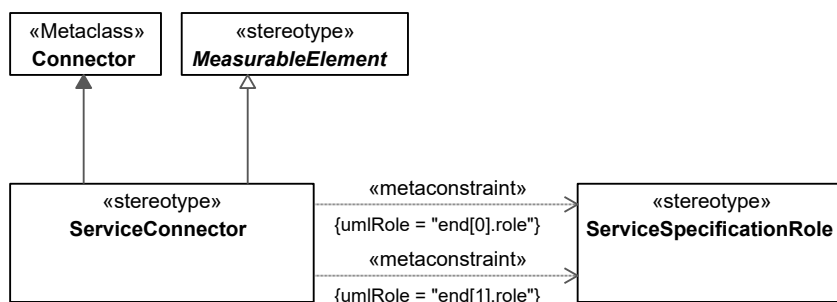


Figure 2:1 – Connector Extension

2.3.2 Metarelationship dependency

«metarelationship» is a stereotype for dependency, showing that certain domain concepts will be implemented using regular UML relationships.

For example: A Capability may depend on other Capabilities or be subtype of a Capability, but this concept cannot be visualized on the diagram:

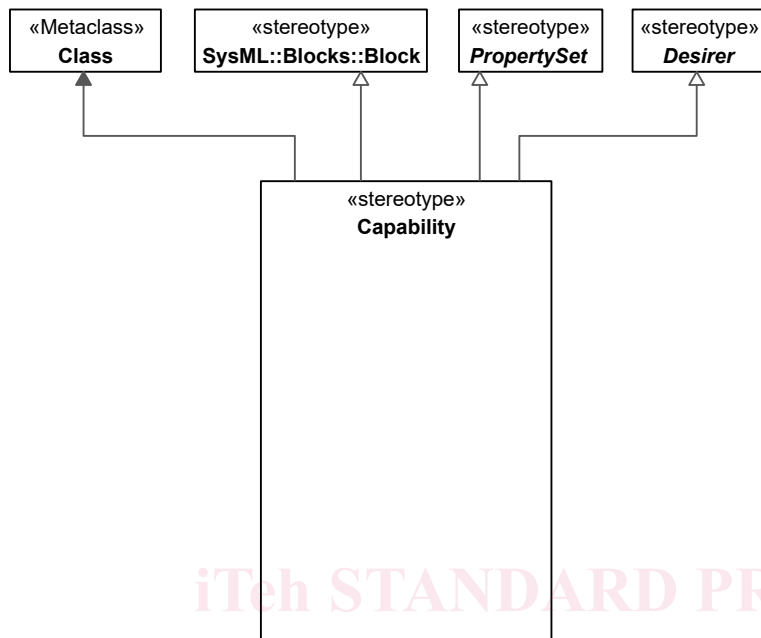


Figure 2.3 – Capabilities Generalization

We are using the «metarelationship» dependency to visualize the dependency and the generalization concept.

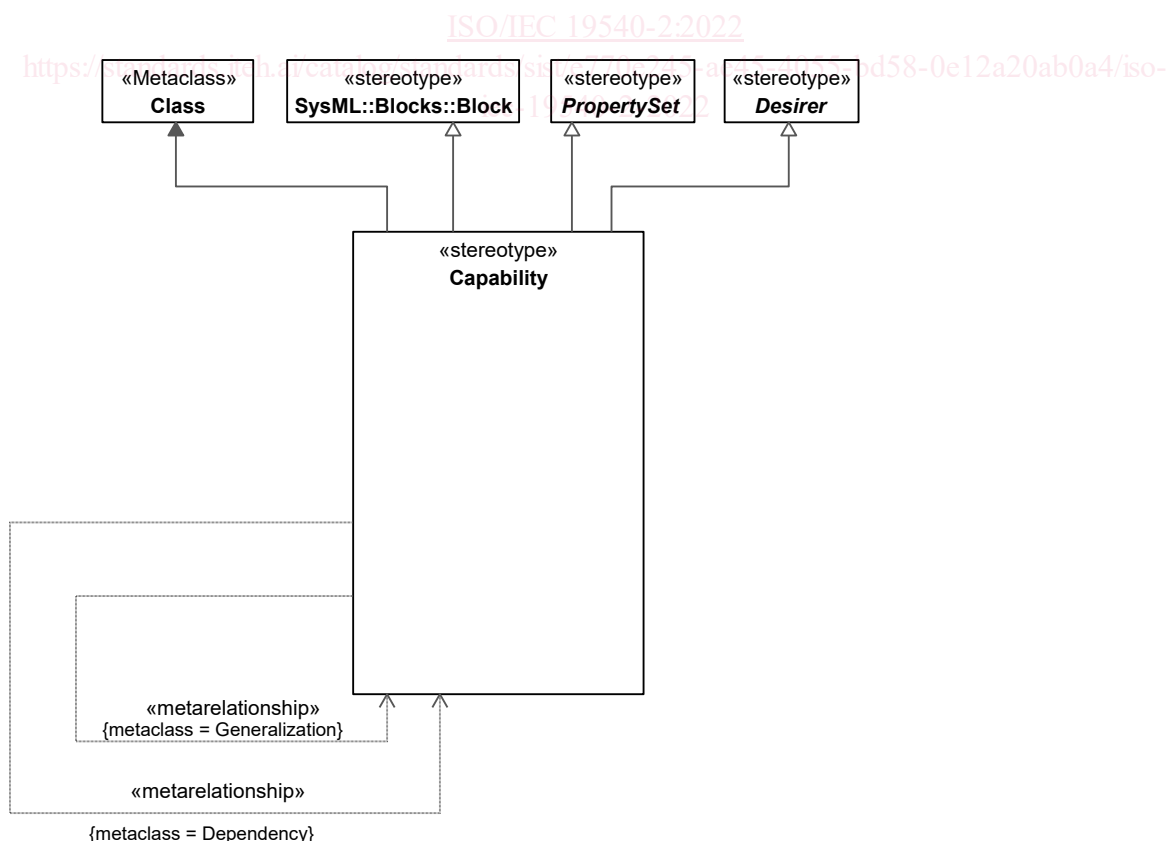


Figure 2.4 – Visualizing «metarelationship»