



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

Committee Specification Draft **0405** /
Public Review Draft **0405**

~~08-August~~**27 October** 2012

Specification URIs

This version:

~~icom-ics-v1.0-csprd04.doc (Authoritative)~~
~~icom-ics-v1.0-csprd04.html~~
~~icom-ics-v1.0-csprd04.pdf~~
<http://docs.oasis-open.org/icom/icom-ics/v1.0/csprd05/icom-ics-v1.0-csprd05.doc> (Authoritative)
<http://docs.oasis-open.org/icom/icom-ics/v1.0/csprd05/icom-ics-v1.0-csprd05.html>
<http://docs.oasis-open.org/icom/icom-ics/v1.0/csprd05/icom-ics-v1.0-csprd05.pdf>

Previous version:

<http://docs.oasis-open.org/icom/icom-ics/v1.0/csprd03/icom-ics-v1.0-csprd03.doc> (Authoritative)
<http://docs.oasis-open.org/icom/icom-ics/v1.0/csprd03/icom-ics-v1.0-csprd03.html>
<http://docs.oasis-open.org/icom/icom-ics/v1.0/csprd03/icom-ics-v1.0-csprd03.pdf>
<http://www.oasis-open.org/committees/download.php/46823/icom-ics-v1.0-csprd04.zip>

Latest version:

~~N/A~~
<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: /schemas/~~
- XML schemas: <http://docs.oasis-open.org/icom/icom-ics/v1.0/csprd05/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>
<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]

~~OASIS~~ Integrated Collaboration Object Model (ICOM) for Interoperable Collaboration Services Version 1.0. ~~08-August~~27 October 2012. OASIS Committee Specification Draft ~~0405~~ / Public Review Draft ~~04-~~

05.

<http://docs.oasis-open.org/icom/icom-ics/v1.0/csprd05/icom-ics-v1.0-csprd05.html>.

Notices

Copyright © OASIS Open 2012. All Rights Reserved.

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

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

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

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

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

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

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

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

Table of Contents

1	Introduction	17
1.1	Terminology	18
1.2	Normative References	18
1.3	Non-Normative References	18
2	Modeling Language	20
2.1	Introduction	20
2.2	Class Definition Grammar	20
2.3	Property Definition Grammar	22
2.4	Namespaces	25
3	Core Model	26
3.1	Main Branch	26
3.1.1	Entity and Top-Level Subclasses	26
3.1.2	Identifiable	26
3.1.3	Parental	27
3.1.4	Extent	28
3.1.5	Entity	29
3.1.6	EntityDefinition	33
3.1.7	Overview of Scope, Subject, and Artifact Branches	34
3.2	Scope Branch	35
3.2.1	Scope and Top-Level Subclasses	35
3.2.2	Scope	36
3.2.3	Community	38
3.2.4	Space	40
3.3	Subject Branch	42
3.3.1	Subject and Top-Level Subclasses	42
3.3.2	Subject	42
3.3.3	Group	44
3.3.4	Actor	46
3.3.5	Person	48
3.3.6	Resource	52
3.3.7	ResourceType	54
3.3.8	ResourceTypeEnum	55
3.3.9	ResourceBookingRule	56
3.3.10	ResourceBookingRuleEnum	57
3.4	Artifact Branch	58
3.4.1	Artifact and Top-Level Subclasses	58
3.4.2	Item	58
3.4.3	SpaceItem	60
3.4.4	Container	60
3.4.5	FolderContainer	61
3.4.6	Artifact	62
3.4.7	Folder	65
3.4.8	HeterogeneousFolder	66

3.5 Access Control Model	68
3.5.1 Accessor	68
3.5.2 Owner	68
3.5.3 RoleDefinition	69
3.5.4 Role	70
3.5.5 Privilege	72
3.5.6 PrivilegeEnum	73
3.5.7 AccessControlList	74
3.5.8 AccessControlEntry	75
3.5.9 AccessType	76
3.5.10 AccessTypeEnum	77
3.6 Metadata Model	78
3.6.1 PropertyDefinition	83
3.6.2 Property	86
3.6.3 PropertyChoiceType	88
3.6.4 PropertyType	89
3.6.5 PropertyTypeEnum	90
3.6.6 Updatability	91
3.6.7 UpdatabilityEnum	92
3.6.8 Cardinality	93
3.6.9 CardinalityEnum	93
3.6.10 Marker and Subclasses	94
3.6.11 Marker	95
3.6.12 Category	96
3.6.13 CategoryApplication	98
3.6.14 Tag	99
3.6.15 TagApplication	101
3.6.16 RelationshipBondable	102
3.6.17 RelationshipDefinition	103
3.6.18 Relationship	105
3.7 Common Concepts	107
3.7.1 Addressable	107
3.7.2 EntityAddress	108
3.7.3 Participant	109
3.7.4 Priority	111
3.7.5 PriorityEnum	112
3.7.6 DateTimeResolution	112
3.7.7 DateTimeResolutionEnum	113
3.7.8 TimeZone	114
3.7.9 Location	115
3.7.10 GeoCoordinates	117
4 Extension Modules	119
4.1 Overview of Extension Modules	119
4.2 Content Module	121
4.2.1 MimeConvertible	121

4.2.2 Content	121
4.2.3 MultiContent	123
4.2.4 SimpleContent	124
4.2.5 OnlineContent	126
4.2.6 ContentDispositionType	127
4.2.7 ContentDispositionTypeEnum	128
4.2.8 AttachedItem	129
4.3 Document Module	130
4.3.1 Versionable	130
4.3.2 VersionControlMetadata	132
4.3.3 VersionSeries	134
4.3.4 Version	136
4.3.5 VersionType	138
4.3.6 VersionTypeEnum	139
4.3.7 Document	140
4.3.8 WikiPage	141
4.4 Message Module	143
4.4.1 Message	143
4.4.2 UnifiedMessage	144
4.4.3 UnifiedMessageParticipant	149
4.4.4 UnifiedMessageFlag	150
4.4.5 UnifiedMessageFlagEnum	151
4.4.6 UnifiedMessageDeliveryStatusNotificationRequest	152
4.4.7 UnifiedMessageDeliveryStatusNotificationRequestEnum	152
4.4.8 UnifiedMessageChannel	153
4.4.9 UnifiedMessageChannelEnum	154
4.4.10 UnifiedMessageEditMode	155
4.4.11 UnifiedMessageEditModeEnum	155
4.4.12 InstantMessage	157
4.4.13 InstantMessageType	160
4.4.14 InstantMessageTypeEnum	161
4.4.15 InstantMessageChatStatus	162
4.4.16 InstantMessageChatStatusEnum	162
4.4.17 InstantMessageFeed	163
4.4.18 InstantMessageConnection	165
4.5 Presence Module	168
4.5.1 Presence	168
4.5.2 PresenceEditMode	170
4.5.3 PresenceEditModeEnum	171
4.5.4 ContactMethod	172
4.5.5 ContactReachabilityStatus	174
4.5.6 ContactReachabilityStatusEnum	174
4.5.7 Activity	177
4.5.8 ActivityType	178
4.5.9 ActivityTypeEnum	179

4.6 Address Book Module	180
4.6.1 AddressBook	180
4.6.2 PersonContact	181
4.7 Calendar Module	186
4.7.1 Calendar	186
4.7.2 OccurrenceSeries	188
4.7.3 Occurrence	193
4.7.4 OccurrenceStatus	199
4.7.5 OccurrenceStatusEnum	199
4.7.6 OccurrenceType	200
4.7.7 OccurrenceTypeEnum	201
4.7.8 OccurrenceParticipant	202
4.7.9 OccurrenceParticipantStatus	203
4.7.10 OccurrenceParticipantStatusEnum	203
4.7.11 OccurrenceParticipantTransparency	204
4.7.12 OccurrenceParticipantTransparencyEnum	205
4.7.13 OccurrenceEditMode	206
4.7.14 OccurrenceEditModeEnum	207
4.8 Free Busy Module	207
4.8.1 FreeBusy	207
4.8.2 FreeBusyInterval	209
4.8.3 FreeBusyType	211
4.8.4 FreeBusyTypeEnum	212
4.9 Task List Module	213
4.9.1 TaskList	213
4.9.2 Task	214
4.9.3 TaskStatus	219
4.9.4 TaskStatusEnum	219
4.9.5 TaskParticipantStatus	220
4.9.6 TaskParticipantStatusEnum	221
4.9.7 TaskEditMode	222
4.9.8 TaskEditModeEnum	222
4.10 Forum Module	223
4.10.1 Discussion	223
4.10.2 DiscussionContainer	224
4.10.3 DiscussionMessage	225
4.10.4 TopicContainer	226
4.10.5 Forum	228
4.10.6 Topic	230
4.10.7 Announcement	231
4.10.8 AnnouncementStatus	233
4.10.9 AnnouncementStatusEnum	233
4.11 Conference Module	234
4.11.1 Conference	234
4.11.2 ConferenceType	237

4.11.3 ConferenceTypeEnum	238
4.11.4 ConferenceStatus	239
4.11.5 ConferenceStatusEnum	240
4.11.6 ConferenceSession	240
4.11.7 ConferenceSessionEndingReason	243
4.11.8 ConferenceSessionEndingReasonEnum	244
4.11.9 ConferenceSetting	244
4.11.10 ConferenceParticipantRole	246
5 Conformance	248
Appendix A. Acknowledgements	251
Appendix B. Revision History	252
1 Introduction	17
1.1 Terminology	18
1.2 Normative References	18
1.3 Non-Normative References	18
2 Modeling Language	20
2.1 Introduction	20
2.2 Class Definition Grammar	20
2.3 Property Definition Grammar	22
2.4 Namespaces	25
3 Core Model	26
3.1 Main Branch	26
3.1.1 Entity and Top-Level Subclasses	26
3.1.2 Identifiable	26
3.1.3 Parental	27
3.1.4 Extent	28
3.1.5 Entity	29
3.1.6 Overview of Scope, Subject, and Artifact Branches	34
3.2 Scope Branch	35
3.2.1 Scope and Top-Level Subclasses	35
3.2.2 Scope	36
3.2.3 Community	38
3.2.4 Space	40
3.3 Subject Branch	42
3.3.1 Subject and Top-Level Subclasses	42
3.3.2 Subject	42
3.3.3 Group	44
3.3.4 Actor	46
3.3.5 Person	48
3.3.6 Resource	52
3.3.7 ResourceType	54
3.3.8 ResourceTypeEnum	55
3.3.9 ResourceBookingRule	56
3.3.10 ResourceBookingRuleEnum	57
3.4 Artifact Branch	58

3.4.1 Artifact and Top-Level Subclasses.....	58
3.4.2 Item.....	58
3.4.3 SpaceItem	60
3.4.4 Container.....	60
3.4.5 FolderContainer.....	61
3.4.6 Artifact	62
3.4.7 Folder	65
3.4.8 HeterogeneousFolder.....	66
3.5 Access Control Model.....	68
3.5.1 Accessor.....	68
3.5.2 Owner	68
3.5.3 RoleDefinition	69
3.5.4 Role	70
3.5.5 Privilege.....	72
3.5.6 PrivilegeEnum	73
3.5.7 AccessControllist.....	74
3.5.8 AccessControlEntry	75
3.5.9 AccessType	76
3.5.10 AccessTypeEnum	77
3.6 Metadata Model	78
3.6.1 ClassDefinition.....	78
3.6.2 Stereotype	81
3.6.3 StereotypeEnum	82
3.6.4 PropertyDefinition	83
3.6.5 Property	86
3.6.6 PropertyChoiceType.....	88
3.6.7 PropertyType	89
3.6.8 PropertyTypeEnum	90
3.6.9 Updatability	91
3.6.10 UpdatabilityEnum	92
3.6.11 Cardinality.....	93
3.6.12 CardinalityEnum	93
3.6.13 Marker and Subclasses.....	94
3.6.14 Marker	95
3.6.15 Category	96
3.6.16 CategoryApplication	98
3.6.17 Tag	99
3.6.18 TagApplication.....	101
3.6.19 RelationshipBondable	102
3.6.20 RelationshipDefinition.....	103
3.6.21 Relationship.....	105
3.7 Common Concepts	107
3.7.1 Addressable.....	107
3.7.2 EntityAddress	108
3.7.3 Participant.....	109

3.7.4	Priority	111
3.7.5	PriorityEnum	112
3.7.6	DateTimeResolution	112
3.7.7	DateTimeResolutionEnum	113
3.7.8	TimeZone	114
3.7.9	Location	115
3.7.10	GeoCoordinates	117
4	Extension Modules	119
4.1	Overview of Extension Modules	119
4.2	Content Module	121
4.2.1	MimeConvertible	121
4.2.2	Content	121
4.2.3	MultiContent	123
4.2.4	SimpleContent	124
4.2.5	OnlineContent	126
4.2.6	ContentDispositionType	127
4.2.7	ContentDispositionTypeEnum	128
4.2.8	AttachedItem	129
4.3	Document Module	130
4.3.1	Versionable	130
4.3.2	VersionControlMetadata	132
4.3.3	VersionSeries	134
4.3.4	Version	136
4.3.5	VersionType	138
4.3.6	VersionTypeEnum	139
4.3.7	Document	140
4.3.8	WikiPage	141
4.4	Message Module	143
4.4.1	Message	143
4.4.2	UnifiedMessage	144
4.4.3	UnifiedMessageParticipant	149
4.4.4	UnifiedMessageFlag	150
4.4.5	UnifiedMessageFlagEnum	151
4.4.6	UnifiedMessageDeliveryStatusNotificationRequest	152
4.4.7	UnifiedMessageDeliveryStatusNotificationRequestEnum	152
4.4.8	UnifiedMessageChannel	153
4.4.9	UnifiedMessageChannelEnum	154
4.4.10	UnifiedMessageEditMode	155
4.4.11	UnifiedMessageEditModeEnum	155
4.4.12	InstantMessage	157
4.4.13	InstantMessageType	160
4.4.14	InstantMessageTypeEnum	161
4.4.15	InstantMessageChatStatus	162
4.4.16	InstantMessageChatStatusEnum	162
4.4.17	InstantMessageFeed	163

4.4.18 InstantMessageConnection	165
4.5 Presence Module	168
4.5.1 Presence	168
4.5.2 PresenceEditMode	170
4.5.3 PresenceEditModeEnum	171
4.5.4 ContactMethod	172
4.5.5 ContactReachabilityStatus	174
4.5.6 ContactReachabilityStatusEnum	174
4.5.7 Activity	177
4.5.8 ActivityType	178
4.5.9 ActivityTypeEnum	179
4.6 Address Book Module	180
4.6.1 AddressBook	180
4.6.2 PersonContact	181
4.7 Calendar Module	186
4.7.1 Calendar	186
4.7.2 OccurrenceSeries	188
4.7.3 Occurrence	193
4.7.4 OccurrenceStatus	199
4.7.5 OccurrenceStatusEnum	199
4.7.6 OccurrenceType	200
4.7.7 OccurrenceTypeEnum	201
4.7.8 OccurrenceParticipant	202
4.7.9 OccurrenceParticipantStatus	203
4.7.10 OccurrenceParticipantStatusEnum	203
4.7.11 OccurrenceParticipantTransparency	204
4.7.12 OccurrenceParticipantTransparencyEnum	205
4.7.13 OccurrenceEditMode	206
4.7.14 OccurrenceEditModeEnum	207
4.8 Free Busy Module	207
4.8.1 FreeBusy	207
4.8.2 FreeBusyInterval	209
4.8.3 FreeBusyType	211
4.8.4 FreeBusyTypeEnum	212
4.9 Task List Module	213
4.9.1 TaskList	213
4.9.2 Task	214
4.9.3 TaskStatus	219
4.9.4 TaskStatusEnum	219
4.9.5 TaskParticipantStatus	220
4.9.6 TaskParticipantStatusEnum	221
4.9.7 TaskEditMode	222
4.9.8 TaskEditModeEnum	222
4.10 Forum Module	223
4.10.1 Discussion	223

4.10.2 DiscussionContainer	224
4.10.3 DiscussionMessage	225
4.10.4 TopicContainer	226
4.10.5 Forum	228
4.10.6 Topic	230
4.10.7 Announcement	231
4.10.8 AnnouncementStatus	233
4.10.9 AnnouncementStatusEnum	233
4.11 Conference Module	234
4.11.1 Conference	234
4.11.2 ConferenceType	237
4.11.3 ConferenceTypeEnum	238
4.11.4 ConferenceStatus	239
4.11.5 ConferenceStatusEnum	240
4.11.6 ConferenceSession	240
4.11.7 ConferenceSessionEndingReason	243
4.11.8 ConferenceSessionEndingReasonEnum	244
4.11.9 ConferenceSetting	244
4.11.10 ConferenceParticipantRole	246
5 Conformance	248
5.1 Software Architecture or Framework Dependence	248
5.2 Platform Provider Conformance	248
5.2.1 Platform Provider Conformance – No Extension Modules	248
5.2.2 Platform Provider Conformance – One or More Extension Modules	248
5.3 Service Provider Conformance	248
5.3.1 ICOM Service Provider – No Extension Modules	248
5.3.2 ICOM Service Provider – One or More Extension Modules	249
5.4 ICOM Producer Conformance	249
5.4.1 ICOM Producer Conformance – No Extension Modules	249
5.4.2 ICOM Producer Conformance – One or More Extension Modules	249
5.5 ICOM Consumer Conformance	250
5.5.1 ICOM Consumer Conformance – No Extension Modules	250
5.5.2 ICOM Consumer Conformance – Extension Modules	250
Appendix A. Acknowledgements	251
Appendix B. Revision History	252

Table of Figures

Figure 1: Entity and Top-Level Abstract Classes.....	26
Figure 2: Entity Class Diagram.....	33
Figure 3: Scope, Subject, and Artifact Branches.....	35
Figure 4: Scope Branch.....	35
Figure 5: Scope Class Diagram.....	38
Figure 6: Community Class Diagram.....	40
Figure 7: Space Class Diagram.....	41
Figure 8: Subject Branch.....	42
Figure 9: Subject Class Diagram.....	44
Figure 10: Group and Actor Class Diagram.....	46
Figure 11: Person Class Diagram.....	52
Figure 12: Resource Class Diagram.....	54
Figure 13: Artifact Branch.....	58
Figure 14: Artifact Class Diagram.....	65
Figure 15: Heterogeneous Folder Class Diagram.....	67
Figure 16: Role Definition and Role Class Diagram.....	72
Figure 17: Access Control List Class Diagram.....	78
Figure 18: Property Definition and Property Class Diagram.....	88
Figure 19: Marker Branch.....	94
Figure 20: Marker Class Diagram.....	96
Figure 21: Category and Category Application Class Diagram.....	98
Figure 22: Tag and Tag Application Class Diagram.....	101
Figure 23: Relationship Class Diagram.....	107
Figure 24: Containers of Collaboration Activities.....	119
Figure 25: Composite Content Class Diagram.....	123
Figure 26: Document, Version Series, and Version Class Diagram.....	141
Figure 27: Wiki Page Class Diagram.....	143
Figure 28: Unified Message Class Diagram.....	157
Figure 29: Instant Message Class Diagram.....	160
Figure 30: Instant Message Feed and Connection Class Diagram.....	165
Figure 31: Presence Class Diagram.....	170
Figure 32: Presence Contact Method and Instant Message Connection Class Diagram.....	176
Figure 33: Address Book Class Diagram.....	181
Figure 34: Person Contact Class Diagram.....	186
Figure 35: Calendar Class Diagram.....	188
Figure 36: Occurrence Series Class Diagram.....	193
Figure 37: Occurrence Class Diagram.....	198
Figure 38: Free Busy Class Diagram.....	211
Figure 39: Task List Class Diagram.....	214
Figure 40: Task Class Diagram.....	218

Figure 41: Forum Class Diagram.....	228
Figure 42: Conference Class Diagram.....	237
Figure 1: Entity and Top-Level Abstract Classes.....	26
Figure 2: Entity Class Diagram.	33
Figure 3: Scope, Subject, and Artifact Branches.	35
Figure 4: Scope Branch.	35
Figure 5: Scope Class Diagram.	38
Figure 6: Community Class Diagram.	40
Figure 7: Space Class Diagram.	41
Figure 8: Subject Branch.....	42
Figure 9: Subject Class Diagram.	44
Figure 10: Group and Actor Class Diagram.....	46
Figure 11: Person Class Diagram.....	52
Figure 12: Resource Class Diagram.....	54
Figure 13: Artifact Branch.	58
Figure 14: Artifact Class Diagram.	65
Figure 15: Heterogeneous Folder Class Diagram.	67
Figure 16: Role Definition and Role Class Diagram.	72
Figure 17: Access Control List Class Diagram.	78
Figure 18: Class Definition and Property Definition Class Diagram.	81
Figure 19: Property Definition and Property Class Diagram.	88
Figure 20: Marker Branch.	94
Figure 21: Marker Class Diagram.	96
Figure 22: Category and Category Application Class Diagram.	98
Figure 23: Tag and Tag Application Class Diagram.	101
Figure 24: Relationship Class Diagram.	107
Figure 25: Containers of Collaboration Activities.	119
Figure 26: Composite Content Class Diagram.	123
Figure 27: Document, Version Series, and Version Class Diagram.	141
Figure 28: Wiki Page Class Diagram.	143
Figure 29: Unified Message Class Diagram.	157
Figure 30: Instant Message Class Diagram.....	160
Figure 31: Instant Message Feed and Connection Class Diagram.	165
Figure 32: Presence Class Diagram.	170
Figure 33: Presence Contact Method and Instant Message Connection Class Diagram.	176
Figure 34: Address Book Class Diagram.....	181
Figure 35: Person Contact Class Diagram.	186
Figure 36: Calendar Class Diagram.....	188
Figure 37: Occurrence Series Class Diagram.	193
Figure 38: Occurrence Class Diagram.....	198
Figure 39: Free Busy Class Diagram.....	211
Figure 40: Task List Class Diagram.....	214

Figure 41: Task Class Diagram.	218
Figure 42: Forum Class Diagram.	228
Figure 43: Conference Class Diagram.	237

1 Introduction

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

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

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

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

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

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

52

53 1.1 Terminology

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

57 1.2 Normative References

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

70 1.3 Non-Normative References

- 71 [BPEL4People] OASIS Committee Specification, *WS-BPEL Extension for People (BPEL4People)*
72 *Specification Version 1.1*, August 2010. [http://docs.oasis-](http://docs.oasis-open.org/bpel4people/bpel4people-1.1.html)
73 [open.org/bpel4people/bpel4people-1.1.html](http://docs.oasis-open.org/bpel4people/bpel4people-1.1.html)
- 74 [BPMN] OMG, "Business Process Model and Notation (BPMN) Version 2.0", January
75 2011. (<http://www.omg.org/spec/BPMN/2.0/PDF>)
- 76 [FOAF] Brickley, D. and Miller, L., "FOAF Vocabulary Specification", August 2009.
77 (<http://xmlns.com/foaf/spec/>)
- 78 [ICOM Wiki] OASIS ICOM TC Wiki, (<http://wiki.oasis-open.org/icom>)
- 79 [JCR 2.0] Java Specification Request (JSR) 283, *Content Repository for Java™*
80 *Technology API 2.0 Specification*, August 2009.
81 (<http://jcp.org/en/jsr/detail?id=283>)
- 82 [OpenGraph] Facebook Platform Open Graph Core Concepts,
83 (<http://developers.facebook.com/docs/coreconcepts/>)
- 84 [OpenSocial] OpenSocial and Gadgets Specification Group, "Social Data Specification",
85 November 2010. ([http://opensocial-](http://opensocial-resources.googlecode.com/svn/spec/2.0/Social-Data.xml)
86 [resources.googlecode.com/svn/spec/2.0/Social-Data.xml](http://opensocial-resources.googlecode.com/svn/spec/2.0/Social-Data.xml))
- 87 [RFC2446] Crispin, M., "Internet Message Access Protocol – Version 4rev1", RFC 2060,
88 December 1996. (<http://tools.ietf.org/html/rfc2060>)
- 89 [RFC2426] Dawson, F. and Howes, T., "vCard MIME Directory Profile", RFC 2426,
90 September 1998. (<http://tools.ietf.org/html/rfc2426>)
- 91 [RFC3920] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core",
92 RFC 3920, October 2004. (<http://tools.ietf.org/html/rfc3920>)
- 93 [RFC3921] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Instant
94 Messaging and Presence", RFC 3921, October 2004.
95 (<http://tools.ietf.org/html/rfc3921>)
- 96 [RFC4512] Zeilenga, K., "Lightweight Directory Access Protocol (LDAP): Directory
97 Information Models", RFC 4512, June 2006. (<http://tools.ietf.org/html/rfc4512>)
- 98 [RFC4791] Daboo, C. and Desruisseaux, B., "Calendaring Extensions to WebDAV
99 (CalDAV)", RFC 4791, March 2007. (<http://tools.ietf.org/html/rfc4791>)

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

2 Modeling Language

2.1 Introduction

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

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

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

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

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

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

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

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

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

2.2 Class Definition Grammar

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

namespace String

The `namespace` attribute specifies an IRI.

localName String

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

description String (optional)

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

extendsFrom IRI (multi-valued)

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

stereotype Enum

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

The values of `stereotype` attribute are:

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

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

Inheritance is constrained by:

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

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

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

isAbstract Boolean

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

The values of `isAbstract` attribute are:

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

The default value is **FALSE**.

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

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

isEnumeration Boolean

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

The values of `isEnumeration` attribute are:

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

The default value is **FALSE**.

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

instances

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

propertyDefinition property-definition (multi-valued)

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

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

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

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

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

2.3 Property Definition Grammar

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

namespace String

The `namespace` attribute specifies an IRI.

localName String

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

description String (optional)

The `description` attribute specifies a description of a property

propertyType Enum

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

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

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

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

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

cardinality Enum

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

The values of `cardinality` attribute are:

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

updatability Enum

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

The values of `updatability` attribute are:

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

inherited Boolean

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

The values of `inherited` attribute are:

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

required Boolean

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

The values of `required` attribute are:

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

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

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

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

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

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

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

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

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

294 attribute.

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

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

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

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

299 listed in `choices` attribute.

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

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

302 `choices` attribute.

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

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

305

306 **openChoice** Boolean

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

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

309 The values of `openChoice` attribute are:

310

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

312

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

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

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

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

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

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

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

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

321 `propertyType` attribute of a property definition.

322

323 **minValue** Integer | Decimal

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

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

326

327 **maxValue** Integer | Decimal

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

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

330

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

displayName String

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

value **property-type**

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

2.4 Namespaces

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

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

Table 1 Namespace prefixes and IRI references.

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

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

3 Core Model

3.1 Main Branch

3.1.1 Entity and Top-Level Subclasses

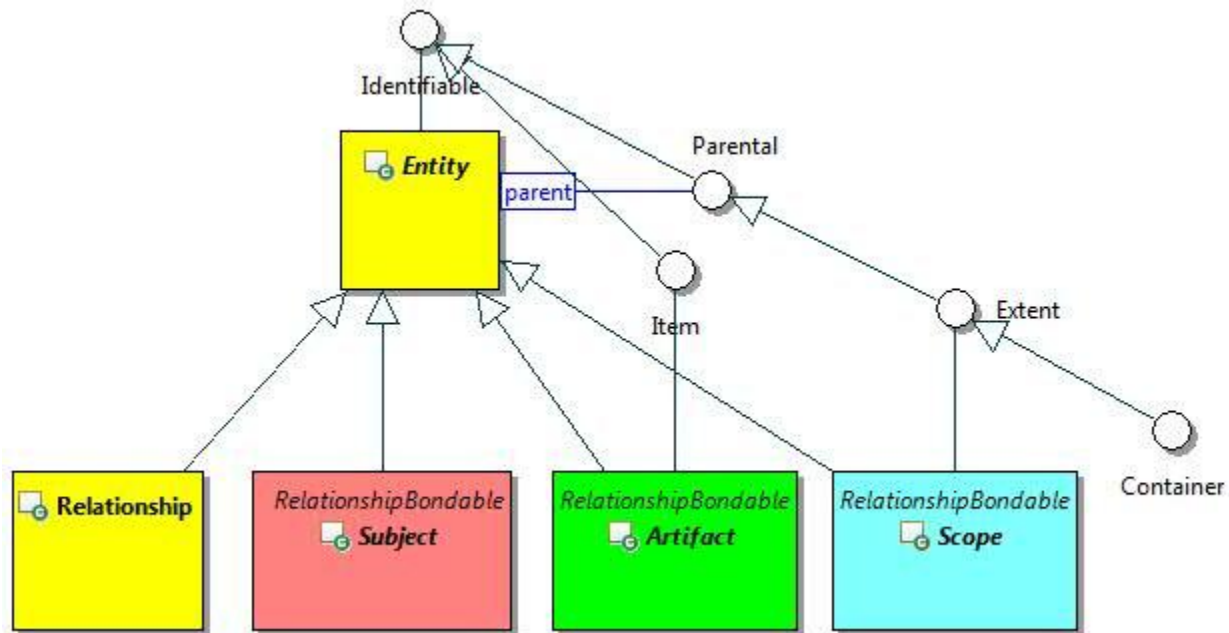


Figure 1: Entity and Top-Level Abstract Classes.

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

3.1.2 Identifiable

3.1.2.1 Description

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

3.1.2.2 Class Definition

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

The `Identifiable` class has attribute values:

localNamespace

Value: `icom_core`

localName

Value: `Identifiable`

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:

388

389 **icom_core:objectId**

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

396

397 **icom_core:changeToken**

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

3.1.3.2 Class Definition

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

The Parental class has attribute values:

localNamespace

Value: icom_core

localName

Value: Parental

extendsFrom

Value: icom_core:Identifiable

stereotype

Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.1.3.3.

3.1.3.3 Property Definitions

The Parental class inherits property definitions from super classes.

The Parental class MUST have the property definition:

icom_core:parent

Description: Parent of an object.

Required: False

Inherited: False

Property Type: icom_core:Parental

Cardinality: Single

Updatability: Read Only

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

3.1.4 Extent

3.1.4.1 Description

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

3.1.4.2 Class Definition

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

The Extent class has attribute values:

localNamespace

Value: icom_core

localName

Value: Extent

extendsFrom

Value: icom_core:Parental

stereotype

Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.1.4.3.

3.1.4.3 Property Definitions

The Extent class inherits property definitions from super classes.

The Extent class MUST have the property definition:

icom_core:parent

Description: Parent of an extent.

Required: False

Inherited: True

Property Type: icom_core:Extent

Cardinality: Single

Updatability: Read Only

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

3.1.5 Entity

3.1.5.1 Description

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

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

492 3.1.5.2 Class Definition

493 The Entity class has attribute values:

494
495 **localNamespace**
496 Value: icom_core
497
498 **localName**
499 Value: Entity
500
501 **extendsFrom**
502 Value: icom_core:Identifiable
503
504 **stereotype**
505 Value: primary
506
507 **isAbstract**
508 Value: TRUE
509
510 **description**
511 Value: An entity is an object with an immutable id and individual access control.
512
513 **propertyDefinitions**
514 The values for this attribute are defined in Section 3.1.5.3.

515 3.1.5.3 Property Definitions

516 The Entity class inherits property definitions from super classes.

517 The Entity class MUST have the property definitions:

518
519 **icom_core:name**
520 Description: Name of an entity.
521 Required: False
522 Inherited: False
523 Property Type: String
524 Cardinality: Single
525 Updatability: Read Write
526
527 **icom_core:createdBy**
528 Description: An actor who created an entity.
529 Required: False
530 Inherited: False

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

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

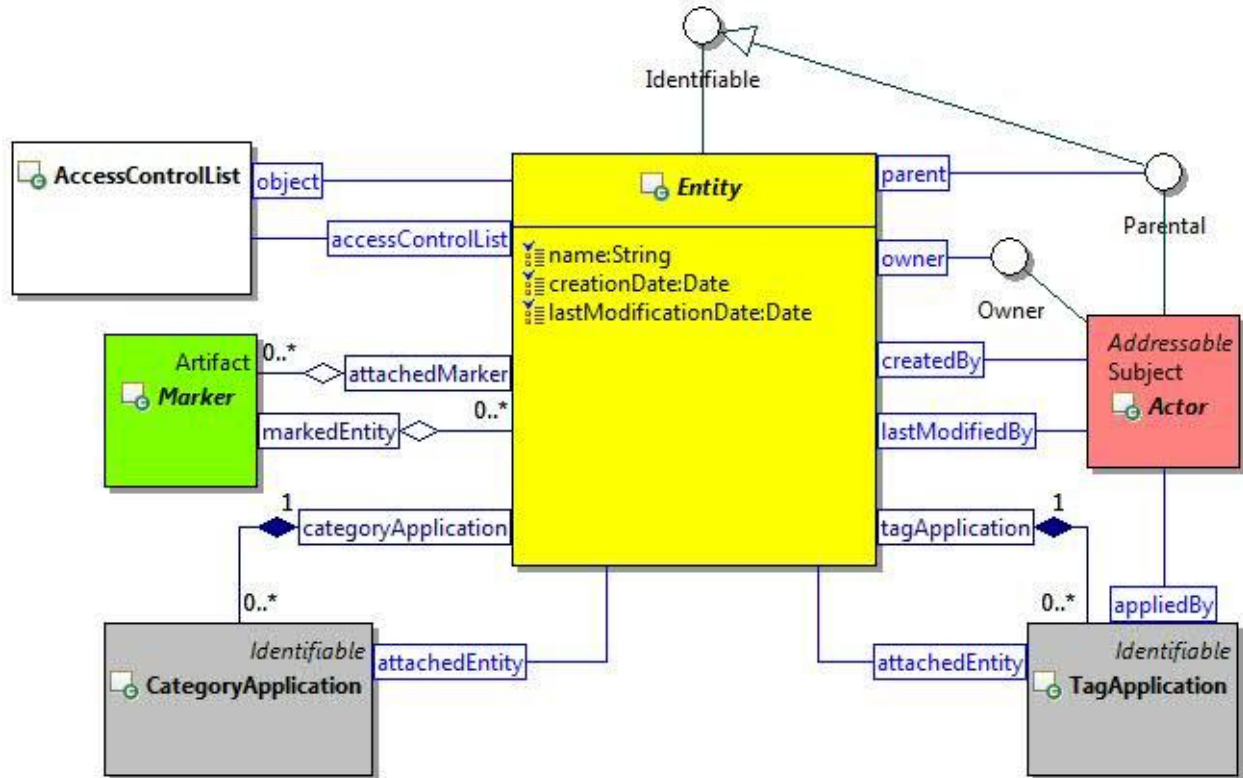


Figure 2: Entity Class Diagram.

3.1.6 EntityDefinition

3.1.6.1 Description

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

3.1.6.2 Class Definition

The EntityDefinition class has attribute values:

localNamespace

Value: icom_core

localName

Value: EntityDefinition

extendsFrom

Value: icom_core:Entity, icom_mota:RelationshipBondable

stereotype

Value: primary

isAbstract

~~Value: TRUE~~

~~description~~

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

~~propertyDefinitions~~

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

~~3.1.6.31.1.1~~ **Property Definitions**

~~The EntityDefinition class inherits property definitions from super classes.~~

~~The EntityDefinition class MUST have the property definition:~~

~~icom_core:description~~

~~Description: A description of an entity definition.~~

~~Required: False~~

~~Inherited: False~~

~~Property Type: String~~

~~Cardinality: Single~~

~~Updatability: Read Write~~

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

~~3.1.7~~ **3.1.6 Overview of Scope, Subject, and Artifact Branches**

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

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

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

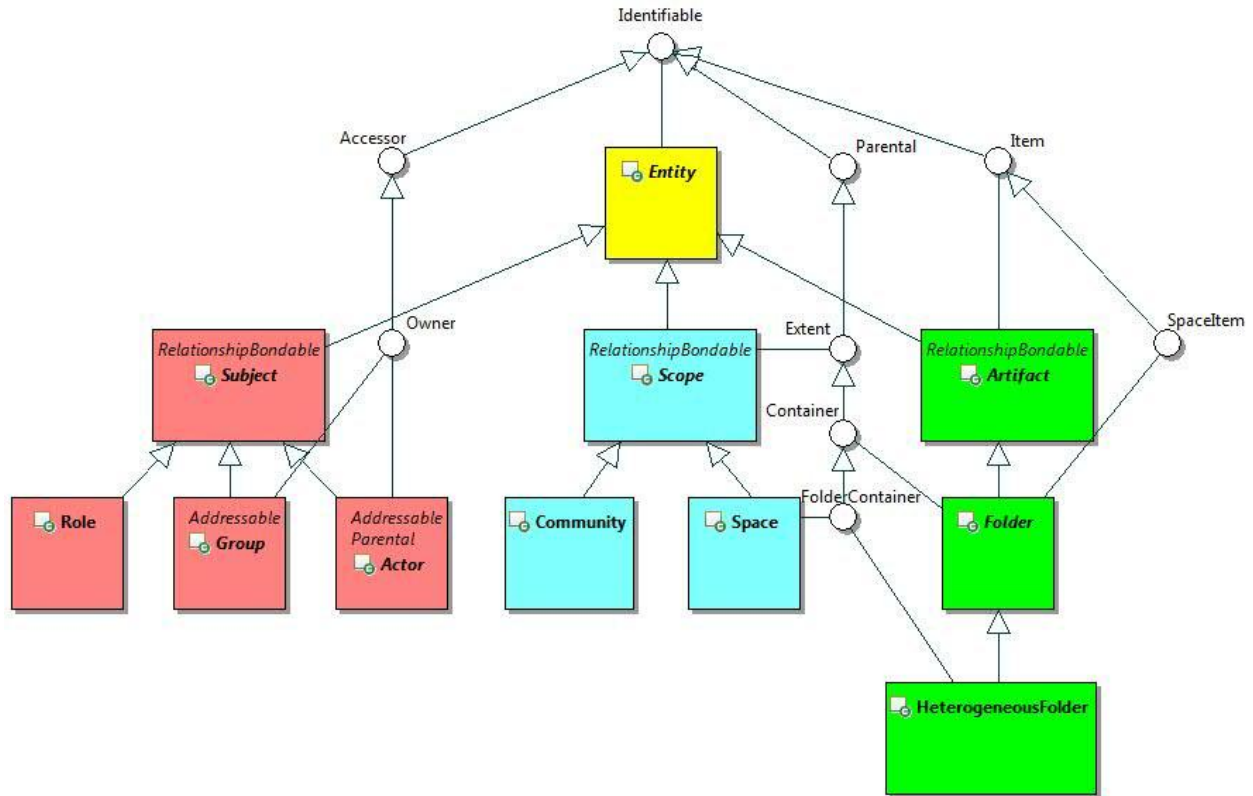


Figure 3: Scope, Subject, and Artifact Branches.

3.2 Scope Branch

3.2.1 Scope and Top-Level Subclasses

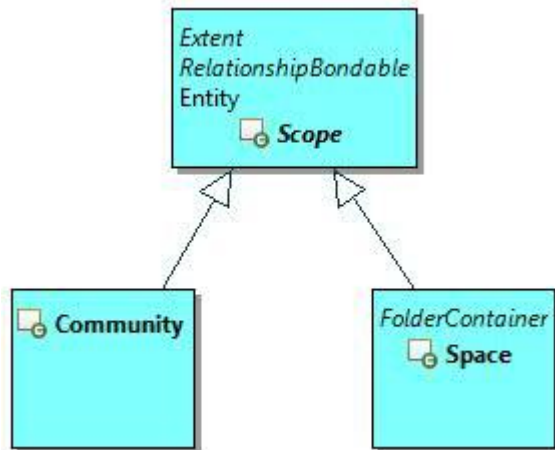


Figure 4: Scope Branch.

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

3.2.2 Scope

3.2.2.1 Description

A scope is an extent of an administrative policy.

3.2.2.2 Class Definition

The Scope class has attribute values:

localNamespace

Value: icom_core

localName

Value: Scope

extendsFrom

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

stereotype

Value: primary

isAbstract

Value: TRUE

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.2.2.3.

3.2.2.3 Property Definitions

The Scope class inherits property definitions from super classes.

The Scope class MUST have the property definitions:

icom_core:description

Description: A description of a scope.

Required: False

Inherited: False

Property Type: String

Cardinality: Single

Updatability: Read Write

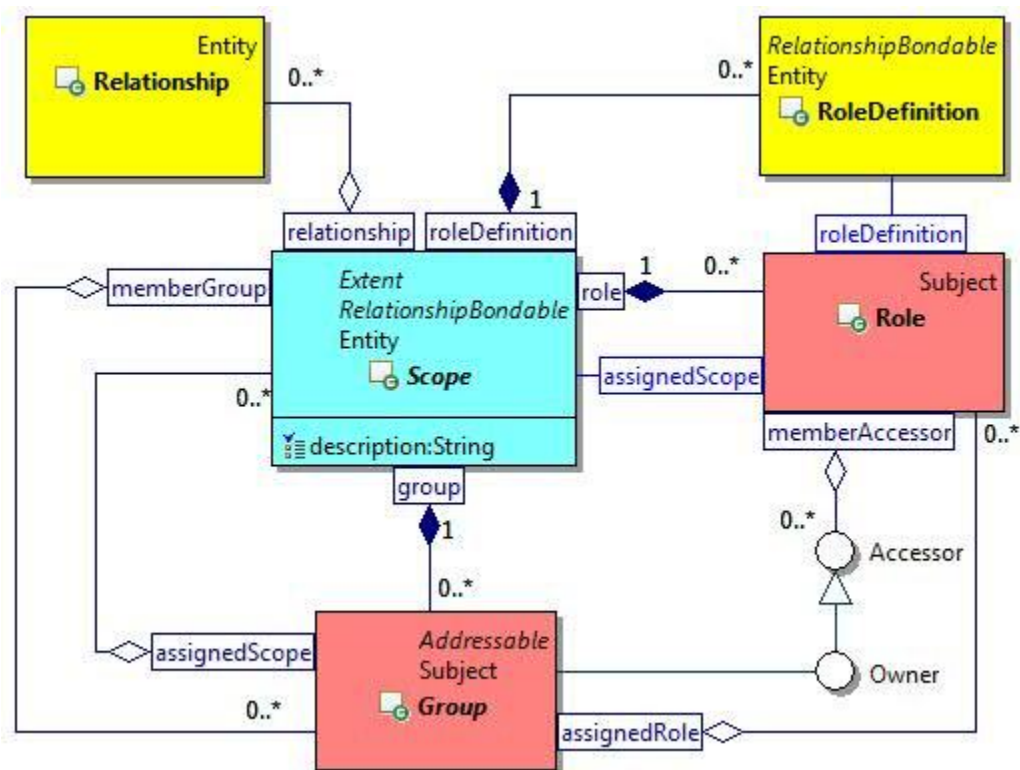
icom_core:parent

Description: A community which contains a scope.

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

756 Property Type: icom_meta:Relationship
 757 Cardinality: Multi
 758 Updatability: Read Only
 759

760 The Scope class MAY include additional property definitions which are implementation-defined.
 761



762
 763 Figure 5: Scope Class Diagram.
 764

765 3.2.3 Community

766 3.2.3.1 Description

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

770 3.2.3.2 Class Definition

771 The Community class has attribute values:

772
 773 **localNamespace**
 774 Value: icom_core

775
 776 **localName**
 777 Value: Community

778

779 **extendsFrom**

780 Value: icom_core:Scope

781

782 **stereotype**

783 Value: primary

784

785 **description**

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

787 of spaces.

788

789 **propertyDefinitions**

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

791 **3.2.3.3 Property Definitions**

792 The Community class inherits property definitions from super classes.

793 The Community class MUST have the property definitions:

794

795 **icom_core:community**

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

802

803 **icom_core:space**

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

810

811 **icom_core:actor**

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

819

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

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

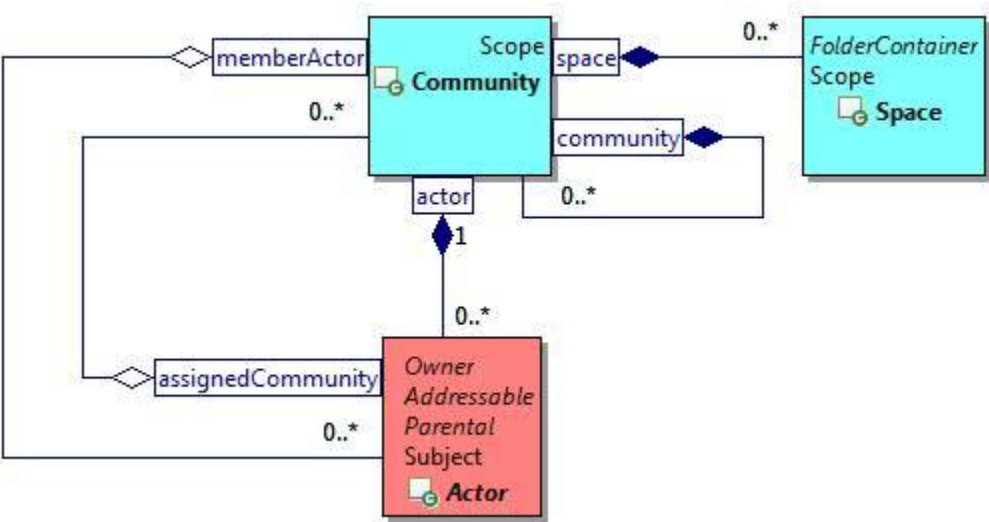


Figure 6: Community Class Diagram.

3.2.4 Space

3.2.4.1 Description

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

3.2.4.2 Class Definition

The Space class has attribute values:

localNamespace
Value: icom_core
localName
Value: Space
extendsFrom
Value: icom_core:Scope, icom_core:FolderContainer

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.2.4.3.

3.2.4.3 Property Definitions

The Space class inherits property definitions from super classes.

The Space class MUST have the property definition:

icom_core:element

Description: Elements of a space.

Required: False

Inherited: True

Property Type: icom_core:Spaceltem

Cardinality: Multi

Updatability: Read Only

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

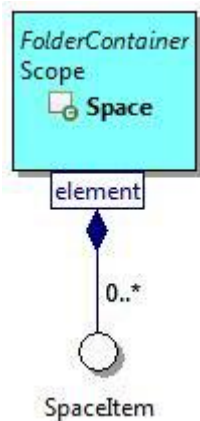


Figure 7: Space Class Diagram.

3.3 Subject Branch

3.3.1 Subject and Top-Level Subclasses

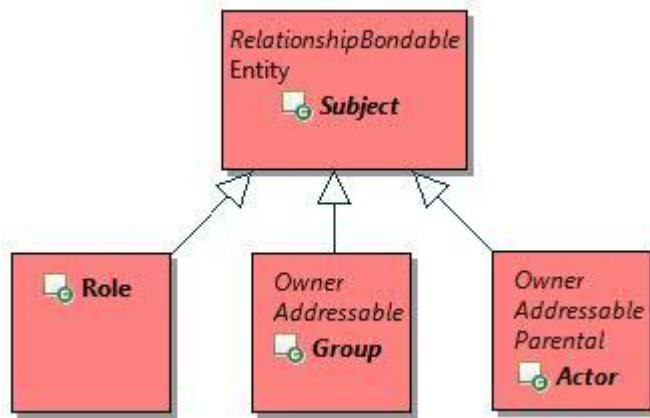


Figure 8: Subject Branch.

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

3.3.2 Subject

3.3.2.1 Description

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

3.3.2.2 Class Definition

The Subject class has attribute values:

```
localNamespace
    Value: icom_core

localName
    Value: Subject

extendsFrom
    Value: icom_core:Entity, icom_meta:RelationshipBondable

stereotype
    Value: primary

isAbstract
    Value: TRUE

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

904 **propertyDefinitions**

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

906 **3.3.2.3 Property Definitions**

907 The Subject class inherits property definitions from super classes.

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

909

910 **icom_core:description**

911 Description: A description of a subject.

912 Required: False

913 Inherited: False

914 Property Type: String

915 Cardinality: Single

916 Updatability: Read Write

917

918 **icom_core:parent**

919 Description: A scope which contains a subject.

920 Required: False

921 Inherited: True

922 Property Type: icom_core:Scope

923 Cardinality: Single

924 Updatability: Read Only

925

926 **icom_meta:relationship**

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

928 Required: False

929 Inherited: False

930 Property Type: icom_meta:Relationship

931 Cardinality: Multi

932 Updatability: Read Only

933

934 **icom_meta:property**

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

936 Required: False

937 Inherited: False

938 Property Type: icom_meta:Property

939 Cardinality: Multi

940 Updatability: Read Write

941

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

943

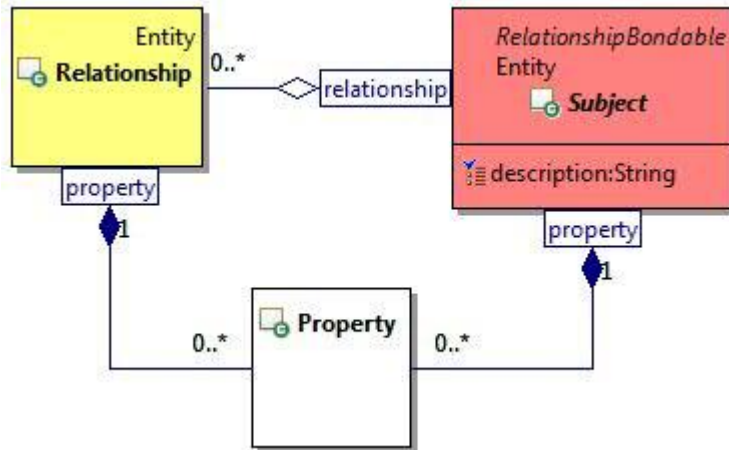


Figure 9: Subject Class Diagram.

3.3.3 Group

3.3.3.1 Description

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

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

It can be an owner of one or more entities.

3.3.3.2 Class Definition

The Group class has attribute values:

localNamespace

Value: icom_core

localName

Value: Group

extendsFrom

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

Optional Value: icom_ac:Owner

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.3.3.3.

3.3.3.3 Property Definitions

The Group class inherits property definitions from super classes.

The Group class MUST have the property definitions:

icom_core:assignedGroup

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

icom_core:assignedScope

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

icom_core:memberGroup

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

icom_core:memberActor

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

icom_ac:assignedRole

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

Updatability: Read Write

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

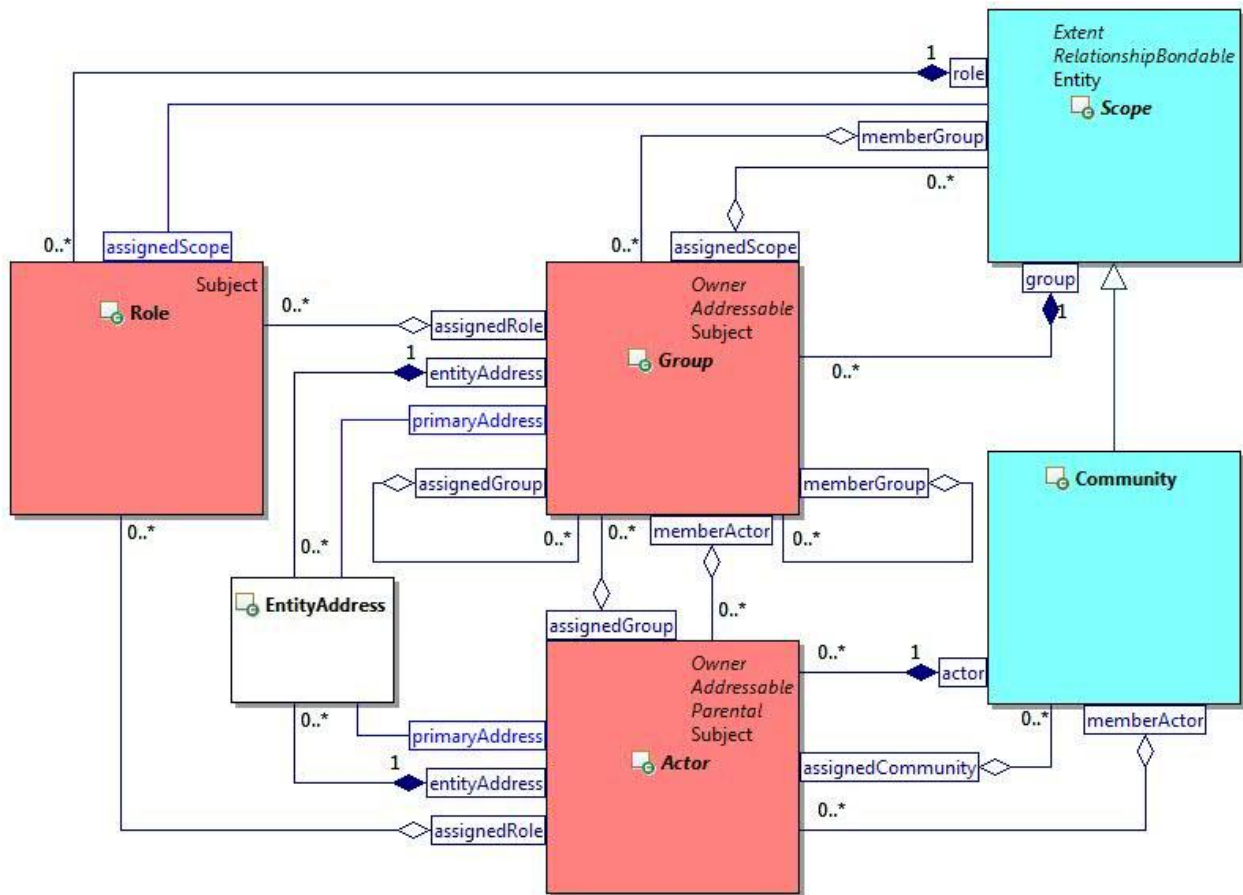


Figure 10: Group and Actor Class Diagram.

3.3.4 Actor

3.3.4.1 Description

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

It can be an owner of entities.

3.3.4.2 Class Definition

The Actor class has attribute values:

localNamespace

Value: icom_core

localName

Value: Actor

1035

1036 **extendsFrom**

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

1038

1039 **stereotype**

1040 Value: primary

1041

1042 **isAbstract**

1043 Value: TRUE

1044

1045 **description**

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

1047

1048 **propertyDefinitions**

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

1050 **3.3.4.3 Property Definitions**

1051 The Actor class inherits property definitions from super classes.

1052 The Actor class MUST have the property definitions:

1053

1054 **icom_core:parent**

1055	Description:	A community which contains an actor.
1056	Required:	False
1057	Inherited:	True
1058	Property Type:	icom_core:Community
1059	Cardinality:	Single
1060	Updatability:	Read Only

1061

1062 **icom_core:assignedGroup**

1063	Description:	An actor's groups.
1064	Required:	False
1065	Inherited:	False
1066	Property Type:	icom_core:Group
1067	Cardinality:	Multi
1068	Updatability:	Read Write

1069

1070 **icom_core:assignedCommunity**

1071	Description:	An actor's communities.
1072	Required:	False
1073	Inherited:	False
1074	Property Type:	icom_core:Community
1075	Cardinality:	Multi
1076	Updatability:	Read Write

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

1085

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

1087

1088 3.3.5 Person

1089 3.3.5.1 Description

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

1091 A person has a personal space.

1092 3.3.5.2 Class Definition

1093 The Person class has attribute values:

1094

1095 **localNamespace**

1096 Value: icom_core

1097

1098 **localName**

1099 Value: Person

1100

1101 **extendsFrom**

1102 Value: icom_core:Actor

1103

1104 **stereotype**

1105 Value: primary

1106

1107 **description**

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

1109

1110 **propertyDefinitions**

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

1112 3.3.5.3 Property Definitions

1113 The Person class inherits property definitions from super classes.

1114 The Person class MUST have the property definitions:

1115

1116 **icom_core:givenName**

1117 Description: Given name of a person.

1118 Required: False

1119 Inherited: False

1120 Property Type: String

1121 Cardinality: Single

1122 Updatability: Read Write

1123

1124 **icom_core:middleName**

1125 Description: Middle name of a person. Can include multiple names
1126 concatenated.

1127 Required: False

1128 Inherited: False

1129 Property Type: String

1130 Cardinality: Single

1131 Updatability: Read Write

1132

1133 **icom_core:familyName**

1134 Description: Family name of a person.

1135 Required: False

1136 Inherited: False

1137 Property Type: String

1138 Cardinality: Single

1139 Updatability: Read Write

1140

1141 **icom_core:prefix**

1142 Description: Prefix of a person's name.

1143 Required: False

1144 Inherited: False

1145 Property Type: String

1146 Cardinality: Single

1147 Updatability: Read Write

1148

1149 **icom_core:suffix**

1150 Description: Suffix of a person's name.

1151 Required: False

1152 Inherited: False

1153 Property Type: String

1154 Cardinality: Single

1155 Updatability: Read Write

1156

1157	icom_core:nickname	
1158	Description:	Nickname of a person.
1159	Required:	False
1160	Inherited:	False
1161	Property Type:	String
1162	Cardinality:	Multi
1163	Updatability:	Read Write
1164		
1165	icom_core:jobTitle	
1166	Description:	Job title of a person.
1167	Required:	False
1168	Inherited:	False
1169	Property Type:	String
1170	Cardinality:	Single
1171	Updatability:	Read Write
1172		
1173	icom_core:department	
1174	Description:	A person's affiliated department.
1175	Required:	False
1176	Inherited:	False
1177	Property Type:	String
1178	Cardinality:	Single
1179	Updatability:	Read Write
1180		
1181	icom_core:officeLocation	
1182	Description:	Location of a person's department.
1183	Required:	False
1184	Inherited:	False
1185	Property Type:	String
1186	Cardinality:	Single
1187	Updatability:	Read Write
1188		
1189	icom_core:company	
1190	Description:	A person's affiliated company.
1191	Required:	False
1192	Inherited:	False
1193	Property Type:	String
1194	Cardinality:	Single
1195	Updatability:	Read Write
1196		
1197	icom_core:profession	
1198	Description:	A person's profession.

1199	Required:	False
1200	Inherited:	False
1201	Property Type:	String
1202	Cardinality:	Single
1203	Updatability:	Read Write
1204		
1205	icom_core:personalSpace	
1206	Description:	Personal space of a person.
1207	Required:	False
1208	Inherited:	False
1209	Property Type:	icom_core:Space
1210	Cardinality:	Single
1211	Updatability:	Read Only
1212		
1213	icom_presence:presence	
1214	Description:	Presence of a person.
1215	Required:	False
1216	Inherited:	False
1217	Property Type:	icom_presence:Presence
1218	Cardinality:	Single
1219	Updatability:	Read Only
1220		
1221	icom_msg:instantMessageFeed	
1222	Description:	Instant message feed for a person.
1223	Required:	False
1224	Inherited:	False
1225	Property Type:	icom_msg:InstantMessageFeed
1226	Cardinality:	Single
1227	Updatability:	Read Only
1228		
1229	The Person class MAY include additional property definitions which are implementation-defined.	
1230		

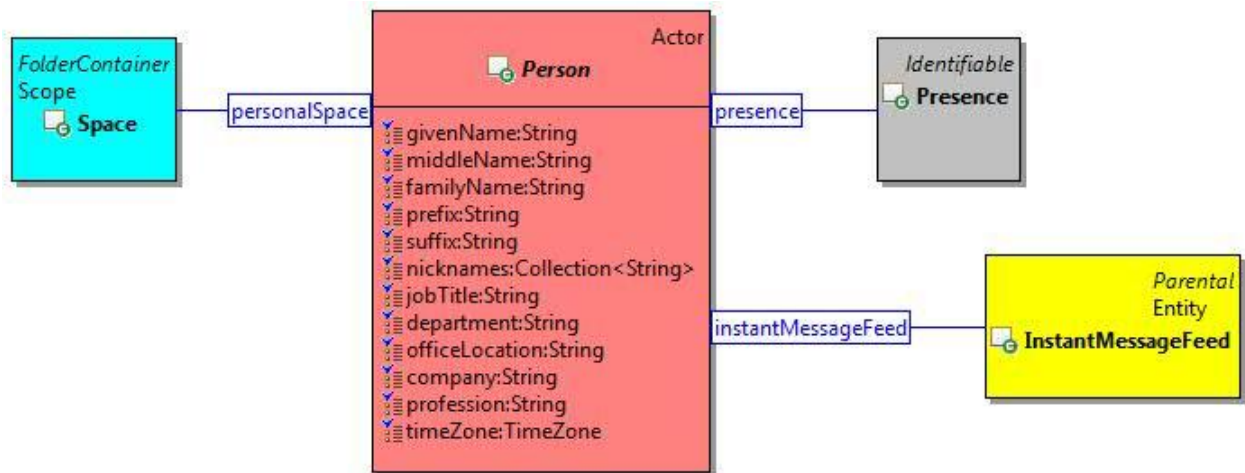


Figure 11: Person Class Diagram.

3.3.6 Resource

3.3.6.1 Description

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

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

3.3.6.2 Class Definition

The Resource class has attribute values:

localNamespace

Value: icom_core

localName

Value: Resource

extendsFrom

Value: icom_core:Actor

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.3.6.3.

3.3.6.3 Property Definitions

The Resource class inherits property definitions from super classes.

The Resource class MUST have the property definitions:

icom_core:resourceSpace

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

icom_core:location

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

icom_core:capacity

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

icom_core:resourceType

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

icom_core:bookingRule

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

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

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

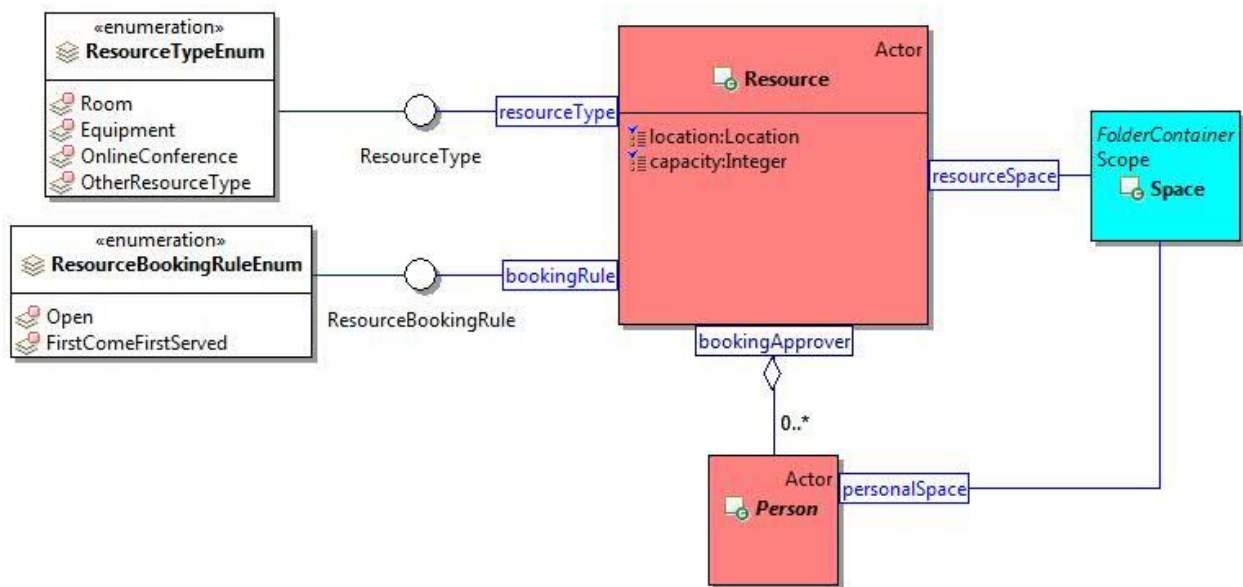


Figure 12: Resource Class Diagram.

3.3.7 Resource Type

3.3.7.1 Description

A resource type is a category of resources.

3.3.7.2 Class Definition

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

The ResourceType class has attribute values:

localNamespace

Value: icom_core

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

1342 **3.3.7.3 Property Definitions**

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

1345 **3.3.8 ResourceTypeEnum**

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

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

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

1371
1372 ICOM defines four resource types:

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

1377

1378 **3.3.9 ResourceBookingRule**

1379 **3.3.9.1 Description**

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

1381 **3.3.9.2 Class Definition**

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

1383 The ResourceBookingRule class has attribute values:

1384

1385 **localNamespace**
1386 Value: icom_core

1387

1388 **localName**
1389 Value: ResourceBookingRule

1390

1391 **extendsFrom**
1392 Value:

1393

1394 **stereotype**
1395 Value: mixin

1396

1397 **description**
1398 Value: ResourceBookingRule is a mixin class which defines a rule for allocating resources for
1399 calendar scheduling.

1400

1401 **propertyDefinitions**
1402 The values for this attribute are defined in Section 3.3.9.3.

1403 **3.3.9.3 Property Definitions**

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

1406

3.3.10 ResourceBookingRuleEnum

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

The ResourceBookingRuleEnum class has attribute values:

localNamespace

Value: icom_core

localName

Value: ResourceBookingRuleEnum

extendsFrom

Value: icom_core:ResourceBookingRule

stereotype

Value: primary

isEnumeration

Value: TRUE

description

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

instances

Value: <icom_core:Open, icom_core:FirstComeFirstServed>

ICOM defines two resource booking rules:

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

3.4 Artifact Branch

3.4.1 Artifact and Top-Level Subclasses

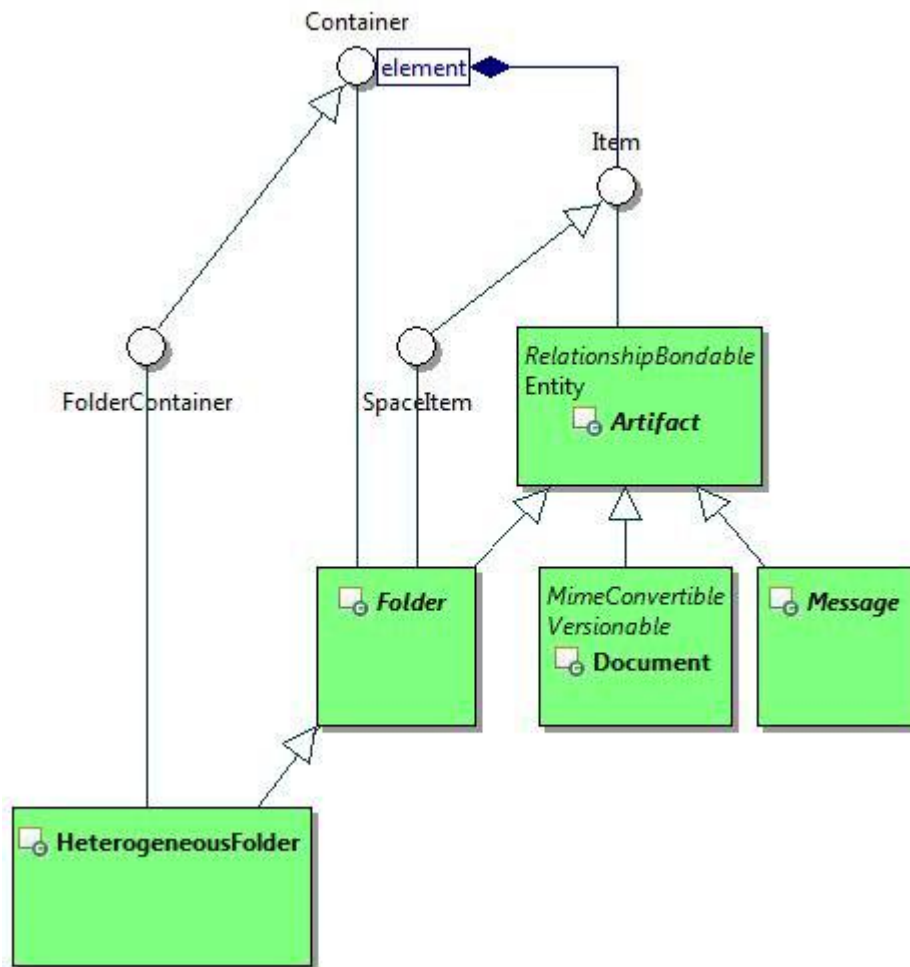


Figure 13: Artifact Branch.

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

3.4.2 Item

3.4.2.1 Description

An item is an element of a container.

The parent of an item MUST be a container.

3.4.2.2 Class Definition

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

The Item class has attribute values:

1452 **localNamespace**
1453 Value: icom_core
1454
1455 **localName**
1456 Value: Item
1457
1458 **extendsFrom**
1459 Value: icom_core:Identifiable
1460
1461 **stereotype**
1462 Value: mixin
1463
1464 **description**
1465 Value: Item is a mixin class which defines the characteristics of entities that can be placed in a
1466 Container.
1467
1468 **propertyDefinitions**
1469 The values for this attribute are defined in Section 3.4.2.3.

1470 **3.4.2.3 Property Definitions**

1471 The Item class inherits property definitions from super classes.

1472 The Item class **MUST** have the property definition:

1473
1474 **icom_core:parent**
1475 Description: A parent container of an item.
1476 Required: False
1477 Inherited: True
1478 Property Type: icom_core:Container
1479 Cardinality: Single
1480 Updatability: Read Only

1481
1482 The Item class **MAY** have the optional property definition:

1483
1484 **icom_core:container**
1485 Description: Zero, one, or more containers of an item, including the parent
1486 container.
1487 Required: False
1488 Inherited: False
1489 Property Type: icom_core:Container
1490 Cardinality: Multi
1491 Updatability: Read Write

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

1494

1495 **3.4.3 Spaceltem**

1496 **3.4.3.1 Description**

1497 A space item is an item that can be an element of a space.

1498 **3.4.3.2 Class Definition**

1499 The Spaceltem class is a mixin class which defines the characteristics of items that can be elements of a
1500 Space.

1501 The Spaceltem class has attribute values:

1502

1503 **localNamespace**

1504 Value: icom_core

1505

1506 **localName**

1507 Value: Spaceltem

1508

1509 **extendsFrom**

1510 Value: icom_core:Item

1511

1512 **stereotype**

1513 Value: mixin

1514

1515 **description**

1516 Value: Spaceltem is a mixin class which defines the characteristics of entities that can be
1517 elements of a Space.

1518

1519 **propertyDefinitions**

1520 The values for this attribute are defined in Section 3.4.3.3.

1521 **3.4.3.3 Property Definitions**

1522 The Spaceltem class inherits property definitions from super classes.

1523 The Spaceltem class MAY include additional property definitions which are implementation-defined.

1524

1525 **3.4.4 Container**

1526 **3.4.4.1 Description**

1527 A container is an extent that contains items.

1528 **3.4.4.2 Class Definition**

1529 The Container class is a mixin class which defines the characteristics of extents that contain items.

1530 The Container class has attribute values:

1531

1532 **localNamespace**

1533 Value: icom_core

1534

1535 **localName**

1536 Value: Container

1537

1538 **extendsFrom**

1539 Value: icom_core:Extent

1540

1541 **stereotype**

1542 Value: mixin

1543

1544 **description**

1545 Value: A container is an extent that contains items.

1546

1547 **propertyDefinitions**

1548 The values for this attribute are defined in Section 3.4.4.3.

1549 **3.4.4.3 Property Definitions**

1550 The Container class inherits property definitions from super classes.

1551 The Container class MUST have the property definition:

1552

1553 **icom_core:element**

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

1561

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

1563

1564 **3.4.5 FolderContainer**

1565 **3.4.5.1 Description**

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

1567 containers.

3.4.5.2 Class Definition

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

The FolderContainer class has attribute values:

localNamespace

Value: icom_core

localName

Value: FolderContainer

extendsFrom

Value: icom_core:Container

stereotype

Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.4.5.3.

3.4.5.3 Property Definitions

The FolderContainer class inherits property definitions from super classes.

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

3.4.6 Artifact

3.4.6.1 Description

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

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

3.4.6.2 Class Definition

The Artifact class has attribute values:

localNamespace

Value: icom_core

localName

Value: Artifact

1608 **extendsFrom**
1609 Value: icom_core:Entity, icom_core:Item, icom_meta:RelationshipBondable
1610 Optional Value: icom_core:SpaceItem
1611
1612 **stereotype**
1613 Value: primary
1614
1615 **isAbstract**
1616 Value: TRUE
1617
1618 **description**
1619 Value: An artifact is a result of a communication, cooperation, content creation, or collaboration
1620 activity.
1621
1622 **propertyDefinitions**
1623 The values for this attribute are defined in Section 3.4.6.3.

1624 **3.4.6.3 Property Definitions**

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

1627
1628 **icom_core:description**
1629 Description: A description of an artifact.
1630 Required: False
1631 Inherited: False
1632 Property Type: String
1633 Cardinality: Single
1634 Updatability: Read Write
1635
1636 **icom_core:userCreationDate**
1637 Description: Date and time when an artifact was created.
1638 Required: False
1639 Inherited: False
1640 Property Type: DateTime
1641 Cardinality: Single
1642 Updatability: Read Write
1643
1644 **icom_core:userLastModificationDate**
1645 Description: Date and time when an artifact was last modified.
1646 Required: False
1647 Inherited: False
1648 Property Type: DateTime
1649 Cardinality: Single

1650	Updatability:	Read Write
1651		
1652	icom_meta:property	
1653	Description:	Zero or more extended properties of an artifact.
1654	Required:	False
1655	Inherited:	False
1656	Property Type:	icom_meta:Property
1657	Cardinality:	Multi
1658	Updatability:	Read Write
1659		
1660	icom_meta:viewerProperty	
1661	Description:	Zero or more extended properties of an artifact visible to a viewer.
1662		
1663	Required:	False
1664	Inherited:	False
1665	Property Type:	icom_meta:Property
1666	Cardinality:	Multi
1667	Updatability:	Read Write
1668		
1669	icom_meta:relationship	
1670	Description:	Zero or more relationships associated with an artifact.
1671	Required:	False
1672	Inherited:	False
1673	Property Type:	icom_meta:Relationship
1674	Cardinality:	Multi
1675	Updatability:	Read Only
1676		
1677	The Artifact class MAY include additional property definitions which are implementation-defined.	
1678		

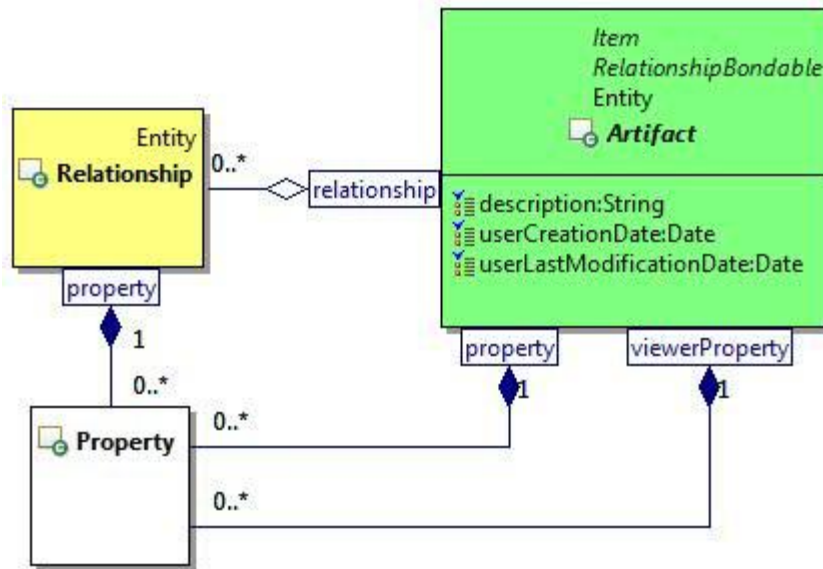


Figure 14: Artifact Class Diagram.

3.4.7 Folder

3.4.7.1 Description

A folder is an artifact that may contain other artifacts.

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

3.4.7.2 Class Definition

The Folder class has attribute values:

localNamespace

Value: icom_core

localName

Value: Folder

extendsFrom

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

stereotype

Value: primary

isAbstract

Value: TRUE

1705 **description**
1706 Value: A folder is an artifact that may contain other artifacts.
1707
1708 **propertyDefinitions**
1709 The values for this attribute are defined in Section 3.4.7.3.

1710 **3.4.7.3 Property Definitions**

1711 The Folder class inherits property definitions from super classes.
1712 The Folder class MUST have the property definition:

1713
1714 **icom_core:parent**
1715 Description: A parent container of a folder.
1716 Required: False
1717 Inherited: True
1718 Property Type: icom_core:FolderContainer
1719 Cardinality: Single
1720 Updatability: Read Only

1721
1722 The Folder class MAY include additional property definitions which are implementation-defined.
1723

1724 **3.4.8 HeterogeneousFolder**

1725 **3.4.8.1 Description**

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

1728 **3.4.8.2 Class Definition**

1729 The HeterogeneousFolder class has attribute values:

1730
1731 **localNamespace**
1732 Value: icom_core
1733
1734 **localName**
1735 Value: HeterogeneousFolder
1736
1737 **extendsFrom**
1738 Value: icom_core:Folder, icom_core:FolderContainer
1739
1740 **stereotype**
1741 Value: primary
1742

description
Value: A heterogeneous folder is an unconstrained folder to contain any type of artifacts.
propertyDefinitions
The values for this attribute are defined in Section 3.4.8.3.

3.4.8.3 Property Definitions

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

icom_core:element	
Description:	Elements of a heterogeneous folder.
Required:	False
Inherited:	True
Property Type:	icom_core:Artifact
Cardinality:	Multi
Updatability:	Read Only

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

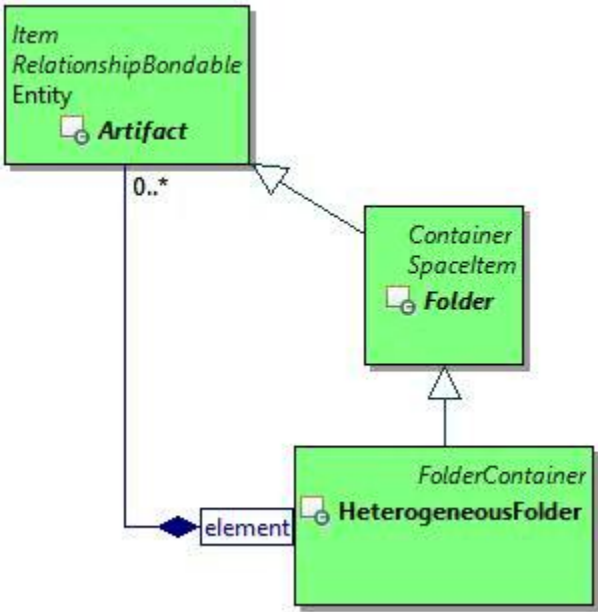


Figure 15: Heterogeneous Folder Class Diagram.

3.5 Access Control Model

3.5.1 Accessor

3.5.1.1 Description

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

3.5.1.2 Class Definition

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

The Accessor class has attribute values:

localNamespace

Value: icom_ac

localName

Value: Accessor

extendsFrom

Value: icom_core:Identifiable

stereotype

Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.5.1.3.

3.5.1.3 Property Definitions

The Accessor class inherits property definitions from super classes.

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

3.5.2 Owner

3.5.2.1 Description

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

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

3.5.2.2 Class Definition

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

The Owner class has attribute values:

localNamespace

Value: icom_ac

localName

Value: Owner

extendsFrom

Value: icom_ac:Accessor

stereotype

Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.5.2.3.

3.5.2.3 Property Definitions

The Owner class inherits property definitions from super classes.

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

3.5.3 RoleDefinition

3.5.3.1 Description

A role definition is a named set of privileges.

3.5.3.2 Class Definition

The RoleDefinition class has attribute values:

localNamespace

Value: icom_ac

localName

Value: RoleDefinition

1842 **extendsFrom**
1843 Value: icom_core:~~EntityDefinition~~Entity, icom_meta:RelationshipBondable
1844
1845 **stereotype**
1846 Value: primary
1847
1848 **description**
1849 Value: A role definition is a named set of privileges.
1850
1851 **propertyDefinitions**
1852 The values for this attribute are defined in Section 3.5.3.3.

1853 **3.5.3.3 Property Definitions**

1854 The RoleDefinition class inherits property definitions from super classes.
1855 The RoleDefinition class MUST have the property definition:

1856

<u>icom_core:description</u>	
<u>Description:</u>	<u>A description of a role definition.</u>
<u>Required:</u>	<u>False</u>
<u>Inherited:</u>	<u>False</u>
<u>Property Type:</u>	<u>String</u>
<u>Cardinality:</u>	<u>Single</u>
<u>Updatability:</u>	<u>Read Write</u>

1864

icom_ac:privilege	
Description:	A set of privileges.
Required:	True
Inherited:	False
Property Type:	icom_ac:Privilege
Cardinality:	Multi
Updatability:	Read Write

1872
1873 The RoleDefinition class MAY include additional property definitions which are implementation-defined.
1874

1875 **3.5.4 Role**

1876 **3.5.4.1 Description**

1877 A role assigns a named set of privileges to a set of accessors for operations within an assigned scope.

1878 **3.5.4.2 Class Definition**

1879 The Role class has attribute values:
1880

1881 **localNamespace**
1882 Value: icom_ac
1883
1884 **localName**
1885 Value: Role
1886
1887 **extendsFrom**
1888 Value: icom_core:Subject
1889
1890 **stereotype**
1891 Value: primary
1892
1893 **description**
1894 Value: A role assigns a named set of rights to a set of accessors for operations within an
1895 assigned scope.
1896
1897 **propertyDefinitions**
1898 The values for this attribute are defined in Section 3.5.4.3.

1899 **3.5.4.3 Property Definitions**

1900 The Role class inherits property definitions from super classes.

1901 The Role class **MUST** have the property definitions:

1902
1903 **icom_ac:roleDefinition**
1904 Description: A role definition containing a set of privileges.
1905 Required: True
1906 Inherited: False
1907 Property Type: icom_ac:RoleDefinition
1908 Cardinality: Single
1909 Updatability: On Create

1910
1911 **icom_ac:assignedScope**
1912 Description: A scope in which a role is assigned.
1913 Required: True
1914 Inherited: False
1915 Property Type: icom_core:Scope
1916 Cardinality: Single
1917 Updatability: Read Write

1918
1919 **icom_ac:memberAccessor**
1920 Description: Accessors (actors and groups) assigned to a role.
1921 Required: False
1922 Inherited: False

Property Type:	icom_ac:Accessor
Cardinality:	Multi
Updatability:	Read Write

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

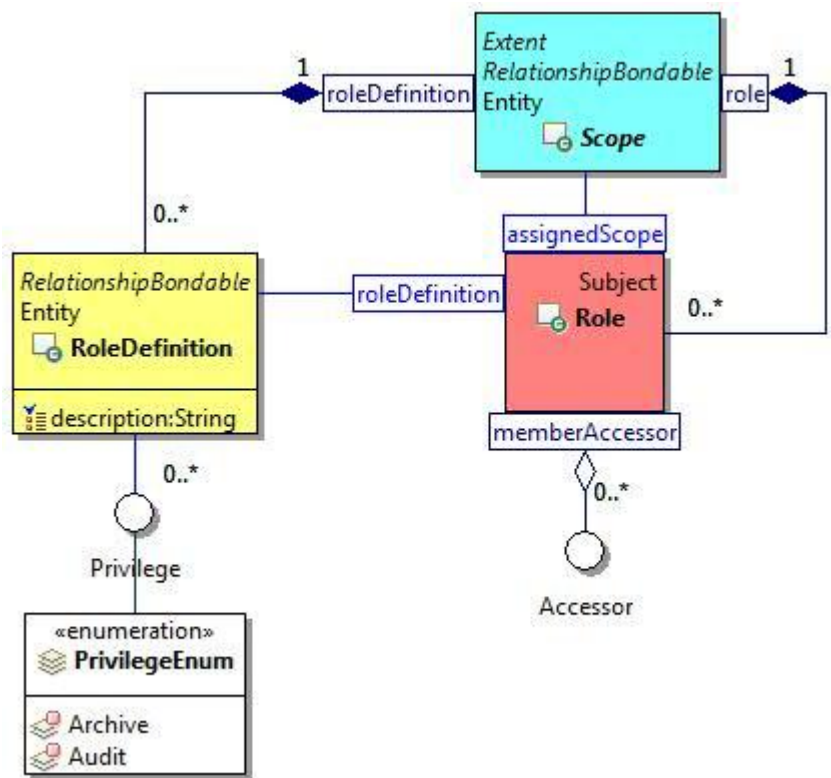


Figure 16: Role Definition and Role Class Diagram.

3.5.5 Privilege

3.5.5.1 Description

A privilege is an access right granted through roles.

3.5.5.2 Class Definition

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

```

localNamespace
  Value: icom_ac

```

```

localName
  Value: Privilege

```


1945 **extendsFrom**
 1946 Value:
 1947
 1948 **stereotype**
 1949 Value: mixin
 1950
 1951 **description**
 1952 Value: Privilege is a mixin class which defines access rights that can be included in role
 1953 definitions.
 1954
 1955 **propertyDefinitions**
 1956 The values for this attribute are defined in Section 3.5.5.3.

1957 3.5.5.3 Property Definitions

1958 The Privilege class MAY include additional property definitions which are implementation-defined.
 1959

1960 3.5.6 PrivilegeEnum

1961 The PrivilegeEnum class is an enum class that enumerates the instances each of which expresses a
 1962 privilege that can be assigned to a role.
 1963 The PrivilegeEnum class has attribute values:

1964
 1965 **localNamespace**
 1966 Value: icom_ac
 1967
 1968 **localName**
 1969 Value: PrivilegeEnum
 1970
 1971 **extendsFrom**
 1972 Value: icom_ac:Privilege
 1973
 1974 **stereotype**
 1975 Value: primary
 1976
 1977 **isEnumeration**
 1978 Value: TRUE
 1979
 1980 **description**
 1981 Value: Privilege that can be assigned to a role.
 1982
 1983 **instances**
 1984 Value: <icom_ac:Archive, icom_ac:Audit>
 1985

- 1986
- ICOM defines two privileges:
- 1987
- **icom_ac:Archive** a right to archive contents in a scope.
- 1988
- **icom_ac:Audit** a right to audit activities in a scope.
- 1989

1990

3.5.7 AccessControlList

1991

3.5.7.1 Description

1992

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

1993

entity.

1994

3.5.7.2 Class Definition

1995

The AccessControlList class has attribute values:

1996

1997	localNamespace	
1998	Value:	icom_ac
1999		
2000	localName	
2001	Value:	AccessControlList
2002		
2003	extendsFrom	
2004	Value:	
2005		
2006	stereotype	
2007	Value:	primary
2008		
2009	description	
2010	Value:	An access control list (ACL) is an object attached to an entity to specify a list of
2011		permissions to access the entity.
2012		
2013	propertyDefinitions	
2014		The values for this attribute are defined in Section 3.5.7.3.

2015

3.5.7.3 Property Definitions

2016

The AccessControlList class MUST have the property definitions:

2017

2018	icom_ac:object	
2019	Description:	Associated object.
2020	Required:	True
2021	Inherited:	False
2022	Property Type:	icom_core:Entity
2023	Cardinality:	Single
2024	Updatability:	On Create
2025		

2026 **icom_ac:accessControlEntry**

2027 Description: One or more access control entries.

2028 Required: True

2029 Inherited: False

2030 Property Type: icom_ac:AccessControlEntry

2031 Cardinality: Multi

2032 Updatability: Read Write

2033

2034 AccessControlList class MAY include additional property definitions which are implementation-defined.

2035

2036 **3.5.8 AccessControlEntry**

2037 **3.5.8.1 Description**

2038 An access control entry specifies access types granted to or denied for an accessor.

2039 **3.5.8.2 Class Definition**

2040 The AccessControlEntry class has attribute values:

2041

2042 **localNamespace**

2043 Value: icom_ac

2044

2045 **localName**

2046 Value: AccessControlEntry

2047

2048 **extendsFrom**

2049 Value:

2050

2051 **stereotype**

2052 Value: primary

2053

2054 **description**

2055 Value: An access control entry is associated with an accessor and contains a list of access

2056 types (permissions) granted to or denied from the accessor.

2057

2058 **propertyDefinitions**

2059 The values for this attribute are defined in Section 3.5.8.3.

2060 **3.5.8.3 Property Definitions**

2061 The AccessControlEntry class MUST have the property definitions:

2062

2063 **icom_ac:subject**

2064 Description: Associated subject.

2065 Required: True

2066	Inherited:	False
2067	Property Type:	icom_ac:Accessor
2068	Cardinality:	Single
2069	Updatability:	On Create

2070

2071 **icom_ac:grant**

2072	Description:	One or more access types granted to a subject.
2073	Required:	False
2074	Inherited:	False
2075	Property Type:	icom_ac:AccessType
2076	Cardinality:	Multi
2077	Updatability:	Read Write

2078

2079 **icom_ac:deny**

2080	Description:	One or more access type denied for a subject.
2081	Required:	False
2082	Inherited:	False
2083	Property Type:	icom_ac:AccessType
2084	Cardinality:	Multi
2085	Updatability:	Read Write

2086

2087 The AccessControlEntry class MAY include additional property definitions which are implementation-
2088 defined.

2089

2090 **3.5.9 AccessType**

2091 An AccessType is an access right granted through an access control entry.

2092 **3.5.9.1 Class Definition**

2093 The AccessType class is a mixin class which defines access rights that can be granted or denied in an
2094 access control entry.

2095 The AccessType class has attribute values:

2096

2097 **localNamespace**

2098 Value: icom_ac

2099

2100 **localName**

2101 Value: AccessType

2102

2103 **extendsFrom**

2104 Value:

2105

2106 | **stereotype**
2107 | Value: mixin
2108 |
2109 | **description**
2110 | Value: AccessType is a mixin class which defines access rights that can be granted or denied in
2111 | an access control entry.
2112 |
2113 | **propertyDefinitions**
2114 | The values for this attribute are defined in Section 3.5.9.2.

2115 3.5.9.2 Property Definitions

2116 | The AccessType class inherits property definitions from super classes.
2117 | The AccessType class MAY include additional property definitions which are implementation-defined.
2118 |

2119 3.5.10 AccessTypeEnum

2120 | The AccessTypeEnum class is an enum class that enumerates the instances each of which expresses an
2121 | access type that can be granted or denied in an access control entry.
2122 | The AccessTypeEnum class has attribute values:

2123 |
2124 | **localNamespace**
2125 | Value: icom_ac
2126 |
2127 | **localName**
2128 | Value: AccessTypeEnum
2129 |
2130 | **extendsFrom**
2131 | Value: icom_ac:AccessType
2132 |
2133 | **stereotype**
2134 | Value: primary
2135 |
2136 | **isEnumeration**
2137 | Value: TRUE
2138 |
2139 | **description**
2140 | Value: Access type that can be granted or denied in an access control entry.
2141 |
2142 | **instances**
2143 | Value: <icom_ac:Read, icom_ac:Write, icom_ac>Delete>
2144 |
2145 | ICOM defines three access types:
2146 | • **icom_ac:Read** a right to retrieve an entity.

- **icom_ac:Write** a right to update an entity.
- **icom_ac:Delete** a right to delete an entity.

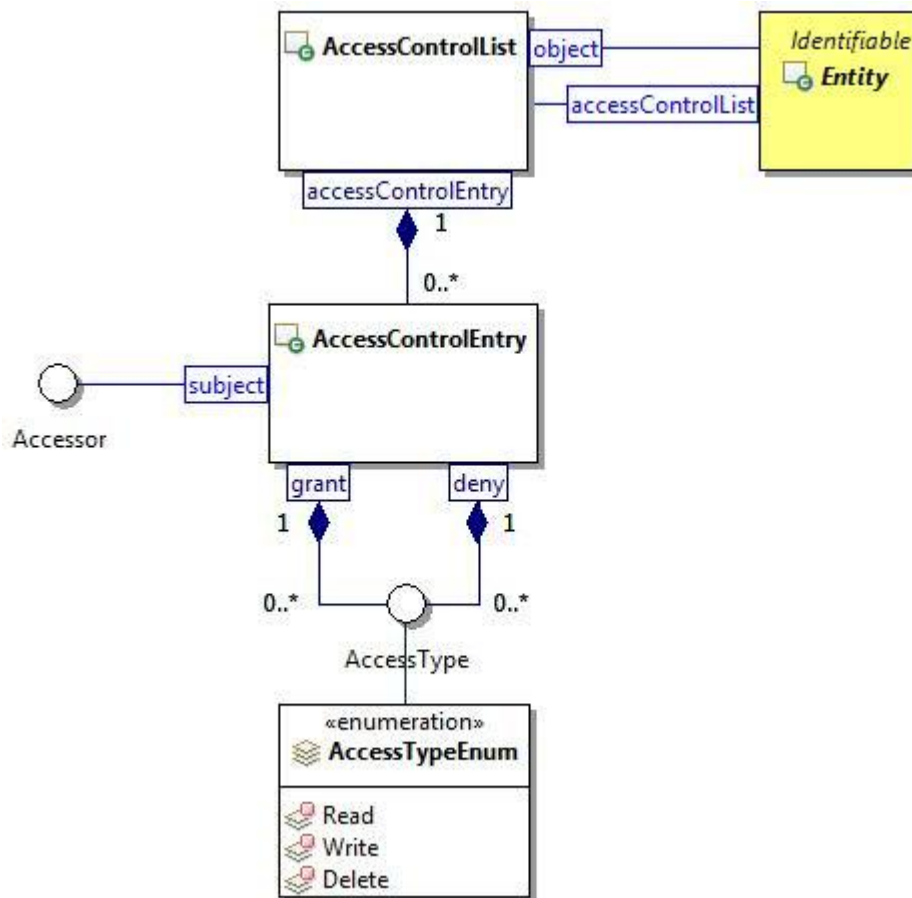


Figure 17: Access Control List Class Diagram.

3.6 Metadata Model

3.6.1 ClassDefinition

3.6.1.1 Description

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

3.6.1.2 Class Definition

The ClassDefinition class has attribute values:

localNamespace

Value: icom_meta

localName

Value: ClassDefinition

2165
 2166 extendsFrom
 2167 Value: icom_core:Entity, icom_meta:RelationshipBondable
 2168
 2169 stereotype
 2170 Value: primary
 2171
 2172 isAbstract
 2173 Value: FALSE
 2174
 2175 description
 2176 Value: A class definition defines a type of entities.
 2177
 2178 propertyDefinitions
 2179 The values for this attribute are defined in Section 3.6.1.3.

2180 3.6.1.3 Property Definitions

2181 The ClassDefinition class inherits property definitions from super classes.
 2182 The ClassDefinition class MUST have the property definition:

2184 icom_core:namespace

Description:	Namespace for a class name.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

2192 icom_core:description

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

2200 icom_meta:extendsFrom

Description:	One or more generalizations of a class.
Required:	True
Inherited:	False
Property Type:	icom_meta:ClassDefinition
Cardinality:	Multi
Updatability:	Read Write

2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249

icom meta:stereoType

<u>Description:</u>	<u>Stereo type of a class.</u>
<u>Required:</u>	<u>True</u>
<u>Inherited:</u>	<u>False</u>
<u>Property Type:</u>	<u>icom meta:StereoType</u>
<u>Cardinality:</u>	<u>Single</u>
<u>Updatability:</u>	<u>Read Write</u>

icom meta:abstract

<u>Description:</u>	<u>Indicates whether a class is abstract or concrete.</u>
<u>Required:</u>	<u>False</u>
<u>Inherited:</u>	<u>False</u>
<u>Property Type:</u>	<u>Boolean</u>
<u>Cardinality:</u>	<u>Single</u>
<u>Updatability:</u>	<u>Read Write</u>

icom meta:enumeration

<u>Description:</u>	<u>Indicates whether instances of a class are enumerated. This property is applicable only if the stereo type property is primary.</u>
<u>Required:</u>	<u>False</u>
<u>Inherited:</u>	<u>False</u>
<u>Property Type:</u>	<u>Boolean</u>
<u>Cardinality:</u>	<u>Single</u>
<u>Updatability:</u>	<u>Read Write</u>

icom meta:instances

<u>Description:</u>	<u>Instances of an enumeration class. This property is applicable only if the enumeration property is true.</u>
<u>Required:</u>	<u>False</u>
<u>Inherited:</u>	<u>False</u>
<u>Property Type:</u>	<u>IRI</u>
<u>Cardinality:</u>	<u>Multi</u>
<u>Updatability:</u>	<u>Read Write</u>

icom meta:propertyDefinition

<u>Description:</u>	<u>One or more property definitions of a class definition.</u>
<u>Required:</u>	<u>False</u>
<u>Inherited:</u>	<u>False</u>
<u>Property Type:</u>	<u>icom meta:PropertyDefinition</u>
<u>Cardinality:</u>	<u>Multi</u>
<u>Updatability:</u>	<u>Read Write</u>

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

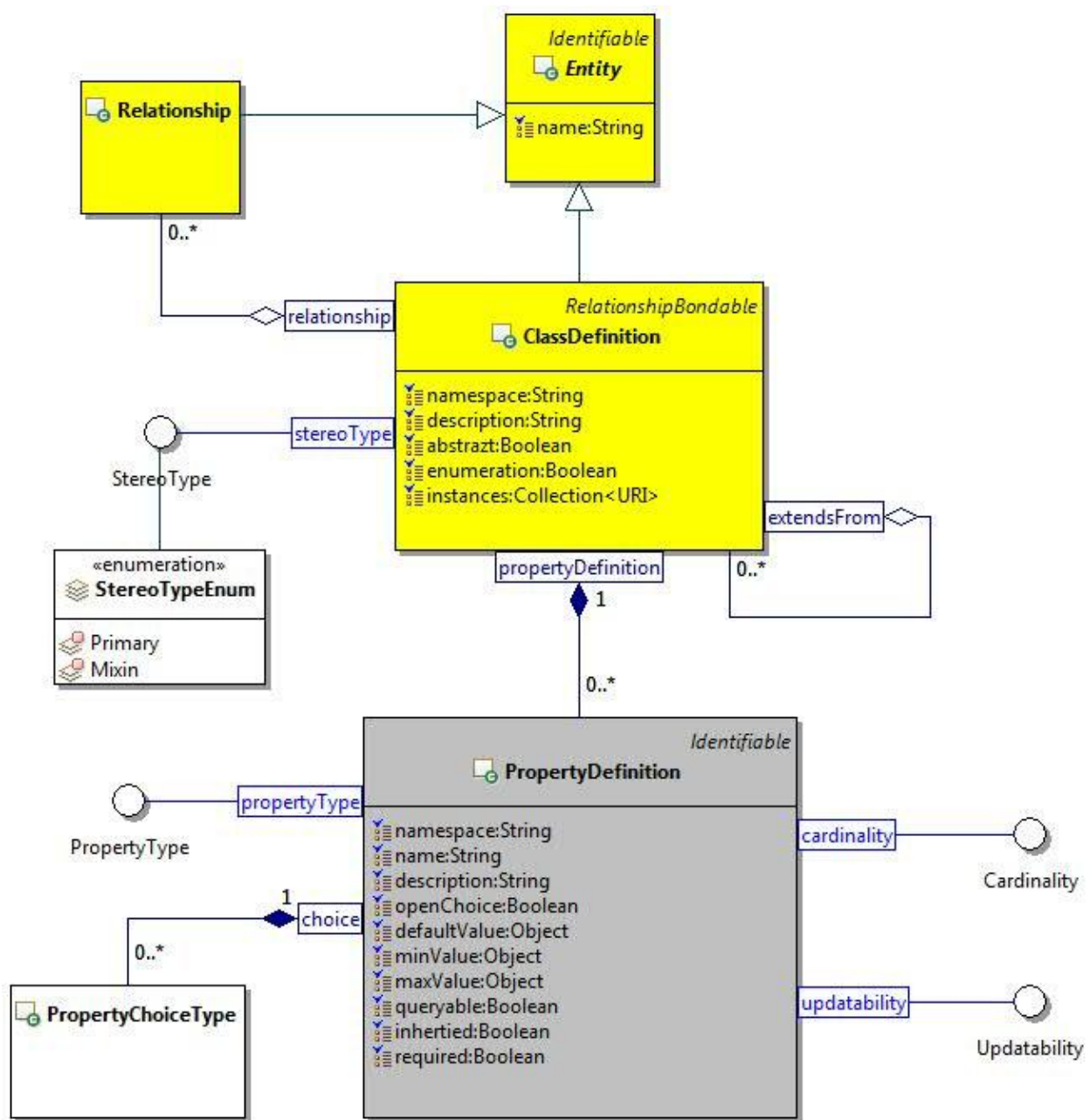


Figure 18: Class Definition and Property Definition Class Diagram.

3.6.2 Stereotype

3.6.2.1 Description

A stereo type of a class definition.

3.6.2.2 Class Definition

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

The Stereotype class has attribute values:

localNamespace

Value: icom meta

localName

Value: Stereotype

extendsFrom

Value:

stereotype

Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.6.2.3.

3.6.2.3 Property Definitions

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

3.6.3 StereotypeEnum

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

The StereotypeEnum has attribute values:

localNamespace

Value: icom meta

localName

Value: StereotypeEnum

extendsFrom

Value: icom meta:Stereotype

stereotype

Value: primary

2300 isEnumeration

2301 Value: TRUE

2302

2303 description

2304 Value: Stereo type of a class definition.

2305

2306 instances

2307 Value: <icom_meta:Primary, icom_meta:Mixin>

2308

2309 ICOM defines two stereo types:

2310 • icom_meta:Primary a primary class.

2311 • icom_meta:Mixin a mixin class.

2312

2313 ~~3.6.13.6.4~~ **PropertyDefinition**

2314 ~~3.6.1.13.6.4.1~~ **Description**

2315 A property definition specifies the name, type, choice, and cardinality of values for properties.

2316 ~~3.6.1.23.6.4.2~~ **Class Definition**

2317 The PropertyDefinition class has attribute values:

2318

2319 **localNamespace**

2320 Value: icom_meta

2321

2322 **localName**

2323 Value: PropertyDefinition

2324

2325 **extendsFrom**

2326 Value: icom_core:Identifiable

2327

2328 **stereotype**

2329 Value: primary

2330

2331 **description**

2332 Value: A property definition specifies the name, type, choice, and cardinality of values for
2333 properties.

2334

2335 **propertyDefinitions**

2336 The values for this attribute are defined in Section 3.6.4.3.

2337 ~~3.6.1.33.6.4.3~~ **Property Definitions**

2338 The PropertyDefinition class inherits property definitions from super classes.

2339 | The PropertyDefinition class MUST have the property definitions:

2340

2341 **icom_core:namespace**

2342 Description: Namespace for a property name.

2343 Required: False

2344 Inherited: False

2345 Property Type: String

2346 Cardinality: Single

2347 Updatability: Read Write

2348

2349 **icom_core:name**

2350 Description: Name for a property.

2351 Required: True

2352 Inherited: False

2353 Property Type: String

2354 Cardinality: Single

2355 Updatability: Read Write

2356

2357 **icom_core:description**

2358 Description: A description of a property definition.

2359 Required: False

2360 Inherited: False

2361 Property Type: String

2362 Cardinality: Single

2363 Updatability: Read Write

2364

2365 **icom_meta:propertyType**

2366 Description: Type of a property.

2367 | Required: ~~True~~False

2368 Inherited: False

2369 Property Type: icom_meta:PropertyType

2370 Cardinality: Single

2371 Updatability: On Create

2372 Choices: {PropertyChoiceType}

2373 Open Choice: False

2374

2375 Note: The notation {PropertyChoiceType} represents a set of PropertyChoiceType.

2376

2377 **icom_meta:defaultValue**

2378 Description: A default value for a property.

2379 Required: False

2380 Inherited: False

2381	Property Type:	property-type
2382	Cardinality:	Single
2383	Updatability:	Read Write
2384		
2385	icom_meta:choice	
2386	Description:	An allowed value for a property.
2387	Required:	False
2388	Inherited:	False
2389	Property Type:	icom_meta:PropertyChoiceType
2390	Cardinality:	Multi
2391	Updatability:	Read Write
2392		
2393	icom_meta:openChoice	
2394	Description:	Indicates whether value of the property must be listed among
2395		the choices.
2396	Required:	False
2397	Inherited:	False
2398	Property Type:	Boolean
2399	Cardinality:	Single
2400	Updatability:	Read Write
2401		
2402	icom_meta:inherited	
2403	Description:	Indicates whether a property definition is inherited from a
2404		super class.
2405	Required:	False
2406	Inherited:	False
2407	Property Type:	Boolean
2408	Cardinality:	Single
2409	Updatability:	Read Write
2410		
2411	icom_meta:required	
2412	Description:	Indicates whether a property value must be provided. It is
2413		applicable only when the updatability of the property is read-
2414		write or on-create.
2415	Required:	True
2416	Inherited:	False
2417	Property Type:	Boolean
2418	Cardinality:	Single
2419	Updatability:	Read Write
2420		
2421	icom_meta:updatability	
2422	Description:	Updatability of a property specifying under what
2423		circumstances the property value can be updated.
2424	Required:	True

2425	Inherited:	False
2426	Property Type:	icom_meta:Updatability
2427	Cardinality:	Single
2428	Updatability:	On Create
2429		
2430	icom_meta:cardinality	
2431	Description:	Cardinality of a property specifying whether the property can
2432		have “zero or one” or “zero or more” values.
2433	Required:	True
2434	Inherited:	False
2435	Property Type:	icom_meta:Cardinality
2436	Cardinality:	Single
2437	Updatability:	On Create
2438		
2439	icom_meta:minValue	
2440	Description:	Minimum value for an integer or decimal property.
2441	Required:	False
2442	Inherited:	False
2443	Property Type:	Integer Decimal
2444	Cardinality:	Single
2445	Updatability:	Read Write
2446		
2447	icom_meta:maxValue	
2448	Description:	Maximum value for an integer or decimal property.
2449	Required:	False
2450	Inherited:	False
2451	Property Type:	Integer Decimal
2452	Cardinality:	Single
2453	Updatability:	Read Write

2454

2455 The PropertyDefinition class MAY include additional property definitions which are implementation-

2456 defined.

2457

2458 ~~3.6.2~~3.6.5 Property

2459 ~~3.6.2.1~~3.6.5.1 Description

2460 The property holds a property value.

2461 ~~3.6.2.2~~3.6.5.2 Class Definition

2462 The Property class has attribute values:

2463		
2464	localNamespace	
2465	Value: icom_meta	

2466

2467 **localName**

2468 Value: Property

2469

2470 **extendsFrom**

2471 Value:

2472

2473 **stereotype**

2474 Value: primary

2475

2476 **description**

2477 Value: A property value.

2478

2479 **propertyDefinitions**

2480 The values for this attribute are defined in Section 3.6.5.3.

~~3.6.2~~3.6.5.3 Property Definitions

The Property class MUST have the property definitions:

icom_meta:propertyDefinition

Description:	A property definition that specifies the name, type, and cardinality of a property.
Required:	True
Inherited:	False
Property Type:	icom_meta:PropertyDefinition
Cardinality:	Single
Updatability:	On Create

icom_meta:value

Description:	A value of a property.
Required:	True
Inherited:	False
Property Type:	property-type
Cardinality:	Single
Updatability:	Read Write

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

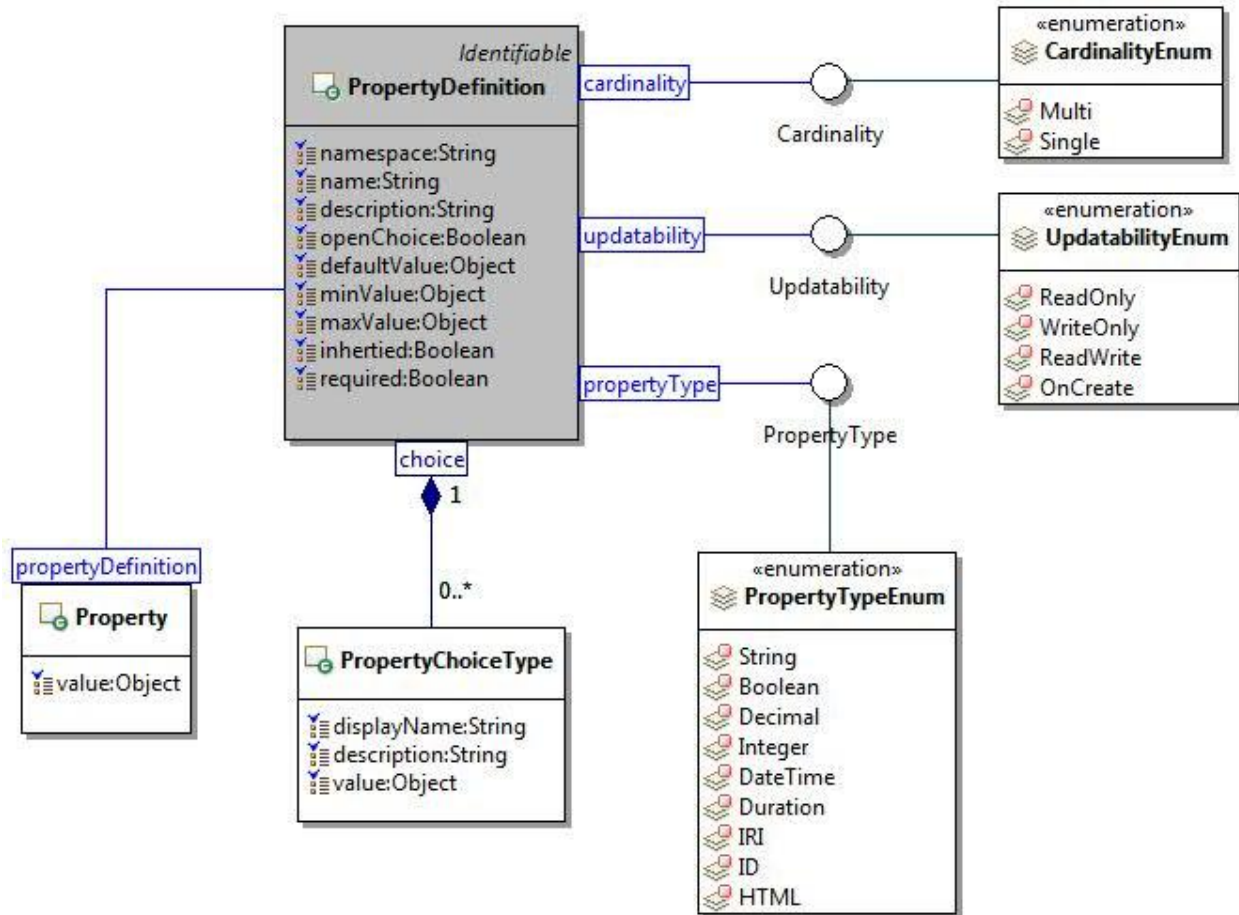


Figure 19: Property Definition and Property Class Diagram.

3.6.33.6.6 PropertyChoiceType

3.6.3.13.6.6.1 Description

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

3.6.3.23.6.6.2 Class Definition

The PropertyChoiceType class has attribute values:

localNamespace

Value: icom_meta

localName

Value: PropertyChoiceType

extendsFrom

Value:

2522 **stereotype**
2523 Value: primary
2524
2525 **description**
2526 Value: A choice for a property value.

2527
2528 **propertyDefinitions**
2529 The values for this attribute are defined Section 3.6.6.3.

2530 ~~3.6.3~~3.6.6.3 **Property Definitions**

2531 The PropertyChoiceType class MUST have the property definitions:

2532
2533 **icom_core:description**
2534 Description: A description of a property choice.
2535 Required: False
2536 Inherited: False
2537 Property Type: String
2538 Cardinality: Single
2539 Updatability: Read Write

2540
2541 **icom_meta:displayName**
2542 Description: Display name of a property choice.
2543 Required: True
2544 Inherited: False
2545 Property Type: String
2546 Cardinality: Single
2547 Updatability: Read Write

2548
2549 **icom_meta:value**
2550 Description: A value of a property choice.
2551 Required: True
2552 Inherited: False
2553 Property Type: **property-type**
2554 Cardinality: Single
2555 Updatability: Read Write

2556
2557 The PropertyChoiceType class MAY include additional property definitions which are implementation-
2558 defined.

2560 ~~3.6.4~~3.6.7 **PropertyType**

2561 A PropertyType expresses a name of a **property-type**.

3.6.4.13.6.7.1 Class Definition

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

The PropertyType class has attribute values:

localNamespace

Value: icom_meta

localName

Value: PropertyType

extendsFrom

Value:

stereotype

Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.6.7.2.

3.6.4.23.6.7.2 Property Definitions

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

3.6.53.6.8 PropertyTypeEnum

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

The PropertyTypeEnum class has attribute values:

localNamespace

Value: icom_meta

localName

Value: PropertyTypeEnum

extendsFrom

Value: icom_meta:PropertyType

stereotype

Value: primary

2603 **isEnumeration**
 2604 Value: TRUE
 2605
 2606 **description**
 2607 Value: Name of a basic data type.
 2608
 2609 **instances**
 2610 Value: <icom_meta:String, icom_meta:Boolean, icom_meta:Decimal, icom_meta:Integer,
 2611 icom_meta:Datetime, icom_meta:Duration, icom_meta:IRI, icom_meta:ID, icom_meta:HTML>
 2612

2613 ICOM defines nine data types:

- 2614 • **icom_meta:String** is equivalent to XML schema type **xsd:string**.
- 2615 • **icom_meta:Boolean** is equivalent to XML schema type **xsd:boolean**.
- 2616 • **icom_meta:Decimal** is equivalent to XML schema type **xsd:decimal**.
- 2617 • **icom_meta:Integer** is equivalent to XML schema type **xsd:integer**.
- 2618 • **icom_meta:Datetime** is equivalent to XML schema type **xsd:dateTime**.
- 2619 • **icom_meta:Duration** is equivalent to XML schema type **xsd:duration**.
- 2620 • **icom_meta:IRI** is equivalent to XML schema type **xsd:anyURI**.
- 2621 • **icom_meta:ID** opaque object identifiers.
- 2622 • **icom_meta:HTML** documents or fragments of Hypertext Markup Language (HTML) content

2623

2624 Note: ICOM uses basic data types defined by “XML Schema Part 2: Datatypes Second Edition” (W3C
 2625 Recommendation, 28 October 2004, <http://www.w3.org/TR/xmlschema-2/>).
 2626

2627 **3.6.6.9 Updatability**

2628 **3.6.6.13.6.9.1 Description**

2629 Updatability specifies under what circumstances a property value can be updated.

2630 **3.6.6.23.6.9.2 Class Definition**

2631 The Updatability class is a mixin class which specifies under what circumstances a property value can be
 2632 updated.

2633 The Updatability class has attribute values:

2634

2635 **localNamespace**
 2636 Value: icom_meta

2637

2638 **localName**
 2639 Value: Updatability

2640

2641 **extendsFrom**
 2642 Value:

2643

2644 **stereotype**
2645 Value: mixin
2646
2647 **description**
2648 Value: Updatability is a mixin class which specifies under what circumstances a property value
2649 can be updated.
2650
2651 **propertyDefinitions**
2652 The values for this attribute are defined in Section 3.6.9.3.

2653 | ~~3.6.6~~**3.6.9.3 Property Definitions**

2654 | The Updatability class MAY include additional property definitions which are implementation-defined.
2655 |

2656 | ~~3.6.7~~**3.6.10 UpdatabilityEnum**

2657 | The UpdatabilityEnum class is an enum class that enumerates instances each of which expresses the
2658 | updatability of a property.
2659 | The UpdatabilityEnum has attribute values:

2660
2661 **localNamespace**
2662 Value: icom_meta
2663
2664 **localName**
2665 Value: UpdatabilityEnum
2666
2667 **extendsFrom**
2668 Value: icom_meta:Updatability
2669
2670 | **stereotype**
2671 | Value: primary
2672
2673 **isEnumeration**
2674 Value: TRUE
2675
2676 **description**
2677 Value: Updatability of a property.
2678
2679 **instances**
2680 Value: <icom_meta:ReadOnly, icom_meta:WriteOnly, icom_meta:ReadWrite,
2681 icom_meta:OnCreate>
2682
2683 | ICOM defines four updatability types:

- 2684 | • **icom_meta:ReadOnly** a property must not be updated directly by application.
- 2685 | • **icom_meta:WriteOnly** a property can be updated but cannot be read by application.

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

~~3.6.8~~3.6.11 Cardinality

~~3.6.8.13~~3.6.11.1 Description

Cardinality specifies whether a property is single or multi valued.

~~3.6.8.23~~3.6.11.2 Class Definition

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

The Cardinality class has attribute values:

localNamespace

Value: icom_meta

localName

Value: Cardinality

extendsFrom

Value:

stereotype

Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.6.11.3.

~~3.6.8.33~~3.6.11.3 Property Definitions

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

~~3.6.9~~3.6.12 CardinalityEnum

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

The CardinalityEnum has attribute values:

localNamespace

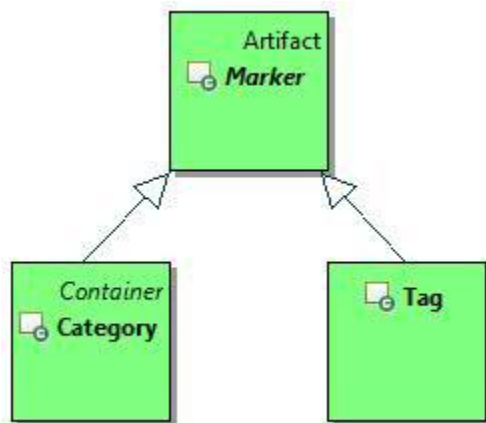
Value: icom_meta

2724 | **localName**
2725 | Value: CardinalityEnum
2726 |
2727 | **extendsFrom**
2728 | Value: icom_meta:Cardinality
2729 |
2730 | **stereotype**
2731 | Value: primary
2732 |
2733 | **isEnumeration**
2734 | Value: TRUE
2735 |
2736 | **description**
2737 | Value: Cardinality of a property.
2738 |
2739 | **instances**
2740 | Value: <icom_meta:Single, icom_meta:Multi>
2741 |

2742 | ICOM defines two cardinality types:

- 2743 | • **icom_meta:Single** a property can have zero or one value (if property is not required), or exactly
2744 | one value (if property is required).
2745 | • **icom_meta:Multi** a property can have zero or more values (if property is not required), or one or
2746 | more values (if property is required).
2747 |

2748 | **3.6.103.6.13 Marker and Subclasses**



2749 |
2750 | *Figure 20: Marker Branch.*

2751 | Figure 20 depicts the main classes of Marker Branch, which includes Marker, Category, and Tag.

~~3.6.11~~3.6.14 Marker

~~3.6.11.13~~3.6.14.1 Description

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

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

~~3.6.11.23~~3.6.14.2 Class Definition

The Marker class has attribute values:

localNamespace

Value: icom_meta

localName

Value: Marker

extendsFrom

Value: icom_core:Artifact

stereotype

Value: primary

isAbstract

Value: TRUE

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.6.14.3.

~~3.6.11.33~~3.6.14.3 Property Definitions

The Marker class inherits property definitions from super classes.

The Marker class MUST have the property definition:

icom_meta:markedEntity

Description: A marked entity.

Required: False

Inherited: False

Property Type: icom_core:Entity

Cardinality: Multi

Updatability: Read Only

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

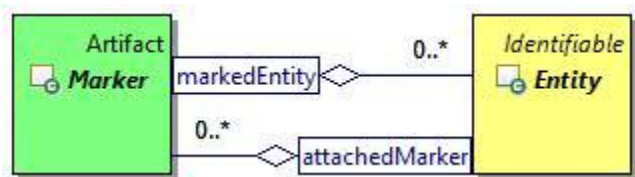


Figure 21: Marker Class Diagram.

3.6.123.6.15 Category

3.6.12-13.6.15.1 Description

A category is a marker that classifies entities.

3.6.12-23.6.15.2 Class Definition

The Category class has attribute values:

localNamespace

Value: icom_meta

localName

Value: Category

extendsFrom

Value: icom_meta:Marker, icom_core:Container

stereotype

Value: primary

description

Value: A category is a marker that classifies entities.

propertyDefinitions

The values for this attribute are defined in Section 3.6.15.3.

3.6.12-33.6.15.3 Property Definitions

The Category class inherits property definitions from super classes.

The Category class MUST have the property definitions:

icom_meta:superCategory

Description: **AZero or more** super category**ies**.

Required: False

2829	Inherited:	False
2830	Property Type:	icom_meta:Category
2831	Cardinality:	Single Multi
2832	Updatability:	Read Only
2833		
2834	icom_meta:subCategory	
2835	Description:	Zero or more sub categories.
2836	Required:	False
2837	Inherited:	False
2838	Property Type:	icom_meta:Category
2839	Cardinality:	Multi
2840	Updatability:	Read Only
2841		
2842	icom_meta:abstract	
2843	Description:	Indicates whether a category is abstract or concrete.
2844	Required:	False
2845	Inherited:	False
2846	Property Type:	Boolean
2847	Cardinality:	Single
2848	Updatability:	Read Write
2849		
2850	icom_meta:propertyDefinition	
2851	Description:	Optional or mandatory properties for a category application.
2852	Required:	False
2853	Inherited:	False
2854	Property Type:	icom_meta:PropertyDefinition
2855	Cardinality:	Multi
2856	Updatability:	Read Write
2857		
2858	The Category class MAY include additional property definitions which are implementation-defined.	
2859		

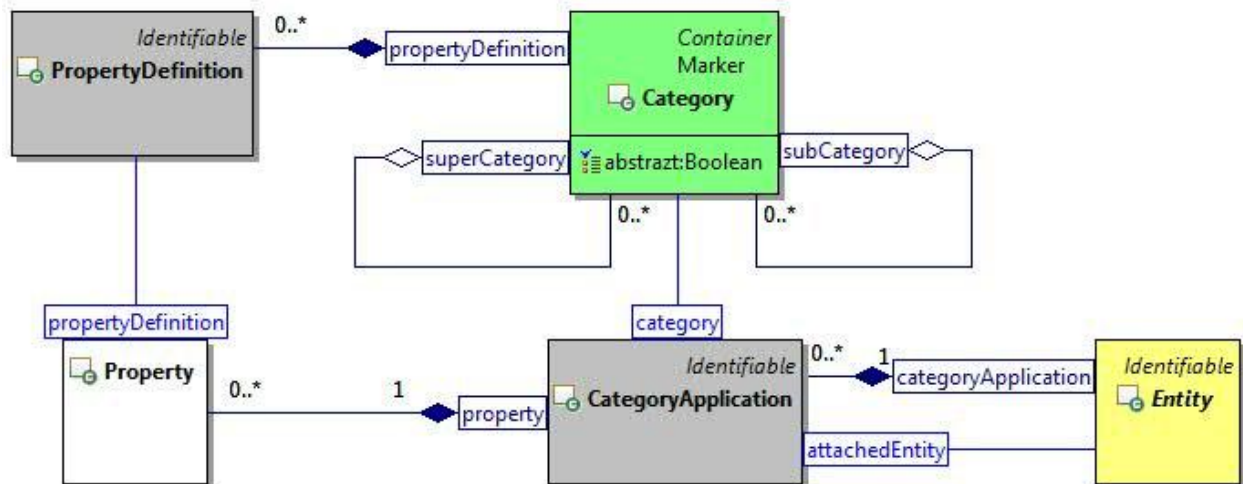


Figure 22: Category and Category Application Class Diagram.

3.6.13.6.16 CategoryApplication

3.6.13.13.6.16.1 Description

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

3.6.13.23.6.16.2 Class Definition

The CategoryApplication class has attribute values:

localNamespace

Value: icom_meta

localName

Value: CategoryApplication

extendsFrom

Value: icom_core:Identifiable

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.6.16.3.

~~3.6.13~~3.6.16.3 Property Definitions

The CategoryApplication class inherits property definitions from super classes.

The CategoryApplication class MUST have the property definitions:

icom_meta:attachedEntity

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

icom_meta:category

Description:	A category which is applied on an entity.
Required:	True
Inherited:	False
Property Type:	icom_meta:Category
Cardinality:	Single
Updatability:	On Create

icom_meta:property

Description:	Zero or more properties.
Required:	False
Inherited:	False
Property Type:	icom_meta:Property
Cardinality:	Multi
Updatability:	Read Write

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

~~3.6.14~~3.6.17 Tag

~~3.6.14.13~~3.6.17.1 Description

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

~~3.6.14.23~~3.6.17.2 Class Definition

The Tag class has attribute values:

localNamespace

Value: icom_meta

2927 **localName**
 2928 Value: Tag
 2929
 2930 **extendsFrom**
 2931 Value: icom_meta:Marker
 2932
 2933 **stereotype**
 2934 Value: primary
 2935
 2936 **description**
 2937 Value: A tag is a marker that labels entities by a keyword.
 2938
 2939 **propertyDefinitions**
 2940 The values for this attribute are defined in Section 3.6.17.3.

3.6.14.33.6.17.3 Property Definitions

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

icom_meta:applicationCount

Description:	An estimate of the number of times a tag is applied on entities.
Required:	False
Inherited:	False
Property Type:	Integer
Cardinality:	Single
Updatability:	Read Only

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

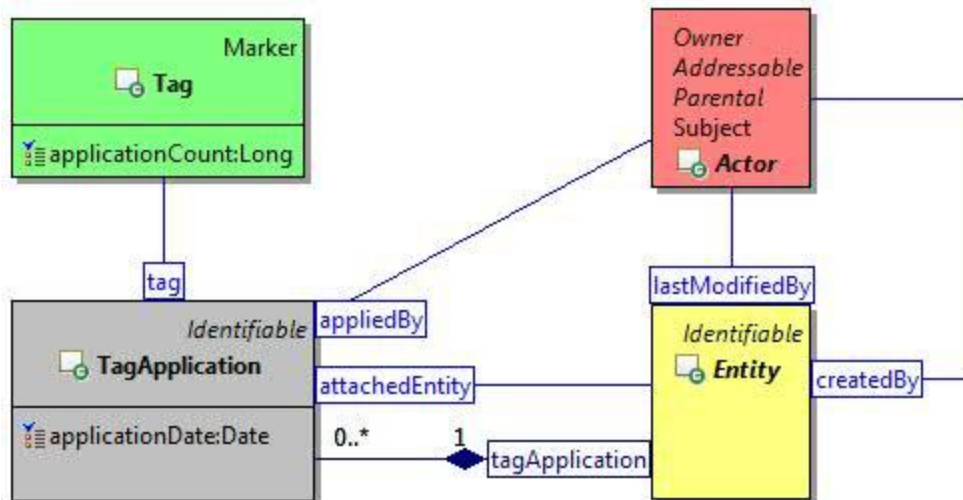


Figure 23: Tag and Tag Application Class Diagram.

3.6.15.3.6.18 TagApplication

3.6.15.13.6.18.1 Description

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

3.6.15.23.6.18.2 Class Definition

The TagApplication class has attribute values:

localNamespace

Value: icom_meta

localName

Value: TagApplication

extendsFrom

Value: icom_core:Identifiable

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.6.18.3.

3.6.15.33.6.18.3 Property Definitions

The TagApplication class inherits property definitions from super classes.

2984 The TagApplication class MUST have the property definitions:

2985

2986 **icom_meta:attachedEntity**

2987 Description: An entity on which a tag is applied.

2988 Required: True

2989 Inherited: False

2990 Property Type: icom_core:Entity

2991 Cardinality: Single

2992 Updatability: On Create

2993

2994 **icom_meta:tag**

2995 Description: A tag which is applied to an entity.

2996 Required: True

2997 Inherited: False

2998 Property Type: icom_meta:Tag

2999 Cardinality: Single

3000 Updatability: On Create

3001

3002 **icom_meta:appliedBy**

3003 Description: A user who applies a tag to an entity.

3004 Required: False

3005 Inherited: False

3006 Property Type: icom_core:Actor

3007 Cardinality: Single

3008 Updatability: Read Only

3009

3010 **icom_meta:applicationDate**

3011 Description: A date and time when a tag is applied to an entity.

3012 Required: False

3013 Inherited: False

3014 Property Type: DateTime

3015 Cardinality: Single

3016 Updatability: Read Write

3017

3018 The TagApplication class MAY include additional property definitions which are implementation-defined.

3019

3020 **~~3.6.16~~3.6.19 RelationshipBondable**

3021 **~~3.6.16~~3.6.19.1 Description**

3022 A relationship bondable entity is an entity which may be related to other entities by a relationship.

3023 Note: A relationship can exist among entities that are not relationships.

~~3.6.16~~3.6.19.2 Class Definition

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

The RelationshipBondable class has attribute values:

localNamespace

Value: icom_meta

localName

Value: RelationshipBondable

extendsFrom

Value: icom_core:Identifiable

stereotype

Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.6.19.3.

~~3.6.16~~3.6.19.3 Property Definitions

The RelationshipBondable class inherits property definitions from super classes.

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

~~3.6.17~~3.6.20 RelationshipDefinition

~~3.6.17.13~~3.6.20.1 Description

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

~~3.6.17.23~~3.6.20.2 Class Definition

The RelationshipDefinition class has attribute values:

localNamespace

Value: icom_meta

3063 **localName**

3064 Value: RelationshipDefinition

3065

3066 **extendsFrom**

3067 Value: icom_core:~~EntityDefinition~~Entity, icom_meta:RelationshipBondable

3068

3069 **stereotype**

3070 Value: primary

3071

3072 **description**

3073 Value: A relationship definition is an entity that defines a type of relationship.

3074

3075 **propertyDefinitions**

3076 The values for this attribute are defined in Section 3.6.20.3.

3077 3.6.17-33.6.20.3 **Property Definitions**

3078 The RelationshipDefinition class inherits property definitions from super classes.

3079 The RelationshipDefinition class MUST have the property definitions:

3080

3081 **icom_core:description**

3082 <u>Description:</u>	<u>A description of a relationship definition.</u>
3083 <u>Required:</u>	<u>False</u>
3084 <u>Inherited:</u>	<u>False</u>
3085 <u>Property Type:</u>	<u>String</u>
3086 <u>Cardinality:</u>	<u>Single</u>
3087 <u>Updatability:</u>	<u>Read Write</u>

3088

3089 **icom_meta:propertyDefinition**

3090 Description:	Optional or mandatory properties for a relationship.
3091 Required:	False
3092 Inherited:	False
3093 Property Type:	icom_meta:PropertyDefinition
3094 Cardinality:	Multi
3095 Updatability:	Read Write

3096

3097 **icom_meta:allowedSourceType**

3098 Description:	A list of expanded names of relationship bondable classes, indicating that the source entity of a relationship MUST be an instance of a class in the list.
3099	
3100	
3101 Required:	False
3102 Inherited:	False
3103 Property Type:	IRI
3104 Cardinality:	Multi

3105	Updatability:	Read Write
3106		
3107	icom_meta:allowedTargetType	
3108	Description:	A list of expanded names of relationship bondable classes,
3109		indicating that the target entity of a relationship MUST be an
3110		instance of a class in the list.
3111	Required:	False
3112	Inherited:	False
3113	Property Type:	IRI
3114	Cardinality:	Multi
3115	Updatability:	Read Write
3116		
3117	The RelationshipDefinition class MAY include additional property definitions which are implementation-	
3118	defined.	
3119		

3120 ~~3.6.18~~3.6.21 Relationship

3121 ~~3.6.18.13~~3.6.21.1 Description

3122 A relationship is an entity that relates a set of entities by a predicate.

3123 ~~3.6.18.23~~3.6.21.2 Class Definition

3124 The Relationship class has attribute values:

3125	
3126	localNamespace
3127	Value: icom_meta
3128	
3129	localName
3130	Value: Relationship
3131	
3132	extendsFrom
3133	Value: icom_core:Entity
3134	
3135	stereotype
3136	Value: primary
3137	
3138	description
3139	Value: A relationship is an entity that relates a set of entities by a predicate.
3140	
3141	propertyDefinitions
3142	The values for this attribute are defined in Section 3.6.21.3.

3143 ~~3.6.18.33~~3.6.21.3 Property Definitions

3144 The Relationship class inherits property definitions from super classes.

3145 The Relationship class MUST have the property definitions:

3146

3147 **icom_meta:relationshipDefinition**

3148 Description: A definition of relationships.

3149 Required: True

3150 Inherited: False

3151 Property Type: icom_meta:RelationshipDefinition

3152 Cardinality: Single

3153 Updatability: On Create

3154

3155 **icom_meta:sourceEntity**

3156 Description: A source entity of a relationship.

3157 Required: True

3158 Inherited: False

3159 Property Type: icom_meta:RelationshipBondable

3160 Cardinality: Single

3161 Updatability: On Create

3162

3163 **icom_meta:targetEntity**

3164 Description: One or more target entities of a relationship.

3165 Required: True

3166 Inherited: False

3167 Property Type: icom_meta:RelationshipBondable

3168 Cardinality: Multi

3169 Updatability: Read Write

3170

3171 **icom_meta:property**

3172 Description: Zero or more properties.

3173 Required: False

3174 Inherited: False

3175 Property Type: icom_meta:Property

3176 Cardinality: Multi

3177 Updatability: Read Write

3178

3179 The Relationship class MAY include additional property definitions which are implementation-defined.

3180

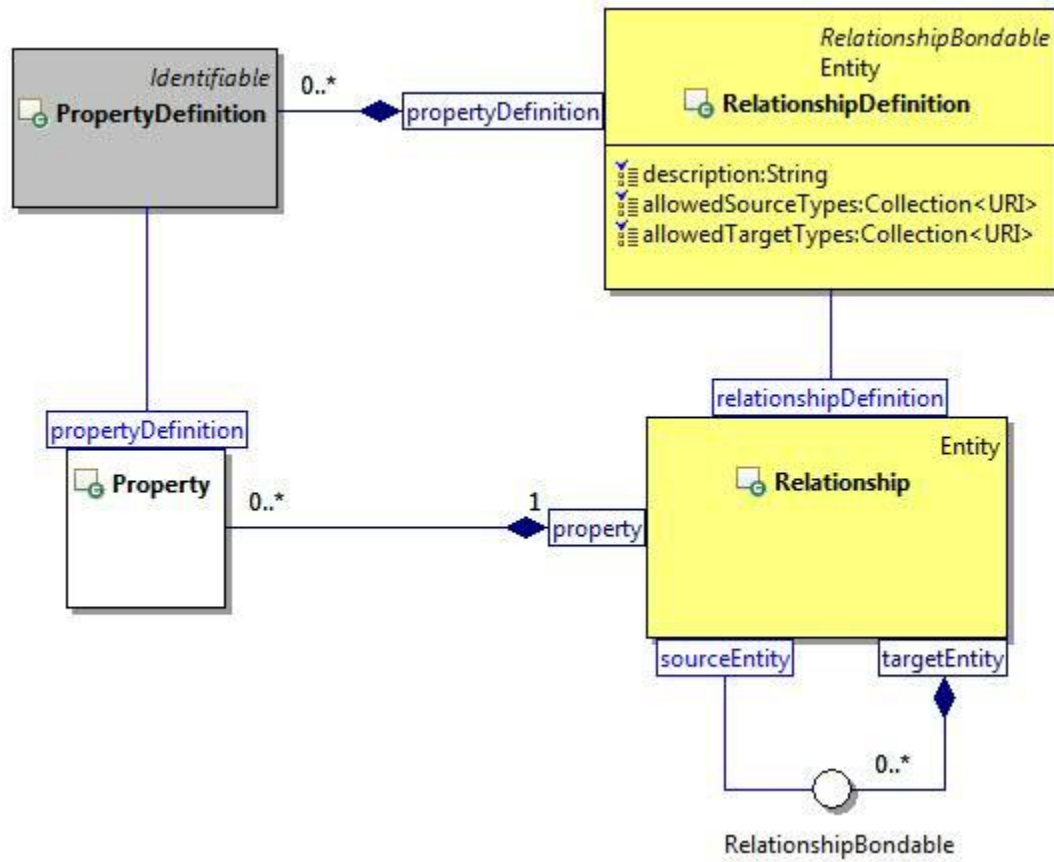


Figure 24: Relationship Class Diagram.

3.7 Common Concepts

3.7.1 Addressable

3.7.1.1 Description

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

3.7.1.2 Class Definition

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

The Addressable class has attribute values:

localNamespace

Value: icom_core

localName

Value: Addressable

3199 | **extendsFrom**
 3200 Value: icom_core:Identifiable
 3201
 3202 **stereotype**
 3203 Value: mixin
 3204
 3205 **description**
 3206 Value: Addressable is a mixin class which defines the characteristics of entities that has one or
 3207 more addresses.
 3208
 3209 **propertyDefinitions**
 3210 The values for this attribute are defined in Section 3.7.1.3.

3211 3.7.1.3 Property Definitions

3212 The Addressable class inherits property definitions from super classes.
 3213 The Addressable class MUST have the property definitions:

3214
 3215 **icom_core:entityAddress**
 3216 Description: Zero or more addresses of an addressable object.
 3217 Required: False
 3218 Inherited: False
 3219 Property Type: icom_core:EntityAddress
 3220 Cardinality: Multi
 3221 Updatability: Read Write
 3222
 3223 **icom_core:primaryAddress**
 3224 Description: The primary address of an addressable object.
 3225 Required: False
 3226 Inherited: False
 3227 Property Type: icom_core:EntityAddress
 3228 Cardinality: Single
 3229 Updatability: Read Write

3230
 3231 The Addressable class MAY include additional property definitions which are implementation-defined.
 3232

3233 3.7.2 EntityAddress

3234 3.7.2.1 Description

3235 An entity address object represents an address which is defined by type and IRI.

3236 3.7.2.2 Class Definition

3237 The EntityAddress class has attribute values:
 3238

3239 **localNamespace**
3240 Value: icom_core
3241
3242 **localName**
3243 Value: EntityAddress
3244
3245 **extendsFrom**
3246 Value:
3247
3248 **stereotype**
3249 Value: primary
3250
3251 **description**
3252 Value: An entity address object represents an address which is defined by type and IRI.
3253
3254 **propertyDefinitions**
3255 The values for this attribute are defined in Section 3.7.2.3.

3256 **3.7.2.3 Property Definitions**

3257 The EntityAddress class MUST have the property definitions:

3258
3259 **icom_core:addressType**
3260 Description: Type of an address.
3261 Required: False
3262 Inherited: False
3263 Property Type: String
3264 Cardinality: Single
3265 Updatability: Read Write
3266
3267 **icom_core:address**
3268 Description: A IRI representing an address.
3269 Required: False
3270 Inherited: False
3271 Property Type: IRI
3272 Cardinality: Single
3273 Updatability: Read Write

3275 **3.7.3 Participant**

3276 **3.7.3.1 Description**

3277 A participant object represents the participation of any addressable entity in a collaboration activity such
3278 as an occurrence, task, conference, discussion, and message.

3279 If an addressable entity is not specified, an address must be specified.

3280 3.7.3.2 Class Definition

3281 The Participant class has attribute values:

3282

3283 **localNamespace**

3284 Value: icom_core

3285

3286 **localName**

3287 Value: Participant

3288

3289 **extendsFrom**

3290 Value:

3291

3292 **stereotype**

3293 Value: primary

3294

3295 **description**

3296 Value: A participant object represents the participation of any addressable entity in a
3297 collaboration activity such as an occurrence, task, conference, discussion, and message.

3298

3299 **propertyDefinitions**

3300 The values for this attribute are defined in Section 3.7.3.3.

3301 3.7.3.3 Property Definitions

3302 The Participant class inherits property definitions from super classes.

3303 The Participant class MUST have the property definitions:

3304

3305 **icom_core:participant**

3306 Description: An addressable entity to participate in a collaboration activity.

3307 Required: False

3308 Inherited: False

3309 Property Type: icom_core:Addressable

3310 Cardinality: Single

3311 Updatability: On Create

3312

3313 **icom_core:address**

3314 Description: An address of a participant in a collaboration activity.

3315 Required: False

3316 Inherited: False

3317 Property Type: IRI

3318 Cardinality: Single

3319 Updatability: On Create

3320		
3321	icom_core:name	
3322	Description:	Name of a participant in a collaboration activity.
3323	Required:	False
3324	Inherited:	False
3325	Property Type:	String
3326	Cardinality:	Single
3327	Updatability:	On Create

3328

3329 The Participant class MAY include additional property definitions which are implementation-defined.

3330

3331 **3.7.4 Priority**

3332 **3.7.4.1 Description**

3333 A priority level for delivery of information.

3334 **3.7.4.2 Class Definition**

3335 The Priority class is a mixin class which defines a priority level for delivery of information.

3336 The Priority class has attribute values:

3337		
3338	localNamespace	
3339	Value:	icom_core
3340		
3341	localName	
3342	Value:	Priority
3343		
3344	extendsFrom	
3345	Value:	
3346		
3347	stereotype	
3348	Value:	mixin
3349		
3350	description	
3351	Value:	Priority is a mixin class which defines a priority level for delivery of information.
3352		
3353	propertyDefinitions	
3354		The values for this attribute are defined in Section 3.7.4.3.

3355 **3.7.4.3 Property Definitions**

3356 The Priority class MAY include additional property definitions which are implementation-defined.

3357

3.7.5 PriorityEnum

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

The PriorityEnum has attribute values:

localNamespace

Value: icom_core

localName

Value: PriorityEnum

extendsFrom

Value: icom_core:Priority

stereotype

Value: primary

isEnumeration

Value: TRUE

description

Value: Priority level for delivery of information.

instances

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

ICOM defines four priorities:

- **icom_core:Normal** a normal priority.
- **icom_core:Low** a low priority.
- **icom_core:Medium** a medium priority.
- **icom_core:High** a high priority.

3.7.6 DateTimeResolution

3.7.6.1 Description

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

3.7.6.2 Class Definition

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

3395 The DateTimeResolution class has attribute values:

3396

3397 **localNamespace**

3398 Value: icom_core

3399

3400 **localName**

3401 Value: DateTimeResolution

3402

3403 **extendsFrom**

3404 Value:

3405

3406 **stereotype**

3407 Value: mixin

3408

3409 **description**

3410 Value: DateTimeResolution is a mixin class which defines a resolution of date time value.

3411

3412 **propertyDefinitions**

3413 The values for this attribute are defined in Section 3.7.6.3.

3414 3.7.6.3 Property Definitions

3415 The DateTimeResolution class MAY include additional property definitions which are implementation-
3416 defined.

3417

3418 3.7.7 DateTimeResolutionEnum

3419 The DateTimeResolutionEnum class is an enum class that enumerates the instances each of which
3420 expresses a resolution of a date time value.

3421 The DateTimeResolutionEnum has attribute values:

3422

3423 **localNamespace**

3424 Value: icom_core

3425

3426 **localName**

3427 Value: DateTimeResolutionEnum

3428

3429 **extendsFrom**

3430 Value: icom_core:DateTimeResolution

3431

3432 **stereotype**

3433 Value: primary

3434

3435 **isEnumeration**
3436 Value: TRUE
3437
3438 **description**
3439 Value: Resolution of a date time value.
3440
3441 **instances**
3442 Value: <icom_core:Year, icom_core:Date, icom_core:Time>
3443
3444 ICOM defines three date time resolutions:
3445

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

3448

3449 **3.7.8 TimeZone**

3450 **3.7.8.1 Description**

3451 A time zone is a region that has a uniform standard time.

3452 **3.7.8.2 Class Definition**

3453 The TimeZone class has attribute values:

3454
3455 **localNamespace**
3456 Value: icom_core
3457
3458 **localName**
3459 Value: TimeZone
3460
3461 **extendsFrom**
3462 Value:
3463
3464 **stereotype**
3465 Value: primary
3466
3467 **description**
3468 Value: A time zone is a region that has a uniform standard time.
3469
3470 **propertyDefinitions**
3471 The values for this attribute are defined in Section 3.7.8.3.

3472 **3.7.8.3 Property Definitions**

3473 The TimeZone class inherits property definitions from super classes.

3474 The TimeZone class MUST have the property definitions:

3475

3476 **icom_core:ID**

3477 Description: Identifier of a time zone.

3478 Required: False

3479 Inherited: False

3480 Property Type: String

3481 Cardinality: Single

3482 Updatability: On Create

3483

3484 **icom_core:rawOffset**

3485 Description: An offset to add to Universal Coordinated Time (UTC) to get
3486 local time. If Daylight Saving Time is in effect at the specified
3487 date, the offset value is adjusted with the amount of daylight
3488 saving.

3489 Required: False

3490 Inherited: False

3491 Property Type: Integer

3492 Cardinality: Single

3493 Updatability: On Create

3494

3495 The TimeZone class MAY include additional property definitions which are implementation-defined.

3496

3497 3.7.9 Location

3498 3.7.9.1 Description

3499 A location object represents a physical location which is defined by name, description, and geo
3500 coordinates.

3501 Note: The name of a location may remain unchanged while a physical location may be changing. For
3502 example, a location name might be "On an airplane" while a physical location might be the geo
3503 coordinates of a flight path or current coordinates of a plane.

3504 3.7.9.2 Class Definition

3505 The Location class has attribute values:

3506

3507 **localNamespace**

3508 Value: icom_core

3509

3510 **localName**

3511 Value: Location

3512

3513 **extendsFrom**

3514 Value:

3515

3516 **stereotype**
3517 Value: primary
3518
3519 **description**
3520 Value: A location object represents a physical location which is defined by name, description, or
3521 geo coordinates.
3522
3523 **propertyDefinitions**
3524 The values for this attribute are defined in Section 3.7.9.3.

3525 **3.7.9.3 Property Definitions**

3526 The Location class MUST have the property definitions:

3527

3528 **icom_core:name**

3529	Description:	Name of a location.
3530	Required:	False
3531	Inherited:	False
3532	Property Type:	String
3533	Cardinality:	Single
3534	Updatability:	Read Write

3535

3536 **icom_core:description**

3537	Description:	A description of a location.
3538	Required:	False
3539	Inherited:	False
3540	Property Type:	String
3541	Cardinality:	Single
3542	Updatability:	Read Write

3543

3544 **icom_core:timeZone**

3545	Description:	Time zone of a location.
3546	Required:	False
3547	Inherited:	False
3548	Property Type:	icom_core:TimeZone
3549	Cardinality:	Single
3550	Updatability:	Read Write

3551

3552 **icom_core:coordinates**

3553	Description:	A list of geo coordinates marking a point, path, or area of a
3554		physical location.
3555	Required:	False
3556	Inherited:	False
3557	Property Type:	icom_core:GeoCoordinates

3558 Cardinality: Multi
3559 Updatability: Read Write

3560
3561 The Location class MAY include additional property definitions which are implementation-defined.
3562

3563 **3.7.10 GeoCoordinates**

3564 **3.7.10.1 Description**

3565 A geo coordinates object specifies the latitude, longitude, and altitude of a physical location.

3566 **3.7.10.2 Class Definition**

3567 The GeoCoordinates class has attribute values:

3568
3569 **localNamespace**
3570 Value: icom_core
3571
3572 **localName**
3573 Value: GeoCoordinates
3574
3575 **extendsFrom**
3576 Value:
3577
3578 **stereotype**
3579 Value: primary
3580
3581 **description**
3582 Value: A geo coordinates object specifies the latitude, longitude, and altitude of a physical
3583 location.
3584
3585 **propertyDefinitions**
3586 The values for this attribute are defined in Section 3.7.10.3.

3587 **3.7.10.3 Property Definitions**

3588 The GeoCoordinates class MUST have the property definitions:

3589
3590 **icom_core:latitude**
3591 Description: Latitude of a location.
3592 Required: False
3593 Inherited: False
3594 Property Type: Float
3595 Cardinality: Single
3596 Updatability: Read Write
3597

3598	icom_core:longitude	
3599	Description:	Longitude of a location.
3600	Required:	False
3601	Inherited:	False
3602	Property Type:	Float
3603	Cardinality:	Single
3604	Updatability:	Read Write
3605		
3606	icom_core:altitude	
3607	Description:	Altitude of a location.
3608	Required:	False
3609	Inherited:	False
3610	Property Type:	Float
3611	Cardinality:	Single
3612	Updatability:	Read Write
3613		
3614	The GeoCoordinates class MAY include additional property definitions which are implementation-defined.	
3615		

4 Extension Modules

4.1 Overview of Extension Modules

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

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

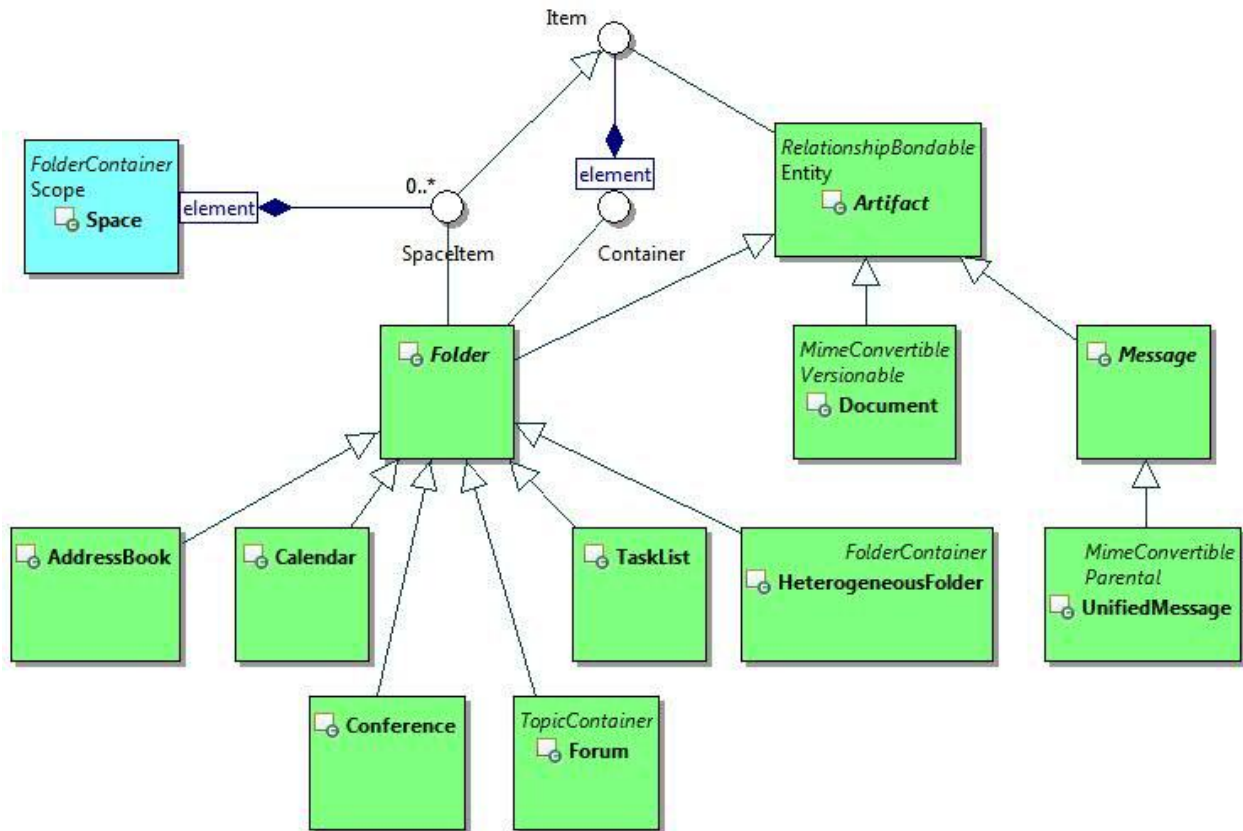


Figure 25: Containers of Collaboration Activities.

ICOM defines containers that provide contexts and structures for specific areas of collaborative activities. The UML class diagram in [Figure 24](#)Figure 25 depicts a Space as a hub of containers, including HeterogeneousFolder, AddressBook, Calendar, TaskList, Forum, and Conference. These containers are briefly described as follows:

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

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

- 3639 • an inbox or outbox for communication, or
- 3640 • a trash folder to archive all types of artifacts deleted from a space.
- 3641 **AddressBook** is a specialized container to manage contact or personal information, such as
- 3642 addresses, phone numbers, birthdays, anniversaries, and other entries.
- 3643 **Calendar** is a specialized container to support time management.
- 3644 **TaskList** is a specialized container to support task coordination.
- 3645 **Forum** is a specialized container to support
- 3646 • **Topic** sub-containers for threaded discussions and
- 3647 • **Announcement** sub-containers for time-sensitive communication.
- 3648 **Conference** is a specialized container that provides a durable context for real-time interactions.
- 3649

3650 The following ten modules are specified as extension modules of ICOM:

- 3651 | 1. Content Module (in ~~s~~Section 4.2) defines Content, MultiContent, and SimpleContent. A content
- 3652 | represents a piece of data in a document or message. Content, multi-content, simple content, and
- 3653 | online content form a composite design pattern.
- 3654 | 2. Document Module (in Section 4.3) defines Document, WikiPage, and version control model. A
- 3655 | document can contain a composite content defined in ~~section 4.2~~Section 4.2. Documents are
- 3656 | typically contained by heterogeneous folders.
- 3657 | 3. Message Module (in Section 4.4) defines Message, UnifiedMessage, InstantMessage, and
- 3658 | related classes. A message can contain a composite content defined in ~~section 4.2~~Section 4.2.
- 3659 | Unified messages are typically contained by heterogeneous folders.
- 3660 | 4. Presence Module (in Section 4.5) defines Presence, Activity, and Contact Method. Presence
- 3661 | represents a watchable state of a presentity (which is usually a person). Presence state is derived
- 3662 | using an actor's subscriptions.

3663 Note: Since a Presence is derived using a viewer's subscriptions, a Presence should not be shared

3664 with other viewers. For this reason, Presence is not modeled as Entity and is not assigned an access

3665 control list.

- 3666 | 5. Address Book Module (in Section 4.6) defines AddressBook and PersonContact. A person
- 3667 | contact can bookmark a reference to a person in an ICOM community as well as store addresses,
- 3668 | phone numbers, and other entries about a person who may not be in any ICOM community.
- 3669 | 6. Calendar Module (in Section 4.7) defines Calendar, Occurrence, and OccurrenceSeries.
- 3670 | Occurrence artifacts are used to resolve the free-busy times of participants for scheduling of
- 3671 | meetings and booking of rooms and other resources.
- 3672 | 7. Free Busy Module (in Section 4.8) defines FreeBusy. FreeBusy is a view derived from
- 3673 | occurrences in a calendar or a set of calendars using an actor's privileges to determine the free
- 3674 | or busy states of calendar occurrences.

3675 Note: Since a FreeBusy view is derived using a viewer's privileges, a FreeBusy should not be shared

3676 with other viewers. For this reason, FreeBusy is not modeled as Entity and is not assigned an access

3677 control list.

- 3678 | 8. Task List Module (in Section 4.9) defines TaskList and Task. Tasks are used to coordinate the
- 3679 | assignment of tasks and to track the progress of task activities.
- 3680 | 9. Forum Module (in Section 4.10) defines Forum, Topic, Announcement, and DiscussionMessage.
- 3681 | Topics, announcements, and discussions are used for treaded discussions. Moderators of a
- 3682 | forum can prune, merge, or fork the discussion threads.
- 3683 | 10. Conference Module (in Section 4.11) defines Conference and related classes. A conference can
- 3684 | contain visual, audio, and chat transcripts of the conference sessions. It also contains the current
- 3685 | status, conference settings, past sessions, active session, and activity logs.

4.2 Content Module

4.2.1 MimeConvertible

4.2.1.1 Description

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

4.2.1.2 Class Definition

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

The MimeConvertible class has attribute values:

localNamespace

Value: icom_content

localName

Value: MimeConvertible

extendsFrom

Value: icom_core:Identifiable

stereotype

Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.2.1.3.

4.2.1.3 Property Definitions

The MimeConvertible class inherits property definitions from super classes.

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

4.2.2 Content

4.2.2.1 Description

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

4.2.2.2 Class Definition

The Content class has attribute values:

localNamespace

Value: icom_content

localName

Value: Content

extendsFrom

Value: icom_core:Identifiable, icom_content:MimeConvertible

stereotype

Value: primary

isAbstract

Value: TRUE

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.2.2.3.

4.2.2.3 Property Definitions

The Content class inherits property definitions from super classes.

The Content class MUST have the property definitions:

icom_content:contentId

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

Required: False

Inherited: False

Property Type: String

Cardinality: Single

Updatability: Read Write

icom_content:mediaType

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

Required: False

Inherited: False

Property Type: String

Cardinality: Single

4.2.3.2 Class Definition

The MultiContent class has attribute values:

localNamespace

Value: icom_content

localName

Value: MultiContent

extendsFrom

Value: icom_content:Content

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.2.3.3.

4.2.3.3 Property Definitions

The MultiContent class inherits property definitions from super classes.

The MultiContent class MUST have the property definitions:

icom_content:part

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

Required: False

Inherited: False

Property Type: icom_content:MimeConvertible

Cardinality: Multi

Updatability: Read Write

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

4.2.4 SimpleContent

4.2.4.1 Description

A simple content holds a single piece of data.

4.2.4.2 Class Definition

The SimpleContent class has attribute values:

localNamespace

Value: icom_content

localName

Value: SimpleContent

extendsFrom

Value: icom_content:Content

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.2.4.3.

4.2.4.3 Property Definitions

The SimpleContent class inherits property definitions from super classes.

The SimpleContent class MUST have the property definitions:

icom_content:characterEncoding

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

Required: False

Inherited: False

Property Type: String

Cardinality: Single

Updatability: Read Write

icom_content:contentEncoding

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

Required: False

Inherited: False

Property Type: String

Cardinality: Single

Updatability: Read Write

3903 **extendsFrom**
3904 Value: icom_content:Content
3905
3906 **stereotype**
3907 Value: primary
3908
3909 **description**
3910 Value: An online content holds an online artifact attached to a message or invitation.
3911
3912 **propertyDefinitions**
3913 The values for this attribute are defined in Section 4.2.5.3.

3914 **4.2.5.3 Property Definitions**

3915 The OnlineContent class inherits property definitions from super classes.
3916 The OnlineContent class MUST have the property definition:

3917
3918 **icom_content:onlineAttachment**
3919 Description: An online artifact attached to a message.
3920 Required: True
3921 Inherited: False
3922 Property Type: icom_core:Artifact
3923 Cardinality: Single
3924 Updatability: Read Write
3925

3926 The OnlineContent class MAY include additional property definitions which are implementation-defined.
3927

3928 **4.2.6 ContentDispositionType**

3929 **4.2.6.1 Description**

3930 A content disposition type is a presentation style of content.

3931 **4.2.6.2 Class Definition**

3932 The ContentDispositionType class is a mixin class which defines a presentation style of content.
3933 The ContentDispositionType class has attribute values:

3934
3935 **localNamespace**
3936 Value: icom_content
3937
3938 **localName**
3939 Value: ContentDispositionType
3940

3941 **extendsFrom**
3942 Value:
3943
3944 **stereotype**
3945 Value: mixin
3946
3947 **description**
3948 Value: ContentDispositionType is a mixin class which defines a presentation style of content.
3949
3950 **propertyDefinitions**
3951 The values for this attribute are defined in Section 4.2.6.3.

3952 **4.2.6.3 Property Definitions**

3953 The ContentDispositionType class MAY include additional property definitions which are implementation-
3954 defined.
3955

3956 **4.2.7 ContentDispositionTypeEnum**

3957 The ContentDispositionTypeEnum class is an enum class that enumerates the instances each of which
3958 expresses a presentation style of content.

3959 The ContentDispositionTypeEnum class has attribute values:

3960
3961 **localNamespace**
3962 Value: icom_content
3963
3964 **localName**
3965 Value: ContentDispositionTypeEnum
3966
3967 **extendsFrom**
3968 Value: icom_content:ContentDispositionType
3969
3970 **stereotype**
3971 Value: primary
3972
3973 **isEnumeration**
3974 Value: TRUE
3975
3976 **description**
3977 Value: A presentation style of content.
3978
3979 **instances**
3980 Value: <icom_content:Inline, icom_content:Attachment>
3981

ICOM defines two content disposition types:

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

4.2.8 AttachedItem

4.2.8.1 Description

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

4.2.8.2 Class Definition

The AttachedItem class has attribute values:

localNamespace

Value: icom_content

localName

Value: AttachedItem

extendsFrom

Value:

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.2.8.3.

4.2.8.3 Property Definitions

The AttachedItem class MUST have the property definitions:

icom_core:name

Description: Name of a content attachment.

Required: True

Inherited: False

Property Type: String

Cardinality: Single

Updatability: Read Write

4022	icom_content:content	
4023	Description:	A content attached to an occurrence, task, or contact artifact.
4024	Required:	True
4025	Inherited:	False
4026	Property Type:	icom_content:Content
4027	Cardinality:	Single
4028	Updatability:	Read Write

4029

4030 The AttachedItem class MAY include additional property definitions which are implementation-defined.

4031

4032 4.3 Document Module

4033 4.3.1 Versionable

4034 4.3.1.1 Description

4035 A versionable artifact is

- 4036 1. a non-version-controlled copy,
- 4037 2. a specific versioned copy,
- 4038 3. a private working copy, or
- 4039 4. a representative copy (optional)

4040 of an artifact version series.

4041 When a versionable artifact is not under version control, a non-version-controlled copy MUST be the only
4042 copy in a version series, i.e. there is only one copy and one *objectId*.

4043 When a versionable artifact is under version control, a representative copy MAY provide a version-
4044 independent view of a versionable artifact.

4045 When a non-version-controlled copy is placed under version control, a versioned copy MUST be created.

4046 Assignment of an object identifier to a versioned copy is implementation-dependent:

- 4047 • if a versioned copy retains the object identifier of a non-version-controlled copy, the version type
4048 of a versionable artifact MUST change from NonVersionControlledCopy to VersionedCopy;
- 4049 • if a versioned copy is assigned a new object identifier that is different from the object identifier of
4050 a non-version-controlled copy, a representative copy MAY retain the object identifier of the non-
4051 version-controlled copy;
- 4052 • if both versioned copy and representative copy are assigned new object identifiers that are
4053 different from the object identifier of a non-version-controlled copy, the non-version-controlled
4054 copy SHALL be discarded.

4055 When a private working copy is checked in, a versioned copy MUST be created. Assignment of an object
4056 identifier to a versioned copy is implementation-dependent:

- 4057 • if a versioned copy retains the object identifier of a private working copy, the version type of a
4058 versionable artifact MUST change from PrivateWorkingCopy to VersionedCopy;
- 4059 • if a versioned copy is assigned a new object identifier that is different from the object identifier of
4060 a private working copy, the private working copy SHALL be discarded.

4061 It is optional for a service provider to provide a representative copy for a version series. If a representative
4062 copy is provided:

- 4063 • a representative copy MUST have its own object identifier that is different from the object
4064 identifier of any versioned copy or private working copy;

- 4065 • assignment of an object identifier to a representative copy is implementation-dependent:
 - 4066 ○ a representative copy MAY retain the object identifier of a non-version-controlled copy; if
 - 4067 so the version type of a versionable artifact MUST change from
 - 4068 NonVersionControlledCopy to RepresentativeCopy;
 - 4069 ○ a representative copy MAY be assigned a new object identifier that is different from the
 - 4070 object identifier of a non-version-controlled copy;
- 4071 • content and state of a representative copy is implementation-dependent:
 - 4072 ○ a representative copy MAY be a copy of the content and state of the latest versioned
 - 4073 copy or the latest major versioned copy in a version series;
 - 4074 ○ a representative copy MAY be a copy of the content and state of a private working copy if
 - 4075 the current user loading the representative copy is the same user who checks out a
 - 4076 version series.

4077 Note: Each versioned copy of a versionable artifact is itself a versionable artifact, i.e. it has its own
 4078 *objectId*. A versioned copy has a version number, label, and check in comment.

4079 Note: A private working copy is a versionable artifact created by an explicit checkout operation on a
 4080 versionable artifact under version control. The properties for a private working copy are identical to the
 4081 properties of a versioned copy on which a checkout operation was performed. Certain properties such as
 4082 *objectId* and *creationDate* are different from a versioned copy. The content of a private working copy is
 4083 identical to the content of a versioned copy. Its object identifier is different from that of the representative
 4084 copy or any versioned copy.

4085 A private working copy MAY be saved in a version series for sharing and co-editing, however, it needs
 4086 not be visible to users who may only have permissions to view other versioned copies in a version series.

4087 Note: Until it is checked in using an explicit check-in operation, a private working copy must not be
 4088 considered the LatestMajorVersion in a version series.

4089 A container of a versionable artifact CAN contain a representative copy so that it provides a version-
 4090 independent view of a state of the version series.

4091 Note: Starting from a representative copy in a container, an actor can traverse a version series to retrieve
 4092 any versioned copy or private working copy.

4093 ICOM version control model is based on the CMIS version control model specified in Section 2.1.9 of
 4094 Content Management Interoperability Services Version 1.0 [CMIS].

4095 4.3.1.2 Class Definition

4096 The Versionable class is a mixin class that defines the characteristics of artifacts that can be versioned.

4097 The Versionable class has attribute values:

```

4098
4099     localNamespace
4100         Value: icom_doc
4101
4102     localName
4103         Value: Versionable
4104
4105     extendsFrom
4106         Value: icom_core:Identifiable
4107
4108     stereotype
4109         Value: mixin
4110
  
```

4111 **description**
4112 Value: Versionable class is a mixin class that defines the characteristics of artifacts that can be
4113 versioned.

4114
4115 **propertyDefinitions**
4116 The values for this attribute are defined in Section 4.3.1.3.

4117 4.3.1.3 Property Definitions

4118 The Versionable class inherits property definitions from super classes.
4119 The Versionable class MUST have the property definitions:

4120
4121 **icom_doc:versionControlMetadata**
4122 Description: A version control metadata object attached to a versionable
4123 artifact.
4124 Required: False
4125 Inherited: False
4126 Property Type: icom_doc:VersionControlMetadata
4127 Cardinality: Single
4128 Updatability: Read Only

4129
4130 **icom_doc:versionType**
4131 Description: A type of version controlled copy of a versionable artifact.
4132 Required: False
4133 Inherited: False
4134 Property Type: icom_doc:VersionType
4135 Cardinality: Single
4136 Updatability: Read Only

4137
4138 The Versionable class MAY include additional property definitions which are implementation-defined.
4139

4140 4.3.2 VersionControlMetadata

4141 4.3.2.1 Description

4142 A version control metadata is an object that contains version control information.
4143 There are two classes of version control metadata: version series and version. A version control metadata
4144 of a versionable artifact is either a version series or a version depending on the version type.

- 4145 • If the version type is icom_doc:NonVersionControlledCopy then metadata is optional; if metadata
4146 is present, it MUST be a version series object.
- 4147 • If the version type is icom_doc:RepresentativeCopy, then metadata MUST be a version series
4148 object.
- 4149 • If the version type is icom_doc:VersionedCopy or icom_doc:PrivateWorkingCopy, then metadata
4150 MUST be a version object.

4.3.2.2 Class Definition

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

The VersionControlMetadata class has attribute values:

localNamespace

Value: icom_doc

localName

Value: VersionControlMetadata

extendsFrom

Value: icom_core:Identifiable

stereotype

Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.3.2.3.

4.3.2.3 Property Definitions

The VersionControlMetadata class inherits property definitions from super classes.

The VersionControlMetadata class MUST have the property definition:

icom_doc:representativeCopy

Description: A representative copy of a versionable artifact.

Required: False

Inherited: False

Property Type: icom_doc:Versionable

Cardinality: Single

Updatability: Read Only

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

4.3.3 VersionSeries

4.3.3.1 Description

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

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

4.3.3.2 Class Definition

The VersionSeries class has attribute values:

localNamespace

Value: icom_doc

localName

Value: VersionSeries

extendsFrom

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

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.3.3.3.

4.3.3.3 Property Definitions

The VersionSeries class inherits property definitions from super classes.

The VersionSeries class MUST have the property definitions:

icom_doc:versionHistory

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

Required: False

Inherited: False

Property Type: icom_doc:Version

Cardinality: Multi

Updatability: Read Only

icom_doc:versionableHistory

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

4229	Required:	False
4230	Inherited:	False
4231	Property Type:	icom_doc:Versionable
4232	Cardinality:	Multi
4233	Updatability:	Read Only
4234		
4235	icom_doc:latestVersionedCopy	
4236	Description:	Latest versioned copy of a versionable artifact.
4237	Required:	False
4238	Inherited:	False
4239	Property Type:	icom_doc:Versionable
4240	Cardinality:	Single
4241	Updatability:	Read Only
4242		
4243	icom_doc:privateWorkingCopy	
4244	Description:	A private working copy of a versionable artifact.
4245	Required:	False
4246	Inherited:	False
4247	Property Type:	icom_doc:Versionable
4248	Cardinality:	Single
4249	Updatability:	Read Only
4250		
4251	icom_doc:versionSeriesCheckedOut	
4252	Description:	Indicates whether a version series is checked out.
4253	Required:	False
4254	Inherited:	False
4255	Property Type:	Boolean
4256	Cardinality:	Single
4257	Updatability:	Read Only
4258		
4259	icom_doc:versionSeriesCheckedOutBy	
4260	Description:	An actor who checks out a version series.
4261	Required:	False
4262	Inherited:	False
4263	Property Type:	icom_core:Actor
4264	Cardinality:	Single
4265	Updatability:	Read Only
4266		
4267	icom_doc:versionSeriesCheckedOutOn	
4268	Description:	The time when a version series is checked out.
4269	Required:	False
4270	Inherited:	False

4271 Property Type: DateTime
4272 Cardinality: Single
4273 Updatability: Read Only

4274

4275 **icom_doc:versionSeriesCheckoutComment**

4276 Description: A check out comment of a version series.
4277 Required: False
4278 Inherited: False
4279 Property Type: String
4280 Cardinality: Single
4281 Updatability: Read Only

4282

4283 **icom_doc:totalSize**

4284 Description: Total size of all versioned copies of a versionable artifact in a
4285 version series.
4286 Required: False
4287 Inherited: False
4288 Property Type: Integer
4289 Cardinality: Single
4290 Updatability: Read Only

4291

4292 The VersionSeries class MAY include additional property definitions which are implementation-defined.

4293

4294 **4.3.4 Version**

4295 **4.3.4.1 Description**

4296 A version is a version control metadata that contains a version number, label, and description.
4297 A version object is a version control metadata of a versioned copy or a private working copy of a
4298 versionable artifact.

4299 **4.3.4.2 Class Definition**

4300 The Version class has attribute values:

4301

4302 **localNamespace**

4303 Value: icom_doc

4304

4305 **localName**

4306 Value: Version

4307

4308 **extendsFrom**

4309 Value: icom_core:Entity, icom_doc:VersionControlMetadata, icom_meta:RelationshipBondable

4310

4311 **stereotype**
4312 Value: primary
4313
4314 **description**
4315 Value: A version is a version control metadata that contains a version number, label, and
4316 description.
4317
4318 **propertyDefinitions**
4319 The values for this attribute are defined in Section 4.3.4.3.

4320 **4.3.4.3 Property Definitions**

4321 The Version class inherits property definitions from super classes.
4322 The Version class **MUST** have the property definitions;

4324 **icom_doc:checkinComment**

4325	Description:	A check in comment of a versioned copy.
4326	Required:	False
4327	Inherited:	False
4328	Property Type:	String
4329	Cardinality:	Single
4330	Updatability:	Read Write

4331

4332 **icom_doc:versionNumber**

4333	Description:	A version number of a versioned copy.
4334	Required:	True
4335	Inherited:	False
4336	Property Type:	Integer
4337	Cardinality:	Single
4338	Updatability:	Read Write

4339

4340 **icom_doc:versionLabel**

4341	Description:	A version label of a versioned copy.
4342	Required:	True
4343	Inherited:	False
4344	Property Type:	String
4345	Cardinality:	Single
4346	Updatability:	Read Write

4347

4348 **icom_doc:majorVersion**

4349	Description:	Indicates whether a versioned copy is a major version.
4350	Required:	True
4351	Inherited:	False
4352	Property Type:	Boolean

4353	Cardinality:	Single
4354	Updatability:	Read Write
4355		
4356	icom_doc:versionedOrPrivateWorkingCopy	
4357	Description:	A versioned copy or private working copy corresponding to a
4358		version of a versionable artifact.
4359	Required:	False
4360	Inherited:	False
4361	Property Type:	icom_doc:Versionable
4362	Cardinality:	Single
4363	Updatability:	Read Only
4364		

4365 The Version class MAY include additional property definitions which are implementation-defined.
4366

4367 4.3.5 VersionType

4368 4.3.5.1 Description

4369 A version type is a version state of a copy of versionable document.

4370 4.3.5.2 Class Definition

4371 The VersionType class is a mixin class which defines a version state of a copy of versionable document.
4372 The VersionType class has attribute values:

4373		
4374	localNamespace	
4375	Value:	icom_doc
4376		
4377	localName	
4378	Value:	VersionType
4379		
4380	extendsFrom	
4381	Value:	
4382		
4383	stereotype	
4384	Value:	mixin
4385		
4386	description	
4387	Value:	VersionType is a mixin class which defines a version state of a copy of versionable
4388		document.
4389		
4390	propertyDefinitions	
4391	The values for this attribute are defined in	Section 4.3.5.3.

4.3.5.3 Property Definitions

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

4.3.6 VersionTypeEnum

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

The VersionTypeEnum class has attribute values:

localNamespace

Value: icom_doc

localName

Value: VersionTypeEnum

extendsFrom

Value: icom_doc:VersionType

stereotype

Value: primary

isEnumeration

Value: TRUE

description

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

instances

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

ICOM defines four version types:

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

4.3.7 Document

4.3.7.1 Description

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

4.3.7.2 Class Definition

The Document class has attribute values:

localNamespace

Value: icom_doc

localName

Value: Document

extendsFrom

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

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.3.7.3.

4.3.7.3 Property Definitions

The Document class inherits property definitions from super classes.

The Document class MUST have the property definitions:

icom_content:content

Description: Content of a document.

Required: False

Inherited: False

Property Type: icom_content:Content

Cardinality: Single

Updatability: Read Write

icom_doc:size

Description: The size of a copy of a document.

Required: False

4470

Inherited:

False

4471

Property Type:

Integer

4472

Cardinality:

Single

4473

Updatability:

Read Only

4474

4475

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

4476

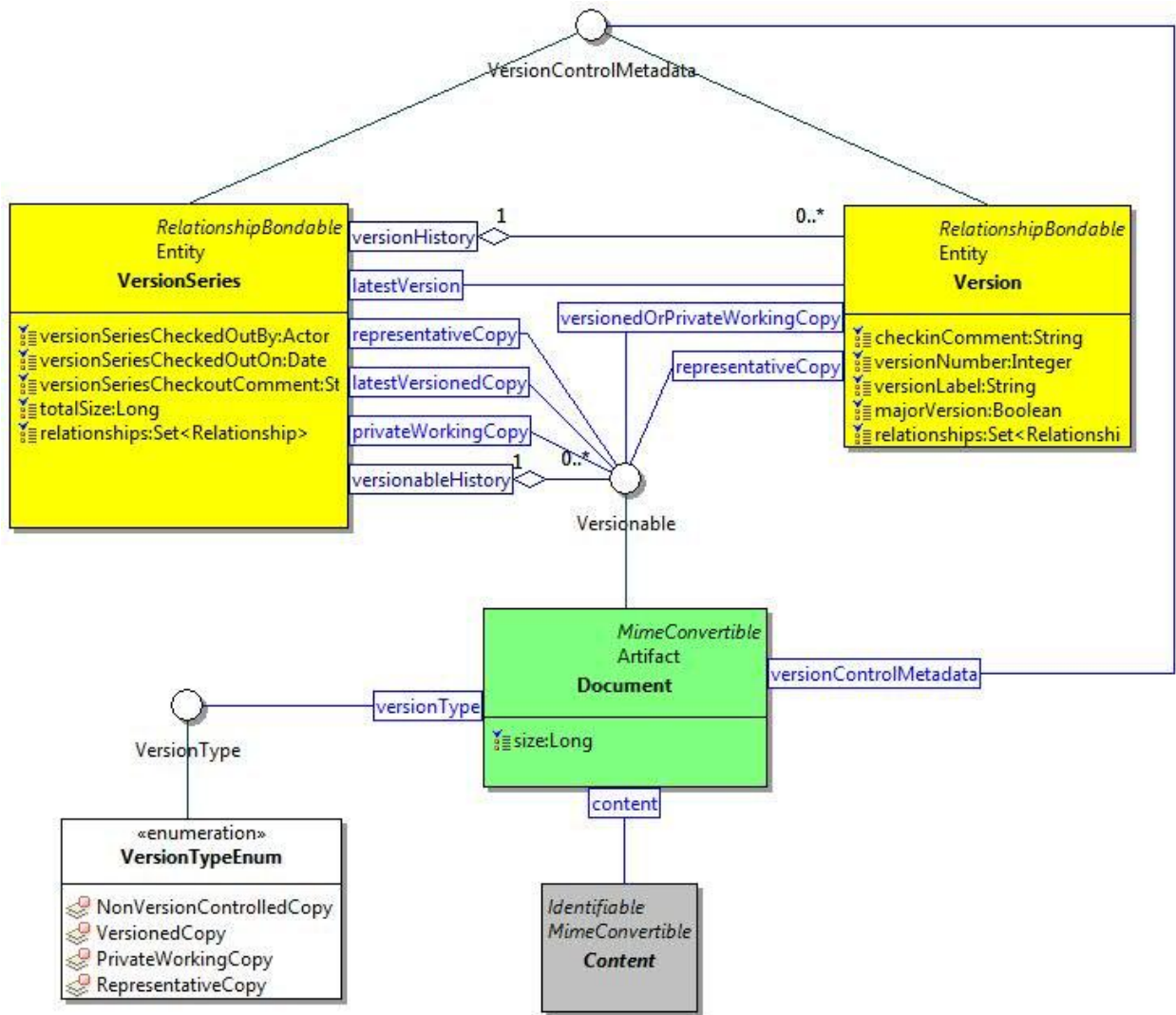


Figure 27: Document, Version Series, and Version Class Diagram.

4.3.8 WikiPage

4.3.8.1 Description

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

4.3.8.2 Class Definition

The WikiPage class has attribute values:

localNamespace

Value: icom_doc

localName

Value: WikiPage

extendsFrom

Value: icom_doc:Document

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.3.8.3.

4.3.8.3 Property Definitions

The WikiPage class inherits property definitions from super classes.

The WikiPage class MUST have the property definitions:

icom_doc:renderedPage

Description: An html page generated from a wiki content.

Required: False

Inherited: False

Property Type: String

Cardinality: Single

Updatability: Read Only

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

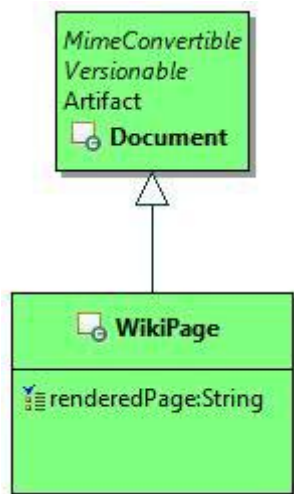


Figure 28: Wiki Page Class Diagram.

4.4 Message Module

4.4.1 Message

4.4.1.1 Description

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

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

4.4.1.2 Class Definition

The Message class has attribute values:

localNamespace

Value: icom_msg

localName

Value: Message

extendsFrom

Value: icom_core:Artifact

stereotype

Value: primary

isAbstract

Value: TRUE

4548 **description**
4549 Value: A message is a unit of conversation.
4550
4551 **propertyDefinitions**
4552 The values for this attribute are defined in Section 4.4.1.3.

4553 **4.4.1.3 Property Definitions**

4554 The Message class inherits property definitions from super classes.
4555 The Message class MUST have the property definitions:

4556

4557 **icom_content:content**

4558 Description:	Content of a message
4559 Required:	False
4560 Inherited:	False
4561 Property Type:	icom_content:Content
4562 Cardinality:	Single
4563 Updatability:	Read Write

4564

4565 **icom_msg:sender**

4566 Description:	Sender of a message.
4567 Required:	False
4568 Inherited:	False
4569 Property Type:	icom_core:Participant
4570 Cardinality:	Single
4571 Updatability:	Read Write

4572

4573 **icom_msg:deliveredTime**

4574 Description:	The date and time when a message is delivered to a given recipient.
4575	
4576 Required:	False
4577 Inherited:	False
4578 Property Type:	DateTime
4579 Cardinality:	Single
4580 Updatability:	Read Only

4581

4582 The Message class MAY include additional property definitions which are implementation-defined.

4583

4584 **4.4.2 UnifiedMessage**

4585 **4.4.2.1 Description**

4586 A unified message is a type of message delivered electronically over a computer, voice, fax, and other
4587 networks.

A unified message can be one of these types:

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

4.4.2.2 Class Definition

The UnifiedMessage class has attribute values:

localNamespace

Value: icom_msg

localName

Value: UnifiedMessage

extendsFrom

Value: icom_msg:Message, icom_content:MimeConvertible

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.4.2.3.

4.4.2.3 Property Definitions

The UnifiedMessage class inherits property definitions from super classes.

The UnifiedMessage class MUST have the property definitions:

icom_core:priority

Description: The priority of a message.

Required: False

Inherited: False

Property Type: icom_core:Priority

Cardinality: Single

Updatability: Read Write

icom_content:contentId

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

4629	Required:	False
4630	Inherited:	False
4631	Property Type:	String
4632	Cardinality:	Single
4633	Updatability:	Read Write
4634		
4635	icom_content:mediaType	
4636	Description:	Media type is a two-part identifier for Internet file formats.
4637	Required:	False
4638	Inherited:	False
4639	Property Type:	String
4640	Cardinality:	Single
4641	Updatability:	Read Write
4642		
4643	icom_content:contentDisposition	
4644	Description:	Content disposition specifies a presentation style.
4645	Required:	False
4646	Inherited:	False
4647	Property Type:	icom_content:ContentDispositionType
4648	Cardinality:	Single
4649	Updatability:	Read Write
4650		
4651	icom_msg:envelopeSender	
4652	Description:	An envelope sender is a participant to receive bounced message. It is also known as return path.
4653		
4654	Required:	False
4655	Inherited:	False
4656	Property Type:	icom_core:Participant
4657	Cardinality:	Single
4658	Updatability:	Read Write
4659		
4660	icom_msg:toReceivers	
4661	Description:	A list of participants to receive a message.
4662	Required:	False
4663	Inherited:	False
4664	Property Type:	icom_core:Participant
4665	Cardinality:	Multi
4666	Updatability:	Read Write
4667		
4668	icom_msg:ccReceivers	
4669	Description:	A list of participants to receive carbon-copies of a message.
4670	Required:	False
4671	Inherited:	False

4672	Property Type:	icom_core:Participant
4673	Cardinality:	Multi
4674	Updatability:	Read Write
4675		
4676	icom_msg:bccReceivers	
4677	Description:	A list of participants to receive blind-carbon-copies of a message.
4678		
4679	Required:	False
4680	Inherited:	False
4681	Property Type:	icom_core:Participant
4682	Cardinality:	Multi
4683	Updatability:	Read Write
4684		
4685	icom_msg:replyTo	
4686	Description:	A list of participants to receive a reply message.
4687	Required:	False
4688	Inherited:	False
4689	Property Type:	icom_core:Participant
4690	Cardinality:	Multi
4691	Updatability:	Read Write
4692		
4693	icom_msg:flag	
4694	Description:	Zero or more flags on a message.
4695	Required:	False
4696	Inherited:	False
4697	Property Type:	icom_msg:UnifiedMessageFlag
4698	Cardinality:	Multi
4699	Updatability:	Read Write
4700		
4701	icom_msg:messageDispositionNotificationRequested	
4702	Description:	A message disposition notification requested for a message.
4703	Required:	False
4704	Inherited:	False
4705	Property Type:	Boolean
4706	Cardinality:	Single
4707	Updatability:	Read Write
4708		
4709	icom_msg:messageDeliveryStatusNotificationRequest	
4710	Description:	Indicates the types of delivery status notifications requested for a message. Default is icom_msg:Failure.
4711		
4712	Required:	False
4713	Inherited:	False
4714	Property Type:	icom_msg:UnifiedMessageDeliveryStatusNotificationRequest

4715	Cardinality:	Multi
4716	Updatability:	Read Write
4717		
4718	icom_msg:channel	
4719	Description:	Indicates the delivery channel of a message.
4720	Required:	False
4721	Inherited:	False
4722	Property Type:	icom_msg:UnifiedMessageChannel
4723	Cardinality:	Single
4724	Updatability:	Read Write
4725		
4726	icom_msg:editMode	
4727	Description:	Indicates an editable mode (new, draft, or delivered) of a message.
4728		
4729	Required:	False
4730	Inherited:	False
4731	Property Type:	icom_msg:UnifiedMessageEditMode
4732	Cardinality:	Single
4733	Updatability:	Read Only
4734		
4735	icom_msg:mimeHeader	
4736	Description:	A list of headers. Each header is represented by a multi-valued property.
4737		
4738	Required:	False
4739	Inherited:	False
4740	Property Type:	icom_meta:Property
4741	Cardinality:	Multi
4742	Updatability:	Read Write
4743		
4744	icom_msg:size	
4745	Description:	The size of a unified message.
4746	Required:	False
4747	Inherited:	False
4748	Property Type:	Integer
4749	Cardinality:	Single
4750	Updatability:	Read Only
4751		
4752	The UnifiedMessage class MAY include additional property definitions which are implementation-defined.	
4753		

4.4.3 UnifiedMessageParticipant

4.4.3.1 Description

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

4.4.3.2 Class Definition

The UnifiedMessageParticipant class has attribute values:

localNamespace

Value: icom_msg

localName

Value: UnifiedMessageParticipant

extendsFrom

Value: icom_core:Participant

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.4.3.3.

4.4.3.3 Property Definitions

The UnifiedMessageParticipant class inherits property definitions from super classes.

The UnifiedMessageParticipant class MUST have the property definitions:

icom_msg:fullAddress

Description: Full address of a participant.

Required: False

Inherited: False

Property Type: IRI

Cardinality: Single

Updatability: Read Write

icom_msg:localPart

Description: Local part of a full address.

Required: False

4794	Inherited:	False
4795	Property Type:	String
4796	Cardinality:	Single
4797	Updatability:	Read Write

4798

4799 **icom_msg:domainPart**

4800	Description:	Domain part of a full address.
4801	Required:	False
4802	Inherited:	False
4803	Property Type:	String
4804	Cardinality:	Single
4805	Updatability:	Read Write

4806

4807 The UnifiedMessageParticipant class MAY include additional property definitions which are
4808 implementation-defined.

4809

4810 **4.4.4 UnifiedMessageFlag**

4811 **4.4.4.1 Description**

4812 A unified message flag is a flag on a message.

4813 **4.4.4.2 Class Definition**

4814 The UnifiedMessageFlag class is a mixin class which defines a flag on a message.

4815 The UnifiedMessageFlag class has attribute values:

4816

4817 **localNamespace**

4818 Value: icom_msg

4819

4820 **localName**

4821 Value: UnifiedMessageFlag

4822

4823 **extendsFrom**

4824 Value:

4825

4826 **stereotype**

4827 Value: mixin

4828

4829 **description**

4830 Value: UnifiedMessageFlag is a mixin class which defines a flag on a message.

4831

4832 **propertyDefinitions**

4833 The values for this attribute are defined in Section 4.4.4.3.

4.4.4.3 Property Definitions

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

4.4.5 UnifiedMessageFlagEnum

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

The UnifiedMessageFlagEnum class has attribute values:

localNamespace

Value: icom_msg

localName

Value: UnifiedMessageFlagEnum

extendsFrom

Value: icom_msg:UnifiedMessageFlag

stereotype

Value: primary

isEnumeration

Value: TRUE

description

Value: A flag on a message.

instances

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

ICOM defines eight flags:

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

4876

4877 **4.4.6 UnifiedMessageDeliveryStatusNotificationRequest**

4878 **4.4.6.1 Description**

4879 A unified message delivery status notification request is a directive for notifying a participant of delivery
4880 status of a message.

4881 **4.4.6.2 Class Definition**

4882 The UnifiedMessageDeliveryStatusNotificationRequest class is a mixin class which defines a directive for
4883 notifying a participant of delivery status of a message.

4884 The UnifiedMessageDeliveryStatusNotificationRequest class has attribute values:

4885

4886 **localNamespace**

4887 Value: icom_msg

4888

4889 **localName**

4890 Value: UnifiedMessageDeliveryStatusNotificationRequest

4891

4892 **extendsFrom**

4893 Value:

4894

4895 **stereotype**

4896 Value: mixin

4897

4898 **description**

4899 Value: UnifiedMessageDeliveryStatusNotificationRequest is a mixin class which defines a
4900 directive for notifying a participant of delivery status of a message.

4901

4902 **propertyDefinitions**

4903 The values for this attribute are defined in Section 4.4.6.3.

4904 **4.4.6.3 Property Definitions**

4905 The UnifiedMessageDeliveryStatusNotificationRequest class MAY include additional property definitions
4906 which are implementation-defined.

4907

4908 **4.4.7 UnifiedMessageDeliveryStatusNotificationRequestEnum**

4909 The UnifiedMessageDeliveryStatusNotificationRequestEnum class is an enum class that enumerates the
4910 instances each of which expresses a request for one of several types of delivery status notification.

4911 The UnifiedMessageDeliveryStatusNotificationRequestEnum class has attribute values:

4912

4913 **localNamespace**

4914 Value: icom_msg

4915

4916 **localName**
4917 Value: UnifiedMessageDeliveryStatusNotificationRequestEnum
4918
4919 **extendsFrom**
4920 Value: icom_msg:UnifiedMessageDeliveryStatusNotificationRequest
4921
4922 **stereotype**
4923 Value: primary
4924
4925 **isEnumeration**
4926 Value: TRUE
4927
4928 **description**
4929 Value: A request for one of several types of delivery status notification.
4930
4931 **instances**
4932 Value: <icom_msg:Never, icom_msg:Success, icom_msg:Failure, icom_msg:Delay>
4933
4934 ICOM defines four delivery status notification requests:
4935 • **icom_msg:Never** a sender requests status notification not be returned to the sender under any
4936 condition.
4937 • **icom_msg:Success** a sender requests a status notification for successful delivery of a message.
4938 • **icom_msg:Failure** a sender requests a status notification for delivery failure of a message.
4939 • **icom_msg:Delay** a sender requests a status notification when delivery of a message has been
4940 delayed for an unusual length of time.
4941

4942 4.4.8 UnifiedMessageChannel

4943 4.4.8.1 Description

4944 A message channel used to deliver a unified message.

4945 4.4.8.2 Class Definition

4946 The UnifiedMessageChannel class is a mixin class which defines a channel used to deliver a unified
4947 message.

4948 The UnifiedMessageChannel class has attribute values:

4949
4950 **localNamespace**
4951 Value: icom_msg
4952
4953 **localName**
4954 Value: UnifiedMessageChannel
4955

4956 **extendsFrom**
4957 Value:
4958
4959 **stereotype**
4960 Value: mixin
4961
4962 **description**
4963 Value: UnifiedMessageChannel is a mixin class which defines a channel used to deliver a
4964 unified message.
4965
4966 **propertyDefinitions**
4967 The values for this attribute are defined in Section 4.4.8.3.

4968 **4.4.8.3 Property Definitions**

4969 The UnifiedMessageChannel class MAY include additional property definitions which are implementation-
4970 defined.
4971

4972 **4.4.9 UnifiedMessageChannelEnum**

4973 The UnifiedMessageChannelEnum class is an enum class that enumerates the instances each of which
4974 expresses a type of delivery channel.

4975 The UnifiedMessageChannelEnum class has attribute values:

4976
4977 **localNamespace**
4978 Value: icom_msg
4979
4980 **localName**
4981 Value: UnifiedMessageChannelEnum
4982
4983 **extendsFrom**
4984 Value: icom_msg:UnifiedMessageChannel
4985
4986 **stereotype**
4987 Value: primary
4988
4989 **isEnumeration**
4990 Value: TRUE
4991
4992 **description**
4993 Value: A delivery channel.
4994
4995 **instances**
4996 Value: <icom_msg:Email, icom_msg:Voice, icom_msg:Fax, icom_msg:Notification>
4997

- 4998 ICOM defines four channel types:
- 4999 • **icom_msg:Email** delivery channel is email.
 - 5000 • **icom_msg:Voice** delivery channel is voice.
 - 5001 • **icom_msg:Fax** delivery channel is fax.
 - 5002 • **icom_msg:Notification** delivery channel is notification.
- 5003

5004 **4.4.10 UnifiedMessageEditMode**

5005 **4.4.10.1 Description**

5006 A unified message edit mode is a mode that indicates whether a unified message is editable.

5007 **4.4.10.2 Class Definition**

5008 The UnifiedMessageEditMode class is a mixin class which defines a mode that indicates whether a
5009 unified message is editable.

5010 The UnifiedMessageEditMode class has attribute values:

5011

5012 **localNamespace**

5013 Value: icom_msg

5014

5015 **localName**

5016 Value: UnifiedMessageEditMode

5017

5018 **extendsFrom**

5019 Value:

5020

5021 **stereotype**

5022 Value: mixin

5023

5024 **description**

5025 Value: UnifiedMessageEditMode is a mixin class which defines a mode that indicates whether a
5026 unified message is editable.

5027

5028 **propertyDefinitions**

5029 The values for this attribute are defined in Section 4.4.10.3.

5030 **4.4.10.3 Property Definitions**

5031 The UnifiedMessageEditMode class MAY include additional property definitions which are
5032 implementation-defined.

5033

5034 **4.4.11 UnifiedMessageEditModeEnum**

5035 The UnifiedMessageEditModeEnum class is an enum class that enumerates the instances each of which
5036 expresses whether a message is a new copy, saved draft copy, or delivered copy.

5037 The UnifiedMessageEditModeEnum class has attribute values:

5038

5039 **localNamespace**

5040 Value: icom_msg

5041

5042 **localName**

5043 Value: UnifiedMessageEditModeEnum

5044

5045 **extendsFrom**

5046 Value: icom_msg:UnifiedMessageEditMode

5047

5048 **stereotype**

5049 Value: primary

5050

5051 **isEnumeration**

5052 Value: TRUE

5053

5054 **description**

5055 Value: A message is a new copy, a saved draft copy, or a delivered copy. New or draft copies

5056 are usually editable while delivered copies are usually not editable.

5057

5058 **instances**

5059 Value: <icom_msg:NewCopy, icom_msg:DraftCopy, icom_msg:DeliveredCopy>

5060

5061 ICOM defines three modes:

5062 • **icom_msg:NewCopy** a message is a new message.

5063 • **icom_msg:DraftCopy** a message is saved as a draft.

5064 • **icom_msg:DeliveredCopy** a message is a sent or received message.

5065

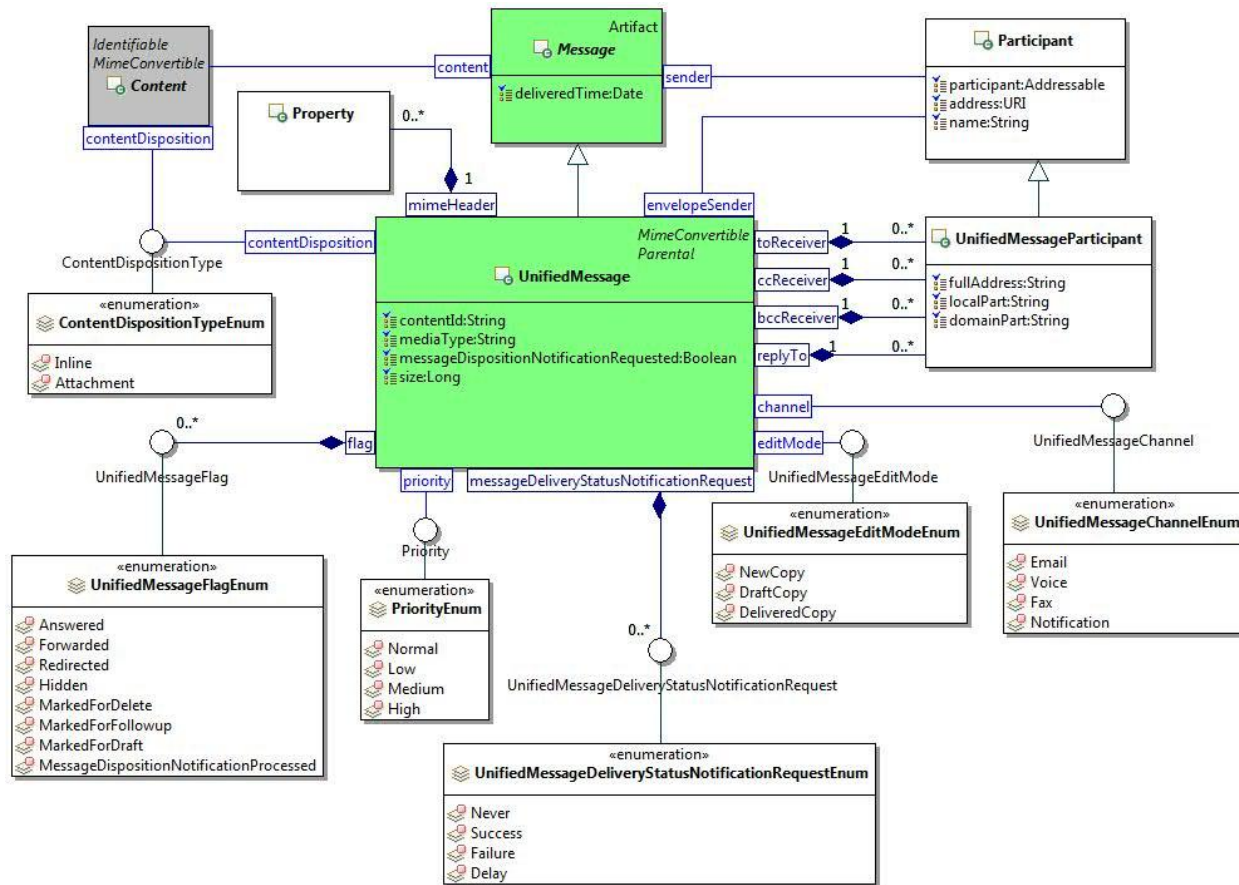


Figure 29: Unified Message Class Diagram.

4.4.12 InstantMessage

4.4.12.1 Description

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

4.4.12.2 Class Definition

The InstantMessage class has attribute values:

localNamespace

Value: icom_msg

localName

Value: InstantMessage

extendsFrom

Value: icom_msg:Message

5084 **stereotype**
5085 Value: primary
5086
5087 **isAbstract**
5088 Value: ~~TRUE~~FALSE
5089
5090 **description**
5091 Value: An instant message is a type of message for synchronous, usually text based,
5092 conversation.
5093
5094 **propertyDefinitions**
5095 The values for this attribute are defined in Section 4.4.12.3.

5096 **4.4.12.3 Property Definitions**

5097 The InstantMessage class inherits property definitions from super classes.

5098 The InstantMessage class **MUST** have the property definitions:

5099

5100 **icom_msg:toReceivers**

5101	Description:	A list of participants to receive a message.
5102	Required:	False
5103	Inherited:	False
5104	Property Type:	icom_core:Participant
5105	Cardinality:	Multi
5106	Updatability:	Read Write

5107

5108 **icom_msg:conversationId**

5109	Description:	An identifier of a conversation involving one or more instant
5110		messages.
5111	Required:	False
5112	Inherited:	False
5113	Property Type:	Integer
5114	Cardinality:	Single
5115	Updatability:	Read Write

5116

5117 **icom_msg:clientId**

5118	Description:	An identifier of a client.
5119	Required:	False
5120	Inherited:	False
5121	Property Type:	String
5122	Cardinality:	Single
5123	Updatability:	Read Write

5124

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

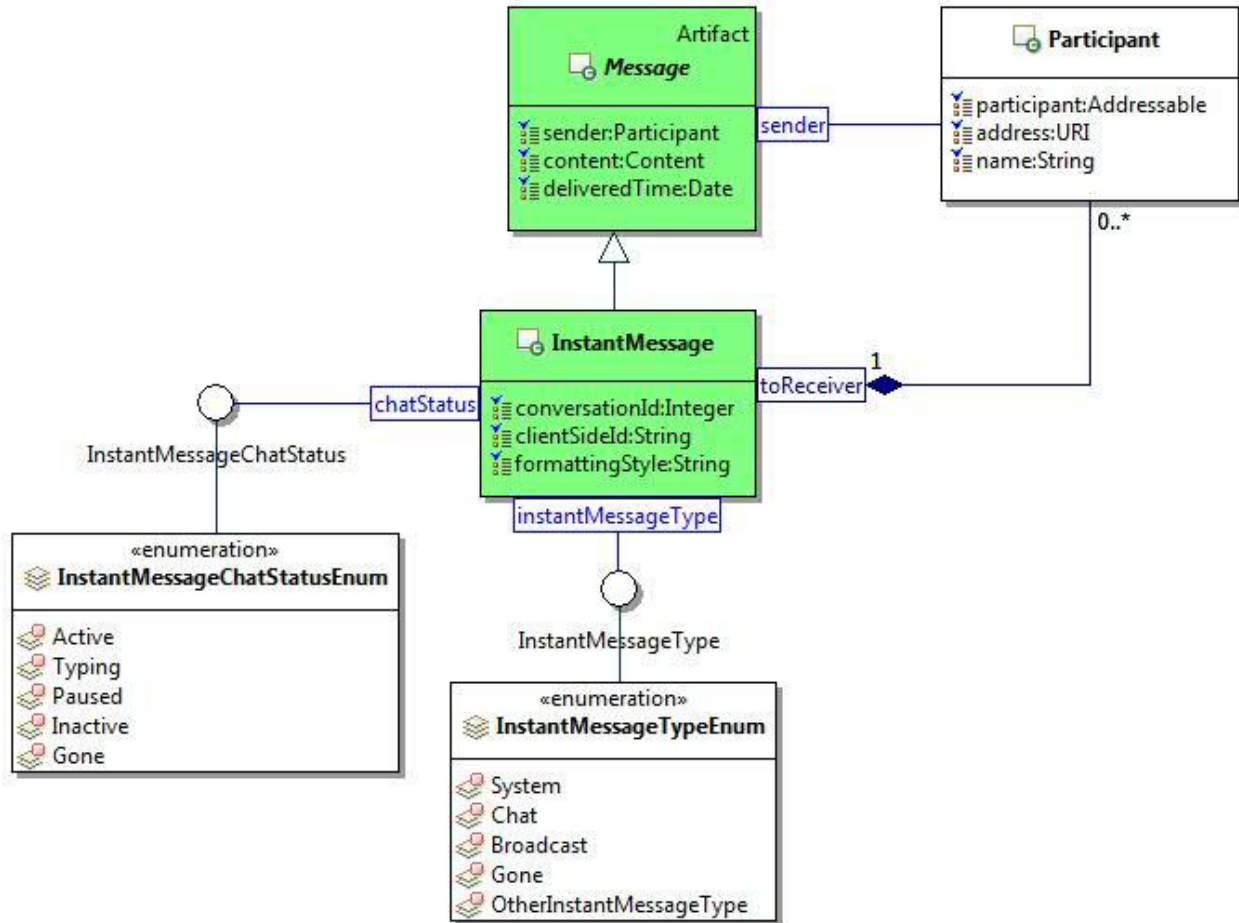


Figure 30: Instant Message Class Diagram.

4.4.13 InstantMessageType

4.4.13.1 Description

An instant message type.

4.4.13.2 Class Definition

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

The InstantMessageType class has attribute values:

localNamespace

Value: `icom_msg`

localName

Value: `InstantMessageType`

extendsFrom

Value:

5167
5168 **stereotype**
5169 Value: mixin
5170
5171 **description**
5172 Value: InstantMessageType is a mixin class which defines a type of instant message.
5173
5174 **propertyDefinitions**
5175 The values for this attribute are defined in Section 4.4.13.3.

5176 **4.4.13.3 Property Definitions**

5177 The InstantMessageType class MAY include additional property definitions which are implementation-
5178 defined.
5179

5180 **4.4.14 InstantMessageTypeEnum**

5181 The InstantMessageTypeEnum class is an enum class that enumerates the instances each of which
5182 expresses a type of instant message.

5183 The InstantMessageTypeEnum class has attribute values:

5184
5185 **localNamespace**
5186 Value: icom_msg
5187
5188 **localName**
5189 Value: InstantMessageTypeEnum
5190
5191 **extendsFrom**
5192 Value: icom_msg:InstantMessageType
5193
5194 **stereotype**
5195 Value: primary
5196
5197 **isEnumeration**
5198 Value: TRUE
5199
5200 **description**
5201 Value: A type of instant message.
5202
5203 **instances**
5204 Value: <icom_msg:System, icom_msg:Chat, icom_msg:Broadcast, icom_msg:Gone,
5205 icom_msg:OtherInstantMessageType>
5206

5207 ICOM defines five instant message types:

- 5208 • **icom_msg:System** an instant message is a system message.

- 5209 • **icom_msg:Chat** an instant message is a chat message.
 - 5210 • **icom_msg:Broadcast** an instant message is a broadcast message.
 - 5211 • **icom_msg:Gone** an instant message is a message indicating that a user is gone.
 - 5212 • **icom_msg:OtherInstantMessageType** an instant message is of other type.
- 5213

5214 4.4.15 InstantMessageChatStatus

5215 4.4.15.1 Description

5216 An instant message chat status defines a vocabulary of chat status.

5217 4.4.15.2 Class Definition

5218 The InstantMessageChatStatus class is a mixin class which defines a chat status.

5219 The InstantMessageChatStatus class has attribute values:

5220

5221 **localNamespace**

5222 Value: icom_msg

5223

5224 **localName**

5225 Value: InstantMessageChatStatus

5226

5227 **extendsFrom**

5228 Value:

5229

5230 **stereotype**

5231 Value: mixin

5232

5233 **description**

5234 Value: InstantMessageChatStatus is a mixin class which defines a chat status.

5235

5236 **propertyDefinitions**

5237 The values for this attribute are defined in Section 4.4.15.3.

5238 4.4.15.3 Property Definitions

5239 The InstantMessageChatStatus class MAY include additional property definitions which are
5240 implementation-defined.

5241

5242 4.4.16 InstantMessageChatStatusEnum

5243 The InstantMessageChatStatusEnum class is an enum class that enumerates the instances each of
5244 which expresses a chat status of a user.

5245 The InstantMessageChatStatusEnum class has attribute values:

5246

5247 **localNamespace**
5248 Value: icom_msg
5249
5250 **localName**
5251 Value: InstantMessageChatStatusEnum
5252
5253 **extendsFrom**
5254 Value: icom_msg:InstantMessageChatStatus
5255
5256 **stereotype**
5257 Value: primary
5258
5259 **isEnumeration**
5260 Value: TRUE
5261
5262 **description**
5263 Value: A chat status of a user.
5264
5265 **instances**
5266 Value: <icom_msg:Active, icom_msg:Typing, icom_msg:Paused, icom_msg:Inactive,
5267 icom_msg:Gone>
5268
5269 ICOM defines five chat status:

- 5270 • **icom_msg:Active** a user is active.
- 5271 • **icom_msg:Typing** a user is typing.
- 5272 • **icom_msg:Paused** a user has paused.
- 5273 • **icom_msg:Inactive** a user is inactive.
- 5274 • **icom_msg:Gone** a user is gone.

5275

5276 **4.4.17 InstantMessageFeed**

5277 **4.4.17.1 Description**

5278 An instant message feed contains a set of instant message connections and a queue of outbound instant
5279 messages.

5280 **4.4.17.2 Class Definition**

5281 The InstantMessageFeed class has attribute values:

5282
5283 **localNamespace**
5284 Value: icom_msg
5285
5286 **localName**
5287 Value: InstantMessageFeed

5288

5289 **extendsFrom**

5290 Value: icom_core:Entity

5291

5292 **stereotype**

5293 Value: primary

5294

5295 **description**

5296 Value: An instant message feed contains a set of instant message connections and a queue of

5297 outbound instant messages.

5298

5299 **propertyDefinitions**

5300 The values for this attribute are defined in Section 4.4.17.3.

5301 **4.4.17.3 Property Definitions**

5302 The InstantMessageFeed class inherits property definitions from super classes.

5303 The InstantMessageFeed class MUST have the property definitions:

5304

5305 **icom_msg:connection**

5306	Description:	One or more instant messaging connections.
5307	Required:	False
5308	Inherited:	False
5309	Property Type:	icom_msg:InstantMessageConnection
5310	Cardinality:	Multi
5311	Updatability:	Read Only

5312

5313 **icom_msg:outboundInstantMessage**

5314	Description:	Outbound instant messages.
5315	Required:	False
5316	Inherited:	False
5317	Property Type:	icom_msg:InstantMessage
5318	Cardinality:	Multi
5319	Updatability:	Write Only

5320

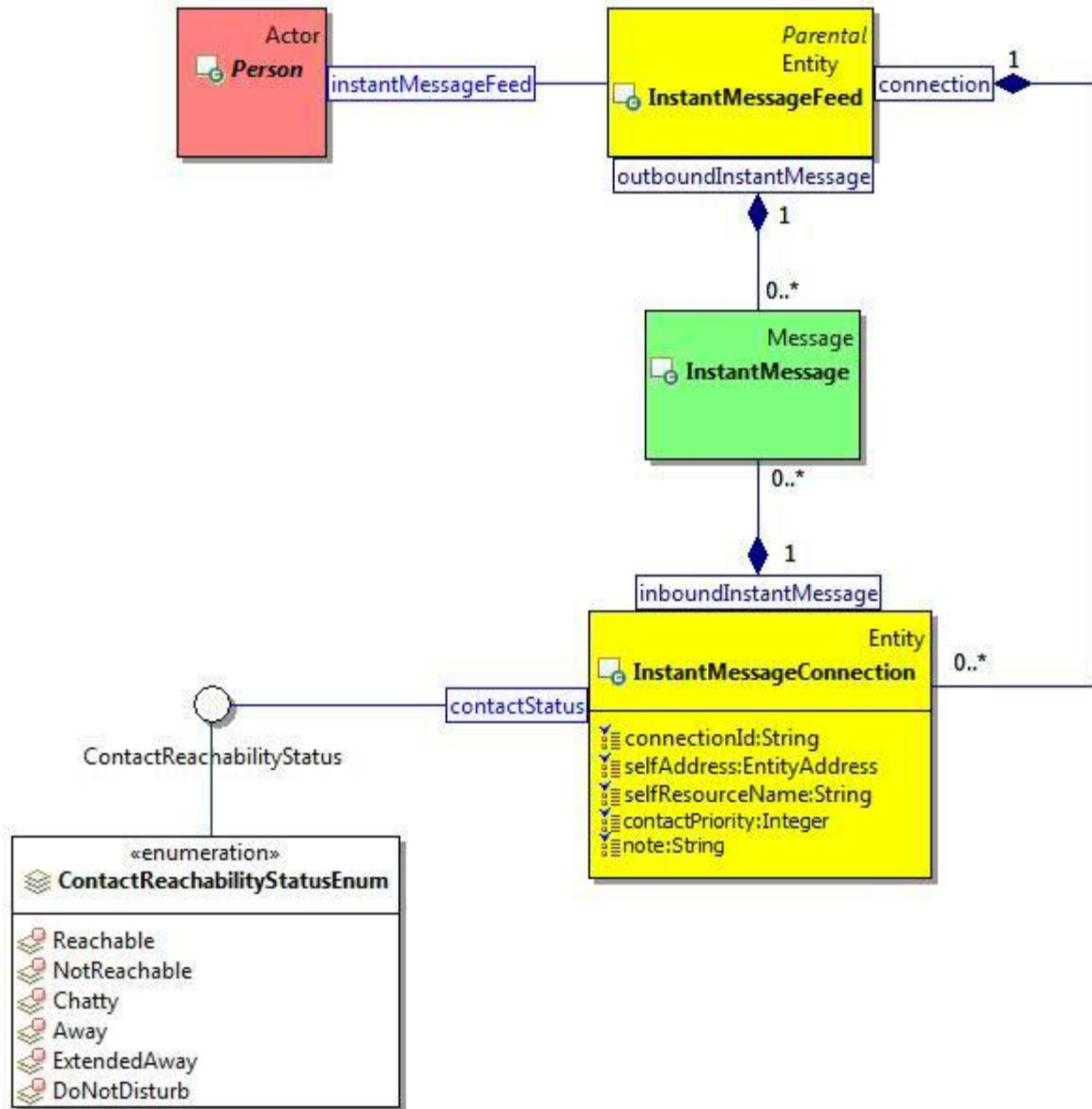


Figure 31: Instant Message Feed and Connection Class Diagram.

4.4.18 InstantMessageConnection

4.4.18.1 Description

An instant message connection contains queues for inbound instant messages.

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

4.4.18.2 Class Definition

The InstantMessageConnection class has attribute values:

localNamespace

Value: icom_msg

localName

Value: InstantMessageConnection

extendsFrom

Value: icom_core:Entity

stereotype

Value: primary

description

Value: An instant message connection contains queues for inbound instant messages.

propertyDefinitions

The values for this attribute are defined in Section 4.4.18.3.

4.4.18.3 Property Definitions

The InstantMessageConnection class inherits property definitions from super classes.

The InstantMessageConnection class MUST have the property definitions:

icom_msg:connectionId

Description: An identifier of a connection.

Required: False

Inherited: False

Property Type: String

Cardinality: Single

Updatability: Read Only

icom_msg:selfAddress

Description: Address of a presentity who opens a connection.

Required: True

Inherited: False

Property Type: IRI

Cardinality: Single

Updatability: On Create

5369	icom_msg:selfResourceName	
5370	Description:	Resource name associated with a connection.
5371	Required:	True
5372	Inherited:	False
5373	Property Type:	String
5374	Cardinality:	Single
5375	Updatability:	On Create
5376		
5377	icom_msg:inboundInstantMessage	
5378	Description:	Inbound instant messages.
5379	Required:	False
5380	Inherited:	False
5381	Property Type:	icom_msg:InstantMessage
5382	Cardinality:	Multi
5383	Updatability:	Read Only
5384		
5385	icom_presence:contactStatus	
5386	Description:	Reachability status to be propagated to an associated contact
5387		method in presence.
5388	Required:	False
5389	Inherited:	False
5390	Property Type:	icom_presence:ContactReachabilityStatus
5391	Cardinality:	Single
5392	Updatability:	Write Only
5393		
5394	icom_presence:contactPriority	
5395	Description:	Priority to be propagated to an associated contact method in
5396		presence.
5397	Required:	False
5398	Inherited:	False
5399	Property Type:	Integer
5400	Cardinality:	Single
5401	Updatability:	Write Only
5402		
5403	icom_presence:note	
5404	Description:	Note to be propagated to an associated contact method in
5405		presence.
5406	Required:	False
5407	Inherited:	False
5408	Property Type:	String
5409	Cardinality:	Single
5410	Updatability:	Write Only
5411		

4.5 Presence Module

4.5.1 Presence

4.5.1.1 Description

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

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

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

4.5.1.2 Class Definition

The Presence class has attribute values:

localNamespace

Value: icom_presence

localName

Value: Presence

extendsFrom

Value: icom_core:Identifiable

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.5.1.3.

4.5.1.3 Property Definitions

The Presence class inherits property definitions from super classes.

The Presence class MUST have the property definitions:

icom_core:lastModificationDate

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

Required: False

Inherited: False

Property Type: DateTime

Cardinality: Single

Updatability: Read Only

5451	icom_core:location	
5452	Description:	Location of a presentity.
5453	Required:	False
5454	Inherited:	False
5455	Property Type:	icom_core:Location
5456	Cardinality:	Single
5457	Updatability:	Read Only
5458		
5459	icom_presence:editMode	
5460	Description:	Indicates a mode which determines whether a presence is
5461		editable.
5462	Required:	False
5463	Inherited:	False
5464	Property Type:	icom_presence:PresenceEditMode
5465	Cardinality:	Single
5466	Updatability:	Read Only
5467		
5468	icom_presence:contactMethod	
5469	Description:	A collection of contact methods describing how to contact a
5470		presentity. A viewer may choose any one of the contact
5471		methods based on circumstances.
5472	Required:	False
5473	Inherited:	False
5474	Property Type:	icom_presence:ContactMethod
5475	Cardinality:	Multi
5476	Updatability:	Read Only
5477		
5478	icom_presence:activity	
5479	Description:	A collection of activities describing what a presentity is doing.
5480	Required:	False
5481	Inherited:	False
5482	Property Type:	icom_presence:Activity
5483	Cardinality:	Multi
5484	Updatability:	Read Only
5485		
5486	The Presence class MAY include additional property definitions which are implementation-defined.	
5487		

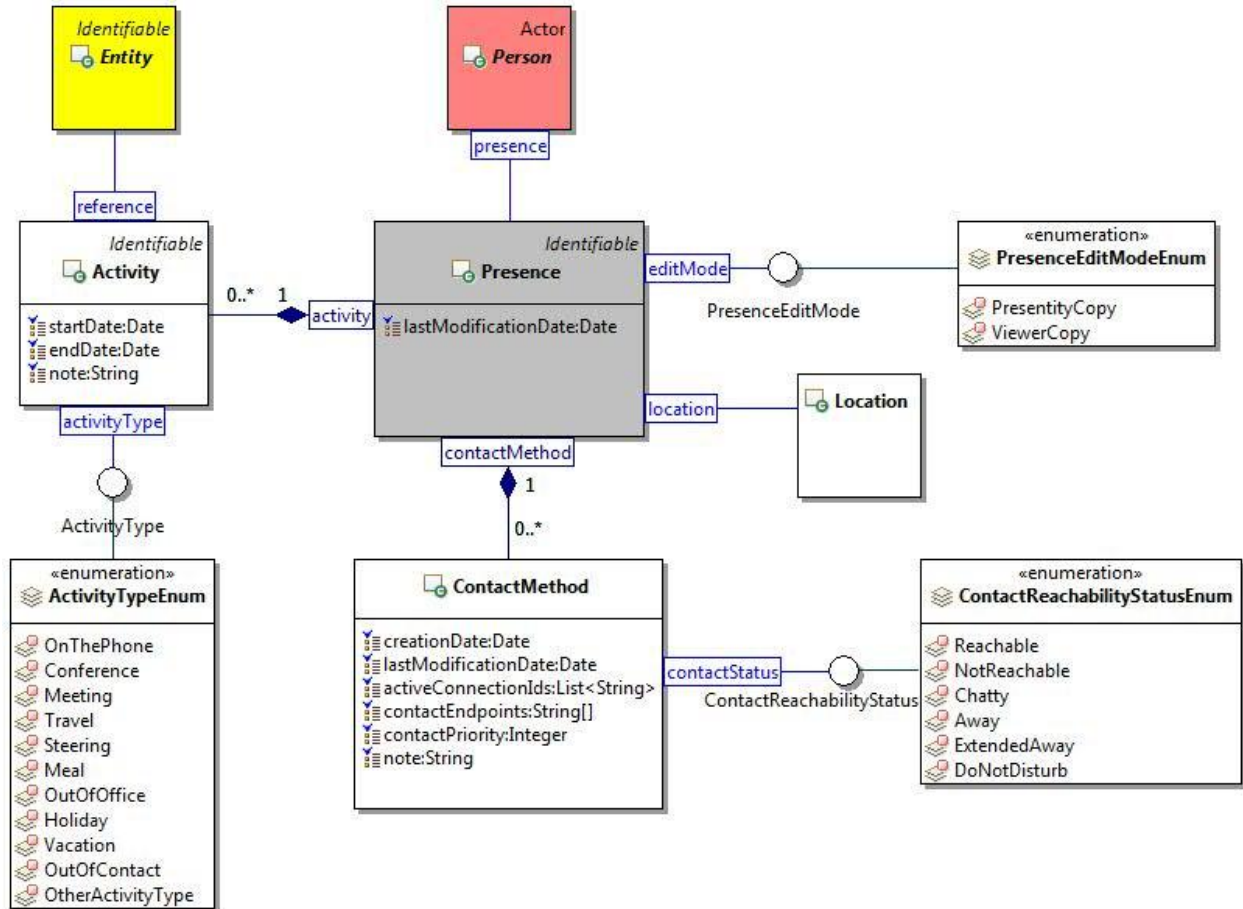


Figure 32: Presence Class Diagram.

4.5.2 PresenceEditMode

4.5.2.1 Description

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

4.5.2.2 Class Definition

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

The PresenceEditMode class has attribute values:

localNamespace

Value: icom_presence

localName

Value: PresenceEditMode

extendsFrom

Value:

5507
5508 **stereotype**
5509 Value: mixin
5510
5511 **description**
5512 Value: PresenceEditMode is a mixin class which defines a mode that indicates whether a
5513 presence is editable.
5514
5515 **propertyDefinitions**
5516 The values for this attribute are defined in Section 4.5.2.3.

5517 **4.5.2.3 Property Definitions**

5518 The PresenceEditMode class MAY include additional property definitions which are implementation-
5519 defined.
5520

5521 **4.5.3 PresenceEditModeEnum**

5522 The PresenceEditModeEnum class is an enum class that enumerates the instances each of which
5523 expresses a mode that indicates whether a presence is editable.

5524 The PresenceEditModeEnum class has attribute values:

5525
5526 **localNamespace**
5527 Value: icom_presence
5528
5529 **localName**
5530 Value: PresenceEditModeEnum
5531
5532 **extendsFrom**
5533 Value: icom_presence:PresenceEditMode
5534
5535 **stereotype**
5536 Value: primary
5537
5538 **isEnumeration**
5539 Value: TRUE
5540
5541 **description**
5542 Value: A mode that indicates whether a presence is editable.
5543
5544 **instances**
5545 Value: <icom_presence:PresententityCopy, icom_presence:ViewerCopy>
5546
5547 ICOM defines two presence editable modes:

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

4.5.4 ContactMethod

4.5.4.1 Description

A contact method object describes reachability circumstances of a presentity.

4.5.4.2 Class Definition

The ContactMethod class has attribute values:

localNamespace

Value: icom_presence

localName

Value: ContactMethod

extendsFrom

Value:

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.5.4.3

4.5.4.3 Property Definitions

The ContactMethod class MUST have the property definitions:

icom_core:creationDate

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

Required: False

Inherited: False

Property Type: DateTime

Cardinality: Single

Updatability: Read Only

5587	icom_core:lastModificationDate	
5588	Description:	Last modification date and time of information in a contact
5589		method.
5590	Required:	False
5591	Inherited:	False
5592	Property Type:	DateTime
5593	Cardinality:	Single
5594	Updatability:	Read Only
5595		
5596	icom_presence:activeConnectionId	
5597	Description:	A list of active connection ids of a presentity.
5598	Required:	False
5599	Inherited:	False
5600	Property Type:	String
5601	Cardinality:	Multi
5602	Updatability:	Read Only
5603		
5604	icom_presence:contactEndpoint	
5605	Description:	A list of endpoints or IRIs for contacting a presentity.
5606	Required:	False
5607	Inherited:	False
5608	Property Type:	String
5609	Cardinality:	Multi
5610	Updatability:	Read Only
5611		
5612	icom_presence:contactPriority	
5613	Description:	Priority of a contact method relative to other contact methods
5614		in a presence.
5615	Required:	False
5616	Inherited:	False
5617	Property Type:	Integer
5618	Cardinality:	Single
5619	Updatability:	Read Only
5620		
5621	icom_presence:contactStatus	
5622	Description:	Status of a contact method in a presence.
5623	Required:	False
5624	Inherited:	False
5625	Property Type:	icom_presence:ContactReachabilityStatus
5626	Cardinality:	Single
5627	Updatability:	Read Only
5628		

5629	icom_presence:note	
5630	Description:	A note about contacting a presentity.
5631	Required:	False
5632	Inherited:	False
5633	Property Type:	String
5634	Cardinality:	Single
5635	Updatability:	Read Only
5636		

5637 4.5.5 ContactReachabilityStatus

5638 4.5.5.1 Description

5639 A contact reachability status is a status of a contact method.

5640 4.5.5.2 Class Definition

5641 The ContactReachabilityStatus class is a mixin class which defines a status of a contact method.

5642 The ContactReachabilityStatus class has attribute values:

5643		
5644	localNamespace	
5645	Value:	icom_presence
5646		
5647	localName	
5648	Value:	ContactReachabilityStatus
5649		
5650	extendsFrom	
5651	Value:	
5652		
5653	stereotype	
5654	Value:	mixin
5655		
5656	description	
5657	Value:	ContactReachabilityStatus is a mixin class which defines a status of a contact method.
5658		
5659	propertyDefinitions	
5660	The values for this attribute are defined in Section 4.5.5.3.	

5661 4.5.5.3 Property Definitions

5662 The ContactReachabilityStatus class MAY include additional property definitions which are
5663 implementation-defined.

5664

5665 4.5.6 ContactReachabilityStatusEnum

5666 The ContactReachabilityStatusEnum class is an enum class that enumerates the instances each of which
5667 expresses a reachability status of a contact method.

5668 The ContactReachabilityStatusEnum class has attribute values:

5669

5670 **localNamespace**

5671 Value: icom_presence

5672

5673 **localName**

5674 Value: ContactReachabilityStatusEnum

5675

5676 **extendsFrom**

5677 Value: icom_presence:ContactReachabilityStatus

5678

5679 **stereotype**

5680 Value: primary

5681

5682 **isEnumeration**

5683 Value: TRUE

5684

5685 **description**

5686 Value: A reachability status of a contact method.

5687

5688 **instances**

5689 Value: <icom_presence:Reachable, icom_presence:NotReachable, icom_presence:Chatty,

5690 icom_presence:Away, icom_presence:ExtendedAway, icom_presence:DoNotDisturb>

5691

5692 ICOM defines six reachability status:

- 5693 • **icom_presence:Reachable** a presentity is reachable through a contact method.
- 5694 • **icom_presence:NotReachable** a presentity is not reachable through a contact method.
- 5695 • **icom_presence:Chatty** a presentity is chatty.
- 5696 • **icom_presence:Away** a presentity is away.
- 5697 • **icom_presence:ExtendedAway** a presentity is away for an extended period.
- 5698 • **icom_presence:DoNotDisturb** a presentity prefers not to be disturbed.

5699

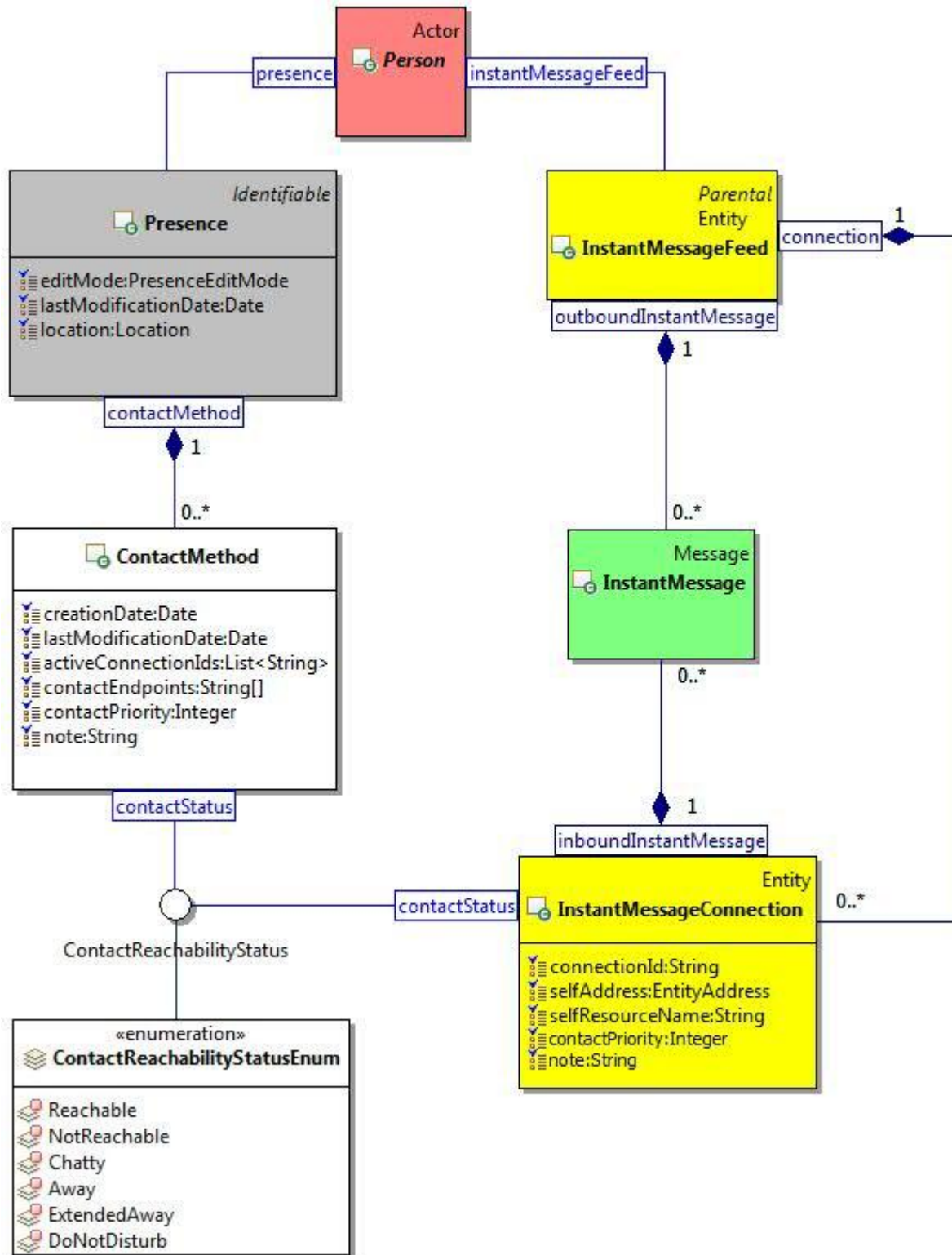


Figure 33: Presence Contact Method and Instant Message Connection Class Diagram.

4.5.7 Activity

4.5.7.1 Description

An activity object describes what a presentity is doing.

4.5.7.2 Class Definition

The Activity class has attribute values:

localNamespace

Value: icom_presence

localName

Value: Activity

extendsFrom

Value:

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.5.7.3.

4.5.7.3 Property Definitions

The Activity class MUST have the property definitions:

icom_core:startDate

Description: Start date and time of an activity.

Required: True

Inherited: False

Property Type: DateTime

Cardinality: Single

Updatability: Read Write

icom_core:endDate

Description: End date and time of an activity.

Required: True

Inherited: False

Property Type: DateTime

Cardinality: Single

5743	Updatability:	Read Write
5744		
5745	icom_presence:activityType	
5746	Description:	Type of an activity.
5747	Required:	true
5748	Inherited:	False
5749	Property Type:	icom_presence:ActivityType
5750	Cardinality:	Single
5751	Updatability:	Read Write
5752		
5753	icom_presence:note	
5754	Description:	A note describing an activity.
5755	Required:	False
5756	Inherited:	False
5757	Property Type:	String
5758	Cardinality:	Single
5759	Updatability:	Read Write
5760		
5761	icom_presence:reference	
5762	Description:	An entity, such as occurrence, task, conference, etc., which is
5763		the source of or reference for an activity.
5764	Required:	False
5765	Inherited:	False
5766	Property Type:	icom_core:Entity
5767	Cardinality:	Single
5768	Updatability:	Read Write
5769		

5770 4.5.8 ActivityType

5771 4.5.8.1 Description

5772 An activity type is a vocabulary of activities for rich presence information model.

5773 4.5.8.2 Class Definition

5774 The ActivityType class is a mixin class which defines an activity.

5775 The ActivityType class has attribute values:

5776		
5777	localNamespace	
5778	Value:	icom_presence
5779		
5780	localName	
5781	Value:	ActivityType
5782		

5783 **extendsFrom**
5784 Value:
5785
5786 **stereotype**
5787 Value: mixin
5788
5789 **description**
5790 Value: ActivityType is a mixin class which defines a type of activity.
5791
5792 **propertyDefinitions**
5793 The values for this attribute are defined in Section 4.5.8.3.

5794 **4.5.8.3 Property Definitions**

5795 The ActivityType class MAY include additional property definitions which are implementation-defined.
5796

5797 **4.5.9 ActivityTypeEnum**

5798 The ActivityTypeEnum class is an enum class that enumerates the instances each of which expresses a
5799 type of activity.

5800 The ActivityTypeEnum class has attribute values:

5801
5802 **localNamespace**
5803 Value: icom_presence
5804
5805 **localName**
5806 Value: ActivityTypeEnum
5807
5808 **extendsFrom**
5809 Value: icom_presence:ActivityType
5810
5811 **stereotype**
5812 Value: primary
5813
5814 **isEnumeration**
5815 Value: TRUE
5816
5817 **description**
5818 Value: A type of activity.
5819
5820 **instances**
5821 Value: <icom_presence:OnThePhone, icom_presence:Conference, icom_presence:Meeting,
5822 icom_presence:Travel, icom_presence:Steering, icom_presence:Meal,
5823 icom_presence:OutOfOffice, icom_presence:Holiday, icom_presence:Vacation,
5824 icom_presence:OutOfContact, icom_presence:OtherActivityType>

5825
5826 ICOM defines eleven activity types:
5827 • **icom_presence:OnThePhone** a presentity is on the phone.
5828 • **icom_presence:Conference** a presentity is in a conference.
5829 • **icom_presence:Meeting** a presentity is in a meeting.
5830 • **icom_presence:Travel** a presentity is traveling.
5831 • **icom_presence:Steering** a presentity is steering a vehicle.
5832 • **icom_presence:Meal** a presentity is having a meal.
5833 • **icom_presence:OutOfOffice** a presentity is out of office.
5834 • **icom_presence:Holiday** a presentity is on holiday.
5835 • **icom_presence:Vacation** a presentity is on vacation.
5836 • **icom_presence:OutOfContact** a presentity is out of contact.
5837 • **icom_presence:OtherActivityType** a presentity is involved in an unspecified activity.
5838

5839 **4.6 Address Book Module**

5840 **4.6.1 AddressBook**

5841 **4.6.1.1 Description**

5842 An address book is a folder that contains sub-address books and addressable contacts.

5843 **4.6.1.2 Class Definition**

5844 The AddressBook class has attribute values:

5845
5846 **localNamespace**
5847 Value: icom_card
5848
5849 **localName**
5850 Value: AddressBook
5851
5852 **extendsFrom**
5853 Value: icom_core:Folder
5854
5855 **stereotype**
5856 Value: primary
5857
5858 **description**
5859 Value: An address book is a folder that contains sub-address books and addressable contacts.
5860
5861 **propertyDefinitions**
5862 The values for this attribute are defined in Section 4.6.1.3.

5863 **4.6.1.3 Property Definitions**

5864 The AddressBook class inherits property definitions from super classes.

5865 The AddressBook class MUST have the property definitions:

5866

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

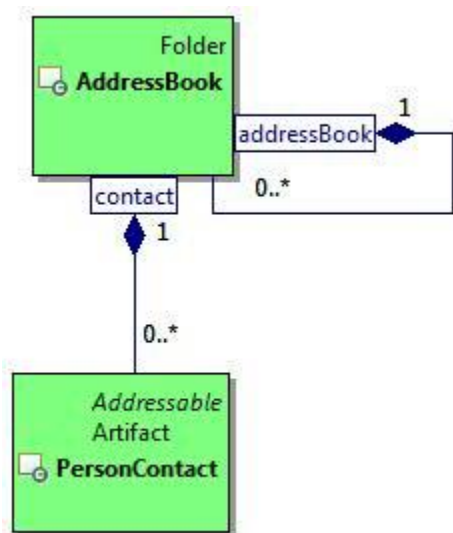
5874

5875

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

5882

5883 The AddressBook class MAY include additional property definitions which are implementation-defined.



5885
5886 *Figure 34: Address Book Class Diagram.*

5887

5888 **4.6.2 PersonContact**

5889 **4.6.2.1 Description**

5890 A person contact is an artifact that contains address information about a person.

4.6.2.2 Class Definition

The PersonContact class has attribute values:

localNamespace

Value: icom_card

localName

Value: PersonContact

extendsFrom

Value: icom_core:Artifact, icom_core:Addressable

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.6.2.3.

4.6.2.3 Property Definitions

The PersonContact class inherits property definitions from super classes.

The PersonContact class MUST have the property definitions:

icom_core:timeZone

Description: Time zone of a person.

Required: False

Inherited: False

Property Type: icom_core:TimeZone

Cardinality: Single

Updatability: Read Write

icom_core:givenName

Description: Given name of a person.

Required: False

Inherited: False

Property Type: String

Cardinality: Single

Updatability: Read Write

5931	icom_core:middleName	
5932	Description:	Middle name of a person. Can include multiple names concatenated.
5933		
5934	Required:	False
5935	Inherited:	False
5936	Property Type:	String
5937	Cardinality:	Single
5938	Updatability:	Read Write
5939		
5940	icom_core:familyName	
5941	Description:	Family name of a person.
5942	Required:	False
5943	Inherited:	False
5944	Property Type:	String
5945	Cardinality:	Single
5946	Updatability:	Read Write
5947		
5948	icom_core:prefix	
5949	Description:	Prefix of a person's name.
5950	Required:	False
5951	Inherited:	False
5952	Property Type:	String
5953	Cardinality:	Single
5954	Updatability:	Read Write
5955		
5956	icom_core:suffix	
5957	Description:	Suffix of a person's name.
5958	Required:	False
5959	Inherited:	False
5960	Property Type:	String
5961	Cardinality:	Single
5962	Updatability:	Read Write
5963		
5964	icom_core:nickname	
5965	Description:	Nickname of a person.
5966	Required:	False
5967	Inherited:	False
5968	Property Type:	String
5969	Cardinality:	Multi
5970	Updatability:	Read Write
5971		
5972	icom_core:jobTitle	
5973	Description:	Job title of a person.

5974	Required:	False
5975	Inherited:	False
5976	Property Type:	String
5977	Cardinality:	Single
5978	Updatability:	Read Write
5979		
5980	icom_core:department	
5981	Description:	A person's affiliated department.
5982	Required:	False
5983	Inherited:	False
5984	Property Type:	String
5985	Cardinality:	Single
5986	Updatability:	Read Write
5987		
5988	icom_core:officeLocation	
5989	Description:	Location of a person's department.
5990	Required:	False
5991	Inherited:	False
5992	Property Type:	String
5993	Cardinality:	Single
5994	Updatability:	Read Write
5995		
5996	icom_core:company	
5997	Description:	A person's affiliated company.
5998	Required:	False
5999	Inherited:	False
6000	Property Type:	String
6001	Cardinality:	Single
6002	Updatability:	Read Write
6003		
6004	icom_core:profession	
6005	Description:	A person's profession.
6006	Required:	False
6007	Inherited:	False
6008	Property Type:	String
6009	Cardinality:	Single
6010	Updatability:	Read Write
6011		
6012	icom_content:attachment	
6013	Description:	One or more content attachments in a contact.
6014	Required:	False
6015	Inherited:	False

6016	Property Type:	icom_content:AttachedItem
6017	Cardinality:	Multi
6018	Updatability:	Read Write
6019		
6020	icom_card:bookmark	
6021	Description:	A person which is bookmarked by a contact.
6022	Required:	False
6023	Inherited:	False
6024	Property Type:	icom_core:Person
6025	Cardinality:	Single
6026	Updatability:	On Create
6027		
6028	The PersonContact class MAY include additional property definitions which are implementation-defined.	
6029		

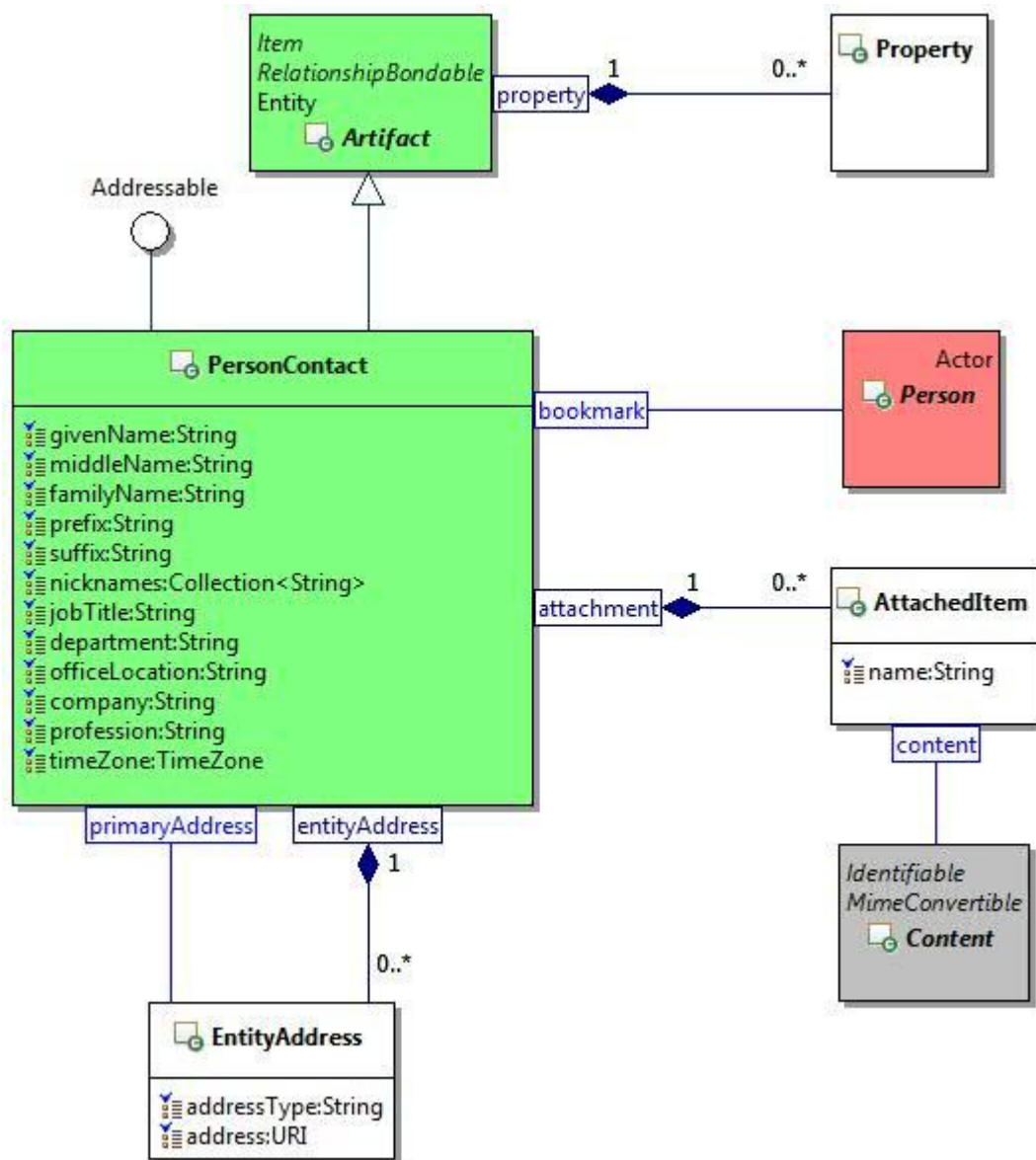


Figure 35: Person Contact Class Diagram.

4.7 Calendar Module

4.7.1 Calendar

4.7.1.1 Description

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

4.7.1.2 Class Definition

The Calendar class has attribute values:

6040 **localNamespace**
6041 Value: icom_cal
6042
6043 **localName**
6044 Value: Calendar
6045
6046 **extendsFrom**
6047 Value: icom_core:Folder
6048
6049 **stereotype**
6050 Value: primary
6051
6052 **description**
6053 Value: A calendar contains time management artifacts that include occurrences and occurrence
6054 series.
6055
6056 **propertyDefinitions**
6057 The values for this attribute are defined in 4.7.1.3.

6058 **4.7.1.3 Property Definitions**

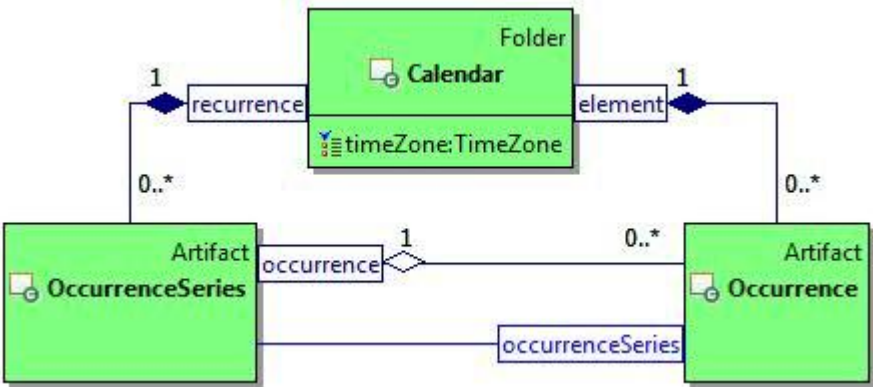
6059 The Calendar class inherits property definitions from super classes.
6060 The Calendar class **MUST** have the property definitions:

6061
6062 **icom_core:timeZone**
6063 Description: Time zone setting for a calendar.
6064 Required: True
6065 Inherited: False
6066 Property Type: icom_core:TimeZone
6067 Cardinality: Single
6068 Updatability: Read Write

6069
6070 **icom_core:element**
6071 Description: Elements of a calendar.
6072 Required: False
6073 Inherited: True
6074 Property Type: icom_cal:Occurrence
6075 Cardinality: Multi
6076 Updatability: Read Only

6077
6078 **icom_cal:recurrence**
6079 Description: Occurrence series of a calendar.
6080 Required: False
6081 Inherited: False

6082 Property Type: icom_cal:OccurrenceSeries
6083 Cardinality: Multi
6084 Updatability: Read Only
6085



6086
6087 *Figure 36: Calendar Class Diagram.*
6088

6089 **4.7.2 OccurrenceSeries**

6090 **4.7.2.1 Description**

6091 An occurrence series represents a series of occurrences associated with the same calendar event.

6092 **4.7.2.2 Class Definition**

6093 The OccurrenceSeries class has attribute values:

6094
6095 **localNamespace**
6096 Value: icom_cal
6097
6098 **localName**
6099 Value: OccurrenceSeries
6100
6101 **extendsFrom**
6102 Value: icom_core:Artifact
6103
6104 **stereotype**
6105 Value: primary
6106
6107 **description**
6108 Value: An occurrence series represents a series of occurrences associated with the same
6109 calendar event.
6110
6111 **propertyDefinitions**
6112 The values for this attribute are defined in 4.7.2.3.

4.7.2.3 Property Definitions

The OccurrenceSeries class inherits property definitions from super classes.

The OccurrenceSeries class MUST have the property definitions:

icom_core:location

Description:	Location of an occurrence series.
Required:	False
Inherited:	False
Property Type:	icom_core:Location
Cardinality:	Single
Updatability:	Read Write

icom_core:organizer

Description:	Organizer of an occurrence series.
Required:	True
Inherited:	False
Property Type:	icom_core:Participant
Cardinality:	Single
Updatability:	On Create

icom_core:participant

Description:	Participants in an occurrence series.
Required:	False
Inherited:	False
Property Type:	icom_cal:OccurrenceParticipant
Cardinality:	Multi
Updatability:	Read Write

icom_core:priority

Description:	Priority for an attendee of an occurrence series.
Required:	False
Inherited:	False
Property Type:	icom_core:Priority
Cardinality:	Single
Updatability:	Read Write

icom_content:attachment

Description:	One or more content attachments in an occurrence series.
Required:	False
Inherited:	False
Property Type:	icom_content:AttachedItem
Cardinality:	Multi

6155	Updatability:	Read Write
6156		
6157	icom_cal:recurrenceStartDate	
6158	Description:	Start date and time of an occurrence series.
6159	Required:	True
6160	Inherited:	False
6161	Property Type:	DateTime
6162	Cardinality:	Single
6163	Updatability:	On Create
6164		
6165	icom_cal:recurrenceStartDateResolution	
6166	Description:	Resolution of start date and time of an occurrence series.
6167	Required:	True
6168	Inherited:	False
6169	Property Type:	icom_core:DateTimeResolution
6170	Cardinality:	Single
6171	Updatability:	On Create
6172		
6173	icom_cal:duration	
6174	Description:	Duration of each occurrence in an occurrence series.
6175	Required:	True
6176	Inherited:	False
6177	Property Type:	Duration
6178	Cardinality:	Single
6179	Updatability:	On Create
6180		
6181	icom_cal:recurrenceRule	
6182	Description:	A recurrence rule of an occurrence series.
6183	Required:	True
6184	Inherited:	False
6185	Property Type:	String
6186	Cardinality:	Single
6187	Updatability:	On Create
6188		
6189	icom_cal:occurrenceStatus	
6190	Description:	Status of an occurrence series.
6191	Required:	True
6192	Inherited:	False
6193	Property Type:	icom_cal:OccurrenceStatus
6194	Cardinality:	Single
6195	Updatability:	Read Write
6196		

6197	icom_cal:occurrenceType	
6198	Description:	Type of an occurrence series.
6199	Required:	True
6200	Inherited:	False
6201	Property Type:	icom_cal:OccurrenceType
6202	Cardinality:	Single
6203	Updatability:	Read Write
6204		
6205	icom_cal:editMode	
6206	Description:	Indicates a mode which determines whether an occurrence
6207		series is editable.
6208	Required:	False
6209	Inherited:	False
6210	Property Type:	icom_cal:OccurrenceEditMode
6211	Cardinality:	Single
6212	Updatability:	Read Only
6213		
6214	icom_cal:occurrence	
6215	Description:	Occurrences in an occurrence series.
6216	Required:	False
6217	Inherited:	False
6218	Property Type:	icom_cal:Occurrence
6219	Cardinality:	Multi
6220	Updatability:	Read Only
6221		
6222	icom_cal:attendee	
6223	Description:	An attendee of an occurrence series.
6224	Required:	False
6225	Inherited:	False
6226	Property Type:	icom_core:Participant
6227	Cardinality:	Single
6228	Updatability:	Read Only
6229		
6230	icom_cal:attendeeParticipantStatus	
6231	Description:	Participation status for an attendee of an occurrence series.
6232	Required:	False
6233	Inherited:	False
6234	Property Type:	icom_cal:OccurrenceParticipantStatus
6235	Cardinality:	Single
6236	Updatability:	Read Write
6237		

6238	icom_cal:transparency	
6239	Description:	Participant transparency for an attendee of an occurrence series.
6240		
6241	Required:	False
6242	Inherited:	False
6243	Property Type:	icom_cal:OccurrenceParticipantTransparency
6244	Cardinality:	Single
6245	Updatability:	Read Write
6246		
6247	icom_cal:attendeeProperty	
6248	Description:	Extensible properties for an attendee of an occurrence series.
6249	Required:	False
6250	Inherited:	False
6251	Property Type:	icom_meta:Property
6252	Cardinality:	Multi
6253	Updatability:	Read Write
6254		
6255	icom_conf:conference	
6256	Description:	One or more conferences for an occurrence series.
6257	Required:	False
6258	Inherited:	False
6259	Property Type:	icom_conf:Conference
6260	Cardinality:	Multi
6261	Updatability:	Read Write
6262		

6277

6278 **extendsFrom**

6279 Value: icom_core:Artifact

6280

6281 **stereotype**

6282 Value: primary

6283

6284 **description**

6285 Value: An occurrence represents an event in a calendar.

6286

6287 **propertyDefinitions**

6288 The values for this attribute are defined in 4.7.3.3.

6289

6290 4.7.3.3 Property Definitions

6291 The Occurrence class inherits property definitions from super classes.

6292 The Occurrence class MUST have the property definitions:

6293

6294 **icom_core:location**

6295 Description:	Location of an occurrence.
6296 Required:	False
6297 Inherited:	False
6298 Property Type:	icom_core:Location
6299 Cardinality:	Single
6300 Updatability:	Read Write

6301

6302 **icom_core:organizer**

6303 Description:	Organizer of an occurrence.
6304 Required:	True
6305 Inherited:	False
6306 Property Type:	icom_core:Participant
6307 Cardinality:	Single
6308 Updatability:	On Create

6309

6310 **icom_core:participant**

6311 Description:	Participants of an occurrence.
6312 Required:	False
6313 Inherited:	False
6314 Property Type:	icom_cal:OccurrenceParticipant
6315 Cardinality:	Multi
6316 Updatability:	Read Write

6317

6318	icom_core:priority	
6319	Description:	Priority for an attendee of an occurrence.
6320	Required:	False
6321	Inherited:	False
6322	Property Type:	icom_core:Priority
6323	Cardinality:	Single
6324	Updatability:	Read Write
6325		
6326	icom_core:startDate	
6327	Description:	Start date and time of an occurrence.
6328	Required:	True
6329	Inherited:	False
6330	Property Type:	DateTime
6331	Cardinality:	Single
6332	Updatability:	On Create
6333		
6334	icom_core:startDateResolution	
6335	Description:	Resolution of start date and time of an occurrence.
6336	Required:	True
6337	Inherited:	False
6338	Property Type:	icom_core:DateTimeResolution
6339	Cardinality:	Single
6340	Updatability:	On Create
6341		
6342	icom_core:endDate	
6343	Description:	End date and time of an occurrence.
6344	Required:	True
6345	Inherited:	False
6346	Property Type:	DateTime
6347	Cardinality:	Single
6348	Updatability:	On Create
6349		
6350	icom_core:endDateResolution	
6351	Description:	Resolution of end date and time of an occurrence.
6352	Required:	True
6353	Inherited:	False
6354	Property Type:	icom_core:DateTimeResolution
6355	Cardinality:	Single
6356	Updatability:	On Create
6357		
6358	icom_content:attachment	
6359	Description:	One or more content attachments in an occurrence.

6360	Required:	False
6361	Inherited:	False
6362	Property Type:	icom_content:AttachedItem
6363	Cardinality:	Multi
6364	Updatability:	Read Write
6365		
6366	icom_cal:occurrenceSeries	
6367	Description:	An occurrence is part of this occurrence series.
6368	Required:	False
6369	Inherited:	False
6370	Property Type:	icom_cal:OccurrenceSeries
6371	Cardinality:	Single
6372	Updatability:	Read Only
6373		
6374	icom_cal:fromRecurringOccurrenceSeries	
6375	Description:	Occurrence is part of a recurring occurrence series.
6376	Required:	False
6377	Inherited:	False
6378	Property Type:	Boolean
6379	Cardinality:	Single
6380	Updatability:	Read Only
6381		
6382	icom_cal:exceptionToOccurrenceSeries	
6383	Description:	Occurrence is an exception to an occurrence series.
6384	Required:	False
6385	Inherited:	False
6386	Property Type:	Boolean
6387	Cardinality:	Single
6388	Updatability:	Read Only
6389		
6390	icom_cal:occurrenceStatus	
6391	Description:	Status of an occurrence.
6392	Required:	True
6393	Inherited:	False
6394	Property Type:	icom_cal:OccurrenceStatus
6395	Cardinality:	Single
6396	Updatability:	Read Write
6397		
6398	icom_cal:occurrenceType	
6399	Description:	Type of an occurrence.
6400	Required:	True
6401	Inherited:	False

6402	Property Type:	icom_cal:OccurrenceType
6403	Cardinality:	Single
6404	Updatability:	Read Write
6405		
6406	icom_cal:editMode	
6407	Description:	Indicates a mode which determines whether an occurrence is
6408		editable.
6409	Required:	False
6410	Inherited:	False
6411	Property Type:	icom_cal:OccurrenceEditMode
6412	Cardinality:	Single
6413	Updatability:	Read Only
6414		
6415	icom_cal:attendee	
6416	Description:	An attendee of an occurrence.
6417	Required:	False
6418	Inherited:	False
6419	Property Type:	icom_core:Participant
6420	Cardinality:	Single
6421	Updatability:	Read Only
6422		
6423	icom_cal:attendeeParticipantStatus	
6424	Description:	Status for an attendee of an occurrence.
6425	Required:	False
6426	Inherited:	False
6427	Property Type:	icom_cal:OccurrenceParticipantStatus
6428	Cardinality:	Single
6429	Updatability:	Read Write
6430		
6431	icom_cal:transparency	
6432	Description:	Transparency for an attendee of an occurrence.
6433	Required:	False
6434	Inherited:	False
6435	Property Type:	icom_cal:OccurrenceParticipantTransparency
6436	Cardinality:	Single
6437	Updatability:	Read Write
6438		
6439	icom_cal:attendeeProperty	
6440	Description:	Extensible properties for an attendee of an occurrence.
6441	Required:	False
6442	Inherited:	False
6443	Property Type:	icom_meta:Property
6444	Cardinality:	Multi

6445 Updatability: Read Write

6446

6447 **icom_conf:conference**

6448 Description: One or more conferences for an occurrence.

6449 Required: False

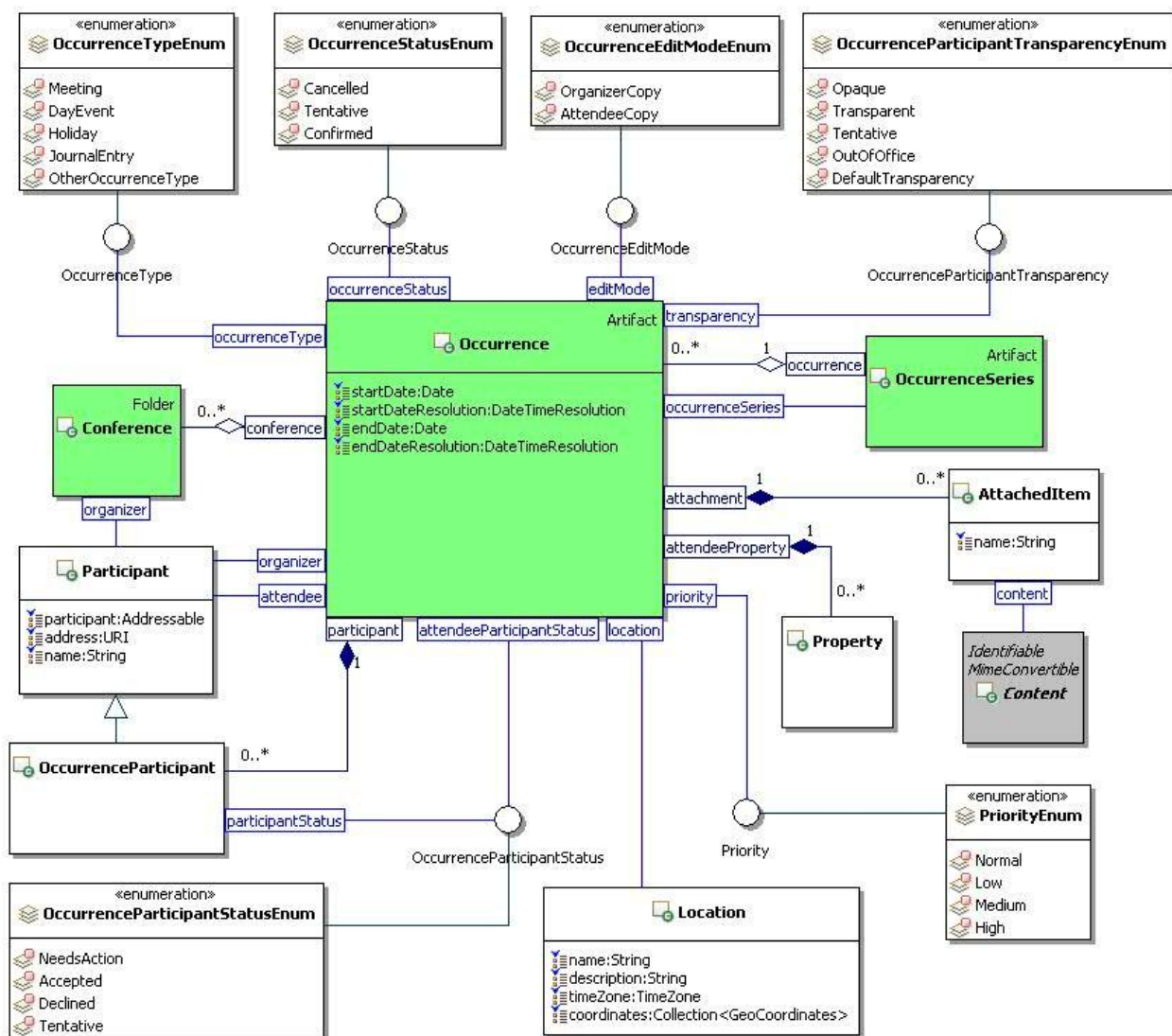
6450 Inherited: False

6451 Property Type: icom_conf:Conference

6452 Cardinality: Multi

6453 Updatability: Read Write

6454



6455

6456 Figure 38: Occurrence Class Diagram.

6457

6458 **4.7.4 OccurrenceStatus**

6459 **4.7.4.1 Description**

6460 An occurrence status is a status of a calendar occurrence.

6461 **4.7.4.2 Class Definition**

6462 The OccurrenceStatus class is a mixin class which defines status of a calendar occurrence.

6463 The OccurrenceStatus class has attribute values:

6464

6465 **localNamespace**

6466 Value: icom_cal

6467

6468 **localName**

6469 Value: OccurrenceStatus

6470

6471 **extendsFrom**

6472 Value:

6473

6474 **stereotype**

6475 Value: mixin

6476

6477 **description**

6478 Value: OccurrenceStatus is a mixin class which defines status of a calendar occurrence.

6479

6480 **propertyDefinitions**

6481 The values for this attribute are defined in Section 4.7.4.3.

6482 **4.7.4.3 Property Definitions**

6483 The OccurrenceStatus class MAY include additional property definitions which are implementation-
6484 defined.

6485

6486 **4.7.5 OccurrenceStatusEnum**

6487 The OccurrenceStatusEnum class is an enum class that enumerates the instances each of which
6488 expresses a status of an occurrence or occurrence series.

6489 The OccurrenceStatusEnum class has attribute values:

6490

6491 **localNamespace**

6492 Value: icom_cal

6493

6494 **localName**

6495 Value: OccurrenceStatusEnum

6496

6497 **extendsFrom**
6498 Value: icom_cal:OccurrenceStatus
6499
6500 **stereotype**
6501 Value: primary
6502
6503 **isEnumeration**
6504 Value: TRUE
6505
6506 **description**
6507 Value: Status of an occurrence or occurrence series.
6508
6509 **instances**
6510 Value: <icom_cal:Cancelled, icom_cal:Tentative, icom_cal:Confirmed>
6511
6512 ICOM defines three occurrence status:

- 6513 • **icom_cal:Cancelled** an occurrence or occurrence series is cancelled.
- 6514 • **icom_cal:Tentative** an occurrence or occurrence series is tentative.
- 6515 • **icom_cal:Confirmed** an occurrence or occurrence series is confirmed.

6516

6517 **4.7.6 OccurrenceType**

6518 **4.7.6.1 Description**

6519 An occurrence type is a category of calendar occurrences.

6520 **4.7.6.2 Class Definition**

6521 The OccurrenceType class is a mixin class which defines a type of occurrence.
6522 The OccurrenceType class has attribute values:

6523
6524 **localNamespace**
6525 Value: icom_cal
6526
6527 **localName**
6528 Value: OccurrenceType
6529
6530 **extendsFrom**
6531 Value:
6532
6533 **stereotype**
6534 Value: mixin
6535

6536 **description**
6537 Value: OccurrenceType is a mixin class which defines a type of occurrence.

6538
6539 **propertyDefinitions**
6540 The values for this attribute are defined in Section 4.7.6.3.

6541 **4.7.6.3 Property Definitions**

6542 The OccurrenceType class MAY include additional property definitions which are implementation-defined.
6543

6544 **4.7.7 OccurrenceTypeEnum**

6545 The OccurrenceTypeEnum class is an enum class that enumerates the instances each of which
6546 expresses a type of an occurrence or occurrence series.

6547 The OccurrenceTypeEnum class has attribute values:

6548
6549 **localNamespace**
6550 Value: icom_cal
6551
6552 **localName**
6553 Value: OccurrenceTypeEnum
6554
6555 **extendsFrom**
6556 Value: icom_cal:OccurrenceType
6557

6558 **stereotype**
6559 Value: primary
6560

6561 **isEnumeration**
6562 Value: TRUE
6563

6564 **description**
6565 Value: Type of an occurrence or occurrence series.
6566

6567 **instances**
6568 Value: <icom_cal:Meeting, icom_cal:DayEvent, icom_cal:Holiday, icom_cal:JournalEntry,
6569 icom_cal:OtherOccurrenceType>
6570

6571 ICOM defines five occurrence types:

- 6572 • **icom_cal:Meeting** an occurrence or occurrence series is a meeting.
- 6573 • **icom_cal:DayEvent** an occurrence or occurrence series is a day event.
- 6574 • **icom_cal:Holiday** an occurrence or occurrence series is a holiday.
- 6575 • **icom_cal:JournalEntry** an occurrence or occurrence series is a journal entry.
- 6576 • **icom_cal:OtherOccurrenceType** an occurrence or occurrence series is of other type.

6577

6578 **4.7.8 OccurrenceParticipant**

6579 **4.7.8.1 Description**

6580 An occurrence participant object is a participant object that contains an occurrence participant status.

6581 **4.7.8.2 Class Definition**

6582 The OccurrenceParticipant class has attribute values:

6583

6584 **localNamespace**

6585 Value: icom_cal

6586

6587 **localName**

6588 Value: OccurrenceParticipant

6589

6590 **extendsFrom**

6591 Value: icom_core:Participant

6592

6593 **stereotype**

6594 Value: primary

6595

6596 **description**

6597 Value: An occurrence participant object is a participant object that contains an occurrence
6598 participant status.

6599

6600 **propertyDefinitions**

6601 The values for this attribute are defined in Section 4.7.8.3.

6602 **4.7.8.3 Property Definitions**

6603 The OccurrenceParticipant class inherits property definitions from super classes.

6604 The OccurrenceParticipant class MUST have the property definition:

6605

6606 **icom_cal:participantStatus**

6607 Description: Status of an occurrence participant.

6608 Required: False

6609 Inherited: False

6610 Property Type: icom_cal:OccurrenceParticipantStatus

6611 Cardinality: Single

6612 Updatability: Read Write

6613

6614 4.7.9 OccurrenceParticipantStatus

6615 4.7.9.1 Description

6616 An occurrence participant status is a participant's response status for an occurrence or occurrence series.

6617 4.7.9.2 Class Definition

6618 The OccurrenceParticipantStatus class is a mixin class which defines a participant's response status for
6619 an occurrence or occurrence series.

6620 The OccurrenceParticipantStatus class has attribute values:

6621

6622 **localNamespace**

6623 Value: icom_cal

6624

6625 **localName**

6626 Value: OccurrenceParticipantStatus

6627

6628 **extendsFrom**

6629 Value:

6630

6631 **stereotype**

6632 Value: mixin

6633

6634 **description**

6635 Value: OccurrenceParticipantStatus is a mixin class which defines a participant's response
6636 status for an occurrence or occurrence series.

6637

6638 **propertyDefinitions**

6639 The values for this attribute are defined in Section 4.7.9.3.

6640 4.7.9.3 Property Definitions

6641 The OccurrenceParticipantStatus class MAY include additional property definitions which are
6642 implementation-defined.

6643

6644 4.7.10 OccurrenceParticipantStatusEnum

6645 The OccurrenceParticipantStatusEnum class is an enum class that enumerates the instances each of
6646 which expresses a participant's response status for an occurrence or occurrence series.

6647 The OccurrenceParticipantStatusEnum class has attribute values:

6648

6649 **localNamespace**

6650 Value: icom_cal

6651

6652 **localName**

6653 Value: OccurrenceParticipantStatusEnum

6654

6655 **extendsFrom**

6656 Value: icom_cal:OccurrenceParticipantStatus

6657

6658 **stereotype**

6659 Value: primary

6660

6661 **isEnumeration**

6662 Value: TRUE

6663

6664 **description**

6665 Value: Participant's response status for an occurrence or occurrence series.

6666

6667 **instances**

6668 Value: <icom_cal:NeedsAction, icom_cal:Accepted, icom_cal:Declined, icom_cal:Tentative>

6669

6670 ICOM defines four occurrence participant's status:

6671 • **icom_cal:NeedsAction** an attendee needs to act on an occurrence or occurrence series.

6672 • **icom_cal:Accepted** an attendee accepted an occurrence or occurrence series.

6673 • **icom_cal:Declined** an attendee declined an occurrence or occurrence series.

6674 • **icom_cal:Tentative** an attendee is tentative about attending an occurrence or occurrence series.

6675

6676 4.7.11 OccurrenceParticipantTransparency

6677 4.7.11.1 Description

6678 An occurrence participant transparency is visibility of an occurrence or occurrence series in a participant's

6679 calendar or free busy.

6680 4.7.11.2 Class Definition

6681 The OccurrenceParticipantTransparency class is a mixin class which defines visibility of an occurrence or

6682 occurrence series in a participant's calendar or free busy.

6683 The OccurrenceParticipantTransparency class has attribute values:

6684

6685 **localNamespace**

6686 Value: icom_cal

6687

6688 **localName**

6689 Value: OccurrenceParticipantTransparency

6690

6691 **extendsFrom**

6692 Value:

6693

6694 **stereotype**
6695 Value: mixin
6696
6697 **description**
6698 Value: OccurrenceParticipantTransparency is a mixin class which defines visibility of an
6699 occurrence or occurrence series in a participant's calendar or free busy.
6700
6701 **propertyDefinitions**
6702 The values for this attribute are defined in Section 4.7.11.3.

6703 **4.7.11.3 Property Definitions**

6704 The OccurrenceParticipantTransparency class MAY include additional property definitions which are
6705 implementation-defined.
6706

6707 **4.7.12 OccurrenceParticipantTransparencyEnum**

6708 The OccurrenceParticipantTransparencyEnum class is an enum class that enumerates the instances
6709 each of which expresses an occurrence or occurrence series transparency in a participant's calendar or
6710 free busy.
6711 The OccurrenceParticipantTransparencyEnum class has attribute values:

6712
6713 **localNamespace**
6714 Value: icom_cal
6715
6716 **localName**
6717 Value: OccurrenceParticipantTransparencyEnum
6718
6719 **extendsFrom**
6720 Value: icom_cal:OccurrenceParticipantTransparency
6721
6722 **stereotype**
6723 Value: primary
6724
6725 **isEnumeration**
6726 Value: TRUE
6727
6728 **description**
6729 Value: Occurrence or occurrence series transparency in a participant's calendar or free busy.
6730
6731 **instances**
6732 Value: <icom_cal:Opaque, icom_cal:Transparent, icom_cal:Tentative, icom_cal:OutOfOffice,
6733 icom_cal:DefaultTransparency>
6734

6735 ICOM defines five participant transparencies:

- 6736 • **icom_cal:Opaque** an occurrence or occurrence series is opaque in a participant's calendar or
6737 free busy.
- 6738 • **icom_cal:Transparent** an occurrence or occurrence series is transparent in a participant's
6739 calendar or free busy.
- 6740 • **icom_cal:Tentative** an occurrence or occurrence series has a tentative transparency in a
6741 participant's calendar or free busy.
- 6742 • **icom_cal:OutOfOffice** an occurrence or occurrence series has out of office transparency in a
6743 participant's calendar or free busy.
- 6744 • **icom_cal:DefaultTransparency** an occurrence or occurrence series has default transparency in
6745 a participant's calendar or free busy.

6746

6747 **4.7.13 OccurrenceEditMode**

6748 **4.7.13.1 Description**

6749 An occurrence edit mode is a mode that indicates whether an occurrence or occurrence series is editable.

6750 **4.7.13.2 Class Definition**

6751 The OccurrenceEditMode class is a mixin class which defines a mode that indicates whether an
6752 occurrence or occurrence series is editable.

6753 The OccurrenceEditMode class has attribute values:

6754

6755 **localNamespace**

6756 Value: icom_cal

6757

6758 **localName**

6759 Value: OccurrenceEditMode

6760

6761 **extendsFrom**

6762 Value:

6763

6764 **stereotype**

6765 Value: mixin

6766

6767 **description**

6768 Value: OccurrenceEditMode is a mixin class which defines a mode that indicates whether an
6769 occurrence or occurrence series is editable.

6770

6771 **propertyDefinitions**

6772 The values for this attribute are defined in Section 4.7.13.3.

6773 **4.7.13.3 Property Definitions**

6774 The OccurrenceEditMode class MAY include additional property definitions which are implementation-
6775 defined.

6776

6777 4.7.14 OccurrenceEditModeEnum

6778 The OccurrenceEditModeEnum class is an enum class that enumerates the instances each of which
6779 expresses a mode that indicates whether an occurrence or occurrence series is editable.

6780 The OccurrenceEditModeEnum class has attribute values:

6781

6782 **localNamespace**

6783 Value: icom_cal

6784

6785 **localName**

6786 Value: OccurrenceEditModeEnum

6787

6788 **extendsFrom**

6789 Value: icom_cal:OccurrenceEditMode

6790

6791 **stereotype**

6792 Value: primary

6793

6794 **isEnumeration**

6795 Value: TRUE

6796

6797 **description**

6798 Value: A mode that indicates whether an occurrence or occurrence series is editable.

6799

6800 **instances**

6801 Value: <icom_cal:OrganizerCopy, icom_cal:AttendeeCopy>

6802

6803 ICOM defines two occurrence editable modes:

- 6804 • **icom_cal:OrganizerCopy** an occurrence or occurrence series is a copy created by an organizer
6805 who may update the properties such as occurrence type, occurrence status, etc.
- 6806 • **icom_cal:AttendeeCopy** an occurrence or occurrence series is a copy delivered to an attendee
6807 who may only update the attendee properties such as priority, transparency, etc .

6808

6809 4.8 Free Busy Module

6810 4.8.1 FreeBusy

6811 4.8.1.1 Description

6812 A free busy object specifies the free time and busy time intervals of one or more participants.

4.8.1.2 Class Definition

The FreeBusy class has attribute values:

localNamespace

Value: icom_cal

localName

Value: FreeBusy

extendsFrom

Value:

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.8.1.3.

4.8.1.3 Property Definitions

The FreeBusy class MUST have the property definitions:

icom_core:participant

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

Required: False

Inherited: False

Property Type: icom_core:Participant

Cardinality: Multi

Updatability: Read Only

icom_core:creationDate

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

Required: False

Inherited: False

Property Type: DateTime

Cardinality: Single

Updatability: Read Only

icom_core:startDate

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

6855	Required:	False
6856	Inherited:	False
6857	Property Type:	DateTime
6858	Cardinality:	Single
6859	Updatability:	Read Only
6860		
6861	icom_core:endDate	
6862	Description:	End date and time of a list of free busy intervals.
6863	Required:	False
6864	Inherited:	False
6865	Property Type:	DateTime
6866	Cardinality:	Single
6867	Updatability:	Read Only
6868		
6869	icom_cal:interval	
6870	Description:	A list of free busy intervals.
6871	Required:	False
6872	Inherited:	False
6873	Property Type:	icom_cal:FreeBusyInterval
6874	Cardinality:	Multi
6875	Updatability:	Read Only
6876		

6877 4.8.2 FreeBusyInterval

6878 4.8.2.1 Description

6879 A free busy interval specifies an interval of free or busy time.

6880 If a free busy type is icom_cal:Free, then a time interval is free for scheduling.

6881 If a free busy type is icom_cal:Busy, then a time interval is busy because one or more events have been
6882 scheduled for the interval.

6883 4.8.2.2 Class Definition

6884 The FreeBusyInterval class has attribute values:

6885		
6886	localNamespace	
6887	Value:	icom_cal
6888		
6889	localName	
6890	Value:	FreeBusyInterval
6891		
6892	extendsFrom	
6893	Value:	
6894		

6895 stereotype

6896 Value: primary

6897

6898	description
------	-------------

6899 Value: A free busy interval object specifies an interval of free or busy time.

6900

```
6901      propertyDefinitions
```

6902 The values for this attribute are defined in Section 4.8.2.3

6903 4.8.2.3 Property Definitions

6904 The FreeBusyInterval class **MUST** have the property definitions:

6905

6906 **icom_core:startDate**

6907	Description:	Start date and time of a free busy interval.
------	--------------	--

6908 Required: False

6909 Inherited: False

6910	Property Type:	DateTime
------	----------------	----------

6911 Cardinality: Single

6912 Updatability: Read Only

6913

6914 **icom_core:endDate**

6915	Description:	End date and time of a free busy interval.
------	--------------	--

6916 Required: False

6917 Inherited: False

6918 Property Type: DateTime

6919 Cardinality: Single

6920 Updatability: Read Only

6921

6922 **icom_cal:freeBusyType**

6923	Description:	A type of free busy interval.
------	--------------	-------------------------------

6924 Required: False

6925 Inherited: False

6926 Property Type: icom_cal:FreeBusyType

6927 Cardinality: Single

6928 Updatability: Read Only

6929

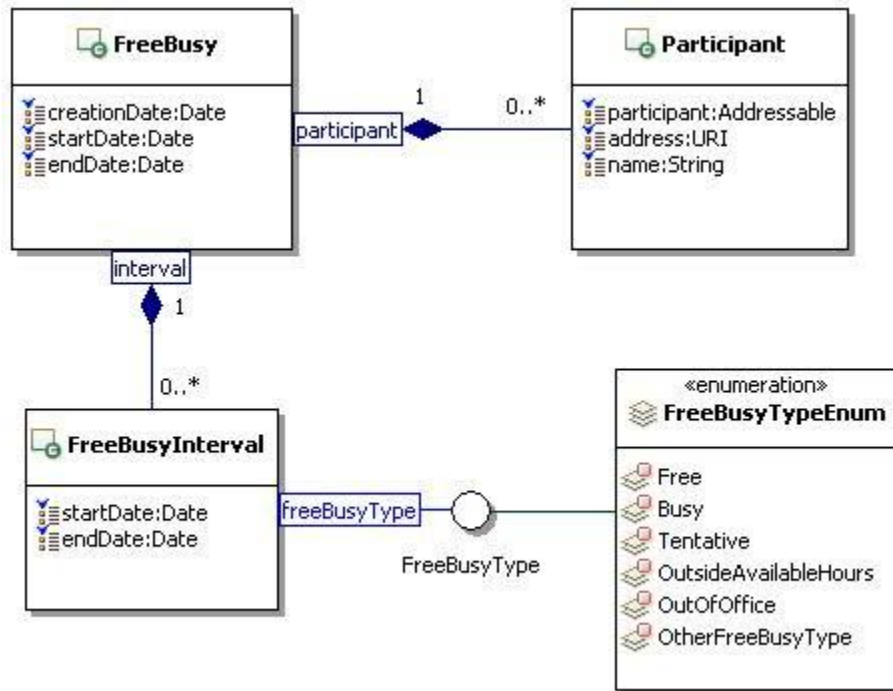


Figure 39: Free Busy Class Diagram.

4.8.3 FreeBusyType

4.8.3.1 Description

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

4.8.3.2 Class Definition

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

The FreeBusyType class has attribute values:

localNamespace

Value: icom_cal

localName

Value: FreeBusyType

extendsFrom

Value:

stereotype

Value: mixin

description

Value: FreeBusyType is a mixin class which defines different types to classify a time interval.

6954
6955 **propertyDefinitions**
6956 The values for this attribute are defined in Section 4.8.3.3.

6957 **4.8.3.3 Property Definitions**

6958 The FreeBusyType class MAY include additional property definitions which are implementation-defined.
6959

6960 **4.8.4 FreeBusyTypeEnum**

6961 The FreeBusyTypeEnum class is an enum class that enumerates the instances each of which expresses
6962 a type of free busy interval.

6963 The FreeBusyTypeEnum class has attribute values:

6964
6965 **localNamespace**
6966 Value: icom_cal
6967
6968 **localName**
6969 Value: FreeBusyTypeEnum
6970
6971 **extendsFrom**
6972 Value: icom_cal:FreeBusyType
6973
6974 **stereotype**
6975 Value: primary
6976
6977 **isEnumeration**
6978 Value: TRUE
6979
6980 **description**
6981 Value: A type of free busy interval.
6982
6983 **instances**
6984 Value: <icom_cal:Free, icom_cal:Busy, icom_cal:Tentative, icom_cal:OutsideAvailableHours,
6985 icom_cal:OutOfOffice, icom_cal:OtherFreeBusyType>
6986

6987 ICOM defines six free busy types:

- 6988 • **icom_cal:Free** a free busy interval is free.
- 6989 • **icom_cal:Busy** a free busy interval is busy.
- 6990 • **icom_cal:Tentative** a free busy interval is tentative.
- 6991 • **icom_cal:OutsideAvailableHours** a free busy interval is outside available hours.
- 6992 • **icom_cal:OutOfOffice** a free busy interval is within out of office hours.
- 6993 • **icom_cal:OtherFreeBusyType** a free busy interval is of other type.

6994

4.9 Task List Module

4.9.1 TaskList

4.9.1.1 Description

A task list contains task management artifacts.

4.9.1.2 Class Definition

The TaskList class has attribute values:

localNamespace

Value: icom_task

localName

Value: TaskList

extendsFrom

Value: icom_core:Folder

stereotype

Value: primary

description

Value: A task list contains task management artifacts.

propertyDefinitions

The values for this attribute are defined in 4.9.1.3.

4.9.1.3 Property Definitions

The TaskList class inherits property definitions from super classes.

The TaskList class MUST have the property definitions:

icom_core:timeZone

Description: Time zone of a task list.

Required: True

Inherited: False

Property Type: icom_core:TimeZone

Cardinality: Single

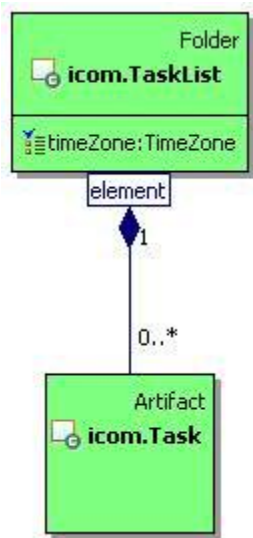
Updatability: Read Write

icom_core:element

Description: Elements of a task list.

Required: False

7034 Inherited: True
7035 Property Type: icom_task:Task
7036 Cardinality: Multi
7037 Updatability: Read Only
7038



7039
7040 Figure 40: Task List Class Diagram.
7041

7042 **4.9.2 Task**

7043 **4.9.2.1 Description**

7044 A task is an artifact that represents a task to do or a task assignment in a task list.

7045 **4.9.2.2 Class Definition**

7046 The Task class has attribute values:

7047
7048 **localNamespace**
7049 Value: icom_task
7050
7051 **localName**
7052 Value: Task
7053
7054 **extendsFrom**
7055 Value: icom_core:Artifact
7056
7057 **stereotype**
7058 Value: primary
7059
7060 **description**
7061 Value: A task is an artifact that represents a task to do or a task assignment in a task list.

7062
7063 **propertyDefinitions**
7064 The values for this attribute are defined in 4.9.2.3.

7065 **4.9.2.3 Property Definitions**

7066 The Task class inherits property definitions from super classes.
7067 The Task class MUST have the property definitions:

7068
7069 **icom_core:location**
7070 Description: Location of a task.
7071 Required: False
7072 Inherited: False
7073 Property Type: icom_core:Location
7074 Cardinality: Single
7075 Updatability: Read Write

7076
7077 **icom_core:organizer**
7078 Description: Organizer of a task.
7079 Required: True
7080 Inherited: False
7081 Property Type: icom_core:Participant
7082 Cardinality: Single
7083 Updatability: On Create

7084
7085 **icom_core:priority**
7086 Description: Priority of a task.
7087 Required: False
7088 Inherited: False
7089 Property Type: icom_core:Priority
7090 Cardinality: Single
7091 Updatability: Read Write

7092
7093 **icom_core:startDate**
7094 Description: Start date and time of a task.
7095 Required: True
7096 Inherited: False
7097 Property Type: DateTime
7098 Cardinality: Single
7099 Updatability: On Create

7100
7101 **icom_core:startDateResolution**
7102 Description: Resolution of start date and time of a task.
7103 Required: True

7104	Inherited:	False
7105	Property Type:	icom_core:DateTimeResolution
7106	Cardinality:	Single
7107	Updatability:	On Create
7108		
7109	icom_content:attachment	
7110	Description:	One or more content attachments in a task.
7111	Required:	False
7112	Inherited:	False
7113	Property Type:	icom_content:AttachedItem
7114	Cardinality:	Multi
7115	Updatability:	Read Write
7116		
7117	icom_task:dueDate	
7118	Description:	Due date and time of a task.
7119	Required:	True
7120	Inherited:	False
7121	Property Type:	DateTime
7122	Cardinality:	Single
7123	Updatability:	On Create
7124		
7125	icom_task:dueDateResolution	
7126	Description:	Resolution of due date and time of a task.
7127	Required:	True
7128	Inherited:	False
7129	Property Type:	icom_core:DateTimeResolution
7130	Cardinality:	Single
7131	Updatability:	On Create
7132		
7133	icom_task:editMode	
7134	Description:	Indicates a mode which determines whether a task is
7135		editable.
7136	Required:	False
7137	Inherited:	False
7138	Property Type:	icom_task:TaskEditMode
7139	Cardinality:	Single
7140	Updatability:	Read Only
7141		
7142	icom_task:taskStatus	
7143	Description:	Status of a task.
7144	Required:	True
7145	Inherited:	False
7146	Property Type:	icom_task:TaskStatus

7147	Cardinality:	Single
7148	Updatability:	Read Write
7149		
7150	icom_task:assignee	
7151	Description:	An assignee of a task.
7152	Required:	False
7153	Inherited:	False
7154	Property Type:	icom_core:Participant
7155	Cardinality:	Single
7156	Updatability:	Read Only
7157		
7158	icom_task:participantStatus	
7159	Description:	Participation status of a task.
7160	Required:	False
7161	Inherited:	False
7162	Property Type:	icom_task:TaskParticipantStatus
7163	Cardinality:	Single
7164	Updatability:	Read Write
7165		
7166	icom_task:completionDate	
7167	Description:	Completion date and time of a task.
7168	Required:	False
7169	Inherited:	False
7170	Property Type:	DateTime
7171	Cardinality:	Single
7172	Updatability:	Read Write
7173		
7174	icom_task:completionDateResolution	
7175	Description:	Resolution of completion date and time of a task.
7176	Required:	False
7177	Inherited:	False
7178	Property Type:	icom_core:DateTimeResolution
7179	Cardinality:	Single
7180	Updatability:	Read Write
7181		
7182	icom_task:percentComplete	
7183	Description:	Percentage of task completed.
7184	Required:	False
7185	Inherited:	False
7186	Property Type:	Integer
7187	Cardinality:	Single
7188	Updatability:	Read Write

icom_task:assigneeProperty

Description:	Extensible properties for an assignee of a task.
Required:	False
Inherited:	False
Property Type:	icom_meta:Property
Cardinality:	Multi
Updatability:	Read Write

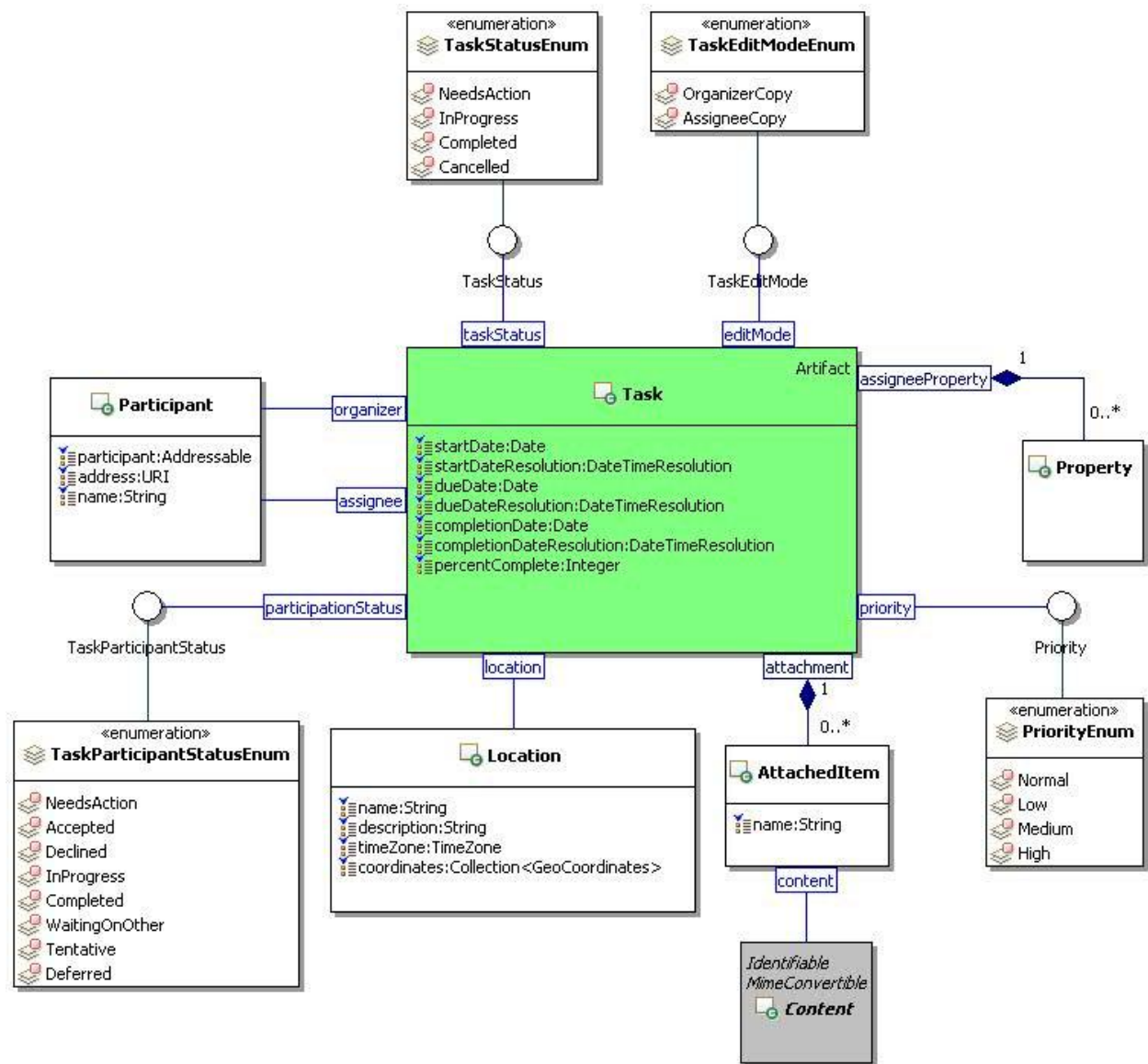


Figure 41: Task Class Diagram.

7201 4.9.3 TaskStatus

7202 4.9.3.1 Description

7203 A task status is a status of a task.

7204 4.9.3.2 Class Definition

7205 The TaskStatus class is a mixin class which defines status of a task.

7206 The TaskStatus class has attribute values:

7207

7208 **localNamespace**

7209 Value: icom_task

7210

7211 **localName**

7212 Value: TaskStatus

7213

7214 **extendsFrom**

7215 Value:

7216

7217 **stereotype**

7218 Value: mixin

7219

7220 **description**

7221 Value: TaskStatus is a mixin class which defines status of a task.

7222

7223 **propertyDefinitions**

7224 The values for this attribute are defined in Section 4.9.3.3.

7225 4.9.3.3 Property Definitions

7226 The TaskStatus class MAY include additional property definitions which are implementation-defined.

7227

7228 4.9.4 TaskStatusEnum

7229 The TaskStatusEnum class is an enum class that enumerates the instances each of which expresses a status of task.

7231 The TaskStatusEnum class has attribute values:

7232

7233 **localNamespace**

7234 Value: icom_task

7235

7236 **localName**

7237 Value: TaskStatusEnum

7238

7239 **extendsFrom**

7240 Value: icom_task:TaskStatus

7241

7242 **stereotype**

7243 Value: primary

7244

7245 **isEnumeration**

7246 Value: TRUE

7247

7248 **description**

7249 Value: Status of a task.

7250

7251 **instances**

7252 Value: <icom_task:NeedsAction, icom_task:InProgress, icom_task:Completed,

7253 icom_task:Cancelled>

7254

7255 ICOM defines four task status:

- 7256 • **icom_task:NeedsAction** a task needs action.
- 7257 • **icom_task:InProgress** a task is in progress.
- 7258 • **icom_task:Completed** a task is completed.
- 7259 • **icom_task:Cancelled** a task is cancelled.

7260

7261 4.9.5 TaskParticipantStatus

7262 4.9.5.1 Description

7263 A task participant status is a participant's response status for a task assignment.

7264 4.9.5.2 Class Definition

7265 The TaskParticipantStatus class is a mixin class which defines a participant's response status for a task

7266 assignment.

7267 The TaskParticipantStatus class has attribute values:

7268

7269 **localNamespace**

7270 Value: icom_task

7271

7272 **localName**

7273 Value: TaskParticipantStatus

7274

7275 **extendsFrom**

7276 Value:

7277

7278 **stereotype**

7279 Value: mixin

7280

7281 **description**
7282 Value: TaskParticipantStatus is a mixin class which defines a participant's response status for a
7283 task assignment.

7284
7285 **propertyDefinitions**
7286 The values for this attribute are defined in Section 4.9.5.3.

7287 **4.9.5.3 Property Definitions**

7288 The TaskParticipantStatus class MAY include additional property definitions which are implementation-
7289 defined.

7290

7291 **4.9.6 TaskParticipantStatusEnum**

7292 The TaskParticipantStatusEnum class is an enum class that enumerates the instances each of which
7293 expresses a participant's response status for a task.

7294 The TaskParticipantStatusEnum class has attribute values:

7295

7296 **localNamespace**
7297 Value: icom_task
7298

7299 **localName**
7300 Value: TaskParticipantStatusEnum
7301

7302 **extendsFrom**
7303 Value: icom_task:TaskParticipantStatus
7304

7305 **stereotype**
7306 Value: primary
7307

7308 **isEnumeration**
7309 Value: TRUE
7310

7311 **description**
7312 Value: Participant's response status for a task.
7313

7314 **instances**
7315 Value: <icom_task:NeedsAction, icom_task:Accepted, icom_task:Declined,
7316 icom_task:InProgress, icom_task:Completed, icom_task:WaitingOnOther, icom_task:Tentative,
7317 icom_task:Deferred>
7318

7319 ICOM defines eight task participant's status:

- 7320
 - **icom_task:NeedsAction** an assignee needs to act on a task.
 - **icom_task:Accepted** an assignee accepted a task.
 - **icom_task:Declined** an assignee declined a task.

7321

7322

- 7323 • **icom_task:InProgress** a task is in progress.
- 7324 • **icom_task:Completed** a task is completed.
- 7325 • **icom_task:WaitingOnOther** an assignee is waiting on other.
- 7326 • **icom_task:Tentative** an assignee is tentative about a task.
- 7327 • **icom_task:Deferred** an assignee deferred a task.

7328

7329 **4.9.7 TaskEditMode**

7330 **4.9.7.1 Description**

7331 A task edit mode is a mode that indicates whether a task is editable.

7332 **4.9.7.2 Class Definition**

7333 The TaskEditMode class is a mixin class which defines a mode that indicates whether a task is editable.

7334 The TaskEditMode class has attribute values:

7335

7336 **localNamespace**

7337 Value: icom_task

7338

7339 **localName**

7340 Value: TaskEditMode

7341

7342 **extendsFrom**

7343 Value:

7344

7345 **stereotype**

7346 Value: mixin

7347

7348 **description**

7349 Value: TaskEditMode is a mixin class which defines a mode that indicates whether task is
7350 editable.

7351

7352 **propertyDefinitions**

7353 The values for this attribute are defined in Section 4.9.7.3.

7354 **4.9.7.3 Property Definitions**

7355 The TaskEditMode class MAY include additional property definitions which are implementation-defined.

7356

7357 **4.9.8 TaskEditModeEnum**

7358 The TaskEditModeEnum class is an enum class that enumerates the instances each of which expresses
7359 an editable mode of a task.

7360 The TaskEditModeEnum class has attribute values:

7361

7362 **localNamespace**

7363 Value: icom_task

7364

7365 **localName**

7366 Value: TaskEditModeEnum

7367

7368 **extendsFrom**

7369 Value: icom_task:TaskEditMode

7370

7371 **stereotype**

7372 Value: primary

7373

7374 **isEnumeration**

7375 Value: TRUE

7376

7377 **description**

7378 Value: A mode that indicates whether a task is editable.

7379

7380 **instances**

7381 Value: <icom_task:OrganizerCopy, icom_task:AssigneeCopy>

7382

7383 ICOM defines two task editable modes:

- 7384 • **icom_task:OrganizerCopy**: a task is a copy created by an organizer who may update the
- 7385 properties such as start time, due time.
- 7386 • **icom_task:AssigneeCopy**: a task is a copy delivered to an assignee who may only update the
- 7387 assignee properties such as completion time, participant status, percent completed.

7388

7389 4.10 Forum Module

7390 4.10.1 Discussion

7391 4.10.1.1 Description

7392 A discussion is an item in a discussion container.

7393 4.10.1.2 Class Definition

7394 The Discussion class is a mixin class that defines the characteristics of artifacts that can be elements of

7395 discussion containers.

7396 The Discussion class has attribute values:

7397

7398 **localNamespace**

7399 Value: icom_forum

7400

7401 **localName**

7402 Value: Discussion

7403

7404 **extendsFrom**

7405 Value: icom_core:Item

7406

7407 **stereotype**

7408 Value: mixin

7409

7410 **description**

7411 Value: Discussion is a mixin class that defines the characteristics of artifacts that can be placed

7412 in a discussion container.

7413

7414 **propertyDefinitions**

7415 The values for this attribute are defined in Section 4.10.1.3.

7416 **4.10.1.3 Property Definitions**

7417 The Discussion class inherits property definitions from super classes.

7418 The Discussion class MUST have the property definition:

7419

7420 **icom_forum:inReplyTo**

7421 Description:	Another discussion object that a discussion object is replying to.
7422	
7423 Required:	False
7424 Inherited:	False
7425 Property Type:	icom_forum:Discussion
7426 Cardinality:	Single
7427 Updatability:	Read Write

7428

7429 The Discussion class MAY include additional property definitions which are implementation-defined.

7430

7431 **4.10.2 DiscussionContainer**

7432 **4.10.2.1 Description**

7433 A discussion container contains discussion items.

7434 **4.10.2.2 Class Definition**

7435 The DiscussionContainer class is a mixin class that defines the characteristics of folders that contain

7436 Discussion items.

7437 The DiscussionContainer class has attribute values:

7438

7439 **localNamespace**

7440 Value: icom_forum

7441

7442 **localName**

7443 Value: DiscussionContainer

7444

7445 **extendsFrom**

7446 Value: icom_core:Container

7447

7448 **stereotype**

7449 Value: mixin

7450

7451 **description**

7452 Value: DiscussionContainer is a mixin class that defines the characteristics of folders that

7453 contain Discussion items.

7454

7455 **propertyDefinitions**

7456 The values for this attribute are defined in Section 4.10.2.3.

7457 4.10.2.3 Property Definitions

7458 The DiscussionContainer class inherits property definitions from super classes.

7459 The DiscussionContainer class MUST have the property definition:

7460

7461 **icom_core:element**

7462 Description:	Elements of a discussion container.
7463 Required:	False
7464 Inherited:	True
7465 Property Type:	icom_forum:Discussion
7466 Cardinality:	Multi
7467 Updatability:	Read Only

7468

7469 The DiscussionContainer class MAY include additional property definitions which are implementation-

7470 defined.

7471

7472 4.10.3 DiscussionMessage

7473 4.10.3.1 Description

7474 A discussion message is a message in a forum discussion thread.

4.10.3.2 Class Definition

The DiscussionMessage class has attribute values:

localNamespace

Value: icom_forum

localName

Value: DiscussionMessage

extendsFrom

Value: icom_msg:Message, icom_forum:Discussion

stereotype

Value: primary

description

Value: Discussion message is a message in a forum discussion thread.

propertyDefinitions

The values for this attribute are defined in Section 4.10.3.3.

4.10.3.3 Property Definitions

The DiscussionMessage class inherits property definitions from super classes.

The DiscussionMessage class MUST have the property definition:

icom_forum:inReplyTo

Description: Another discussion message that a discussion message is replying to.

Required: False

Inherited: True

Property Type: icom_forum:DiscussionMessage

Cardinality: Single

Updatability: Read Write

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

4.10.4 TopicContainer

4.10.4.1 Description

A topic container contains topics.

4.10.4.2 Class Definition

The TopicContainer class is a mixin class which defines the characteristics of folders that contain Topics.

The TopicContainer class has attribute values:

localNamespace

Value: icom_forum

localName

Value: TopicContainer

extendsFrom

Value: icom_core:Container

stereotype

Value: mixin

description

Value: TopicContainer is a mixin class that defines the characteristics of folders that contain topics.

propertyDefinitions

The values for this attribute are defined in Section 4.10.4.3.

4.10.4.3 Property Definitions

The TopicContainer class inherits property definitions from super classes.

The TopicContainer class MUST have the property definitions:

icom_core:element

Description: Elements of a topic container.

Required: False

Inherited: True

Property Type: icom_forum:Topic

Cardinality: Multi

Updatability: Read Only

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

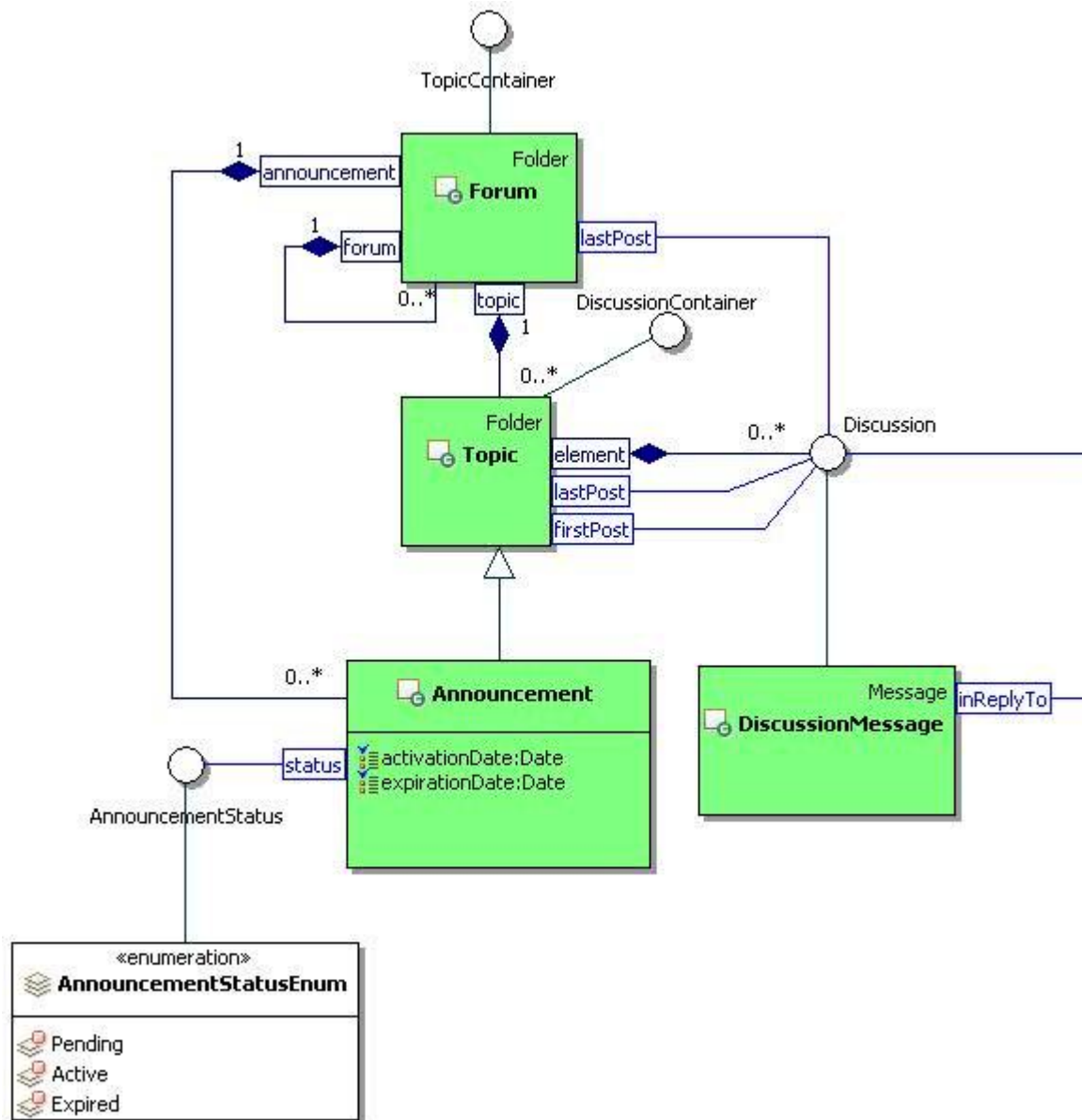


Figure 42: Forum Class Diagram.

4.10.5 Forum

4.10.5.1 Description

A forum contains sub-forums, topics, and announcements.

4.10.5.2 Class Definition

The Forum class has attribute values:

localNamespace

Value: icom_forum

7562 **localName**
7563 Value: Forum
7564
7565 **extendsFrom**
7566 Value: icom_core:Folder, icom_forum:TopicContainer
7567
7568 **stereotype**
7569 Value: primary
7570
7571 **description**
7572 Value: A forum contains sub-forums, topics, and announcements.
7573
7574 **propertyDefinitions**
7575 The values for this attribute are defined in Section 4.10.5.3.

7576 **4.10.5.3 Property Definitions**

7577 The Forum class inherits property definitions from super classes.
7578 The Forum class MUST have the property definitions:

7579
7580 **icom_forum:lastPost**
7581 Description: The last posted discussion in a forum.
7582 Required: False
7583 Inherited: False
7584 Property Type: icom_forum:Discussion
7585 Cardinality: Single
7586 Updatability: Read Only
7587
7588 **icom_forum:forum**
7589 Description: Sub-forums of a forum.
7590 Required: False
7591 Inherited: False
7592 Property Type: icom_forum:Forum
7593 Cardinality: Multi
7594 Updatability: Read Only
7595
7596 **icom_forum:topic**
7597 Description: Topics of a forum.
7598 Required: False
7599 Inherited: False
7600 Property Type: icom_forum:Topic
7601 Cardinality: Multi
7602 Updatability: Read Only
7603

7604	icom_forum:announcement	
7605	Description:	Announcements of a forum.
7606	Required:	False
7607	Inherited:	False
7608	Property Type:	icom_forum:Announcement
7609	Cardinality:	Multi
7610	Updatability:	Read Only

7611

7612 The Forum class MAY include additional property definitions which are implementation-defined.

7613

7614 4.10.6 Topic

7615 4.10.6.1 Description

7616 A topic contains conversations among forum participants. The discussions in a topic may be sorted in

7617 chronological order or threaded by reply.

7618 4.10.6.2 Class Definition

7619 The Topic class has attribute values:

7620		
7621	localNamespace	
7622	Value:	icom_forum
7623		
7624	localName	
7625	Value:	Topic
7626		
7627	extendsFrom	
7628	Value:	icom_core:Folder, icom_forum:DiscussionContainer
7629		
7630	stereotype	
7631	Value:	primary
7632		
7633	description	
7634	Value:	A topic contains discussion threads.
7635		
7636	propertyDefinitions	
7637	The values for this attribute are defined in Section 4.10.6.3.	

7638 4.10.6.3 Property Definitions

7639 The Topic class inherits property definitions from super classes.

7640 The Topic class MUST have the property definitions:

7641		
7642	icom_core:element	
7643	Description:	Elements of a topic.

7644	Required:	False
7645	Inherited:	True
7646	Property Type:	icom_forum:Discussion
7647	Cardinality:	Multi
7648	Updatability:	Read Only
7649		
7650	icom_forum:firstPost	
7651	Description:	The first posted discussion in a topic.
7652	Required:	False
7653	Inherited:	False
7654	Property Type:	icom_forum:Discussion
7655	Cardinality:	Single
7656	Updatability:	Read Only

7657		
7658	icom_forum:lastPost	
7659	Description:	The last posted discussion in a topic.
7660	Required:	False
7661	Inherited:	False
7662	Property Type:	icom_forum:Discussion
7663	Cardinality:	Single
7664	Updatability:	Read Only

7665

7666 The Topic class MAY include additional property definitions which are implementation-defined.

7667

7668 4.10.7 Announcement

7669 4.10.7.1 Description

7670 An announcement contains time-sensitive discussion posts that are valid for a specified period of time,
7671 depending on activation and expiration times.

7672 4.10.7.2 Class Definition

7673 The Announcement class has attribute values:

7674		
7675	localNamespace	
7676	Value:	icom_forum
7677		
7678	localName	
7679	Value:	Announcement
7680		
7681	extendsFrom	
7682	Value:	icom_forum:Topic
7683		

7684 **stereotype**
7685 Value: primary
7686
7687 **description**
7688 Value: An announcement contains discussion items that are valid for a specified period of time.
7689
7690 **propertyDefinitions**
7691 The values for this attribute are defined in Section 4.10.7.3.

7692 **4.10.7.3 Property Definitions**

7693 The Announcement class inherits property definitions from super classes.
7694 The Announcement class **MUST** have the property definitions:

7695
7696 **icom_forum:activationDate**
7697 Description: Date and time when an announcement becomes active.
7698 Required: False
7699 Inherited: False
7700 Property Type: DateTime
7701 Cardinality: Single
7702 Updatability: Read Write
7703

7704 **icom_forum:expirationDate**
7705 Description: Date and time when an announcement expires.
7706 Required: False
7707 Inherited: False
7708 Property Type: DateTime
7709 Cardinality: Single
7710 Updatability: Read Write
7711

7712 **icom_forum:announcementStatus**
7713 Description: Status of an announcement.
7714 Required: True
7715 Inherited: False
7716 Property Type: icom_forum:AnnouncementStatus
7717 Cardinality: Single
7718 Updatability: Read Write
7719

7720 The Announcement class **MAY** include additional property definitions which are implementation-defined.
7721

7722 4.10.8 AnnouncementStatus

7723 4.10.8.1 Description

7724 An announcement status is status of an announcement.

7725 4.10.8.2 Class Definition

7726 The AnnouncementStatus class is a mixin class which defines status of an announcement.

7727 The AnnouncementStatus class has attribute values:

7728

7729 **localNamespace**

7730 Value: icom_forum

7731

7732 **localName**

7733 Value: AnnouncementStatus

7734

7735 **extendsFrom**

7736 Value:

7737

7738 **stereotype**

7739 Value: mixin

7740

7741 **description**

7742 Value: AnnouncementStatus is a mixin class which defines status of an announcement.

7743

7744 **propertyDefinitions**

7745 The values for this attribute are defined in Section 4.10.8.3.

7746 4.10.8.3 Property Definitions

7747 The AnnouncementStatus class MAY include additional property definitions which are implementation-
7748 defined.

7749

7750 4.10.9 AnnouncementStatusEnum

7751 The AnnouncementStatusEnum class is an enum class that enumerates the instances each of which
7752 expresses a status of announcement.

7753 The AnnouncementStatusEnum class has attribute values:

7754

7755 **localNamespace**

7756 Value: icom_forum

7757

7758 **localName**

7759 Value: AnnouncementStatusEnum

7760

7761 **extendsFrom**
7762 Value: icom_forum:AnnouncementStatus
7763
7764 **stereotype**
7765 Value: primary
7766
7767 **isEnumeration**
7768 Value: TRUE
7769
7770 **description**
7771 Value: Status of announcement.
7772
7773 **instances**
7774 Value: <icom_forum:Pending, icom_forum:Active, icom_forum:Expired>
7775
7776 ICOM defines three announcement status:
7777 • **icom_forum:Pending** an announcement is pending.
7778 • **icom_forum:Active** an announcement is active.
7779 • **icom_forum:Expired** an announcement is expired.
7780

7781 4.11 Conference Module

7782 4.11.1 Conference

7783 4.11.1.1 Description

7784 A conference is a container that represents a durable context for conference sessions.
7785 It contains conference metadata, settings, and transcripts.

7786 4.11.1.2 Class Definition

7787 The Conference class has attribute values:

7788
7789 **localNamespace**
7790 Value: icom_conf
7791
7792 **localName**
7793 Value: Conference
7794
7795 **extendsFrom**
7796 Value: icom_core:Folder
7797
7798 **stereotype**
7799 Value: primary

7800
7801 **description**
7802 Value: A conference represents a durable context for online conference sessions.
7803
7804 **propertyDefinitions**
7805 The values for this attribute are defined in Section 4.11.1.3.

7806 **4.11.1.3 Property Definitions**

7807 The Conference class inherits property definitions from super classes.
7808 The Conference class MUST have the property definitions:

7809
7810 **icom_core:organizer**
7811 Description: Organizer of a conference.
7812 Required: False
7813 Inherited: False
7814 Property Type: icom_core:Participant
7815 Cardinality: Single
7816 Updatability: On Create

7817
7818 **icom_conf:conferenceType**
7819 Description: Type of a conference.
7820 Required: False
7821 Inherited: False
7822 Property Type: icom_conf:ConferenceType
7823 Cardinality: Single
7824 Updatability: Read Write

7825
7826 **icom_conf:conferenceStatus**
7827 Description: Status of a conference.
7828 Required: False
7829 Inherited: False
7830 Property Type: icom_conf:ConferenceStatus
7831 Cardinality: Single
7832 Updatability: Read Only

7833
7834 **icom_conf:runningSession**
7835 Description: Current session of a conference.
7836 Required: False
7837 Inherited: False
7838 Property Type: icom_conf:ConferenceSession
7839 Cardinality: Single
7840 Updatability: Read Only

7841

7842	icom_conf:conferenceSetting	
7843	Description:	Configurable settings of a conference.
7844	Required:	False
7845	Inherited:	False
7846	Property Type:	icom_conf:ConferenceSetting
7847	Cardinality:	Single
7848	Updatability:	Read Only
7849		
7850	icom_conf:transcript	
7851	Description:	Transcripts from ended sessions of a conference.
7852	Required:	False
7853	Inherited:	False
7854	Property Type:	icom_doc:Document
7855	Cardinality:	Multi
7856	Updatability:	Read Write
7857		
7858	icom_conf:scheduledStartDate	
7859	Description:	Scheduled start date and time of a conference session.
7860	Required:	False
7861	Inherited:	False
7862	Property Type:	DateTime
7863	Cardinality:	Single
7864	Updatability:	Read Write
7865		
7866	icom_conf:scheduledEndDate	
7867	Description:	Scheduled end date and time of a conference session.
7868	Required:	False
7869	Inherited:	False
7870	Property Type:	DateTime
7871	Cardinality:	Single
7872	Updatability:	Read Write
7873		
7874	The Conference class MAY include additional property definitions which are implementation-defined.	
7875		

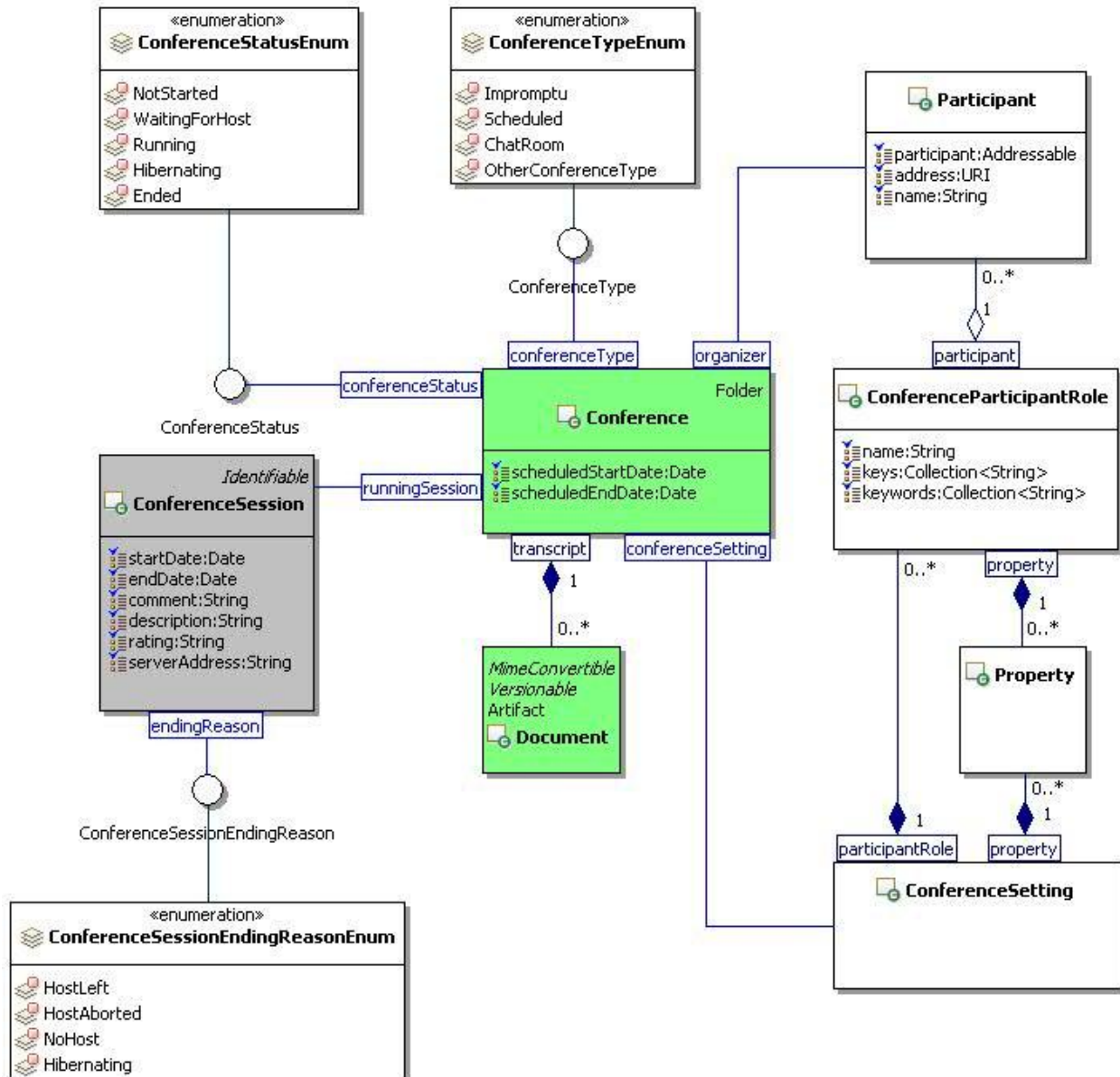


Figure 43: Conference Class Diagram.

4.11.2 ConferenceType

4.11.2.1 Description

A conference type represents a category of conferences.

4.11.2.2 Class Definition

The ConferenceType class is a mixin class which defines a type of conference.

The ConferenceType class has attribute values:

localNamespace

Value: icom_conf

7888

7889 **localName**

7890 Value: ConferenceType

7891

7892 **extendsFrom**

7893 Value:

7894

7895 **stereotype**

7896 Value: mixin

7897

7898 **description**

7899 Value: ConferenceType is a mixin class which defines type of conference.

7900

7901 **propertyDefinitions**

7902 The values for this attribute are defined in Section 4.11.2.3.

7903 **4.11.2.3 Property Definitions**

7904 The ConferenceType class MAY include additional property definitions which are implementation-defined.

7905

7906 **4.11.3 ConferenceTypeEnum**

7907 The ConferenceTypeEnum class is an enum class that enumerates the instances each of which

7908 expresses a type of a conference.

7909 The ConferenceTypeEnum class has attribute values:

7910

7911 **localNamespace**

7912 Value: icom_conf

7913

7914 **localName**

7915 Value: ConferenceTypeEnum

7916

7917 **extendsFrom**

7918 Value: icom_conf:ConferenceType

7919

7920 **stereotype**

7921 Value: primary

7922

7923 **isEnumeration**

7924 Value: TRUE

7925

7926 **description**

7927 Value: A type of a conference.

7928

7929 **instances**
7930 Value: <icom_conf:Impromptu, icom_conf:Scheduled, icom_conf:ChatRoom,
7931 icom_conf:OtherConferenceType>

7932

7933 ICOM defines four conference types:

- 7934 • **icom_conf:Impromptu** a conference session is started impromptu.
- 7935 • **icom_conf:Scheduled** a conference session is scheduled.
- 7936 • **icom_conf:ChatRoom** a conference is used for a chat room.
- 7937 • **icom_conf:OtherConferenceType** a conference is of other type.

7938

7939 **4.11.4 ConferenceStatus**

7940 **4.11.4.1 Description**

7941 A conference status is status of an online conference.

7942 **4.11.4.2 Class Definition**

7943 The ConferenceStatus class is a mixin class which defines status of an online conference.

7944 The ConferenceStatus class has attribute values:

7945

7946 **localNamespace**

7947 Value: icom_conf

7948

7949 **localName**

7950 Value: ConferenceStatus

7951

7952 **extendsFrom**

7953 Value:

7954

7955 **stereotype**

7956 Value: mixin

7957

7958 **description**

7959 Value: ConferenceStatus is a mixin class which defines status of an online conference.

7960

7961 **propertyDefinitions**

7962 The values for this attribute are defined in Section 4.11.4.3.

7963 **4.11.4.3 Property Definitions**

7964 The ConferenceStatus class MAY include additional property definitions which are implementation-
7965 defined.

7966

4.11.5 ConferenceStatusEnum

The ConferenceStatusEnum class is an enum class that enumerates the instances each of which expresses a status of a conference.

The ConferenceStateEnum class has attribute values:

localNamespace

Value: icom_conf

localName

Value: ConferenceStatusEnum

extendsFrom

Value: icom_conf:ConferenceStatus

stereotype

Value: primary

isEnumeration

Value: TRUE

description

Value: Status of a conference.

instances

Value: <icom_conf:NotStarted, icom_conf:WaitingForHost, icom_conf:Running, icom_conf:Hibernate, icom_conf:Ended>

ICOM defines five conference status:

- **icom_conf:NotStarted** a conference session is not started .
- **icom_conf:WaitingForHost** a conference session is waiting for a host.
- **icom_conf:Running** a conference session is running.
- **icom_conf:Hibernate** a conference session is hibernating.
- **icom_conf:Ended** a conference session is ended.

4.11.6 ConferenceSession

4.11.6.1 Description

A conference session represents the metadata for a session of a conference.

4.11.6.2 Class Definition

The ConferenceSession class has attribute values:

localNamespace

Value: icom_conf

localName

Value: ConferenceSession

extendsFrom

Value: icom_core:Identifiable

stereotype

Value: primary

description

Value: A conference session represents the metadata for a session of a conference.

propertyDefinitions

The values for this attribute are defined in Section 4.11.6.3.

4.11.6.3 Property Definitions

The ConferenceSession class inherits property definitions from super classes.

The ConferenceSession class MUST have the property definitions:

icom_core:description

Description: Description of a conference session.

Required: False

Inherited: False

Property Type: String

Cardinality: Single

Updatability: Read Write

icom_core:startDate

Description: Start date and time of a conference session.

Required: False

Inherited: False

Property Type: DateTime

Cardinality: Single

Updatability: Read Only

8044	icom_core:endDate	
8045	Description:	End date and time of a conference session.
8046	Required:	False
8047	Inherited:	False
8048	Property Type:	DateTime
8049	Cardinality:	Single
8050	Updatability:	Read Only
8051		
8052	icom_conf:comment	
8053	Description:	Comment on a conference session.
8054	Required:	False
8055	Inherited:	False
8056	Property Type:	String
8057	Cardinality:	Single
8058	Updatability:	Read Write
8059		
8060	icom_conf:description	
8061	Description:	Description of a conference session.
8062	Required:	False
8063	Inherited:	False
8064	Property Type:	String
8065	Cardinality:	Single
8066	Updatability:	Read Write
8067		
8068	icom_conf:rating	
8069	Description:	Rating of a conference session.
8070	Required:	False
8071	Inherited:	False
8072	Property Type:	String
8073	Cardinality:	Single
8074	Updatability:	Read Write
8075		
8076	icom_conf:serverAddress	
8077	Description:	Address of a server that hosts a conference session.
8078	Required:	False
8079	Inherited:	False
8080	Property Type:	String
8081	Cardinality:	Single
8082	Updatability:	Read Only
8083		
8084	icom_conf:endingReason	
8085	Description:	Reason for ending a conference session.

8086	Required:	False
8087	Inherited:	False
8088	Property Type:	icom_conf:ConferenceSessionEndingReason
8089	Cardinality:	Single
8090	Updatability:	Read Only

8091

8092 The ConferenceSession class MAY include additional property definitions which are implementation-
8093 defined.

8094

8095 **4.11.7 ConferenceSessionEndingReason**

8096 **4.11.7.1 Description**

8097 A conference session ending reason is an indication of how a conference session ended.

8098 **4.11.7.2 Class Definition**

8099 The ConferenceSessionEndingReason class is a mixin class which defines an indication of how a
8100 conference session ended.

8101 The ConferenceSessionEndingReason class has attribute values:

8102

8103 **localNamespace**

8104 Value: icom_conf

8105

8106 **localName**

8107 Value: ConferenceSessionEndingReason

8108

8109 **extendsFrom**

8110 Value:

8111

8112 **stereotype**

8113 Value: mixin

8114

8115 **description**

8116 Value: ConferenceSessionEndingReason is a mixin class which defines an indication of how a
8117 conference session ended.

8118

8119 **propertyDefinitions**

8120 The values for this attribute are defined in Section 4.11.7.3.

8121 **4.11.7.3 Property Definitions**

8122 The ConferenceSessionEndingReason class MAY include additional property definitions which are
8123 implementation-defined.

8124

4.11.8 ConferenceSessionEndingReasonEnum

The ConferenceSessionEndingReasonEnum class is an enum class that enumerates the instances each of which expresses a reason for ending a conference session.

The ConferenceSessionEndingReasonEnum class has attribute values:

localNamespace

Value: icom_conf

localName

Value: ConferenceSessionEndingReasonEnum

extendsFrom

Value: icom_conf:ConferenceSessionEndingReason

stereotype

Value: primary

isEnumeration

Value: TRUE

description

Value: Reason for ending a conference session.

instances

Value: <icom_conf:HostLeft, icom_conf:HostAborted, icom_conf:NoHost, icom_conf:Hibernating>

ICOM defines four conference session states:

- **icom_conf:HostLeft** a conference session ended after the host left.
- **icom_conf:HostAborted** a conference session ended after the host aborted it.
- **icom_conf:NoHost** a conference session ended due to no one hosting.
- **icom_conf:Hibernating** a conference session is hibernating.

4.11.9 ConferenceSetting

4.11.9.1 Description

A conference setting represents configuration settings for sessions of a conference.

4.11.9.2 Class Definition

The ConferenceSetting class has attribute values:

localNamespace

Value: icom_conf

8165

8166 **localName**

8167 Value: ConferenceSetting

8168

8169 **extendsFrom**

8170 Value:

8171

8172 **stereotype**

8173 Value: primary

8174

8175 **description**

8176 Value: A conference setting represents configuration settings for sessions of a conference.

8177

8178 **propertyDefinitions**

8179 The values for this attribute are defined in Section 4.11.9.3.

8180 **4.11.9.3 Property Definitions**

8181 The ConferenceSetting class inherits property definitions from super classes.

8182 The ConferenceSetting class MUST have the property definitions:

8183

8184 **icom_meta:property**

8185 Description:	Configurable properties for a conference.
8186 Required:	False
8187 Inherited:	False
8188 Property Type:	icom_meta:property
8189 Cardinality:	Multi
8190 Updatability:	Read Write

8191

8192 **icom_conf:participantRole**

8193 Description:	Role settings for conference participants.
8194 Required:	False
8195 Inherited:	False
8196 Property Type:	icom_conf:ConferenceParticipantRole
8197 Cardinality:	Multi
8198 Updatability:	Read Write

8199

8200 The ConferenceSetting class MAY include additional property definitions which are implementation-
8201 defined.

8202

8203 4.11.10 ConferenceParticipantRole

8204 4.11.10.1 Description

8205 A conference participant role defines roles settings for a conference participant.

8206 4.11.10.2 Class Definition

8207 The ConferenceParticipantRole class has attribute values:

8208

8209 **localNamespace**

8210 Value: icom_conf

8211

8212 **localName**

8213 Value: ConferenceParticipantRole

8214

8215 **extendsFrom**

8216 Value:

8217

8218 **stereotype**

8219 Value: primary

8220

8221 **description**

8222 Value: A conference participant role contains roles settings for a conference.

8223

8224 **propertyDefinitions**

8225 The values for this attribute are defined in Section 4.11.10.3.

8226 4.11.10.3 Property Definitions

8227 The ConferenceParticipantRole class MUST have the property definitions:

8228

8229 **icom_core:name**

8230 Description: Name of a role setting in a conference.

8231 Required: False

8232 Inherited: False

8233 Property Type: String

8234 Cardinality: Single

8235 Updatability: Read Write

8236

8237 **icom_core:participant**

8238 Description: One or more participants in a role setting.

8239 Required: False

8240 Inherited: False

8241 Property Type: icom_core:Participant

8242 Cardinality: Multi

8243	Updatability:	Read Write
8244		
8245	icom_meta:property	
8246	Description:	Configurable properties for a role setting.
8247	Required:	False
8248	Inherited:	False
8249	Property Type:	icom_meta:Property
8250	Cardinality:	Multi
8251	Updatability:	Read Write
8252		
8253	icom_conf:key	
8254	Description:	One or more sign on keys to activate a role setting.
8255	Required:	False
8256	Inherited:	False
8257	Property Type:	String
8258	Cardinality:	Multi
8259	Updatability:	Read Write
8260		
8261	icom_conf:keyword	
8262	Description:	One or more key words to activate a role setting.
8263	Required:	False
8264	Inherited:	False
8265	Property Type:	String
8266	Cardinality:	Multi
8267	Updatability:	Read Write
8268		
8269	The ConferenceParticipantRole class MAY include additional property definitions which are	
8270	implementation-defined.	

5 ~~5~~ Conformance

5.1 Software Architecture or Framework Dependence

The ICOM specification does not presume a particular software architecture or framework for use of the ICOM model.

~~Conformance to the ICOM specification is defined using use case roles played by the following four stakeholders of a typical software architecture or framework:~~

- ~~1. An ICOM platform provider supplies an environment for one or more ICOM service providers, producers, and consumers to exchange ICOM objects.~~
- ~~2. An ICOM service provider manages objects produced by one or more ICOM producers for access by one or more ICOM consumers.~~
- ~~3. An ICOM producer creates objects managed by an ICOM service provider.~~
- ~~4. An ICOM consumer accepts objects managed by an ICOM service provider.~~

Fulfillment of ICOM use case roles and accompanying responsibilities is implementation dependent. ~~An ICOM implementation may fulfill one or more of ICOM use case roles and accompanying responsibilities.~~

5.2 Platform Provider Conformance

5.2.1 Platform Provider Conformance by platform provider:— No Extension Modules

- ~~4.~~ An ICOM platform provider: with no extension modules (Section 4):
- a. SHALL conform to all mandatory statements and
 - b. MAY conform to optional statements
- of the core ICOM model as defined in Section 3 of this standard.

5.2.2 Platform Provider Conformance – One or More Extension Modules

- ~~2.~~ An ICOM platform provider: with extension modules (Section 4):
- a. SHALL conform to Section 5.2.1 and
 - ~~a-b.~~ SHALL conform to all mandatory statements and
 - ~~b-c.~~ MAY conform to optional statements
- as defined in Section 4 for each extension module.

5.3 Service Provider Conformance by

5.3.1 ICOM Service Provider – No Extension Modules

~~An ICOM service provider: may provide one or more services defined in Section 3. For each such service provided, an ICOM service provider:~~

- a. SHALL conform to all mandatory statements and
b. MAY conform to optional statements
for the classes, super classes, and related classes defined in Section 3 of this standard.

5.3.2 ICOM Service Provider – One or More Extension Modules

~~1.~~ An ICOM service provider MAY support one or more extension modules as defined in Section 4 of this standard. For each service provided, an ICOM service provider:

~~2. An ICOM service provider that supports an extension module:~~

~~a. SHALL conform to all mandatory statements and~~

~~b. a. MAY conform to optional statements~~

a. SHALL conform to Section 5.3.1 (if an offered service is defined in Section 3) and
~~as defined in Section 4 for that extension module.~~

~~3. Depending on the classes extended by an extension module, an ICOM service provider:~~

~~a. b. SHALL conform to all mandatory statements and~~

~~b. c. MAY conform to optional statements~~

~~for inherited super classes and related classes defined in Section 3 of this standard.~~

~~Note: ICOM environment may include multiple service providers each of which provides different subsets of extension modules.~~

as defined in Section 4 for that extension module.

5.4 ICOM Producer Conformance by ICOM producer:

5.4.1 ICOM Producer Conformance – No Extension Modules

~~1.~~ An ICOM producer that produces no objects of a class conforming to Section 4:

a. SHALL conform to all mandatory statements and

b. MAY conform to optional statements

for the class and super classes thereof in Section 3 of this standard, for any object produced.

5.4.2 ICOM Producer Conformance – One or More Extension Modules

~~2.~~ An ICOM producer ~~may support one or more extension modules as defined in that produces~~ objects of a class conforming to Section 4 ~~of this standard. ICOM producers that support an extension module:~~

a. SHALL conform to Section 5.4.1 and

~~a. b. SHALL conform to all mandatory statements and~~

~~b. c. MAY conform to optional statements~~

as defined in Section 4 for that extension module.

5.5 ICOM Consumer Conformance by ICOM consumer:

5.5.1 ICOM Consumer Conformance – No Extension Modules

- ~~1.~~ An ICOM consumer that consumes no objects of a class conforming to Section 4:
- a. SHALL conform to all mandatory statements and
 - b. MAY conform to optional statements
- for the class and super classes thereof in Section 3 of this standard, for any object consumed.

5.5.2 ICOM Consumer Conformance – Extension Modules

- ~~2.~~ An ICOM consumer ~~may support one or more extension modules as defined in~~ that consumes objects of a class conforming to Section 4 ~~of this standard. ICOM consumers that support an extension module:~~
- a. SHALL conform to Section 5.5.1 and
 - ~~a-b.~~ SHALL conform to all mandatory statements and
 - ~~b-c.~~ MAY conform to optional statements
- as defined in Section 4 for that extension module.

Appendix A. Acknowledgements

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:

- Rafiul Ahad, Oracle Corporation
- Kenneth P. Baclawski, Northeastern University
- Eric S. Chan, Oracle Corporation
- Martin Chapman, Oracle Corporation
- Scott Conroy, Individual
- Stefan Decker, Digital Enterprise Research Institute (DERI)
- Laura Dragan, Digital Enterprise Research Institute (DERI)
- Patrick Durusau, Individual
- Siegfried Handschuh, Digital Enterprise Research Institute (DERI)
- Deirdre Lee, Digital Enterprise Research Institute (DERI)
- Marc Pallot, ESoCE-NET
- Chancellor Pascale, Johns Hopkins University Applied Physics Laboratory
- Vassilios Peristeras, Digital Enterprise Research Institute (DERI)
- Peter Saint-Andre, Cisco Systems, Inc.
- Ramesh Vasudevan, Oracle Corporation
- Peter Yim, Individual

Appendix B. Revision History

Revision	Date	Editor	Changes Made
CSPRD 01	March 16, 2011	Eric S. Chan Patrick Durusau	Committee Specification Draft for Public Review
CSPRD 02	November 8, 2011	Eric S. Chan Patrick Durusau	Changes in response to public review comments.
CSPRD 03	March 20, 2012	Eric S. Chan Patrick Durusau Laura Dragan	Changes in response to TC members review comments.
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.