



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

Committee Specification Draft ~~0203~~ /
Public Review Draft ~~0203~~

~~16 November 2011~~

28 March 2012

Specification URIs

This version:

<http://docs.oasis-open.org/icom/icom-ics/v1.0/csprd03/icom-ics-v1.0-csprd03.doc> ~~N/A~~
(Authoritative)
<http://docs.oasis-open.org/icom/icom-ics/v1.0/csprd03/icom-ics-v1.0-csprd03.html>
<http://docs.oasis-open.org/icom/icom-ics/v1.0/csprd03/icom-ics-v1.0-csprd03.pdf>

Previous version:

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

Latest version:

<http://docs.oasis-open.org/icom/icom-ics/v1.0/icom-ics-v1.0.doc> ~~N/A~~
(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 ~~that~~which also includes:

- XML schemas: <http://docs.oasis-open.org/icom/icom-ics/v1.0/csprd03/schemas/>
 - ~~icom-ac.xsd~~
 - ~~icom-cal.xsd~~
 - ~~icom-card.xsd~~
 - ~~icom-conf.xsd~~
 - ~~icom-content.xsd~~
 - ~~icom-core.xsd~~

- ~~icom-doc.xsd~~
- ~~icom-forum.xsd~~
- ~~icom-msg.xsd~~
- ~~icom-presence.xsd~~
- ~~icom-meta.xsd~~
- ~~icom-task.xsd~~

Related work:

N/A

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 ~~standard~~ 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 ~~aggregate~~ 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 the ~~the~~ "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. ~~11 November 2011-28 March 2012.~~ OASIS Committee Specification Draft ~~0203~~ / Public

Review Draft ~~02~~₀₃ <http://docs.oasis-open.org/icom/icom-ics/v1.0/csprd03/icom-ics-v1.0-csprd03.html>

Notices

Copyright © OASIS Open 201~~1~~². 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	12
1.1	Terminology	13
1.2	Normative References	13
1.3	Non-Normative References	13
2	Modeling Language	15
2.1	Introduction	15
2.2	Class Definition Grammar	15
2.3	Property Definition Grammar	17
2.4	Namespaces	20
3	Core Model	21
3.1	Main Branch	21
3.1.1	Entity and Top-Level Subclasses	21
3.1.2	Identifiable	22
3.1.3	Parental	23
3.1.4	Extent	24
3.1.5	Entity	25
3.1.6	EntityDefinition	29
3.1.7	Overview of Scope, Subject, and Artifact Branches	30
3.2	Scope Branch	32
3.2.1	Scope and Top-Level Subclasses	32
3.2.2	Scope	32
3.2.3	Community	37
3.2.4	Space	39
3.3	Subject Branch	41
3.3.1	Subject and Top-Level Subclasses	41
3.3.2	Subject	41
3.3.3	Group	43
3.3.4	Actor	47
3.3.5	Person	49
3.3.6	Resource	53
3.3.7	ResourceType	56
3.3.8	ResourceTypeEnum	57
3.3.9	ResourceBookingRule	58
3.3.10	ResourceBookingRuleEnum	59
3.4	Artifact Branch	61
3.4.1	Artifact and Top-Level Subclasses	61
3.4.2	Item	62
3.4.3	SpaceItem	63
3.4.4	Container	64
3.4.5	FolderContainer	65
3.4.6	Artifact	66
3.4.7	Folder	69
3.4.8	HeterogeneousFolder	70

3.5 Access Control Model	72
3.5.1 Accessor	72
3.5.2 Owner	73
3.5.3 RoleDefinition	74
3.5.4 Role	75
3.5.5 Privilege	78
3.5.6 PrivilegeEnum	78
3.5.7 AccessControlList	79
3.5.8 AccessControlEntry	80
3.5.9 AccessType	81
3.5.10 AccessTypeEnum	82
3.6 Metadata Model	85
3.6.1 PropertyDefinition	85
3.6.2 Property	88
3.6.3 PropertyChoiceType	91
3.6.4 PropertyType	92
3.6.5 PropertyTypeEnum	93
3.6.6 Cardinality	94
3.6.7 CardinalityEnum	95
3.6.8 Marker and Subclasses	96
3.6.9 Marker	96
3.6.10 Category	97
3.6.11 CategoryApplication	99
3.6.12 Tag	101
3.6.13 TagApplication	102
3.6.14 RelationshipBondable	104
3.6.15 RelationshipDefinition	105
3.6.16 Relationship	106
3.7 Common Concepts	109
3.7.1 Addressable	109
3.7.2 EntityAddress	110
3.7.3 Participant	111
3.7.4 Priority	112
3.7.5 PriorityEnum	113
3.7.6 DateTimeResolution	114
3.7.7 DateTimeResolutionEnum	115
3.7.8 TimeZone	115
3.7.9 Location	117
3.7.10 GeoCoordinates	118
4 Extension Modules	120
4.1 Overview of Extension Modules	120
4.2 Content Module	123
4.2.1 MimeConvertible	123
4.2.2 Content	124
4.2.3 MultiContent	128

4.2.4 SimpleContent	129
4.2.5 OnlineContent.....	131
4.2.6 ContentDispositionType	132
4.2.7 ContentDispositionTypeEnum	132
4.2.8 Attachment	133
4.3 Document Module.....	134
4.3.1 Versionable.....	134
4.3.2 VersionControlMetadata.....	137
4.3.3 VersionSeries	138
4.3.4 Version	141
4.3.5 VersionType	142
4.3.6 VersionTypeEnum	143
4.3.7 Document	144
4.3.8 WikiPage	147
4.4 Message Module	149
4.4.1 Message	149
4.4.2 UnifiedMessage	151
4.4.3 UnifiedMessageParticipant.....	156
4.4.4 UnifiedMessageFlag.....	157
4.4.5 UnifiedMessageFlagEnum	158
4.4.6 UnifiedMessageDeliveryStatusNotificationRequest	159
4.4.7 UnifiedMessageDeliveryStatusNotificationRequestEnum	160
4.4.8 UnifiedMessageChannel	161
4.4.9 UnifiedMessageChannelEnum.....	162
4.4.10 UnifiedMessageEditMode	163
4.4.11 UnifiedMessageEditModeEnum	163
4.4.12 InstantMessage	166
4.4.13 InstantMessageType	170
4.4.14 InstantMessageTypeEnum.....	171
4.4.15 InstantMessageChatStatus	172
4.4.16 InstantMessageChatStatusEnum.....	173
4.4.17 InstantMessageFeed.....	174
4.4.18 InstantMessageConnection.....	177
4.5 Presence Module	179
4.5.1 Presence	180
4.5.2 PresenceEditMode	184
4.5.3 PresenceEditModeEnum.....	185
4.5.4 ContactMethod	186
4.5.5 ContactReachabilityStatus	188
4.5.6 ContactReachabilityStatusEnum	189
4.5.7 Activity	192
4.5.8 ActivityType	193
4.5.9 ActivityTypeEnum.....	194
4.6 Address Book Module.....	195
4.6.1 AddressBook	195

4.6.2 PersonContact.....	197
4.7 Calendar Module	203
4.7.1 Calendar	203
4.7.2 OccurrenceSeries.....	205
4.7.3 Occurrence	213
4.7.4 OccurrenceStatus.....	221
4.7.5 OccurrenceStatusEnum	222
4.7.6 OccurrenceType	223
4.7.7 OccurrenceTypeEnum	224
4.7.8 OccurrenceParticipant.....	224
4.7.9 OccurrenceParticipantStatus.....	225
4.7.10 OccurrenceParticipantStatusEnum	226
4.7.11 OccurrenceParticipantTransparency.....	227
4.7.12 OccurrenceParticipantTransparencyEnum	228
4.7.13 OccurrenceEditMode.....	229
4.7.14 OccurrenceEditModeEnum	230
4.8 Free Busy Module.....	231
4.8.1 FreeBusy	231
4.8.2 FreeBusyInterval	232
4.8.3 FreeBusyType	235
4.8.4 FreeBusyTypeEnum.....	236
4.9 Task List Module.....	237
4.9.1 TaskList	237
4.9.2 Task.....	239
4.9.3 TaskStatus.....	246
4.9.4 TaskStatusEnum	247
4.9.5 TaskParticipantStatus	248
4.9.6 TaskParticipantStatusEnum	249
4.9.7 TaskEditMode.....	250
4.9.8 TaskEditModeEnum	250
4.10 Forum Module.....	251
4.10.1 Discussion	251
4.10.2 DiscussionContainer	252
4.10.3 DiscussionMessage	253
4.10.4 TopicContainer	254
4.10.5 Forum	257
4.10.6 Topic.....	259
4.10.7 Announcement	260
4.10.8 AnnouncementStatus	262
4.10.9 AnnouncementStatusEnum.....	262
4.11 Conference Module	263
4.11.1 Conference	263
4.11.2 ConferenceType	267
4.11.3 ConferenceTypeEnum	268
4.11.4 ConferenceStatus.....	269

4.11.5 ConferenceStatusEnum	270
4.11.6 ConferenceSession	271
4.11.7 ConferenceSessionEndingReason	273
4.11.8 ConferenceSessionEndingReasonEnum	273
4.11.9 ConferenceSetting	274
4.11.10 ConferenceParticipantRole	276
5 Conformance.....	279
Appendix A. Acknowledgements	280
Appendix B. Revision History	282

Table of Figures

Figure 1: Entity and Top-Level Abstract Classes.....	21
Figure 2: Entity Class Diagram.	28
Figure 3: Scope, Subject, and Artifact Branches.	31
Figure 4: Scope Branch.	32
Figure 5: Scope Class Diagram.	36
Figure 6: Community Class Diagram.	38
Figure 7: Space Class Diagram.	40
Figure 8: Subject Branch.....	41
Figure 9: Subject Class Diagram.	43
Figure 10: Group and Actor Class Diagram.....	47
Figure 11: Person Class Diagram.	53
Figure 12: Resource Class Diagram.....	56
Figure 13: Artifact Branch.	62
Figure 14: Artifact Class Diagram.	69
Figure 15: Heterogeneous Folder Class Diagram.	72
Figure 16: Role Definition and Role Class Diagram.	77
Figure 17: Access Control List Class Diagram.	85
Figure 18: Property Definition and Property Class Diagram.....	90
Figure 19: Marker Branch.	96
Figure 20: Marker Class Diagram.	97
Figure 21: Category and Category Application Class Diagram.	99
Figure 22: Tag and Tag Application Class Diagram.	102
Figure 23: Relationship Class Diagram.	108
Figure 24: Containers of Collaboration Activities.	121
Figure 25: Composite Content Class Diagram.	128
Figure 26: Document, Version Series, and Version Class Diagram.	147
Figure 27: Wiki Page Class Diagram.	149
Figure 28: Unified Message Class Diagram.	166
Figure 29: Instant Message Class Diagram.....	170
Figure 30: Instant Message Feed and Connection Class Diagram.	177
Figure 31: Presence Class Diagram.	184
Figure 32: Presence Contact Method and Instant Message Connection Class Diagram.	191
Figure 33: Address Book Class Diagram.....	197
Figure 34: Person Contact Class Diagram.	203
Figure 35: Calendar Class Diagram.....	205
Figure 36: Occurrence Series Class Diagram.	213
Figure 37: Occurrence Class Diagram.....	221
Figure 38: Free Busy Class Diagram.....	235
Figure 39: Task List Class Diagram.....	239
Figure 40: Task Class Diagram.	246

Figure 41: Forum Class Diagram.....	257
Figure 42: Conference Class Diagram.....	267

1 Introduction

The Integrated Collaboration Object Model (ICOM) for Interoperable Collaboration Services ~~standards~~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 ~~super~~-classes (Section 3.1 Main Branch) that brings 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 ~~separate the~~support separation of concerns ~~effor~~ user administration and content management. Subject branch includes the model of actors, groups of actors, and role assignment of actors. Actors, groups, and roles ~~are~~ typically appear as the ~~subjects of subject in the~~ (subject, privilege, object) triples ~~in~~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 ~~range of~~ 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 **schemaset** 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 ~~more~~ 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)
- ~~`uri`~~ `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.

311

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 UML Diagram of Entity and Top-Level Subclasses

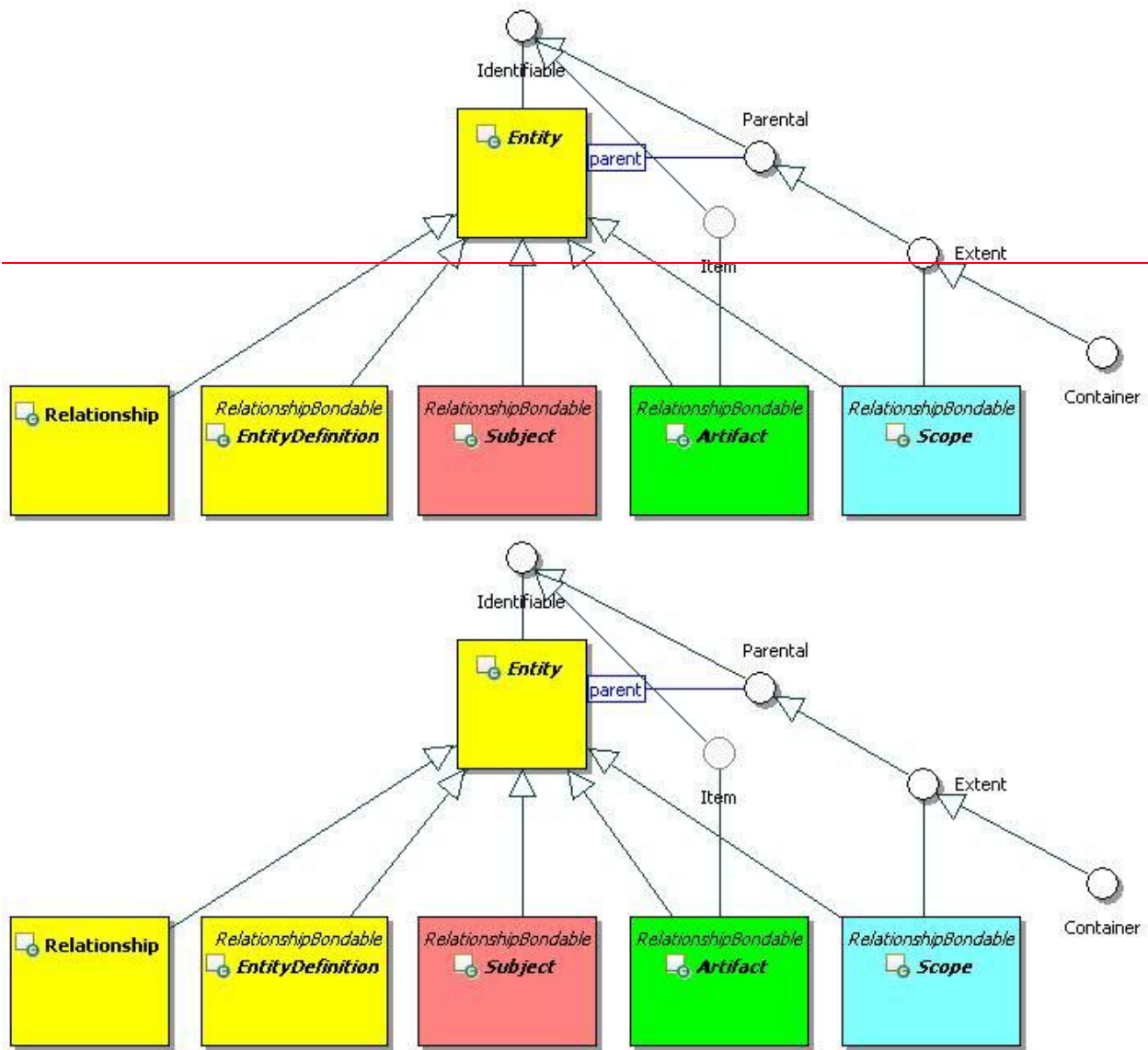


Figure 1: Entity and Top-Level Abstract Classes.

The UML diagram in 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 ~~can be uniquely identified~~enables unique identification.

The `Identifiable` class ~~is defined by the~~has attribute values:

localNamespace

Value: `icom_core`

localName

Value: `Identifiable`

extendsFrom

Value:

stereotype

Value: `mixin`

description

Value: `Identifiable` is a mixin class which defines the characteristics of all entities and some non-entities that ~~can be uniquely identified~~enables unique identification.

propertyDefinitions

The values for this attribute are defined in Section 3.1.2.3.

3.1.2.3 Property Definitions

The `Identifiable` class MUST have the property definitions:

icom_core:objectId

Description: A persistent identifier of an object.

Required: False

Inherited: False

Property Type: ~~ID~~String

Cardinality: Single

Updatability: Read Only

icom_core:changeToken

Description: An opaque token used for optimistic locking & concurrency checking.

401	Required:	False
402	Inherited:	False
403	Property Type:	String
404	Cardinality:	Single
405	Updatability:	Read Only

406

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

408

409 3.1.3 Parental

410 3.1.3.1 Description

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

412 3.1.3.2 Class Definition

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

415 | The Parental class ~~is defined by the~~has attribute values:

416

417 **localNamespace**

418 Value: icom_core

419

420 **localName**

421 Value: Parental

422

423 **extendsFrom**

424 Value: icom_core:Identifiable

425

426 **stereotype**

427 Value: mixin

428

429 **description**

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

431 parents of other entities or identifiable objects.

432

433 **propertyDefinitions**

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

435 3.1.3.3 Property Definitions

436 The Parental class inherits property definitions from super classes.

437 The Parental class MUST have the property definition:

438

439 **icom_core:parent**

440 Description: Parent of an object.

441	Required:	False
442	Inherited:	False
443	Property Type:	icom_core:Parental
444	Cardinality:	Single
445	Updatability:	Read Only

446

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

448

449 3.1.4 Extent

450 3.1.4.1 Description

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

452 3.1.4.2 Class Definition

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

454 The Extent class ~~is defined by the~~has attribute values:

455

456 **localNamespace**

457 Value: icom_core

458

459 **localName**

460 Value: Extent

461

462 **extendsFrom**

463 Value: icom_core:Parental

464

465 **stereotype**

466 Value: mixin

467

468 **description**

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

470 entities.

471

472 **propertyDefinitions**

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

474 3.1.4.3 Property Definitions

475 The Extent class inherits property definitions from super classes.

476 The Extent class MUST have the property definition:

477

478	icom_core:parent	
479	Description:	Parent of an extent.
480	Required:	False

481 Inherited: True
482 Property Type: icom_core:Extent
483 Cardinality: Single
484 Updatability: Read Only

485

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

487

488 3.1.5 Entity

489 3.1.5.1 Description

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

491 Each entity is assigned an internationalized resource identifier (IRI) composed from its *objectId*. The form
492 of the IRI is implementation-dependent.

493 3.1.5.2 Class Definition

494 The Entity class ~~is defined by the~~has attribute values:

495

496 **localNamespace**

497 Value: icom_core

498

499 **localName**

500 Value: Entity

501

502 **extendsFrom**

503 Value: icom_core:Identifiable

504

505 **stereotype**

506 Value: primary

507

508 **isAbstract**

509 Value: TRUE

510

511 **description**

512 Value: An entity is an object ~~that has~~with an immutable id and ~~can be individually~~individual
513 access control~~led~~.

514

515 **propertyDefinitions**

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

517 3.1.5.3 Property Definitions

518 The Entity class inherits property definitions from super classes.

519 The Entity class MUST have the property definitions:

520

521	icom_core:name	
522	Description:	Name of an entity.
523	Required:	False
524	Inherited:	False
525	Property Type:	String
526	Cardinality:	Single
527	Updatability:	Read Write
528		
529	icom_core:createdBy	
530	Description:	An actor who creates sd an entity.
531	Required:	False
532	Inherited:	False
533	Property Type:	icom_core:Actor
534	Cardinality:	Single
535	Updatability:	Read Only
536		
537	icom_core:creationDate	
538	Description:	Date and time when an entity is created. It is immutable.
539	Required:	False
540	Inherited:	False
541	Property Type:	DateTime
542	Cardinality:	Single
543	Updatability:	Read Only
544		
545	icom_core:lastModifiedBy	
546	Description:	An actor who last modified an entity.
547	Required:	False
548	Inherited:	False
549	Property Type:	icom_core:Actor
550	Cardinality:	Single
551	Updatability:	Read Only
552		
553	icom_core:lastModificationDate	
554	Description:	Date and time when an entity is of last modified ed <u>ation</u> .
555	Required:	False
556	Inherited:	False
557	Property Type:	DateTime
558	Cardinality:	Single
559	Updatability:	Read Only
560		
561	icom_ac:owner	
562	Description:	A subject who owns an entity.

563 ~~Required: True~~
564 ~~Inherited: False~~
565 ~~Property Type: icom_ac:Owner~~
566 ~~Cardinality: Single~~
567 ~~Updatability: Read Write~~
568

569 **icom_core:parent**

570 Description: A parental entity which contains an entity.
571 Required: False
572 Inherited: False
573 Property Type: icom_core:Parental
574 Cardinality: Single
575 Updatability: Read Only
576

577 **icom_ac:owner**

578 Description: A subject who owns an entity.
579 Required: True
580 Inherited: False
581 Property Type: icom_ac:Owner
582 Cardinality: Single
583 Updatability: Read Write
584

585 **icom_ac:accessControlList**

586 Description: Access control list on an entity.
587 Required: False
588 Inherited: False
589 Property Type: icom_ac:AccessControlList
590 Cardinality: Single
591 Updatability: Read Write
592

593 **icom_meta:attachedMarker**

594 Description: Zero or more markers applied on an entity.
595 Required: False
596 Inherited: False
597 Property Type: icom_meta:Marker
598 Cardinality: Multi
599 Updatability: Read Only
600

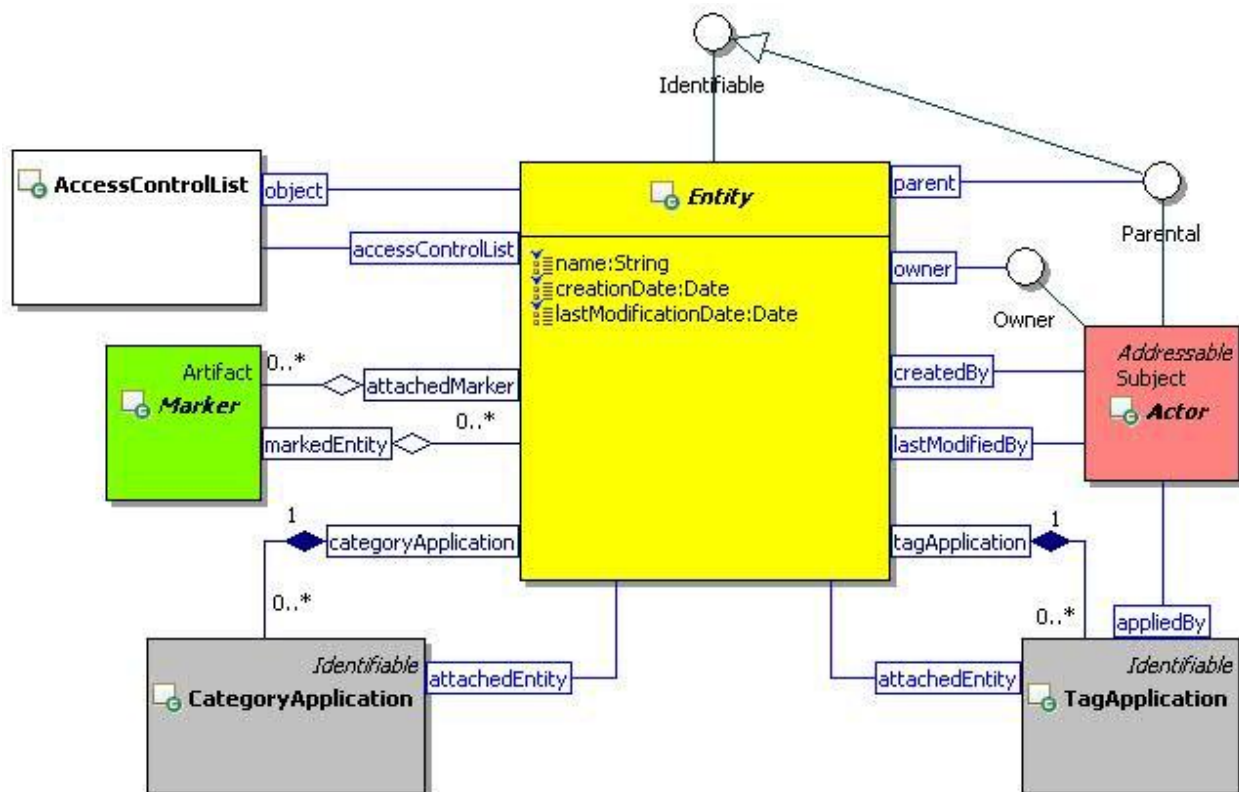
601 **icom_meta:categoryApplication**

602 Description: Zero or more category applications on an entity.
603 Required: False
604 Inherited: False

605 Property Type: icom_meta:CategoryApplication
606 Cardinality: Multi
607 Updatability: Read Only
608
609 **icom_meta:tagApplication**
610 Description: Zero or more tag applications on an entity.
611 Required: False
612 Inherited: False
613 Property Type: icom_meta:TagApplication
614 Cardinality: Multi
615 Updatability: Read Only

617 ~~icom_ac:accessControllist~~
618 ~~Description: Access control list on an entity.~~
619 ~~Required: False~~
620 ~~Inherited: False~~
621 ~~Property Type: icom_ac:AccessControllist~~
622 ~~Cardinality: Single~~
623 ~~Updatability: Read Write~~

625 The Entity class MAY include additional property definitions which are implementation-defined.
626



627
628 Figure 2: Entity Class Diagram.

629

630 3.1.6 EntityDefinition

631 3.1.6.1 Description

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

633 3.1.6.2 Class Definition

634 The EntityDefinition class ~~is defined by the~~has attribute values:

635

636 **localNamespace**

637 Value: icom_core

638

639 **localName**

640 Value: EntityDefinition

641

642 **extendsFrom**

643 Value: icom_core:Entity, icom_meta:RelationshipBondable

644

645 **stereotype**

646 Value: primary

647

648 **isAbstract**

649 Value: TRUE

650

651 **description**

652 Value: An entity definition defines a type of entities.

653

654 **propertyDefinitions**

655 The values for this attribute are defined in Section 3.1.6.3.

656 3.1.6.3 Property Definitions

657 The EntityDefinition class inherits property definitions from super classes.

658 The EntityDefinition class MUST have the property definition:

659

660 **icom_core:description**

661 Description: A description of an entity definition.

662 Required: False

663 Inherited: False

664 Property Type: String

665 Cardinality: Single

666 Updatability: Read Write

667

668 The EntityDefinition class MAY include additional property definitions which are implementation-defined.
669

670 3.1.7 Overview of Scope, Subject, and Artifact Branches

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

675 Note: The Subject and Artifact branches support the separation of concerns of user administration and
676 content management. Typically subjects and artifacts are joined in the (subject, privilege, artifact) triples
677 of access control model. Some of the (subject, privilege, artifact) triples are derived from the scopes of the
678 role assignments and the artifacts contained by the scopes. ~~For example, OASIS consortium can be~~
679 ~~represented in ICOM by a community which contains a set of spaces to represent the TC workspaces. In~~
680 ~~the OASIS community, an organizational member can designate one person to approve the participation~~
681 ~~of its representatives in the OASIS TC's. Once an organization's representative is added as a member of~~
682 ~~a TC space, he or she gains the access privileges for artifacts in the space.~~ The communities and spaces
683 contain subjects and artifacts; however, membership of subjects in a space is administered separately
684 from management of artifacts in the space.

685 ~~The classes in~~ Scope, Subject, and Artifact ~~branches~~ are defined, ~~respectively,~~ in Section 3.2, ~~Section~~
686 3.3, and 3.4 ~~Section-, respectively.~~

687

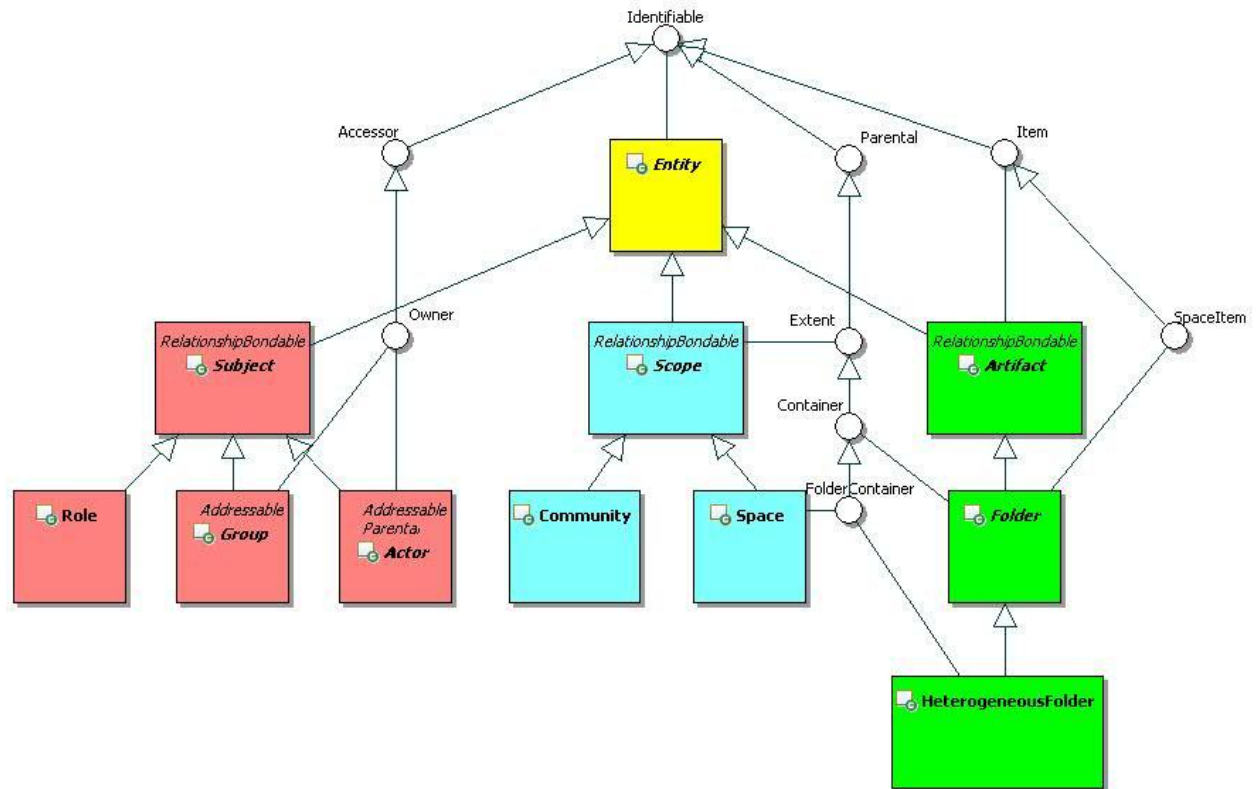
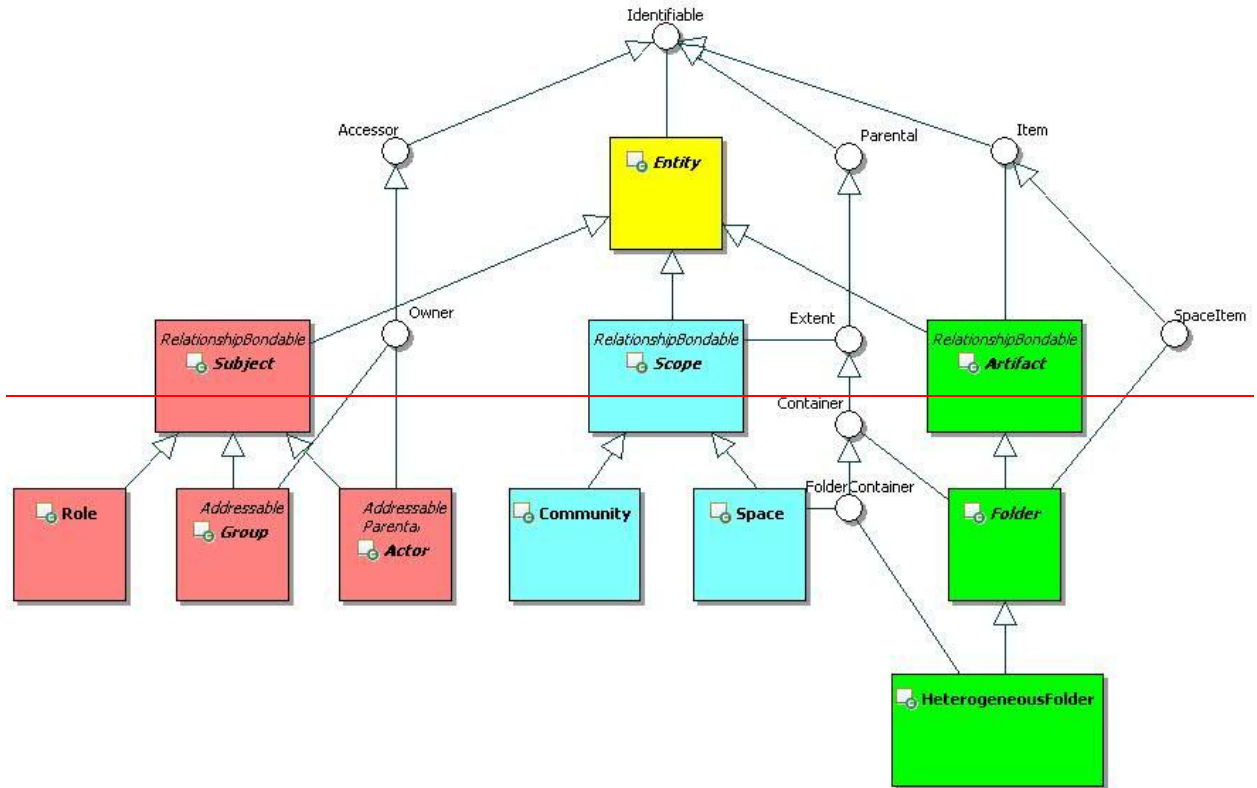


Figure 3—UML Diagram of: Scope, Subject, and Artifact Branches.

3.2 Scope Branch

3.2.1 ~~UML Diagram of~~ Scope and Top-Level Subclasses

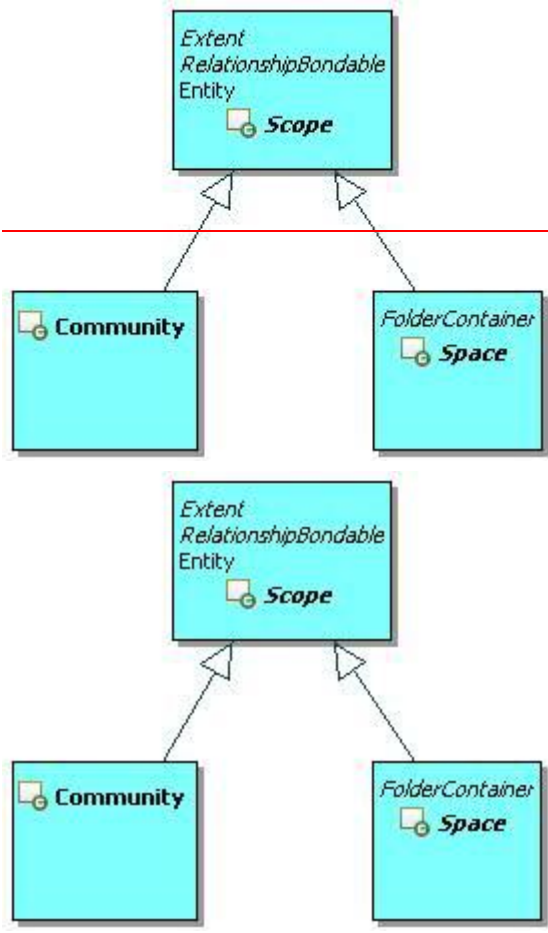


Figure 4: Scope Branch.

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 ~~realm~~ policy.

3.2.2.2 Class Definition

The Scope class ~~is defined by the~~ has attribute values:

localNamespace

Value: icom_core

localName

Value: Scope

710 **extendsFrom**
 711 Value: icom_core:Entity, icom_core:Extent, icom_meta:RelationshipBondable
 712
 713 **stereotype**
 714 Value: primary
 715
 716 **isAbstract**
 717 Value: TRUE
 718
 719 **description**
 720 Value: A scope is an extent of an administrative realm.
 721
 722 **propertyDefinitions**
 723 The values for this attribute are defined in Section 3.2.2.3.

724 3.2.2.3 Property Definitions

725 The Scope class inherits property definitions from super classes.
 726 The Scope class MUST have the property definitions:

727
 728 **icom_core:description**
 729 Description: A description of a scope.
 730 Required: False
 731 Inherited: False
 732 Property Type: String
 733 Cardinality: Single
 734 Updatability: Read Write
 735
 736 **icom_core:parent**
 737 Description: A community which contains a scope.
 738 Required: False
 739 Inherited: True
 740 Property Type: icom_core:Community
 741 Cardinality: Single
 742 Updatability: Read Only
 743
 744 ~~**icom_ac:roleDefinition**~~
 745 ~~Description: Zero or more role definitions defined in a scope.~~
 746 ~~Required: False~~
 747 ~~Inherited: False~~
 748 ~~Property Type: icom_ac:RoleDefinition~~
 749 ~~Cardinality: Multi~~
 750 ~~Updatability: Read Only~~
 751

icom_ac:role

Description: Zero or more roles defined in a scope.
Required: False
Inherited: False
Property Type: icom_ac:Role
Cardinality: Multi
Updatability: Read Only

icom_core:group

Description: Zero or more groups defined in a scope.
Required: False
Inherited: False
Property Type: icom_core:Group
Cardinality: Multi
Updatability: Read Only

icom_core:memberGroup

Description: Member groups of a scope, i.e. groups whose assigned scopes include this scope.
Required: False
Inherited: False
Property Type: icom_core:Group
Cardinality: Multi
Updatability: Read Only

icom_ac:roleDefinition

Description: Zero or more role definitions defined in a scope.
Required: False
Inherited: False
Property Type: icom_ac:RoleDefinition
Cardinality: Multi
Updatability: Read Only

icom_ac:role

Description: Zero or more roles defined in a scope.
Required: False
Inherited: False
Property Type: icom_ac:Role
Cardinality: Multi
Updatability: Read Only

icom_meta:relationship

Description: Zero or more relationships associated with a scope.

795	Required:	False
796	Inherited:	False
797	Property Type:	icom_meta:Relationship
798	Cardinality:	Multi
799	Updatability:	Read Only

800

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

802

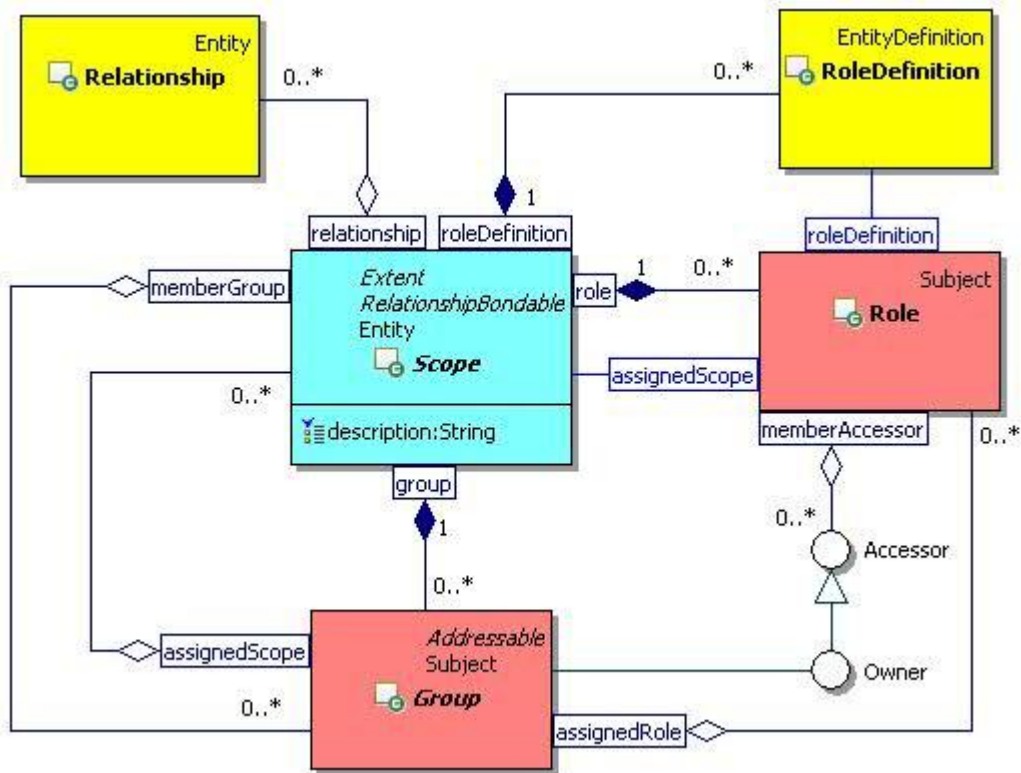
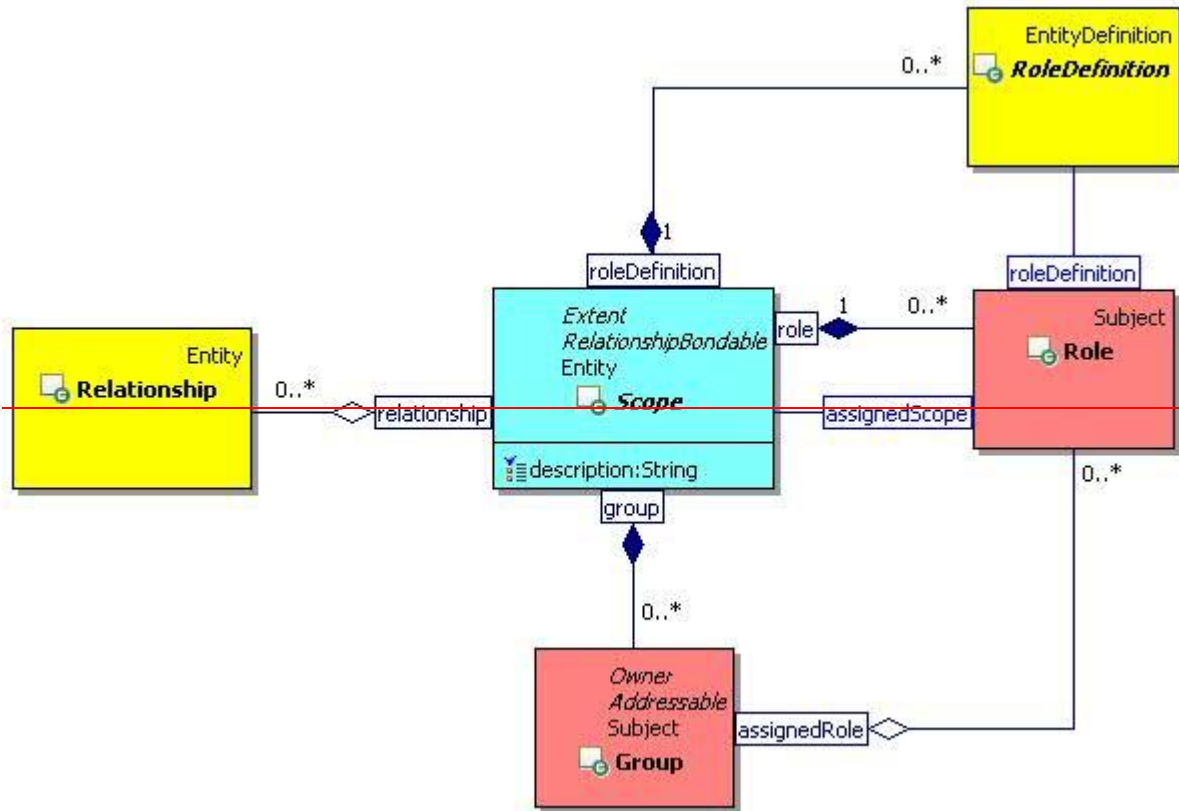


Figure 5: Scope Class Diagram.

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 ~~is defined by the~~has attribute values:

localNamespace

Value: icom_core

localName

Value: Community

extendsFrom

Value: icom_core:Scope

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.2.3.3.

3.2.3.3 Property Definitions

The Community class inherits property definitions from super classes.

The Community class MUST have the property definitions:

icom_core:community

Description: Sub-communities of a community.

Required: False

Inherited: False

Property Type: icom_core:Community

Cardinality: Multi

Updatability: Read Only

icom_core:space

Description: Spaces of a community.

847	Required:	False
848	Inherited:	False
849	Property Type:	icom_core:Space
850	Cardinality:	Multi
851	Updatability:	Read Only
852		
853	icom_core:actor	
854	Description:	Managed actors of a community, i.e. actors whose parent community is this community.
855		
856	Required:	False
857	Inherited:	False
858	Property Type:	icom_core:Actor
859	Cardinality:	Multi
860	Updatability:	Read Only
861		
862	icom_core:memberActor	
863	Description:	Member actors of a community, i.e. actors whose assigned communities include this community.
864		
865	Required:	False
866	Inherited:	False
867	Property Type:	icom_core:Actor
868	Cardinality:	Multi
869	Updatability:	Read Only

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

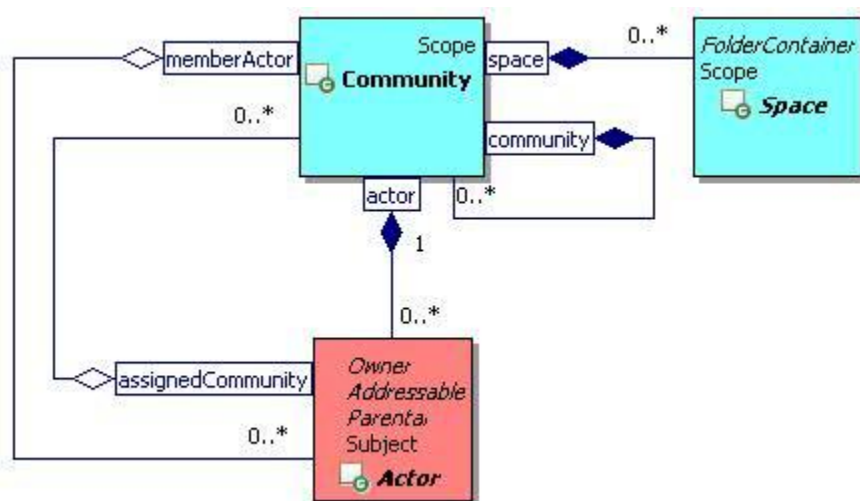


Figure 6: Community Class Diagram.

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 ~~is defined by the~~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.



element

0..*



SpaceItem



element

0..*



SpaceItem

Figure 7: Space Class Diagram.

3.3 Subject Branch

3.3.1 UML Diagram of Subject and Top-Level Subclasses

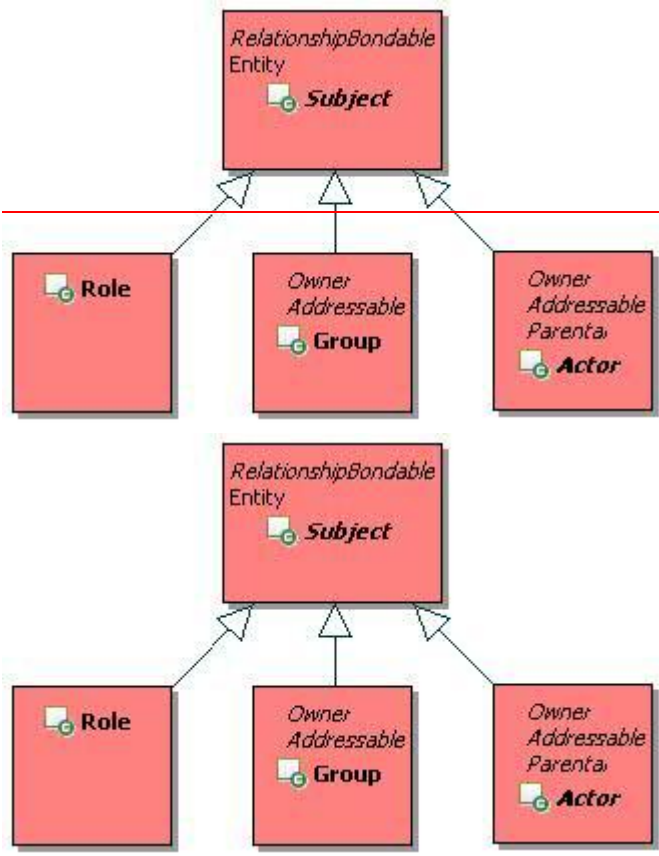


Figure 8: Subject Branch.

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 is defined by the `has` attribute values:

localNamespace

Value: `icom_core`

localName

Value: `Subject`

extendsFrom

Value: `icom_core:Entity, icom_meta:RelationshipBondable`

938

939 **stereotype**

940 Value: primary

941

942 **isAbstract**

943 Value: TRUE

944

945 **description**

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

947

948 **propertyDefinitions**

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

950 3.3.2.3 Property Definitions

951 The Subject class inherits property definitions from super classes.

952 The Subject class MUST have the property definitions:

953

954 **icom_core:description**

955 Description: A description of a subject.

956 Required: False

957 Inherited: False

958 Property Type: String

959 Cardinality: Single

960 Updatability: Read Write

961

962 **icom_core:parent**

963 Description: A scope which contains a subject.

964 Required: False

965 Inherited: True

966 Property Type: icom_core:Scope

967 Cardinality: Single

968 Updatability: Read Only

969

970 **icom_meta:relationship**

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

972 Required: False

973 Inherited: False

974 Property Type: icom_meta:Relationship

975 Cardinality: Multi

976 Updatability: Read Only

977

978 **icom_meta:property**

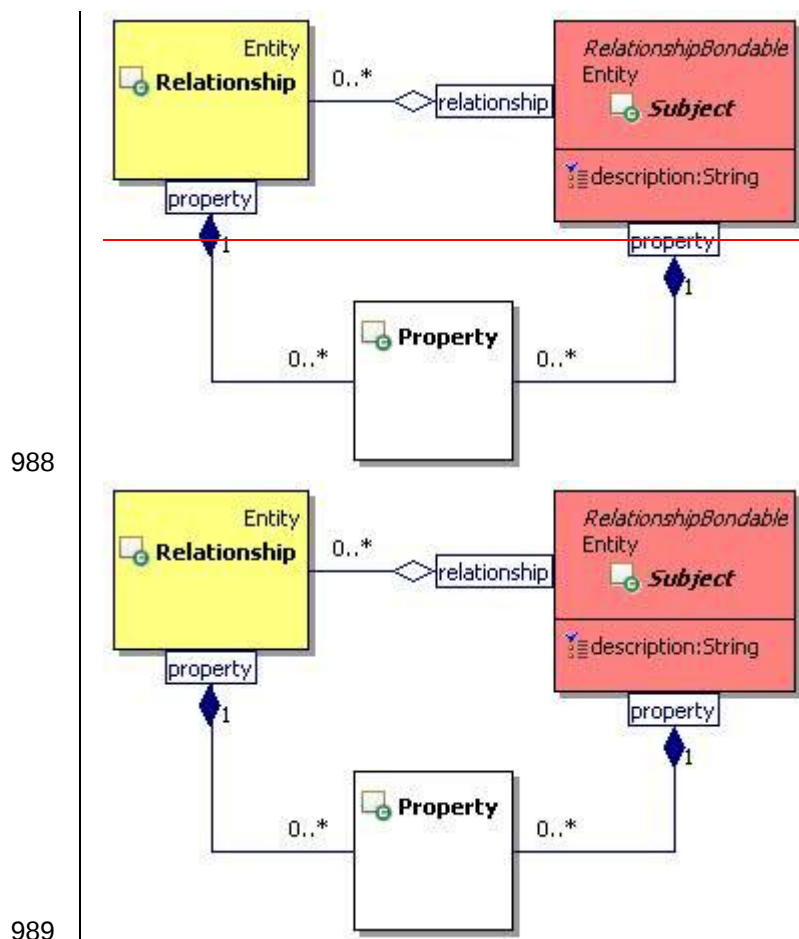
979 Description: Zero or more extended properties of a subject.

980	Required:	False
981	Inherited:	False
982	Property Type:	icom_meta:Property
983	Cardinality:	Multi
984	Updatability:	Read Write

985

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

987



990 *Figure 9: Subject Class Diagram.*

991

992 3.3.3 Group

993 3.3.3.1 Description

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

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

996 It can be an owner of one or more entities.

997 3.3.3.2 Class Definition

998 The Group class ~~is defined by the~~has attribute values:

999

1000 **localNamespace**

1001 Value: icom_core

1002

1003 **localName**

1004 Value: Group

1005

1006 **extendsFrom**

1007 Value: icom_core:Subject, icom_core:Addressable, icom_ac:Accessor

1008 Optional Value: icom_ac:Owner

1009

1010 **stereotype**

1011 Value: primary

1012

1013 **description**

1014 Value: A group is a subject representing a set of actors and sub-groups. A group can be part of

1015 one or more super-groups. It can be an owner of one or more entities.

1016

1017 **propertyDefinitions**

1018 The values for this attribute are defined in Section 3.3.3.3.

1019 **3.3.3.3 Property Definitions**

1020 The Group class inherits property definitions from super classes.

1021 The Group class MUST have the property definitions:

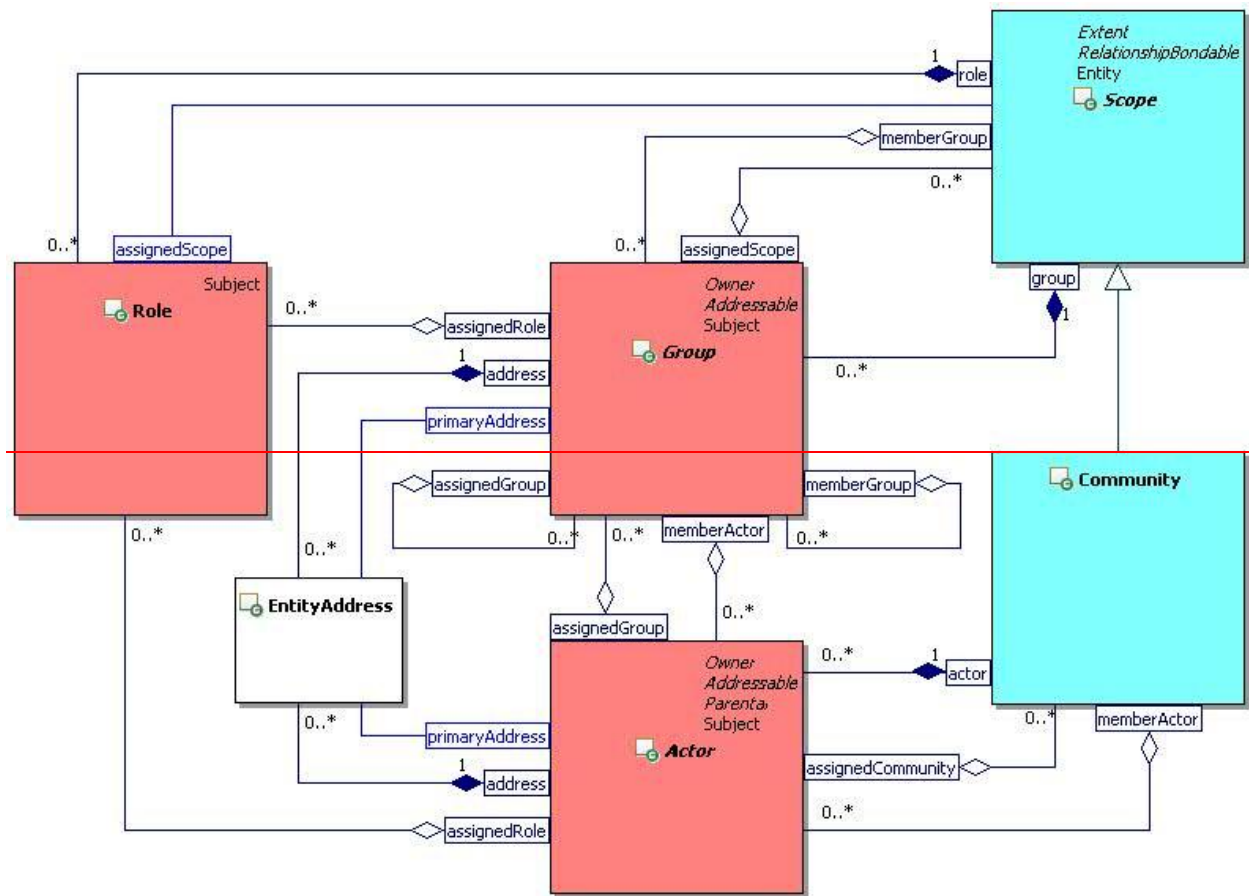
1022		
1023	icom_ac:assignedRole	
1024	Description:	Roles to which a group is assigned.
1025	Required:	False
1026	Inherited:	False
1027	Property Type:	icom_ac:Role
1028	Cardinality:	Multi
1029	Updatability:	Read Write
1030		
1031	icom_core:assignedGroup	
1032	Description:	SuperA group's super-groups to which a group is assigned.
1033	Required:	False
1034	Inherited:	False
1035	Property Type:	icom_core:Group
1036	Cardinality:	Multi
1037	Updatability:	Read Write
1038		
1039	icom_core:assignedScope	
1040	Description:	Scopes to which a group is assignedA group's scopes.

1041	Required:	False
1042	Inherited:	False
1043	Property Type:	icom_core:Scope
1044	Cardinality:	Multi
1045	Updatability:	Read Write
1046		
1047	icom_core:memberGroup	
1048	Description:	Sub-groups assigned to <u>of</u> a group.
1049	Required:	False
1050	Inherited:	False
1051	Property Type:	icom_core:Group
1052	Cardinality:	Multi
1053	Updatability:	Read Only

1054		
1055	icom_core:memberActor	
1056	Description:	Actors assigned to <u>in</u> a group.
1057	Required:	False
1058	Inherited:	False
1059	Property Type:	icom_core:Actor
1060	Cardinality:	Multi
1061	Updatability:	Read Only

1062		
1063	<u>icom_ac:assignedRole</u>	
1064	<u>Description:</u>	<u>A group's roles.</u>
1065	<u>Required:</u>	<u>False</u>
1066	<u>Inherited:</u>	<u>False</u>
1067	<u>Property Type:</u>	<u>icom_ac:Role</u>
1068	<u>Cardinality:</u>	<u>Multi</u>
1069	<u>Updatability:</u>	<u>Read Write</u>

1070		
1071	The Group class MAY include additional property definitions which are implementation-defined.	
1072		



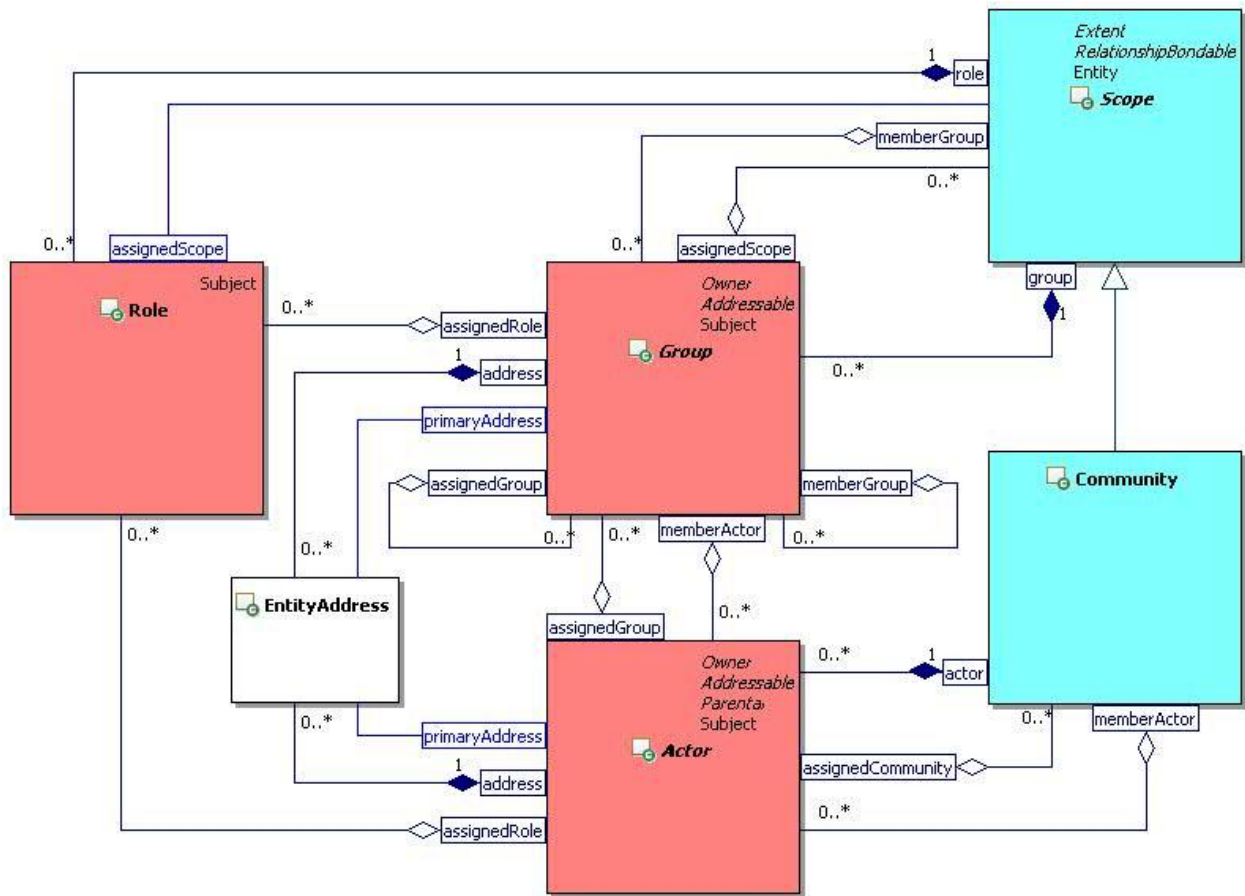


Figure 10: Group and Actor Class Diagram.

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 ~~is defined by the~~ has attribute values:

localNamespace

Value: icom_core

localName

Value: Actor

extendsFrom

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

1093 **stereotype**
1094 Value: primary
1095
1096 **isAbstract**
1097 Value: TRUE
1098
1099 **description**
1100 Value: An actor is a subject that can perform actions on objects.
1101
1102 **propertyDefinitions**
1103 The values for this attribute are defined in Section 3.3.4.3.

1104 3.3.4.3 Property Definitions

1105 The Actor class inherits property definitions from super classes.
1106 The Actor class MUST have the property definitions:

1107
1108 **icom_core:parent**
1109 Description: A community which contains an actor.
1110 Required: False
1111 Inherited: True
1112 Property Type: icom_core:Community
1113 Cardinality: Single
1114 Updatability: Read Only

1115
1116 ~~**icom_ac:assignedRole**~~
1117 ~~Description: Roles to which an actor is assigned.~~
1118 ~~Required: False~~
1119 ~~Inherited: False~~
1120 ~~Property Type: icom_ac:Role~~
1121 ~~Cardinality: Multi~~
1122 ~~Updatability: Read Write~~

1123
1124 **icom_core:assignedGroup**
1125 Description: ~~Groups to which an actor is assigned~~An actor's groups.
1126 Required: False
1127 Inherited: False
1128 Property Type: icom_core:Group
1129 Cardinality: Multi
1130 Updatability: Read Write

1131
1132 **icom_core:assignedCommunity**
1133 Description: ~~Communities to which an actor is assigned~~An actor's
1134 communities.

1135	Required:	False
1136	Inherited:	False
1137	Property Type:	icom_core:Community
1138	Cardinality:	Multi
1139	Updatability:	Read Write

1140

1141 icom_ac:assignedRole

1142 Description: An actor's roles.

1143 Required: False

1144 Inherited: False

1145 Property Type: icom_ac:Role

1146 Cardinality: Multi

1147 Updatability: Read Write

1148

1149 The Actor class MAY include additional property definitions which are implementation-defined.

1150

1151 3.3.5 Person

1152 3.3.5.1 Description

1153 A person is an individual human who may be an actor.

1154 A person has a personal space.

1155 3.3.5.2 Class Definition

1156 The Person class ~~is defined by the~~has attribute values:

1157

1158 **localNamespace**

1159 Value: icom_core

1160

1161 **localName**

1162 Value: Person

1163

1164 **extendsFrom**

1165 Value: icom_core:Actor

1166

1167 **stereotype**

1168 Value: primary

1169

1170 **description**

1171 Value: A person is an individual human who may be an actor.

1172

1173 **propertyDefinitions**

1174 The values for this attribute are defined in Section 3.3.5.3.

3.3.5.3 Property Definitions

The Person class inherits property definitions from super classes.

The Person class MUST have the property definitions:

icom_core:givenName

Description:	Given name of a person.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

icom_core:middleName

Description:	Middle name of a person. <u>Can include multiple names concatenated.</u>
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

icom_core:familyName

Description:	Family name of a person.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

icom_core:prefix

Description:	Prefix of a person's name.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

icom_core:suffix

Description:	Suffix of a person's name.
Required:	False
Inherited:	False
Property Type:	String

1217	Cardinality:	Single
1218	Updatability:	Read Write
1219		
1220	icom_core:nickname	
1221	Description:	Nickname of a person.
1222	Required:	False
1223	Inherited:	False
1224	Property Type:	String
1225	Cardinality:	Single <u>Multi</u>
1226	Updatability:	Read Write
1227		
1228	icom_core:jobTitle	
1229	Description:	Job title of a person.
1230	Required:	False
1231	Inherited:	False
1232	Property Type:	String
1233	Cardinality:	Single
1234	Updatability:	Read Write
1235		
1236	icom_core:department	
1237	Description:	A person's affiliated department.
1238	Required:	False
1239	Inherited:	False
1240	Property Type:	String
1241	Cardinality:	Single
1242	Updatability:	Read Write
1243		
1244	icom_core:officeLocation	
1245	Description:	Location of a person's department.
1246	Required:	False
1247	Inherited:	False
1248	Property Type:	String
1249	Cardinality:	Single
1250	Updatability:	Read Write
1251		
1252	icom_core:company	
1253	Description:	A person's affiliated company.
1254	Required:	False
1255	Inherited:	False
1256	Property Type:	String
1257	Cardinality:	Single
1258	Updatability:	Read Write

1259		
1260	icom_core:profession	
1261	Description:	A person's profession.
1262	Required:	False
1263	Inherited:	False
1264	Property Type:	String
1265	Cardinality:	Single
1266	Updatability:	Read Write
1267		
1268	icom_core:personalSpace	
1269	Description:	Personal space of a person.
1270	Required:	False
1271	Inherited:	False
1272	Property Type:	icom_core:Space
1273	Cardinality:	Single
1274	Updatability:	Read Only
1275		
1276	icom_presence:presence	
1277	Description:	Presence of a person.
1278	Required:	False
1279	Inherited:	False
1280	Property Type:	icom_presence:Presence
1281	Cardinality:	Single
1282	Updatability:	Read Only
1283		
1284	icom_msg:instantMessageFeed	
1285	Description:	Instant message feed for a person.
1286	Required:	False
1287	Inherited:	False
1288	Property Type:	icom_msg:InstantMessageFeed
1289	Cardinality:	Single
1290	Updatability:	Read Only
1291		
1292	The Person class MAY include additional property definitions which are implementation-defined.	
1293		

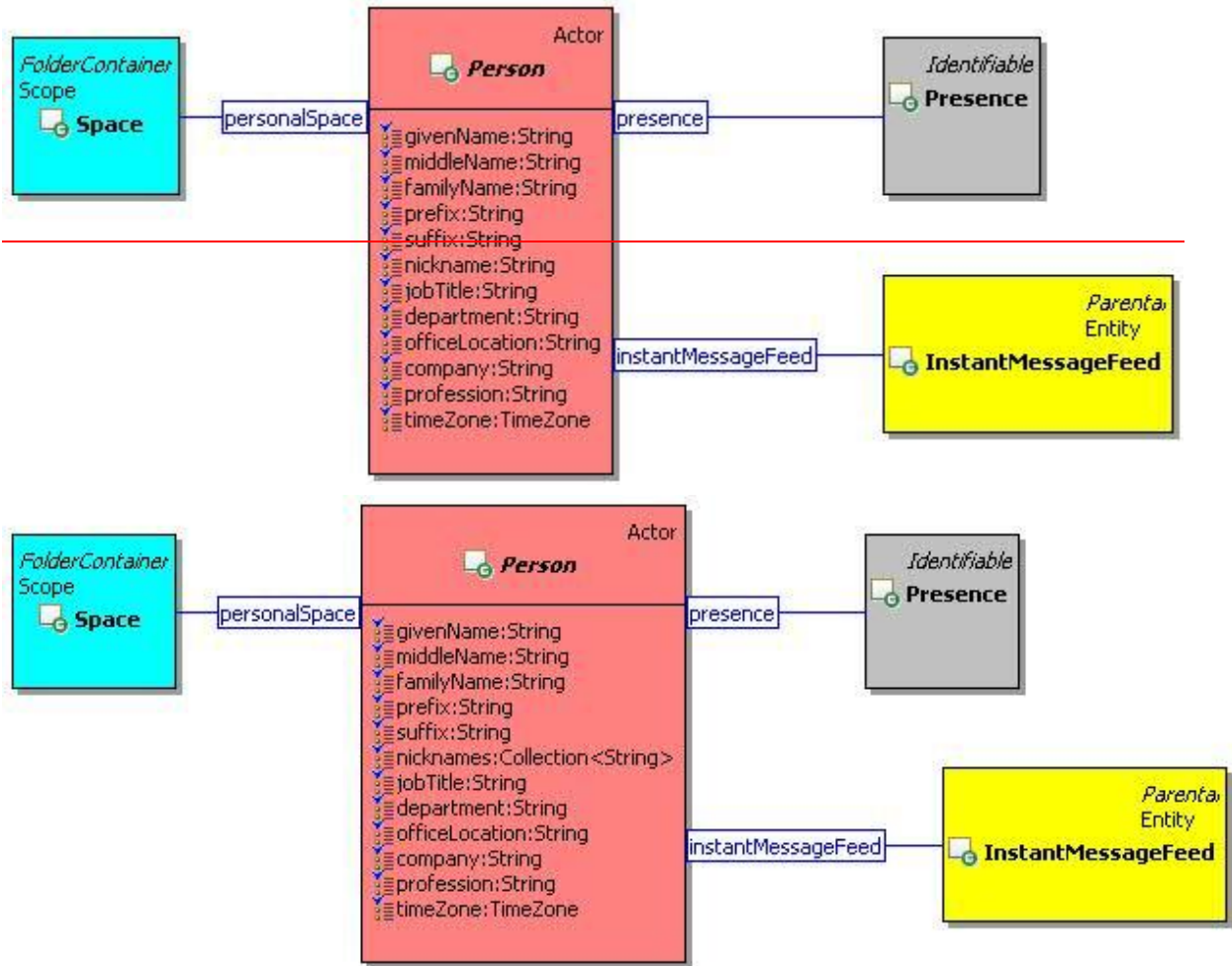


Figure 11: Person Class Diagram.

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 ~~is defined by the~~ has attribute values:

localNamespace

Value: icom_core

localName

Value: Resource

1312

1313 **extendsFrom**

1314 Value: icom_core:Actor

1315

1316 **stereotype**

1317 Value: primary

1318

1319 **description**

1320 Value: A resource actor is an actor representing a bookable resource, such as a conference

1321 room, equipment, or on-line conference.

1322

1323 **propertyDefinitions**

1324 The values for this attribute are defined in Section 3.3.6.3.

1325 3.3.6.3 Property Definitions

1326 The Resource class inherits property definitions from super classes.

1327 The Resource class MUST have the property definitions:

1328

1329 **icom_core:resourceSpace**

1330 Description: Administrative space of a resource actor.

1331 Required: False

1332 Inherited: False

1333 Property Type: icom_core:Space

1334 Cardinality: Single

1335 Updatability: Read Only

1336

1337 **icom_core:location**

1338 Description: Location of a resource.

1339 Required: False

1340 Inherited: False

1341 Property Type: icom_core:Location

1342 Cardinality: Single

1343 Updatability: Read Write

1344

1345 **icom_core:capacity**

1346 Description: Capacity of a resource.

1347 Required: False

1348 Inherited: False

1349 Property Type: Integer

1350 Cardinality: Single

1351 Updatability: Read Write

1352

1353 **icom_core:resourceType**

1354	Description:	Type of a resource.
1355	Required:	False
1356	Inherited:	False
1357	Property Type:	icom_core:ResourceType
1358	Cardinality:	Single
1359	Updatability:	Read Write
1360		
1361	icom_core:bookingRule	
1362	Description:	Resource booking rule.
1363	Required:	False
1364	Inherited:	False
1365	Property Type:	icom_core:ResourceBookingRule
1366	Cardinality:	Single
1367	Updatability:	Read Write
1368		
1369	icom_core:bookingApprover	
1370	Description:	One or more users <u>persons</u> who approve the booking of a
1371	resource.	
1372	Required:	False
1373	Inherited:	False
1374	Property Type:	icom_core:Person
1375	Cardinality:	Single <u>Multi</u>
1376	Updatability:	Read Write
1377		
1378	The Resource class MAY include additional property definitions which are implementation-defined.	
1379		

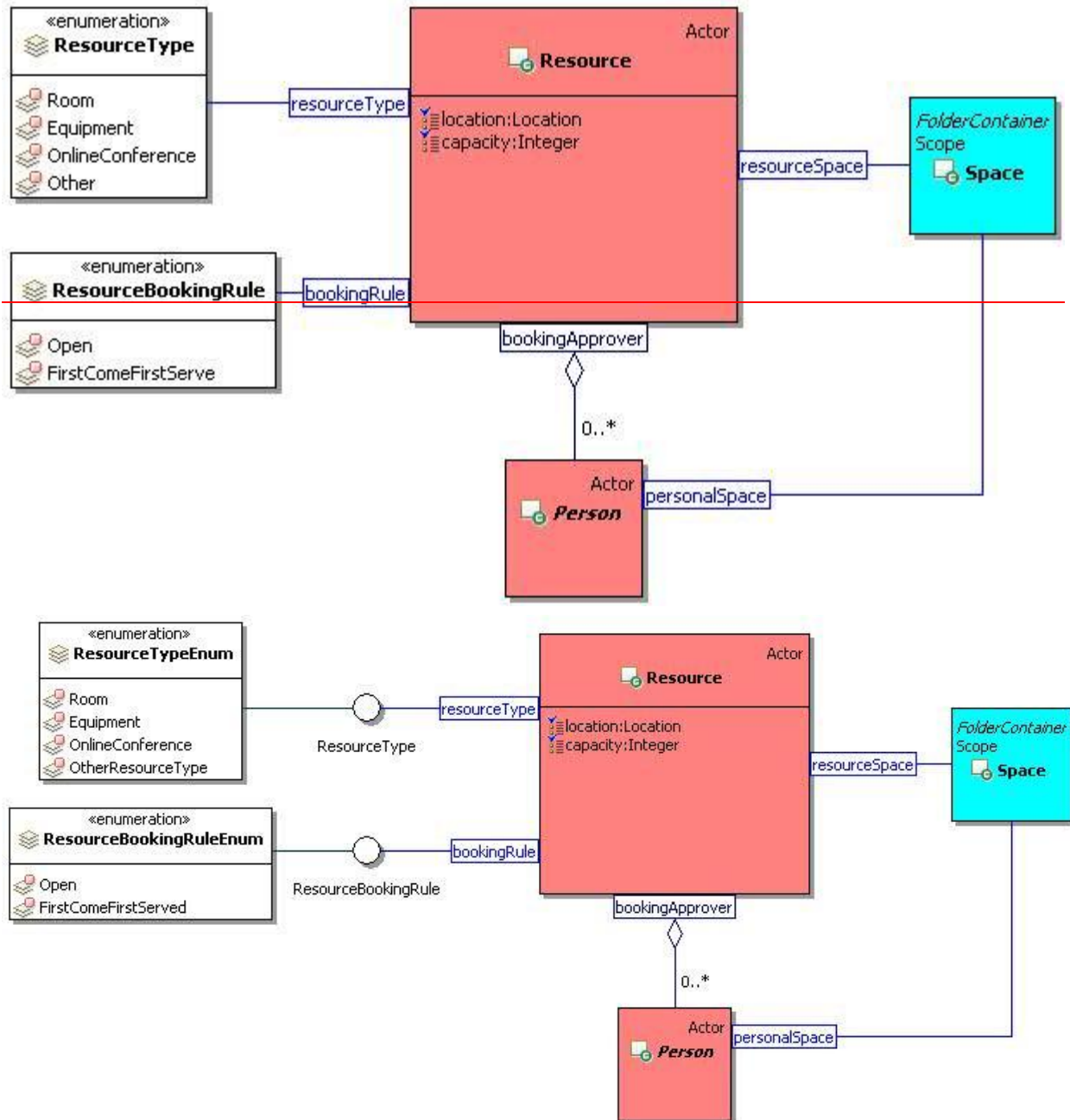


Figure 12: Resource Class Diagram.

3.3.7 ResourceType

Description The ResourceType class is an enum class that enumerates the instances each of which expresses a type of a resource.

3.3.7.1

A resource type is a category of resources.

3.3.7.2 Class Definition

The ResourceType class is ~~defined by the~~ a mixin class which defines a resource type.

The ResourceType class has attribute values:

localNamespace

Value: icom_core

localName

Value: ResourceType

extendsFrom

Value:

stereotype

Value: primarymixin

description

Value: ResourceType is a mixin class which defines a type of resources.

propertyDefinitions

The values for this attribute are defined in Section 3.3.7.3.

3.3.7.3 Property Definitions

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

3.3.8 ResourceTypeEnum

The ResourceTypeEnum class is an enum class that enumerates the

~~**isEnumeration**~~

~~Value: TRUE~~

~~**description**~~

~~Value: An enumeration of~~ instances each of which expresses a type of ~~a resource.~~ resources.

The ResourceTypeEnum class has attribute values:

localNamespace

Value: icom_core

localName

Value: ResourceTypeEnum

extendsFrom

Value: ResourceType

1430

1431 **stereotype**

1432 Value: primary

1433

1434 **isEnumeration**

1435 Value: TRUE

1436

1437 **description**

1438 Value: A type of resources.

1439

1440 **instances**

1441 Value: <icom_core:Room, icom_core:Equipment, icom_core:OnlineConference,

1442 icom_core:~~Other~~OtherResourceType>

1443

1444 ~~The following~~ICOM defines four resource types ~~are defined by ICOM:~~

1445 • **icom_core:Room** ~~to express that~~ a resource represents a room.

1446 • **icom_core:Equipment** ~~to express that~~ a resource represents an equipment.

1447 • **icom_core:OnlineConference** ~~to express that~~ a resource represents an online conference.

1448 • **icom_core:**~~Other~~ ~~to express that~~OtherResourceType a resource represents other things.

1449

1450 ~~3.3.83.3.9~~ **ResourceBookingRule**

1451 ~~The ResourceBookingRule class is an enum class that enumerates the instances each of which~~

1452 ~~expresses a booking rule.~~

1453 **3.3.9.1 Description**

1454 A resource booking rule is a strategy for allocating resources for calendar scheduling.

1455 **3.3.9.2 Class Definition**

1456 The ResourceBookingRule class is ~~defined by the~~a mixin class which defines a resource booking rule.

1457 The ResourceBookingRule class has attribute values:

1458

1459 **localNamespace**

1460 Value: icom_core

1461

1462 **localName**

1463 Value: ResourceBookingRule

1464

1465 **extendsFrom**

1466 Value:

1467

1468 **stereotype**

1469 Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.3.9.3.

3.3.9.3 Property Definitions

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

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: ~~An enumeration of instances each of which expresses a~~A resource booking rule ~~for allocating resources for calendar scheduling.~~

instances

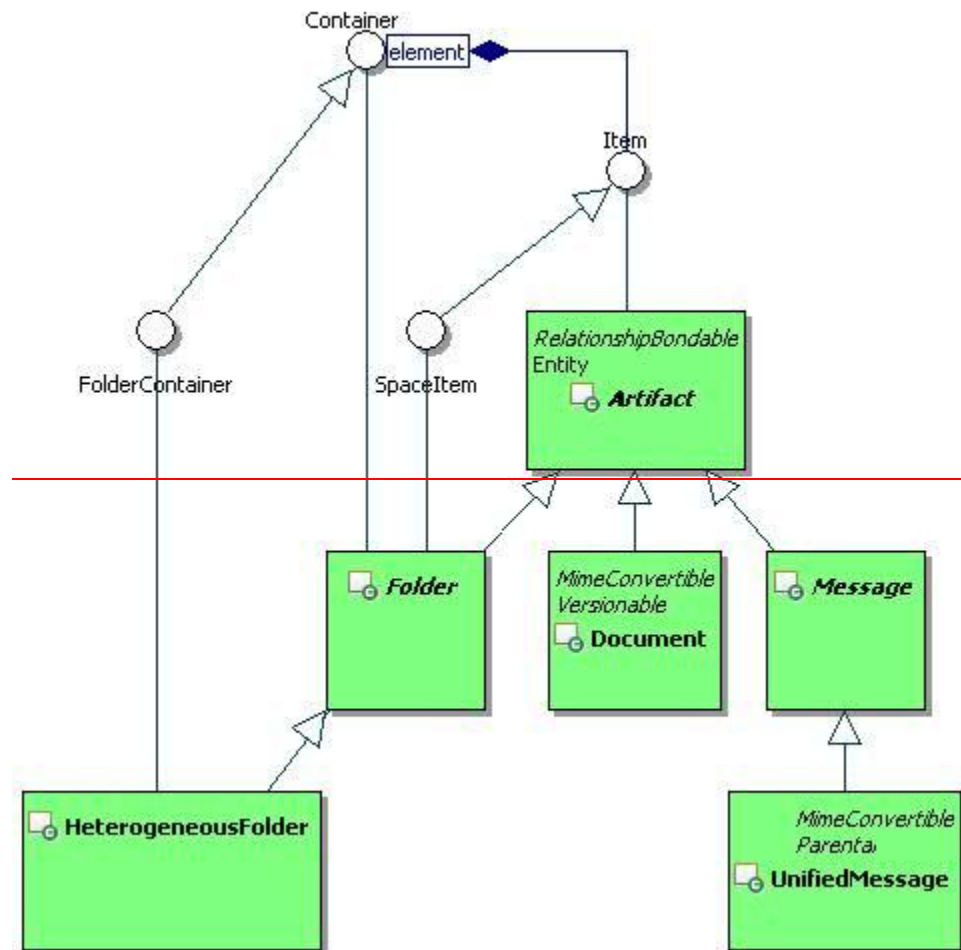
Value: <icom_core:Open, icom_core:FirstComeFirstServed>

~~The following ICOM defines two~~ resource booking rules ~~are defined by ICOM:~~

- ~~icom_core:Open to express that~~ a resource is open for booking.
- ~~icom_core:FirstComeFirstServe to express that~~**FirstComeFirstServed** a resource is first come first served.

3.4 Artifact Branch

3.4.1 ~~UML Diagram of~~ Artifact and Top-Level Subclasses



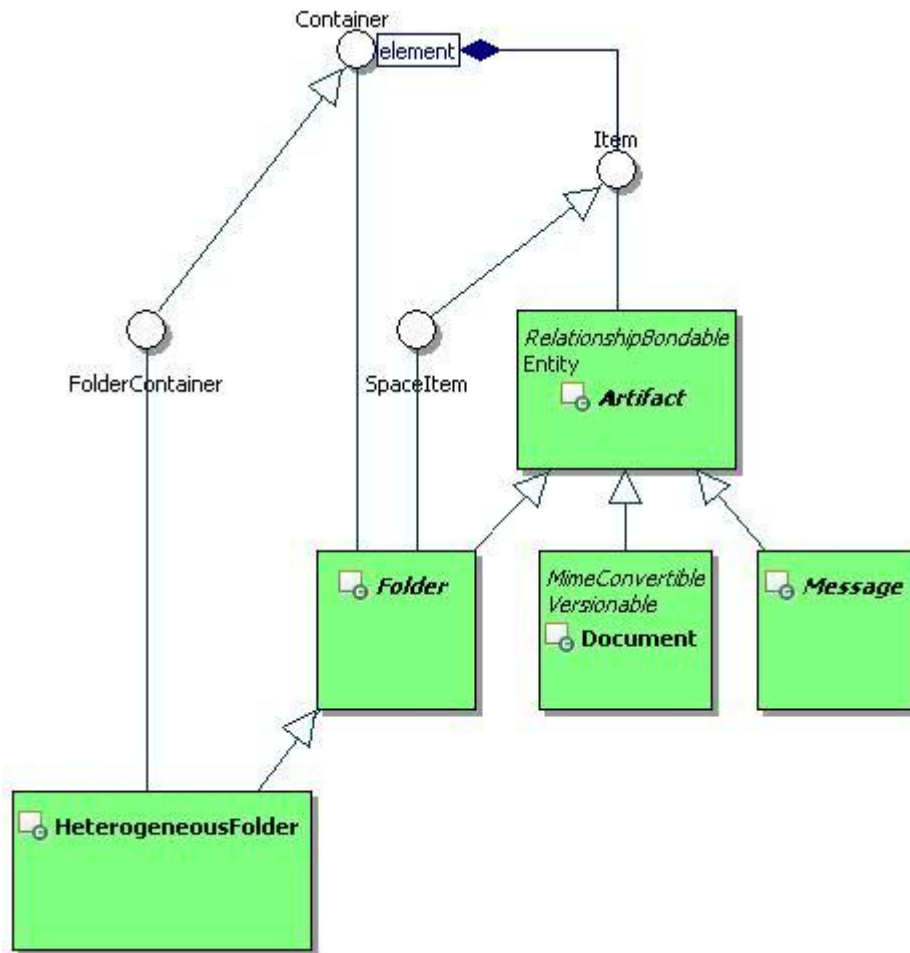


Figure 13: Artifact Branch.

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 ~~is defined by the~~ has attribute values:

localNamespace

Value: icom_core

localName

Value: Item

1534
1535 **extendsFrom**
1536 Value: icom_core:Identifiable
1537
1538 **stereotype**
1539 Value: mixin
1540
1541 **description**
1542 Value: Item is a mixin class which defines the characteristics of entities that can be placed in a
1543 Container.
1544
1545 **propertyDefinitions**
1546 The values for this attribute are defined in Section 3.4.2.3.

1547 **3.4.2.3 Property Definitions**

1548 The Item class inherits property definitions from super classes.
1549 The Item class MUST have the property definition:

1550
1551 **icom_core:parent**
1552 Description: A parent container of an item.
1553 Required: False
1554 Inherited: True
1555 Property Type: icom_core:Container
1556 Cardinality: Single
1557 Updatability: Read Only

1558
1559 The Item class MAY have the optional property definition:

1560
1561 **icom_core:container**
1562 Description: Zero, one, or more containers of an item, including the parent
1563 container.
1564 Required: False
1565 Inherited: False
1566 Property Type: icom_core:Container
1567 Cardinality: Multi
1568 Updatability: Read Write

1569
1570 The Item class MAY include additional property definitions which are implementation-defined.
1571

1572 **3.4.3 Spaceltem**

1573 **3.4.3.1 Description**

1574 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 ~~is defined by the~~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 ~~is defined by the~~has attribute values:

localNamespace

Value: icom_core

localName

Value: Container

1615 **extendsFrom**
 1616 Value: icom_core:Extent
 1617
 1618 **stereotype**
 1619 Value: mixin
 1620
 1621 **description**
 1622 Value: A container is an extent that contains items.
 1623
 1624 **propertyDefinitions**
 1625 The values for this attribute are defined in Section 3.4.4.3.

1626 3.4.4.3 Property Definitions

1627 The Container class inherits property definitions from super classes.
 1628 The Container class MUST have the property definition:

1629
 1630 **icom_core:element**
 1631 Description: Elements of a container, i.e. items whose parent container is
 1632 the container or whose containers include the container.
 1633 Required: False
 1634 Inherited: False
 1635 Property Type: icom_core:Item
 1636 Cardinality: Multi
 1637 Updatability: Read Only

1638
 1639 The Container class MAY include additional property definitions which are implementation-defined.
 1640

1641 3.4.5 FolderContainer

1642 3.4.5.1 Description

1643 A folder container is a container which may contain folders. Space and heterogeneous folder are folder
 1644 containers.

1645 3.4.5.2 Class Definition

1646 The FolderContainer class is a mixin class that defines the characteristics of containers that may contain
 1647 folders.

1648 The FolderContainer class ~~is defined by the~~has attribute values:

1649
 1650 **localNamespace**
 1651 Value: icom_core
 1652
 1653 **localName**
 1654 Value: FolderContainer

1655
1656 **extendsFrom**
1657 Value: icom_core:Container
1658
1659 **stereotype**
1660 Value: mixin
1661
1662 **description**
1663 Value: A folder container is a container which may contain folders.
1664
1665 **propertyDefinitions**
1666 The values for this attribute are defined in Section 3.4.5.3.

1667 3.4.5.3 Property Definitions

1668 The FolderContainer class inherits property definitions from super classes.
1669 The FolderContainer class MAY include additional property definitions which are implementation-defined.
1670

1671 3.4.6 Artifact

1672 3.4.6.1 Description

1673 An artifact is a result of a communication, cooperation, content creation, or collaboration activity.
1674 Note: Document versioning is an example of content creation activity resulting in an artifact (a version of a
1675 document).

1676 3.4.6.2 Class Definition

1677 | The Artifact class ~~is defined by the~~has attribute values:

1678
1679 **localNamespace**
1680 Value: icom_core
1681
1682 **localName**
1683 Value: Artifact
1684
1685 **extendsFrom**
1686 Value: icom_core:Entity, icom_core:Item, icom_meta:RelationshipBondable
1687 Optional Value: icom_core:SpaceItem
1688
1689 **stereotype**
1690 Value: primary
1691
1692 **isAbstract**
1693 Value: TRUE
1694

1695 **description**
1696 Value: An artifact is a result of a communication, cooperation, content creation, or collaboration
1697 activity.

1698
1699 **propertyDefinitions**

1700 The values for this attribute are defined in Section 3.4.6.3.

1701 **3.4.6.3 Property Definitions**

1702 The Artifact class inherits property definitions from super classes.

1703 The Artifact class MUST have the property definitions:

1704
1705 **icom_core:description**

1706	Description:	A description of an artifact.
1707	Required:	False
1708	Inherited:	False
1709	Property Type:	String
1710	Cardinality:	Single
1711	Updatability:	Read Write

1712
1713 **icom_core:userCreationDate**

1714	Description:	Date and time when an artifact <u>iswas</u> created. This field can
1715		be set by application.
1716	Required:	False
1717	Inherited:	False
1718	Property Type:	DateTime
1719	Cardinality:	Single
1720	Updatability:	Read Write

1721
1722 **icom_core:userLastModificationDate**

1723	Description:	Date and time when an artifact <u>iswas</u> last modified. This field
1724		can be set by application.
1725	Required:	False
1726	Inherited:	False
1727	Property Type:	DateTime
1728	Cardinality:	Single
1729	Updatability:	Read Write

1730
1731 **icom_meta:property**

1732	Description:	Zero or more extended properties of an artifact.
1733	Required:	False
1734	Inherited:	False
1735	Property Type:	icom_meta:Property
1736	Cardinality:	Multi

1737	Updatability:	Read Write
1738		
1739	icom_meta:viewerProperty	
1740	Description:	Zero or more extended properties of an artifact visible to a
1741		viewer.
1742	Required:	False
1743	Inherited:	False
1744	Property Type:	icom_meta:Property
1745	Cardinality:	Multi
1746	Updatability:	Read Write
1747		
1748	icom_meta:relationship	
1749	Description:	Zero or more relationships associated with an artifact.
1750	Required:	False
1751	Inherited:	False
1752	Property Type:	icom_meta:Relationship
1753	Cardinality:	Multi
1754	Updatability:	Read Only
1755		
1756	The Artifact class MAY include additional property definitions which are implementation-defined.	
1757		

1772

1773 **localName**

1774 Value: Folder

1775

1776 **extendsFrom**

1777 Value: icom_core:Artifact, icom_core:Container, icom_core:SpaceItem

1778

1779 **stereotype**

1780 Value: primary

1781

1782 **isAbstract**

1783 Value: TRUE

1784

1785 **description**

1786 Value: A folder is an artifact that may contain other artifacts.

1787

1788 **propertyDefinitions**

1789 The values for this attribute are defined in Section 3.4.7.3.

1790 **3.4.7.3 Property Definitions**

1791 The Folder class inherits property definitions from super classes.

1792 The Folder class MUST have the property definition:

1793

1794 **icom_core:parent**

1795 Description:	A parent container of a folder.
1796 Required:	False
1797 Inherited:	True
1798 Property Type:	icom_core:FolderContainer
1799 Cardinality:	Single
1800 Updatability:	Read Only

1801

1802 The Folder class MAY include additional property definitions which are implementation-defined.

1803

1804 **3.4.8 HeterogeneousFolder**

1805 **3.4.8.1 Description**

1806 A heterogeneous folder is an unconstrained folder to contain any type of artifacts. ~~It is typically used for~~

1807 ~~document folders, inbox, and trash folder of a space.~~

1808 Note: It is typically used for document folders, inbox, outbox, and trash folder of a space.

1809 **3.4.8.2 Class Definition**

1810 The HeterogeneousFolder class ~~is defined by the~~has attribute values:

1811

1812 **localNamespace**
1813 Value: icom_core
1814
1815 **localName**
1816 Value: HeterogeneousFolder
1817
1818 **extendsFrom**
1819 Value: icom_core:Folder, icom_core:FolderContainer
1820
1821 **stereotype**
1822 Value: primary
1823
1824 **description**
1825 Value: A heterogeneous folder is an unconstrained folder to contain any type of artifacts.
1826
1827 **propertyDefinitions**
1828 The values for this attribute are defined in Section 3.4.8.3.

1829 **3.4.8.3 Property Definitions**

1830 The HeterogeneousFolder class inherits property definitions from super classes.

1831 The HeterogeneousFolder class MUST have the property definition:

1832
1833 **icom_core:element**
1834 Description: Elements of a heterogeneous folder.
1835 Required: False
1836 Inherited: True
1837 Property Type: icom_core:Artifact
1838 Cardinality: Multi
1839 Updatability: Read Only

1840
1841 The HeterogeneousFolder class MAY include additional property definitions which are implementation-
1842 defined.

1843

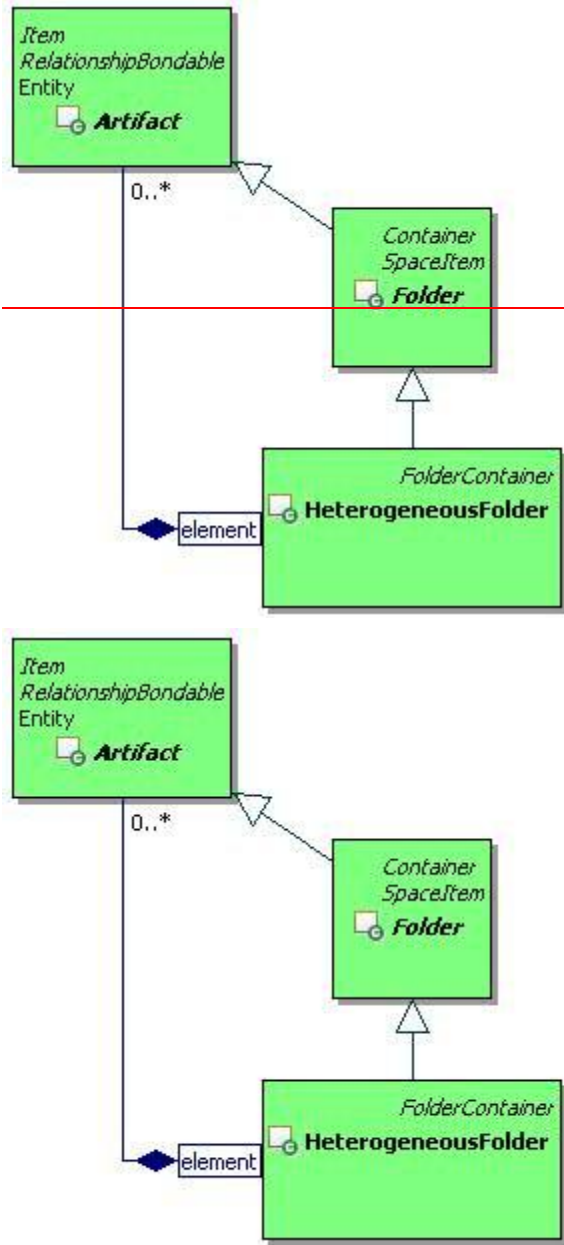


Figure 15: Heterogeneous Folder Class Diagram.

3.5 Access Control Model

3.5.1 Accessor

3.5.1.1 Description

An accessor can be granted or denied access rights to ~~access~~ 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 ~~is defined by the~~has attribute values:

localNamespace

Value: icom_ac

localName

Value: Accessor

extendsFrom

Value: icom_core:Identifiable

stereotype

Value: mixin

description

Value: Accessor 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 granted privileges in role assignments.

propertyDefinitions

The values for this attribute are defined in Section 3.5.1.3.

3.5.1.3 Property Definitions

The Accessor class inherits property definitions from super classes.

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

3.5.2 Owner

3.5.2.1 Description

An owner is a subject that can be the owner of entities.

An owner of an entity MAY always have rights to update the access control list for the entity.

3.5.2.2 Class Definition

The Owner class is a mixin class which defines the characteristics of subjects such as groups and actors that can own entities.

The Owner class ~~is defined by the~~has attribute values:

localNamespace

Value: icom_ac

1892
1893 **localName**
1894 Value: Owner
1895
1896 **extendsFrom**
1897 Value: icom_ac:Accessor
1898
1899 **stereotype**
1900 Value: mixin
1901
1902 **description**
1903 Value: Owner is a mixin class which defines the characteristics of subjects such as groups and
1904 actors that can own entities.
1905
1906 **propertyDefinitions**
1907 The values for this attribute are defined in Section 3.5.2.3.

1908 **3.5.2.3 Property Definitions**

1909 The Owner class inherits property definitions from super classes.
1910 The Owner class MAY include additional property definitions which are implementation-defined.
1911

1912 **3.5.3 RoleDefinition**

1913 **3.5.3.1 Description**

1914 A role definition is a named set of privileges.

1915 **3.5.3.2 Class Definition**

1916 The RoleDefinition class ~~is defined by the~~has attribute values:

1917
1918 **localNamespace**
1919 Value: icom_ac
1920
1921 **localName**
1922 Value: RoleDefinition
1923
1924 **extendsFrom**
1925 Value: icom_core:EntityDefinition
1926
1927 **stereotype**
1928 Value: primary
1929
1930 **description**
1931 Value: A role definition is a named set of privileges.

1932
1933
1934

1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948

1949

1950
1951

1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971

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

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

3.5.4 Role

3.5.4.1 Description

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

3.5.4.2 Class Definition

The Role class ~~is defined by the~~has attribute values:

localNamespace

Value: icom_ac

localName

Value: Role

extendsFrom

Value: icom_core:Subject

stereotype

Value: primary

description

Value: A role assigns a named set of rights to a set of accessors for operations within an assigned scope.

propertyDefinitions

1972 The values for this attribute are defined in Section 3.5.4.3.

1973 **3.5.4.3 Property Definitions**

1974 The Role class inherits property definitions from super classes.

1975 The Role class **MUST** have the property definitions:

1976

1977 **icom_ac:roleDefinition**

1978 Description: A role definition containing a set of privileges.

1979 Required: True

1980 Inherited: False

1981 Property Type: icom_ac:RoleDefinition

1982 Cardinality: Single

1983 Updatability: On Create

1984

1985 **icom_ac:assignedScope**

1986 Description: A scope in which a role is assigned.

1987 Required: True

1988 Inherited: False

1989 Property Type: icom_core:Scope

1990 Cardinality: Single

1991 Updatability: Read Write

1992

1993 **icom_ac:memberAccessor**

1994 Description: Accessors (actors and groups) assigned to a role.

1995 Required: False

1996 Inherited: False

1997 Property Type: icom_ac:Accessor

1998 Cardinality: Multi

1999 Updatability: Read Write

2000

2001 The Role class **MAY** include additional property definitions which are implementation-defined.

2002

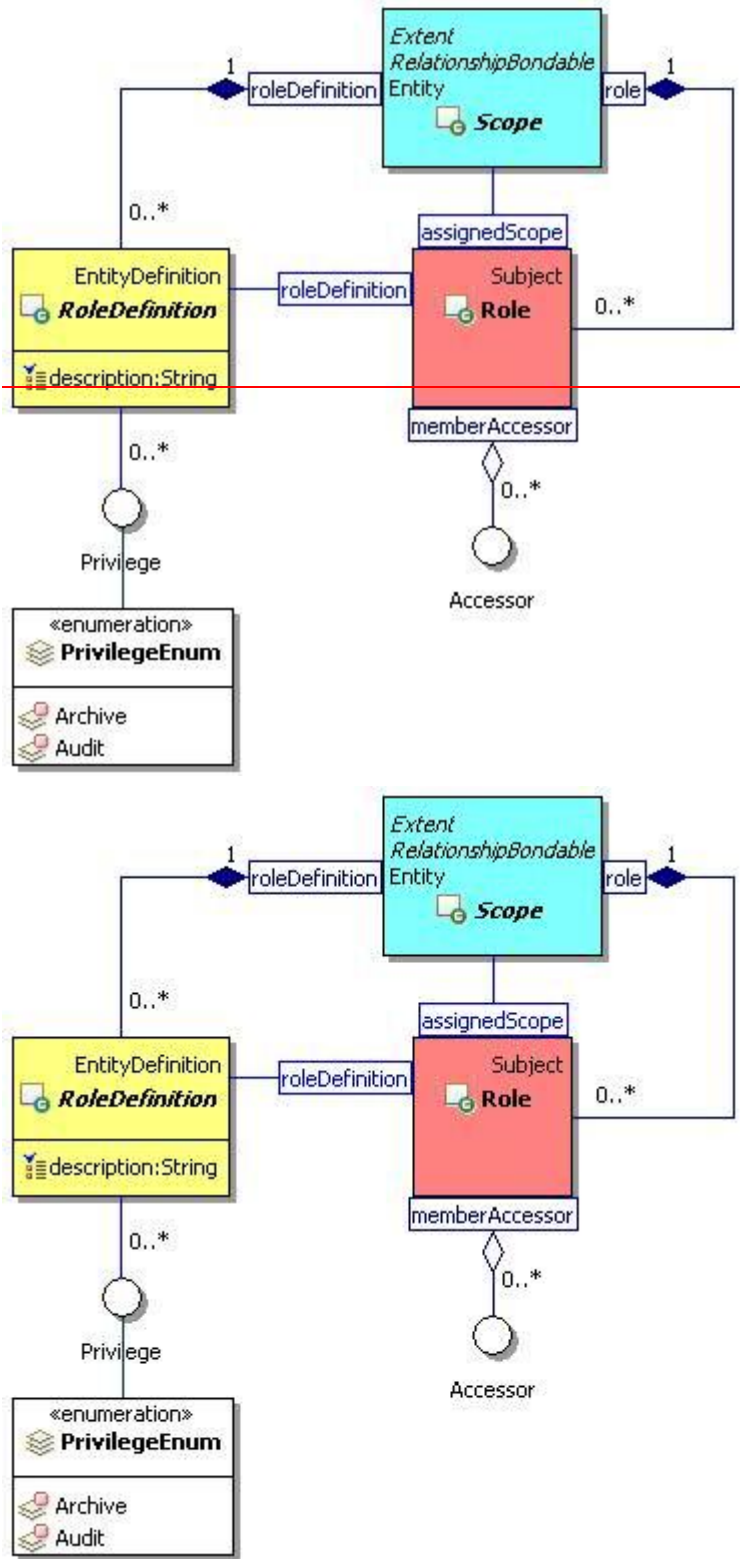


Figure 16: Role Definition and Role Class Diagram.

3.5.5 Privilege

3.5.5.1 Description

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 ~~is defined by the~~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 ~~is defined by the~~has attribute values:

localNamespace

Value: icom_ac

localName

Value: PrivilegeEnum

extendsFrom

2047 Value: icom_ac:Privilege

2048

2049 **stereotype**

2050 Value: primary

2051

2052 **isEnumeration**

2053 Value: TRUE

2054

2055 **description**

2056 | Value: ~~An enumeration of instances each of which expresses a privilege~~Privilege that can be

2057 assigned to a role.

2058

2059 **instances**

2060 Value: <icom_ac:Archive, icom_ac:Audit>

2061

2062 ~~The following~~ICOM defines two privileges ~~are defined by ICOM:~~

2063 | • **icom_ac:Archive** ~~to express~~ a right to archive contents in a scope.

2064 | • **icom_ac:Audit** ~~to express~~ a right to audit activities in a scope.

2065

2066 3.5.7 AccessControlList

2067 3.5.7.1 Description

2068 An access control list (ACL) is an object attached to an entity to specify a list of permissions to access the

2069 entity.

2070 3.5.7.2 Class Definition

2071 | The AccessControlList class ~~is defined by the~~has attribute values:

2072

2073 **localNamespace**

2074 Value: icom_ac

2075

2076 **localName**

2077 Value: AccessControlList

2078

2079 **extendsFrom**

2080 Value:

2081

2082 **stereotype**

2083 Value: primary

2084

2085 **description**

2086 Value: An access control list (ACL) is an object attached to an entity to specify a list of

2087 permissions to access the entity.

2088
2089 **propertyDefinitions**
2090 The values for this attribute are defined in Section 3.5.7.3.

2091 **3.5.7.3 Property Definitions**

2092 The AccessControlList class MUST have the property definitions:

2094 **icom_ac:object**

2095 Description:	Associated object.
2096 Required:	True
2097 Inherited:	False
2098 Property Type:	icom_core:Entity
2099 Cardinality:	Single
2100 Updatability:	On Create

2102 **icom_ac:accessControlEntry**

2103 Description:	One or more access control entries.
2104 Required:	True
2105 Inherited:	False
2106 Property Type:	icom_ac:AccessControlEntry
2107 Cardinality:	Multi
2108 Updatability:	Read Write

2109
2110 AccessControlList class MAY include additional property definitions which are implementation-defined.
2111

2112 **3.5.8 AccessControlEntry**

2113 **3.5.8.1 Description**

2114 An access control entry specifies access types granted to or denied for an accessor.

2115 **3.5.8.2 Class Definition**

2116 The AccessControlEntry class ~~is defined by the~~has attribute values:

2118 **localNamespace**

2119 Value: icom_ac

2121 **localName**

2122 Value: AccessControlEntry

2124 **extendsFrom**

2125 Value:

2126

2127 **stereotype**
 2128 Value: primary
 2129
 2130 **description**
 2131 Value: An access control entry is associated with an accessor and contains a list of access
 2132 types (permissions) granted to or denied from the accessor.
 2133
 2134 **propertyDefinitions**
 2135 The values for this attribute are defined in Section 3.5.8.3.

2136 **3.5.8.3 Property Definitions**

2137 The AccessControlEntry class MUST have the property definitions:

2138

2139 **icom_ac:subject**

2140	Description:	Associated subject.
2141	Required:	True
2142	Inherited:	False
2143	Property Type:	icom_ac:Accessor
2144	Cardinality:	Single
2145	Updatability:	On Create

2146

2147 **icom_ac:grant**

2148	Description:	One or more access types granted to a subject.
2149	Required:	False
2150	Inherited:	False
2151	Property Type:	icom_ac:AccessType
2152	Cardinality:	Multi
2153	Updatability:	Read Write

2154

2155 **icom_ac:deny**

2156	Description:	One or more access type denied for a subject.
2157	Required:	False
2158	Inherited:	False
2159	Property Type:	icom_ac:AccessType
2160	Cardinality:	Multi
2161	Updatability:	Read Write

2162

2163 The AccessControlEntry class MAY include additional property definitions which are implementation-
 2164 defined.

2165

2166 **3.5.9 AccessType**

2167 An AccessType is an access right granted through an access control entry.

3.5.9.1 Class Definition

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

The AccessType class ~~is defined by the~~has attribute values:

localNamespace

Value: icom_ac

localName

Value: AccessType

extendsFrom

Value:

stereotype

Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.5.9.2.

3.5.9.2 Property Definitions

The AccessType class inherits property definitions from super classes.

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

3.5.10 AccessTypeEnum

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

The AccessTypeEnum class ~~is defined by the~~has attribute values:

localNamespace

Value: icom_ac

localName

Value: AccessTypeEnum

extendsFrom

Value: icom_ac:AccessType

2209 **stereotype**

2210 Value: primary

2211

2212 **isEnumeration**

2213 Value: TRUE

2214

2215 **description**

2216 | ~~Value: An enumeration of instances each of which expresses an access~~Value: Access type

2217 that can be granted or denied in an access control entry.

2218

2219 **instances**

2220 Value: <icom_ac:Read, icom_ac:Write, icom_ac>Delete>

2221

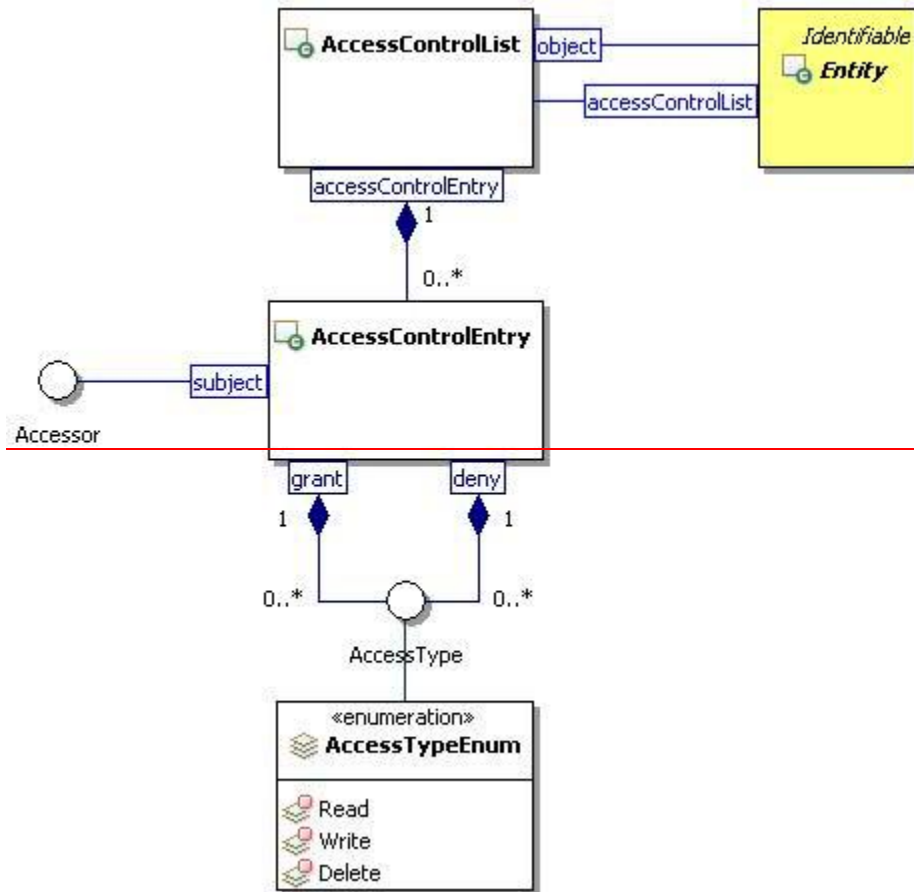
2222 | ~~The following~~ICOM defines three access types~~are defined by ICOM:~~

2223 • **icom_ac:Read** ~~to express~~ a right to retrieve an entity.

2224 • **icom_ac:Write** ~~to express~~ a right to update an entity.

2225 • **icom_ac>Delete** ~~to express~~ a right to delete an entity.

2226



2227

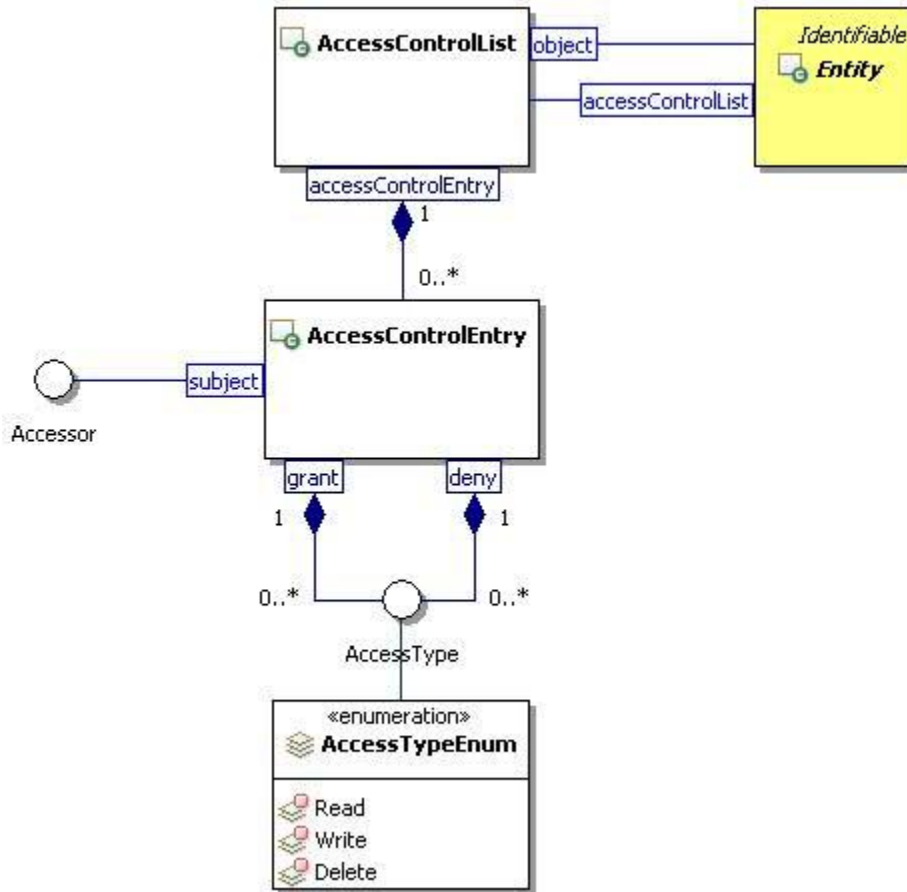


Figure 17: Access Control List Class Diagram.

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. ~~A property type includes string, boolean, decimal, integer, datetime, etc.~~

3.6.1.2 Class Definition

The PropertyDefinition class ~~is defined by the~~ has attribute values:

localNamespace

Value: icom_meta

localName

Value: PropertyDefinition

extendsFrom

2246 Value: icom_core:Identifiable
 2247
 2248 **stereotype**
 2249 Value: primary
 2250
 2251 **description**
 2252 Value: A property definition specifies the name, type, choice, and cardinality of values for
 2253 properties.
 2254
 2255 **propertyDefinitions**
 2256 The values for this attribute are defined in Section 3.6.1.3.

2257 3.6.1.3 Property Definitions

2258 The PropertyDefinition class inherits property definitions from super classes.

2259 The PropertyDefinition class MUST have the property definitions:

2260

2261 **icom_core:namespace**

2262	Description:	Namespace for a property name.
2263	Required:	False
2264	Inherited:	False
2265	Property Type:	String
2266	Cardinality:	Single
2267	Updatability:	Read Write

2268

2269 **icom_core:name**

2270	Description:	Name for a property.
2271	Required:	True
2272	Inherited:	False
2273	Property Type:	String
2274	Cardinality:	Single
2275	Updatability:	Read Write

2276

2277 **icom_core:description**

2278	Description:	A description of a property definition.
2279	Required:	False
2280	Inherited:	False
2281	Property Type:	String
2282	Cardinality:	Single
2283	Updatability:	Read Write

2284

2285 **icom_meta:propertyType**

2286	Description:	Type of a property.
2287	Required:	True

2288	Inherited:	False
2289	Property Type:	icom_meta:PropertyType
2290	Cardinality:	Single
2291	Updatability:	On Create
2292	Choices:	{PropertyChoiceType}
2293	Open Choice:	False

2294

2295 Note: The notation {PropertyChoiceType} represents a set of PropertyChoiceType.

2296

2297 **icom_meta:defaultValue**

2298	Description:	A default value for a property.
2299	Required:	False
2300	Inherited:	False
2301	Property Type:	property-type
2302	Cardinality:	Single
2303	Updatability:	Read Write

2304

2305 **icom_meta:choice**

2306	Description:	An allowed value for a property.
2307	Required:	False
2308	Inherited:	False
2309	Property Type:	icom_meta:PropertyChoiceType
2310	Cardinality:	Multi
2311	Updatability:	Read Write

2312

2313 **icom_meta:cardinality**

2314	Description:	Cardinality of a property specifying whether the property can have “zero or one” or “zero or more” values.
2315		
2316	Required:	True
2317	Inherited:	False
2318	Property Type:	icom_meta:Cardinality
2319	Cardinality:	Single
2320	Updatability:	On Create

2321

2322 **icom_meta:minValue**

2323	Description:	Minimum value for an integer or decimal property.
2324	Required:	False
2325	Inherited:	False
2326	Property Type:	Integer Decimal
2327	Cardinality:	Single
2328	Updatability:	Read Write

2329

2330 **icom_meta:maxValue**

2331	Description:	Maximum value for an integer or decimal property.
2332	Required:	False
2333	Inherited:	False
2334	Property Type:	Integer Decimal
2335	Cardinality:	Single
2336	Updatability:	Read Write
2337		
2338	The PropertyDefinition class MAY include additional property definitions which are implementation-	
2339	defined.	
2340		

2341 **3.6.2 Property**

2342 **3.6.2.1 Description**

2343 The property holds a property value.

2344 **3.6.2.2 Class Definition**

2345 | The Property class ~~is defined by the~~has attribute values:

2346		
2347	localNamespace	
2348	Value:	icom_meta
2349		
2350	localName	
2351	Value:	Property
2352		
2353	extendsFrom	
2354	Value:	
2355		
2356	stereotype	
2357	Value:	primary
2358		
2359	description	
2360	Value:	A property value.
2361		
2362	propertyDefinitions	
2363	The values for this attribute are defined in Section 3.6.2.3.	

2364 **3.6.2.3 Property Definitions**

2365 The Property class MUST have the property definitions:

2366		
2367	icom_meta:propertyDefinition	
2368	Description:	A property definition that specifies the name, type, and
2369		cardinality of a property.
2370	Required:	True

2371	Inherited:	False
2372	Property Type:	icom_meta:PropertyDefinition
2373	Cardinality:	Single
2374	Updatability:	On Create
2375		
2376	icom_meta:value	
2377	Description:	A value of a property.
2378	Required:	True
2379	Inherited:	False
2380	Property Type:	property-type
2381	Cardinality:	Single
2382	Updatability:	Read Write
2383		
2384	The Property class MAY include additional property definitions which are implementation-defined.	
2385		

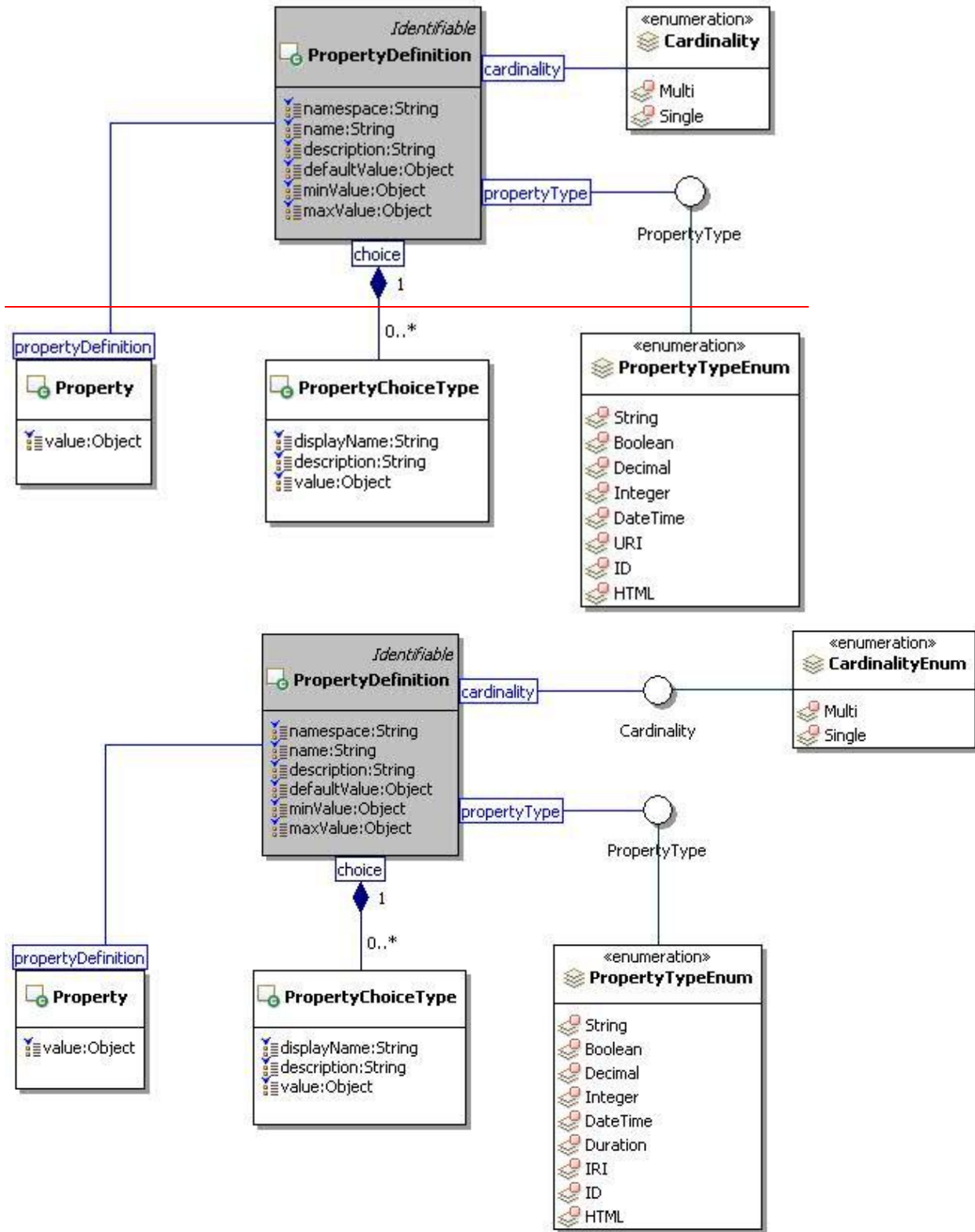


Figure 18: Property Definition and Property Class Diagram.

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 ~~is defined by the~~has attribute values:

localNamespace

Value: icom_meta

localName

Value: PropertyChoiceType

extendsFrom

Value:

stereotype

Value: primary

description

Value: A choice for a property value.

propertyDefinitions

The values for this attribute are defined Section 3.6.3.3.

3.6.3.3 Property Definitions

The PropertyChoiceType class MUST have the property definitions:

~~icom_meta:displayName~~

~~Description: Display name of a property choice.~~

~~Required: True~~

~~Inherited: False~~

~~Property Type: String~~

~~Cardinality: Single~~

~~Updatability: Read Write~~

icom_core:description

Description: A description of a property choice.

Required: False

Inherited: False

Property Type: String

2430 Cardinality: Single
2431 Updatability: Read Write

2432

2433 **icom_meta:displayName**

2434 Description: Display name of a property choice.

2435 Required: True

2436 Inherited: False

2437 Property Type: String

2438 Cardinality: Single

2439 Updatability: Read Write

2440

2441 **icom_meta:value**

2442 Description: A value of a property choice.

2443 Required: True

2444 Inherited: False

2445 Property Type: **property-type**

2446 Cardinality: Single

2447 Updatability: Read Write

2448

2449 The PropertyChoiceType class MAY include additional property definitions which are implementation-
2450 defined.

2451

2452 **3.6.4 PropertyType**

2453 A PropertyType expresses a name of a **property-type**.

2454 **3.6.4.1 Class Definition**

2455 The PropertyType class is a mixin class which expresses a name of a **property-type**.

2456 The PropertyType class ~~is defined by the~~has attribute values:

2457

2458 **localNamespace**

2459 Value: icom_meta

2460

2461 **localName**

2462 Value: PropertyType

2463

2464 **extendsFrom**

2465 Value:

2466

2467 **stereotype**

2468 Value: mixin

2469

2470 **description**

2471 Value: PropertyType is a mixin class which expresses a name of a **property-type**.

2472

2473 **propertyDefinitions**

2474 The values for this attribute are defined in Section 3.6.4.2.

2475 3.6.4.2 Property Definitions

2476 The PropertyType class MAY include additional property definitions which are implementation-defined.

2477

2478 3.6.5 PropertyTypeEnum

2479 The PropertyTypeEnum class is an enum class that enumerates the instances each of which expresses
2480 the name of a **property-type**.

2481 The PropertyTypeEnum class has attribute values:

2482

2483 **localNamespace**

2484 Value: icom_meta

2485

2486 **localName**

2487 Value: PropertyTypeEnum

2488

2489 **extendsFrom**

2490 Value: PropertyType

2491

2492 **stereotype**

2493 Value: primary

2494

2495 **isEnumeration**

2496 Value: TRUE

2497

2498 **description**

2499 ~~Value: An enumeration of instances each of which expresses the name~~Value: Name of a
2500 basic data type.

2501

2502 **instances**

2503 Value: <icom_meta:String, icom_meta:Boolean, icom_meta:Decimal, icom_meta:Integer,
2504 icom_meta:Datetime, icom_meta:~~URI~~Duration, icom_meta:IRI, icom_meta:ID, icom_meta:HTML>

2505

2506 ~~The following names of ICOM defines nine~~ data types ~~are defined by ICOM:~~

- 2507 • **icom_meta:String** is equivalent to expressXML schema type xsd:string.
- 2508 • **icom_meta:Boolean** is equivalent to expressXML schema type xsd:boolean.
- 2509 • **icom_meta:Decimal** is equivalent to expressXML schema type xsd:decimal.
- 2510 • **icom_meta:Integer** is equivalent to expressXML schema type xsd:integer.
- 2511 • **icom_meta:Datetime** is equivalent to expressXML schema type xsd:dateTime.

- icom_meta:URI is equivalent to expressXML schema type xsd:duration.
- icom_meta:IRI is equivalent to XML schema type xsd:anyURI.
- icom_meta:ID ~~to express~~ opaque object identifiers.
- icom_meta:HTML ~~to express~~ documents or fragments of Hypertext Markup Language (HTML) content

Note: ICOM uses ~~several~~ basic data types defined by “XML Schema Part 2: Datatypes Second Edition” (W3C Recommendation, 28 October 2004, <http://www.w3.org/TR/xmlschema-2/>).

3.6.6 Cardinality

3.6.6.1 Description

Cardinality specifies whether a property is single or multi valued.

3.6.6.2 Class Definition

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

The Cardinality class has attribute values:

localNamespace

Value: icom meta

localName

Value: Cardinality

extendsFrom

Value:

stereotype

Value: mixin

description

Value: Cardinality is a mixin class which defines whether a property is single or multi valued.

propertyDefinitions

The values for this attribute are defined in Section 3.6.6.3.

3.6.6.3 Property Definitions

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

3.6.7 CardinalityEnum

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

The ~~Cardinality is defined by the~~ CardinalityEnum has attribute values:

localNamespace

Value: icom_meta

localName

Value: CardinalityEnum

extendsFrom

Value: Cardinality

stereotype

Value: primary

isEnumeration

Value: TRUE

description

Value: ~~An enumeration of instances each of which expresses the cardinality~~ Cardinality of a property.

instances

Value: <icom_meta:Single, icom_meta:Multi>

~~The following ICOM defines two cardinality types are defined by ICOM:~~

- **icom_meta:Single** ~~to express that~~ a property can have zero or one value (if property is not required), or exactly one value (if property is required).
- **icom_meta:Multi** ~~to express that~~ a property can have zero or more values (if property is not required), or one or more values (if property is required).

~~3.6.7~~3.6.8 UML Diagram of Marker and Subclasses

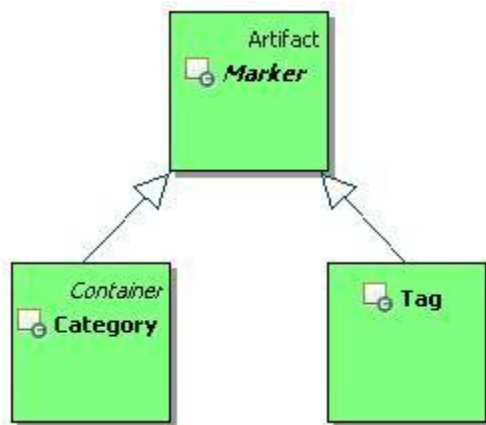


Figure 19: Marker Branch.

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

~~3.6.8~~3.6.9 Marker

~~3.6.8.13~~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.

~~3.6.8.23~~3.6.9.2 Class Definition

The Marker class ~~is defined by the~~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.

3.6.8.3.6.9.3 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.

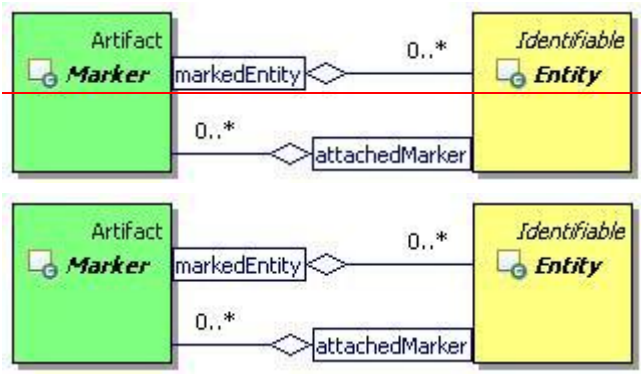


Figure 20: Marker Class Diagram.

3.6.9.3.6.10 Category

3.6.9.13.6.10.1 Description

A category is a marker that classifies entities ~~by taxonomy~~.

3.6.9.23.6.10.2 Class Definition

The Category class ~~is defined by the~~has attribute values:

~~localNamespace~~

localNamespace

Value: icom_meta

2645 ~~localName~~

2646

2647 localName

2648 Value: Category

2649

2650 ~~extendsFrom~~

2651

2652 extendsFrom

2653 Value: icom_meta:Marker, icom_core:Container

2654

2655 **stereotype**

2656 Value: primary

2657

2658 **description**

2659 Value: A category is a marker that classifies entities ~~by taxonomy~~.

2660

2661 **propertyDefinitions**

2662 The values for this attribute are defined in Section 3.6.10.3.

2663 ~~3.6.9.33.6.10.3~~ **3.6.10.3 Property Definitions**

2664 The Category class inherits property definitions from super classes.

2665 The Category class MUST have the property definitions:

2666

2667 **icom_meta:superCategory**

2668	Description:	A super category.
2669	Required:	False
2670	Inherited:	False
2671	Property Type:	icom_meta:Category
2672	Cardinality:	Single
2673	Updatability:	Read Only

2674

2675 **icom_meta:subcategory**

2676	Description:	Zero or more sub categories.
2677	Required:	False
2678	Inherited:	False
2679	Property Type:	icom_meta:Category
2680	Cardinality:	Multi
2681	Updatability:	Read Only

2682

2683 **icom_meta:~~isA~~abstract**

2684	Description:	Indicates whether a category is abstract or concrete.
2685	Required:	False
2686	Inherited:	False

Property Type: Boolean
Cardinality: Single
Updatability: Read Write

icom_meta:propertyDefinition

Description: Optional or mandatory properties for a category application.
Required: False
Inherited: False
Property Type: icom_meta:PropertyDefinition
Cardinality: Multi
Updatability: Read Write

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

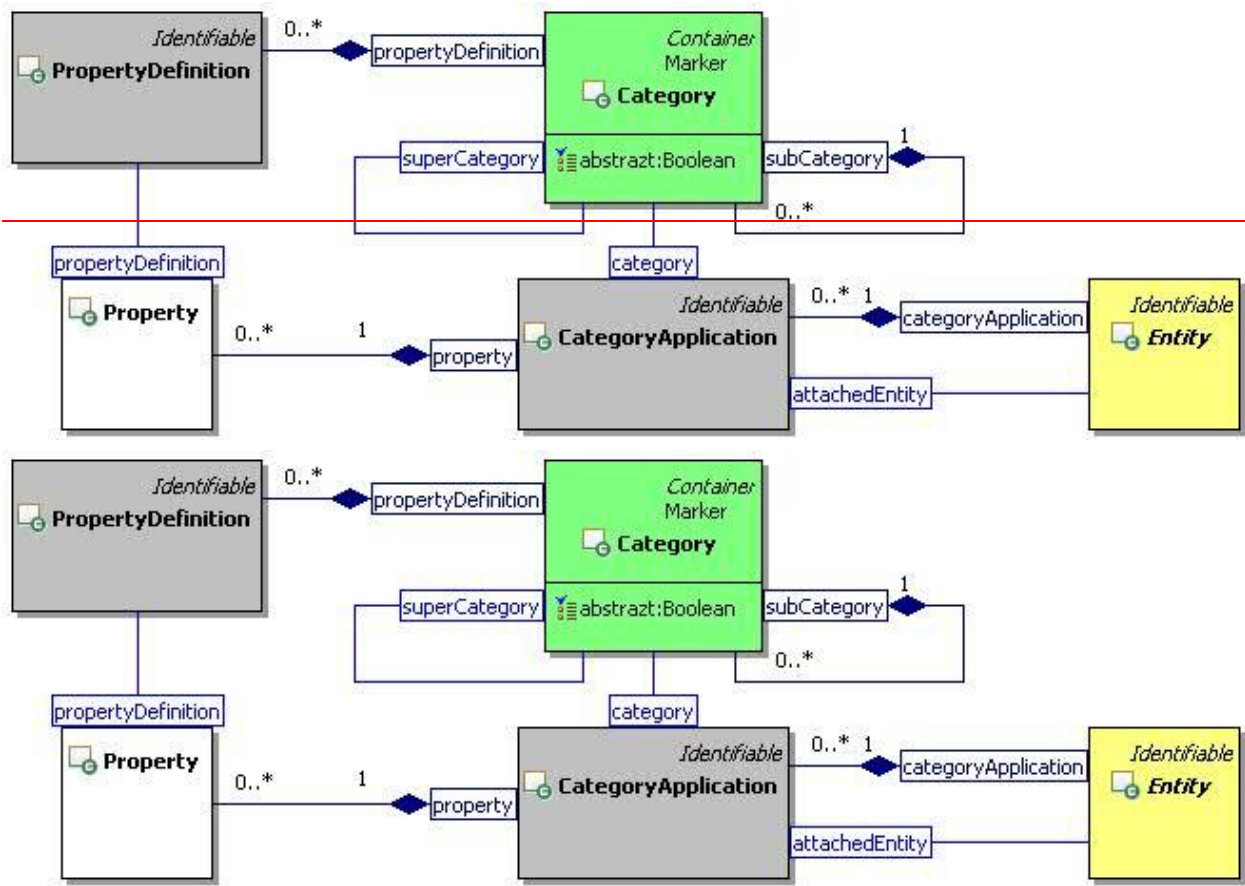


Figure 21: Category and Category Application Class Diagram.

3.6.103.6.11 CategoryApplication

3.6.10.13.6.11.1 Description

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

~~3.6.10~~3.6.11.2 Class Definition

The CategoryApplication class ~~is defined by the~~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.

propertyDefinitions

The values for this attribute are defined in Section 3.6.11.3.

~~3.6.10~~3.6.11.3 Property Definitions

The CategoryApplication class inherits property definitions from super classes.

The CategoryApplication class MUST have the property definitions:

icom_meta:attachedEntity

Description: An entity ~~onto~~to which a category is applied.

Required: True

Inherited: False

Property Type: icom_core:Entity

Cardinality: Single

Updatability: On Create

icom_meta:category

Description: A category which is applied on an entity.

Required: True

Inherited: False

Property Type: icom_meta:Category

Cardinality: Single

Updatability: On Create

icom_meta:property

2750 Description: Zero or more properties.
 2751 Required: False
 2752 Inherited: False
 2753 Property Type: icom_meta:Property
 2754 Cardinality: Multi
 2755 Updatability: Read Write
 2756
 2757 The CategoryApplication class MAY include additional property definitions which are implementation-
 2758 defined.
 2759

2760 ~~3.6.11~~3.6.12 Tag

2761 ~~3.6.11.1~~3.6.12.1 Description

2762 A tag is a marker that labels entities by a keyword.

2763 ~~3.6.11.2~~3.6.12.2 Class Definition

2764 The Tag class ~~is defined by the~~has attribute values:

2765
 2766 **localNamespace**
 2767 Value: icom_meta
 2768
 2769 **localName**
 2770 Value: Tag
 2771
 2772 **extendsFrom**
 2773 Value: icom_meta:Marker
 2774
 2775 **stereotype**
 2776 Value: primary
 2777
 2778 **description**
 2779 Value: A tag is a marker that labels entities by a keyword.
 2780
 2781 **propertyDefinitions**
 2782 The values for this attribute are defined in Section 3.6.12.3.

2783 ~~3.6.11.3~~3.6.12.3 Property Definitions

2784 The Tag class inherits property definitions from super classes.

2785 The Tag class MUST have the property definition:

2786
 2787 **icom_meta:applicationCount**
 2788 Description: ~~The~~An estimate of the number of times a tag is applied on
 2789 entities.

2790	Required:	False
2791	Inherited:	False
2792	Property Type:	Integer
2793	Cardinality:	Single
2794	Updatability:	Read Only

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

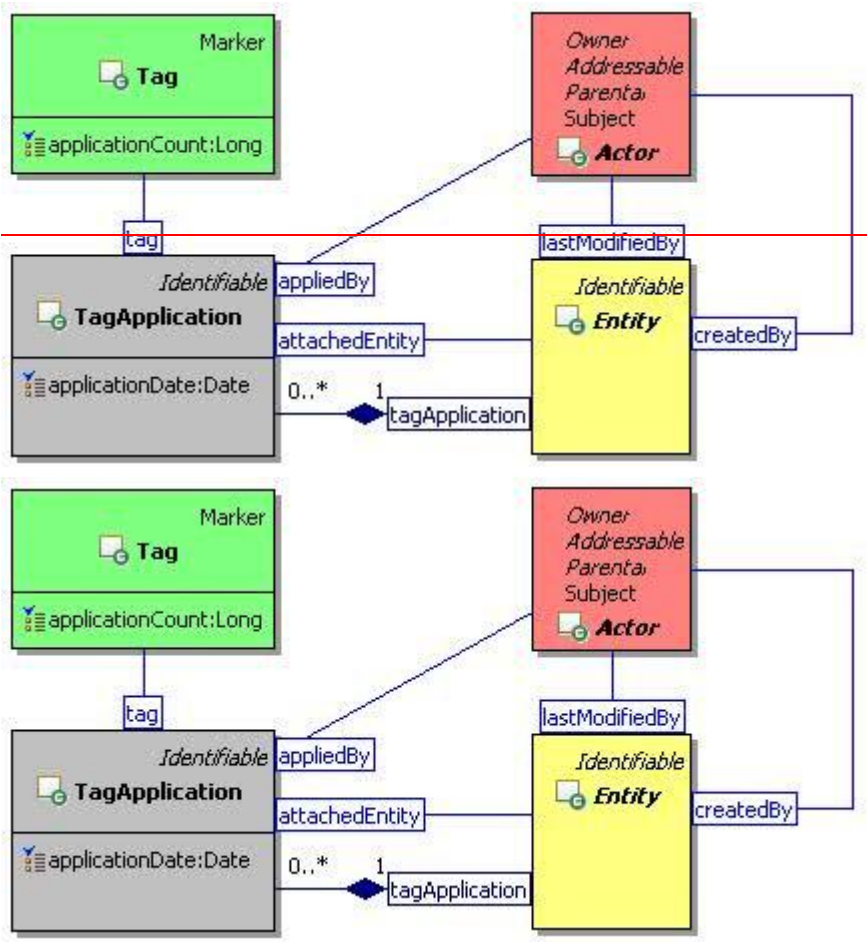


Figure 22: Tag and Tag Application Class Diagram.

3.6.123.6.13 TagApplication

3.6.12.13.6.13.1 Description

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

3.6.12.23.6.13.2 Class Definition

The TagApplication class ~~is defined by the~~has attribute values:

localNamespace

2809 Value: icom_meta
 2810
 2811 **localName**
 2812 Value: TagApplication
 2813
 2814 **extendsFrom**
 2815 Value: icom_core:Identifiable
 2816
 2817 **stereotype**
 2818 Value: primary
 2819
 2820 **description**
 2821 Value: A tag application is an instance of association between a tag and a specific entity.
 2822
 2823 **propertyDefinitions**
 2824 The values for this attribute are defined in Section 3.6.13.3.

3.6.12-3.6.13.3 Property Definitions

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

icom_meta:attachedEntity

Description:	An entity on which a tag is applied.
Required:	True
Inherited:	False
Property Type:	icom_core:Entity
Cardinality:	Single
Updatability:	On Create

icom_meta:tag

Description:	A tag which is applied onto an entity.
Required:	True
Inherited:	False
Property Type:	icom_meta:Tag
Cardinality:	Single
Updatability:	On Create

icom_meta:appliedBy

Description:	A user who applies a tag onto an entity.
Required:	False
Inherited:	False
Property Type:	icom_core:Actor
Cardinality:	Single

2851	Updatability:	Read Only
2852		
2853	icom_meta:applicationDate	
2854	Description:	A date and time when a tag is applied onto an entity.
2855	Required:	False
2856	Inherited:	False
2857	Property Type:	DateTime
2858	Cardinality:	Single
2859	Updatability:	Read Write

2860

2861 The TagApplication class MAY include additional property definitions which are implementation-defined.

2862

2863 ~~3.6.133.6.14~~ RelationshipBondable

2864 ~~3.6.13.13.6.14.1~~ Description

2865 A relationship bondable entity is an entity which may be related to other entities by a relationship ~~bonded~~.

2866 Note: ~~a~~ relationship ~~cannot be relationship bonded by other relationships, i.e. relationships are~~ can exist

2867 among entities that are not ~~relationship bondable~~ relationships.

2868 ~~3.6.13.23.6.14.2~~ Class Definition

2869 The RelationshipBondable class is a mixin class which defines the characteristics of entities that may be

2870 relationship bonded. It includes almost every subclass of Entity except Relationship.

2871 The RelationshipBondable class ~~is defined by the~~ has attribute values:

2872		
2873	localNamespace	
2874	Value:	icom_meta
2875		
2876	localName	
2877	Value:	RelationshipBondable
2878		
2879	extendsFrom	
2880	Value:	icom_core:Identifiable
2881		
2882	stereotype	
2883	Value:	mixin
2884		
2885	description	
2886	Value:	RelationshipBondable is a mixin class which defines the characteristics of entities that
2887		can be relationship bonded.
2888		
2889	propertyDefinitions	
2890	The values for this attribute are defined in Section 3.6.14.3.	

~~3.6.13~~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.

~~3.6.14~~3.6.15 RelationshipDefinition

~~3.6.14.13~~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.

~~3.6.14.23~~3.6.15.2 Class Definition

The RelationshipDefinition class ~~is defined by the~~has attribute values:

localNamespace

Value: icom_meta

localName

Value: RelationshipDefinition

extendsFrom

Value: icom_core:EntityDefinition

stereotype

Value: primary

description

Value: A relationship definition is an entity that defines a type of relationship.

propertyDefinitions

The values for this attribute are defined in Section 3.6.15.3.

~~3.6.14~~3.6.15.3 Property Definitions

The RelationshipDefinition class inherits property definitions from super classes.

The RelationshipDefinition class MUST have the property definitions:

icom_meta:propertyDefinition

Description: Optional or mandatory properties for a relationship.

Required: False

Inherited: False

Property Type: icom_meta:PropertyDefinition

Cardinality: Multi

2931	Updatability:	Read Write
2932		
2933	icom_meta:allowedSourceType	
2934	Description:	A list of expanded names of relationship bondable classes, indicating that the source entity of a relationship MUST be an instance of a class in the list.
2935		
2936		
2937	Required:	False
2938	Inherited:	False
2939	Property Type:	IRI
2940	Cardinality:	Multi
2941	Updatability:	Read Write
2942		
2943	icom_meta:allowedTargetType	
2944	Description:	A list of expanded names of relationship bondable classes, indicating that the target entity of a relationship MUST be an instance of a class in the list.
2945		
2946		
2947	Required:	False
2948	Inherited:	False
2949	Property Type:	IRI
2950	Cardinality:	Multi
2951	Updatability:	Read Write
2952		
2953	The RelationshipDefinition class MAY include additional property definitions which are implementation-defined.	
2954		
2955		

2956 ~~3.6.15~~3.6.16 Relationship

2957 ~~3.6.15-13.6.16.1~~3.6.16.1 Description

2958 A relationship is an entity that relates a set of entities by a predicate.

2959 ~~3.6.15-23.6.16.2~~3.6.16.2 Class Definition

2960 The Relationship class ~~is defined by the~~has attribute values:

2961	
2962	localNamespace
2963	Value: icom_meta
2964	
2965	localName
2966	Value: Relationship
2967	
2968	extendsFrom
2969	Value: icom_core:Entity
2970	
2971	stereotype

2972 Value: primary

2973

2974 **description**

2975 Value: A relationship is an entity that relates a set of entities by a predicate.

2976

2977 **propertyDefinitions**

2978 The values for this attribute are defined in Section 3.6.16.3.

~~3.6.15~~3.6.16.3 Property Definitions

2980 The Relationship class inherits property definitions from super classes.

2981 The Relationship class MUST have the property definitions:

2982

icom_meta:relationshipDefinition

2984	Description:	A definition of relationships.
2985	Required:	True
2986	Inherited:	False
2987	Property Type:	icom_meta:RelationshipDefinition
2988	Cardinality:	Single
2989	Updatability:	On Create

icom_meta:sourceEntity

2992	Description:	A source entity of a relationship.
2993	Required:	True
2994	Inherited:	False
2995	Property Type:	icom_meta:RelationshipBondable
2996	Cardinality:	Single
2997	Updatability:	On Create

icom_meta:targetEntity

3000	Description:	One or more target entities of a relationship.
3001	Required:	True
3002	Inherited:	False
3003	Property Type:	icom_meta:RelationshipBondable
3004	Cardinality:	Multi
3005	Updatability:	Read Write

icom_meta:property

3008	Description:	Zero or more properties.
3009	Required:	False
3010	Inherited:	False
3011	Property Type:	icom_meta:Property
3012	Cardinality:	Multi
3013	Updatability:	Read Write

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

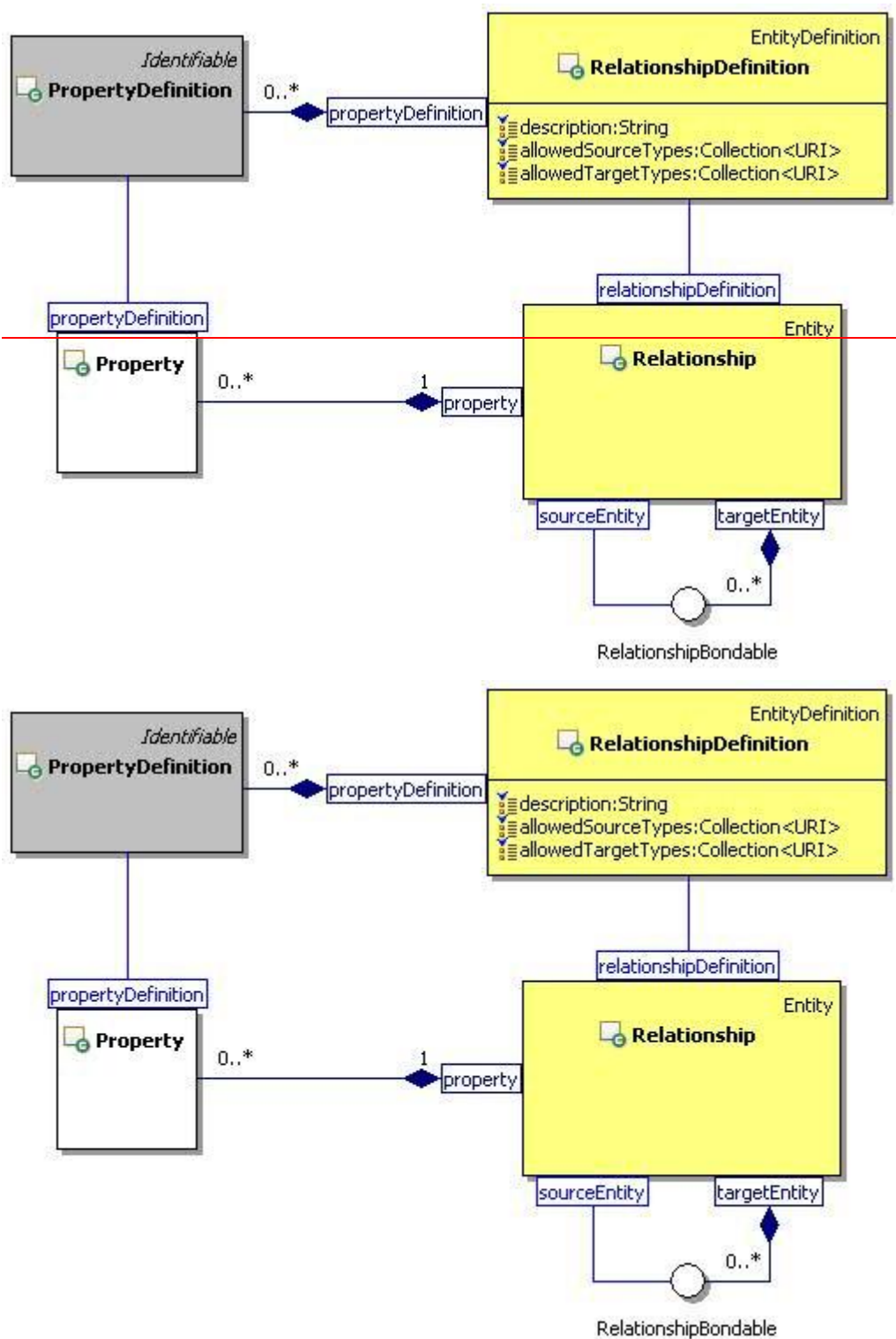


Figure 23: Relationship Class Diagram.

3.7 Common Concepts

3.7.1 Addressable

3.7.1.1 Description

An addressable object is an identifiable object ~~which~~that has ~~email and other~~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 ~~email and other~~one or more addresses.

The Addressable class ~~is defined by the~~has attribute values:

localNamespace

Value: icom_core

localName

Value: Addressable

extendsFrom

Value: icom_core:Identifiable

stereotype

Value: mixin

description

Value: Addressable is a mixin class which defines the characteristics of entities that has ~~email and other~~one or more addresses.

propertyDefinitions

The values for this attribute are defined in Section 3.7.1.3.

3.7.1.3 Property Definitions

The Addressable class inherits property definitions from super classes.

The Addressable class MUST have the property definitions:

icom_core:address

Description: Zero or more addresses of an addressable object.

Required: False

Inherited: False

Property Type: icom_core:EntityAddress

Cardinality: Multi

Updatability: Read Write

icom_core:primaryAddress

3061	Description:	The primary address of an addressable object.
3062	Required:	False
3063	Inherited:	False
3064	Property Type:	icom_core:EntityAddress
3065	Cardinality:	Single
3066	Updatability:	Read Write
3067		
3068	The Addressable class MAY include additional property definitions which are implementation-defined.	
3069		

3070 3.7.2 EntityAddress

3071 3.7.2.1 Description

3072 | An entity address object represents an address which is defined by type and ~~U~~RI.

3073 3.7.2.2 Class Definition

3074 | The EntityAddress class ~~is defined by the~~has attribute values:

3075		
3076	localNamespace	
3077	Value:	icom_core
3078		
3079	localName	
3080	Value:	EntityAddress
3081		
3082	extendsFrom	
3083	Value:	
3084		
3085	stereotype	
3086	Value:	primary
3087		
3088	description	
3089	Value:	An entity address object represents an address which is defined by type and U RI.
3090		
3091	propertyDefinitions	
3092	The values for this attribute are defined in Section 3.7.2.3.	

3093 3.7.2.3 Property Definitions

3094 The EntityAddress class MUST have the property definitions:

3095		
3096	icom_core:addressType	
3097	Description:	Type of an address.
3098	Required:	False
3099	Inherited:	False

3100	Property Type:	String
3101	Cardinality:	Single
3102	Updatability:	Read Write
3103		
3104	icom_core:address	
3105	Description:	A U RI representing an address.
3106	Required:	False
3107	Inherited:	False
3108	Property Type:	U RI
3109	Cardinality:	Single
3110	Updatability:	Read Write
3111		

3112 3.7.3 Participant

3113 3.7.3.1 Description

3114 A participant object represents the participation of any addressable entity in a collaboration activity such
3115 as an occurrence, task, conference, discussion, and message.

3116 If an addressable entity is not specified, an address must be specified.

3117 3.7.3.2 Class Definition

3118 The Participant class ~~is defined by the~~has attribute values:

3119	
3120	localNamespace
3121	Value: icom_core
3122	
3123	localName
3124	Value: Participant
3125	
3126	extendsFrom
3127	Value:
3128	
3129	stereotype
3130	Value: primary
3131	
3132	description
3133	Value: A participant object represents the participation of any addressable entity in a
3134	collaboration activity such as an occurrence, task, conference, discussion, and message.
3135	
3136	propertyDefinitions
3137	The values for this attribute are defined in Section 3.7.3.3.

3138 3.7.3.3 Property Definitions

3139 The Participant class inherits property definitions from super classes.

3140 The Participant class MUST have the property definitions:

3141		
3142	icom_core:participant	
3143	Description:	An addressable entity to participate in a collaboration activity.
3144	Required:	False
3145	Inherited:	False
3146	Property Type:	icom_core:Addressable
3147	Cardinality:	Single
3148	Updatability:	On Create

3149		
3150	icom_core:address	
3151	Description:	An address of a participant in a collaboration activity.
3152	Required:	False
3153	Inherited:	False
3154	Property Type:	URI
3155	Cardinality:	Single
3156	Updatability:	On Create

3157		
3158	icom_core:name	
3159	Description:	Name of a participant in a collaboration activity.
3160	Required:	False
3161	Inherited:	False
3162	Property Type:	String
3163	Cardinality:	Single
3164	Updatability:	On Create

3165

3166 The Participant class MAY include additional property definitions which are implementation-defined.

3167

3168 3.7.4 Priority

3169 3.7.4.1 Description

3170 A priority level for delivery of information.

3171 3.7.4.2 Class Definition

3172 The Priority class is an enuma mixin class ~~that enumerates the instances each of which expresses a~~
3173 ~~precedence ordering~~defines a priority level for delivery of information.

3174 The Priority ~~is defined by the~~class has attribute values:

3175		
3176	localNamespace	
3177	Value:	icom_core
3178		
3179	localName	

3180 Value: Priority

3181

3182 **extendsFrom**

3183 Value:

3184

3185 **stereotype**

3186 Value: mixin

3187

3188 **description**

3189 Value: Priority is a mixin class which defines a priority level for delivery of information.

3190

3191 **propertyDefinitions**

3192 The values for this attribute are defined in Section 3.7.4.3.

3193 **3.7.4.3 Property Definitions**

3194 The Priority class MAY include additional property definitions which are implementation-defined.

3195

3196 **3.7.5 PriorityEnum**

3197 The PriorityEnum class is an enum class that enumerates the instances each of which defines a priority

3198 level for delivery of information.

3199 The PriorityEnum has attribute values:

3200

3201 **localNamespace**

3202 Value: icom_core

3203

3204 **localName**

3205 Value: PriorityEnum

3206

3207 **extendsFrom**

3208 Value: Priority

3209

3210 **stereotype**

3211 Value: primary

3212

3213 **isEnumeration**

3214 Value: TRUE

3215

3216 **description**

3217 Value: ~~An enumeration~~ Priority level for delivery of the information.

3218

3219 **instances** ~~each of which expresses a precedence ordering.~~

3220

3221 ~~instances~~
3222 Value: <icom_core:~~None~~Normal, icom_core:Low, icom_core:Medium, icom_core:High>
3223

3224 The following ICOM defines four priorities ~~are defined by ICOM~~:

- 3225
 - icom_core:~~None to express~~Normal a normal priority.
 - icom_core:Low ~~to express~~ a low priority.
 - icom_core:Medium ~~to express~~ a medium priority.
 - icom_core:High ~~to express~~ a high priority.
- 3226
3227
3228
3229

3230 ~~3.7.5~~3.7.6 DateTimeResolution

3231 3.7.6.1 Description

3232 A date time resolution is a resolution of date time value.

3233 3.7.6.2 Class Definition

3234 The DateTimeResolution class is a mixin class which defines a resolution of date time value.

3235 The DateTimeResolution class has attribute values:

3236 localNamespace

3237 Value: icom_core

3238 localName

3239 Value: DateTimeResolution

3240 extendsFrom

3241 Value:

3242 stereotype

3243 Value: mixin

3244 description

3245 Value: DateTimeResolution is a mixin class which defines a resolution of date time value.

3246 propertyDefinitions

3247 The values for this attribute are defined in Section 3.7.6.3.

3254 3.7.6.3 Property Definitions

3255 The DateTimeResolution class MAY include additional property definitions which are implementation-
3256 defined.

3257

3.7.7 DateTimeResolutionEnum

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

The ~~DateTimeResolution~~ is defined by the DateTimeResolutionEnum has attribute values:

localNamespace

Value: icom_core

localName

Value: DateTimeResolutionEnum

extendsFrom

Value: DateTimeResolution

~~extendsFrom~~

Value:

stereotype

Value: primary

isEnumeration

Value: TRUE

description

Value: ~~An enumeration of instances each of which expresses a resolution~~Resolution of a date time value.

instances

Value: <icom_core:Year, icom_core:Date, icom_core:Time>

The following ICOM defines three date time resolutions ~~are defined by ICOM~~:

- **icom_core:Year** ~~to express~~ date time resolution is in years.
- **icom_core:Date** ~~to express~~ date time resolution is in years and days.
- **icom_core:Time** ~~to express~~ date time resolution is in years, days, and time of day.

3.7.8 TimeZone

3.7.8.1 Description

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

3.7.8.2 Class Definition

The TimeZone class has attribute values:

3298

3299 localNamespace

3300 Value: icom_core

3301

3302 localName

3303 Value: TimeZone

3304

3305 extendsFrom

3306 Value:

3307

3308 stereotype

3309 Value: primary

3310

3311 description

3312 Value: A time zone is a region that has a uniform standard time.

3313

3314 propertyDefinitions

3315 The values for this attribute are defined in Section 3.7.8.3.

3316 **3.7.8.3 Property Definitions**

3317 The TimeZone class inherits property definitions from super classes.

3318 The TimeZone class MUST have the property definitions:

3319

3320 icom_core:ID

3321	<u>Description:</u>	<u>Identifier of a time zone.</u>
3322	<u>Required:</u>	<u>False</u>
3323	<u>Inherited:</u>	<u>False</u>
3324	<u>Property Type:</u>	<u>String</u>
3325	<u>Cardinality:</u>	<u>Single</u>
3326	<u>Updatability:</u>	<u>On Create</u>

3327

3328 icom_core:rawOffset

3329	<u>Description:</u>	<u>An offset to add to Universal Coordinated Time (UTC) to get</u>
3330		<u>local time. If Daylight Saving Time is in effect at the specified</u>
3331		<u>date, the offset value is adjusted with the amount of daylight</u>
3332		<u>saving.</u>
3333	<u>Required:</u>	<u>False</u>
3334	<u>Inherited:</u>	<u>False</u>
3335	<u>Property Type:</u>	<u>Integer</u>
3336	<u>Cardinality:</u>	<u>Single</u>
3337	<u>Updatability:</u>	<u>On Create</u>

3338

3339 The TimeZone class MAY include additional property definitions which are implementation-defined.

3340		
3341	3.7.6 3.7.9	Location
3342	3.7.6-13.7.9.1	Description
3343	A location object represents a physical location which is defined by name, description, and geo	
3344	coordinates.	
3345	Note: The name of a location may remain unchanged while a physical location may be changing. For	
3346	example, a location name might be “On an airplane” while a physical location might be the geo	
3347	coordinates of a flight path or current coordinates of a plane.	
3348	3.7.6-23.7.9.2	Class Definition
3349	The Location class is defined by the <u>has</u> attribute values:	
3350		
3351	localNamespace	
3352	Value:	icom_core
3353		
3354	localName	
3355	Value:	Location
3356		
3357	extendsFrom	
3358	Value:	
3359		
3360	stereotype	
3361	Value:	primary
3362		
3363	description	
3364	Value:	A location object represents a physical location which is defined by name, description, or
3365		geo coordinates.
3366		
3367	propertyDefinitions	
3368	The values for this attribute are defined in Section 3.7.9.3.	
3369	3.7.6-33.7.9.3	Property Definitions
3370	The Location class MUST have the property definitions:	
3371		
3372	icom_core:name	
3373	Description:	Name of a location.
3374	Required:	False
3375	Inherited:	False
3376	Property Type:	String
3377	Cardinality:	Single
3378	Updatability:	Read Write
3379		

3380	icom_core:description	
3381	Description:	A description of a location.
3382	Required:	False
3383	Inherited:	False
3384	Property Type:	String
3385	Cardinality:	Single
3386	Updatability:	Read Write
3387		
3388	icom_core:timeZone	
3389	Description:	Time zone of a location.
3390	Required:	False
3391	Inherited:	False
3392	Property Type:	<u>icom_core:</u> TimeZone
3393	Cardinality:	Single
3394	Updatability:	Read Write
3395		
3396	icom_core:locationMark<u>coordinates</u>	
3397	Description:	A list of geo coordinates marking a point, path, or area of a physical location.
3398		
3399	Required:	False
3400	Inherited:	False
3401	Property Type:	icom_core:GeoCoordinates
3402	Cardinality:	Multi
3403	Updatability:	Read Write
3404		

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

~~3.7.7.3~~3.7.10 GeoCoordinates

~~3.7.7.13~~3.7.10.1 Description

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

~~3.7.7.23~~3.7.10.2 Class Definition

The GeoCoordinates class ~~is defined by the~~has attribute values:

localNamespace

Value: icom_core

localName

Value: GeoCoordinates

extendsFrom

3420 Value:

3421

3422 **stereotype**

3423 Value: primary

3424

3425 **description**

3426 Value: A geo coordinates object specifies the latitude, longitude, and altitude of a physical

3427 location.

3428

3429 **propertyDefinitions**

3430 The values for this attribute are defined in Section 3.7.10.3.

3431 **3.7.7.33.7.10.3 Property Definitions**

3432 The GeoCoordinates class MUST have the property definitions:

3433

3434 **icom_core:latitude**

3435	Description:	Latitude of coordinates <u>a location</u> .
3436	Required:	False
3437	Inherited:	False
3438	Property Type:	Float
3439	Cardinality:	Single
3440	Updatability:	Read Write

3442 **icom_core:longitude**

3443	Description:	Longitude of coordinates <u>a location</u> .
3444	Required:	False
3445	Inherited:	False
3446	Property Type:	Float
3447	Cardinality:	Single
3448	Updatability:	Read Write

3450 **icom_core:altitude**

3451	Description:	Altitude of coordinates <u>a location</u> .
3452	Required:	False
3453	Inherited:	False
3454	Property Type:	Float
3455	Cardinality:	Single
3456	Updatability:	Read Write

3458 The GeoCoordinates class MAY include additional property definitions which are implementation-defined.

3459

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 [aArtifact](#) and [fFolder](#) of Artifact Branch.

Note: ICOM Core Model [of \(Section 3\)](#) 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].

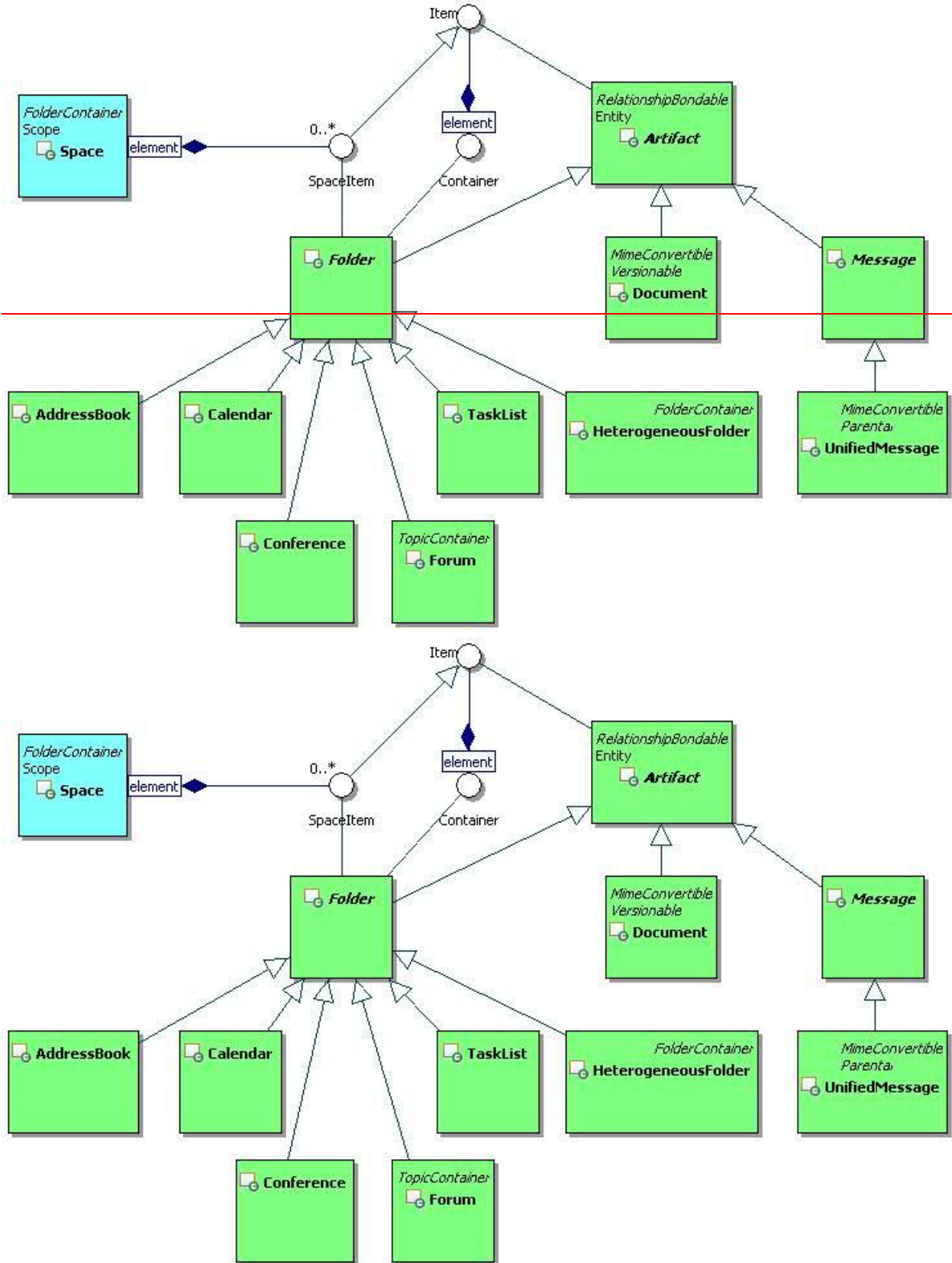


Figure 24 UML Diagram of Containers of Collaboration Activities.

~~Each area of collaboration~~ ICOM defines containers that provide contexts and structures for specific areas of collaborative activities ~~is contained by one or more specialized subclasses of folders.~~ The UML class diagram in Figure 24 depicts the containers of the collaboration activities defined in the 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,
- an inbox or outbox for communication, or
- a trash folder to archive all types of artifacts deleted from a space.

AddressBook is a specialized container to manage contact or personal information, such as addresses, phone numbers, birthdays, anniversaries, and other entries.

Calendar is a specialized container to support time management.

TaskList is a specialized container to support task coordination.

Forum is a specialized container to support

- **Topic** sub-containers for threaded discussions and
- **Announcement** sub-containers for time-sensitive communication.

Conference is a specialized container that provides a durable context for real-time interactions.

~~The following ten modules are specified as extension modules. The diagram shows Space as a container of SpaceItem, which includes different types of folder. A space serves as a hub of containers of different collaboration activities, of ICOM:~~

~~Note: HeterogeneousFolder is a general purpose folder that can serve as inbox, outbox, document or wiki page library, trash folder, etc. Document and UnifiedMessage are artifacts typically contained by the heterogeneous folders. Document and Message use the same composite content model.~~

- ~~1. Content Module (in section 4.2) defines Content, MultiContent, and SimpleContent are defined. A content represents a piece of data in a document or message. Content, multi-content, simple content, and online content form a composite design pattern.~~

- ~~• Document Module (in Section Content Module.~~

- ~~2. 4.3) defines Document, WikiPage, and version control model. A document can contain a composite content defined in section 4.2. Documents are defined typically contained by heterogeneous folders.~~

- ~~• Message Module (in Section Document Module.~~

- ~~3. 4.4) defines Message, UnifiedMessage, InstantMessage, and related classes are defined. A message can contain a composite content defined in section 4.2. Unified messages are typically contained by heterogeneous folders.~~

- ~~• Presence Module (in Section Message Module.~~

- ~~4. Note: 4.5) defines Presence, Activity, and Contact Method. Presence represents a watchable state of a Person. Presence is presentity (which is usually a person). Presence state is derived using an actor's subscriptions.~~

~~Note: Since a Presence is derived using a viewer's subscriptions, a Presence should not be shared with other viewers. For this reason, Presence is not modeled as Entity, hence a Presence instance cannot be and is not assigned an access control list.~~

- ~~• Presence, Activity, and Contact Method are defined in Address Book Module (in Section Presence Module.~~

- ~~1-5. Note: 4.6) defines AddressBook is a folder which contains a special type of artifact called and PersonContact. PersonContactA person contact can contain referencesbookmark a reference to~~

a person in ~~an ICOM~~ community. ~~It can also contain as well as store~~ addresses, phone numbers, and other entries about ~~an external~~ person. ~~who may not be in any ICOM community.~~

- ~~AddressBook and PersonContact are defined in Calendar Module (in Section Address Book Module.~~

~~2.6. Note: Calendar is a folder that contains time management artifacts such as 4.7) defines Calendar, Occurrence, and OccurrenceSeries. These Occurrence artifacts are used to resolve the free-busy times of participants for scheduling of meetings and booking of rooms and other resources.~~

- ~~Calendar, Occurrence, and OccurrenceSeries are defined Free Busy Module (in Section Calendar Module.~~

~~7. Note: 4.8) defines FreeBusy. FreeBusy is a stateview derived from occurrences in a calendar or a set of calendars. using an actor's privileges to determine the free or busy states of calendar occurrences.~~

~~Note: Since a FreeBusy view is derived using a viewer's privileges, a FreeBusy should not be shared with other viewers. For this reason, FreeBusy is not modeled as Entity, a FreeBusy instance cannot be and is not assigned an access control list.~~

- ~~Free Busy is defined Task List Module (in Section Free Busy Module.~~

~~3.8. Note: 4.9) defines TaskList is a folder that contains task management artifacts such as Task and Assignment. These artifacts Task. Tasks are used to coordinate the assignment of tasks and to track the progress of task activities.~~

- ~~TaskList, Task, and Assignment are defined Forum Module (in Section TaskList Module.~~

~~4.9. Note: 4.10) defines Forum is a folder that contains, Topic, Announcement, and DiscussionMessage. Topics, announcements, and DiscussionMessage. These artifacts discussions are used for treaded discussions. Moderators of thea forum can prune, merge, or fork the discussion threads.~~

- ~~Forum, Topic, Announcement, and DiscussionMessage are defined in Section Forum Module.~~

~~5.10. Note: Conference is a folder that provides a durable context for real time interactions. #Conference Module (in Section 4.11) defines Conference and related classes. A conference can contain visual, audio, and chat transcripts of the conference sessions. It also contains the current status, conference settings, past sessions, active session, and activity logs.~~

- ~~Conference and related classes are defined in Section Conference Module.~~

~~Note: In an integrated application, the repertoire of specialized types of folders can potentially grow to include more advanced collaboration activities, such as case management, decision support, simulation, command and control, business process monitoring, to name a few.~~

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.

~~Note: MIME is specified by RFC memoranda: RFC 2045, RFC 2046, RFC 2047, RFC 4288, RFC 4289 and RFC 2049.~~

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 ~~is defined by the~~has attribute values:

3569
3570 **localNamespace**
3571 Value: icom_content
3572
3573 **localName**
3574 Value: MimeConvertible
3575
3576 **extendsFrom**
3577 Value: icom_core:Identifiable
3578
3579 **stereotype**
3580 Value: mixin
3581
3582 **description**
3583 Value: MimeConvertible class is a mixin class that defines the characteristics of objects that can
3584 be represented in MIME format.
3585
3586 **propertyDefinitions**
3587 The values for this attribute are defined in Section 4.2.1.3.

3588 **4.2.1.3 Property Definitions**

3589 The MimeConvertible class inherits property definitions from super classes.
3590 The MimeConvertible class MAY include additional property definitions which are implementation-defined.
3591

3592 **4.2.2 Content**

3593 **4.2.2.1 Description**

3594 A content object represents a piece of data in a document or message. Content, multi-content, simple
3595 content, and online content form a composite design pattern.

3596 **4.2.2.2 Class Definition**

3597 The Content class ~~is defined by the~~has attribute values:

3598
3599 **localNamespace**
3600 Value: icom_content
3601
3602 **localName**
3603 Value: Content
3604
3605 **extendsFrom**
3606 Value: icom_core:Identifiable, icom_content:MimeConvertible
3607
3608 **stereotype**

3609 Value: primary
 3610
 3611 **isAbstract**
 3612 Value: TRUE
 3613
 3614 **description**
 3615 Value: Content represents a piece of data in a document or message.
 3616
 3617 **propertyDefinitions**
 3618 The values for this attribute are defined in Section 4.2.2.3.

3619 4.2.2.3 Property Definitions

3620 The Content class inherits property definitions from super classes.
 3621 The Content class MUST have the property definitions:

3623 **icom_content:contentId**

3624	Description:	A content id is a unique identifier for a part of content in multi-
3625		part contents.
3626	Required:	False
3627	Inherited:	False
3628	Property Type:	String
3629	Cardinality:	Single
3630	Updatability:	Read Write

3632 **icom_content:mediaType**

3633		Description:	Media type is a two-part identifier for Internet file formats as defined in RFC 2046 and additional RFCs including RFC 3236, RFC 1847, etc.
3634			
3635			
3636		Required:	False
3637		Inherited:	False
3638		Property Type:	String
3639		Cardinality:	Single
3640		Updatability:	Read Write

3642 **icom_content:contentDisposition**

3643		Description:	Content disposition is defined in RFC 2183 to specify <u>specifies</u>
3644			a presentation style.
3645		Required:	False
3646		Inherited:	False
3647		Property Type:	ContentDispositionType
3648		Cardinality:	Single
3649		Updatability:	Read Write

3651 The Content class MAY include additional property definitions which are implementation-defined.



Figure 25: Composite Content Class Diagram.

4.2.3 MultiContent

4.2.3.1 Description

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

~~Note: A media type is an official RFC 2046 type.~~

4.2.3.2 Class Definition

The MultiContent class ~~is defined by the~~has attribute values:

localNamespace

Value: icom_content

localName

Value: MultiContent

extendsFrom

Value: icom_content:Content

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.2.3.3.

4.2.3.3 Property Definitions

The MultiContent class inherits property definitions from super classes.

The MultiContent class MUST have the property definitions:

icom_content:part

Description: Zero or more parts of a hierarchical composite content.

Required: False

Inherited: False

Property Type: icom_content:MimeConvertible

Cardinality: Multi

Updatability: Read Write

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

3695

3696 **4.2.4 SimpleContent**

3697 **4.2.4.1 Description**

3698 A simple content holds a single piece of data.

3699 ~~Note: A media type is an official RFC 2046 type. Content encoding specifies RFC 2616 content encoding~~
3700 ~~applied to a content. Character encoding specifies RFC 2616 character set of a content (a missing value~~
3701 ~~means that a content should be treated as binary or raw). Content language specifies RFC 2616 content~~
3702 ~~language for a content (a missing value means non-natural language content).~~

3703 ~~**4.2.4.2.1.1.1 Class Definition**~~

3704 **4.2.4.2 Class Definition**

3705 The SimpleContent class ~~is defined by the~~has attribute values:

3706

3707 **localNamespace**

3708 Value: icom_content

3709

3710 **localName**

3711 Value: SimpleContent

3712

3713 **extendsFrom**

3714 Value: icom_content:Content

3715

3716 **stereotype**

3717 Value: primary

3718

3719 **description**

3720 Value: A simple content holds a single piece of data.

3721

3722 **propertyDefinitions**

3723 The values for this attribute are defined in Section 4.2.4.3.

3724 **4.2.4.3 Property Definitions**

3725 The SimpleContent class inherits property definitions from super classes.

3726 The SimpleContent class MUST have the property definitions:

3727

3728 **icom_content:characterEncoding**

3729 Description: Character encoding specifies ~~RFC 2616~~ character set of a
3730 content (a missing value means that a piece of content should
3731 be treated as binary or raw).

3732 Required: False

3733 Inherited: False

3734 Property Type: String

3735	Cardinality:	Single
3736	Updatability:	Read Write
3737		
3738	icom_content:contentEncoding	
3739	Description:	Content encoding specifies RFC-2616 content encoding
3740		applied to a <u>piece of</u> content.
3741	Required:	False
3742	Inherited:	False
3743	Property Type:	String
3744	Cardinality:	Single
3745	Updatability:	Read Write
3746		
3747	icom_content:contentLanguage	
3748	Description:	Content language specifies RFC-2616 content language for a
3749		<u>piece of</u> content (a missing value means non-natural
3750		language content).
3751	Required:	False
3752	Inherited:	False
3753	Property Type:	Locale
3754	Cardinality:	Single
3755	Updatability:	Read Write
3756		
3757	icom_content:contentLength	
3758	Description:	Length of a <u>piece of</u> content.
3759	Required:	False
3760	Inherited:	False
3761	Property Type:	Integer
3762	Cardinality:	Single
3763	Updatability:	Read Write
3764		
3765	icom_content:contentBody	
3766	Description:	Body of a simple content.
3767	Required:	False
3768	Inherited:	False
3769	Property Type:	Object
3770	Cardinality:	Single
3771	Updatability:	Read Write
3772		
3773	The SimpleContent class MAY include additional property definitions which are implementation-defined.	
3774		

4.2.5 OnlineContent

4.2.5.1 Description

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

Note: An online artifact must be rendered as [a URL an IRI](#) when a message or invitation is delivered to external recipients.

4.2.5.2 Class Definition

~~4.2.5.2.1.1.1 Class Definition~~

The OnlineContent class ~~is defined by the~~[has](#) attribute values:

localNamespace

Value: icom_content

localName

Value: OnlineContent

extendsFrom

Value: icom_content:Content

stereotype

Value: primary

description

Value: An online content holds an online artifact attached to a message or invitation.

propertyDefinitions

The values for this attribute are defined in Section 4.2.5.3.

4.2.5.3 Property Definitions

The OnlineContent class inherits property definitions from super classes.

The OnlineContent class MUST have the property definition:

icom_content:onlineAttachment

Description: An online artifact attached to a message.

Required: True

Inherited: False

Property Type: [icom_core:Artifact](#)

Cardinality: Single

Updatability: Read Write

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

4.2.6 ContentDispositionType

4.2.6.1 Description

A content disposition type is a presentation style of content.

4.2.6.2 Class Definition

The ContentDispositionType class is ~~an enum~~ a mixin class ~~that enumerates the instances each of which expresses~~ defines a presentation style of content.

~~Note: The enumerated instances for this class are content disposition types defined in RFC 2183.~~

The ContentDispositionType class ~~is defined by the~~ has attribute values:

localNamespace

Value: icom_content

localName

Value: ContentDispositionType

extendsFrom

Value:

stereotype

Value: mixin

description

Value: ContentDispositionType is a mixin class which defines a presentation style of content.

propertyDefinitions

The values for this attribute are defined in Section 4.2.6.3.

4.2.6.3 Property Definitions

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

4.2.7 ContentDispositionTypeEnum

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

The ContentDispositionTypeEnum class has attribute values:

localNamespace

Value: icom_content

localName

Value: ContentDispositionTypeEnum

3855
 3856 **extendsFrom**
 3857 Value: ContentDispositionType
 3858
 3859 **stereotype**
 3860 Value: primary
 3861
 3862 **isEnumeration**
 3863 Value: TRUE
 3864
 3865 **description**
 3866 Value: ~~An enumeration of instances each of which expresses a~~ presentation style of content
 3867 ~~defined in RFC 2183.~~
 3868
 3869 **instances**
 3870 Value: <icom_content:Inline, icom_content:Attachment>
 3871
 3872 ~~There are~~ICOM defines two content disposition types ~~defined by ICOM:~~
 3873 • **icom_content:Inline** ~~to express that~~ content is to be displayed automatically upon display of the
 3874 main body of an artifact.
 3875 • **icom_content:Attachment** ~~to express that~~ content is separate from the main body of an artifact,
 3876 and that its display should not be automatic, but contingent upon some further action of a user.
 3877

3878 4.2.74.2.8 Attachment

3879 ~~4.2.7.14~~4.2.8.1 Description

3880 An attachment holds a ~~simple~~ content for an occurrence, task, and contact artifact.

3881 ~~4.2.7.24~~4.2.8.2 Class Definition

3882 The Attachment class ~~is defined by the~~has attribute values:

3883
 3884 **localNamespace**
 3885 Value: icom_content
 3886
 3887 **localName**
 3888 Value: Attachment
 3889
 3890 **extendsFrom**
 3891 Value:
 3892
 3893 **stereotype**
 3894 Value: primary
 3895

3896 **description**
3897 | Value: An attachment holds a ~~simple~~ content for an occurrence, task, and contact artifact.
3898

3899 **propertyDefinitions**

3900 The values for this attribute are defined in Section 4.2.8.3.

3901 ~~4.2.7.34.2.8.3~~ **4.2.8.3 Property Definitions**

3902 The Attachment class MUST have the property definitions:
3903

3904 **icom_core:name**

3905 Description:	Name of a simple content attachment.
3906 Required:	True
3907 Inherited:	False
3908 Property Type:	String
3909 Cardinality:	Single
3910 Updatability:	Read Write

3912 **icom_content:content**

3913 Description:	A simple content attached to an occurrence, task, or contact artifact.
3914	
3915 Required:	True
3916 Inherited:	False
3917 Property Type:	SimpleContent <u>icom_content:Content</u>
3918 Cardinality:	Single
3919 Updatability:	Read Write

3920
3921 The Attachment class MAY include additional property definitions which are implementation-defined.
3922

3923 **4.3 Document Module**

3924 **4.3.1 Versionable**

3925 **4.3.1.1 Description**

3926 A versionable artifact is

- 3927 1. a non-version-controlled copy,
- 3928 2. a specific versioned copy,
- 3929 3. a private working copy, or
- 3930 4. a representative copy (optional)

3931 of an artifact version series.

3932 | When a versionable artifact is not under version control, a non-version-controlled copy ~~of the versionable~~
3933 ~~artifact~~ MUST be the only copy in a version series, i.e. there is only one copy and one *objectId*.

3934 | When a versionable artifact is under version control, a representative copy MAY provide a version-
3935 independent view of a versionable artifact.

3936 | When a non-version-controlled copy ~~of a versionable artifact~~ is placed under version control, a versioned
3937 | copy MUST be created. Assignment of an object identifier to a versioned copy is implementation-
3938 | dependent:

- 3939 | • if a versioned copy retains the object identifier of a non-version-controlled copy, the version type
3940 | of a versionable artifact MUST change from NonVersionControlledCopy to VersionedCopy;
- 3941 | • if a versioned copy is assigned a new object identifier that is different from the object identifier of
3942 | a non-version-controlled copy, a representative copy MAY retain the object identifier of the non-
3943 | version-controlled copy;
- 3944 | • if both versioned copy and representative copy are assigned new object identifiers that are
3945 | different from the object identifier of a non-version-controlled copy, the non-version-controlled
3946 | copy SHALL be discarded.

3947 | When a private working copy ~~of a versionable artifact~~ is checked in, a versioned copy MUST be created.
3948 | Assignment of an object identifier to a versioned copy is implementation-dependent:

- 3949 | • if a versioned copy retains the object identifier of a private working copy, the version type of a
3950 | versionable artifact MUST change from PrivateWorkingCopy to VersionedCopy;
- 3951 | • if a versioned copy is assigned a new object identifier that is different from the object identifier of
3952 | a private working copy, the private working copy SHALL be discarded.

3953 | It is optional for a service provider to provide a representative copy for a version series. If a representative
3954 | copy ~~of a versionable artifact~~ is provided:

- 3955 | • a representative copy MUST have its own object identifier that is different from the object
3956 | identifier of any versioned copy or private working copy ~~of a versionable artifact~~;
- 3957 | • assignment of an object identifier to a representative copy ~~of a versionable artifact~~ is
3958 | implementation-dependent:
 - 3959 | ○ a representative copy MAY retain the object identifier of a non-version-controlled copy ~~of~~
3960 | ~~a versionable artifact~~; if so the version type of a versionable artifact MUST change from
3961 | NonVersionControlledCopy to RepresentativeCopy;
 - 3962 | ○ a representative copy MAY be assigned a new object identifier that is different from the
3963 | object identifier of a non-version-controlled copy ~~of a versionable artifact~~;
 - 3964 | • content and state of a representative copy ~~of a versionable artifact in a version series~~ is
3965 | implementation-dependent:
 - 3966 | ○ a representative copy MAY be a copy of the content and state of the latest versioned
3967 | copy or the latest major versioned copy in a version series;
 - 3968 | ○ a representative copy MAY be a copy of the content and state of a private working copy if
3969 | the current user loading the representative copy is the same user who checks out a
3970 | version series.

3971 | ~~Note: A specific versioned copy of a versionable artifact is an explicit "deep" copy of the content and state~~
3972 | ~~of a versionable artifact, preserving its content and state at a certain point in time.~~ ~~Note:~~ Each versioned
3973 | copy of a versionable artifact is itself a versionable artifact, i.e. it has its own *objectId*. A versioned copy
3974 | has a version number, label, and check in comment.

3975 | Note: A private working copy ~~of a versionable artifact~~ is a versionable artifact created by an explicit
3976 | checkout operation on a versionable artifact under version control. The properties for a private working
3977 | copy ~~should be~~ identical to the properties of a versioned copy ~~of a versionable artifact~~ on which a
3978 | checkout operation was performed. Certain properties such as *objectId* and *creationDate* ~~shall be~~
3979 | different from a versioned copy. The content of a private working copy ~~may be~~ identical to the content of
3980 | a versioned copy. Its object identifier ~~must be~~ different from that of the representative copy or any
3981 | versioned copy.

3982 | ~~Note:~~ A private working copy ~~may~~ MAY be saved in a version series for sharing and co-editing, however, it
3983 | needs not be visible to users who may only have permissions to view other versioned copies in a version
3984 | series.

3985 | **Note:** Until it is checked in using an explicit ~~checkin~~check-in operation, a private working copy must not be
 3986 considered the LatestMajorVersion in a version series.

3987 | **Note:** A container of a versionable artifact ~~can~~CAN contain a representative copy ~~of a version series~~ so
 3988 that it provides a version-independent view of a state of the version series.

3989 | **Note:** Starting from a representative copy in a container, an actor can traverse a version series to retrieve
 3990 any versioned copy or private working copy.

3991 ICOM version control model is based on the CMIS version control model specified in Section 2.1.9 of
 3992 Content Management Interoperability Services Version 1.0 [CMIS].

3993 4.3.1.2 Class Definition

3994 The Versionable class is a mixin class that defines the characteristics of artifacts that can be versioned.
 3995 | The Versionable class ~~is defined by the~~has attribute values:

3996

3997 **localNamespace**
 3998 Value: icom_doc

3999

4000 **localName**
 4001 Value: Versionable

4002

4003 **extendsFrom**
 4004 Value: icom_core:Identifiable

4005

4006 **stereotype**
 4007 Value: mixin

4008

4009 **description**
 4010 Value: Versionable class is a mixin class that defines the characteristics of artifacts that can be
 4011 versioned.

4012

4013 **propertyDefinitions**
 4014 The values for this attribute are defined in Section 4.3.1.3.

4015 4.3.1.3 Property Definitions

4016 The Versionable class inherits property definitions from super classes.
 4017 The Versionable class MUST have the property definitions:

4018

4019 **icom_doc:versionControlMetadata**

4020 Description:	A version control metadata object attached to a versionable artifact.
4021	
4022 Required:	False
4023 Inherited:	False
4024 Property Type:	icom_doc:VersionControlMetadata
4025 Cardinality:	Single
4026 Updatability:	Read Only

4027		
4028	icom_doc:versionType	
4029	Description:	A type of version controlled copy of a versionable artifact.
4030	Required:	False
4031	Inherited:	False
4032	Property Type:	icom_doc:VersionType
4033	Cardinality:	Single
4034	Updatability:	Read Only
4035		

4036 The Versionable class MAY include additional property definitions which are implementation-defined.

4037

4038 4.3.2 VersionControlMetadata

4039 4.3.2.1 Description

4040 A version control metadata is an object that contains version control information.

4041 There are two classes of version control metadata: version series and version. A version control metadata
4042 of a versionable artifact is either a version series or a version depending on the version type.

- 4043 • If the version type is icom_doc:NonVersionControlledCopy then metadata is optional; if metadata
4044 is present, it MUST be a version series object.
- 4045 • If the version type is icom_doc:RepresentativeCopy, then metadata MUST be a version series
4046 object.
- 4047 • If the version type is icom_doc:VersionedCopy or icom_doc:PrivateWorkingCopy, then metadata
4048 MUST be a version object.

4049 4.3.2.2 Class Definition

4050 The VersionControlMetadata class is a mixin class that defines the characteristics of ~~entities that serve~~
4051 ~~as version or version series~~ metadata for version control.

4052 The VersionControlMetadata class ~~is defined by the~~has attribute values:

4053

4054 **localNamespace**

4055 Value: icom_doc

4056

4057 **localName**

4058 Value: VersionControlMetadata

4059

4060 **extendsFrom**

4061 Value: icom_core:Identifiable

4062

4063 **stereotype**

4064 Value: mixin

4065

4066 **description**

4067 Value: VersionControlMetadata is a mixin class that defines the characteristics of entities that
4068 serve as metadata for version control.

4069
4070 **propertyDefinitions**
4071 The values for this attribute are defined in Section 4.3.2.3.

4072 **4.3.2.3 Property Definitions**

4073 The VersionControlMetadata class inherits property definitions from super classes.
4074 The VersionControlMetadata class MUST have the property definition:

4075
4076 **icom_doc:representativeCopy**
4077 Description: A representative copy of a versionable artifact.
4078 Required: False
4079 Inherited: False
4080 Property Type: icom_doc:Versionable
4081 Cardinality: Single
4082 Updatability: Read Only

4083
4084 The VersionControlMetadata class MAY include additional property definitions which are implementation-
4085 defined.
4086

4087 **4.3.3 VersionSeries**

4088 **4.3.3.1 Description**

4089 A version series is a version control metadata that contains a version history and check in/out states of a
4090 versionable artifact.
4091 A version series object is a version control metadata of a representative copy of a versionable artifact.

4092 **4.3.3.2 Class Definition**

4093 The VersionSeries class ~~is defined by the~~has attribute values:

4094
4095 **localNamespace**
4096 Value: icom_doc
4097
4098 **localName**
4099 Value: VersionSeries
4100
4101 **extendsFrom**
4102 Value: icom_core:Entity, icom_doc:VersionControlMetadata, icom_meta:RelationshipBondable
4103
4104 **stereotype**
4105 Value: primary
4106
4107 **description**

4108 | Value: A version series is ~~a~~ version control metadata that contains a version history and check
4109 in/out states of a versionable artifact.

4110

4111 **propertyDefinitions**

4112 The values for this attribute are defined in Section 4.3.3.3.

4113 **4.3.3.3 Property Definitions**

4114 The VersionSeries class inherits property definitions from super classes.

4115 The VersionSeries class MUST have the property definitions:

4116

4117 **icom_doc:versionHistory**

4118 Description: A history of version nodes of a versionable artifact.

4119 Required: False

4120 Inherited: False

4121 Property Type: icom_doc:Version

4122 Cardinality: Multi

4123 Updatability: Read Only

4124

4125 **icom_doc:versionableHistory**

4126 Description: A history of the versioned copies of a versionable artifact.

4127 Required: False

4128 Inherited: False

4129 Property Type: icom_doc:Versionable

4130 Cardinality: Multi

4131 Updatability: Read Only

4132

4133 **icom_doc:latestVersionedCopy**

4134 Description: Latest versioned copy of a versionable artifact.

4135 Required: False

4136 Inherited: False

4137 Property Type: icom_doc:Versionable

4138 Cardinality: Single

4139 Updatability: Read Only

4140

4141 **icom_doc:privateWorkingCopy**

4142 Description: A private working copy of a versionable artifact.

4143 Required: False

4144 Inherited: False

4145 Property Type: icom_doc:Versionable

4146 Cardinality: Single

4147 Updatability: Read Only

4148

4149 **icom_doc:versionSeriesCheckedOut**

4150	Description:	Indicates whether a version series is checked out.
4151	Required:	False
4152	Inherited:	False
4153	Property Type:	Boolean
4154	Cardinality:	Single
4155	Updatability:	Read Only
4156		
4157	icom_doc:versionSeriesCheckedOutBy	
4158	Description:	An actor who checks out a version series.
4159	Required:	False
4160	Inherited:	False
4161	Property Type:	icom_core:Actor
4162	Cardinality:	Single
4163	Updatability:	Read Only
4164		
4165	icom_doc:versionSeriesCheckedOutOn	
4166	Description:	The time when a version series is checked out.
4167	Required:	False
4168	Inherited:	False
4169	Property Type:	DateTime
4170	Cardinality:	Single
4171	Updatability:	Read Only
4172		
4173	icom_doc:versionSeriesCheckoutComment	
4174	Description:	A checked out comment of a version series.
4175	Required:	False
4176	Inherited:	False
4177	Property Type:	String
4178	Cardinality:	Single
4179	Updatability:	Read Only
4180		
4181	icom_doc:totalSize	
4182	Description:	Total size of all versioned copies of a versionable artifact in a version series.
4183		
4184	Required:	False
4185	Inherited:	False
4186	Property Type:	Integer
4187	Cardinality:	Single
4188	Updatability:	Read Only
4189		
4190	The VersionSeries class MAY include additional property definitions which are implementation-defined.	
4191		

4.3.4 Version

4.3.4.1 Description

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

A version object is a version control metadata of a versioned copy or a private working copy of a versionable artifact.

4.3.4.2 Class Definition

The Version class ~~is defined by the~~has attribute values:

localNamespace

Value: icom_doc

localName

Value: Version

extendsFrom

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

stereotype

Value: primary

description

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

propertyDefinitions

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.

4232	Required:	True
4233	Inherited:	False
4234	Property Type:	Integer
4235	Cardinality:	Single
4236	Updatability:	Read Write

4237

4238 **icom_doc:versionLabel**

4239	Description:	A version label of a versioned copy.
4240	Required:	True
4241	Inherited:	False
4242	Property Type:	String
4243	Cardinality:	Single
4244	Updatability:	Read Write

4245

4246 **icom_doc:majorVersion**

4247	Description:	Indicates whether a versioned copy is a major version.
4248	Required:	True
4249	Inherited:	False
4250	Property Type:	Boolean
4251	Cardinality:	Single
4252	Updatability:	Read Write

4253

4254 **icom_doc:versionedOrPrivateWorkingCopy**

4255	Description:	A versioned copy or private working copy corresponding to a version of a versionable artifact.
4256		
4257	Required:	False
4258	Inherited:	False
4259	Property Type:	icom_doc:Versionable
4260	Cardinality:	Multi Single
4261	Updatability:	Read Only

4262

4263 The Version class MAY include additional property definitions which are implementation-defined.

4264

4265 **4.3.5 VersionType**

4266 ~~The VersionType class is an enum class that enumerates the instances each of which expresses a~~

4267 ~~version type.~~

4268 **4.3.5.1 Description**

4269 ~~A version type is a version state of a copy of versionable document.~~

4270 **4.3.5.2 Class Definition**

4271 The VersionType class is ~~defined by the~~ a mixin class which defines a version state of a copy of

4272 versionable document.

4273 | The VersionType class has attribute values:

4274

4275 **localNamespace**

4276 Value: icom_doc

4277

4278 **localName**

4279 Value: VersionType

4280

4281 **extendsFrom**

4282 Value:

4283

4284 **stereotype**

4285 Value: mixin

4286

4287 **description**

4288 Value: VersionType is a mixin class which defines a version state of a copy of versionable
4289 document.

4290

4291 **propertyDefinitions**

4292 The values for this attribute are defined in Section 4.3.5.3.

4293 **4.3.5.3 Property Definitions**

4294 The VersionType class MAY include additional property definitions which are implementation-defined.

4295

4296 **4.3.6 VersionTypeEnum**

4297 The VersionTypeEnum class is an enum class that enumerates the instances each of which expresses a
4298 version type.

4299 The VersionTypeEnum class has attribute values:

4300

4301 **localNamespace**

4302 Value: icom_doc

4303

4304 **localName**

4305 Value: VersionTypeEnum

4306

4307 **extendsFrom**

4308 Value: VersionType

4309

4310 **stereotype**

4311 Value: primary

4312

4313 **isEnumeration**

4314 Value: TRUE
4315
4316 **description**
4317 Value: ~~An enumeration of the instances each of which expresses a~~ A version type of a copy of
4318 versionable document.

4319
4320 **instances**
4321 Value: <icom_doc:NonVersionControlledCopy, icom_doc:VersionedCopy,
4322 icom_doc:PrivateWorkingCopy, icom_doc:RepresentativeCopy>

4323
4324 ~~There are~~ ICOM defines four version types ~~defined by ICOM~~:
4325 • **icom_doc:NonVersionControlledCopy** ~~to express that~~ a versionable artifact is not under
4326 version control.
4327 • **icom_doc:VersionedCopy** ~~to express that~~ a versionable artifact is a version of an artifact
4328 version series.
4329 • **icom_doc:PrivateWorkingCopy** ~~to express that~~ a versionable artifact is a private working copy
4330 of an artifact version series.
4331 • **icom_doc:RepresentativeCopy** ~~to express that~~ a versionable artifact is a version-independent
4332 representative copy of an artifact. This version type is optional and implementation-dependent.

4333
4334 ~~4.3.6.4~~ 4.3.7 **Document**

4335 ~~4.3.6.14~~ 4.3.7.1 **Description**

4336 A document is a versionable artifact that can contain a single content of a media type or composite
4337 contents of ~~any~~ assortment of media types.

4338 ~~4.3.6.24~~ 4.3.7.2 **Class Definition**

4339 The Document class ~~is defined by the~~ has attribute values:

4340
4341 **localNamespace**
4342 Value: icom_doc
4343
4344 **localName**
4345 Value: Document
4346
4347 **extendsFrom**
4348 Value: icom_core:Artifact, icom_doc:Versionable, icom_content:MimeConvertible

4349
4350 **stereotype**
4351 Value: primary

4352
4353 **description**
4354 Value: A document is a versionable artifact that ~~can~~ may contain a single content of a media type
4355 or composite contents of ~~any~~ assortment of media types.

4356
4357 ~~propertyDefinitions~~

4358
4359 propertyDefinitions

4360 The values for this attribute are defined in Section 4.3.7.3.

4361 ~~4.3.6-34.3.7.3~~ **Property Definitions**

4362 The Document class inherits property definitions from super classes.

4363 The Document class MUST have the property definitions:

4364

4365 **icom_content:content**

4366 Description: Content of a document.

4367 Required: False

4368 Inherited: False

4369 Property Type: icom_content:Content

4370 Cardinality: Single

4371 Updatability: Read Write

4372

4373 **icom_doc:size**

4374 Description: The size of a copy of a document.

4375 Required: False

4376 Inherited: False

4377 Property Type: Integer

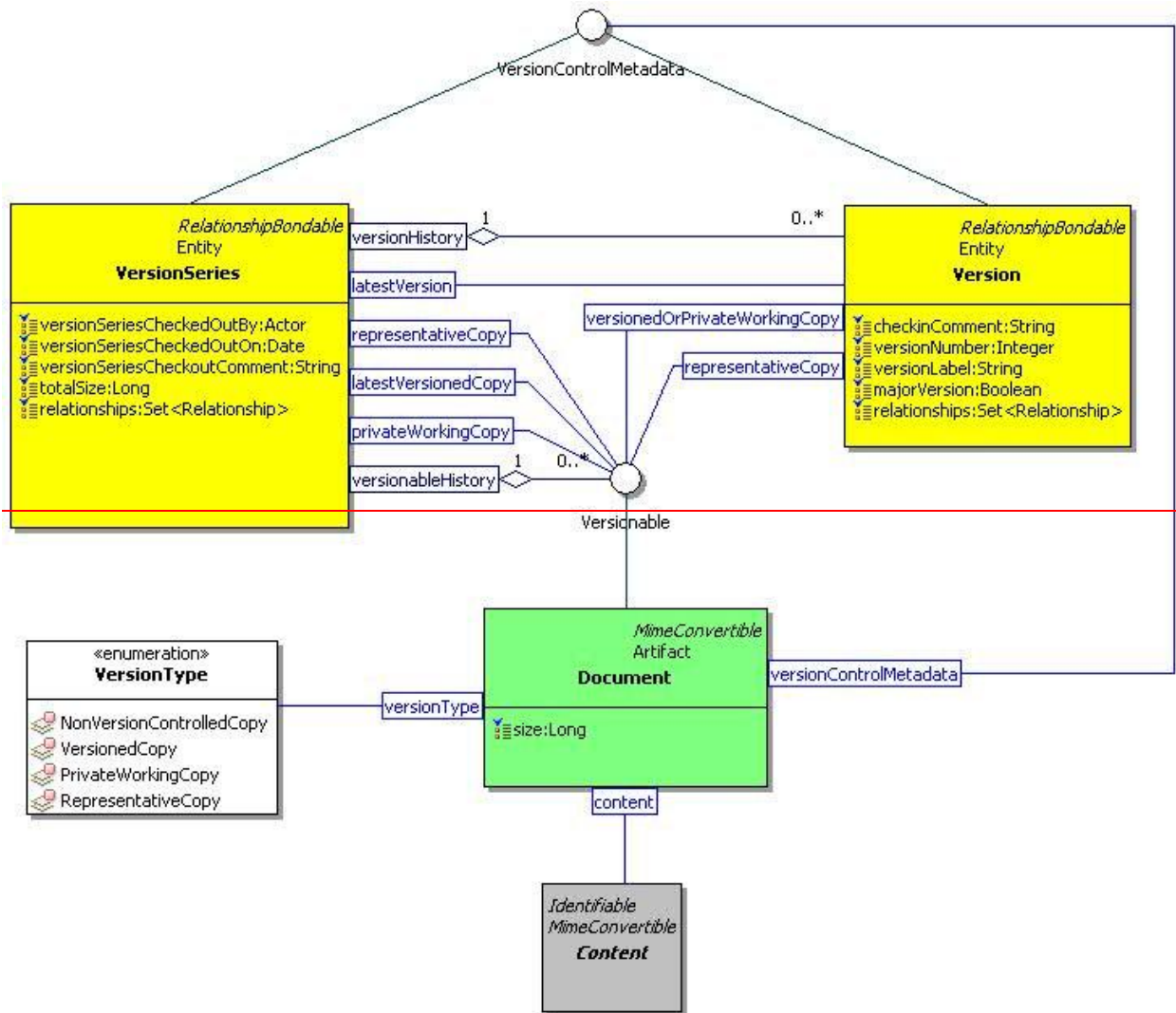
4378 Cardinality: Single

4379 Updatability: Read Only

4380

4381 The Document class MAY include additional property definitions which are implementation-defined.

4382



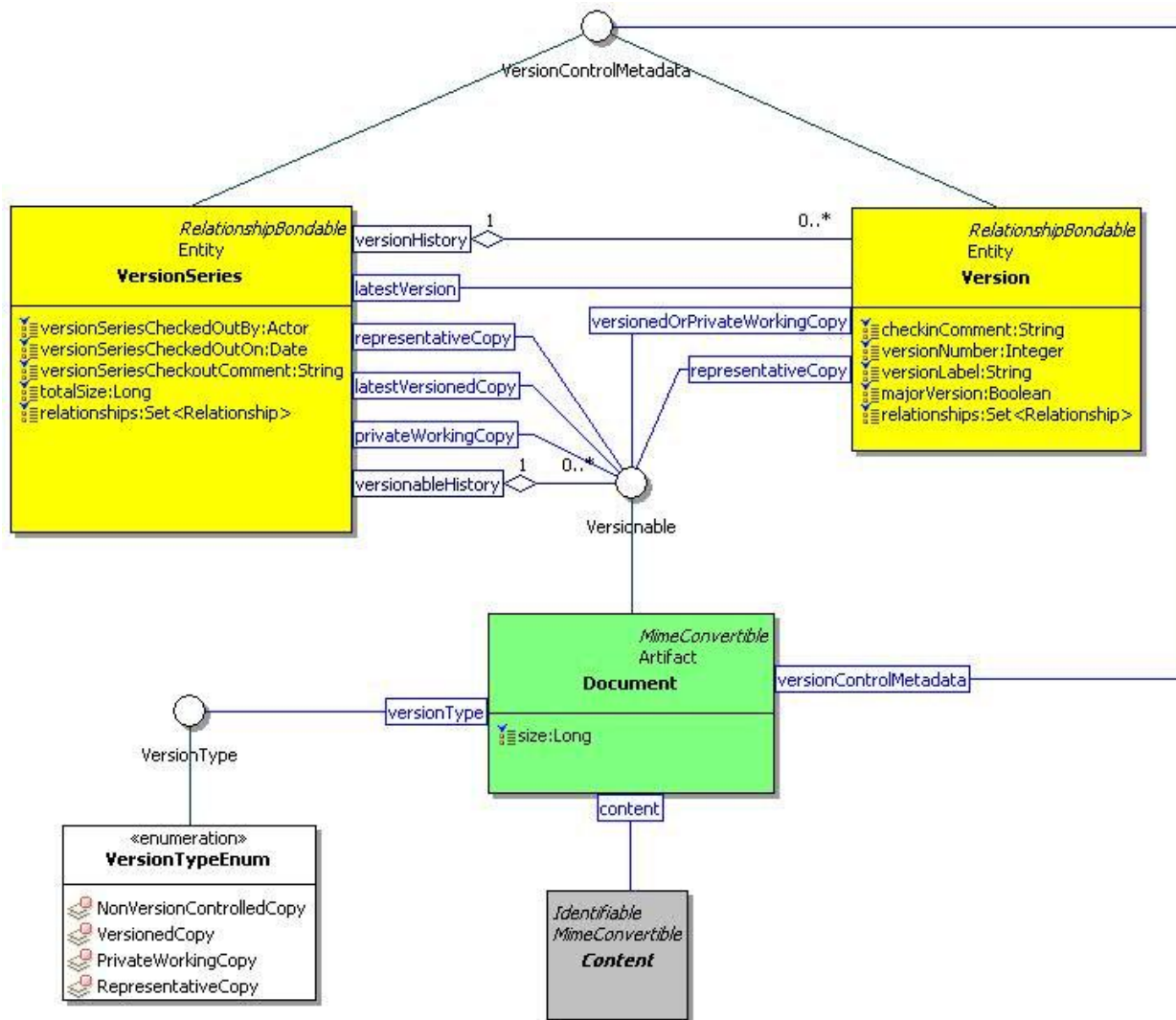


Figure 26: Document, Version Series, and Version Class Diagram.

4.3.7.14.3.8.1 WikiPage

4.3.7.14.3.8.1.1 Description

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

4.3.7.24.3.8.2 Class Definition

The WikiPage class is defined by the has attribute values:

localNamespace

Value: `icom_doc`

localName

Value: `WikiPage`

4399

4400 **extendsFrom**

4401 Value: icom_doc:Document

4402

4403 **stereotype**

4404 Value: primary

4405

4406 **description**

4407 Value: A wiki page is a document that contains a ~~rendered page and a rendered~~ wiki content.

4408 and that provides an html page generated from the wiki content.

4409

4410 **propertyDefinitions**

4411

4412 ~~**propertyDefinitions**~~

4413 The values for this attribute are defined in Section 4.3.8.3.

4414 ~~4.3.7.~~4.3.8.3 **Property Definitions**

4415 The WikiPage class inherits property definitions from super classes.

4416 The WikiPage class MUST have the property definitions:

4417

4418 **icom_doc:renderedPage**

4419 Description: ~~A~~An html page ~~rendered~~ated from ~~a~~wiki ~~page~~ content.

4420 Required: False

4421 Inherited: False

4422 Property Type: String

4423 Cardinality: Single

4424 Updatability: Read Only

4425

4426 ~~**icom_doc:renderedContent**~~

4427 ~~Description: An object rendered from wiki page content.~~

4428 ~~Required: False~~

4429 ~~Inherited: False~~

4430 ~~Property Type: Object~~

4431 ~~Cardinality: Single~~

4432 ~~Updatability: Read Only~~

4433

4434 The WikiPage class MAY include additional property definitions which are implementation-defined.

4435

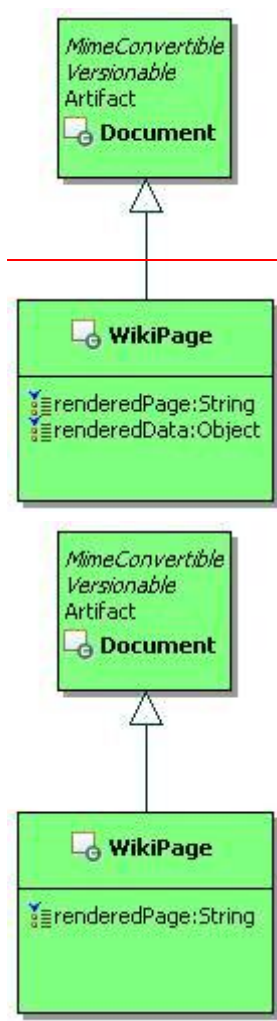


Figure 27 ~~WikiPage~~: Wiki Page Class Diagram.

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 ~~sent time of a message is represented by a~~ user creation date and time ~~of property can be used as~~ the sent date and time of a message. The name property ~~holds~~ can be used as the subject of a message.

~~4.4.1.21.1.1.1 Class Definition~~

4.4.1.2 Class Definition

The Message class ~~is defined by the~~ has attribute values:

4493	Description:	The <u>date and</u> time when a message is delivered to a given
4494		recipient.
4495	Required:	False
4496	Inherited:	False
4497	Property Type:	DateTime
4498	Cardinality:	Single
4499	Updatability:	Read Only

4500

4501 ~~icom_content:content~~

4502	Description:	Content of a message
4503	Required:	False
4504	Inherited:	False
4505	Property Type:	icom_content:Content
4506	Cardinality:	Single
4507	Updatability:	Read Write

4508

4509 The Message class MAY include additional property definitions which are implementation-defined.

4510

4511 4.4.2 UnifiedMessage

4512 4.4.2.1 Description

4513 A unified message is a ~~special~~ type of message delivered electronically over a computer, voice, fax, and
4514 other networks.

4515 A unified message can be one of these types:

- 4516 • Email is a ~~type of~~ message ~~that is~~ delivered electronically over a computer network.
- 4517 • Voice is a ~~type of~~ message that contains a voice or audio stream.
- 4518 • Fax is a ~~type of~~ message that contains an image transmitted via phone lines using the fax protocol.
- 4519 • Notification is a type of message sent by applications.

4520 4.4.2.2 Class Definition

4521 The UnifiedMessage class ~~is defined by the~~has attribute values:

4522

4523 **localNamespace**

4524 Value: icom_msg

4525

4526 **localName**

4527 Value: UnifiedMessage

4528

4529 **extendsFrom**

4530 Value: icom_msg:Message, icom_content:MimeConvertible

4531

4532 **stereotype**

4533 Value: primary

description

Value: A unified message is a ~~special~~ type of message delivered electronically over a computer, voice, fax, and other networks.

propertyDefinitions

The values for this attribute are defined in Section 4.4.2.3.

4.4.2.3 Property Definitions

The UnifiedMessage class inherits property definitions from super classes.

The UnifiedMessage class MUST have the property definitions:

icom_core:priority

Description: The priority of a message.

Required: False

Inherited: False

Property Type: icom_core:Priority

Cardinality: Single

Updatability: Read Write

icom_content:contentId

Description: Content id is a unique identifier for a message part in multi-part messages.

Required: False

Inherited: False

Property Type: String

Cardinality: Single

Updatability: Read Write

icom_content:mediaType

Description: Media type is a two-part identifier for Internet file formats.

Required: False

Inherited: False

Property Type: String

Cardinality: Single

Updatability: Read Write

icom_content:contentDisposition

Description: Content disposition specifies a presentation style.

Required: False

Inherited: False

Property Type: icom_content:ContentDispositionType

Cardinality: Single

4576	<u>Updatability:</u>	<u>Read Write</u>
4577		
4578	icom_msg:envelopeSender	
4579	Description:	An envelope sender (sometimes called <u>is a participant to</u>
4580		<u>receive bounced message. It is also known as</u> return path) of
4581		<u>a message.</u>
4582	Required:	False
4583	Inherited:	False
4584	Property Type:	icom_core:Participant
4585	Cardinality:	Single
4586	Updatability:	Read Write
4587		
4588	icom_msg:toReceivers	
4589	Description:	A list of participants to whom <u>receive</u> a message is sent or to
4590		<u>be sent.</u>
4591	Required:	False
4592	Inherited:	False
4593	Property Type:	icom_core:Participant
4594	Cardinality:	Multi
4595	Updatability:	Read Write
4596		
4597	icom_msg:ccReceivers	
4598	Description:	A list of participants to whom <u>receive carbon-copies of</u> a
4599		message is carbon-copied or to be carbon-copied.
4600	Required:	False
4601	Inherited:	False
4602	Property Type:	icom_core:Participant
4603	Cardinality:	Multi
4604	Updatability:	Read Write
4605		
4606	icom_msg:bccReceivers	
4607	Description:	A list of participants to whom a message is <u>receive</u> blind-
4608		carbon- copied or to be blind-carbon-copied <u>copies of a</u>
4609		<u>message.</u>
4610	Required:	False
4611	Inherited:	False
4612	Property Type:	icom_core:Participant
4613	Cardinality:	Multi
4614	Updatability:	Read Write
4615		
4616	icom_msg:replyTo	
4617	Description:	A list of participants to whom <u>receive a</u> reply messages should
4618		<u>be sent</u> <u>message.</u>
4619	Required:	False

4620	Inherited:	False
4621	Property Type:	icom_core:Participant
4622	Cardinality:	Multi
4623	Updatability:	Read Write
4624		
4625	icom_content:contentId	
4626	Description:	Content id is a unique identifier for a message part in multi-part messages.
4627		
4628	Required:	False
4629	Inherited:	False
4630	Property Type:	String
4631	Cardinality:	Single
4632	Updatability:	Read Write
4633		
4634	icom_content:mediaType	
4635	Description:	Media type is a two-part identifier for Internet file formats as defined in RFC 2046 and additional RFCs including RFC 3236, RFC 1847, etc.
4636		
4637		
4638	Required:	False
4639	Inherited:	False
4640	Property Type:	String
4641	Cardinality:	Single
4642	Updatability:	Read Write
4643		
4644	icom_content:contentDisposition	
4645	Description:	Content disposition is defined in RFC 2183 to specify a presentation style.
4646		
4647	Required:	False
4648	Inherited:	False
4649	Property Type:	icom_content:ContentDispositionType
4650	Cardinality:	Single
4651	Updatability:	Read Write
4652		
4653	icom_core:priority	
4654	Description:	The priority of a message.
4655	Required:	False
4656	Inherited:	False
4657	Property Type:	icom_core:Priority
4658	Cardinality:	Single
4659	Updatability:	Read Write
4660		
4661	icom_msg:flag	
4662	Description:	Zero or more flags on a message.

4663	Required:	False
4664	Inherited:	False
4665	Property Type:	icom_msg:UnifiedMessageFlag
4666	Cardinality:	Multi
4667	Updatability:	Read Write
4668		
4669	icom_msg:messageDispositionNotificationRequested	
4670	Description:	A message disposition notification (RFC 2298) is requested
4671		for a message.
4672	Required:	False
4673	Inherited:	False
4674	Property Type:	Boolean
4675	Cardinality:	Single
4676	Updatability:	Read Write
4677		
4678	icom_msg:messageDeliveryStatusNotificationRequest	
4679	Description:	Indicates the types of delivery status notifications (RFC 1891)
4680		requested for a message. Default is icom_msg:Failure.
4681	Required:	False
4682	Inherited:	False
4683	Property Type:	icom_msg:UnifiedMessageDeliveryStatusNotificationRequest
4684	Cardinality:	Multi
4685	Updatability:	Read Write
4686		
4687	icom_msg:channel	
4688	Description:	Indicates the delivery channel of a message.
4689	Required:	False
4690	Inherited:	False
4691	Property Type:	icom_msg:UnifiedMessageChannel
4692	Cardinality:	Single
4693	Updatability:	Read Write
4694		
4695	icom_msg:mode<u>editMode</u>	
4696	Description:	Indicates an editable mode (new, draft, or delivered) of a
4697		message.
4698	Required:	False
4699	Inherited:	False
4700	Property Type:	icom_msg:UnifiedMessageEditMode
4701	Cardinality:	Single
4702	Updatability:	Read Only
4703		
4704	icom_msg:mimeHeader	

4705 Description: A list of headers ~~including those defined in RFC 822 and~~
4706 ~~other custom headers.~~ Each header is represented by a
4707 multi-valued property. ~~The name of a property is a printable~~
4708 ~~header name. The value of a property is a collection of ascii~~
4709 ~~or non-ascii strings.~~

4710 Required: False
4711 Inherited: False
4712 Property Type: icom_meta:Property
4713 Cardinality: Multi
4714 Updatability: Read Write

4715

4716 **icom_msg:size**

4717 Description: The size of a unified message.
4718 Required: False
4719 Inherited: False
4720 Property Type: Integer
4721 Cardinality: Single
4722 Updatability: Read Only

4723

4724 The UnifiedMessage class MAY include additional property definitions which are implementation-defined.

4725

4726 **4.4.3 UnifiedMessageParticipant**

4727 **4.4.3.1 Description**

4728 A unified message participant object represents the participation of an addressable entity in a unified
4729 message.

4730 **4.4.3.2 Class Definition**

4731 | The UnifiedMessageParticipant class ~~is defined by the~~has attribute values:

4732

4733 **localNamespace**

4734 Value: icom_msg

4735

4736 **localName**

4737 Value: UnifiedMessageParticipant

4738

4739 **extendsFrom**

4740 Value: icom_core:Participant

4741

4742 **stereotype**

4743 Value: primary

4744

4745 **description**

4746 Value: A unified message participant object represents the participation of an addressable entity
4747 in a unified message.

4748

4749 **propertyDefinitions**

4750 The values for this attribute are defined in Section 4.4.3.3.

4751 **4.4.3.3 Property Definitions**

4752 The UnifiedMessageParticipant class inherits property definitions from super classes.

4753 The UnifiedMessageParticipant class MUST have the property definitions:

4754

4755 **icom_msg:fullAddress**

4756 Description: Full address of a participant.

4757 Required: False

4758 Inherited: False

4759 Property Type: ~~URI~~

4760 Cardinality: Single

4761 Updatability: Read Write

4762

4763 **icom_msg:localPart**

4764 Description: Local part of a full address.

4765 Required: False

4766 Inherited: False

4767 Property Type: String

4768 Cardinality: Single

4769 Updatability: Read Write

4770

4771 **icom_msg:domainPart**

4772 Description: Domain part of a full address.

4773 Required: False

4774 Inherited: False

4775 Property Type: String

4776 Cardinality: Single

4777 Updatability: Read Write

4778

4779 The UnifiedMessageParticipant class MAY include additional property definitions which are
4780 implementation-defined.

4781

4782 **4.4.4 UnifiedMessageFlag**

4783 ~~The UnifiedMessageFlag class is an enum class that enumerates the instances each of which expresses~~
4784 ~~a type of flag.~~

4785 **4.4.4.1 Description**

4786 A unified message flag is a flag on a message.

4.4.4.2 Class Definition

The UnifiedMessageFlag class is ~~defined by the~~ a mixin class which defines a flag on a message.
The UnifiedMessageFlag class has attribute values:

localNamespace

Value: icom_msg

localName

Value: UnifiedMessageFlag

extendsFrom

Value:

stereotype

Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.4.4.3.

4.4.4.3 Property Definitions

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

4.4.5 UnifiedMessageFlagEnum

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

The UnifiedMessageFlagEnum class has attribute values:

localNamespace

Value: icom_msg

localName

Value: UnifiedMessageFlagEnum

extendsFrom

Value: UnifiedMessageFlag

stereotype

Value: primary

4828

4829 **isEnumeration**

4830 Value: TRUE

4831

4832 **description**

4833 Value: ~~An enumeration of the instances each of which expresses a type of~~ A flag on a message.

4834

4835 **instances**

4836 Value: <icom_msg:Answered, icom_msg:Forwarded, icom_msg:Redirected, icom_msg:Hidden,

4837 icom_msg:MarkedForDelete, icom_msg:MarkedForFollowUp, icom_msg:MarkedForDraft,

4838 icom_msg:MessageDispositionNotificationProcessed>

4839

4840 ~~There are~~ ICOM defines eight flags ~~defined by ICOM:~~

4841

- **icom_msg:Answered** ~~to express that~~ a message is answered.
- **icom_msg:Forwarded** ~~to express that~~ a message is forwarded.
- **icom_msg:Redirected** ~~to express that~~ a message is redirected.
- **icom_msg:Hidden** ~~to express that~~ a message is hidden.
- **icom_msg:MarkedForDelete** ~~to express that~~ a message is marked for delete.
- **icom_msg:MarkedForFollowUp** ~~to express that~~ a message is marked for follow up.
- **icom_msg:MarkedForDraft** ~~to express that~~ a message is marked for draft.
- **icom_msg:MessageDispositionNotificationProcessed** ~~to express that~~ a message disposition notification ~~(RFC 2298)~~ is processed.

4849

4850

4851 ~~4.4.54.4.6~~ **UnifiedMessageDeliveryStatusNotificationRequest**

4852 **4.4.6.1 Description**

4853 ~~A unified message delivery status notification request is a directive for notifying a participant of delivery~~

4854 ~~status of a message.~~

4855 **4.4.6.2 Class Definition**

4856 The UnifiedMessageDeliveryStatusNotificationRequest ~~class is a mixin class which defines a directive for~~

4857 ~~notifying a participant of delivery status of a message.~~

4858 ~~The UnifiedMessageDeliveryStatusNotificationRequest class has attribute values:~~

4859

4860 **localNamespace**

4861 Value: icom_msg

4862

4863 **localName**

4864 Value: UnifiedMessageDeliveryStatusNotificationRequest

4865

4866 **extendsFrom**

4867 Value:

4868

4869 stereotype

4870 Value: mixin

4871

4872 description

4873 Value: UnifiedMessageDeliveryStatusNotificationRequest is a mixin class which defines a

4874 directive for notifying a participant of delivery status of a message.

4875

4876 propertyDefinitions

4877 The values for this attribute are defined in Section 4.4.6.3.

4878 **4.4.6.3 Property Definitions**

4879 The UnifiedMessageDeliveryStatusNotificationRequest class MAY include additional property definitions

4880 which are implementation-defined.

4881

4882 **4.4.7 UnifiedMessageDeliveryStatusNotificationRequestEnum**

4883 The UnifiedMessageDeliveryStatusNotificationRequestEnum class is an enum class that enumerates the

4884 instances each of which expresses a request for one of several types of delivery status notification

4885 defined in RFC 1891.

4886 The UnifiedMessageDeliveryStatusNotificationRequestEnum class has attribute values:

4887

4888 localNamespace

4889 Value: icom_msg

4890

4891 localName

4892 Value: UnifiedMessageDeliveryStatusNotificationRequestEnum

4893

4894 extendsFrom

4895 Value: UnifiedMessageDeliveryStatusNotificationRequest ~~class is defined by the attribute values:~~

4896

4897 ~~localNamespace~~

4898 ~~Value:~~ ~~icom_msg~~

4899

4900 ~~localName~~

4901 ~~Value:~~ ~~UnifiedMessageDeliveryStatusNotificationRequest~~

4902

4903 ~~extendsFrom~~

4904 ~~Value:~~

4905

4906 ~~stereotype~~

4907 ~~Value:~~

4908

4909 stereotype

4910 Value: primary

4911

4912 **isEnumeration**

4913 Value: TRUE

4914

4915 **description**

4916 | ~~Value:—An enumeration of the instances each of which expresses a~~Value: A request for

4917 one of several types of delivery status notification.

4918

4919 **instances**

4920 Value: <icom_msg:Never, icom_msg:Success, icom_msg:Failure, icom_msg:Delay>

4921

4922 | ~~There are~~ICOM defines four delivery status notification requests~~—defined by ICOM:~~

4923 | • ~~icom_msg:Never—to express that~~ a sender requests status notification not be returned to the

4924 sender under any condition.

4925 | • ~~icom_msg:Success—to express that~~ a sender requests a status notification for successful

4926 delivery of a message.

4927 | • ~~icom_msg:Failure—to express that~~ a sender requests a status notification for delivery failure of a

4928 message.

4929 | • ~~icom_msg:Delay—to express that~~ a sender requests a status notification when delivery of a

4930 message has been delayed for an unusual length of time.

4931

4932 4.4.64.4.8 UnifiedMessageChannel

4933 4.4.8.1 Description

4934 A message channel used to deliver a unified message.

4935 4.4.8.2 Class Definition

4936 The UnifiedMessageChannel class is an enuma mixin class ~~that enumerates the instances each of~~ which

4937 ~~expresses~~defines a ~~a type of delivery~~ channel used to deliver a unified message.

4938 The UnifiedMessageChannel class ~~is defined by the~~has attribute values:

4939

4940 **localNamespace**

4941 Value: icom_msg

4942

4943 **localName**

4944 Value: UnifiedMessageChannel

4945

4946 **extendsFrom**

4947 Value:

4948

4949 **stereotype**

4950 Value: mixin

4951

4952 **description**

Value: UnifiedMessageChannel is a mixin class which defines a channel used to deliver a unified message.

propertyDefinitions

The values for this attribute are defined in Section 4.4.8.3.

4.4.8.3 Property Definitions

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

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: An enumeration of the instances each of which expresses a requested receipt type.~~

~~Value: A delivery channel.~~

instances

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

~~There are~~ICOM defines four channel types ~~defined by ICOM:~~

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

4.4.74.4.10 UnifiedMessageEditMode

4.4.10.1 ~~The UnifiedMessageEditMode class~~ Description

~~A unified message edit mode is an enum class a mode that enumerates the instances each of which expresses~~indicates whether a unified message is ~~a draft copy, delivered copy, or other~~editable.

4.4.10.2 Class Definition

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

The UnifiedMessageEditMode class has attribute values:

localNamespace

Value: icom_msg

localName

Value: UnifiedMessageEditMode

extendsFrom

Value:

stereotype

Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.4.10.3.

4.4.10.3 Property Definitions

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

4.4.11 UnifiedMessageEditModeEnum

The UnifiedMessageEditModeEnum class is an enum class that enumerates the instances each of which expresses whether a message is a new copy, saved draft copy, or delivered copy.

The UnifiedMessageEditModeEnum class has attribute values:

localNamespace

Value: icom_msg

5034 localName

5035 Value: UnifiedMessageEditModeEnum

5036

5037 extendsFrom

5038 Value: UnifiedMessageEditMode

5039

5040 stereotype

5041 Value: primary

5042

5043 **isEnumeration**

5044 Value: TRUE

5045

5046 **description**

5047 Value: ~~An enumeration of the instances each of which expresses whether a~~ message is a new

5048 copy, a saved draft copy, or a delivered copy, or other. New or draft copies are usually editable

5049 while delivered copies are usually not editable.

5050

5051 **instances**

5052 Value: <icom_msg:NewCopy, icom_msg:DraftCopy, icom_msg:DeliveredCopy, ~~icom_msg:Other~~>

5053

5054 ~~There are~~ICOM defines three modes ~~defined by ICOM:~~

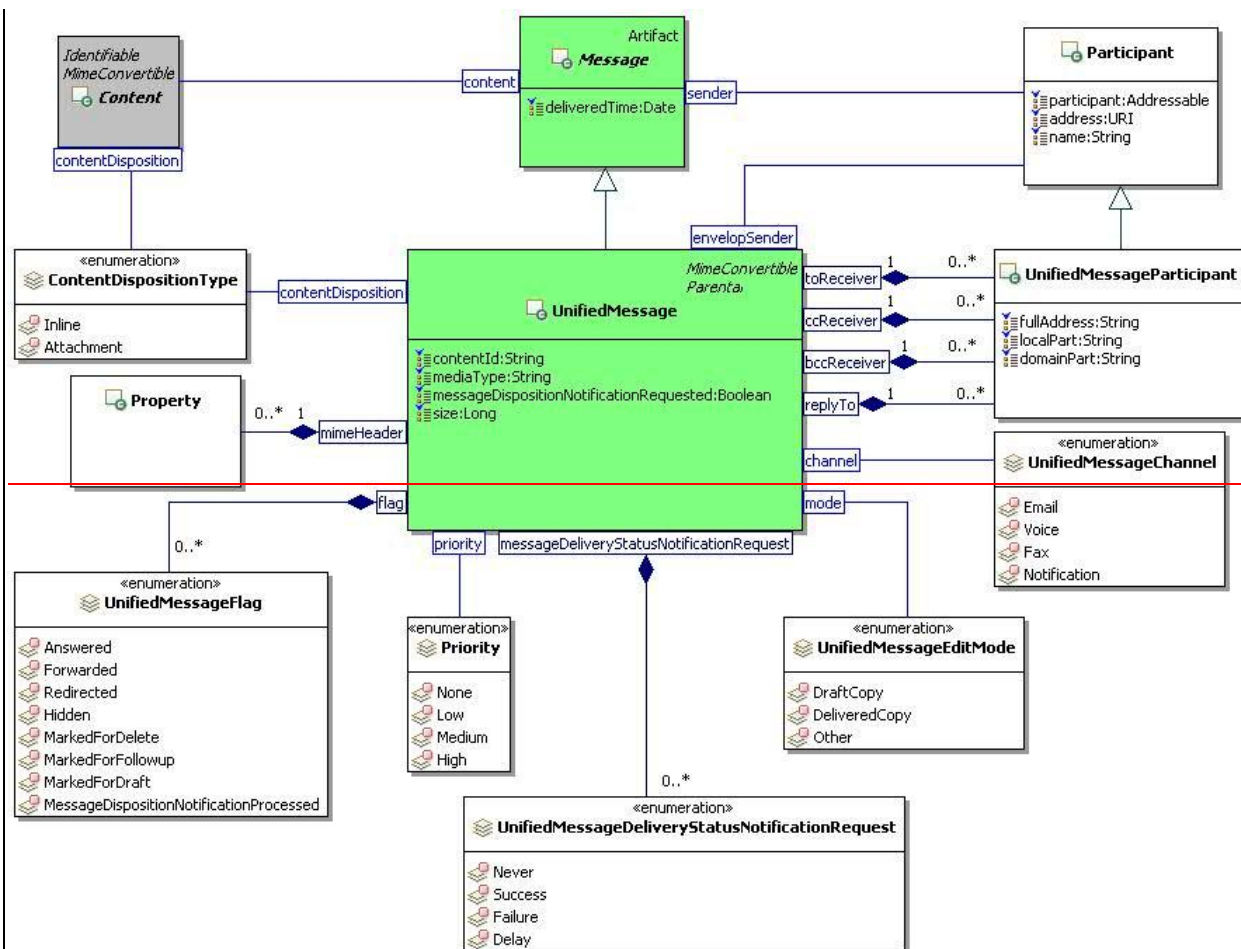
5055 • icom_msg:NewCopy a message is a new message.

5056 • **icom_msg:DraftCopy** ~~to express that~~ a message is saved as a draft.

5057 • **icom_msg:DeliveredCopy** ~~to express that~~ a message is a sent or received message.

5058 • ~~icom_msg:Other to express that a message is other than draft or delivered.~~

5059



5060
5061



4.4.8.14.4.12.1 Description

4.4.8.24.4.12.2 Class Definition

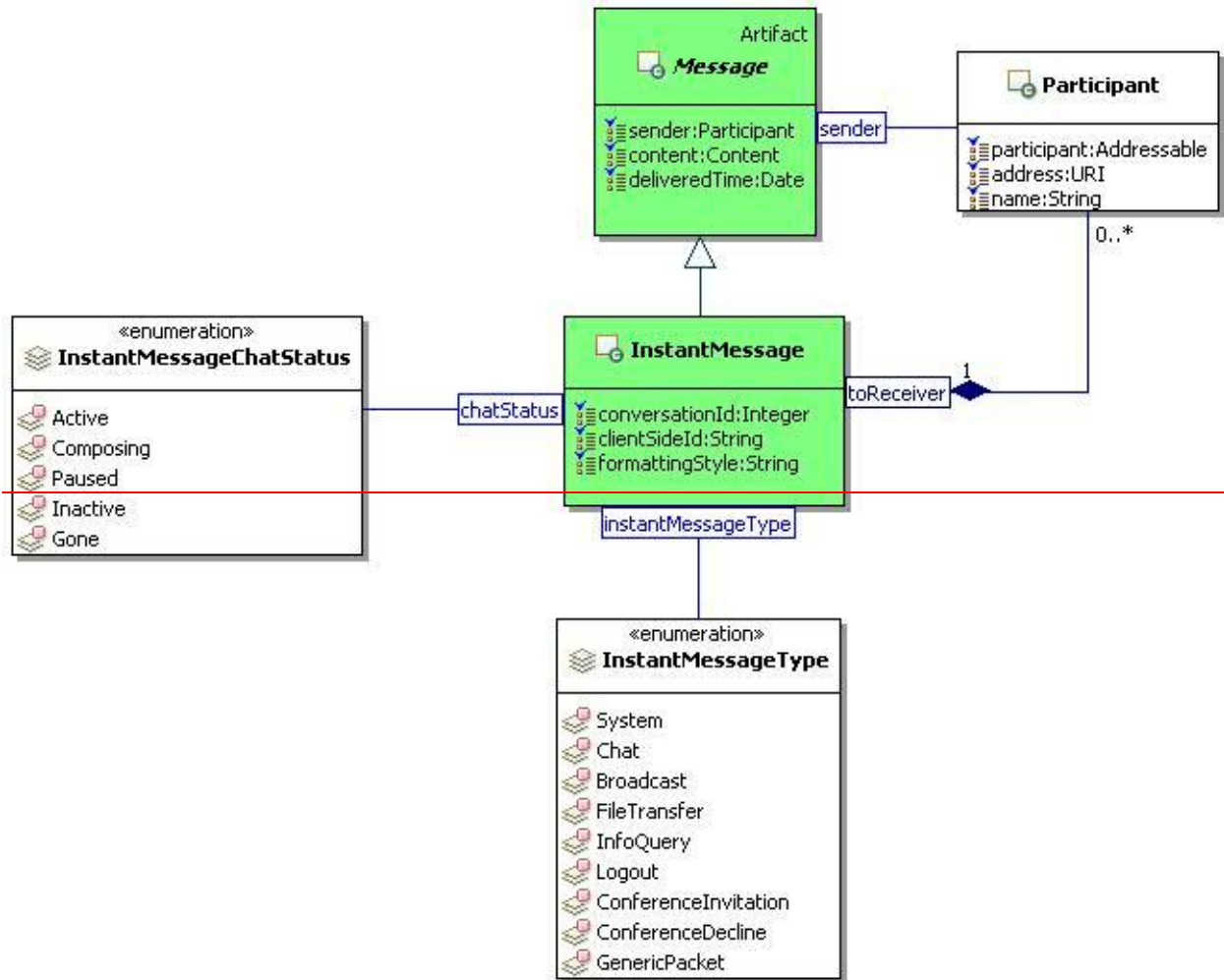
Value: icom msg:Message

5081 **stereotype**
5082 Value: primary
5083
5084 **isAbstract**
5085 Value: TRUE
5086
5087 **description**
5088 Value: An instant message is a ~~special~~ type of message for ~~one-on-one~~, synchronous, usually
5089 text based, conversation.
5090
5091 ~~propertyDefinitions~~
5092
5093 ~~propertyDefinitions~~
5094 The values for this attribute are defined in Section 4.4.12.3.

5095 **4.4.8.34.4.12.3 Property Definitions**
5096 The InstantMessage class inherits property definitions from super classes.
5097 The InstantMessage class MUST have the property definitions:
5098

5099	icom_msg:toReceivers	
5100	Description:	A list of participants to whom <u>receive</u> a message is sent or to
5101		be sent .
5102	Required:	False
5103	Inherited:	False
5104	Property Type:	icom_core:Participant
5105	Cardinality:	Multi
5106	Updatability:	Read Write
5107		
5108	icom_msg:conversationId	
5109	Description:	An identifier of a conversation involving one or more instant
5110		messages.
5111	Required:	False
5112	Inherited:	False
5113	Property Type:	Integer
5114	Cardinality:	Single
5115	Updatability:	Read Write
5116		
5117	icom_msg:clientSidId	
5118	Description:	An identifier of a client.
5119	Required:	False
5120	Inherited:	False
5121	Property Type:	String
5122	Cardinality:	Single

5123	Updatability:	Read Write
5124		
5125	icom_msg:formingStyle	
5126	Description:	A <u>style for</u> formatting style of a rich text message in xhtml .
5127	Required:	False
5128	Inherited:	False
5129	Property Type:	String
5130	Cardinality:	Single
5131	Updatability:	Read Write
5132		
5133	icom_msg:instantMessageType	
5134	Description:	A type of instant message.
5135	Required:	False
5136	Inherited:	False
5137	Property Type:	icom_msg:InstantMessageType
5138	Cardinality:	Single
5139	Updatability:	Read Write
5140		
5141	icom_msg:chatStatus	
5142	Description:	A chat status of a user.
5143	Required:	False
5144	Inherited:	False
5145	Property Type:	icom_msg:InstantMessageChatStatus
5146	Cardinality:	Single
5147	Updatability:	Read Write
5148		



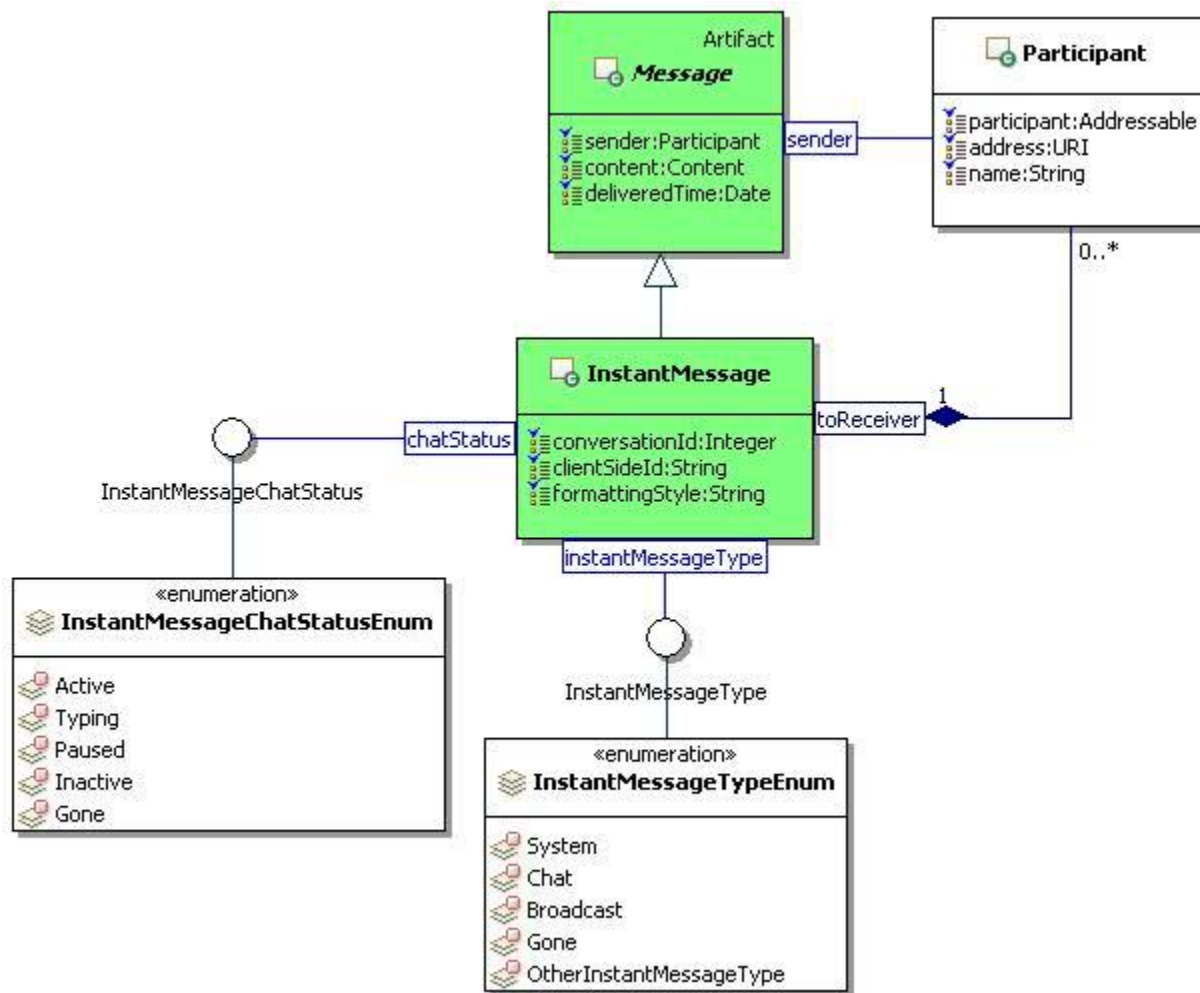


Figure 29 *InstantMessage: Instant Message* Class Diagram.

4.4.94.4.13 InstantMessageType

4.4.13.1 Description

An instant message type.

4.4.13.2 Class Definition

The InstantMessageType class is an enuma mixin class that enumerates the instances each of which expresses/defines a type of instant message.

The InstantMessageType class is defined by the has attribute values:

localNamespace

Value: icom_msg

localName

Value: InstantMessageType

5167 **extendsFrom**
5168 Value:
5169
5170 **stereotype**
5171 Value: mixin
5172
5173 **description**
5174 Value: InstantMessageType is a mixin class which defines a type of instant message.
5175
5176 **propertyDefinitions**
5177 The values for this attribute are defined in Section 4.4.13.3.
5178
5178 **4.4.13.3 Property Definitions**
5179 The InstantMessageType class MAY include additional property definitions which are implementation-
5180 defined.
5181
5182 **4.4.14 InstantMessageTypeEnum**
5183 The InstantMessageTypeEnum class is an enum class that enumerates the instances each of which
5184 expresses a type of instant message.
5185 The InstantMessageTypeEnum class has attribute values:
5186
5187 **localNamespace**
5188 Value: icom_msg
5189
5190 **localName**
5191 Value: InstantMessageTypeEnum
5192
5193 **extendsFrom**
5194 Value: InstantMessageType
5195
5196 **stereotype**
5197 Value: primary
5198
5199 **isEnumeration**
5200 Value: TRUE
5201
5202 **description**
5203 Value: ~~An enumeration of the instances each of which expresses a~~A type of instant message.
5204
5205 ~~**instances**~~
5206
5207 **instances**

5208 Value: <icom_msg:System, icom_msg:Chat, icom_msg:Broadcast, icom_msg:~~FileTransfer,~~
5209 ~~icom_msg:InfoQuery, icom_msg:Logout, icom_msg:ConferenceInvitation,~~
5210 ~~icom_msg:ConferenceDecline, icom_msg:GenericPacketGone,~~
5211 ~~icom_msg:OtherInstantMessageType~~>
5212

5213 ~~There are nine~~ICOM defines five instant message types ~~defined by ICOM:~~

- 5214 • ~~icom_msg:System to express that~~ an instant message is a system message.
- 5215 • ~~icom_msg:Chat to express that~~ an instant message is a chat message.
- 5216 • ~~icom_msg:Broadcast to express that~~ an instant message is a broadcast message.
- 5217 • ~~icom_msg:FileTransfer: to express that~~Gone an instant message is a ~~file transfer message.~~
- 5218 • ~~icom_msg:InfoQuery to express that~~ an instant message is a info query message.
- 5219 • ~~icom_msg:Logout to express that~~ an instant message is a logout message.
- 5220 • ~~icom_msg:ConferenceInvitation to express that~~ an instant message is a conference invitation
- 5221 ~~message.~~
- 5222 • ~~icom_msg:ConferenceDecline to express that~~ an instant message is a decline message to a
- 5223 ~~conference invitation.~~
- 5224 • ~~icom_msg:GenericGone to express that~~ an instant message is a generic message indicating
- 5225 that a user is gone.
- 5226 • icom_msg:OtherInstantMessageType an instant message is of other type.
- 5227

5228 ~~4.4.104.4.15~~ InstantMessageChatStatus

5229 4.4.15.1 Description

5230 An instant message chat status defines a vocabulary of chat status.

5231 4.4.15.2 Class Definition

5232 The InstantMessageChatStatus class is an enuma mixin class ~~that enumerates the instances each of~~
5233 ~~which expressesdefines~~ a chat status ~~of a user.~~

5234 The InstantMessageChatStatus class ~~is defined by the~~has attribute values:

5235

5236 **localNamespace**

5237 Value: icom_msg

5238

5239 **localName**

5240 Value: InstantMessageChatStatus

5241

5242 **extendsFrom**

5243 Value:

5244

5245 **stereotype**

5246 Value: mixin

5247

5248 **description**

5249 Value: InstantMessageChatStatus is a mixin class which defines a chat status.

propertyDefinitions

The values for this attribute are defined in Section 4.4.15.3.

4.4.15.3 Property Definitions

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

4.4.16 InstantMessageChatStatusEnum

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

The InstantMessageChatStatusEnum class has attribute values:

localNamespace

Value: icom_msg

localName

Value: InstantMessageChatStatusEnum

extendsFrom

Value: InstantMessageChatStatus

stereotype

Value: primary

isEnumeration

Value: TRUE

description

Value: ~~An enumeration of the instances each of which expresses a~~ chat status of a user.

instances

Value: <icom_msg:Active, icom_msg:~~Composing~~Typing, icom_msg:Paused, icom_msg:Inactive, icom_msg:Gone>

~~There are~~ICOM defines five chat status ~~defined by ICOM:~~

- **icom_msg:Active** ~~to express that~~ a user is active.
- **icom_msg:~~Composing~~Typing** ~~to express that~~ a user is ~~composing a message~~typing.
- **icom_msg:Paused** ~~to express that~~ a user has paused.
- **icom_msg:Inactive** ~~to express that~~ a user is inactive.
- **icom_msg:Gone** ~~to express that~~ a user is gone.

~~4.4.11~~4.4.17 InstantMessageFeed

~~4.4.11.14~~4.4.17.1 Description

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

~~4.4.11.24~~4.4.17.2 Class Definition

The InstantMessageFeed class ~~is defined by the~~has attribute values:

localNamespace

Value: icom_msg

localName

Value: InstantMessageFeed

extendsFrom

Value: icom_core:Entity

stereotype

Value: primary

description

Value: An instant message feed contains a set of instant message connections and a queue of outbound instant messages.

propertyDefinitions

~~**propertyDefinitions**~~

The values for this attribute are defined in Section 4.4.17.3.

~~4.4.11.34~~4.4.17.3 Property Definitions

The InstantMessageFeed class inherits property definitions from super classes.

The InstantMessageFeed class MUST have the property definitions:

icom_msg:connection

Description: One or more instant messaging connections.

Required: False

Inherited: False

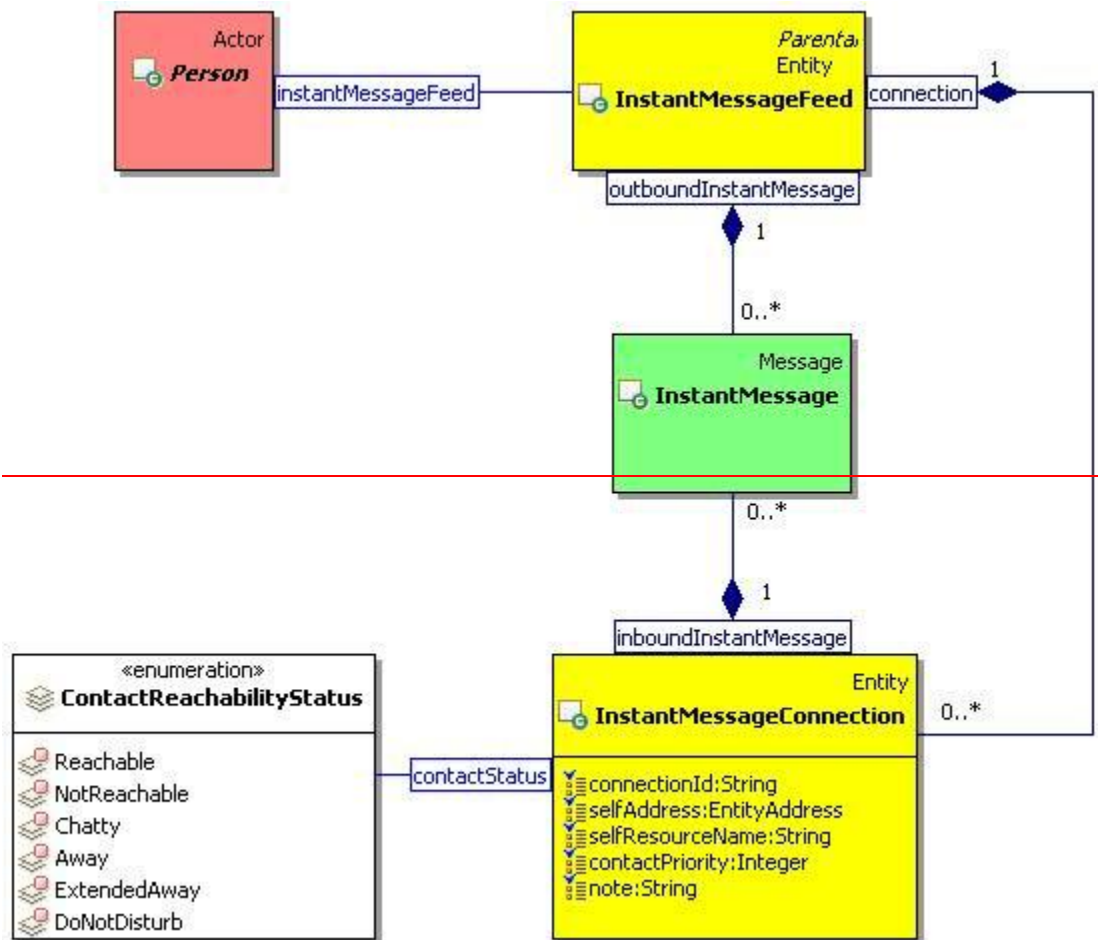
Property Type: icom_msg:InstantMessageConnection

Cardinality: Multi

Updatability: Read Only

icom_msg:outboundInstantMessage

5331 Description: ~~A queue for outbound~~Outbound instant messages.
5332 Required: False
5333 Inherited: False
5334 Property Type: icom_msg:InstantMessage
5335 Cardinality: Multi
5336 Updatability: ~~Required: False~~
5337 ~~Inherited: False~~
5338 ~~Property Type: icom_msg:InstantMessage~~
5339 ~~Cardinality: Multi~~
5340 ~~Updatability: Write Only~~
5341



5342

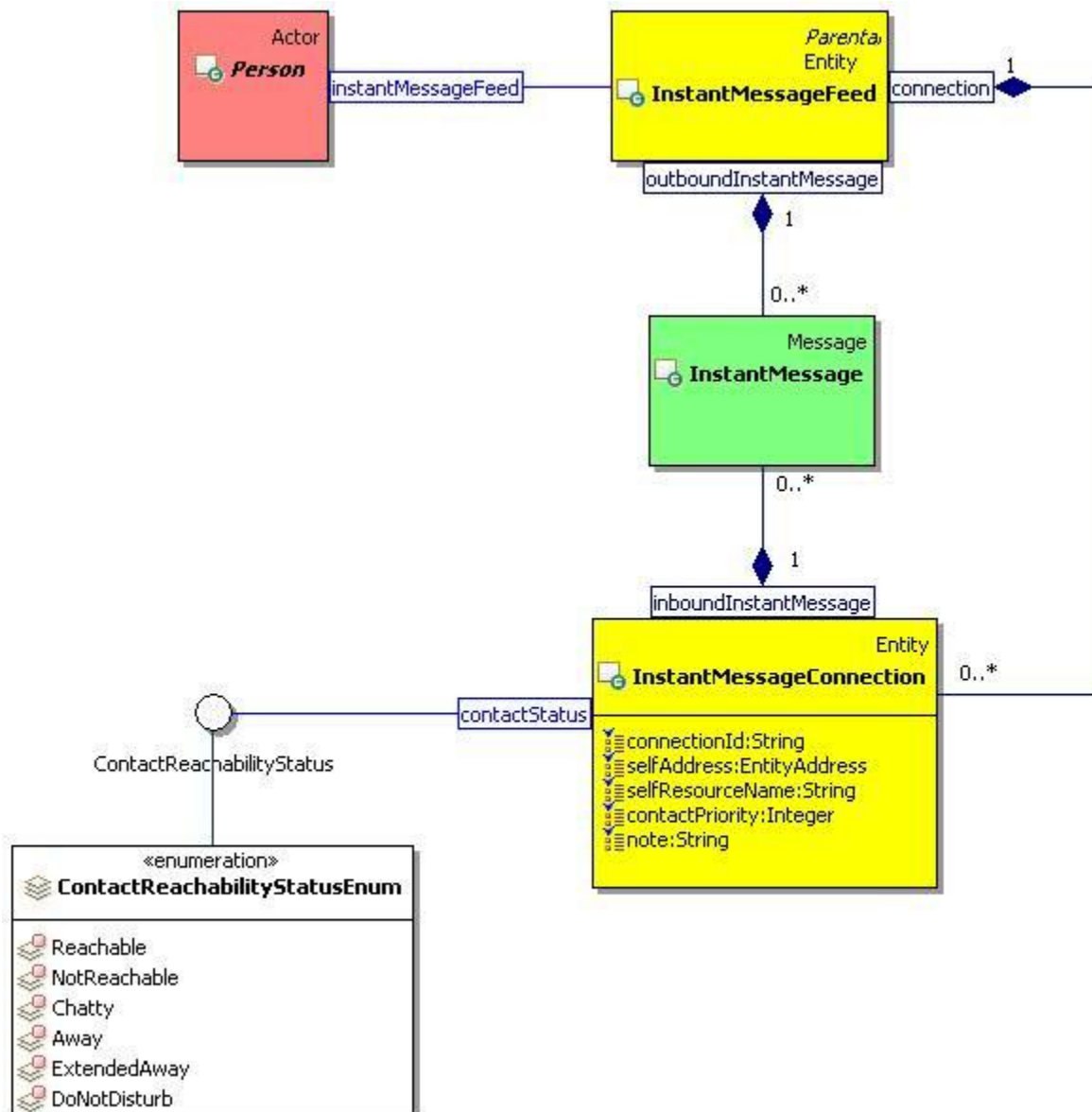


Figure 30: Instant Message Feed and Connection Class Diagram.

4.4.124.4.18 InstantMessageConnection

4.4.12.14.4.18.1 Description

An instant message connection contains ~~the~~ queues for inbound ~~and outbound~~ instant messages.

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

4.4.12.24.4.18.2 Class Definition

The `InstantMessageConnection` class ~~is defined by the~~ `has` attribute values:

5354 **localNamespace**
5355 Value: icom_msg
5356
5357 **localName**
5358 Value: InstantMessageConnection
5359
5360 **extendsFrom**
5361 Value: icom_core:Entity
5362
5363 **stereotype**
5364 Value: primary
5365
5366 **description**
5367 Value: An instant message connection contains ~~the~~ queues for inbound ~~and outbound~~ instant
5368 messages.
5369
5370 **propertyDefinitions**
5371 The values for this attribute are defined in Section 4.4.18.3.

5372 ~~4.4.12.34.4.18.3~~ **Property Definitions**

5373 The InstantMessageConnection class inherits property definitions from super classes.
5374 The InstantMessageConnection class MUST have the property definitions:

5375
5376 **icom_msg:connectionId**
5377 Description: An identifier of a connection.
5378 Required: False
5379 Inherited: False
5380 Property Type: String
5381 Cardinality: Single
5382 Updatability: Read Only
5383
5384 **icom_msg:selfAddress**
5385 Description: Address of a presentity who opens a connection.
5386 Required: True
5387 Inherited: False
5388 Property Type: URI
5389 Cardinality: Single
5390 Updatability: On Create
5391
5392 **icom_msg:selfResourceName**
5393 Description: Resource name associated with a connection.
5394 Required: True
5395 Inherited: False

5396 Property Type: String
5397 Cardinality: Single
5398 Updatability: On Create

5399

5400 **icom_msg:inboundInstantMessage**

5401 Description: Inbound instant messages.
5402 Required: False
5403 Inherited: False
5404 Property Type: icom_msg:InstantMessage
5405 Cardinality: Multi
5406 Updatability: Read Only

5407

5408 **icom_presence:contactStatus**

5409 Description: Reachability status to be propagated to an associated contact
5410 method in presence.
5411 Required: False
5412 Inherited: False
5413 Property Type: icom_presence:ContactReachabilityStatus
5414 Cardinality: Single
5415 Updatability: Write Only

5416

5417 **icom_presence:contactPriority**

5418 Description: Priority to be propagated to an associated contact method in
5419 presence.
5420 Required: False
5421 Inherited: False
5422 Property Type: Integer
5423 Cardinality: Single
5424 Updatability: Write Only

5425

5426 **icom_presence:note**

5427 Description: Note to be propagated to an associated contact method in
5428 presence.
5429 Required: False
5430 Inherited: False
5431 Property Type: String
5432 Cardinality: Single
5433 Updatability: Write Only

5434

5435 **icom_msg:inboundInstantMessage**

5436 Description: A queue for inbound instant messages.
5437 Required: False
5438 Inherited: False

5439 | ~~Property Type:~~ ~~icom_msg:InstantMessage~~
5440 | ~~Cardinality:~~ ~~Multi~~
5441 | ~~Updatability:~~ ~~Read Only~~
5442 |

5443 | **4.5 Presence Module**

5444 | **4.5.1 Presence**

5445 | **4.5.1.1 Description**

5446 | A presence describes the contact methods and activities of a presentity.
5447 | It provides a list of contact methods describing how to contact a presentity. A viewer may choose any one
5448 | of the contact methods based on circumstances.
5449 | It includes a list of activities describing what a presentity is doing.

5450 | **4.5.1.2 Class Definition**

5451 | The Presence class ~~is defined by the~~has attribute values:

5452 |
5453 | **localNamespace**
5454 | Value: icom_presence
5455 |
5456 | **localName**
5457 | Value: Presence
5458 |
5459 | **extendsFrom**
5460 | Value: icom_core:Identifiable
5461 |
5462 | **stereotype**
5463 | Value: primary
5464 |
5465 | **description**
5466 | Value: A presence describes the contact methods and activities of a presentity.
5467 |
5468 | **propertyDefinitions**
5469 | The values for this attribute are defined in Section 4.5.1.3.

5470 | **4.5.1.3 Property Definitions**

5471 | The Presence class inherits property definitions from super classes.
5472 | The Presence class MUST have the property definitions:

5473 |
5474 | **icom_core:lastModificationDate**
5475 | Description: Last modification date and time of information in a presence.
5476 | Required: False
5477 | Inherited: False

5478 Property Type: DateTime
5479 Cardinality: Single
5480 Updatability: Read Only

5481

5482 **icom_presence:mode**

5483 ~~Description: Editable mode of a presence.~~
5484 ~~Required: False~~
5485 ~~Inherited: False~~
5486 ~~Property Type: icom_presence:PresenceEditMode~~
5487 ~~Cardinality: Single~~
5488 ~~Updatability: Read Only~~

5489

5490 **icom_core:location**

5491 Description: Location of a presentity.
5492 Required: False
5493 Inherited: False
5494 Property Type: icom_core:Location
5495 Cardinality: Single
5496 Updatability: Read Only

5497

5498 **icom_presence:editMode**

5499 Description: Indicates a mode which determines whether a presence is
5500 editable.
5501 Required: False
5502 Inherited: False
5503 Property Type: icom_presence:PresenceEditMode
5504 Cardinality: Single
5505 Updatability: Read Only

5506

5507 **icom_presence:contactMethod**

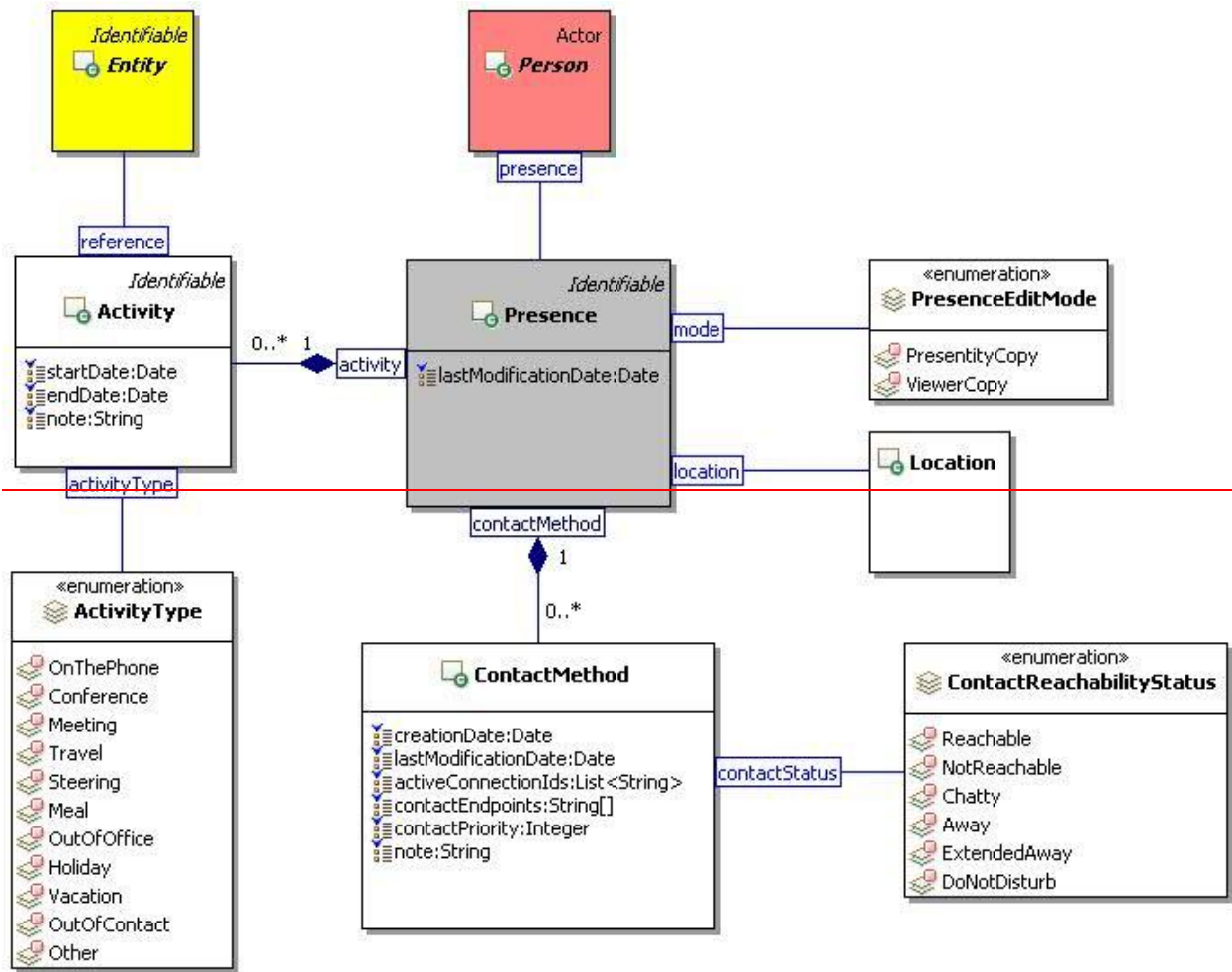
5508 Description: A collection of contact methods describing how to contact a
5509 presentity. A viewer may choose any one of the contact
5510 methods based on circumstances.
5511 Required: False
5512 Inherited: False
5513 Property Type: icom_presence:ContactMethod
5514 Cardinality: Multi
5515 Updatability: Read Only

5516

5517 **icom_presence:activity**

5518 Description: A collection of activities describing what a presentity is doing.
5519 Required: False
5520 Inherited: False

5521	Property Type:	icom_presence:Activity
5522	Cardinality:	Multi
5523	Updatability:	Read Only
5524		
5525	The Presence class MAY include additional property definitions which are implementation-defined.	
5526		



5527

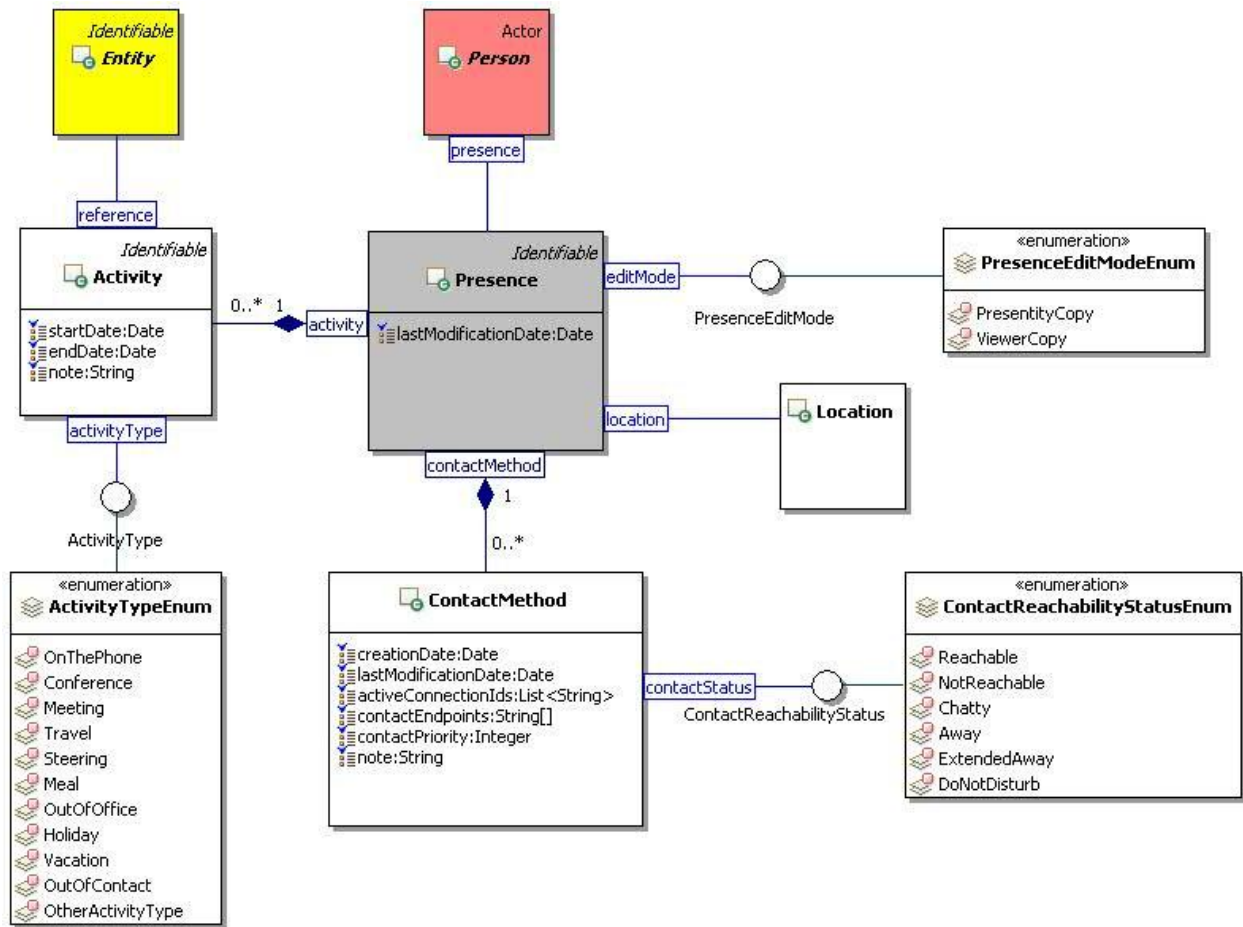


Figure 31: Presence Class Diagram.

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 an **enuma mixin** class that enumerates the instances each of which expresses an **editable** defines a mode of that indicates whether a presence **is** **editable**.

The PresenceEditMode class is defined by the **has** attribute values:

localNamespace

Value: icom_presence

localName

Value: PresenceEditMode

5545 **extendsFrom**
5546 Value:
5547
5548 **stereotype**
5549 Value: mixin
5550
5551 **description**
5552 Value: PresenceEditMode is a mixin class which defines a mode that indicates whether a
5553 presence is editable.
5554
5555 **propertyDefinitions**
5556 The values for this attribute are defined in Section 4.5.2.3.
5557
5558 **4.5.2.3 Property Definitions**
5559 The PresenceEditMode class MAY include additional property definitions which are implementation-
5560 defined.
5561
5562 **4.5.3 PresenceEditModeEnum**
5563 The PresenceEditModeEnum class is an enum class that enumerates the instances each of which
5564 expresses a mode that indicates whether a presence is editable.
5565 The PresenceEditModeEnum class has attribute values:
5566
5567 **localNamespace**
5568 Value: icom_presence
5569
5570 **localName**
5571 Value: PresenceEditModeEnum
5572
5573 **extendsFrom**
5574 Value: PresenceEditMode
5575
5576 **stereotype**
5577 Value: primary
5578
5579 **isEnumeration**
5580 Value: TRUE
5581
5582 **description**
5583 Value: ~~An enumeration of the instances each of which expresses an editable~~ A mode of that
5584 indicates whether a presence is editable.
5585
5586 **instances**
5587 Value: <icom_presence:PresentityCopy, icom_presence:ViewerCopy>

5587

5588 ~~There are~~ICOM defines two presence editable modes ~~defined by ICOM:~~

- 5589 • **icom_presence:PresentityCopy** ~~to express that~~ a presence is a copy belonging to a presentity
- 5590 who may update the properties such as activities.
- 5591 • **icom_presence:ViewerCopy** ~~to express that~~ a presence is a copy visible to a subscriber who
- 5592 may not update the properties.

5594 ~~4.5.3~~4.5.4 **ContactMethod**

5595 ~~4.5.3.1~~4.5.4.1 **Description**

5596 A contact method object describes reachability circumstances of a presentity.

5597 ~~4.5.3.2~~4.5.4.2 **Class Definition**

5598 The ContactMethod class ~~is defined by the~~has attribute values:

5599

5600 **localNamespace**

5601 Value: icom_presence

5602

5603 **localName**

5604 Value: ContactMethod

5605

5606 **extendsFrom**

5607 Value:

5608

5609 **stereotype**

5610 Value: primary

5611

5612 **description**

5613 Value: A contact method object describes reachability circumstances of a presentity.

5614

5615 **propertyDefinitions**

5616 The values for this attribute are defined in Section 4.5.4.3

5617 ~~4.5.3.3~~4.5.4.3 **Property Definitions**

5618 The ContactMethod class MUST have the property definitions:

5619

5620	icom_core:creationDate	
5621	Description:	Creation date and time of information in a contact method.
5622	Required:	False
5623	Inherited:	False
5624	Property Type:	DateTime
5625	Cardinality:	Single
5626	Updatability:	Read Only

5627		
5628	icom_core:lastModificationDate	
5629	Description:	Last modification date and time of information in a contact
5630		method.
5631	Required:	False
5632	Inherited:	False
5633	Property Type:	DateTime
5634	Cardinality:	Single
5635	Updatability:	Read Only
5636		
5637	icom_presence:activeConnectionId	
5638	Description:	A list of active connection ids of a presentity.
5639	Required:	False
5640	Inherited:	False
5641	Property Type:	String
5642	Cardinality:	Multi
5643	Updatability:	Read Only
5644		
5645	icom_presence:contactEndpoint	
5646	Description:	A list of endpoints or URIs for contacting a presentity.
5647	Required:	False
5648	Inherited:	False
5649	Property Type:	String
5650	Cardinality:	Multi
5651	Updatability:	Read Only
5652		
5653	icom_presence:contactPriority	
5654	Description:	Priority of a contact method relative to other contact methods
5655		in a presence.
5656	Required:	False
5657	Inherited:	False
5658	Property Type:	Integer
5659	Cardinality:	Single
5660	Updatability:	Read Only
5661		
5662	icom_presence:contactStatus	
5663	Description:	Status of a contact method in a presence.
5664	Required:	False
5665	Inherited:	False
5666	Property Type:	icom_presence:ContactReachabilityStatus
5667	Cardinality:	Single
5668	Updatability:	Read Only
5669		

5670 **icom_presence:note**

5671 Description: A note about ~~a contact method in a presence~~contacting a

5672 present entity.

5673 Required: False

5674 Inherited: False

5675 Property Type: String

5676 Cardinality: Single

5677 Updatability: Read Only

5678

5679 ~~4.5.44.5.5~~ **ContactReachabilityStatus**

5680 **4.5.5.1 Description**

5681 A contact reachability status is a status of a contact method.

5682 **4.5.5.2 Class Definition**

5683 The ContactReachabilityStatus class is ~~an enuma mixin~~ class ~~that enumerates the instances each of~~

5684 which ~~expresses a reachability~~defines a status of a contact method.

5685 The ContactReachabilityStatus class ~~is defined by the~~has attribute values:

5686

5687 **localNamespace**

5688 Value: icom_presence

5689

5690 **localName**

5691 Value: ContactReachabilityStatus

5692

5693 **extendsFrom**

5694 Value:

5695

5696 **stereotype**

5697 Value: mixin

5698

5699 **description**

5700 Value: ContactReachabilityStatus is a mixin class which defines a status of a contact method.

5701

5702 **propertyDefinitions**

5703 The values for this attribute are defined in Section 4.5.5.3.

5704 **4.5.5.3 Property Definitions**

5705 The ContactReachabilityStatus class MAY include additional property definitions which are

5706 implementation-defined.

5707

4.5.6 ContactReachabilityStatusEnum

The ContactReachabilityStatusEnum class is an enum class that enumerates the instances each of which expresses a reachability status of a contact method.

The ContactReachabilityStatusEnum class has attribute values:

localNamespace

Value: icom_presence

localName

Value: ContactReachabilityStatusEnum

extendsFrom

Value: ContactReachabilityStatus

stereotype

Value: primary

isEnumeration

Value: TRUE

description

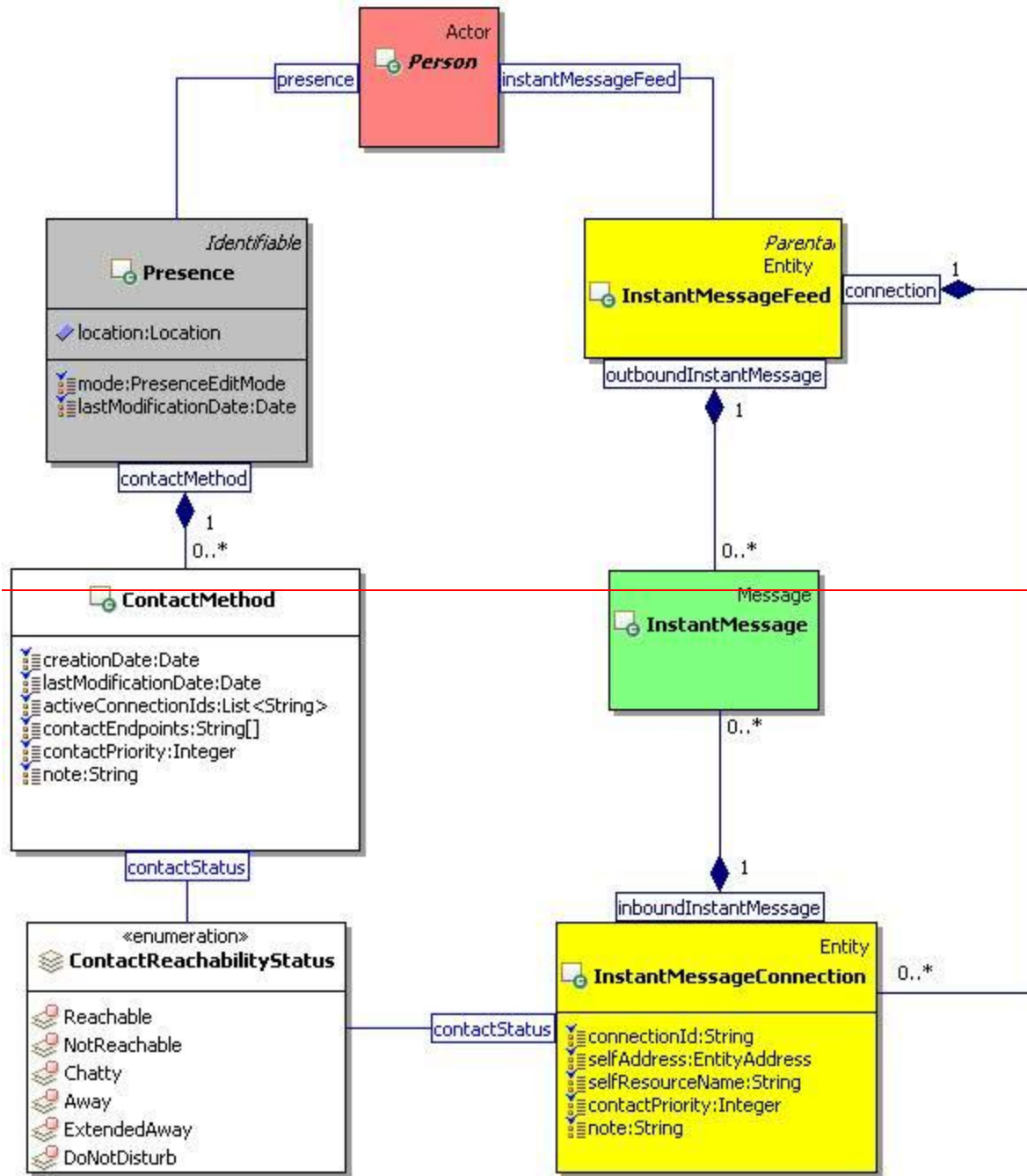
Value: ~~An enumeration of the instances each of which expresses a~~ reachability status of a contact method.

instances

Value: <icom_presence:Reachable, icom_presence:NotReachable, icom_presence:Chatty, icom_presence:Away, icom_presence:ExtendedAway, icom_presence:DoNotDisturb>

~~There are~~ ICOM defines six reachability status ~~defined by ICOM~~:

- **icom_presence:Reachable** ~~to express that~~ a presentity is reachable through a contact method.
- **icom_presence:NotReachable** ~~to express that~~ a presentity is not reachable through a contact method.
- **icom_presence:Chatty** ~~to express that~~ a presentity is chatty.
- **icom_presence:Away** ~~to express that~~ a presentity is away.
- **icom_presence:ExtendedAway** ~~to express that~~ a presentity is away for an extended period.
- **icom_presence:DoNotDisturb** ~~to express that~~ a presentity prefers not to be disturbed.



5745

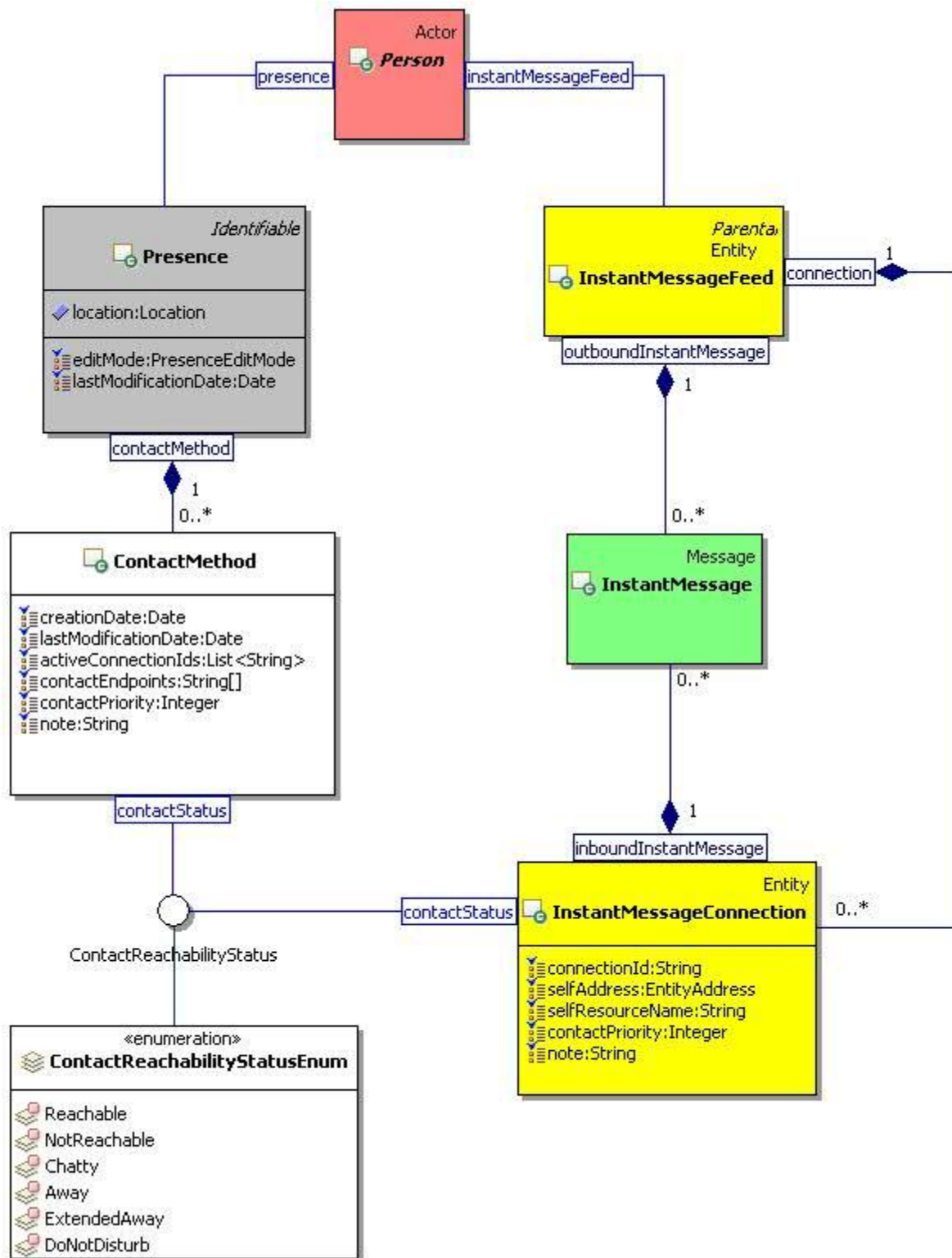


Figure 32: Presence Contact Method and Instant Message Connection Class Diagram.

~~4.5.5.4~~4.5.7 Activity

~~4.5.5.14~~4.5.7.1 Description

An activity object describes what a presentity is ~~currently~~ doing.

~~4.5.5.24~~4.5.7.2 Class Definition

The Activity class ~~is defined by the~~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 ~~currently~~ doing.

~~propertyDefinitions~~

~~propertyDefinitions~~

The values for this attribute are defined in Section 4.5.7.3.

~~4.5.5.34~~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

5789	Property Type:	DateTime
5790	Cardinality:	Single
5791	Updatability:	Read Write
5792		
5793	icom_presence:activityType	
5794	Description:	Type of an activity.
5795	Required:	true
5796	Inherited:	False
5797	Property Type:	icom_presence:ActivityType
5798	Cardinality:	Single
5799	Updatability:	Read Write
5800		
5801	icom_presence:note	
5802	Description:	A note describing an activity.
5803	Required:	False
5804	Inherited:	False
5805	Property Type:	String
5806	Cardinality:	Single
5807	Updatability:	Read Write
5808		
5809	icom_presence:reference	
5810	Description:	An entity, such as occurrence, task, conference, etc., which is
5811		the source of or reference for an activity.
5812	Required:	False
5813	Inherited:	False
5814	Property Type:	icom_core:Entity
5815	Cardinality:	Single
5816	Updatability:	Read Write
5817		

4.5.64.5.8 ActivityType

4.5.8.1 Description

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

4.5.8.2 Class Definition

~~The ActivityType class is an enum class that enumerates the instances each of which expresses a type of activity.~~

The ActivityType class is ~~defined by the~~ a mixin class which defines an activity.

The ActivityType class has attribute values:

localNamespace

Value: icom_presence

5830 **localName**

5831 Value: ActivityType

5832

5833 **extendsFrom**

5834 Value:

5835

5836 **stereotype**

5837 Value: mixin

5838

5839 **description**

5840 Value: ActivityType is a mixin class which defines a type of activity.

5841

5842 **propertyDefinitions**

5843 The values for this attribute are defined in Section 4.5.8.3.

5844 **4.5.8.3 Property Definitions**

5845 The ActivityType class MAY include additional property definitions which are implementation-defined.

5846

5847 **4.5.9 ActivityTypeEnum**

5848 The ActivityTypeEnum class is an enum class that enumerates the instances each of which expresses a

5849 type of activity.

5850 The ActivityTypeEnum class has attribute values:

5851

5852 **localNamespace**

5853 Value: icom_presence

5854

5855 **localName**

5856 Value: ActivityTypeEnum

5857

5858 **extendsFrom**

5859 Value: ActivityType

5860

5861 **stereotype**

5862 Value: primary

5863

5864 **isEnumeration**

5865 Value: TRUE

5866

5867 **description**

5868 Value: ~~An enumeration of the instances each of which expresses a~~A type of activity.

5869

5870 **instances**

5871 Value: <icom_presence:OnThePhone, icom_presence:Conference, icom_presence:Meeting,
5872 icom_presence:Travel, icom_presence:Steering, icom_presence:Meal,
5873 icom_presence:OutOfOffice, icom_presence:Holiday, icom_presence:Vacation,
5874 icom_presence:OutOfContact, icom_presence:~~Other~~OtherActivityType>

5875

5876 ~~There are~~ICOM defines eleven activity types ~~defined by ICOM~~:

- 5877 • ~~icom_presence:OnThePhone to express that~~ a presentity is on the phone.
- 5878 • ~~icom_presence:Conference to express that~~ a presentity is in a conference.
- 5879 • ~~icom_presence:Meeting to express that~~ a presentity is in a meeting.
- 5880 • ~~icom_presence:Travel to express that~~ a presentity is traveling.
- 5881 • ~~icom_presence:Steering to express that~~ a presentity is steering a vehicle.
- 5882 • ~~icom_presence:Meal to express that~~ a presentity is having a meal.
- 5883 • ~~icom_presence:OutOfOffice to express that~~ a presentity is out of office.
- 5884 • ~~icom_presence:Holiday to express that~~ a presentity is on holiday.
- 5885 • ~~icom_presence:Vacation to express that~~ a presentity is on vacation.
- 5886 • ~~icom_presence:OutOfContact to express that~~ a presentity is out of contact.
- 5887 • ~~icom_presence:Other to express that~~OtherActivityType a presentity is involved in an
- 5888 unspecified activity.

5889

5890 4.6 Address Book Module

5891 4.6.1 AddressBook

5892 4.6.1.1 Description

5893 | An address book is a folder that contains sub-address books and addressable contacts.

5894 4.6.1.2 Class Definition

5895 | The AddressBook class ~~is defined by the~~has attribute values:

5896

5897 ~~localNamespace~~

5898

5899 localNamespace

5900 Value: icom_card

5901

5902 ~~localName~~

5903

5904 localName

5905 Value: AddressBook

5906

5907 ~~extendsFrom~~

5908

5909 extendsFrom

5910 Value: icom_core:Folder

5911

5912 **stereotype**

5913 Value: primary

5914

5915 **description**

5916 | Value: An address book is a folder that contains [sub-address books and](#) addressable contacts.

5917

5918 **propertyDefinitions**

5919 The values for this attribute are defined in Section 4.6.1.3.

5920 **4.6.1.3 Property Definitions**

5921 The AddressBook class inherits property definitions from super classes.

5922 The AddressBook class MUST have the property definitions:

5923

5924 **icom_card:addressBook**

5925 Description:	Sub-address books in an address book.
5926 Required:	False
5927 Inherited:	False
5928 Property Type:	icom_card:AddressBook
5929 Cardinality:	Multi
5930 Updatability:	Read Only

5931

5932 **icom_card:contact**

5933 Description:	Contacts in an address book.
5934 Required:	False
5935 Inherited:	False
5936 Property Type:	icom_card:PersonContact
5937 Cardinality:	Multi
5938 Updatability:	Read Only

5939

5940 The AddressBook class MAY include additional property definitions which are implementation-defined.

5941

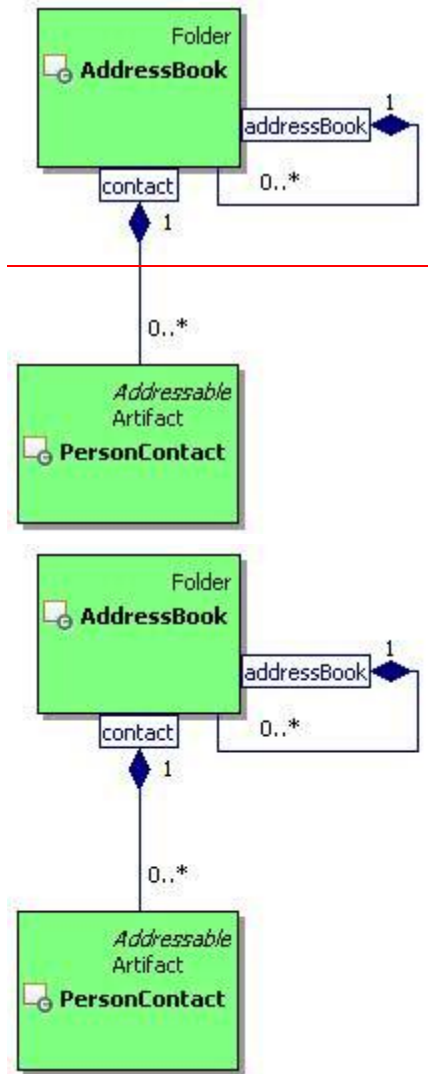


Figure 33-~~AddressBook~~: Address Book Class Diagram.

4.6.2 PersonContact

4.6.2.1 Description

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

4.6.2.2 Class Definition

The PersonContact class ~~is defined by the~~has attribute values:

localNamespace

Value: icom_card

localName

Value: PersonContact

5957

5958 **extendsFrom**

5959 Value: icom_core:Artifact, icom_core:Addressable

5960

5961 **stereotype**

5962 Value: primary

5963

5964 **description**

5965 Value: A person contact is an artifact that contains address information about a person.

5966

5967 **propertyDefinitions**

5968 The values for this attribute are defined in Section 4.6.2.3.

5969 4.6.2.3 Property Definitions

5970 The PersonContact class inherits property definitions from super classes.

5971 The PersonContact class MUST have the property definitions:

5972

5973 ~~icom_card:bookmark~~

5974 ~~Description: A person which is bookmarked by a contact.~~

5975 ~~Required: False~~

5976 ~~Inherited: False~~

5977 ~~Property Type: icom_core:Person~~

5978 ~~Cardinality: Single~~

5979 ~~Updatability: On Create~~

5980

5981 **icom_core:timeZone**

5982 Description: Time zone of a person.

5983 Required: False

5984 Inherited: False

5985 Property Type: icom_core:TimeZone

5986 Cardinality: MultiSingle

5987 Updatability: Read Write

5988

5989 ~~icom_content:attachment~~

5990 ~~Description: One or more simple content attachments in a contact.~~

5991 ~~Required: False~~

5992 ~~Inherited: False~~

5993 ~~Property Type: icom_content:Attachment~~

5994 ~~Cardinality: Multi~~

5995 ~~Updatability: Read Write~~

5996

5997 **icom_core:givenName**

5998 Description: Given name of a person.

5999	Required:	False
6000	Inherited:	False
6001	Property Type:	String
6002	Cardinality:	Single
6003	Updatability:	Read Write
6004		
6005	icom_core:middleName	
6006	Description:	Middle name of a person. Can include multiple names concatenated.
6007		
6008	Required:	False
6009	Inherited:	False
6010	Property Type:	String
6011	Cardinality:	Single
6012	Updatability:	Read Write
6013		
6014	icom_core:familyName	
6015	Description:	Family name of a person.
6016	Required:	False
6017	Inherited:	False
6018	Property Type:	String
6019	Cardinality:	Single
6020	Updatability:	Read Write
6021		
6022	icom_core:prefix	
6023	Description:	Prefix of a person's name.
6024	Required:	False
6025	Inherited:	False
6026	Property Type:	String
6027	Cardinality:	Single
6028	Updatability:	Read Write
6029		
6030	icom_core:suffix	
6031	Description:	Suffix of a person's name.
6032	Required:	False
6033	Inherited:	False
6034	Property Type:	String
6035	Cardinality:	Single
6036	Updatability:	Read Write
6037		
6038	icom_core:nickname	
6039	Description:	Nickname of a person.
6040	Required:	False
6041	Inherited:	False

6042	Property Type:	String
6043	Cardinality:	Single <u>Multi</u>
6044	Updatability:	Read Write
6045		
6046	icom_core:jobTitle	
6047	Description:	Job title of a person.
6048	Required:	False
6049	Inherited:	False
6050	Property Type:	String
6051	Cardinality:	Single
6052	Updatability:	Read Write
6053		
6054	icom_core:department	
6055	Description:	A person's affiliated department.
6056	Required:	False
6057	Inherited:	False
6058	Property Type:	String
6059	Cardinality:	Single
6060	Updatability:	Read Write
6061		
6062	icom_core:officeLocation	
6063	Description:	Location of a person's department.
6064	Required:	False
6065	Inherited:	False
6066	Property Type:	String
6067	Cardinality:	Single
6068	Updatability:	Read Write
6069		
6070	icom_core:company	
6071	Description:	A person's affiliated company.
6072	Required:	False
6073	Inherited:	False
6074	Property Type:	String
6075	Cardinality:	Single
6076	Updatability:	Read Write
6077		
6078	icom_core:profession	
6079	Description:	A person's profession.
6080	Required:	False
6081	Inherited:	False
6082	Property Type:	String
6083	Cardinality:	Single

6084 Updatability: Read Write

6085

6086 icom_content:attachment

6087 Description: One or more content attachments in a contact.

6088 Required: False

6089 Inherited: False

6090 Property Type: icom_content:Attachment

6091 Cardinality: Multi

6092 Updatability: Read Write

6093

6094 icom_card:bookmark

6095 Description: A person which is bookmarked by a contact.

6096 Required: False

6097 Inherited: False

6098 Property Type: icom_core:Person

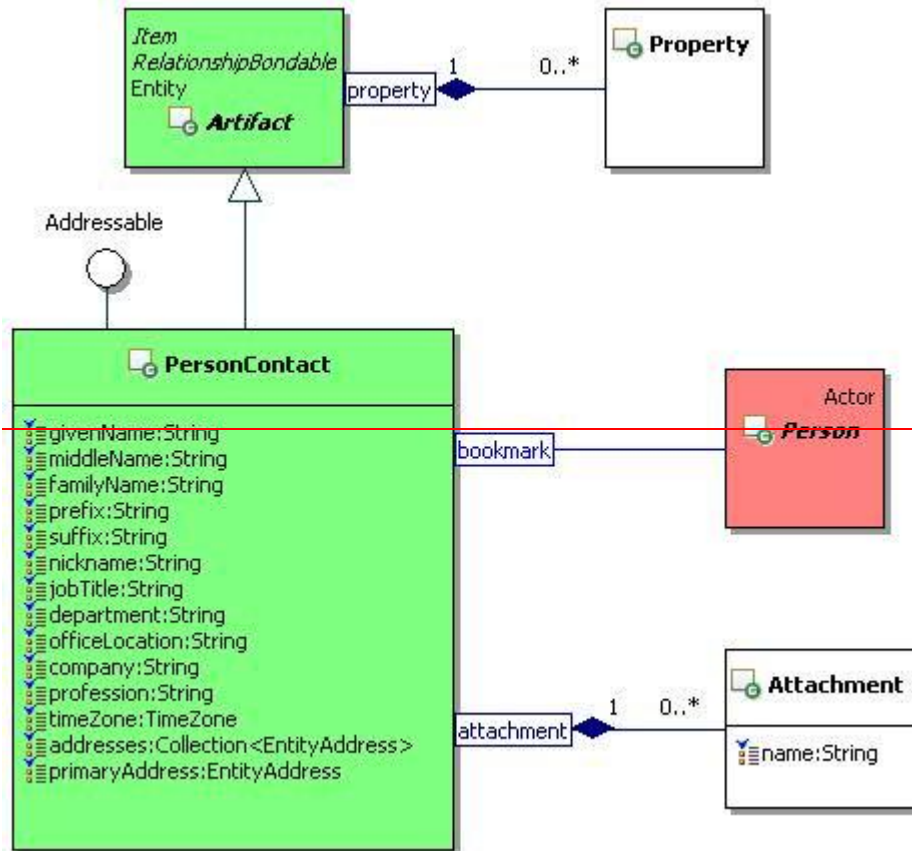
6099 Cardinality: Single

6100 Updatability: On Create

6101

6102 The PersonContact class MAY include additional property definitions which are implementation-defined.

6103



6104

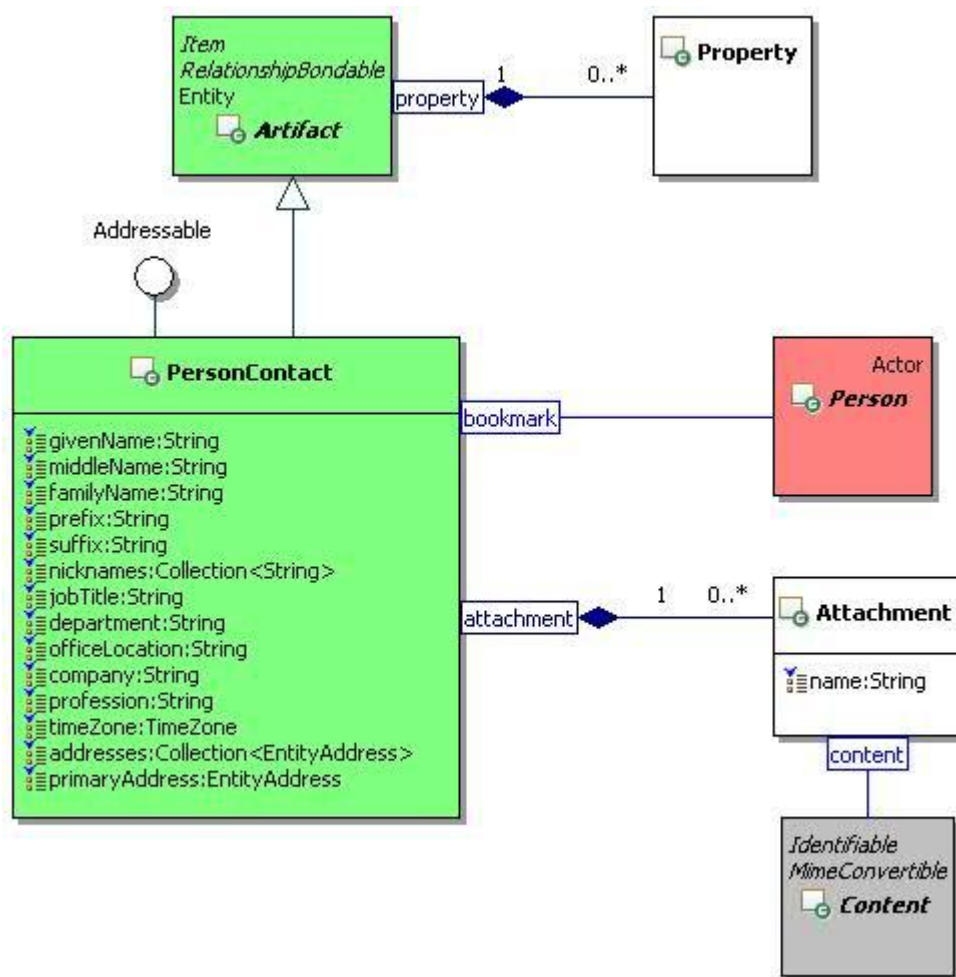


Figure 34-~~PersonContact~~: Person Contact Class Diagram.

4.7 Calendar Module

4.7.1 Calendar

4.7.1.1 Description

A calendar ~~is a folder that~~ contains time management artifacts ~~such as that include~~ occurrences and occurrence series.

4.7.1.2 Class Definition

4.7.1.2.1.1.1 Class Definition

The Calendar class ~~is defined by the~~has attribute values:

~~localNamespace~~

~~Value: icom_cal~~

~~localName~~

6121

6122

6123 localNamespace

6124 Value: icom_cal

6125

6126 localName

6127 Value: Calendar

6128

6129 **extendsFrom**

6130 Value: icom_core:Folder

6131

6132 stereotype

6133 Value: primary

6134

6135 description

6136

6137 ~~stereotype~~

6138 ~~Value: primary~~

6139

6140 ~~description~~

6141 Value: A calendar ~~is a folder that~~ contains time management artifacts ~~such as~~ that include

6142 occurrences and occurrence series.

6143

6144 **propertyDefinitions**

6145 The values for this attribute are defined in 4.7.1.3.

6146 4.7.1.3 Property Definitions

6147 The Calendar class inherits property definitions from super classes.

6148 The Calendar class MUST have the property definitions:

6149

6150 **icom_core:timeZone**

6151 Description: Time zone ~~of~~ setting for a calendar.

6152 Required: True

6153 Inherited: False

6154 Property Type: icom_core:TimeZone

6155 Cardinality: Single

6156 Updatability: Read Write

6157

6158 **icom_core:element**

6159 Description: Elements of a calendar.

6160 Required: False

6161 Inherited: True

6162 Property Type: icom_cal:Occurrence

6163	Cardinality:	Multi
6164	Updatability:	Read Only
6165		
6166	icom_cal:recurrence	
6167	Description:	Recurrence elements <u>Occurrence series</u> of a calendar.
6168	Required:	False
6169	Inherited:	False
6170	Property Type:	icom_cal:OccurrenceSeries
6171	Cardinality:	Multi
6172	Updatability:	Read Only
6173		

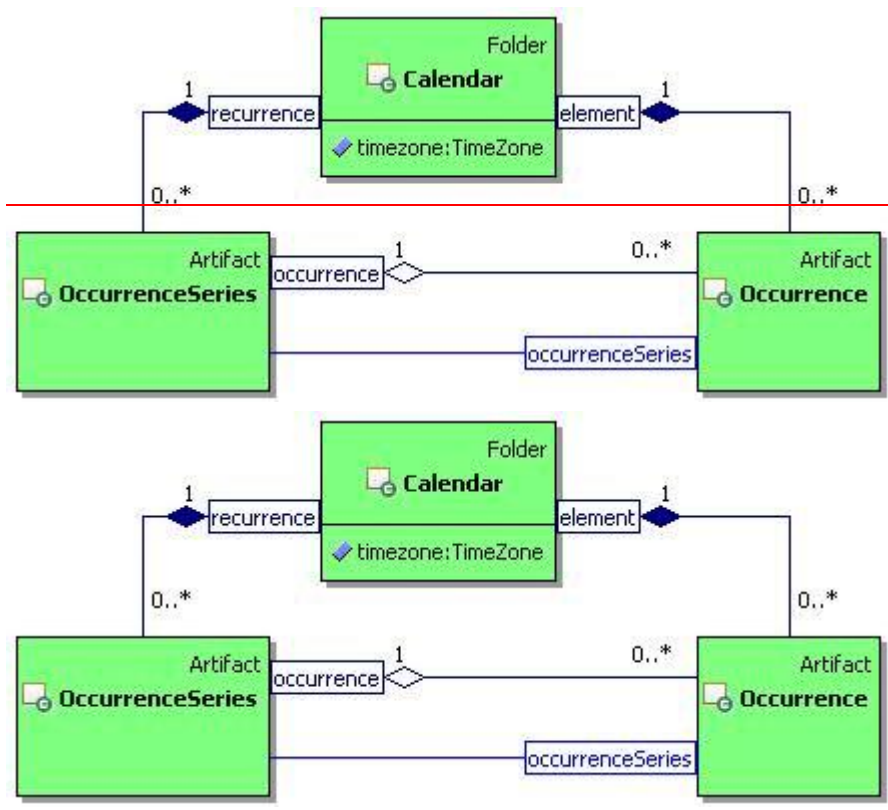


Figure 35: Calendar Class Diagram.

4.7.2 OccurrenceSeries

4.7.2.1 Description

An occurrence series ~~is an artifact that~~ represents a series of occurrences associated with the same calendar event.

4.7.2.2 Class Definition

The OccurrenceSeries class ~~is defined by the~~has attribute values:

6185 ~~localNamespace~~

6186

6187 localNamespace

6188 Value: icom_cal

6189

6190 ~~localName~~

6191

6192 localName

6193 Value: OccurrenceSeries

6194

6195 extendsFrom

6196 Value: icom_core:Artifact

6197

6198 stereotype

6199 Value: primary

6200

6201 description

6202

6203 ~~extendsFrom~~

6204 ~~Value: icom_core:Artifact~~

6205

6206 ~~stereotype~~

6207 ~~Value: primary~~

6208

6209 ~~description~~

6210 Value: An occurrence series ~~is an artifact that~~ represents a series of occurrences associated

6211 with the same calendar event.

6212

6213 **propertyDefinitions**

6214 The values for this attribute are defined in 4.7.2.3.

6215 4.7.2.3 Property Definitions

6216 The OccurrenceSeries class inherits property definitions from super classes.

6217 The OccurrenceSeries class MUST have the property definitions:

6218

6219 icom_core:location

6220 Description: Location of an occurrence series.

6221 Required: False

6222 Inherited: False

6223 Property Type: icom_core:Location

6224 Cardinality: Single

6225 Updatability: Read Write

6226

6227	<u>icom_core:organizer</u>	
6228	<u>Description:</u>	<u>Organizer of an occurrence series.</u>
6229	<u>Required:</u>	<u>True</u>
6230	<u>Inherited:</u>	<u>False</u>
6231	<u>Property Type:</u>	<u>icom_core:Participant</u>
6232	<u>Cardinality:</u>	<u>Single</u>
6233	<u>Updatability:</u>	<u>On Create</u>
6234		
6235	<u>icom_core:participant</u>	
6236	<u>Description:</u>	<u>Participants <u>in an occurrence series.</u></u>
6237	<u>Required:</u>	<u>False</u>
6238	<u>Inherited:</u>	<u>False</u>
6239	<u>Property Type:</u>	<u>icom_cal:OccurrenceParticipant</u>
6240	<u>Cardinality:</u>	<u>Multi</u>
6241	<u>Updatability:</u>	<u>Read Write</u>
6242		
6243	<u>icom_core:priority</u>	
6244	<u>Description:</u>	<u>Priority for an attendee of an occurrence series.</u>
6245	<u>Required:</u>	<u>False</u>
6246	<u>Inherited:</u>	<u>False</u>
6247	<u>Property Type:</u>	<u>icom_core:Priority</u>
6248	<u>Cardinality:</u>	<u>Single</u>
6249	<u>Updatability:</u>	<u>Read Write</u>
6250		
6251	<u>icom_content:attachment</u>	
6252	<u>Description:</u>	<u>One or more <u>content attachments in an occurrence series.</u></u>
6253	<u>Required:</u>	<u>False</u>
6254	<u>Inherited:</u>	<u>False</u>
6255	<u>Property Type:</u>	<u>icom_content:Attachment</u>
6256	<u>Cardinality:</u>	<u>Multi</u>
6257	<u>Updatability:</u>	<u>Read Write</u>
6258		
6259	icom_cal:recurrenceStartDate	
6260	<u>Description:</u>	<u>Start date and time of an occurrence series.</u>
6261	<u>Required:</u>	<u>True</u>
6262	<u>Inherited:</u>	<u>False</u>
6263	<u>Property Type:</u>	<u>DateTime</u>
6264	<u>Cardinality:</u>	<u>Single</u>
6265	<u>Updatability:</u>	<u>On Create</u>
6266		
6267	icom_cal:recurrenceStartDateResolution	
6268	<u>Description:</u>	<u>Resolution of start date and time of an occurrence series.</u>

6269	Required:	True
6270	Inherited:	False
6271	Property Type:	icom_core:DateTimeResolution
6272	Cardinality:	Single
6273	Updatability:	On Create
6274		
6275	icom_cal:duration	
6276	Description:	Duration of each occurrence in an occurrence series.
6277	Required:	True
6278	Inherited:	False
6279	Property Type:	Duration
6280	Cardinality:	Single
6281	Updatability:	On Create
6282		
6283	icom_cal:recurrenceRule	
6284	Description:	A recurrence rule of an occurrence series.
6285	Required:	True
6286	Inherited:	False
6287	Property Type:	String
6288	Cardinality:	Single
6289	Updatability:	On Create
6290		
6291	icom_core:location	
6292	Description:	Location of an occurrence series.
6293	Required:	False
6294	Inherited:	False
6295	Property Type:	icom_core:Location
6296	Cardinality:	Single
6297	Updatability:	Read Write
6298		
6299	icom_core:organizer	
6300	Description:	Organizer of an occurrence series.
6301	Required:	True
6302	Inherited:	False
6303	Property Type:	icom_core:Participant
6304	Cardinality:	Single
6305	Updatability:	On Create
6306		
6307	icom_core:participant	
6308	Description:	Participants of an occurrence series.
6309	Required:	False
6310	Inherited:	False

6311 ~~Property Type: icom_cal:OccurrenceParticipant~~
6312 ~~Cardinality: Multi~~
6313 ~~Updatability: Read Write~~

6314

6315 **icom_cal:occurrenceStatus**

6316 Description: Status of an occurrence series.
6317 Required: True
6318 Inherited: False
6319 Property Type: icom_cal:OccurrenceStatus
6320 Cardinality: Single
6321 Updatability: Read Write

6322

6323 **icom_cal:occurrenceType**

6324 Description: Type of an occurrence series.
6325 Required: True
6326 Inherited: False
6327 Property Type: icom_cal:OccurrenceType
6328 Cardinality: Single
6329 Updatability: Read Write

6330

6331 **icom_cal:~~mode~~editMode**

6332 Description: ~~Mutability~~Indicates a mode ~~of which determines whether~~ an
6333 occurrence series ~~is editable~~.
6334 Required: False
6335 Inherited: False
6336 Property Type: icom_cal:OccurrenceEditMode
6337 Cardinality: Single
6338 Updatability: Read Only

6339

6340 **icom_cal:occurrence**

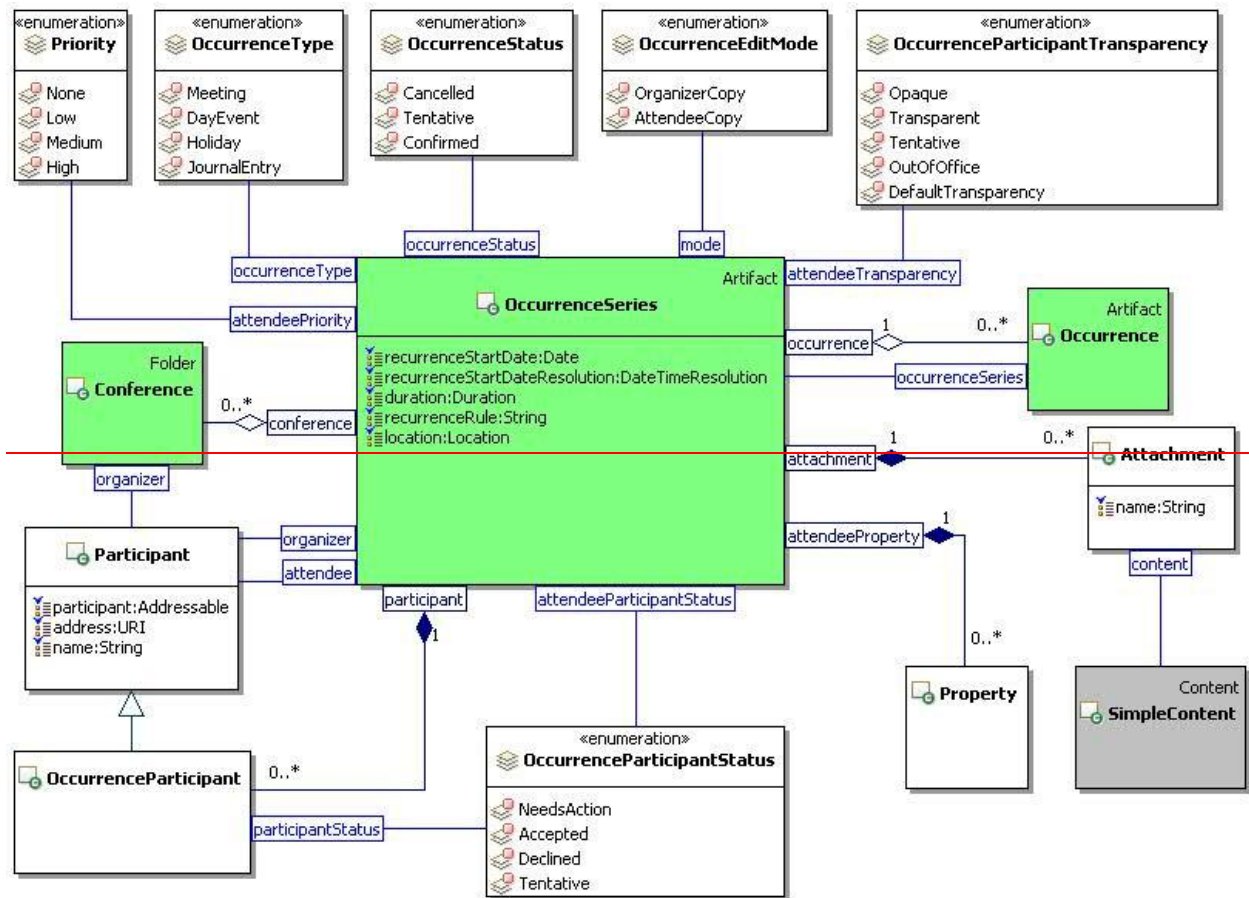
6341 Description: Occurrences in an occurrence series.
6342 Required: False
6343 Inherited: False
6344 Property Type: icom_cal:Occurrence
6345 Cardinality: Multi
6346 Updatability: Read Only

6347

6348 ~~icom_content:attachment~~

6349 ~~Description: One or more simple content attachments in an occurrence~~
6350 ~~series.~~
6351 ~~Required: False~~
6352 ~~Inherited: False~~
6353 ~~Property Type: icom_content:Attachment~~

6397		
6398	icom_conf:conference	
6399	Description:	One or more conferences for an occurrence series.
6400	Required:	False
6401	Inherited:	False
6402	Property Type:	icom_conf:Conference
6403	Cardinality:	Multi
6404	Updatability:	Read Write
6405		



6406

6419

6420 localNamespace

6421 Value: icom_cal

6422

6423 localName

6424

6425 ~~localNamespace~~

6426 ~~Value: _cal~~

6427

6428 ~~localName~~

6429 Value: Occurrence

6430

6431 ~~extendsFrom~~

6432 ~~Value: icom_core:Artifact~~

6433

6434 ~~stereotype~~

6435 ~~Value: primary~~

6436

6437 ~~description~~

6438

6439 extendsFrom

6440 Value: icom_core:Artifact

6441

6442 stereotype

6443 Value: primary

6444

6445 description

6446 Value: An occurrence ~~is an artifact that~~ represents an event in a calendar.

6447

6448 **propertyDefinitions**

6449 The values for this attribute are defined in 4.7.3.3.

6450

6451 4.7.3.3 Property Definitions

6452 The Occurrence class inherits property definitions from super classes.

6453 The Occurrence class MUST have the property definitions:

6454

6455 ~~icom_cal~~icom_core:location

6456 Description: Location of an occurrence.

6457 Required: False

6458 Inherited: False

6459 Property Type: icom_core:Location

6460 Cardinality: Single

6461	<u>Updatability:</u>	<u>Read Write</u>
6462		
6463	<u>icom_core:organizer</u>	
6464	<u>Description:</u>	<u>Organizer of an occurrence.</u>
6465	<u>Required:</u>	<u>True</u>
6466	<u>Inherited:</u>	<u>False</u>
6467	<u>Property Type:</u>	<u>icom_core:Participant</u>
6468	<u>Cardinality:</u>	<u>Single</u>
6469	<u>Updatability:</u>	<u>On Create</u>
6470		
6471	<u>icom_core:participant</u>	
6472	<u>Description:</u>	<u>Participants of an occurrence.</u>
6473	<u>Required:</u>	<u>False</u>
6474	<u>Inherited:</u>	<u>False</u>
6475	<u>Property Type:</u>	<u>icom_cal:OccurrenceParticipant</u>
6476	<u>Cardinality:</u>	<u>Multi</u>
6477	<u>Updatability:</u>	<u>Read Write</u>
6478		
6479	<u>icom_core:priority</u>	
6480	<u>Description:</u>	<u>Priority for an attendee of an occurrence.</u>
6481	<u>Required:</u>	<u>False</u>
6482	<u>Inherited:</u>	<u>False</u>
6483	<u>Property Type:</u>	<u>icom_core:Priority</u>
6484	<u>Cardinality:</u>	<u>Single</u>
6485	<u>Updatability:</u>	<u>Read Write</u>
6486		
6487	<u>icom_core:startDate</u>	
6488	Description:	Start date and time of an occurrence.
6489	Required:	True
6490	Inherited:	False
6491	Property Type:	DateTime
6492	Cardinality:	Single
6493	Updatability:	On Create
6494		
6495	<u>icom_calcore:startDateResolution</u>	
6496	Description:	Resolution of start date and time of an occurrence.
6497	Required:	True
6498	Inherited:	False
6499	Property Type:	icom_core:DateTimeResolution
6500	Cardinality:	Single
6501	Updatability:	On Create
6502		

6503	icom_cal:core:endDate	
6504	Description:	End date and time of an occurrence.
6505	Required:	True
6506	Inherited:	False
6507	Property Type:	DateTime
6508	Cardinality:	Single
6509	Updatability:	On Create
6510		
6511	icom_cal:core:endDateResolution	
6512	Description:	Resolution of end date and time of an occurrence.
6513	Required:	True
6514	Inherited:	False
6515	Property Type:	icom_core:DateTimeResolution
6516	Cardinality:	Single
6517	Updatability:	On Create
6518		
6519	<u>icom_content:attachment</u>	
6520	<u>Description:</u>	<u>One or more content attachments in an occurrence.</u>
6521	<u>Required:</u>	<u>False</u>
6522	<u>Inherited:</u>	<u>False</u>
6523	<u>Property Type:</u>	<u>icom_content:Attachment</u>
6524	<u>Cardinality:</u>	<u>Multi</u>
6525	<u>Updatability:</u>	<u>Read Write</u>
6526		
6527	<u>icom_core:location</u>	
6528	<u>Description:</u>	<u>Location of an occurrence.</u>
6529	<u>Required:</u>	<u>False</u>
6530	<u>Inherited:</u>	<u>False</u>
6531	<u>Property Type:</u>	<u>icom_core:Location</u>
6532	<u>Cardinality:</u>	<u>Single</u>
6533	<u>Updatability:</u>	<u>Read Write</u>
6534		
6535	icom_cal:occurrenceSeries	
6536	Description:	An occurrence <u>is part of this occurrence</u> series that includes
6537		an occurrence.
6538	Required:	False
6539	Inherited:	False
6540	Property Type:	icom_cal:OccurrenceSeries
6541	Cardinality:	Single
6542	Updatability:	Read Only
6543		
6544	icom_cal:fromRecurringOccurrenceSeries	
6545	Description:	Occurrence is part of a recurring occurrence series.

6546	Required:	False
6547	Inherited:	False
6548	Property Type:	Boolean
6549	Cardinality:	Single
6550	Updatability:	Read Only

6551

6552 **icom_cal:exceptionToOccurrenceSeries**

6553	Description:	Occurrence is an exception to an occurrence series.
6554	Required:	False
6555	Inherited:	False
6556	Property Type:	Boolean
6557	Cardinality:	Single
6558	Updatability:	Read Only

6559

6560 ~~icom_core:organizer~~

6561	Description:	Organizer of an occurrence.
6562	Required:	True
6563	Inherited:	False
6564	Property Type:	icom_core:Participant
6565	Cardinality:	Single
6566	Updatability:	On Create

6567

6568 ~~icom_core:participant~~

6569	Description:	Participants of an occurrence.
6570	Required:	False
6571	Inherited:	False
6572	Property Type:	icom_cal:OccurrenceParticipant
6573	Cardinality:	Multi
6574	Updatability:	Read Write

6575

6576 **icom_cal:occurrenceStatus**

6577	Description:	Status of an occurrence.
6578	Required:	True
6579	Inherited:	False
6580	Property Type:	icom_cal:OccurrenceStatus
6581	Cardinality:	Single
6582	Updatability:	Read Write

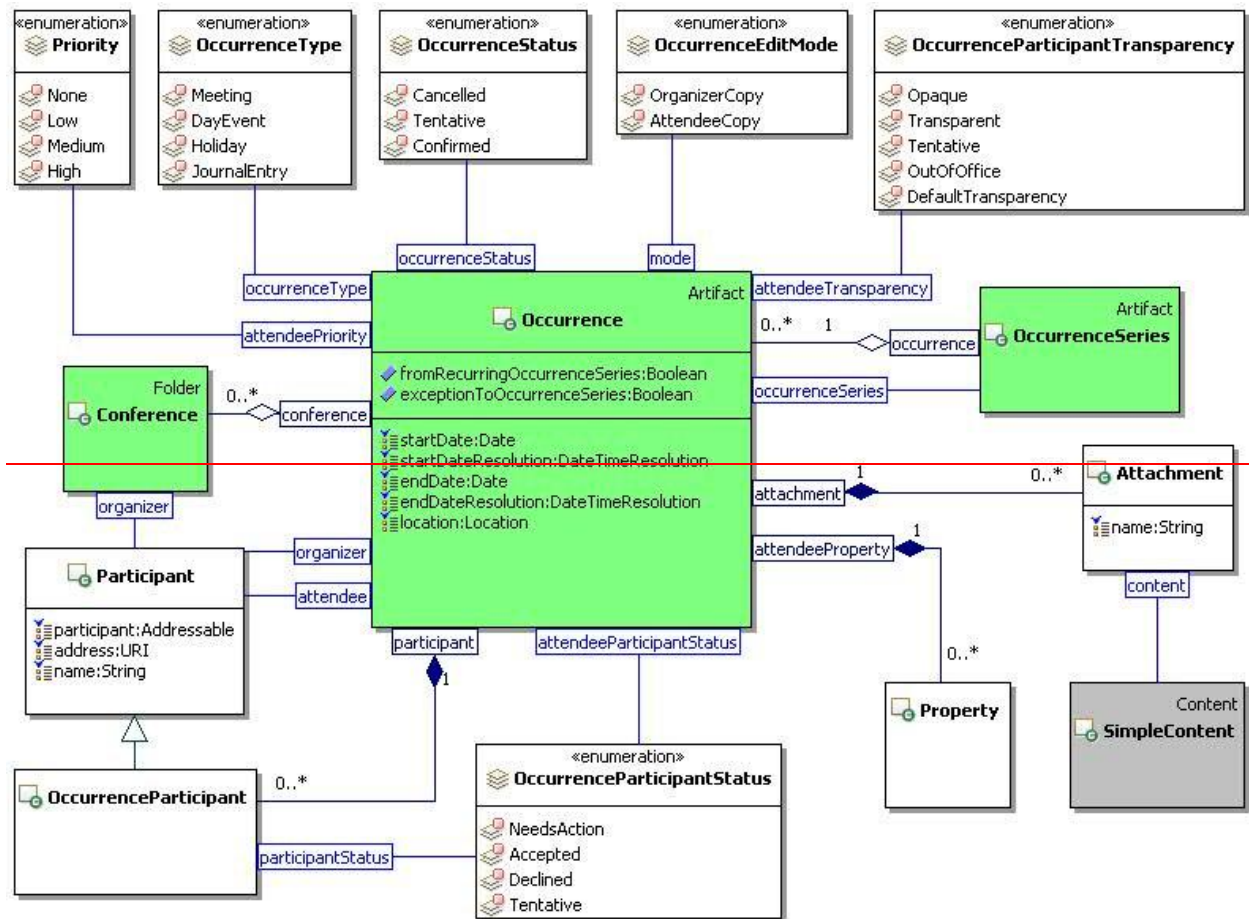
6583

6584 **icom_cal:occurrenceType**

6585	Description:	Type of an occurrence.
6586	Required:	True
6587	Inherited:	False

6588	Property Type:	icom_cal:OccurrenceType
6589	Cardinality:	Single
6590	Updatability:	Read Write
6591		
6592	icom_cal:modeeditMode	
6593	Description:	Mutability Indicates a mode of which determines whether an occurrence <u>is editable</u> .
6594		
6595	Required:	False
6596	Inherited:	False
6597	Property Type:	icom_cal:OccurrenceEditMode
6598	Cardinality:	Single
6599	Updatability:	Read Only
6600		
6601	icom_content:attachment	
6602	Description:	One or more simple content attachments in an occurrence.
6603	Required:	False
6604	Inherited:	False
6605	Property Type:	icom_content:Attachment
6606	Cardinality:	Multi
6607	Updatability:	Read Write
6608		
6609	icom_cal:attende	
6610	Description:	An attendee of an occurrence.
6611	Required:	False
6612	Inherited:	False
6613	Property Type:	icom_core:Participant
6614	Cardinality:	Single
6615	Updatability:	Read Only
6616		
6617	icom_cal:attendePriority	
6618	Description:	Priority for an attendee of an occurrence.
6619	Required:	False
6620	Inherited:	False
6621	Property Type:	icom_core:Priority
6622	Cardinality:	Single
6623	Updatability:	Read Write
6624		
6625	icom_cal:attendeParticipantStatus	
6626	Description:	Participation status <u>Status</u> for an attendee of an occurrence.
6627	Required:	False
6628	Inherited:	False
6629	Property Type:	icom_cal:OccurrenceParticipantStatus
6630	Cardinality:	Single

6631	Updatability:	Read Write
6632		
6633	icom_cal:attendeeTransparency	
6634	Description:	Participant transparency <u>Transparency</u> for an attendee of an
6635		occurrence.
6636	Required:	False
6637	Inherited:	False
6638	Property Type:	icom_cal:OccurrenceParticipantTransparency
6639	Cardinality:	Single
6640	Updatability:	Read Write
6641		
6642	icom_cal:attendeeProperty	
6643	Description:	Extensible properties for an attendee of an occurrence.
6644	Required:	False
6645	Inherited:	False
6646	Property Type:	icom_meta:Property
6647	Cardinality:	Multi
6648	Updatability:	Read Write
6649		
6650	icom_conf:conference	
6651	Description:	One or more conferences for an occurrence.
6652	Required:	False
6653	Inherited:	False
6654	Property Type:	icom_conf:Conference
6655	Cardinality:	Multi
6656	Updatability:	Read Write
6657		



6658

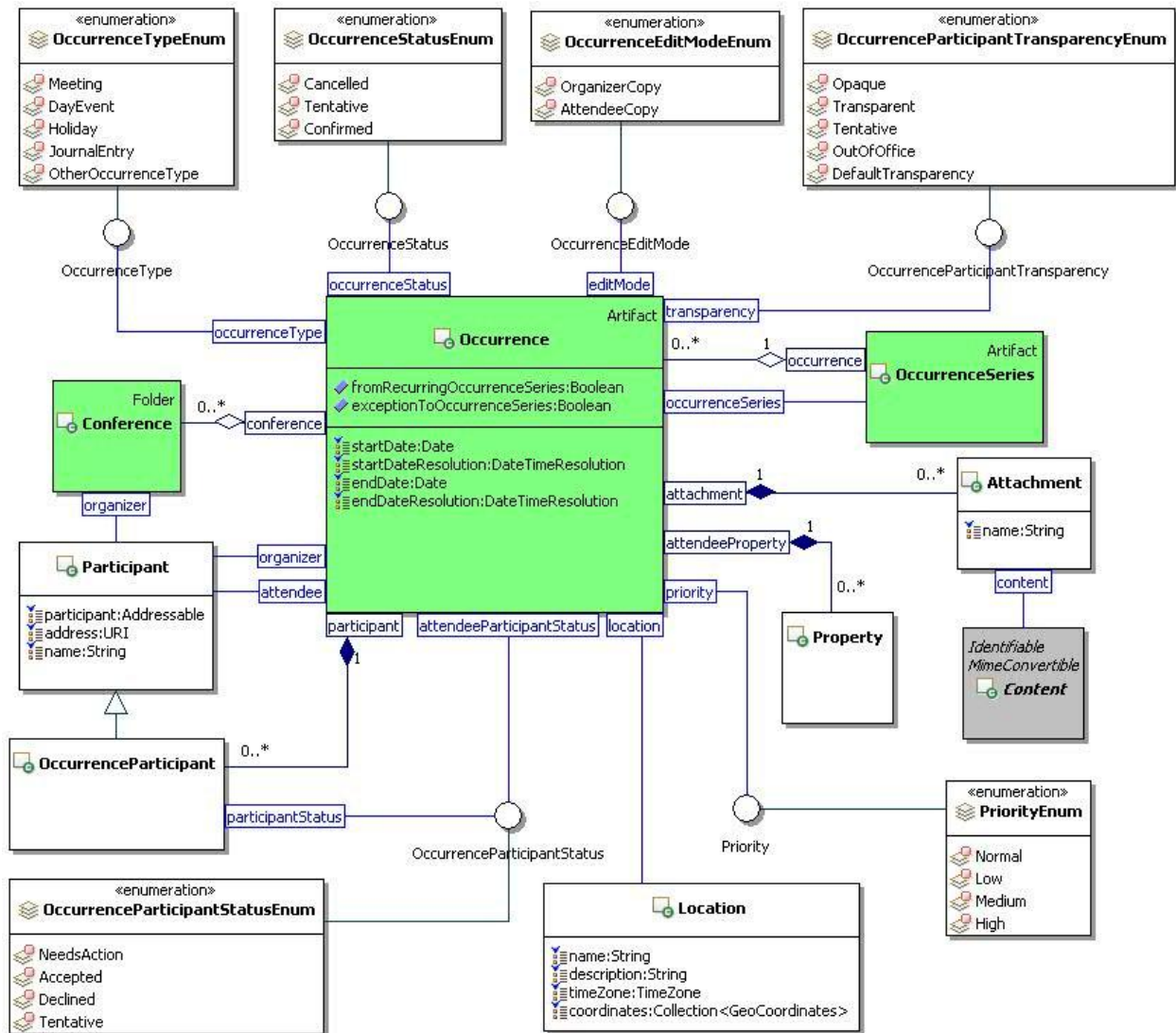


Figure 37: Occurrence Class Diagram.

4.7.4 OccurrenceStatus

4.7.4.1 Description

An occurrence status is a status of a calendar occurrence.

4.7.4.2 Class Definition

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

The OccurrenceStatus class has attribute values:

localNamespace

Value: icom cal

localName

6673 Value: OccurrenceStatus

6674

6675 extendsFrom

6676 Value:

6677

6678 stereotype

6679 Value: mixin

6680

6681 description

6682 Value: OccurrenceStatus is a mixin class which defines status of a calendar occurrence.

6683

6684 propertyDefinitions

6685 The values for this attribute are defined in Section 4.7.4.3.

6686 **4.7.4.3 Property Definitions**

6687 The OccurrenceStatus class MAY include additional property definitions which are implementation-

6688 defined.

6689

6690 **4.7.5 OccurrenceStatusEnum**

6691 The OccurrenceStatusEnum class is an enum class that enumerates the instances each of which

6692 expresses a status of an occurrence or occurrence series.

6693 The OccurrenceStatusEnum class ~~is defined by the~~has attribute values:

6694

6695 **localNamespace**

6696 Value: icom_cal

6697

6698 **localName**

6699 Value: OccurrenceStatusEnum

6700

6701 extendsFrom

6702 Value: OccurrenceStatus

6703

6704 ~~extendsFrom~~

6705 Value:

6706

6707 **stereotype**

6708 Value: primary

6709

6710 **isEnumeration**

6711 Value: TRUE

6712

6713 **description**

6714 | ~~Value: An enumeration of the instances each of which expresses a status~~Value: Status of an
6715 | occurrence or occurrence series.

6716

6717 | **instances**

6718 | Value: <icom_cal:Cancelled, icom_cal:Tentative, icom_cal:Confirmed>

6719

6720 | ~~There are~~ICOM defines three occurrence status~~defined by ICOM:~~

- 6721 | • **icom_cal:Cancelled** ~~to express that~~ an occurrence or occurrence series is cancelled.
- 6722 | • **icom_cal:Tentative** ~~to express that~~ an occurrence or occurrence series is tentative.
- 6723 | • **icom_cal:Confirmed** ~~to express that~~ an occurrence or occurrence series is confirmed.

6724

6725 | ~~4.7.5~~4.7.6 OccurrenceType

6726 | 4.7.6.1 Description

6727 | An occurrence type is a category of calendar occurrences.

6728 | 4.7.6.2 Class Definition

6729 | The OccurrenceType class is a mixin class which defines a type of occurrence.

6730 | The OccurrenceType class has attribute values:

6731

6732 | localNamespace

6733 | Value: icom_cal

6734

6735 | localName

6736 | Value: OccurrenceType

6737

6738 | extendsFrom

6739 | Value:

6740

6741 | stereotype

6742 | Value: mixin

6743

6744 | description

6745 | Value: OccurrenceType is a mixin class which defines a type of occurrence.

6746

6747 | propertyDefinitions

6748 | The values for this attribute are defined in Section 4.7.6.3.

6749 | 4.7.6.3 Property Definitions

6750 | The OccurrenceType class MAY include additional property definitions which are implementation-defined.

6751

4.7.7 OccurrenceTypeEnum

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

The OccurrenceTypeEnum class ~~is defined by the~~has attribute values:

localNamespace

Value: icom_cal

localName

Value: OccurrenceTypeEnum

extendsFrom

Value: OccurrenceType

~~extendsFrom~~

~~Value:~~

stereotype

Value: primary

isEnumeration

Value: TRUE

description

~~Value: An enumeration of the instances each of which expresses a type~~Value: Type of an occurrence or occurrence series.

instances

Value: <icom_cal:Meeting, icom_cal:DayEvent, icom_cal:Holiday, icom_cal:JournalEntry, icom_cal:OtherOccurrenceType>

~~There are four~~ICOM defines five occurrence types ~~defined by ICOM:~~

- **icom_cal:Meeting** ~~to express that~~ an occurrence or occurrence series is a meeting.
- **icom_cal:DayEvent** ~~to express that~~ an occurrence or occurrence series is a day event.
- **icom_cal:Holiday** ~~to express that~~ an occurrence or occurrence series is a holiday.
- **icom_cal:JournalEntry** ~~to express that~~ an occurrence or occurrence series is a journal entry.
- **icom_cal:OtherOccurrenceType** an occurrence or occurrence series is of other type.

4.7.6.14.7.8 OccurrenceParticipant

4.7.6.14.7.8.1 Description

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

4.7.6.24.7.8.2 Class Definition

The OccurrenceParticipant class ~~is defined by the~~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.

~~4.7.6.34~~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

~~4.7.74~~4.7.9 OccurrenceParticipantStatus

4.7.9.1 Description

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

4.7.9.2 Class Definition

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

The OccurrenceParticipantStatus class has attribute values:

6833
6834 localNamespace
6835 Value: icom_cal
6836
6837 localName
6838 Value: OccurrenceParticipantStatus
6839
6840 extendsFrom
6841 Value:
6842
6843 stereotype
6844 Value: mixin
6845
6846 description
6847 Value: OccurrenceParticipantStatus is a mixin class which defines a participant's response
6848 status for an occurrence or occurrence series.
6849
6850 propertyDefinitions
6851 The values for this attribute are defined in Section 4.7.9.3.

6852 **4.7.9.3 Property Definitions**

6853 The OccurrenceParticipantStatus class MAY include additional property definitions which are
6854 implementation-defined.
6855

6856 **4.7.10 OccurrenceParticipantStatusEnum**

6857 The OccurrenceParticipantStatusEnum class is an enum class that enumerates the instances each of
6858 which expresses a participant's response status for an occurrence or occurrence series.

6859 The OccurrenceParticipantStatusEnum class has attribute values:

6860
6861 localNamespace
6862 Value: icom_cal
6863
6864 localName
6865 Value: OccurrenceParticipantStatusEnum
6866
6867 extendsFrom
6868 Value: OccurrenceParticipantStatus~~class is defined by the attribute values:~~
6869
6870 ~~localNamespace~~
6871 ~~Value: icom_cal~~
6872
6873 ~~localName~~
6874 ~~Value: OccurrenceParticipantStatus~~

6875

6876 **extendsFrom**

6877 **Value:**

6878

6879 **stereotype**

6880 **Value:**

6881

6882 **stereotype**

6883 **Value:** primary

6884

6885 **isEnumeration**

6886 **Value:** TRUE

6887

6888 **description**

6889 **Value:** ~~An enumeration of the instances each of which expresses a participant's~~ **Value:**

6890 **Participant's** response status for an occurrence or occurrence series.

6891

6892 **instances**

6893 **Value:** <icom_cal:NeedsAction, icom_cal:Accepted, icom_cal:Declined, icom_cal:Tentative>

6894

6895 ~~There are~~ **ICOM defines** four occurrence participant's status ~~defined by ICOM:~~

6896

6897

- 6898 • **icom_cal:NeedsAction** ~~to express that~~ an attendee needs to act on an occurrence or
- 6899 occurrence series.
- 6900 • **icom_cal:Accepted** ~~to express that~~ an attendee accepted an occurrence or occurrence series.
- 6901 • **icom_cal:Declined** ~~to express that~~ an attendee declined an occurrence or occurrence series.
- 6902 • **icom_cal:Tentative** ~~to express that~~ an attendee is tentative about attending an occurrence or

6903

6904 **4.7.84.7.11 OccurrenceParticipantTransparency**

6905

6906 **4.7.11.1 Description**

6907 **An occurrence participant transparency is visibility of an occurrence or occurrence series in a participant's**

6908 **calendar or free busy.**

6909

6910 **4.7.11.2 Class Definition**

6911

6912 **The OccurrenceParticipantTransparency class is a mixin class which defines visibility of an occurrence or**

6913 **occurrence series in a participant's calendar or free busy.**

6914 **The OccurrenceParticipantTransparency class has attribute values:**

6915

6916 **localNamespace**

6917 **Value:** icom_cal

6918

6919 **localName**

6916 Value: OccurrenceParticipantTransparency

6917

6918 extendsFrom

6919 Value:

6920

6921 stereotype

6922 Value: mixin

6923

6924 description

6925 Value: OccurrenceParticipantTransparency is a mixin class which defines visibility of an

6926 occurrence or occurrence series in a participant's calendar or free busy.

6927

6928 propertyDefinitions

6929 The values for this attribute are defined in Section 4.7.11.3.

6930 **4.7.11.3 Property Definitions**

6931 The OccurrenceParticipantTransparency class MAY include additional property definitions which are

6932 implementation-defined.

6933

6934 **4.7.12 OccurrenceParticipantTransparencyEnum**

6935 The OccurrenceParticipantTransparencyEnum class is an enum class that enumerates the instances

6936 each of which expresses an occurrence or occurrence series transparency in a participant's calendar or

6937 free busy.

6938 The OccurrenceParticipantTransparencyEnum class has attribute values:

6939

6940 localNamespace

6941 Value: icom_cal

6942

6943 localName

6944 Value: OccurrenceParticipantTransparencyEnum

6945

6946 extendsFrom

6947 Value: OccurrenceParticipantTransparency ~~class is defined by the attribute values:~~

6948

6949 ~~localNamespace~~

6950 ~~Value: icom_cal~~

6951

6952 ~~localName~~

6953 ~~Value: OccurrenceParticipantTransparency~~

6954

6955 ~~extendsFrom~~

6956 ~~Value:~~

6957

6958 **stereotype**
 6959 **Value:**
 6960
 6961 **stereotype**
 6962 **Value:** primary
 6963
 6964 **isEnumeration**
 6965 Value: TRUE
 6966
 6967 **description**
 6968 Value: ~~An enumeration of the instances each of which expresses an occurrence~~Occurrence or
 6969 occurrence series transparency in a participant's calendar or free busy.
 6970
 6971 **instances**
 6972
 6973 **instances**
 6974 Value: <icom_cal:Opaque, icom_cal:Transparent, icom_cal:Tentative, icom_cal:OutOfOffice,
 6975 icom_cal:DefaultTransparency>
 6976
 6977 ~~There are~~ICOM defines five participant transparencies ~~defined by ICOM:~~
 6978 • **icom_cal:Opaque** ~~to express that~~ an occurrence or occurrence series is opaque in a
 6979 participant's calendar or free busy.
 6980 • **icom_cal:Transparent** ~~to express that~~ an occurrence or occurrence series is transparent in a
 6981 participant's calendar or free busy.
 6982 • **icom_cal:Tentative** ~~to express that~~ an occurrence or occurrence series has a tentative
 6983 transparency in a participant's calendar or free busy.
 6984 • **icom_cal:OutOfOffice** ~~to express that~~ an occurrence or occurrence series has out of office
 6985 transparency in a participant's calendar or free busy.
 6986 • **icom_cal:DefaultTransparency** ~~to express that~~ an occurrence or occurrence series has default
 6987 transparency in a participant's calendar or free busy.
 6988

6989 **4.7.94.7.13 OccurrenceEditMode**

6990 **4.7.13.1 Description**

6991 An occurrence edit mode is a mode that indicates whether an occurrence or occurrence series is editable.

6992 **4.7.13.2 Class Definition**

6993 The OccurrenceEditMode class is ~~an enuma mixin~~ class ~~that enumerates the instances each of which~~
 6994 ~~expresses~~defines a mode that indicates whether an ~~editable mode of an~~ occurrence or occurrence series
 6995 is editable.

6996 The OccurrenceEditMode class ~~is defined by the~~has attribute values:

6997
 6998 **localNamespace**
 6999 Value: icom_cal

7000

7001 **localName**

7002 Value: OccurrenceEditMode

7003

7004 **extendsFrom**

7005 Value:

7006

7007 **stereotype**

7008 Value: mixin

7009

7010 **description**

7011 Value: OccurrenceEditMode is a mixin class which defines a mode that indicates whether an

7012 occurrence or occurrence series is editable.

7013

7014 **propertyDefinitions**

7015 The values for this attribute are defined in Section 4.7.13.3.

7016 **4.7.13.3 Property Definitions**

7017 The OccurrenceEditMode class MAY include additional property definitions which are implementation-

7018 defined.

7019

7020 **4.7.14 OccurrenceEditModeEnum**

7021 The OccurrenceEditModeEnum class is an enum class that enumerates the instances each of which

7022 expresses a mode that indicates whether an occurrence or occurrence series is editable.

7023 The OccurrenceEditModeEnum class has attribute values:

7024

7025 **localNamespace**

7026 Value: icom_cal

7027

7028 **localName**

7029 Value: OccurrenceEditModeEnum

7030

7031 **extendsFrom**

7032 Value: OccurrenceEditMode

7033

7034 **stereotype**

7035 Value: primary

7036

7037 **isEnumeration**

7038 Value: TRUE

7039

7040 **description**

7041 | Value: ~~An enumeration of the instances each of which expresses an editable~~A mode ~~of that~~
7042 | ~~indicates whether~~ an occurrence or occurrence series is editable.

7043 |
7044 | **instances**

7045 | Value: <icom_cal:OrganizerCopy, icom_cal:AttendeeCopy>

7046 |

7047 | ~~There are~~ICOM defines two occurrence editable modes ~~defined by ICOM~~:

7048 | • **icom_cal:OrganizerCopy** ~~to express that~~ an occurrence or occurrence series is a copy created
7049 | by an organizer who may update the properties such as occurrence type, occurrence status, etc.

7050 | • **icom_cal:AttendeeCopy** ~~to express that~~ an occurrence or occurrence series is a copy delivered
7051 | to an attendee who may only update the attendee properties such as ~~attendee~~priority, ~~attendee~~
7052 | transparency, etc .

7053 |

7054 | 4.8 FreeBusyFree Busy Module

7055 | 4.8.1 FreeBusy

7056 | 4.8.1.1 Description

7057 | A free busy object specifies the free time and busy time intervals of one or more participants.

7058 | 4.8.1.2 Class Definition

7059 | The FreeBusy class ~~is defined by the~~has attribute values:

7060 |

7061 | **localNamespace**

7062 | Value: icom_cal

7063 |

7064 | **localName**

7065 | Value: FreeBusy

7066 |

7067 | **extendsFrom**

7068 | Value:

7069 |

7070 | **stereotype**

7071 | Value: primary

7072 |

7073 | **description**

7074 | Value: A free busy object specifies the free time and busy time intervals of one or more
7075 | participants.

7076 |

7077 | **propertyDefinitions**

7078 | The values for this attribute are defined in Section 4.8.1.3.

7079 | 4.8.1.3 Property Definitions

7080 | The FreeBusy class MUST have the property definitions:

7081		
7082	icom_core:participant	
7083	<u>Description:</u>	<u>A list of participants whose free busy intervals are included.</u>
7084	<u>Required:</u>	<u>False</u>
7085	<u>Inherited:</u>	<u>False</u>
7086	<u>Property Type:</u>	<u>icom_core:Participant</u>
7087	<u>Cardinality:</u>	<u>Multi</u>
7088	<u>Updatability:</u>	<u>Read Only</u>
7089		
7090	icalicom_core:creationDate	
7091	Description:	Creation date and time of a free busy object.
7092	Required:	False
7093	Inherited:	False
7094	Property Type:	DateTime
7095	Cardinality:	Single
7096	Updatability:	Read Only
7097		
7098	icom_icalcore:startDate	
7099	Description:	Start date and time of a list of free busy intervals.
7100	Required:	False
7101	Inherited:	False
7102	Property Type:	DateTime
7103	Cardinality:	Single
7104	Updatability:	Read Only
7105		
7106	icom_icalcore:endDate	
7107	Description:	End date and time of a list of free busy intervals.
7108	Required:	False
7109	Inherited:	False
7110	Property Type:	DateTime
7111	Cardinality:	Single
7112	Updatability:	Read Only
7113		
7114	icom_cal:interval	
7115	Description:	A list of free busy intervals.
7116	Required:	False
7117	Inherited:	False
7118	Property Type:	icom_cal:FreeBusyInterval
7119	Cardinality:	Multi
7120	Updatability:	Read Only
7121		
7122	icom_core:participant	

7123	Description:	A list of participants whose free busy intervals are merged.
7124	Required:	False
7125	Inherited:	False
7126	Property Type:	icom_core:Participant
7127	Cardinality:	Multi
7128	Updatability:	Read Only
7129		

7130 **4.8.2 FreeBusyInterval**

7131 **4.8.2.1 Description**

7132 | A free busy interval ~~object~~ specifies an interval of free or busy time.

7133 | If a free busy type is icom_cal:Free, then a time interval is free for scheduling.

7134 | If a free busy type is icom_cal:Busy, then a time interval is busy because one or more events have been

7135 | scheduled for the interval.

7136 **4.8.2.2 Class Definition**

7137 | The FreeBusyInterval class ~~is defined by the~~has attribute values:

7138		
7139	localNamespace	
7140	Value:	icom_cal
7141		
7142	localName	
7143	Value:	FreeBusy <u>Interval</u>
7144		
7145	extendsFrom	
7146	Value:	
7147		
7148	stereotype	
7149	Value:	primary
7150		
7151	description	
7152	Value:	A free busy interval object specifies an interval of free or busy time.
7153		
7154	propertyDefinitions	
7155		The values for this attribute are defined in Section 4.8.2.3

7156 **4.8.2.3 Property Definitions**

7157 | The FreeBusyInterval class MUST have the property definitions:

7158		
7159	icom_calcore:startDate	
7160	Description:	Start date and time of a free busy interval.
7161	Required:	False
7162	Inherited:	False

7163	Property Type:	DateTime
7164	Cardinality:	Single
7165	Updatability:	Read Only
7166		
7167	icom_calcore:endDate	
7168	Description:	End date and time of a free busy interval.
7169	Required:	False
7170	Inherited:	False
7171	Property Type:	DateTime
7172	Cardinality:	Single
7173	Updatability:	Read Only
7174		
7175	icom_cal:freeBusyType	
7176	Description:	A type of free busy interval.
7177	Required:	False
7178	Inherited:	False
7179	Property Type:	icom_cal:FreeBusyType
7180	Cardinality:	Single
7181	Updatability:	Read Only
7182		

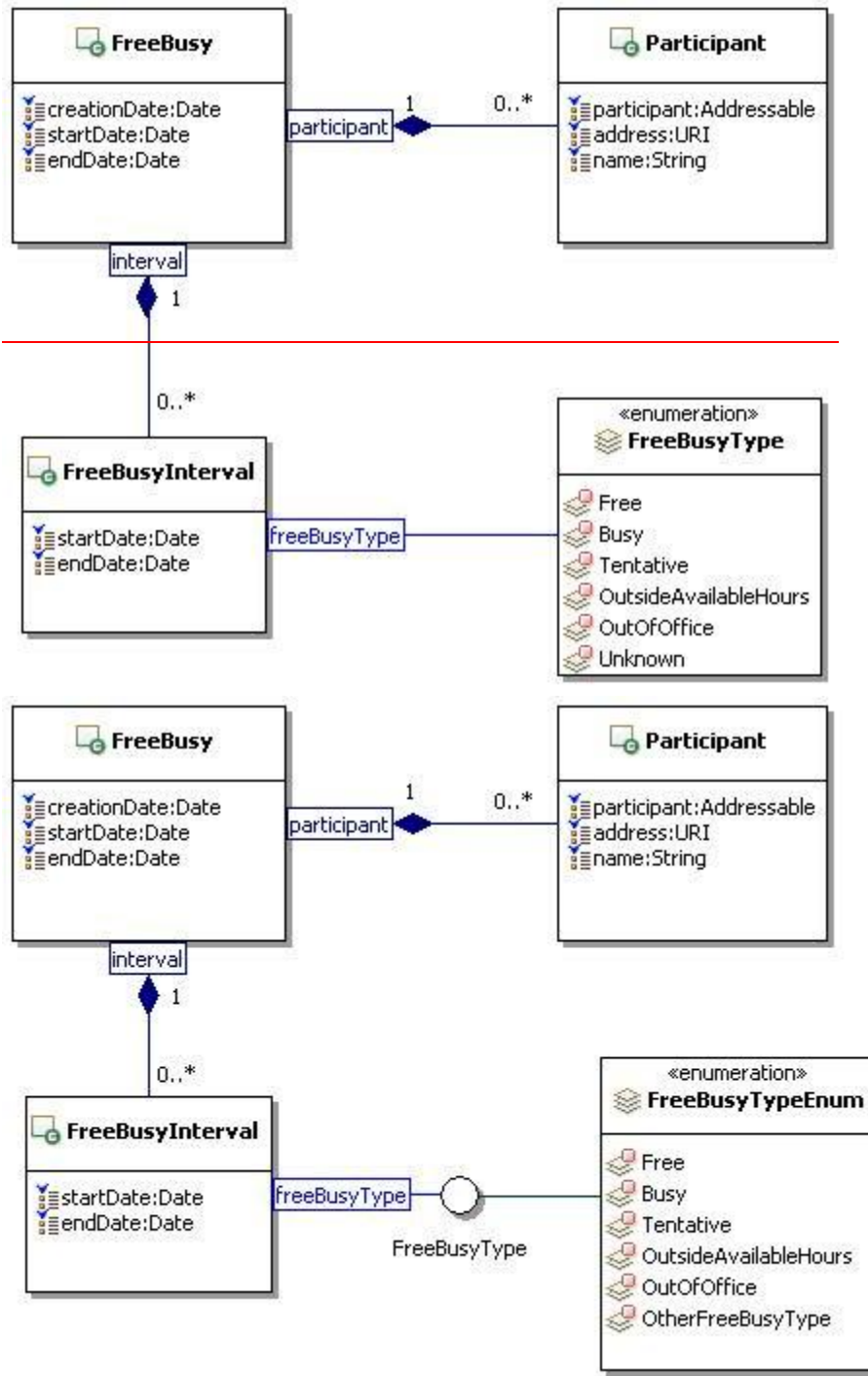


Figure 38-~~FreeBusy~~: Free Busy Class Diagram.

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.

propertyDefinitions

The values for this attribute are defined in Section 4.8.3.3.

4.8.3.3 Property Definitions

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

4.8.4 FreeBusyTypeEnum

The FreeBusyTypeEnum class is an enum class that enumerates the instances each of which expresses a type of free busy interval.

The FreeBusyType Enum class ~~is defined by the~~ has attribute values:

localNamespace

Value: icom_cal

localName

Value: FreeBusyTypeEnum

extendsFrom

Value: FreeBusyType

extendsFrom

Value:

7231 **stereotype**
7232 Value: primary
7233
7234 **isEnumeration**
7235 Value: TRUE
7236
7237 **description**
7238 Value: ~~An enumeration of the instances each of which expresses a~~A type of free busy interval.
7239
7240 **instances**
7241 Value: <icom_cal:Free, icom_cal:Busy, icom_cal:Tentative, icom_cal:OutsideAvailableHours,
7242 icom_cal:OutOfOffice, icom_cal:~~Unknown~~OtherFreeBusyType>
7243
7244 ~~There are~~ICOM defines six free busy types ~~defined by ICOM~~:
7245 • **icom_cal:Free** ~~to express that~~ a free busy interval is free.
7246 • **icom_cal:Busy** ~~to express that~~ a free busy interval is busy.
7247 • **icom_cal:Tentative** ~~to express that~~ a free busy interval is tentative.
7248 • **icom_cal:OutsideAvailableHours** ~~to express that~~ a free busy interval is outside available hours.
7249 • **icom_cal:OutOfOffice** ~~to express that~~ a free busy interval is ~~coincides with~~within out of office
7250 hours.
7251 • **icom_cal:**~~Unknown~~ ~~to express that~~OtherFreeBusyType a free busy interval is ~~unknown~~of other
7252 type.
7253

7254 **4.9 TaskList**Task List Module

7255 **4.9.1 TaskList**

7256 **4.9.1.1 Description**

7257 A task list ~~is a folder that~~ contains task management artifacts.

7258 **4.9.1.2 Class Definition**

7259 The TaskList class ~~is defined by the~~has attribute values:

7260
7261 **localNamespace**
7262 Value: icom_task
7263
7264 **localName**
7265 Value: TaskList
7266
7267 **extendsFrom**
7268 Value: icom_core:Folder
7269
7270 **stereotype**

7271 Value: primary
7272
7273 **description**
7274 | Value: A task list ~~is a folder that~~ contains task management artifacts.
7275
7276 **propertyDefinitions**
7277 The values for this attribute are defined in 4.9.1.3.

7278 **4.9.1.3 Property Definitions**

7279 The TaskList class inherits property definitions from super classes.
7280 The TaskList class MUST have the property definitions:

7281
7282 **icom_core:timeZone**
7283 Description: Time zone of a task list.
7284 Required: True
7285 Inherited: False
7286 | Property Type: icom_core:TimeZone
7287 Cardinality: Single
7288 Updatability: Read Write

7289
7290 **icom_core:element**
7291 Description: Elements of a task list.
7292 Required: False
7293 Inherited: True
7294 Property Type: icom_task:Task
7295 Cardinality: Multi
7296 Updatability: Read Only
7297

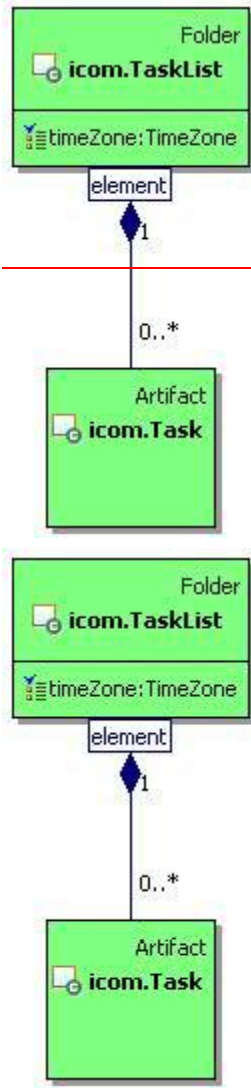


Figure 39-~~TaskList~~: Task List Class Diagram.

4.9.2 Task

4.9.2.1 Description

A task is an artifact that represents a task to do or a task assignment in a task list.

4.9.2.2 Class Definition

The Task class ~~is defined by the~~has attribute values:

localNamespace

Value: icom_task

localName

Value: Task

7313

7314 **extendsFrom**

7315 Value: icom_core:Artifact

7316

7317 **stereotype**

7318 Value: primary

7319

7320 **description**

7321 Value: A task is an artifact that represents a task to do or a task assignment in a task list.

7322

7323 **propertyDefinitions**

7324 The values for this attribute are defined in 4.9.2.3.

7325 4.9.2.3 Property Definitions

7326 The Task class inherits property definitions from super classes.

7327 The Task class MUST have the property definitions:

7329 ~~icom_task~~icom_core:location

7330 Description: Location of a task.

7331 Required: False

7332 Inherited: False

7333 Property Type: icom_core:Location

7334 Cardinality: Single

7335 Updatability: Read Write

7337 icom_core:organizer

7338 Description: Organizer of a task.

7339 Required: True

7340 Inherited: False

7341 Property Type: icom_core:Participant

7342 Cardinality: Single

7343 Updatability: On Create

7345 icom_core:priority

7346 Description: Priority of a task.

7347 Required: False

7348 Inherited: False

7349 Property Type: icom_core:Priority

7350 Cardinality: Single

7351 Updatability: Read Write

7353 icom_core:startDate

7354 Description: Start date and time of a task.

7355	Required:	True
7356	Inherited:	False
7357	Property Type:	DateTime
7358	Cardinality:	Single
7359	Updatability:	On Create

7360

7361 | **icom_taskcore:startDateResolution**

7362	Description:	Resolution of start date and time of a task.
7363	Required:	True
7364	Inherited:	False
7365	Property Type:	icom_core:DateTimeResolution
7366	Cardinality:	Single
7367	Updatability:	On Create

7368

7369 | **icom_task:icom_content:attachment**

7370	<u>Description:</u>	<u>One or more content attachments in a task.</u>
7371	<u>Required:</u>	<u>False</u>
7372	<u>Inherited:</u>	<u>False</u>
7373	<u>Property Type:</u>	<u>icom_content:Attachment</u>
7374	<u>Cardinality:</u>	<u>Multi</u>
7375	<u>Updatability:</u>	<u>Read Write</u>

7376

7377 | **icom_task:dueDate**

7378	Description:	Due date and time of a task.
7379	Required:	True
7380	Inherited:	False
7381	Property Type:	DateTime
7382	Cardinality:	Single
7383	Updatability:	On Create

7384

7385 | **icom_task:dueDateResolution**

7386	Description:	Resolution of due date and time of a task.
7387	Required:	True
7388	Inherited:	False
7389	Property Type:	icom_core:DateTimeResolution
7390	Cardinality:	Single
7391	Updatability:	On Create

7392

7393 | **icom_task:editMode**

7394	<u>Description:</u>	<u>Indicates a mode which determines whether a task is</u>
7395		<u>editable.</u>
7396	<u>Required:</u>	<u>False</u>
7397	<u>Inherited:</u>	<u>False</u>

7398 Property Type: icom_task:TaskEditMode
7399 Cardinality: Single
7400 Updatability: Read Only

7401
7402 **icom_core:location**

7403 Description: Location of a task.
7404 Required: False
7405 Inherited: False
7406 Property Type: icom_core:Location
7407 Cardinality: Single
7408 Updatability: Read Write

7409
7410 **icom_core:organizer**

7411 Description: Organizer of a task.
7412 Required: True
7413 Inherited: False
7414 Property Type: icom_core:Participant
7415 Cardinality: Single
7416 Updatability: On Create

7417
7418 **icom_task:taskStatus**

7419 Description: Status of a task.
7420 Required: True
7421 Inherited: False
7422 Property Type: icom_task:TaskStatus
7423 Cardinality: Single
7424 Updatability: Read Write

7425
7426 **icom_task:mode**

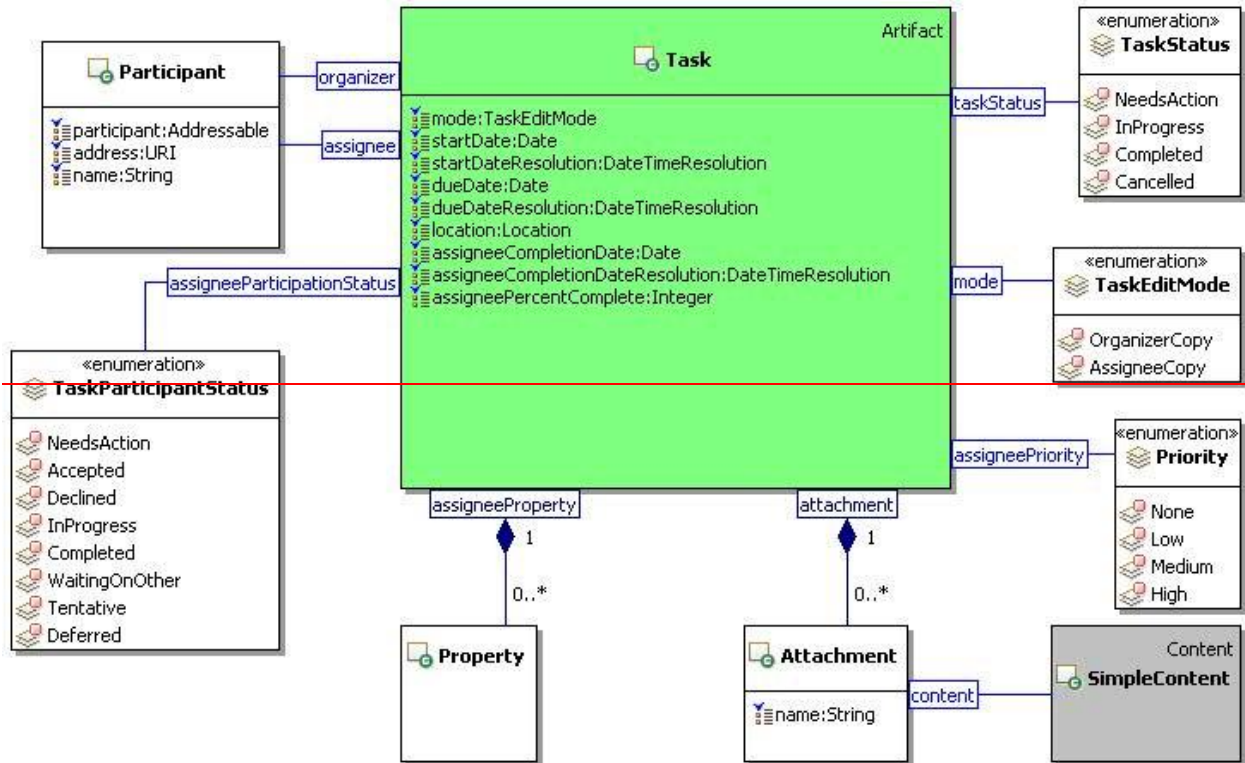
7427 Description: Mutability mode of a task.
7428 Required: False
7429 Inherited: False
7430 Property Type: icom_task:TaskEditMode
7431 Cardinality: Single
7432 Updatability: Read Only

7433
7434 **icom_content:attachment**

7435 Description: One or more simple content attachments in a task.
7436 Required: False
7437 Inherited: False
7438 Property Type: icom_content:Attachment
7439 Cardinality: Multi

7440	Updatability:	Read Write
7441		
7442	icom_task:assignee	
7443	Description:	An assignee of a task.
7444	Required:	False
7445	Inherited:	False
7446	Property Type:	icom_core:Participant
7447	Cardinality:	Single
7448	Updatability:	Read Only
7449		
7450	icom_task:assignee<u>Priority</u>participant<u>Status</u>	
7451	Description:	Priority for an assignee of a task.
7452	Required:	False
7453	Inherited:	False
7454	Property Type:	icom_core:Priority
7455	Cardinality:	Single
7456	Updatability:	Read Write
7457		
7458	icom_task:assignee<u>Participant</u><u>Status</u>	
7459	Description:	Participation status for an assignee of a task.
7460	Required:	False
7461	Inherited:	False
7462	Property Type:	icom_task:TaskParticipantStatus
7463	Cardinality:	Single
7464	Updatability:	Read Write
7465		
7466	icom_task:assignee<u>C</u><u>completionDate</u>	
7467	Description:	Completion date and time of a task.
7468	Required:	False
7469	Inherited:	False
7470	Property Type:	DateTime
7471	Cardinality:	Single
7472	Updatability:	Read Write
7473		
7474	icom_task:assignee<u>C</u><u>completionDate</u><u>Resolution</u>	
7475	Description:	Resolution of completion date and time of a task.
7476	Required:	False
7477	Inherited:	False
7478	Property Type:	icom_core:DateTimeResolution
7479	Cardinality:	Single
7480	Updatability:	Read Write
7481		

7482		icom_task:assigneePercentComplete	
7483		Description:	Percentage of task completed.
7484		Required:	False
7485		Inherited:	False
7486		Property Type:	Integer
7487		Cardinality:	Single
7488		Updatability:	Read Write
7489			
7490		icom_task:assigneeProperty	
7491		Description:	Extensible properties for an assignee of a task.
7492		Required:	False
7493		Inherited:	False
7494		Property Type:	icom_meta:Property
7495		Cardinality:	Multi
7496		Updatability:	Read Write
7497			



7498

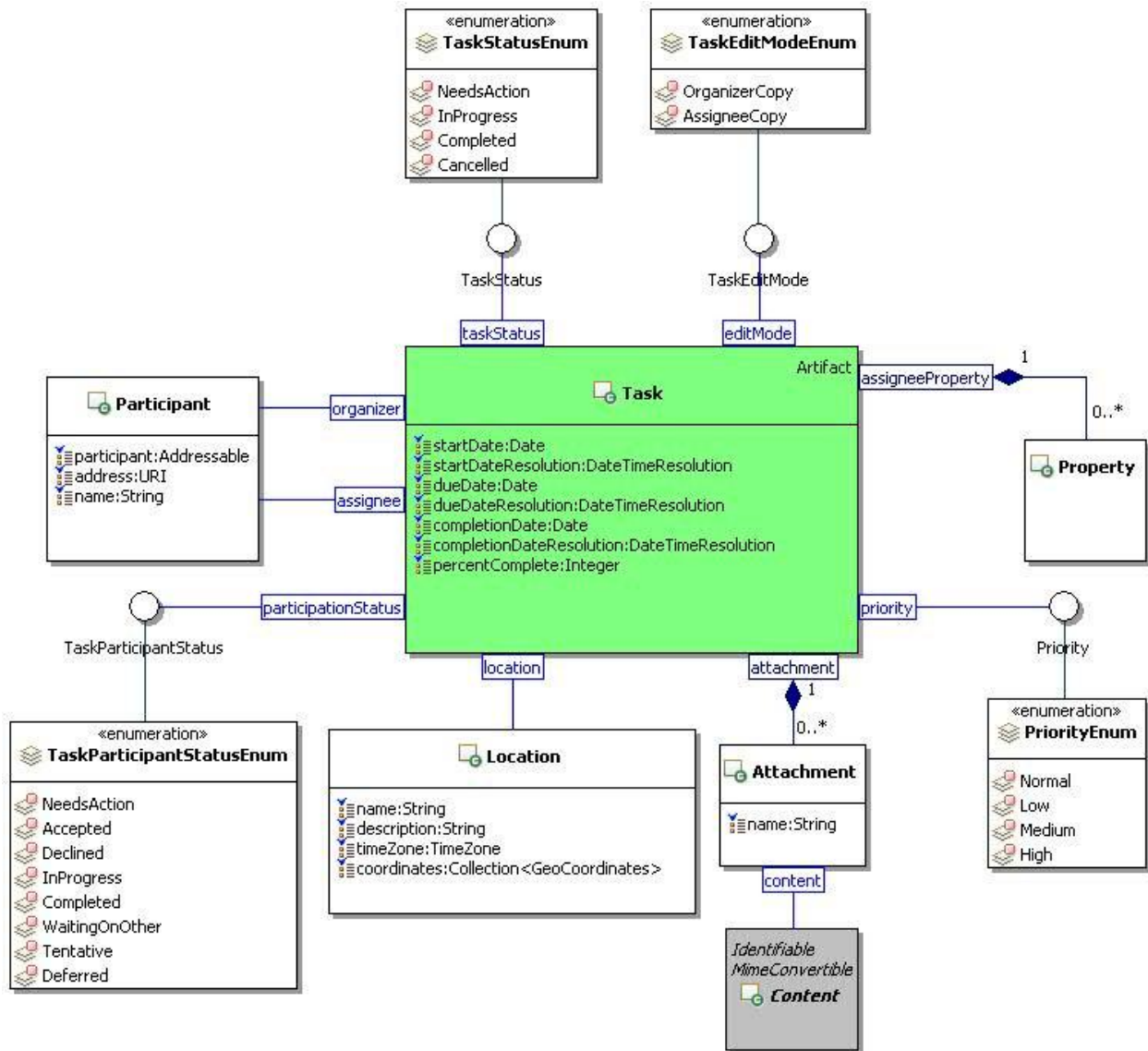


Figure 40: Task Class Diagram.

4.9.3 TaskStatus

The TaskStatus class is an enum class that enumerates the instances each of which expresses a status of task.

4.9.3.1 Description

A task status is a status of a task.

4.9.3.2 Class Definition

The TaskStatus class is defined by the mixin class which defines status of a task.

The TaskStatus class has attribute values:

localNamespace

7512 Value: icom_task

7513

7514 **localName**

7515 Value: TaskStatus

7516

7517 **extendsFrom**

7518 Value:

7519

7520 **stereotype**

7521 Value: mixin

7522

7523 **description**

7524 Value: TaskStatus is a mixin class which defines status of a task.

7525

7526 **propertyDefinitions**

7527 The values for this attribute are defined in Section 4.9.3.3.

7528 **4.9.3.3 Property Definitions**

7529 The TaskStatus class MAY include additional property definitions which are implementation-defined.

7530

7531 **4.9.4 TaskStatusEnum**

7532 The TaskStatusEnum class is an enum class that enumerates the instances each of which expresses a
7533 status of task.

7534 The TaskStatusEnum class has attribute values:

7535

7536 **localNamespace**

7537 Value: icom task

7538

7539 **localName**

7540 Value: TaskStatusEnum

7541

7542 **extendsFrom**

7543 Value: TaskStatus

7544

7545 **stereotype**

7546 Value: primary

7547

7548 **isEnumeration**

7549 Value: TRUE

7550

7551 **description**

7552 Value: ~~An enumeration of the instances each of which expresses a status~~Status of a task.

7553
7554 **instances**
7555 Value: <icom_task:NeedsAction, icom_task:InProgress, icom_task:Completed,
7556 icom_task:Cancelled>
7557

7558 ~~There are~~ ICOM defines four task status ~~defined by ICOM~~:
7559 • **icom_task:NeedsAction** ~~to express that~~ a task needs action.
7560 • **icom_task:InProgress** ~~to express that~~ a task is in progress.
7561 • **icom_task:Completed** ~~to express that~~ a task is completed.
7562 • **icom_task:Cancelled** ~~to express that~~ a task is cancelled.
7563

7564 ~~4.9.44.9.5~~ **TaskParticipantStatus**

7565 **4.9.5.1 Description**

7566 A task participant status is a participant's response status for a task assignment.

7567 **4.9.5.2 Class Definition**

7568 The TaskParticipantStatus class is a mixin class which defines a participant's response status for a task
7569 assignment.

7570 The TaskParticipantStatus class has attribute values:

7571
7572 **localNamespace**
7573 Value: icom_task
7574

7575 **localName**
7576 Value: TaskParticipantStatus
7577

7578 **extendsFrom**
7579 Value:
7580

7581 **stereotype**
7582 Value: mixin
7583

7584 **description**
7585 Value: TaskParticipantStatus is a mixin class which defines a participant's response status for a
7586 task assignment.
7587

7588 **propertyDefinitions**
7589 The values for this attribute are defined in Section 4.9.5.3.

7590 **4.9.5.3 Property Definitions**

7591 The TaskParticipantStatus class MAY include additional property definitions which are implementation-
7592 defined.

4.9.6 TaskParticipantStatusEnum

The TaskParticipantStatusEnum class is an enum class that enumerates the instances each of which expresses a participant's response status for a task.

The TaskParticipantStatusEnum class ~~is defined by the~~has attribute values:

localNamespace

Value: icom_task

localName

Value: TaskParticipantStatusEnum

extendsFrom

Value: TaskParticipantStatus

~~extendsFrom~~

~~Value:~~

stereotype

Value: primary

isEnumeration

Value: TRUE

description

Value: ~~An enumeration of the instances each of which expresses a participant's~~Participant's response status for a task.

instances

Value: <icom_task:NeedsAction, icom_task:Accepted, icom_task:Declined, icom_task:InProgress, icom_task:Completed, icom_task:WaitingOnOther, icom_task:Tentative, icom_task:Deferred>

~~There are~~ICOM defines eight task participant's status ~~defined by ICOM:~~

- **icom_task:NeedsAction** ~~to express that~~ an assignee needs to act on a task.
- **icom_task:Accepted** ~~to express that~~ an assignee accepted a task.
- **icom_task:Declined** ~~to express that~~ an assignee declined a task.
- **icom_task:InProgress** ~~to express that~~ a task is in progress.
- **icom_task:Completed** ~~to express that~~ a task is completed.
- **icom_task:WaitingOnOther** ~~to express that~~ an assignee is waiting on other.
- **icom_task:Tentative** ~~to express that~~ an assignee is tentative about a task ~~status~~.
- **icom_task:Deferred** ~~to express that~~ an assignee deferred a task.

4.9.54.9.7 TaskEditMode

4.9.7.1 Description

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

4.9.7.2 Class Definition

The TaskEditMode class is a mixin class which defines a mode that indicates whether a task is editable.

The TaskEditMode class has attribute values:

localNamespace

Value: icom task

localName

Value: TaskEditMode

extendsFrom

Value:

stereotype

Value: mixin

description

Value: TaskEditMode is a mixin class which defines a mode that indicates whether task is editable.

propertyDefinitions

The values for this attribute are defined in Section 4.9.7.3.

4.9.7.3 Property Definitions

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

4.9.8 TaskEditModeEnum

The TaskEditModeEnum class is an enum class that enumerates the instances each of which expresses an editable mode of a task.

The TaskEditModeEnum class is defined by the has attribute values:

localNamespace

Value: icom_task

localName

Value: TaskEditModeEnum

extendsFrom

7676 | Value: TaskEditMode
 7677 |
 7678 | ~~extendsFrom~~
 7679 | Value:
 7680 |
 7681 | **stereotype**
 7682 | Value: primary
 7683 |
 7684 | **isEnumeration**
 7685 | Value: TRUE
 7686 |
 7687 | **description**
 7688 | Value: ~~An enumeration of the instances each of which expresses an editable~~ A mode of that
 7689 | indicates whether a task is editable.
 7690 |
 7691 | **instances**
 7692 | Value: <icom_task:OrganizerCopy, icom_task:AssigneeCopy>
 7693 |

7694 | ~~There are~~ ICOM defines two task editable modes ~~defined by ICOM~~:

- 7695 | • **icom_task:OrganizerCopy:** ~~to express that~~ a task is a copy created by an organizer who may
 7696 | update the properties such as start time, due time, ~~etc.~~
- 7697 | • **icom_task:AssigneeCopy:** ~~to express that~~ a task is a copy delivered to an assignee who may
 7698 | only update the assignee properties such as ~~assignee~~ completion time, ~~assignee~~ participant
 7699 | status, ~~assignee~~ percent completed, ~~etc.~~

7701 | 4.10 Forum Module

7702 | 4.10.1 Discussion

7703 | 4.10.1.1 Description

7704 | A discussion is an item in a discussion container.

7705 | 4.10.1.2 Class Definition

7706 | The Discussion class is a mixin class ~~which that~~ defines the characteristics of ~~entities~~ artifacts that can be
 7707 | elements of ~~a DiscussionContainer~~ discussion containers.

7708 | The Discussion class ~~is defined by the~~ has attribute values:

7709 |
 7710 | **localNamespace**
 7711 | Value: icom_forum
 7712 |
 7713 | **localName**
 7714 | Value: Discussion
 7715 |

7716 | **extendsFrom**

7717 Value: icom_core:Item

7718

7719 **stereotype**

7720 Value: mixin

7721

7722 **description**

7723 | Value: Discussion is a mixin class ~~which~~that defines the characteristics of ~~entities~~artifacts that

7724 | can be placed in a ~~DiscussionContainer~~discussion container.

7725

7726 **propertyDefinitions**

7727 The values for this attribute are defined in Section 4.10.1.3.

7728 4.10.1.3 Property Definitions

7729 The Discussion class inherits property definitions from super classes.

7730 The Discussion class MUST have the property definition:

7731

7732 **icom_forum:inReplyTo**

7733 Description: Another discussion object that a discussion object is replying

7734 to.

7735 Required: False

7736 Inherited: False

7737 Property Type: icom_forum:Discussion

7738 Cardinality: Single

7739 Updatability: Read Write

7740

7741 The Discussion class MAY include additional property definitions which are implementation-defined.

7742

7743 4.10.2 DiscussionContainer

7744 4.10.2.1 Description

7745 | A discussion container ~~is a container of discussions~~contains discussion items.

7746 4.10.2.2 Class Definition

7747 | The DiscussionContainer class is a mixin class ~~which~~that defines the characteristics of ~~entities~~folders that

7748 | contain Discussion items.

7749 | The DiscussionContainer class ~~is defined by the~~has attribute values:

7750

7751 **localNamespace**

7752 Value: icom_forum

7753

7754 **localName**

7755 Value: DiscussionContainer

7756

7757 **extendsFrom**
7758 Value: icom_core:Container
7759
7760 **stereotype**
7761 Value: mixin
7762
7763 **description**
7764 Value: DiscussionContainer is a mixin class ~~which~~that defines the characteristics of
7765 ~~entities~~folders that contain Discussion items.
7766
7767 **propertyDefinitions**
7768 The values for this attribute are defined in Section 4.10.2.3.

7769 4.10.2.3 Property Definitions

7770 The DiscussionContainer class inherits property definitions from super classes.
7771 The DiscussionContainer class MUST have the property definition:

7772
7773 **icom_core:element**
7774 Description: Elements of a discussion container.
7775 Required: False
7776 Inherited: True
7777 Property Type: icom_forum:Discussion
7778 Cardinality: Multi
7779 Updatability: Read Only

7780
7781 The DiscussionContainer class MAY include additional property definitions which are implementation-
7782 defined.

7784 4.10.3 DiscussionMessage

7785 4.10.3.1 Description

7786 A discussion message is a message in a forum discussion thread.

7787 4.10.3.2 Class Definition

7788 The DiscussionMessage class ~~is defined by the~~has attribute values:

7789
7790 **localNamespace**
7791 Value: icom_forum
7792
7793 **localName**
7794 Value: DiscussionMessage
7795
7796 **extendsFrom**

7797 Value: icom_msg:Message, icom_forum:Discussion

7798

7799 **stereotype**

7800 Value: primary

7801

7802 **description**

7803 Value: Discussion message is a message in a forum discussion thread.

7804

7805 **propertyDefinitions**

7806 The values for this attribute are defined in Section 4.10.3.3.

7807 4.10.3.3 Property Definitions

7808 The DiscussionMessage class inherits property definitions from super classes.

7809 The DiscussionMessage class MUST have the property definition:

7810

7811 **icom_forum:inReplyTo**

7812 Description: Another discussion message that a discussion message is
7813 replying to.

7814 Required: False

7815 Inherited: True

7816 Property Type: icom_forum:DiscussionMessage

7817 Cardinality: Single

7818 Updatability: Read Write

7819

7820 The DiscussionMessage class MAY include additional property definitions which are implementation-
7821 defined.

7822

7823 4.10.4 TopicContainer

7824 4.10.4.1 Description

7825 | A topic container ~~is a container of~~contains topics.

7826 4.10.4.2 Class Definition

7827 | The TopicContainer class is a mixin class which defines the characteristics of ~~entities~~folders that contain
7828 Topics.

7829 | The TopicContainer class ~~is defined by the~~has attribute values:

7830

7831 **localNamespace**

7832 Value: icom_forum

7833

7834 **localName**

7835 Value: TopicContainer

7836

7837 **extendsFrom**
7838 Value: icom_core:Container
7839
7840 **stereotype**
7841 Value: mixin
7842
7843 **description**
7844 Value: TopicContainer is a mixin class ~~which~~that defines the characteristics of ~~entities~~folders
7845 that contain ~~T~~htopics.
7846
7847 **propertyDefinitions**
7848 The values for this attribute are defined in Section 4.10.4.3.

7849 **4.10.4.3 Property Definitions**

7850 The TopicContainer class inherits property definitions from super classes.
7851 The TopicContainer class MUST have the property definitions:

7852

7853	icom_core:element	
7854	Description:	Elements of a topic container.
7855	Required:	False
7856	Inherited:	True
7857	Property Type:	icom_forum:Topic
7858	Cardinality:	Multi
7859	Updatability:	Read Only

7860
7861 The TopicContainer class MAY include additional property definitions which are implementation-defined.
7862

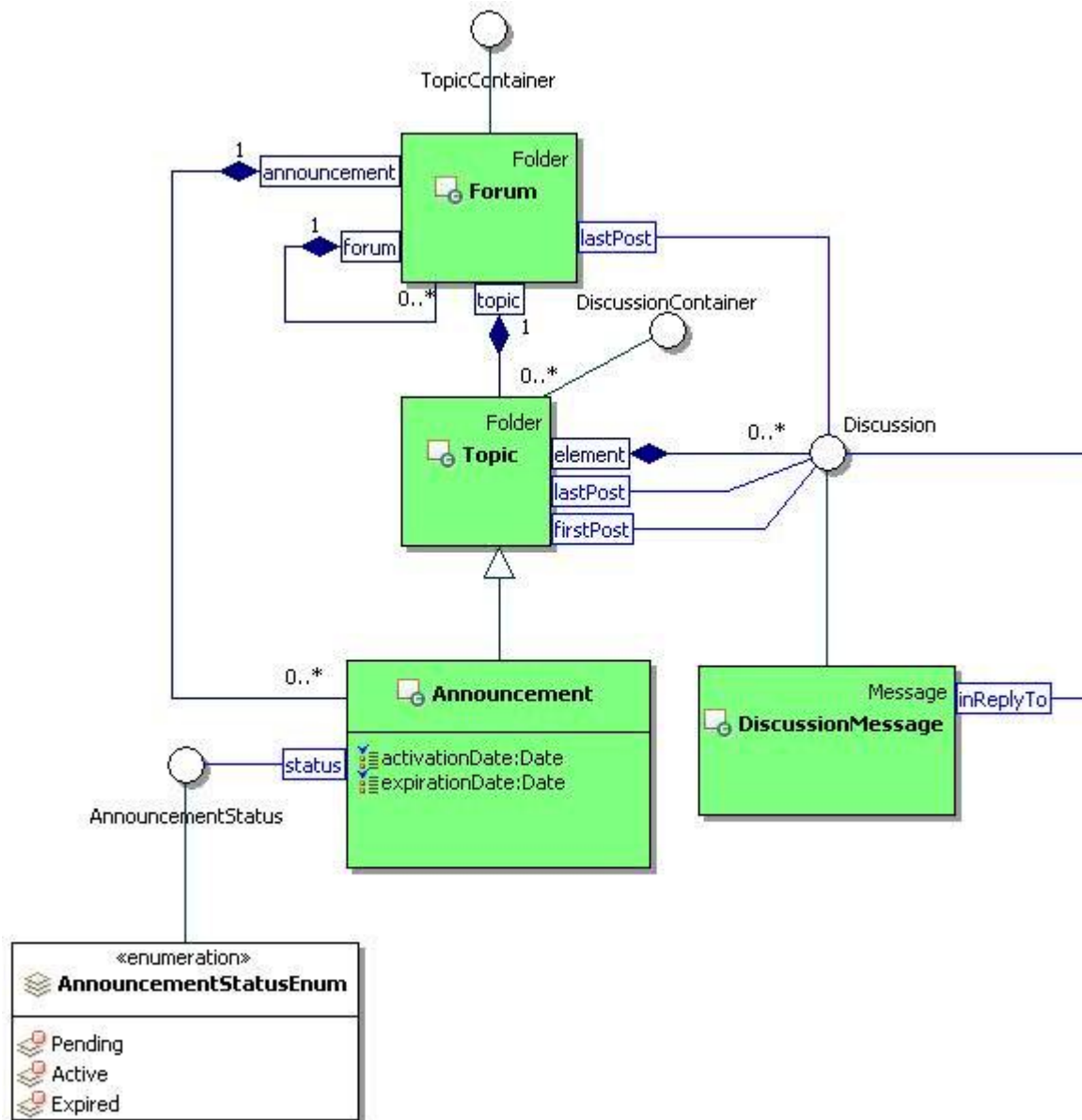


Figure 41: Forum Class Diagram.

4.10.5 Forum

4.10.5.1 Description

A forum ~~is a folder that~~ contains sub-forums, topics, and announcements.

~~4.10.5.2.1.1.1 Class Definition~~

4.10.5.2 Class Definition

The Forum class ~~is defined by the~~has attribute values:

localNamespace

7875 Value: icom_forum

7876

7877 **localName**

7878 Value: Forum

7879

7880 **extendsFrom**

7881 Value: icom_core:Folder, icom_forum:TopicContainer

7882

7883 **stereotype**

7884 Value: primary

7885

7886 **description**

7887 | Value: A forum ~~is a folder that~~ contains sub-forums, topics, and announcements.

7888

7889 **propertyDefinitions**

7890 The values for this attribute are defined in Section 4.10.5.3.

7891 4.10.5.3 Property Definitions

7892 The Forum class inherits property definitions from super classes.

7893 The Forum class MUST have the property definitions:

7894

7895 **icom_forum:lastPost**

7896	Description:	The last posted discussion in a forum.
7897	Required:	False
7898	Inherited:	False
7899	Property Type:	icom_forum:Discussion
7900	Cardinality:	Single
7901	Updatability:	Read Only

7902

7903 **icom_forum:forum**

7904	Description:	Sub-forums of a forum.
7905	Required:	False
7906	Inherited:	False
7907	Property Type:	icom_forum:Forum
7908	Cardinality:	Multi
7909	Updatability:	Read Only

7910

7911 **icom_forum:topic**

7912	Description:	Topics of a forum.
7913	Required:	False
7914	Inherited:	False
7915	Property Type:	icom_forum:Topic
7916	Cardinality:	Multi

7917	Updatability:	Read Only
7918		
7919	icom_forum:announcement	
7920	Description:	Announcements of a forum.
7921	Required:	False
7922	Inherited:	False
7923	Property Type:	icom_forum:Announcement
7924	Cardinality:	Multi
7925	Updatability:	Read Only
7926		
7927	The Forum class MAY include additional property definitions which are implementation-defined.	
7928		

7929 4.10.6 Topic

7930 4.10.6.1 Description

7931 | A topic ~~is a folder that~~ contains ~~a conversation~~conversations among forum participants. The discussions
7932 | in a topic may be sorted in chronological order or threaded by reply.

7933 4.10.6.2 Class Definition

7934 | The Topic class ~~is defined by the~~has attribute values:

7935	
7936	localNamespace
7937	Value: icom_forum
7938	
7939	localName
7940	Value: Topic
7941	
7942	extendsFrom
7943	Value: icom_core:Folder, icom_forum:DiscussionContainer
7944	
7945	stereotype
7946	Value: primary
7947	
7948	description
7949	Value: A topic is a folder that contains discussion threads.
7950	
7951	propertyDefinitions
7952	The values for this attribute are defined in Section 4.10.6.3.

7953 4.10.6.3 Property Definitions

7954 The Topic class inherits property definitions from super classes.

7955 The Topic class MUST have the property definitions:

7956

7957	icom_core:element	
7958	Description:	Elements of a topic.
7959	Required:	False
7960	Inherited:	True
7961	Property Type:	icom_forum:Discussion
7962	Cardinality:	Multi
7963	Updatability:	Read Only
7964		
7965	icom_forum:firstPost	
7966	Description:	The first posted discussion in a topic.
7967	Required:	False
7968	Inherited:	False
7969	Property Type:	icom_forum:Discussion
7970	Cardinality:	Single
7971	Updatability:	Read Only
7972		
7973	icom_forum:lastPost	
7974	Description:	The last posted discussion in a topic.
7975	Required:	False
7976	Inherited:	False
7977	Property Type:	icom_forum:Discussion
7978	Cardinality:	Single
7979	Updatability:	Read Only
7980		
7981	The Topic class MAY include additional property definitions which are implementation-defined.	
7982		

7983 4.10.7 Announcement

7984 4.10.7.1 Description

7985 | An announcement ~~is a special topic for~~contains time-sensitive discussion posts that are valid for a
7986 | specified period of time, depending on activation and expiration times.

7987 4.10.7.2 Class Definition

7988 ~~4.10.7.2.1.1.1 Class Definition~~

7989 | The Announcement class ~~is defined by the~~has attribute values:

7990		
7991	localNamespace	
7992	Value:	icom_forum
7993		
7994	localName	
7995	Value:	Announcement
7996		

7997 **extendsFrom**
7998 Value: icom_forum:Topic
7999
8000 **stereotype**
8001 Value: primary
8002
8003 **description**
8004 Value: An announcement ~~is a special topic for discussions~~contains discussion items that are
8005 valid for a specified period of time.
8006
8007 **propertyDefinitions**
8008 The values for this attribute are defined in Section 4.10.7.3.

8009 **4.10.7.3 Property Definitions**

8010 The Announcement class inherits property definitions from super classes.
8011 The Announcement class **MUST** have the property definitions:

8012
8013 **icom_forum:activationDate**
8014 Description: Date and time when an announcement becomes active.
8015 Required: False
8016 Inherited: False
8017 Property Type: DateTime
8018 Cardinality: Single
8019 Updatability: Read Write

8020
8021 **icom_forum:expirationDate**
8022 Description: Date and time when an announcement expires.
8023 Required: False
8024 Inherited: False
8025 Property Type: DateTime
8026 Cardinality: Single
8027 Updatability: Read Write

8028
8029 **icom_forum:announcementStatus**
8030 Description: Status of an announcement.
8031 Required: True
8032 Inherited: False
8033 Property Type: icom_forum:AnnouncementStatus
8034 Cardinality: Single
8035 Updatability: Read Write

8036
8037 The Announcement class **MAY** include additional property definitions which are implementation-defined.
8038

4.10.8 AnnouncementStatus

~~The AnnouncementStatus class is an enum class that enumerates the instances each of which expresses a status of announcement.~~

4.10.8.1 Description

~~An announcement status is status of an announcement.~~

4.10.8.2 Class Definition

~~The AnnouncementStatus class is defined by the a mixin class which defines status of an announcement.~~

~~The AnnouncementStatus class has~~ attribute values:

localNamespace

Value: `icom_forum`

localName

Value: `AnnouncementStatus`

extendsFrom

Value:

stereotype

Value: `mixin`

description

Value: `AnnouncementStatus` is a mixin class which defines status of an announcement.

propertyDefinitions

The values for this attribute are defined in Section 4.10.8.3.

4.10.8.3 Property Definitions

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

4.10.9 AnnouncementStatusEnum

The AnnouncementStatusEnum class is an enum class that enumerates the instances each of which expresses a status of announcement.

The AnnouncementStatusEnum class has attribute values:

localNamespace

Value: `icom_forum`

localName

Value: `AnnouncementStatusEnum`

8079
8080 **extendsFrom**
8081 Value: AnnouncementStatus
8082
8083 **stereotype**
8084 Value: primary
8085
8086 **isEnumeration**
8087 Value: TRUE
8088
8089 **description**
8090 Value: ~~An enumeration of the instances each of which expresses a status~~Status of
8091 announcement.
8092
8093 **instances**
8094 Value: <icom_forum:Pending, icom_forum:Active, icom_forum:Expired>
8095
8096 ~~There are~~ICOM defines three announcement status ~~defined by ICOM:~~
8097 • **icom_forum:Pending** ~~to express that~~ an announcement is pending.
8098 • **icom_forum:Active** ~~to express that~~ an announcement is active.
8099 • **icom_forum:Expired** ~~to express that~~ an announcement is expired.
8100

8101 4.11 Conference Module

8102 4.11.1 Conference

8103 4.11.1.1 Description

8104 A conference is a foldercontainer that represents a durable context for conference sessions.
8105 It contains conference metadata, settings, and transcripts.

8106 4.11.1.2 Class Definition

8107 The Conference class ~~is defined by the~~has attribute values:

8108
8109 **localNamespace**
8110 Value: icom_conf
8111
8112 **localName**
8113 Value: Conference
8114
8115 **extendsFrom**
8116 Value: icom_core:Folder
8117
8118 **stereotype**

8119 Value: primary

8120

8121 **description**

8122 Value: A conference ~~is a folder that~~ represents a durable context for online conference sessions.

8123

8124 **propertyDefinitions**

8125 The values for this attribute are defined in Section 4.11.1.3.

8126 4.11.1.3 Property Definitions

8127 The Conference class inherits property definitions from super classes.

8128 The Conference class MUST have the property definitions:

8129

8130 **icom_core:organizer**

8131 Description: Organizer of a conference.

8132 Required: False

8133 Inherited: False

8134 Property Type: icom_core:Participant

8135 Cardinality: Single

8136 Updatability: On Create

8137

8138 **icom_conf:conferenceType**

8139 Description: Type of a conference.

8140 Required: False

8141 Inherited: False

8142 Property Type: icom_conf:ConferenceType

8143 Cardinality: Single

8144 Updatability: Read Write

8145

8146 **icom_conf:conferenceState~~us~~**

8147 Description: ~~Session state~~Status of a conference.

8148 Required: False

8149 Inherited: False

8150 Property Type: icom_conf:ConferenceState~~us~~

8151 Cardinality: Single

8152 Updatability: Read Only

8153

8154 **icom_conf:runningSession**

8155 Description: Current session of a conference.

8156 Required: False

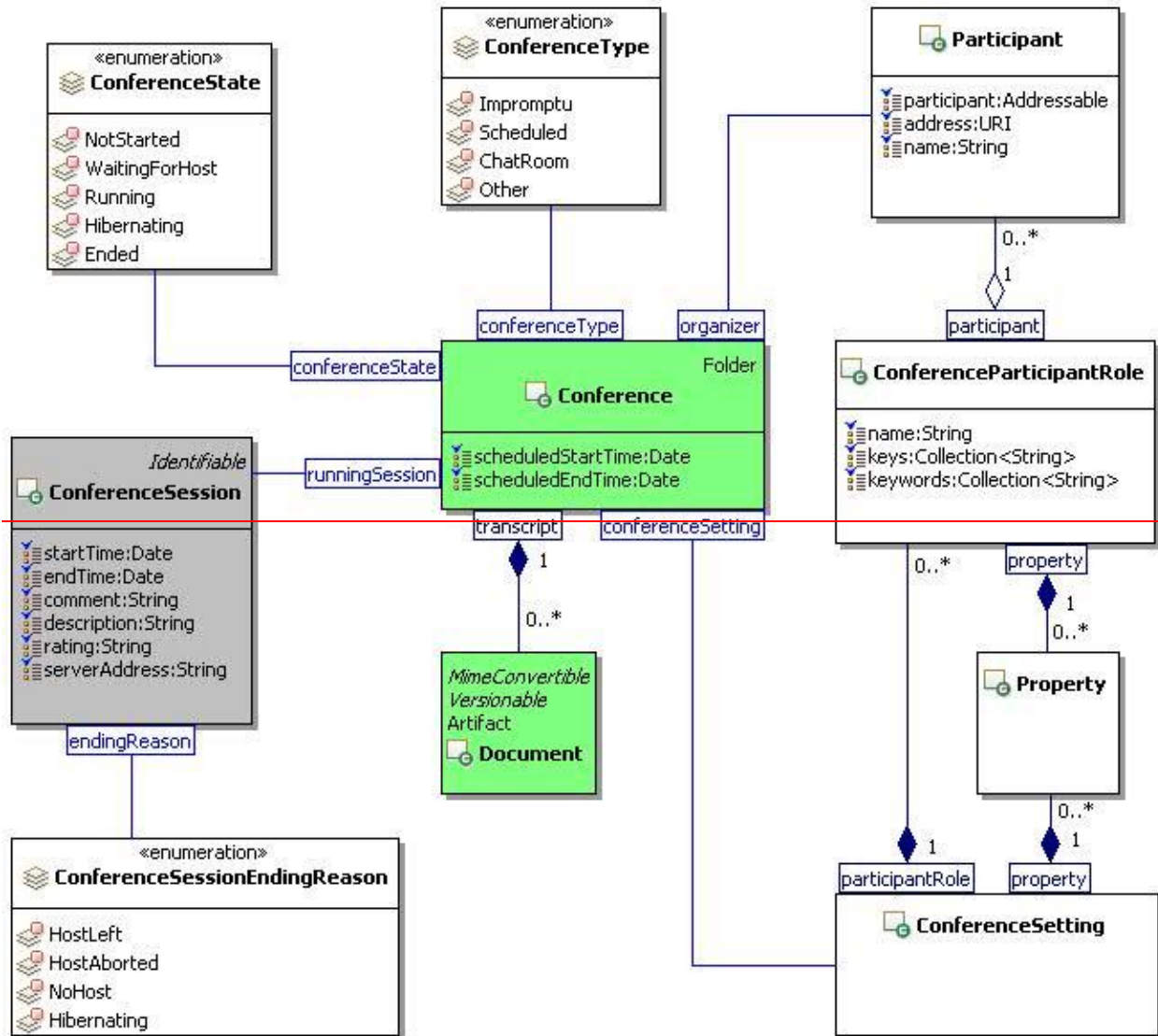
8157 Inherited: False

8158 Property Type: icom_conf:ConferenceSession

8159 Cardinality: Single

8160 Updatability: Read Only

8161		
8162	icom_conf:conferenceSetting	
8163	Description:	Configurable settings of a conference.
8164	Required:	False
8165	Inherited:	False
8166	Property Type:	icom_conf:ConferenceSetting
8167	Cardinality:	Single
8168	Updatability:	Read Only
8169		
8170	icom_conf:transcript	
8171	Description:	Transcripts from ended sessions of a conference.
8172	Required:	False
8173	Inherited:	False
8174	Property Type:	icom_doc:Document
8175	Cardinality:	Multi
8176	Updatability:	Read Write
8177		
8178	icom_conf:scheduledStartTimeDate	
8179	Description:	Scheduled start <u>date and</u> time of the current or next 8180 <u>conference</u> session- of a conference .
8181	Required:	False
8182	Inherited:	False
8183	Property Type:	Date Time
8184	Cardinality:	Single
8185	Updatability:	Read Write
8186		
8187	icom_conf:scheduledEndTimeDate	
8188	Description:	Scheduled end <u>date and</u> time of the current or next 8189 <u>conference</u> session- of a conference .
8190	Required:	False
8191	Inherited:	False
8192	Property Type:	Date Time
8193	Cardinality:	Single
8194	Updatability:	Read Write
8195		
8196	The Conference class MAY include additional property definitions which are implementation-defined.	
8197		



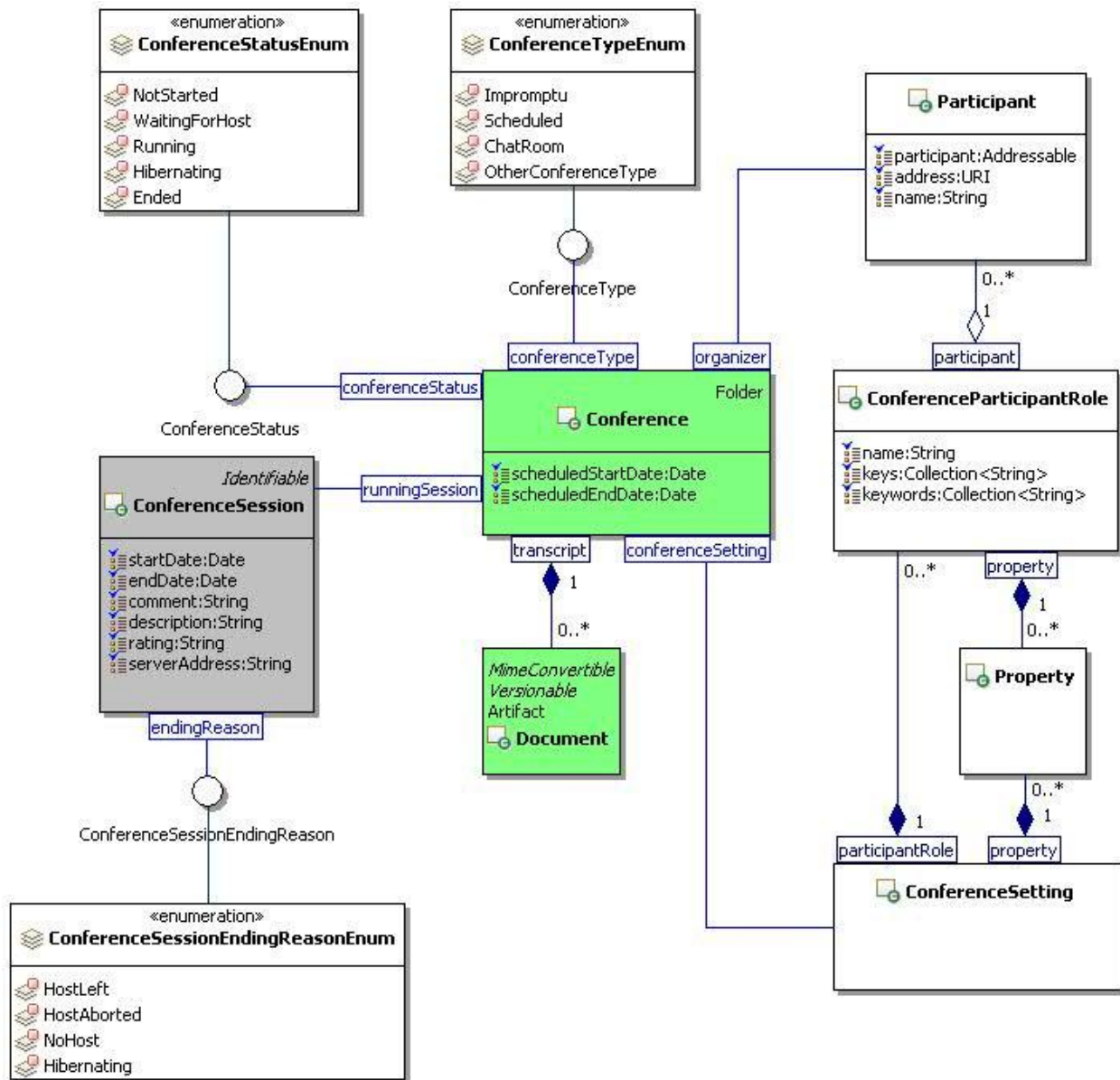


Figure 42: Conference Class Diagram.

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 an **enuma mixin** class that enumerates the instances each of which expresses defines a type of a conference.

The ConferenceType class is defined by the **has** attribute values:

localNamespace

8211 Value: icom_conf

8212

8213 **localName**

8214 Value: ConferenceType

8215

8216 **extendsFrom**

8217 Value:

8218

8219 **stereotype**

8220 Value: mixin

8221

8222 **description**

8223 Value: ConferenceType is a mixin class which defines type of conference.

8224

8225 **propertyDefinitions**

8226 The values for this attribute are defined in Section 4.11.2.3.

8227 **4.11.2.3 Property Definitions**

8228 The ConferenceType class MAY include additional property definitions which are implementation-defined.

8229

8230 **4.11.3 ConferenceTypeEnum**

8231 The ConferenceTypeEnum class is an enum class that enumerates the instances each of which
8232 expresses a type of a conference.

8233 The ConferenceTypeEnum class has attribute values:

8234

8235 **localNamespace**

8236 Value: icom_conf

8237

8238 **localName**

8239 Value: ConferenceTypeEnum

8240

8241 **extendsFrom**

8242 Value: ConferenceType

8243

8244 **stereotype**

8245 Value: primary

8246

8247 **isEnumeration**

8248 Value: TRUE

8249

8250 **description**

8251 Value: ~~An enumeration of the instances each of which expresses a~~ type of a conference.

8252

8253 **instances**

8254 Value: <icom_conf:Impromptu, icom_conf:Scheduled, icom_conf:ChatRoom,

8255 icom_conf:~~Other~~OtherConferenceType>

8256

8257 ~~There are~~ICOM defines four conference types ~~defined by ICOM:~~

8258 • **icom_conf:Impromptu** ~~to express that~~ a conference session is started impromptu.

8259 • **icom_conf:Scheduled** ~~to express that~~ a conference session is scheduled.

8260 • **icom_conf:ChatRoom** ~~to express that~~ a conference is used for a chat room.

8261 • **icom_conf:**~~Other~~ ~~to express that~~OtherConferenceType a conference is of other type.

8262

8263 **4.11.3 ConferenceState**

8264 **4.11.4 The ConferenceState class**ConferenceStatus

8265 **4.11.4.1 Description**

8266 A conference status is an enum~~status of an online conference.~~

8267 **4.11.4.2 Class Definition**

8268 The ConferenceStatus class is a mixin class that enumerates the instances each of which expresses a

8269 session state~~defines status of an online conference.~~

8270 The ConferenceStateus class ~~is defined by the~~has attribute values:

8271

8272 **localNamespace**

8273 Value: icom_conf

8274

8275 **localName**

8276 Value: ConferenceStateus

8277

8278 **extendsFrom**

8279 Value:

8280

8281 **stereotype**

8282 Value: mixin

8283

8284 **description**

8285 Value: ConferenceStatus is a mixin class which defines status of an online conference.

8286

8287 **propertyDefinitions**

8288 The values for this attribute are defined in Section 4.11.4.3.

4.11.4.3 Property Definitions

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

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: ~~An enumeration of the instances each of which expresses a session state~~Status of a conference.

instances

Value: `<icom_conf:NotStarted, icom_conf:WaitingForHost, icom_conf:Running, icom_conf:Hibernating, icom_conf:Ended>`

~~There are~~ICOM defines five conference ~~session states defined by ICOM~~status:

- `icom_conf:NotStarted` ~~to express that~~ a conference session is not started .
- `icom_conf:WaitingForHost` ~~to express that~~ a conference session is waiting for a host.
- `icom_conf:Running` ~~to express that~~ a conference session is running.
- `icom_conf:Hibernating` ~~to express that~~ a conference session is hibernating.
- `icom_conf:Ended` ~~to express that~~ a conference session is ended.

8328 **4.11.44.11.6 ConferenceSession**

8329 **4.11.4.14.11.6.1 Description**

8330 A conference session represents the metadata for a ~~runtime~~ session of a conference.

8331 ~~**4.11.4.21.1.1.1 Class Definition**~~

8332 **4.11.6.2 Class Definition**

8333 The ConferenceSession class ~~is defined by the~~has attribute values:

8334

8335 **localNamespace**

8336 Value: icom_conf

8337

8338 **localName**

8339 Value: ConferenceSession

8340

8341 **extendsFrom**

8342 Value: icom_core:Identifiable

8343

8344 **stereotype**

8345 Value: primary

8346

8347 **description**

8348 Value: A conference session represents the metadata for a ~~runtime~~ session of a conference.

8349

8350 **propertyDefinitions**

8351 The values for this attribute are defined in Section 4.11.6.3.

8352 **4.11.4.34.11.6.3 Property Definitions**

8353 The ConferenceSession class inherits property definitions from super classes.

8354 The ConferenceSession class MUST have the property definitions:

8355

8356 **icom_conf:startTimecore:startDate**

8357 Description: Start date and time of a conference session.

8358 Required: False

8359 Inherited: False

8360 Property Type: DateTime

8361 Cardinality: Single

8362 Updatability: Read Only

8363

8364 **icom_conf:endTimecore:endDate**

8365 Description: End date and time of a conference session.

8366 Required: False

8367	Inherited:	False
8368	Property Type:	Date <u>Time</u>
8369	Cardinality:	Single
8370	Updatability:	Read Only
8371		
8372	icom_conf:comment	
8373	Description:	Comment on a conference session.
8374	Required:	False
8375	Inherited:	False
8376	Property Type:	String
8377	Cardinality:	Single
8378	Updatability:	Read Write
8379		
8380	icom_conf:description	
8381	Description:	Description of a conference session.
8382	Required:	False
8383	Inherited:	False
8384	Property Type:	String
8385	Cardinality:	Single
8386	Updatability:	Read Write
8387		
8388	icom_conf:rating	
8389	Description:	Rating of a conference session.
8390	Required:	False
8391	Inherited:	False
8392	Property Type:	String
8393	Cardinality:	Single
8394	Updatability:	Read Write
8395		
8396	icom_conf:serverAddress	
8397	Description:	Address of a server that hosts a conference session.
8398	Required:	False
8399	Inherited:	False
8400	Property Type:	String
8401	Cardinality:	Single
8402	Updatability:	Read Only
8403		
8404	icom_conf:endingReason	
8405	Description:	Reason for ending a conference session.
8406	Required:	False
8407	Inherited:	False
8408	Property Type:	icom_conf:ConferenceSessionEndingReason

8409 Cardinality: Single
8410 Updatability: Read Only

8411
8412 The ConferenceSession class MAY include additional property definitions which are implementation-
8413 defined.
8414

8415 ~~4.11.54.11.7~~ **ConferenceSessionEndingReason**

8416 **4.11.7.1 Description**

8417 A conference session ending reason is an indication of how a conference session ended.

8418 **4.11.7.2 Class Definition**

8419 The ConferenceSessionEndingReason class is ~~an enuma mixin class that enumerates the instances~~
8420 ~~each of which expresses a reason for ending~~defines an indication of how a conference session ended.

8421 The ConferenceSessionEndingReason class ~~is defined by the~~has attribute values:

8422
8423 **localNamespace**
8424 Value: icom_conf

8425
8426 **localName**
8427 Value: ConferenceSessionEndingReason

8428
8429 **extendsFrom**
8430 Value:

8431
8432 **stereotype**
8433 Value: mixin

8434
8435 **description**
8436 Value: ConferenceSessionEndingReason is a mixin class which defines an indication of how a
8437 conference session ended.

8438
8439 **propertyDefinitions**
8440 The values for this attribute are defined in Section 4.11.7.3.

8441 **4.11.7.3 Property Definitions**

8442 The ConferenceSessionEndingReason class MAY include additional property definitions which are
8443 implementation-defined.

8444

8445 **4.11.8 ConferenceSessionEndingReasonEnum**

8446 The ConferenceSessionEndingReasonEnum class is an enum class that enumerates the instances each
8447 of which expresses a reason for ending a conference session.

8448 The ConferenceSessionEndingReasonEnum class has attribute values:

8449

8450 **localNamespace**

8451 Value: icom_conf

8452

8453 **localName**

8454 Value: ConferenceSessionEndingReasonEnum

8455

8456 **extendsFrom**

8457 Value: ConferenceSessionEndingReason

8458

8459 **stereotype**

8460 Value: primary

8461

8462 **isEnumeration**

8463 Value: TRUE

8464

8465 **description**

8466 Value: ~~An enumeration of the instances each of which expresses a reason~~Reason for ending a

8467 conference session.

8468

8469 **instances**

8470 Value: <icom_conf:HostLeft, icom_conf:HostAborted, icom_conf:NoHost, icom_conf:Hibernating>

8471

8472 ~~There are~~ICOM defines four conference session states ~~defined by ICOM:~~

8473

- 8474 • ~~icom_conf:HostLeft to express that~~ a conference session ended after the host left.
- 8474 • ~~icom_conf:HostAborted to express that~~ a conference session ended after the host aborted it.
- 8475 • ~~icom_conf:NoHost to express that~~ a conference session ended due to no one hosting.
- 8476 • ~~icom_conf:Hibernating to express that~~ a conference session is hibernating.

8477

8478 ~~4.11.6~~4.11.9 **ConferenceSetting**

8479 ~~4.11.6.14~~4.11.9.1 **Description**

8480 A conference setting represents configuration settings for sessions of a conference.

8481 ~~4.11.6.24~~4.11.9.2 **Class Definition**

8482 The ConferenceSetting class ~~is defined by the~~has attribute values:

8483

8484 **localNamespace**

8485 Value: icom_conf

8486

8487 **localName**

8488 Value: ConferenceSetting

8489

8490 **extendsFrom**
8491 Value:
8492
8493 **stereotype**
8494 Value: primary
8495
8496 **description**
8497 Value: A conference setting represents configuration settings for sessions of a conference.
8498
8499 **propertyDefinitions**
8500 The values for this attribute are defined in Section 4.11.9.3.

8501 ~~4.11.6.3~~4.11.9.3 **Property Definitions**

8502 The ConferenceSetting class inherits property definitions from super classes.
8503 The ConferenceSetting class MUST have the property definitions:

8504
8505 **icom_meta:property**
8506 Description: Configurable properties for a conference.
8507 Required: False
8508 Inherited: False
8509 Property Type: icom_meta:property
8510 Cardinality: Multi
8511 Updatability: Read Write

8512
8513 **icom_conf:participantRole**
8514 Description: Role settings for conference participants.
8515 Required: False
8516 Inherited: False
8517 Property Type: icom_conf:ConferenceParticipantRole
8518 Cardinality: Multi
8519 Updatability: Read Write

8520
8521 **icom_meta:property**
8522 Description: Configurable properties for a conference.
8523 Required: False
8524 Inherited: False
8525 Property Type: icom_meta:property
8526 Cardinality: Multi
8527 Updatability: Read Write

8528
8529 The ConferenceSetting class MAY include additional property definitions which are implementation-
8530 defined.

8531

4.11.74.11.10 ConferenceParticipantRole

4.11.7.14.11.10.1 Description

A conference participant role ~~contains~~defines roles settings for a conference participant.

4.11.7.24.11.10.2 Class Definition

The ConferenceParticipantRole class ~~is defined by the~~has attribute values:

localNamespace

Value: icom_conf

localName

Value: ConferenceParticipantRole

extendsFrom

Value:

stereotype

Value: primary

description

Value: A conference participant role contains roles settings for a conference.

propertyDefinitions

The values for this attribute are defined in Section 4.11.10.3.

4.11.7.34.11.10.3 Property Definitions

The ConferenceParticipantRole class MUST have the property definitions:

icom_core:name

Description: Name of a role setting in a conference.

Required: False

Inherited: False

Property Type: String

Cardinality: Single

Updatability: Read Write

~~icom_conf:key~~

~~Description: One or more sign-on keys to activate a role setting.~~

~~Required: False~~

~~Inherited: False~~

~~Property Type: String~~

~~Cardinality: Multi~~

8572 ~~Updatability: Read Write~~

8573

8574 **icom_conf:keyword**

8575 ~~Description: One or more key words to activate a role setting.~~

8576 ~~Required: False~~

8577 ~~Inherited: False~~

8578 ~~Property Type: String~~

8579 ~~Cardinality: Multi~~

8580 ~~Updatability: Read Write~~

8581

8582 **icom_core:participant**

8583 Description: One or more participants in a role setting.

8584 Required: False

8585 Inherited: False

8586 Property Type: icom_core:Participant

8587 Cardinality: Multi

8588 Updatability: Read Write

8589

8590 **icom_meta:property**

8591 Description: Configurable properties for a role setting.

8592 Required: False

8593 Inherited: False

8594 Property Type: icom_meta:Property

8595 Cardinality: Multi

8596 Updatability: Read Write

8597

8598 **icom_conf:key**

8599 ~~Description: One or more sign on keys to activate a role setting.~~

8600 ~~Required: False~~

8601 ~~Inherited: False~~

8602 ~~Property Type: String~~

8603 ~~Cardinality: Multi~~

8604 ~~Updatability: Read Write~~

8605

8606 **icom_conf:keyword**

8607 ~~Description: One or more key words to activate a role setting.~~

8608 ~~Required: False~~

8609 ~~Inherited: False~~

8610 ~~Property Type: String~~

8611 ~~Cardinality: Multi~~

8612 ~~Updatability: Read Write~~

8613

8614 The ConferenceParticipantRole class MAY include additional property definitions which are
8615 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 ~~in terms of the~~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 ~~to be~~ 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 to the ICOM model is defined by use case roles.~~

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 ~~these classes and inherited~~ super classes ~~as and related classes~~ defined in Section 3 of this standard.

Note: ~~A service provider may choose one or more extension modules to support in an ICOM environment. It may be the case that include multiple service providers are involved in an ICOM environment to provide each of which provides different subsets of the extension modules.~~

~~Note: Examples of optional statements of the core ICOM model in Section 3 include 1) Artifact to extend from Spaceltem and 2) Group to extend from Owner.~~

~~Note: An example of optional statements of the extension modules in Section 4 is the version type called RepresentativeCopy that provides a version-independent view of a versionable artifact.~~

Conformance by ICOM producer:

1. An ICOM producer that produces objects of a class:
 - a. SHALL conform to all mandatory statements and
 - b. MAY conform to optional statementsfor the class and super classes thereof in Section 3 of this standard, for any object produced.
2. An ICOM producer may support one or more extension modules as defined in Section 4 of this standard. ICOM producers that support 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.

~~Note: Implementations in the ICOM producer role are not required to produce any particular ICOM objects, but any which are produced to be managed by an ICOM service provider, must conform to this standard.~~

Conformance by ICOM consumer:

1. An ICOM consumer that consumes objects of a class:
 - a. SHALL conform to all mandatory statements and
 - b. MAY conform to optional statementsfor the class and super classes thereof in Section 3 of this standard, for any object consumed.
2. An ICOM consumer may support one or more extension modules as defined in Section 4 of this standard. ICOM consumers that support 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.

~~Note: Implementations in the ICOM consumer role are not required to consume any particular ICOM objects, but any which are consumed must conform to this standard.~~

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

~~Non-Normative Text~~

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.