



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

Committee Specification 01

31 January 2013

Specification URIs

This version:

<http://docs.oasis-open.org/icom/icom-ics/v1.0/cs01/icom-ics-v1.0-cs01.doc> (Authoritative)
<http://docs.oasis-open.org/icom/icom-ics/v1.0/cs01/icom-ics-v1.0-cs01.html>
<http://docs.oasis-open.org/icom/icom-ics/v1.0/cs01/icom-ics-v1.0-cs01.pdf>

Previous version:

<http://www.oasis-open.org/committees/download.php/46823/icom-ics-v1.0-csprd04.zip>

Latest version:

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

Technical Committee:

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

Chairs:

Eric S. Chan (eric.s.chan@oracle.com), Oracle
Kenneth P. Baclawski (kenb@ccs.neu.edu), Northeastern University

Editors:

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

Additional artifacts:

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

- XML schemas: <http://docs.oasis-open.org/icom/icom-ics/v1.0/cs01/schemas/>

Declared XML namespaces:

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

Abstract:

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

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

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

Status:

This document was last revised or approved by the OASIS Integrated Collaboration Object Model for Interoperable Collaboration Services (ICOM) TC on the above date. The level of approval is also listed above.

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

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

Citation format:

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

[ICOM-ics-v1.0]

Integrated Collaboration Object Model (ICOM) for Interoperable Collaboration Services Version 1.0. 31 January 2013. OASIS Committee Specification 01.

<http://docs.oasis-open.org/icom/icom-ics/v1.0/cs01/icom-ics-v1.0-cs01.html>.

Notices

Copyright © OASIS Open 2013. 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 <https://www.oasis-open.org/policies-guidelines/trademark> for above guidance.

Table of Contents

1	Introduction	11
1.1	Terminology	12
1.2	Normative References	12
1.3	Non-Normative References	12
2	Modeling Language	14
2.1	Introduction	14
2.2	Class Definition Grammar	14
2.3	Property Definition Grammar	16
2.4	Namespaces	19
3	Core Model	20
3.1	Main Branch	20
3.1.1	Entity and Top-Level Subclasses	20
3.1.2	Identifiable	20
3.1.3	Parental	21
3.1.4	Extent	22
3.1.5	Entity	23
3.1.6	Overview of Scope, Subject, and Artifact Branches	27
3.2	Scope Branch	28
3.2.1	Scope and Top-Level Subclasses	28
3.2.2	Scope	29
3.2.3	Community	31
3.2.4	Space	33
3.3	Subject Branch	35
3.3.1	Subject and Top-Level Subclasses	35
3.3.2	Subject	35
3.3.3	Group	37
3.3.4	Actor	39
3.3.5	Person	41
3.3.6	Resource	45
3.3.7	ResourceType	47
3.3.8	ResourceTypeEnum	48
3.3.9	ResourceBookingRule	49
3.3.10	ResourceBookingRuleEnum	50
3.4	Artifact Branch	51
3.4.1	Artifact and Top-Level Subclasses	51
3.4.2	Item	51
3.4.3	SpaceItem	53
3.4.4	Container	53
3.4.5	FolderContainer	54
3.4.6	Artifact	55
3.4.7	Folder	58
3.4.8	HeterogeneousFolder	59
3.5	Access Control Model	61

3.5.1	Accessor	61
3.5.2	Owner	61
3.5.3	RoleDefinition	62
3.5.4	Role	63
3.5.5	Privilege	65
3.5.6	PrivilegeEnum	66
3.5.7	AccessControlList	67
3.5.8	AccessControlEntry	68
3.5.9	AccessType	69
3.5.10	AccessTypeEnum	70
3.6	Metadata Model	71
3.6.1	ClassDefinition	71
3.6.2	StereoType	74
3.6.3	StereoTypeEnum	75
3.6.4	PropertyDefinition	76
3.6.5	Property	79
3.6.6	PropertyChoiceType	81
3.6.7	PropertyType	82
3.6.8	PropertyTypeEnum	83
3.6.9	Updatability	84
3.6.10	UpdatabilityEnum	85
3.6.11	Cardinality	86
3.6.12	CardinalityEnum	86
3.6.13	Marker and Subclasses	87
3.6.14	Marker	88
3.6.15	Category	89
3.6.16	CategoryApplication	91
3.6.17	Tag	92
3.6.18	TagApplication	94
3.6.19	RelationshipBondable	95
3.6.20	RelationshipDefinition	96
3.6.21	Relationship	98
3.7	Common Concepts	100
3.7.1	Addressable	100
3.7.2	EntityAddress	101
3.7.3	Participant	102
3.7.4	Priority	104
3.7.5	PriorityEnum	105
3.7.6	DateTimeResolution	105
3.7.7	DateTimeResolutionEnum	106
3.7.8	TimeZone	107
3.7.9	Location	108
3.7.10	GeoCoordinates	110
4	Extension Modules	112
4.1	Overview of Extension Modules	112

4.2 Content Module	114
4.2.1 MimeConvertible	114
4.2.2 Content	114
4.2.3 MultiContent	116
4.2.4 SimpleContent	117
4.2.5 OnlineContent.....	119
4.2.6 ContentDispositionType	120
4.2.7 ContentDispositionTypeEnum.....	121
4.2.8 AttachedItem	122
4.3 Document Module.....	123
4.3.1 Versionable.....	123
4.3.2 VersionControlMetadata.....	125
4.3.3 VersionSeries	127
4.3.4 Version	129
4.3.5 VersionType	131
4.3.6 VersionTypeEnum.....	132
4.3.7 Document	133
4.3.8 WikiPage	134
4.4 Message Module	136
4.4.1 Message	136
4.4.2 UnifiedMessage.....	137
4.4.3 UnifiedMessageParticipant.....	142
4.4.4 UnifiedMessageFlag.....	143
4.4.5 UnifiedMessageFlagEnum	144
4.4.6 UnifiedMessageDeliveryStatusNotificationRequest	145
4.4.7 UnifiedMessageDeliveryStatusNotificationRequestEnum	145
4.4.8 UnifiedMessageChannel	146
4.4.9 UnifiedMessageChannelEnum.....	147
4.4.10 UnifiedMessageEditMode	148
4.4.11 UnifiedMessageEditModeEnum.....	148
4.4.12 InstantMessage	150
4.4.13 InstantMessageType	153
4.4.14 InstantMessageTypeEnum.....	154
4.4.15 InstantMessageChatStatus	155
4.4.16 InstantMessageChatStatusEnum.....	155
4.4.17 InstantMessageFeed.....	156
4.4.18 InstantMessageConnection.....	158
4.5 Presence Module.....	161
4.5.1 Presence	161
4.5.2 PresenceEditMode	163
4.5.3 PresenceEditModeEnum.....	164
4.5.4 ContactMethod	165
4.5.5 ContactReachabilityStatus	167
4.5.6 ContactReachabilityStatusEnum.....	167
4.5.7 Activity	170

4.5.8	ActivityType	171
4.5.9	ActivityTypeEnum.....	172
4.6	Address Book Module.....	173
4.6.1	AddressBook	173
4.6.2	PersonContact.....	174
4.7	Calendar Module	179
4.7.1	Calendar	179
4.7.2	OccurrenceSeries.....	181
4.7.3	Occurrence	186
4.7.4	OccurrenceStatus.....	192
4.7.5	OccurrenceStatusEnum	192
4.7.6	OccurrenceType	193
4.7.7	OccurrenceTypeEnum	194
4.7.8	OccurrenceParticipant.....	195
4.7.9	OccurrenceParticipantStatus.....	196
4.7.10	OccurrenceParticipantStatusEnum	196
4.7.11	OccurrenceParticipantTransparency.....	197
4.7.12	OccurrenceParticipantTransparencyEnum	198
4.7.13	OccurrenceEditMode.....	199
4.7.14	OccurrenceEditModeEnum	200
4.8	Free Busy Module.....	200
4.8.1	FreeBusy	200
4.8.2	FreeBusyInterval	202
4.8.3	FreeBusyType	204
4.8.4	FreeBusyTypeEnum.....	205
4.9	Task List Module.....	206
4.9.1	TaskList	206
4.9.2	Task.....	207
4.9.3	TaskStatus.....	212
4.9.4	TaskStatusEnum	212
4.9.5	TaskParticipantStatus	213
4.9.6	TaskParticipantStatusEnum	214
4.9.7	TaskEditMode.....	215
4.9.8	TaskEditModeEnum	215
4.10	Forum Module.....	216
4.10.1	Discussion	216
4.10.2	DiscussionContainer	217
4.10.3	DiscussionMessage	218
4.10.4	TopicContainer	219
4.10.5	Forum	221
4.10.6	Topic.....	223
4.10.7	Announcement	224
4.10.8	AnnouncementStatus	226
4.10.9	AnnouncementStatusEnum.....	226
4.11	Conference Module	227

4.11.1	Conference	227
4.11.2	ConferenceType	230
4.11.3	ConferenceTypeEnum	231
4.11.4	ConferenceStatus	232
4.11.5	ConferenceStatusEnum	233
4.11.6	ConferenceSession	233
4.11.7	ConferenceSessionEndingReason	236
4.11.8	ConferenceSessionEndingReasonEnum	236
4.11.9	ConferenceSetting	237
4.11.10	ConferenceParticipantRole	238
5	Conformance	241
5.1	Software Architecture or Framework Dependence	241
5.2	Platform Provider Conformance	241
5.2.1	Platform Provider Conformance – No Extension Modules	241
5.2.2	Platform Provider Conformance – One or More Extension Modules	241
5.3	Service Provider Conformance	241
5.3.1	ICOM Service Provider – No Extension Modules	241
5.3.2	ICOM Service Provider – One or More Extension Modules	241
5.4	ICOM Producer Conformance	242
5.4.1	ICOM Producer Conformance – No Extension Modules	242
5.4.2	ICOM Producer Conformance – One or More Extension Modules	242
5.5	ICOM Consumer Conformance	242
5.5.1	ICOM Consumer Conformance – No Extension Modules	242
5.5.2	ICOM Consumer Conformance – Extension Modules	242
Appendix A.	Acknowledgements	243
Appendix B.	Revision History	244

Table of Figures

Figure 1: Entity and Top-Level Abstract Classes.....	20
Figure 2: Entity Class Diagram.....	27
Figure 3: Scope, Subject, and Artifact Branches.....	28
Figure 4: Scope Branch.....	28
Figure 5: Scope Class Diagram.....	31
Figure 6: Community Class Diagram.....	33
Figure 7: Space Class Diagram.....	34
Figure 8: Subject Branch.....	35
Figure 9: Subject Class Diagram.....	37
Figure 10: Group and Actor Class Diagram.....	39
Figure 11: Person Class Diagram.....	45
Figure 12: Resource Class Diagram.....	47
Figure 13: Artifact Branch.....	51
Figure 14: Artifact Class Diagram.....	58
Figure 15: Heterogeneous Folder Class Diagram.....	60
Figure 16: Role Definition and Role Class Diagram.....	65
Figure 17: Access Control List Class Diagram.....	71
Figure 18: Class Definition and Property Definition Class Diagram.....	74
Figure 19: Property Definition and Property Class Diagram.....	81
Figure 20: Marker Branch.....	87
Figure 21: Marker Class Diagram.....	89
Figure 22: Category and Category Application Class Diagram.....	91
Figure 23: Tag and Tag Application Class Diagram.....	94
Figure 24: Relationship Class Diagram.....	100
Figure 25: Containers of Collaboration Activities.....	112
Figure 26: Composite Content Class Diagram.....	116
Figure 27: Document, Version Series, and Version Class Diagram.....	134
Figure 28: Wiki Page Class Diagram.....	136
Figure 29: Unified Message Class Diagram.....	150
Figure 30: Instant Message Class Diagram.....	153
Figure 31: Instant Message Feed and Connection Class Diagram.....	158
Figure 32: Presence Class Diagram.....	163
Figure 33: Presence Contact Method and Instant Message Connection Class Diagram.....	169
Figure 34: Address Book Class Diagram.....	174
Figure 35: Person Contact Class Diagram.....	179
Figure 36: Calendar Class Diagram.....	181
Figure 37: Occurrence Series Class Diagram.....	186
Figure 38: Occurrence Class Diagram.....	191
Figure 39: Free Busy Class Diagram.....	204
Figure 40: Task List Class Diagram.....	207

Figure 41: Task Class Diagram.	211
Figure 42: Forum Class Diagram.....	221
Figure 43: Conference Class Diagram.....	230

1 Introduction

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

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

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

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

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

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

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 **[RFC2060]** 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)

114 2 Modeling Language

115 2.1 Introduction

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

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

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

131 Note: ICOM extends the CMIS base types in several ways. ICOM Relationship class defined in Section
132 3.6.21 can represent n-ary relationships whereas CMIS Relationship base type represents binary
133 relationships. ICOM version control model defined in Section 4.3.1 adopts the CMIS version control
134 model and extends it with the concept of representative copy.

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

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

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

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

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

150

151 2.2 Class Definition Grammar

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

153 **namespace** String

154 The `namespace` attribute specifies an IRI.

155

156 **localName** String

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

158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200

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.

201
202 **instances**
203 The `instances` attribute enumerates instances of an enum class. It is applicable only when the
204 value of `stereotype` attribute is **Primary** and the value of `isEnumeration` attribute is **TRUE**.
205
206 **propertyDefinition** **property-definition** (multi-valued)
207 The `propertyDefinition` attribute defines a set of zero or more property definitions for a
208 class.
209 Property definitions of a class are a union of inherited property definitions from super classes and
210 property definitions explicitly defined on a class.
211 The order of property definitions within a class is not significant.
212 Property definitions **MUST** be uniquely named to avoid conflicts from multiple inheritances.
213 Note: It is possible for the same property definition to be inherited through different paths in a
214 super class hierarchy. Duplicate property definitions are eliminated from the set of property
215 definitions of a class.
216

217 2.3 Property Definition Grammar

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

219 **namespace** String

220 The `namespace` attribute specifies an IRI.

221

222 **localName** String

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

225

226 **description** String (optional)

227 The `description` attribute specifies a description of a property

228

229 **propertyType** Enum

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

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

- 234 • `string` (xsd:string)
- 235 • `boolean` (xsd:boolean)
- 236 • `decimal` (xsd:decimal)
- 237 • `integer` (xsd:integer)
- 238 • `datetime` (xsd:dateTime)
- 239 • `duration` (xsd:duration)
- 240 • `iri` (xsd:anyURI)

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

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

244

245 **cardinality** Enum

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

247 The values of `cardinality` attribute are:

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

252

253 **updatability** Enum

254 The `updatability` attribute specifies under what circumstances the value of this property MAY

255 be updated.

256 The values of `updatability` attribute are:

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

264

265 **inherited** Boolean

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

267 The values of `inherited` attribute are:

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

270

271 **required** Boolean

272 The `required` attribute is only applicable to read-write and on-create properties, i.e. properties

273 whose value is provided by application.

274 The values of `required` attribute are:

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

282 This attribute is not applicable when the value `updatability` attribute is **ReadOnly**. In that

283 case, `required` attribute SHOULD be set to **FALSE**.

284 Note: The value of a read-only property (such as `icom_core:objectId`,

285 `icom_core:createdBy`) is set by a service provider. Hence, the value of the `required`

286 attribute SHOULD be **FALSE** because it is read only for applications.

287

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;

311 • **FALSE** if a value of a property **MUST** be among those listed in `choices` attribute.

312

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

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

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

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

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

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

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

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

321 `propertyType` attribute of a property definition.

322

323 **minValue** Integer | Decimal

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

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

326

327 **maxValue** Integer | Decimal

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

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

330

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

332 **displayName** String

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

334

335 **value** **property-type**

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

338

339 2.4 Namespaces

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

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

344

345 *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

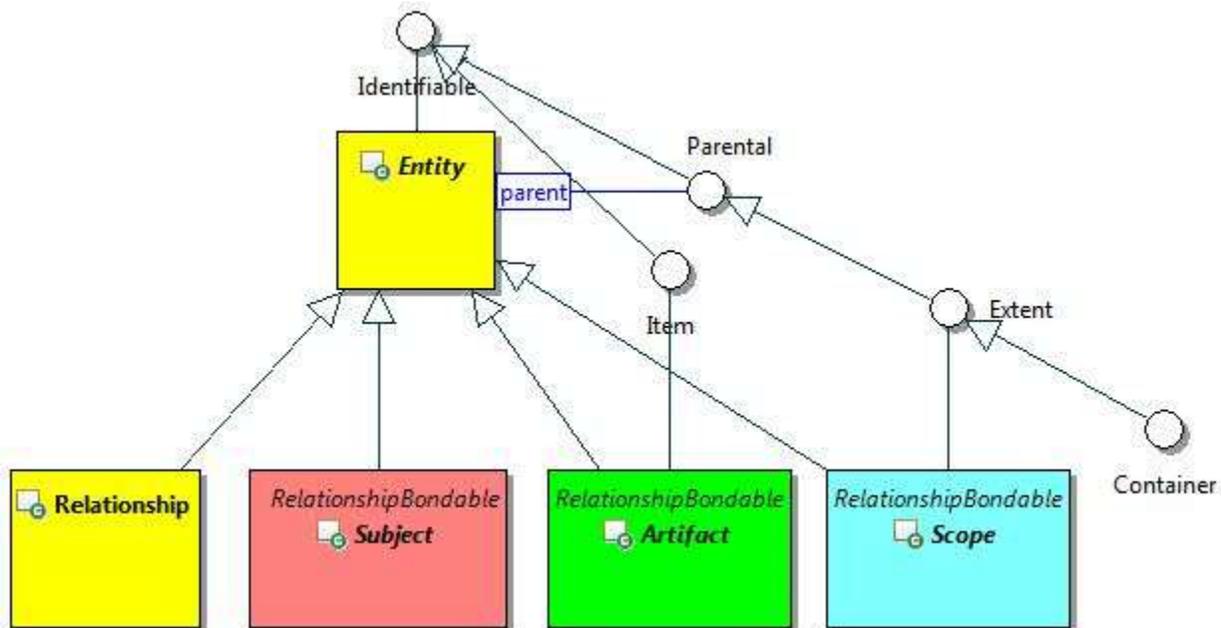
346

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

351 3 Core Model

352 3.1 Main Branch

353 3.1.1 Entity and Top-Level Subclasses



354
355 Figure 1: Entity and Top-Level Abstract Classes.

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

359 3.1.2 Identifiable

360 3.1.2.1 Description

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

363 3.1.2.2 Class Definition

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

366 The *Identifiable* class has attribute values:

367
368 **localNamespace**
369 Value: icom_core
370
371 **localName**
372 Value: Identifiable

373
 374 **extendsFrom**
 375 Value:
 376
 377 **stereotype**
 378 Value: mixin
 379
 380 **description**
 381 Value: Identifiable is a mixin class which defines the characteristics of all entities and some non-
 382 entities that enables unique identification.
 383
 384 **propertyDefinitions**
 385 The values for this attribute are defined in Section 3.1.2.3.

386 **3.1.2.3 Property Definitions**

387 The Identifiable class **MUST** have the property definitions:

389 **icom_core:objectId**

390	Description:	A persistent identifier of an object.
391	Required:	False
392	Inherited:	False
393	Property Type:	String
394	Cardinality:	Single
395	Updatability:	Read Only

397 **icom_core:changeToken**

398	Description:	An opaque token used for optimistic locking & concurrency checking.
399		
400	Required:	False
401	Inherited:	False
402	Property Type:	String
403	Cardinality:	Single
404	Updatability:	Read Only

405
 406 The Identifiable class **MAY** include additional property definitions which are implementation-defined.
 407

408 **3.1.3 Parental**

409 **3.1.3.1 Description**

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

411 **3.1.3.2 Class Definition**

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

414 The Parental class has attribute values:

415

416 **localNamespace**

417 Value: icom_core

418

419 **localName**

420 Value: Parental

421

422 **extendsFrom**

423 Value: icom_core:Identifiable

424

425 **stereotype**

426 Value: mixin

427

428 **description**

429 Value: Parental is a mixin class which defines the characteristics of the entities that can be
430 parents of other entities or identifiable objects.

431

432 **propertyDefinitions**

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

434 **3.1.3.3 Property Definitions**

435 The Parental class inherits property definitions from super classes.

436 The Parental class **MUST** have the property definition:

437

438 **icom_core:parent**

439 Description: Parent of an object.

440 Required: False

441 Inherited: False

442 Property Type: icom_core:Parental

443 Cardinality: Single

444 Updatability: Read Only

445

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

447

448 **3.1.4 Extent**

449 **3.1.4.1 Description**

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

451 **3.1.4.2 Class Definition**

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

453 The Extent class has attribute values:

454

455 **localNamespace**

456 Value: icom_core

457

458 **localName**

459 Value: Extent

460

461 **extendsFrom**

462 Value: icom_core:Parental

463

464 **stereotype**

465 Value: mixin

466

467 **description**

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

470

471 **propertyDefinitions**

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

473 **3.1.4.3 Property Definitions**

474 The Extent class inherits property definitions from super classes.

475 The Extent class MUST have the property definition:

476

477 **icom_core:parent**

478 Description: Parent of an extent.

479 Required: False

480 Inherited: True

481 Property Type: icom_core:Extent

482 Cardinality: Single

483 Updatability: Read Only

484

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

486

487 **3.1.5 Entity**

488 **3.1.5.1 Description**

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

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

492 3.1.5.2 Class Definition

493 The Entity class has attribute values:

494

495 **localNamespace**

496 Value: icom_core

497

498 **localName**

499 Value: Entity

500

501 **extendsFrom**

502 Value: icom_core:Identifiable

503

504 **stereotype**

505 Value: primary

506

507 **isAbstract**

508 Value: TRUE

509

510 **description**

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

512

513 **propertyDefinitions**

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

515 3.1.5.3 Property Definitions

516 The Entity class inherits property definitions from super classes.

517 The Entity class MUST have the property definitions:

518

519 **icom_core:name**

520 Description: Name of an entity.

521 Required: False

522 Inherited: False

523 Property Type: String

524 Cardinality: Single

525 Updatability: Read Write

526

527 **icom_core:createdBy**

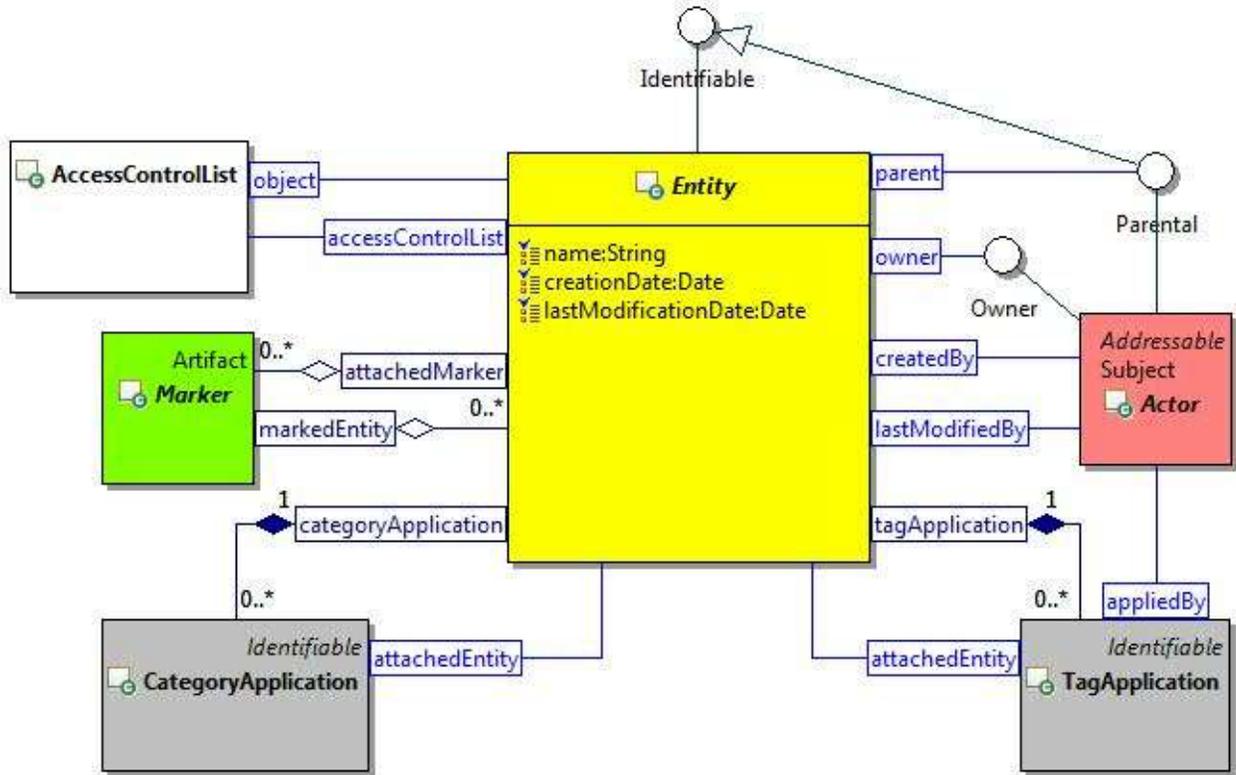
528 Description: An actor who created an entity.

529 Required: False

530 Inherited: False

531	Property Type:	icom_core:Actor
532	Cardinality:	Single
533	Updatability:	Read Only
534		
535	icom_core:creationDate	
536	Description:	Date and time when an entity is created. It is immutable.
537	Required:	False
538	Inherited:	False
539	Property Type:	DateTime
540	Cardinality:	Single
541	Updatability:	Read Only
542		
543	icom_core:lastModifiedBy	
544	Description:	An actor who last modified an entity.
545	Required:	False
546	Inherited:	False
547	Property Type:	icom_core:Actor
548	Cardinality:	Single
549	Updatability:	Read Only
550		
551	icom_core:lastModificationDate	
552	Description:	Date and time of last modification.
553	Required:	False
554	Inherited:	False
555	Property Type:	DateTime
556	Cardinality:	Single
557	Updatability:	Read Only
558		
559	icom_core:parent	
560	Description:	A parental entity which contains an entity.
561	Required:	False
562	Inherited:	False
563	Property Type:	icom_core:Parental
564	Cardinality:	Single
565	Updatability:	Read Only
566		
567	icom_ac:owner	
568	Description:	A subject who owns an entity.
569	Required:	True
570	Inherited:	False
571	Property Type:	icom_ac:Owner
572	Cardinality:	Single

573	Updatability:	Read Write
574		
575	icom_ac:accessControlList	
576	Description:	Access control list on an entity.
577	Required:	False
578	Inherited:	False
579	Property Type:	icom_ac:AccessControlList
580	Cardinality:	Single
581	Updatability:	Read Write
582		
583	icom_meta:attachedMarker	
584	Description:	Zero or more markers applied on an entity.
585	Required:	False
586	Inherited:	False
587	Property Type:	icom_meta:Marker
588	Cardinality:	Multi
589	Updatability:	Read Only
590		
591	icom_meta:categoryApplication	
592	Description:	Zero or more category applications on an entity.
593	Required:	False
594	Inherited:	False
595	Property Type:	icom_meta:CategoryApplication
596	Cardinality:	Multi
597	Updatability:	Read Only
598		
599	icom_meta:tagApplication	
600	Description:	Zero or more tag applications on an entity.
601	Required:	False
602	Inherited:	False
603	Property Type:	icom_meta:TagApplication
604	Cardinality:	Multi
605	Updatability:	Read Only
606		
607	The Entity class MAY include additional property definitions which are implementation-defined.	
608		



609
 610 *Figure 2: Entity Class Diagram.*
 611

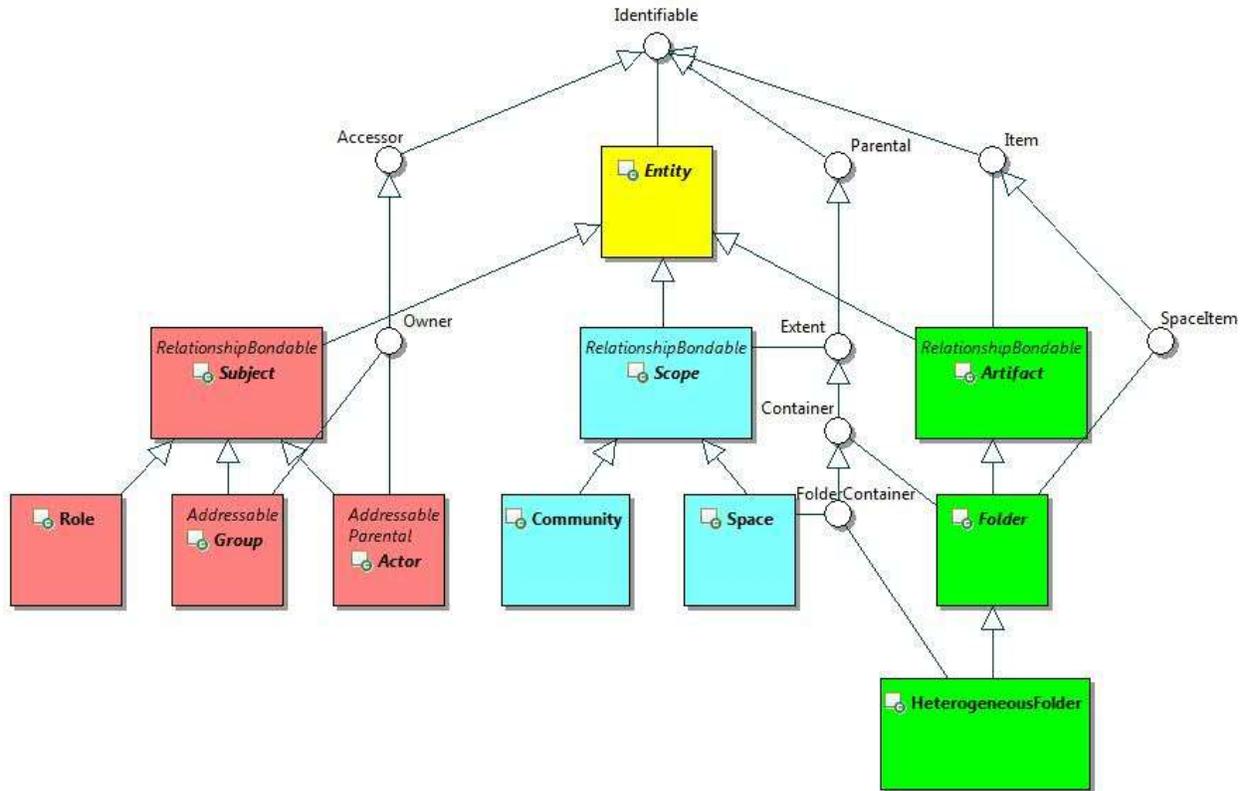
612 **3.1.6 Overview of Scope, Subject, and Artifact Branches**

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

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

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

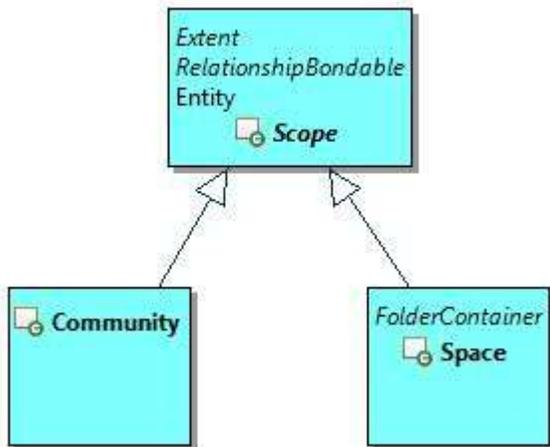
624



625
 626 *Figure 3: Scope, Subject, and Artifact Branches.*
 627

628 3.2 Scope Branch

629 3.2.1 Scope and Top-Level Subclasses



630
 631 *Figure 4: Scope Branch.*

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

633 **3.2.2 Scope**

634 **3.2.2.1 Description**

635 A scope is an extent of an administrative policy.

636 **3.2.2.2 Class Definition**

637 The Scope class has attribute values:

638

639 **localNamespace**

640 Value: icom_core

641

642 **localName**

643 Value: Scope

644

645 **extendsFrom**

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

647

648 **stereotype**

649 Value: primary

650

651 **isAbstract**

652 Value: TRUE

653

654 **description**

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

656

657 **propertyDefinitions**

658 The values for this attribute are defined in Section 3.2.2.3.

659 **3.2.2.3 Property Definitions**

660 The Scope class inherits property definitions from super classes.

661 The Scope class MUST have the property definitions:

662

663 **icom_core:description**

664 Description: A description of a scope.

665 Required: False

666 Inherited: False

667 Property Type: String

668 Cardinality: Single

669 Updatability: Read Write

670

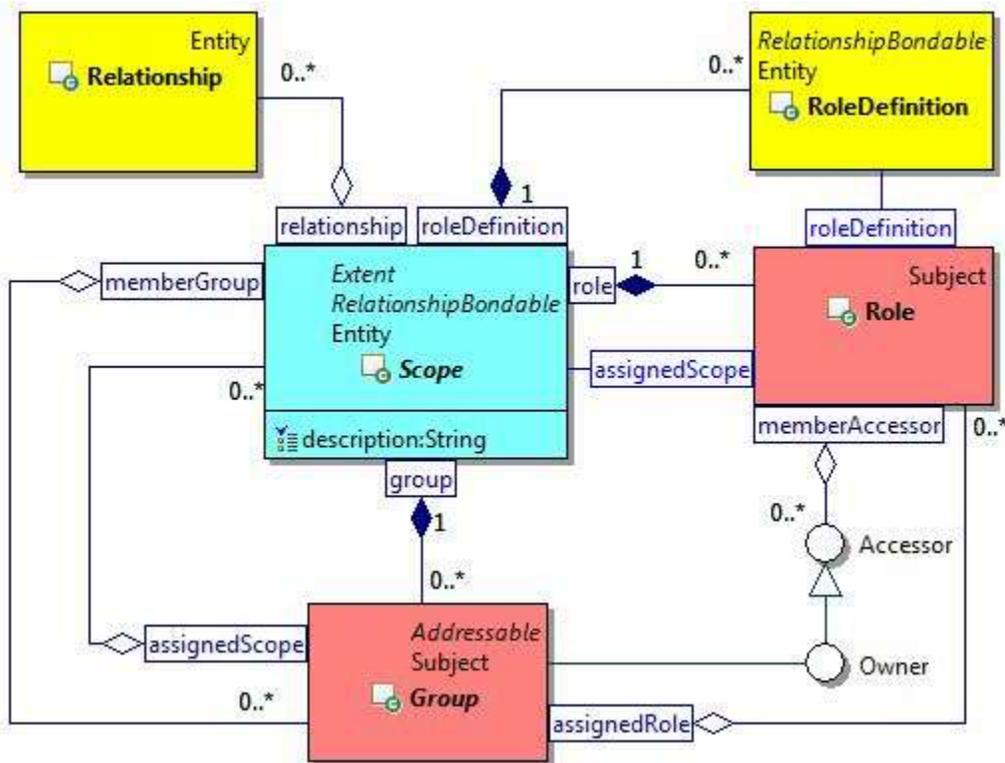
671 **icom_core:parent**

672 Description: A community which contains a scope.

673	Required:	False
674	Inherited:	True
675	Property Type:	icom_core:Community
676	Cardinality:	Single
677	Updatability:	Read Only
678		
679	icom_core:group	
680	Description:	Zero or more groups defined in a scope.
681	Required:	False
682	Inherited:	False
683	Property Type:	icom_core:Group
684	Cardinality:	Multi
685	Updatability:	Read Only
686		
687	icom_core:memberGroup	
688	Description:	Member groups of a scope, i.e. groups whose assigned
689		scopes include this scope.
690	Required:	False
691	Inherited:	False
692	Property Type:	icom_core:Group
693	Cardinality:	Multi
694	Updatability:	Read Only
695		
696	icom_ac:roleDefinition	
697	Description:	Zero or more role definitions defined in a scope.
698	Required:	False
699	Inherited:	False
700	Property Type:	icom_ac:RoleDefinition
701	Cardinality:	Multi
702	Updatability:	Read Only
703		
704	icom_ac:role	
705	Description:	Zero or more roles defined in a scope.
706	Required:	False
707	Inherited:	False
708	Property Type:	icom_ac:Role
709	Cardinality:	Multi
710	Updatability:	Read Only
711		
712	icom_meta:relationship	
713	Description:	Zero or more relationships associated with a scope.
714	Required:	False
715	Inherited:	False

716 Property Type: icom_meta:Relationship
 717 Cardinality: Multi
 718 Updatability: Read Only

719
 720 The Scope class MAY include additional property definitions which are implementation-defined.
 721



722
 723 Figure 5: Scope Class Diagram.

724
 725 **3.2.3 Community**

726 **3.2.3.1 Description**

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

730 **3.2.3.2 Class Definition**

731 The Community class has attribute values:

732
 733 **localNamespace**
 734 Value: icom_core

735
 736 **localName**
 737 Value: Community

738
 739 **extendsFrom**
 740 Value: icom_core:Scope
 741
 742 **stereotype**
 743 Value: primary
 744
 745 **description**
 746 Value: A community is a scope that has a set of actors as members who can participate in a set
 747 of spaces.
 748
 749 **propertyDefinitions**
 750 The values for this attribute are defined in Section 3.2.3.3.

751 **3.2.3.3 Property Definitions**

752 The Community class inherits property definitions from super classes.
 753 The Community class **MUST** have the property definitions:

754 **icom_core:community**

756 Description: Sub-communities of a community.
 757 Required: False
 758 Inherited: False
 759 Property Type: icom_core:Community
 760 Cardinality: Multi
 761 Updatability: Read Only

762 **icom_core:space**

764 Description: Spaces of a community.
 765 Required: False
 766 Inherited: False
 767 Property Type: icom_core:Space
 768 Cardinality: Multi
 769 Updatability: Read Only

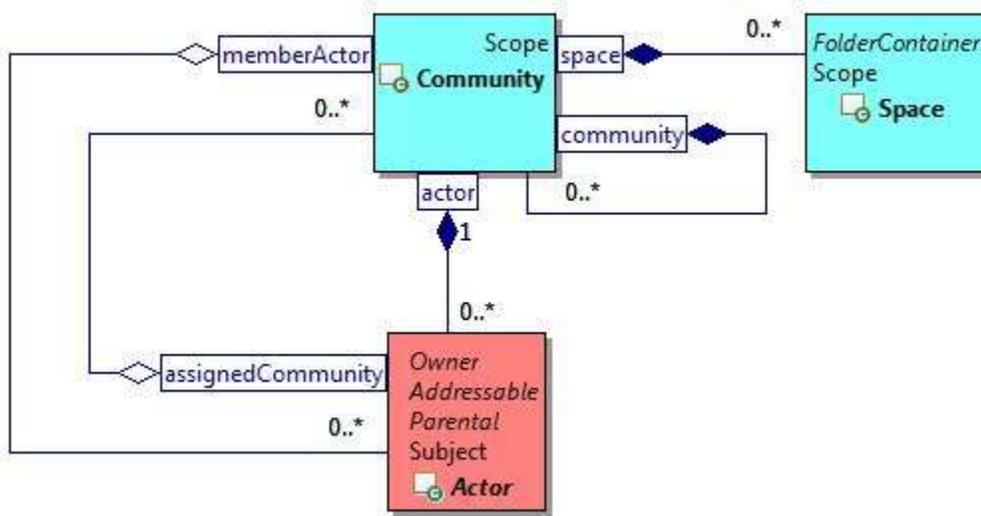
770 **icom_core:actor**

772 Description: Managed actors of a community, i.e. actors whose parent
 773 community is this community.
 774 Required: False
 775 Inherited: False
 776 Property Type: icom_core:Actor
 777 Cardinality: Multi
 778 Updatability: Read Only

779

780 **icom_core:memberActor**
 781 Description: Member actors of a community, i.e. actors whose assigned
 782 communities include this community.
 783 Required: False
 784 Inherited: False
 785 Property Type: icom_core:Actor
 786 Cardinality: Multi
 787 Updatability: Read Only

788
 789 The Community class MAY include additional property definitions which are implementation-defined.
 790



791
 792 *Figure 6: Community Class Diagram.*
 793

794 3.2.4 Space

795 3.2.4.1 Description

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

797 3.2.4.2 Class Definition

798 The Space class has attribute values:

799
 800 **localNamespace**
 801 Value: icom_core
 802
 803 **localName**
 804 Value: Space
 805
 806 **extendsFrom**
 807 Value: icom_core:Scope, icom_core:FolderContainer

808

809 **stereotype**

810 Value: primary

811

812 **description**

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

814 collaborate.

815

816 **propertyDefinitions**

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

818 3.2.4.3 Property Definitions

819 The Space class inherits property definitions from super classes.

820 The Space class MUST have the property definition:

821

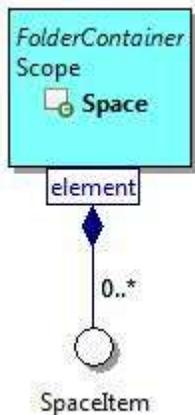
822 **icom_core:element**

823 Description:	Elements of a space.
824 Required:	False
825 Inherited:	True
826 Property Type:	icom_core:SpaceItem
827 Cardinality:	Multi
828 Updatability:	Read Only

829

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

831



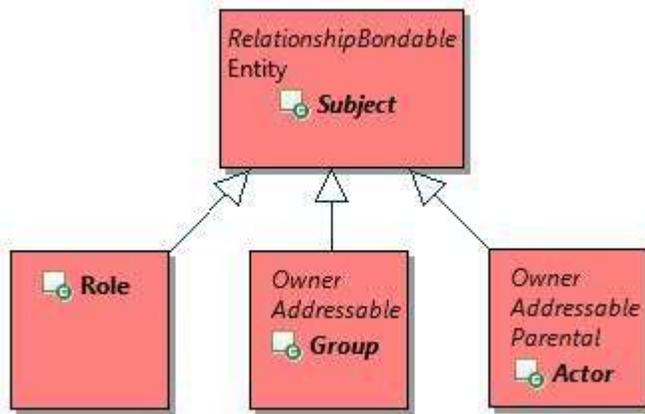
832

833 *Figure 7: Space Class Diagram.*

834

835 3.3 Subject Branch

836 3.3.1 Subject and Top-Level Subclasses



837
838 *Figure 8: Subject Branch.*

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

840 3.3.2 Subject

841 3.3.2.1 Description

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

843 3.3.2.2 Class Definition

844 The Subject class has attribute values:

845

846 **localNamespace**
847 Value: icom_core

848

849 **localName**
850 Value: Subject

851

852 **extendsFrom**
853 Value: icom_core:Entity, icom_meta:RelationshipBondable

854

855 **stereotype**
856 Value: primary

857

858 **isAbstract**
859 Value: TRUE

860

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

863

864 **propertyDefinitions**

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

866 **3.3.2.3 Property Definitions**

867 The Subject class inherits property definitions from super classes.

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

869

870 **icom_core:description**

871 Description: A description of a subject.

872 Required: False

873 Inherited: False

874 Property Type: String

875 Cardinality: Single

876 Updatability: Read Write

877

878 **icom_core:parent**

879 Description: A scope which contains a subject.

880 Required: False

881 Inherited: True

882 Property Type: icom_core:Scope

883 Cardinality: Single

884 Updatability: Read Only

885

886 **icom_meta:relationship**

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

888 Required: False

889 Inherited: False

890 Property Type: icom_meta:Relationship

891 Cardinality: Multi

892 Updatability: Read Only

893

894 **icom_meta:property**

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

896 Required: False

897 Inherited: False

898 Property Type: icom_meta:Property

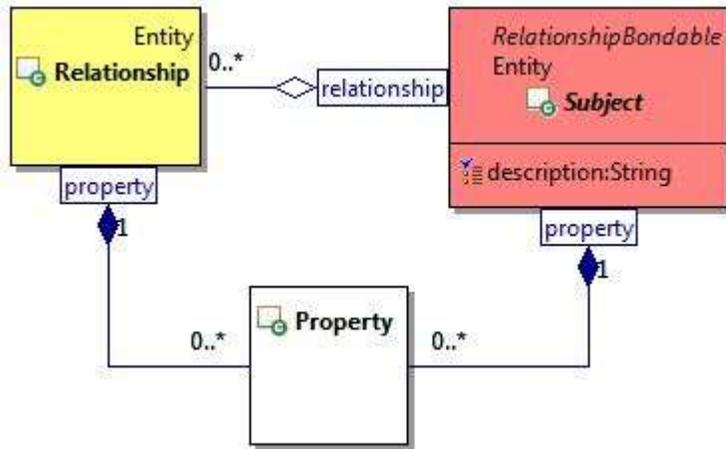
899 Cardinality: Multi

900 Updatability: Read Write

901

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

903



904
905 *Figure 9: Subject Class Diagram.*
906

907 3.3.3 Group

908 3.3.3.1 Description

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

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

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

912 3.3.3.2 Class Definition

913 The Group class has attribute values:

914

915 **localNamespace**

916 Value: icom_core

917

918 **localName**

919 Value: Group

920

921 **extendsFrom**

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

923 Optional Value: icom_ac:Owner

924

925 **stereotype**

926 Value: primary

927

928 **description**

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

931

932 **propertyDefinitions**

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

934 **3.3.3.3 Property Definitions**

935 The Group class inherits property definitions from super classes.

936 The Group class MUST have the property definitions:

937

938 **icom_core:assignedGroup**

939 Description: A group's super-groups.

940 Required: False

941 Inherited: False

942 Property Type: icom_core:Group

943 Cardinality: Multi

944 Updatability: Read Write

945

946 **icom_core:assignedScope**

947 Description: A group's scopes.

948 Required: False

949 Inherited: False

950 Property Type: icom_core:Scope

951 Cardinality: Multi

952 Updatability: Read Write

953

954 **icom_core:memberGroup**

955 Description: Sub-groups of a group.

956 Required: False

957 Inherited: False

958 Property Type: icom_core:Group

959 Cardinality: Multi

960 Updatability: Read Only

961

962 **icom_core:memberActor**

963 Description: Actors in a group.

964 Required: False

965 Inherited: False

966 Property Type: icom_core:Actor

967 Cardinality: Multi

968 Updatability: Read Only

969

970 **icom_ac:assignedRole**

971 Description: A group's roles.

972 Required: False

973 Inherited: False

974 Property Type: icom_ac:Role

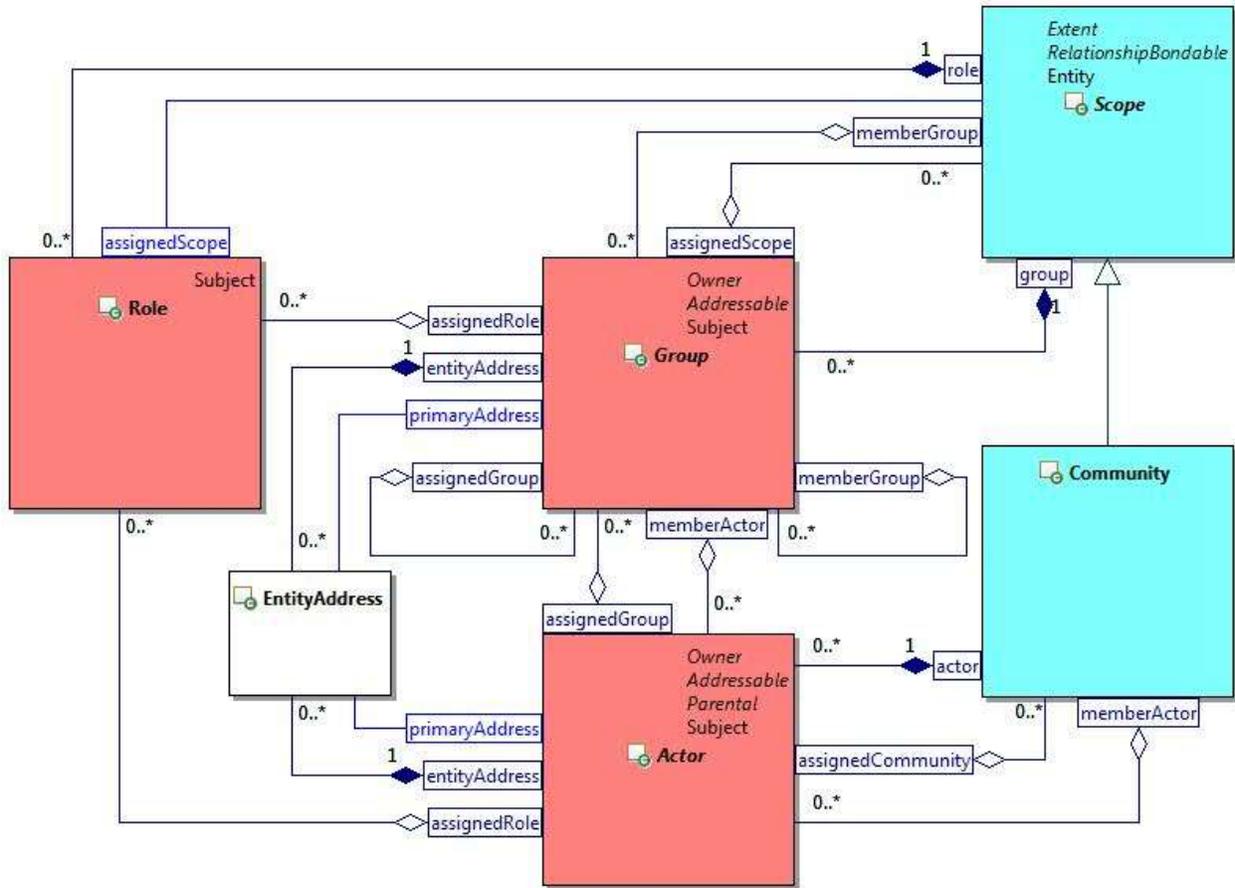
975 Cardinality: Multi

976
977
978
979

Updatability:

Read Write

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



980
981
982

Figure 10: Group and Actor Class Diagram.

983 3.3.4 Actor

984 3.3.4.1 Description

985 An actor is a subject that can perform actions on objects.
986 It can be an owner of entities.

987 3.3.4.2 Class Definition

988 The Actor class has attribute values:

989
990
991
992
993
994

localNamespace

Value: icom_core

localName

Value: Actor

995
 996 **extendsFrom**
 997 Value: icom_core:Subject, icom_core:Addressable, icom_ac:Owner
 998
 999 **stereotype**
 1000 Value: primary
 1001
 1002 **isAbstract**
 1003 Value: TRUE
 1004
 1005 **description**
 1006 Value: An actor is a subject that can perform actions on objects.
 1007
 1008 **propertyDefinitions**
 1009 The values for this attribute are defined in Section 3.3.4.3.

3.3.4.3 Property Definitions

1010
 1011 The Actor class inherits property definitions from super classes.
 1012 The Actor class MUST have the property definitions:

1013
 1014 **icom_core:parent**
 1015 Description: A community which contains an actor.
 1016 Required: False
 1017 Inherited: True
 1018 Property Type: icom_core:Community
 1019 Cardinality: Single
 1020 Updatability: Read Only

1021
 1022 **icom_core:assignedGroup**
 1023 Description: An actor's groups.
 1024 Required: False
 1025 Inherited: False
 1026 Property Type: icom_core:Group
 1027 Cardinality: Multi
 1028 Updatability: Read Write

1029
 1030 **icom_core:assignedCommunity**
 1031 Description: An actor's communities.
 1032 Required: False
 1033 Inherited: False
 1034 Property Type: icom_core:Community
 1035 Cardinality: Multi
 1036 Updatability: Read Write

1037		
1038	icom_ac:assignedRole	
1039	Description:	An actor's roles.
1040	Required:	False
1041	Inherited:	False
1042	Property Type:	icom_ac:Role
1043	Cardinality:	Multi
1044	Updatability:	Read Write

1045

1046 The Actor class MAY include additional property definitions which are implementation-defined.

1047

1048 **3.3.5 Person**

1049 **3.3.5.1 Description**

1050 A person is an individual human who may be an actor.

1051 A person has a personal space.

1052 **3.3.5.2 Class Definition**

1053 The Person class has attribute values:

1054

1055 **localNamespace**

1056 Value: icom_core

1057

1058 **localName**

1059 Value: Person

1060

1061 **extendsFrom**

1062 Value: icom_core:Actor

1063

1064 **stereotype**

1065 Value: primary

1066

1067 **description**

1068 Value: A person is an individual human who may be an actor.

1069

1070 **propertyDefinitions**

1071 The values for this attribute are defined in Section 3.3.5.3.

1072 **3.3.5.3 Property Definitions**

1073 The Person class inherits property definitions from super classes.

1074 The Person class MUST have the property definitions:

1075

1076 **icom_core:givenName**

1077 Description: Given name of a person.

1078 Required: False

1079 Inherited: False

1080 Property Type: String

1081 Cardinality: Single

1082 Updatability: Read Write

1083

1084 **icom_core:middleName**

1085 Description: Middle name of a person. Can include multiple names
1086 concatenated.

1087 Required: False

1088 Inherited: False

1089 Property Type: String

1090 Cardinality: Single

1091 Updatability: Read Write

1092

1093 **icom_core:familyName**

1094 Description: Family name of a person.

1095 Required: False

1096 Inherited: False

1097 Property Type: String

1098 Cardinality: Single

1099 Updatability: Read Write

1100

1101 **icom_core:prefix**

1102 Description: Prefix of a person's name.

1103 Required: False

1104 Inherited: False

1105 Property Type: String

1106 Cardinality: Single

1107 Updatability: Read Write

1108

1109 **icom_core:suffix**

1110 Description: Suffix of a person's name.

1111 Required: False

1112 Inherited: False

1113 Property Type: String

1114 Cardinality: Single

1115 Updatability: Read Write

1116

1117	icom_core:nickname	
1118	Description:	Nickname of a person.
1119	Required:	False
1120	Inherited:	False
1121	Property Type:	String
1122	Cardinality:	Multi
1123	Updatability:	Read Write
1124		
1125	icom_core:jobTitle	
1126	Description:	Job title of a person.
1127	Required:	False
1128	Inherited:	False
1129	Property Type:	String
1130	Cardinality:	Single
1131	Updatability:	Read Write
1132		
1133	icom_core:department	
1134	Description:	A person's affiliated department.
1135	Required:	False
1136	Inherited:	False
1137	Property Type:	String
1138	Cardinality:	Single
1139	Updatability:	Read Write
1140		
1141	icom_core:officeLocation	
1142	Description:	Location of a person's department.
1143	Required:	False
1144	Inherited:	False
1145	Property Type:	String
1146	Cardinality:	Single
1147	Updatability:	Read Write
1148		
1149	icom_core:company	
1150	Description:	A person's affiliated company.
1151	Required:	False
1152	Inherited:	False
1153	Property Type:	String
1154	Cardinality:	Single
1155	Updatability:	Read Write
1156		
1157	icom_core:profession	
1158	Description:	A person's profession.

1159 Required: False
 1160 Inherited: False
 1161 Property Type: String
 1162 Cardinality: Single
 1163 Updatability: Read Write
 1164

icom_core:personalSpace

1166 Description: Personal space of a person.
 1167 Required: False
 1168 Inherited: False
 1169 Property Type: icom_core:Space
 1170 Cardinality: Single
 1171 Updatability: Read Only
 1172

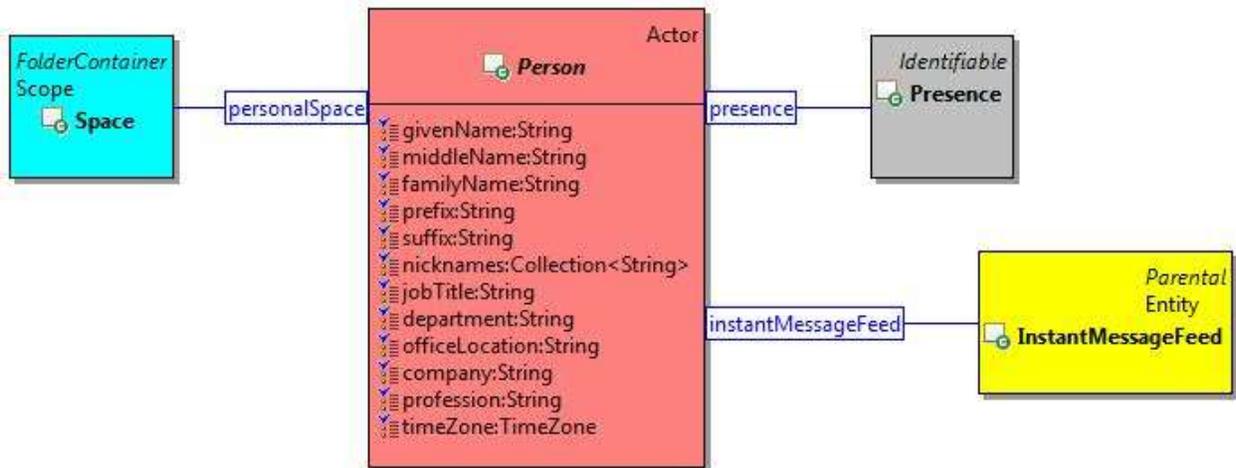
icom_presence:presence

1174 Description: Presence of a person.
 1175 Required: False
 1176 Inherited: False
 1177 Property Type: icom_presence:Presence
 1178 Cardinality: Single
 1179 Updatability: Read Only
 1180

icom_msg:instantMessageFeed

1182 Description: Instant message feed for a person.
 1183 Required: False
 1184 Inherited: False
 1185 Property Type: icom_msg:InstantMessageFeed
 1186 Cardinality: Single
 1187 Updatability: Read Only
 1188

1189 The Person class MAY include additional property definitions which are implementation-defined.
 1190



1191
1192 *Figure 11: Person Class Diagram.*
1193

1194 3.3.6 Resource

1195 3.3.6.1 Description

1196 A resource is an actor representing a bookable resource, such as a conference room, equipment, or on-
1197 line conference.
1198 A resource is associated with a resource space that contains a resource scheduling calendar and on-line
1199 conference.

1200 3.3.6.2 Class Definition

1201 The Resource class has attribute values:

- 1202
- 1203 **localNamespace**
 - 1204 Value: icom_core
 - 1205
 - 1206 **localName**
 - 1207 Value: Resource
 - 1208
 - 1209 **extendsFrom**
 - 1210 Value: icom_core:Actor
 - 1211
 - 1212 **stereotype**
 - 1213 Value: primary
 - 1214
 - 1215 **description**
 - 1216 Value: A resource actor is an actor representing a bookable resource, such as a conference
 - 1217 room, equipment, or on-line conference.
 - 1218
 - 1219 **propertyDefinitions**
 - 1220 The values for this attribute are defined in Section 3.3.6.3.

1221 3.3.6.3 Property Definitions

1222 The Resource class inherits property definitions from super classes.

1223 The Resource class MUST have the property definitions:

1224

1225 **icom_core:resourceSpace**

1226 Description: Administrative space of a resource actor.

1227 Required: False

1228 Inherited: False

1229 Property Type: icom_core:Space

1230 Cardinality: Single

1231 Updatability: Read Only

1232

1233 **icom_core:location**

1234 Description: Location of a resource.

1235 Required: False

1236 Inherited: False

1237 Property Type: icom_core:Location

1238 Cardinality: Single

1239 Updatability: Read Write

1240

1241 **icom_core:capacity**

1242 Description: Capacity of a resource.

1243 Required: False

1244 Inherited: False

1245 Property Type: Integer

1246 Cardinality: Single

1247 Updatability: Read Write

1248

1249 **icom_core:resourceType**

1250 Description: Type of a resource.

1251 Required: False

1252 Inherited: False

1253 Property Type: icom_core:ResourceType

1254 Cardinality: Single

1255 Updatability: Read Write

1256

1257 **icom_core:bookingRule**

1258 Description: Resource booking rule.

1259 Required: False

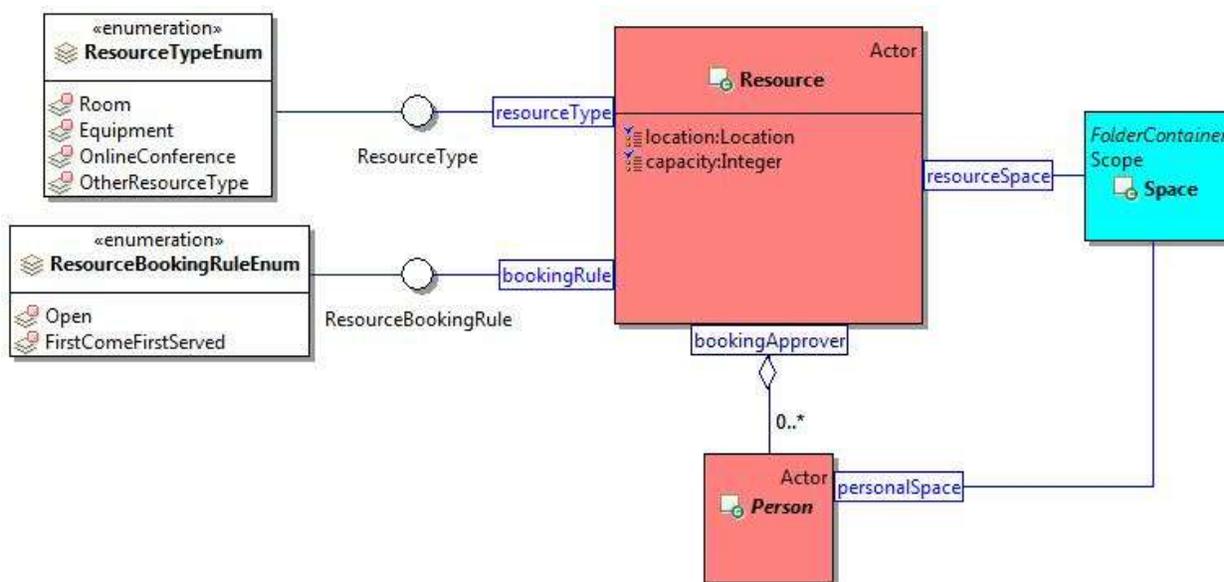
1260 Inherited: False

1261 Property Type: icom_core:ResourceBookingRule

1262 Cardinality: Single

1263 Updatability: Read Write
 1264
 1265 **icom_core:bookingApprover**
 1266 Description: One or more persons who approve the booking of a resource.
 1267 Required: False
 1268 Inherited: False
 1269 Property Type: icom_core:Person
 1270 Cardinality: Multi
 1271 Updatability: Read Write
 1272

1273 The Resource class MAY include additional property definitions which are implementation-defined.
 1274



1275
 1276 *Figure 12: Resource Class Diagram.*
 1277

1278 3.3.7 ResourceType

1279 3.3.7.1 Description

1280 A resource type is a category of resources.

1281 3.3.7.2 Class Definition

1282 The ResourceType class is a mixin class which defines a resource type.

1283 The ResourceType class has attribute values:

1284
 1285 **localNamespace**
 1286 Value: icom_core
 1287

1288 **localName**
1289 Value: ResourceType
1290
1291 **extendsFrom**
1292 Value:
1293
1294 **stereotype**
1295 Value: mixin
1296
1297 **description**
1298 Value: ResourceType is a mixin class which defines a type of resources.
1299
1300 **propertyDefinitions**
1301 The values for this attribute are defined in Section 3.3.7.3.

1302 **3.3.7.3 Property Definitions**

1303 The ResourceType class MAY include additional property definitions which are implementation-defined.
1304

1305 **3.3.8 ResourceTypeEnum**

1306 The ResourceTypeEnum class is an enum class that enumerates the instances each of which expresses
1307 a type of resources.

1308 The ResourceTypeEnum class has attribute values:

1309
1310 **localNamespace**
1311 Value: icom_core
1312
1313 **localName**
1314 Value: ResourceTypeEnum
1315
1316 **extendsFrom**
1317 Value: icom_core:ResourceType
1318
1319 **stereotype**
1320 Value: primary
1321
1322 **isEnumeration**
1323 Value: TRUE
1324
1325 **description**
1326 Value: A type of resources.
1327

1328 **instances**
1329 Value: <icom_core:Room, icom_core:Equipment, icom_core:OnlineConference,
1330 icom_core:OtherResourceType>

1331
1332 ICOM defines four resource types:

- 1333 • **icom_core:Room** a resource represents a room.
- 1334 • **icom_core:Equipment** a resource represents an equipment.
- 1335 • **icom_core:OnlineConference** a resource represents an online conference.
- 1336 • **icom_core:OtherResourceType** a resource represents other things.

1337

1338 **3.3.9 ResourceBookingRule**

1339 **3.3.9.1 Description**

1340 A resource booking rule is a strategy for allocating resources for calendar scheduling.

1341 **3.3.9.2 Class Definition**

1342 The ResourceBookingRule class is a mixin class which defines a resource booking rule.

1343 The ResourceBookingRule class has attribute values:

1344

1345 **localNamespace**
1346 Value: icom_core

1347

1348 **localName**
1349 Value: ResourceBookingRule

1350

1351 **extendsFrom**
1352 Value:

1353

1354 **stereotype**
1355 Value: mixin

1356

1357 **description**
1358 Value: ResourceBookingRule is a mixin class which defines a rule for allocating resources for
1359 calendar scheduling.

1360

1361 **propertyDefinitions**
1362 The values for this attribute are defined in Section 3.3.9.3.

1363 **3.3.9.3 Property Definitions**

1364 The ResourceBookingRule class MAY include additional property definitions which are implementation-
1365 defined.

1366

1367 **3.3.10 ResourceBookingRuleEnum**

1368 The ResourceBookingRuleEnum class is an enum class that enumerates the instances each of which
1369 expresses a booking rule.

1370 The ResourceBookingRuleEnum class has attribute values:

1371

1372 **localNamespace**

1373 Value: icom_core

1374

1375 **localName**

1376 Value: ResourceBookingRuleEnum

1377

1378 **extendsFrom**

1379 Value: icom_core:ResourceBookingRule

1380

1381 **stereotype**

1382 Value: primary

1383

1384 **isEnumeration**

1385 Value: TRUE

1386

1387 **description**

1388 Value: A resource booking rule for allocating resources for calendar scheduling.

1389

1390 **instances**

1391 Value: <icom_core:Open, icom_core:FirstComeFirstServed>

1392

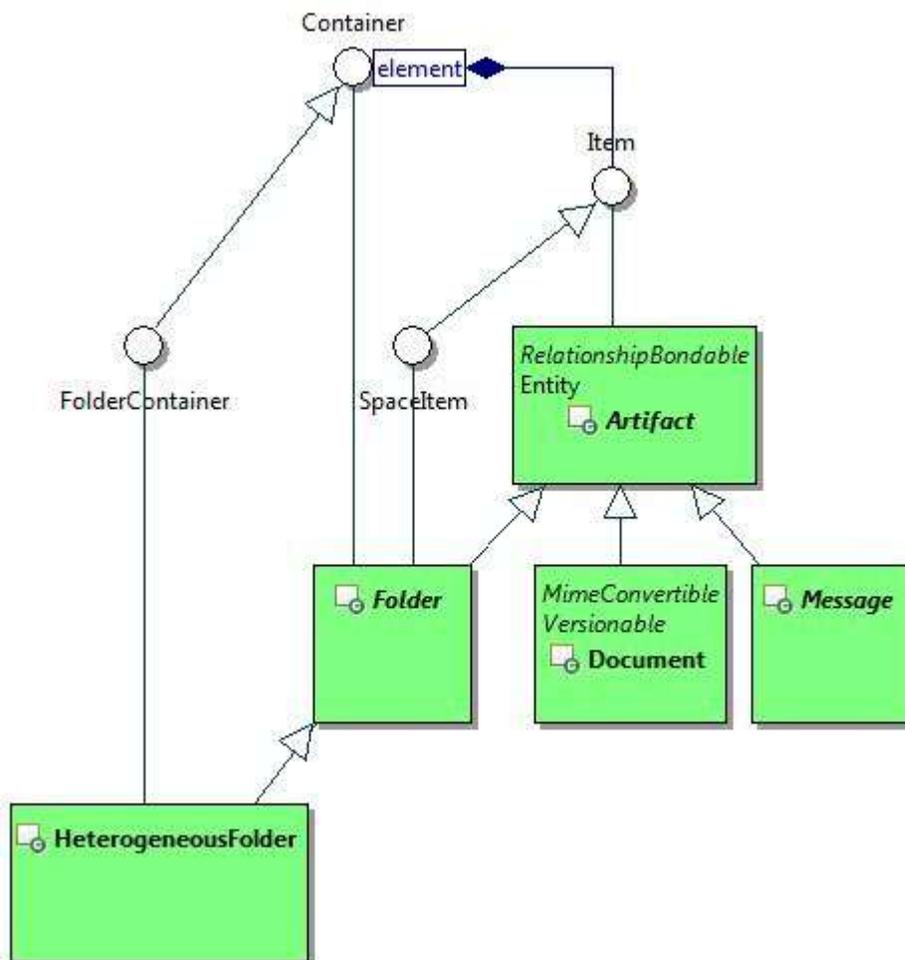
1393 ICOM defines two resource booking rules:

- 1394 • **icom_core:Open** a resource is open for booking.
- 1395 • **icom_core:FirstComeFirstServed** a resource is first come first served.

1396

1397 **3.4 Artifact Branch**

1398 **3.4.1 Artifact and Top-Level Subclasses**



1399
1400 *Figure 13: Artifact Branch.*

1401 Figure 13 depicts the top-level classes of Artifact Branch, which includes Artifact, Folder,
1402 HeterogeneousFolder, Document, and Message.

1403 **3.4.2 Item**

1404 **3.4.2.1 Description**

1405 An item is an element of a container.
1406 The parent of an item **MUST** be a container.

1407 **3.4.2.2 Class Definition**

1408 The Item class is a mixin class which defines the characteristics of entities that can be elements of a
1409 Container.

1410 The Item class has attribute values:

1411

1412 **localNamespace**
 1413 Value: icom_core
 1414
 1415 **localName**
 1416 Value: Item
 1417
 1418 **extendsFrom**
 1419 Value: icom_core:Identifiable
 1420
 1421 **stereotype**
 1422 Value: mixin
 1423
 1424 **description**
 1425 Value: Item is a mixin class which defines the characteristics of entities that can be placed in a
 1426 Container.
 1427
 1428 **propertyDefinitions**
 1429 The values for this attribute are defined in Section 3.4.2.3.

1430 **3.4.2.3 Property Definitions**

1431 The Item class inherits property definitions from super classes.
 1432 The Item class MUST have the property definition:

1433
 1434 **icom_core:parent**
 1435 Description: A parent container of an item.
 1436 Required: False
 1437 Inherited: True
 1438 Property Type: icom_core:Container
 1439 Cardinality: Single
 1440 Updatability: Read Only

1441
 1442 The Item class MAY have the optional property definition:

1443
 1444 **icom_core:container**
 1445 Description: Zero, one, or more containers of an item, including the parent
 1446 container.
 1447 Required: False
 1448 Inherited: False
 1449 Property Type: icom_core:Container
 1450 Cardinality: Multi
 1451 Updatability: Read Write

1452
 1453 The Item class MAY include additional property definitions which are implementation-defined.

1454

1455 **3.4.3 Spaceltem**

1456 **3.4.3.1 Description**

1457 A space item is an item that can be an element of a space.

1458 **3.4.3.2 Class Definition**

1459 The Spaceltem class is a mixin class which defines the characteristics of items that can be elements of a
1460 Space.

1461 The Spaceltem class has attribute values:

1462

1463 **localNamespace**

1464 Value: icom_core

1465

1466 **localName**

1467 Value: Spaceltem

1468

1469 **extendsFrom**

1470 Value: icom_core:Item

1471

1472 **stereotype**

1473 Value: mixin

1474

1475 **description**

1476 Value: Spaceltem is a mixin class which defines the characteristics of entities that can be
1477 elements of a Space.

1478

1479 **propertyDefinitions**

1480 The values for this attribute are defined in Section 3.4.3.3.

1481 **3.4.3.3 Property Definitions**

1482 The Spaceltem class inherits property definitions from super classes.

1483 The Spaceltem class MAY include additional property definitions which are implementation-defined.

1484

1485 **3.4.4 Container**

1486 **3.4.4.1 Description**

1487 A container is an extent that contains items.

1488 **3.4.4.2 Class Definition**

1489 The Container class is a mixin class which defines the characteristics of extents that contain items.

1490 The Container class has attribute values:

1491

1492 **localNamespace**

1493 Value: icom_core

1494

1495 **localName**

1496 Value: Container

1497

1498 **extendsFrom**

1499 Value: icom_core:Extent

1500

1501 **stereotype**

1502 Value: mixin

1503

1504 **description**

1505 Value: A container is an extent that contains items.

1506

1507 **propertyDefinitions**

1508 The values for this attribute are defined in Section 3.4.4.3.

1509 **3.4.4.3 Property Definitions**

1510 The Container class inherits property definitions from super classes.

1511 The Container class **MUST** have the property definition:

1512

1513 **icom_core:element**

1514 Description:	Elements of a container, i.e. items whose parent container is the container or whose containers include the container.
1515	
1516 Required:	False
1517 Inherited:	False
1518 Property Type:	icom_core:Item
1519 Cardinality:	Multi
1520 Updatability:	Read Only

1521

1522 The Container class **MAY** include additional property definitions which are implementation-defined.

1523

1524 **3.4.5 FolderContainer**

1525 **3.4.5.1 Description**

1526 A folder container is a container which may contain folders. Space and heterogeneous folder are folder containers.

1527

1528 **3.4.5.2 Class Definition**

1529 The FolderContainer class is a mixin class that defines the characteristics of containers that may contain
1530 folders.

1531 The FolderContainer class has attribute values:

1532

1533 **localNamespace**

1534 Value: icom_core

1535

1536 **localName**

1537 Value: FolderContainer

1538

1539 **extendsFrom**

1540 Value: icom_core:Container

1541

1542 **stereotype**

1543 Value: mixin

1544

1545 **description**

1546 Value: A folder container is a container which may contain folders.

1547

1548 **propertyDefinitions**

1549 The values for this attribute are defined in Section 3.4.5.3.

1550 **3.4.5.3 Property Definitions**

1551 The FolderContainer class inherits property definitions from super classes.

1552 The FolderContainer class MAY include additional property definitions which are implementation-defined.

1553

1554 **3.4.6 Artifact**

1555 **3.4.6.1 Description**

1556 An artifact is a result of a communication, cooperation, content creation, or collaboration activity.

1557 Note: Document versioning is an example of content creation activity resulting in an artifact (a version of a
1558 document).

1559 **3.4.6.2 Class Definition**

1560 The Artifact class has attribute values:

1561

1562 **localNamespace**

1563 Value: icom_core

1564

1565 **localName**

1566 Value: Artifact

1567

1568 **extendsFrom**
1569 Value: icom_core:Entity, icom_core:Item, icom_meta:RelationshipBondable
1570 Optional Value: icom_core:SpaceItem
1571
1572 **stereotype**
1573 Value: primary
1574
1575 **isAbstract**
1576 Value: TRUE
1577
1578 **description**
1579 Value: An artifact is a result of a communication, cooperation, content creation, or collaboration
1580 activity.
1581
1582 **propertyDefinitions**
1583 The values for this attribute are defined in Section 3.4.6.3.

1584 **3.4.6.3 Property Definitions**

1585 The Artifact class inherits property definitions from super classes.
1586 The Artifact class **MUST** have the property definitions:

1588 **icom_core:description**

1589	Description:	A description of an artifact.
1590	Required:	False
1591	Inherited:	False
1592	Property Type:	String
1593	Cardinality:	Single
1594	Updatability:	Read Write

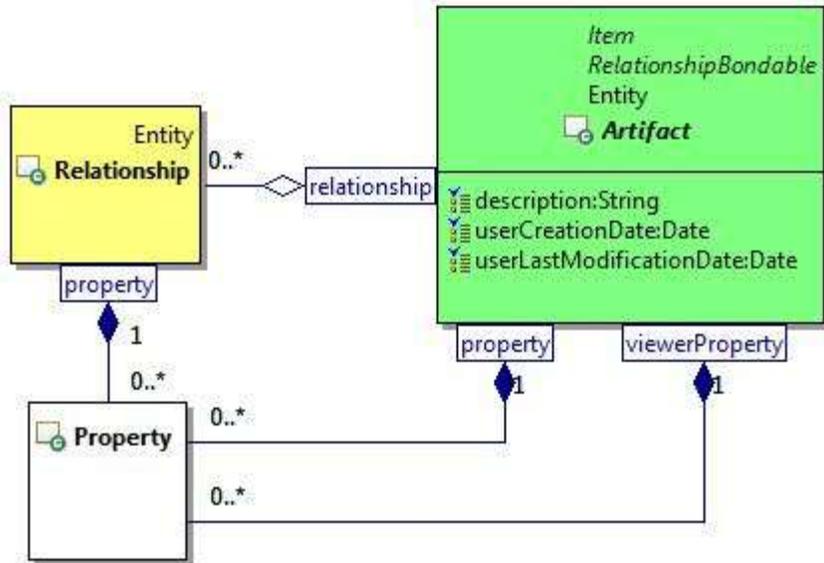
1596 **icom_core:userCreationDate**

1597	Description:	Date and time when an artifact was created.
1598	Required:	False
1599	Inherited:	False
1600	Property Type:	DateTime
1601	Cardinality:	Single
1602	Updatability:	Read Write

1604 **icom_core:userLastModificationDate**

1605	Description:	Date and time when an artifact was last modified.
1606	Required:	False
1607	Inherited:	False
1608	Property Type:	DateTime
1609	Cardinality:	Single

1610	Updatability:	Read Write
1611		
1612	icom_meta:property	
1613	Description:	Zero or more extended properties of an artifact.
1614	Required:	False
1615	Inherited:	False
1616	Property Type:	icom_meta:Property
1617	Cardinality:	Multi
1618	Updatability:	Read Write
1619		
1620	icom_meta:viewerProperty	
1621	Description:	Zero or more extended properties of an artifact visible to a viewer.
1622		
1623	Required:	False
1624	Inherited:	False
1625	Property Type:	icom_meta:Property
1626	Cardinality:	Multi
1627	Updatability:	Read Write
1628		
1629	icom_meta:relationship	
1630	Description:	Zero or more relationships associated with an artifact.
1631	Required:	False
1632	Inherited:	False
1633	Property Type:	icom_meta:Relationship
1634	Cardinality:	Multi
1635	Updatability:	Read Only
1636		
1637	The Artifact class MAY include additional property definitions which are implementation-defined.	
1638		



1639
1640 Figure 14: Artifact Class Diagram.

1641
1642 **3.4.7 Folder**

1643 **3.4.7.1 Description**

1644 A folder is an artifact that may contain other artifacts.
1645 Note: Every folder except root folders has at least one parent folder. The parent of a root folder is a
1646 space. Subclasses of Folder class should enforce their own semantics on elements.

1647 **3.4.7.2 Class Definition**

1648 The Folder class has attribute values:

- 1649
- 1650 **localNamespace**
- 1651 Value: icom_core
- 1652
- 1653 **localName**
- 1654 Value: Folder
- 1655
- 1656 **extendsFrom**
- 1657 Value: icom_core:Artifact, icom_core:Container, icom_core:SpaceItem
- 1658
- 1659 **stereotype**
- 1660 Value: primary
- 1661
- 1662 **isAbstract**
- 1663 Value: TRUE
- 1664

1665 **description**
1666 Value: A folder is an artifact that may contain other artifacts.

1667
1668 **propertyDefinitions**
1669 The values for this attribute are defined in Section 3.4.7.3.

1670 **3.4.7.3 Property Definitions**

1671 The Folder class inherits property definitions from super classes.
1672 The Folder class MUST have the property definition:

1673
1674 **icom_core:parent**
1675 Description: A parent container of a folder.
1676 Required: False
1677 Inherited: True
1678 Property Type: icom_core:FolderContainer
1679 Cardinality: Single
1680 Updatability: Read Only

1681
1682 The Folder class MAY include additional property definitions which are implementation-defined.
1683

1684 **3.4.8 HeterogeneousFolder**

1685 **3.4.8.1 Description**

1686 A heterogeneous folder is an unconstrained folder to contain any type of artifacts.
1687 Note: It is typically used for document folders, inbox, outbox, and trash folder of a space.

1688 **3.4.8.2 Class Definition**

1689 The HeterogeneousFolder class has attribute values:

1690
1691 **localNamespace**
1692 Value: icom_core
1693
1694 **localName**
1695 Value: HeterogeneousFolder
1696
1697 **extendsFrom**
1698 Value: icom_core:Folder, icom_core:FolderContainer
1699
1700 **stereotype**
1701 Value: primary
1702

1703 **description**
1704 Value: A heterogeneous folder is an unconstrained folder to contain any type of artifacts.

1705
1706 **propertyDefinitions**
1707 The values for this attribute are defined in Section 3.4.8.3.

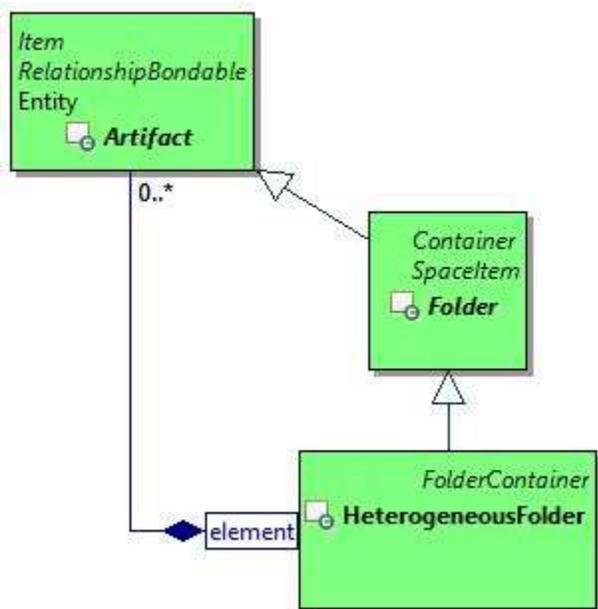
3.4.8.3 Property Definitions

1709 The HeterogeneousFolder class inherits property definitions from super classes.
1710 The HeterogeneousFolder class **MUST** have the property definition:

1711
1712 **icom_core:element**
1713 Description: Elements of a heterogeneous folder.
1714 Required: False
1715 Inherited: True
1716 Property Type: icom_core:Artifact
1717 Cardinality: Multi
1718 Updatability: Read Only

1719
1720 The HeterogeneousFolder class **MAY** include additional property definitions which are implementation-
1721 defined.

1722



1723
1724 *Figure 15: Heterogeneous Folder Class Diagram.*

1725

1726 **3.5 Access Control Model**

1727 **3.5.1 Accessor**

1728 **3.5.1.1 Description**

1729 An accessor can be granted or denied access rights to objects.

1730 **3.5.1.2 Class Definition**

1731 The Accessor class is a mixin class which defines the characteristics of subjects such as groups and
1732 actors that can be granted or denied access types in access control lists and privileges in role
1733 assignments.

1734 The Accessor class has attribute values:

1735

1736 **localNamespace**

1737 Value: icom_ac

1738

1739 **localName**

1740 Value: Accessor

1741

1742 **extendsFrom**

1743 Value: icom_core:Identifiable

1744

1745 **stereotype**

1746 Value: mixin

1747

1748 **description**

1749 Value: Accessor is a mixin class which defines the characteristics of subjects such as groups
1750 and actors that can be granted or denied access types in access control lists and granted
1751 privileges in role assignments.

1752

1753 **propertyDefinitions**

1754 The values for this attribute are defined in Section 3.5.1.3.

1755 **3.5.1.3 Property Definitions**

1756 The Accessor class inherits property definitions from super classes.

1757 The Accessor class MAY include additional property definitions which are implementation-defined.

1758

1759 **3.5.2 Owner**

1760 **3.5.2.1 Description**

1761 An owner is a subject that can be the owner of entities.

1762 An owner of an entity MAY always have rights to update the access control list for the entity.

1763 **3.5.2.2 Class Definition**

1764 The Owner class is a mixin class which defines the characteristics of subjects such as groups and actors
1765 that can own entities.

1766 The Owner class has attribute values:

1767

1768 **localNamespace**

1769 Value: icom_ac

1770

1771 **localName**

1772 Value: Owner

1773

1774 **extendsFrom**

1775 Value: icom_ac:Accessor

1776

1777 **stereotype**

1778 Value: mixin

1779

1780 **description**

1781 Value: Owner is a mixin class which defines the characteristics of subjects such as groups and
1782 actors that can own entities.

1783

1784 **propertyDefinitions**

1785 The values for this attribute are defined in Section 3.5.2.3.

1786 **3.5.2.3 Property Definitions**

1787 The Owner class inherits property definitions from super classes.

1788 The Owner class MAY include additional property definitions which are implementation-defined.

1789

1790 **3.5.3 RoleDefinition**

1791 **3.5.3.1 Description**

1792 A role definition is a named set of privileges.

1793 **3.5.3.2 Class Definition**

1794 The RoleDefinition class has attribute values:

1795

1796 **localNamespace**

1797 Value: icom_ac

1798

1799 **localName**

1800 Value: RoleDefinition

1801

1802 **extendsFrom**
1803 Value: icom_core:Entity, icom_meta:RelationshipBondable

1804
1805 **stereotype**
1806 Value: primary

1807
1808 **description**
1809 Value: A role definition is a named set of privileges.

1810
1811 **propertyDefinitions**
1812 The values for this attribute are defined in Section 3.5.3.3.

1813 **3.5.3.3 Property Definitions**

1814 The RoleDefinition class inherits property definitions from super classes.
1815 The RoleDefinition class MUST have the property definition:

1816
1817 **icom_core:description**
1818 Description: A description of a role definition.
1819 Required: False
1820 Inherited: False
1821 Property Type: String
1822 Cardinality: Single
1823 Updatability: Read Write

1824
1825 **icom_ac:privilege**
1826 Description: A set of privileges.
1827 Required: True
1828 Inherited: False
1829 Property Type: icom_ac:Privilege
1830 Cardinality: Multi
1831 Updatability: Read Write

1832
1833 The RoleDefinition class MAY include additional property definitions which are implementation-defined.
1834

1835 **3.5.4 Role**

1836 **3.5.4.1 Description**

1837 A role assigns a named set of privileges to a set of accessors for operations within an assigned scope.

1838 **3.5.4.2 Class Definition**

1839 The Role class has attribute values:
1840

1841 **localNamespace**
 1842 Value: icom_ac
 1843
 1844 **localName**
 1845 Value: Role
 1846
 1847 **extendsFrom**
 1848 Value: icom_core:Subject
 1849
 1850 **stereotype**
 1851 Value: primary
 1852
 1853 **description**
 1854 Value: A role assigns a named set of rights to a set of accessors for operations within an
 1855 assigned scope.
 1856
 1857 **propertyDefinitions**
 1858 The values for this attribute are defined in Section 3.5.4.3.

1859 **3.5.4.3 Property Definitions**

1860 The Role class inherits property definitions from super classes.

1861 The Role class **MUST** have the property definitions:

1862

1863 **icom_ac:roleDefinition**

1864	Description:	A role definition containing a set of privileges.
1865	Required:	True
1866	Inherited:	False
1867	Property Type:	icom_ac:RoleDefinition
1868	Cardinality:	Single
1869	Updatability:	On Create

1870

1871 **icom_ac:assignedScope**

1872	Description:	A scope in which a role is assigned.
1873	Required:	True
1874	Inherited:	False
1875	Property Type:	icom_core:Scope
1876	Cardinality:	Single
1877	Updatability:	Read Write

1878

1879 **icom_ac:memberAccessor**

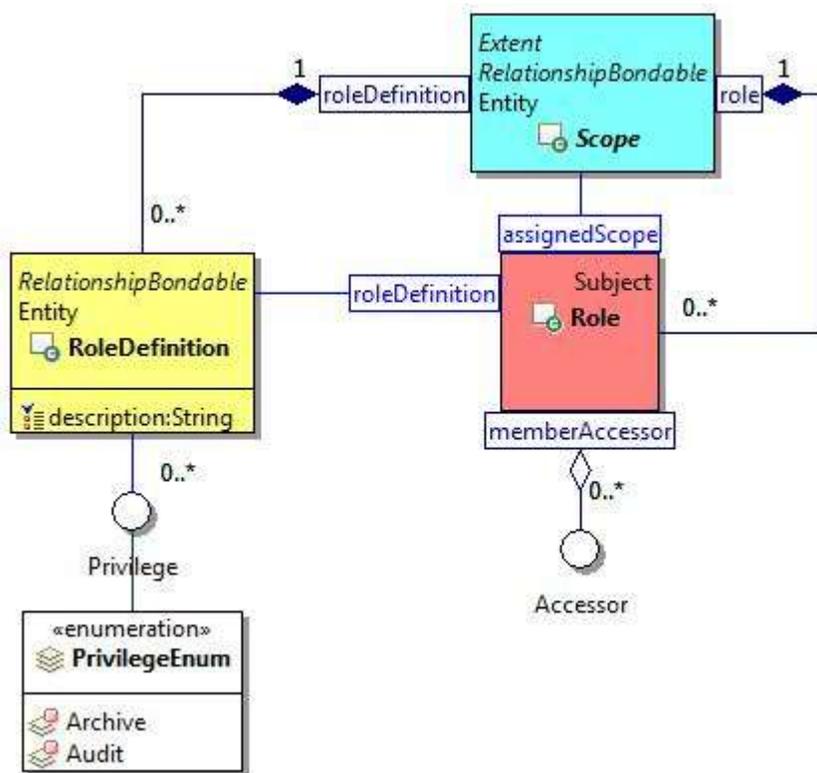
1880	Description:	Accessors (actors and groups) assigned to a role.
1881	Required:	False
1882	Inherited:	False

1883 Property Type: icom_ac:Accessor
 1884 Cardinality: Multi
 1885 Updatability: Read Write

1886

1887 The Role class MAY include additional property definitions which are implementation-defined.

1888



1889

1890 Figure 16: Role Definition and Role Class Diagram.

1891

1892 3.5.5 Privilege

1893 3.5.5.1 Description

1894 A privilege is an access right granted through roles.

1895 3.5.5.2 Class Definition

1896 The Privilege class is a mixin class which defines access rights that can be included in role definitions.

1897 The Privilege class has attribute values:

1898

1899 **localNamespace**

1900 Value: icom_ac

1901

1902 **localName**

1903 Value: Privilege

1904

1905 **extendsFrom**
1906 Value:
1907
1908 **stereotype**
1909 Value: mixin
1910
1911 **description**
1912 Value: Privilege is a mixin class which defines access rights that can be included in role
1913 definitions.
1914
1915 **propertyDefinitions**
1916 The values for this attribute are defined in Section 3.5.5.3.

1917 **3.5.5.3 Property Definitions**

1918 The Privilege class MAY include additional property definitions which are implementation-defined.
1919

1920 **3.5.6 PrivilegeEnum**

1921 The PrivilegeEnum class is an enum class that enumerates the instances each of which expresses a
1922 privilege that can be assigned to a role.

1923 The PrivilegeEnum class has attribute values:

1924
1925 **localNamespace**
1926 Value: icom_ac
1927
1928 **localName**
1929 Value: PrivilegeEnum
1930
1931 **extendsFrom**
1932 Value: icom_ac:Privilege
1933
1934 **stereotype**
1935 Value: primary
1936
1937 **isEnumeration**
1938 Value: TRUE
1939
1940 **description**
1941 Value: Privilege that can be assigned to a role.
1942
1943 **instances**
1944 Value: <icom_ac:Archive, icom_ac:Audit>
1945

- 1946 ICOM defines two privileges:
- 1947 • **icom_ac:Archive** a right to archive contents in a scope.
 - 1948 • **icom_ac:Audit** a right to audit activities in a scope.

1949

1950 **3.5.7 AccessControlList**

1951 **3.5.7.1 Description**

1952 An access control list (ACL) is an object attached to an entity to specify a list of permissions to access the
 1953 entity.

1954 **3.5.7.2 Class Definition**

1955 The AccessControlList class has attribute values:

1956

1957 **localNamespace**

1958 Value: icom_ac

1959

1960 **localName**

1961 Value: AccessControlList

1962

1963 **extendsFrom**

1964 Value:

1965

1966 **stereotype**

1967 Value: primary

1968

1969 **description**

1970 Value: An access control list (ACL) is an object attached to an entity to specify a list of
 1971 permissions to access the entity.

1972

1973 **propertyDefinitions**

1974 The values for this attribute are defined in Section 3.5.7.3.

1975 **3.5.7.3 Property Definitions**

1976 The AccessControlList class **MUST** have the property definitions:

1977

1978 **icom_ac:object**

1979 Description: Associated object.

1980 Required: True

1981 Inherited: False

1982 Property Type: icom_core:Entity

1983 Cardinality: Single

1984 Updatability: On Create

1985

1986	icom_ac:accessControlEntry	
1987	Description:	One or more access control entries.
1988	Required:	True
1989	Inherited:	False
1990	Property Type:	icom_ac:AccessControlEntry
1991	Cardinality:	Multi
1992	Updatability:	Read Write

1993

1994 AccessControlList class MAY include additional property definitions which are implementation-defined.

1995

1996 **3.5.8 AccessControlEntry**

1997 **3.5.8.1 Description**

1998 An access control entry specifies access types granted to or denied for an accessor.

1999 **3.5.8.2 Class Definition**

2000 The AccessControlEntry class has attribute values:

2001

2002 **localNamespace**

2003 Value: icom_ac

2004

2005 **localName**

2006 Value: AccessControlEntry

2007

2008 **extendsFrom**

2009 Value:

2010

2011 **stereotype**

2012 Value: primary

2013

2014 **description**

2015 Value: An access control entry is associated with an accessor and contains a list of access
2016 types (permissions) granted to or denied from the accessor.

2017

2018 **propertyDefinitions**

2019 The values for this attribute are defined in Section 3.5.8.3.

2020 **3.5.8.3 Property Definitions**

2021 The AccessControlEntry class MUST have the property definitions:

2022

2023 **icom_ac:subject**

2024 Description: Associated subject.

2025 Required: True

2026 Inherited: False
 2027 Property Type: icom_ac:Accessor
 2028 Cardinality: Single
 2029 Updatability: On Create
 2030
 2031 **icom_ac:grant**
 2032 Description: One or more access types granted to a subject.
 2033 Required: False
 2034 Inherited: False
 2035 Property Type: icom_ac:AccessType
 2036 Cardinality: Multi
 2037 Updatability: Read Write

2038
 2039 **icom_ac:deny**
 2040 Description: One or more access type denied for a subject.
 2041 Required: False
 2042 Inherited: False
 2043 Property Type: icom_ac:AccessType
 2044 Cardinality: Multi
 2045 Updatability: Read Write

2046
 2047 The AccessControlEntry class MAY include additional property definitions which are implementation-
 2048 defined.
 2049

2050 **3.5.9 AccessType**

2051 An AccessType is an access right granted through an access control entry.

2052 **3.5.9.1 Class Definition**

2053 The AccessType class is a mixin class which defines access rights that can be granted or denied in an
 2054 access control entry.

2055 The AccessType class has attribute values:

2056
 2057 **localNamespace**
 2058 Value: icom_ac
 2059
 2060 **localName**
 2061 Value: AccessType
 2062
 2063 **extendsFrom**
 2064 Value:
 2065

2066 **stereotype**
2067 Value: mixin
2068
2069 **description**
2070 Value: AccessType is a mixin class which defines access rights that can be granted or denied in
2071 an access control entry.
2072
2073 **propertyDefinitions**
2074 The values for this attribute are defined in Section 3.5.9.2.

2075 **3.5.9.2 Property Definitions**

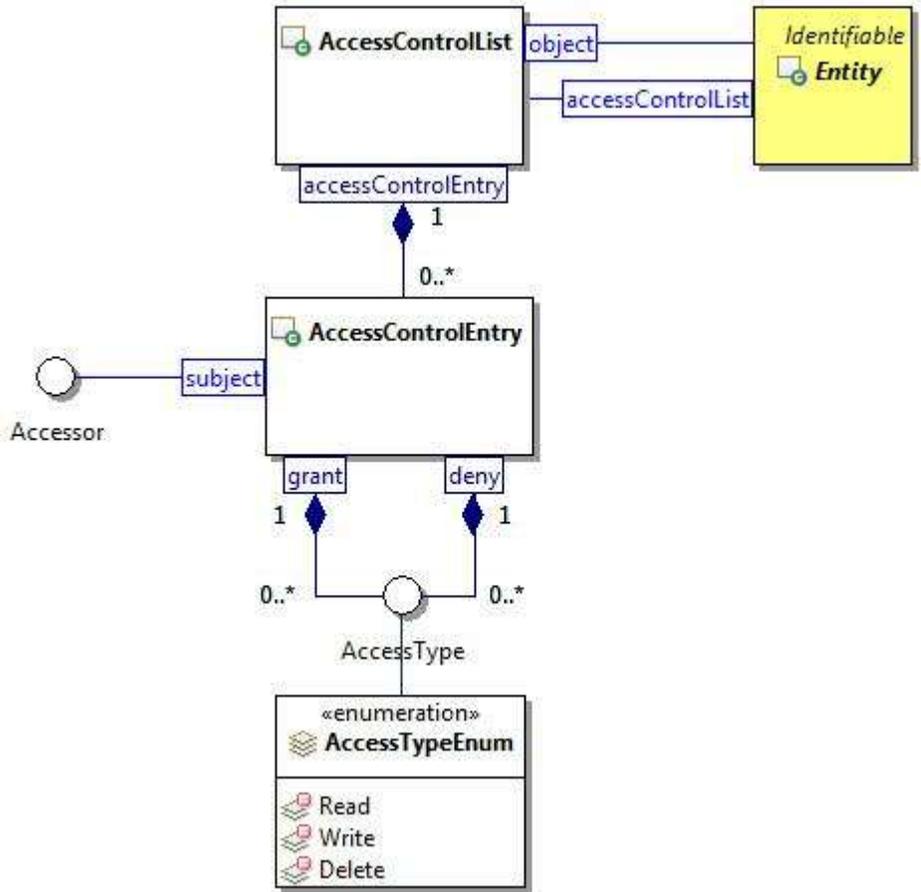
2076 The AccessType class inherits property definitions from super classes.
2077 The AccessType class MAY include additional property definitions which are implementation-defined.
2078

2079 **3.5.10 AccessTypeEnum**

2080 The AccessTypeEnum class is an enum class that enumerates the instances each of which expresses an
2081 access type that can be granted or denied in an access control entry.
2082 The AccessTypeEnum class has attribute values:

2083
2084 **localNamespace**
2085 Value: icom_ac
2086
2087 **localName**
2088 Value: AccessTypeEnum
2089
2090 **extendsFrom**
2091 Value: icom_ac:AccessType
2092
2093 **stereotype**
2094 Value: primary
2095
2096 **isEnumeration**
2097 Value: TRUE
2098
2099 **description**
2100 Value: Access type that can be granted or denied in an access control entry.
2101
2102 **instances**
2103 Value: <icom_ac:Read, icom_ac:Write, icom_ac>Delete>
2104
2105 ICOM defines three access types:
2106 • **icom_ac:Read** a right to retrieve an entity.

- 2107 • **icom_ac:Write** a right to update an entity.
 - 2108 • **icom_ac:Delete** a right to delete an entity.
- 2109



2110
2111 *Figure 17: Access Control List Class Diagram.*

2112

2113 3.6 Metadata Model

2114 3.6.1 ClassDefinition

2115 3.6.1.1 Description

2116 A class definition is an entity that defines a type of entities.

2117 3.6.1.2 Class Definition

2118 The ClassDefinition class has attribute values:

- 2119
- 2120 **localNamespace**
Value: icom_meta
 - 2121
 - 2122
 - 2123 **localName**
Value: ClassDefinition
 - 2124

2125
 2126 **extendsFrom**
 2127 Value: icom_core:Entity, icom_meta:RelationshipBondable
 2128
 2129 **stereotype**
 2130 Value: primary
 2131
 2132 **isAbstract**
 2133 Value: FALSE
 2134
 2135 **description**
 2136 Value: A class definition defines a type of entities.
 2137
 2138 **propertyDefinitions**
 2139 The values for this attribute are defined in Section 3.6.1.3.

2140 **3.6.1.3 Property Definitions**

2141 The ClassDefinition class inherits property definitions from super classes.
 2142 The ClassDefinition class **MUST** have the property definition:

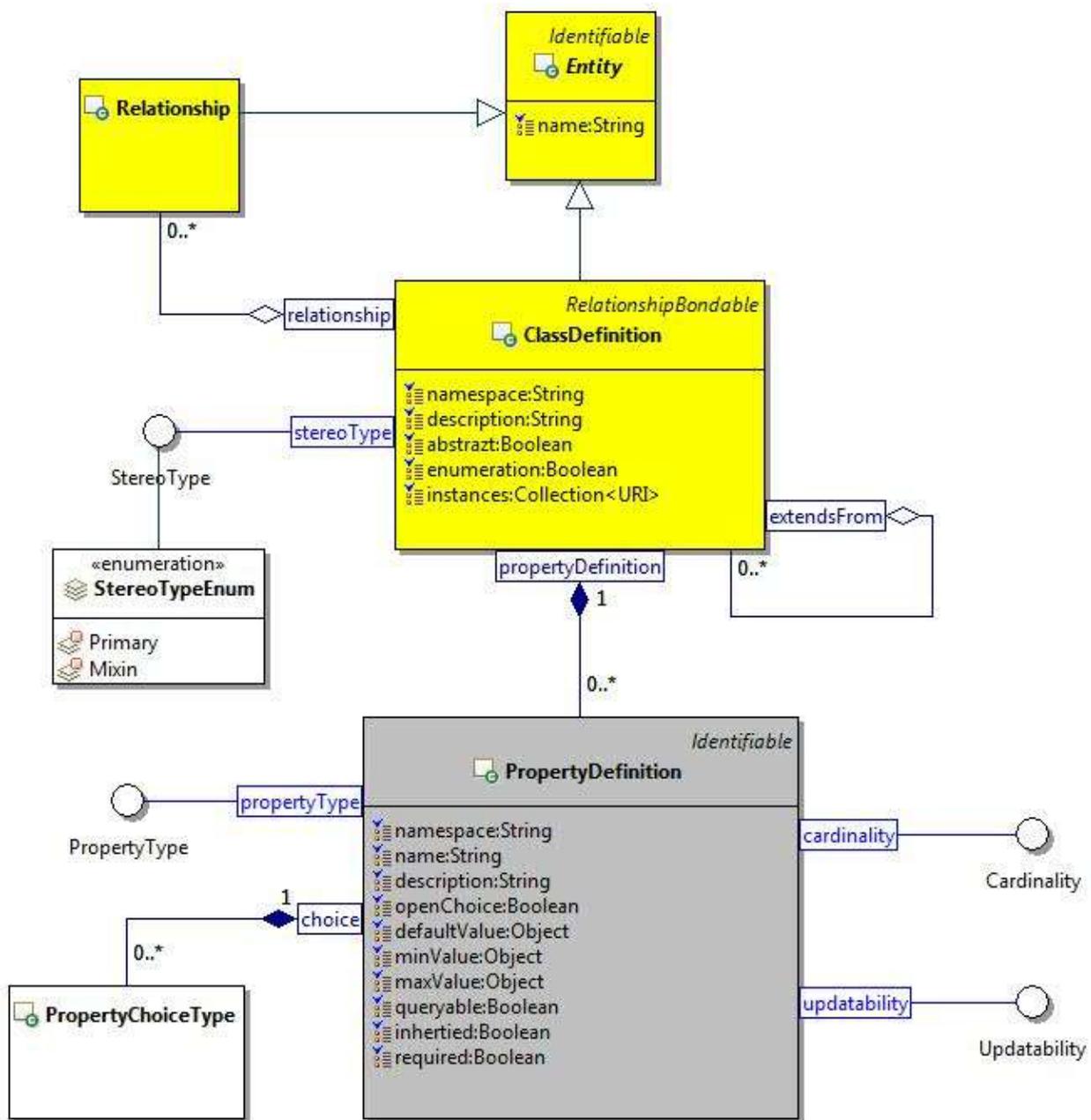
2143
 2144 **icom_core:namespace**
 2145 Description: Namespace for a class name.
 2146 Required: False
 2147 Inherited: False
 2148 Property Type: String
 2149 Cardinality: Single
 2150 Updatability: Read Write

2151
 2152 **icom_core:description**
 2153 Description: A description of a class.
 2154 Required: False
 2155 Inherited: False
 2156 Property Type: String
 2157 Cardinality: Single
 2158 Updatability: Read Write

2159
 2160 **icom_meta:extendsFrom**
 2161 Description: One or more generalizations of a class.
 2162 Required: True
 2163 Inherited: False
 2164 Property Type: icom_meta:ClassDefinition
 2165 Cardinality: Multi
 2166 Updatability: Read Write

2167		
2168	icom_meta:stereoType	
2169	Description:	Stereo type of a class.
2170	Required:	True
2171	Inherited:	False
2172	Property Type:	icom_meta:StereoType
2173	Cardinality:	Single
2174	Updatability:	Read Write
2175		
2176	icom_meta:abstract	
2177	Description:	Indicates whether a class is abstract or concrete.
2178	Required:	False
2179	Inherited:	False
2180	Property Type:	Boolean
2181	Cardinality:	Single
2182	Updatability:	Read Write
2183		
2184	icom_meta:enumeration	
2185	Description:	Indicates whether instances of a class are enumerated. This
2186		property is applicable only if the stereo type property is
2187		primary.
2188	Required:	False
2189	Inherited:	False
2190	Property Type:	Boolean
2191	Cardinality:	Single
2192	Updatability:	Read Write
2193		
2194	icom_meta:instances	
2195	Description:	Instances of an enumeration class. This property is applicable
2196		only if the enumeration property is true.
2197	Required:	False
2198	Inherited:	False
2199	Property Type:	IRI
2200	Cardinality:	Multi
2201	Updatability:	Read Write
2202		
2203	icom_meta:propertyDefinition	
2204	Description:	One or more property definitions of a class definition.
2205	Required:	False
2206	Inherited:	False
2207	Property Type:	icom_meta:PropertyDefinition
2208	Cardinality:	Multi
2209	Updatability:	Read Write

2210
 2211 The ClassDefinition class MAY include additional property definitions which are implementation-defined.
 2212



2213
 2214 Figure 18: Class Definition and Property Definition Class Diagram.
 2215

2216 3.6.2 Stereotype

2217 3.6.2.1 Description

2218 A stereo type of a class definition.

2219 **3.6.2.2 Class Definition**

2220 The Stereotype class is a mixin class which defines a stereo type of a class definition.

2221 The Stereotype class has attribute values:

2222

2223 **localNamespace**

2224 Value: icom_meta

2225

2226 **localName**

2227 Value: Stereotype

2228

2229 **extendsFrom**

2230 Value:

2231

2232 **stereotype**

2233 Value: mixin

2234

2235 **description**

2236 Value: Stereotype is a mixin class which defines a stereo type of a class definition.

2237

2238 **propertyDefinitions**

2239 The values for this attribute are defined in Section 3.6.2.3.

2240 **3.6.2.3 Property Definitions**

2241 The Stereotype class MAY include additional property definitions which are implementation-defined.

2242

2243 **3.6.3 StereotypeEnum**

2244 The StereotypeEnum class is an enum class that enumerates the instances each of which defines a
2245 stereo type of a class definition.

2246 The StereotypeEnum has attribute values:

2247

2248 **localNamespace**

2249 Value: icom_meta

2250

2251 **localName**

2252 Value: StereotypeEnum

2253

2254 **extendsFrom**

2255 Value: icom_meta:Stereotype

2256

2257 **stereotype**

2258 Value: primary

2259

2260 **isEnumeration**
2261 Value: TRUE
2262
2263 **description**
2264 Value: Stereo type of a class definition.
2265
2266 **instances**
2267 Value: <icom_meta:Primary, icom_meta:Mixin>
2268

2269 ICOM defines two stereo types:
2270

- **icom_meta:Primary** a primary class.
- **icom_meta:Mixin** a mixin class.

2271
2272

2273 **3.6.4 PropertyDefinition**

2274 **3.6.4.1 Description**

2275 A property definition specifies the name, type, choice, and cardinality of values for properties.

2276 **3.6.4.2 Class Definition**

2277 The PropertyDefinition class has attribute values:

2278
2279 **localNamespace**
2280 Value: icom_meta
2281
2282 **localName**
2283 Value: PropertyDefinition
2284
2285 **extendsFrom**
2286 Value: icom_core:Identifiable
2287
2288 **stereotype**
2289 Value: primary
2290
2291 **description**
2292 Value: A property definition specifies the name, type, choice, and cardinality of values for
2293 properties.
2294
2295 **propertyDefinitions**
2296 The values for this attribute are defined in Section 3.6.4.3.

2297 **3.6.4.3 Property Definitions**

2298 The PropertyDefinition class inherits property definitions from super classes.

2299 The PropertyDefinition class MUST have the property definitions:

2300

2301 **icom_core:namespace**

2302 Description: Namespace for a property name.

2303 Required: False

2304 Inherited: False

2305 Property Type: String

2306 Cardinality: Single

2307 Updatability: Read Write

2308

2309 **icom_core:name**

2310 Description: Name for a property.

2311 Required: True

2312 Inherited: False

2313 Property Type: String

2314 Cardinality: Single

2315 Updatability: Read Write

2316

2317 **icom_core:description**

2318 Description: A description of a property definition.

2319 Required: False

2320 Inherited: False

2321 Property Type: String

2322 Cardinality: Single

2323 Updatability: Read Write

2324

2325 **icom_meta:propertyType**

2326 Description: Type of a property.

2327 Required: False

2328 Inherited: False

2329 Property Type: icom_meta:PropertyType

2330 Cardinality: Single

2331 Updatability: On Create

2332 Choices: {PropertyChoiceType}

2333 Open Choice: False

2334

2335 Note: The notation {PropertyChoiceType} represents a set of PropertyChoiceType.

2336

2337 **icom_meta:defaultValue**

2338 Description: A default value for a property.

2339 Required: False

2340 Inherited: False

2341	Property Type:	property-type
2342	Cardinality:	Single
2343	Updatability:	Read Write
2344		
2345	icom_meta:choice	
2346	Description:	An allowed value for a property.
2347	Required:	False
2348	Inherited:	False
2349	Property Type:	icom_meta:PropertyChoiceType
2350	Cardinality:	Multi
2351	Updatability:	Read Write
2352		
2353	icom_meta:openChoice	
2354	Description:	Indicates whether value of the property must be listed among
2355		the choices.
2356	Required:	False
2357	Inherited:	False
2358	Property Type:	Boolean
2359	Cardinality:	Single
2360	Updatability:	Read Write
2361		
2362	icom_meta:inherited	
2363	Description:	Indicates whether a property definition is inherited from a
2364		super class.
2365	Required:	False
2366	Inherited:	False
2367	Property Type:	Boolean
2368	Cardinality:	Single
2369	Updatability:	Read Write
2370		
2371	icom_meta:required	
2372	Description:	Indicates whether a property value must be provided. It is
2373		applicable only when the updatability of the property is read-
2374		write or on-create.
2375	Required:	True
2376	Inherited:	False
2377	Property Type:	Boolean
2378	Cardinality:	Single
2379	Updatability:	Read Write
2380		
2381	icom_meta:updatability	
2382	Description:	Updatability of a property specifying under what
2383		circumstances the property value can be updated.
2384	Required:	True

2385	Inherited:	False
2386	Property Type:	icom_meta:Updatability
2387	Cardinality:	Single
2388	Updatability:	On Create
2389		
2390	icom_meta:cardinality	
2391	Description:	Cardinality of a property specifying whether the property can have “zero or one” or “zero or more” values.
2392		
2393	Required:	True
2394	Inherited:	False
2395	Property Type:	icom_meta:Cardinality
2396	Cardinality:	Single
2397	Updatability:	On Create
2398		
2399	icom_meta:minValue	
2400	Description:	Minimum value for an integer or decimal property.
2401	Required:	False
2402	Inherited:	False
2403	Property Type:	Integer Decimal
2404	Cardinality:	Single
2405	Updatability:	Read Write
2406		
2407	icom_meta:maxValue	
2408	Description:	Maximum value for an integer or decimal property.
2409	Required:	False
2410	Inherited:	False
2411	Property Type:	Integer Decimal
2412	Cardinality:	Single
2413	Updatability:	Read Write

2415 The PropertyDefinition class MAY include additional property definitions which are implementation-
2416 defined.

2418 3.6.5 Property

2419 3.6.5.1 Description

2420 The property holds a property value.

2421 3.6.5.2 Class Definition

2422 The Property class has attribute values:

2423		
2424	localNamespace	
2425	Value:	icom_meta

2426
 2427 **localName**
 2428 Value: Property
 2429
 2430 **extendsFrom**
 2431 Value:
 2432
 2433 **stereotype**
 2434 Value: primary
 2435
 2436 **description**
 2437 Value: A property value.
 2438
 2439 **propertyDefinitions**
 2440 The values for this attribute are defined in Section 3.6.5.3.

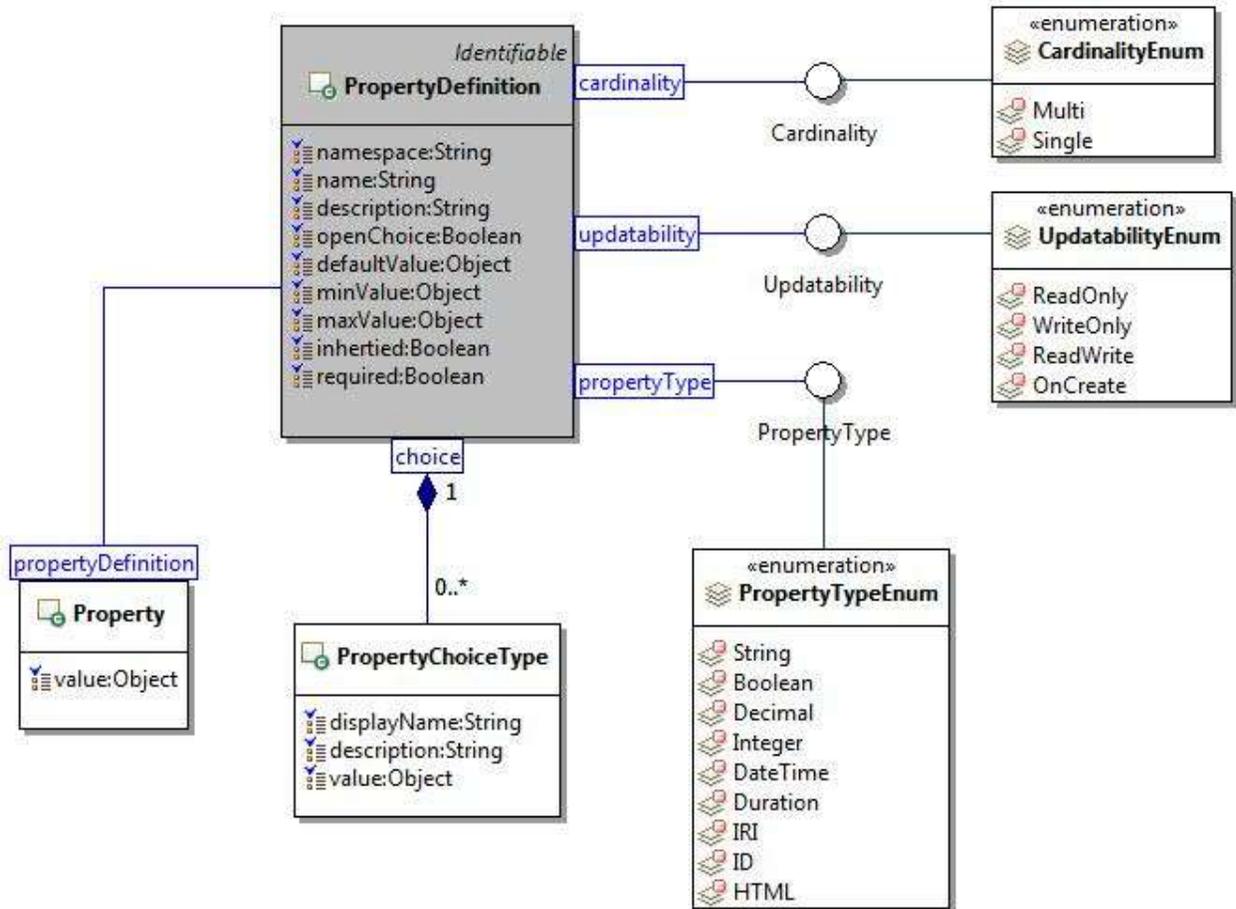
2441 **3.6.5.3 Property Definitions**

2442 The Property class **MUST** have the property definitions:

2443
 2444 **icom_meta:propertyDefinition**
 2445 Description: A property definition that specifies the name, type, and
 2446 cardinality of a property.
 2447 Required: True
 2448 Inherited: False
 2449 Property Type: icom_meta:PropertyDefinition
 2450 Cardinality: Single
 2451 Updatability: On Create

2452
 2453 **icom_meta:value**
 2454 Description: A value of a property.
 2455 Required: True
 2456 Inherited: False
 2457 Property Type: **property-type**
 2458 Cardinality: Single
 2459 Updatability: Read Write

2460
 2461 The Property class **MAY** include additional property definitions which are implementation-defined.
 2462



2463
 2464 *Figure 19: Property Definition and Property Class Diagram.*
 2465

2466 3.6.6 PropertyChoiceType

2467 3.6.6.1 Description

2468 The property choice type represents a value choice for a property. Each choice includes a display name
 2469 to be used for presentation purpose and a value to be stored in a property when a choice is selected.

2470 3.6.6.2 Class Definition

2471 The PropertyChoiceType class has attribute values:

- 2472
- 2473 **localNamespace**
- 2474 Value: icom_meta
- 2475
- 2476 **localName**
- 2477 Value: PropertyChoiceType
- 2478
- 2479 **extendsFrom**
- 2480 Value:
- 2481

2482 **stereotype**
2483 Value: primary
2484
2485 **description**
2486 Value: A choice for a property value.
2487
2488 **propertyDefinitions**
2489 The values for this attribute are defined Section 3.6.6.3.

2490 **3.6.6.3 Property Definitions**

2491 The PropertyChoiceType class MUST have the property definitions:

2492
2493 **icom_core:description**
2494 Description: A description of a property choice.
2495 Required: False
2496 Inherited: False
2497 Property Type: String
2498 Cardinality: Single
2499 Updatability: Read Write

2500
2501 **icom_meta:displayName**
2502 Description: Display name of a property choice.
2503 Required: True
2504 Inherited: False
2505 Property Type: String
2506 Cardinality: Single
2507 Updatability: Read Write

2508
2509 **icom_meta:value**
2510 Description: A value of a property choice.
2511 Required: True
2512 Inherited: False
2513 Property Type: **property-type**
2514 Cardinality: Single
2515 Updatability: Read Write

2516
2517 The PropertyChoiceType class MAY include additional property definitions which are implementation-
2518 defined.

2519

2520 **3.6.7 PropertyType**

2521 A PropertyType expresses a name of a **property-type**.

2522 **3.6.7.1 Class Definition**

2523 The PropertyType class is a mixin class which expresses a name of a **property-type**.

2524 The PropertyType class has attribute values:

2525

2526 **localNamespace**

2527 Value: icom_meta

2528

2529 **localName**

2530 Value: PropertyType

2531

2532 **extendsFrom**

2533 Value:

2534

2535 **stereotype**

2536 Value: mixin

2537

2538 **description**

2539 Value: PropertyType is a mixin class which expresses a name of a **property-type**.

2540

2541 **propertyDefinitions**

2542 The values for this attribute are defined in Section 3.6.7.2.

2543 **3.6.7.2 Property Definitions**

2544 The PropertyType class MAY include additional property definitions which are implementation-defined.

2545

2546 **3.6.8 PropertyTypeEnum**

2547 The PropertyTypeEnum class is an enum class that enumerates the instances each of which expresses
2548 the name of a **property-type**.

2549 The PropertyTypeEnum class has attribute values:

2550

2551 **localNamespace**

2552 Value: icom_meta

2553

2554 **localName**

2555 Value: PropertyTypeEnum

2556

2557 **extendsFrom**

2558 Value: icom_meta:PropertyType

2559

2560 **stereotype**

2561 Value: primary

2562

2563 **isEnumeration**
2564 Value: TRUE
2565
2566 **description**
2567 Value: Name of a basic data type.
2568
2569 **instances**
2570 Value: <icom_meta:String, icom_meta:Boolean, icom_meta:Decimal, icom_meta:Integer,
2571 icom_meta:Datetime, icom_meta:Duration, icom_meta:IRI, icom_meta:ID, icom_meta:HTML>
2572

2573 ICOM defines nine data types:

- 2574 • **icom_meta:String** is equivalent to XML schema type **xsd:string**.
- 2575 • **icom_meta:Boolean** is equivalent to XML schema type **xsd:boolean**.
- 2576 • **icom_meta:Decimal** is equivalent to XML schema type **xsd:decimal**.
- 2577 • **icom_meta:Integer** is equivalent to XML schema type **xsd:integer**.
- 2578 • **icom_meta:Datetime** is equivalent to XML schema type **xsd:dateTime**.
- 2579 • **icom_meta:Duration** is equivalent to XML schema type **xsd:duration**.
- 2580 • **icom_meta:IRI** is equivalent to XML schema type **xsd:anyURI**.
- 2581 • **icom_meta:ID** opaque object identifiers.
- 2582 • **icom_meta:HTML** documents or fragments of Hypertext Markup Language (HTML) content

2583
2584 Note: ICOM uses basic data types defined by “XML Schema Part 2: Datatypes Second Edition” (W3C
2585 Recommendation, 28 October 2004, <http://www.w3.org/TR/xmlschema-2/>).
2586

2587 **3.6.9 Updatability**

2588 **3.6.9.1 Description**

2589 Updatability specifies under what circumstances a property value can be updated.

2590 **3.6.9.2 Class Definition**

2591 The Updatability class is a mixin class which specifies under what circumstances a property value can be
2592 updated.

2593 The Updatability class has attribute values:

2594
2595 **localNamespace**
2596 Value: icom_meta
2597
2598 **localName**
2599 Value: Updatability
2600
2601 **extendsFrom**
2602 Value:
2603

2604 **stereotype**
2605 Value: mixin
2606
2607 **description**
2608 Value: Updatability is a mixin class which specifies under what circumstances a property value
2609 can be updated.
2610
2611 **propertyDefinitions**
2612 The values for this attribute are defined in Section 3.6.9.3.

2613 **3.6.9.3 Property Definitions**

2614 The Updatability class MAY include additional property definitions which are implementation-defined.
2615

2616 **3.6.10 UpdatabilityEnum**

2617 The UpdatabilityEnum class is an enum class that enumerates instances each of which expresses the
2618 updatability of a property.

2619 The UpdatabilityEnum has attribute values:

2620
2621 **localNamespace**
2622 Value: icom_meta
2623
2624 **localName**
2625 Value: UpdatabilityEnum
2626
2627 **extendsFrom**
2628 Value: icom_meta:Updatability
2629
2630 **stereotype**
2631 Value: primary
2632
2633 **isEnumeration**
2634 Value: TRUE
2635
2636 **description**
2637 Value: Updatability of a property.
2638
2639 **instances**
2640 Value: <icom_meta:ReadOnly, icom_meta:WriteOnly, icom_meta:ReadWrite,
2641 icom_meta:OnCreate>

2642
2643 ICOM defines four updatability types:

- 2644 • **icom_meta:ReadOnly** a property must not be updated directly by application.
- 2645 • **icom_meta:WriteOnly** a property can be updated but cannot be read by application.

- 2646 • **icom_meta:ReadWrite** a property can be updated by application.
 - 2647 • **icom_meta:WriteOnly** a property can be updated during the creation by application.
- 2648

2649 **3.6.11 Cardinality**

2650 **3.6.11.1 Description**

2651 Cardinality specifies whether a property is single or multi valued.

2652 **3.6.11.2 Class Definition**

2653 The Cardinality class is a mixin class which defines whether a property is single or multi valued.

2654 The Cardinality class has attribute values:

2655

2656 **localNamespace**
2657 Value: icom_meta

2658

2659 **localName**
2660 Value: Cardinality

2661

2662 **extendsFrom**
2663 Value:

2664

2665 **stereotype**
2666 Value: mixin

2667

2668 **description**
2669 Value: Cardinality is a mixin class which defines whether a property is single or multi valued.

2670

2671 **propertyDefinitions**
2672 The values for this attribute are defined in Section 3.6.11.3.

2673 **3.6.11.3 Property Definitions**

2674 The Cardinality class MAY include additional property definitions which are implementation-defined.

2675

2676 **3.6.12 CardinalityEnum**

2677 The CardinalityEnum class is an enum class that enumerates instances each of which expresses the
2678 cardinality of a property.

2679 The CardinalityEnum has attribute values:

2680

2681 **localNamespace**
2682 Value: icom_meta

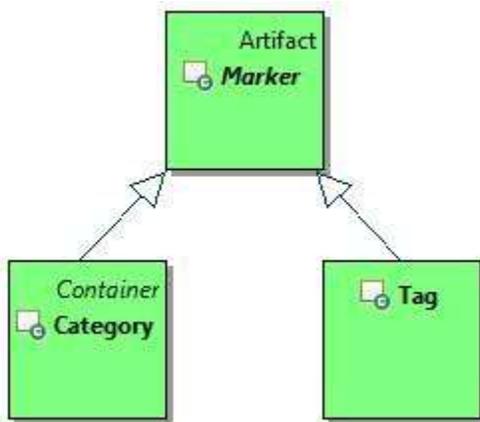
2683

2684 **localName**
 2685 Value: CardinalityEnum
 2686
 2687 **extendsFrom**
 2688 Value: icom_meta:Cardinality
 2689
 2690 **stereotype**
 2691 Value: primary
 2692
 2693 **isEnumeration**
 2694 Value: TRUE
 2695
 2696 **description**
 2697 Value: Cardinality of a property.
 2698
 2699 **instances**
 2700 Value: <icom_meta:Single, icom_meta:Multi>

2702 ICOM defines two cardinality types:

- 2703 • **icom_meta:Single** a property can have zero or one value (if property is not required), or exactly
- 2704 one value (if property is required).
- 2705 • **icom_meta:Multi** a property can have zero or more values (if property is not required), or one or
- 2706 more values (if property is required).

2708 3.6.13 Marker and Subclasses



2709
 2710 *Figure 20: Marker Branch.*

2711 Figure 20 depicts the main classes of Marker Branch, which includes Marker, Category, and Tag.

2712 **3.6.14 Marker**

2713 **3.6.14.1 Description**

2714 A marker is an artifact that groups together entities by a criterion. Markers can be flat or hierarchical. Flat
2715 markers are modeled by tag and hierarchical markers are modeled by category.

2716 Note: In some cases when a user applies a marker to an entity, the marker application should be private
2717 such that only the user who applies the marker can browse or locate the entity through the marker. This is
2718 especially the case when markers are created by a user and visible only to the user who created them.

2719 **3.6.14.2 Class Definition**

2720 The Marker class has attribute values:

2721

2722 **localNamespace**

2723 Value: icom_meta

2724

2725 **localName**

2726 Value: Marker

2727

2728 **extendsFrom**

2729 Value: icom_core:Artifact

2730

2731 **stereotype**

2732 Value: primary

2733

2734 **isAbstract**

2735 Value: TRUE

2736

2737 **description**

2738 Value: A marker is an artifact that groups together entities by a criterion.

2739

2740 **propertyDefinitions**

2741 The values for this attribute are defined in Section 3.6.14.3.

2742 **3.6.14.3 Property Definitions**

2743 The Marker class inherits property definitions from super classes.

2744 The Marker class MUST have the property definition:

2745

2746 **icom_meta:markedEntity**

2747 Description: A marked entity.

2748 Required: False

2749 Inherited: False

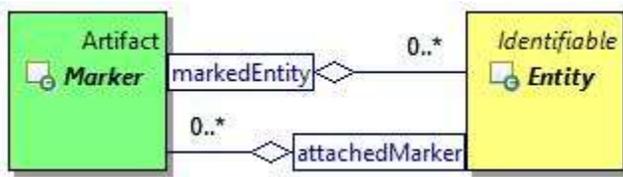
2750 Property Type: icom_core:Entity

2751 Cardinality: Multi

2752 Updatability: Read Only

2753
2754
2755

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



2756
2757
2758

Figure 21: Marker Class Diagram.

2759 3.6.15 Category

2760 3.6.15.1 Description

2761 A category is a marker that classifies entities.

2762 3.6.15.2 Class Definition

2763 The Category class has attribute values:

2764

localNamespace

2765 Value: icom_meta

2767

localName

2768 Value: Category

2770

extendsFrom

2771 Value: icom_meta:Marker, icom_core:Container

2773

stereotype

2774 Value: primary

2776

description

2777 Value: A category is a marker that classifies entities.

2779

propertyDefinitions

2780 The values for this attribute are defined in Section 3.6.15.3.

2782 3.6.15.3 Property Definitions

2783 The Category class inherits property definitions from super classes.

2784 The Category class MUST have the property definitions:

2785

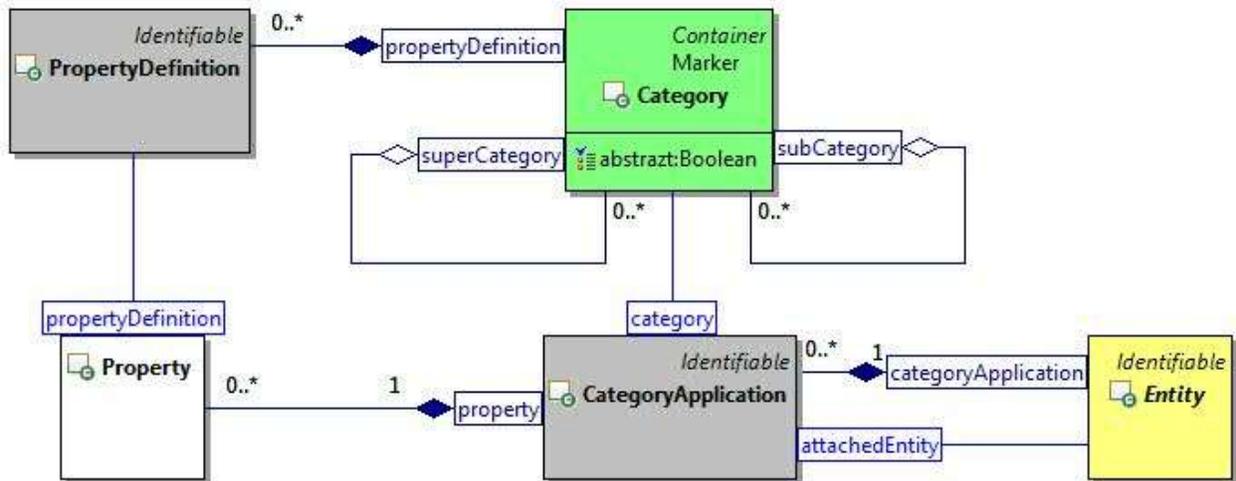
icom_meta:superCategory

2786 Description: Zero or more super categories.

2787 Required: False

2788

2789	Inherited:	False
2790	Property Type:	icom_meta:Category
2791	Cardinality:	Multi
2792	Updatability:	Read Only
2793		
2794	icom_meta:subCategory	
2795	Description:	Zero or more sub categories.
2796	Required:	False
2797	Inherited:	False
2798	Property Type:	icom_meta:Category
2799	Cardinality:	Multi
2800	Updatability:	Read Only
2801		
2802	icom_meta:abstract	
2803	Description:	Indicates whether a category is abstract or concrete.
2804	Required:	False
2805	Inherited:	False
2806	Property Type:	Boolean
2807	Cardinality:	Single
2808	Updatability:	Read Write
2809		
2810	icom_meta:propertyDefinition	
2811	Description:	Optional or mandatory properties for a category application.
2812	Required:	False
2813	Inherited:	False
2814	Property Type:	icom_meta:PropertyDefinition
2815	Cardinality:	Multi
2816	Updatability:	Read Write
2817		
2818	The Category class MAY include additional property definitions which are implementation-defined.	
2819		



2820

2821 *Figure 22: Category and Category Application Class Diagram.*

2822

2823 3.6.16 CategoryApplication

2824 3.6.16.1 Description

2825 A category application is an instance of association between a category and a specific entity.

2826 3.6.16.2 Class Definition

2827 The CategoryApplication class has attribute values:

2828

localNamespace

2830 Value: icom_meta

2831

localName

2833 Value: CategoryApplication

2834

extendsFrom

2836 Value: icom_core:Identifiable

2837

stereotype

2839 Value: primary

2840

description

2842 Value: A category application is an instance of association between a category and a specific
2843 entity.

2844

propertyDefinitions

2846 The values for this attribute are defined in Section 3.6.16.3.

2847 **3.6.16.3 Property Definitions**

2848 The CategoryApplication class inherits property definitions from super classes.

2849 The CategoryApplication class MUST have the property definitions:

2850

2851 **icom_meta:attachedEntity**

2852 Description: An entity to which a category is applied.

2853 Required: True

2854 Inherited: False

2855 Property Type: icom_core:Entity

2856 Cardinality: Single

2857 Updatability: On Create

2858

2859 **icom_meta:category**

2860 Description: A category which is applied on an entity.

2861 Required: True

2862 Inherited: False

2863 Property Type: icom_meta:Category

2864 Cardinality: Single

2865 Updatability: On Create

2866

2867 **icom_meta:property**

2868 Description: Zero or more properties.

2869 Required: False

2870 Inherited: False

2871 Property Type: icom_meta:Property

2872 Cardinality: Multi

2873 Updatability: Read Write

2874

2875 The CategoryApplication class MAY include additional property definitions which are implementation-
2876 defined.

2877

2878 **3.6.17 Tag**

2879 **3.6.17.1 Description**

2880 A tag is a marker that labels entities by a keyword.

2881 **3.6.17.2 Class Definition**

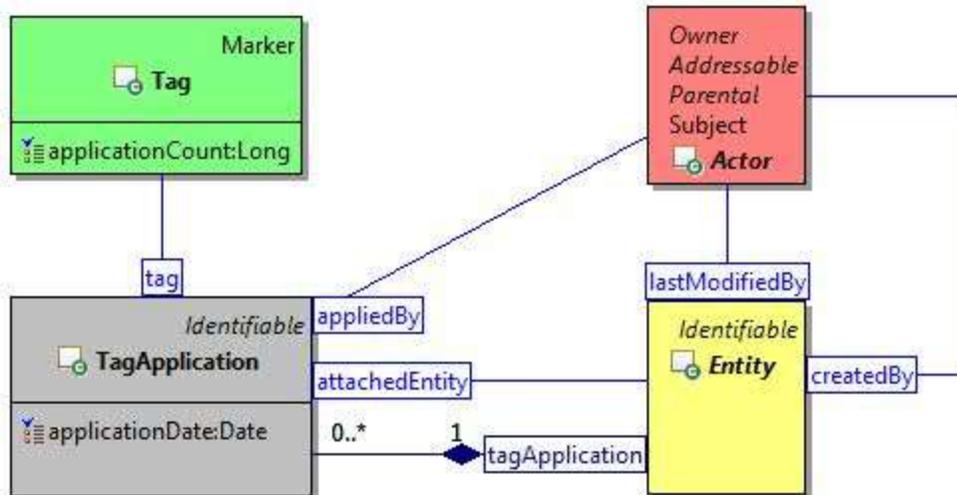
2882 The Tag class has attribute values:

2883

2884 **localNamespace**

2885 Value: icom_meta

2886



2916
2917 *Figure 23: Tag and Tag Application Class Diagram.*

2918

2919 **3.6.18 TagApplication**

2920 **3.6.18.1 Description**

2921 A tag application is an instance of association between a tag and a specific entity.

2922 **3.6.18.2 Class Definition**

2923 The TagApplication class has attribute values:

2924

2925 **localNamespace**
2926 Value: icom_meta

2927

2928 **localName**
2929 Value: TagApplication

2930

2931 **extendsFrom**
2932 Value: icom_core:Identifiable

2933

2934 **stereotype**
2935 Value: primary

2936

2937 **description**
2938 Value: A tag application is an instance of association between a tag and a specific entity.

2939

2940 **propertyDefinitions**
2941 The values for this attribute are defined in Section 3.6.18.3.

2942 **3.6.18.3 Property Definitions**

2943 The TagApplication class inherits property definitions from super classes.

2944 The TagApplication class MUST have the property definitions:

2945

2946 **icom_meta:attachedEntity**

2947 Description: An entity on which a tag is applied.

2948 Required: True

2949 Inherited: False

2950 Property Type: icom_core:Entity

2951 Cardinality: Single

2952 Updatability: On Create

2953

2954 **icom_meta:tag**

2955 Description: A tag which is applied to an entity.

2956 Required: True

2957 Inherited: False

2958 Property Type: icom_meta:Tag

2959 Cardinality: Single

2960 Updatability: On Create

2961

2962 **icom_meta:appliedBy**

2963 Description: A user who applies a tag to an entity.

2964 Required: False

2965 Inherited: False

2966 Property Type: icom_core:Actor

2967 Cardinality: Single

2968 Updatability: Read Only

2969

2970 **icom_meta:applicationDate**

2971 Description: A date and time when a tag is applied to an entity.

2972 Required: False

2973 Inherited: False

2974 Property Type: DateTime

2975 Cardinality: Single

2976 Updatability: Read Write

2977

2978 The TagApplication class MAY include additional property definitions which are implementation-defined.

2979

2980 **3.6.19 RelationshipBondable**

2981 **3.6.19.1 Description**

2982 A relationship bondable entity is an entity which may be related to other entities by a relationship.

2983 Note: A relationship can exist among entities that are not relationships.

2984 **3.6.19.2 Class Definition**

2985 The RelationshipBondable class is a mixin class which defines the characteristics of entities that may be
2986 relationship bonded. It includes almost every subclass of Entity except Relationship.

2987 The RelationshipBondable class has attribute values:

2988

2989 **localNamespace**

2990 Value: icom_meta

2991

2992 **localName**

2993 Value: RelationshipBondable

2994

2995 **extendsFrom**

2996 Value: icom_core:Identifiable

2997

2998 **stereotype**

2999 Value: mixin

3000

3001 **description**

3002 Value: RelationshipBondable is a mixin class which defines the characteristics of entities that
3003 can be relationship bonded.

3004

3005 **propertyDefinitions**

3006 The values for this attribute are defined in Section 3.6.19.3.

3007 **3.6.19.3 Property Definitions**

3008 The RelationshipBondable class inherits property definitions from super classes.

3009 The RelationshipBondable class MAY include additional property definitions which are implementation-
3010 defined.

3011

3012 **3.6.20 RelationshipDefinition**

3013 **3.6.20.1 Description**

3014 A relationship definition is an entity that defines a type of relationship, including a name and a description
3015 of the relationship type, types of source entity and target entities of a relationship, and definition of
3016 properties in a relationship.

3017 **3.6.20.2 Class Definition**

3018 The RelationshipDefinition class has attribute values:

3019

3020 **localNamespace**

3021 Value: icom_meta

3022

3105 The Relationship class MUST have the property definitions:

3106

3107 **icom_meta:relationshipDefinition**

3108 Description: A definition of relationships.
3109 Required: True
3110 Inherited: False
3111 Property Type: icom_meta:RelationshipDefinition
3112 Cardinality: Single
3113 Updatability: On Create

3114

3115 **icom_meta:sourceEntity**

3116 Description: A source entity of a relationship.
3117 Required: True
3118 Inherited: False
3119 Property Type: icom_meta:RelationshipBondable
3120 Cardinality: Single
3121 Updatability: On Create

3122

3123 **icom_meta:targetEntity**

3124 Description: One or more target entities of a relationship.
3125 Required: True
3126 Inherited: False
3127 Property Type: icom_meta:RelationshipBondable
3128 Cardinality: Multi
3129 Updatability: Read Write

3130

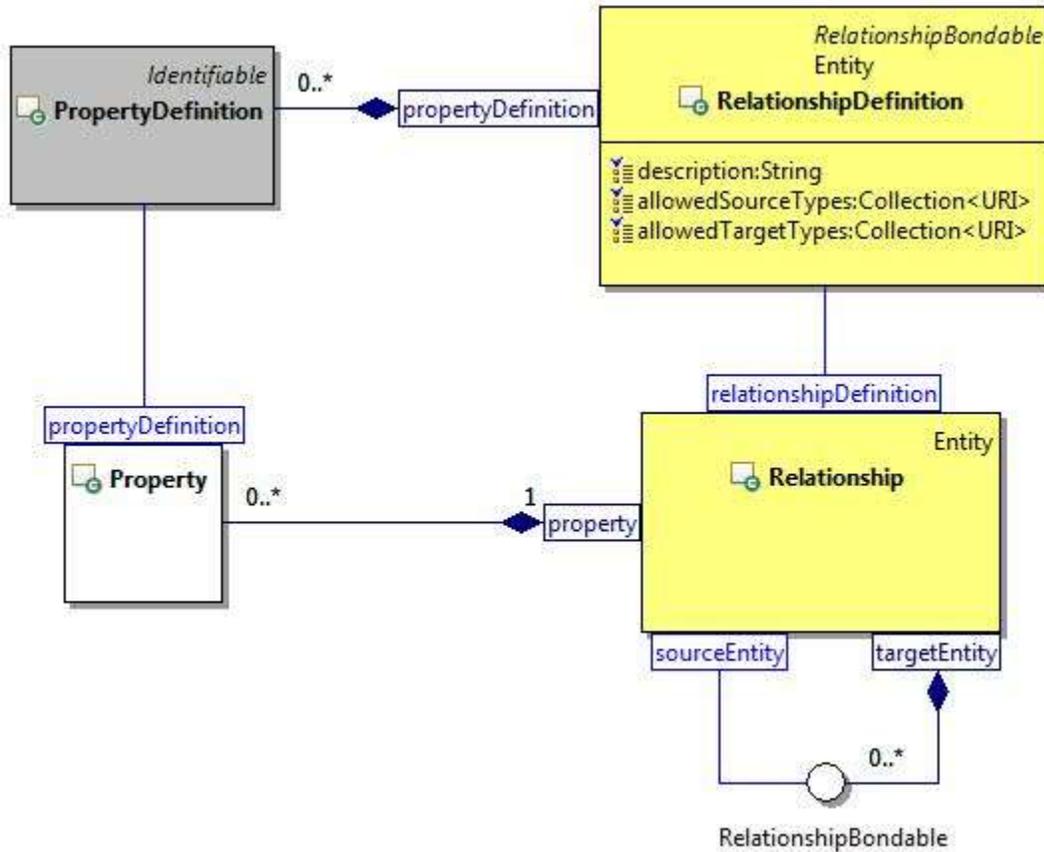
3131 **icom_meta:property**

3132 Description: Zero or more properties.
3133 Required: False
3134 Inherited: False
3135 Property Type: icom_meta:Property
3136 Cardinality: Multi
3137 Updatability: Read Write

3138

3139 The Relationship class MAY include additional property definitions which are implementation-defined.

3140



3141
3142 *Figure 24: Relationship Class Diagram.*

3143
3144 **3.7 Common Concepts**

3145 **3.7.1 Addressable**

3146 **3.7.1.1 Description**

3147 An addressable object is an identifiable object that has one or more addresses.

3148 **3.7.1.2 Class Definition**

3149 The Addressable class is a mixin class which defines the characteristics of entities that has one or more
3150 addresses.

3151 The Addressable class has attribute values:

- 3152
- 3153 **localNamespace**
 - 3154 Value: icom_core
 - 3155
 - 3156 **localName**
 - 3157 Value: Addressable
 - 3158

3159 **extendsFrom**
3160 Value: icom_core:Identifiable
3161
3162 **stereotype**
3163 Value: mixin
3164
3165 **description**
3166 Value: Addressable is a mixin class which defines the characteristics of entities that has one or
3167 more addresses.
3168
3169 **propertyDefinitions**
3170 The values for this attribute are defined in Section 3.7.1.3.

3171 **3.7.1.3 Property Definitions**

3172 The Addressable class inherits property definitions from super classes.
3173 The Addressable class **MUST** have the property definitions:

3175 **icom_core:entityAddress**

3176	Description:	Zero or more addresses of an addressable object.
3177	Required:	False
3178	Inherited:	False
3179	Property Type:	icom_core:EntityAddress
3180	Cardinality:	Multi
3181	Updatability:	Read Write

3183 **icom_core:primaryAddress**

3184	Description:	The primary address of an addressable object.
3185	Required:	False
3186	Inherited:	False
3187	Property Type:	icom_core:EntityAddress
3188	Cardinality:	Single
3189	Updatability:	Read Write

3190
3191 The Addressable class **MAY** include additional property definitions which are implementation-defined.
3192

3193 **3.7.2 EntityAddress**

3194 **3.7.2.1 Description**

3195 An entity address object represents an address which is defined by type and IRI.

3196 **3.7.2.2 Class Definition**

3197 The EntityAddress class has attribute values:
3198

3199 **localNamespace**
3200 Value: icom_core
3201
3202 **localName**
3203 Value: EntityAddress
3204
3205 **extendsFrom**
3206 Value:
3207
3208 **stereotype**
3209 Value: primary
3210
3211 **description**
3212 Value: An entity address object represents an address which is defined by type and IRI.
3213
3214 **propertyDefinitions**
3215 The values for this attribute are defined in Section 3.7.2.3.

3216 **3.7.2.3 Property Definitions**

3217 The EntityAddress class MUST have the property definitions:

3218
3219 **icom_core:addressType**
3220 Description: Type of an address.
3221 Required: False
3222 Inherited: False
3223 Property Type: String
3224 Cardinality: Single
3225 Updatability: Read Write
3226

3227 **icom_core:address**
3228 Description: A IRI representing an address.
3229 Required: False
3230 Inherited: False
3231 Property Type: IRI
3232 Cardinality: Single
3233 Updatability: Read Write
3234

3235 **3.7.3 Participant**

3236 **3.7.3.1 Description**

3237 A participant object represents the participation of any addressable entity in a collaboration activity such
3238 as an occurrence, task, conference, discussion, and message.

3239 If an addressable entity is not specified, an address must be specified.

3240 **3.7.3.2 Class Definition**

3241 The Participant class has attribute values:

3242

3243 **localNamespace**

3244 Value: icom_core

3245

3246 **localName**

3247 Value: Participant

3248

3249 **extendsFrom**

3250 Value:

3251

3252 **stereotype**

3253 Value: primary

3254

3255 **description**

3256 Value: A participant object represents the participation of any addressable entity in a
3257 collaboration activity such as an occurrence, task, conference, discussion, and message.

3258

3259 **propertyDefinitions**

3260 The values for this attribute are defined in Section 3.7.3.3.

3261 **3.7.3.3 Property Definitions**

3262 The Participant class inherits property definitions from super classes.

3263 The Participant class **MUST** have the property definitions:

3264

3265 **icom_core:participant**

3266 Description: An addressable entity to participate in a collaboration activity.

3267 Required: False

3268 Inherited: False

3269 Property Type: icom_core:Addressable

3270 Cardinality: Single

3271 Updatability: On Create

3272

3273 **icom_core:address**

3274 Description: An address of a participant in a collaboration activity.

3275 Required: False

3276 Inherited: False

3277 Property Type: IRI

3278 Cardinality: Single

3279 Updatability: On Create

3280
3281 **icom_core:name**
3282 Description: Name of a participant in a collaboration activity.
3283 Required: False
3284 Inherited: False
3285 Property Type: String
3286 Cardinality: Single
3287 Updatability: On Create

3288
3289 The Participant class MAY include additional property definitions which are implementation-defined.
3290

3291 **3.7.4 Priority**

3292 **3.7.4.1 Description**

3293 A priority level for delivery of information.

3294 **3.7.4.2 Class Definition**

3295 The Priority class is a mixin class which defines a priority level for delivery of information.

3296 The Priority class has attribute values:

3297
3298 **localNamespace**
3299 Value: icom_core
3300
3301 **localName**
3302 Value: Priority
3303
3304 **extendsFrom**
3305 Value:
3306
3307 **stereotype**
3308 Value: mixin
3309
3310 **description**
3311 Value: Priority is a mixin class which defines a priority level for delivery of information.
3312
3313 **propertyDefinitions**
3314 The values for this attribute are defined in Section 3.7.4.3.

3315 **3.7.4.3 Property Definitions**

3316 The Priority class MAY include additional property definitions which are implementation-defined.

3317

3318 **3.7.5 PriorityEnum**

3319 The PriorityEnum class is an enum class that enumerates the instances each of which defines a priority
3320 level for delivery of information.

3321 The PriorityEnum has attribute values:

3322

3323 **localNamespace**

3324 Value: icom_core

3325

3326 **localName**

3327 Value: PriorityEnum

3328

3329 **extendsFrom**

3330 Value: icom_core:Priority

3331

3332 **stereotype**

3333 Value: primary

3334

3335 **isEnumeration**

3336 Value: TRUE

3337

3338 **description**

3339 Value: Priority level for delivery of information.

3340

3341 **instances**

3342 Value: <icom_core:Normal, icom_core:Low, icom_core:Medium, icom_core:High>

3343

3344 ICOM defines four priorities:

- 3345 • **icom_core:Normal** a normal priority.
- 3346 • **icom_core:Low** a low priority.
- 3347 • **icom_core:Medium** a medium priority.
- 3348 • **icom_core:High** a high priority.

3349

3350 **3.7.6 DateTimeResolution**

3351 **3.7.6.1 Description**

3352 A date time resolution is a resolution of date time value.

3353 **3.7.6.2 Class Definition**

3354 The DateTimeResolution class is a mixin class which defines a resolution of date time value.

3355 The DateTimeResolution class has attribute values:

3356

3357 **localNamespace**

3358 Value: icom_core

3359

3360 **localName**

3361 Value: DateTimeResolution

3362

3363 **extendsFrom**

3364 Value:

3365

3366 **stereotype**

3367 Value: mixin

3368

3369 **description**

3370 Value: DateTimeResolution is a mixin class which defines a resolution of date time value.

3371

3372 **propertyDefinitions**

3373 The values for this attribute are defined in Section 3.7.6.3.

3374 **3.7.6.3 Property Definitions**

3375 The DateTimeResolution class MAY include additional property definitions which are implementation-
3376 defined.

3377

3378 **3.7.7 DateTimeResolutionEnum**

3379 The DateTimeResolutionEnum class is an enum class that enumerates the instances each of which
3380 expresses a resolution of a date time value.

3381 The DateTimeResolutionEnum has attribute values:

3382

3383 **localNamespace**

3384 Value: icom_core

3385

3386 **localName**

3387 Value: DateTimeResolutionEnum

3388

3389 **extendsFrom**

3390 Value: icom_core:DateTimeResolution

3391

3392 **stereotype**

3393 Value: primary

3394

3395 **isEnumeration**
3396 Value: TRUE
3397
3398 **description**
3399 Value: Resolution of a date time value.
3400
3401 **instances**
3402 Value: <icom_core:Year, icom_core:Date, icom_core:Time>
3403

3404 ICOM defines three date time resolutions:

- 3405 • **icom_core:Year** date time resolution is in years.
 - 3406 • **icom_core:Date** date time resolution is in years and days.
 - 3407 • **icom_core:Time** date time resolution is in years, days, and time of day.
- 3408

3409 **3.7.8 TimeZone**

3410 **3.7.8.1 Description**

3411 A time zone is a region that has a uniform standard time.

3412 **3.7.8.2 Class Definition**

3413 The TimeZone class has attribute values:

3414
3415 **localNamespace**
3416 Value: icom_core
3417
3418 **localName**
3419 Value: TimeZone
3420
3421 **extendsFrom**
3422 Value:
3423
3424 **stereotype**
3425 Value: primary
3426
3427 **description**
3428 Value: A time zone is a region that has a uniform standard time.
3429
3430 **propertyDefinitions**
3431 The values for this attribute are defined in Section 3.7.8.3.

3432 **3.7.8.3 Property Definitions**

3433 The TimeZone class inherits property definitions from super classes.

3434 The TimeZone class MUST have the property definitions:

3435

3436 **icom_core:ID**

3437 Description: Identifier of a time zone.

3438 Required: False

3439 Inherited: False

3440 Property Type: String

3441 Cardinality: Single

3442 Updatability: On Create

3443

3444 **icom_core:rawOffset**

3445 Description: An offset to add to Universal Coordinated Time (UTC) to get
3446 local time. If Daylight Saving Time is in effect at the specified
3447 date, the offset value is adjusted with the amount of daylight
3448 saving.

3449 Required: False

3450 Inherited: False

3451 Property Type: Integer

3452 Cardinality: Single

3453 Updatability: On Create

3454

3455 The TimeZone class MAY include additional property definitions which are implementation-defined.

3456

3457 **3.7.9 Location**

3458 **3.7.9.1 Description**

3459 A location object represents a physical location which is defined by name, description, and geo
3460 coordinates.

3461 Note: The name of a location may remain unchanged while a physical location may be changing. For
3462 example, a location name might be "On an airplane" while a physical location might be the geo
3463 coordinates of a flight path or current coordinates of a plane.

3464 **3.7.9.2 Class Definition**

3465 The Location class has attribute values:

3466

3467 **localNamespace**

3468 Value: icom_core

3469

3470 **localName**

3471 Value: Location

3472

3473 **extendsFrom**

3474 Value:

3475

3476 **stereotype**
 3477 Value: primary
 3478
 3479 **description**
 3480 Value: A location object represents a physical location which is defined by name, description, or
 3481 geo coordinates.
 3482
 3483 **propertyDefinitions**
 3484 The values for this attribute are defined in Section 3.7.9.3.

3.7.9.3 Property Definitions

3485
 3486 The Location class MUST have the property definitions:
 3487

icom_core:name

3489 Description: Name of a location.
 3490 Required: False
 3491 Inherited: False
 3492 Property Type: String
 3493 Cardinality: Single
 3494 Updatability: Read Write

3495

icom_core:description

3497 Description: A description of a location.
 3498 Required: False
 3499 Inherited: False
 3500 Property Type: String
 3501 Cardinality: Single
 3502 Updatability: Read Write

3503

icom_core:timeZone

3505 Description: Time zone of a location.
 3506 Required: False
 3507 Inherited: False
 3508 Property Type: icom_core:TimeZone
 3509 Cardinality: Single
 3510 Updatability: Read Write

3511

icom_core:coordinates

3513 Description: A list of geo coordinates marking a point, path, or area of a
 3514 physical location.
 3515 Required: False
 3516 Inherited: False
 3517 Property Type: icom_core:GeoCoordinates

3518 Cardinality: Multi
3519 Updatability: Read Write

3520
3521 The Location class MAY include additional property definitions which are implementation-defined.
3522

3523 3.7.10 GeoCoordinates

3524 3.7.10.1 Description

3525 A geo coordinates object specifies the latitude, longitude, and altitude of a physical location.

3526 3.7.10.2 Class Definition

3527 The GeoCoordinates class has attribute values:

3528
3529 **localNamespace**
3530 Value: icom_core

3531
3532 **localName**
3533 Value: GeoCoordinates

3534
3535 **extendsFrom**
3536 Value:

3537
3538 **stereotype**
3539 Value: primary

3540
3541 **description**
3542 Value: A geo coordinates object specifies the latitude, longitude, and altitude of a physical
3543 location.

3544
3545 **propertyDefinitions**
3546 The values for this attribute are defined in Section 3.7.10.3.

3547 3.7.10.3 Property Definitions

3548 The GeoCoordinates class MUST have the property definitions:

3549
3550 **icom_core:latitude**
3551 Description: Latitude of a location.
3552 Required: False
3553 Inherited: False
3554 Property Type: Float
3555 Cardinality: Single
3556 Updatability: Read Write

3557

3558 **icom_core:longitude**
3559 Description: Longitude of a location.
3560 Required: False
3561 Inherited: False
3562 Property Type: Float
3563 Cardinality: Single
3564 Updatability: Read Write

3565

3566 **icom_core:altitude**
3567 Description: Altitude of a location.
3568 Required: False
3569 Inherited: False
3570 Property Type: Float
3571 Cardinality: Single
3572 Updatability: Read Write

3573

3574 The GeoCoordinates class MAY include additional property definitions which are implementation-defined.

3575

3576

4 Extension Modules

3577

4.1 Overview of Extension Modules

3578

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

3582

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

3583

3584

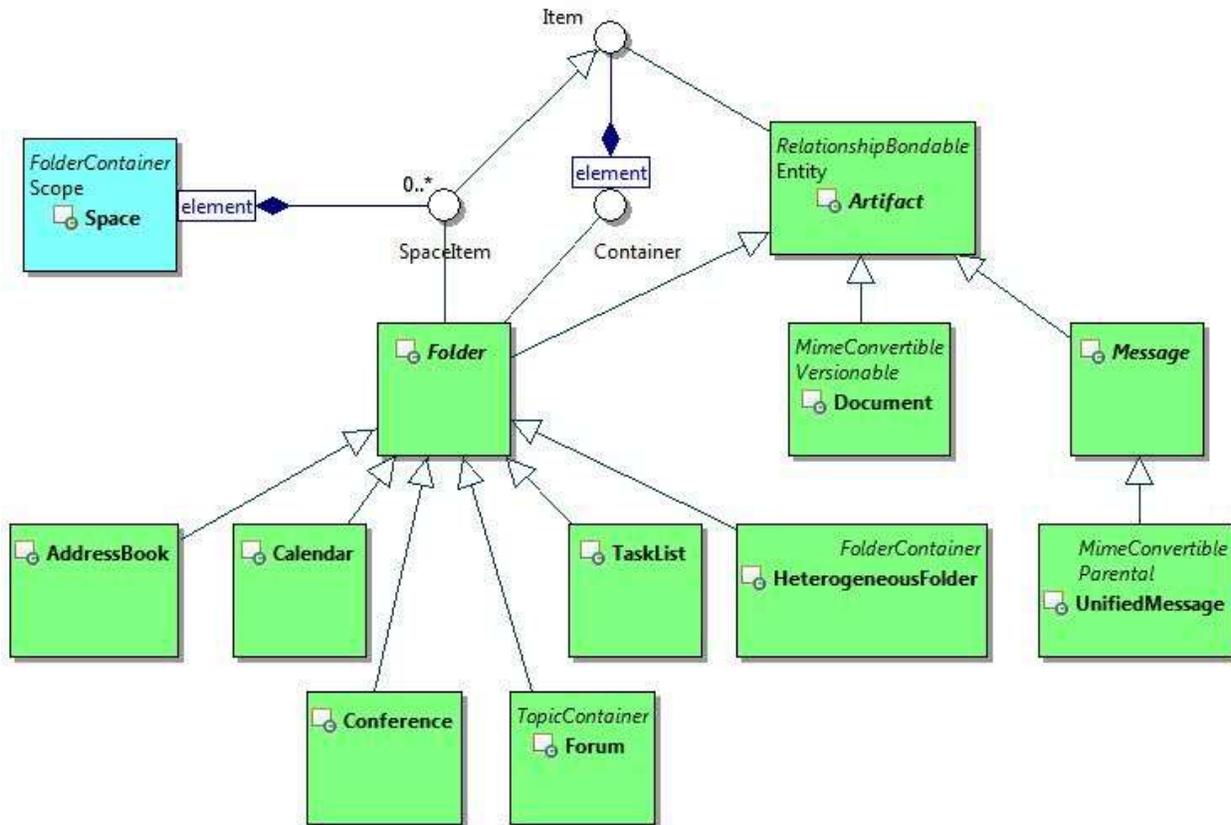
3585

3586

3587

3588

3589



3590

3591 *Figure 25: Containers of Collaboration Activities.*

3592 ICOM defines containers that provide contexts and structures for specific areas of collaborative activities.
3593 The UML class diagram in Figure 25 depicts a Space as a hub of containers, including
3594 HeterogeneousFolder, AddressBook, Calendar, TaskList, Forum, and Conference. These containers are
3595 briefly described as follows:

3596

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

3597

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

3598

- 3599 • an inbox or outbox for communication, or
 - 3600 • a trash folder to archive all types of artifacts deleted from a space.
- 3601 **AddressBook** is a specialized container to manage contact or personal information, such as
- 3602 addresses, phone numbers, birthdays, anniversaries, and other entries.
- 3603 **Calendar** is a specialized container to support time management.
- 3604 **TaskList** is a specialized container to support task coordination.
- 3605 **Forum** is a specialized container to support
- 3606 • **Topic** sub-containers for threaded discussions and
 - 3607 • **Announcement** sub-containers for time-sensitive communication.
- 3608 **Conference** is a specialized container that provides a durable context for real-time interactions.
- 3609

3610 The following ten modules are specified as extension modules of ICOM:

- 3611 1. Content Module (in Section 4.2) defines Content, MultiContent, and SimpleContent. A content
- 3612 represents a piece of data in a document or message. Content, multi-content, simple content, and
- 3613 online content form a composite design pattern.
- 3614 2. Document Module (in Section 4.3) defines Document, WikiPage, and version control model. A
- 3615 document can contain a composite content defined in Section 4.2. Documents are typically
- 3616 contained by heterogeneous folders.
- 3617 3. Message Module (in Section 4.4) defines Message, UnifiedMessage, InstantMessage, and
- 3618 related classes. A message can contain a composite content defined in Section 4.2. Unified
- 3619 messages are typically contained by heterogeneous folders.
- 3620 4. Presence Module (in Section 4.5) defines Presence, Activity, and Contact Method. Presence
- 3621 represents a watchable state of a presentity (which is usually a person). Presence state is derived
- 3622 using an actor's subscriptions.

3623 Note: Since a Presence is derived using a viewer's subscriptions, a Presence should not be shared

3624 with other viewers. For this reason, Presence is not modeled as Entity and is not assigned an access

3625 control list.

- 3626 5. Address Book Module (in Section 4.6) defines AddressBook and PersonContact. A person
- 3627 contact can bookmark a reference to a person in an ICOM community as well as store addresses,
- 3628 phone numbers, and other entries about a person who may not be in any ICOM community.
- 3629 6. Calendar Module (in Section 4.7) defines Calendar, Occurrence, and OccurrenceSeries.
- 3630 Occurrence artifacts are used to resolve the free-busy times of participants for scheduling of
- 3631 meetings and booking of rooms and other resources.
- 3632 7. Free Busy Module (in Section 4.8) defines FreeBusy. FreeBusy is a view derived from
- 3633 occurrences in a calendar or a set of calendars using an actor's privileges to determine the free
- 3634 or busy states of calendar occurrences.

3635 Note: Since a FreeBusy view is derived using a viewer's privileges, a FreeBusy should not be shared

3636 with other viewers. For this reason, FreeBusy is not modeled as Entity and is not assigned an access

3637 control list.

- 3638 8. Task List Module (in Section 4.9) defines TaskList and Task. Tasks are used to coordinate the
- 3639 assignment of tasks and to track the progress of task activities.
- 3640 9. Forum Module (in Section 4.10) defines Forum, Topic, Announcement, and DiscussionMessage.
- 3641 Topics, announcements, and discussions are used for treaded discussions. Moderators of a
- 3642 forum can prune, merge, or fork the discussion threads.
- 3643 10. Conference Module (in Section 4.11) defines Conference and related classes. A conference can
- 3644 contain visual, audio, and chat transcripts of the conference sessions. It also contains the current
- 3645 status, conference settings, past sessions, active session, and activity logs.

3646 **4.2 Content Module**

3647 **4.2.1 MimeConvertible**

3648 **4.2.1.1 Description**

3649 A MimeConvertible object represents an object that has Multipurpose Internet Mail Extensions (MIME)
3650 characteristics such as headers, content transfer encoding, and possible hierarchy of sub-contents.

3651 **4.2.1.2 Class Definition**

3652 The MimeConvertible class is a mixin class that defines the characteristics of objects that can be
3653 represented in MIME format.

3654 The MimeConvertible class has attribute values:

3655

3656 **localNamespace**

3657 Value: icom_content

3658

3659 **localName**

3660 Value: MimeConvertible

3661

3662 **extendsFrom**

3663 Value: icom_core:Identifiable

3664

3665 **stereotype**

3666 Value: mixin

3667

3668 **description**

3669 Value: MimeConvertible class is a mixin class that defines the characteristics of objects that can
3670 be represented in MIME format.

3671

3672 **propertyDefinitions**

3673 The values for this attribute are defined in Section 4.2.1.3.

3674 **4.2.1.3 Property Definitions**

3675 The MimeConvertible class inherits property definitions from super classes.

3676 The MimeConvertible class MAY include additional property definitions which are implementation-defined.

3677

3678 **4.2.2 Content**

3679 **4.2.2.1 Description**

3680 A content object represents a piece of data in a document or message. Content, multi-content, simple
3681 content, and online content form a composite design pattern.

3682 **4.2.2.2 Class Definition**

3683 The Content class has attribute values:

3684

3685 **localNamespace**

3686 Value: icom_content

3687

3688 **localName**

3689 Value: Content

3690

3691 **extendsFrom**

3692 Value: icom_core:Identifiable, icom_content:MimeConvertible

3693

3694 **stereotype**

3695 Value: primary

3696

3697 **isAbstract**

3698 Value: TRUE

3699

3700 **description**

3701 Value: Content represents a piece of data in a document or message.

3702

3703 **propertyDefinitions**

3704 The values for this attribute are defined in Section 4.2.2.3.

3705 **4.2.2.3 Property Definitions**

3706 The Content class inherits property definitions from super classes.

3707 The Content class MUST have the property definitions:

3708

3709 **icom_content:contentId**

3710 Description: A content id is a unique identifier for a part of content in multi-
3711 part contents.

3712 Required: False

3713 Inherited: False

3714 Property Type: String

3715 Cardinality: Single

3716 Updatability: Read Write

3717

3718 **icom_content:mediaType**

3719 Description: Media type is a two-part identifier for Internet file formats.

3720 Required: False

3721 Inherited: False

3722 Property Type: String

3723 Cardinality: Single

3743 **4.2.3.2 Class Definition**

3744 The MultiContent class has attribute values:

- 3745
- 3746 **localNamespace**
- 3747 Value: icom_content
- 3748
- 3749 **localName**
- 3750 Value: MultiContent
- 3751
- 3752 **extendsFrom**
- 3753 Value: icom_content:Content
- 3754
- 3755 **stereotype**
- 3756 Value: primary
- 3757
- 3758 **description**
- 3759 Value: A multi-content object represents the multiple parts of a message or document.
- 3760
- 3761 **propertyDefinitions**
- 3762 The values for this attribute are defined in Section 4.2.3.3.

3763 **4.2.3.3 Property Definitions**

3764 The MultiContent class inherits property definitions from super classes.

3765 The MultiContent class MUST have the property definitions:

- 3766
- 3767 **icom_content:part**
- 3768 Description: Zero or more parts of a hierarchical composite content.
- 3769 Required: False
- 3770 Inherited: False
- 3771 Property Type: icom_content:MimeConvertible
- 3772 Cardinality: Multi
- 3773 Updatability: Read Write

3774

3775 The MultiContent class MAY include additional property definitions which are implementation-defined.

3776

3777 **4.2.4 SimpleContent**

3778 **4.2.4.1 Description**

3779 A simple content holds a single piece of data.

3780 **4.2.4.2 Class Definition**

3781 The SimpleContent class has attribute values:

3782

3783 **localNamespace**

3784 Value: icom_content

3785

3786 **localName**

3787 Value: SimpleContent

3788

3789 **extendsFrom**

3790 Value: icom_content:Content

3791

3792 **stereotype**

3793 Value: primary

3794

3795 **description**

3796 Value: A simple content holds a single piece of data.

3797

3798 **propertyDefinitions**

3799 The values for this attribute are defined in Section 4.2.4.3.

3800 **4.2.4.3 Property Definitions**

3801 The SimpleContent class inherits property definitions from super classes.

3802 The SimpleContent class MUST have the property definitions:

3803

3804 **icom_content:characterEncoding**

3805 Description: Character encoding specifies character set of a content (a
3806 missing value means that a piece of content should be
3807 treated as binary or raw).

3808 Required: False

3809 Inherited: False

3810 Property Type: String

3811 Cardinality: Single

3812 Updatability: Read Write

3813

3814 **icom_content:contentEncoding**

3815 Description: Content encoding specifies encoding of a piece of content.

3816 Required: False

3817 Inherited: False

3818 Property Type: String

3819 Cardinality: Single

3820 Updatability: Read Write

3821

3863 **extendsFrom**
3864 Value: icom_content:Content
3865
3866 **stereotype**
3867 Value: primary
3868
3869 **description**
3870 Value: An online content holds an online artifact attached to a message or invitation.
3871
3872 **propertyDefinitions**
3873 The values for this attribute are defined in Section 4.2.5.3.

3874 **4.2.5.3 Property Definitions**

3875 The OnlineContent class inherits property definitions from super classes.
3876 The OnlineContent class **MUST** have the property definition:

3877
3878 **icom_content:onlineAttachment**
3879 Description: An online artifact attached to a message.
3880 Required: True
3881 Inherited: False
3882 Property Type: icom_core:Artifact
3883 Cardinality: Single
3884 Updatability: Read Write

3885
3886 The OnlineContent class **MAY** include additional property definitions which are implementation-defined.
3887

3888 **4.2.6 ContentDispositionType**

3889 **4.2.6.1 Description**

3890 A content disposition type is a presentation style of content.

3891 **4.2.6.2 Class Definition**

3892 The ContentDispositionType class is a mixin class which defines a presentation style of content.
3893 The ContentDispositionType class has attribute values:

3894
3895 **localNamespace**
3896 Value: icom_content
3897
3898 **localName**
3899 Value: ContentDispositionType
3900

3901 **extendsFrom**
3902 Value:
3903
3904 **stereotype**
3905 Value: mixin
3906
3907 **description**
3908 Value: ContentDispositionType is a mixin class which defines a presentation style of content.
3909
3910 **propertyDefinitions**
3911 The values for this attribute are defined in Section 4.2.6.3.

3912 **4.2.6.3 Property Definitions**

3913 The ContentDispositionType class MAY include additional property definitions which are implementation-
3914 defined.
3915

3916 **4.2.7 ContentDispositionTypeEnum**

3917 The ContentDispositionTypeEnum class is an enum class that enumerates the instances each of which
3918 expresses a presentation style of content.
3919 The ContentDispositionTypeEnum class has attribute values:

3920
3921 **localNamespace**
3922 Value: icom_content
3923
3924 **localName**
3925 Value: ContentDispositionTypeEnum
3926
3927 **extendsFrom**
3928 Value: icom_content:ContentDispositionType
3929
3930 **stereotype**
3931 Value: primary
3932
3933 **isEnumeration**
3934 Value: TRUE
3935
3936 **description**
3937 Value: A presentation style of content.
3938
3939 **instances**
3940 Value: <icom_content:Inline, icom_content:Attachment>
3941

3942 ICOM defines two content disposition types:
3943 • **icom_content:Inline** content is to be displayed automatically upon display of the main body of an
3944 artifact.
3945 • **icom_content:Attachment** content is separate from the main body of an artifact, and that its
3946 display should not be automatic, but contingent upon some further action of a user.
3947

3948 **4.2.8 AttachedItem**

3949 **4.2.8.1 Description**

3950 An attached item holds a content for an occurrence, task, and contact artifact.

3951 **4.2.8.2 Class Definition**

3952 The AttachedItem class has attribute values:

3953 **localNamespace**
3954 Value: icom_content
3955
3956 **localName**
3957 Value: AttachedItem
3958
3959 **extendsFrom**
3960 Value:
3961
3962 **stereotype**
3963 Value: primary
3964
3965 **description**
3966 Value: An attachedItem holds a content for an occurrence, task, and contact artifact.
3967
3968 **propertyDefinitions**
3969 The values for this attribute are defined in Section 4.2.8.3.
3970

3971 **4.2.8.3 Property Definitions**

3972 The AttachedItem class MUST have the property definitions:

3973 **icom_core:name**
3974 Description: Name of a content attachment.
3975 Required: True
3976 Inherited: False
3977 Property Type: String
3978 Cardinality: Single
3979 Updatability: Read Write
3980
3981

3982	icom_content:content	
3983	Description:	A content attached to an occurrence, task, or contact artifact.
3984	Required:	True
3985	Inherited:	False
3986	Property Type:	icom_content:Content
3987	Cardinality:	Single
3988	Updatability:	Read Write

3989

3990 The AttachedItem class MAY include additional property definitions which are implementation-defined.

3991

3992 4.3 Document Module

3993 4.3.1 Versionable

3994 4.3.1.1 Description

3995 A versionable artifact is

- 3996 1. a non-version-controlled copy,
- 3997 2. a specific versioned copy,
- 3998 3. a private working copy, or
- 3999 4. a representative copy (optional)

4000 of an artifact version series.

4001 When a versionable artifact is not under version control, a non-version-controlled copy MUST be the only
4002 copy in a version series, i.e. there is only one copy and one *objectId*.

4003 When a versionable artifact is under version control, a representative copy MAY provide a version-
4004 independent view of a versionable artifact.

4005 When a non-version-controlled copy is placed under version control, a versioned copy MUST be created.
4006 Assignment of an object identifier to a versioned copy is implementation-dependent:

- 4007 • if a versioned copy retains the object identifier of a non-version-controlled copy, the version type
4008 of a versionable artifact MUST change from NonVersionControlledCopy to VersionedCopy;
- 4009 • if a versioned copy is assigned a new object identifier that is different from the object identifier of
4010 a non-version-controlled copy, a representative copy MAY retain the object identifier of the non-
4011 version-controlled copy;
- 4012 • if both versioned copy and representative copy are assigned new object identifiers that are
4013 different from the object identifier of a non-version-controlled copy, the non-version-controlled
4014 copy SHALL be discarded.

4015 When a private working copy is checked in, a versioned copy MUST be created. Assignment of an object
4016 identifier to a versioned copy is implementation-dependent:

- 4017 • if a versioned copy retains the object identifier of a private working copy, the version type of a
4018 versionable artifact MUST change from PrivateWorkingCopy to VersionedCopy;
- 4019 • if a versioned copy is assigned a new object identifier that is different from the object identifier of
4020 a private working copy, the private working copy SHALL be discarded.

4021 It is optional for a service provider to provide a representative copy for a version series. If a representative
4022 copy is provided:

- 4023 • a representative copy MUST have its own object identifier that is different from the object
4024 identifier of any versioned copy or private working copy;

- 4025 • assignment of an object identifier to a representative copy is implementation-dependent:
 - 4026 ○ a representative copy MAY retain the object identifier of a non-version-controlled copy; if
 - 4027 so the version type of a versionable artifact MUST change from
 - 4028 NonVersionControlledCopy to RepresentativeCopy;
 - 4029 ○ a representative copy MAY be assigned a new object identifier that is different from the
 - 4030 object identifier of a non-version-controlled copy;
- 4031 • content and state of a representative copy is implementation-dependent:
 - 4032 ○ a representative copy MAY be a copy of the content and state of the latest versioned
 - 4033 copy or the latest major versioned copy in a version series;
 - 4034 ○ a representative copy MAY be a copy of the content and state of a private working copy if
 - 4035 the current user loading the representative copy is the same user who checks out a
 - 4036 version series.

4037 Note: Each versioned copy of a versionable artifact is itself a versionable artifact, i.e. it has its own
 4038 *objectId*. A versioned copy has a version number, label, and check in comment.

4039 Note: A private working copy is a versionable artifact created by an explicit checkout operation on a
 4040 versionable artifact under version control. The properties for a private working copy are identical to the
 4041 properties of a versioned copy on which a checkout operation was performed. Certain properties such as
 4042 *objectId* and *creationDate* are different from a versioned copy. The content of a private working copy is
 4043 identical to the content of a versioned copy. Its object identifier is different from that of the representative
 4044 copy or any versioned copy.

4045 A private working copy MAY be saved in a version series for sharing and co-editing, however, it needs
 4046 not be visible to users who may only have permissions to view other versioned copies in a version series.

4047 Note: Until it is checked in using an explicit check-in operation, a private working copy must not be
 4048 considered the LatestMajorVersion in a version series.

4049 A container of a versionable artifact CAN contain a representative copy so that it provides a version-
 4050 independent view of a state of the version series.

4051 Note: Starting from a representative copy in a container, an actor can traverse a version series to retrieve
 4052 any versioned copy or private working copy.

4053 ICOM version control model is based on the CMIS version control model specified in Section 2.1.9 of
 4054 Content Management Interoperability Services Version 1.0 [CMIS].

4055 4.3.1.2 Class Definition

4056 The Versionable class is a mixin class that defines the characteristics of artifacts that can be versioned.

4057 The Versionable class has attribute values:

4058

4059 **localNamespace**

4060 Value: icom_doc

4061

4062 **localName**

4063 Value: Versionable

4064

4065 **extendsFrom**

4066 Value: icom_core:Identifiable

4067

4068 **stereotype**

4069 Value: mixin

4070

4071 **description**
4072 Value: Versionable class is a mixin class that defines the characteristics of artifacts that can be
4073 versioned.

4074

4075 **propertyDefinitions**

4076 The values for this attribute are defined in Section 4.3.1.3.

4077 **4.3.1.3 Property Definitions**

4078 The Versionable class inherits property definitions from super classes.

4079 The Versionable class MUST have the property definitions:

4080

4081 **icom_doc:versionControlMetadata**

4082 Description: A version control metadata object attached to a versionable
4083 artifact.

4084 Required: False

4085 Inherited: False

4086 Property Type: icom_doc:VersionControlMetadata

4087 Cardinality: Single

4088 Updatability: Read Only

4089

4090 **icom_doc:versionType**

4091 Description: A type of version controlled copy of a versionable artifact.

4092 Required: False

4093 Inherited: False

4094 Property Type: icom_doc:VersionType

4095 Cardinality: Single

4096 Updatability: Read Only

4097

4098 The Versionable class MAY include additional property definitions which are implementation-defined.

4099

4100 **4.3.2 VersionControlMetadata**

4101 **4.3.2.1 Description**

4102 A version control metadata is an object that contains version control information.

4103 There are two classes of version control metadata: version series and version. A version control metadata
4104 of a versionable artifact is either a version series or a version depending on the version type.

4105 • If the version type is icom_doc:NonVersionControlledCopy then metadata is optional; if metadata
4106 is present, it MUST be a version series object.

4107 • If the version type is icom_doc:RepresentativeCopy, then metadata MUST be a version series
4108 object.

4109 • If the version type is icom_doc:VersionedCopy or icom_doc:PrivateWorkingCopy, then metadata
4110 MUST be a version object.

4111 **4.3.2.2 Class Definition**

4112 The VersionControlMetadata class is a mixin class that defines the characteristics of version or version
4113 series metadata for version control.

4114 The VersionControlMetadata class has attribute values:

4115

4116 **localNamespace**

4117 Value: icom_doc

4118

4119 **localName**

4120 Value: VersionControlMetadata

4121

4122 **extendsFrom**

4123 Value: icom_core:Identifiable

4124

4125 **stereotype**

4126 Value: mixin

4127

4128 **description**

4129 Value: VersionControlMetadata is a mixin class that defines the characteristics of entities that
4130 serve as metadata for version control.

4131

4132 **propertyDefinitions**

4133 The values for this attribute are defined in Section 4.3.2.3.

4134 **4.3.2.3 Property Definitions**

4135 The VersionControlMetadata class inherits property definitions from super classes.

4136 The VersionControlMetadata class MUST have the property definition:

4137

4138 **icom_doc:representativeCopy**

4139 Description: A representative copy of a versionable artifact.

4140 Required: False

4141 Inherited: False

4142 Property Type: icom_doc:Versionable

4143 Cardinality: Single

4144 Updatability: Read Only

4145

4146 The VersionControlMetadata class MAY include additional property definitions which are implementation-
4147 defined.

4148

4149 **4.3.3 VersionSeries**

4150 **4.3.3.1 Description**

4151 A version series is a version control metadata that contains a version history and check in/out states of a
4152 versionable artifact.

4153 A version series object is a version control metadata of a representative copy of a versionable artifact.

4154 **4.3.3.2 Class Definition**

4155 The VersionSeries class has attribute values:

4156

4157 **localNamespace**

4158 Value: icom_doc

4159

4160 **localName**

4161 Value: VersionSeries

4162

4163 **extendsFrom**

4164 Value: icom_core:Entity, icom_doc:VersionControlMetadata, icom_meta:RelationshipBondable

4165

4166 **stereotype**

4167 Value: primary

4168

4169 **description**

4170 Value: A version series is version control metadata that contains a version history and check
4171 in/out states of a versionable artifact.

4172

4173 **propertyDefinitions**

4174 The values for this attribute are defined in Section 4.3.3.3.

4175 **4.3.3.3 Property Definitions**

4176 The VersionSeries class inherits property definitions from super classes.

4177 The VersionSeries class MUST have the property definitions:

4178

4179 **icom_doc:versionHistory**

4180 Description: A history of version nodes of a versionable artifact.

4181 Required: False

4182 Inherited: False

4183 Property Type: icom_doc:Version

4184 Cardinality: Multi

4185 Updatability: Read Only

4186

4187 **icom_doc:versionableHistory**

4188 Description: A history of the versioned copies of a versionable artifact.

4189	Required:	False
4190	Inherited:	False
4191	Property Type:	icom_doc:Versionable
4192	Cardinality:	Multi
4193	Updatability:	Read Only
4194		
4195	icom_doc:latestVersionedCopy	
4196	Description:	Latest versioned copy of a versionable artifact.
4197	Required:	False
4198	Inherited:	False
4199	Property Type:	icom_doc:Versionable
4200	Cardinality:	Single
4201	Updatability:	Read Only
4202		
4203	icom_doc:privateWorkingCopy	
4204	Description:	A private working copy of a versionable artifact.
4205	Required:	False
4206	Inherited:	False
4207	Property Type:	icom_doc:Versionable
4208	Cardinality:	Single
4209	Updatability:	Read Only
4210		
4211	icom_doc:versionSeriesCheckedOut	
4212	Description:	Indicates whether a version series is checked out.
4213	Required:	False
4214	Inherited:	False
4215	Property Type:	Boolean
4216	Cardinality:	Single
4217	Updatability:	Read Only
4218		
4219	icom_doc:versionSeriesCheckedOutBy	
4220	Description:	An actor who checks out a version series.
4221	Required:	False
4222	Inherited:	False
4223	Property Type:	icom_core:Actor
4224	Cardinality:	Single
4225	Updatability:	Read Only
4226		
4227	icom_doc:versionSeriesCheckedOutOn	
4228	Description:	The time when a version series is checked out.
4229	Required:	False
4230	Inherited:	False

4231 Property Type: DateTime
4232 Cardinality: Single
4233 Updatability: Read Only

4234

4235 **icom_doc:versionSeriesCheckoutComment**

4236 Description: A check out comment of a version series.
4237 Required: False
4238 Inherited: False
4239 Property Type: String
4240 Cardinality: Single
4241 Updatability: Read Only

4242

4243 **icom_doc:totalSize**

4244 Description: Total size of all versioned copies of a versionable artifact in a
4245 version series.
4246 Required: False
4247 Inherited: False
4248 Property Type: Integer
4249 Cardinality: Single
4250 Updatability: Read Only

4251

4252 The VersionSeries class MAY include additional property definitions which are implementation-defined.

4253

4254 **4.3.4 Version**

4255 **4.3.4.1 Description**

4256 A version is a version control metadata that contains a version number, label, and description.

4257 A version object is a version control metadata of a versioned copy or a private working copy of a
4258 versionable artifact.

4259 **4.3.4.2 Class Definition**

4260 The Version class has attribute values:

4261

4262 **localNamespace**

4263 Value: icom_doc

4264

4265 **localName**

4266 Value: Version

4267

4268 **extendsFrom**

4269 Value: icom_core:Entity, icom_doc:VersionControlMetadata, icom_meta:RelationshipBondable

4270

4313	Cardinality:	Single
4314	Updatability:	Read Write
4315		
4316	icom_doc:versionedOrPrivateWorkingCopy	
4317	Description:	A versioned copy or private working copy corresponding to a
4318		version of a versionable artifact.
4319	Required:	False
4320	Inherited:	False
4321	Property Type:	icom_doc:Versionable
4322	Cardinality:	Single
4323	Updatability:	Read Only

4324

4325 The Version class MAY include additional property definitions which are implementation-defined.

4326

4327 4.3.5 VersionType

4328 4.3.5.1 Description

4329 A version type is a version state of a copy of versionable document.

4330 4.3.5.2 Class Definition

4331 The VersionType class is a mixin class which defines a version state of a copy of versionable document.

4332 The VersionType class has attribute values:

4333		
4334	localNamespace	
4335	Value:	icom_doc
4336		
4337	localName	
4338	Value:	VersionType
4339		
4340	extendsFrom	
4341	Value:	
4342		
4343	stereotype	
4344	Value:	mixin
4345		
4346	description	
4347	Value:	VersionType is a mixin class which defines a version state of a copy of versionable
4348		document.
4349		
4350	propertyDefinitions	
4351		The values for this attribute are defined in Section 4.3.5.3.

4352 **4.3.5.3 Property Definitions**

4353 The VersionType class MAY include additional property definitions which are implementation-defined.

4354

4355 **4.3.6 VersionTypeEnum**

4356 The VersionTypeEnum class is an enum class that enumerates the instances each of which expresses a
4357 version type.

4358 The VersionTypeEnum class has attribute values:

4359

4360 **localNamespace**

4361 Value: icom_doc

4362

4363 **localName**

4364 Value: VersionTypeEnum

4365

4366 **extendsFrom**

4367 Value: icom_doc:VersionType

4368

4369 **stereotype**

4370 Value: primary

4371

4372 **isEnumeration**

4373 Value: TRUE

4374

4375 **description**

4376 Value: A version type of a copy of versionable document.

4377

4378 **instances**

4379 Value: <icom_doc:NonVersionControlledCopy, icom_doc:VersionedCopy,
4380 icom_doc:PrivateWorkingCopy, icom_doc:RepresentativeCopy>

4381

4382 ICOM defines four version types:

- 4383 • **icom_doc:NonVersionControlledCopy** a versionable artifact is not under version control.
- 4384 • **icom_doc:VersionedCopy** a versionable artifact is a version of an artifact version series.
- 4385 • **icom_doc:PrivateWorkingCopy** a versionable artifact is a private working copy of an artifact
4386 version series.
- 4387 • **icom_doc:RepresentativeCopy** a versionable artifact is a version-independent representative
4388 copy of an artifact. This version type is optional and implementation-dependent.

4389

4390 **4.3.7 Document**

4391 **4.3.7.1 Description**

4392 A document is a versionable artifact that can contain a single content of a media type or composite
4393 contents of an assortment of media types.

4394 **4.3.7.2 Class Definition**

4395 The Document class has attribute values:

4396

4397 **localNamespace**

4398 Value: icom_doc

4399

4400 **localName**

4401 Value: Document

4402

4403 **extendsFrom**

4404 Value: icom_core:Artifact, icom_doc:Versionable, icom_content:MimeConvertible

4405

4406 **stereotype**

4407 Value: primary

4408

4409 **description**

4410 Value: A document is a versionable artifact that may contain a single content of a media type or
4411 composite contents of an assortment of media types.

4412

4413 **propertyDefinitions**

4414 The values for this attribute are defined in Section 4.3.7.3.

4415 **4.3.7.3 Property Definitions**

4416 The Document class inherits property definitions from super classes.

4417 The Document class MUST have the property definitions:

4418

4419 **icom_content:content**

4420 Description: Content of a document.

4421 Required: False

4422 Inherited: False

4423 Property Type: icom_content:Content

4424 Cardinality: Single

4425 Updatability: Read Write

4426

4427 **icom_doc:size**

4428 Description: The size of a copy of a document.

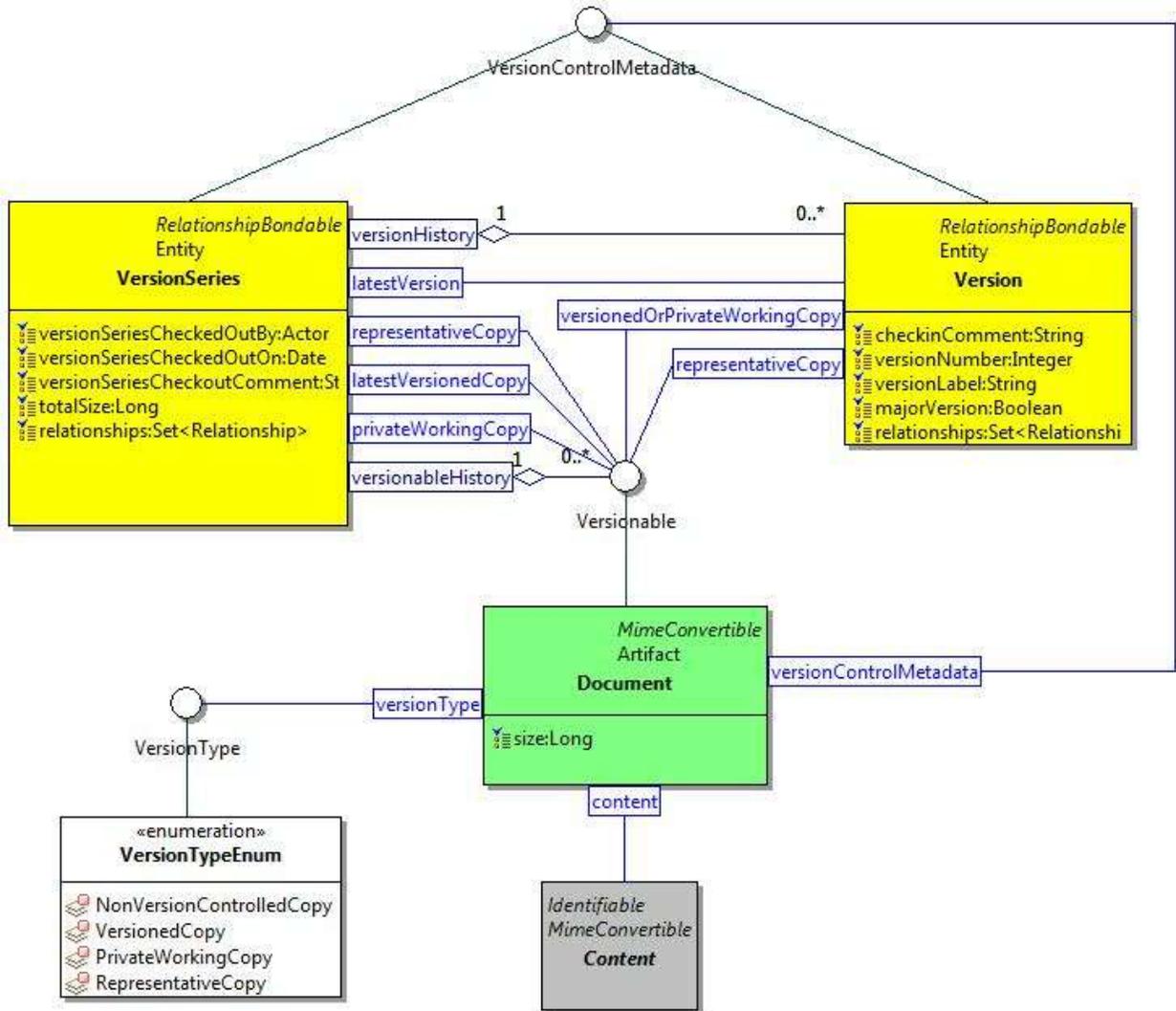
4429 Required: False

4430 Inherited: False
 4431 Property Type: Integer
 4432 Cardinality: Single
 4433 Updatability: Read Only

4434

4435 The Document class MAY include additional property definitions which are implementation-defined.

4436



4437

4438 Figure 27: Document, Version Series, and Version Class Diagram.

4439

4440 4.3.8 WikiPage

4441 4.3.8.1 Description

4442 A wiki page is a document that contains a wiki content and that provides an html page generated from the
 4443 wiki content.

4444 **4.3.8.2 Class Definition**

4445 The WikiPage class has attribute values:

4446

4447 **localNamespace**

4448 Value: icom_doc

4449

4450 **localName**

4451 Value: WikiPage

4452

4453 **extendsFrom**

4454 Value: icom_doc:Document

4455

4456 **stereotype**

4457 Value: primary

4458

4459 **description**

4460 Value: A wiki page is a document that contains a wiki content and that provides an html page
4461 generated from the wiki content.

4462

4463 **propertyDefinitions**

4464 The values for this attribute are defined in Section 4.3.8.3.

4465 **4.3.8.3 Property Definitions**

4466 The WikiPage class inherits property definitions from super classes.

4467 The WikiPage class MUST have the property definitions:

4468

4469 **icom_doc:renderedPage**

4470 Description: An html page generated from a wiki content.

4471 Required: False

4472 Inherited: False

4473 Property Type: String

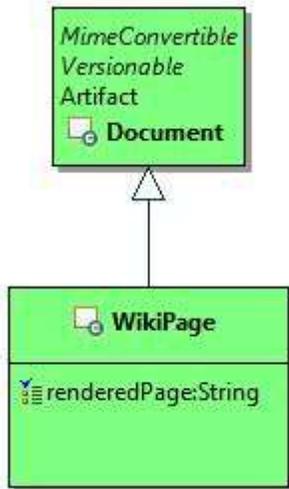
4474 Cardinality: Single

4475 Updatability: Read Only

4476

4477 The WikiPage class MAY include additional property definitions which are implementation-defined.

4478



4479
4480 *Figure 28: Wiki Page Class Diagram.*

4481
4482 **4.4 Message Module**

4483 **4.4.1 Message**

4484 **4.4.1.1 Description**

4485 A message is a unit of conversation. It holds a simple content or multipart message contents in a content
4486 property. It has a single sender.

4487 Note: The delivered time is the time when a message is delivered to a given recipient. The user creation
4488 date and time property can be used as the sent date and time of a message. The name property can be
4489 used as the subject of a message.

4490 **4.4.1.2 Class Definition**

4491 The Message class has attribute values:

- 4492
- 4493 **localNamespace**
- 4494 Value: icom_msg
- 4495
- 4496 **localName**
- 4497 Value: Message
- 4498
- 4499 **extendsFrom**
- 4500 Value: icom_core:Artifact
- 4501
- 4502 **stereotype**
- 4503 Value: primary
- 4504
- 4505 **isAbstract**
- 4506 Value: TRUE
- 4507

4508 **description**
4509 Value: A message is a unit of conversation.

4510 **propertyDefinitions**
4511 The values for this attribute are defined in Section 4.4.1.3.

4513 **4.4.1.3 Property Definitions**

4514 The Message class inherits property definitions from super classes.
4515 The Message class MUST have the property definitions:

4517 **icom_content:content**

4518	Description:	Content of a message
4519	Required:	False
4520	Inherited:	False
4521	Property Type:	icom_content:Content
4522	Cardinality:	Single
4523	Updatability:	Read Write

4525 **icom_msg:sender**

4526	Description:	Sender of a message.
4527	Required:	False
4528	Inherited:	False
4529	Property Type:	icom_core:Participant
4530	Cardinality:	Single
4531	Updatability:	Read Write

4533 **icom_msg:deliveredTime**

4534	Description:	The date and time when a message is delivered to a given
4535		recipient.
4536	Required:	False
4537	Inherited:	False
4538	Property Type:	DateTime
4539	Cardinality:	Single
4540	Updatability:	Read Only

4541
4542 The Message class MAY include additional property definitions which are implementation-defined.

4544 **4.4.2 UnifiedMessage**

4545 **4.4.2.1 Description**

4546 A unified message is a type of message delivered electronically over a computer, voice, fax, and other
4547 networks.

- 4548 A unified message can be one of these types:
- 4549 • Email is a message delivered electronically over a computer network.
 - 4550 • Voice is a message that contains a voice or audio stream.
 - 4551 • Fax is a message that contains an image transmitted via phone lines using the fax protocol.
 - 4552 • Notification is a type of message sent by applications.

4553 4.4.2.2 Class Definition

4554 The UnifiedMessage class has attribute values:

4555 **localNamespace**

4557 Value: icom_msg

4558

4559 **localName**

4560 Value: UnifiedMessage

4561

4562 **extendsFrom**

4563 Value: icom_msg:Message, icom_content:MimeConvertible

4564

4565 **stereotype**

4566 Value: primary

4567

4568 **description**

4569 Value: A unified message is a type of message delivered electronically over a computer, voice,

4570 fax, and other networks.

4571

4572 **propertyDefinitions**

4573 The values for this attribute are defined in Section 4.4.2.3.

4574 4.4.2.3 Property Definitions

4575 The UnifiedMessage class inherits property definitions from super classes.

4576 The UnifiedMessage class MUST have the property definitions:

4577

4578 **icom_core:priority**

4579 Description:	The priority of a message.
4580 Required:	False
4581 Inherited:	False
4582 Property Type:	icom_core:Priority
4583 Cardinality:	Single
4584 Updatability:	Read Write

4585

4586 **icom_content:contentId**

4587 Description:	Content id is a unique identifier for a message part in multi-
4588	part messages.

4589	Required:	False
4590	Inherited:	False
4591	Property Type:	String
4592	Cardinality:	Single
4593	Updatability:	Read Write
4594		
4595	icom_content:mediaType	
4596	Description:	Media type is a two-part identifier for Internet file formats.
4597	Required:	False
4598	Inherited:	False
4599	Property Type:	String
4600	Cardinality:	Single
4601	Updatability:	Read Write
4602		
4603	icom_content:contentDisposition	
4604	Description:	Content disposition specifies a presentation style.
4605	Required:	False
4606	Inherited:	False
4607	Property Type:	icom_content:ContentDispositionType
4608	Cardinality:	Single
4609	Updatability:	Read Write
4610		
4611	icom_msg:envelopeSender	
4612	Description:	An envelope sender is a participant to receive bounced message. It is also known as return path.
4613		
4614	Required:	False
4615	Inherited:	False
4616	Property Type:	icom_core:Participant
4617	Cardinality:	Single
4618	Updatability:	Read Write
4619		
4620	icom_msg:toReceivers	
4621	Description:	A list of participants to receive a message.
4622	Required:	False
4623	Inherited:	False
4624	Property Type:	icom_core:Participant
4625	Cardinality:	Multi
4626	Updatability:	Read Write
4627		
4628	icom_msg:ccReceivers	
4629	Description:	A list of participants to receive carbon-copies of a message.
4630	Required:	False
4631	Inherited:	False

4632	Property Type:	icom_core:Participant
4633	Cardinality:	Multi
4634	Updatability:	Read Write
4635		
4636	icom_msg:bccReceivers	
4637	Description:	A list of participants to receive blind-carbon-copies of a message.
4638		
4639	Required:	False
4640	Inherited:	False
4641	Property Type:	icom_core:Participant
4642	Cardinality:	Multi
4643	Updatability:	Read Write
4644		
4645	icom_msg:replyTo	
4646	Description:	A list of participants to receive a reply message.
4647	Required:	False
4648	Inherited:	False
4649	Property Type:	icom_core:Participant
4650	Cardinality:	Multi
4651	Updatability:	Read Write
4652		
4653	icom_msg:flag	
4654	Description:	Zero or more flags on a message.
4655	Required:	False
4656	Inherited:	False
4657	Property Type:	icom_msg:UnifiedMessageFlag
4658	Cardinality:	Multi
4659	Updatability:	Read Write
4660		
4661	icom_msg:messageDispositionNotificationRequested	
4662	Description:	A message disposition notification requested for a message.
4663	Required:	False
4664	Inherited:	False
4665	Property Type:	Boolean
4666	Cardinality:	Single
4667	Updatability:	Read Write
4668		
4669	icom_msg:messageDeliveryStatusNotificationRequest	
4670	Description:	Indicates the types of delivery status notifications requested for a message. Default is icom_msg:Failure.
4671		
4672	Required:	False
4673	Inherited:	False
4674	Property Type:	icom_msg:UnifiedMessageDeliveryStatusNotificationRequest

4675	Cardinality:	Multi
4676	Updatability:	Read Write
4677		
4678	icom_msg:channel	
4679	Description:	Indicates the delivery channel of a message.
4680	Required:	False
4681	Inherited:	False
4682	Property Type:	icom_msg:UnifiedMessageChannel
4683	Cardinality:	Single
4684	Updatability:	Read Write
4685		
4686	icom_msg:editMode	
4687	Description:	Indicates an editable mode (new, draft, or delivered) of a message.
4688		
4689	Required:	False
4690	Inherited:	False
4691	Property Type:	icom_msg:UnifiedMessageEditMode
4692	Cardinality:	Single
4693	Updatability:	Read Only
4694		
4695	icom_msg:mimeHeader	
4696	Description:	A list of headers. Each header is represented by a multi-valued property.
4697		
4698	Required:	False
4699	Inherited:	False
4700	Property Type:	icom_meta:Property
4701	Cardinality:	Multi
4702	Updatability:	Read Write
4703		
4704	icom_msg:size	
4705	Description:	The size of a unified message.
4706	Required:	False
4707	Inherited:	False
4708	Property Type:	Integer
4709	Cardinality:	Single
4710	Updatability:	Read Only
4711		
4712	The UnifiedMessage class MAY include additional property definitions which are implementation-defined.	
4713		

4714 **4.4.3 UnifiedMessageParticipant**

4715 **4.4.3.1 Description**

4716 A unified message participant object represents the participation of an addressable entity in a unified
4717 message.

4718 **4.4.3.2 Class Definition**

4719 The UnifiedMessageParticipant class has attribute values:

4720

4721 **localNamespace**

4722 Value: icom_msg

4723

4724 **localName**

4725 Value: UnifiedMessageParticipant

4726

4727 **extendsFrom**

4728 Value: icom_core:Participant

4729

4730 **stereotype**

4731 Value: primary

4732

4733 **description**

4734 Value: A unified message participant object represents the participation of an addressable entity
4735 in a unified message.

4736

4737 **propertyDefinitions**

4738 The values for this attribute are defined in Section 4.4.3.3.

4739 **4.4.3.3 Property Definitions**

4740 The UnifiedMessageParticipant class inherits property definitions from super classes.

4741 The UnifiedMessageParticipant class MUST have the property definitions:

4742

4743 **icom_msg:fullAddress**

4744 Description: Full address of a participant.

4745 Required: False

4746 Inherited: False

4747 Property Type: IRI

4748 Cardinality: Single

4749 Updatability: Read Write

4750

4751 **icom_msg:localPart**

4752 Description: Local part of a full address.

4753 Required: False

4754 Inherited: False
4755 Property Type: String
4756 Cardinality: Single
4757 Updatability: Read Write

4758

4759 **icom_msg:domainPart**

4760 Description: Domain part of a full address.
4761 Required: False
4762 Inherited: False
4763 Property Type: String
4764 Cardinality: Single
4765 Updatability: Read Write

4766

4767 The UnifiedMessageParticipant class MAY include additional property definitions which are
4768 implementation-defined.

4769

4770 **4.4.4 UnifiedMessageFlag**

4771 **4.4.4.1 Description**

4772 A unified message flag is a flag on a message.

4773 **4.4.4.2 Class Definition**

4774 The UnifiedMessageFlag class is a mixin class which defines a flag on a message.

4775 The UnifiedMessageFlag class has attribute values:

4776

4777 **localNamespace**

4778 Value: icom_msg

4779

4780 **localName**

4781 Value: UnifiedMessageFlag

4782

4783 **extendsFrom**

4784 Value:

4785

4786 **stereotype**

4787 Value: mixin

4788

4789 **description**

4790 Value: UnifiedMessageFlag is a mixin class which defines a flag on a message.

4791

4792 **propertyDefinitions**

4793 The values for this attribute are defined in Section 4.4.4.3.

4794 **4.4.4.3 Property Definitions**

4795 The UnifiedMessageFlag class MAY include additional property definitions which are implementation-
4796 defined.

4797

4798 **4.4.5 UnifiedMessageFlagEnum**

4799 The UnifiedMessageFlagEnum class is an enum class that enumerates the instances each of which
4800 expresses a flag on a message.

4801 The UnifiedMessageFlagEnum class has attribute values:

4802

4803 **localNamespace**

4804 Value: icom_msg

4805

4806 **localName**

4807 Value: UnifiedMessageFlagEnum

4808

4809 **extendsFrom**

4810 Value: icom_msg:UnifiedMessageFlag

4811

4812 **stereotype**

4813 Value: primary

4814

4815 **isEnumeration**

4816 Value: TRUE

4817

4818 **description**

4819 Value: A flag on a message.

4820

4821 **instances**

4822 Value: <icom_msg:Answered, icom_msg:Forwarded, icom_msg:Redirected, icom_msg:Hidden,
4823 icom_msg:MarkedForDelete, icom_msg:MarkedForFollowUp, icom_msg:MarkedForDraft,
4824 icom_msg:MessageDispositionNotificationProcessed>

4825

4826 ICOM defines eight flags:

- 4827 • **icom_msg:Answered** a message is answered.
- 4828 • **icom_msg:Forwarded** a message is forwarded.
- 4829 • **icom_msg:Redirected** a message is redirected.
- 4830 • **icom_msg:Hidden** a message is hidden.
- 4831 • **icom_msg:MarkedForDelete** a message is marked for delete.
- 4832 • **icom_msg:MarkedForFollowUp** a message is marked for follow up.
- 4833 • **icom_msg:MarkedForDraft** a message is marked for draft.
- 4834 • **icom_msg:MessageDispositionNotificationProcessed** a message disposition notification is
4835 processed.

4836

4837 **4.4.6 UnifiedMessageDeliveryStatusNotificationRequest**

4838 **4.4.6.1 Description**

4839 A unified message delivery status notification request is a directive for notifying a participant of delivery
4840 status of a message.

4841 **4.4.6.2 Class Definition**

4842 The UnifiedMessageDeliveryStatusNotificationRequest class is a mixin class which defines a directive for
4843 notifying a participant of delivery status of a message.

4844 The UnifiedMessageDeliveryStatusNotificationRequest class has attribute values:

4845

4846 **localNamespace**

4847 Value: icom_msg

4848

4849 **localName**

4850 Value: UnifiedMessageDeliveryStatusNotificationRequest

4851

4852 **extendsFrom**

4853 Value:

4854

4855 **stereotype**

4856 Value: mixin

4857

4858 **description**

4859 Value: UnifiedMessageDeliveryStatusNotificationRequest is a mixin class which defines a
4860 directive for notifying a participant of delivery status of a message.

4861

4862 **propertyDefinitions**

4863 The values for this attribute are defined in Section 4.4.6.3.

4864 **4.4.6.3 Property Definitions**

4865 The UnifiedMessageDeliveryStatusNotificationRequest class MAY include additional property definitions
4866 which are implementation-defined.

4867

4868 **4.4.7 UnifiedMessageDeliveryStatusNotificationRequestEnum**

4869 The UnifiedMessageDeliveryStatusNotificationRequestEnum class is an enum class that enumerates the
4870 instances each of which expresses a request for one of several types of delivery status notification.

4871 The UnifiedMessageDeliveryStatusNotificationRequestEnum class has attribute values:

4872

4873 **localNamespace**

4874 Value: icom_msg

4875

4876 **localName**
4877 Value: UnifiedMessageDeliveryStatusNotificationRequestEnum
4878
4879 **extendsFrom**
4880 Value: icom_msg:UnifiedMessageDeliveryStatusNotificationRequest
4881
4882 **stereotype**
4883 Value: primary
4884
4885 **isEnumeration**
4886 Value: TRUE
4887
4888 **description**
4889 Value: A request for one of several types of delivery status notification.
4890
4891 **instances**
4892 Value: <icom_msg:Never, icom_msg:Success, icom_msg:Failure, icom_msg:Delay>
4893

4894 ICOM defines four delivery status notification requests:

- 4895 • **icom_msg:Never** a sender requests status notification not be returned to the sender under any
4896 condition.
- 4897 • **icom_msg:Success** a sender requests a status notification for successful delivery of a message.
- 4898 • **icom_msg:Failure** a sender requests a status notification for delivery failure of a message.
- 4899 • **icom_msg:Delay** a sender requests a status notification when delivery of a message has been
4900 delayed for an unusual length of time.

4901

4902 **4.4.8 UnifiedMessageChannel**

4903 **4.4.8.1 Description**

4904 A message channel used to deliver a unified message.

4905 **4.4.8.2 Class Definition**

4906 The UnifiedMessageChannel class is a mixin class which defines a channel used to deliver a unified
4907 message.

4908 The UnifiedMessageChannel class has attribute values:

4909
4910 **localNamespace**
4911 Value: icom_msg
4912
4913 **localName**
4914 Value: UnifiedMessageChannel
4915

4916 **extendsFrom**
4917 Value:
4918
4919 **stereotype**
4920 Value: mixin
4921
4922 **description**
4923 Value: UnifiedMessageChannel is a mixin class which defines a channel used to deliver a
4924 unified message.
4925
4926 **propertyDefinitions**
4927 The values for this attribute are defined in Section 4.4.8.3.

4928 **4.4.8.3 Property Definitions**

4929 The UnifiedMessageChannel class MAY include additional property definitions which are implementation-
4930 defined.
4931

4932 **4.4.9 UnifiedMessageChannelEnum**

4933 The UnifiedMessageChannelEnum class is an enum class that enumerates the instances each of which
4934 expresses a type of delivery channel.

4935 The UnifiedMessageChannelEnum class has attribute values:

4936
4937 **localNamespace**
4938 Value: icom_msg
4939
4940 **localName**
4941 Value: UnifiedMessageChannelEnum
4942
4943 **extendsFrom**
4944 Value: icom_msg:UnifiedMessageChannel
4945
4946 **stereotype**
4947 Value: primary
4948
4949 **isEnumeration**
4950 Value: TRUE
4951
4952 **description**
4953 Value: A delivery channel.
4954
4955 **instances**
4956 Value: <icom_msg:Email, icom_msg:Voice, icom_msg:Fax, icom_msg:Notification>
4957

4958 ICOM defines four channel types:
4959 • **icom_msg:Email** delivery channel is email.
4960 • **icom_msg:Voice** delivery channel is voice.
4961 • **icom_msg:Fax** delivery channel is fax.
4962 • **icom_msg:Notification** delivery channel is notification.
4963

4964 **4.4.10 UnifiedMessageEditMode**

4965 **4.4.10.1 Description**

4966 A unified message edit mode is a mode that indicates whether a unified message is editable.

4967 **4.4.10.2 Class Definition**

4968 The UnifiedMessageEditMode class is a mixin class which defines a mode that indicates whether a
4969 unified message is editable.

4970 The UnifiedMessageEditMode class has attribute values:

4971
4972 **localNamespace**
4973 Value: icom_msg
4974
4975 **localName**
4976 Value: UnifiedMessageEditMode
4977
4978 **extendsFrom**
4979 Value:
4980
4981 **stereotype**
4982 Value: mixin
4983
4984 **description**
4985 Value: UnifiedMessageEditMode is a mixin class which defines a mode that indicates whether a
4986 unified message is editable.
4987
4988 **propertyDefinitions**
4989 The values for this attribute are defined in Section 4.4.10.3.

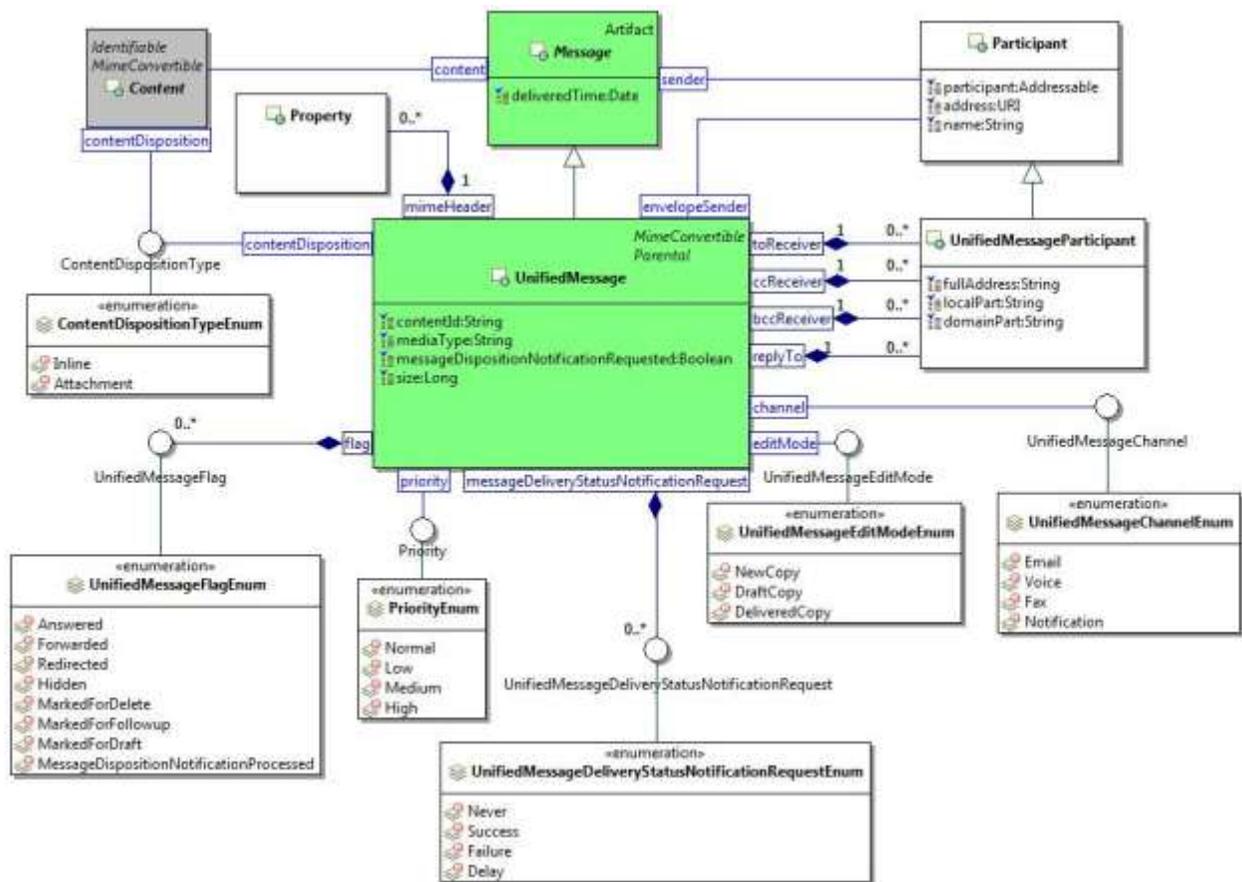
4990 **4.4.10.3 Property Definitions**

4991 The UnifiedMessageEditMode class MAY include additional property definitions which are
4992 implementation-defined.
4993

4994 **4.4.11 UnifiedMessageEditModeEnum**

4995 The UnifiedMessageEditModeEnum class is an enum class that enumerates the instances each of which
4996 expresses whether a message is a new copy, saved draft copy, or delivered copy.

4997 The UnifiedMessageEditModeEnum class has attribute values:
4998
4999 **localNamespace**
5000 Value: icom_msg
5001
5002 **localName**
5003 Value: UnifiedMessageEditModeEnum
5004
5005 **extendsFrom**
5006 Value: icom_msg:UnifiedMessageEditMode
5007
5008 **stereotype**
5009 Value: primary
5010
5011 **isEnumeration**
5012 Value: TRUE
5013
5014 **description**
5015 Value: A message is a new copy, a saved draft copy, or a delivered copy. New or draft copies
5016 are usually editable while delivered copies are usually not editable.
5017
5018 **instances**
5019 Value: <icom_msg:NewCopy, icom_msg:DraftCopy, icom_msg:DeliveredCopy>
5020
5021 ICOM defines three modes:
5022 • **icom_msg:NewCopy** a message is a new message.
5023 • **icom_msg:DraftCopy** a message is saved as a draft.
5024 • **icom_msg:DeliveredCopy** a message is a sent or received message.
5025



5026

5027 *Figure 29: Unified Message Class Diagram.*

5028

5029 4.4.12 InstantMessage

5030 4.4.12.1 Description

5031 An instant message is a type of message for synchronous, usually text based, conversation.

5032 4.4.12.2 Class Definition

5033 The InstantMessage class has attribute values:

5034

5035 **localNamespace**

5036 Value: icom_msg

5037

5038 **localName**

5039 Value: InstantMessage

5040

5041 **extendsFrom**

5042 Value: icom_msg:Message

5043

5044 **stereotype**
5045 Value: primary
5046
5047 **isAbstract**
5048 Value: FALSE
5049
5050 **description**
5051 Value: An instant message is a type of message for synchronous, usually text based,
5052 conversation.
5053
5054 **propertyDefinitions**
5055 The values for this attribute are defined in Section 4.4.12.3.

5056 **4.4.12.3 Property Definitions**

5057 The InstantMessage class inherits property definitions from super classes.

5058 The InstantMessage class **MUST** have the property definitions:

5059

5060 **icom_msg:toReceivers**

5061 Description: A list of participants to receive a message.
5062 Required: False
5063 Inherited: False
5064 Property Type: icom_core:Participant
5065 Cardinality: Multi
5066 Updatability: Read Write

5067

5068 **icom_msg:conversationId**

5069 Description: An identifier of a conversation involving one or more instant
5070 messages.
5071 Required: False
5072 Inherited: False
5073 Property Type: Integer
5074 Cardinality: Single
5075 Updatability: Read Write

5076

5077 **icom_msg:clientId**

5078 Description: An identifier of a client.
5079 Required: False
5080 Inherited: False
5081 Property Type: String
5082 Cardinality: Single
5083 Updatability: Read Write

5084

5085 **icom_msg:formattingStyle**
5086 Description: A style for formatting a rich text message.
5087 Required: False
5088 Inherited: False
5089 Property Type: String
5090 Cardinality: Single
5091 Updatability: Read Write

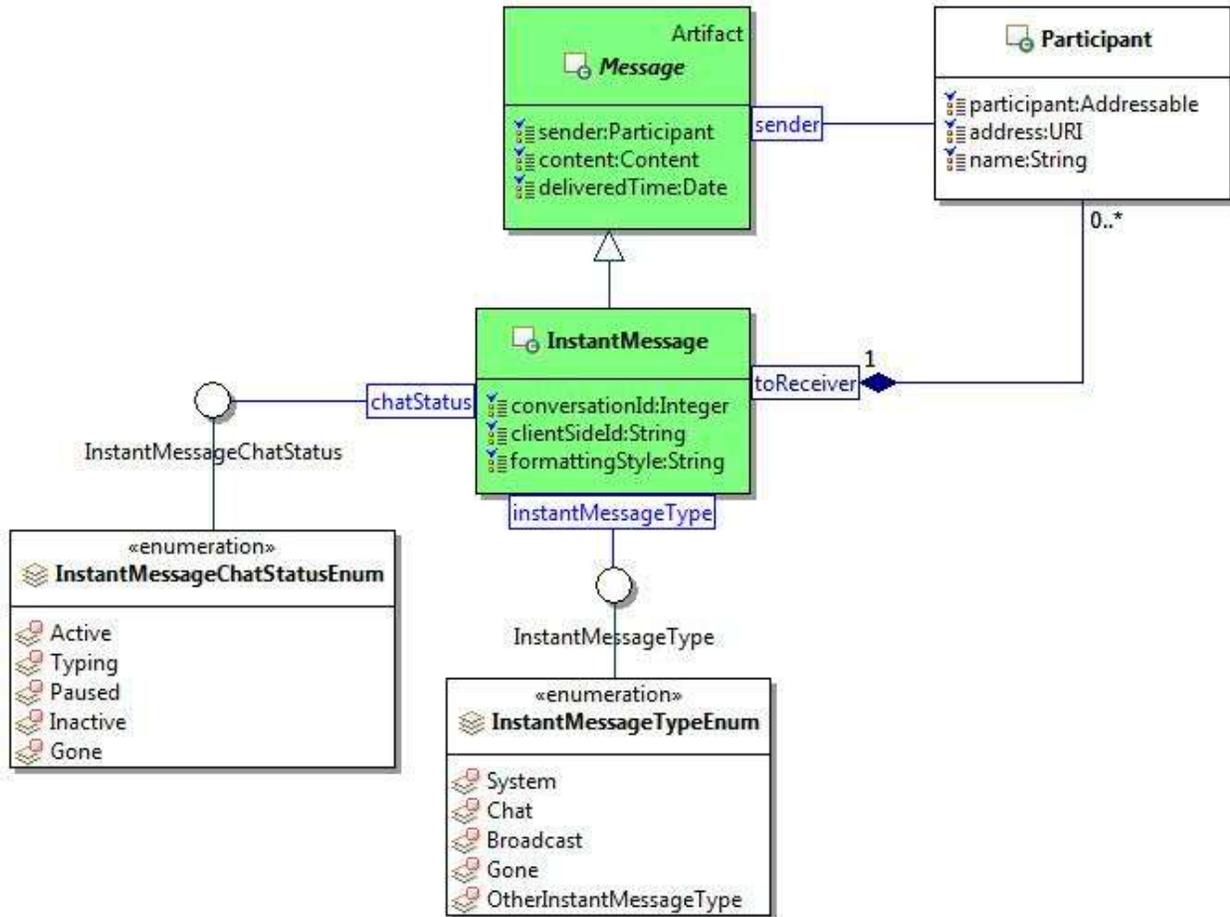
5092

5093 **icom_msg:instantMessageType**
5094 Description: A type of instant message.
5095 Required: False
5096 Inherited: False
5097 Property Type: icom_msg:InstantMessageType
5098 Cardinality: Single
5099 Updatability: Read Write

5100

5101 **icom_msg:chatStatus**
5102 Description: A chat status of a user.
5103 Required: False
5104 Inherited: False
5105 Property Type: icom_msg:InstantMessageChatStatus
5106 Cardinality: Single
5107 Updatability: Read Write

5108



5109
 5110 *Figure 30: Instant Message Class Diagram.*
 5111

5112 **4.4.13 InstantMessageType**

5113 **4.4.13.1 Description**

5114 An instant message type.

5115 **4.4.13.2 Class Definition**

5116 The InstantMessageType class is a mixin class which defines a type of instant message.

5117 The InstantMessageType class has attribute values:

5118
 5119 **localNamespace**
 5120 Value: icom_msg

5121
 5122 **localName**
 5123 Value: InstantMessageType

5124
 5125 **extendsFrom**
 5126 Value:

5127
5128 **stereotype**
5129 Value: mixin
5130
5131 **description**
5132 Value: InstantMessageType is a mixin class which defines a type of instant message.
5133
5134 **propertyDefinitions**
5135 The values for this attribute are defined in Section 4.4.13.3.

5136 **4.4.13.3 Property Definitions**

5137 The InstantMessageType class MAY include additional property definitions which are implementation-
5138 defined.
5139

5140 **4.4.14 InstantMessageTypeEnum**

5141 The InstantMessageTypeEnum class is an enum class that enumerates the instances each of which
5142 expresses a type of instant message.

5143 The InstantMessageTypeEnum class has attribute values:

5144
5145 **localNamespace**
5146 Value: icom_msg
5147
5148 **localName**
5149 Value: InstantMessageTypeEnum
5150
5151 **extendsFrom**
5152 Value: icom_msg:InstantMessageType
5153
5154 **stereotype**
5155 Value: primary
5156
5157 **isEnumeration**
5158 Value: TRUE
5159
5160 **description**
5161 Value: A type of instant message.
5162
5163 **instances**
5164 Value: <icom_msg:System, icom_msg:Chat, icom_msg:Broadcast, icom_msg:Gone,
5165 icom_msg:OtherInstantMessageType>

5167 ICOM defines five instant message types:

- 5168 • **icom_msg:System** an instant message is a system message.

- 5169 • **icom_msg:Chat** an instant message is a chat message.
- 5170 • **icom_msg:Broadcast** an instant message is a broadcast message.
- 5171 • **icom_msg:Gone** an instant message is a message indicating that a user is gone.
- 5172 • **icom_msg:OtherInstantMessageType** an instant message is of other type.
- 5173

5174 **4.4.15 InstantMessageChatStatus**

5175 **4.4.15.1 Description**

5176 An instant message chat status defines a vocabulary of chat status.

5177 **4.4.15.2 Class Definition**

5178 The InstantMessageChatStatus class is a mixin class which defines a chat status.

5179 The InstantMessageChatStatus class has attribute values:

5180

5181 **localNamespace**

5182 Value: icom_msg

5183

5184 **localName**

5185 Value: InstantMessageChatStatus

5186

5187 **extendsFrom**

5188 Value:

5189

5190 **stereotype**

5191 Value: mixin

5192

5193 **description**

5194 Value: InstantMessageChatStatus is a mixin class which defines a chat status.

5195

5196 **propertyDefinitions**

5197 The values for this attribute are defined in Section 4.4.15.3.

5198 **4.4.15.3 Property Definitions**

5199 The InstantMessageChatStatus class MAY include additional property definitions which are
5200 implementation-defined.

5201

5202 **4.4.16 InstantMessageChatStatusEnum**

5203 The InstantMessageChatStatusEnum class is an enum class that enumerates the instances each of
5204 which expresses a chat status of a user.

5205 The InstantMessageChatStatusEnum class has attribute values:

5206

5207 **localNamespace**
5208 Value: icom_msg
5209
5210 **localName**
5211 Value: InstantMessageChatStatusEnum
5212
5213 **extendsFrom**
5214 Value: icom_msg:InstantMessageChatStatus
5215
5216 **stereotype**
5217 Value: primary
5218
5219 **isEnumeration**
5220 Value: TRUE
5221
5222 **description**
5223 Value: A chat status of a user.
5224
5225 **instances**
5226 Value: <icom_msg:Active, icom_msg:Typing, icom_msg:Paused, icom_msg:Inactive,
5227 icom_msg:Gone>
5228
5229 ICOM defines five chat status:
5230 • **icom_msg:Active** a user is active.
5231 • **icom_msg:Typing** a user is typing.
5232 • **icom_msg:Paused** a user has paused.
5233 • **icom_msg:Inactive** a user is inactive.
5234 • **icom_msg:Gone** a user is gone.
5235

5236 **4.4.17 InstantMessageFeed**

5237 **4.4.17.1 Description**

5238 An instant message feed contains a set of instant message connections and a queue of outbound instant
5239 messages.

5240 **4.4.17.2 Class Definition**

5241 The InstantMessageFeed class has attribute values:

5242
5243 **localNamespace**
5244 Value: icom_msg
5245
5246 **localName**
5247 Value: InstantMessageFeed

5248

5249 **extendsFrom**

5250 Value: icom_core:Entity

5251

5252 **stereotype**

5253 Value: primary

5254

5255 **description**

5256 Value: An instant message feed contains a set of instant message connections and a queue of

5257 outbound instant messages.

5258

5259 **propertyDefinitions**

5260 The values for this attribute are defined in Section 4.4.17.3.

5261 **4.4.17.3 Property Definitions**

5262 The InstantMessageFeed class inherits property definitions from super classes.

5263 The InstantMessageFeed class **MUST** have the property definitions:

5264

5265 **icom_msg:connection**

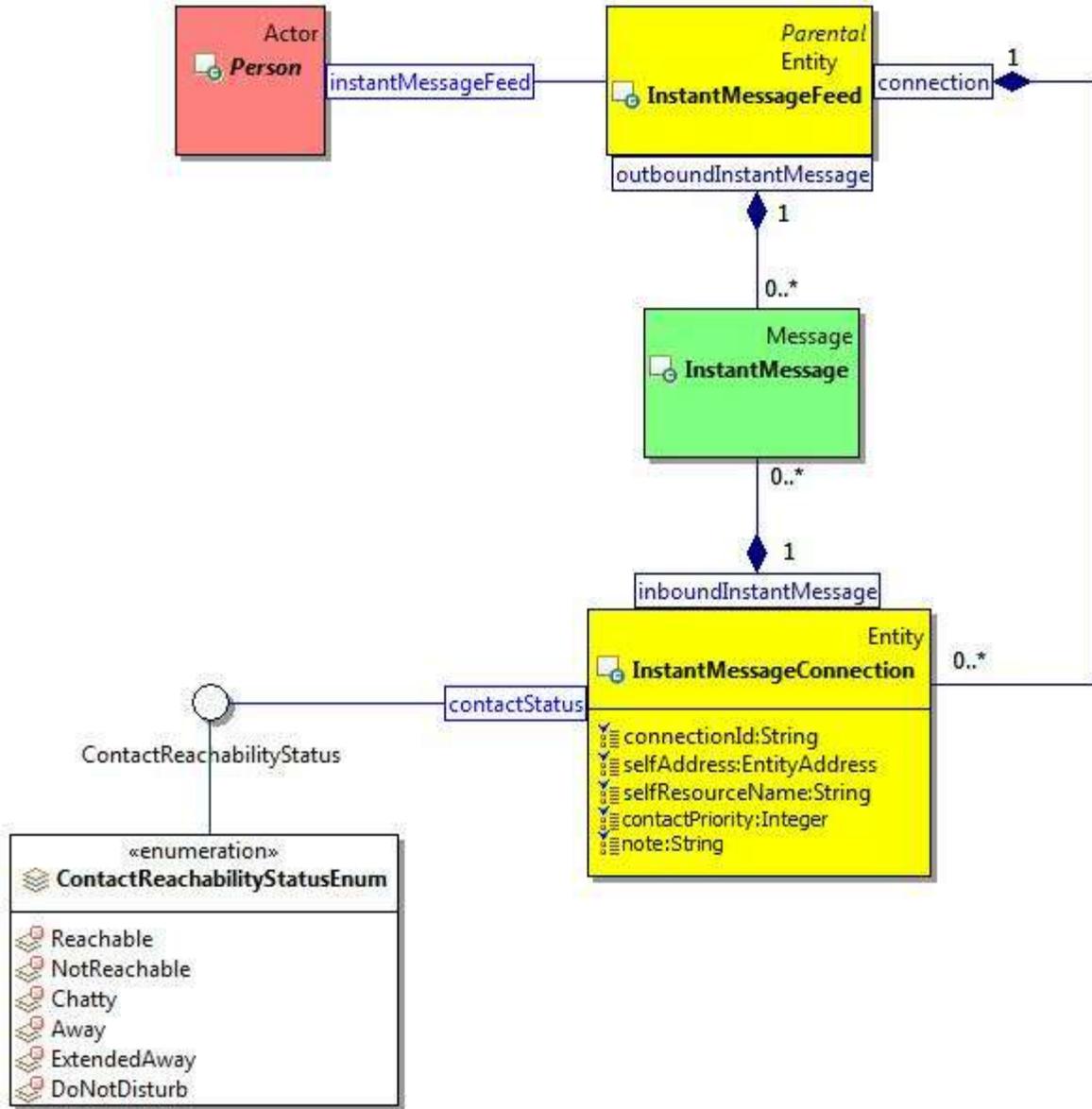
5266	Description:	One or more instant messaging connections.
5267	Required:	False
5268	Inherited:	False
5269	Property Type:	icom_msg:InstantMessageConnection
5270	Cardinality:	Multi
5271	Updatability:	Read Only

5272

5273 **icom_msg:outboundInstantMessage**

5274	Description:	Outbound instant messages.
5275	Required:	False
5276	Inherited:	False
5277	Property Type:	icom_msg:InstantMessage
5278	Cardinality:	Multi
5279	Updatability:	Write Only

5280



5281
5282 *Figure 31: Instant Message Feed and Connection Class Diagram.*

5283

5284 4.4.18 InstantMessageConnection

5285 4.4.18.1 Description

5286 An instant message connection contains queues for inbound instant messages.

5287 A presentity can update the contact status, contact priority, and note for a contact method associated with
5288 a connection.

5289 **4.4.18.2 Class Definition**

5290 The InstantMessageConnection class has attribute values:

- 5291
- 5292 **localNamespace**
- 5293 Value: icom_msg
- 5294
- 5295 **localName**
- 5296 Value: InstantMessageConnection
- 5297
- 5298 **extendsFrom**
- 5299 Value: icom_core:Entity
- 5300
- 5301 **stereotype**
- 5302 Value: primary
- 5303
- 5304 **description**
- 5305 Value: An instant message connection contains queues for inbound instant messages.
- 5306
- 5307 **propertyDefinitions**
- 5308 The values for this attribute are defined in Section 4.4.18.3.

5309 **4.4.18.3 Property Definitions**

5310 The InstantMessageConnection class inherits property definitions from super classes.

5311 The InstantMessageConnection class MUST have the property definitions:

- 5312
- 5313 **icom_msg:connectionId**
- 5314 Description: An identifier of a connection.
- 5315 Required: False
- 5316 Inherited: False
- 5317 Property Type: String
- 5318 Cardinality: Single
- 5319 Updatability: Read Only
- 5320
- 5321 **icom_msg:selfAddress**
- 5322 Description: Address of a presentity who opens a connection.
- 5323 Required: True
- 5324 Inherited: False
- 5325 Property Type: IRI
- 5326 Cardinality: Single
- 5327 Updatability: On Create
- 5328

5329	icom_msg:selfResourceName	
5330	Description:	Resource name associated with a connection.
5331	Required:	True
5332	Inherited:	False
5333	Property Type:	String
5334	Cardinality:	Single
5335	Updatability:	On Create
5336		
5337	icom_msg:inboundInstantMessage	
5338	Description:	Inbound instant messages.
5339	Required:	False
5340	Inherited:	False
5341	Property Type:	icom_msg:InstantMessage
5342	Cardinality:	Multi
5343	Updatability:	Read Only
5344		
5345	icom_presence:contactStatus	
5346	Description:	Reachability status to be propagated to an associated contact
5347		method in presence.
5348	Required:	False
5349	Inherited:	False
5350	Property Type:	icom_presence:ContactReachabilityStatus
5351	Cardinality:	Single
5352	Updatability:	Write Only
5353		
5354	icom_presence:contactPriority	
5355	Description:	Priority to be propagated to an associated contact method in
5356		presence.
5357	Required:	False
5358	Inherited:	False
5359	Property Type:	Integer
5360	Cardinality:	Single
5361	Updatability:	Write Only
5362		
5363	icom_presence:note	
5364	Description:	Note to be propagated to an associated contact method in
5365		presence.
5366	Required:	False
5367	Inherited:	False
5368	Property Type:	String
5369	Cardinality:	Single
5370	Updatability:	Write Only
5371		

5372 **4.5 Presence Module**

5373 **4.5.1 Presence**

5374 **4.5.1.1 Description**

5375 A presence describes the contact methods and activities of a presentity.

5376 It provides a list of contact methods describing how to contact a presentity. A viewer may choose any one
5377 of the contact methods based on circumstances.

5378 It includes a list of activities describing what a presentity is doing.

5379 **4.5.1.2 Class Definition**

5380 The Presence class has attribute values:

5381

5382 **localNamespace**

Value: icom_presence

5384

5385 **localName**

Value: Presence

5387

5388 **extendsFrom**

Value: icom_core:Identifiable

5390

5391 **stereotype**

Value: primary

5393

5394 **description**

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

5396

5397 **propertyDefinitions**

The values for this attribute are defined in Section 4.5.1.3.

5399 **4.5.1.3 Property Definitions**

5400 The Presence class inherits property definitions from super classes.

5401 The Presence class MUST have the property definitions:

5402

5403 **icom_core:lastModificationDate**

5404 Description: Last modification date and time of information in a presence.

5405 Required: False

5406 Inherited: False

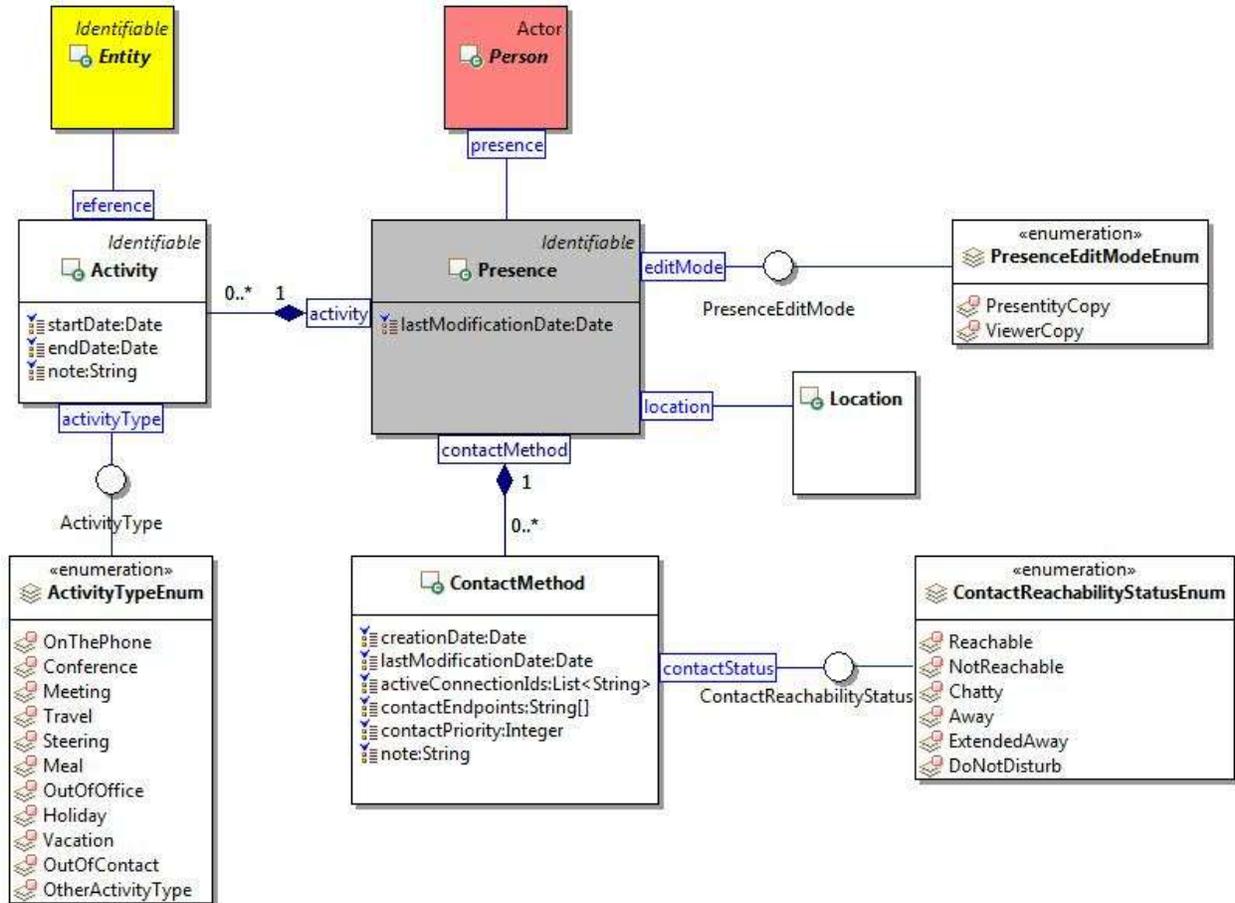
5407 Property Type: DateTime

5408 Cardinality: Single

5409 Updatability: Read Only

5410

5411	icom_core:location	
5412	Description:	Location of a presentity.
5413	Required:	False
5414	Inherited:	False
5415	Property Type:	icom_core:Location
5416	Cardinality:	Single
5417	Updatability:	Read Only
5418		
5419	icom_presence:editMode	
5420	Description:	Indicates a mode which determines whether a presence is
5421		editable.
5422	Required:	False
5423	Inherited:	False
5424	Property Type:	icom_presence:PresenceEditMode
5425	Cardinality:	Single
5426	Updatability:	Read Only
5427		
5428	icom_presence:contactMethod	
5429	Description:	A collection of contact methods describing how to contact a
5430		presentity. A viewer may choose any one of the contact
5431		methods based on circumstances.
5432	Required:	False
5433	Inherited:	False
5434	Property Type:	icom_presence>ContactMethod
5435	Cardinality:	Multi
5436	Updatability:	Read Only
5437		
5438	icom_presence:activity	
5439	Description:	A collection of activities describing what a presentity is doing.
5440	Required:	False
5441	Inherited:	False
5442	Property Type:	icom_presence:Activity
5443	Cardinality:	Multi
5444	Updatability:	Read Only
5445		
5446	The Presence class MAY include additional property definitions which are implementation-defined.	
5447		



5448
5449 *Figure 32: Presence Class Diagram.*

5450
5451 **4.5.2 PresenceEditMode**

5452 **4.5.2.1 Description**

5453 A presence edit mode is a mode that indicates whether a presence is editable.

5454 **4.5.2.2 Class Definition**

5455 The PresenceEditMode class is a mixin class which defines a mode that indicates whether a presence is
5456 editable.

5457 The PresenceEditMode class has attribute values:

- 5458
- 5459 **localNamespace**
 - 5460 Value: icom_presence
 - 5461
 - 5462 **localName**
 - 5463 Value: PresenceEditMode
 - 5464
 - 5465 **extendsFrom**
 - 5466 Value:

5467
5468 **stereotype**
5469 Value: mixin
5470
5471 **description**
5472 Value: PresenceEditMode is a mixin class which defines a mode that indicates whether a
5473 presence is editable.
5474
5475 **propertyDefinitions**
5476 The values for this attribute are defined in Section 4.5.2.3.

5477 **4.5.2.3 Property Definitions**

5478 The PresenceEditMode class MAY include additional property definitions which are implementation-
5479 defined.
5480

5481 **4.5.3 PresenceEditModeEnum**

5482 The PresenceEditModeEnum class is an enum class that enumerates the instances each of which
5483 expresses a mode that indicates whether a presence is editable.

5484 The PresenceEditModeEnum class has attribute values:

5485
5486 **localNamespace**
5487 Value: icom_presence
5488
5489 **localName**
5490 Value: PresenceEditModeEnum
5491
5492 **extendsFrom**
5493 Value: icom_presence:PresenceEditMode
5494
5495 **stereotype**
5496 Value: primary
5497
5498 **isEnumeration**
5499 Value: TRUE
5500
5501 **description**
5502 Value: A mode that indicates whether a presence is editable.
5503
5504 **instances**
5505 Value: <icom_presence:PresentityCopy, icom_presence:ViewerCopy>
5506
5507 ICOM defines two presence editable modes:

- 5508 • **icom_presence:PresentityCopy** a presence is a copy belonging to a presentity who may update
5509 the properties such as activities.
- 5510 • **icom_presence:ViewerCopy** a presence is a copy visible to a subscriber who may not update
5511 the properties.

5512

5513 4.5.4 ContactMethod

5514 4.5.4.1 Description

5515 A contact method object describes reachability circumstances of a presentity.

5516 4.5.4.2 Class Definition

5517 The ContactMethod class has attribute values:

5518

5519 **localNamespace**

5520 Value: icom_presence

5521

5522 **localName**

5523 Value: ContactMethod

5524

5525 **extendsFrom**

5526 Value:

5527

5528 **stereotype**

5529 Value: primary

5530

5531 **description**

5532 Value: A contact method object describes reachability circumstances of a presentity.

5533

5534 **propertyDefinitions**

5535 The values for this attribute are defined in Section 4.5.4.3

5536 4.5.4.3 Property Definitions

5537 The ContactMethod class MUST have the property definitions:

5538

5539 **icom_core:creationDate**

5540 Description: Creation date and time of information in a contact method.

5541 Required: False

5542 Inherited: False

5543 Property Type: DateTime

5544 Cardinality: Single

5545 Updatability: Read Only

5546

5547	icom_core:lastModificationDate	
5548	Description:	Last modification date and time of information in a contact
5549		method.
5550	Required:	False
5551	Inherited:	False
5552	Property Type:	DateTime
5553	Cardinality:	Single
5554	Updatability:	Read Only
5555		
5556	icom_presence:activeConnectionId	
5557	Description:	A list of active connection ids of a presentity.
5558	Required:	False
5559	Inherited:	False
5560	Property Type:	String
5561	Cardinality:	Multi
5562	Updatability:	Read Only
5563		
5564	icom_presence:contactEndpoint	
5565	Description:	A list of endpoints or IRIs for contacting a presentity.
5566	Required:	False
5567	Inherited:	False
5568	Property Type:	String
5569	Cardinality:	Multi
5570	Updatability:	Read Only
5571		
5572	icom_presence:contactPriority	
5573	Description:	Priority of a contact method relative to other contact methods
5574		in a presence.
5575	Required:	False
5576	Inherited:	False
5577	Property Type:	Integer
5578	Cardinality:	Single
5579	Updatability:	Read Only
5580		
5581	icom_presence:contactStatus	
5582	Description:	Status of a contact method in a presence.
5583	Required:	False
5584	Inherited:	False
5585	Property Type:	icom_presence:ContactReachabilityStatus
5586	Cardinality:	Single
5587	Updatability:	Read Only
5588		

5589	icom_presence:note	
5590	Description:	A note about contacting a presentity.
5591	Required:	False
5592	Inherited:	False
5593	Property Type:	String
5594	Cardinality:	Single
5595	Updatability:	Read Only
5596		

5597 **4.5.5 ContactReachabilityStatus**

5598 **4.5.5.1 Description**

5599 A contact reachability status is a status of a contact method.

5600 **4.5.5.2 Class Definition**

5601 The ContactReachabilityStatus class is a mixin class which defines a status of a contact method.

5602 The ContactReachabilityStatus class has attribute values:

5603		
5604	localNamespace	
5605	Value:	icom_presence
5606		
5607	localName	
5608	Value:	ContactReachabilityStatus
5609		
5610	extendsFrom	
5611	Value:	
5612		
5613	stereotype	
5614	Value:	mixin
5615		
5616	description	
5617	Value:	ContactReachabilityStatus is a mixin class which defines a status of a contact method.
5618		
5619	propertyDefinitions	
5620		The values for this attribute are defined in Section 4.5.5.3.

5621 **4.5.5.3 Property Definitions**

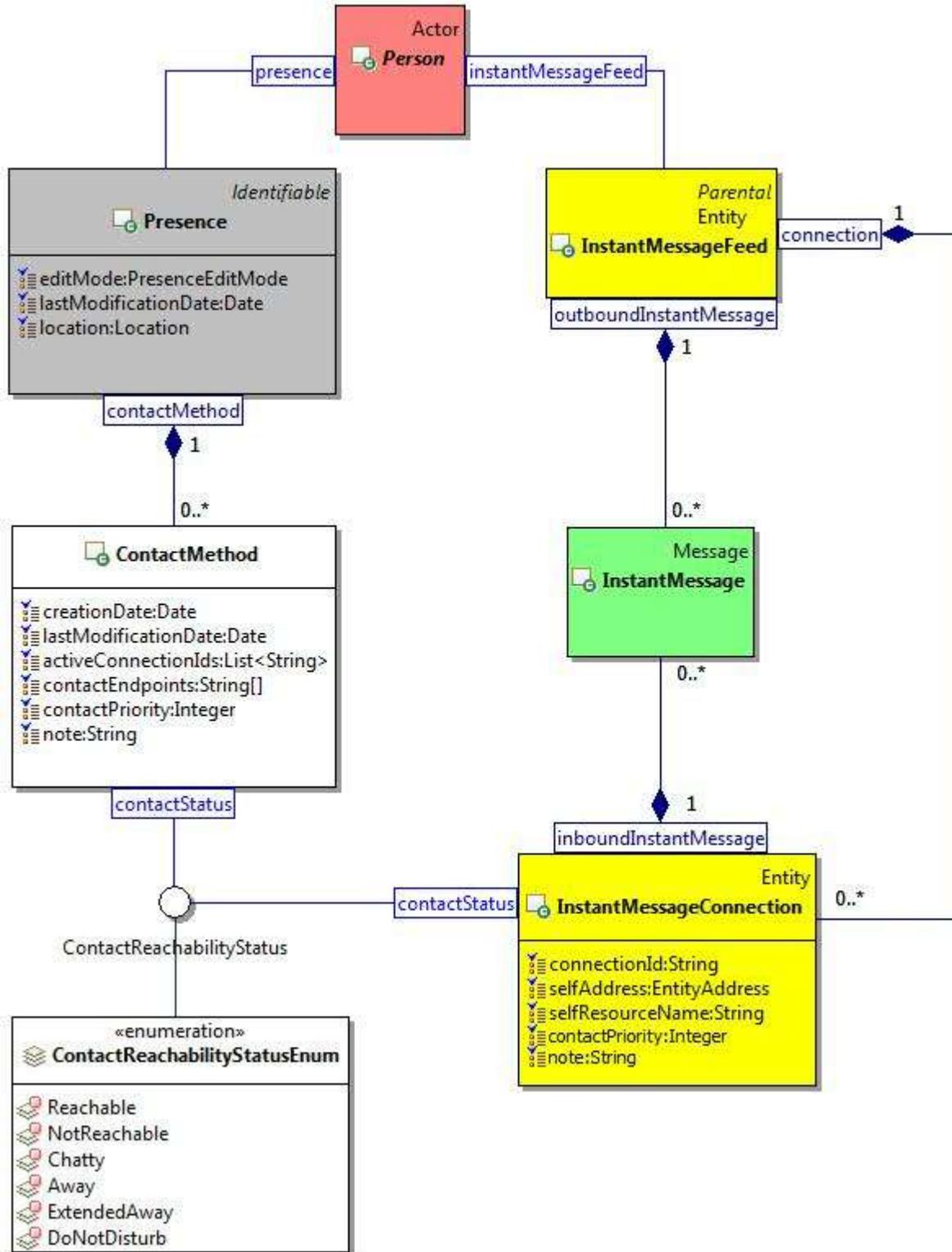
5622 The ContactReachabilityStatus class MAY include additional property definitions which are
5623 implementation-defined.

5624

5625 **4.5.6 ContactReachabilityStatusEnum**

5626 The ContactReachabilityStatusEnum class is an enum class that enumerates the instances each of which
5627 expresses a reachability status of a contact method.

5628 The ContactReachabilityStatusEnum class has attribute values:
5629
5630 **localNamespace**
5631 Value: icom_presence
5632
5633 **localName**
5634 Value: ContactReachabilityStatusEnum
5635
5636 **extendsFrom**
5637 Value: icom_presence:ContactReachabilityStatus
5638
5639 **stereotype**
5640 Value: primary
5641
5642 **isEnumeration**
5643 Value: TRUE
5644
5645 **description**
5646 Value: A reachability status of a contact method.
5647
5648 **instances**
5649 Value: <icom_presence:Reachable, icom_presence:NotReachable, icom_presence:Chatty,
5650 icom_presence:Away, icom_presence:ExtendedAway, icom_presence:DoNotDisturb>
5651
5652 ICOM defines six reachability status:
5653 • **icom_presence:Reachable** a presentity is reachable through a contact method.
5654 • **icom_presence:NotReachable** a presentity is not reachable through a contact method.
5655 • **icom_presence:Chatty** a presentity is chatty.
5656 • **icom_presence:Away** a presentity is away.
5657 • **icom_presence:ExtendedAway** a presentity is away for an extended period.
5658 • **icom_presence:DoNotDisturb** a presentity prefers not to be disturbed.
5659



5660
 5661
 5662

Figure 33: Presence Contact Method and Instant Message Connection Class Diagram.

5663 **4.5.7 Activity**

5664 **4.5.7.1 Description**

5665 An activity object describes what a presentity is doing.

5666 **4.5.7.2 Class Definition**

5667 The Activity class has attribute values:

5668

5669 **localNamespace**

5670 Value: icom_presence

5671

5672 **localName**

5673 Value: Activity

5674

5675 **extendsFrom**

5676 Value:

5677

5678 **stereotype**

5679 Value: primary

5680

5681 **description**

5682 Value: An activity object describes what a presentity is doing.

5683

5684 **propertyDefinitions**

5685 The values for this attribute are defined in Section 4.5.7.3.

5686 **4.5.7.3 Property Definitions**

5687 The Activity class MUST have the property definitions:

5688

5689 **icom_core:startDate**

5690 Description: Start date and time of an activity.

5691 Required: True

5692 Inherited: False

5693 Property Type: DateTime

5694 Cardinality: Single

5695 Updatability: Read Write

5696

5697 **icom_core:endDate**

5698 Description: End date and time of an activity.

5699 Required: True

5700 Inherited: False

5701 Property Type: DateTime

5702 Cardinality: Single

5703	Updatability:	Read Write
5704		
5705	icom_presence:activityType	
5706	Description:	Type of an activity.
5707	Required:	true
5708	Inherited:	False
5709	Property Type:	icom_presence:ActivityType
5710	Cardinality:	Single
5711	Updatability:	Read Write
5712		
5713	icom_presence:note	
5714	Description:	A note describing an activity.
5715	Required:	False
5716	Inherited:	False
5717	Property Type:	String
5718	Cardinality:	Single
5719	Updatability:	Read Write
5720		
5721	icom_presence:reference	
5722	Description:	An entity, such as occurrence, task, conference, etc., which is
5723		the source of or reference for an activity.
5724	Required:	False
5725	Inherited:	False
5726	Property Type:	icom_core:Entity
5727	Cardinality:	Single
5728	Updatability:	Read Write
5729		

5730 **4.5.8 ActivityType**

5731 **4.5.8.1 Description**

5732 An activity type is a vocabulary of activities for rich presence information model.

5733 **4.5.8.2 Class Definition**

5734 The ActivityType class is a mixin class which defines an activity.

5735 The ActivityType class has attribute values:

5736

5737 **localNamespace**

5738 Value: icom_presence

5739

5740 **localName**

5741 Value: ActivityType

5742

5743 **extendsFrom**
5744 Value:
5745
5746 **stereotype**
5747 Value: mixin
5748
5749 **description**
5750 Value: ActivityType is a mixin class which defines a type of activity.
5751
5752 **propertyDefinitions**
5753 The values for this attribute are defined in Section 4.5.8.3.

5754 **4.5.8.3 Property Definitions**

5755 The ActivityType class MAY include additional property definitions which are implementation-defined.
5756

5757 **4.5.9 ActivityTypeEnum**

5758 The ActivityTypeEnum class is an enum class that enumerates the instances each of which expresses a
5759 type of activity.

5760 The ActivityTypeEnum class has attribute values:

5761
5762 **localNamespace**
5763 Value: icom_presence
5764
5765 **localName**
5766 Value: ActivityTypeEnum
5767
5768 **extendsFrom**
5769 Value: icom_presence:ActivityType
5770
5771 **stereotype**
5772 Value: primary
5773
5774 **isEnumeration**
5775 Value: TRUE
5776
5777 **description**
5778 Value: A type of activity.
5779
5780 **instances**
5781 Value: <icom_presence:OnThePhone, icom_presence:Conference, icom_presence:Meeting,
5782 icom_presence:Travel, icom_presence:Steering, icom_presence:Meal,
5783 icom_presence:OutOfOffice, icom_presence:Holiday, icom_presence:Vacation,
5784 icom_presence:OutOfContact, icom_presence:OtherActivityType>

5785
5786
5787
5788
5789
5790
5791
5792
5793
5794
5795
5796
5797
5798

ICOM defines eleven activity types:

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

5799 4.6 Address Book Module

5800 4.6.1 AddressBook

5801 4.6.1.1 Description

5802 An address book is a folder that contains sub-address books and addressable contacts.

5803 4.6.1.2 Class Definition

5804 The AddressBook class has attribute values:

5805
5806
5807
5808
5809
5810
5811
5812
5813
5814
5815
5816
5817
5818
5819
5820
5821
5822

localNamespace

Value: icom_card

localName

Value: AddressBook

extendsFrom

Value: icom_core:Folder

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.6.1.3.

5823 **4.6.1.3 Property Definitions**

5824 The AddressBook class inherits property definitions from super classes.

5825 The AddressBook class MUST have the property definitions:

5826

5827 **icom_card:addressBook**

5828 Description: Sub-address books in an address book.

5829 Required: False

5830 Inherited: False

5831 Property Type: icom_card:AddressBook

5832 Cardinality: Multi

5833 Updatability: Read Only

5834

5835 **icom_card:contact**

5836 Description: Contacts in an address book.

5837 Required: False

5838 Inherited: False

5839 Property Type: icom_card:PersonContact

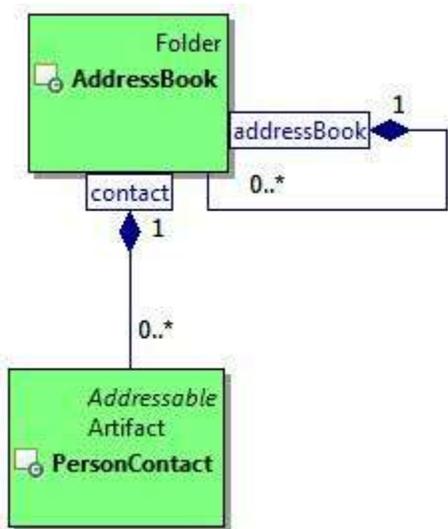
5840 Cardinality: Multi

5841 Updatability: Read Only

5842

5843 The AddressBook class MAY include additional property definitions which are implementation-defined.

5844



5845

5846 *Figure 34: Address Book Class Diagram.*

5847

5848 **4.6.2 PersonContact**

5849 **4.6.2.1 Description**

5850 A person contact is an artifact that contains address information about a person.

5851 **4.6.2.2 Class Definition**

5852 The PersonContact class has attribute values:

5853

5854 **localNamespace**

5855 Value: icom_card

5856

5857 **localName**

5858 Value: PersonContact

5859

5860 **extendsFrom**

5861 Value: icom_core:Artifact, icom_core:Addressable

5862

5863 **stereotype**

5864 Value: primary

5865

5866 **description**

5867 Value: A person contact is an artifact that contains address information about a person.

5868

5869 **propertyDefinitions**

5870 The values for this attribute are defined in Section 4.6.2.3.

5871 **4.6.2.3 Property Definitions**

5872 The PersonContact class inherits property definitions from super classes.

5873 The PersonContact class MUST have the property definitions:

5874

5875 **icom_core:timeZone**

5876 Description: Time zone of a person.

5877 Required: False

5878 Inherited: False

5879 Property Type: icom_core:TimeZone

5880 Cardinality: Single

5881 Updatability: Read Write

5882

5883 **icom_core:givenName**

5884 Description: Given name of a person.

5885 Required: False

5886 Inherited: False

5887 Property Type: String

5888 Cardinality: Single

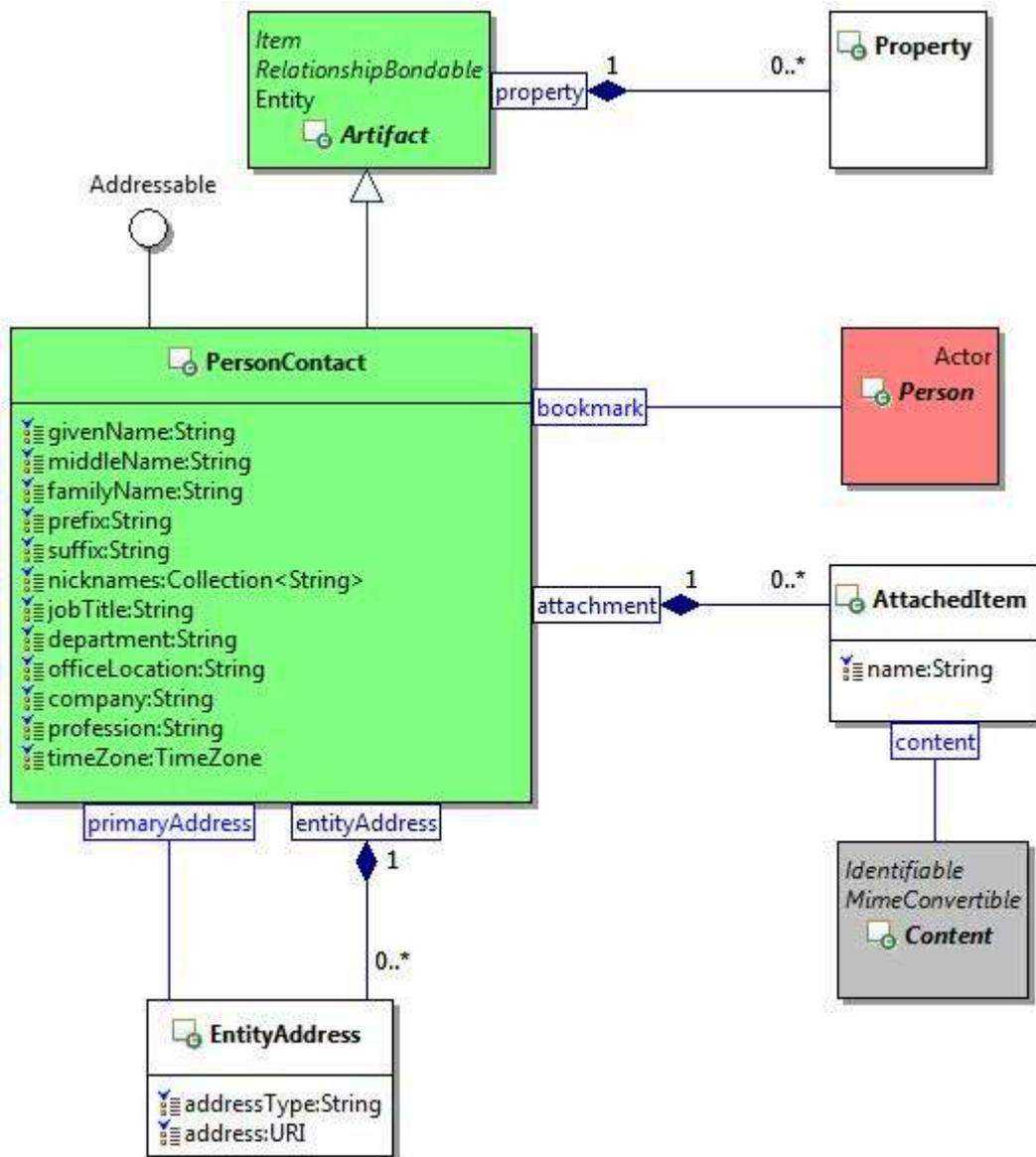
5889 Updatability: Read Write

5890

5891	icom_core:middleName	
5892	Description:	Middle name of a person. Can include multiple names concatenated.
5893		
5894	Required:	False
5895	Inherited:	False
5896	Property Type:	String
5897	Cardinality:	Single
5898	Updatability:	Read Write
5899		
5900	icom_core:familyName	
5901	Description:	Family name of a person.
5902	Required:	False
5903	Inherited:	False
5904	Property Type:	String
5905	Cardinality:	Single
5906	Updatability:	Read Write
5907		
5908	icom_core:prefix	
5909	Description:	Prefix of a person's name.
5910	Required:	False
5911	Inherited:	False
5912	Property Type:	String
5913	Cardinality:	Single
5914	Updatability:	Read Write
5915		
5916	icom_core:suffix	
5917	Description:	Suffix of a person's name.
5918	Required:	False
5919	Inherited:	False
5920	Property Type:	String
5921	Cardinality:	Single
5922	Updatability:	Read Write
5923		
5924	icom_core:nickname	
5925	Description:	Nickname of a person.
5926	Required:	False
5927	Inherited:	False
5928	Property Type:	String
5929	Cardinality:	Multi
5930	Updatability:	Read Write
5931		
5932	icom_core:jobTitle	
5933	Description:	Job title of a person.

5934	Required:	False
5935	Inherited:	False
5936	Property Type:	String
5937	Cardinality:	Single
5938	Updatability:	Read Write
5939		
5940	icom_core:department	
5941	Description:	A person's affiliated department.
5942	Required:	False
5943	Inherited:	False
5944	Property Type:	String
5945	Cardinality:	Single
5946	Updatability:	Read Write
5947		
5948	icom_core:officeLocation	
5949	Description:	Location of a person's department.
5950	Required:	False
5951	Inherited:	False
5952	Property Type:	String
5953	Cardinality:	Single
5954	Updatability:	Read Write
5955		
5956	icom_core:company	
5957	Description:	A person's affiliated company.
5958	Required:	False
5959	Inherited:	False
5960	Property Type:	String
5961	Cardinality:	Single
5962	Updatability:	Read Write
5963		
5964	icom_core:profession	
5965	Description:	A person's profession.
5966	Required:	False
5967	Inherited:	False
5968	Property Type:	String
5969	Cardinality:	Single
5970	Updatability:	Read Write
5971		
5972	icom_content:attachment	
5973	Description:	One or more content attachments in a contact.
5974	Required:	False
5975	Inherited:	False

5976	Property Type:	icom_content:AttachedItem
5977	Cardinality:	Multi
5978	Updatability:	Read Write
5979		
5980	icom_card:bookmark	
5981	Description:	A person which is bookmarked by a contact.
5982	Required:	False
5983	Inherited:	False
5984	Property Type:	icom_core:Person
5985	Cardinality:	Single
5986	Updatability:	On Create
5987		
5988	The PersonContact class MAY include additional property definitions which are implementation-defined.	
5989		



5990
5991 *Figure 35: Person Contact Class Diagram.*

5992

5993 **4.7 Calendar Module**

5994 **4.7.1 Calendar**

5995 **4.7.1.1 Description**

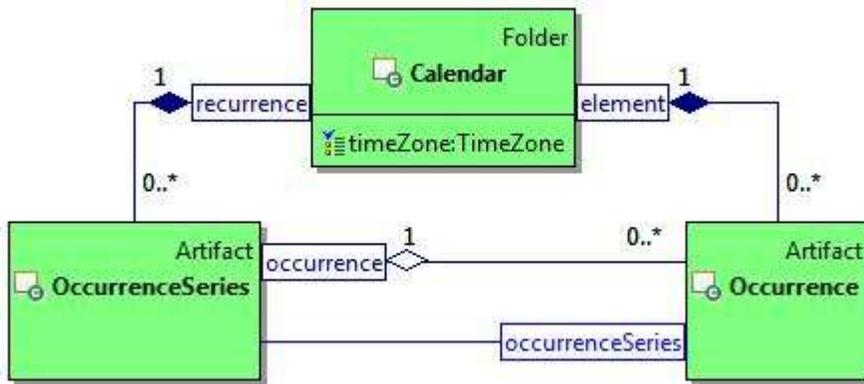
5996 A calendar contains time management artifacts that include occurrences and occurrence series.

5997 **4.7.1.2 Class Definition**

5998 The Calendar class has attribute values:

5999

6042 Property Type: icom_cal:OccurrenceSeries
 6043 Cardinality: Multi
 6044 Updatability: Read Only
 6045



6046
 6047 *Figure 36: Calendar Class Diagram.*
 6048

6049 **4.7.2 OccurrenceSeries**

6050 **4.7.2.1 Description**

6051 An occurrence series represents a series of occurrences associated with the same calendar event.

6052 **4.7.2.2 Class Definition**

6053 The OccurrenceSeries class has attribute values:

6054

6055 **localNamespace**
 6056 Value: icom_cal

6057

6058 **localName**
 6059 Value: OccurrenceSeries

6060

6061 **extendsFrom**
 6062 Value: icom_core:Artifact

6063

6064 **stereotype**
 6065 Value: primary

6066

6067 **description**
 6068 Value: An occurrence series represents a series of occurrences associated with the same
 6069 calendar event.

6070

6071 **propertyDefinitions**
 6072 The values for this attribute are defined in 4.7.2.3.

6073 **4.7.2.3 Property Definitions**

6074 The OccurrenceSeries class inherits property definitions from super classes.

6075 The OccurrenceSeries class MUST have the property definitions:

6076

6077 **icom_core:location**

6078 Description: Location of an occurrence series.

6079 Required: False

6080 Inherited: False

6081 Property Type: icom_core:Location

6082 Cardinality: Single

6083 Updatability: Read Write

6084

6085 **icom_core:organizer**

6086 Description: Organizer of an occurrence series.

6087 Required: True

6088 Inherited: False

6089 Property Type: icom_core:Participant

6090 Cardinality: Single

6091 Updatability: On Create

6092

6093 **icom_core:participant**

6094 Description: Participants in an occurrence series.

6095 Required: False

6096 Inherited: False

6097 Property Type: icom_cal:OccurrenceParticipant

6098 Cardinality: Multi

6099 Updatability: Read Write

6100

6101 **icom_core:priority**

6102 Description: Priority for an attendee of an occurrence series.

6103 Required: False

6104 Inherited: False

6105 Property Type: icom_core:Priority

6106 Cardinality: Single

6107 Updatability: Read Write

6108

6109 **icom_content:attachment**

6110 Description: One or more content attachments in an occurrence series.

6111 Required: False

6112 Inherited: False

6113 Property Type: icom_content:AttachedItem

6114 Cardinality: Multi

6115	Updatability:	Read Write
6116		
6117	icom_cal:recurrenceStartDate	
6118	Description:	Start date and time of an occurrence series.
6119	Required:	True
6120	Inherited:	False
6121	Property Type:	DateTime
6122	Cardinality:	Single
6123	Updatability:	On Create
6124		
6125	icom_cal:recurrenceStartDateResolution	
6126	Description:	Resolution of start date and time of an occurrence series.
6127	Required:	True
6128	Inherited:	False
6129	Property Type:	icom_core:DateTimeResolution
6130	Cardinality:	Single
6131	Updatability:	On Create
6132		
6133	icom_cal:duration	
6134	Description:	Duration of each occurrence in an occurrence series.
6135	Required:	True
6136	Inherited:	False
6137	Property Type:	Duration
6138	Cardinality:	Single
6139	Updatability:	On Create
6140		
6141	icom_cal:recurrenceRule	
6142	Description:	A recurrence rule of an occurrence series.
6143	Required:	True
6144	Inherited:	False
6145	Property Type:	String
6146	Cardinality:	Single
6147	Updatability:	On Create
6148		
6149	icom_cal:occurrenceStatus	
6150	Description:	Status of an occurrence series.
6151	Required:	True
6152	Inherited:	False
6153	Property Type:	icom_cal:OccurrenceStatus
6154	Cardinality:	Single
6155	Updatability:	Read Write
6156		

6157	icom_cal:occurrenceType	
6158	Description:	Type of an occurrence series.
6159	Required:	True
6160	Inherited:	False
6161	Property Type:	icom_cal:OccurrenceType
6162	Cardinality:	Single
6163	Updatability:	Read Write
6164		
6165	icom_cal:editMode	
6166	Description:	Indicates a mode which determines whether an occurrence series is editable.
6167		
6168	Required:	False
6169	Inherited:	False
6170	Property Type:	icom_cal:OccurrenceEditMode
6171	Cardinality:	Single
6172	Updatability:	Read Only
6173		
6174	icom_cal:occurrence	
6175	Description:	Occurrences in an occurrence series.
6176	Required:	False
6177	Inherited:	False
6178	Property Type:	icom_cal:Occurrence
6179	Cardinality:	Multi
6180	Updatability:	Read Only
6181		
6182	icom_cal:attendee	
6183	Description:	An attendee of an occurrence series.
6184	Required:	False
6185	Inherited:	False
6186	Property Type:	icom_core:Participant
6187	Cardinality:	Single
6188	Updatability:	Read Only
6189		
6190	icom_cal:attendeeParticipantStatus	
6191	Description:	Participation status for an attendee of an occurrence series.
6192	Required:	False
6193	Inherited:	False
6194	Property Type:	icom_cal:OccurrenceParticipantStatus
6195	Cardinality:	Single
6196	Updatability:	Read Write
6197		

6198	icom_cal:transparency	
6199	Description:	Participant transparency for an attendee of an occurrence series.
6200		
6201	Required:	False
6202	Inherited:	False
6203	Property Type:	icom_cal:OccurrenceParticipantTransparency
6204	Cardinality:	Single
6205	Updatability:	Read Write
6206		
6207	icom_cal:attendeeProperty	
6208	Description:	Extensible properties for an attendee of an occurrence series.
6209	Required:	False
6210	Inherited:	False
6211	Property Type:	icom_meta:Property
6212	Cardinality:	Multi
6213	Updatability:	Read Write
6214		
6215	icom_conf:conference	
6216	Description:	One or more conferences for an occurrence series.
6217	Required:	False
6218	Inherited:	False
6219	Property Type:	icom_conf:Conference
6220	Cardinality:	Multi
6221	Updatability:	Read Write
6222		

6237
6238 **extendsFrom**
6239 Value: icom_core:Artifact

6240
6241 **stereotype**
6242 Value: primary

6243
6244 **description**
6245 Value: An occurrence represents an event in a calendar.

6246
6247 **propertyDefinitions**
6248 The values for this attribute are defined in 4.7.3.3.
6249

6250 **4.7.3.3 Property Definitions**

6251 The Occurrence class inherits property definitions from super classes.
6252 The Occurrence class **MUST** have the property definitions:

6253
6254 **icom_core:location**
6255 Description: Location of an occurrence.
6256 Required: False
6257 Inherited: False
6258 Property Type: icom_core:Location
6259 Cardinality: Single
6260 Updatability: Read Write

6261
6262 **icom_core:organizer**
6263 Description: Organizer of an occurrence.
6264 Required: True
6265 Inherited: False
6266 Property Type: icom_core:Participant
6267 Cardinality: Single
6268 Updatability: On Create

6269
6270 **icom_core:participant**
6271 Description: Participants of an occurrence.
6272 Required: False
6273 Inherited: False
6274 Property Type: icom_cal:OccurrenceParticipant
6275 Cardinality: Multi
6276 Updatability: Read Write

6277

6278	icom_core:priority	
6279	Description:	Priority for an attendee of an occurrence.
6280	Required:	False
6281	Inherited:	False
6282	Property Type:	icom_core:Priority
6283	Cardinality:	Single
6284	Updatability:	Read Write
6285		
6286	icom_core:startDate	
6287	Description:	Start date and time of an occurrence.
6288	Required:	True
6289	Inherited:	False
6290	Property Type:	DateTime
6291	Cardinality:	Single
6292	Updatability:	On Create
6293		
6294	icom_core:startDateResolution	
6295	Description:	Resolution of start date and time of an occurrence.
6296	Required:	True
6297	Inherited:	False
6298	Property Type:	icom_core:DateTimeResolution
6299	Cardinality:	Single
6300	Updatability:	On Create
6301		
6302	icom_core:endDate	
6303	Description:	End date and time of an occurrence.
6304	Required:	True
6305	Inherited:	False
6306	Property Type:	DateTime
6307	Cardinality:	Single
6308	Updatability:	On Create
6309		
6310	icom_core:endDateResolution	
6311	Description:	Resolution of end date and time of an occurrence.
6312	Required:	True
6313	Inherited:	False
6314	Property Type:	icom_core:DateTimeResolution
6315	Cardinality:	Single
6316	Updatability:	On Create
6317		
6318	icom_content:attachment	
6319	Description:	One or more content attachments in an occurrence.

6320	Required:	False
6321	Inherited:	False
6322	Property Type:	icom_content:AttachedItem
6323	Cardinality:	Multi
6324	Updatability:	Read Write
6325		
6326	icom_cal:occurrenceSeries	
6327	Description:	An occurrence is part of this occurrence series.
6328	Required:	False
6329	Inherited:	False
6330	Property Type:	icom_cal:OccurrenceSeries
6331	Cardinality:	Single
6332	Updatability:	Read Only
6333		
6334	icom_cal:fromRecurringOccurrenceSeries	
6335	Description:	Occurrence is part of a recurring occurrence series.
6336	Required:	False
6337	Inherited:	False
6338	Property Type:	Boolean
6339	Cardinality:	Single
6340	Updatability:	Read Only
6341		
6342	icom_cal:exceptionToOccurrenceSeries	
6343	Description:	Occurrence is an exception to an occurrence series.
6344	Required:	False
6345	Inherited:	False
6346	Property Type:	Boolean
6347	Cardinality:	Single
6348	Updatability:	Read Only
6349		
6350	icom_cal:occurrenceStatus	
6351	Description:	Status of an occurrence.
6352	Required:	True
6353	Inherited:	False
6354	Property Type:	icom_cal:OccurrenceStatus
6355	Cardinality:	Single
6356	Updatability:	Read Write
6357		
6358	icom_cal:occurrenceType	
6359	Description:	Type of an occurrence.
6360	Required:	True
6361	Inherited:	False

6362	Property Type:	icom_cal:OccurrenceType
6363	Cardinality:	Single
6364	Updatability:	Read Write
6365		
6366	icom_cal:editMode	
6367	Description:	Indicates a mode which determines whether an occurrence is
6368		editable.
6369	Required:	False
6370	Inherited:	False
6371	Property Type:	icom_cal:OccurrenceEditMode
6372	Cardinality:	Single
6373	Updatability:	Read Only
6374		
6375	icom_cal:attendee	
6376	Description:	An attendee of an occurrence.
6377	Required:	False
6378	Inherited:	False
6379	Property Type:	icom_core:Participant
6380	Cardinality:	Single
6381	Updatability:	Read Only
6382		
6383	icom_cal:attendeeParticipantStatus	
6384	Description:	Status for an attendee of an occurrence.
6385	Required:	False
6386	Inherited:	False
6387	Property Type:	icom_cal:OccurrenceParticipantStatus
6388	Cardinality:	Single
6389	Updatability:	Read Write
6390		
6391	icom_cal:transparency	
6392	Description:	Transparency for an attendee of an occurrence.
6393	Required:	False
6394	Inherited:	False
6395	Property Type:	icom_cal:OccurrenceParticipantTransparency
6396	Cardinality:	Single
6397	Updatability:	Read Write
6398		
6399	icom_cal:attendeeProperty	
6400	Description:	Extensible properties for an attendee of an occurrence.
6401	Required:	False
6402	Inherited:	False
6403	Property Type:	icom_meta:Property
6404	Cardinality:	Multi

6418 **4.7.4 OccurrenceStatus**

6419 **4.7.4.1 Description**

6420 An occurrence status is a status of a calendar occurrence.

6421 **4.7.4.2 Class Definition**

6422 The OccurrenceStatus class is a mixin class which defines status of a calendar occurrence.

6423 The OccurrenceStatus class has attribute values:

6424

6425 **localNamespace**

6426 Value: icom_cal

6427

6428 **localName**

6429 Value: OccurrenceStatus

6430

6431 **extendsFrom**

6432 Value:

6433

6434 **stereotype**

6435 Value: mixin

6436

6437 **description**

6438 Value: OccurrenceStatus is a mixin class which defines status of a calendar occurrence.

6439

6440 **propertyDefinitions**

6441 The values for this attribute are defined in Section 4.7.4.3.

6442 **4.7.4.3 Property Definitions**

6443 The OccurrenceStatus class MAY include additional property definitions which are implementation-
6444 defined.

6445

6446 **4.7.5 OccurrenceStatusEnum**

6447 The OccurrenceStatusEnum class is an enum class that enumerates the instances each of which
6448 expresses a status of an occurrence or occurrence series.

6449 The OccurrenceStatusEnum class has attribute values:

6450

6451 **localNamespace**

6452 Value: icom_cal

6453

6454 **localName**

6455 Value: OccurrenceStatusEnum

6456

6457 **extendsFrom**
6458 Value: icom_cal:OccurrenceStatus
6459
6460 **stereotype**
6461 Value: primary
6462
6463 **isEnumeration**
6464 Value: TRUE
6465
6466 **description**
6467 Value: Status of an occurrence or occurrence series.
6468
6469 **instances**
6470 Value: <icom_cal:Cancelled, icom_cal:Tentative, icom_cal:Confirmed>
6471
6472 ICOM defines three occurrence status:
6473 • **icom_cal:Cancelled** an occurrence or occurrence series is cancelled.
6474 • **icom_cal:Tentative** an occurrence or occurrence series is tentative.
6475 • **icom_cal:Confirmed** an occurrence or occurrence series is confirmed.
6476

6477 **4.7.6 OccurrenceType**

6478 **4.7.6.1 Description**

6479 An occurrence type is a category of calendar occurrences.

6480 **4.7.6.2 Class Definition**

6481 The OccurrenceType class is a mixin class which defines a type of occurrence.

6482 The OccurrenceType class has attribute values:

6483
6484 **localNamespace**
6485 Value: icom_cal
6486
6487 **localName**
6488 Value: OccurrenceType
6489
6490 **extendsFrom**
6491 Value:
6492
6493 **stereotype**
6494 Value: mixin
6495

6496 **description**
6497 Value: OccurrenceType is a mixin class which defines a type of occurrence.

6498
6499 **propertyDefinitions**
6500 The values for this attribute are defined in Section 4.7.6.3.

6501 **4.7.6.3 Property Definitions**

6502 The OccurrenceType class MAY include additional property definitions which are implementation-defined.
6503

6504 **4.7.7 OccurrenceTypeEnum**

6505 The OccurrenceTypeEnum class is an enum class that enumerates the instances each of which
6506 expresses a type of an occurrence or occurrence series.

6507 The OccurrenceTypeEnum class has attribute values:

6508
6509 **localNamespace**
6510 Value: icom_cal
6511

6512 **localName**
6513 Value: OccurrenceTypeEnum
6514

6515 **extendsFrom**
6516 Value: icom_cal:OccurrenceType
6517

6518 **stereotype**
6519 Value: primary
6520

6521 **isEnumeration**
6522 Value: TRUE
6523

6524 **description**
6525 Value: Type of an occurrence or occurrence series.
6526

6527 **instances**
6528 Value: <icom_cal:Meeting, icom_cal:DayEvent, icom_cal:Holiday, icom_cal:JournalEntry,
6529 icom_cal:OtherOccurrenceType>
6530

6531 ICOM defines five occurrence types:

- 6532 • **icom_cal:Meeting** an occurrence or occurrence series is a meeting.
- 6533 • **icom_cal:DayEvent** an occurrence or occurrence series is a day event.
- 6534 • **icom_cal:Holiday** an occurrence or occurrence series is a holiday.
- 6535 • **icom_cal:JournalEntry** an occurrence or occurrence series is a journal entry.
- 6536 • **icom_cal:OtherOccurrenceType** an occurrence or occurrence series is of other type.

6537

6538 **4.7.8 OccurrenceParticipant**

6539 **4.7.8.1 Description**

6540 An occurrence participant object is a participant object that contains an occurrence participant status.

6541 **4.7.8.2 Class Definition**

6542 The OccurrenceParticipant class has attribute values:

6543

6544 **localNamespace**

Value: icom_cal

6546

6547 **localName**

Value: OccurrenceParticipant

6549

6550 **extendsFrom**

Value: icom_core:Participant

6552

6553 **stereotype**

Value: primary

6555

6556 **description**

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

6557

6558

6559

6560 **propertyDefinitions**

The values for this attribute are defined in Section 4.7.8.3.

6561

6562 **4.7.8.3 Property Definitions**

6563 The OccurrenceParticipant class inherits property definitions from super classes.

6564 The OccurrenceParticipant class MUST have the property definition:

6565

6566 **icom_cal:participantStatus**

6567 Description:

Status of an occurrence participant.

6568 Required:

False

6569 Inherited:

False

6570 Property Type:

icom_cal:OccurrenceParticipantStatus

6571 Cardinality:

Single

6572 Updatability:

Read Write

6573

6574 **4.7.9 OccurrenceParticipantStatus**

6575 **4.7.9.1 Description**

6576 An occurrence participant status is a participant's response status for an occurrence or occurrence series.

6577 **4.7.9.2 Class Definition**

6578 The OccurrenceParticipantStatus class is a mixin class which defines a participant's response status for
6579 an occurrence or occurrence series.

6580 The OccurrenceParticipantStatus class has attribute values:

6581

6582 **localNamespace**

6583 Value: icom_cal

6584

6585 **localName**

6586 Value: OccurrenceParticipantStatus

6587

6588 **extendsFrom**

6589 Value:

6590

6591 **stereotype**

6592 Value: mixin

6593

6594 **description**

6595 Value: OccurrenceParticipantStatus is a mixin class which defines a participant's response
6596 status for an occurrence or occurrence series.

6597

6598 **propertyDefinitions**

6599 The values for this attribute are defined in Section 4.7.9.3.

6600 **4.7.9.3 Property Definitions**

6601 The OccurrenceParticipantStatus class MAY include additional property definitions which are
6602 implementation-defined.

6603

6604 **4.7.10 OccurrenceParticipantStatusEnum**

6605 The OccurrenceParticipantStatusEnum class is an enum class that enumerates the instances each of
6606 which expresses a participant's response status for an occurrence or occurrence series.

6607 The OccurrenceParticipantStatusEnum class has attribute values:

6608

6609 **localNamespace**

6610 Value: icom_cal

6611

6612 **localName**

6613 Value: OccurrenceParticipantStatusEnum

6614
6615 **extendsFrom**
6616 Value: icom_cal:OccurrenceParticipantStatus
6617
6618 **stereotype**
6619 Value: primary
6620
6621 **isEnumeration**
6622 Value: TRUE
6623
6624 **description**
6625 Value: Participant's response status for an occurrence or occurrence series.
6626
6627 **instances**
6628 Value: <icom_cal:NeedsAction, icom_cal:Accepted, icom_cal:Declined, icom_cal:Tentative>
6629

6630 ICOM defines four occurrence participant's status:

- 6631 • **icom_cal:NeedsAction** an attendee needs to act on an occurrence or occurrence series.
 - 6632 • **icom_cal:Accepted** an attendee accepted an occurrence or occurrence series.
 - 6633 • **icom_cal:Declined** an attendee declined an occurrence or occurrence series.
 - 6634 • **icom_cal:Tentative** an attendee is tentative about attending an occurrence or occurrence series.
- 6635

6636 **4.7.11 OccurrenceParticipantTransparency**

6637 **4.7.11.1 Description**

6638 An occurrence participant transparency is visibility of an occurrence or occurrence series in a participant's
6639 calendar or free busy.

6640 **4.7.11.2 Class Definition**

6641 The OccurrenceParticipantTransparency class is a mixin class which defines visibility of an occurrence or
6642 occurrence series in a participant's calendar or free busy.

6643 The OccurrenceParticipantTransparency class has attribute values:

6644
6645 **localNamespace**
6646 Value: icom_cal
6647
6648 **localName**
6649 Value: OccurrenceParticipantTransparency
6650
6651 **extendsFrom**
6652 Value:
6653

6654 **stereotype**
6655 Value: mixin
6656
6657 **description**
6658 Value: OccurrenceParticipantTransparency is a mixin class which defines visibility of an
6659 occurrence or occurrence series in a participant's calendar or free busy.
6660
6661 **propertyDefinitions**
6662 The values for this attribute are defined in Section 4.7.11.3.

6663 **4.7.11.3 Property Definitions**

6664 The OccurrenceParticipantTransparency class MAY include additional property definitions which are
6665 implementation-defined.
6666

6667 **4.7.12 OccurrenceParticipantTransparencyEnum**

6668 The OccurrenceParticipantTransparencyEnum class is an enum class that enumerates the instances
6669 each of which expresses an occurrence or occurrence series transparency in a participant's calendar or
6670 free busy.

6671 The OccurrenceParticipantTransparencyEnum class has attribute values:

6672
6673 **localNamespace**
6674 Value: icom_cal
6675
6676 **localName**
6677 Value: OccurrenceParticipantTransparencyEnum
6678
6679 **extendsFrom**
6680 Value: icom_cal:OccurrenceParticipantTransparency
6681
6682 **stereotype**
6683 Value: primary
6684
6685 **isEnumeration**
6686 Value: TRUE
6687
6688 **description**
6689 Value: Occurrence or occurrence series transparency in a participant's calendar or free busy.
6690
6691 **instances**
6692 Value: <icom_cal:Opaque, icom_cal:Transparent, icom_cal:Tentative, icom_cal:OutOfOffice,
6693 icom_cal:DefaultTransparency>
6694

6695 ICOM defines five participant transparencies:

- 6696 • **icom_cal:Opaque** an occurrence or occurrence series is opaque in a participant's calendar or
6697 free busy.
- 6698 • **icom_cal:Transparent** an occurrence or occurrence series is transparent in a participant's
6699 calendar or free busy.
- 6700 • **icom_cal:Tentative** an occurrence or occurrence series has a tentative transparency in a
6701 participant's calendar or free busy.
- 6702 • **icom_cal:OutOfOffice** an occurrence or occurrence series has out of office transparency in a
6703 participant's calendar or free busy.
- 6704 • **icom_cal:DefaultTransparency** an occurrence or occurrence series has default transparency in
6705 a participant's calendar or free busy.

6706

6707 **4.7.13 OccurrenceEditMode**

6708 **4.7.13.1 Description**

6709 An occurrence edit mode is a mode that indicates whether an occurrence or occurrence series is editable.

6710 **4.7.13.2 Class Definition**

6711 The OccurrenceEditMode class is a mixin class which defines a mode that indicates whether an
6712 occurrence or occurrence series is editable.

6713 The OccurrenceEditMode class has attribute values:

6714

6715 **localNamespace**

6716 Value: icom_cal

6717

6718 **localName**

6719 Value: OccurrenceEditMode

6720

6721 **extendsFrom**

6722 Value:

6723

6724 **stereotype**

6725 Value: mixin

6726

6727 **description**

6728 Value: OccurrenceEditMode is a mixin class which defines a mode that indicates whether an
6729 occurrence or occurrence series is editable.

6730

6731 **propertyDefinitions**

6732 The values for this attribute are defined in Section 4.7.13.3.

6733 **4.7.13.3 Property Definitions**

6734 The OccurrenceEditMode class MAY include additional property definitions which are implementation-
6735 defined.

6736

6737 4.7.14 OccurrenceEditModeEnum

6738 The OccurrenceEditModeEnum class is an enum class that enumerates the instances each of which
6739 expresses a mode that indicates whether an occurrence or occurrence series is editable.

6740 The OccurrenceEditModeEnum class has attribute values:

6741

6742 **localNamespace**

6743 Value: icom_cal

6744

6745 **localName**

6746 Value: OccurrenceEditModeEnum

6747

6748 **extendsFrom**

6749 Value: icom_cal:OccurrenceEditMode

6750

6751 **stereotype**

6752 Value: primary

6753

6754 **isEnumeration**

6755 Value: TRUE

6756

6757 **description**

6758 Value: A mode that indicates whether an occurrence or occurrence series is editable.

6759

6760 **instances**

6761 Value: <icom_cal:OrganizerCopy, icom_cal:AttendeeCopy>

6762

6763 ICOM defines two occurrence editable modes:

- 6764 • **icom_cal:OrganizerCopy** an occurrence or occurrence series is a copy created by an organizer
6765 who may update the properties such as occurrence type, occurrence status, etc.
- 6766 • **icom_cal:AttendeeCopy** an occurrence or occurrence series is a copy delivered to an attendee
6767 who may only update the attendee properties such as priority, transparency, etc .

6768

6769 4.8 Free Busy Module

6770 4.8.1 FreeBusy

6771 4.8.1.1 Description

6772 A free busy object specifies the free time and busy time intervals of one or more participants.

6773 **4.8.1.2 Class Definition**

6774 The FreeBusy class has attribute values:

6775

6776 **localNamespace**

6777 Value: icom_cal

6778

6779 **localName**

6780 Value: FreeBusy

6781

6782 **extendsFrom**

6783 Value:

6784

6785 **stereotype**

6786 Value: primary

6787

6788 **description**

6789 Value: A free busy object specifies the free time and busy time intervals of one or more
6790 participants.

6791

6792 **propertyDefinitions**

6793 The values for this attribute are defined in Section 4.8.1.3.

6794 **4.8.1.3 Property Definitions**

6795 The FreeBusy class MUST have the property definitions:

6796

6797 **icom_core:participant**

6798 Description: A list of participants whose free busy intervals are included.

6799 Required: False

6800 Inherited: False

6801 Property Type: icom_core:Participant

6802 Cardinality: Multi

6803 Updatability: Read Only

6804

6805 **icom_core:creationDate**

6806 Description: Creation date and time of a free busy object.

6807 Required: False

6808 Inherited: False

6809 Property Type: DateTime

6810 Cardinality: Single

6811 Updatability: Read Only

6812

6813 **icom_core:startDate**

6814 Description: Start date and time of a list of free busy intervals.

6815	Required:	False
6816	Inherited:	False
6817	Property Type:	DateTime
6818	Cardinality:	Single
6819	Updatability:	Read Only
6820		
6821	icom_core:endDate	
6822	Description:	End date and time of a list of free busy intervals.
6823	Required:	False
6824	Inherited:	False
6825	Property Type:	DateTime
6826	Cardinality:	Single
6827	Updatability:	Read Only
6828		
6829	icom_cal:interval	
6830	Description:	A list of free busy intervals.
6831	Required:	False
6832	Inherited:	False
6833	Property Type:	icom_cal:FreeBusyInterval
6834	Cardinality:	Multi
6835	Updatability:	Read Only
6836		

6837 **4.8.2 FreeBusyInterval**

6838 **4.8.2.1 Description**

6839 A free busy interval specifies an interval of free or busy time.

6840 If a free busy type is icom_cal:Free, then a time interval is free for scheduling.

6841 If a free busy type is icom_cal:Busy, then a time interval is busy because one or more events have been
6842 scheduled for the interval.

6843 **4.8.2.2 Class Definition**

6844 The FreeBusyInterval class has attribute values:

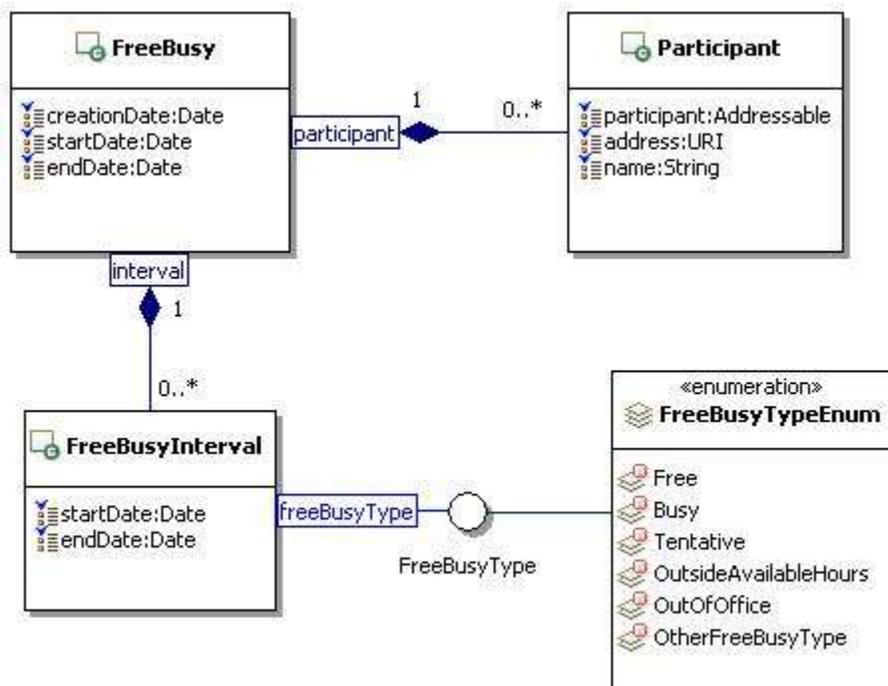
6845		
6846	localNamespace	
6847	Value:	icom_cal
6848		
6849	localName	
6850	Value:	FreeBusyInterval
6851		
6852	extendsFrom	
6853	Value:	
6854		

6855 **stereotype**
6856 Value: primary
6857
6858 **description**
6859 Value: A free busy interval object specifies an interval of free or busy time.
6860
6861 **propertyDefinitions**
6862 The values for this attribute are defined in Section 4.8.2.3

4.8.2.3 Property Definitions

The FreeBusyInterval class MUST have the property definitions:

6866	icom_core:startDate	
6867	Description:	Start date and time of a free busy interval.
6868	Required:	False
6869	Inherited:	False
6870	Property Type:	DateTime
6871	Cardinality:	Single
6872	Updatability:	Read Only
6873		
6874	icom_core:endDate	
6875	Description:	End date and time of a free busy interval.
6876	Required:	False
6877	Inherited:	False
6878	Property Type:	DateTime
6879	Cardinality:	Single
6880	Updatability:	Read Only
6881		
6882	icom_cal:freeBusyType	
6883	Description:	A type of free busy interval.
6884	Required:	False
6885	Inherited:	False
6886	Property Type:	icom_cal:FreeBusyType
6887	Cardinality:	Single
6888	Updatability:	Read Only
6889		



6890
 6891 *Figure 39: Free Busy Class Diagram.*
 6892

6893 4.8.3 FreeBusyType

6894 4.8.3.1 Description

6895 A free busy type classifies a time interval as free, busy, or other.

6896 4.8.3.2 Class Definition

6897 The FreeBusyType class is a mixin class which defines different types to classify a time interval.

6898 The FreeBusyType class has attribute values:

- 6899
- 6900 **localNamespace**
- 6901 Value: icom_cal
- 6902
- 6903 **localName**
- 6904 Value: FreeBusyType
- 6905
- 6906 **extendsFrom**
- 6907 Value:
- 6908
- 6909 **stereotype**
- 6910 Value: mixin
- 6911
- 6912 **description**
- 6913 Value: FreeBusyType is a mixin class which defines different types to classify a time interval.

6914
6915 **propertyDefinitions**
6916 The values for this attribute are defined in Section 4.8.3.3.

6917 **4.8.3.3 Property Definitions**

6918 The FreeBusyType class MAY include additional property definitions which are implementation-defined.
6919

6920 **4.8.4 FreeBusyTypeEnum**

6921 The FreeBusyTypeEnum class is an enum class that enumerates the instances each of which expresses
6922 a type of free busy interval.

6923 The FreeBusyTypeEnum class has attribute values:

6924
6925 **localNamespace**
6926 Value: icom_cal
6927
6928 **localName**
6929 Value: FreeBusyTypeEnum
6930
6931 **extendsFrom**
6932 Value: icom_cal:FreeBusyType
6933
6934 **stereotype**
6935 Value: primary
6936
6937 **isEnumeration**
6938 Value: TRUE
6939
6940 **description**
6941 Value: A type of free busy interval.
6942
6943 **instances**
6944 Value: <icom_cal:Free, icom_cal:Busy, icom_cal:Tentative, icom_cal:OutsideAvailableHours,
6945 icom_cal:OutOfOffice, icom_cal:OtherFreeBusyType>

6947 ICOM defines six free busy types:

- 6948 • **icom_cal:Free** a free busy interval is free.
- 6949 • **icom_cal:Busy** a free busy interval is busy.
- 6950 • **icom_cal:Tentative** a free busy interval is tentative.
- 6951 • **icom_cal:OutsideAvailableHours** a free busy interval is outside available hours.
- 6952 • **icom_cal:OutOfOffice** a free busy interval is within out of office hours.
- 6953 • **icom_cal:OtherFreeBusyType** a free busy interval is of other type.

6954

6955 **4.9 Task List Module**

6956 **4.9.1 TaskList**

6957 **4.9.1.1 Description**

6958 A task list contains task management artifacts.

6959 **4.9.1.2 Class Definition**

6960 The TaskList class has attribute values:

6961

6962 **localNamespace**

6963 Value: icom_task

6964

6965 **localName**

6966 Value: TaskList

6967

6968 **extendsFrom**

6969 Value: icom_core:Folder

6970

6971 **stereotype**

6972 Value: primary

6973

6974 **description**

6975 Value: A task list contains task management artifacts.

6976

6977 **propertyDefinitions**

6978 The values for this attribute are defined in 4.9.1.3.

6979 **4.9.1.3 Property Definitions**

6980 The TaskList class inherits property definitions from super classes.

6981 The TaskList class MUST have the property definitions:

6982

6983 **icom_core:timeZone**

6984 Description: Time zone of a task list.

6985 Required: True

6986 Inherited: False

6987 Property Type: icom_core:TimeZone

6988 Cardinality: Single

6989 Updatability: Read Write

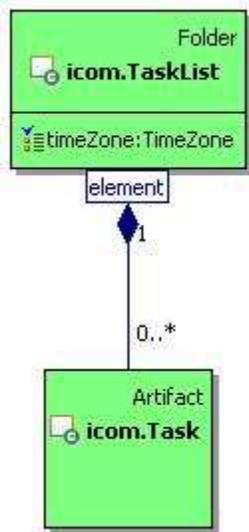
6990

6991 **icom_core:element**

6992 Description: Elements of a task list.

6993 Required: False

6994 Inherited: True
 6995 Property Type: icom_task:Task
 6996 Cardinality: Multi
 6997 Updatability: Read Only
 6998



6999
 7000 *Figure 40: Task List Class Diagram.*

7001

7002 4.9.2 Task

7003 4.9.2.1 Description

7004 A task is an artifact that represents a task to do or a task assignment in a task list.

7005 4.9.2.2 Class Definition

7006 The Task class has attribute values:

7007

7008 **localNamespace**
 7009 Value: icom_task

7010

7011 **localName**
 7012 Value: Task

7013

7014 **extendsFrom**
 7015 Value: icom_core:Artifact

7016

7017 **stereotype**
 7018 Value: primary

7019

7020 **description**
 7021 Value: A task is an artifact that represents a task to do or a task assignment in a task list.

7022

7023 **propertyDefinitions**

7024 The values for this attribute are defined in 4.9.2.3.

7025 **4.9.2.3 Property Definitions**

7026 The Task class inherits property definitions from super classes.

7027 The Task class MUST have the property definitions:

7028

7029 **icom_core:location**

7030 Description: Location of a task.

7031 Required: False

7032 Inherited: False

7033 Property Type: icom_core:Location

7034 Cardinality: Single

7035 Updatability: Read Write

7036

7037 **icom_core:organizer**

7038 Description: Organizer of a task.

7039 Required: True

7040 Inherited: False

7041 Property Type: icom_core:Participant

7042 Cardinality: Single

7043 Updatability: On Create

7044

7045 **icom_core:priority**

7046 Description: Priority of a task.

7047 Required: False

7048 Inherited: False

7049 Property Type: icom_core:Priority

7050 Cardinality: Single

7051 Updatability: Read Write

7052

7053 **icom_core:startDate**

7054 Description: Start date and time of a task.

7055 Required: True

7056 Inherited: False

7057 Property Type: DateTime

7058 Cardinality: Single

7059 Updatability: On Create

7060

7061 **icom_core:startDateResolution**

7062 Description: Resolution of start date and time of a task.

7063 Required: True

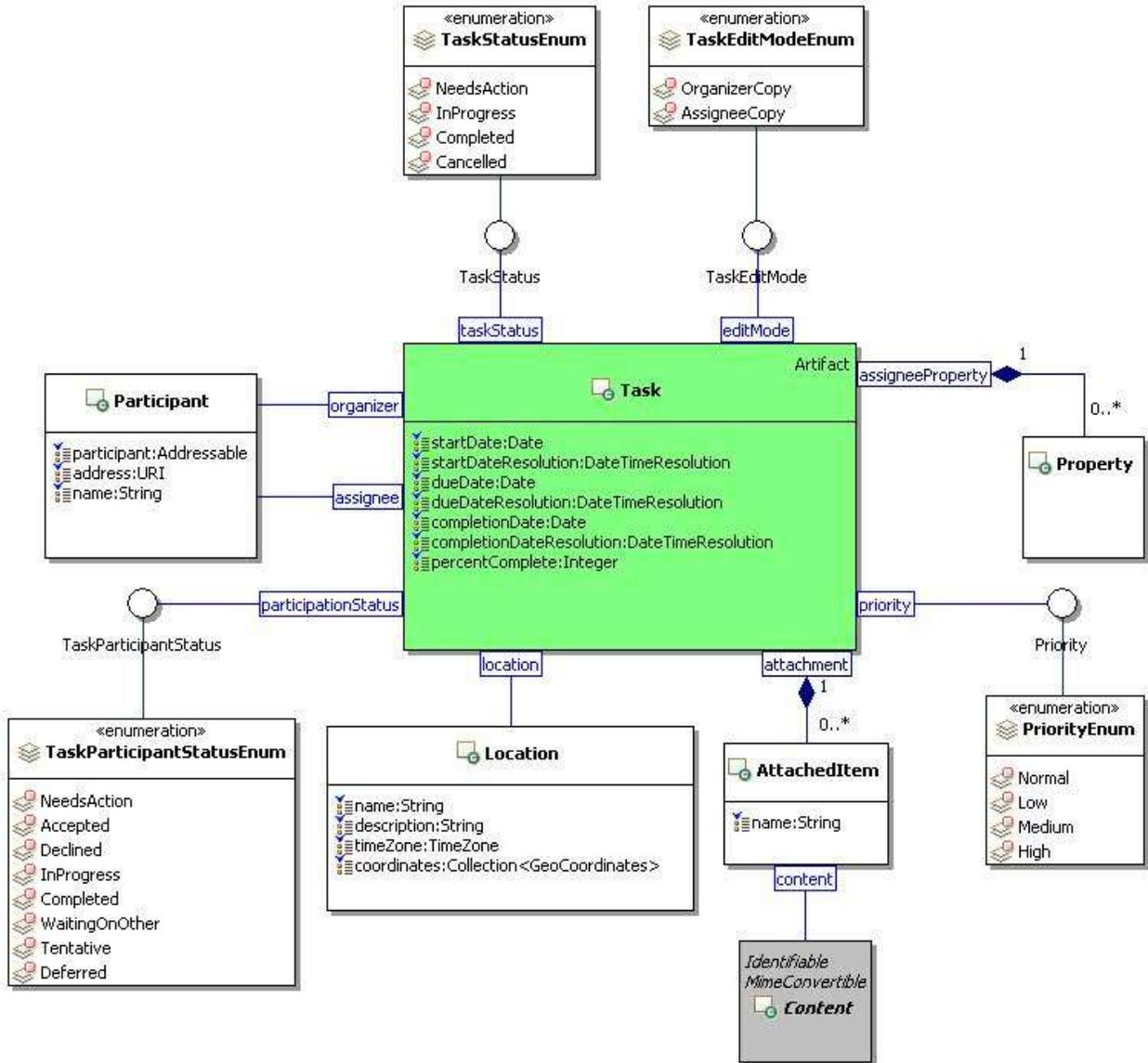
7064	Inherited:	False
7065	Property Type:	icom_core:DateTimeResolution
7066	Cardinality:	Single
7067	Updatability:	On Create
7068		
7069	icom_content:attachment	
7070	Description:	One or more content attachments in a task.
7071	Required:	False
7072	Inherited:	False
7073	Property Type:	icom_content:AttachedItem
7074	Cardinality:	Multi
7075	Updatability:	Read Write
7076		
7077	icom_task:dueDate	
7078	Description:	Due date and time of a task.
7079	Required:	True
7080	Inherited:	False
7081	Property Type:	DateTime
7082	Cardinality:	Single
7083	Updatability:	On Create
7084		
7085	icom_task:dueDateResolution	
7086	Description:	Resolution of due date and time of a task.
7087	Required:	True
7088	Inherited:	False
7089	Property Type:	icom_core:DateTimeResolution
7090	Cardinality:	Single
7091	Updatability:	On Create
7092		
7093	icom_task:editMode	
7094	Description:	Indicates a mode which determines whether a task is
7095		editable.
7096	Required:	False
7097	Inherited:	False
7098	Property Type:	icom_task:TaskEditMode
7099	Cardinality:	Single
7100	Updatability:	Read Only
7101		
7102	icom_task:taskStatus	
7103	Description:	Status of a task.
7104	Required:	True
7105	Inherited:	False
7106	Property Type:	icom_task:TaskStatus

7107	Cardinality:	Single
7108	Updatability:	Read Write
7109		
7110	icom_task:assignee	
7111	Description:	An assignee of a task.
7112	Required:	False
7113	Inherited:	False
7114	Property Type:	icom_core:Participant
7115	Cardinality:	Single
7116	Updatability:	Read Only
7117		
7118	icom_task:participantStatus	
7119	Description:	Participation status of a task.
7120	Required:	False
7121	Inherited:	False
7122	Property Type:	icom_task:TaskParticipantStatus
7123	Cardinality:	Single
7124	Updatability:	Read Write
7125		
7126	icom_task:completionDate	
7127	Description:	Completion date and time of a task.
7128	Required:	False
7129	Inherited:	False
7130	Property Type:	DateTime
7131	Cardinality:	Single
7132	Updatability:	Read Write
7133		
7134	icom_task:completionDateResolution	
7135	Description:	Resolution of completion date and time of a task.
7136	Required:	False
7137	Inherited:	False
7138	Property Type:	icom_core:DateTimeResolution
7139	Cardinality:	Single
7140	Updatability:	Read Write
7141		
7142	icom_task:percentComplete	
7143	Description:	Percentage of task completed.
7144	Required:	False
7145	Inherited:	False
7146	Property Type:	Integer
7147	Cardinality:	Single
7148	Updatability:	Read Write

7149
7150
7151
7152
7153
7154
7155
7156
7157

icom_task:assigneeProperty

Description: Extensible properties for an assignee of a task.
 Required: False
 Inherited: False
 Property Type: icom_meta:Property
 Cardinality: Multi
 Updatability: Read Write



7158
7159 *Figure 41: Task Class Diagram.*
7160

7161 **4.9.3 TaskStatus**

7162 **4.9.3.1 Description**

7163 A task status is a status of a task.

7164 **4.9.3.2 Class Definition**

7165 The TaskStatus class is a mixin class which defines status of a task.

7166 The TaskStatus class has attribute values:

7167

7168 **localNamespace**

7169 Value: icom_task

7170

7171 **localName**

7172 Value: TaskStatus

7173

7174 **extendsFrom**

7175 Value:

7176

7177 **stereotype**

7178 Value: mixin

7179

7180 **description**

7181 Value: TaskStatus is a mixin class which defines status of a task.

7182

7183 **propertyDefinitions**

7184 The values for this attribute are defined in Section 4.9.3.3.

7185 **4.9.3.3 Property Definitions**

7186 The TaskStatus class MAY include additional property definitions which are implementation-defined.

7187

7188 **4.9.4 TaskStatusEnum**

7189 The TaskStatusEnum class is an enum class that enumerates the instances each of which expresses a status of task.

7191 The TaskStatusEnum class has attribute values:

7192

7193 **localNamespace**

7194 Value: icom_task

7195

7196 **localName**

7197 Value: TaskStatusEnum

7198

7199 **extendsFrom**

7200 Value: icom_task:TaskStatus

7201
7202 **stereotype**
7203 Value: primary
7204
7205 **isEnumeration**
7206 Value: TRUE
7207
7208 **description**
7209 Value: Status of a task.
7210
7211 **instances**
7212 Value: <icom_task:NeedsAction, icom_task:InProgress, icom_task:Completed,
7213 icom_task:Cancelled>
7214
7215 ICOM defines four task status:
7216 • **icom_task:NeedsAction** a task needs action.
7217 • **icom_task:InProgress** a task is in progress.
7218 • **icom_task:Completed** a task is completed.
7219 • **icom_task:Cancelled** a task is cancelled.
7220

7221 **4.9.5 TaskParticipantStatus**

7222 **4.9.5.1 Description**

7223 A task participant status is a participant's response status for a task assignment.

7224 **4.9.5.2 Class Definition**

7225 The TaskParticipantStatus class is a mixin class which defines a participant's response status for a task
7226 assignment.

7227 The TaskParticipantStatus class has attribute values:

7228
7229 **localNamespace**
7230 Value: icom_task
7231
7232 **localName**
7233 Value: TaskParticipantStatus
7234
7235 **extendsFrom**
7236 Value:
7237
7238 **stereotype**
7239 Value: mixin
7240

7241 **description**
7242 Value: TaskParticipantStatus is a mixin class which defines a participant's response status for a
7243 task assignment.

7244

7245 **propertyDefinitions**

7246 The values for this attribute are defined in Section 4.9.5.3.

7247 **4.9.5.3 Property Definitions**

7248 The TaskParticipantStatus class MAY include additional property definitions which are implementation-
7249 defined.

7250

7251 **4.9.6 TaskParticipantStatusEnum**

7252 The TaskParticipantStatusEnum class is an enum class that enumerates the instances each of which
7253 expresses a participant's response status for a task.

7254 The TaskParticipantStatusEnum class has attribute values:

7255

7256 **localNamespace**

7257 Value: icom_task

7258

7259 **localName**

7260 Value: TaskParticipantStatusEnum

7261

7262 **extendsFrom**

7263 Value: icom_task:TaskParticipantStatus

7264

7265 **stereotype**

7266 Value: primary

7267

7268 **isEnumeration**

7269 Value: TRUE

7270

7271 **description**

7272 Value: Participant's response status for a task.

7273

7274 **instances**

7275 Value: <icom_task:NeedsAction, icom_task:Accepted, icom_task:Declined,
7276 icom_task:InProgress, icom_task:Completed, icom_task:WaitingOnOther, icom_task:Tentative,
7277 icom_task:Deferred>

7278

7279 ICOM defines eight task participant's status:

7280 • **icom_task:NeedsAction** an assignee needs to act on a task.

7281 • **icom_task:Accepted** an assignee accepted a task.

7282 • **icom_task:Declined** an assignee declined a task.

- 7283 • **icom_task:InProgress** a task is in progress.
- 7284 • **icom_task:Completed** a task is completed.
- 7285 • **icom_task:WaitingOnOther** an assignee is waiting on other.
- 7286 • **icom_task:Tentative** an assignee is tentative about a task.
- 7287 • **icom_task:Deferred** an assignee deferred a task.

7288

7289 **4.9.7 TaskEditMode**

7290 **4.9.7.1 Description**

7291 A task edit mode is a mode that indicates whether a task is editable.

7292 **4.9.7.2 Class Definition**

7293 The TaskEditMode class is a mixin class which defines a mode that indicates whether a task is editable.

7294 The TaskEditMode class has attribute values:

7295

7296 **localNamespace**

7297 Value: icom_task

7298

7299 **localName**

7300 Value: TaskEditMode

7301

7302 **extendsFrom**

7303 Value:

7304

7305 **stereotype**

7306 Value: mixin

7307

7308 **description**

7309 Value: TaskEditMode is a mixin class which defines a mode that indicates whether task is
7310 editable.

7311

7312 **propertyDefinitions**

7313 The values for this attribute are defined in Section 4.9.7.3.

7314 **4.9.7.3 Property Definitions**

7315 The TaskEditMode class MAY include additional property definitions which are implementation-defined.

7316

7317 **4.9.8 TaskEditModeEnum**

7318 The TaskEditModeEnum class is an enum class that enumerates the instances each of which expresses
7319 an editable mode of a task.

7320 The TaskEditModeEnum class has attribute values:

7321

7322 **localNamespace**

7323 Value: icom_task

7324

7325 **localName**

7326 Value: TaskEditModeEnum

7327

7328 **extendsFrom**

7329 Value: icom_task:TaskEditMode

7330

7331 **stereotype**

7332 Value: primary

7333

7334 **isEnumeration**

7335 Value: TRUE

7336

7337 **description**

7338 Value: A mode that indicates whether a task is editable.

7339

7340 **instances**

7341 Value: <icom_task:OrganizerCopy, icom_task:AssigneeCopy>

7342

7343 ICOM defines two task editable modes:

- 7344 • **icom_task:OrganizerCopy**: a task is a copy created by an organizer who may update the
- 7345 properties such as start time, due time.
- 7346 • **icom_task:AssigneeCopy**: a task is a copy delivered to an assignee who may only update the
- 7347 assignee properties such as completion time, participant status, percent completed.

7348

7349 4.10 Forum Module

7350 4.10.1 Discussion

7351 4.10.1.1 Description

7352 A discussion is an item in a discussion container.

7353 4.10.1.2 Class Definition

7354 The Discussion class is a mixin class that defines the characteristics of artifacts that can be elements of

7355 discussion containers.

7356 The Discussion class has attribute values:

7357

7358 **localNamespace**

7359 Value: icom_forum

7360

7361 **localName**

7362 Value: Discussion

7363

7364 **extendsFrom**

7365 Value: icom_core:Item

7366

7367 **stereotype**

7368 Value: mixin

7369

7370 **description**

7371 Value: Discussion is a mixin class that defines the characteristics of artifacts that can be placed

7372 in a discussion container.

7373

7374 **propertyDefinitions**

7375 The values for this attribute are defined in Section 4.10.1.3.

7376 **4.10.1.3 Property Definitions**

7377 The Discussion class inherits property definitions from super classes.

7378 The Discussion class **MUST** have the property definition:

7379

7380 **icom_forum:inReplyTo**

7381 Description:	Another discussion object that a discussion object is replying to.
7382	
7383 Required:	False
7384 Inherited:	False
7385 Property Type:	icom_forum:Discussion
7386 Cardinality:	Single
7387 Updatability:	Read Write

7388

7389 The Discussion class **MAY** include additional property definitions which are implementation-defined.

7390

7391 **4.10.2 DiscussionContainer**

7392 **4.10.2.1 Description**

7393 A discussion container contains discussion items.

7394 **4.10.2.2 Class Definition**

7395 The DiscussionContainer class is a mixin class that defines the characteristics of folders that contain Discussion items.

7396

7397 The DiscussionContainer class has attribute values:

7398

7399 **localNamespace**

7400 Value: icom_forum

7401

7402 **localName**

7403 Value: DiscussionContainer

7404

7405 **extendsFrom**

7406 Value: icom_core:Container

7407

7408 **stereotype**

7409 Value: mixin

7410

7411 **description**

7412 Value: DiscussionContainer is a mixin class that defines the characteristics of folders that

7413 contain Discussion items.

7414

7415 **propertyDefinitions**

7416 The values for this attribute are defined in Section 4.10.2.3.

7417 **4.10.2.3 Property Definitions**

7418 The DiscussionContainer class inherits property definitions from super classes.

7419 The DiscussionContainer class **MUST** have the property definition:

7420

7421 **icom_core:element**

7422 Description:	Elements of a discussion container.
7423 Required:	False
7424 Inherited:	True
7425 Property Type:	icom_forum:Discussion
7426 Cardinality:	Multi
7427 Updatability:	Read Only

7428

7429 The DiscussionContainer class **MAY** include additional property definitions which are implementation-

7430 defined.

7431

7432 **4.10.3 DiscussionMessage**

7433 **4.10.3.1 Description**

7434 A discussion message is a message in a forum discussion thread.

7435 **4.10.3.2 Class Definition**

7436 The DiscussionMessage class has attribute values:

7437

7438 **localNamespace**

7439 Value: icom_forum

7440

7441 **localName**

7442 Value: DiscussionMessage

7443

7444 **extendsFrom**

7445 Value: icom_msg:Message, icom_forum:Discussion

7446

7447 **stereotype**

7448 Value: primary

7449

7450 **description**

7451 Value: Discussion message is a message in a forum discussion thread.

7452

7453 **propertyDefinitions**

7454 The values for this attribute are defined in Section 4.10.3.3.

7455 **4.10.3.3 Property Definitions**

7456 The DiscussionMessage class inherits property definitions from super classes.

7457 The DiscussionMessage class MUST have the property definition:

7458

7459 **icom_forum:inReplyTo**

7460 Description: Another discussion message that a discussion message is
7461 replying to.

7462 Required: False

7463 Inherited: True

7464 Property Type: icom_forum:DiscussionMessage

7465 Cardinality: Single

7466 Updatability: Read Write

7467

7468 The DiscussionMessage class MAY include additional property definitions which are implementation-
7469 defined.

7470

7471 **4.10.4 TopicContainer**

7472 **4.10.4.1 Description**

7473 A topic container contains topics.

7474 **4.10.4.2 Class Definition**

7475 The TopicContainer class is a mixin class which defines the characteristics of folders that contain Topics.

7476 The TopicContainer class has attribute values:

7477

7478 **localNamespace**

7479 Value: icom_forum

7480

7481 **localName**

7482 Value: TopicContainer

7483

7484 **extendsFrom**

7485 Value: icom_core:Container

7486

7487 **stereotype**

7488 Value: mixin

7489

7490 **description**

7491 Value: TopicContainer is a mixin class that defines the characteristics of folders that contain
7492 topics.

7493

7494 **propertyDefinitions**

7495 The values for this attribute are defined in Section 4.10.4.3.

7496 **4.10.4.3 Property Definitions**

7497 The TopicContainer class inherits property definitions from super classes.

7498 The TopicContainer class MUST have the property definitions:

7499

7500 **icom_core:element**

7501 Description: Elements of a topic container.

7502 Required: False

7503 Inherited: True

7504 Property Type: icom_forum:Topic

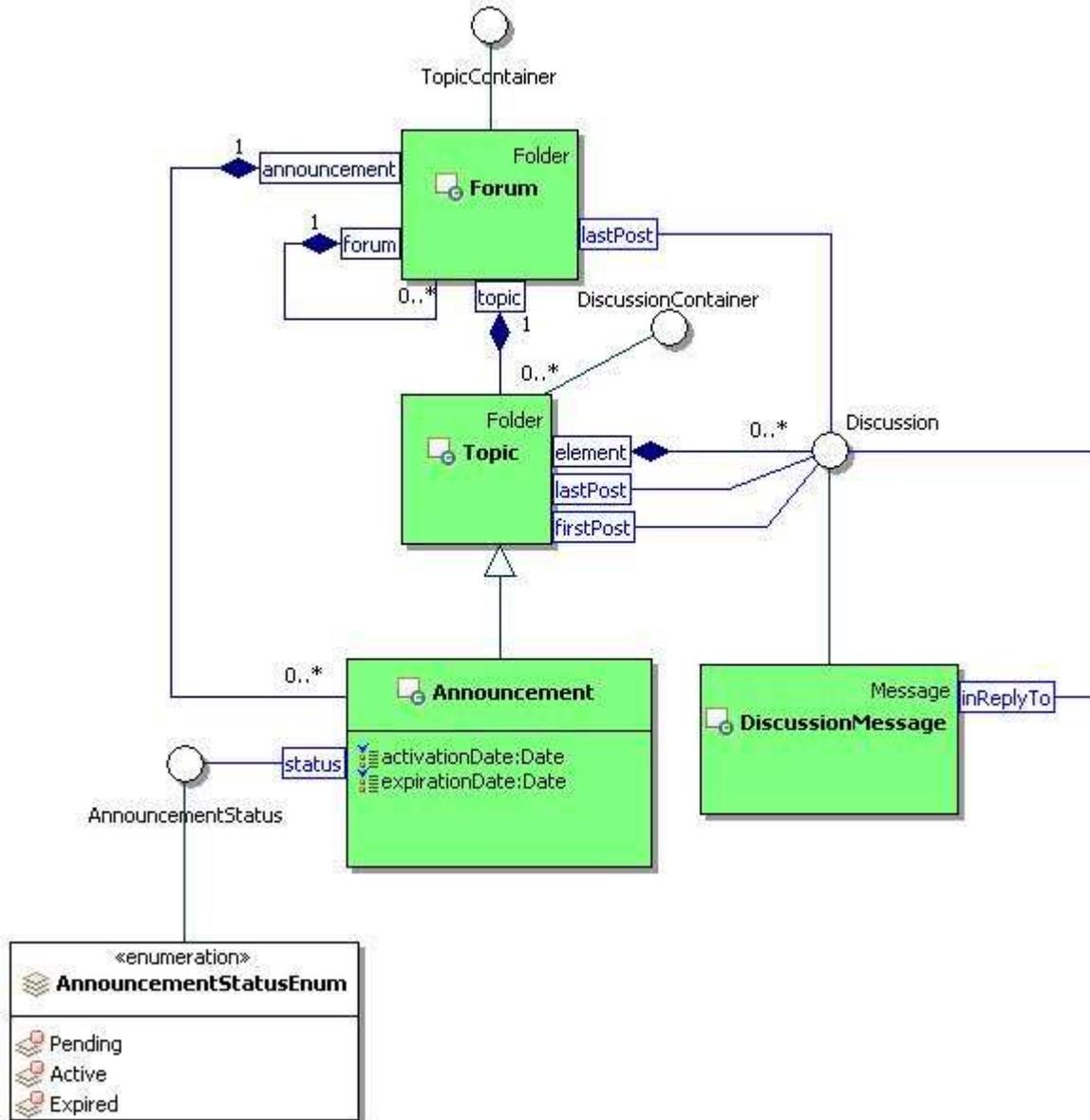
7505 Cardinality: Multi

7506 Updatability: Read Only

7507

7508 The TopicContainer class MAY include additional property definitions which are implementation-defined.

7509



7510
7511 *Figure 42: Forum Class Diagram.*

7512

7513 4.10.5 Forum

7514 4.10.5.1 Description

7515 A forum contains sub-forums, topics, and announcements.

7516 4.10.5.2 Class Definition

7517 The Forum class has attribute values:

7518

7519 `localNamespace`

7520 Value: `icom_forum`

7521

7522 **localName**
7523 Value: Forum
7524
7525 **extendsFrom**
7526 Value: icom_core:Folder, icom_forum:TopicContainer
7527
7528 **stereotype**
7529 Value: primary
7530
7531 **description**
7532 Value: A forum contains sub-forums, topics, and announcements.
7533
7534 **propertyDefinitions**
7535 The values for this attribute are defined in Section 4.10.5.3.

7536 **4.10.5.3 Property Definitions**

7537 The Forum class inherits property definitions from super classes.
7538 The Forum class MUST have the property definitions:

7539
7540 **icom_forum:lastPost**
7541 Description: The last posted discussion in a forum.
7542 Required: False
7543 Inherited: False
7544 Property Type: icom_forum:Discussion
7545 Cardinality: Single
7546 Updatability: Read Only

7547
7548 **icom_forum:forum**
7549 Description: Sub-forums of a forum.
7550 Required: False
7551 Inherited: False
7552 Property Type: icom_forum:Forum
7553 Cardinality: Multi
7554 Updatability: Read Only

7555
7556 **icom_forum:topic**
7557 Description: Topics of a forum.
7558 Required: False
7559 Inherited: False
7560 Property Type: icom_forum:Topic
7561 Cardinality: Multi
7562 Updatability: Read Only

7563

7564	icom_forum:announcement	
7565	Description:	Announcements of a forum.
7566	Required:	False
7567	Inherited:	False
7568	Property Type:	icom_forum:Announcement
7569	Cardinality:	Multi
7570	Updatability:	Read Only

7571

7572 The Forum class MAY include additional property definitions which are implementation-defined.

7573

7574 **4.10.6 Topic**

7575 **4.10.6.1 Description**

7576 A topic contains conversations among forum participants. The discussions in a topic may be sorted in
7577 chronological order or threaded by reply.

7578 **4.10.6.2 Class Definition**

7579 The Topic class has attribute values:

7580		
7581	localNamespace	
7582	Value:	icom_forum
7583		
7584	localName	
7585	Value:	Topic
7586		
7587	extendsFrom	
7588	Value:	icom_core:Folder, icom_forum:DiscussionContainer
7589		
7590	stereotype	
7591	Value:	primary
7592		
7593	description	
7594	Value:	A topic contains discussion threads.
7595		
7596	propertyDefinitions	
7597	The values for this attribute are defined in Section 4.10.6.3.	

7598 **4.10.6.3 Property Definitions**

7599 The Topic class inherits property definitions from super classes.

7600 The Topic class MUST have the property definitions:

7601		
7602	icom_core:element	
7603	Description:	Elements of a topic.

7604	Required:	False
7605	Inherited:	True
7606	Property Type:	icom_forum:Discussion
7607	Cardinality:	Multi
7608	Updatability:	Read Only
7609		
7610	icom_forum:firstPost	
7611	Description:	The first posted discussion in a topic.
7612	Required:	False
7613	Inherited:	False
7614	Property Type:	icom_forum:Discussion
7615	Cardinality:	Single
7616	Updatability:	Read Only

7617		
7618	icom_forum:lastPost	
7619	Description:	The last posted discussion in a topic.
7620	Required:	False
7621	Inherited:	False
7622	Property Type:	icom_forum:Discussion
7623	Cardinality:	Single
7624	Updatability:	Read Only

7625

7626 The Topic class MAY include additional property definitions which are implementation-defined.

7627

7628 **4.10.7 Announcement**

7629 **4.10.7.1 Description**

7630 An announcement contains time-sensitive discussion posts that are valid for a specified period of time,
7631 depending on activation and expiration times.

7632 **4.10.7.2 Class Definition**

7633 The Announcement class has attribute values:

7634		
7635	localNamespace	
7636	Value:	icom_forum
7637		
7638	localName	
7639	Value:	Announcement
7640		
7641	extendsFrom	
7642	Value:	icom_forum:Topic
7643		

7682 **4.10.8 AnnouncementStatus**

7683 **4.10.8.1 Description**

7684 An announcement status is status of an announcement.

7685 **4.10.8.2 Class Definition**

7686 The AnnouncementStatus class is a mixin class which defines status of an announcement.

7687 The AnnouncementStatus class has attribute values:

7688

7689 **localNamespace**

7690 Value: icom_forum

7691

7692 **localName**

7693 Value: AnnouncementStatus

7694

7695 **extendsFrom**

7696 Value:

7697

7698 **stereotype**

7699 Value: mixin

7700

7701 **description**

7702 Value: AnnouncementStatus is a mixin class which defines status of an announcement.

7703

7704 **propertyDefinitions**

7705 The values for this attribute are defined in Section 4.10.8.3.

7706 **4.10.8.3 Property Definitions**

7707 The AnnouncementStatus class MAY include additional property definitions which are implementation-
7708 defined.

7709

7710 **4.10.9 AnnouncementStatusEnum**

7711 The AnnouncementStatusEnum class is an enum class that enumerates the instances each of which
7712 expresses a status of announcement.

7713 The AnnouncementStatusEnum class has attribute values:

7714

7715 **localNamespace**

7716 Value: icom_forum

7717

7718 **localName**

7719 Value: AnnouncementStatusEnum

7720

7721 **extendsFrom**
7722 Value: icom_forum:AnnouncementStatus
7723
7724 **stereotype**
7725 Value: primary
7726
7727 **isEnumeration**
7728 Value: TRUE
7729
7730 **description**
7731 Value: Status of announcement.
7732
7733 **instances**
7734 Value: <icom_forum:Pending, icom_forum:Active, icom_forum:Expired>
7735
7736 ICOM defines three announcement status:
7737 • **icom_forum:Pending** an announcement is pending.
7738 • **icom_forum:Active** an announcement is active.
7739 • **icom_forum:Expired** an announcement is expired.
7740

7741 **4.11 Conference Module**

7742 **4.11.1 Conference**

7743 **4.11.1.1 Description**

7744 A conference is a container that represents a durable context for conference sessions.
7745 It contains conference metadata, settings, and transcripts.

7746 **4.11.1.2 Class Definition**

7747 The Conference class has attribute values:

7748
7749 **localNamespace**
7750 Value: icom_conf
7751
7752 **localName**
7753 Value: Conference
7754
7755 **extendsFrom**
7756 Value: icom_core:Folder
7757
7758 **stereotype**
7759 Value: primary

7760
7761 **description**
7762 Value: A conference represents a durable context for online conference sessions.

7763
7764 **propertyDefinitions**
7765 The values for this attribute are defined in Section 4.11.1.3.

7766 **4.11.1.3 Property Definitions**

7767 The Conference class inherits property definitions from super classes.
7768 The Conference class **MUST** have the property definitions:

7769
7770 **icom_core:organizer**
7771 Description: Organizer of a conference.
7772 Required: False
7773 Inherited: False
7774 Property Type: icom_core:Participant
7775 Cardinality: Single
7776 Updatability: On Create

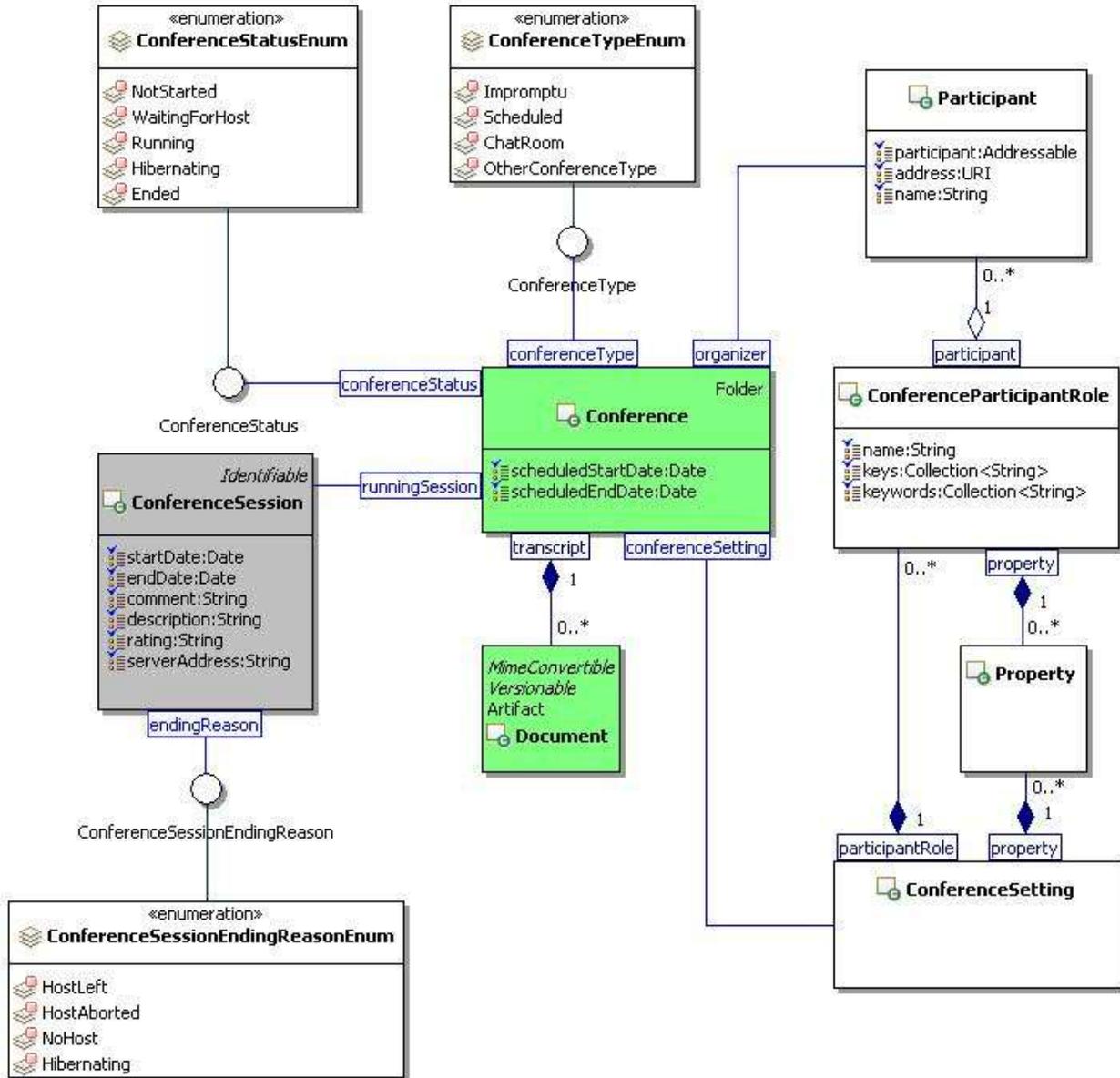
7777
7778 **icom_conf:conferenceType**
7779 Description: Type of a conference.
7780 Required: False
7781 Inherited: False
7782 Property Type: icom_conf:ConferenceType
7783 Cardinality: Single
7784 Updatability: Read Write

7785
7786 **icom_conf:conferenceStatus**
7787 Description: Status of a conference.
7788 Required: False
7789 Inherited: False
7790 Property Type: icom_conf:ConferenceStatus
7791 Cardinality: Single
7792 Updatability: Read Only

7793
7794 **icom_conf:runningSession**
7795 Description: Current session of a conference.
7796 Required: False
7797 Inherited: False
7798 Property Type: icom_conf:ConferenceSession
7799 Cardinality: Single
7800 Updatability: Read Only

7801

7802	icom_conf:conferenceSetting	
7803	Description:	Configurable settings of a conference.
7804	Required:	False
7805	Inherited:	False
7806	Property Type:	icom_conf:ConferenceSetting
7807	Cardinality:	Single
7808	Updatability:	Read Only
7809		
7810	icom_conf:transcript	
7811	Description:	Transcripts from ended sessions of a conference.
7812	Required:	False
7813	Inherited:	False
7814	Property Type:	icom_doc:Document
7815	Cardinality:	Multi
7816	Updatability:	Read Write
7817		
7818	icom_conf:scheduledStartDate	
7819	Description:	Scheduled start date and time of a conference session.
7820	Required:	False
7821	Inherited:	False
7822	Property Type:	DateTime
7823	Cardinality:	Single
7824	Updatability:	Read Write
7825		
7826	icom_conf:scheduledEndDate	
7827	Description:	Scheduled end date and time of a conference session.
7828	Required:	False
7829	Inherited:	False
7830	Property Type:	DateTime
7831	Cardinality:	Single
7832	Updatability:	Read Write
7833		
7834	The Conference class MAY include additional property definitions which are implementation-defined.	
7835		



7836
7837 *Figure 43: Conference Class Diagram.*

7838

7839 4.11.2 ConferenceType

7840 4.11.2.1 Description

7841 A conference type represents a category of conferences.

7842 4.11.2.2 Class Definition

7843 The ConferenceType class is a mixin class which defines a type of conference.

7844 The ConferenceType class has attribute values:

7845

7846 **localNamespace**

7847 Value: icom_conf

7848
7849 **localName**
7850 Value: ConferenceType
7851
7852 **extendsFrom**
7853 Value:
7854
7855 **stereotype**
7856 Value: mixin
7857
7858 **description**
7859 Value: ConferenceType is a mixin class which defines type of conference.
7860
7861 **propertyDefinitions**
7862 The values for this attribute are defined in Section 4.11.2.3.

7863 **4.11.2.3 Property Definitions**

7864 The ConferenceType class MAY include additional property definitions which are implementation-defined.
7865

7866 **4.11.3 ConferenceTypeEnum**

7867 The ConferenceTypeEnum class is an enum class that enumerates the instances each of which
7868 expresses a type of a conference.

7869 The ConferenceTypeEnum class has attribute values:

7870
7871 **localNamespace**
7872 Value: icom_conf
7873
7874 **localName**
7875 Value: ConferenceTypeEnum
7876
7877 **extendsFrom**
7878 Value: icom_conf:ConferenceType
7879
7880 **stereotype**
7881 Value: primary
7882
7883 **isEnumeration**
7884 Value: TRUE
7885
7886 **description**
7887 Value: A type of a conference.
7888

7889 **instances**
7890 Value: <icom_conf:Impromptu, icom_conf:Scheduled, icom_conf:ChatRoom,
7891 icom_conf:OtherConferenceType>

7892
7893 ICOM defines four conference types:

- 7894 • **icom_conf:Impromptu** a conference session is started impromptu.
- 7895 • **icom_conf:Scheduled** a conference session is scheduled.
- 7896 • **icom_conf:ChatRoom** a conference is used for a chat room.
- 7897 • **icom_conf:OtherConferenceType** a conference is of other type.

7898

7899 **4.11.4 ConferenceStatus**

7900 **4.11.4.1 Description**

7901 A conference status is status of an online conference.

7902 **4.11.4.2 Class Definition**

7903 The ConferenceStatus class is a mixin class which defines status of an online conference.

7904 The ConferenceStatus class has attribute values:

7905

7906 **localNamespace**

7907 Value: icom_conf

7908

7909 **localName**

7910 Value: ConferenceStatus

7911

7912 **extendsFrom**

7913 Value:

7914

7915 **stereotype**

7916 Value: mixin

7917

7918 **description**

7919 Value: ConferenceStatus is a mixin class which defines status of an online conference.

7920

7921 **propertyDefinitions**

7922 The values for this attribute are defined in Section 4.11.4.3.

7923 **4.11.4.3 Property Definitions**

7924 The ConferenceStatus class MAY include additional property definitions which are implementation-
7925 defined.

7926

7927 4.11.5 ConferenceStatusEnum

7928 The ConferenceStatusEnum class is an enum class that enumerates the instances each of which
7929 expresses a status of a conference.

7930 The ConferenceStateEnum class has attribute values:

7931

7932 **localNamespace**

7933 Value: icom_conf

7934

7935 **localName**

7936 Value: ConferenceStatusEnum

7937

7938 **extendsFrom**

7939 Value: icom_conf:ConferenceStatus

7940

7941 **stereotype**

7942 Value: primary

7943

7944 **isEnumeration**

7945 Value: TRUE

7946

7947 **description**

7948 Value: Status of a conference.

7949

7950 **instances**

7951 Value: <icom_conf:NotStarted, icom_conf:WaitingForHost, icom_conf:Running,
7952 icom_conf:Hibernating, icom_conf:Ended>

7953

7954 ICOM defines five conference status:

- 7955 • **icom_conf:NotStarted** a conference session is not started .
- 7956 • **icom_conf:WaitingForHost** a conference session is waiting for a host.
- 7957 • **icom_conf:Running** a conference session is running.
- 7958 • **icom_conf:Hibernating** a conference session is hibernating.
- 7959 • **icom_conf:Ended** a conference session is ended.

7960

7961 4.11.6 ConferenceSession

7962 4.11.6.1 Description

7963 A conference session represents the metadata for a session of a conference.

7964 **4.11.6.2 Class Definition**

7965 The ConferenceSession class has attribute values:

7966

7967 **localNamespace**

7968 Value: icom_conf

7969

7970 **localName**

7971 Value: ConferenceSession

7972

7973 **extendsFrom**

7974 Value: icom_core:Identifiable

7975

7976 **stereotype**

7977 Value: primary

7978

7979 **description**

7980 Value: A conference session represents the metadata for a session of a conference.

7981

7982 **propertyDefinitions**

7983 The values for this attribute are defined in Section 4.11.6.3.

7984 **4.11.6.3 Property Definitions**

7985 The ConferenceSession class inherits property definitions from super classes.

7986 The ConferenceSession class MUST have the property definitions:

7987

7988 **icom_core:description**

7989 Description: Description of a conference session.

7990 Required: False

7991 Inherited: False

7992 Property Type: String

7993 Cardinality: Single

7994 Updatability: Read Write

7995

7996 **icom_core:startDate**

7997 Description: Start date and time of a conference session.

7998 Required: False

7999 Inherited: False

8000 Property Type: DateTime

8001 Cardinality: Single

8002 Updatability: Read Only

8003

8004	icom_core:endDate	
8005	Description:	End date and time of a conference session.
8006	Required:	False
8007	Inherited:	False
8008	Property Type:	DateTime
8009	Cardinality:	Single
8010	Updatability:	Read Only
8011		
8012	icom_conf:comment	
8013	Description:	Comment on a conference session.
8014	Required:	False
8015	Inherited:	False
8016	Property Type:	String
8017	Cardinality:	Single
8018	Updatability:	Read Write
8019		
8020	icom_conf:rating	
8021	Description:	Rating of a conference session.
8022	Required:	False
8023	Inherited:	False
8024	Property Type:	String
8025	Cardinality:	Single
8026	Updatability:	Read Write
8027		
8028	icom_conf:serverAddress	
8029	Description:	Address of a server that hosts a conference session.
8030	Required:	False
8031	Inherited:	False
8032	Property Type:	String
8033	Cardinality:	Single
8034	Updatability:	Read Only
8035		
8036	icom_conf:endingReason	
8037	Description:	Reason for ending a conference session.
8038	Required:	False
8039	Inherited:	False
8040	Property Type:	icom_conf:ConferenceSessionEndingReason
8041	Cardinality:	Single
8042	Updatability:	Read Only
8043		
8044	The ConferenceSession class MAY include additional property definitions which are implementation-	
8045	defined.	
8046		

8047 **4.11.7 ConferenceSessionEndingReason**

8048 **4.11.7.1 Description**

8049 A conference session ending reason is an indication of how a conference session ended.

8050 **4.11.7.2 Class Definition**

8051 The ConferenceSessionEndingReason class is a mixin class which defines an indication of how a
8052 conference session ended.

8053 The ConferenceSessionEndingReason class has attribute values:

8054

8055 **localNamespace**

8056 Value: icom_conf

8057

8058 **localName**

8059 Value: ConferenceSessionEndingReason

8060

8061 **extendsFrom**

8062 Value:

8063

8064 **stereotype**

8065 Value: mixin

8066

8067 **description**

8068 Value: ConferenceSessionEndingReason is a mixin class which defines an indication of how a
8069 conference session ended.

8070

8071 **propertyDefinitions**

8072 The values for this attribute are defined in Section 4.11.7.3.

8073 **4.11.7.3 Property Definitions**

8074 The ConferenceSessionEndingReason class MAY include additional property definitions which are
8075 implementation-defined.

8076

8077 **4.11.8 ConferenceSessionEndingReasonEnum**

8078 The ConferenceSessionEndingReasonEnum class is an enum class that enumerates the instances each
8079 of which expresses a reason for ending a conference session.

8080 The ConferenceSessionEndingReasonEnum class has attribute values:

8081

8082 **localNamespace**

8083 Value: icom_conf

8084

8085 **localName**

8086 Value: ConferenceSessionEndingReasonEnum

8087
8088 **extendsFrom**
8089 Value: icom_conf:ConferenceSessionEndingReason
8090
8091 **stereotype**
8092 Value: primary
8093
8094 **isEnumeration**
8095 Value: TRUE
8096
8097 **description**
8098 Value: Reason for ending a conference session.
8099
8100 **instances**
8101 Value: <icom_conf:HostLeft, icom_conf:HostAborted, icom_conf:NoHost, icom_conf:Hibernating>

8102

8103 ICOM defines four conference session states:

- 8104 • **icom_conf:HostLeft** a conference session ended after the host left.
- 8105 • **icom_conf:HostAborted** a conference session ended after the host aborted it.
- 8106 • **icom_conf:NoHost** a conference session ended due to no one hosting.
- 8107 • **icom_conf:Hibernating** a conference session is hibernating.

8108

8109 **4.11.9 ConferenceSetting**

8110 **4.11.9.1 Description**

8111 A conference setting represents configuration settings for sessions of a conference.

8112 **4.11.9.2 Class Definition**

8113 The ConferenceSetting class has attribute values:

8114

8115 **localNamespace**
8116 Value: icom_conf

8117

8118 **localName**
8119 Value: ConferenceSetting

8120

8121 **extendsFrom**
8122 Value:

8123

8124 **stereotype**
8125 Value: primary

8126

8127 **description**
8128 Value: A conference setting represents configuration settings for sessions of a conference.

8129
8130 **propertyDefinitions**
8131 The values for this attribute are defined in Section 4.11.9.3.

8132 **4.11.9.3 Property Definitions**

8133 The ConferenceSetting class inherits property definitions from super classes.
8134 The ConferenceSetting class MUST have the property definitions:

8135
8136 **icom_meta:property**
8137 Description: Configurable properties for a conference.
8138 Required: False
8139 Inherited: False
8140 Property Type: icom_meta:property
8141 Cardinality: Multi
8142 Updatability: Read Write

8143
8144 **icom_conf:participantRole**
8145 Description: Role settings for conference participants.
8146 Required: False
8147 Inherited: False
8148 Property Type: icom_conf:ConferenceParticipantRole
8149 Cardinality: Multi
8150 Updatability: Read Write

8151
8152 The ConferenceSetting class MAY include additional property definitions which are implementation-
8153 defined.

8154

8155 **4.11.10 ConferenceParticipantRole**

8156 **4.11.10.1 Description**

8157 A conference participant role defines roles settings for a conference participant.

8158 **4.11.10.2 Class Definition**

8159 The ConferenceParticipantRole class has attribute values:

8160
8161 **localNamespace**
8162 Value: icom_conf

8163
8164 **localName**
8165 Value: ConferenceParticipantRole

8166

8209	Property Type:	String
8210	Cardinality:	Multi
8211	Updatability:	Read Write
8212		
8213	icom_conf:keyword	
8214	Description:	One or more key words to activate a role setting.
8215	Required:	False
8216	Inherited:	False
8217	Property Type:	String
8218	Cardinality:	Multi
8219	Updatability:	Read Write
8220		
8221	The ConferenceParticipantRole class MAY include additional property definitions which are	
8222	implementation-defined.	

8223 **5 Conformance**

8224 **5.1 Software Architecture or Framework Dependence**

8225 The ICOM specification does not presume a particular software architecture or framework for use of the
8226 ICOM model.

8227 Fulfillment of ICOM use case roles and accompanying responsibilities is implementation dependent.

8228

8229 **5.2 Platform Provider Conformance**

8230 **5.2.1 Platform Provider Conformance – No Extension Modules**

8231 An ICOM platform provider with no extension modules (Section 4):

8232 a. SHALL conform to all mandatory statements and

8233 b. MAY conform to optional statements

8234 of the core ICOM model as defined in Section 3 of this standard.

8235

8236 **5.2.2 Platform Provider Conformance – One or More Extension Modules**

8237 An ICOM platform provider with extension modules (Section 4):

8238 a. SHALL conform to Section 5.2.1 and

8239 b. SHALL conform to all mandatory statements and

8240 c. MAY conform to optional statements

8241 as defined in Section 4 for each extension module.

8242

8243 **5.3 Service Provider Conformance**

8244 **5.3.1 ICOM Service Provider – No Extension Modules**

8245 An ICOM service provider may provide one or more services defined in Section 3. For each such service
8246 provided, an ICOM service provider:

8247 a. SHALL conform to all mandatory statements and

8248 b. MAY conform to optional statements

8249 for the classes, super classes, and related classes defined in Section 3 of this standard.

8250

8251 **5.3.2 ICOM Service Provider – One or More Extension Modules**

8252 An ICOM service provider MAY support one or more extension modules as defined in Section 4 of this
8253 standard. For each service provided, an ICOM service provider:

8254 a. SHALL conform to Section 5.3.1 (if an offered service is defined in Section 3) and

8255 b. SHALL conform to all mandatory statements and

8256 c. MAY conform to optional statements

8257 as defined in Section 4 for that extension module.

8258

8259 **5.4 ICOM Producer Conformance**

8260 **5.4.1 ICOM Producer Conformance – No Extension Modules**

8261 An ICOM producer that produces no objects of a class conforming to Section 4:

- 8262 a. SHALL conform to all mandatory statements and
- 8263 b. MAY conform to optional statements

8264 for the class and super classes thereof in Section 3 of this standard, for any object produced.

8265

8266 **5.4.2 ICOM Producer Conformance – One or More Extension Modules**

8267 An ICOM producer that produces objects of a class conforming to Section 4:

- 8268 a. SHALL conform to Section 5.4.1 and
- 8269 b. SHALL conform to all mandatory statements and
- 8270 c. MAY conform to optional statements

8271 as defined in Section 4 for that extension module.

8272

8273 **5.5 ICOM Consumer Conformance**

8274 **5.5.1 ICOM Consumer Conformance – No Extension Modules**

8275 An ICOM consumer that consumes no objects of a class conforming to Section 4:

- 8276 a. SHALL conform to all mandatory statements and
- 8277 b. MAY conform to optional statements

8278 for the class and super classes thereof in Section 3 of this standard, for any object consumed.

8279

8280 **5.5.2 ICOM Consumer Conformance – Extension Modules**

8281 An ICOM consumer that consumes objects of a class conforming to Section 4:

- 8282 a. SHALL conform to Section 5.5.1 and
- 8283 b. SHALL conform to all mandatory statements and
- 8284 c. MAY conform to optional statements

8285 as defined in Section 4 for that extension module.

8286

Appendix A. Acknowledgements

8287 The following individuals have participated in the creation of this specification and are gratefully
8288 acknowledged:

8289 **Participants:**

8290 Rafiul Ahad, Oracle Corporation
8291 Kenneth P. Baclawski, Northeastern University
8292 Eric S. Chan, Oracle Corporation
8293 Martin Chapman, Oracle Corporation
8294 Scott Conroy, Individual
8295 Stefan Decker, Digital Enterprise Research Institute (DERI)
8296 Laura Dragan, Digital Enterprise Research Institute (DERI)
8297 Patrick Durusau, Individual
8298 Siegfried Handschuh, Digital Enterprise Research Institute (DERI)
8299 Deirdre Lee, Digital Enterprise Research Institute (DERI)
8300 Marc Pallot, ESoCE-NET
8301 Chancellor Pascale, Johns Hopkins University Applied Physics Laboratory
8302 Vassilios Peristeras, Digital Enterprise Research Institute (DERI)
8303 Peter Saint-Andre, Cisco Systems, Inc.
8304 Ramesh Vasudevan, Oracle Corporation
8305 Peter Yim, Individual
8306

8307

Appendix B. Revision History

8308

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.
CSPRD 04	June 26, 2012	Ken Baclawski	Add 4 additional attributes from grammar to PropertyDefinition metadata model, corrected spelling of Cardinality, renamed the address property of Addressable to entityAddress to avoid clashing with the address properties of EntityAddress and Participant, and specified the omitted namespaces of the superCategories of some of the enumerations.
CSPRD 05	October 15, 2012	Ken Baclawski Eric S. Chan Patrick Durusau	Change InstantMessage isAbstract to false, change PropertyType to optional in PropertyDefinition, change cardinality of superCategory property in Category to multi, add ClassDefinition, Stereotype, StereotypeEnum in icom_meta, add Figure 18 ClassDefinition UML diagram, remove EntityDefinition in icom_core. Updated the conformance clauses in Section 5.

8309

8310

8311