



Integrated Collaboration Object Model (ICOM) for Interoperable Collaboration Services Version 1.0

Committee Specification Draft 03

28 March 2012

Specification URIs

This version:

<http://docs.oasis-open.org/icom/icom-ics/v1.0/csd03/icom-ics-v1.0-csd03.doc> (Authoritative)
<http://docs.oasis-open.org/icom/icom-ics/v1.0/csd03/icom-ics-v1.0-csd03.html>
<http://docs.oasis-open.org/icom/icom-ics/v1.0/csd03/icom-ics-v1.0-csd03.pdf>

Previous version:

<http://www.oasis-open.org/committees/download.php/44405/icom-ics-v1.0-csprd02.zip>

Latest version:

<http://docs.oasis-open.org/icom/icom-ics/v1.0/icom-ics-v1.0.doc> (Authoritative)
<http://docs.oasis-open.org/icom/icom-ics/v1.0/icom-ics-v1.0.html>
<http://docs.oasis-open.org/icom/icom-ics/v1.0/icom-ics-v1.0.pdf>

Technical Committee:

OASIS Integrated Collaboration Object Model for Interoperable Collaboration Services (ICOM) TC

Chair:

Eric S. Chan (eric.s.chan@oracle.com), Oracle

Editors:

Eric S. Chan (eric.s.chan@oracle.com), Oracle
Patrick Durusau, (patrick@durusau.net), Individual

Additional artifacts:

This prose specification is one component of a Work Product which also includes:

- XML schemas: <http://docs.oasis-open.org/icom/icom-ics/v1.0/csd03/schemas/>

Declared XML namespaces:

<http://docs.oasis-open.org/ns/icom/core/201008>
<http://docs.oasis-open.org/ns/icom/accesscontrol/201008>
<http://docs.oasis-open.org/ns/icom/metadata/201008>
<http://docs.oasis-open.org/ns/icom/content/201008>
<http://docs.oasis-open.org/ns/icom/document/201008>
<http://docs.oasis-open.org/ns/icom/message/201008>
<http://docs.oasis-open.org/ns/icom/presence/201008>
<http://docs.oasis-open.org/ns/icom/contact/201008>
<http://docs.oasis-open.org/ns/icom/calendar/201008>
<http://docs.oasis-open.org/ns/icom/task/201008>
<http://docs.oasis-open.org/ns/icom/forum/201008>
<http://docs.oasis-open.org/ns/icom/conference/201008>

Abstract:

The Integrated Collaboration Object Model (ICOM) for Interoperable Collaboration Services defines a framework for integrating a broad range of domain models for collaboration activities in an integrated and interoperable collaboration environment.

The framework is not intended to prescribe how applications or services conforming to its model implement, store, or transport the data for objects. It is intended as a basis for integrating a broad range of collaboration objects to enable seamless transitions across collaboration activities. This enables applications to maintain a complete thread of conversations across multiple collaboration activities.

The model integrates a broad range of collaboration activities, by encompassing and improving on a range of models which are part of existing standards and technologies. The model is modular to allow extensibility. The core concepts, metadata concepts, and their relations are included in the Core, while the specific concepts and relations for each area of collaboration activities are defined in separate extension modules.

Status:

This document was last revised or approved by the OASIS Integrated Collaboration Object Model for Interoperable Collaboration Services (ICOM) TC on the above date. The level of approval is also listed above. Check the "Latest version" location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using "Send A Comment" button on the Technical Committee's web page at <http://www.oasis-open.org/committees/icom/>.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (<http://www.oasis-open.org/committees/icom/ipr.php>).

Citation format:

When referencing this specification the following citation format should be used:

[ICOM-ics-v1.0]

Integrated Collaboration Object Model (ICOM) for Interoperable Collaboration Services Version 1.0. 28 March 2012. OASIS Committee Specification Draft 03. <http://docs.oasis-open.org/icom/icom-ics/v1.0/csd03/icom-ics-v1.0-csd03.html>.

Notices

Copyright © OASIS Open 2012. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full [Policy](#) may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is a trademark of [OASIS](#), the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for above guidance.

Table of Contents

1	Introduction	11
1.1	Terminology	12
1.2	Normative References	12
1.3	Non-Normative References	12
2	Modeling Language	14
2.1	Introduction	14
2.2	Class Definition Grammar	14
2.3	Property Definition Grammar	16
2.4	Namespaces	19
3	Core Model	20
3.1	Main Branch	20
3.1.1	Entity and Top-Level Subclasses	20
3.1.2	Identifiable	20
3.1.3	Parental	21
3.1.4	Extent	22
3.1.5	Entity	23
3.1.6	EntityDefinition	27
3.1.7	Overview of Scope, Subject, and Artifact Branches	28
3.2	Scope Branch	29
3.2.1	Scope and Top-Level Subclasses	29
3.2.2	Scope	30
3.2.3	Community	32
3.2.4	Space	34
3.3	Subject Branch	36
3.3.1	Subject and Top-Level Subclasses	36
3.3.2	Subject	36
3.3.3	Group	38
3.3.4	Actor	40
3.3.5	Person	42
3.3.6	Resource	46
3.3.7	ResourceType	48
3.3.8	ResourceTypeEnum	49
3.3.9	ResourceBookingRule	50
3.3.10	ResourceBookingRuleEnum	51
3.4	Artifact Branch	52
3.4.1	Artifact and Top-Level Subclasses	52
3.4.2	Item	52
3.4.3	SpaceItem	54
3.4.4	Container	54
3.4.5	FolderContainer	55
3.4.6	Artifact	56
3.4.7	Folder	59
3.4.8	HeterogeneousFolder	60

3.5 Access Control Model	61
3.5.1 Accessor	61
3.5.2 Owner	62
3.5.3 RoleDefinition	63
3.5.4 Role	64
3.5.5 Privilege	65
3.5.6 PrivilegeEnum	66
3.5.7 AccessControlList	67
3.5.8 AccessControlEntry	68
3.5.9 AccessType	69
3.5.10 AccessTypeEnum	70
3.6 Metadata Model	72
3.6.1 PropertyDefinition	72
3.6.2 Property	74
3.6.3 PropertyChoiceType	76
3.6.4 PropertyType	77
3.6.5 PropertyTypeEnum	78
3.6.6 Cardinality	79
3.6.7 CardinalityEnum	80
3.6.8 Marker and Subclasses	81
3.6.9 Marker	81
3.6.10 Category	82
3.6.11 CategoryApplication	84
3.6.12 Tag	85
3.6.13 TagApplication	87
3.6.14 RelationshipBondable	88
3.6.15 RelationshipDefinition	89
3.6.16 Relationship	91
3.7 Common Concepts	93
3.7.1 Addressable	93
3.7.2 EntityAddress	94
3.7.3 Participant	95
3.7.4 Priority	97
3.7.5 PriorityEnum	97
3.7.6 DateTimeResolution	98
3.7.7 DateTimeResolutionEnum	99
3.7.8 TimeZone	100
3.7.9 Location	101
3.7.10 GeoCoordinates	102
4 Extension Modules	105
4.1 Overview of Extension Modules	105
4.2 Content Module	107
4.2.1 MimeConvertible	107
4.2.2 Content	107
4.2.3 MultiContent	109

4.2.4 SimpleContent	110
4.2.5 OnlineContent.....	112
4.2.6 ContentDispositionType	113
4.2.7 ContentDispositionTypeEnum	114
4.2.8 Attachment	115
4.3 Document Module.....	116
4.3.1 Versionable.....	116
4.3.2 VersionControlMetadata.....	118
4.3.3 VersionSeries	119
4.3.4 Version	122
4.3.5 VersionType	124
4.3.6 VersionTypeEnum	124
4.3.7 Document	125
4.3.8 WikiPage	127
4.4 Message Module	129
4.4.1 Message	129
4.4.2 UnifiedMessage.....	130
4.4.3 UnifiedMessageParticipant.....	134
4.4.4 UnifiedMessageFlag.....	136
4.4.5 UnifiedMessageFlagEnum	136
4.4.6 UnifiedMessageDeliveryStatusNotificationRequest	137
4.4.7 UnifiedMessageDeliveryStatusNotificationRequestEnum	138
4.4.8 UnifiedMessageChannel	139
4.4.9 UnifiedMessageChannelEnum.....	140
4.4.10 UnifiedMessageEditMode	140
4.4.11 UnifiedMessageEditModeEnum	141
4.4.12 InstantMessage	142
4.4.13 InstantMessageType	145
4.4.14 InstantMessageTypeEnum.....	146
4.4.15 InstantMessageChatStatus	147
4.4.16 InstantMessageChatStatusEnum.....	147
4.4.17 InstantMessageFeed.....	148
4.4.18 InstantMessageConnection.....	150
4.5 Presence Module	153
4.5.1 Presence	153
4.5.2 PresenceEditMode	155
4.5.3 PresenceEditModeEnum.....	156
4.5.4 ContactMethod	157
4.5.5 ContactReachabilityStatus	159
4.5.6 ContactReachabilityStatusEnum	159
4.5.7 Activity	162
4.5.8 ActivityType	163
4.5.9 ActivityTypeEnum.....	164
4.6 Address Book Module.....	165
4.6.1 AddressBook	165

4.6.2 PersonContact	166
4.7 Calendar Module	171
4.7.1 Calendar	171
4.7.2 OccurrenceSeries	172
4.7.3 Occurrence	177
4.7.4 OccurrenceStatus	183
4.7.5 OccurrenceStatusEnum	183
4.7.6 OccurrenceType	184
4.7.7 OccurrenceTypeEnum	185
4.7.8 OccurrenceParticipant	186
4.7.9 OccurrenceParticipantStatus	187
4.7.10 OccurrenceParticipantStatusEnum	187
4.7.11 OccurrenceParticipantTransparency	188
4.7.12 OccurrenceParticipantTransparencyEnum	189
4.7.13 OccurrenceEditMode	190
4.7.14 OccurrenceEditModeEnum	191
4.8 Free Busy Module	191
4.8.1 FreeBusy	191
4.8.2 FreeBusyInterval	193
4.8.3 FreeBusyType	195
4.8.4 FreeBusyTypeEnum	196
4.9 Task List Module	197
4.9.1 TaskList	197
4.9.2 Task	198
4.9.3 TaskStatus	203
4.9.4 TaskStatusEnum	203
4.9.5 TaskParticipantStatus	204
4.9.6 TaskParticipantStatusEnum	205
4.9.7 TaskEditMode	206
4.9.8 TaskEditModeEnum	206
4.10 Forum Module	207
4.10.1 Discussion	207
4.10.2 DiscussionContainer	208
4.10.3 DiscussionMessage	209
4.10.4 TopicContainer	210
4.10.5 Forum	212
4.10.6 Topic	214
4.10.7 Announcement	215
4.10.8 AnnouncementStatus	217
4.10.9 AnnouncementStatusEnum	217
4.11 Conference Module	218
4.11.1 Conference	218
4.11.2 ConferenceType	221
4.11.3 ConferenceTypeEnum	222
4.11.4 ConferenceStatus	223

4.11.5 ConferenceStatusEnum	224
4.11.6 ConferenceSession	224
4.11.7 ConferenceSessionEndingReason	226
4.11.8 ConferenceSessionEndingReasonEnum	227
4.11.9 ConferenceSetting	228
4.11.10 ConferenceParticipantRole	229
5 Conformance.....	232
Appendix A. Acknowledgements	234
Appendix B. Revision History	235

Table of Figures

Figure 1: Entity and Top-Level Abstract Classes.....	20
Figure 2: Entity Class Diagram.	27
Figure 3: Scope, Subject, and Artifact Branches.	29
Figure 4: Scope Branch.	29
Figure 5: Scope Class Diagram.	32
Figure 6: Community Class Diagram.	34
Figure 7: Space Class Diagram.	35
Figure 8: Subject Branch.....	36
Figure 9: Subject Class Diagram.	38
Figure 10: Group and Actor Class Diagram.....	40
Figure 11: Person Class Diagram.....	46
Figure 12: Resource Class Diagram.....	48
Figure 13: Artifact Branch.	52
Figure 14: Artifact Class Diagram.	58
Figure 15: Heterogeneous Folder Class Diagram.	61
Figure 16: Role Definition and Role Class Diagram.	65
Figure 17: Access Control List Class Diagram.	71
Figure 18: Property Definition and Property Class Diagram.....	76
Figure 19: Marker Branch.	81
Figure 20: Marker Class Diagram.	82
Figure 21: Category and Category Application Class Diagram.	84
Figure 22: Tag and Tag Application Class Diagram.	87
Figure 23: Relationship Class Diagram.	93
Figure 24: Containers of Collaboration Activities.	105
Figure 25: Composite Content Class Diagram.	109
Figure 26: Document, Version Series, and Version Class Diagram.	127
Figure 27: Wiki Page Class Diagram.	128
Figure 28: Unified Message Class Diagram.	142
Figure 29: Instant Message Class Diagram.....	145
Figure 30: Instant Message Feed and Connection Class Diagram.	150
Figure 31: Presence Class Diagram.	155
Figure 32: Presence Contact Method and Instant Message Connection Class Diagram.	161
Figure 33: Address Book Class Diagram.....	166
Figure 34: Person Contact Class Diagram.	170
Figure 35: Calendar Class Diagram.....	172
Figure 36: Occurrence Series Class Diagram.	177
Figure 37: Occurrence Class Diagram.....	182
Figure 38: Free Busy Class Diagram.....	195
Figure 39: Task List Class Diagram.....	198
Figure 40: Task Class Diagram.	202

Figure 41: Forum Class Diagram.....	212
Figure 42: Conference Class Diagram.....	221

1 Introduction

The Integrated Collaboration Object Model (ICOM) for Interoperable Collaboration Services specification defines a framework for integrating a broad range of domain model for collaboration activities in an interoperable collaboration environment. The standard promotes an integrated user experience with seamless transitions across collaboration activities. It enables applications to support continuity of conversations across diverse collaboration activities. For example, applications can aggregate conversation threads in email with other conversations on the same topic in instant message, over the phone or via real-time conferencing, by discussion threads in community forum, weblog or micro blog, and activity stream of participants from all channels.

The specification defines a core model and a set of extension modules. The core model (Section 3) defines the classes (Section 3.1 Main Branch) that bring together the model of directory (Section 3.2 Scope Branch), identity management (Section 3.3 Subject Branch), and content management (Section 3.4 Artifact Branch) in a framework with a common access control model (Section 3.5) and metadata model (Section 3.6). The extension modules in Section 4 extend the artifact and folder model of Artifact Branch (Section 3.4) to define the specialized model for different collaboration activities. The range of collaboration model includes content sharing and co-creation, asynchronous communication, instant communication, presence awareness, moderated group discussion, time management, coordination, real-time interaction, etc.

The Subject and Artifact branches support separation of concerns for user administration and content management. Subject branch includes the model of actors, groups of actors, and role assignment of actors. Actors, groups, and roles typically appear as the subject in the (subject, privilege, object) triples of an access control model. The Artifact branch includes the model of content and metadata produced by actors. The Scope branch includes the model of communities and spaces that contain subjects and artifacts. Communities and spaces join the subjects and artifacts in a role-based access control model where a role is assigned to an actor in a specific scope. Thus Scope, Subject, and Artifact form a framework for applications to integrate and interoperate with directory, identity management, content management, and collaboration services.

The model specified in ICOM is part of existing standards and technologies, several of which are referenced in Section 1.3 Non-Normative References. The model is modular and extensible, with common concepts, metadata concepts, and their relations provided in the Core, while the specific concepts and relations for each area of collaboration activities defined in separate extension modules. ICOM core model encompasses LDAP Directory Information Models [RFC4512]. The extension modules integrate models from Content Management Interoperability Services [CMIS], Java Content Repository API [JCR 2.0], Web Distributed Authoring and Versioning (WebDAV) [RFC4918], Internet Message Access Protocol (IMAP) [RFC2119], Simple Mail Transfer Protocol (SMTP) [RFC5321], Extensible Messaging and Presence Protocol (XMPP) [RFC3920], XMPP Instant Messaging and Presence [RFC3921], vCard MIME Directory Profile [RFC2426], Internet Calendaring and Scheduling Core Object Specification (iCalendar) [RFC5545], and Calendaring Extensions to WebDAV (CalDAV) [RFC4791].

ICOM is open for extensions with additional domain models to enable seamless integration with business processes and social networks: for example in process integration domain which includes Business Process Model and Notation [BPMN], Web Services Business Process Execution Language [WS-BPEL], WS-BPEL Extension for People [BPEL4People], and Web Services for Human Task [WS-HumanTask]; in social networking domain, which includes Friend of a Friend [FOAF], Semantically-Interlinked Online Communities [SIOC], Open Social [OpenSocial], and Facebook Platform Open Graph [OpenGraph]. The OASIS ICOM TC Wiki [ICOM Wiki] provides Non-Normative supplemental information, including overview, primer, extensions, use cases, and mappings to various standard and proprietary data models.

The integrated model can be the foundation for defining the application programming interfaces (API) for application developers to develop integrated collaboration applications to interoperate with collaboration services. A service provider interface (SPI) can be specified to support interchangeable and interoperable services that conform to the ICOM application framework. ICOM does not prescribe how applications or services conforming to its model implement, store, or transport the data for objects.

52

53 1.1 Terminology

54 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD
55 NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be
56 interpreted as described in [RFC2119].

57 1.2 Normative References

- 58 [CMIS] OASIS Standard, *Content Management Interoperability Services (CMIS) Version*
59 *1.0*, May 2010. ([http://docs.oasis-open.org/cmisis/CMIS/v1.0/os/cmisis-spec-](http://docs.oasis-open.org/cmisis/CMIS/v1.0/os/cmisis-spec-v1.0.doc)
60 [v1.0.doc](http://docs.oasis-open.org/cmisis/CMIS/v1.0/os/cmisis-spec-v1.0.doc))
- 61 [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP
62 14, RFC 2119, March 1997. (<http://www.ietf.org/rfc/rfc2119.txt>)
- 63 [RFC3986] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifier
64 (URI): Generic Syntax", STD 66, RFC 3986, January 2005.
65 (<http://www.ietf.org/rfc/rfc3986.txt>)
- 66 [RFC3987] Duerst, M. and Suignard, M., "Internationalized Resource Identifiers (IRIs)", RFC
67 3987, January 2005. (<http://www.ietf.org/rfc/rfc3987.txt>)
- 68 [XML SCHEMA] Biron, P.V. and Malhotra, A., "XML Schema Part 2: Datatypes Second Edition",
69 W3C Recommendation, 28 October 2004. (<http://www.w3.org/TR/xmlschema-2/>)

70 1.3 Non-Normative References

- 71 [BPEL4People] OASIS Committee Specification, *WS-BPEL Extension for People (BPEL4People)*
72 *Specification Version 1.1*, August 2010. [http://docs.oasis-](http://docs.oasis-open.org/bpel4people/bpel4people-1.1.html)
73 [open.org/bpel4people/bpel4people-1.1.html](http://docs.oasis-open.org/bpel4people/bpel4people-1.1.html)
- 74 [BPMN] OMG, "Business Process Model and Notation (BPMN) Version 2.0", January
75 2011. (<http://www.omg.org/spec/BPMN/2.0/PDF>)
- 76 [FOAF] Brickley, D. and Miller, L., "FOAF Vocabulary Specification", August 2009.
77 (<http://xmlns.com/foaf/spec/>)
- 78 [ICOM Wiki] OASIS ICOM TC Wiki, (<http://wiki.oasis-open.org/icom>)
- 79 [JCR 2.0] Java Specification Request (JSR) 283, *Content Repository for Java™*
80 *Technology API 2.0 Specification*, August 2009.
81 (<http://jcp.org/en/jsr/detail?id=283>)
- 82 [OpenGraph] Facebook Platform Open Graph Core Concepts,
83 (<http://developers.facebook.com/docs/coreconcepts/>)
- 84 [OpenSocial] OpenSocial and Gadgets Specification Group, "Social Data Specification",
85 November 2010. ([http://opensocial-](http://opensocial-resources.googlecode.com/svn/spec/2.0/Social-Data.xml)
86 [resources.googlecode.com/svn/spec/2.0/Social-Data.xml](http://opensocial-resources.googlecode.com/svn/spec/2.0/Social-Data.xml))
- 87 [RFC2119] Crispin, M., "Internet Message Access Protocol – Version 4rev1", RFC 2060,
88 December 1996. (<http://tools.ietf.org/html/rfc2060>)
- 89 [RFC2426] Dawson, F. and Howes, T., "vCard MIME Directory Profile", RFC 2426,
90 September 1998. (<http://tools.ietf.org/html/rfc2426>)
- 91 [RFC3920] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core",
92 RFC 3920, October 2004. (<http://tools.ietf.org/html/rfc3920>)
- 93 [RFC3921] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Instant
94 Messaging and Presence", RFC 3921, October 2004.
95 (<http://tools.ietf.org/html/rfc3921>)
- 96 [RFC4512] Zeilenga, K., "Lightweight Directory Access Protocol (LDAP): Directory
97 Information Models", RFC 4512, June 2006. (<http://tools.ietf.org/html/rfc4512>)
- 98 [RFC4791] Daboo, C. and Desruisseaux, B., "Calendaring Extensions to WebDAV
99 (CalDAV)", RFC 4791, March 2007. (<http://tools.ietf.org/html/rfc4791>)

100 **[RFC4918]** Dusseault, L., "HTTP Extensions for Web Distributed Authoring and Versioning
101 (WebDAV)", RFC 4918, June 2007. (<http://tools.ietf.org/html/rfc4918>)
102 **[RFC5321]** Klensin, J., "Simple Mail Transfer Protocol, Draft Standard" RFC 5321, October
103 2008. (<http://tools.ietf.org/html/rfc5321>)
104 **[RFC5545]** Desruisseaux, B., "Internet Calendaring and Scheduling Core Object
105 Specification (iCalendar)", RFC 5545, September 2009.
106 (<http://tools.ietf.org/html/rfc5545>)
107 **[SIOC]** W3C Member Submission, "SIOC Core Ontology Specification", June 2007.
108 (<http://www.w3.org/Submission/2007/SUBM-sioc-spec-20070612/>)
109 **[WS-BPEL]** OASIS Standard, *Web Services Business Process Execution Language Version*
110 2.0, April 2007. <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html>
111 **[WS-HumanTask]** OASIS Committee Specification, *Web Services – Human Task (WS-HumanTask)*
112 *Specification Version 1.1, CS-01*, August 2010. [http://docs.oasis-](http://docs.oasis-open.org/bpel4people/ws-humantask-1.1-spec-cs-01.html)
113 [open.org/bpel4people/ws-humantask-1.1-spec-cs-01.html](http://docs.oasis-open.org/bpel4people/ws-humantask-1.1-spec-cs-01.html)

2 Modeling Language

2.1 Introduction

ICOM specifies a set of objects in a collaboration environment, in terms of class definitions and property definitions of the classes. Objects comprise the information structures in a common application framework. An ICOM information structure MAY be composed of information from multiple repositories or collaboration services.

Note: To offer closer interoperability with OASIS Content Management Interoperability Services, ICOM specification follows the class and property definitions grammar of CMIS specification [CMIS], which is a normative reference for ICOM specification. ICOM specification adapts the CMIS class and property definitions grammar to introduce mixed-in types, enumeration types, and other base types which are not part of the domain model of CMIS Version 1 specification.

Note: One objective of ICOM standard is to offer seamless interoperability among identity management, content management, and collaboration services. Scope and Subject classes, defined respectively in Section 3.2 Scope Branch and Section 3.3 Subject Branch, can represent objects in Identity Management domain (such as LDAP). Artifact classes defined in Section 3.4 Artifact Branch can represent the extensions of CMIS Folder and Document base types. The extension modules in Section 4 define specialized subclasses of artifact and folder in Artifact Branch to support collaboration activities.

Note: ICOM extends the CMIS base types in several ways. ICOM Relationship class defined in Section 3.6.16 can represent n-ary relationships whereas CMIS Relationship base type represents binary relationships. ICOM version control model defined in Section 4.3.1 adopts the CMIS version control model and extends it with the concept of representative copy.

ICOM application framework includes a core model and a set of extension modules. All objects in the framework must be instances of at least one class.

Each class is defined in the class definition grammar, which specifies a `namespace` attribute, a `localName` attribute, a `description` attribute, an `extendsFrom` attribute representing a set of zero or more super classes, a `stereotype` attribute indicating whether a class is primary or mixin, an `isAbstract` attribute indicating whether a primary class is abstract, an `isEnumeration` attribute indicating whether instances of a primary class are enumerated, and a `propertyDefinition` attribute defining a set of zero or more properties of objects of the class. The properties are defined in the property definition grammar.

Note: The class and property definitions grammar corresponds to the UML meta-model, which is an OMG Meta Object Facility (MOF) M2-model. Each of the classes and properties thus defined are faithfully depicted by UML 2.0 diagrams in this specification.

A fully expanded class name, `namespace/localName`, MUST be unique within a domain.

Note: A namespace IRI reference qualifies a local name by associating the local name with the IRI reference to derive an expanded name.

2.2 Class Definition Grammar

A **class-definition** MUST contain the following attributes:

namespace String

The `namespace` attribute specifies an IRI.

localName String

The `localName` attribute specifies a local name portion of an expanded name or qualified name.

description String (optional)

The `description` attribute describes the nature and intended use of a class.

extendsFrom IRI (multi-valued)

The `extendsFrom` attribute specifies a set of zero or more super classes.

stereotype Enum

The `stereotype` attribute specifies whether a class is a primary or mixin class.

The values of `stereotype` attribute are:

- **Primary**: A primary class is part of a single inheritance class hierarchy;
- **Mixin**: A mixin class is part of multiple inheritance class hierarchy.

A particular class is either a primary class or a mixin class, i.e. it cannot be both.

Inheritance is constrained by:

- a primary class **MUST** extend from one and only one primary class;
- a primary or mixin class **MAY** extend from zero or more mixin classes;
- a mixin class **MUST NOT** extend from a primary class.

An object **MUST** be an instance of one and only one primary class.

Note: When there is more than one super class in a class definition, at most one of the super classes is a primary class and the rest of the super classes are mixin classes. For example, `Scope` extends from `Entity`, `RelationshipBondable`, and `Extent`. `Scope` is a primary class. Among its super classes, only `Entity` is a primary class while `RelationshipBondable` and `Extent` are mixin classes.

isAbstract Boolean

The `isAbstract` attribute specifies whether a primary class is an abstract class. It is applicable only when the value of `stereotype` attribute is **Primary**.

The values of `isAbstract` attribute are:

- **TRUE** if the primary class is an abstract class;
- **FALSE** if the primary class is not an abstract class.

The default value is **FALSE**.

Note: An abstract class typically does not provide a complete declaration and cannot be instantiated. An abstract class is intended to be extended by other primary classes.

An abstract primary class **MUST NOT** extend from any non-abstract primary class.

isEnumeration Boolean

The `isEnumeration` attribute specifies whether instances of a primary class are enumerated in a class definition. It is applicable only when the value of `stereotype` attribute is **Primary**.

The values of `isEnumeration` attribute are:

- **TRUE** if the instances of a primary class are enumerated in a class definition;
- **FALSE** if the instances of a primary class are not enumerated in a class definition.

The default value is **FALSE**.

Note: A primary class which is an enumeration of instances is also known as an enum class.

instances

The `instances` attribute enumerates instances of an enum class. It is applicable only when the value of `stereotype` attribute is **Primary** and the value of `isEnumeration` attribute is **TRUE**.

propertyDefinition property-definition (multi-valued)

The `propertyDefinition` attribute defines a set of zero or more property definitions for a class.

Property definitions of a class are a union of inherited property definitions from super classes and property definitions explicitly defined on a class.

The order of property definitions within a class is not significant.

Property definitions **MUST** be uniquely named to avoid conflicts from multiple inheritances.

Note: It is possible for the same property definition to be inherited through different paths in a super class hierarchy. Duplicate property definitions are eliminated from the set of property definitions of a class.

2.3 Property Definition Grammar

A **property-definition** **MUST** contain the following attributes:

namespace String

The `namespace` attribute specifies an IRI.

localName String

The `localName` attribute specifies the local name portion of an expanded name or qualified name.

description String (optional)

The `description` attribute specifies a description of a property

propertyType Enum

The `propertyType` attribute specifies a **property-type** for property values.

The value of `propertyType` attribute is one of the **property-type** names. The **property-type** names include names for the following data type defined by XML Schema Part 2 [XML SCHEMA]:

- `string` (xsd:string)
- `boolean` (xsd:boolean)
- `decimal` (xsd:decimal)
- `integer` (xsd:integer)
- `datetime` (xsd:dateTime)
- `duration` (xsd:duration)
- `iri` (xsd:anyURI)

In addition, the following data type names are also specified by ICOM:

- `id` (an opaque string representing an object id of an identifiable object)
- `html` (a document or fragment of Hypertext Markup Language)

cardinality Enum

The `cardinality` attribute specifies a cardinality of property values.

The values of `cardinality` attribute are:

- **Single:** Property can have zero or one value (if property is not required), or exactly one value (if property is required)
- **Multi:** Property can have zero or more values (if property is not required), or one or more values (if property is required).

updatability Enum

The `updatability` attribute specifies under what circumstances the value of this property MAY be updated.

The values of `updatability` attribute are:

- **ReadOnly:** The value of this property MUST NOT be set directly by application. It is a property that is either maintained or computed by a service provider.
- **WriteOnly:** The value of this property can be set by application. It is a property whose value MAY be propagated into another **ReadOnly** property by a service provider.
- **ReadWrite:** The property value can be modified.
- **OnCreate:** The property value MUST only be update-able during the creation (a create operation) of an object.

inherited Boolean

The `inherited` attribute specifies whether a property definition is inherited from a super class.

The values of `inherited` attribute are:

- **TRUE** if a property definition is inherited from a super class;
- **FALSE** if a property definition is explicitly defined for a class.

required Boolean

The `required` attribute is only applicable to read-write and on-create properties, i.e. properties whose value is provided by application.

The values of `required` attribute are:

- **TRUE** if the value of a property MUST never be set to the “not set” state when an object of this type is created or updated. If a value is not provided during a create or update operation, a service provider MUST provide a value for the property. If a value is not provided, then a default value defined for the property MUST be set. If no default value is defined, a service provider MUST throw an exception.
- **FALSE** if the value of a property MAY be set to the “not set” state when an object of this type is created or updated.

This attribute is not applicable when the value `updatability` attribute is **ReadOnly**. In that case, `required` attribute SHOULD be set to **FALSE**.

Note: The value of a read-only property (such as `icom_core:objectId`, `icom_core:createdBy`) is set by a service provider. Hence, the value of the `required` attribute SHOULD be **FALSE** because it is read only for applications.

288 **choices** **property-choice-type** (multi-valued)

289 The `choices` attribute specifies a set of single values allowed for this property.

290 Each value of `choices` attribute is an instance of **property-choice-type** that specifies a display

291 name and a value to be stored in a property when selected.

292 If the value of `cardinality` attribute is **Single** and the value of `openChoice` attribute

293 is **FALSE**, then a property value **MUST** be at most one of the values listed in `choices`

294 attribute.

295 If the value of `cardinality` attribute is **Single** and the value of `openChoice` attribute

296 is **TRUE**, then a property value **MAY** be one of the values listed in `choices` attribute.

297 If the value of `cardinality` attribute is **Multi** and the value of `openChoice` attribute

298 is **FALSE**, then a property value **MUST** be zero, one, or more than one of the values

299 listed in `choices` attribute.

300 If the value of `cardinality` attribute is **Multi** and the value of `openChoice` attribute

301 is **TRUE**, then a property value **MAY** be zero, one, or more than one of the values listed in

302 `choices` attribute.

303 If `choices` attribute is “not set”, then a property value **MAY** be an instance of the **property-type**

304 specified by the `propertyType` attribute of a property definition.

305

306 **openChoice** Boolean

307 The `openChoice` attribute specifies whether the value of a property must be listed in `choices`

308 attribute. It is applicable only when `choices` attribute is set.

309 The values of `openChoice` attribute are:

310

- **TRUE** if a value of a property **MAY** be other than those listed in `choices` attribute;
- **FALSE** if a value of a property **MUST** be among those listed in `choices` attribute.

312

313 **defaultValue** **property-type**

314 The `defaultValue` attribute specifies a value that a service provider **MUST** set for a property if

315 a value is not provided by application when an object is created.

316 If no default value is specified and application creates an object of this class without setting a

317 value for a property of this property definition, a service provider **MUST** attempt to store a “not

318 set” state for the property value. If this occurs for a property that is defined to be required, then a

319 service provider **MUST** throw an exception.

320 The value of the `defaultValue` attribute is an instance of the **property-type** specified by the

321 `propertyType` attribute of a property definition.

322

323 **minValue** Integer | Decimal

324 The minimum value allowed for a property. It is applicable only when the `propertyType`

325 attribute of a property definition specifies the property types Integer or Decimal.

326

327 **maxValue** Integer | Decimal

328 The maximum value allowed for a property. It is applicable only when the `propertyType`

329 attribute of a property definition specifies the property types Integer or Decimal.

330

331 A **property-choice-type** **MUST** contain the following attributes:

332 **displayName** String

The `displayName` attribute specifies a string for presentation by application.

value **property-type**

The `value` attribute specifies a value compatible with the **property-type** specified by the `propertyType` attribute of a property definition.

2.4 Namespaces

Qualified names are subject to namespace interpretation depending on the namespace prefixes.

A class definition includes the two attributes: `namespace` and `localName`. The `namespace` specifies one of the namespace prefixes in Table 1. The `localName` specifies an unprefix name of a class. Syntactically, the namespace qualifies the local name.

Table 1 Namespace prefixes and IRI references.

<code>icom_core</code>	=	http://docs.oasis-open.org/ns/icom/core/201008
<code>icom_ac</code>	=	http://docs.oasis-open.org/ns/icom/accesscontrol/201008
<code>icom_meta</code>	=	http://docs.oasis-open.org/ns/icom/metadata/201008
<code>icom_content</code>	=	http://docs.oasis-open.org/ns/icom/content/201008
<code>icom_doc</code>	=	http://docs.oasis-open.org/ns/icom/document/201008
<code>icom_msg</code>	=	http://docs.oasis-open.org/ns/icom/message/201008
<code>icom_card</code>	=	http://docs.oasis-open.org/ns/icom/contact/201008
<code>icom_presence</code>	=	http://docs.oasis-open.org/ns/icom/presence/201008
<code>icom_cal</code>	=	http://docs.oasis-open.org/ns/icom/calendar/201008
<code>icom_task</code>	=	http://docs.oasis-open.org/ns/icom/task/201008
<code>icom_forum</code>	=	http://docs.oasis-open.org/ns/icom/forum/201008
<code>icom_conf</code>	=	http://docs.oasis-open.org/ns/icom/conference/201008

Note: The namespace prefix `icom_core` represents the IRI reference <http://docs.oasis-open.org/ns/icom/core/201008> for ICOM core namespace. Both the unprefix name `Entity` and prefixed name `icom_core:Entity` are qualified names that SHALL be interpreted by the expanded name <http://docs.oasis-open.org/ns/icom/core/201008#Entity>.

3 Core Model

3.1 Main Branch

3.1.1 Entity and Top-Level Subclasses

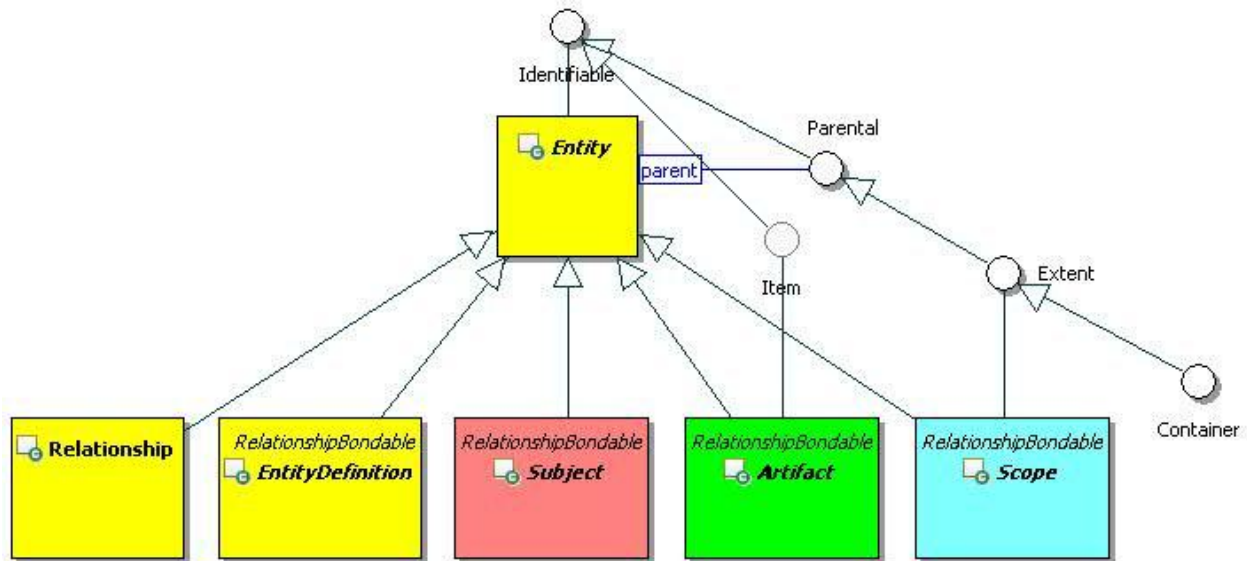


Figure 1: Entity and Top-Level Abstract Classes.

Figure 1 depicts Entity and top-level abstract classes forming the main branch of the ICOM class hierarchy. It depicts the Scope, Subject, and Artifact classes that represent the roots of the three major sub-branches of ICOM class hierarchy.

3.1.2 Identifiable

3.1.2.1 Description

An identifiable object has *objectId* and *changeToken* properties. The assignment of an *objectId* is implementation-dependent. The *objectId* is read only (immutable) once it is assigned.

3.1.2.2 Class Definition

The `Identifiable` class is a mixin class which defines the characteristics of entities and non-entities that enables unique identification.

The `Identifiable` class has attribute values:

localNamespace

Value: `icom_core`

localName

Value: `Identifiable`

374 **extendsFrom**
375 Value:
376
377 **stereotype**
378 Value: mixin
379
380 **description**
381 Value: Identifiable is a mixin class which defines the characteristics of all entities and some non-
382 entities that enables unique identification.
383
384 **propertyDefinitions**
385 The values for this attribute are defined in Section 3.1.2.3.

386 **3.1.2.3 Property Definitions**

387 The Identifiable class MUST have the property definitions:

388
389 **icom_core:objectId**
390 Description: A persistent identifier of an object.
391 Required: False
392 Inherited: False
393 Property Type: String
394 Cardinality: Single
395 Updatability: Read Only
396
397 **icom_core:changeToken**
398 Description: An opaque token used for optimistic locking & concurrency
399 checking.
400 Required: False
401 Inherited: False
402 Property Type: String
403 Cardinality: Single
404 Updatability: Read Only
405

406 The Identifiable class MAY include additional property definitions which are implementation-defined.
407

408 **3.1.3 Parental**

409 **3.1.3.1 Description**

410 A parental object may be a parent of other objects.

411 **3.1.3.2 Class Definition**

412 The Parental class is a mixin class which defines the characteristics of entities that may be parents of
413 other entities or identifiable objects.

414 The Parental class has attribute values:

415

416 **localNamespace**

417 Value: icom_core

418

419 **localName**

420 Value: Parental

421

422 **extendsFrom**

423 Value: icom_core:Identifiable

424

425 **stereotype**

426 Value: mixin

427

428 **description**

429 Value: Parental is a mixin class which defines the characteristics of the entities that can be

430 parents of other entities or identifiable objects.

431

432 **propertyDefinitions**

433 The values for this attribute are defined in Section 3.1.3.3.

434 3.1.3.3 Property Definitions

435 The Parental class inherits property definitions from super classes.

436 The Parental class MUST have the property definition:

437

438 **icom_core:parent**

439 Description: Parent of an object.

440 Required: False

441 Inherited: False

442 Property Type: icom_core:Parental

443 Cardinality: Single

444 Updatability: Read Only

445

446 The Parental class MAY include additional property definitions which are implementation-defined.

447

448 3.1.4 Extent

449 3.1.4.1 Description

450 An extent object is a parental object which may contain other entities.

451 3.1.4.2 Class Definition

452 The Extent class is a mixin class which defines characteristics of entities that may contain other entities.

453 The Extent class has attribute values:

454

455 **localNamespace**

456 Value: icom_core

457

458 **localName**

459 Value: Extent

460

461 **extendsFrom**

462 Value: icom_core:Parental

463

464 **stereotype**

465 Value: mixin

466

467 **description**

468 Value: Extent is a mixin class which defines the characteristics of entities that may contain other

469 entities.

470

471 **propertyDefinitions**

472 The values for this attribute are defined in Section 3.1.4.3.

473 3.1.4.3 Property Definitions

474 The Extent class inherits property definitions from super classes.

475 The Extent class MUST have the property definition:

476

477 **icom_core:parent**

478 Description:	Parent of an extent.
479 Required:	False
480 Inherited:	True
481 Property Type:	icom_core:Extent
482 Cardinality:	Single
483 Updatability:	Read Only

484

485 The Extent class MAY include additional property definitions which are implementation-defined.

486

487 3.1.5 Entity

488 3.1.5.1 Description

489 An entity is an identifiable object that can be persisted and that has an access control list.

490 Each entity is assigned an internationalized resource identifier (IRI) composed from its *objectId*. The form

491 of the IRI is implementation-dependent.

492 3.1.5.2 Class Definition

493 The Entity class has attribute values:

494

495 **localNamespace**

496 Value: icom_core

497

498 **localName**

499 Value: Entity

500

501 **extendsFrom**

502 Value: icom_core:Identifiable

503

504 **stereotype**

505 Value: primary

506

507 **isAbstract**

508 Value: TRUE

509

510 **description**

511 Value: An entity is an object with an immutable id and individual access control.

512

513 **propertyDefinitions**

514 The values for this attribute are defined in Section 3.1.5.3.

515 **3.1.5.3 Property Definitions**

516 The Entity class inherits property definitions from super classes.

517 The Entity class MUST have the property definitions:

518

519 **icom_core:name**

520 Description:	Name of an entity.
521 Required:	False
522 Inherited:	False
523 Property Type:	String
524 Cardinality:	Single
525 Updatability:	Read Write

526

527 **icom_core:createdBy**

528 Description:	An actor who created an entity.
529 Required:	False
530 Inherited:	False
531 Property Type:	icom_core:Actor
532 Cardinality:	Single
533 Updatability:	Read Only

534

535	icom_core:creationDate	
536	Description:	Date and time when an entity is created. It is immutable.
537	Required:	False
538	Inherited:	False
539	Property Type:	DateTime
540	Cardinality:	Single
541	Updatability:	Read Only
542		
543	icom_core:lastModifiedBy	
544	Description:	An actor who last modified an entity.
545	Required:	False
546	Inherited:	False
547	Property Type:	icom_core:Actor
548	Cardinality:	Single
549	Updatability:	Read Only
550		
551	icom_core:lastModificationDate	
552	Description:	Date and time of last modification.
553	Required:	False
554	Inherited:	False
555	Property Type:	DateTime
556	Cardinality:	Single
557	Updatability:	Read Only
558		
559	icom_core:parent	
560	Description:	A parental entity which contains an entity.
561	Required:	False
562	Inherited:	False
563	Property Type:	icom_core:Parental
564	Cardinality:	Single
565	Updatability:	Read Only
566		
567	icom_ac:owner	
568	Description:	A subject who owns an entity.
569	Required:	True
570	Inherited:	False
571	Property Type:	icom_ac:Owner
572	Cardinality:	Single
573	Updatability:	Read Write
574		
575	icom_ac:accessControlList	
576	Description:	Access control list on an entity.

577	Required:	False
578	Inherited:	False
579	Property Type:	icom_ac:AccessControllist
580	Cardinality:	Single
581	Updatability:	Read Write

582

583 **icom_meta:attachedMarker**

584	Description:	Zero or more markers applied on an entity.
585	Required:	False
586	Inherited:	False
587	Property Type:	icom_meta:Marker
588	Cardinality:	Multi
589	Updatability:	Read Only

590

591 **icom_meta:categoryApplication**

592	Description:	Zero or more category applications on an entity.
593	Required:	False
594	Inherited:	False
595	Property Type:	icom_meta:CategoryApplication
596	Cardinality:	Multi
597	Updatability:	Read Only

598

599 **icom_meta:tagApplication**

600	Description:	Zero or more tag applications on an entity.
601	Required:	False
602	Inherited:	False
603	Property Type:	icom_meta:TagApplication
604	Cardinality:	Multi
605	Updatability:	Read Only

606

607 The Entity class MAY include additional property definitions which are implementation-defined.

608

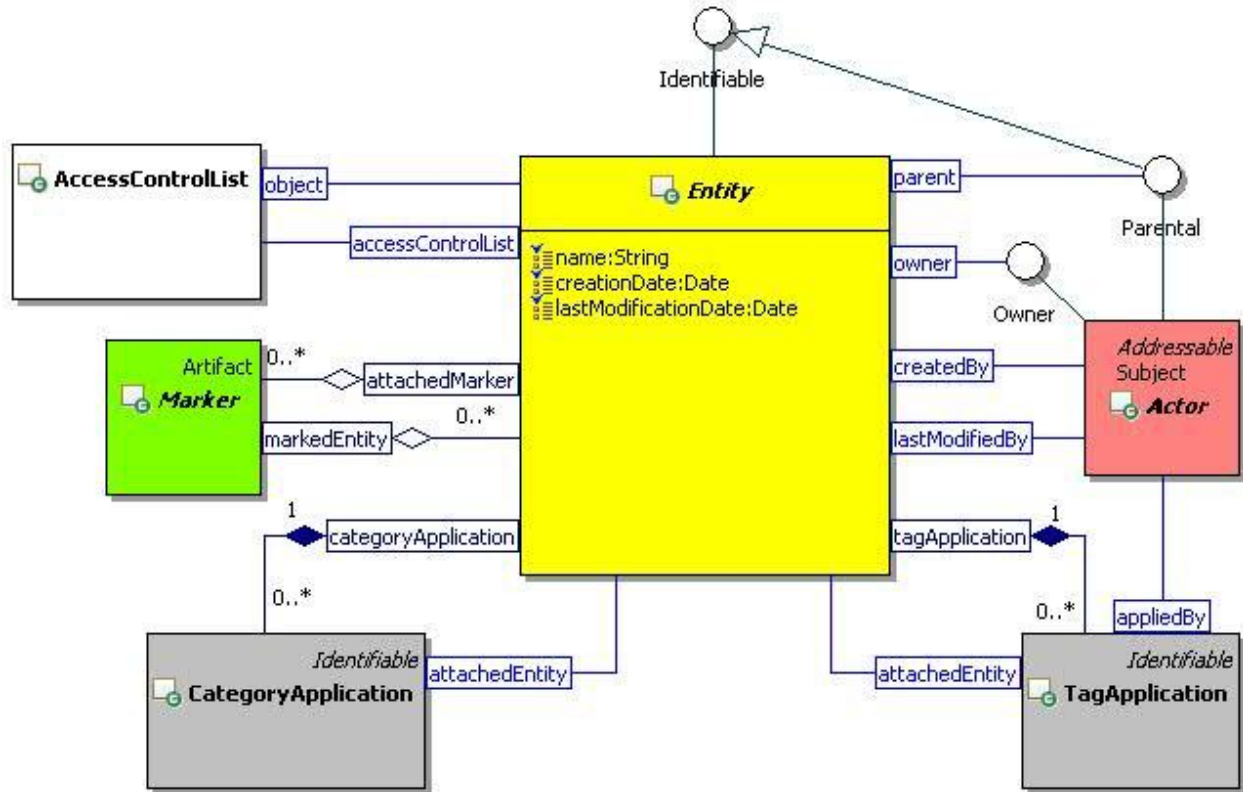


Figure 2: Entity Class Diagram.

3.1.6 EntityDefinition

3.1.6.1 Description

An entity definition is an entity that defines a type of entities.

3.1.6.2 Class Definition

The EntityDefinition class has attribute values:

localNamespace

Value: icom_core

localName

Value: EntityDefinition

extendsFrom

Value: icom_core:Entity, icom_meta:RelationshipBondable

stereotype

Value: primary

630 **isAbstract**
631 Value: TRUE
632
633 **description**
634 Value: An entity definition defines a type of entities.
635
636 **propertyDefinitions**
637 The values for this attribute are defined in Section 3.1.6.3.

638 **3.1.6.3 Property Definitions**

639 The EntityDefinition class inherits property definitions from super classes.
640 The EntityDefinition class MUST have the property definition:

641
642 **icom_core:description**
643 Description: A description of an entity definition.
644 Required: False
645 Inherited: False
646 Property Type: String
647 Cardinality: Single
648 Updatability: Read Write

649
650 The EntityDefinition class MAY include additional property definitions which are implementation-defined.
651

652 **3.1.7 Overview of Scope, Subject, and Artifact Branches**

653 The UML diagram in Figure 3 depicts the core classes in the Scope, Subject, and Artifact branches of
654 ICOM class hierarchy. Scope branch includes the model of communities and spaces which are containers
655 of subjects and artifacts. Subject branch includes the model of actors, groups, and roles. Artifact branch
656 includes the model of content and metadata produced by actors.

657 Note: The Subject and Artifact branches support the separation of concerns of user administration and
658 content management. Typically subjects and artifacts are joined in the (subject, privilege, artifact) triples
659 of access control model. Some of the (subject, privilege, artifact) triples are derived from the scopes of the
660 role assignments and the artifacts contained by the scopes. The communities and spaces contain
661 subjects and artifacts; however, membership of subjects in a space is administered separately from
662 management of artifacts in the space.

663 Scope, Subject, and Artifact are defined in Section 3.2, 3.3, and 3.4, respectively.

664

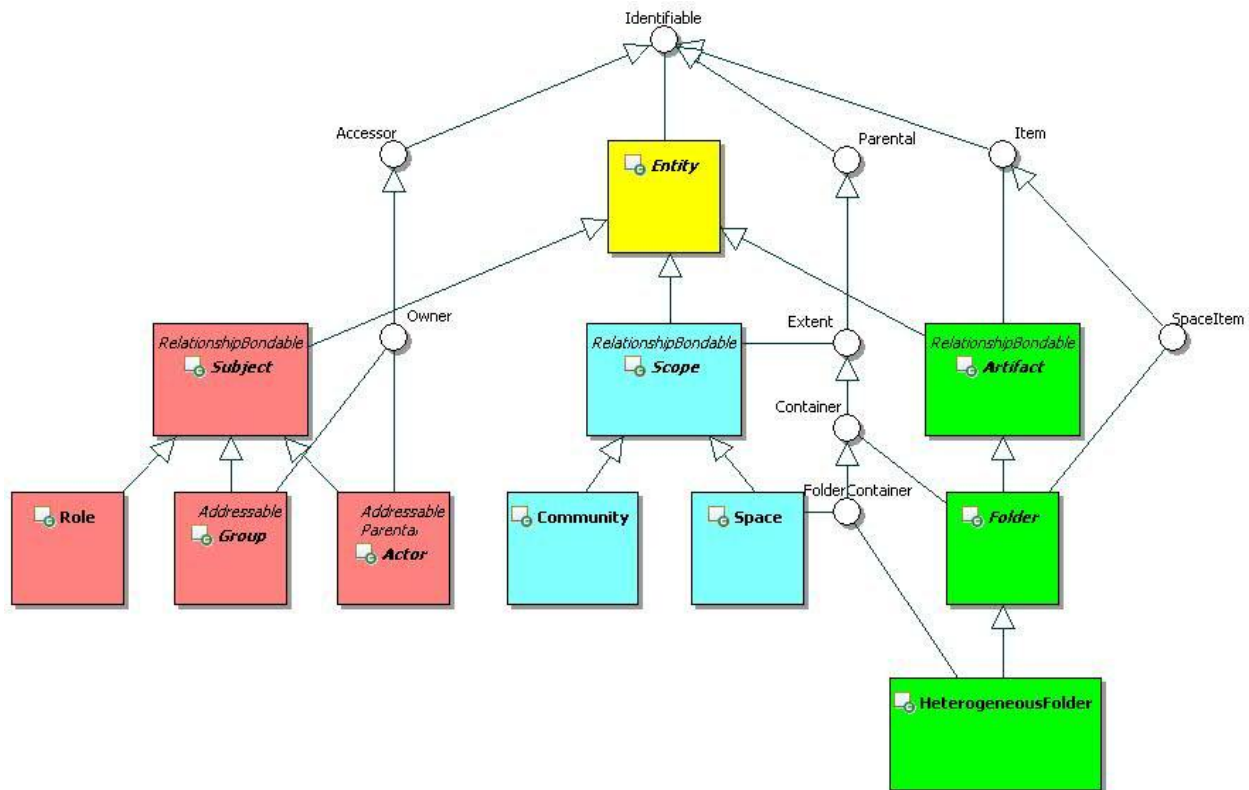


Figure 3: Scope, Subject, and Artifact Branches.

3.2 Scope Branch

3.2.1 Scope and Top-Level Subclasses

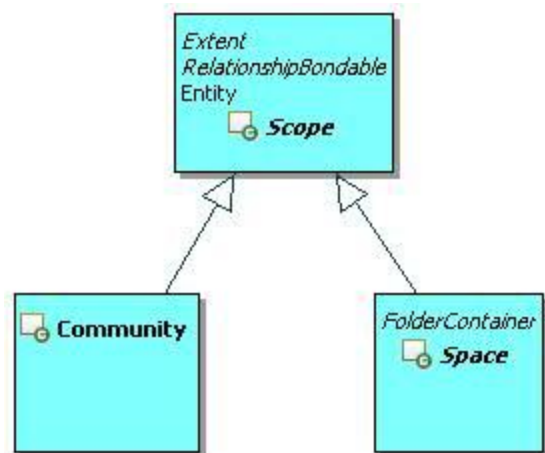


Figure 4: Scope Branch.

Figure 4 depicts the top-level classes of Scope Branch, which includes Scope, Community, and Space.

3.2.2 Scope

3.2.2.1 Description

A scope is an extent of an administrative policy.

3.2.2.2 Class Definition

The Scope class has attribute values:

localNamespace

Value: icom_core

localName

Value: Scope

extendsFrom

Value: icom_core:Entity, icom_core:Extent, icom_meta:RelationshipBondable

stereotype

Value: primary

isAbstract

Value: TRUE

description

Value: A scope is an extent of an administrative realm.

propertyDefinitions

The values for this attribute are defined in Section 3.2.2.3.

3.2.2.3 Property Definitions

The Scope class inherits property definitions from super classes.

The Scope class MUST have the property definitions:

icom_core:description

Description: A description of a scope.

Required: False

Inherited: False

Property Type: String

Cardinality: Single

Updatability: Read Write

icom_core:parent

Description: A community which contains a scope.

713	Required:	False
714	Inherited:	True
715	Property Type:	icom_core:Community
716	Cardinality:	Single
717	Updatability:	Read Only
718		
719	icom_core:group	
720	Description:	Zero or more groups defined in a scope.
721	Required:	False
722	Inherited:	False
723	Property Type:	icom_core:Group
724	Cardinality:	Multi
725	Updatability:	Read Only
726		
727	icom_core:memberGroup	
728	Description:	Member groups of a scope, i.e. groups whose assigned
729		scopes include this scope.
730	Required:	False
731	Inherited:	False
732	Property Type:	icom_core:Group
733	Cardinality:	Multi
734	Updatability:	Read Only
735		
736	icom_ac:roleDefinition	
737	Description:	Zero or more role definitions defined in a scope.
738	Required:	False
739	Inherited:	False
740	Property Type:	icom_ac:RoleDefinition
741	Cardinality:	Multi
742	Updatability:	Read Only
743		
744	icom_ac:role	
745	Description:	Zero or more roles defined in a scope.
746	Required:	False
747	Inherited:	False
748	Property Type:	icom_ac:Role
749	Cardinality:	Multi
750	Updatability:	Read Only
751		
752	icom_meta:relationship	
753	Description:	Zero or more relationships associated with a scope.
754	Required:	False
755	Inherited:	False

Property Type: icom_meta:Relationship
Cardinality: Multi
Updatability: Read Only

The Scope class MAY include additional property definitions which are implementation-defined.

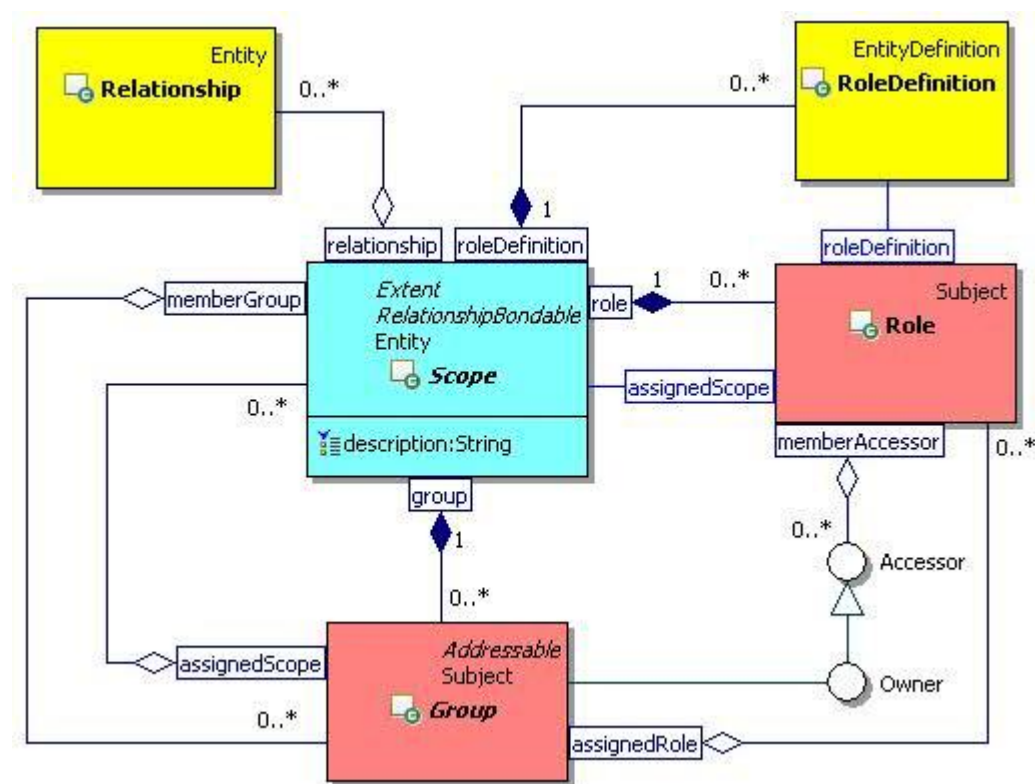


Figure 5: Scope Class Diagram.

3.2.3 Community

3.2.3.1 Description

A community is a scope that has a set of actors as members who can participate in a set of spaces. It is implementation-dependent whether or not a space in a community can include participating actors who are not members of a parent community or ancestor communities.

3.2.3.2 Class Definition

The Community class has attribute values:

localNamespace

Value: icom_core

localName

Value: Community

778

779 **extendsFrom**

780 Value: icom_core:Scope

781

782 **stereotype**

783 Value: primary

784

785 **description**

786 Value: A community is a scope that has a set of actors as members who can participate in a set

787 of spaces.

788

789 **propertyDefinitions**

790 The values for this attribute are defined in Section 3.2.3.3.

791 **3.2.3.3 Property Definitions**

792 The Community class inherits property definitions from super classes.

793 The Community class MUST have the property definitions:

794

795 **icom_core:community**

796	Description:	Sub-communities of a community.
797	Required:	False
798	Inherited:	False
799	Property Type:	icom_core:Community
800	Cardinality:	Multi
801	Updatability:	Read Only

802

803 **icom_core:space**

804	Description:	Spaces of a community.
805	Required:	False
806	Inherited:	False
807	Property Type:	icom_core:Space
808	Cardinality:	Multi
809	Updatability:	Read Only

810

811 **icom_core:actor**

812	Description:	Managed actors of a community, i.e. actors whose parent
813		community is this community.
814	Required:	False
815	Inherited:	False
816	Property Type:	icom_core:Actor
817	Cardinality:	Multi
818	Updatability:	Read Only

819

icom_core:memberActor

Description:	Member actors of a community, i.e. actors whose assigned communities include this community.
Required:	False
Inherited:	False
Property Type:	icom_core:Actor
Cardinality:	Multi
Updatability:	Read Only

The Community class MAY include additional property definitions which are implementation-defined.

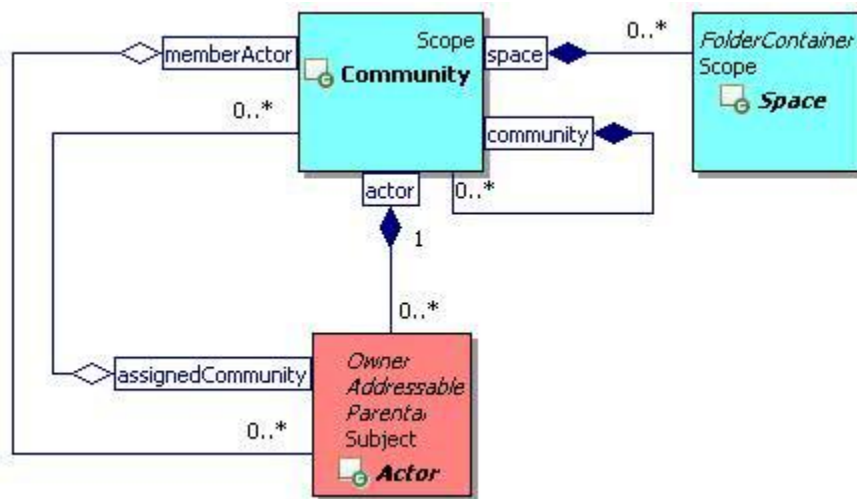


Figure 6: Community Class Diagram.

3.2.4 Space

3.2.4.1 Description

A space is a scope that defines a durable context and place for actors to work or collaborate.

3.2.4.2 Class Definition

The Space class has attribute values:

localNamespace

Value: icom_core

localName

Value: Space

extendsFrom

Value: icom_core:Scope, icom_core:FolderContainer

stereotype
Value: primary

description
Value: A space is a scope that defines a durable context and place for actors to work or collaborate.

propertyDefinitions
The values for this attribute are defined in Section 3.2.4.3.

3.2.4.3 Property Definitions

The Space class inherits property definitions from super classes.
The Space class MUST have the property definition:

icom_core:element	
Description:	Elements of a space.
Required:	False
Inherited:	True
Property Type:	icom_core:SpaceItem
Cardinality:	Multi
Updatability:	Read Only

The Space class MAY include additional property definitions which are implementation-defined.

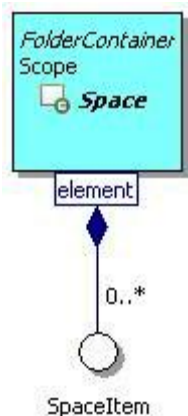


Figure 7: Space Class Diagram.

3.3 Subject Branch

3.3.1 Subject and Top-Level Subclasses

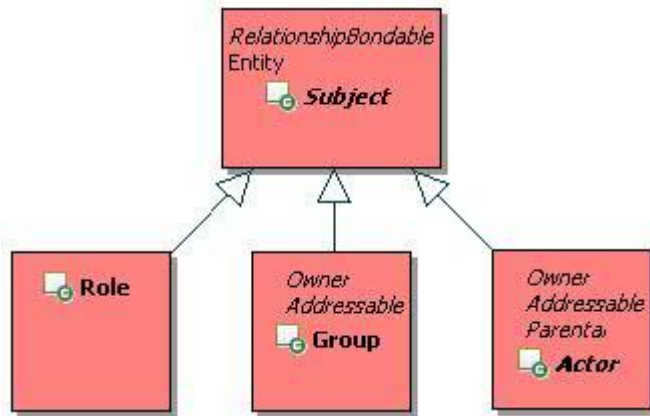


Figure 8: Subject Branch.

Figure 8 depicts the top-level classes of Subject Branch, which includes Subject, Role, Group, and Actor.

3.3.2 Subject

3.3.2.1 Description

A subject is an entity that can have rights to perform actions.

3.3.2.2 Class Definition

The Subject class has attribute values:

```
localNamespace
    Value: icom_core

localName
    Value: Subject

extendsFrom
    Value: icom_core:Entity, icom_meta:RelationshipBondable

stereotype
    Value: primary

isAbstract
    Value: TRUE

description
    Value: A subject is an entity that can have rights to perform actions.
```

904 **propertyDefinitions**

905 The values for this attribute are defined in Section 3.3.2.3.

906 **3.3.2.3 Property Definitions**

907 The Subject class inherits property definitions from super classes.

908 The Subject class **MUST** have the property definitions:

909

910 **icom_core:description**

911 Description: A description of a subject.

912 Required: False

913 Inherited: False

914 Property Type: String

915 Cardinality: Single

916 Updatability: Read Write

917

918 **icom_core:parent**

919 Description: A scope which contains a subject.

920 Required: False

921 Inherited: True

922 Property Type: icom_core:Scope

923 Cardinality: Single

924 Updatability: Read Only

925

926 **icom_meta:relationship**

927 Description: Zero or more relationships associated with a subject.

928 Required: False

929 Inherited: False

930 Property Type: icom_meta:Relationship

931 Cardinality: Multi

932 Updatability: Read Only

933

934 **icom_meta:property**

935 Description: Zero or more extended properties of a subject.

936 Required: False

937 Inherited: False

938 Property Type: icom_meta:Property

939 Cardinality: Multi

940 Updatability: Read Write

941

942 The Subject class **MAY** include additional property definitions which are implementation-defined.

943

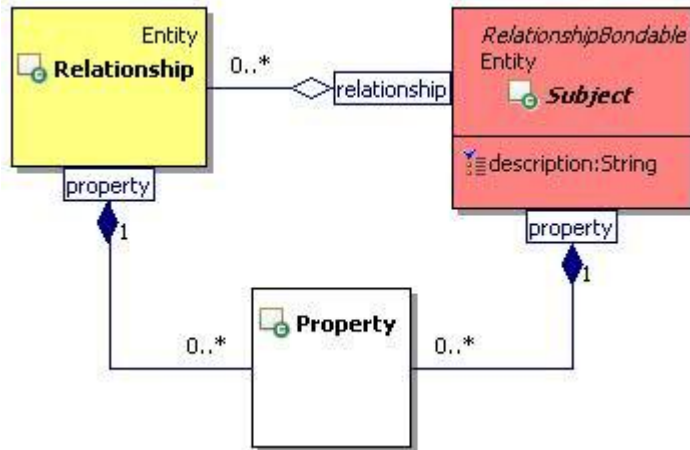


Figure 9: Subject Class Diagram.

3.3.3 Group

3.3.3.1 Description

A group is a subject representing a set of actors and sub-groups.

A group can be part of one or more super-groups.

It can be an owner of one or more entities.

3.3.3.2 Class Definition

The Group class has attribute values:

localNamespace

Value: icom_core

localName

Value: Group

extendsFrom

Value: icom_core:Subject, icom_core:Addressable, icom_ac:Accessor

Optional Value: icom_ac:Owner

stereotype

Value: primary

description

Value: A group is a subject representing a set of actors and sub-groups. A group can be part of one or more super-groups. It can be an owner of one or more entities.

propertyDefinitions

The values for this attribute are defined in Section 3.3.3.3.

3.3.3.3 Property Definitions

The Group class inherits property definitions from super classes.

The Group class MUST have the property definitions:

icom_core:assignedGroup

Description:	A group's super-groups.
Required:	False
Inherited:	False
Property Type:	icom_core:Group
Cardinality:	Multi
Updatability:	Read Write

icom_core:assignedScope

Description:	A group's scopes.
Required:	False
Inherited:	False
Property Type:	icom_core:Scope
Cardinality:	Multi
Updatability:	Read Write

icom_core:memberGroup

Description:	Sub-groups of a group.
Required:	False
Inherited:	False
Property Type:	icom_core:Group
Cardinality:	Multi
Updatability:	Read Only

icom_core:memberActor

Description:	Actors in a group.
Required:	False
Inherited:	False
Property Type:	icom_core:Actor
Cardinality:	Multi
Updatability:	Read Only

icom_ac:assignedRole

Description:	A group's roles.
Required:	False
Inherited:	False
Property Type:	icom_ac:Role
Cardinality:	Multi

Updatability: Read Write

The Group class MAY include additional property definitions which are implementation-defined.

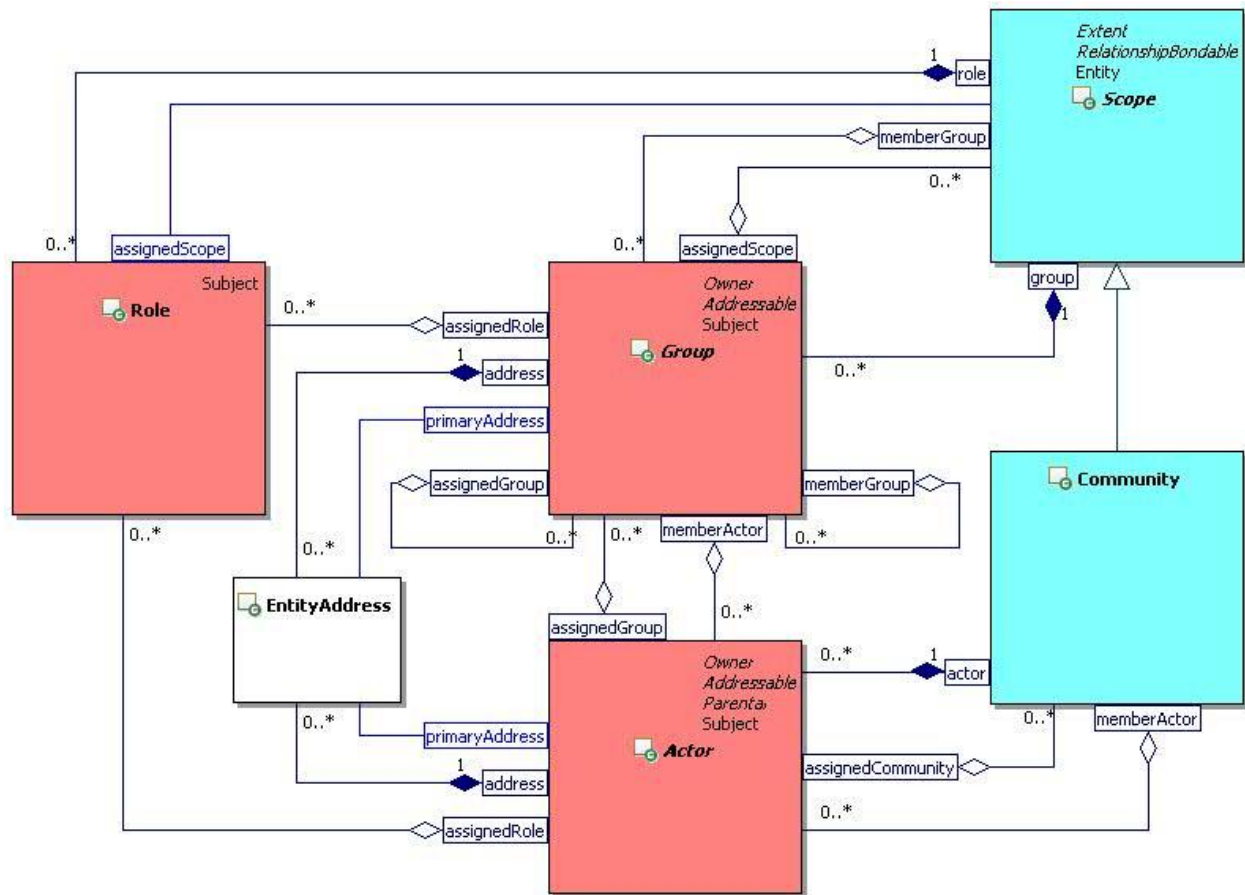


Figure 10: Group and Actor Class Diagram.

3.3.4 Actor

3.3.4.1 Description

An actor is a subject that can perform actions on objects.

It can be an owner of entities.

3.3.4.2 Class Definition

The Actor class has attribute values:

localNamespace

Value: icom_core

localName

Value: Actor

1035

1036 **extendsFrom**

1037 Value: icom_core:Subject, icom_core:Addressable, icom_ac:Owner

1038

1039 **stereotype**

1040 Value: primary

1041

1042 **isAbstract**

1043 Value: TRUE

1044

1045 **description**

1046 Value: An actor is a subject that can perform actions on objects.

1047

1048 **propertyDefinitions**

1049 The values for this attribute are defined in Section 3.3.4.3.

1050 3.3.4.3 Property Definitions

1051 The Actor class inherits property definitions from super classes.

1052 The Actor class MUST have the property definitions:

1053

1054 **icom_core:parent**

1055 Description: A community which contains an actor.

1056 Required: False

1057 Inherited: True

1058 Property Type: icom_core:Community

1059 Cardinality: Single

1060 Updatability: Read Only

1061

1062 **icom_core:assignedGroup**

1063 Description: An actor's groups.

1064 Required: False

1065 Inherited: False

1066 Property Type: icom_core:Group

1067 Cardinality: Multi

1068 Updatability: Read Write

1069

1070 **icom_core:assignedCommunity**

1071 Description: An actor's communities.

1072 Required: False

1073 Inherited: False

1074 Property Type: icom_core:Community

1075 Cardinality: Multi

1076 Updatability: Read Write

1077		
1078	icom_ac:assignedRole	
1079	Description:	An actor's roles.
1080	Required:	False
1081	Inherited:	False
1082	Property Type:	icom_ac:Role
1083	Cardinality:	Multi
1084	Updatability:	Read Write

1085

1086 The Actor class MAY include additional property definitions which are implementation-defined.

1087

1088 3.3.5 Person

1089 3.3.5.1 Description

1090 A person is an individual human who may be an actor.

1091 A person has a personal space.

1092 3.3.5.2 Class Definition

1093 The Person class has attribute values:

1094

1095 **localNamespace**

1096 Value: icom_core

1097

1098 **localName**

1099 Value: Person

1100

1101 **extendsFrom**

1102 Value: icom_core:Actor

1103

1104 **stereotype**

1105 Value: primary

1106

1107 **description**

1108 Value: A person is an individual human who may be an actor.

1109

1110 **propertyDefinitions**

1111 The values for this attribute are defined in Section 3.3.5.3.

1112 3.3.5.3 Property Definitions

1113 The Person class inherits property definitions from super classes.

1114 The Person class MUST have the property definitions:

1115

1116	icom_core:givenName	
1117	Description:	Given name of a person.
1118	Required:	False
1119	Inherited:	False
1120	Property Type:	String
1121	Cardinality:	Single
1122	Updatability:	Read Write
1123		
1124	icom_core:middleName	
1125	Description:	Middle name of a person. Can include multiple names concatenated.
1126		
1127	Required:	False
1128	Inherited:	False
1129	Property Type:	String
1130	Cardinality:	Single
1131	Updatability:	Read Write
1132		
1133	icom_core:familyName	
1134	Description:	Family name of a person.
1135	Required:	False
1136	Inherited:	False
1137	Property Type:	String
1138	Cardinality:	Single
1139	Updatability:	Read Write
1140		
1141	icom_core:prefix	
1142	Description:	Prefix of a person's name.
1143	Required:	False
1144	Inherited:	False
1145	Property Type:	String
1146	Cardinality:	Single
1147	Updatability:	Read Write
1148		
1149	icom_core:suffix	
1150	Description:	Suffix of a person's name.
1151	Required:	False
1152	Inherited:	False
1153	Property Type:	String
1154	Cardinality:	Single
1155	Updatability:	Read Write
1156		
1157	icom_core:nickname	
1158	Description:	Nickname of a person.

1159	Required:	False
1160	Inherited:	False
1161	Property Type:	String
1162	Cardinality:	Multi
1163	Updatability:	Read Write
1164		
1165	icom_core:jobTitle	
1166	Description:	Job title of a person.
1167	Required:	False
1168	Inherited:	False
1169	Property Type:	String
1170	Cardinality:	Single
1171	Updatability:	Read Write
1172		
1173	icom_core:department	
1174	Description:	A person's affiliated department.
1175	Required:	False
1176	Inherited:	False
1177	Property Type:	String
1178	Cardinality:	Single
1179	Updatability:	Read Write
1180		
1181	icom_core:officeLocation	
1182	Description:	Location of a person's department.
1183	Required:	False
1184	Inherited:	False
1185	Property Type:	String
1186	Cardinality:	Single
1187	Updatability:	Read Write
1188		
1189	icom_core:company	
1190	Description:	A person's affiliated company.
1191	Required:	False
1192	Inherited:	False
1193	Property Type:	String
1194	Cardinality:	Single
1195	Updatability:	Read Write
1196		
1197	icom_core:profession	
1198	Description:	A person's profession.
1199	Required:	False
1200	Inherited:	False

1201	Property Type:	String
1202	Cardinality:	Single
1203	Updatability:	Read Write
1204		
1205	icom_core:personalSpace	
1206	Description:	Personal space of a person.
1207	Required:	False
1208	Inherited:	False
1209	Property Type:	icom_core:Space
1210	Cardinality:	Single
1211	Updatability:	Read Only
1212		
1213	icom_presence:presence	
1214	Description:	Presence of a person.
1215	Required:	False
1216	Inherited:	False
1217	Property Type:	icom_presence:Presence
1218	Cardinality:	Single
1219	Updatability:	Read Only
1220		
1221	icom_msg:instantMessageFeed	
1222	Description:	Instant message feed for a person.
1223	Required:	False
1224	Inherited:	False
1225	Property Type:	icom_msg:InstantMessageFeed
1226	Cardinality:	Single
1227	Updatability:	Read Only
1228		
1229	The Person class MAY include additional property definitions which are implementation-defined.	
1230		

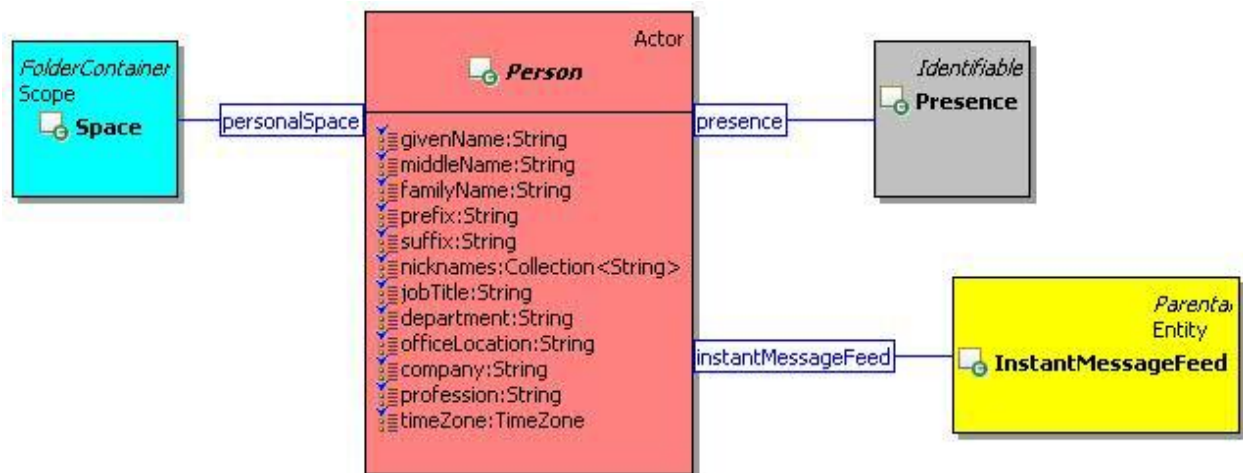


Figure 11: Person Class Diagram.

3.3.6 Resource

3.3.6.1 Description

A resource is an actor representing a bookable resource, such as a conference room, equipment, or on-line conference.

A resource is associated with a resource space that contains a resource scheduling calendar and on-line conference.

3.3.6.2 Class Definition

The Resource class has attribute values:

localNamespace

Value: icom_core

localName

Value: Resource

extendsFrom

Value: icom_core:Actor

stereotype

Value: primary

description

Value: A resource actor is an actor representing a bookable resource, such as a conference room, equipment, or on-line conference.

propertyDefinitions

The values for this attribute are defined in Section 3.3.6.3.

3.3.6.3 Property Definitions

The Resource class inherits property definitions from super classes.

The Resource class MUST have the property definitions:

icom_core:resourceSpace

Description:	Administrative space of a resource actor.
Required:	False
Inherited:	False
Property Type:	icom_core:Space
Cardinality:	Single
Updatability:	Read Only

icom_core:location

Description:	Location of a resource.
Required:	False
Inherited:	False
Property Type:	icom_core:Location
Cardinality:	Single
Updatability:	Read Write

icom_core:capacity

Description:	Capacity of a resource.
Required:	False
Inherited:	False
Property Type:	Integer
Cardinality:	Single
Updatability:	Read Write

icom_core:resourceType

Description:	Type of a resource.
Required:	False
Inherited:	False
Property Type:	icom_core:ResourceType
Cardinality:	Single
Updatability:	Read Write

icom_core:bookingRule

Description:	Resource booking rule.
Required:	False
Inherited:	False
Property Type:	icom_core:ResourceBookingRule
Cardinality:	Single

1303	Updatability:	Read Write
1304		
1305	icom_core:bookingApprover	
1306	Description:	One or more persons who approve the booking of a resource.
1307	Required:	False
1308	Inherited:	False
1309	Property Type:	icom_core:Person
1310	Cardinality:	Multi
1311	Updatability:	Read Write

The Resource class MAY include additional property definitions which are implementation-defined.

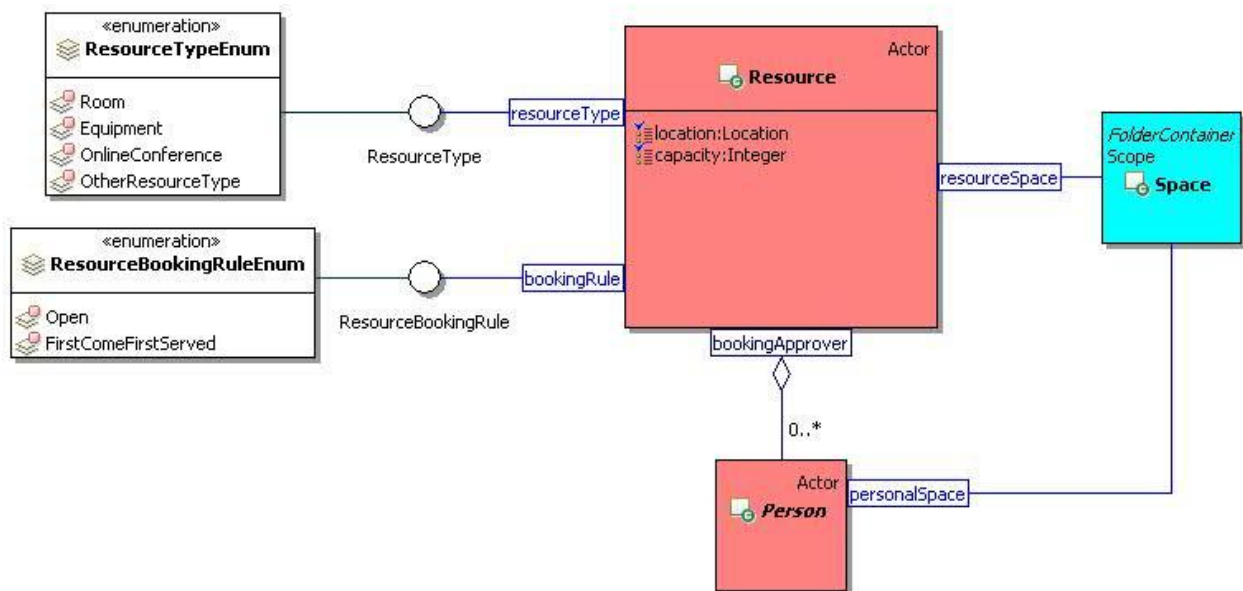


Figure 12: Resource Class Diagram.

3.3.7 ResourceType

3.3.7.1 Description

A resource type is a category of resources.

3.3.7.2 Class Definition

The ResourceType class is a mixin class which defines a resource type.

The ResourceType class has attribute values:

localNamespace
Value: icom_core

1328 **localName**
 1329 Value: ResourceType
 1330
 1331 **extendsFrom**
 1332 Value:
 1333
 1334 **stereotype**
 1335 Value: mixin
 1336
 1337 **description**
 1338 Value: ResourceType is a mixin class which defines a type of resources.
 1339
 1340 **propertyDefinitions**
 1341 The values for this attribute are defined in Section 3.3.7.3.

1342 **3.3.7.3 Property Definitions**

1343 The ResourceType class MAY include additional property definitions which are implementation-defined.
 1344

1345 **3.3.8 ResourceTypeEnum**

1346 The ResourceTypeEnum class is an enum class that enumerates the instances each of which expresses
 1347 a type of resources.
 1348 The ResourceTypeEnum class has attribute values:

1349
 1350 **localNamespace**
 1351 Value: icom_core
 1352
 1353 **localName**
 1354 Value: ResourceTypeEnum
 1355
 1356 **extendsFrom**
 1357 Value: ResourceType
 1358
 1359 **stereotype**
 1360 Value: primary
 1361
 1362 **isEnumeration**
 1363 Value: TRUE
 1364
 1365 **description**
 1366 Value: A type of resources.
 1367

1368 **instances**
1369 Value: <icom_core:Room, icom_core:Equipment, icom_core:OnlineConference,
1370 icom_core:OtherResourceType>

1371

1372 ICOM defines four resource types:

- 1373 • **icom_core:Room** a resource represents a room.
- 1374 • **icom_core:Equipment** a resource represents an equipment.
- 1375 • **icom_core:OnlineConference** a resource represents an online conference.
- 1376 • **icom_core:OtherResourceType** a resource represents other things.

1377

1378 **3.3.9 ResourceBookingRule**

1379 **3.3.9.1 Description**

1380 A resource booking rule is a strategy for allocating resources for calendar scheduling.

1381 **3.3.9.2 Class Definition**

1382 The ResourceBookingRule class is a mixin class which defines a resource booking rule.

1383 The ResourceBookingRule class has attribute values:

1384

1385 **localNamespace**

1386 Value: icom_core

1387

1388 **localName**

1389 Value: ResourceBookingRule

1390

1391 **extendsFrom**

1392 Value:

1393

1394 **stereotype**

1395 Value: mixin

1396

1397 **description**

1398 Value: ResourceBookingRule is a mixin class which defines a rule for allocating resources for
1399 calendar scheduling.

1400

1401 **propertyDefinitions**

1402 The values for this attribute are defined in Section 3.3.9.3.

1403 **3.3.9.3 Property Definitions**

1404 The ResourceBookingRule class MAY include additional property definitions which are implementation-
1405 defined.

1406

3.3.10 ResourceBookingRuleEnum

The ResourceBookingRuleEnum class is an enum class that enumerates the instances each of which expresses a booking rule.

The ResourceBookingRuleEnum class has attribute values:

localNamespace

Value: icom_core

localName

Value: ResourceBookingRuleEnum

extendsFrom

Value: ResourceBookingRule

stereotype

Value: primary

isEnumeration

Value: TRUE

description

Value: A resource booking rule for allocating resources for calendar scheduling.

instances

Value: <icom_core:Open, icom_core:FirstComeFirstServed>

ICOM defines two resource booking rules:

- **icom_core:Open** a resource is open for booking.
- **icom_core:FirstComeFirstServed** a resource is first come first served.

3.4 Artifact Branch

3.4.1 Artifact and Top-Level Subclasses

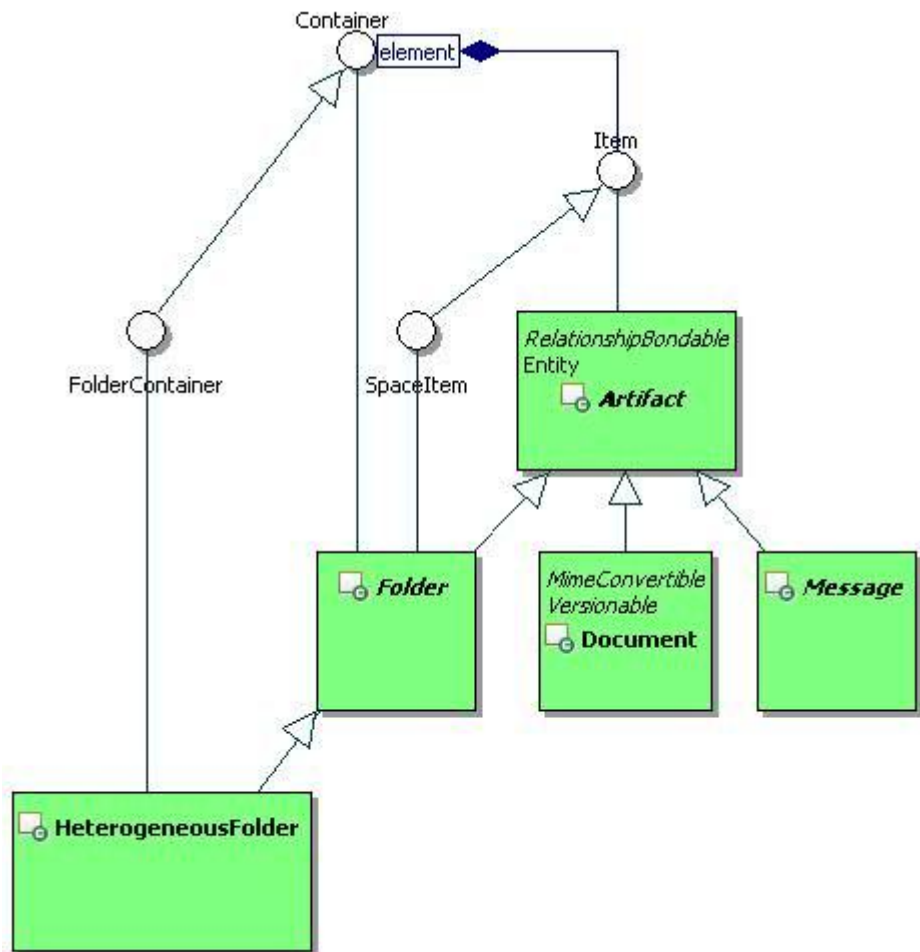


Figure 13: Artifact Branch.

Figure 13 depicts the top-level classes of Artifact Branch, which includes Artifact, Folder, HeterogeneousFolder, Document, and Message.

3.4.2 Item

3.4.2.1 Description

An item is an element of a container.

The parent of an item MUST be a container.

3.4.2.2 Class Definition

The Item class is a mixin class which defines the characteristics of entities that can be elements of a Container.

The Item class has attribute values:

localNamespace

Value: icom_core

1454

1455 **localName**

1456 Value: Item

1457

1458 **extendsFrom**

1459 Value: icom_core:Identifiable

1460

1461 **stereotype**

1462 Value: mixin

1463

1464 **description**

1465 Value: Item is a mixin class which defines the characteristics of entities that can be placed in a

1466 Container.

1467

1468 **propertyDefinitions**

1469 The values for this attribute are defined in Section 3.4.2.3.

1470 3.4.2.3 Property Definitions

1471 The Item class inherits property definitions from super classes.

1472 The Item class MUST have the property definition:

1473

1474 **icom_core:parent**

1475 Description:	A parent container of an item.
1476 Required:	False
1477 Inherited:	True
1478 Property Type:	icom_core:Container
1479 Cardinality:	Single
1480 Updatability:	Read Only

1481

1482 The Item class MAY have the optional property definition:

1483

1484 **icom_core:container**

1485 Description:	Zero, one, or more containers of an item, including the parent
1486	container.
1487 Required:	False
1488 Inherited:	False
1489 Property Type:	icom_core:Container
1490 Cardinality:	Multi
1491 Updatability:	Read Write

1492

1493 The Item class MAY include additional property definitions which are implementation-defined.

1494

3.4.3 Spaceltem

3.4.3.1 Description

A space item is an item that can be an element of a space.

3.4.3.2 Class Definition

The Spaceltem class is a mixin class which defines the characteristics of items that can be elements of a Space.

The Spaceltem class has attribute values:

localNamespace

Value: icom_core

localName

Value: Spaceltem

extendsFrom

Value: icom_core:Item

stereotype

Value: mixin

description

Value: Spaceltem is a mixin class which defines the characteristics of entities that can be elements of a Space.

propertyDefinitions

The values for this attribute are defined in Section 3.4.3.3.

3.4.3.3 Property Definitions

The Spaceltem class inherits property definitions from super classes.

The Spaceltem class MAY include additional property definitions which are implementation-defined.

3.4.4 Container

3.4.4.1 Description

A container is an extent that contains items.

3.4.4.2 Class Definition

The Container class is a mixin class which defines the characteristics of extents that contain items.

The Container class has attribute values:

1532 **localNamespace**
1533 Value: icom_core
1534
1535 **localName**
1536 Value: Container
1537
1538 **extendsFrom**
1539 Value: icom_core:Extent
1540
1541 **stereotype**
1542 Value: mixin
1543
1544 **description**
1545 Value: A container is an extent that contains items.
1546
1547 **propertyDefinitions**
1548 The values for this attribute are defined in Section 3.4.4.3.

1549 **3.4.4.3 Property Definitions**

1550 The Container class inherits property definitions from super classes.
1551 The Container class MUST have the property definition:

1552

1553	icom_core:element	
1554	Description:	Elements of a container, i.e. items whose parent container is the container or whose containers include the container.
1555		
1556	Required:	False
1557	Inherited:	False
1558	Property Type:	icom_core:Item
1559	Cardinality:	Multi
1560	Updatability:	Read Only

1561

1562 The Container class MAY include additional property definitions which are implementation-defined.

1563

1564 **3.4.5 FolderContainer**

1565 **3.4.5.1 Description**

1566 A folder container is a container which may contain folders. Space and heterogeneous folder are folder
1567 containers.

1568 **3.4.5.2 Class Definition**

1569 The FolderContainer class is a mixin class that defines the characteristics of containers that may contain
1570 folders.

1571 The FolderContainer class has attribute values:

1572
1573 **localNamespace**
1574 Value: icom_core
1575
1576 **localName**
1577 Value: FolderContainer
1578
1579 **extendsFrom**
1580 Value: icom_core:Container
1581
1582 **stereotype**
1583 Value: mixin
1584
1585 **description**
1586 Value: A folder container is a container which may contain folders.
1587
1588 **propertyDefinitions**
1589 The values for this attribute are defined in Section 3.4.5.3.

1590 **3.4.5.3 Property Definitions**

1591 The FolderContainer class inherits property definitions from super classes.
1592 The FolderContainer class MAY include additional property definitions which are implementation-defined.
1593

1594 **3.4.6 Artifact**

1595 **3.4.6.1 Description**

1596 An artifact is a result of a communication, cooperation, content creation, or collaboration activity.
1597 Note: Document versioning is an example of content creation activity resulting in an artifact (a version of a
1598 document).

1599 **3.4.6.2 Class Definition**

1600 The Artifact class has attribute values:

1601
1602 **localNamespace**
1603 Value: icom_core
1604
1605 **localName**
1606 Value: Artifact
1607
1608 **extendsFrom**
1609 Value: icom_core:Entity, icom_core:Item, icom_meta:RelationshipBondable
1610 Optional Value: icom_core:SpaceItem
1611

1612 **stereotype**
 1613 Value: primary
 1614
 1615 **isAbstract**
 1616 Value: TRUE
 1617
 1618 **description**
 1619 Value: An artifact is a result of a communication, cooperation, content creation, or collaboration
 1620 activity.
 1621
 1622 **propertyDefinitions**
 1623 The values for this attribute are defined in Section 3.4.6.3.

1624 **3.4.6.3 Property Definitions**

1625 The Artifact class inherits property definitions from super classes.
 1626 The Artifact class **MUST** have the property definitions:

1627 **icom_core:description**

1629	Description:	A description of an artifact.
1630	Required:	False
1631	Inherited:	False
1632	Property Type:	String
1633	Cardinality:	Single
1634	Updatability:	Read Write

1635 **icom_core:userCreationDate**

1637	Description:	Date and time when an artifact was created.
1638	Required:	False
1639	Inherited:	False
1640	Property Type:	DateTime
1641	Cardinality:	Single
1642	Updatability:	Read Write

1643 **icom_core:userLastModificationDate**

1645	Description:	Date and time when an artifact was last modified.
1646	Required:	False
1647	Inherited:	False
1648	Property Type:	DateTime
1649	Cardinality:	Single
1650	Updatability:	Read Write

1651 **icom_meta:property**

1653	Description:	Zero or more extended properties of an artifact.
------	--------------	--

1654	Required:	False
1655	Inherited:	False
1656	Property Type:	icom_meta:Property
1657	Cardinality:	Multi
1658	Updatability:	Read Write
1659		
1660	icom_meta:viewerProperty	
1661	Description:	Zero or more extended properties of an artifact visible to a viewer.
1662		
1663	Required:	False
1664	Inherited:	False
1665	Property Type:	icom_meta:Property
1666	Cardinality:	Multi
1667	Updatability:	Read Write
1668		
1669	icom_meta:relationship	
1670	Description:	Zero or more relationships associated with an artifact.
1671	Required:	False
1672	Inherited:	False
1673	Property Type:	icom_meta:Relationship
1674	Cardinality:	Multi
1675	Updatability:	Read Only

The Artifact class MAY include additional property definitions which are implementation-defined.

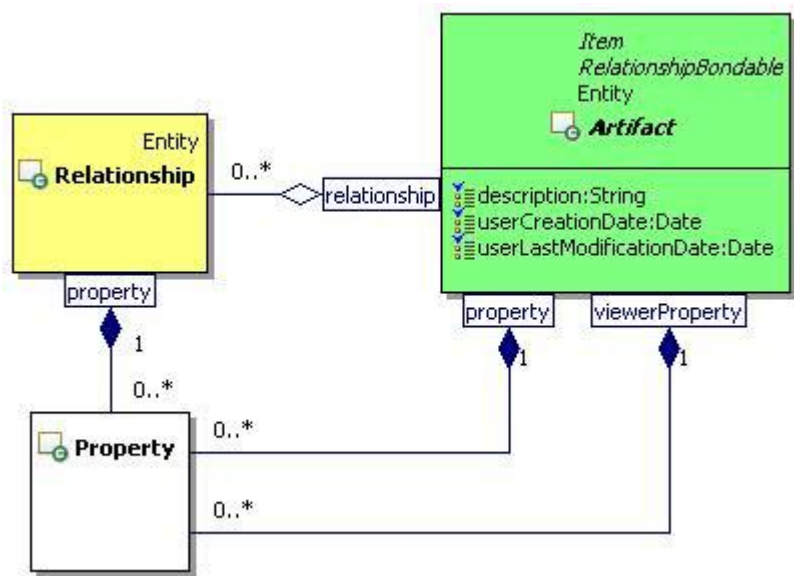


Figure 14: Artifact Class Diagram.

3.4.7 Folder

3.4.7.1 Description

A folder is an artifact that may contain other artifacts.

Note: Every folder except root folders has at least one parent folder. The parent of a root folder is a space. Subclasses of Folder class should enforce their own semantics on elements.

3.4.7.2 Class Definition

The Folder class has attribute values:

localNamespace

Value: icom_core

localName

Value: Folder

extendsFrom

Value: icom_core:Artifact, icom_core:Container, icom_core:SpaceItem

stereotype

Value: primary

isAbstract

Value: TRUE

description

Value: A folder is an artifact that may contain other artifacts.

propertyDefinitions

The values for this attribute are defined in Section 3.4.7.3.

3.4.7.3 Property Definitions

The Folder class inherits property definitions from super classes.

The Folder class MUST have the property definition:

icom_core:parent

Description: A parent container of a folder.

Required: False

Inherited: True

Property Type: icom_core:FolderContainer

Cardinality: Single

Updatability: Read Only

1722 The Folder class MAY include additional property definitions which are implementation-defined.
1723

1724 **3.4.8 HeterogeneousFolder**

1725 **3.4.8.1 Description**

1726 A heterogeneous folder is an unconstrained folder to contain any type of artifacts.
1727 Note: It is typically used for document folders, inbox, outbox, and trash folder of a space.

1728 **3.4.8.2 Class Definition**

1729 The HeterogeneousFolder class has attribute values:

1730
1731 **localNamespace**
1732 Value: icom_core
1733
1734 **localName**
1735 Value: HeterogeneousFolder
1736
1737 **extendsFrom**
1738 Value: icom_core:Folder, icom_core:FolderContainer
1739
1740 **stereotype**
1741 Value: primary
1742
1743 **description**
1744 Value: A heterogeneous folder is an unconstrained folder to contain any type of artifacts.
1745
1746 **propertyDefinitions**
1747 The values for this attribute are defined in Section 3.4.8.3.

1748 **3.4.8.3 Property Definitions**

1749 The HeterogeneousFolder class inherits property definitions from super classes.
1750 The HeterogeneousFolder class MUST have the property definition:

1751
1752 **icom_core:element**
1753 Description: Elements of a heterogeneous folder.
1754 Required: False
1755 Inherited: True
1756 Property Type: icom_core:Artifact
1757 Cardinality: Multi
1758 Updatability: Read Only
1759

1760 The HeterogeneousFolder class MAY include additional property definitions which are implementation-
1761 defined.

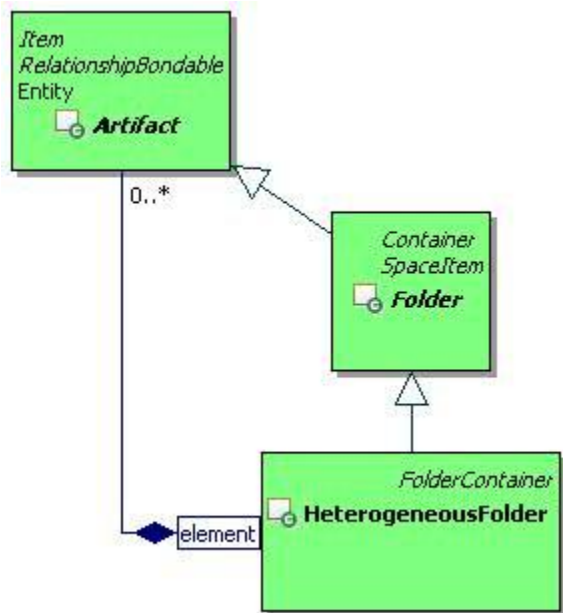


Figure 15: Heterogeneous Folder Class Diagram.

3.5 Access Control Model

3.5.1 Accessor

3.5.1.1 Description

An accessor can be granted or denied access rights to objects.

3.5.1.2 Class Definition

The Accessor class is a mixin class which defines the characteristics of subjects such as groups and actors that can be granted or denied access types in access control lists and privileges in role assignments.

The Accessor class has attribute values:

localNamespace

Value: icom_ac

localName

Value: Accessor

extendsFrom

Value: icom_core:Identifiable

stereotype

Value: mixin

1787
1788 **description**
1789 Value: Accessor is a mixin class which defines the characteristics of subjects such as groups
1790 and actors that can be granted or denied access types in access control lists and granted
1791 privileges in role assignments.

1792
1793 **propertyDefinitions**
1794 The values for this attribute are defined in Section 3.5.1.3.

1795 **3.5.1.3 Property Definitions**

1796 The Accessor class inherits property definitions from super classes.
1797 The Accessor class MAY include additional property definitions which are implementation-defined.
1798

1799 **3.5.2 Owner**

1800 **3.5.2.1 Description**

1801 An owner is a subject that can be the owner of entities.
1802 An owner of an entity MAY always have rights to update the access control list for the entity.

1803 **3.5.2.2 Class Definition**

1804 The Owner class is a mixin class which defines the characteristics of subjects such as groups and actors
1805 that can own entities.
1806 The Owner class has attribute values:

1807
1808 **localNamespace**
1809 Value: icom_ac
1810
1811 **localName**
1812 Value: Owner
1813
1814 **extendsFrom**
1815 Value: icom_ac:Accessor
1816

1817 **stereotype**
1818 Value: mixin
1819

1820 **description**
1821 Value: Owner is a mixin class which defines the characteristics of subjects such as groups and
1822 actors that can own entities.

1823
1824 **propertyDefinitions**
1825 The values for this attribute are defined in Section 3.5.2.3.

3.5.2.3 Property Definitions

The Owner class inherits property definitions from super classes.

The Owner class MAY include additional property definitions which are implementation-defined.

3.5.3 RoleDefinition

3.5.3.1 Description

A role definition is a named set of privileges.

3.5.3.2 Class Definition

The RoleDefinition class has attribute values:

localNamespace

Value: icom_ac

localName

Value: RoleDefinition

extendsFrom

Value: icom_core:EntityDefinition

stereotype

Value: primary

description

Value: A role definition is a named set of privileges.

propertyDefinitions

The values for this attribute are defined in Section 3.5.3.3.

3.5.3.3 Property Definitions

The RoleDefinition class inherits property definitions from super classes.

The RoleDefinition class MUST have the property definition:

icom_ac:privilege

Description: A set of privileges.

Required: True

Inherited: False

Property Type: icom_ac:Privilege

Cardinality: Multi

Updatability: Read Write

1865 The RoleDefinition class MAY include additional property definitions which are implementation-defined.
1866

1867 **3.5.4 Role**

1868 **3.5.4.1 Description**

1869 A role assigns a named set of privileges to a set of accessors for operations within an assigned scope.

1870 **3.5.4.2 Class Definition**

1871 The Role class has attribute values:

1872
1873 **localNamespace**
1874 Value: icom_ac
1875
1876 **localName**
1877 Value: Role
1878
1879 **extendsFrom**
1880 Value: icom_core:Subject
1881
1882 **stereotype**
1883 Value: primary
1884
1885 **description**
1886 Value: A role assigns a named set of rights to a set of accessors for operations within an
1887 assigned scope.
1888
1889 **propertyDefinitions**
1890 The values for this attribute are defined in Section 3.5.4.3.

1891 **3.5.4.3 Property Definitions**

1892 The Role class inherits property definitions from super classes.

1893 The Role class MUST have the property definitions:

1894
1895 **icom_ac:roleDefinition**
1896 Description: A role definition containing a set of privileges.
1897 Required: True
1898 Inherited: False
1899 Property Type: icom_ac:RoleDefinition
1900 Cardinality: Single
1901 Updatability: On Create
1902
1903 **icom_ac:assignedScope**
1904 Description: A scope in which a role is assigned.

1905	Required:	True
1906	Inherited:	False
1907	Property Type:	icom_core:Scope
1908	Cardinality:	Single
1909	Updatability:	Read Write

1910

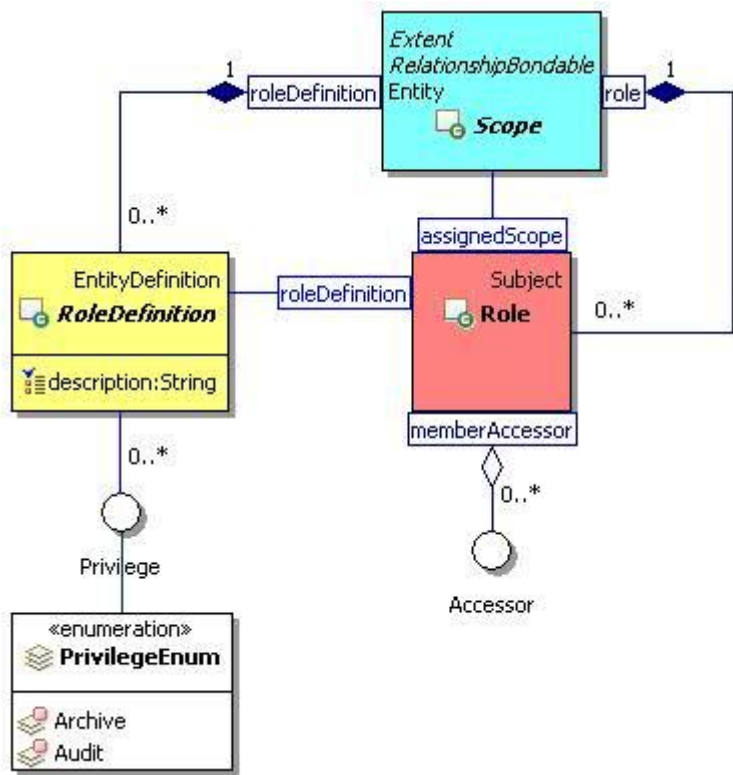
1911 **icom_ac:memberAccessor**

1912	Description:	Accessors (actors and groups) assigned to a role.
1913	Required:	False
1914	Inherited:	False
1915	Property Type:	icom_ac:Accessor
1916	Cardinality:	Multi
1917	Updatability:	Read Write

1918

1919 The Role class MAY include additional property definitions which are implementation-defined.

1920



1921

1922 Figure 16: Role Definition and Role Class Diagram.

1923

1924

3.5.5 Privilege

1925

3.5.5.1 Description

1926 A privilege is an access right granted through roles.

3.5.5.2 Class Definition

The Privilege class is a mixin class which defines access rights that can be included in role definitions.

The Privilege class has attribute values:

localNamespace

Value: icom_ac

localName

Value: Privilege

extendsFrom

Value:

stereotype

Value: mixin

description

Value: Privilege is a mixin class which defines access rights that can be included in role definitions.

propertyDefinitions

The values for this attribute are defined in Section 3.5.5.3.

3.5.5.3 Property Definitions

The Privilege class MAY include additional property definitions which are implementation-defined.

3.5.6 PrivilegeEnum

The PrivilegeEnum class is an enum class that enumerates the instances each of which expresses a privilege that can be assigned to a role.

The PrivilegeEnum class has attribute values:

localNamespace

Value: icom_ac

localName

Value: PrivilegeEnum

extendsFrom

Value: icom_ac:Privilege

stereotype

Value: primary

1968

1969 **isEnumeration**

1970 Value: TRUE

1971

1972 **description**

1973 Value: Privilege that can be assigned to a role.

1974

1975 **instances**

1976 Value: <icom_ac:Archive, icom_ac:Audit>

1977

1978 ICOM defines two privileges:

1979 • **icom_ac:Archive** a right to archive contents in a scope.

1980 • **icom_ac:Audit** a right to audit activities in a scope.

1981

1982 **3.5.7 AccessControlList**

1983 **3.5.7.1 Description**

1984 An access control list (ACL) is an object attached to an entity to specify a list of permissions to access the

1985 entity.

1986 **3.5.7.2 Class Definition**

1987 The AccessControlList class has attribute values:

1988

1989 **localNamespace**

1990 Value: icom_ac

1991

1992 **localName**

1993 Value: AccessControlList

1994

1995 **extendsFrom**

1996 Value:

1997

1998 **stereotype**

1999 Value: primary

2000

2001 **description**

2002 Value: An access control list (ACL) is an object attached to an entity to specify a list of

2003 permissions to access the entity.

2004

2005 **propertyDefinitions**

2006 The values for this attribute are defined in Section 3.5.7.3.

3.5.7.3 Property Definitions

The AccessControlList class MUST have the property definitions:

icom_ac:object

Description:	Associated object.
Required:	True
Inherited:	False
Property Type:	icom_core:Entity
Cardinality:	Single
Updatability:	On Create

icom_ac:accessControlEntry

Description:	One or more access control entries.
Required:	True
Inherited:	False
Property Type:	icom_ac:AccessControlEntry
Cardinality:	Multi
Updatability:	Read Write

AccessControlList class MAY include additional property definitions which are implementation-defined.

3.5.8 AccessControlEntry

3.5.8.1 Description

An access control entry specifies access types granted to or denied for an accessor.

3.5.8.2 Class Definition

The AccessControlEntry class has attribute values:

localNamespace

Value: icom_ac

localName

Value: AccessControlEntry

extendsFrom

Value:

stereotype

Value: primary

2046 **description**
2047 Value: An access control entry is associated with an accessor and contains a list of access
2048 types (permissions) granted to or denied from the accessor.

2049
2050 **propertyDefinitions**
2051 The values for this attribute are defined in Section 3.5.8.3.

2052 **3.5.8.3 Property Definitions**

2053 The AccessControlEntry class MUST have the property definitions:

2054
2055 **icom_ac:subject**
2056 Description: Associated subject.
2057 Required: True
2058 Inherited: False
2059 Property Type: icom_ac:Accessor
2060 Cardinality: Single
2061 Updatability: On Create

2062
2063 **icom_ac:grant**
2064 Description: One or more access types granted to a subject.
2065 Required: False
2066 Inherited: False
2067 Property Type: icom_ac:AccessType
2068 Cardinality: Multi
2069 Updatability: Read Write

2070
2071 **icom_ac:deny**
2072 Description: One or more access type denied for a subject.
2073 Required: False
2074 Inherited: False
2075 Property Type: icom_ac:AccessType
2076 Cardinality: Multi
2077 Updatability: Read Write

2078
2079 The AccessControlEntry class MAY include additional property definitions which are implementation-
2080 defined.

2081

2082 **3.5.9 AccessType**

2083 An AccessType is an access right granted through an access control entry.

2084 **3.5.9.1 Class Definition**

2085 The AccessType class is a mixin class which defines access rights that can be granted or denied in an
2086 access control entry.

2087 The AccessType class has attribute values:

2088

2089 **localNamespace**

2090 Value: icom_ac

2091

2092 **localName**

2093 Value: AccessType

2094

2095 **extendsFrom**

2096 Value:

2097

2098 **stereotype**

2099 Value: mixin

2100

2101 **description**

2102 Value: AccessType is a mixin class which defines access rights that can be granted or denied in
2103 an access control entry.

2104

2105 **propertyDefinitions**

2106 The values for this attribute are defined in Section 3.5.9.2.

2107 3.5.9.2 Property Definitions

2108 The AccessType class inherits property definitions from super classes.

2109 The AccessType class MAY include additional property definitions which are implementation-defined.

2110

2111 3.5.10 AccessTypeEnum

2112 The AccessTypeEnum class is an enum class that enumerates the instances each of which expresses an
2113 access type that can be granted or denied in an access control entry.

2114 The AccessTypeEnum class has attribute values:

2115

2116 **localNamespace**

2117 Value: icom_ac

2118

2119 **localName**

2120 Value: AccessTypeEnum

2121

2122 **extendsFrom**

2123 Value: icom_ac:AccessType

2124

2125 **stereotype**

2126 Value: primary

2127

2128 **isEnumeration**
 2129 Value: TRUE
 2130
 2131 **description**
 2132 Value: Access type that can be granted or denied in an access control entry.
 2133
 2134 **instances**
 2135 Value: <icom_ac:Read, icom_ac:Write, icom_ac>Delete>
 2136

2137 ICOM defines three access types:

- 2138 • **icom_ac:Read** a right to retrieve an entity.
- 2139 • **icom_ac:Write** a right to update an entity.
- 2140 • **icom_ac>Delete** a right to delete an entity.

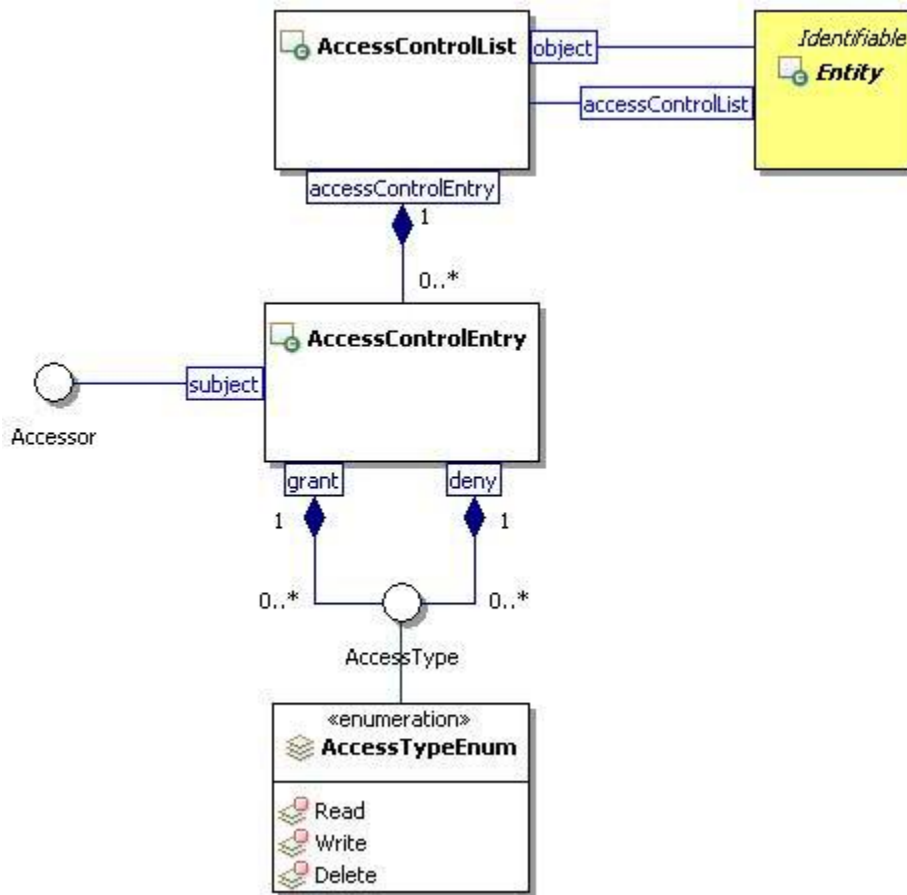


Figure 17: Access Control List Class Diagram.

3.6 Metadata Model

3.6.1 PropertyDefinition

3.6.1.1 Description

A property definition specifies the name, type, choice, and cardinality of values for properties.

3.6.1.2 Class Definition

The PropertyDefinition class has attribute values:

localNamespace

Value: icom_meta

localName

Value: PropertyDefinition

extendsFrom

Value: icom_core:Identifiable

stereotype

Value: primary

description

Value: A property definition specifies the name, type, choice, and cardinality of values for properties.

propertyDefinitions

The values for this attribute are defined in Section 3.6.1.3.

3.6.1.3 Property Definitions

The PropertyDefinition class inherits property definitions from super classes.

The PropertyDefinition class MUST have the property definitions:

icom_core:namespace

Description: Namespace for a property name.

Required: False

Inherited: False

Property Type: String

Cardinality: Single

Updatability: Read Write

icom_core:name

Description: Name for a property.

2184	Required:	True
2185	Inherited:	False
2186	Property Type:	String
2187	Cardinality:	Single
2188	Updatability:	Read Write
2189		
2190	icom_core:description	
2191	Description:	A description of a property definition.
2192	Required:	False
2193	Inherited:	False
2194	Property Type:	String
2195	Cardinality:	Single
2196	Updatability:	Read Write
2197		
2198	icom_meta:propertyType	
2199	Description:	Type of a property.
2200	Required:	True
2201	Inherited:	False
2202	Property Type:	icom_meta:PropertyType
2203	Cardinality:	Single
2204	Updatability:	On Create
2205	Choices:	{PropertyChoiceType}
2206	Open Choice:	False
2207		
2208	Note: The notation {PropertyChoiceType} represents a set of PropertyChoiceType.	
2209		
2210	icom_meta:defaultValue	
2211	Description:	A default value for a property.
2212	Required:	False
2213	Inherited:	False
2214	Property Type:	property-type
2215	Cardinality:	Single
2216	Updatability:	Read Write
2217		
2218	icom_meta:choice	
2219	Description:	An allowed value for a property.
2220	Required:	False
2221	Inherited:	False
2222	Property Type:	icom_meta:PropertyChoiceType
2223	Cardinality:	Multi
2224	Updatability:	Read Write
2225		

2226 **icom_meta:cardinality**

2227 Description: Cardinality of a property specifying whether the property can

2228 have “zero or one” or “zero or more” values.

2229 Required: True

2230 Inherited: False

2231 Property Type: icom_meta:Cardinality

2232 Cardinality: Single

2233 Updatability: On Create

2234

2235 **icom_meta:minValue**

2236 Description: Minimum value for an integer or decimal property.

2237 Required: False

2238 Inherited: False

2239 Property Type: Integer | Decimal

2240 Cardinality: Single

2241 Updatability: Read Write

2242

2243 **icom_meta:maxValue**

2244 Description: Maximum value for an integer or decimal property.

2245 Required: False

2246 Inherited: False

2247 Property Type: Integer | Decimal

2248 Cardinality: Single

2249 Updatability: Read Write

2250

2251 The PropertyDefinition class MAY include additional property definitions which are implementation-

2252 defined.

2253

2254 **3.6.2 Property**

2255 **3.6.2.1 Description**

2256 The property holds a property value.

2257 **3.6.2.2 Class Definition**

2258 The Property class has attribute values:

2259

2260 **localNamespace**

2261 Value: icom_meta

2262

2263 **localName**

2264 Value: Property

2265

2266 **extendsFrom**
 2267 Value:
 2268
 2269 **stereotype**
 2270 Value: primary
 2271
 2272 **description**
 2273 Value: A property value.
 2274
 2275 **propertyDefinitions**
 2276 The values for this attribute are defined in Section 3.6.2.3.

2277 **3.6.2.3 Property Definitions**

2278 The Property class MUST have the property definitions:

2279
 2280 **icom_meta:propertyDefinition**
 2281 Description: A property definition that specifies the name, type, and
 2282 cardinality of a property.
 2283 Required: True
 2284 Inherited: False
 2285 Property Type: icom_meta:PropertyDefinition
 2286 Cardinality: Single
 2287 Updatability: On Create
 2288
 2289 **icom_meta:value**
 2290 Description: A value of a property.
 2291 Required: True
 2292 Inherited: False
 2293 Property Type: **property-type**
 2294 Cardinality: Single
 2295 Updatability: Read Write

2296
 2297 The Property class MAY include additional property definitions which are implementation-defined.
 2298

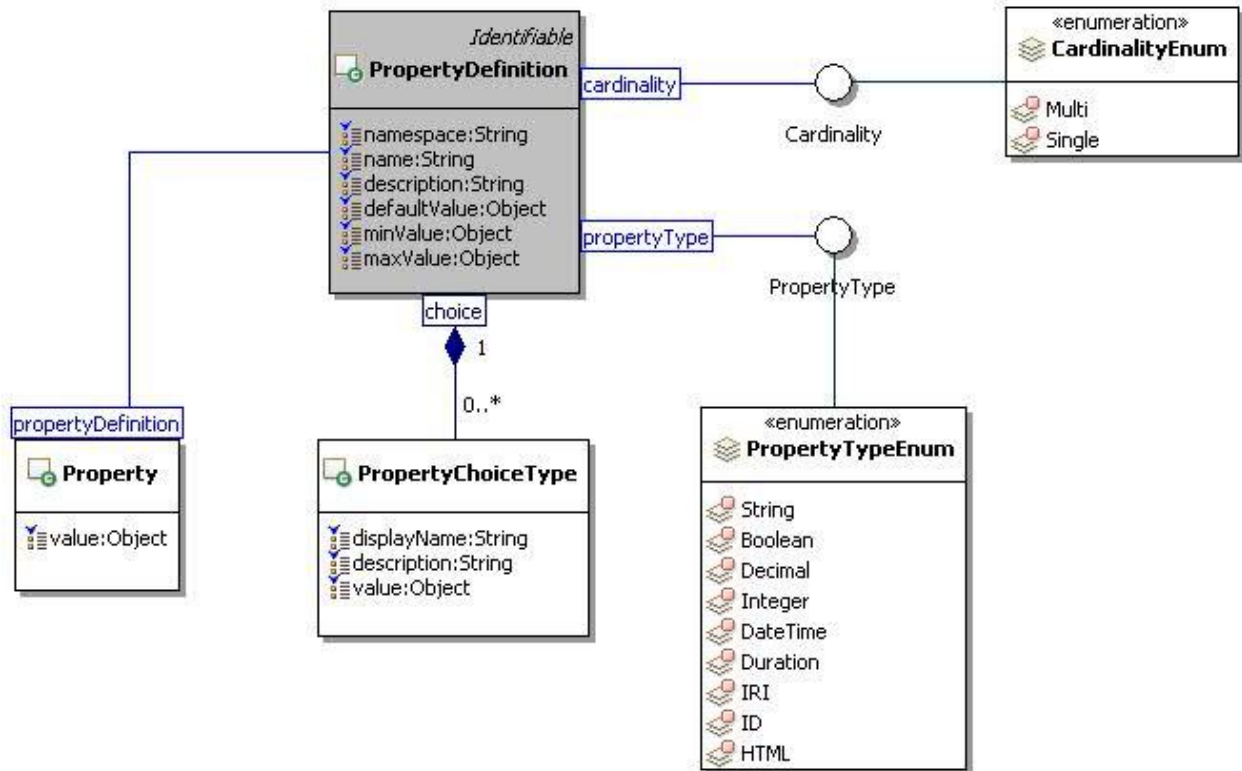


Figure 18: Property Definition and Property Class Diagram.

3.6.3 PropertyChoiceType

3.6.3.1 Description

The property choice type represents a value choice for a property. Each choice includes a display name to be used for presentation purpose and a value to be stored in a property when a choice is selected.

3.6.3.2 Class Definition

The PropertyChoiceType class has attribute values:

localNamespace

Value: icom_meta

localName

Value: PropertyChoiceType

extendsFrom

Value:

stereotype

Value: primary

2321 **description**
 2322 Value: A choice for a property value.
 2323
 2324 **propertyDefinitions**
 2325 The values for this attribute are defined Section 3.6.3.3.

2326 3.6.3.3 Property Definitions

2327 The PropertyChoiceType class MUST have the property definitions:

2328
 2329 **icom_core:description**
 2330 Description: A description of a property choice.
 2331 Required: False
 2332 Inherited: False
 2333 Property Type: String
 2334 Cardinality: Single
 2335 Updatability: Read Write

2336
 2337 **icom_meta:displayName**
 2338 Description: Display name of a property choice.
 2339 Required: True
 2340 Inherited: False
 2341 Property Type: String
 2342 Cardinality: Single
 2343 Updatability: Read Write

2344
 2345 **icom_meta:value**
 2346 Description: A value of a property choice.
 2347 Required: True
 2348 Inherited: False
 2349 Property Type: **property-type**
 2350 Cardinality: Single
 2351 Updatability: Read Write

2352
 2353 The PropertyChoiceType class MAY include additional property definitions which are implementation-
 2354 defined.

2356 3.6.4 PropertyType

2357 A PropertyType expresses a name of a **property-type**.

2358 3.6.4.1 Class Definition

2359 The PropertyType class is a mixin class which expresses a name of a **property-type**.

2360 The PropertyType class has attribute values:

2361

2362 **localNamespace**

2363 Value: icom_meta

2364

2365 **localName**

2366 Value: PropertyType

2367

2368 **extendsFrom**

2369 Value:

2370

2371 **stereotype**

2372 Value: mixin

2373

2374 **description**

2375 Value: PropertyType is a mixin class which expresses a name of a **property-type**.

2376

2377 **propertyDefinitions**

2378 The values for this attribute are defined in Section 3.6.4.2.

2379 **3.6.4.2 Property Definitions**

2380 The PropertyType class MAY include additional property definitions which are implementation-defined.

2381

2382 **3.6.5 PropertyTypeEnum**

2383 The PropertyTypeEnum class is an enum class that enumerates the instances each of which expresses

2384 the name of a **property-type**.

2385 The PropertyTypeEnum class has attribute values:

2386

2387 **localNamespace**

2388 Value: icom_meta

2389

2390 **localName**

2391 Value: PropertyTypeEnum

2392

2393 **extendsFrom**

2394 Value: PropertyType

2395

2396 **stereotype**

2397 Value: primary

2398

2399 **isEnumeration**

2400 Value: TRUE

2401

2402 **description**

2403 Value: Name of a basic data type.

2404

2405 **instances**

2406 Value: <icom_meta:String, icom_meta:Boolean, icom_meta:Decimal, icom_meta:Integer,

2407 icom_meta:Datetime, icom_meta:Duration, icom_meta:IRI, icom_meta:ID, icom_meta:HTML>

2408

2409 ICOM defines nine data types:

2410 • **icom_meta:String** is equivalent to XML schema type **xsd:string**.

2411 • **icom_meta:Boolean** is equivalent to XML schema type **xsd:boolean**.

2412 • **icom_meta:Decimal** is equivalent to XML schema type **xsd:decimal**.

2413 • **icom_meta:Integer** is equivalent to XML schema type **xsd:integer**.

2414 • **icom_meta:Datetime** is equivalent to XML schema type **xsd:dateTime**.

2415 • **icom_meta:Duration** is equivalent to XML schema type **xsd:duration**.

2416 • **icom_meta:IRI** is equivalent to XML schema type **xsd:anyURI**.

2417 • **icom_meta:ID** opaque object identifiers.

2418 • **icom_meta:HTML** documents or fragments of Hypertext Markup Language (HTML) content

2419

2420 Note: ICOM uses basic data types defined by “XML Schema Part 2: Datatypes Second Edition” (W3C

2421 Recommendation, 28 October 2004, <http://www.w3.org/TR/xmlschema-2/>).

2422

2423 3.6.6 Cardinality

2424 3.6.6.1 Description

2425 Cardinality specifies whether a property is single or multi valued.

2426 3.6.6.2 Class Definition

2427 The Cardinality class is a mixin class which defines whether a property is single or multi valued.

2428 The Cardinality class has attribute values:

2429

2430 **localNamespace**

2431 Value: icom_meta

2432

2433 **localName**

2434 Value: Cardinality

2435

2436 **extendsFrom**

2437 Value:

2438

2439 **stereotype**

2440 Value: mixin

2441

2442 **description**
2443 Value: Cardinality is a mixin class which defines whether a property is single or multi valued.
2444
2445 **propertyDefinitions**
2446 The values for this attribute are defined in Section 3.6.6.3.

2447 3.6.6.3 Property Definitions

2448 The Cardinality class MAY include additional property definitions which are implementation-defined.
2449

2450 3.6.7 CardinalityEnum

2451 The CardinalityEnum class is an enum class that enumerates instances each of which expresses the
2452 cardinality of a property.

2453 The CardinalityEnum has attribute values:

2454
2455 **localNamespace**
2456 Value: icom_meta
2457
2458 **localName**
2459 Value: CardinalityEnum
2460
2461 **extendsFrom**
2462 Value: Cardinality
2463
2464 **stereotype**
2465 Value: primary
2466
2467 **isEnumeration**
2468 Value: TRUE
2469
2470 **description**
2471 Value: Cardinality of a property.
2472
2473 **instances**
2474 Value: <icom_meta:Single, icom_meta:Multi>

2475
2476 ICOM defines two cardinality types:
2477

- 2478 • **icom_meta:Single** a property can have zero or one value (if property is not required), or exactly
2479 one value (if property is required).
- 2480 • **icom_meta:Multi** a property can have zero or more values (if property is not required), or one or
2481 more values (if property is required).

3.6.8 Marker and Subclasses

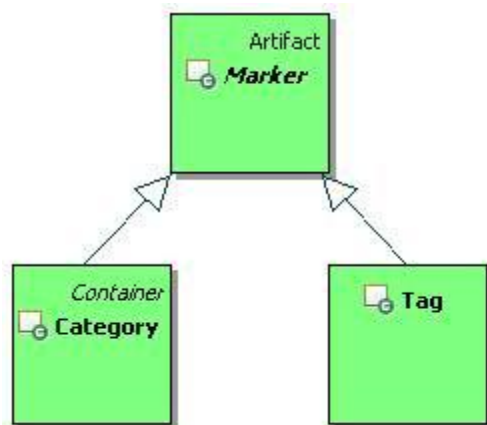


Figure 19: Marker Branch.

Figure 19 depicts the main classes of Marker Branch, which includes Marker, Category, and Tag.

3.6.9 Marker

3.6.9.1 Description

A marker is an artifact that groups together entities by a criterion. Markers can be flat or hierarchical. Flat markers are modeled by tag and hierarchical markers are modeled by category.

Note: In some cases when a user applies a marker to an entity, the marker application should be private such that only the user who applies the marker can browse or locate the entity through the marker. This is especially the case when markers are created by a user and visible only to the user who created them.

3.6.9.2 Class Definition

The Marker class has attribute values:

localNamespace

Value: icom-meta

localName

Value: Marker

extendsFrom

Value: icom_core:Artifact

stereotype

Value: primary

isAbstract

Value: TRUE

description

Value: A marker is an artifact that groups together entities by a criterion.

propertyDefinitions

The values for this attribute are defined in Section 3.6.9.3.

3.6.9.3 Property Definitions

The Marker class inherits property definitions from super classes.
The Marker class MUST have the property definition:

icom_meta:markedEntity

Description:	A marked entity.
Required:	False
Inherited:	False
Property Type:	icom_core:Entity
Cardinality:	Multi
Updatability:	Read Only

The Marker class MAY include additional property definitions which are implementation-defined.

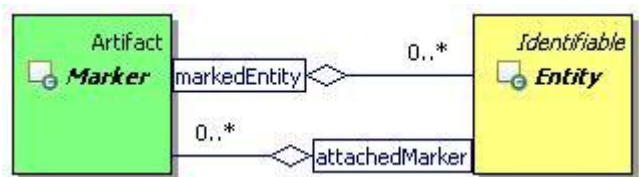


Figure 20: Marker Class Diagram.

3.6.10 Category

3.6.10.1 Description

A category is a marker that classifies entities.

3.6.10.2 Class Definition

The Category class has attribute values:

localNamespace

Value: icom_meta

localName

Value: Category

extendsFrom

Value: icom_meta:Marker, icom_core:Container

2548 **stereotype**
 2549 Value: primary
 2550
 2551 **description**
 2552 Value: A category is a marker that classifies entities.
 2553
 2554 **propertyDefinitions**
 2555 The values for this attribute are defined in Section 3.6.10.3.

2556 **3.6.10.3 Property Definitions**

2557 The Category class inherits property definitions from super classes.
 2558 The Category class MUST have the property definitions:

2559
 2560 **icom_meta:superCategory**
 2561 Description: A super category.
 2562 Required: False
 2563 Inherited: False
 2564 Property Type: icom_meta:Category
 2565 Cardinality: Single
 2566 Updatability: Read Only

2567
 2568 **icom_meta:subcategory**
 2569 Description: Zero or more sub categories.
 2570 Required: False
 2571 Inherited: False
 2572 Property Type: icom_meta:Category
 2573 Cardinality: Multi
 2574 Updatability: Read Only

2575
 2576 **icom_meta:abstract**
 2577 Description: Indicates whether a category is abstract or concrete.
 2578 Required: False
 2579 Inherited: False
 2580 Property Type: Boolean
 2581 Cardinality: Single
 2582 Updatability: Read Write

2583
 2584 **icom_meta:propertyDefinition**
 2585 Description: Optional or mandatory properties for a category application.
 2586 Required: False
 2587 Inherited: False
 2588 Property Type: icom_meta:PropertyDefinition
 2589 Cardinality: Multi

Updatability: Read Write

The Category class MAY include additional property definitions which are implementation-defined.

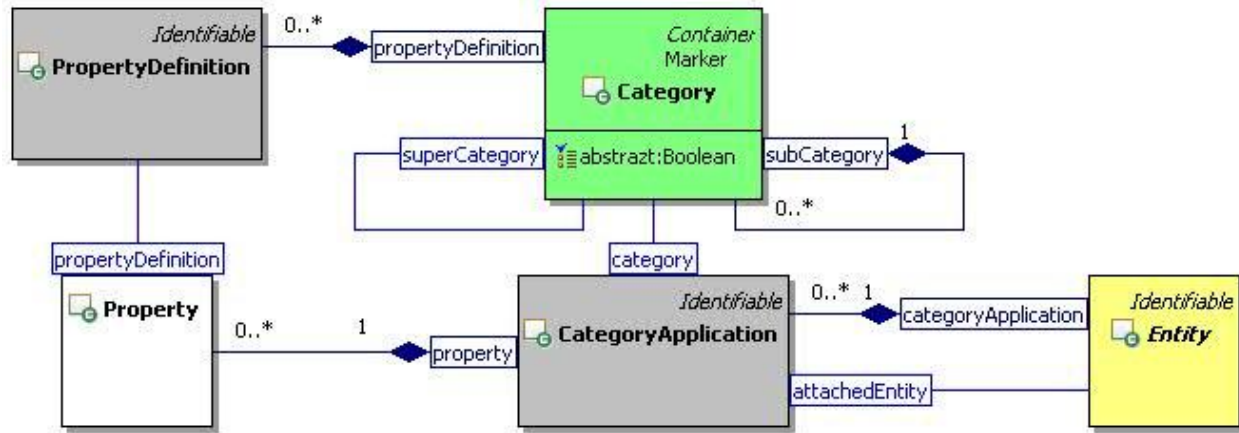


Figure 21: Category and Category Application Class Diagram.

3.6.11 CategoryApplication

3.6.11.1 Description

A category application is an instance of association between a category and a specific entity.

3.6.11.2 Class Definition

The CategoryApplication class has attribute values:

localNamespace

Value: icom_meta

localName

Value: CategoryApplication

extendsFrom

Value: icom_core:Identifiable

stereotype

Value: primary

description

Value: A category application is an instance of association between a category and a specific entity.

2619 **propertyDefinitions**

2620 The values for this attribute are defined in Section 3.6.11.3.

2621 **3.6.11.3 Property Definitions**

2622 The CategoryApplication class inherits property definitions from super classes.

2623 The CategoryApplication class MUST have the property definitions:

2624

2625 **icom_meta:attachedEntity**

2626 Description: An entity to which a category is applied.

2627 Required: True

2628 Inherited: False

2629 Property Type: icom_core:Entity

2630 Cardinality: Single

2631 Updatability: On Create

2632

2633 **icom_meta:category**

2634 Description: A category which is applied on an entity.

2635 Required: True

2636 Inherited: False

2637 Property Type: icom_meta:Category

2638 Cardinality: Single

2639 Updatability: On Create

2640

2641 **icom_meta:property**

2642 Description: Zero or more properties.

2643 Required: False

2644 Inherited: False

2645 Property Type: icom_meta:Property

2646 Cardinality: Multi

2647 Updatability: Read Write

2648

2649 The CategoryApplication class MAY include additional property definitions which are implementation-
2650 defined.

2651

2652 **3.6.12 Tag**

2653 **3.6.12.1 Description**

2654 A tag is a marker that labels entities by a keyword.

2655 **3.6.12.2 Class Definition**

2656 The Tag class has attribute values:

2657

2658 **localNamespace**
 2659 Value: icom_meta
 2660
 2661 **localName**
 2662 Value: Tag
 2663
 2664 **extendsFrom**
 2665 Value: icom_meta:Marker
 2666
 2667 **stereotype**
 2668 Value: primary
 2669
 2670 **description**
 2671 Value: A tag is a marker that labels entities by a keyword.
 2672
 2673 **propertyDefinitions**
 2674 The values for this attribute are defined in Section 3.6.12.3.

2675 **3.6.12.3 Property Definitions**

2676 The Tag class inherits property definitions from super classes.

2677 The Tag class MUST have the property definition:

2678
 2679 **icom_meta:applicationCount**
 2680 Description: An estimate of the number of times a tag is applied on
 2681 entities.
 2682 Required: False
 2683 Inherited: False
 2684 Property Type: Integer
 2685 Cardinality: Single
 2686 Updatability: Read Only

2687
 2688 The Tag class MAY include additional property definitions which are implementation-defined.
 2689

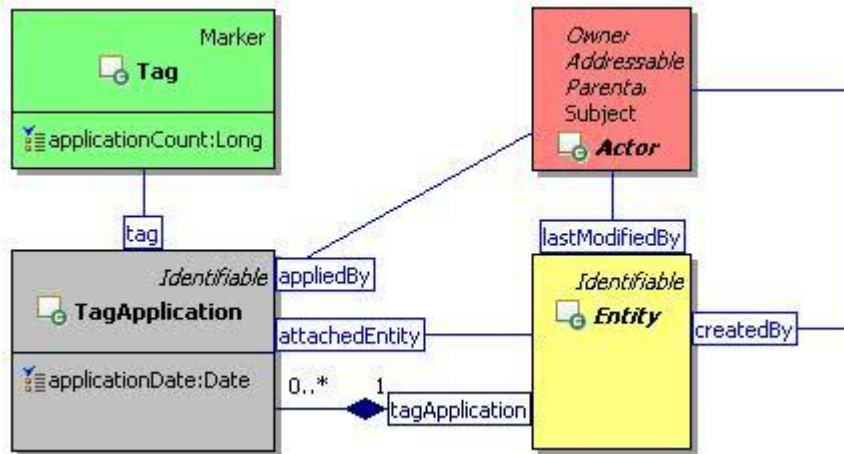


Figure 22: Tag and Tag Application Class Diagram.

3.6.13 TagApplication

3.6.13.1 Description

A tag application is an instance of association between a tag and a specific entity.

3.6.13.2 Class Definition

The TagApplication class has attribute values:

localNamespace

Value: icom_meta

localName

Value: TagApplication

extendsFrom

Value: icom_core:Identifiable

stereotype

Value: primary

description

Value: A tag application is an instance of association between a tag and a specific entity.

propertyDefinitions

The values for this attribute are defined in Section 3.6.13.3.

3.6.13.3 Property Definitions

The TagApplication class inherits property definitions from super classes.

The TagApplication class MUST have the property definitions:

2719		
2720	icom_meta:attachedEntity	
2721	Description:	An entity on which a tag is applied.
2722	Required:	True
2723	Inherited:	False
2724	Property Type:	icom_core:Entity
2725	Cardinality:	Single
2726	Updatability:	On Create
2727		
2728	icom_meta:tag	
2729	Description:	A tag which is applied to an entity.
2730	Required:	True
2731	Inherited:	False
2732	Property Type:	icom_meta:Tag
2733	Cardinality:	Single
2734	Updatability:	On Create
2735		
2736	icom_meta:appliedBy	
2737	Description:	A user who applies a tag to an entity.
2738	Required:	False
2739	Inherited:	False
2740	Property Type:	icom_core:Actor
2741	Cardinality:	Single
2742	Updatability:	Read Only
2743		
2744	icom_meta:applicationDate	
2745	Description:	A date and time when a tag is applied to an entity.
2746	Required:	False
2747	Inherited:	False
2748	Property Type:	DateTime
2749	Cardinality:	Single
2750	Updatability:	Read Write
2751		
2752	The TagApplication class MAY include additional property definitions which are implementation-defined.	
2753		

2754 3.6.14 RelationshipBondable

2755 3.6.14.1 Description

2756 A relationship bondable entity is an entity which may be related to other entities by a relationship.

2757 Note: A relationship can exist among entities that are not relationships.

3.6.14.2 Class Definition

The RelationshipBondable class is a mixin class which defines the characteristics of entities that may be relationship bonded. It includes almost every subclass of Entity except Relationship.

The RelationshipBondable class has attribute values:

localNamespace

Value: icom_meta

localName

Value: RelationshipBondable

extendsFrom

Value: icom_core:Identifiable

stereotype

Value: mixin

description

Value: RelationshipBondable is a mixin class which defines the characteristics of entities that can be relationship bonded.

propertyDefinitions

The values for this attribute are defined in Section 3.6.14.3.

3.6.14.3 Property Definitions

The RelationshipBondable class inherits property definitions from super classes.

The RelationshipBondable class MAY include additional property definitions which are implementation-defined.

3.6.15 RelationshipDefinition

3.6.15.1 Description

A relationship definition is an entity that defines a type of relationship, including a name and a description of the relationship type, types of source entity and target entities of a relationship, and definition of properties in a relationship.

3.6.15.2 Class Definition

The RelationshipDefinition class has attribute values:

localNamespace

Value: icom_meta

2797 **localName**
 2798 Value: RelationshipDefinition
 2799
 2800 **extendsFrom**
 2801 Value: icom_core:EntityDefinition
 2802
 2803 **stereotype**
 2804 Value: primary
 2805
 2806 **description**
 2807 Value: A relationship definition is an entity that defines a type of relationship.
 2808
 2809 **propertyDefinitions**
 2810 The values for this attribute are defined in Section 3.6.15.3.

2811 **3.6.15.3 Property Definitions**

2812 The RelationshipDefinition class inherits property definitions from super classes.

2813 The RelationshipDefinition class MUST have the property definitions:

2814
 2815 **icom_meta:propertyDefinition**
 2816 Description: Optional or mandatory properties for a relationship.
 2817 Required: False
 2818 Inherited: False
 2819 Property Type: icom_meta:PropertyDefinition
 2820 Cardinality: Multi
 2821 Updatability: Read Write
 2822
 2823 **icom_meta:allowedSourceType**
 2824 Description: A list of expanded names of relationship bondable classes,
 2825 indicating that the source entity of a relationship MUST be an
 2826 instance of a class in the list.
 2827 Required: False
 2828 Inherited: False
 2829 Property Type: IRI
 2830 Cardinality: Multi
 2831 Updatability: Read Write
 2832
 2833 **icom_meta:allowedTargetType**
 2834 Description: A list of expanded names of relationship bondable classes,
 2835 indicating that the target entity of a relationship MUST be an
 2836 instance of a class in the list.
 2837 Required: False
 2838 Inherited: False
 2839 Property Type: IRI

2840	Cardinality:	Multi
2841	Updatability:	Read Write
2842		
2843	The RelationshipDefinition class MAY include additional property definitions which are implementation-	
2844	defined.	
2845		

2846 3.6.16 Relationship

2847 3.6.16.1 Description

2848 A relationship is an entity that relates a set of entities by a predicate.

2849 3.6.16.2 Class Definition

2850 The Relationship class has attribute values:

2851		
2852	localNamespace	
2853	Value:	icom_meta
2854		
2855	localName	
2856	Value:	Relationship
2857		
2858	extendsFrom	
2859	Value:	icom_core:Entity
2860		
2861	stereotype	
2862	Value:	primary
2863		
2864	description	
2865	Value:	A relationship is an entity that relates a set of entities by a predicate.
2866		
2867	propertyDefinitions	
2868		The values for this attribute are defined in Section 3.6.16.3.

2869 3.6.16.3 Property Definitions

2870 The Relationship class inherits property definitions from super classes.

2871 The Relationship class MUST have the property definitions:

2872		
2873	icom_meta:relationshipDefinition	
2874	Description:	A definition of relationships.
2875	Required:	True
2876	Inherited:	False
2877	Property Type:	icom_meta:RelationshipDefinition
2878	Cardinality:	Single
2879	Updatability:	On Create

2880		
2881	icom_meta:sourceEntity	
2882	Description:	A source entity of a relationship.
2883	Required:	True
2884	Inherited:	False
2885	Property Type:	icom_meta:RelationshipBondable
2886	Cardinality:	Single
2887	Updatability:	On Create
2888		
2889	icom_meta:targetEntity	
2890	Description:	One or more target entities of a relationship.
2891	Required:	True
2892	Inherited:	False
2893	Property Type:	icom_meta:RelationshipBondable
2894	Cardinality:	Multi
2895	Updatability:	Read Write
2896		
2897	icom_meta:property	
2898	Description:	Zero or more properties.
2899	Required:	False
2900	Inherited:	False
2901	Property Type:	icom_meta:Property
2902	Cardinality:	Multi
2903	Updatability:	Read Write
2904		
2905	The Relationship class MAY include additional property definitions which are implementation-defined.	
2906		

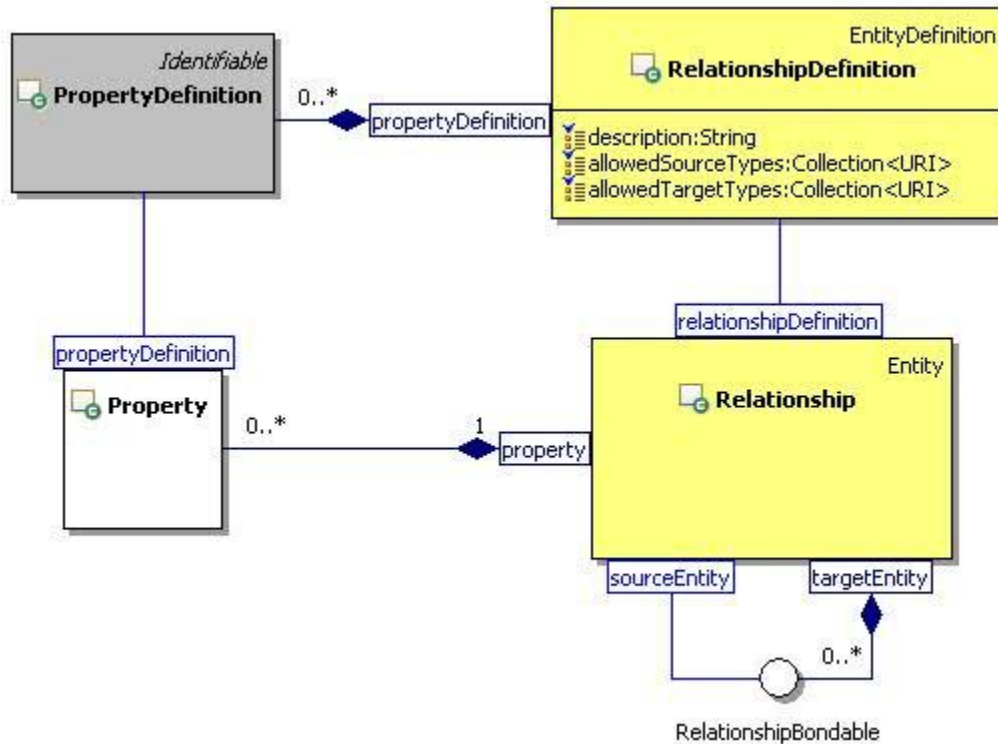


Figure 23: Relationship Class Diagram.

3.7 Common Concepts

3.7.1 Addressable

3.7.1.1 Description

An addressable object is an identifiable object that has one or more addresses.

3.7.1.2 Class Definition

The Addressable class is a mixin class which defines the characteristics of entities that has one or more addresses.

The Addressable class has attribute values:

localNamespace

Value: icom_core

localName

Value: Addressable

extendsFrom

Value: icom_core:Identifiable

2928 **stereotype**
2929 Value: mixin
2930
2931 **description**
2932 Value: Addressable is a mixin class which defines the characteristics of entities that has one or
2933 more addresses.
2934
2935 **propertyDefinitions**
2936 The values for this attribute are defined in Section 3.7.1.3.

2937 **3.7.1.3 Property Definitions**

2938 The Addressable class inherits property definitions from super classes.
2939 The Addressable class MUST have the property definitions:

2941 **icom_core:address**
2942 Description: Zero or more addresses of an addressable object.
2943 Required: False
2944 Inherited: False
2945 Property Type: icom_core:EntityAddress
2946 Cardinality: Multi
2947 Updatability: Read Write

2948
2949 **icom_core:primaryAddress**
2950 Description: The primary address of an addressable object.
2951 Required: False
2952 Inherited: False
2953 Property Type: icom_core:EntityAddress
2954 Cardinality: Single
2955 Updatability: Read Write

2956
2957 The Addressable class MAY include additional property definitions which are implementation-defined.
2958

2959 **3.7.2 EntityAddress**

2960 **3.7.2.1 Description**

2961 An entity address object represents an address which is defined by type and IRI.

2962 **3.7.2.2 Class Definition**

2963 The EntityAddress class has attribute values:

2964
2965 **localNamespace**
2966 Value: icom_core

2967

2968 **localName**
 2969 Value: EntityAddress
 2970
 2971 **extendsFrom**
 2972 Value:
 2973
 2974 **stereotype**
 2975 Value: primary
 2976
 2977 **description**
 2978 Value: An entity address object represents an address which is defined by type and IRI.
 2979
 2980 **propertyDefinitions**
 2981 The values for this attribute are defined in Section 3.7.2.3.

2982 **3.7.2.3 Property Definitions**

2983 The EntityAddress class MUST have the property definitions:
 2984

2985	icom_core:addressType	
2986	Description:	Type of an address.
2987	Required:	False
2988	Inherited:	False
2989	Property Type:	String
2990	Cardinality:	Single
2991	Updatability:	Read Write
2992		
2993	icom_core:address	
2994	Description:	A IRI representing an address.
2995	Required:	False
2996	Inherited:	False
2997	Property Type:	IRI
2998	Cardinality:	Single
2999	Updatability:	Read Write
3000		

3001 **3.7.3 Participant**

3002 **3.7.3.1 Description**

3003 A participant object represents the participation of any addressable entity in a collaboration activity such
 3004 as an occurrence, task, conference, discussion, and message.
 3005 If an addressable entity is not specified, an address must be specified.

3006 **3.7.3.2 Class Definition**

3007 The Participant class has attribute values:

3008

3009 **localNamespace**

3010 Value: icom_core

3011

3012 **localName**

3013 Value: Participant

3014

3015 **extendsFrom**

3016 Value:

3017

3018 **stereotype**

3019 Value: primary

3020

3021 **description**

3022 Value: A participant object represents the participation of any addressable entity in a

3023 collaboration activity such as an occurrence, task, conference, discussion, and message.

3024

3025 **propertyDefinitions**

3026 The values for this attribute are defined in Section 3.7.3.3.

3027 **3.7.3.3 Property Definitions**

3028 The Participant class inherits property definitions from super classes.

3029 The Participant class **MUST** have the property definitions:

3030

3031 **icom_core:participant**

3032	Description:	An addressable entity to participate in a collaboration activity.
3033	Required:	False
3034	Inherited:	False
3035	Property Type:	icom_core:Addressable
3036	Cardinality:	Single
3037	Updatability:	On Create

3038

3039 **icom_core:address**

3040	Description:	An address of a participant in a collaboration activity.
3041	Required:	False
3042	Inherited:	False
3043	Property Type:	IRI
3044	Cardinality:	Single
3045	Updatability:	On Create

3046

3047 **icom_core:name**

3048	Description:	Name of a participant in a collaboration activity.
3049	Required:	False

3050 Inherited: False
3051 Property Type: String
3052 Cardinality: Single
3053 Updatability: On Create

3054

3055 The Participant class MAY include additional property definitions which are implementation-defined.

3056

3057 **3.7.4 Priority**

3058 **3.7.4.1 Description**

3059 A priority level for delivery of information.

3060 **3.7.4.2 Class Definition**

3061 The Priority class is a mixin class which defines a priority level for delivery of information.

3062 The Priority class has attribute values:

3063

3064 **localNamespace**

3065 Value: icom_core

3066

3067 **localName**

3068 Value: Priority

3069

3070 **extendsFrom**

3071 Value:

3072

3073 **stereotype**

3074 Value: mixin

3075

3076 **description**

3077 Value: Priority is a mixin class which defines a priority level for delivery of information.

3078

3079 **propertyDefinitions**

3080 The values for this attribute are defined in Section 3.7.4.3.

3081 **3.7.4.3 Property Definitions**

3082 The Priority class MAY include additional property definitions which are implementation-defined.

3083

3084 **3.7.5 PriorityEnum**

3085 The PriorityEnum class is an enum class that enumerates the instances each of which defines a priority
3086 level for delivery of information.

3087 The PriorityEnum has attribute values:

3088

3089 **localNamespace**
3090 Value: icom_core
3091
3092 **localName**
3093 Value: PriorityEnum
3094
3095 **extendsFrom**
3096 Value: Priority
3097
3098 **stereotype**
3099 Value: primary
3100
3101 **isEnumeration**
3102 Value: TRUE
3103
3104 **description**
3105 Value: Priority level for delivery of information.
3106
3107 **instances**
3108 Value: <icom_core:Normal, icom_core:Low, icom_core:Medium, icom_core:High>

3109
3110 ICOM defines four priorities:
3111 • **icom_core:Normal** a normal priority.
3112 • **icom_core:Low** a low priority.
3113 • **icom_core:Medium** a medium priority.
3114 • **icom_core:High** a high priority.
3115

3116 **3.7.6 DateTimeResolution**

3117 **3.7.6.1 Description**

3118 A date time resolution is a resolution of date time value.

3119 **3.7.6.2 Class Definition**

3120 The DateTimeResolution class is a mixin class which defines a resolution of date time value.

3121 The DateTimeResolution class has attribute values:

3122
3123 **localNamespace**
3124 Value: icom_core
3125
3126 **localName**
3127 Value: DateTimeResolution
3128

3129 **extendsFrom**
3130 Value:
3131
3132 **stereotype**
3133 Value: mixin
3134
3135 **description**
3136 Value: DateTimeResolution is a mixin class which defines a resolution of date time value.
3137
3138 **propertyDefinitions**
3139 The values for this attribute are defined in Section 3.7.6.3.

3140 **3.7.6.3 Property Definitions**

3141 The DateTimeResolution class MAY include additional property definitions which are implementation-
3142 defined.
3143

3144 **3.7.7 DateTimeResolutionEnum**

3145 The DateTimeResolutionEnum class is an enum class that enumerates the instances each of which
3146 expresses a resolution of a date time value.

3147 The DateTimeResolutionEnum has attribute values:

3148
3149 **localNamespace**
3150 Value: icom_core
3151
3152 **localName**
3153 Value: DateTimeResolutionEnum
3154
3155 **extendsFrom**
3156 Value: DateTimeResolution
3157
3158 **stereotype**
3159 Value: primary
3160
3161 **isEnumeration**
3162 Value: TRUE
3163
3164 **description**
3165 Value: Resolution of a date time value.
3166
3167 **instances**
3168 Value: <icom_core:Year, icom_core:Date, icom_core:Time>
3169

3170 ICOM defines three date time resolutions:
3171 • **icom_core:Year** date time resolution is in years.
3172 • **icom_core:Date** date time resolution is in years and days.
3173 • **icom_core:Time** date time resolution is in years, days, and time of day.
3174

3175 3.7.8 TimeZone

3176 3.7.8.1 Description

3177 A time zone is a region that has a uniform standard time.

3178 3.7.8.2 Class Definition

3179 The TimeZone class has attribute values:

3180
3181 **localNamespace**
3182 Value: icom_core
3183
3184 **localName**
3185 Value: TimeZone
3186
3187 **extendsFrom**
3188 Value:
3189
3190 **stereotype**
3191 Value: primary
3192
3193 **description**
3194 Value: A time zone is a region that has a uniform standard time.
3195
3196 **propertyDefinitions**
3197 The values for this attribute are defined in Section 3.7.8.3.

3198 3.7.8.3 Property Definitions

3199 The TimeZone class inherits property definitions from super classes.

3200 The TimeZone class MUST have the property definitions:

3201
3202 **icom_core:ID**
3203 Description: Identifier of a time zone.
3204 Required: False
3205 Inherited: False
3206 Property Type: String
3207 Cardinality: Single
3208 Updatability: On Create

3209		
3210	icom_core:rawOffset	
3211	Description:	An offset to add to Universal Coordinated Time (UTC) to get local time. If Daylight Saving Time is in effect at the specified date, the offset value is adjusted with the amount of daylight saving.
3212		
3213		
3214		
3215	Required:	False
3216	Inherited:	False
3217	Property Type:	Integer
3218	Cardinality:	Single
3219	Updatability:	On Create

3220

3221 The TimeZone class MAY include additional property definitions which are implementation-defined.

3222

3223 3.7.9 Location

3224 3.7.9.1 Description

3225 A location object represents a physical location which is defined by name, description, and geo
3226 coordinates.

3227 Note: The name of a location may remain unchanged while a physical location may be changing. For
3228 example, a location name might be “On an airplane” while a physical location might be the geo
3229 coordinates of a flight path or current coordinates of a plane.

3230 3.7.9.2 Class Definition

3231 The Location class has attribute values:

3232		
3233	localNamespace	
3234	Value: icom_core	
3235		
3236	localName	
3237	Value: Location	
3238		
3239	extendsFrom	
3240	Value:	
3241		
3242	stereotype	
3243	Value: primary	
3244		
3245	description	
3246	Value: A location object represents a physical location which is defined by name, description, or 3247 geo coordinates.	
3248		
3249	propertyDefinitions	
3250	The values for this attribute are defined in Section 3.7.9.3.	

3.7.9.3 Property Definitions

The Location class MUST have the property definitions:

icom_core:name

Description:	Name of a location.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

icom_core:description

Description:	A description of a location.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

icom_core:timeZone

Description:	Time zone of a location.
Required:	False
Inherited:	False
Property Type:	icom_core:TimeZone
Cardinality:	Single
Updatability:	Read Write

icom_core:coordinates

Description:	A list of geo coordinates marking a point, path, or area of a physical location.
Required:	False
Inherited:	False
Property Type:	icom_core:GeoCoordinates
Cardinality:	Multi
Updatability:	Read Write

The Location class MAY include additional property definitions which are implementation-defined.

3.7.10 GeoCoordinates

3.7.10.1 Description

A geo coordinates object specifies the latitude, longitude, and altitude of a physical location.

3.7.10.2 Class Definition

The GeoCoordinates class has attribute values:

localNamespace

Value: icom_core

localName

Value: GeoCoordinates

extendsFrom

Value:

stereotype

Value: primary

description

Value: A geo coordinates object specifies the latitude, longitude, and altitude of a physical location.

propertyDefinitions

The values for this attribute are defined in Section 3.7.10.3.

3.7.10.3 Property Definitions

The GeoCoordinates class MUST have the property definitions:

icom_core:latitude

Description: Latitude of a location.

Required: False

Inherited: False

Property Type: Float

Cardinality: Single

Updatability: Read Write

icom_core:longitude

Description: Longitude of a location.

Required: False

Inherited: False

Property Type: Float

Cardinality: Single

Updatability: Read Write

icom_core:altitude

Description: Altitude of a location.

3334	Required:	False
3335	Inherited:	False
3336	Property Type:	Float
3337	Cardinality:	Single
3338	Updatability:	Read Write

3339

3340 The GeoCoordinates class MAY include additional property definitions which are implementation-defined.

3341

4 Extension Modules

4.1 Overview of Extension Modules

Each extension module defines a model of a collaboration activity. Different models of collaboration activities in this specification include content creation, communication, coordination, discussion forum, and conference. Except for the Presence Module and Free Busy Module, the extension modules in this section introduce specialized subclasses of Artifact and Folder of Artifact Branch.

Note: ICOM Core Model (Section 3) establishes a framework to integrate specialized collaboration activities of the extension modules, which more or less represent technology or protocol channels. The framework is extensible with additional extension modules. For example, applications can adopt a model for CMIS Policy base type as a new extension module, which can be used to integrate with BPMN or BPEL processes outside the ICOM domain. An ICOM space can provide a durable context for continuity of conversations and activities related to a business process type or process instance. Some new extension modules may import the models from related standards. For example, social network model may be imported from [OpenGraph] or [OpenSocial].

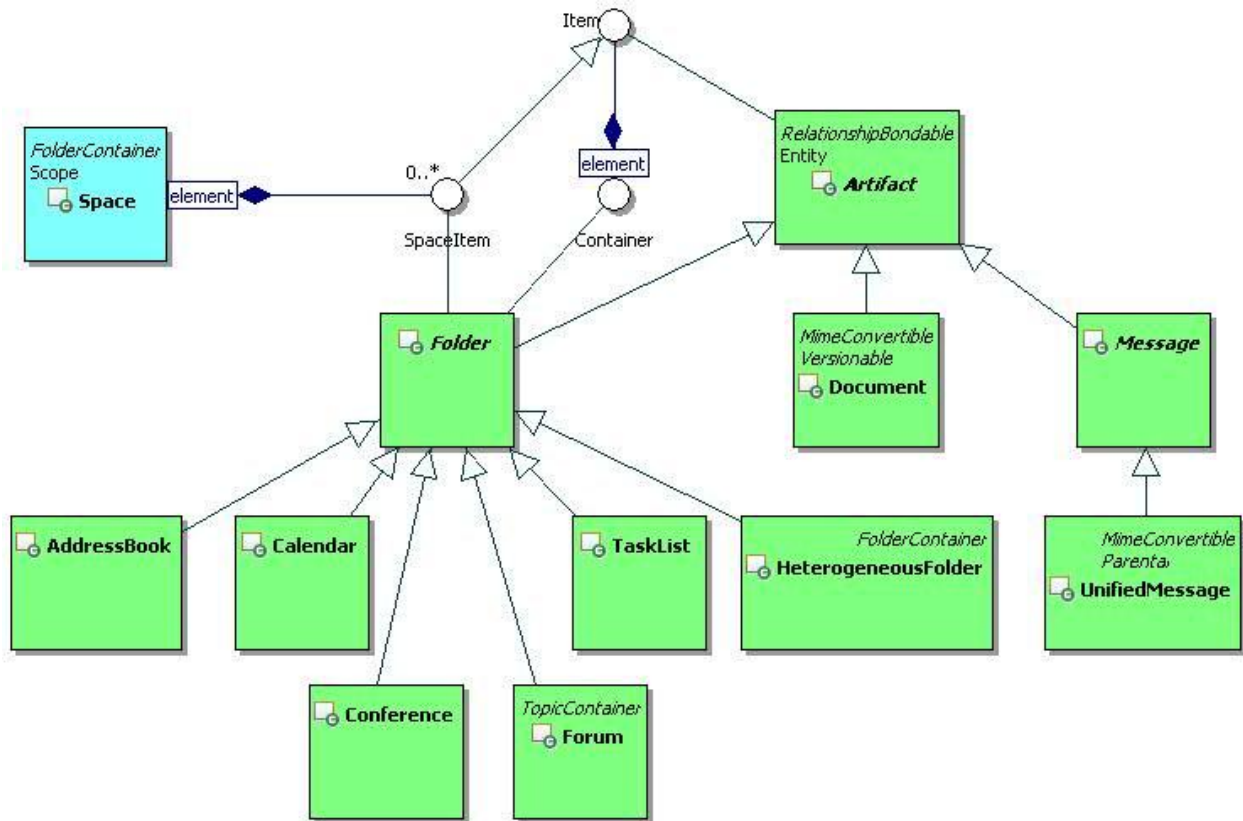


Figure 24: Containers of Collaboration Activities.

ICOM defines containers that provide contexts and structures for specific areas of collaborative activities. The UML class diagram in Figure 24 depicts a Space as a hub of containers, including HeterogeneousFolder, AddressBook, Calendar, TaskList, Forum, and Conference. These containers are briefly described as follows:

HeterogeneousFolder (defined in Core Model) is a general purpose container that can contain any type of artifacts, and therefore, can serve as

- a library of documents and wiki pages to support content sharing and co-creation,

- 3365 • an inbox or outbox for communication, or
- 3366 • a trash folder to archive all types of artifacts deleted from a space.
- 3367 **AddressBook** is a specialized container to manage contact or personal information, such as
- 3368 addresses, phone numbers, birthdays, anniversaries, and other entries.
- 3369 **Calendar** is a specialized container to support time management.
- 3370 **TaskList** is a specialized container to support task coordination.
- 3371 **Forum** is a specialized container to support
- 3372 • **Topic** sub-containers for threaded discussions and
- 3373 • **Announcement** sub-containers for time-sensitive communication.
- 3374 **Conference** is a specialized container that provides a durable context for real-time interactions.
- 3375

3376 The following ten modules are specified as extension modules of ICOM:

- 3377 1. Content Module (in section 4.2) defines Content, MultiContent, and SimpleContent. A content
- 3378 represents a piece of data in a document or message. Content, multi-content, simple content, and
- 3379 online content form a composite design pattern.
- 3380 2. Document Module (in Section 4.3) defines Document, WikiPage, and version control model. A
- 3381 document can contain a composite content defined in section 4.2. Documents are typically
- 3382 contained by heterogeneous folders.
- 3383 3. Message Module (in Section 4.4) defines Message, UnifiedMessage, InstantMessage, and
- 3384 related classes. A message can contain a composite content defined in section 4.2. Unified
- 3385 messages are typically contained by heterogeneous folders.
- 3386 4. Presence Module (in Section 4.5) defines Presence, Activity, and Contact Method. Presence
- 3387 represents a watchable state of a presentity (which is usually a person). Presence state is derived
- 3388 using an actor's subscriptions.

3389 Note: Since a Presence is derived using a viewer's subscriptions, a Presence should not be shared

3390 with other viewers. For this reason, Presence is not modeled as Entity and is not assigned an access

3391 control list.

- 3392 5. Address Book Module (in Section 4.6) defines AddressBook and PersonContact. A person
- 3393 contact can bookmark a reference to a person in an ICOM community as well as store addresses,
- 3394 phone numbers, and other entries about a person who may not be in any ICOM community.
- 3395 6. Calendar Module (in Section 4.7) defines Calendar, Occurrence, and OccurrenceSeries.
- 3396 Occurrence artifacts are used to resolve the free-busy times of participants for scheduling of
- 3397 meetings and booking of rooms and other resources.
- 3398 7. Free Busy Module (in Section 4.8) defines FreeBusy. FreeBusy is a view derived from
- 3399 occurrences in a calendar or a set of calendars using an actor's privileges to determine the free
- 3400 or busy states of calendar occurrences.

3401 Note: Since a FreeBusy view is derived using a viewer's privileges, a FreeBusy should not be shared

3402 with other viewers. For this reason, FreeBusy is not modeled as Entity and is not assigned an access

3403 control list.

- 3404 8. Task List Module (in Section 4.9) defines TaskList and Task. Tasks are used to coordinate the
- 3405 assignment of tasks and to track the progress of task activities.
- 3406 9. Forum Module (in Section 4.10) defines Forum, Topic, Announcement, and DiscussionMessage.
- 3407 Topics, announcements, and discussions are used for treaded discussions. Moderators of a
- 3408 forum can prune, merge, or fork the discussion threads.
- 3409 10. Conference Module (in Section 4.11) defines Conference and related classes. A conference can
- 3410 contain visual, audio, and chat transcripts of the conference sessions. It also contains the current
- 3411 status, conference settings, past sessions, active session, and activity logs.

4.2 Content Module

4.2.1 MimeConvertible

4.2.1.1 Description

A MimeConvertible object represents an object that has Multipurpose Internet Mail Extensions (MIME) characteristics such as headers, content transfer encoding, and possible hierarchy of sub-contents.

4.2.1.2 Class Definition

The MimeConvertible class is a mixin class that defines the characteristics of objects that can be represented in MIME format.

The MimeConvertible class has attribute values:

localNamespace

Value: icom_content

localName

Value: MimeConvertible

extendsFrom

Value: icom_core:Identifiable

stereotype

Value: mixin

description

Value: MimeConvertible class is a mixin class that defines the characteristics of objects that can be represented in MIME format.

propertyDefinitions

The values for this attribute are defined in Section 4.2.1.3.

4.2.1.3 Property Definitions

The MimeConvertible class inherits property definitions from super classes.

The MimeConvertible class MAY include additional property definitions which are implementation-defined.

4.2.2 Content

4.2.2.1 Description

A content object represents a piece of data in a document or message. Content, multi-content, simple content, and online content form a composite design pattern.

4.2.2.2 Class Definition

The Content class has attribute values:

3450

3451 **localNamespace**

3452 Value: icom_content

3453

3454 **localName**

3455 Value: Content

3456

3457 **extendsFrom**

3458 Value: icom_core:Identifiable, icom_content:MimeConvertible

3459

3460 **stereotype**

3461 Value: primary

3462

3463 **isAbstract**

3464 Value: TRUE

3465

3466 **description**

3467 Value: Content represents a piece of data in a document or message.

3468

3469 **propertyDefinitions**

3470 The values for this attribute are defined in Section 4.2.2.3.

3471 **4.2.2.3 Property Definitions**

3472 The Content class inherits property definitions from super classes.

3473 The Content class **MUST** have the property definitions:

3474

3475 **icom_content:contentId**

3476 Description:	A content id is a unique identifier for a part of content in multi-
3477 Required:	False
3478 Inherited:	False
3479 Property Type:	String
3480 Cardinality:	Single
3481 Updatability:	Read Write

3483

3484 **icom_content:mediaType**

3485 Description:	Media type is a two-part identifier for Internet file formats.
3486 Required:	False
3487 Inherited:	False
3488 Property Type:	String
3489 Cardinality:	Single
3490 Updatability:	Read Write

3491

3492
3493
3494
3495
3496
3497
3498
3499
3500
3501

Content disposition specifies a presentation style.

False

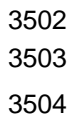
False

ContentDispositionType

Single

Read Write

3500 The Content class MAY include additional property definitions which are implementation-defined.
3501



3503 *Figure 25: Composite Content Class Diagram.*
3504

3505 4.2.3 MultiContent

3506 4.2.3.1 Description

3507 A multi-content object represents multiple parts of a message or document. It is a composite content that
3508 can contain a list of simple or composite contents.

3509 4.2.3.2 Class Definition

3510 The MultiContent class has attribute values:

3511
3512
3513

Value: icom content

3514

3515 **localName**

3516 Value: MultiContent

3517

3518 **extendsFrom**

3519 Value: icom_content:Content

3520

3521 **stereotype**

3522 Value: primary

3523

3524 **description**

3525 Value: A multi-content object represents the multiple parts of a message or document.

3526

3527 **propertyDefinitions**

3528 The values for this attribute are defined in Section 4.2.3.3.

3529 4.2.3.3 Property Definitions

3530 The MultiContent class inherits property definitions from super classes.

3531 The MultiContent class MUST have the property definitions:

3532

3533 **icom_content:part**

3534 Description:	Zero or more parts of a hierarchical composite content.
3535 Required:	False
3536 Inherited:	False
3537 Property Type:	icom_content:MimeConvertible
3538 Cardinality:	Multi
3539 Updatability:	Read Write

3540

3541 The MultiContent class MAY include additional property definitions which are implementation-defined.

3542

3543 4.2.4 SimpleContent

3544 4.2.4.1 Description

3545 A simple content holds a single piece of data.

3546 4.2.4.2 Class Definition

3547 The SimpleContent class has attribute values:

3548

3549 **localNamespace**

3550 Value: icom_content

3551

3552 **localName**
3553 Value: SimpleContent
3554
3555 **extendsFrom**
3556 Value: icom_content:Content
3557
3558 **stereotype**
3559 Value: primary
3560
3561 **description**
3562 Value: A simple content holds a single piece of data.
3563
3564 **propertyDefinitions**
3565 The values for this attribute are defined in Section 4.2.4.3.

3566 **4.2.4.3 Property Definitions**

3567 The SimpleContent class inherits property definitions from super classes.
3568 The SimpleContent class MUST have the property definitions:

3569
3570 **icom_content:characterEncoding**
3571 Description: Character encoding specifies character set of a content (a
3572 missing value means that a piece of content should be
3573 treated as binary or raw).
3574 Required: False
3575 Inherited: False
3576 Property Type: String
3577 Cardinality: Single
3578 Updatability: Read Write
3579
3580 **icom_content:contentEncoding**
3581 Description: Content encoding specifies encoding of a piece of content.
3582 Required: False
3583 Inherited: False
3584 Property Type: String
3585 Cardinality: Single
3586 Updatability: Read Write
3587
3588 **icom_content:contentLanguage**
3589 Description: Content language specifies language for a piece of content (a
3590 missing value means non-natural language content).
3591 Required: False
3592 Inherited: False
3593 Property Type: Locale

3594	Cardinality:	Single
3595	Updatability:	Read Write
3596		
3597	icom_content:contentLength	
3598	Description:	Length of a piece of content.
3599	Required:	False
3600	Inherited:	False
3601	Property Type:	Integer
3602	Cardinality:	Single
3603	Updatability:	Read Write

3604		
3605	icom_content:contentBody	
3606	Description:	Body of a simple content.
3607	Required:	False
3608	Inherited:	False
3609	Property Type:	Object
3610	Cardinality:	Single
3611	Updatability:	Read Write

3612

3613 The SimpleContent class MAY include additional property definitions which are implementation-defined.

3614

3615 4.2.5 OnlineContent

3616 4.2.5.1 Description

3617 An online content holds an online artifact attached to a document, message, or invitation.

3618 Note: An online artifact must be rendered as an IRI when a message or invitation is delivered to external

3619 recipients.

3620 4.2.5.2 Class Definition

3621 The OnlineContent class has attribute values:

3622	
3623	localNamespace
3624	Value: icom_content
3625	
3626	localName
3627	Value: OnlineContent
3628	
3629	extendsFrom
3630	Value: icom_content:Content
3631	
3632	stereotype
3633	Value: primary

3634
3635 **description**
3636 Value: An online content holds an online artifact attached to a message or invitation.

3637
3638 **propertyDefinitions**
3639 The values for this attribute are defined in Section 4.2.5.3.

3640 **4.2.5.3 Property Definitions**

3641 The OnlineContent class inherits property definitions from super classes.

3642 The OnlineContent class MUST have the property definition:

3643
3644 **icom_content:onlineAttachment**
3645 Description: An online artifact attached to a message.
3646 Required: True
3647 Inherited: False
3648 Property Type: icom_core:Artifact
3649 Cardinality: Single
3650 Updatability: Read Write

3651
3652 The OnlineContent class MAY include additional property definitions which are implementation-defined.
3653

3654 **4.2.6 ContentDispositionType**

3655 **4.2.6.1 Description**

3656 A content disposition type is a presentation style of content.

3657 **4.2.6.2 Class Definition**

3658 The ContentDispositionType class is a mixin class which defines a presentation style of content.

3659 The ContentDispositionType class has attribute values:

3660
3661 **localNamespace**
3662 Value: icom_content
3663
3664 **localName**
3665 Value: ContentDispositionType
3666
3667 **extendsFrom**
3668 Value:
3669
3670 **stereotype**
3671 Value: mixin
3672

3673 **description**
3674 Value: ContentDispositionType is a mixin class which defines a presentation style of content.

3675
3676 **propertyDefinitions**
3677 The values for this attribute are defined in Section 4.2.6.3.

3678 4.2.6.3 Property Definitions

3679 The ContentDispositionType class MAY include additional property definitions which are implementation-
3680 defined.

3681

3682 4.2.7 ContentDispositionTypeEnum

3683 The ContentDispositionTypeEnum class is an enum class that enumerates the instances each of which
3684 expresses a presentation style of content.

3685 The ContentDispositionTypeEnum class has attribute values:

3686

3687 **localNamespace**
3688 Value: icom_content

3689

3690 **localName**
3691 Value: ContentDispositionTypeEnum

3692

3693 **extendsFrom**
3694 Value: ContentDispositionType

3695

3696 **stereotype**
3697 Value: primary

3698

3699 **isEnumeration**
3700 Value: TRUE

3701

3702 **description**
3703 Value: A presentation style of content.

3704

3705 **instances**
3706 Value: <icom_content:Inline, icom_content:Attachment>

3707

3708 ICOM defines two content disposition types:

3709

- **icom_content:Inline** content is to be displayed automatically upon display of the main body of an
3710 artifact.

3711

- **icom_content:Attachment** content is separate from the main body of an artifact, and that its
3712 display should not be automatic, but contingent upon some further action of a user.

3713

4.2.8 Attachment

4.2.8.1 Description

An attachment holds a content for an occurrence, task, and contact artifact.

4.2.8.2 Class Definition

The Attachment class has attribute values:

localNamespace

Value: icom_content

localName

Value: Attachment

extendsFrom

Value:

stereotype

Value: primary

description

Value: An attachment holds a content for an occurrence, task, and contact artifact.

propertyDefinitions

The values for this attribute are defined in Section 4.2.8.3.

4.2.8.3 Property Definitions

The Attachment class MUST have the property definitions:

icom_core:name

Description: Name of a content attachment.

Required: True

Inherited: False

Property Type: String

Cardinality: Single

Updatability: Read Write

icom_content:content

Description: A content attached to an occurrence, task, or contact artifact.

Required: True

Inherited: False

Property Type: icom_content:Content

Cardinality: Single

3754	Updatability:	Read Write
------	---------------	------------

3755

3756 The Attachment class MAY include additional property definitions which are implementation-defined.

3757

3758 4.3 Document Module

3759 4.3.1 Versionable

3760 4.3.1.1 Description

3761 A versionable artifact is

1. a non-version-controlled copy,
2. a specific versioned copy,
3. a private working copy, or
4. a representative copy (optional)

3766 of an artifact version series.

3767 When a versionable artifact is not under version control, a non-version-controlled copy MUST be the only
3768 copy in a version series, i.e. there is only one copy and one *objectId*.

3769 When a versionable artifact is under version control, a representative copy MAY provide a version-
3770 independent view of a versionable artifact.

3771 When a non-version-controlled copy is placed under version control, a versioned copy **MUST** be created.

3772 Assignment of an object identifier to a versioned copy is implementation-dependent:

- 3773 • if a versioned copy retains the object identifier of a non-version-controlled copy, the version type
3774 of a versionable artifact MUST change from NonVersionControlledCopy to VersionedCopy;
- 3775 • if a versioned copy is assigned a new object identifier that is different from the object identifier of
3776 a non-version-controlled copy, a representative copy MAY retain the object identifier of the non-
3777 version-controlled copy;
- 3778 • if both versioned copy and representative copy are assigned new object identifiers that are
3779 different from the object identifier of a non-version-controlled copy, the non-version-controlled
3780 copy SHALL be discarded.

3781 When a private working copy is checked in, a versioned copy **MUST** be created. Assignment of an object
3782 identifier to a versioned copy is implementation-dependent:

- 3783 • if a versioned copy retains the object identifier of a private working copy, the version type of a
3784 versionable artifact MUST change from PrivateWorkingCopy to VersionedCopy;
- 3785 • if a versioned copy is assigned a new object identifier that is different from the object identifier of
3786 a private working copy, the private working copy SHALL be discarded.

3787 It is optional for a service provider to provide a representative copy for a version series. If a representative
3788 copy is provided:

- 3789 • a representative copy MUST have its own object identifier that is different from the object
3790 identifier of any versioned copy or private working copy;
- 3791 • assignment of an object identifier to a representative copy is implementation-dependent:
- 3792 ○ a representative copy MAY retain the object identifier of a non-version-controlled copy; if
3793 so the version type of a versionable artifact MUST change from
3794 NonVersionControlledCopy to RepresentativeCopy;
- 3795 ○ a representative copy MAY be assigned a new object identifier that is different from the
3796 object identifier of a non-version-controlled copy;
- 3797 • content and state of a representative copy is implementation-dependent:

- 3798 ○ a representative copy MAY be a copy of the content and state of the latest versioned
3799 copy or the latest major versioned copy in a version series;
- 3800 ○ a representative copy MAY be a copy of the content and state of a private working copy if
3801 the current user loading the representative copy is the same user who checks out a
3802 version series.

3803 Note: Each versioned copy of a versionable artifact is itself a versionable artifact, i.e. it has its own
3804 *objectId*. A versioned copy has a version number, label, and check in comment.

3805 Note: A private working copy is a versionable artifact created by an explicit checkout operation on a
3806 versionable artifact under version control. The properties for a private working copy are identical to the
3807 properties of a versioned copy on which a checkout operation was performed. Certain properties such as
3808 *objectId* and *creationDate* are different from a versioned copy. The content of a private working copy is
3809 identical to the content of a versioned copy. Its object identifier is different from that of the representative
3810 copy or any versioned copy.

3811 A private working copy MAY be saved in a version series for sharing and co-editing, however, it needs
3812 not be visible to users who may only have permissions to view other versioned copies in a version series.

3813 Note: Until it is checked in using an explicit check-in operation, a private working copy must not be
3814 considered the LatestMajorVersion in a version series.

3815 A container of a versionable artifact CAN contain a representative copy so that it provides a version-
3816 independent view of a state of the version series.

3817 Note: Starting from a representative copy in a container, an actor can traverse a version series to retrieve
3818 any versioned copy or private working copy.

3819 ICOM version control model is based on the CMIS version control model specified in Section 2.1.9 of
3820 Content Management Interoperability Services Version 1.0 [CMIS].

3821 4.3.1.2 Class Definition

3822 The Versionable class is a mixin class that defines the characteristics of artifacts that can be versioned.

3823 The Versionable class has attribute values:

3824

3825 **localNamespace**

3826 Value: icom_doc

3827

3828 **localName**

3829 Value: Versionable

3830

3831 **extendsFrom**

3832 Value: icom_core:Identifiable

3833

3834 **stereotype**

3835 Value: mixin

3836

3837 **description**

3838 Value: Versionable class is a mixin class that defines the characteristics of artifacts that can be
3839 versioned.

3840

3841 **propertyDefinitions**

3842 The values for this attribute are defined in Section 4.3.1.3.

4.3.1.3 Property Definitions

The Versionable class inherits property definitions from super classes.

The Versionable class MUST have the property definitions:

icom_doc:versionControlMetadata

Description:	A version control metadata object attached to a versionable artifact.
Required:	False
Inherited:	False
Property Type:	icom_doc:VersionControlMetadata
Cardinality:	Single
Updatability:	Read Only

icom_doc:versionType

Description:	A type of version controlled copy of a versionable artifact.
Required:	False
Inherited:	False
Property Type:	icom_doc:VersionType
Cardinality:	Single
Updatability:	Read Only

The Versionable class MAY include additional property definitions which are implementation-defined.

4.3.2 VersionControlMetadata

4.3.2.1 Description

A version control metadata is an object that contains version control information.

There are two classes of version control metadata: version series and version. A version control metadata of a versionable artifact is either a version series or a version depending on the version type.

- If the version type is icom_doc:NonVersionControlledCopy then metadata is optional; if metadata is present, it MUST be a version series object.
- If the version type is icom_doc:RepresentativeCopy, then metadata MUST be a version series object.
- If the version type is icom_doc:VersionedCopy or icom_doc:PrivateWorkingCopy, then metadata MUST be a version object.

4.3.2.2 Class Definition

The VersionControlMetadata class is a mixin class that defines the characteristics of version or version series metadata for version control.

The VersionControlMetadata class has attribute values:

localNamespace

Value: icom_doc

3884

3885 **localName**

3886 Value: VersionControlMetadata

3887

3888 **extendsFrom**

3889 Value: icom_core:Identifiable

3890

3891 **stereotype**

3892 Value: mixin

3893

3894 **description**

3895 Value: VersionControlMetadata is a mixin class that defines the characteristics of entities that

3896 serve as metadata for version control.

3897

3898 **propertyDefinitions**

3899 The values for this attribute are defined in Section 4.3.2.3.

3900 **4.3.2.3 Property Definitions**

3901 The VersionControlMetadata class inherits property definitions from super classes.

3902 The VersionControlMetadata class MUST have the property definition:

3903

3904 **icom_doc:representativeCopy**

3905 Description:	A representative copy of a versionable artifact.
3906 Required:	False
3907 Inherited:	False
3908 Property Type:	icom_doc:Versionable
3909 Cardinality:	Single
3910 Updatability:	Read Only

3911

3912 The VersionControlMetadata class MAY include additional property definitions which are implementation-

3913 defined.

3915 **4.3.3 VersionSeries**

3916 **4.3.3.1 Description**

3917 A version series is a version control metadata that contains a version history and check in/out states of a

3918 versionable artifact.

3919 A version series object is a version control metadata of a representative copy of a versionable artifact.

3920 **4.3.3.2 Class Definition**

3921 The VersionSeries class has attribute values:

3922

3923 **localNamespace**
3924 Value: icom_doc
3925
3926 **localName**
3927 Value: VersionSeries
3928
3929 **extendsFrom**
3930 Value: icom_core:Entity, icom_doc:VersionControlMetadata, icom_meta:RelationshipBondable
3931
3932 **stereotype**
3933 Value: primary
3934
3935 **description**
3936 Value: A version series is version control metadata that contains a version history and check
3937 in/out states of a versionable artifact.
3938
3939 **propertyDefinitions**
3940 The values for this attribute are defined in Section 4.3.3.3.

3941 **4.3.3.3 Property Definitions**

3942 The VersionSeries class inherits property definitions from super classes.
3943 The VersionSeries class MUST have the property definitions:

3944
3945 **icom_doc:versionHistory**
3946 Description: A history of version nodes of a versionable artifact.
3947 Required: False
3948 Inherited: False
3949 Property Type: icom_doc:Version
3950 Cardinality: Multi
3951 Updatability: Read Only
3952
3953 **icom_doc:versionableHistory**
3954 Description: A history of the versioned copies of a versionable artifact.
3955 Required: False
3956 Inherited: False
3957 Property Type: icom_doc:Versionable
3958 Cardinality: Multi
3959 Updatability: Read Only
3960
3961 **icom_doc:latestVersionedCopy**
3962 Description: Latest versioned copy of a versionable artifact.
3963 Required: False
3964 Inherited: False

3965	Property Type:	icom_doc:Versionable
3966	Cardinality:	Single
3967	Updatability:	Read Only
3968		
3969	icom_doc:privateWorkingCopy	
3970	Description:	A private working copy of a versionable artifact.
3971	Required:	False
3972	Inherited:	False
3973	Property Type:	icom_doc:Versionable
3974	Cardinality:	Single
3975	Updatability:	Read Only
3976		
3977	icom_doc:versionSeriesCheckedOut	
3978	Description:	Indicates whether a version series is checked out.
3979	Required:	False
3980	Inherited:	False
3981	Property Type:	Boolean
3982	Cardinality:	Single
3983	Updatability:	Read Only
3984		
3985	icom_doc:versionSeriesCheckedOutBy	
3986	Description:	An actor who checks out a version series.
3987	Required:	False
3988	Inherited:	False
3989	Property Type:	icom_core:Actor
3990	Cardinality:	Single
3991	Updatability:	Read Only
3992		
3993	icom_doc:versionSeriesCheckedOutOn	
3994	Description:	The time when a version series is checked out.
3995	Required:	False
3996	Inherited:	False
3997	Property Type:	DateTime
3998	Cardinality:	Single
3999	Updatability:	Read Only
4000		
4001	icom_doc:versionSeriesCheckoutComment	
4002	Description:	A check out comment of a version series.
4003	Required:	False
4004	Inherited:	False
4005	Property Type:	String
4006	Cardinality:	Single

4007	Updatability:	Read Only
4008		
4009	icom_doc:totalSize	
4010	Description:	Total size of all versioned copies of a versionable artifact in a
4011		version series.
4012	Required:	False
4013	Inherited:	False
4014	Property Type:	Integer
4015	Cardinality:	Single
4016	Updatability:	Read Only

4017

4018 The VersionSeries class MAY include additional property definitions which are implementation-defined.

4019

4020 4.3.4 Version

4021 4.3.4.1 Description

4022 A version is a version control metadata that contains a version number, label, and description.

4023 A version object is a version control metadata of a versioned copy or a private working copy of a

4024 versionable artifact.

4025 4.3.4.2 Class Definition

4026 The Version class has attribute values:

4027

4028 **localNamespace**

4029 Value: icom_doc

4030

4031 **localName**

4032 Value: Version

4033

4034 **extendsFrom**

4035 Value: icom_core:Entity, icom_doc:VersionControlMetadata, icom_meta:RelationshipBondable

4036

4037 **stereotype**

4038 Value: primary

4039

4040 **description**

4041 Value: A version is a version control metadata that contains a version number, label, and

4042 description.

4043

4044 **propertyDefinitions**

4045 The values for this attribute are defined in Section 4.3.4.3.

4.3.4.3 Property Definitions

The Version class inherits property definitions from super classes.

The Version class MUST have the property definitions;

icom_doc:checkinComment

Description:	A check in comment of a versioned copy.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

icom_doc:versionNumber

Description:	A version number of a versioned copy.
Required:	True
Inherited:	False
Property Type:	Integer
Cardinality:	Single
Updatability:	Read Write

icom_doc:versionLabel

Description:	A version label of a versioned copy.
Required:	True
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

icom_doc:majorVersion

Description:	Indicates whether a versioned copy is a major version.
Required:	True
Inherited:	False
Property Type:	Boolean
Cardinality:	Single
Updatability:	Read Write

icom_doc:versionedOrPrivateWorkingCopy

Description:	A versioned copy or private working copy corresponding to a version of a versionable artifact.
Required:	False
Inherited:	False
Property Type:	icom_doc:Versionable

4088 Cardinality: Single
4089 Updatability: Read Only

4090
4091 The Version class MAY include additional property definitions which are implementation-defined.
4092

4093 **4.3.5 VersionType**

4094 **4.3.5.1 Description**

4095 A version type is a version state of a copy of versionable document.

4096 **4.3.5.2 Class Definition**

4097 The VersionType class is a mixin class which defines a version state of a copy of versionable document.
4098 The VersionType class has attribute values:

4099
4100 **localNamespace**
4101 Value: icom_doc
4102
4103 **localName**
4104 Value: VersionType
4105
4106 **extendsFrom**
4107 Value:
4108
4109 **stereotype**
4110 Value: mixin
4111
4112 **description**
4113 Value: VersionType is a mixin class which defines a version state of a copy of versionable
4114 document.
4115
4116 **propertyDefinitions**
4117 The values for this attribute are defined in Section 4.3.5.3.

4118 **4.3.5.3 Property Definitions**

4119 The VersionType class MAY include additional property definitions which are implementation-defined.
4120

4121 **4.3.6 VersionTypeEnum**

4122 The VersionTypeEnum class is an enum class that enumerates the instances each of which expresses a
4123 version type.

4124 The VersionTypeEnum class has attribute values:

4125

4126 **localNamespace**
4127 Value: icom_doc
4128
4129 **localName**
4130 Value: VersionTypeEnum
4131
4132 **extendsFrom**
4133 Value: VersionType
4134
4135 **stereotype**
4136 Value: primary
4137
4138 **isEnumeration**
4139 Value: TRUE
4140
4141 **description**
4142 Value: A version type of a copy of versionable document.
4143
4144 **instances**
4145 Value: <icom_doc:NonVersionControlledCopy, icom_doc:VersionedCopy,
4146 icom_doc:PrivateWorkingCopy, icom_doc:RepresentativeCopy>
4147
4148 ICOM defines four version types:
4149

- **icom_doc:NonVersionControlledCopy** a versionable artifact is not under version control.
- **icom_doc:VersionedCopy** a versionable artifact is a version of an artifact version series.
- **icom_doc:PrivateWorkingCopy** a versionable artifact is a private working copy of an artifact version series.
- **icom_doc:RepresentativeCopy** a versionable artifact is a version-independent representative copy of an artifact. This version type is optional and implementation-dependent.

4155

4156 **4.3.7 Document**

4157 **4.3.7.1 Description**

4158 A document is a versionable artifact that can contain a single content of a media type or composite
4159 contents of an assortment of media types.

4160 **4.3.7.2 Class Definition**

4161 The Document class has attribute values:

4162
4163 **localNamespace**
4164 Value: icom_doc
4165

4166 **localName**
4167 Value: Document
4168
4169 **extendsFrom**
4170 Value: icom_core:Artifact, icom_doc:Versionable, icom_content:MimeConvertible
4171
4172 **stereotype**
4173 Value: primary
4174
4175 **description**
4176 Value: A document is a versionable artifact that may contain a single content of a media type or
4177 composite contents of an assortment of media types.
4178
4179 **propertyDefinitions**
4180 The values for this attribute are defined in Section 4.3.7.3.

4181 **4.3.7.3 Property Definitions**

4182 The Document class inherits property definitions from super classes.
4183 The Document class **MUST** have the property definitions:

4184
4185 **icom_content:content**
4186 Description: Content of a document.
4187 Required: False
4188 Inherited: False
4189 Property Type: icom_content:Content
4190 Cardinality: Single
4191 Updatability: Read Write
4192
4193 **icom_doc:size**
4194 Description: The size of a copy of a document.
4195 Required: False
4196 Inherited: False
4197 Property Type: Integer
4198 Cardinality: Single
4199 Updatability: Read Only

4200
4201 The Document class **MAY** include additional property definitions which are implementation-defined.
4202

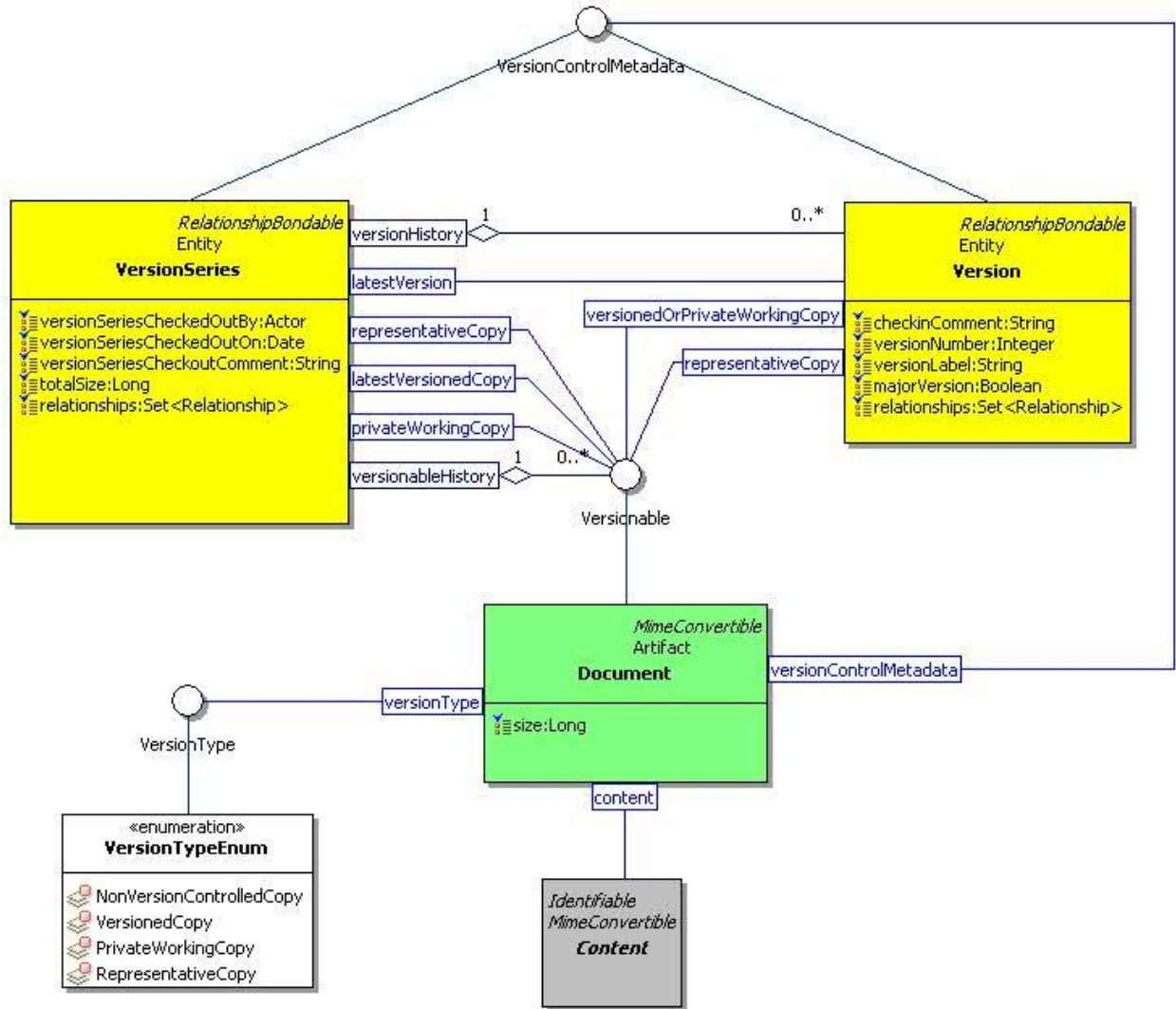


Figure 26: Document, Version Series, and Version Class Diagram.

4.3.8 WikiPage

4.3.8.1 Description

A wiki page is a document that contains a wiki content and that provides an html page generated from the wiki content.

4.3.8.2 Class Definition

The WikiPage class has attribute values:

localNamespace

Value: icom doc

localName

Value: WikiPage

4218
4219 **extendsFrom**
4220 Value: icom_doc:Document
4221
4222 **stereotype**
4223 Value: primary
4224
4225 **description**
4226 Value: A wiki page is a document that contains a wiki content and that provides an html page
4227 generated from the wiki content.
4228
4229 **propertyDefinitions**
4230 The values for this attribute are defined in Section 4.3.8.3.

4231 **4.3.8.3 Property Definitions**

4232 The WikiPage class inherits property definitions from super classes.
4233 The WikiPage class MUST have the property definitions:

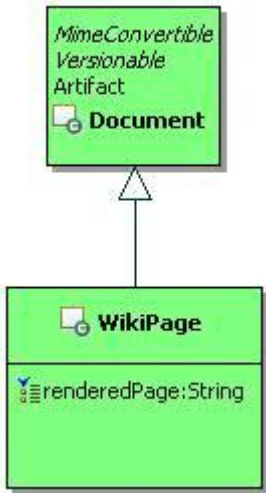
4234

4235	icom_doc:renderedPage	
4236	Description:	An html page generated from a wiki content.
4237	Required:	False
4238	Inherited:	False
4239	Property Type:	String
4240	Cardinality:	Single
4241	Updatability:	Read Only

4242

4243 The WikiPage class MAY include additional property definitions which are implementation-defined.

4244



4245
4246 *Figure 27: Wiki Page Class Diagram.*

4247

4.4 Message Module

4.4.1 Message

4.4.1.1 Description

A message is a unit of conversation. It holds a simple content or multipart message contents in a content property. It has a single sender.

Note: The delivered time is the time when a message is delivered to a given recipient. The user creation date and time property can be used as the sent date and time of a message. The name property can be used as the subject of a message.

4.4.1.2 Class Definition

The Message class has attribute values:

localNamespace

Value: icom_msg

localName

Value: Message

extendsFrom

Value: icom_core:Artifact

stereotype

Value: primary

isAbstract

Value: TRUE

description

Value: A message is a unit of conversation.

propertyDefinitions

The values for this attribute are defined in Section 4.4.1.3.

4.4.1.3 Property Definitions

The Message class inherits property definitions from super classes.

The Message class MUST have the property definitions:

icom_content:content

Description: Content of a message

Required: False

Inherited: False

Property Type: icom_content:Content

4288	Cardinality:	Single
4289	Updatability:	Read Write
4290		
4291	icom_msg:sender	
4292	Description:	Sender of a message.
4293	Required:	False
4294	Inherited:	False
4295	Property Type:	icom_core:Participant
4296	Cardinality:	Single
4297	Updatability:	Read Write
4298		
4299	icom_msg:deliveredTime	
4300	Description:	The date and time when a message is delivered to a given
4301		recipient.
4302	Required:	False
4303	Inherited:	False
4304	Property Type:	DateTime
4305	Cardinality:	Single
4306	Updatability:	Read Only

4307

4308 The Message class MAY include additional property definitions which are implementation-defined.

4309

4310 4.4.2 UnifiedMessage

4311 4.4.2.1 Description

4312 A unified message is a type of message delivered electronically over a computer, voice, fax, and other
4313 networks.

4314 A unified message can be one of these types:

- 4315 • Email is a message delivered electronically over a computer network.
- 4316 • Voice is a message that contains a voice or audio stream.
- 4317 • Fax is a message that contains an image transmitted via phone lines using the fax protocol.
- 4318 • Notification is a type of message sent by applications.

4319 4.4.2.2 Class Definition

4320 The UnifiedMessage class has attribute values:

4321

4322 **localNamespace**

4323 Value: icom_msg

4324

4325 **localName**

4326 Value: UnifiedMessage

4327

4328 **extendsFrom**
4329 Value: icom_msg:Message, icom_content:MimeConvertible
4330
4331 **stereotype**
4332 Value: primary
4333
4334 **description**
4335 Value: A unified message is a type of message delivered electronically over a computer, voice,
4336 fax, and other networks.
4337
4338 **propertyDefinitions**
4339 The values for this attribute are defined in Section 4.4.2.3.

4340 **4.4.2.3 Property Definitions**

4341 The UnifiedMessage class inherits property definitions from super classes.
4342 The UnifiedMessage class MUST have the property definitions:

4343
4344 **icom_core:priority**
4345 Description: The priority of a message.
4346 Required: False
4347 Inherited: False
4348 Property Type: icom_core:Priority
4349 Cardinality: Single
4350 Updatability: Read Write
4351
4352 **icom_content:contentId**
4353 Description: Content id is a unique identifier for a message part in multi-
4354 part messages.
4355 Required: False
4356 Inherited: False
4357 Property Type: String
4358 Cardinality: Single
4359 Updatability: Read Write
4360
4361 **icom_content:mediaType**
4362 Description: Media type is a two-part identifier for Internet file formats.
4363 Required: False
4364 Inherited: False
4365 Property Type: String
4366 Cardinality: Single
4367 Updatability: Read Write
4368

4369	icom_content:contentDisposition	
4370	Description:	Content disposition specifies a presentation style.
4371	Required:	False
4372	Inherited:	False
4373	Property Type:	icom_content:ContentDispositionType
4374	Cardinality:	Single
4375	Updatability:	Read Write
4376		
4377	icom_msg:envelopeSender	
4378	Description:	An envelope sender is a participant to receive bounced message. It is also known as return path.
4379		
4380	Required:	False
4381	Inherited:	False
4382	Property Type:	icom_core:Participant
4383	Cardinality:	Single
4384	Updatability:	Read Write
4385		
4386	icom_msg:toReceivers	
4387	Description:	A list of participants to receive a message.
4388	Required:	False
4389	Inherited:	False
4390	Property Type:	icom_core:Participant
4391	Cardinality:	Multi
4392	Updatability:	Read Write
4393		
4394	icom_msg:ccReceivers	
4395	Description:	A list of participants to receive carbon-copies of a message.
4396	Required:	False
4397	Inherited:	False
4398	Property Type:	icom_core:Participant
4399	Cardinality:	Multi
4400	Updatability:	Read Write
4401		
4402	icom_msg:bccReceivers	
4403	Description:	A list of participants to receive blind-carbon-copies of a message.
4404		
4405	Required:	False
4406	Inherited:	False
4407	Property Type:	icom_core:Participant
4408	Cardinality:	Multi
4409	Updatability:	Read Write
4410		

4411	icom_msg:replyTo	
4412	Description:	A list of participants to receive a reply message.
4413	Required:	False
4414	Inherited:	False
4415	Property Type:	icom_core:Participant
4416	Cardinality:	Multi
4417	Updatability:	Read Write
4418		
4419	icom_msg:flag	
4420	Description:	Zero or more flags on a message.
4421	Required:	False
4422	Inherited:	False
4423	Property Type:	icom_msg:UnifiedMessageFlag
4424	Cardinality:	Multi
4425	Updatability:	Read Write
4426		
4427	icom_msg:messageDispositionNotificationRequested	
4428	Description:	A message disposition notification requested for a message.
4429	Required:	False
4430	Inherited:	False
4431	Property Type:	Boolean
4432	Cardinality:	Single
4433	Updatability:	Read Write
4434		
4435	icom_msg:messageDeliveryStatusNotificationRequest	
4436	Description:	Indicates the types of delivery status notifications requested for a message. Default is icom_msg:Failure.
4437		
4438	Required:	False
4439	Inherited:	False
4440	Property Type:	icom_msg:UnifiedMessageDeliveryStatusNotificationRequest
4441	Cardinality:	Multi
4442	Updatability:	Read Write
4443		
4444	icom_msg:channel	
4445	Description:	Indicates the delivery channel of a message.
4446	Required:	False
4447	Inherited:	False
4448	Property Type:	icom_msg:UnifiedMessageChannel
4449	Cardinality:	Single
4450	Updatability:	Read Write
4451		

4452	icom_msg:editMode	
4453	Description:	Indicates an editable mode (new, draft, or delivered) of a message.
4454		
4455	Required:	False
4456	Inherited:	False
4457	Property Type:	icom_msg:UnifiedMessageEditMode
4458	Cardinality:	Single
4459	Updatability:	Read Only

4460		
4461	icom_msg:mimeHeader	
4462	Description:	A list of headers. Each header is represented by a multi-valued property.
4463		
4464	Required:	False
4465	Inherited:	False
4466	Property Type:	icom_meta:Property
4467	Cardinality:	Multi
4468	Updatability:	Read Write

4469		
4470	icom_msg:size	
4471	Description:	The size of a unified message.
4472	Required:	False
4473	Inherited:	False
4474	Property Type:	Integer
4475	Cardinality:	Single
4476	Updatability:	Read Only

4477

4478 The UnifiedMessage class MAY include additional property definitions which are implementation-defined.

4479

4480 4.4.3 UnifiedMessageParticipant

4481 4.4.3.1 Description

4482 A unified message participant object represents the participation of an addressable entity in a unified message.

4483

4484 4.4.3.2 Class Definition

4485 The UnifiedMessageParticipant class has attribute values:

4486		
4487	localNamespace	
4488	Value:	icom_msg
4489		
4490	localName	
4491	Value:	UnifiedMessageParticipant
4492		

4493 **extendsFrom**
4494 Value: icom_core:Participant
4495
4496 **stereotype**
4497 Value: primary
4498
4499 **description**
4500 Value: A unified message participant object represents the participation of an addressable entity
4501 in a unified message.
4502
4503 **propertyDefinitions**
4504 The values for this attribute are defined in Section 4.4.3.3.

4505 **4.4.3.3 Property Definitions**

4506 The UnifiedMessageParticipant class inherits property definitions from super classes.
4507 The UnifiedMessageParticipant class **MUST** have the property definitions:

4509 **icom_msg:fullAddress**
4510 Description: Full address of a participant.
4511 Required: False
4512 Inherited: False
4513 Property Type: IRI
4514 Cardinality: Single
4515 Updatability: Read Write

4517 **icom_msg:localPart**
4518 Description: Local part of a full address.
4519 Required: False
4520 Inherited: False
4521 Property Type: String
4522 Cardinality: Single
4523 Updatability: Read Write

4525 **icom_msg:domainPart**
4526 Description: Domain part of a full address.
4527 Required: False
4528 Inherited: False
4529 Property Type: String
4530 Cardinality: Single
4531 Updatability: Read Write

4533 The UnifiedMessageParticipant class **MAY** include additional property definitions which are
4534 implementation-defined.

4535

4536 **4.4.4 UnifiedMessageFlag**

4537 **4.4.4.1 Description**

4538 A unified message flag is a flag on a message.

4539 **4.4.4.2 Class Definition**

4540 The UnifiedMessageFlag class is a mixin class which defines a flag on a message.

4541 The UnifiedMessageFlag class has attribute values:

4542

4543 **localNamespace**

4544 Value: icom_msg

4545

4546 **localName**

4547 Value: UnifiedMessageFlag

4548

4549 **extendsFrom**

4550 Value:

4551

4552 **stereotype**

4553 Value: mixin

4554

4555 **description**

4556 Value: UnifiedMessageFlag is a mixin class which defines a flag on a message.

4557

4558 **propertyDefinitions**

4559 The values for this attribute are defined in Section 4.4.4.3.

4560 **4.4.4.3 Property Definitions**

4561 The UnifiedMessageFlag class MAY include additional property definitions which are implementation-
4562 defined.

4563

4564 **4.4.5 UnifiedMessageFlagEnum**

4565 The UnifiedMessageFlagEnum class is an enum class that enumerates the instances each of which
4566 expresses a flag on a message.

4567 The UnifiedMessageFlagEnum class has attribute values:

4568

4569 **localNamespace**

4570 Value: icom_msg

4571

4572 **localName**

4573 Value: UnifiedMessageFlagEnum

4574
4575 **extendsFrom**
4576 Value: UnifiedMessageFlag
4577
4578 **stereotype**
4579 Value: primary
4580
4581 **isEnumeration**
4582 Value: TRUE
4583
4584 **description**
4585 Value: A flag on a message.
4586
4587 **instances**
4588 Value: <icom_msg:Answered, icom_msg:Forwarded, icom_msg:Redirected, icom_msg:Hidden,
4589 icom_msg:MarkedForDelete, icom_msg:MarkedForFollowUp, icom_msg:MarkedForDraft,
4590 icom_msg:MessageDispositionNotificationProcessed>

4591
4592 ICOM defines eight flags:
4593

- **icom_msg:Answered** a message is answered.
- **icom_msg:Forwarded** a message is forwarded.
- **icom_msg:Redirected** a message is redirected.
- **icom_msg:Hidden** a message is hidden.
- **icom_msg:MarkedForDelete** a message is marked for delete.
- **icom_msg:MarkedForFollowUp** a message is marked for follow up.
- **icom_msg:MarkedForDraft** a message is marked for draft.
- **icom_msg:MessageDispositionNotificationProcessed** a message disposition notification is processed.

4602

4603 4.4.6 UnifiedMessageDeliveryStatusNotificationRequest

4604 4.4.6.1 Description

4605 A unified message delivery status notification request is a directive for notifying a participant of delivery
4606 status of a message.

4607 4.4.6.2 Class Definition

4608 The UnifiedMessageDeliveryStatusNotificationRequest class is a mixin class which defines a directive for
4609 notifying a participant of delivery status of a message.

4610 The UnifiedMessageDeliveryStatusNotificationRequest class has attribute values:

4611
4612 **localNamespace**
4613 Value: icom_msg
4614

4615 **localName**
4616 Value: UnifiedMessageDeliveryStatusNotificationRequest
4617
4618 **extendsFrom**
4619 Value:
4620
4621 **stereotype**
4622 Value: mixin
4623
4624 **description**
4625 Value: UnifiedMessageDeliveryStatusNotificationRequest is a mixin class which defines a
4626 directive for notifying a participant of delivery status of a message.
4627
4628 **propertyDefinitions**
4629 The values for this attribute are defined in Section 4.4.6.3.

4630 **4.4.6.3 Property Definitions**

4631 The UnifiedMessageDeliveryStatusNotificationRequest class MAY include additional property definitions
4632 which are implementation-defined.
4633

4634 **4.4.7 UnifiedMessageDeliveryStatusNotificationRequestEnum**

4635 The UnifiedMessageDeliveryStatusNotificationRequestEnum class is an enum class that enumerates the
4636 instances each of which expresses a request for one of several types of delivery status notification.

4637 The UnifiedMessageDeliveryStatusNotificationRequestEnum class has attribute values:

4638
4639 **localNamespace**
4640 Value: icom_msg
4641
4642 **localName**
4643 Value: UnifiedMessageDeliveryStatusNotificationRequestEnum
4644
4645 **extendsFrom**
4646 Value: UnifiedMessageDeliveryStatusNotificationRequest
4647
4648 **stereotype**
4649 Value: primary
4650
4651 **isEnumeration**
4652 Value: TRUE
4653
4654 **description**
4655 Value: A request for one of several types of delivery status notification.
4656

4657 **instances**

4658 Value: <icom_msg:Never, icom_msg:Success, icom_msg:Failure, icom_msg:Delay>

4659

4660 ICOM defines four delivery status notification requests:

- 4661 • **icom_msg:Never** a sender requests status notification not be returned to the sender under any
- 4662 condition.
- 4663 • **icom_msg:Success** a sender requests a status notification for successful delivery of a message.
- 4664 • **icom_msg:Failure** a sender requests a status notification for delivery failure of a message.
- 4665 • **icom_msg:Delay** a sender requests a status notification when delivery of a message has been
- 4666 delayed for an unusual length of time.

4667

4668 **4.4.8 UnifiedMessageChannel**

4669 **4.4.8.1 Description**

4670 A message channel used to deliver a unified message.

4671 **4.4.8.2 Class Definition**

4672 The UnifiedMessageChannel class is a mixin class which defines a channel used to deliver a unified

4673 message.

4674 The UnifiedMessageChannel class has attribute values:

4675

4676 **localNamespace**

4677 Value: icom_msg

4678

4679 **localName**

4680 Value: UnifiedMessageChannel

4681

4682 **extendsFrom**

4683 Value:

4684

4685 **stereotype**

4686 Value: mixin

4687

4688 **description**

4689 Value: UnifiedMessageChannel is a mixin class which defines a channel used to deliver a

4690 unified message.

4691

4692 **propertyDefinitions**

4693 The values for this attribute are defined in Section 4.4.8.3.

4694 **4.4.8.3 Property Definitions**

4695 The UnifiedMessageChannel class MAY include additional property definitions which are implementation-

4696 defined.

4697

4.4.9 UnifiedMessageChannelEnum

The UnifiedMessageChannelEnum class is an enum class that enumerates the instances each of which expresses a type of delivery channel.

The UnifiedMessageChannelEnum class has attribute values:

localNamespace

Value: icom_msg

localName

Value: UnifiedMessageChannelEnum

extendsFrom

Value: UnifiedMessageChannel

stereotype

Value: primary

isEnumeration

Value: TRUE

description

Value: A delivery channel.

instances

Value: <icom_msg:Email, icom_msg:Voice, icom_msg:Fax, icom_msg:Notification>

ICOM defines four channel types:

- **icom_msg:Email** delivery channel is email.
- **icom_msg:Voice** delivery channel is voice.
- **icom_msg:Fax** delivery channel is fax.
- **icom_msg:Notification** delivery channel is notification.

4.4.10 UnifiedMessageEditMode

4.4.10.1 Description

A unified message edit mode is a mode that indicates whether a unified message is editable.

4.4.10.2 Class Definition

The UnifiedMessageEditMode class is a mixin class which defines a mode that indicates whether a unified message is editable.

The UnifiedMessageEditMode class has attribute values:

4738 **localNamespace**
4739 Value: icom_msg
4740
4741 **localName**
4742 Value: UnifiedMessageEditMode
4743
4744 **extendsFrom**
4745 Value:
4746
4747 **stereotype**
4748 Value: mixin
4749
4750 **description**
4751 Value: UnifiedMessageEditMode is a mixin class which defines a mode that indicates whether a
4752 unified message is editable.
4753
4754 **propertyDefinitions**
4755 The values for this attribute are defined in Section 4.4.10.3.

4756 **4.4.10.3 Property Definitions**

4757 The UnifiedMessageEditMode class MAY include additional property definitions which are
4758 implementation-defined.
4759

4760 **4.4.11 UnifiedMessageEditModeEnum**

4761 The UnifiedMessageEditModeEnum class is an enum class that enumerates the instances each of which
4762 expresses whether a message is a new copy, saved draft copy, or delivered copy.
4763 The UnifiedMessageEditModeEnum class has attribute values:

4764
4765 **localNamespace**
4766 Value: icom_msg
4767
4768 **localName**
4769 Value: UnifiedMessageEditModeEnum
4770
4771 **extendsFrom**
4772 Value: UnifiedMessageEditMode
4773
4774 **stereotype**
4775 Value: primary
4776
4777 **isEnumeration**
4778 Value: TRUE
4779

description

Value: A message is a new copy, a saved draft copy, or a delivered copy. New or draft copies are usually editable while delivered copies are usually not editable.

instances

Value: <icom_msg:NewCopy, icom_msg:DraftCopy, icom_msg:DeliveredCopy>

ICOM defines three modes:

- **icom_msg:NewCopy** a message is a new message.
- **icom_msg:DraftCopy** a message is saved as a draft.
- **icom_msg:DeliveredCopy** a message is a sent or received message.

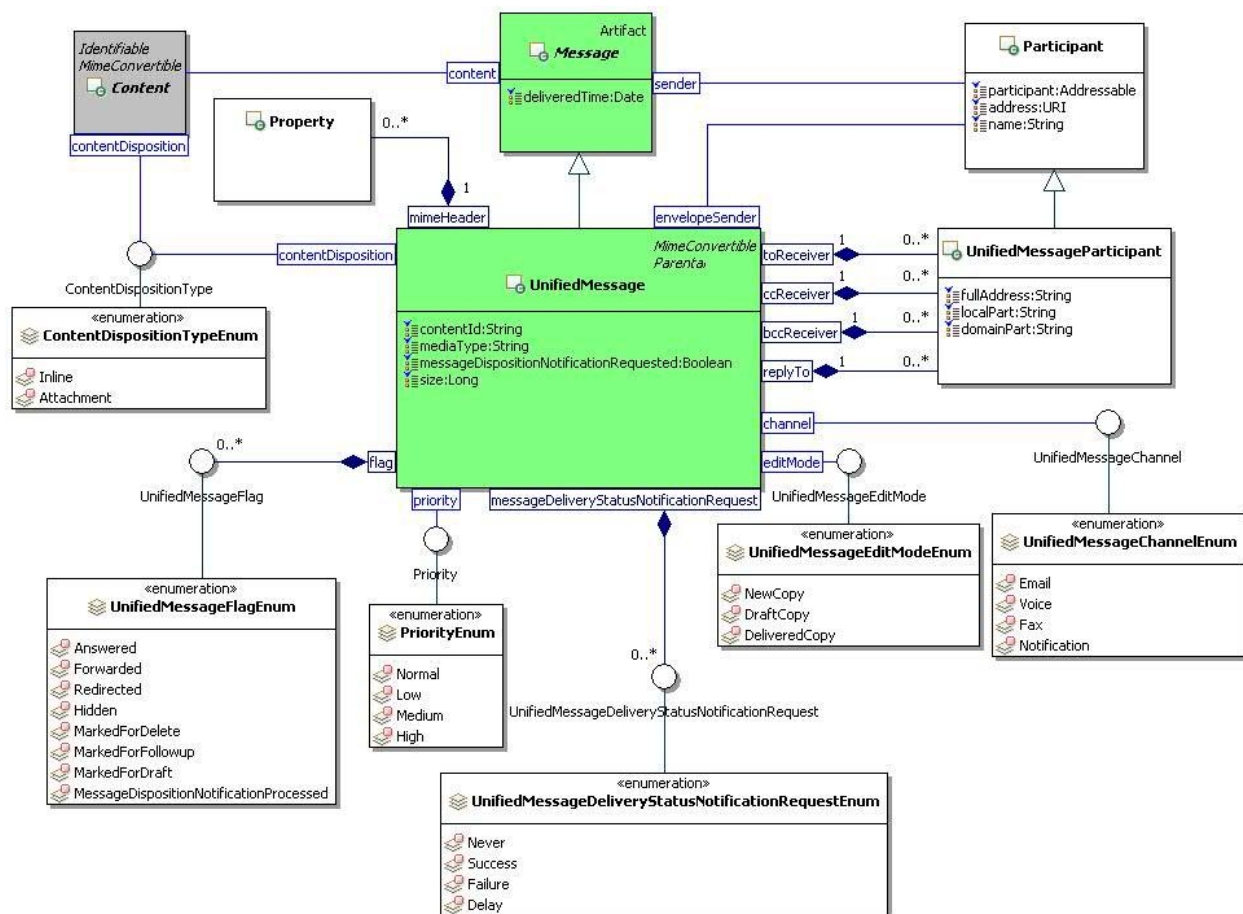


Figure 28: Unified Message Class Diagram.

4.4.12 InstantMessage

4.4.12.1 Description

An instant message is a type of message for synchronous, usually text based, conversation.

4.4.12.2 Class Definition

The InstantMessage class has attribute values:

localNamespace

Value: icom_msg

localName

Value: InstantMessage

extendsFrom

Value: icom_msg:Message

stereotype

Value: primary

isAbstract

Value: TRUE

description

Value: An instant message is a type of message for synchronous, usually text based, conversation.

propertyDefinitions

The values for this attribute are defined in Section 4.4.12.3.

4.4.12.3 Property Definitions

The InstantMessage class inherits property definitions from super classes.

The InstantMessage class MUST have the property definitions:

icom_msg:toReceivers

Description: A list of participants to receive a message.

Required: False

Inherited: False

Property Type: icom_core:Participant

Cardinality: Multi

Updatability: Read Write

icom_msg:conversationId

Description: An identifier of a conversation involving one or more instant messages.

Required: False

Inherited: False

Property Type: Integer

4840	Cardinality:	Single
4841	Updatability:	Read Write
4842		
4843	icom_msg:clientSideld	
4844	Description:	An identifier of a client.
4845	Required:	False
4846	Inherited:	False
4847	Property Type:	String
4848	Cardinality:	Single
4849	Updatability:	Read Write
4850		
4851	icom_msg:formattingStyle	
4852	Description:	A style for formatting a rich text message.
4853	Required:	False
4854	Inherited:	False
4855	Property Type:	String
4856	Cardinality:	Single
4857	Updatability:	Read Write
4858		
4859	icom_msg:instantMessageType	
4860	Description:	A type of instant message.
4861	Required:	False
4862	Inherited:	False
4863	Property Type:	icom_msg:InstantMessageType
4864	Cardinality:	Single
4865	Updatability:	Read Write
4866		
4867	icom_msg:chatStatus	
4868	Description:	A chat status of a user.
4869	Required:	False
4870	Inherited:	False
4871	Property Type:	icom_msg:InstantMessageChatStatus
4872	Cardinality:	Single
4873	Updatability:	Read Write
4874		

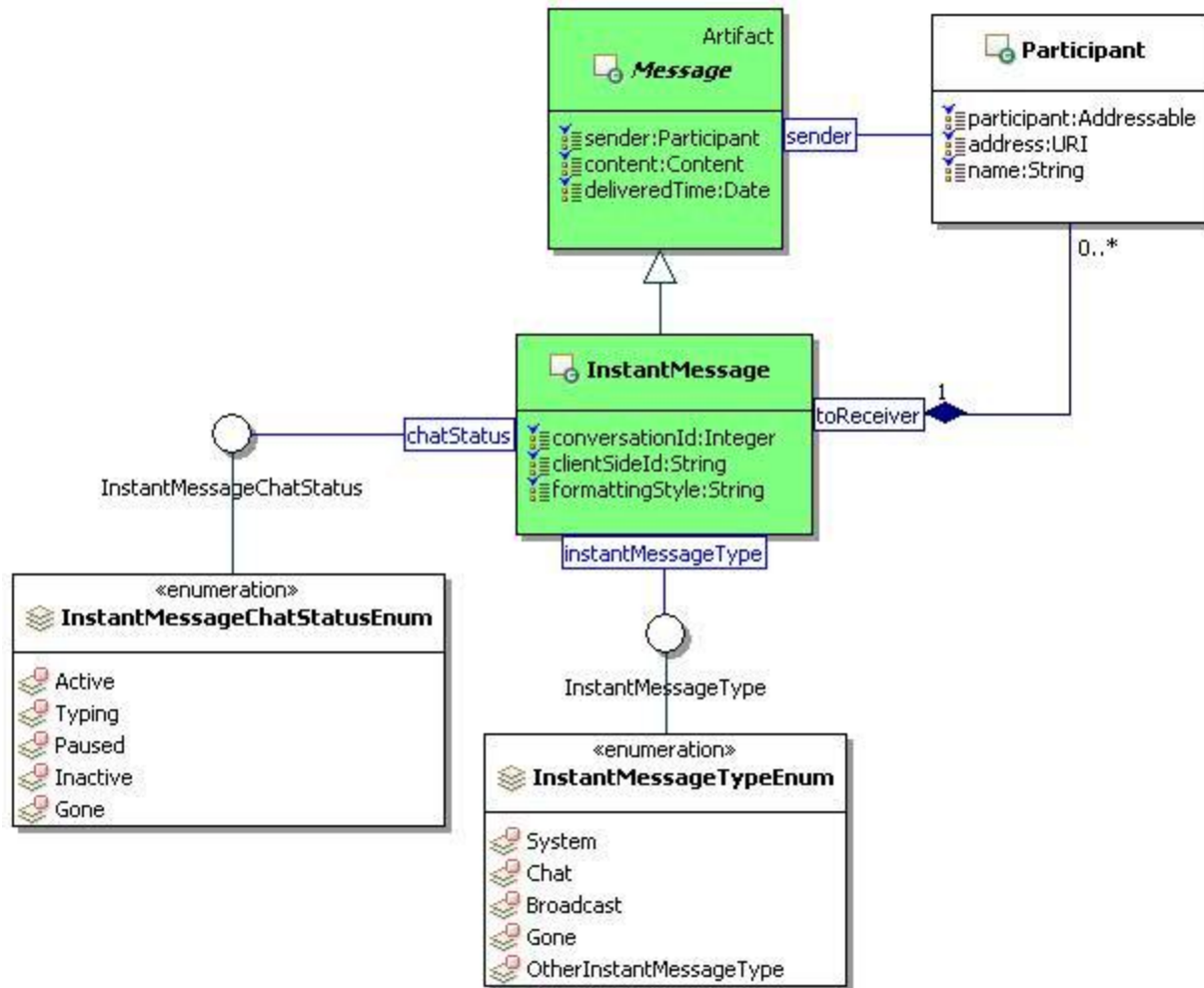


Figure 29: Instant Message Class Diagram.

4.4.13 InstantMessageType

4.4.13.1 Description

An instant message type.

4.4.13.2 Class Definition

The InstantMessageType class is a mixin class which defines a type of instant message.

The InstantMessageType class has attribute values:

localNamespace

Value: icom_msg

localName

Value: InstantMessageType

4891 **extendsFrom**
4892 Value:
4893
4894 **stereotype**
4895 Value: mixin
4896
4897 **description**
4898 Value: InstantMessageType is a mixin class which defines a type of instant message.
4899
4900 **propertyDefinitions**
4901 The values for this attribute are defined in Section 4.4.13.3.

4902 **4.4.13.3 Property Definitions**

4903 The InstantMessageType class MAY include additional property definitions which are implementation-
4904 defined.
4905

4906 **4.4.14 InstantMessageTypeEnum**

4907 The InstantMessageTypeEnum class is an enum class that enumerates the instances each of which
4908 expresses a type of instant message.

4909 The InstantMessageTypeEnum class has attribute values:

4910
4911 **localNamespace**
4912 Value: icom_msg
4913
4914 **localName**
4915 Value: InstantMessageTypeEnum
4916
4917 **extendsFrom**
4918 Value: InstantMessageType
4919
4920 **stereotype**
4921 Value: primary
4922
4923 **isEnumeration**
4924 Value: TRUE
4925
4926 **description**
4927 Value: A type of instant message.
4928
4929 **instances**
4930 Value: <icom_msg:System, icom_msg:Chat, icom_msg:Broadcast, icom_msg:Gone,
4931 icom_msg:OtherInstantMessageType>
4932

4933 ICOM defines five instant message types:

- 4934 • **icom_msg:System** an instant message is a system message.
- 4935 • **icom_msg:Chat** an instant message is a chat message.
- 4936 • **icom_msg:Broadcast** an instant message is a broadcast message.
- 4937 • **icom_msg:Gone** an instant message is a message indicating that a user is gone.
- 4938 • **icom_msg:OtherInstantMessageType** an instant message is of other type.

4939

4940 4.4.15 InstantMessageChatStatus

4941 4.4.15.1 Description

4942 An instant message chat status defines a vocabulary of chat status.

4943 4.4.15.2 Class Definition

4944 The InstantMessageChatStatus class is a mixin class which defines a chat status.

4945 The InstantMessageChatStatus class has attribute values:

4946

4947 **localNamespace**

4948 Value: icom_msg

4949

4950 **localName**

4951 Value: InstantMessageChatStatus

4952

4953 **extendsFrom**

4954 Value:

4955

4956 **stereotype**

4957 Value: mixin

4958

4959 **description**

4960 Value: InstantMessageChatStatus is a mixin class which defines a chat status.

4961

4962 **propertyDefinitions**

4963 The values for this attribute are defined in Section 4.4.15.3.

4964 4.4.15.3 Property Definitions

4965 The InstantMessageChatStatus class MAY include additional property definitions which are
4966 implementation-defined.

4967

4968 4.4.16 InstantMessageChatStatusEnum

4969 The InstantMessageChatStatusEnum class is an enum class that enumerates the instances each of
4970 which expresses a chat status of a user.

4971 The InstantMessageChatStatusEnum class has attribute values:

4972
4973 **localNamespace**
4974 Value: icom_msg
4975
4976 **localName**
4977 Value: InstantMessageChatStatusEnum
4978
4979 **extendsFrom**
4980 Value: InstantMessageChatStatus
4981
4982 **stereotype**
4983 Value: primary
4984
4985 **isEnumeration**
4986 Value: TRUE
4987
4988 **description**
4989 Value: A chat status of a user.
4990
4991 **instances**
4992 Value: <icom_msg:Active, icom_msg:Typing, icom_msg:Paused, icom_msg:Inactive,
4993 icom_msg:Gone>
4994
4995 ICOM defines five chat status:
4996

- **icom_msg:Active** a user is active.
- **icom_msg:Typing** a user is typing.
- **icom_msg:Paused** a user has paused.
- **icom_msg:Inactive** a user is inactive.
- **icom_msg:Gone** a user is gone.

5001

5002 **4.4.17 InstantMessageFeed**

5003 **4.4.17.1 Description**

5004 An instant message feed contains a set of instant message connections and a queue of outbound instant
5005 messages.

5006 **4.4.17.2 Class Definition**

5007 The InstantMessageFeed class has attribute values:

5008
5009 **localNamespace**
5010 Value: icom_msg
5011

5012 **localName**
5013 Value: InstantMessageFeed
5014
5015 **extendsFrom**
5016 Value: icom_core:Entity
5017
5018 **stereotype**
5019 Value: primary
5020
5021 **description**
5022 Value: An instant message feed contains a set of instant message connections and a queue of
5023 outbound instant messages.
5024
5025 **propertyDefinitions**
5026 The values for this attribute are defined in Section 4.4.17.3.

5027 **4.4.17.3 Property Definitions**

5028 The InstantMessageFeed class inherits property definitions from super classes.
5029 The InstantMessageFeed class MUST have the property definitions:

5030
5031 **icom_msg:connection**
5032 Description: One or more instant messaging connections.
5033 Required: False
5034 Inherited: False
5035 Property Type: icom_msg:InstantMessageConnection
5036 Cardinality: Multi
5037 Updatability: Read Only

5038
5039 **icom_msg:outboundInstantMessage**
5040 Description: Outbound instant messages.
5041 Required: False
5042 Inherited: False
5043 Property Type: icom_msg:InstantMessage
5044 Cardinality: Multi
5045 Updatability: Write Only

5046

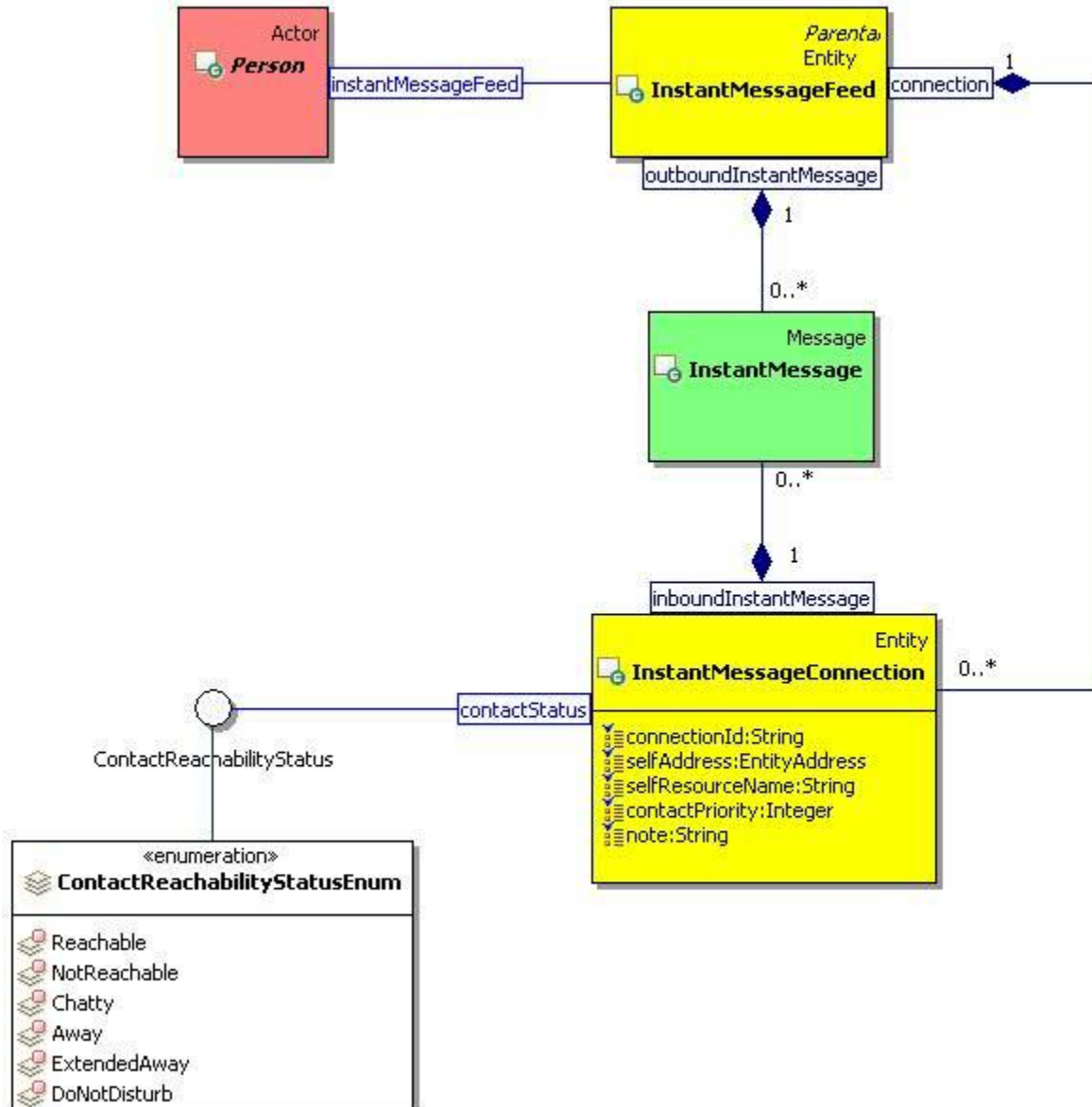


Figure 30: Instant Message Feed and Connection Class Diagram.

4.4.18 InstantMessageConnection

4.4.18.1 Description

An instant message connection contains queues for inbound instant messages.

A presentity can update the contact status, contact priority, and note for a contact method associated with a connection.

4.4.18.2 Class Definition

The **InstantMessageConnection** class has attribute values:

5058 **localNamespace**
5059 Value: icom_msg
5060
5061 **localName**
5062 Value: InstantMessageConnection
5063
5064 **extendsFrom**
5065 Value: icom_core:Entity
5066
5067 **stereotype**
5068 Value: primary
5069
5070 **description**
5071 Value: An instant message connection contains queues for inbound instant messages.
5072
5073 **propertyDefinitions**
5074 The values for this attribute are defined in Section 4.4.18.3.

5075 **4.4.18.3 Property Definitions**

5076 The InstantMessageConnection class inherits property definitions from super classes.

5077 The InstantMessageConnection class MUST have the property definitions:

5078
5079 **icom_msg:connectionId**
5080 Description: An identifier of a connection.
5081 Required: False
5082 Inherited: False
5083 Property Type: String
5084 Cardinality: Single
5085 Updatability: Read Only
5086
5087 **icom_msg:selfAddress**
5088 Description: Address of a presentity who opens a connection.
5089 Required: True
5090 Inherited: False
5091 Property Type: IRI
5092 Cardinality: Single
5093 Updatability: On Create
5094
5095 **icom_msg:selfResourceName**
5096 Description: Resource name associated with a connection.
5097 Required: True
5098 Inherited: False
5099 Property Type: String

5100	Cardinality:	Single
5101	Updatability:	On Create
5102		
5103	icom_msg:inboundInstantMessage	
5104	Description:	Inbound instant messages.
5105	Required:	False
5106	Inherited:	False
5107	Property Type:	icom_msg:InstantMessage
5108	Cardinality:	Multi
5109	Updatability:	Read Only
5110		
5111	icom_presence:contactStatus	
5112	Description:	Reachability status to be propagated to an associated contact
5113		method in presence.
5114	Required:	False
5115	Inherited:	False
5116	Property Type:	icom_presence:ContactReachabilityStatus
5117	Cardinality:	Single
5118	Updatability:	Write Only
5119		
5120	icom_presence:contactPriority	
5121	Description:	Priority to be propagated to an associated contact method in
5122		presence.
5123	Required:	False
5124	Inherited:	False
5125	Property Type:	Integer
5126	Cardinality:	Single
5127	Updatability:	Write Only
5128		
5129	icom_presence:note	
5130	Description:	Note to be propagated to an associated contact method in
5131		presence.
5132	Required:	False
5133	Inherited:	False
5134	Property Type:	String
5135	Cardinality:	Single
5136	Updatability:	Write Only
5137		

4.5 Presence Module

4.5.1 Presence

4.5.1.1 Description

A presence describes the contact methods and activities of a presentity.

It provides a list of contact methods describing how to contact a presentity. A viewer may choose any one of the contact methods based on circumstances.

It includes a list of activities describing what a presentity is doing.

4.5.1.2 Class Definition

The Presence class has attribute values:

localNamespace

Value: icom_presence

localName

Value: Presence

extendsFrom

Value: icom_core:Identifiable

stereotype

Value: primary

description

Value: A presence describes the contact methods and activities of a presentity.

propertyDefinitions

The values for this attribute are defined in Section 4.5.1.3.

4.5.1.3 Property Definitions

The Presence class inherits property definitions from super classes.

The Presence class MUST have the property definitions:

icom_core:lastModificationDate

Description: Last modification date and time of information in a presence.

Required: False

Inherited: False

Property Type: DateTime

Cardinality: Single

Updatability: Read Only

5177	icom_core:location	
5178	Description:	Location of a presentity.
5179	Required:	False
5180	Inherited:	False
5181	Property Type:	icom_core:Location
5182	Cardinality:	Single
5183	Updatability:	Read Only
5184		
5185	icom_presence:editMode	
5186	Description:	Indicates a mode which determines whether a presence is
5187		editable.
5188	Required:	False
5189	Inherited:	False
5190	Property Type:	icom_presence:PresenceEditMode
5191	Cardinality:	Single
5192	Updatability:	Read Only
5193		
5194	icom_presence:contactMethod	
5195	Description:	A collection of contact methods describing how to contact a
5196		presentity. A viewer may choose any one of the contact
5197		methods based on circumstances.
5198	Required:	False
5199	Inherited:	False
5200	Property Type:	icom_presence:ContactMethod
5201	Cardinality:	Multi
5202	Updatability:	Read Only
5203		
5204	icom_presence:activity	
5205	Description:	A collection of activities describing what a presentity is doing.
5206	Required:	False
5207	Inherited:	False
5208	Property Type:	icom_presence:Activity
5209	Cardinality:	Multi
5210	Updatability:	Read Only
5211		
5212	The Presence class MAY include additional property definitions which are implementation-defined.	
5213		

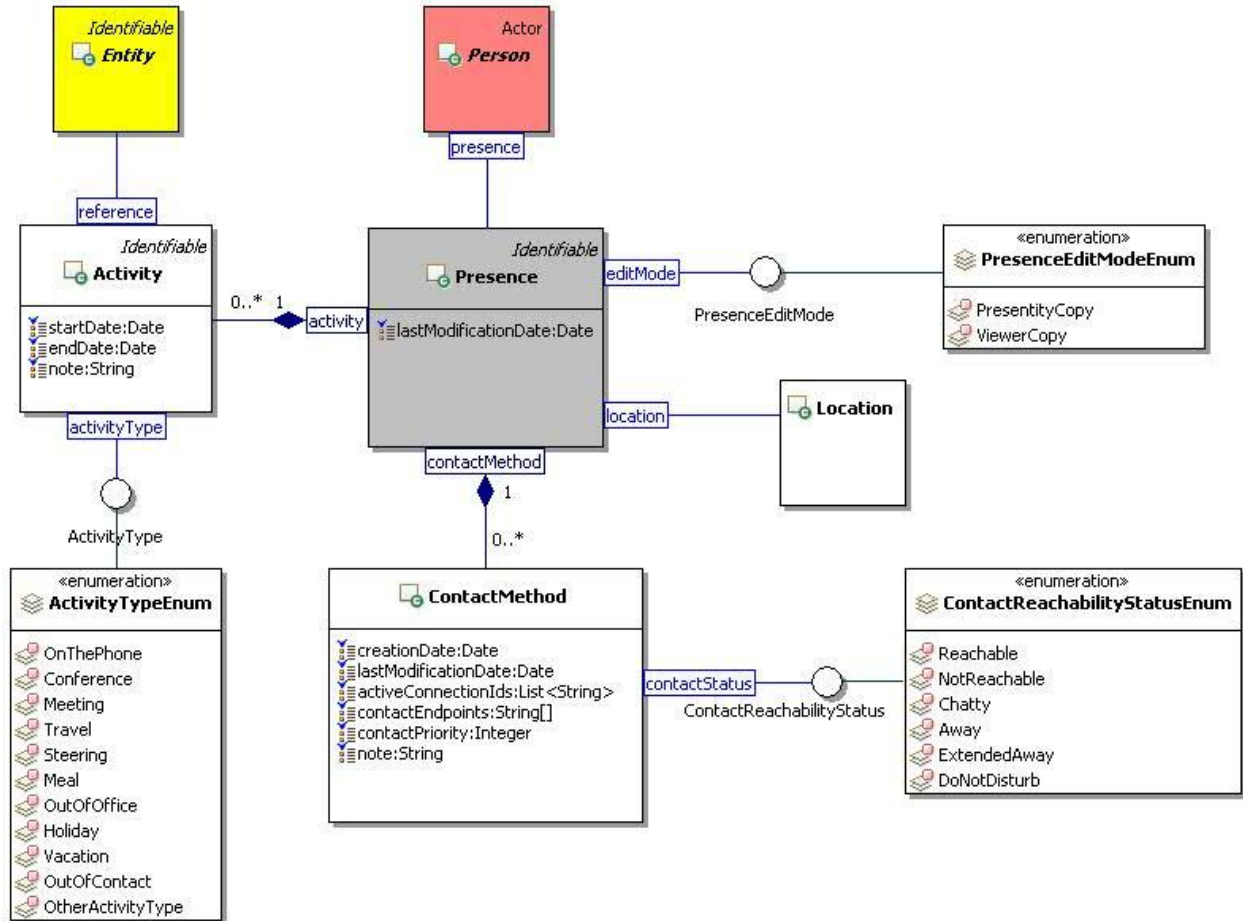


Figure 31: Presence Class Diagram.

4.5.2 PresenceEditMode

4.5.2.1 Description

A presence edit mode is a mode that indicates whether a presence is editable.

4.5.2.2 Class Definition

The PresenceEditMode class is a mixin class which defines a mode that indicates whether a presence is editable.

The PresenceEditMode class has attribute values:

localNamespace

Value: icom_presence

localName

Value: PresenceEditMode

5231 **extendsFrom**
5232 Value:
5233
5234 **stereotype**
5235 Value: mixin
5236
5237 **description**
5238 Value: PresenceEditMode is a mixin class which defines a mode that indicates whether a
5239 presence is editable.
5240
5241 **propertyDefinitions**
5242 The values for this attribute are defined in Section 4.5.2.3.

5243 **4.5.2.3 Property Definitions**

5244 The PresenceEditMode class MAY include additional property definitions which are implementation-
5245 defined.
5246

5247 **4.5.3 PresenceEditModeEnum**

5248 The PresenceEditModeEnum class is an enum class that enumerates the instances each of which
5249 expresses a mode that indicates whether a presence is editable.
5250 The PresenceEditModeEnum class has attribute values:

5251
5252 **localNamespace**
5253 Value: icom_presence
5254
5255 **localName**
5256 Value: PresenceEditModeEnum
5257
5258 **extendsFrom**
5259 Value: PresenceEditMode
5260
5261 **stereotype**
5262 Value: primary
5263
5264 **isEnumeration**
5265 Value: TRUE
5266
5267 **description**
5268 Value: A mode that indicates whether a presence is editable.
5269
5270 **instances**
5271 Value: <icom_presence:PresententityCopy, icom_presence:ViewerCopy>
5272

5273 ICOM defines two presence editable modes:

- 5274 • **icom_presence:PresentityCopy** a presence is a copy belonging to a presentity who may update
- 5275 the properties such as activities.
- 5276 • **icom_presence:ViewerCopy** a presence is a copy visible to a subscriber who may not update
- 5277 the properties.
- 5278

5279 4.5.4 ContactMethod

5280 4.5.4.1 Description

5281 A contact method object describes reachability circumstances of a presentity.

5282 4.5.4.2 Class Definition

5283 The ContactMethod class has attribute values:

5284

5285 **localNamespace**

5286 Value: icom_presence

5287

5288 **localName**

5289 Value: ContactMethod

5290

5291 **extendsFrom**

5292 Value:

5293

5294 **stereotype**

5295 Value: primary

5296

5297 **description**

5298 Value: A contact method object describes reachability circumstances of a presentity.

5299

5300 **propertyDefinitions**

5301 The values for this attribute are defined in Section 4.5.4.3

5302 4.5.4.3 Property Definitions

5303 The ContactMethod class MUST have the property definitions:

5304

5305 **icom_core:creationDate**

5306	Description:	Creation date and time of information in a contact method.
5307	Required:	False
5308	Inherited:	False
5309	Property Type:	DateTime
5310	Cardinality:	Single
5311	Updatability:	Read Only

5312

5313	icom_core:lastModificationDate	
5314	Description:	Last modification date and time of information in a contact
5315		method.
5316	Required:	False
5317	Inherited:	False
5318	Property Type:	DateTime
5319	Cardinality:	Single
5320	Updatability:	Read Only
5321		
5322	icom_presence:activeConnectionId	
5323	Description:	A list of active connection ids of a presentity.
5324	Required:	False
5325	Inherited:	False
5326	Property Type:	String
5327	Cardinality:	Multi
5328	Updatability:	Read Only
5329		
5330	icom_presence:contactEndpoint	
5331	Description:	A list of endpoints or IRIs for contacting a presentity.
5332	Required:	False
5333	Inherited:	False
5334	Property Type:	String
5335	Cardinality:	Multi
5336	Updatability:	Read Only
5337		
5338	icom_presence:contactPriority	
5339	Description:	Priority of a contact method relative to other contact methods
5340		in a presence.
5341	Required:	False
5342	Inherited:	False
5343	Property Type:	Integer
5344	Cardinality:	Single
5345	Updatability:	Read Only
5346		
5347	icom_presence:contactStatus	
5348	Description:	Status of a contact method in a presence.
5349	Required:	False
5350	Inherited:	False
5351	Property Type:	icom_presence:ContactReachabilityStatus
5352	Cardinality:	Single
5353	Updatability:	Read Only
5354		

5355	icom_presence:note	
5356	Description:	A note about contacting a presentity.
5357	Required:	False
5358	Inherited:	False
5359	Property Type:	String
5360	Cardinality:	Single
5361	Updatability:	Read Only
5362		

5363 4.5.5 ContactReachabilityStatus

5364 4.5.5.1 Description

5365 A contact reachability status is a status of a contact method.

5366 4.5.5.2 Class Definition

5367 The ContactReachabilityStatus class is a mixin class which defines a status of a contact method.

5368 The ContactReachabilityStatus class has attribute values:

5369

5370 **localNamespace**

5371 Value: icom_presence

5372

5373 **localName**

5374 Value: ContactReachabilityStatus

5375

5376 **extendsFrom**

5377 Value:

5378

5379 **stereotype**

5380 Value: mixin

5381

5382 **description**

5383 Value: ContactReachabilityStatus is a mixin class which defines a status of a contact method.

5384

5385 **propertyDefinitions**

5386 The values for this attribute are defined in Section 4.5.5.3.

5387 4.5.5.3 Property Definitions

5388 The ContactReachabilityStatus class MAY include additional property definitions which are

5389 implementation-defined.

5390

5391 4.5.6 ContactReachabilityStatusEnum

5392 The ContactReachabilityStatusEnum class is an enum class that enumerates the instances each of which

5393 expresses a reachability status of a contact method.

5394 The ContactReachabilityStatusEnum class has attribute values:

5395

5396 **localNamespace**

5397 Value: icom_presence

5398

5399 **localName**

5400 Value: ContactReachabilityStatusEnum

5401

5402 **extendsFrom**

5403 Value: ContactReachabilityStatus

5404

5405 **stereotype**

5406 Value: primary

5407

5408 **isEnumeration**

5409 Value: TRUE

5410

5411 **description**

5412 Value: A reachability status of a contact method.

5413

5414 **instances**

5415 Value: <icom_presence:Reachable, icom_presence:NotReachable, icom_presence:Chatty,

5416 icom_presence:Away, icom_presence:ExtendedAway, icom_presence:DoNotDisturb>

5417

5418 ICOM defines six reachability status:

- 5419 • **icom_presence:Reachable** a presentity is reachable through a contact method.
- 5420 • **icom_presence:NotReachable** a presentity is not reachable through a contact method.
- 5421 • **icom_presence:Chatty** a presentity is chatty.
- 5422 • **icom_presence:Away** a presentity is away.
- 5423 • **icom_presence:ExtendedAway** a presentity is away for an extended period.
- 5424 • **icom_presence:DoNotDisturb** a presentity prefers not to be disturbed.

5425

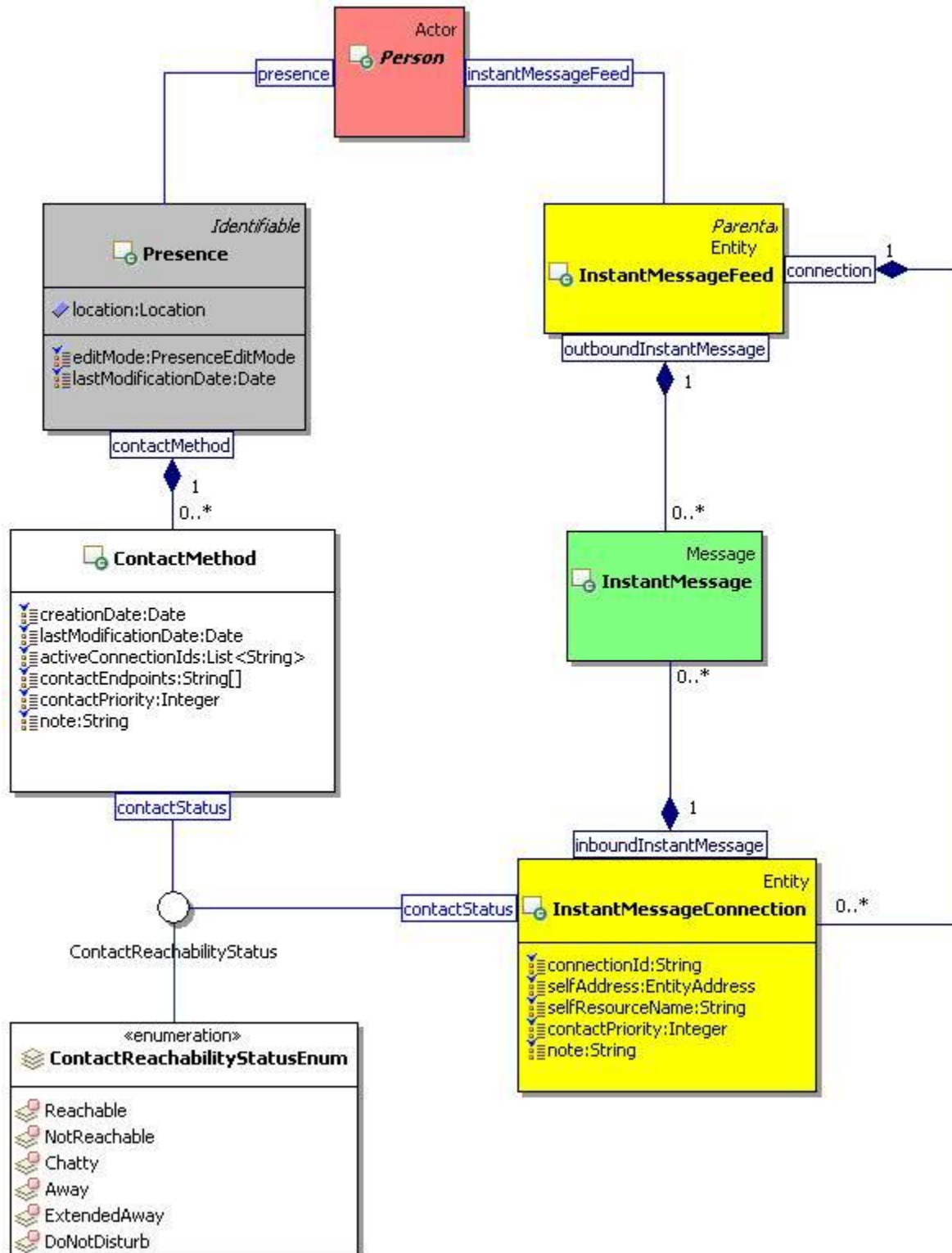


Figure 32: Presence Contact Method and Instant Message Connection Class Diagram.

4.5.7 Activity

4.5.7.1 Description

An activity object describes what a presentity is doing.

4.5.7.2 Class Definition

The Activity class has attribute values:

localNamespace

Value: icom_presence

localName

Value: Activity

extendsFrom

Value:

stereotype

Value: primary

description

Value: An activity object describes what a presentity is doing.

propertyDefinitions

The values for this attribute are defined in Section 4.5.7.3.

4.5.7.3 Property Definitions

The Activity class MUST have the property definitions:

icom_core:startDate

Description: Start date and time of an activity.

Required: True

Inherited: False

Property Type: DateTime

Cardinality: Single

Updatability: Read Write

icom_core:endDate

Description: End date and time of an activity.

Required: True

Inherited: False

Property Type: DateTime

Cardinality: Single

5469	Updatability:	Read Write
5470		
5471	icom_presence:activityType	
5472	Description:	Type of an activity.
5473	Required:	true
5474	Inherited:	False
5475	Property Type:	icom_presence:ActivityType
5476	Cardinality:	Single
5477	Updatability:	Read Write
5478		
5479	icom_presence:note	
5480	Description:	A note describing an activity.
5481	Required:	False
5482	Inherited:	False
5483	Property Type:	String
5484	Cardinality:	Single
5485	Updatability:	Read Write
5486		
5487	icom_presence:reference	
5488	Description:	An entity, such as occurrence, task, conference, etc., which is
5489		the source of or reference for an activity.
5490	Required:	False
5491	Inherited:	False
5492	Property Type:	icom_core:Entity
5493	Cardinality:	Single
5494	Updatability:	Read Write
5495		

5496 4.5.8 ActivityType

5497 4.5.8.1 Description

5498 An activity type is a vocabulary of activities for rich presence information model.

5499 4.5.8.2 Class Definition

5500 The ActivityType class is a mixin class which defines an activity.

5501 The ActivityType class has attribute values:

5502		
5503	localNamespace	
5504	Value:	icom_presence
5505		
5506	localName	
5507	Value:	ActivityType
5508		

5509 **extendsFrom**
5510 Value:
5511
5512 **stereotype**
5513 Value: mixin
5514
5515 **description**
5516 Value: ActivityType is a mixin class which defines a type of activity.
5517
5518 **propertyDefinitions**
5519 The values for this attribute are defined in Section 4.5.8.3.

5520 **4.5.8.3 Property Definitions**

5521 The ActivityType class MAY include additional property definitions which are implementation-defined.
5522

5523 **4.5.9 ActivityTypeEnum**

5524 The ActivityTypeEnum class is an enum class that enumerates the instances each of which expresses a
5525 type of activity.

5526 The ActivityTypeEnum class has attribute values:

5527
5528 **localNamespace**
5529 Value: icom_presence
5530
5531 **localName**
5532 Value: ActivityTypeEnum
5533
5534 **extendsFrom**
5535 Value: ActivityType
5536
5537 **stereotype**
5538 Value: primary
5539
5540 **isEnumeration**
5541 Value: TRUE
5542
5543 **description**
5544 Value: A type of activity.
5545
5546 **instances**
5547 Value: <icom_presence:OnThePhone, icom_presence:Conference, icom_presence:Meeting,
5548 icom_presence:Travel, icom_presence:Steering, icom_presence:Meal,
5549 icom_presence:OutOfOffice, icom_presence:Holiday, icom_presence:Vacation,
5550 icom_presence:OutOfContact, icom_presence:OtherActivityType>

5551
5552
5553
5554
5555
5556
5557
5558
5559
5560
5561
5562
5563
5564

ICOM defines eleven activity types:

- **icom_presence:OnThePhone** a presentity is on the phone.
- **icom_presence:Conference** a presentity is in a conference.
- **icom_presence:Meeting** a presentity is in a meeting.
- **icom_presence:Travel** a presentity is traveling.
- **icom_presence:Steering** a presentity is steering a vehicle.
- **icom_presence:Meal** a presentity is having a meal.
- **icom_presence:OutOfOffice** a presentity is out of office.
- **icom_presence:Holiday** a presentity is on holiday.
- **icom_presence:Vacation** a presentity is on vacation.
- **icom_presence:OutOfContact** a presentity is out of contact.
- **icom_presence:OtherActivityType** a presentity is involved in an unspecified activity.

5565 4.6 Address Book Module

5566 4.6.1 AddressBook

5567 4.6.1.1 Description

5568 An address book is a folder that contains sub-address books and addressable contacts.

5569 4.6.1.2 Class Definition

5570 The AddressBook class has attribute values:

5571
5572
5573
5574
5575
5576
5577
5578
5579
5580
5581
5582
5583
5584
5585
5586
5587
5588

localNamespace

Value: icom_card

localName

Value: AddressBook

extendsFrom

Value: icom_core:Folder

stereotype

Value: primary

description

Value: An address book is a folder that contains sub-address books and addressable contacts.

propertyDefinitions

The values for this attribute are defined in Section 4.6.1.3.

5589 **4.6.1.3 Property Definitions**

5590 The AddressBook class inherits property definitions from super classes.

5591 The AddressBook class MUST have the property definitions:

5592

icom_card:addressBook		
Description:		Sub-address books in an address book.
Required:		False
Inherited:		False
Property Type:		icom_card:AddressBook
Cardinality:		Multi
Updatability:		Read Only

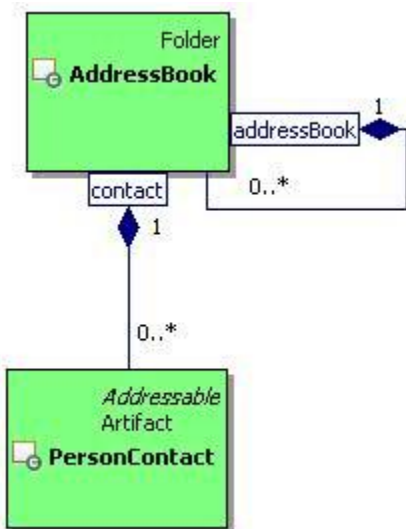
5600

5601 **icom_card:contact**

Description:		Contacts in an address book.
Required:		False
Inherited:		False
Property Type:		icom_card:PersonContact
Cardinality:		Multi
Updatability:		Read Only

5608

5609 The AddressBook class MAY include additional property definitions which are implementation-defined.



5611

5612 *Figure 33: Address Book Class Diagram.*

5613

5614 **4.6.2 PersonContact**

5615 **4.6.2.1 Description**

5616 A person contact is an artifact that contains address information about a person.

4.6.2.2 Class Definition

The PersonContact class has attribute values:

localNamespace

Value: icom_card

localName

Value: PersonContact

extendsFrom

Value: icom_core:Artifact, icom_core:Addressable

stereotype

Value: primary

description

Value: A person contact is an artifact that contains address information about a person.

propertyDefinitions

The values for this attribute are defined in Section 4.6.2.3.

4.6.2.3 Property Definitions

The PersonContact class inherits property definitions from super classes.

The PersonContact class MUST have the property definitions:

icom_core:timeZone

Description: Time zone of a person.

Required: False

Inherited: False

Property Type: icom_core:TimeZone

Cardinality: Single

Updatability: Read Write

icom_core:givenName

Description: Given name of a person.

Required: False

Inherited: False

Property Type: String

Cardinality: Single

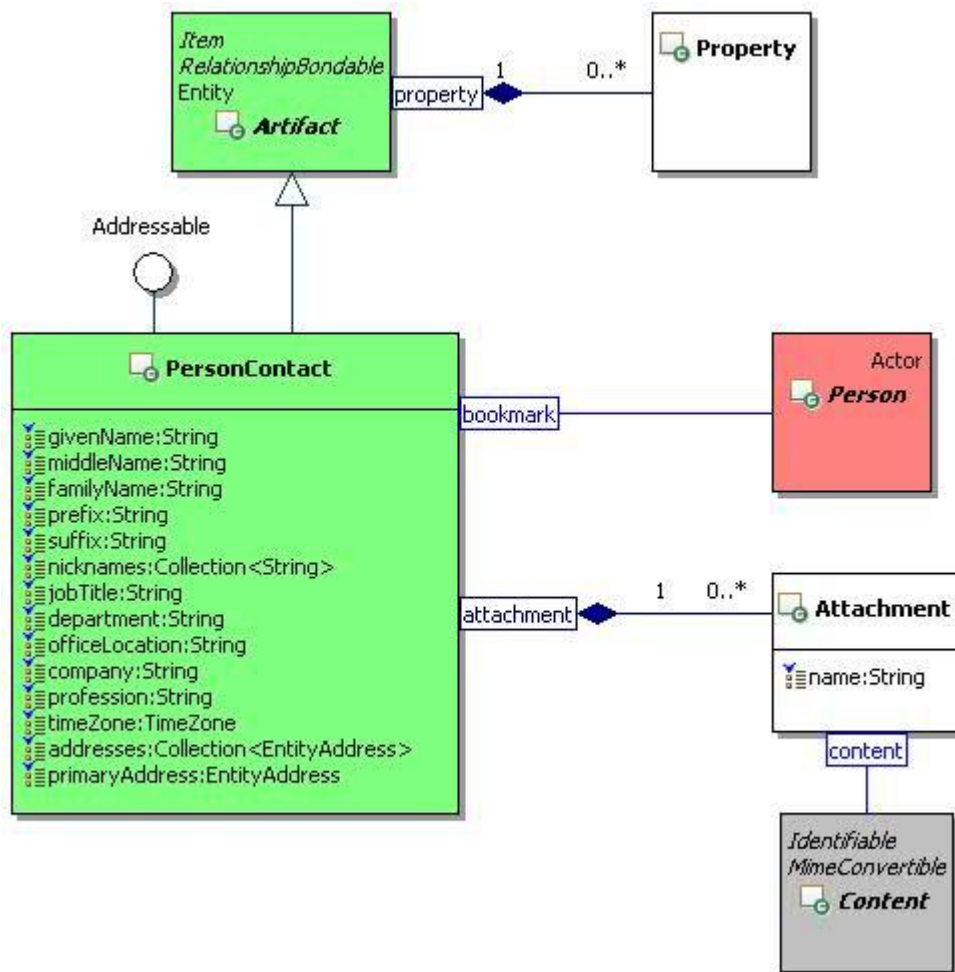
Updatability: Read Write

5657	icom_core:middleName	
5658	Description:	Middle name of a person. Can include multiple names concatenated.
5659		
5660	Required:	False
5661	Inherited:	False
5662	Property Type:	String
5663	Cardinality:	Single
5664	Updatability:	Read Write
5665		
5666	icom_core:familyName	
5667	Description:	Family name of a person.
5668	Required:	False
5669	Inherited:	False
5670	Property Type:	String
5671	Cardinality:	Single
5672	Updatability:	Read Write
5673		
5674	icom_core:prefix	
5675	Description:	Prefix of a person's name.
5676	Required:	False
5677	Inherited:	False
5678	Property Type:	String
5679	Cardinality:	Single
5680	Updatability:	Read Write
5681		
5682	icom_core:suffix	
5683	Description:	Suffix of a person's name.
5684	Required:	False
5685	Inherited:	False
5686	Property Type:	String
5687	Cardinality:	Single
5688	Updatability:	Read Write
5689		
5690	icom_core:nickname	
5691	Description:	Nickname of a person.
5692	Required:	False
5693	Inherited:	False
5694	Property Type:	String
5695	Cardinality:	Multi
5696	Updatability:	Read Write
5697		
5698	icom_core:jobTitle	
5699	Description:	Job title of a person.

5700	Required:	False
5701	Inherited:	False
5702	Property Type:	String
5703	Cardinality:	Single
5704	Updatability:	Read Write
5705		
5706	icom_core:department	
5707	Description:	A person's affiliated department.
5708	Required:	False
5709	Inherited:	False
5710	Property Type:	String
5711	Cardinality:	Single
5712	Updatability:	Read Write
5713		
5714	icom_core:officeLocation	
5715	Description:	Location of a person's department.
5716	Required:	False
5717	Inherited:	False
5718	Property Type:	String
5719	Cardinality:	Single
5720	Updatability:	Read Write
5721		
5722	icom_core:company	
5723	Description:	A person's affiliated company.
5724	Required:	False
5725	Inherited:	False
5726	Property Type:	String
5727	Cardinality:	Single
5728	Updatability:	Read Write
5729		
5730	icom_core:profession	
5731	Description:	A person's profession.
5732	Required:	False
5733	Inherited:	False
5734	Property Type:	String
5735	Cardinality:	Single
5736	Updatability:	Read Write
5737		
5738	icom_content:attachment	
5739	Description:	One or more content attachments in a contact.
5740	Required:	False
5741	Inherited:	False

5742 Property Type: icom_content:Attachment
5743 Cardinality: Multi
5744 Updatability: Read Write
5745
5746 **icom_card:bookmark**
5747 Description: A person which is bookmarked by a contact.
5748 Required: False
5749 Inherited: False
5750 Property Type: icom_core:Person
5751 Cardinality: Single
5752 Updatability: On Create

5754 The PersonContact class MAY include additional property definitions which are implementation-defined.
5755



5756
5757 Figure 34: Person Contact Class Diagram.
5758

4.7 Calendar Module

4.7.1 Calendar

4.7.1.1 Description

A calendar contains time management artifacts that include occurrences and occurrence series.

4.7.1.2 Class Definition

The Calendar class has attribute values:

localNamespace

Value: icom_cal

localName

Value: Calendar

extendsFrom

Value: icom_core:Folder

stereotype

Value: primary

description

Value: A calendar contains time management artifacts that include occurrences and occurrence series.

propertyDefinitions

The values for this attribute are defined in 4.7.1.3.

4.7.1.3 Property Definitions

The Calendar class inherits property definitions from super classes.

The Calendar class MUST have the property definitions:

icom_core:timeZone

Description: Time zone setting for a calendar.

Required: True

Inherited: False

Property Type: icom_core:TimeZone

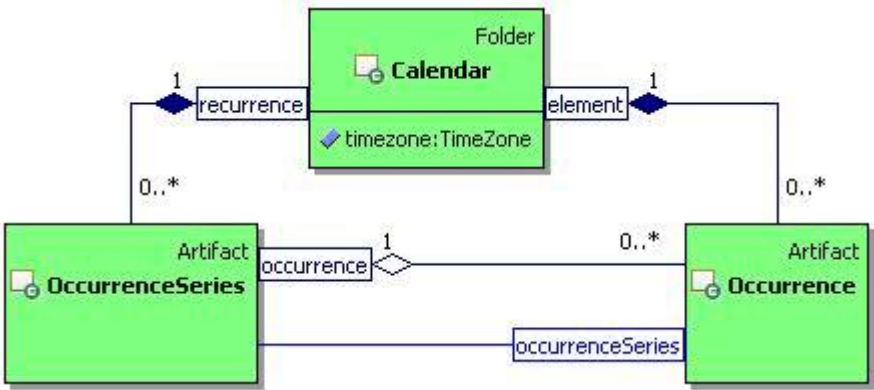
Cardinality: Single

Updatability: Read Write

icom_core:element

Description: Elements of a calendar.

5798	Required:	False
5799	Inherited:	True
5800	Property Type:	icom_cal:Occurrence
5801	Cardinality:	Multi
5802	Updatability:	Read Only
5803		
5804	icom_cal:recurrence	
5805	Description:	Occurrence series of a calendar.
5806	Required:	False
5807	Inherited:	False
5808	Property Type:	icom_cal:OccurrenceSeries
5809	Cardinality:	Multi
5810	Updatability:	Read Only
5811		



5812
5813 *Figure 35: Calendar Class Diagram.*
5814

5815 **4.7.2 OccurrenceSeries**

5816 **4.7.2.1 Description**

5817 An occurrence series represents a series of occurrences associated with the same calendar event.

5818 **4.7.2.2 Class Definition**

5819 The OccurrenceSeries class has attribute values:

5820	
5821	localNamespace
5822	Value: icom_cal
5823	
5824	localName
5825	Value: OccurrenceSeries
5826	
5827	extendsFrom
5828	Value: icom_core:Artifact

5829

5830 **stereotype**

5831 Value: primary

5832

5833 **description**

5834 Value: An occurrence series represents a series of occurrences associated with the same

5835 calendar event.

5836

5837 **propertyDefinitions**

5838 The values for this attribute are defined in 4.7.2.3.

5839 **4.7.2.3 Property Definitions**

5840 The OccurrenceSeries class inherits property definitions from super classes.

5841 The OccurrenceSeries class MUST have the property definitions:

5842

5843 **icom_core:location**

5844 Description: Location of an occurrence series.

5845 Required: False

5846 Inherited: False

5847 Property Type: icom_core:Location

5848 Cardinality: Single

5849 Updatability: Read Write

5850

5851 **icom_core:organizer**

5852 Description: Organizer of an occurrence series.

5853 Required: True

5854 Inherited: False

5855 Property Type: icom_core:Participant

5856 Cardinality: Single

5857 Updatability: On Create

5858

5859 **icom_core:participant**

5860 Description: Participants in an occurrence series.

5861 Required: False

5862 Inherited: False

5863 Property Type: icom_cal:OccurrenceParticipant

5864 Cardinality: Multi

5865 Updatability: Read Write

5866

5867 **icom_core:priority**

5868 Description: Priority for an attendee of an occurrence series.

5869 Required: False

5870 Inherited: False

5871	Property Type:	icom_core:Priority
5872	Cardinality:	Single
5873	Updatability:	Read Write
5874		
5875	icom_content:attachment	
5876	Description:	One or more content attachments in an occurrence series.
5877	Required:	False
5878	Inherited:	False
5879	Property Type:	icom_content:Attachment
5880	Cardinality:	Multi
5881	Updatability:	Read Write
5882		
5883	icom_cal:recurrenceStartDate	
5884	Description:	Start date and time of an occurrence series.
5885	Required:	True
5886	Inherited:	False
5887	Property Type:	DateTime
5888	Cardinality:	Single
5889	Updatability:	On Create
5890		
5891	icom_cal:recurrenceStartDateResolution	
5892	Description:	Resolution of start date and time of an occurrence series.
5893	Required:	True
5894	Inherited:	False
5895	Property Type:	icom_core:DateTimeResolution
5896	Cardinality:	Single
5897	Updatability:	On Create
5898		
5899	icom_cal:duration	
5900	Description:	Duration of each occurrence in an occurrence series.
5901	Required:	True
5902	Inherited:	False
5903	Property Type:	Duration
5904	Cardinality:	Single
5905	Updatability:	On Create
5906		
5907	icom_cal:recurrenceRule	
5908	Description:	A recurrence rule of an occurrence series.
5909	Required:	True
5910	Inherited:	False
5911	Property Type:	String
5912	Cardinality:	Single

5913	Updatability:	On Create
5914		
5915	icom_cal:occurrenceStatus	
5916	Description:	Status of an occurrence series.
5917	Required:	True
5918	Inherited:	False
5919	Property Type:	icom_cal:OccurrenceStatus
5920	Cardinality:	Single
5921	Updatability:	Read Write
5922		
5923	icom_cal:occurrenceType	
5924	Description:	Type of an occurrence series.
5925	Required:	True
5926	Inherited:	False
5927	Property Type:	icom_cal:OccurrenceType
5928	Cardinality:	Single
5929	Updatability:	Read Write
5930		
5931	icom_cal:editMode	
5932	Description:	Indicates a mode which determines whether an occurrence
5933		series is editable.
5934	Required:	False
5935	Inherited:	False
5936	Property Type:	icom_cal:OccurrenceEditMode
5937	Cardinality:	Single
5938	Updatability:	Read Only
5939		
5940	icom_cal:occurrence	
5941	Description:	Occurrences in an occurrence series.
5942	Required:	False
5943	Inherited:	False
5944	Property Type:	icom_cal:Occurrence
5945	Cardinality:	Multi
5946	Updatability:	Read Only
5947		
5948	icom_cal:attendee	
5949	Description:	An attendee of an occurrence series.
5950	Required:	False
5951	Inherited:	False
5952	Property Type:	icom_core:Participant
5953	Cardinality:	Single
5954	Updatability:	Read Only
5955		

5956	icom_cal:attendeeParticipantStatus	
5957	Description:	Participation status for an attendee of an occurrence series.
5958	Required:	False
5959	Inherited:	False
5960	Property Type:	icom_cal:OccurrenceParticipantStatus
5961	Cardinality:	Single
5962	Updatability:	Read Write
5963		
5964	icom_cal:transparency	
5965	Description:	Participant transparency for an attendee of an occurrence series.
5966		
5967	Required:	False
5968	Inherited:	False
5969	Property Type:	icom_cal:OccurrenceParticipantTransparency
5970	Cardinality:	Single
5971	Updatability:	Read Write
5972		
5973	icom_cal:attendeeProperty	
5974	Description:	Extensible properties for an attendee of an occurrence series.
5975	Required:	False
5976	Inherited:	False
5977	Property Type:	icom_meta:Property
5978	Cardinality:	Multi
5979	Updatability:	Read Write
5980		
5981	icom_conf:conference	
5982	Description:	One or more conferences for an occurrence series.
5983	Required:	False
5984	Inherited:	False
5985	Property Type:	icom_conf:Conference
5986	Cardinality:	Multi
5987	Updatability:	Read Write
5988		

6003

6004 **extendsFrom**

6005 Value: icom_core:Artifact

6006

6007 **stereotype**

6008 Value: primary

6009

6010 **description**

6011 Value: An occurrence represents an event in a calendar.

6012

6013 **propertyDefinitions**

6014 The values for this attribute are defined in 4.7.3.3.

6015

6016 4.7.3.3 Property Definitions

6017 The Occurrence class inherits property definitions from super classes.

6018 The Occurrence class MUST have the property definitions:

6019

6020 **icom_core:location**

6021	Description:	Location of an occurrence.
6022	Required:	False
6023	Inherited:	False
6024	Property Type:	icom_core:Location
6025	Cardinality:	Single
6026	Updatability:	Read Write

6027

6028 **icom_core:organizer**

6029	Description:	Organizer of an occurrence.
6030	Required:	True
6031	Inherited:	False
6032	Property Type:	icom_core:Participant
6033	Cardinality:	Single
6034	Updatability:	On Create

6035

6036 **icom_core:participant**

6037	Description:	Participants of an occurrence.
6038	Required:	False
6039	Inherited:	False
6040	Property Type:	icom_cal:OccurrenceParticipant
6041	Cardinality:	Multi
6042	Updatability:	Read Write

6043

6044	icom_core:priority	
6045	Description:	Priority for an attendee of an occurrence.
6046	Required:	False
6047	Inherited:	False
6048	Property Type:	icom_core:Priority
6049	Cardinality:	Single
6050	Updatability:	Read Write
6051		
6052	icom_core:startDate	
6053	Description:	Start date and time of an occurrence.
6054	Required:	True
6055	Inherited:	False
6056	Property Type:	DateTime
6057	Cardinality:	Single
6058	Updatability:	On Create
6059		
6060	icom_core:startDateResolution	
6061	Description:	Resolution of start date and time of an occurrence.
6062	Required:	True
6063	Inherited:	False
6064	Property Type:	icom_core:DateTimeResolution
6065	Cardinality:	Single
6066	Updatability:	On Create
6067		
6068	icom_core:endDate	
6069	Description:	End date and time of an occurrence.
6070	Required:	True
6071	Inherited:	False
6072	Property Type:	DateTime
6073	Cardinality:	Single
6074	Updatability:	On Create
6075		
6076	icom_core:endDateResolution	
6077	Description:	Resolution of end date and time of an occurrence.
6078	Required:	True
6079	Inherited:	False
6080	Property Type:	icom_core:DateTimeResolution
6081	Cardinality:	Single
6082	Updatability:	On Create
6083		
6084	icom_content:attachment	
6085	Description:	One or more content attachments in an occurrence.

6086	Required:	False
6087	Inherited:	False
6088	Property Type:	icom_content:Attachment
6089	Cardinality:	Multi
6090	Updatability:	Read Write
6091		
6092	icom_cal:occurrenceSeries	
6093	Description:	An occurrence is part of this occurrence series.
6094	Required:	False
6095	Inherited:	False
6096	Property Type:	icom_cal:OccurrenceSeries
6097	Cardinality:	Single
6098	Updatability:	Read Only
6099		
6100	icom_cal:fromRecurringOccurrenceSeries	
6101	Description:	Occurrence is part of a recurring occurrence series.
6102	Required:	False
6103	Inherited:	False
6104	Property Type:	Boolean
6105	Cardinality:	Single
6106	Updatability:	Read Only
6107		
6108	icom_cal:exceptionToOccurrenceSeries	
6109	Description:	Occurrence is an exception to an occurrence series.
6110	Required:	False
6111	Inherited:	False
6112	Property Type:	Boolean
6113	Cardinality:	Single
6114	Updatability:	Read Only
6115		
6116	icom_cal:occurrenceStatus	
6117	Description:	Status of an occurrence.
6118	Required:	True
6119	Inherited:	False
6120	Property Type:	icom_cal:OccurrenceStatus
6121	Cardinality:	Single
6122	Updatability:	Read Write
6123		
6124	icom_cal:occurrenceType	
6125	Description:	Type of an occurrence.
6126	Required:	True
6127	Inherited:	False

6128	Property Type:	icom_cal:OccurrenceType
6129	Cardinality:	Single
6130	Updatability:	Read Write
6131		
6132	icom_cal:editMode	
6133	Description:	Indicates a mode which determines whether an occurrence is
6134		editable.
6135	Required:	False
6136	Inherited:	False
6137	Property Type:	icom_cal:OccurrenceEditMode
6138	Cardinality:	Single
6139	Updatability:	Read Only
6140		
6141	icom_cal:attendee	
6142	Description:	An attendee of an occurrence.
6143	Required:	False
6144	Inherited:	False
6145	Property Type:	icom_core:Participant
6146	Cardinality:	Single
6147	Updatability:	Read Only
6148		
6149	icom_cal:attendeeParticipantStatus	
6150	Description:	Status for an attendee of an occurrence.
6151	Required:	False
6152	Inherited:	False
6153	Property Type:	icom_cal:OccurrenceParticipantStatus
6154	Cardinality:	Single
6155	Updatability:	Read Write
6156		
6157	icom_cal:transparency	
6158	Description:	Transparency for an attendee of an occurrence.
6159	Required:	False
6160	Inherited:	False
6161	Property Type:	icom_cal:OccurrenceParticipantTransparency
6162	Cardinality:	Single
6163	Updatability:	Read Write
6164		
6165	icom_cal:attendeeProperty	
6166	Description:	Extensible properties for an attendee of an occurrence.
6167	Required:	False
6168	Inherited:	False
6169	Property Type:	icom_meta:Property
6170	Cardinality:	Multi

6171 Updatability: Read Write

6172

6173 **icom_conf:conference**

6174 Description: One or more conferences for an occurrence.

6175 Required: False

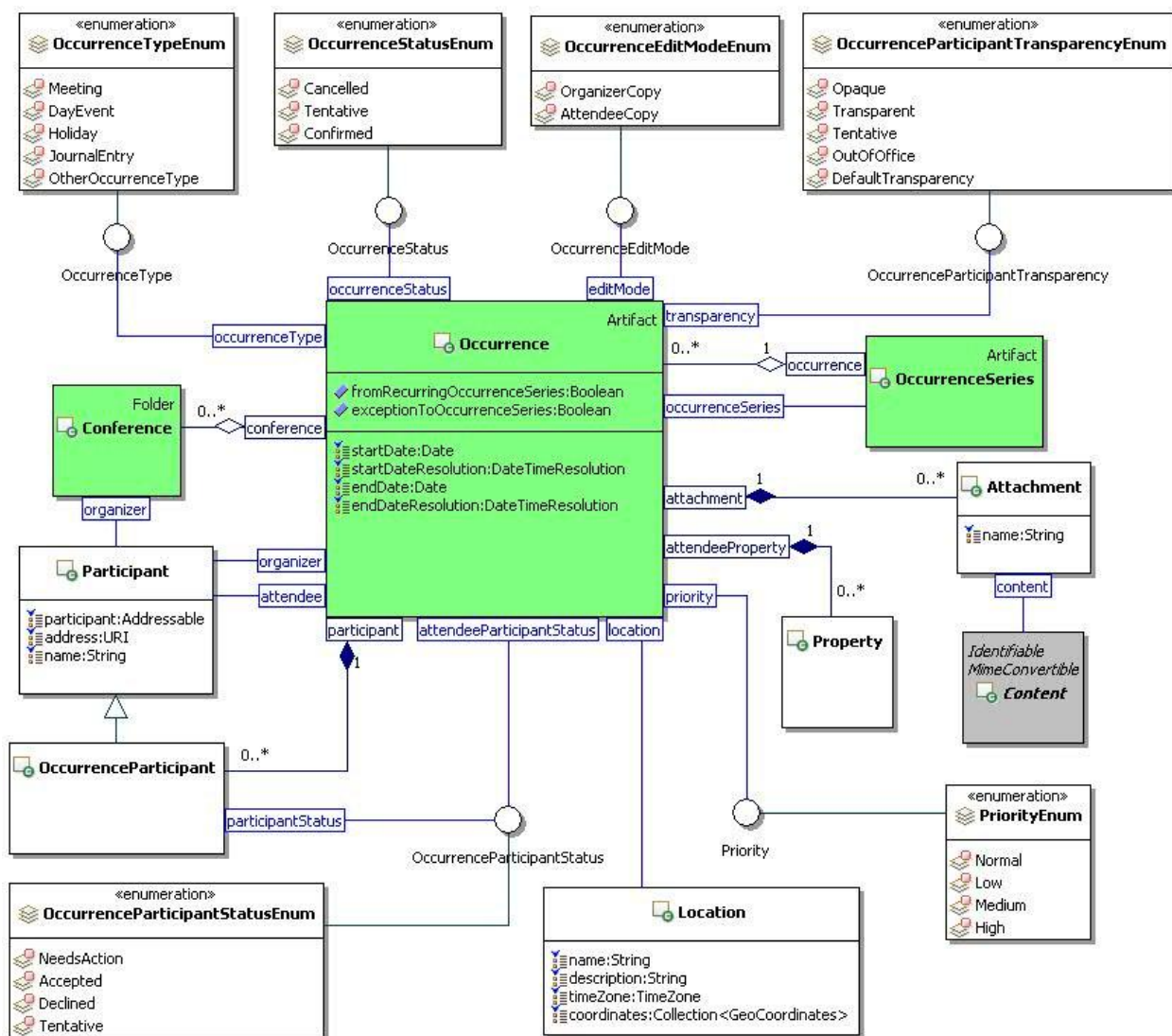
6176 Inherited: False

6177 Property Type: icom_conf:Conference

6178 Cardinality: Multi

6179 Updatability: Read Write

6180



6181

6182 Figure 37: Occurrence Class Diagram.

6183

6184 4.7.4 OccurrenceStatus

6185 4.7.4.1 Description

6186 An occurrence status is a status of a calendar occurrence.

6187 4.7.4.2 Class Definition

6188 The OccurrenceStatus class is a mixin class which defines status of a calendar occurrence.

6189 The OccurrenceStatus class has attribute values:

6190

6191 **localNamespace**

6192 Value: icom_cal

6193

6194 **localName**

6195 Value: OccurrenceStatus

6196

6197 **extendsFrom**

6198 Value:

6199

6200 **stereotype**

6201 Value: mixin

6202

6203 **description**

6204 Value: OccurrenceStatus is a mixin class which defines status of a calendar occurrence.

6205

6206 **propertyDefinitions**

6207 The values for this attribute are defined in Section 4.7.4.3.

6208 4.7.4.3 Property Definitions

6209 The OccurrenceStatus class MAY include additional property definitions which are implementation-
6210 defined.

6211

6212 4.7.5 OccurrenceStatusEnum

6213 The OccurrenceStatusEnum class is an enum class that enumerates the instances each of which
6214 expresses a status of an occurrence or occurrence series.

6215 The OccurrenceStatusEnum class has attribute values:

6216

6217 **localNamespace**

6218 Value: icom_cal

6219

6220 **localName**

6221 Value: OccurrenceStatusEnum

6222

6223 **extendsFrom**
6224 Value: OccurrenceStatus
6225
6226 **stereotype**
6227 Value: primary
6228
6229 **isEnumeration**
6230 Value: TRUE
6231
6232 **description**
6233 Value: Status of an occurrence or occurrence series.
6234
6235 **instances**
6236 Value: <icom_cal:Cancelled, icom_cal:Tentative, icom_cal:Confirmed>
6237
6238 ICOM defines three occurrence status:

- 6239 • **icom_cal:Cancelled** an occurrence or occurrence series is cancelled.
- 6240 • **icom_cal:Tentative** an occurrence or occurrence series is tentative.
- 6241 • **icom_cal:Confirmed** an occurrence or occurrence series is confirmed.

6242

6243 4.7.6 OccurrenceType

6244 4.7.6.1 Description

6245 An occurrence type is a category of calendar occurrences.

6246 4.7.6.2 Class Definition

6247 The OccurrenceType class is a mixin class which defines a type of occurrence.

6248 The OccurrenceType class has attribute values:

6249
6250 **localNamespace**
6251 Value: icom_cal
6252
6253 **localName**
6254 Value: OccurrenceType
6255
6256 **extendsFrom**
6257 Value:
6258
6259 **stereotype**
6260 Value: mixin
6261

6262 **description**
6263 Value: OccurrenceType is a mixin class which defines a type of occurrence.

6264
6265 **propertyDefinitions**
6266 The values for this attribute are defined in Section 4.7.6.3.

6267 **4.7.6.3 Property Definitions**

6268 The OccurrenceType class MAY include additional property definitions which are implementation-defined.
6269

6270 **4.7.7 OccurrenceTypeEnum**

6271 The OccurrenceTypeEnum class is an enum class that enumerates the instances each of which
6272 expresses a type of an occurrence or occurrence series.

6273 The OccurrenceTypeEnum class has attribute values:

6274
6275 **localNamespace**
6276 Value: icom_cal
6277

6278 **localName**
6279 Value: OccurrenceTypeEnum

6280
6281 **extendsFrom**
6282 Value: OccurrenceType
6283

6284 **stereotype**
6285 Value: primary
6286

6287 **isEnumeration**
6288 Value: TRUE
6289

6290 **description**
6291 Value: Type of an occurrence or occurrence series.
6292

6293 **instances**
6294 Value: <icom_cal:Meeting, icom_cal:DayEvent, icom_cal:Holiday, icom_cal:JournalEntry,
6295 icom_cal:OtherOccurrenceType>
6296

6297 ICOM defines five occurrence types:

- 6298 • **icom_cal:Meeting** an occurrence or occurrence series is a meeting.
- 6299 • **icom_cal:DayEvent** an occurrence or occurrence series is a day event.
- 6300 • **icom_cal:Holiday** an occurrence or occurrence series is a holiday.
- 6301 • **icom_cal:JournalEntry** an occurrence or occurrence series is a journal entry.
- 6302 • **icom_cal:OtherOccurrenceType** an occurrence or occurrence series is of other type.

6303
6304
6305
6306
6307
6308
6309
6310
6311
6312
6313
6314
6315
6316
6317
6318
6319
6320
6321
6322
6323
6324
6325
6326
6327
6328
6329
6330
6331
6332
6333
6334
6335
6336
6337
6338
6339

4.7.8 OccurrenceParticipant

4.7.8.1 Description

An occurrence participant object is a participant object that contains an occurrence participant status.

4.7.8.2 Class Definition

The OccurrenceParticipant class has attribute values:

localNamespace
Value: icom_cal
localName
Value: OccurrenceParticipant
extendsFrom
Value: icom_core:Participant
stereotype
Value: primary
description
Value: An occurrence participant object is a participant object that contains an occurrence participant status.
propertyDefinitions
The values for this attribute are defined in Section 4.7.8.3.

4.7.8.3 Property Definitions

The OccurrenceParticipant class inherits property definitions from super classes.

The OccurrenceParticipant class MUST have the property definition:

icom_cal:participantStatus	
Description:	Status of an occurrence participant.
Required:	False
Inherited:	False
Property Type:	icom_cal:OccurrenceParticipantStatus
Cardinality:	Single
Updatability:	Read Write

6340 4.7.9 OccurrenceParticipantStatus

6341 4.7.9.1 Description

6342 An occurrence participant status is a participant's response status for an occurrence or occurrence series.

6343 4.7.9.2 Class Definition

6344 The OccurrenceParticipantStatus class is a mixin class which defines a participant's response status for
6345 an occurrence or occurrence series.

6346 The OccurrenceParticipantStatus class has attribute values:

6347

6348 **localNamespace**

6349 Value: icom_cal

6350

6351 **localName**

6352 Value: OccurrenceParticipantStatus

6353

6354 **extendsFrom**

6355 Value:

6356

6357 **stereotype**

6358 Value: mixin

6359

6360 **description**

6361 Value: OccurrenceParticipantStatus is a mixin class which defines a participant's response
6362 status for an occurrence or occurrence series.

6363

6364 **propertyDefinitions**

6365 The values for this attribute are defined in Section 4.7.9.3.

6366 4.7.9.3 Property Definitions

6367 The OccurrenceParticipantStatus class MAY include additional property definitions which are
6368 implementation-defined.

6369

6370 4.7.10 OccurrenceParticipantStatusEnum

6371 The OccurrenceParticipantStatusEnum class is an enum class that enumerates the instances each of
6372 which expresses a participant's response status for an occurrence or occurrence series.

6373 The OccurrenceParticipantStatusEnum class has attribute values:

6374

6375 **localNamespace**

6376 Value: icom_cal

6377

6378 **localName**

6379 Value: OccurrenceParticipantStatusEnum

6380

6381 **extendsFrom**

6382 Value: OccurrenceParticipantStatus

6383

6384 **stereotype**

6385 Value: primary

6386

6387 **isEnumeration**

6388 Value: TRUE

6389

6390 **description**

6391 Value: Participant's response status for an occurrence or occurrence series.

6392

6393 **instances**

6394 Value: <icom_cal:NeedsAction, icom_cal:Accepted, icom_cal:Declined, icom_cal:Tentative>

6395

6396 ICOM defines four occurrence participant's status:

6397 • **icom_cal:NeedsAction** an attendee needs to act on an occurrence or occurrence series.

6398 • **icom_cal:Accepted** an attendee accepted an occurrence or occurrence series.

6399 • **icom_cal:Declined** an attendee declined an occurrence or occurrence series.

6400 • **icom_cal:Tentative** an attendee is tentative about attending an occurrence or occurrence series.

6401

6402 **4.7.11 OccurrenceParticipantTransparency**

6403 **4.7.11.1 Description**

6404 An occurrence participant transparency is visibility of an occurrence or occurrence series in a participant's

6405 calendar or free busy.

6406 **4.7.11.2 Class Definition**

6407 The OccurrenceParticipantTransparency class is a mixin class which defines visibility of an occurrence or

6408 occurrence series in a participant's calendar or free busy.

6409 The OccurrenceParticipantTransparency class has attribute values:

6410

6411 **localNamespace**

6412 Value: icom_cal

6413

6414 **localName**

6415 Value: OccurrenceParticipantTransparency

6416

6417 **extendsFrom**

6418 Value:

6419

6420 **stereotype**
6421 Value: mixin
6422
6423 **description**
6424 Value: OccurrenceParticipantTransparency is a mixin class which defines visibility of an
6425 occurrence or occurrence series in a participant's calendar or free busy.
6426
6427 **propertyDefinitions**
6428 The values for this attribute are defined in Section 4.7.11.3.

6429 **4.7.11.3 Property Definitions**

6430 The OccurrenceParticipantTransparency class MAY include additional property definitions which are
6431 implementation-defined.
6432

6433 **4.7.12 OccurrenceParticipantTransparencyEnum**

6434 The OccurrenceParticipantTransparencyEnum class is an enum class that enumerates the instances
6435 each of which expresses an occurrence or occurrence series transparency in a participant's calendar or
6436 free busy.
6437 The OccurrenceParticipantTransparencyEnum class has attribute values:

6438
6439 **localNamespace**
6440 Value: icom_cal
6441
6442 **localName**
6443 Value: OccurrenceParticipantTransparencyEnum
6444
6445 **extendsFrom**
6446 Value: OccurrenceParticipantTransparency
6447
6448 **stereotype**
6449 Value: primary
6450
6451 **isEnumeration**
6452 Value: TRUE
6453
6454 **description**
6455 Value: Occurrence or occurrence series transparency in a participant's calendar or free busy.
6456
6457 **instances**
6458 Value: <icom_cal:Opaque, icom_cal:Transparent, icom_cal:Tentative, icom_cal:OutOfOffice,
6459 icom_cal:DefaultTransparency>
6460

6461 ICOM defines five participant transparencies:

- 6462 • **icom_cal:Opaque** an occurrence or occurrence series is opaque in a participant's calendar or
- 6463 free busy.
- 6464 • **icom_cal:Transparent** an occurrence or occurrence series is transparent in a participant's
- 6465 calendar or free busy.
- 6466 • **icom_cal:Tentative** an occurrence or occurrence series has a tentative transparency in a
- 6467 participant's calendar or free busy.
- 6468 • **icom_cal:OutOfOffice** an occurrence or occurrence series has out of office transparency in a
- 6469 participant's calendar or free busy.
- 6470 • **icom_cal:DefaultTransparency** an occurrence or occurrence series has default transparency in
- 6471 a participant's calendar or free busy.
- 6472

6473 4.7.13 OccurrenceEditMode

6474 4.7.13.1 Description

6475 An occurrence edit mode is a mode that indicates whether an occurrence or occurrence series is editable.

6476 4.7.13.2 Class Definition

6477 The OccurrenceEditMode class is a mixin class which defines a mode that indicates whether an

6478 occurrence or occurrence series is editable.

6479 The OccurrenceEditMode class has attribute values:

6480

6481 **localNamespace**

6482 Value: icom_cal

6483

6484 **localName**

6485 Value: OccurrenceEditMode

6486

6487 **extendsFrom**

6488 Value:

6489

6490 **stereotype**

6491 Value: mixin

6492

6493 **description**

6494 Value: OccurrenceEditMode is a mixin class which defines a mode that indicates whether an

6495 occurrence or occurrence series is editable.

6496

6497 **propertyDefinitions**

6498 The values for this attribute are defined in Section 4.7.13.3.

6499 4.7.13.3 Property Definitions

6500 The OccurrenceEditMode class MAY include additional property definitions which are implementation-

6501 defined.

6502

6503 4.7.14 OccurrenceEditModeEnum

6504 The OccurrenceEditModeEnum class is an enum class that enumerates the instances each of which
6505 expresses a mode that indicates whether an occurrence or occurrence series is editable.

6506 The OccurrenceEditModeEnum class has attribute values:

6507

6508 **localNamespace**

6509 Value: icom_cal

6510

6511 **localName**

6512 Value: OccurrenceEditModeEnum

6513

6514 **extendsFrom**

6515 Value: OccurrenceEditMode

6516

6517 **stereotype**

6518 Value: primary

6519

6520 **isEnumeration**

6521 Value: TRUE

6522

6523 **description**

6524 Value: A mode that indicates whether an occurrence or occurrence series is editable.

6525

6526 **instances**

6527 Value: <icom_cal:OrganizerCopy, icom_cal:AttendeeCopy>

6528

6529 ICOM defines two occurrence editable modes:

- 6530 • **icom_cal:OrganizerCopy** an occurrence or occurrence series is a copy created by an organizer
6531 who may update the properties such as occurrence type, occurrence status, etc.
- 6532 • **icom_cal:AttendeeCopy** an occurrence or occurrence series is a copy delivered to an attendee
6533 who may only update the attendee properties such as priority, transparency, etc .

6534

6535 4.8 Free Busy Module

6536 4.8.1 FreeBusy

6537 4.8.1.1 Description

6538 A free busy object specifies the free time and busy time intervals of one or more participants.

6539 4.8.1.2 Class Definition

6540 The FreeBusy class has attribute values:

6541

6542 **localNamespace**

6543 Value: icom_cal

6544

6545 **localName**

6546 Value: FreeBusy

6547

6548 **extendsFrom**

6549 Value:

6550

6551 **stereotype**

6552 Value: primary

6553

6554 **description**

6555 Value: A free busy object specifies the free time and busy time intervals of one or more

6556 participants.

6557

6558 **propertyDefinitions**

6559 The values for this attribute are defined in Section 4.8.1.3.

6560 **4.8.1.3 Property Definitions**

6561 The FreeBusy class MUST have the property definitions:

6562

6563 **icom_core:participant**

6564 Description: A list of participants whose free busy intervals are included.

6565 Required: False

6566 Inherited: False

6567 Property Type: icom_core:Participant

6568 Cardinality: Multi

6569 Updatability: Read Only

6570

6571 **icom_core:creationDate**

6572 Description: Creation date and time of a free busy object.

6573 Required: False

6574 Inherited: False

6575 Property Type: DateTime

6576 Cardinality: Single

6577 Updatability: Read Only

6578

6579 **icom_core:startDate**

6580 Description: Start date and time of a list of free busy intervals.

6581 Required: False

6582 Inherited: False

6583	Property Type:	DateTime
6584	Cardinality:	Single
6585	Updatability:	Read Only
6586		
6587	icom_core:endDate	
6588	Description:	End date and time of a list of free busy intervals.
6589	Required:	False
6590	Inherited:	False
6591	Property Type:	DateTime
6592	Cardinality:	Single
6593	Updatability:	Read Only
6594		
6595	icom_cal:interval	
6596	Description:	A list of free busy intervals.
6597	Required:	False
6598	Inherited:	False
6599	Property Type:	icom_cal:FreeBusyInterval
6600	Cardinality:	Multi
6601	Updatability:	Read Only
6602		

6603 4.8.2 FreeBusyInterval

6604 4.8.2.1 Description

6605 A free busy interval specifies an interval of free or busy time.

6606 If a free busy type is icom_cal:Free, then a time interval is free for scheduling.

6607 If a free busy type is icom_cal:Busy, then a time interval is busy because one or more events have been
6608 scheduled for the interval.

6609 4.8.2.2 Class Definition

6610 The FreeBusyInterval class has attribute values:

6611		
6612	localNamespace	
6613	Value:	icom_cal
6614		
6615	localName	
6616	Value:	FreeBusyInterval
6617		
6618	extendsFrom	
6619	Value:	
6620		
6621	stereotype	
6622	Value:	primary

6623

6624 **description**

6625 Value: A free busy interval object specifies an interval of free or busy time.

6626

6627 **propertyDefinitions**

6628 The values for this attribute are defined in Section 4.8.2.3

6629 4.8.2.3 Property Definitions

6630 The FreeBusyInterval class MUST have the property definitions:

6631

6632 **icom_core:startDate**

6633 Description:	Start date and time of a free busy interval.
6634 Required:	False
6635 Inherited:	False
6636 Property Type:	DateTime
6637 Cardinality:	Single
6638 Updatability:	Read Only

6639

6640 **icom_core:endDate**

6641 Description:	End date and time of a free busy interval.
6642 Required:	False
6643 Inherited:	False
6644 Property Type:	DateTime
6645 Cardinality:	Single
6646 Updatability:	Read Only

6647

6648 **icom_cal:freeBusyType**

6649 Description:	A type of free busy interval.
6650 Required:	False
6651 Inherited:	False
6652 Property Type:	icom_cal:FreeBusyType
6653 Cardinality:	Single
6654 Updatability:	Read Only

6655

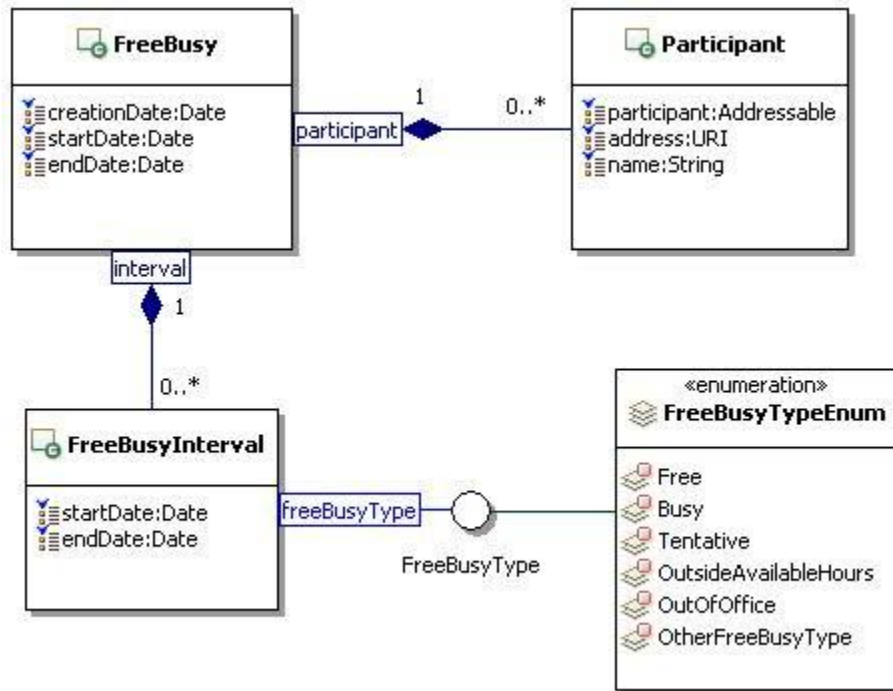


Figure 38: Free Busy Class Diagram.

4.8.3 FreeBusyType

4.8.3.1 Description

A free busy type classifies a time interval as free, busy, or other.

4.8.3.2 Class Definition

The FreeBusyType class is a mixin class which defines different types to classify a time interval.

The FreeBusyType class has attribute values:

localNamespace

Value: icom_cal

localName

Value: FreeBusyType

extendsFrom

Value:

stereotype

Value: mixin

description

Value: FreeBusyType is a mixin class which defines different types to classify a time interval.

6680
6681 **propertyDefinitions**
6682 The values for this attribute are defined in Section 4.8.3.3.

6683 **4.8.3.3 Property Definitions**

6684 The FreeBusyType class MAY include additional property definitions which are implementation-defined.
6685

6686 **4.8.4 FreeBusyTypeEnum**

6687 The FreeBusyTypeEnum class is an enum class that enumerates the instances each of which expresses
6688 a type of free busy interval.

6689 The FreeBusyTypeEnum class has attribute values:

6690
6691 **localNamespace**
6692 Value: icom_cal
6693
6694 **localName**
6695 Value: FreeBusyTypeEnum
6696
6697 **extendsFrom**
6698 Value: FreeBusyType
6699
6700 **stereotype**
6701 Value: primary
6702
6703 **isEnumeration**
6704 Value: TRUE
6705
6706 **description**
6707 Value: A type of free busy interval.
6708
6709 **instances**
6710 Value: <icom_cal:Free, icom_cal:Busy, icom_cal:Tentative, icom_cal:OutsideAvailableHours,
6711 icom_cal:OutOfOffice, icom_cal:OtherFreeBusyType>
6712

6713 ICOM defines six free busy types:

- 6714 • **icom_cal:Free** a free busy interval is free.
- 6715 • **icom_cal:Busy** a free busy interval is busy.
- 6716 • **icom_cal:Tentative** a free busy interval is tentative.
- 6717 • **icom_cal:OutsideAvailableHours** a free busy interval is outside available hours.
- 6718 • **icom_cal:OutOfOffice** a free busy interval is within out of office hours.
- 6719 • **icom_cal:OtherFreeBusyType** a free busy interval is of other type.

6720

4.9 Task List Module

4.9.1 TaskList

4.9.1.1 Description

A task list contains task management artifacts.

4.9.1.2 Class Definition

The TaskList class has attribute values:

localNamespace

Value: icom_task

localName

Value: TaskList

extendsFrom

Value: icom_core:Folder

stereotype

Value: primary

description

Value: A task list contains task management artifacts.

propertyDefinitions

The values for this attribute are defined in 4.9.1.3.

4.9.1.3 Property Definitions

The TaskList class inherits property definitions from super classes.

The TaskList class MUST have the property definitions:

icom_core:timeZone

Description: Time zone of a task list.

Required: True

Inherited: False

Property Type: icom_core:TimeZone

Cardinality: Single

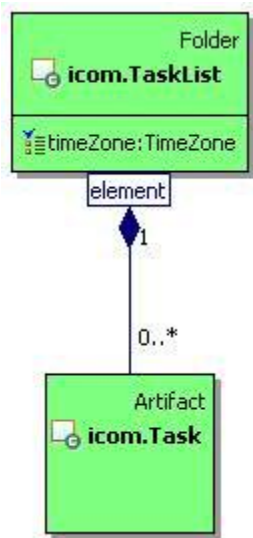
Updatability: Read Write

icom_core:element

Description: Elements of a task list.

Required: False

6760 Inherited: True
6761 Property Type: icom_task:Task
6762 Cardinality: Multi
6763 Updatability: Read Only
6764



6765
6766 Figure 39: Task List Class Diagram.
6767

6768 **4.9.2 Task**

6769 **4.9.2.1 Description**

6770 A task is an artifact that represents a task to do or a task assignment in a task list.

6771 **4.9.2.2 Class Definition**

6772 The Task class has attribute values:

6773
6774 **localNamespace**
6775 Value: icom_task
6776
6777 **localName**
6778 Value: Task
6779
6780 **extendsFrom**
6781 Value: icom_core:Artifact
6782
6783 **stereotype**
6784 Value: primary
6785
6786 **description**
6787 Value: A task is an artifact that represents a task to do or a task assignment in a task list.

6788
6789 **propertyDefinitions**
6790 The values for this attribute are defined in 4.9.2.3.

6791 **4.9.2.3 Property Definitions**

6792 The Task class inherits property definitions from super classes.
6793 The Task class MUST have the property definitions:

6794
6795 **icom_core:location**
6796 Description: Location of a task.
6797 Required: False
6798 Inherited: False
6799 Property Type: icom_core:Location
6800 Cardinality: Single
6801 Updatability: Read Write

6802
6803 **icom_core:organizer**
6804 Description: Organizer of a task.
6805 Required: True
6806 Inherited: False
6807 Property Type: icom_core:Participant
6808 Cardinality: Single
6809 Updatability: On Create

6810
6811 **icom_core:priority**
6812 Description: Priority of a task.
6813 Required: False
6814 Inherited: False
6815 Property Type: icom_core:Priority
6816 Cardinality: Single
6817 Updatability: Read Write

6818
6819 **icom_core:startDate**
6820 Description: Start date and time of a task.
6821 Required: True
6822 Inherited: False
6823 Property Type: DateTime
6824 Cardinality: Single
6825 Updatability: On Create

6826
6827 **icom_core:startDateResolution**
6828 Description: Resolution of start date and time of a task.
6829 Required: True

6830	Inherited:	False
6831	Property Type:	icom_core:DateTimeResolution
6832	Cardinality:	Single
6833	Updatability:	On Create
6834		
6835	icom_content:attachment	
6836	Description:	One or more content attachments in a task.
6837	Required:	False
6838	Inherited:	False
6839	Property Type:	icom_content:Attachment
6840	Cardinality:	Multi
6841	Updatability:	Read Write
6842		
6843	icom_task:dueDate	
6844	Description:	Due date and time of a task.
6845	Required:	True
6846	Inherited:	False
6847	Property Type:	DateTime
6848	Cardinality:	Single
6849	Updatability:	On Create
6850		
6851	icom_task:dueDateResolution	
6852	Description:	Resolution of due date and time of a task.
6853	Required:	True
6854	Inherited:	False
6855	Property Type:	icom_core:DateTimeResolution
6856	Cardinality:	Single
6857	Updatability:	On Create
6858		
6859	icom_task:editMode	
6860	Description:	Indicates a mode which determines whether a task is
6861		editable.
6862	Required:	False
6863	Inherited:	False
6864	Property Type:	icom_task:TaskEditMode
6865	Cardinality:	Single
6866	Updatability:	Read Only
6867		
6868	icom_task:taskStatus	
6869	Description:	Status of a task.
6870	Required:	True
6871	Inherited:	False
6872	Property Type:	icom_task:TaskStatus

6873	Cardinality:	Single
6874	Updatability:	Read Write
6875		
6876	icom_task:assignee	
6877	Description:	An assignee of a task.
6878	Required:	False
6879	Inherited:	False
6880	Property Type:	icom_core:Participant
6881	Cardinality:	Single
6882	Updatability:	Read Only
6883		
6884	icom_task:participantStatus	
6885	Description:	Participation status of a task.
6886	Required:	False
6887	Inherited:	False
6888	Property Type:	icom_task:TaskParticipantStatus
6889	Cardinality:	Single
6890	Updatability:	Read Write
6891		
6892	icom_task:completionDate	
6893	Description:	Completion date and time of a task.
6894	Required:	False
6895	Inherited:	False
6896	Property Type:	DateTime
6897	Cardinality:	Single
6898	Updatability:	Read Write
6899		
6900	icom_task:completionDateResolution	
6901	Description:	Resolution of completion date and time of a task.
6902	Required:	False
6903	Inherited:	False
6904	Property Type:	icom_core:DateTimeResolution
6905	Cardinality:	Single
6906	Updatability:	Read Write
6907		
6908	icom_task:percentComplete	
6909	Description:	Percentage of task completed.
6910	Required:	False
6911	Inherited:	False
6912	Property Type:	Integer
6913	Cardinality:	Single
6914	Updatability:	Read Write

icom_task:assigneeProperty

Description:	Extensible properties for an assignee of a task.
Required:	False
Inherited:	False
Property Type:	icom_meta:Property
Cardinality:	Multi
Updatability:	Read Write

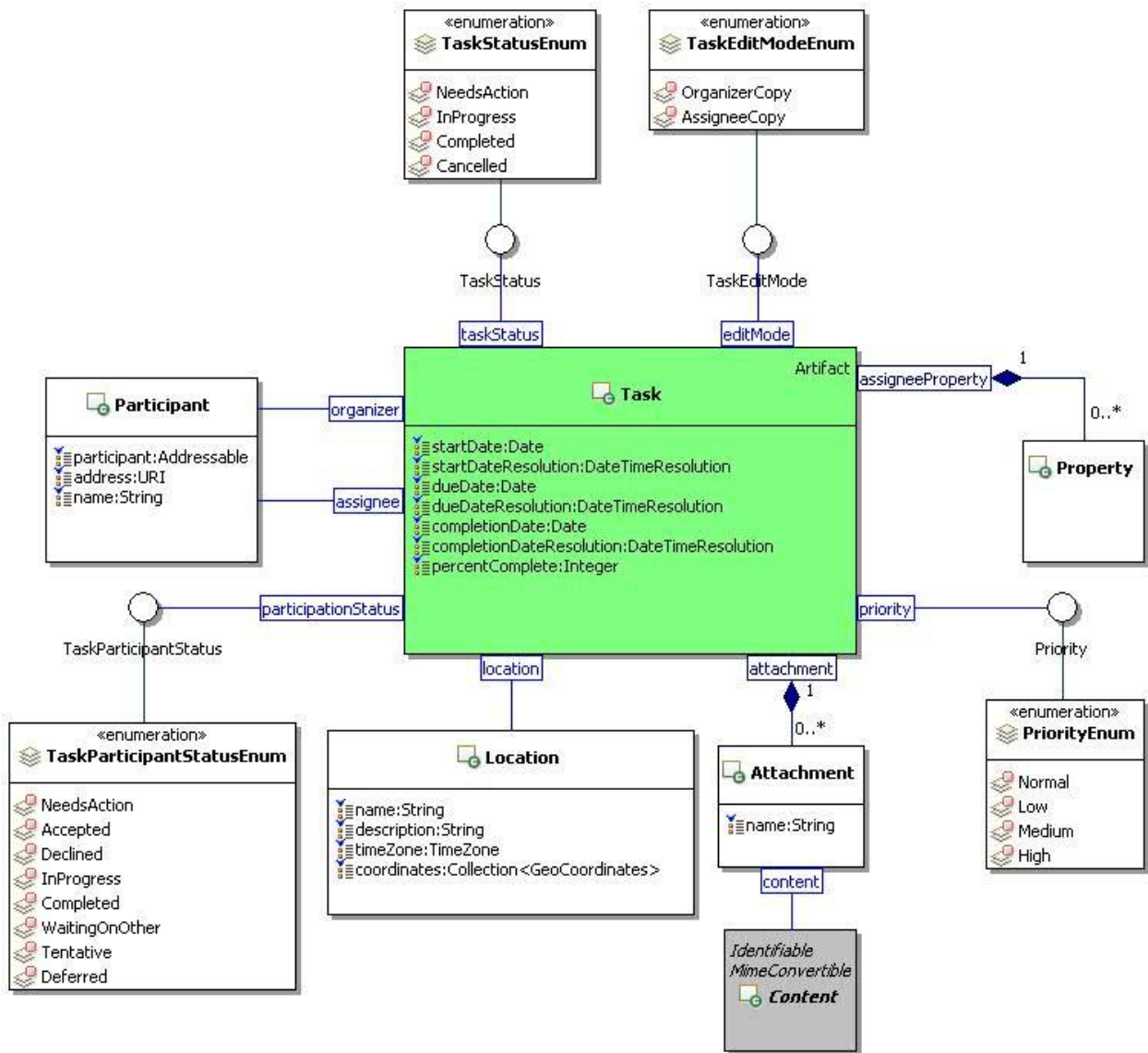


Figure 40: Task Class Diagram.

6927 4.9.3 TaskStatus

6928 4.9.3.1 Description

6929 A task status is a status of a task.

6930 4.9.3.2 Class Definition

6931 The TaskStatus class is a mixin class which defines status of a task.

6932 The TaskStatus class has attribute values:

6933

6934 **localNamespace**

6935 Value: icom_task

6936

6937 **localName**

6938 Value: TaskStatus

6939

6940 **extendsFrom**

6941 Value:

6942

6943 **stereotype**

6944 Value: mixin

6945

6946 **description**

6947 Value: TaskStatus is a mixin class which defines status of a task.

6948

6949 **propertyDefinitions**

6950 The values for this attribute are defined in Section 4.9.3.3.

6951 4.9.3.3 Property Definitions

6952 The TaskStatus class MAY include additional property definitions which are implementation-defined.

6953

6954 4.9.4 TaskStatusEnum

6955 The TaskStatusEnum class is an enum class that enumerates the instances each of which expresses a
6956 status of task.

6957 The TaskStatusEnum class has attribute values:

6958

6959 **localNamespace**

6960 Value: icom_task

6961

6962 **localName**

6963 Value: TaskStatusEnum

6964

6965 **extendsFrom**

6966 Value: TaskStatus

6967

6968 **stereotype**

6969 Value: primary

6970

6971 **isEnumeration**

6972 Value: TRUE

6973

6974 **description**

6975 Value: Status of a task.

6976

6977 **instances**

6978 Value: <icom_task:NeedsAction, icom_task:InProgress, icom_task:Completed,

6979 icom_task:Cancelled>

6980

6981 ICOM defines four task status:

- 6982 • **icom_task:NeedsAction** a task needs action.
- 6983 • **icom_task:InProgress** a task is in progress.
- 6984 • **icom_task:Completed** a task is completed.
- 6985 • **icom_task:Cancelled** a task is cancelled.

6986

6987 **4.9.5 TaskParticipantStatus**

6988 **4.9.5.1 Description**

6989 A task participant status is a participant's response status for a task assignment.

6990 **4.9.5.2 Class Definition**

6991 The TaskParticipantStatus class is a mixin class which defines a participant's response status for a task

6992 assignment.

6993 The TaskParticipantStatus class has attribute values:

6994

6995 **localNamespace**

6996 Value: icom_task

6997

6998 **localName**

6999 Value: TaskParticipantStatus

7000

7001 **extendsFrom**

7002 Value:

7003

7004 **stereotype**

7005 Value: mixin

7006

7007 **description**
7008 Value: TaskParticipantStatus is a mixin class which defines a participant's response status for a
7009 task assignment.

7010
7011 **propertyDefinitions**
7012 The values for this attribute are defined in Section 4.9.5.3.

7013 **4.9.5.3 Property Definitions**

7014 The TaskParticipantStatus class MAY include additional property definitions which are implementation-
7015 defined.

7016

7017 **4.9.6 TaskParticipantStatusEnum**

7018 The TaskParticipantStatusEnum class is an enum class that enumerates the instances each of which
7019 expresses a participant's response status for a task.

7020 The TaskParticipantStatusEnum class has attribute values:

7021
7022 **localNamespace**
7023 Value: icom_task
7024

7025 **localName**
7026 Value: TaskParticipantStatusEnum

7027
7028 **extendsFrom**
7029 Value: TaskParticipantStatus
7030

7031 **stereotype**
7032 Value: primary
7033

7034 **isEnumeration**
7035 Value: TRUE
7036

7037 **description**
7038 Value: Participant's response status for a task.

7039
7040 **instances**
7041 Value: <icom_task:NeedsAction, icom_task:Accepted, icom_task:Declined,
7042 icom_task:InProgress, icom_task:Completed, icom_task:WaitingOnOther, icom_task:Tentative,
7043 icom_task:Deferred>
7044

7045 ICOM defines eight task participant's status:

- 7046 • **icom_task:NeedsAction** an assignee needs to act on a task.
7047 • **icom_task:Accepted** an assignee accepted a task.
7048 • **icom_task:Declined** an assignee declined a task.

- 7049 • **icom_task:InProgress** a task is in progress.
- 7050 • **icom_task:Completed** a task is completed.
- 7051 • **icom_task:WaitingOnOther** an assignee is waiting on other.
- 7052 • **icom_task:Tentative** an assignee is tentative about a task.
- 7053 • **icom_task:Deferred** an assignee deferred a task.

7054

7055 **4.9.7 TaskEditMode**

7056 **4.9.7.1 Description**

7057 A task edit mode is a mode that indicates whether a task is editable.

7058 **4.9.7.2 Class Definition**

7059 The TaskEditMode class is a mixin class which defines a mode that indicates whether a task is editable.

7060 The TaskEditMode class has attribute values:

7061

7062 **localNamespace**

7063 Value: icom_task

7064

7065 **localName**

7066 Value: TaskEditMode

7067

7068 **extendsFrom**

7069 Value:

7070

7071 **stereotype**

7072 Value: mixin

7073

7074 **description**

7075 Value: TaskEditMode is a mixin class which defines a mode that indicates whether task is
7076 editable.

7077

7078 **propertyDefinitions**

7079 The values for this attribute are defined in Section 4.9.7.3.

7080 **4.9.7.3 Property Definitions**

7081 The TaskEditMode class MAY include additional property definitions which are implementation-defined.

7082

7083 **4.9.8 TaskEditModeEnum**

7084 The TaskEditModeEnum class is an enum class that enumerates the instances each of which expresses
7085 an editable mode of a task.

7086 The TaskEditModeEnum class has attribute values:

7087

7088 **localNamespace**
7089 Value: icom_task
7090
7091 **localName**
7092 Value: TaskEditModeEnum
7093
7094 **extendsFrom**
7095 Value: TaskEditMode
7096
7097 **stereotype**
7098 Value: primary
7099
7100 **isEnumeration**
7101 Value: TRUE
7102
7103 **description**
7104 Value: A mode that indicates whether a task is editable.
7105
7106 **instances**
7107 Value: <icom_task:OrganizerCopy, icom_task:AssigneeCopy>
7108

7109 ICOM defines two task editable modes:

- 7110 • **icom_task:OrganizerCopy**: a task is a copy created by an organizer who may update the
- 7111 properties such as start time, due time.
- 7112 • **icom_task:AssigneeCopy**: a task is a copy delivered to an assignee who may only update the
- 7113 assignee properties such as completion time, participant status, percent completed.
- 7114

7115 4.10 Forum Module

7116 4.10.1 Discussion

7117 4.10.1.1 Description

7118 A discussion is an item in a discussion container.

7119 4.10.1.2 Class Definition

7120 The Discussion class is a mixin class that defines the characteristics of artifacts that can be elements of
7121 discussion containers.

7122 The Discussion class has attribute values:

7123
7124 **localNamespace**
7125 Value: icom_forum
7126

7127 **localName**
7128 Value: Discussion
7129
7130 **extendsFrom**
7131 Value: icom_core:Item
7132
7133 **stereotype**
7134 Value: mixin
7135
7136 **description**
7137 Value: Discussion is a mixin class that defines the characteristics of artifacts that can be placed
7138 in a discussion container.
7139
7140 **propertyDefinitions**
7141 The values for this attribute are defined in Section 4.10.1.3.

7142 **4.10.1.3 Property Definitions**

7143 The Discussion class inherits property definitions from super classes.
7144 The Discussion class MUST have the property definition:

7145
7146 **icom_forum:inReplyTo**
7147 Description: Another discussion object that a discussion object is replying
7148 to.
7149 Required: False
7150 Inherited: False
7151 Property Type: icom_forum:Discussion
7152 Cardinality: Single
7153 Updatability: Read Write
7154

7155 The Discussion class MAY include additional property definitions which are implementation-defined.
7156

7157 **4.10.2 DiscussionContainer**

7158 **4.10.2.1 Description**

7159 A discussion container contains discussion items.

7160 **4.10.2.2 Class Definition**

7161 The DiscussionContainer class is a mixin class that defines the characteristics of folders that contain
7162 Discussion items.

7163 The DiscussionContainer class has attribute values:

7164
7165 **localNamespace**
7166 Value: icom_forum

7167

7168 **localName**

7169 Value: DiscussionContainer

7170

7171 **extendsFrom**

7172 Value: icom_core:Container

7173

7174 **stereotype**

7175 Value: mixin

7176

7177 **description**

7178 Value: DiscussionContainer is a mixin class that defines the characteristics of folders that

7179 contain Discussion items.

7180

7181 **propertyDefinitions**

7182 The values for this attribute are defined in Section 4.10.2.3.

7183 **4.10.2.3 Property Definitions**

7184 The DiscussionContainer class inherits property definitions from super classes.

7185 The DiscussionContainer class **MUST** have the property definition:

7186

7187 **icom_core:element**

7188 Description:	Elements of a discussion container.
7189 Required:	False
7190 Inherited:	True
7191 Property Type:	icom_forum:Discussion
7192 Cardinality:	Multi
7193 Updatability:	Read Only

7194

7195 The DiscussionContainer class **MAY** include additional property definitions which are implementation-

7196 defined.

7197

7198 **4.10.3 DiscussionMessage**

7199 **4.10.3.1 Description**

7200 A discussion message is a message in a forum discussion thread.

7201 **4.10.3.2 Class Definition**

7202 The DiscussionMessage class has attribute values:

7203

7204 **localNamespace**

7205 Value: icom_forum

7206

7207 **localName**
7208 Value: DiscussionMessage
7209
7210 **extendsFrom**
7211 Value: icom_msg:Message, icom_forum:Discussion
7212
7213 **stereotype**
7214 Value: primary
7215
7216 **description**
7217 Value: Discussion message is a message in a forum discussion thread.
7218
7219 **propertyDefinitions**
7220 The values for this attribute are defined in Section 4.10.3.3.

7221 **4.10.3.3 Property Definitions**

7222 The DiscussionMessage class inherits property definitions from super classes.
7223 The DiscussionMessage class MUST have the property definition:

7224
7225 **icom_forum:inReplyTo**
7226 Description: Another discussion message that a discussion message is
7227 replying to.
7228 Required: False
7229 Inherited: True
7230 Property Type: icom_forum:DiscussionMessage
7231 Cardinality: Single
7232 Updatability: Read Write

7233
7234 The DiscussionMessage class MAY include additional property definitions which are implementation-
7235 defined.
7236

7237 **4.10.4 TopicContainer**

7238 **4.10.4.1 Description**

7239 A topic container contains topics.

7240 **4.10.4.2 Class Definition**

7241 The TopicContainer class is a mixin class which defines the characteristics of folders that contain Topics.
7242 The TopicContainer class has attribute values:

7243
7244 **localNamespace**
7245 Value: icom_forum
7246

7247 **localName**
7248 Value: TopicContainer
7249
7250 **extendsFrom**
7251 Value: icom_core:Container
7252
7253 **stereotype**
7254 Value: mixin
7255
7256 **description**
7257 Value: TopicContainer is a mixin class that defines the characteristics of folders that contain
7258 topics.
7259
7260 **propertyDefinitions**
7261 The values for this attribute are defined in Section 4.10.4.3.

7262 **4.10.4.3 Property Definitions**

7263 The TopicContainer class inherits property definitions from super classes.
7264 The TopicContainer class **MUST** have the property definitions:

7265
7266 **icom_core:element**
7267 Description: Elements of a topic container.
7268 Required: False
7269 Inherited: True
7270 Property Type: icom_forum:Topic
7271 Cardinality: Multi
7272 Updatability: Read Only

7273
7274 The TopicContainer class **MAY** include additional property definitions which are implementation-defined.
7275

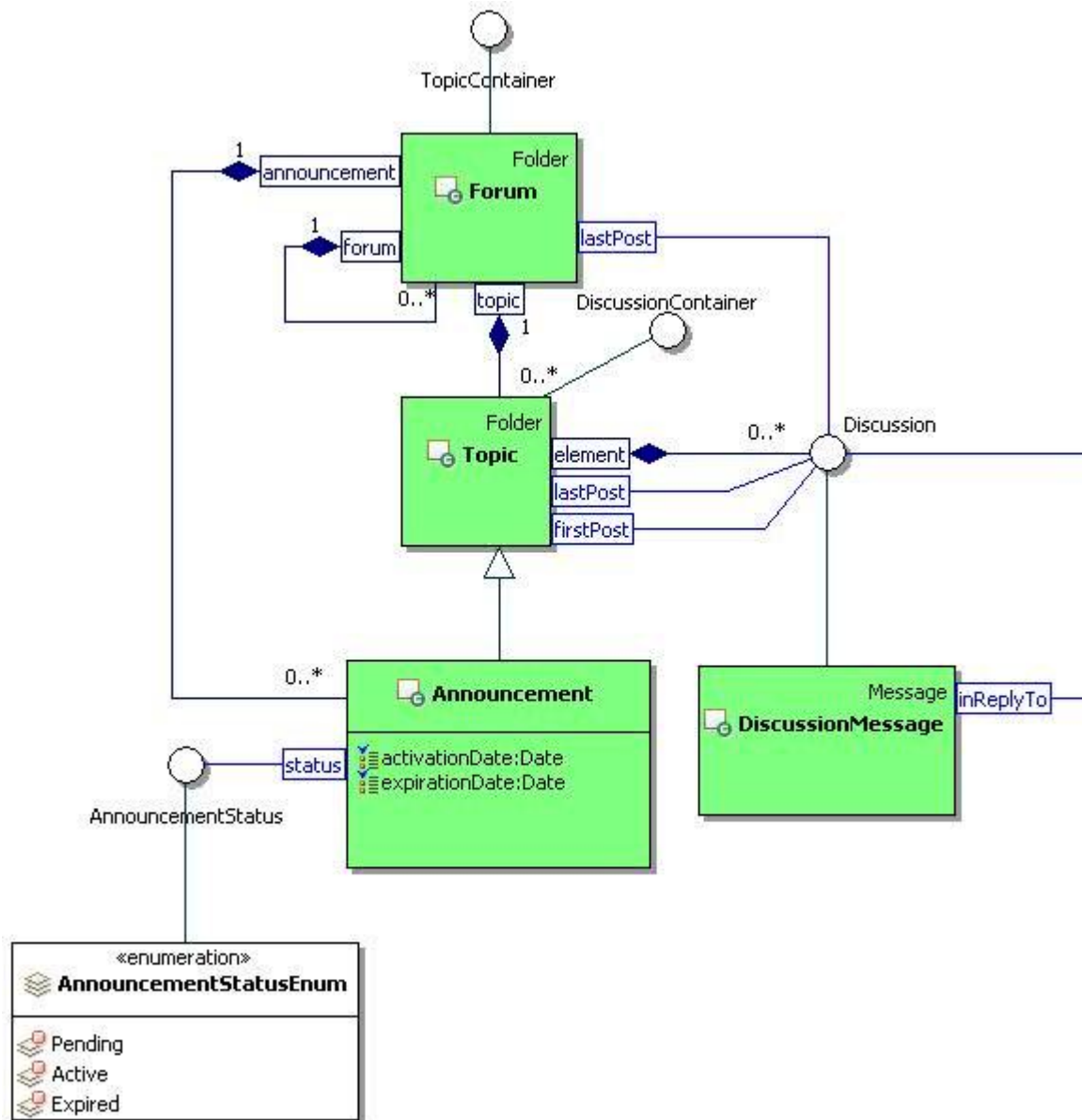


Figure 41: Forum Class Diagram.

4.10.5 Forum

4.10.5.1 Description

A forum contains sub-forums, topics, and announcements.

4.10.5.2 Class Definition

The Forum class has attribute values:

localNamespace

Value: icom_forum

7288 **localName**
7289 Value: Forum
7290
7291 **extendsFrom**
7292 Value: icom_core:Folder, icom_forum:TopicContainer
7293
7294 **stereotype**
7295 Value: primary
7296
7297 **description**
7298 Value: A forum contains sub-forums, topics, and announcements.
7299
7300 **propertyDefinitions**
7301 The values for this attribute are defined in Section 4.10.5.3.

7302 **4.10.5.3 Property Definitions**

7303 The Forum class inherits property definitions from super classes.
7304 The Forum class MUST have the property definitions:

7305
7306 **icom_forum:lastPost**
7307 Description: The last posted discussion in a forum.
7308 Required: False
7309 Inherited: False
7310 Property Type: icom_forum:Discussion
7311 Cardinality: Single
7312 Updatability: Read Only
7313
7314 **icom_forum:forum**
7315 Description: Sub-forums of a forum.
7316 Required: False
7317 Inherited: False
7318 Property Type: icom_forum:Forum
7319 Cardinality: Multi
7320 Updatability: Read Only
7321
7322 **icom_forum:topic**
7323 Description: Topics of a forum.
7324 Required: False
7325 Inherited: False
7326 Property Type: icom_forum:Topic
7327 Cardinality: Multi
7328 Updatability: Read Only
7329

7330 **icom_forum:announcement**

7331 Description: Announcements of a forum.

7332 Required: False

7333 Inherited: False

7334 Property Type: icom_forum:Announcement

7335 Cardinality: Multi

7336 Updatability: Read Only

7337

7338 The Forum class MAY include additional property definitions which are implementation-defined.

7339

7340 **4.10.6 Topic**

7341 **4.10.6.1 Description**

7342 A topic contains conversations among forum participants. The discussions in a topic may be sorted in
7343 chronological order or threaded by reply.

7344 **4.10.6.2 Class Definition**

7345 The Topic class has attribute values:

7346

7347 **localNamespace**

7348 Value: icom_forum

7349

7350 **localName**

7351 Value: Topic

7352

7353 **extendsFrom**

7354 Value: icom_core:Folder, icom_forum:DiscussionContainer

7355

7356 **stereotype**

7357 Value: primary

7358

7359 **description**

7360 Value: A topic contains discussion threads.

7361

7362 **propertyDefinitions**

7363 The values for this attribute are defined in Section 4.10.6.3.

7364 **4.10.6.3 Property Definitions**

7365 The Topic class inherits property definitions from super classes.

7366 The Topic class MUST have the property definitions:

7367

7368 **icom_core:element**

7369 Description: Elements of a topic.

7370	Required:	False
7371	Inherited:	True
7372	Property Type:	icom_forum:Discussion
7373	Cardinality:	Multi
7374	Updatability:	Read Only
7375		
7376	icom_forum:firstPost	
7377	Description:	The first posted discussion in a topic.
7378	Required:	False
7379	Inherited:	False
7380	Property Type:	icom_forum:Discussion
7381	Cardinality:	Single
7382	Updatability:	Read Only

7383		
7384	icom_forum:lastPost	
7385	Description:	The last posted discussion in a topic.
7386	Required:	False
7387	Inherited:	False
7388	Property Type:	icom_forum:Discussion
7389	Cardinality:	Single
7390	Updatability:	Read Only

7391

7392 The Topic class MAY include additional property definitions which are implementation-defined.

7393

7394 4.10.7 Announcement

7395 4.10.7.1 Description

7396 An announcement contains time-sensitive discussion posts that are valid for a specified period of time,
7397 depending on activation and expiration times.

7398 4.10.7.2 Class Definition

7399 The Announcement class has attribute values:

7400

7401 **localNamespace**

7402 Value: icom_forum

7403

7404 **localName**

7405 Value: Announcement

7406

7407 **extendsFrom**

7408 Value: icom_forum:Topic

7409

7410 **stereotype**
7411 Value: primary
7412
7413 **description**
7414 Value: An announcement contains discussion items that are valid for a specified period of time.
7415
7416 **propertyDefinitions**
7417 The values for this attribute are defined in Section 4.10.7.3.

7418 **4.10.7.3 Property Definitions**

7419 The Announcement class inherits property definitions from super classes.
7420 The Announcement class **MUST** have the property definitions:

7421
7422 **icom_forum:activationDate**
7423 Description: Date and time when an announcement becomes active.
7424 Required: False
7425 Inherited: False
7426 Property Type: DateTime
7427 Cardinality: Single
7428 Updatability: Read Write
7429

7430 **icom_forum:expirationDate**
7431 Description: Date and time when an announcement expires.
7432 Required: False
7433 Inherited: False
7434 Property Type: DateTime
7435 Cardinality: Single
7436 Updatability: Read Write
7437

7438 **icom_forum:announcementStatus**
7439 Description: Status of an announcement.
7440 Required: True
7441 Inherited: False
7442 Property Type: icom_forum:AnnouncementStatus
7443 Cardinality: Single
7444 Updatability: Read Write
7445

7446 The Announcement class **MAY** include additional property definitions which are implementation-defined.
7447

7448 **4.10.8 AnnouncementStatus**

7449 **4.10.8.1 Description**

7450 An announcement status is status of an announcement.

7451 **4.10.8.2 Class Definition**

7452 The AnnouncementStatus class is a mixin class which defines status of an announcement.

7453 The AnnouncementStatus class has attribute values:

7454

7455 **localNamespace**

7456 Value: icom_forum

7457

7458 **localName**

7459 Value: AnnouncementStatus

7460

7461 **extendsFrom**

7462 Value:

7463

7464 **stereotype**

7465 Value: mixin

7466

7467 **description**

7468 Value: AnnouncementStatus is a mixin class which defines status of an announcement.

7469

7470 **propertyDefinitions**

7471 The values for this attribute are defined in Section 4.10.8.3.

7472 **4.10.8.3 Property Definitions**

7473 The AnnouncementStatus class MAY include additional property definitions which are implementation-
7474 defined.

7475

7476 **4.10.9 AnnouncementStatusEnum**

7477 The AnnouncementStatusEnum class is an enum class that enumerates the instances each of which
7478 expresses a status of announcement.

7479 The AnnouncementStatusEnum class has attribute values:

7480

7481 **localNamespace**

7482 Value: icom_forum

7483

7484 **localName**

7485 Value: AnnouncementStatusEnum

7486

7487 **extendsFrom**
7488 Value: AnnouncementStatus
7489
7490 **stereotype**
7491 Value: primary
7492
7493 **isEnumeration**
7494 Value: TRUE
7495
7496 **description**
7497 Value: Status of announcement.
7498
7499 **instances**
7500 Value: <icom_forum:Pending, icom_forum:Active, icom_forum:Expired>
7501
7502 ICOM defines three announcement status:

- 7503 • **icom_forum:Pending** an announcement is pending.
- 7504 • **icom_forum:Active** an announcement is active.
- 7505 • **icom_forum:Expired** an announcement is expired.

7506

7507 4.11 Conference Module

7508 4.11.1 Conference

7509 4.11.1.1 Description

7510 A conference is a container that represents a durable context for conference sessions.
7511 It contains conference metadata, settings, and transcripts.

7512 4.11.1.2 Class Definition

7513 The Conference class has attribute values:

7514
7515 **localNamespace**
7516 Value: icom_conf
7517
7518 **localName**
7519 Value: Conference
7520
7521 **extendsFrom**
7522 Value: icom_core:Folder
7523
7524 **stereotype**
7525 Value: primary

7526

7527 **description**

7528 Value: A conference represents a durable context for online conference sessions.

7529

7530 **propertyDefinitions**

7531 The values for this attribute are defined in Section 4.11.1.3.

7532 **4.11.1.3 Property Definitions**

7533 The Conference class inherits property definitions from super classes.

7534 The Conference class MUST have the property definitions:

7535

7536 **icom_core:organizer**

7537 Description: Organizer of a conference.

7538 Required: False

7539 Inherited: False

7540 Property Type: icom_core:Participant

7541 Cardinality: Single

7542 Updatability: On Create

7543

7544 **icom_conf:conferenceType**

7545 Description: Type of a conference.

7546 Required: False

7547 Inherited: False

7548 Property Type: icom_conf:ConferenceType

7549 Cardinality: Single

7550 Updatability: Read Write

7551

7552 **icom_conf:conferenceStatus**

7553 Description: Status of a conference.

7554 Required: False

7555 Inherited: False

7556 Property Type: icom_conf:ConferenceStatus

7557 Cardinality: Single

7558 Updatability: Read Only

7559

7560 **icom_conf:runningSession**

7561 Description: Current session of a conference.

7562 Required: False

7563 Inherited: False

7564 Property Type: icom_conf:ConferenceSession

7565 Cardinality: Single

7566 Updatability: Read Only

7567

7568	icom_conf:conferenceSetting	
7569	Description:	Configurable settings of a conference.
7570	Required:	False
7571	Inherited:	False
7572	Property Type:	icom_conf:ConferenceSetting
7573	Cardinality:	Single
7574	Updatability:	Read Only
7575		
7576	icom_conf:transcript	
7577	Description:	Transcripts from ended sessions of a conference.
7578	Required:	False
7579	Inherited:	False
7580	Property Type:	icom_doc:Document
7581	Cardinality:	Multi
7582	Updatability:	Read Write
7583		
7584	icom_conf:scheduledStartDate	
7585	Description:	Scheduled start date and time of a conference session.
7586	Required:	False
7587	Inherited:	False
7588	Property Type:	DateTime
7589	Cardinality:	Single
7590	Updatability:	Read Write
7591		
7592	icom_conf:scheduledEndDate	
7593	Description:	Scheduled end date and time of a conference session.
7594	Required:	False
7595	Inherited:	False
7596	Property Type:	DateTime
7597	Cardinality:	Single
7598	Updatability:	Read Write
7599		
7600	The Conference class MAY include additional property definitions which are implementation-defined.	
7601		

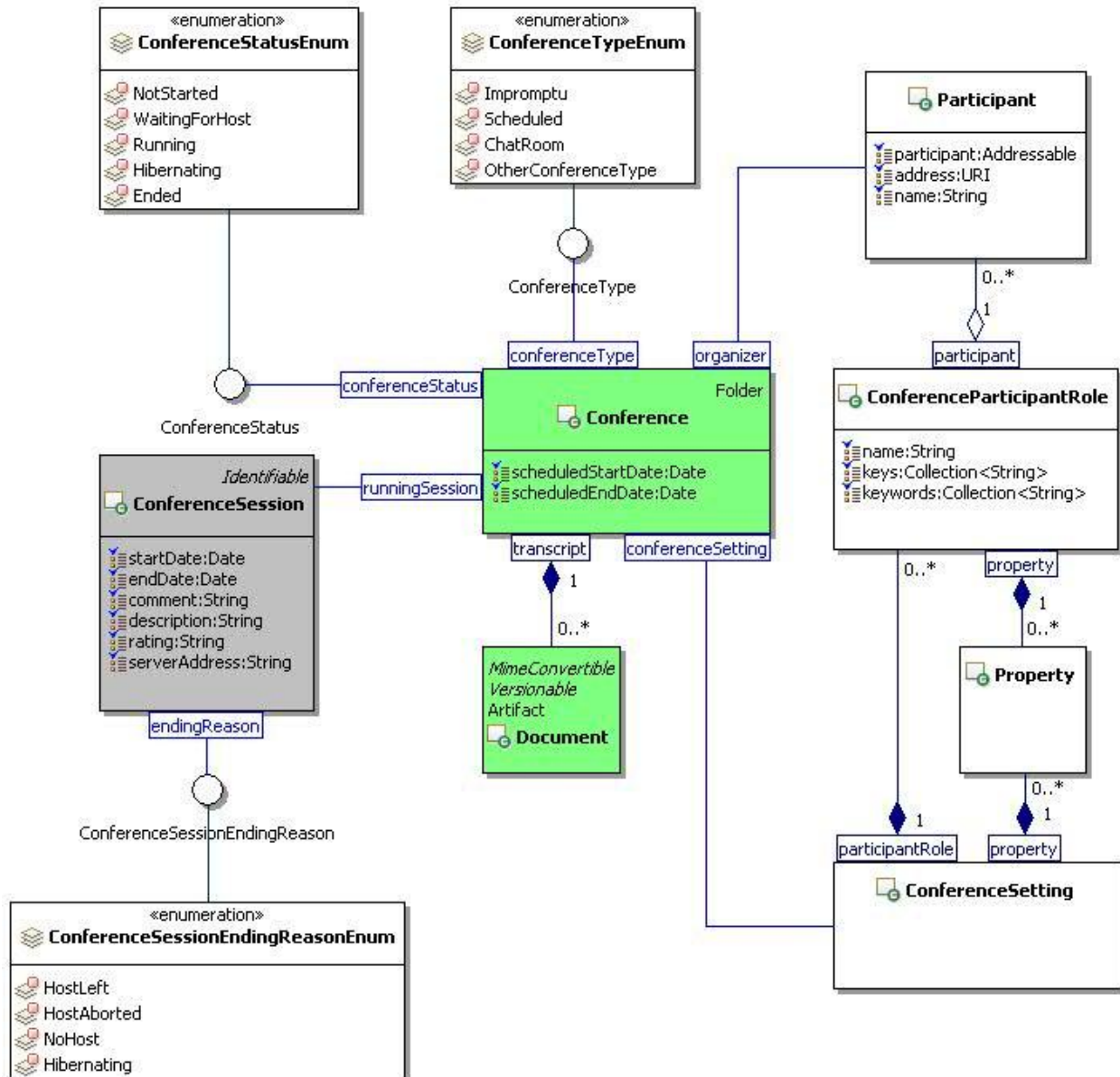


Figure 42: Conference Class Diagram.

4.11.2 ConferenceType

4.11.2.1 Description

A conference type represents a category of conferences.

4.11.2.2 Class Definition

The ConferenceType class is a mixin class which defines a type of conference.

The ConferenceType class has attribute values:

localNamespace

Value: icom_conf

7614

7615 **localName**

7616 Value: ConferenceType

7617

7618 **extendsFrom**

7619 Value:

7620

7621 **stereotype**

7622 Value: mixin

7623

7624 **description**

7625 Value: ConferenceType is a mixin class which defines type of conference.

7626

7627 **propertyDefinitions**

7628 The values for this attribute are defined in Section 4.11.2.3.

7629 **4.11.2.3 Property Definitions**

7630 The ConferenceType class MAY include additional property definitions which are implementation-defined.

7631

7632 **4.11.3 ConferenceTypeEnum**

7633 The ConferenceTypeEnum class is an enum class that enumerates the instances each of which

7634 expresses a type of a conference.

7635 The ConferenceTypeEnum class has attribute values:

7636

7637 **localNamespace**

7638 Value: icom_conf

7639

7640 **localName**

7641 Value: ConferenceTypeEnum

7642

7643 **extendsFrom**

7644 Value: ConferenceType

7645

7646 **stereotype**

7647 Value: primary

7648

7649 **isEnumeration**

7650 Value: TRUE

7651

7652 **description**

7653 Value: A type of a conference.

7654

7655 **instances**
7656 Value: <icom_conf:Impromptu, icom_conf:Scheduled, icom_conf:ChatRoom,
7657 icom_conf:OtherConferenceType>
7658

7659 ICOM defines four conference types:

- 7660 • **icom_conf:Impromptu** a conference session is started impromptu.
 - 7661 • **icom_conf:Scheduled** a conference session is scheduled.
 - 7662 • **icom_conf:ChatRoom** a conference is used for a chat room.
 - 7663 • **icom_conf:OtherConferenceType** a conference is of other type.
- 7664

7665 **4.11.4 ConferenceStatus**

7666 **4.11.4.1 Description**

7667 A conference status is status of an online conference.

7668 **4.11.4.2 Class Definition**

7669 The ConferenceStatus class is a mixin class which defines status of an online conference.

7670 The ConferenceStatus class has attribute values:

7671

7672 **localNamespace**
7673 Value: icom_conf
7674

7675 **localName**
7676 Value: ConferenceStatus
7677

7678 **extendsFrom**
7679 Value:
7680

7681 **stereotype**
7682 Value: mixin
7683

7684 **description**
7685 Value: ConferenceStatus is a mixin class which defines status of an online conference.
7686

7687 **propertyDefinitions**
7688 The values for this attribute are defined in Section 4.11.4.3.

7689 **4.11.4.3 Property Definitions**

7690 The ConferenceStatus class MAY include additional property definitions which are implementation-
7691 defined.

7692

4.11.5 ConferenceStatusEnum

The ConferenceStatusEnum class is an enum class that enumerates the instances each of which expresses a status of a conference.

The ConferenceStateEnum class has attribute values:

localNamespace

Value: icom_conf

localName

Value: ConferenceStatusEnum

extendsFrom

Value: ConferenceStatus

stereotype

Value: primary

isEnumeration

Value: TRUE

description

Value: Status of a conference.

instances

Value: <icom_conf:NotStarted, icom_conf:WaitingForHost, icom_conf:Running, icom_conf:Hibernate, icom_conf:Ended>

ICOM defines five conference status:

- **icom_conf:NotStarted** a conference session is not started .
- **icom_conf:WaitingForHost** a conference session is waiting for a host.
- **icom_conf:Running** a conference session is running.
- **icom_conf:Hibernate** a conference session is hibernating.
- **icom_conf:Ended** a conference session is ended.

4.11.6 ConferenceSession

4.11.6.1 Description

A conference session represents the metadata for a session of a conference.

4.11.6.2 Class Definition

The ConferenceSession class has attribute values:

7733 **localNamespace**
7734 Value: icom_conf
7735
7736 **localName**
7737 Value: ConferenceSession
7738
7739 **extendsFrom**
7740 Value: icom_core:Identifiable
7741
7742 **stereotype**
7743 Value: primary
7744
7745 **description**
7746 Value: A conference session represents the metadata for a session of a conference.
7747
7748 **propertyDefinitions**
7749 The values for this attribute are defined in Section 4.11.6.3.

7750 **4.11.6.3 Property Definitions**

7751 The ConferenceSession class inherits property definitions from super classes.

7752 The ConferenceSession class MUST have the property definitions:

7753
7754 **icom_core:startDate**
7755 Description: Start date and time of a conference session.
7756 Required: False
7757 Inherited: False
7758 Property Type: DateTime
7759 Cardinality: Single
7760 Updatability: Read Only
7761
7762 **icom_core:endDate**
7763 Description: End date and time of a conference session.
7764 Required: False
7765 Inherited: False
7766 Property Type: DateTime
7767 Cardinality: Single
7768 Updatability: Read Only
7769
7770 **icom_conf:comment**
7771 Description: Comment on a conference session.
7772 Required: False
7773 Inherited: False
7774 Property Type: String

7775	Cardinality:	Single
7776	Updatability:	Read Write
7777		
7778	icom_conf:description	
7779	Description:	Description of a conference session.
7780	Required:	False
7781	Inherited:	False
7782	Property Type:	String
7783	Cardinality:	Single
7784	Updatability:	Read Write
7785		
7786	icom_conf:rating	
7787	Description:	Rating of a conference session.
7788	Required:	False
7789	Inherited:	False
7790	Property Type:	String
7791	Cardinality:	Single
7792	Updatability:	Read Write
7793		
7794	icom_conf:serverAddress	
7795	Description:	Address of a server that hosts a conference session.
7796	Required:	False
7797	Inherited:	False
7798	Property Type:	String
7799	Cardinality:	Single
7800	Updatability:	Read Only
7801		
7802	icom_conf:endingReason	
7803	Description:	Reason for ending a conference session.
7804	Required:	False
7805	Inherited:	False
7806	Property Type:	icom_conf:ConferenceSessionEndingReason
7807	Cardinality:	Single
7808	Updatability:	Read Only
7809		
7810	The ConferenceSession class MAY include additional property definitions which are implementation-	
7811	defined.	
7812		

7813 4.11.7 ConferenceSessionEndingReason

7814 4.11.7.1 Description

7815 A conference session ending reason is an indication of how a conference session ended.

4.11.7.2 Class Definition

The ConferenceSessionEndingReason class is a mixin class which defines an indication of how a conference session ended.

The ConferenceSessionEndingReason class has attribute values:

localNamespace

Value: icom_conf

localName

Value: ConferenceSessionEndingReason

extendsFrom

Value:

stereotype

Value: mixin

description

Value: ConferenceSessionEndingReason is a mixin class which defines an indication of how a conference session ended.

propertyDefinitions

The values for this attribute are defined in Section 4.11.7.3.

4.11.7.3 Property Definitions

The ConferenceSessionEndingReason class MAY include additional property definitions which are implementation-defined.

4.11.8 ConferenceSessionEndingReasonEnum

The ConferenceSessionEndingReasonEnum class is an enum class that enumerates the instances each of which expresses a reason for ending a conference session.

The ConferenceSessionEndingReasonEnum class has attribute values:

localNamespace

Value: icom_conf

localName

Value: ConferenceSessionEndingReasonEnum

extendsFrom

Value: ConferenceSessionEndingReason

7857 **stereotype**
7858 Value: primary
7859
7860 **isEnumeration**
7861 Value: TRUE
7862
7863 **description**
7864 Value: Reason for ending a conference session.
7865
7866 **instances**
7867 Value: <icom_conf:HostLeft, icom_conf:HostAborted, icom_conf:NoHost, icom_conf:Hibernating>
7868
7869 ICOM defines four conference session states:

- 7870 • **icom_conf:HostLeft** a conference session ended after the host left.
- 7871 • **icom_conf:HostAborted** a conference session ended after the host aborted it.
- 7872 • **icom_conf:NoHost** a conference session ended due to no one hosting.
- 7873 • **icom_conf:Hibernating** a conference session is hibernating.

7874

7875 4.11.9 ConferenceSetting

7876 4.11.9.1 Description

7877 A conference setting represents configuration settings for sessions of a conference.

7878 4.11.9.2 Class Definition

7879 The ConferenceSetting class has attribute values:

7880
7881 **localNamespace**
7882 Value: icom_conf
7883
7884 **localName**
7885 Value: ConferenceSetting
7886
7887 **extendsFrom**
7888 Value:
7889
7890 **stereotype**
7891 Value: primary
7892
7893 **description**
7894 Value: A conference setting represents configuration settings for sessions of a conference.
7895

7896 **propertyDefinitions**

7897 The values for this attribute are defined in Section 4.11.9.3.

7898 **4.11.9.3 Property Definitions**

7899 The ConferenceSetting class inherits property definitions from super classes.

7900 The ConferenceSetting class MUST have the property definitions:

7901

7902 **icom_meta:property**

7903 Description: Configurable properties for a conference.

7904 Required: False

7905 Inherited: False

7906 Property Type: icom_meta:property

7907 Cardinality: Multi

7908 Updatability: Read Write

7909

7910 **icom_conf:participantRole**

7911 Description: Role settings for conference participants.

7912 Required: False

7913 Inherited: False

7914 Property Type: icom_conf:ConferenceParticipantRole

7915 Cardinality: Multi

7916 Updatability: Read Write

7917

7918 The ConferenceSetting class MAY include additional property definitions which are implementation-
7919 defined.

7920

7921 **4.11.10 ConferenceParticipantRole**

7922 **4.11.10.1 Description**

7923 A conference participant role defines roles settings for a conference participant.

7924 **4.11.10.2 Class Definition**

7925 The ConferenceParticipantRole class has attribute values:

7926

7927 **localNamespace**

7928 Value: icom_conf

7929

7930 **localName**

7931 Value: ConferenceParticipantRole

7932

7933 **extendsFrom**

7934 Value:

7935

7936 **stereotype**
7937 Value: primary
7938
7939 **description**
7940 Value: A conference participant role contains roles settings for a conference.
7941
7942 **propertyDefinitions**
7943 The values for this attribute are defined in Section 4.11.10.3.

7944 **4.11.10.3 Property Definitions**

7945 The ConferenceParticipantRole class MUST have the property definitions:

7946

7947 **icom_core:name**
7948 Description: Name of a role setting in a conference.
7949 Required: False
7950 Inherited: False
7951 Property Type: String
7952 Cardinality: Single
7953 Updatability: Read Write

7954

7955 **icom_core:participant**
7956 Description: One or more participants in a role setting.
7957 Required: False
7958 Inherited: False
7959 Property Type: icom_core:Participant
7960 Cardinality: Multi
7961 Updatability: Read Write

7962

7963 **icom_meta:property**
7964 Description: Configurable properties for a role setting.
7965 Required: False
7966 Inherited: False
7967 Property Type: icom_meta:Property
7968 Cardinality: Multi
7969 Updatability: Read Write

7970

7971 **icom_conf:key**
7972 Description: One or more sign on keys to activate a role setting.
7973 Required: False
7974 Inherited: False
7975 Property Type: String
7976 Cardinality: Multi
7977 Updatability: Read Write

7978

7979 **icom_conf:keyword**

7980	Description:	One or more key words to activate a role setting.
7981	Required:	False
7982	Inherited:	False
7983	Property Type:	String
7984	Cardinality:	Multi
7985	Updatability:	Read Write

7986

7987 The ConferenceParticipantRole class MAY include additional property definitions which are

7988 implementation-defined.

5 Conformance

The ICOM specification does not presume a particular software architecture or framework for use of the ICOM model.

Conformance to the ICOM specification is defined using use case roles played by the following four stakeholders of a typical software architecture or framework:

1. An ICOM platform provider supplies an environment for one or more ICOM service providers, producers, and consumers to exchange ICOM objects.
2. An ICOM service provider manages objects produced by one or more ICOM producers for access by one or more ICOM consumers.
3. An ICOM producer creates objects managed by an ICOM service provider.
4. An ICOM consumer accepts objects managed by an ICOM service provider.

Fulfillment of ICOM use case roles and accompanying responsibilities is implementation dependent. An ICOM implementation may fulfill one or more of ICOM use case roles and accompanying responsibilities.

Conformance by platform provider:

1. An ICOM platform provider:
 - a. SHALL conform to all mandatory statements and
 - b. MAY conform to optional statementsof the core ICOM model as defined in Section 3 of this standard
2. An ICOM platform provider:
 - a. SHALL conform to all mandatory statements and
 - b. MAY conform to optional statementsas defined in Section 4 for each extension module.

Conformance by service provider:

1. An ICOM service provider MAY support one or more extension modules as defined in Section 4 of this standard.
2. An ICOM service provider that supports an extension module:
 - a. SHALL conform to all mandatory statements and
 - b. MAY conform to optional statementsas defined in Section 4 for that extension module.
3. Depending on the classes extended by an extension module, an ICOM service provider:
 - a. SHALL conform to all mandatory statements and
 - b. MAY conform to optional statementsfor inherited super classes and related classes defined in Section 3 of this standard.

Note: ICOM environment may include multiple service providers each of which provides different subsets of extension modules.

- 8031 Conformance by ICOM producer:
- 8032 1. An ICOM producer that produces objects of a class:
- 8033 a. SHALL conform to all mandatory statements and
- 8034 b. MAY conform to optional statements
- 8035 for the class and super classes thereof in Section 3 of this standard, for any object produced.
- 8036 2. An ICOM producer may support one or more extension modules as defined in Section 4 of this
- 8037 standard. ICOM producers that support an extension module:
- 8038 a. SHALL conform to all mandatory statements and
- 8039 b. MAY conform to optional statements
- 8040 as defined in Section 4 for that extension module.
- 8041
- 8042 Conformance by ICOM consumer:
- 8043 1. An ICOM consumer that consumes objects of a class:
- 8044 a. SHALL conform to all mandatory statements and
- 8045 b. MAY conform to optional statements
- 8046 for the class and super classes thereof in Section 3 of this standard, for any object consumed.
- 8047 2. An ICOM consumer may support one or more extension modules as defined in Section 4 of this
- 8048 standard. ICOM consumers that support an extension module:
- 8049 a. SHALL conform to all mandatory statements and
- 8050 b. MAY conform to optional statements
- 8051 as defined in Section 4 for that extension module.

Appendix A. Acknowledgements

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:

Rafiul Ahad, Oracle Corporation
Eric S. Chan, Oracle Corporation
Martin Chapman, Oracle Corporation
Scott Conroy, Individual
Stefan Decker, Digital Enterprise Research Institute (DERI)
Laura Dragan, Digital Enterprise Research Institute (DERI)
Patrick Durusau, Individual
Siegfried Handschuh, Digital Enterprise Research Institute (DERI)
Deirdre Lee, Digital Enterprise Research Institute (DERI)
Marc Pallot, ESoCE-NET
Chancellor Pascale, Johns Hopkins University Applied Physics Laboratory
Vassilios Peristeras, Digital Enterprise Research Institute (DERI)
Peter Saint-Andre, Cisco Systems, Inc.
Ramesh Vasudevan, Oracle Corporation
Peter Yim, Individual

Appendix B. Revision History

Revision	Date	Editor	Changes Made
CSPRD 01	March 16, 2011	Eric S. Chan Patrick Durusau	Committee Specification Draft for Public Review
CSPRD 02	November 8, 2011	Eric S. Chan Patrick Durusau	Changes in response to public review comments.
CSPRD 03	March 20, 2012	Eric S. Chan Patrick Durusau Laura Dragan	Changes in response to TC members review comments.