



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

Committee Specification Draft 01

16 March 2011

Specification URIs:

This Version:

<http://docs.oasis-open.org/icom/icom-ics/v1.0/csd01/icom-ics-v1.0-csd01.doc> (Authoritative)
<http://docs.oasis-open.org/icom/icom-ics/v1.0/csd01/icom-ics-v1.0-csd01.html>
<http://docs.oasis-open.org/icom/icom-ics/v1.0/csd01/icom-ics-v1.0-csd01.pdf>

Previous Version:

N/A

Latest Version:

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

Technical Committee:

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

Chair(s):

Eric S. Chan, Oracle

Editor(s):

Eric S. Chan, Oracle
Patrick Durusau, Individual

Related work:

N/A

Declared XML Namespace(s):

<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/addressbook/201008>
<http://docs.oasis-open.org/ns/icom/calendar/201008>
<http://docs.oasis-open.org/ns/icom/forum/201008>
<http://docs.oasis-open.org/ns/icom/conference/201008>

Abstract:

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

The framework is not intended to prescribe how applications or services conforming to its model implement, store, or transport the objects. It is intended as a basis for integrating a broad range of collaboration objects to enable seamless transitions across collaboration activities. This enables applications to aggregate discussion threads across multiple 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 model integrates a broad range of collaboration activities, by encompassing and improving on a range of models which are part of existing standards and technologies. The model is modular to allow extensibility. The core concepts, metadata concepts, and their relations are included in the Core, while the specific concepts and relations for each area of collaboration activities are defined in separate extension modules.

Status:

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

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

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

Citation Format:

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

[ICOM-ics-v1.0]

Integrated Collaboration Object Model (ICOM) for Interoperable Collaboration Services Version 1.0. 16 March 2011. OASIS Committee Specification Draft 01. <http://docs.oasis-open.org/icom/icom-ics/v1.0/csd01/icom-ics-v1.0-csd01.doc>.

Notices

Copyright © OASIS Open 2011. 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 names "OASIS" and "ICOM" are trademarks of [OASIS](#), the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for above guidance.

Table of Contents

1	Introduction.....	8
1.1	Terminology	8
1.2	Normative References	8
1.3	Non-Normative References	9
2	Modeling Language.....	10
2.1	Introduction	10
2.2	Class Definition Grammar.....	10
2.3	Property Definition Grammar	12
2.4	Namespaces.....	14
3	Core Model	16
3.1	Main Branch.....	16
3.1.1	UML Diagram of Entity and Top-Level Subclasses.....	16
3.1.2	Identifiable	16
3.1.3	Parental	17
3.1.4	Extent	18
3.1.5	Entity.....	19
3.1.6	EntityDefinition.....	23
3.2	Scope Branch	24
3.2.1	UML Diagram of Scope and Top-Level Subclasses	24
3.2.2	Scope	24
3.2.3	Community	27
3.2.4	Space	29
3.3	Subject Branch	31
3.3.1	UML Diagram of Subject and Top-Level Subclasses	31
3.3.2	Subject.....	31
3.3.3	Group.....	33
3.3.4	Actor	35
3.3.5	User	37
3.3.6	ResourceActor.....	39
3.3.7	ResourceType	42
3.3.8	ResourceBookingRule.....	43
3.4	Artifact Branch	44
3.4.1	UML Diagram of Artifact and Top-Level Subclasses	44
3.4.2	Item.....	44
3.4.3	SpaceItem	46
3.4.4	Container	46
3.4.5	FolderContainer.....	47
3.4.6	Artifact	48
3.4.7	Folder	50
3.4.8	HeterogeneousFolder.....	51
3.5	Access Control Model.....	53
3.5.1	Accessor.....	53

3.5.2 Owner	54
3.5.3 RoleDefinition	55
3.5.4 Role	56
3.5.5 Privilege.....	57
3.5.6 PrivilegeEnum	58
3.5.7 AccessControlList.....	59
3.5.8 AccessControlEntry	60
3.5.9 AccessType	61
3.5.10 AccessTypeEnum	62
3.6 Metadata Model	64
3.6.1 PropertyDefinition.....	64
3.6.2 Property.....	66
3.6.3 PropertyChoiceType.....	68
3.6.4 PropertyType	69
3.6.5 PropertyTypeEnum	70
3.6.6 Cardinality.....	71
3.6.7 UML Diagram of Marker and Subclasses	72
3.6.8 Marker	72
3.6.9 Category	73
3.6.10 CategoryApplication	75
3.6.11 Tag	76
3.6.12 TagApplication.....	78
3.6.13 RelationshipBondable	79
3.6.14 RelationshipDefinition.....	80
3.6.15 Relationship.....	81
3.7 Common Concepts	83
3.7.1 Participant.....	83
3.7.2 Priority	85
3.7.3 DateTimeResolution.....	86
3.7.4 Location	86
3.7.5 GeoCoordinates	88
4 Extension Modules	90
4.1 Content Module	90
4.1.1 Content	90
4.1.2 MultiContent	92
4.1.3 SimpleContent.....	93
4.1.4 OnlineContent.....	95
4.1.5 ContentDispositionType	96
4.1.6 Attachment	97
4.2 Document Module.....	98
4.2.1 Versionable.....	98
4.2.2 VersionControlMetadata.....	100
4.2.3 VersionSeries	101
4.2.4 Version	103

4.2.5 VersionType	105
4.2.6 Document	106
4.2.7 WikiPage	108
4.3 Message Module	110
4.3.1 MimeConvertible	110
4.3.2 Message	111
4.3.3 UnifiedMessage.....	112
4.3.4 UnifiedMessageFlag.....	116
4.3.5 UnifiedMessageDeliveryStatusNotificationRequest	117
4.3.6 UnifiedMessageChannel	118
4.3.7 UnifiedMessageEditMode	119
4.3.8 InstantMessage	120
4.3.9 InstantMessageType	123
4.3.10 InstantMessageChatStatus	124
4.3.11 InstantMessageFeed.....	125
4.3.12 InstantMessageConnection.....	127
4.4 Presence Module	129
4.4.1 Presence	129
4.4.2 PresenceEditMode	131
4.4.3 ContactMethod	132
4.4.4 ContactReachabilityStatus	134
4.4.5 Activity	136
4.4.6 ActivityType	138
4.5 Address Book Module.....	139
4.5.1 Addressable.....	139
4.5.2 Person	140
4.5.3 AddressBook	143
4.5.4 Contact	145
4.6 Calendar Module	149
4.6.1 Calendar	149
4.6.2 OccurrenceSeries.....	151
4.6.3 Occurrence	156
4.6.4 OccurrenceStatus.....	161
4.6.5 OccurrenceType	162
4.6.6 OccurrenceEditMode.....	163
4.6.7 ParticipantTransparency	164
4.6.8 OccurrenceParticipant.....	164
4.6.9 OccurrenceParticipantStatus.....	165
4.7 FreeBusy Module.....	166
4.7.1 FreeBusy	166
4.7.2 FreeBusyInterval	168
4.7.3 FreeBusyType	170
4.8 TaskList Module.....	171
4.8.1 TaskList	171

4.8.2 Task	172
4.8.3 TaskStatus.....	177
4.8.4 TaskEditMode.....	177
4.8.5 TaskParticipantStatus	178
4.9 Forum Module.....	179
4.9.1 Discussion	179
4.9.2 DiscussionContainer	180
4.9.3 DiscussionMessage	181
4.9.4 Forum	183
4.9.5 Topic.....	185
4.9.6 Announcement	186
4.9.7 AnnouncementStatus	187
4.10 Conference Module	188
4.10.1 Conference	188
4.10.2 ConferenceType.....	191
4.10.3 ConferenceState	192
4.10.4 ConferenceSession	193
4.10.5 ConferenceSessionEndingReason	195
4.10.6 ConferenceSetting.....	196
4.10.7 ConferenceParticipantRole	197
5 Conformance.....	199
A. Acknowledgements	200
B. Non-Normative Text	201
C. Revision History.....	202

1 Introduction

The Integrated Collaboration Object Model (ICOM) for Interoperable Collaboration Services standard defines a framework for a broad range of domain models for collaboration activities in an integrated and interoperable collaboration environment. The framework is intended as a basis for integrating a broad range of collaboration objects to enable seamless transitions across collaboration activities. This enables applications to support continuity of conversations across multiple collaboration activities.

ICOM encompasses and integrates a range of models which are part of existing standards and technologies, several of which are referenced in Section 1.3 Non-Normative References. The model is defined in modular and extensible way, with core concepts, metadata concepts, and their relations included 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: for example in process integration domain which includes 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 objects.

1.1 Terminology

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

1.2 Normative References

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

1.3 Non-Normative References

- [BPEL4People]** OASIS Committee Specification, *WS-BPEL Extension for People (BPEL4People) Specification Version 1.1*, August 2010. <http://docs.oasis-open.org/bpel4people/bpel4people-1.1.html>
- [FOAF]** Brickley, D. and Miller, L., "FOAF Vocabulary Specification", August 2009. (<http://xmlns.com/foaf/spec/>)
- [ICOM Wiki]** OASIS ICOM TC Wiki, (<http://wiki.oasis-open.org/icom>)
- [JCR 2.0]** Java Specification Request (JSR) 283, *Content Repository for Java™ Technology API 2.0 Specification*, August 2009. (<http://jcp.org/en/jsr/detail?id=283>)
- [OpenGraph]** Facebook Platform Open Graph Core Concepts, (<http://developers.facebook.com/docs/coreconcepts/>)
- [OpenSocial]** OpenSocial and Gadgets Specification Group, "Social Data Specification", November 2010. (<http://opensocial-resources.googlecode.com/svn/spec/2.0/Social-Data.xml>)
- [RFC2119]** Crispin, M., "Internet Message Access Protocol – Version 4rev1", RFC 2060, December 1996. (<http://tools.ietf.org/html/rfc2060>)
- [RFC2426]** Dawson, F. and Howes, T., "vCard MIME Directory Profile", RFC 2426, September 1998. (<http://tools.ietf.org/html/rfc2426>)
- [RFC3920]** Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 3920, October 2004. (<http://tools.ietf.org/html/rfc3920>)
- [RFC3921]** Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence", RFC 3921, October 2004. (<http://tools.ietf.org/html/rfc3921>)
- [RFC4512]** Zeilenga, K., "Lightweight Directory Access Protocol (LDAP): Directory Information Models", RFC 4512, June 2006. (<http://tools.ietf.org/html/rfc4512>)
- [RFC4791]** Daboo, C. and Desruisseaux, B., "Calendaring Extensions to WebDAV (CalDAV)", RFC 4791, March 2007. (<http://tools.ietf.org/html/rfc4791>)
- [RFC4918]** Dusseault, L., "HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)", RFC 4918, June 2007. (<http://tools.ietf.org/html/rfc4918>)
- [RFC5321]** Klensin, J., "Simple Mail Transfer Protocol, Draft Standard" RFC 5321, October 2008. (<http://tools.ietf.org/html/rfc5321>)
- [RFC5545]** Desruisseaux, B., "Internet Calendaring and Scheduling Core Object Specification (iCalendar)", RFC 5545, September 2009. (<http://tools.ietf.org/html/rfc5545>)
- [SIOC]** W3C Member Submission, "SIOC Core Ontology Specification", June 2007. (<http://www.w3.org/Submission/2007/SUBM-sioc-spec-20070612/>)
- [WS-BPEL]** OASIS Standard, *Web Services Business Process Execution Language Version 2.0*, April 2007. <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html>
- [WS-HumanTask]** OASIS Committee Specification, *Web Services – Human Task (WS-HumanTask) Specification Version 1.1*, CS-01, August 2010. <http://docs.oasis-open.org/bpel4people/ws-humantask-1.1-spec-cs-01.html>

2 Modeling Language

2.1 Introduction

ICOM specifies a schema of objects in a collaboration environment, in terms of classes 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.

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

Each class is defined by 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 representing a set of zero or more property definitions.

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 `isEnumerated` attribute is **TRUE**.

propertyDefinition property-definition (multi-valued)

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

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

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

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

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

2.3 Property Definition Grammar

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

namespace String

The **namespace** attribute specifies an IRI.

localName String

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

description String (optional)

The **description** attribute specifies a description of a property

propertyType Enum

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

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

- **string** (xsd:string)
- **boolean** (xsd:boolean)
- **decimal** (xsd:decimal)
- **integer** (xsd:integer)
- **datetime** (xsd:dateTime)
- **uri** (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.

218 • **ReadWrite:** The property value can be modified.

219 • **OnCreate:** The property value **MUST** only be update-able during the creation (a create

220 operation) of an object.

221

222 **inherited** Boolean

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

224 The values of **inherited** attribute are:

225 • **TRUE** if a property definition is inherited from a super class;

226 • **FALSE** if a property definition is explicitly defined for a class.

227

228 **required** Boolean

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

230 whose value is provided by application.

231 The values of **required** attribute are:

232 • **TRUE** if the value of a property **MUST** never be set to the “not set” state when an object of

233 this type is created or updated. If a value is not provided during a create or update

234 operation, a service provider **MUST** provide a value for the property. If a value is not

235 provided, then a default value defined for the property **MUST** be set. If no default value is

236 defined, a service provider **MUST** throw an exception.

237 • **FALSE** if the value of a property **MAY** be set to the “not set” state when an object of this

238 type is created or updated.

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

240 case, **required** attribute **SHOULD** be set to **FALSE**.

241 Note: The value of a read-only property (such as **icom:objectId**, **icom:createdBy**) is set by

242 a service provider. Hence, the value of the **required** attribute **SHOULD** be **FALSE** because it is

243 read only for applications.

244

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

246 The **choices** attribute specifies an explicit ordered set of single values allowed for this property.

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

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

249 If the value of **cardinality** attribute is **single** and the value of **openChoice** attribute

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

251 attribute.

252 If the value of **cardinality** attribute is **single** and the value of **openChoice** attribute

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

254 If the value of **cardinality** attribute is **Multi** and the value of **openChoice** attribute

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

256 listed in **choices** attribute.

257 If the value of **cardinality** attribute is **Multi** and the value of **openChoice** attribute

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

259 **choices** attribute.

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

261 specified by the **propertyType** attribute of a property definition.

262

263 **openChoice** Boolean

264 The **openChoice** attribute specifies whether the value of a property must be listed in **choices**

265 attribute. It is applicable only when **choices** attribute is set.

266 The values of **openChoice** attribute are:

267 • **TRUE** if a value of a property MAY be other than those listed in **choices** attribute;

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

269

270 **defaultValue** **property-type**

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

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

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

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

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

276 service provider MUST throw an exception.

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

278 **propertyType** attribute of a property definition.

279

280 **minValue** Integer | Decimal

281 The minimum value allowed for a property. It is applicable only when the **propertyType**

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

283

284 **maxValue** Integer | Decimal

285 The maximum value allowed for a property. It is applicable only when the **propertyType**

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

287

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

289 **displayName** String

290 The **displayName** attribute specifies a string for presentation by application.

291

292 **value** **property-type**

293 The **value** attribute specifies a value compatible with the **property-type** specified by the

294 **propertyType** attribute of a property definition.

295

296 2.4 Namespaces

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

298 The class definition includes the two attributes: **namespace** **and** **localName**. The **namespace**

299 specifies one of the namespace prefixes in Table 1. The **localName** specifies an unprefix name of a

300 class. Syntactically, the namespace qualifies the local name.

301

302 *Table 1 Namespace prefixes and IRI references.*

icom = http://docs.oasis-open.org/ns/icom/core/201008
icom_ac = http://docs.oasis-open.org/ns/icom/accesscontrol/201008

icom_mata = http://docs.oasis-open.org/ns/icom/metadata/201008
icom_content = http://docs.oasis-open.org/ns/icom/content/201008
icom_doc = http://docs.oasis-open.org/ns/icom/document/201008
icom_msg = http://docs.oasis-open.org/ns/icom/message/201008
icom_card = http://docs.oasis-open.org/ns/icom/addressbook/201008
icom_presence = http://docs.oasis-open.org/ns/icom/presence/201008
icom_cal = http://docs.oasis-open.org/ns/icom/calendar/201008
icom_forum = http://docs.oasis-open.org/ns/icom/forum/201008
icom_conf = http://docs.oasis-open.org/ns/icom/conference/201008

303
304 Note: The namespace prefix icom represents the IRI reference [http://docs.oasis-](http://docs.oasis-open.org/ns/icom/core/201008)
305 [open.org/ns/icom/core/201008](http://docs.oasis-open.org/ns/icom/core/201008) for ICOM core namespace. Both the unprefixd name Entity and prefixed
306 name icom:Entity are qualified names that SHALL be interpreted by the expanded name
307 <http://docs.oasis-open.org/ns/icom/core/201008#Entity>.

3 Core Model

3.1 Main Branch

3.1.1 UML Diagram of Entity and Top-Level Subclasses

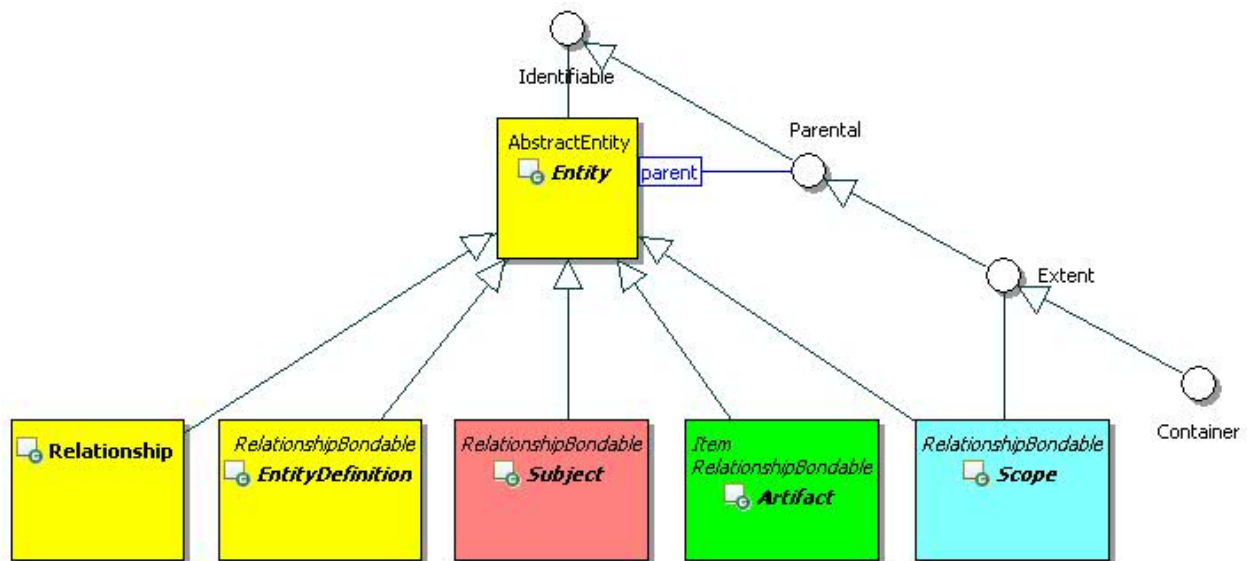


Figure 1 Entity and Top-Level Subclasses

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 mix-in class which defines the characteristics of entities and non-entities that can be uniquely identified.

The **Identifiable** class is defined by the attribute values:

localNamespace

Value: icom

localName

Value: Identifiable

extendsFrom

Value:

331

332 **stereotype**

333 Value: mixin

334

335 **description**

336 Value: Identifiable is a mixin class which defines the characteristics of all entities and some non-

337 entities that can be uniquely identified.

338

339 **propertyDefinitions**

340 The values for this attribute are defined in Section 3.1.2.3.

341 **3.1.2.3 Property Definitions**

342 The Identifiable class MUST have the property definitions:

343

344 **icom:objectId**

345 Description:	A persistent identifier of an object.
346 Required:	False
347 Inherited:	False
348 Property Type:	ID
349 Cardinality:	Single
350 Updatability:	Read Only

351

352 **icom:changeToken**

353 Description:	An opaque token used for optimistic locking & concurrency
354 checking:	
355 Required:	False
356 Inherited:	False
357 Property Type:	String
358 Cardinality:	Single
359 Updatability:	Read Only

360

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

362

363 **3.1.3 Parental**

364 **3.1.3.1 Description**

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

366 **3.1.3.2 Class Definition**

367 The Parental class is a mixin class which defines the characteristics of entities that may be parents of

368 other entities or identifiable objects.

369 The Parental class is defined by the attribute values:

370

371 **localNamespace**
372 Value: icom
373
374 **localName**
375 Value: Parental
376
377 **extendsFrom**
378 Value: icom:Identifiable
379
380 **stereotype**
381 Value: mixin
382
383 **description**
384 Value: Parental is a mixin class which defines the characteristics of the entities that can be
385 parents of other entities or identifiable objects.
386
387 **propertyDefinitions**
388 The values for this attribute are defined in Section 3.1.3.3.

389 **3.1.3.3 Property Definitions**

390 The Parental class inherits property definitions from super classes.
391 The Parental class MAY include additional property definitions which are implementation-defined.
392

393 **3.1.4 Extent**

394 **3.1.4.1 Description**

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

396 **3.1.4.2 Class Definition**

397 The Extent class is a mixin class which defines characteristics of entities that may contain other entities.
398 The Extent class is defined by the attribute values:

399
400 **localNamespace**
401 Value: icom
402
403 **localName**
404 Value: Extent
405
406 **extendsFrom**
407 Value: icom:Parental
408
409 **stereotype**
410 Value: mixin

411
412 **description**
413 Value: Extent is a mixin class which defines the characteristics of entities that may contain other
414 entities.
415
416 **propertyDefinitions**
417 The values for this attribute are defined in Section 3.1.4.3.

418 3.1.4.3 Property Definitions

419 The Extent class inherits property definitions from super classes.
420 The Extent class MUST have the property definitions:

421
422 **icom:parent**
423 Description: Parent of an extent.
424 Required: False
425 Inherited: False
426 Property Type: icom:Extent
427 Cardinality: Single
428 Updatability: Read Only
429

430 The Extent class MAY include additional property definitions which are implementation-defined.
431

432 3.1.5 Entity

433 3.1.5.1 Description

434 An entity is an identifiable object that can be persisted and that has an access control list.
435 Each entity is assigned an internationalized resource identifier (IRI) composed from its *objectId*. The form
436 of the IRI is implementation-dependent.

437 3.1.5.2 Class Definition

438 The Entity class is defined by the attribute values:

439
440 **localNamespace**
441 Value: icom
442
443 **localName**
444 Value: Entity
445
446 **extendsFrom**
447 Value: icom:Identifiable
448
449 **stereotype**
450 Value: primary

451
452 **isAbstract**
453 Value: TRUE
454
455 **description**
456 Value: An entity is an object that has an immutable id and can be individually access controlled.
457
458 **propertyDefinitions**
459 The values for this attribute are defined in Section 3.1.5.3.

460 **3.1.5.3 Property Definitions**

461 The Entity class inherits property definitions from super classes.
462 The Entity class MUST have the property definitions:

463 **icom:name**

465 Description:	Name of an entity.
466 Required:	False
467 Inherited:	False
468 Property Type:	String
469 Cardinality:	Single
470 Updatability:	Read Write

471

472 **icom:createdBy**

473 Description:	An actor who creates an entity.
474 Required:	False
475 Inherited:	False
476 Property Type:	icom:Actor
477 Cardinality:	Single
478 Updatability:	Read Only

479

480 **icom:creationDate**

481 Description:	Date and time when an entity is created. It is immutable.
482 Required:	False
483 Inherited:	False
484 Property Type:	DateTime
485 Cardinality:	Single
486 Updatability:	Read Only

487

488 **icom:lastModifiedBy**

489 Description:	An actor who last modified an entity.
490 Required:	False
491 Inherited:	False
492 Property Type:	icom:Actor

493	Cardinality:	Single
494	Updatability:	Read Only
495		
496	icom:lastModificationDate	
497	Description:	Date and time when an entity is last modified.
498	Required:	False
499	Inherited:	False
500	Property Type:	DateTime
501	Cardinality:	Single
502	Updatability:	Read Only
503		
504	icom_ac:owner	
505	Description:	A subject who owns an entity.
506	Required:	True
507	Inherited:	False
508	Property Type:	icom_ac:Owner
509	Cardinality:	Single
510	Updatability:	Read Write
511		
512	icom:parent	
513	Description:	A parental entity which contains an entity.
514	Required:	False
515	Inherited:	False
516	Property Type:	icom:Parental
517	Cardinality:	Single
518	Updatability:	Read Only
519		
520	icom_meta:attachedMarker	
521	Description:	Zero or more markers applied on an entity.
522	Required:	False
523	Inherited:	False
524	Property Type:	icom_meta:Marker
525	Cardinality:	Multi
526	Updatability:	Read Only
527		
528	icom_meta:categoryApplication	
529	Description:	Zero or more category applications on an entity.
530	Required:	False
531	Inherited:	False
532	Property Type:	icom_meta:CategoryApplication
533	Cardinality:	Multi
534	Updatability:	Read Only

icom_meta:tagApplication

Description:	Zero or more tag applications on an entity.
Required:	False
Inherited:	False
Property Type:	icom_meta:TagApplication
Cardinality:	Multi
Updatability:	Read Only

icom_ac:accessControlList

Description:	Access control list on an entity.
Required:	False
Inherited:	False
Property Type:	icom_ac:AccessControlList
Cardinality:	Single
Updatability:	Read Write

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

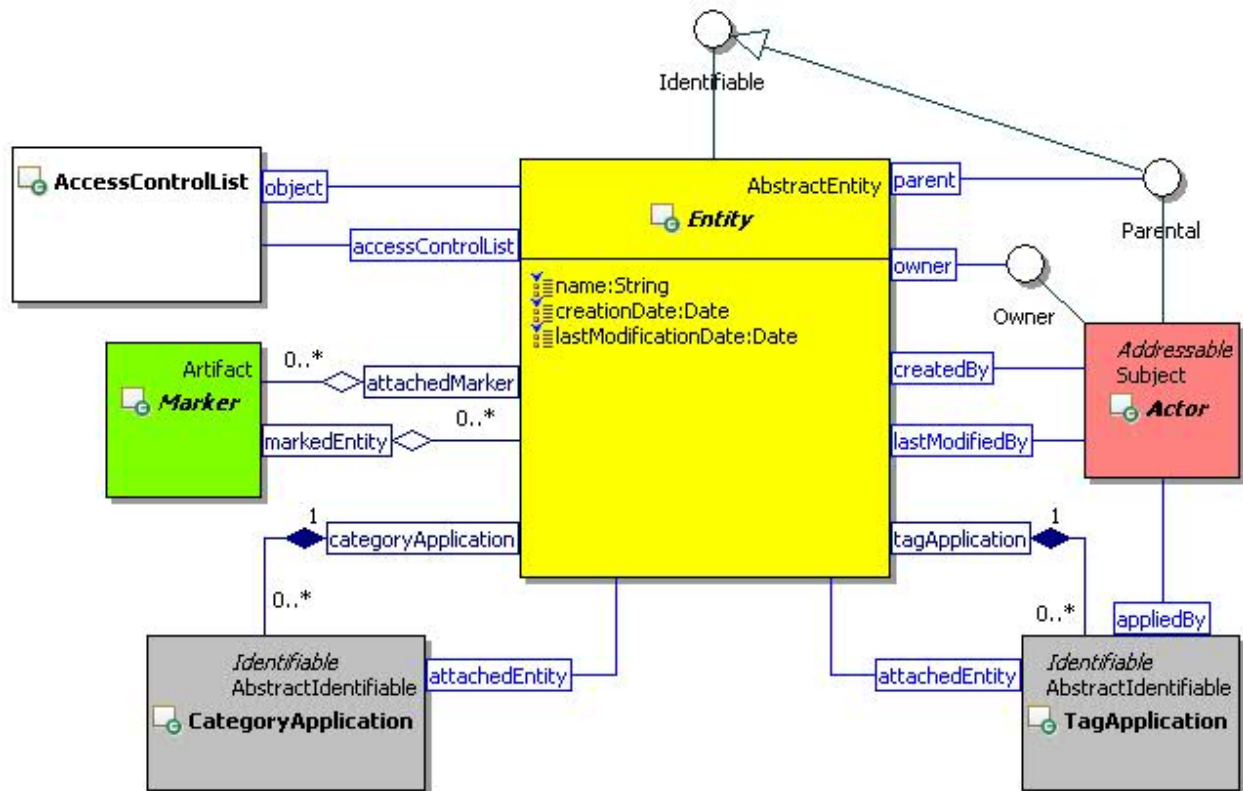


Figure 2 Entity Class Diagram

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 is defined by the attribute values:

localNamespace

Value: icom

localName

Value: EntityDefinition

extendsFrom

Value: icom:Entity, icom_meta: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.

3.1.6.3 Property Definitions

The EntityDefinition class inherits property definitions from super classes.

The EntityDefinition class MUST have the property definitions:

icom: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.

3.2 Scope Branch

3.2.1 UML Diagram of Scope and Top-Level Subclasses

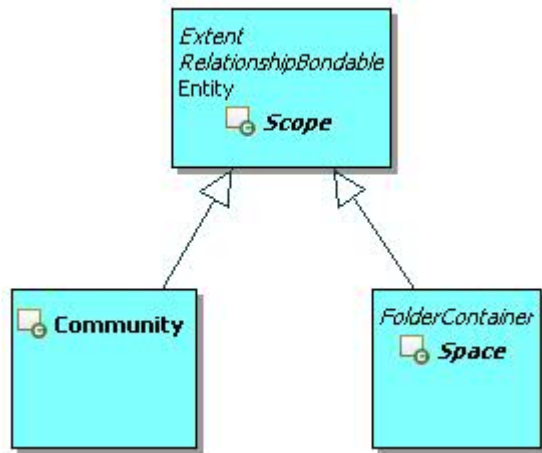


Figure 3 Scope Branch

3.2.2 Scope

3.2.2.1 Description

A scope is an extent of an administrative realm.

3.2.2.2 Class Definition

The Scope class is defined by the attribute values:

localNamespace

Value: icom

localName

Value: Scope

extendsFrom

Value: icom:Entity, icom:Extent, icom_meta:RelationshipBondable

stereotype

Value: primary

isAbstract

Value: TRUE

description

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

625
626 **propertyDefinitions**
627 The values for this attribute are defined in Section 3.2.2.3.

628 **3.2.2.3 Property Definitions**

629 The Scope class inherits property definitions from super classes.
630 The Scope class MUST have the property definitions:

631
632 **icom:description**
633 Description: A description of a scope.
634 Required: False
635 Inherited: False
636 Property Type: String
637 Cardinality: Single
638 Updatability: Read Write

639
640 **icom:parent**
641 Description: A community which contains a scope.
642 Required: False
643 Inherited: True
644 Property Type: icom:Community
645 Cardinality: Single
646 Updatability: Read Only

647
648 **icom:roleDefinition**
649 Description: Zero or more role definitions defined in a scope.
650 Required: False
651 Inherited: False
652 Property Type: icom_ac:RoleDefinition
653 Cardinality: Multi
654 Updatability: Read Only

655
656 **icom:role**
657 Description: Zero or more roles defined in a scope.
658 Required: False
659 Inherited: False
660 Property Type: icom_ac:Role
661 Cardinality: Multi
662 Updatability: Read Only

663
664 **icom:group**
665 Description: Zero or more groups defined in a scope.
666 Required: False

667	Inherited:	False
668	Property Type:	icom:Group
669	Cardinality:	Multi
670	Updatability:	Read Only
671		
672	icom:memberGroup	
673	Description:	Member groups of a scope, i.e. groups whose assigned
674		scopes include this scope.
675	Required:	False
676	Inherited:	False
677	Property Type:	icom:Group
678	Cardinality:	Multi
679	Updatability:	Read Only
680		
681	icom_meta:relationship	
682	Description:	Zero or more relationships associated with a scope.
683	Required:	False
684	Inherited:	False
685	Property Type:	icom_meta:Relationship
686	Cardinality:	Multi
687	Updatability:	Read Only
688		
689	The Scope class MAY include additional property definitions which are implementation-defined.	
690		

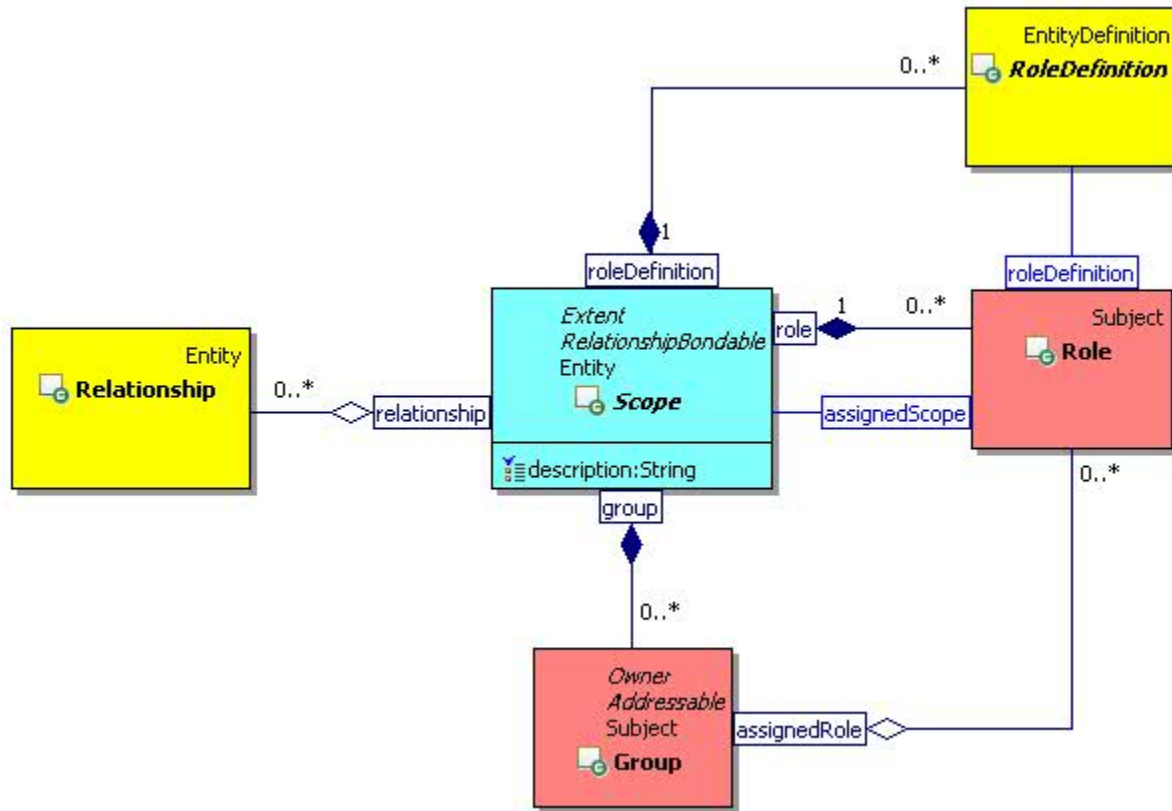


Figure 4 Scope Class Diagram

3.2.3 Community

3.2.3.1 Description

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

3.2.3.2 Class Definition

The Community class is defined by the attribute values:

localNamespace

Value: icom

localName

Value: Community

extendsFrom

Value: icom:Scope

stereotype

712 Value: primary
 713
 714 **description**
 715 Value: A community is a scope that has a set of actors as members who can participate in a set
 716 of spaces.
 717
 718 **propertyDefinitions**
 719 The values for this attribute are defined in Section 3.2.3.3.

720 3.2.3.3 Property Definitions

721 The Community class inherits property definitions from super classes.
 722 The Community class MUST have the property definitions:

724 **icom:community**

725	Description:	Sub-communities of a community.
726	Required:	False
727	Inherited:	False
728	Property Type:	icom:Community
729	Cardinality:	Multi
730	Updatability:	Read Only

732 **icom:space**

733	Description:	Spaces of a community.
734	Required:	False
735	Inherited:	False
736	Property Type:	icom:Space
737	Cardinality:	Multi
738	Updatability:	Read Only

740 **icom:actor**

741	Description:	Managed actors of a community, i.e. actors whose parent
742		community is this community.
743	Required:	False
744	Inherited:	False
745	Property Type:	icom:Actor
746	Cardinality:	Multi
747	Updatability:	Read Only

749 **icom:memberActor**

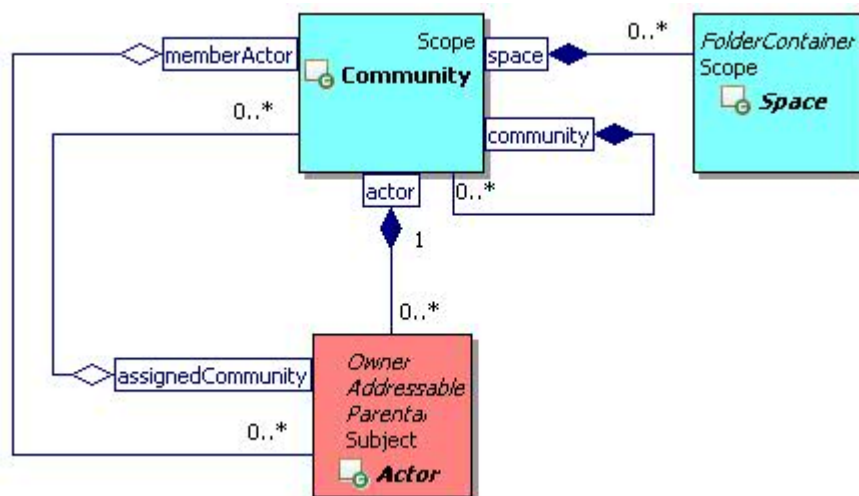
750	Description:	Member actors of a community, i.e. actors whose assigned
751		communities include this community.
752	Required:	False
753	Inherited:	False

754 Property Type: icom:Actor
 755 Cardinality: Multi
 756 Updatability: Read Only

757

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

759



760

761 Figure 5 Community Class Diagram

762

763 3.2.4 Space

764 3.2.4.1 Description

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

766 3.2.4.2 Class Definition

767 The Space class is defined by the attribute values:

768

769 **localNamespace**

770 Value: icom

771

772 **localName**

773 Value: Space

774

775 **extendsFrom**

776 Value: icom:Scope, icom:FolderContainer

777

778 **stereotype**

779 Value: primary

780

781 **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 definitions:

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

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

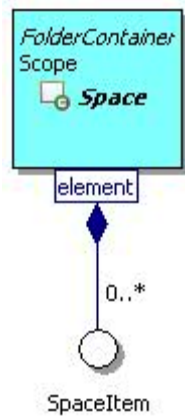


Figure 6 Space Class Diagram

3.3 Subject Branch

3.3.1 UML Diagram of Subject and Top-Level Subclasses

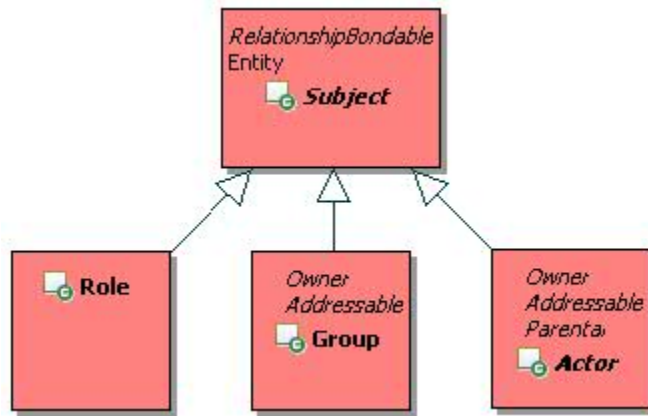


Figure 7 Subject Branch

3.3.2 Subject

3.3.2.1 Description

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

3.3.2.2 Class Definition

The Subject class is defined by the attribute values:

```
localNamespace
    Value: icom

localName
    Value: Subject

extendsFrom
    Value: icom:Entity, icom_meta:RelationshipBondable

stereotype
    Value: primary

isAbstract
    Value: TRUE

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

833 **propertyDefinitions**

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

835 3.3.2.3 Property Definitions

836 The Subject class inherits property definitions from super classes.

837 The Subject class MUST have the property definitions:

838

839 **icom:description**

840 Description: A description of a subject.

841 Required: False

842 Inherited: False

843 Property Type: String

844 Cardinality: Single

845 Updatability: Read Write

846

847 **icom:parent**

848 Description: A scope which contains a subject.

849 Required: False

850 Inherited: True

851 Property Type: icom:Scope

852 Cardinality: Single

853 Updatability: Read Only

854

855 **icom_meta:relationship**

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

857 Required: False

858 Inherited: False

859 Property Type: icom_meta:Relationship

860 Cardinality: Multi

861 Updatability: Read Only

862

863 **icom_meta:property**

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

865 Required: False

866 Inherited: False

867 Property Type: icom_meta:Property

868 Cardinality: Multi

869 Updatability: Read Write

870

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

872

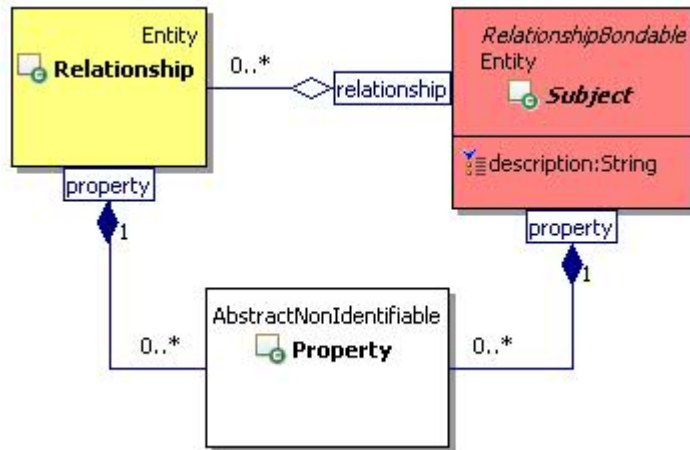


Figure 8 Subject Class Diagram

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 is defined by the attribute values:

localNamespace

Value: icom

localName

Value: Group

extendsFrom

Value: icom:Subject, icom_card: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:assignedRole

Description:	Roles to which a group is assigned.
Required:	False
Inherited:	False
Property Type:	icom:Role
Cardinality:	Multi
Updatability:	Read Write

icom:assignedGroup

Description:	Super-groups to which a group is assigned.
Required:	False
Inherited:	False
Property Type:	icom:Group
Cardinality:	Multi
Updatability:	Read Write

icom:assignedScope

Description:	Scopes to which a group is assigned.
Required:	False
Inherited:	False
Property Type:	icom:Scope
Cardinality:	Multi
Updatability:	Read Write

icom:memberGroup

Description:	Sub-groups assigned to a group.
Required:	False
Inherited:	False
Property Type:	icom:Group
Cardinality:	Multi
Updatability:	Read Only

icom:memberActor

Description:	Actors assigned to a group.
Required:	False
Inherited:	False
Property Type:	icom:Actor
Cardinality:	Multi

Updatability: Read Only

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

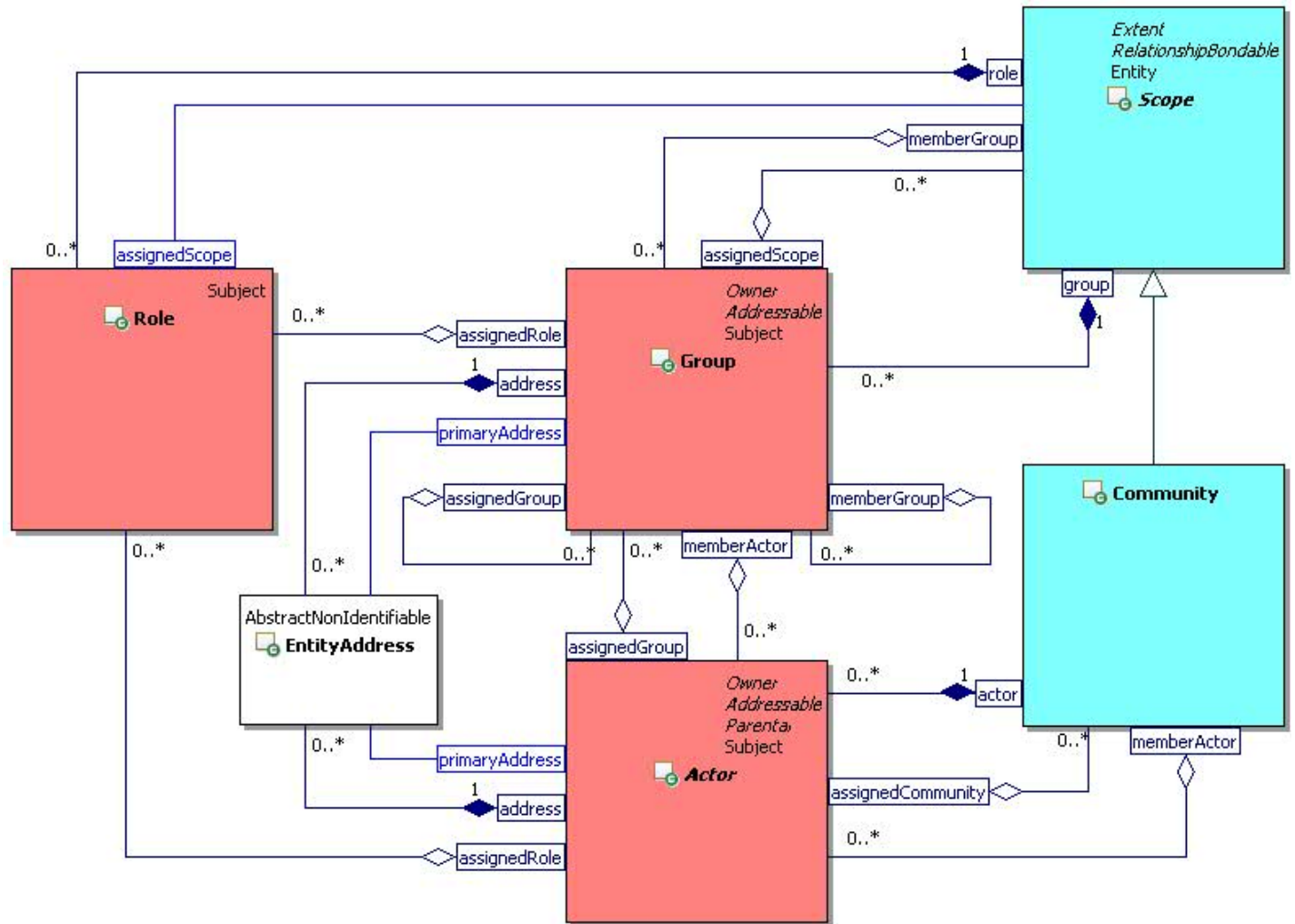


Figure 9 Group and Actor Class Diagram

3.3.4 Actor

3.3.4.1 Description

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

3.3.4.2 Class Definition

The Actor class is defined by the attribute values:

localNamespace

Value: icom

961

962 **localName**

963 Value: Actor

964

965 **extendsFrom**

966 Value: icom:Subject, icom_card:Addressable, icom_ac:Owner

967

968 **stereotype**

969 Value: primary

970

971 **isAbstract**

972 Value: TRUE

973

974 **description**

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

976

977 **propertyDefinitions**

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

979 3.3.4.3 Property Definitions

980 The Actor class inherits property definitions from super classes.

981 The Actor class MUST have the property definitions:

982

983 **icom:parent**

984 Description: A community which contains an actor.

985 Required: False

986 Inherited: True

987 Property Type: icom:Community

988 Cardinality: Single

989 Updatability: Read Only

990

991 **icom:assignedRole**

992 Description: Roles to which an actor is assigned.

993 Required: False

994 Inherited: False

995 Property Type: icom:Role

996 Cardinality: Multi

997 Updatability: Read Write

998

999 **icom:assignedGroup**

1000 Description: Groups to which an actor is assigned.

1001 Required: False

1002 Inherited: False

1003	Property Type:	icom:Group
1004	Cardinality:	Multi
1005	Updatability:	Read Write
1006		
1007	icom:assignedCommunity	
1008	Description:	Communities to which an actor is assigned.
1009	Required:	False
1010	Inherited:	False
1011	Property Type:	icom:Community
1012	Cardinality:	Multi
1013	Updatability:	Read Write
1014		
1015	The Actor class MAY include additional property definitions which are implementation-defined.	
1016		

1017 3.3.5 User

1018 3.3.5.1 Description

1019 A user is an actor who is also a person.
 1020 A user has a personal space.

1021 3.3.5.2 Class Definition

1022 The User class is defined by the attribute values:

1023	
1024	localNamespace
1025	Value: icom
1026	
1027	localName
1028	Value: User
1029	
1030	extendsFrom
1031	Value: icom:Actor, icom_card:Person
1032	
1033	stereotype
1034	Value: primary
1035	
1036	description
1037	Value: A user is an actor who is also a person.
1038	
1039	propertyDefinitions
1040	The values for this attribute are defined in Section 3.3.5.3.

3.3.5.3 Property Definitions

The User class inherits property definitions from super classes.

The User class MUST have the property definitions:

icom:personalSpace

Description:	Personal space of a person.
Required:	False
Inherited:	False
Property Type:	icom:Space
Cardinality:	Single
Updatability:	Read Only

icom_presence:presence

Description:	Presence of a person.
Required:	False
Inherited:	False
Property Type:	icom_presence:Presence
Cardinality:	Single
Updatability:	Read Only

icom_msg:instantMessageFeed

Description:	Instant message feed for a person.
Required:	False
Inherited:	False
Property Type:	icom_msg:InstantMessageFeed
Cardinality:	Single
Updatability:	Read Only

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

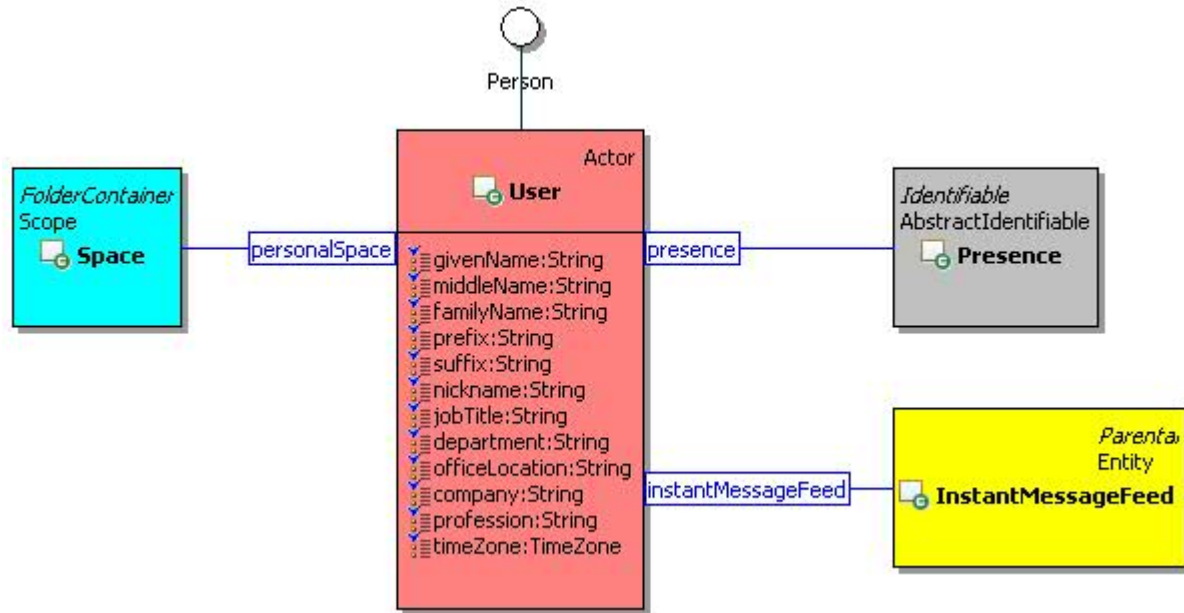


Figure 10 User Class Diagram.

3.3.6 ResourceActor

3.3.6.1 Description

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

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

3.3.6.2 Class Definition

The ResourceActor class is defined by the attribute values:

localNamespace

Value: icom

localName

Value: ResourceActor

extendsFrom

Value: icom:Actor

stereotype

Value: primary

description

1096 Value: A resource actor is an actor representing a bookable resource, such as a conference
1097 room, equipment, or on-line conference.

1098

1099 **propertyDefinitions**

1100 The values for this attribute are defined in Section 3.3.6.3..

1101 **3.3.6.3 Property Definitions**

1102 The ResourceActor class inherits property definitions from super classes.

1103 The ResourceActor class MUST have the property definitions:

1104

1105 **icom:resourceSpace**

1106 Description: Administrative space of a resource actor.

1107 Required: False

1108 Inherited: False

1109 Property Type: icom:Space

1110 Cardinality: Single

1111 Updatability: Read Only

1112

1113 **icom:description**

1114 Description: Description of a resource.

1115 Required: False

1116 Inherited: False

1117 Property Type: String

1118 Cardinality: Single

1119 Updatability: Read Write

1120

1121 **icom:location**

1122 Description: Location of a resource.

1123 Required: False

1124 Inherited: False

1125 Property Type: icom:Location

1126 Cardinality: Single

1127 Updatability: Read Write

1128

1129 **icom:capacity**

1130 Description: Capacity of a resource.

1131 Required: False

1132 Inherited: False

1133 Property Type: Integer

1134 Cardinality: Single

1135 Updatability: Read Write

1136

1137 **icom:resourceType**

1138	Description:	Type of a resource.
1139	Required:	False
1140	Inherited:	False
1141	Property Type:	icom:ResourceType
1142	Cardinality:	Single
1143	Updatability:	Read Write
1144		
1145	icom:bookingRule	
1146	Description:	Resource booking rule.
1147	Required:	False
1148	Inherited:	False
1149	Property Type:	icom:ResourceBookingRule
1150	Cardinality:	Single
1151	Updatability:	Read Write
1152		
1153	icom:bookingApprover	
1154	Description:	One or more users who approve the booking of a resource.
1155	Required:	False
1156	Inherited:	False
1157	Property Type:	icom:User
1158	Cardinality:	Single
1159	Updatability:	Read Write
1160		
1161	The ResourceActor class MAY include additional property definitions which are implementation-defined.	
1162		

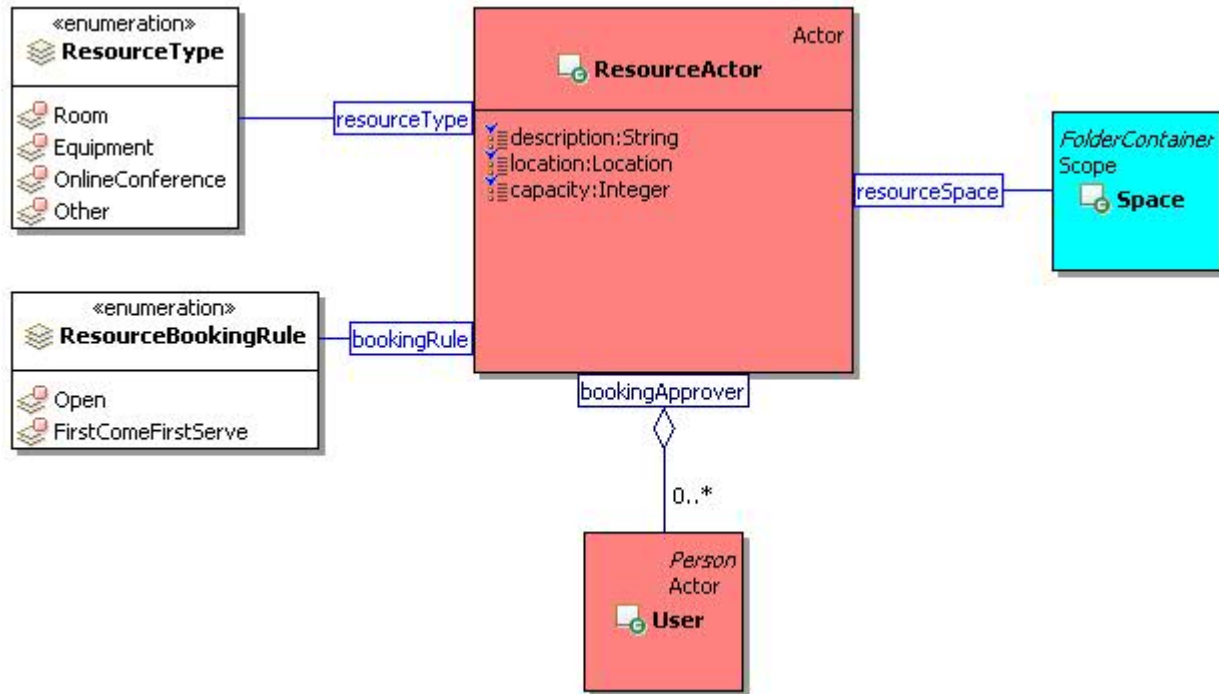


Figure 11 ResourceActor Class Diagram.

3.3.7 ResourceType

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

The ResourceType class is defined by the attribute values:

localNamespace

Value: icom

localName

Value: ResourceType

extendsFrom

Value:

stereotype

Value: primary

isEnumeration

Value: TRUE

description

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

1188
1189 **instances**
1190 Value: <icom:Room, icom:Equipment, icom:OnlineConference, icom:Other>

1191

1192 The following resource types are defined by ICOM:

- 1193 • **icom:Room** to express that a resource represents a room.
- 1194 • **icom:Equipment** to express that a resource represents an equipment.
- 1195 • **icom:OnlineConference** to express that a resource represents an online conference.
- 1196 • **icom:Other** to express that a resource represents other things.

1197

1198 **3.3.8 ResourceBookingRule**

1199 The ResourceBookingRule class is an enum class that enumerates the instances each of which
1200 expresses a booking rule.

1201 The ResourceBookingRule class is defined by the attribute values:

1202

1203 **localNamespace**

1204 Value: icom

1205

1206 **localName**

1207 Value: ResourceBookingRule

1208

1209 **extendsFrom**

1210 Value:

1211

1212 **stereotype**

1213 Value: primary

1214

1215 **isEnumeration**

1216 Value: TRUE

1217

1218 **description**

1219 Value: An enumeration of instances each of which expresses a booking rule.

1220

1221 **instances**

1222 Value: <icom:Open, icom:FirstComeFirstServe>

1223

1224 The following resource booking rules are defined by ICOM:

- 1225 • **icom:Open** to express that a resource is open for booking.
- 1226 • **icom:FirstComeFirstServe** to express that a resource is first come first serve.

1227

3.4 Artifact Branch

3.4.1 UML Diagram of Artifact and Top-Level Subclasses

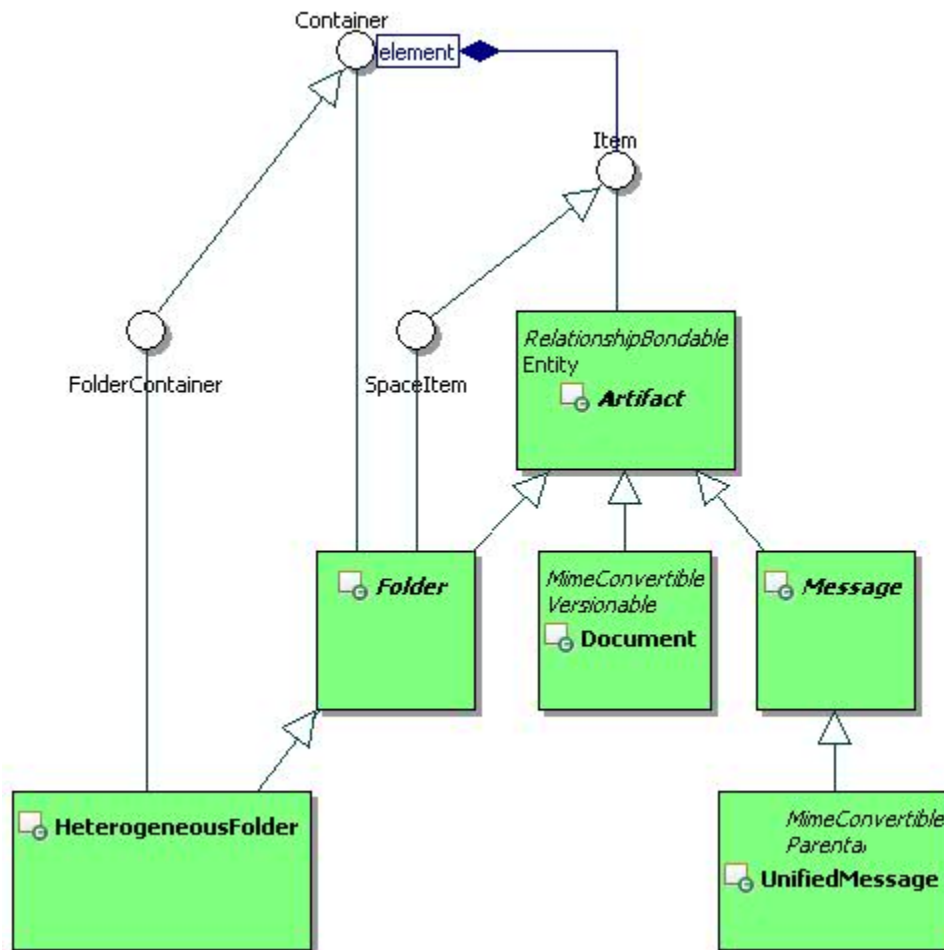


Figure 12 Artifact Branch

3.4.2 Item

3.4.2.1 Description

An item is an element of a container.

The parent of an item MUST be a container.

3.4.2.2 Class Definition

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

The Item class is defined by the attribute values:

localNamespace

Value: icom

1244

1245 **localName**

1246 Value: Item

1247

1248 **extendsFrom**

1249 Value: icom:Identifiable

1250

1251 **stereotype**

1252 Value: mixin

1253

1254 **description**

1255 Value: Item is a mixin class which defines the characteristics of entities that can be placed in a

1256 Container.

1257

1258 **propertyDefinitions**

1259 The values for this attribute are defined in Section 3.4.2.3.

1260 **3.4.2.3 Property Definitions**

1261 The Item class inherits property definitions from super classes.

1262 The Item class **MUST** have the property definition:

1263

1264 **icom:parent**

1265 Description:	A parent container of an item.
1266 Required:	False
1267 Inherited:	True
1268 Property Type:	icom:Container
1269 Cardinality:	Single
1270 Updatability:	Read Only

1271

1272 The Item class **MAY** have the optional property definition:

1273

1274 **icom:container**

1275 Description:	Zero, one, or more containers of an item.
1276 Required:	False
1277 Inherited:	False
1278 Property Type:	icom:Container
1279 Cardinality:	Multi
1280 Updatability:	Read Write

1281

1282 The Item class **MAY** include additional property definitions which are implementation-defined.

1283

1284 3.4.3 Spaceltem

1285 3.4.3.1 Description

1286 A space item is an element of a space.

1287 3.4.3.2 Class Definition

1288 The Spaceltem class is a mixin class which defines the characteristics of entities that can be elements of
1289 a Space.

1290 The Spaceltem class is defined by the attribute values:

1291

1292 **localNamespace**

1293 Value: icom

1294

1295 **localName**

1296 Value: Spaceltem

1297

1298 **extendsFrom**

1299 Value: icom:Item

1300

1301 **stereotype**

1302 Value: mixin

1303

1304 **description**

1305 Value: Spaceltem is a mixin class which defines the characteristics of entities that can be
1306 elements of a Space.

1307

1308 **propertyDefinitions**

1309 The values for this attribute are defined in Section 3.4.3.3.

1310 3.4.3.3 Property Definitions

1311 The Spaceltem class inherits property definitions from super classes.

1312 The Spaceltem class MAY include additional property definitions which are implementation-defined.

1313

1314 3.4.4 Container

1315 3.4.4.1 Description

1316 A container is an extent that contains items.

1317 3.4.4.2 Class Definition

1318 The Container class is mixin class which defines the characteristics of extents that contain items.

1319 The Container class is defined by the attribute values:

1320

1321 **localNamespace**

1322 Value: icom
1323
1324 **localName**
1325 Value: Container
1326
1327 **extendsFrom**
1328 Value: icom:Extent
1329
1330 **stereotype**
1331 Value: mixin
1332
1333 **description**
1334 Value: A container is an extent that contains items.
1335
1336 **propertyDefinitions**
1337 The values for this attribute are defined in Section 3.4.4.3.

1338 **3.4.4.3 Property Definitions**

1339 The Container class inherits property definitions from super classes.
1340 The Container class MUST have the property definitions:

1341

1342	icom:element	
1343	Description:	Elements of a container, i.e. items whose parent container is the container or whose containers include the container.
1344		
1345	Required:	False
1346	Inherited:	False
1347	Property Type:	icom:Item
1348	Cardinality:	Multi
1349	Updatability:	Read Only

1350
1351 The Container class MAY include additional property definitions which are implementation-defined.
1352

1353 **3.4.5 FolderContainer**

1354 **3.4.5.1 Description**

1355 A folder container is a container which may contain folders. Space and heterogeneous folder are folder
1356 containers.

1357 **3.4.5.2 Class Definition**

1358 The FolderContainer class is a mixin class that defines the characteristics of containers that may contain
1359 folders.

1360 The FolderContainer class is defined by the attribute values:

1361

1362 **localNamespace**
1363 Value: icom
1364
1365 **localName**
1366 Value: FolderContainer
1367
1368 **extendsFrom**
1369 Value: icom:Container
1370
1371 **stereotype**
1372 Value: mixin
1373
1374 **description**
1375 Value: A folder container is a container which may contain folders.
1376
1377 **propertyDefinitions**
1378 The values for this attribute are defined in Section 3.4.5.3.

1379 **3.4.5.3 Property Definitions**

1380 The FolderContainer class inherits property definitions from super classes.
1381 The FolderContainer class MAY include additional property definitions which are implementation-defined.
1382

1383 **3.4.6 Artifact**

1384 **3.4.6.1 Description**

1385 An artifact is a result of a communication, cooperation, content creation, or collaboration activity.
1386 Note: Document versioning is an example of content creation activity resulting in an artifact (a version of
1387 document).

1388 **3.4.6.2 Class Definition**

1389 The Artifact class is defined by the attribute values:

1390
1391 **localNamespace**
1392 Value: icom
1393
1394 **localName**
1395 Value: Artifact
1396
1397 **extendsFrom**
1398 Value: icom:Entity, icom:Item, icom_meta:RelationshipBondable
1399 Optional Value: icom:SpaceItem
1400
1401 **stereotype**

1402 Value: primary

1403

1404 **isAbstract**

1405 Value: TRUE

1406

1407 **description**

1408 Value: An artifact is a result of a communication, cooperation, content creation, or collaboration

1409 activity.

1410

1411 **propertyDefinitions**

1412 The values for this attribute are defined in Section 3.4.6.3.

1413 3.4.6.3 Property Definitions

1414 The Artifact class inherits property definitions from super classes.

1415 The Artifact class MUST have the property definitions:

1416

1417 **icom:description**

1418	Description:	A description of an artifact.
1419	Required:	False
1420	Inherited:	False
1421	Property Type:	String
1422	Cardinality:	Single
1423	Updatability:	Read Write

1424

1425 **icom:userCreationDate**

1426	Description:	Date and time when an artifact is created. This field can be
1427		set by application.
1428	Required:	False
1429	Inherited:	False
1430	Property Type:	DateTime
1431	Cardinality:	Single
1432	Updatability:	Read Write

1433

1434 **icom:userLastModificationDate**

1435	Description:	Date and time when an artifact is last modified. This field can
1436		be set by application.
1437	Required:	False
1438	Inherited:	False
1439	Property Type:	DateTime
1440	Cardinality:	Single
1441	Updatability:	Read Write

1442

1443 **icom_meta:property**

1444	Description:	Zero or more extended properties of an artifact.
1445	Required:	False
1446	Inherited:	False
1447	Property Type:	icom_meta:Property
1448	Cardinality:	Multi
1449	Updatability:	Read Write
1450		
1451	icom_meta:relationship	
1452	Description:	Zero or more relationships associated with an artifact.
1453	Required:	False
1454	Inherited:	False
1455	Property Type:	icom_meta:Relationship
1456	Cardinality:	Multi
1457	Updatability:	Read Only
1458		

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

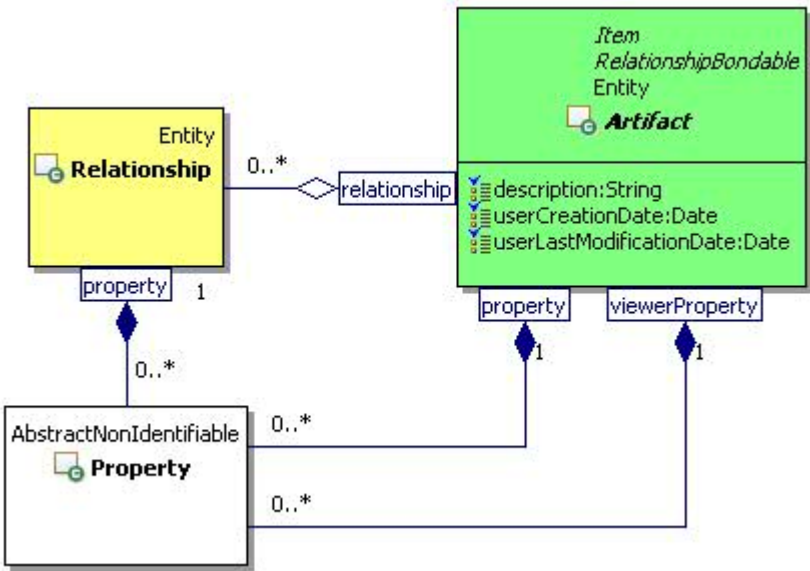


Figure 13 Artifact Class Diagram

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 is defined by the attribute values:

1471

1472 **localNamespace**

1473 Value: icom

1474

1475 **localName**

1476 Value: Folder

1477

1478 **extendsFrom**

1479 Value: icom:Artifact, icom:Container, icom:SpaceItem

1480

1481 **stereotype**

1482 Value: primary

1483

1484 **isAbstract**

1485 Value: TRUE

1486

1487 **description**

1488 Value: A folder is an artifact that may contain other artifacts.

1489

1490 **propertyDefinitions**

1491 The values for this attribute are defined in Section 3.4.7.3.

1492 3.4.7.3 Property Definitions

1493 The Folder class inherits property definitions from super classes.

1494 The Folder class MUST have the property definitions:

1495

1496	icom:parent	
1497	Description:	A parent container of a folder.
1498	Required:	False
1499	Inherited:	True
1500	Property Type:	icom:FolderContainer
1501	Cardinality:	Single
1502	Updatability:	Read Only

1503

1504 The Folder class MAY include additional property definitions which are implementation-defined.

1505

1506 3.4.8 HeterogeneousFolder

1507 3.4.8.1 Description

1508 A heterogeneous folder is an unconstrained folder to contain any type of artifacts. It is typically used for

1509 document folders, inbox, and trash folder of a space.

3.4.8.2 Class Definition

The HeterogeneousFolder class is defined by the attribute values:

localNamespace

Value: icom

localName

Value: HeterogeneousFolder

extendsFrom

Value: icom:Folder, icom:FolderContainer

stereotype

Value: primary

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 definitions:

icom:element

Description: Elements of a heterogeneous folder.

Required: False

Inherited: True

Property Type: icom:Artifact

Cardinality: Multi

Updatability: Read Only

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

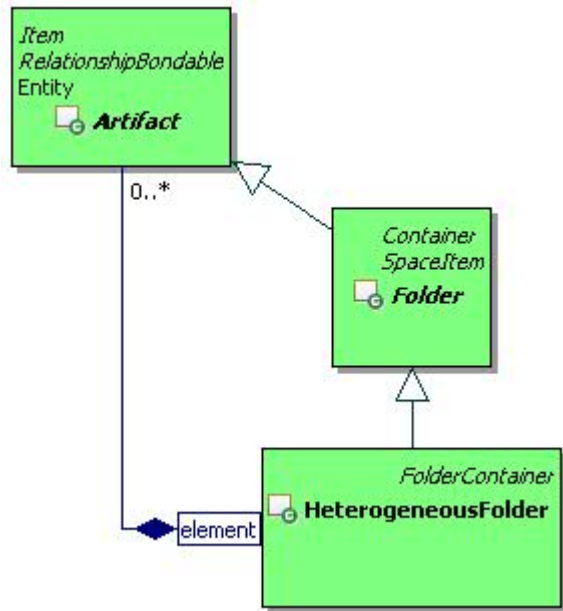


Figure 14 Heterogeneous Folder Class Diagram

3.5 Access Control Model

3.5.1 Accessor

3.5.1.1 Description

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

3.5.1.2 Class Definition

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

The Accessor class is defined by the attribute values:

localNamespace

Value: icom_ac

localName

Value: Accessor

extendsFrom

Value: icom:Identifiable

stereotype

Value: mixin

1570 **description**
1571 Value: Accessor is a mixin class which defines the characteristics of subjects such as groups
1572 and actors that can be granted or denied access types in access control lists and granted
1573 privileges in role assignments.

1574
1575 **propertyDefinitions**
1576 The values for this attribute are defined in Section 3.5.1.3.

1577 **3.5.1.3 Property Definitions**

1578 The Accessor class inherits property definitions from super classes.
1579 The Accessor class MAY include additional property definitions which are implementation-defined.
1580

1581 **3.5.2 Owner**

1582 **3.5.2.1 Description**

1583 An owner is a subject that can be the owner of entities.
1584 An owner of an entity MAY always have rights to update the access control list for the entity.

1585 **3.5.2.2 Class Definition**

1586 The Owner class is a mixin class which defines the characteristics of subjects such as groups and actors
1587 that can own entities.
1588 The Owner class is defined by the attribute values:

1589
1590 **localNamespace**
1591 Value: icom_ac
1592
1593 **localName**
1594 Value: Owner
1595
1596 **extendsFrom**
1597 Value: icom_ac:Accessor
1598
1599 **stereotype**
1600 Value: mixin
1601
1602 **description**
1603 Value: Owner is a mixin class which defines the characteristics of subjects such as groups and
1604 actors that can own entities.

1605
1606 **propertyDefinitions**
1607 The values for this attribute are defined in Section 3.5.2.3.

1608 **3.5.2.3 Property Definitions**

1609 The Owner class inherits property definitions from super classes.

1610 The Owner class MAY include additional property definitions which are implementation-defined.
1611

1612 **3.5.3 RoleDefinition**

1613 **3.5.3.1 Description**

1614 A role definition is a named set of privileges.

1615 **3.5.3.2 Class Definition**

1616 The RoleDefinition class is defined by the attribute values:

1617
1618 **localNamespace**
1619 Value: icom_ac
1620
1621 **localName**
1622 Value: RoleDefinition
1623
1624 **extendsFrom**
1625 Value: icom:EntityDefinition
1626
1627 **stereotype**
1628 Value: primary
1629
1630 **description**
1631 Value: A role definition is a named set of privileges.
1632
1633 **propertyDefinitions**
1634 The values for this attribute are defined in Section 3.5.3.3.

1635 **3.5.3.3 Property Definitions**

1636 The RoleDefinition class inherits property definitions from super classes.

1637 The RoleDefinition class MUST have the property definitions:

1638
1639 **icom_ac:privilege**
1640 Description: A set of privileges.
1641 Required: True
1642 Inherited: False
1643 Property Type: icom_ac:Privilege
1644 Cardinality: Multi
1645 Updatability: Read Write
1646

1647 The RoleDefinition class MAY include additional property definitions which are implementation-defined.
1648

3.5.4 Role

3.5.4.1 Description

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

3.5.4.2 Class Definition

The Role class is defined by the attribute values:

localNamespace

Value: icom_ac

localName

Value: Role

extendsFrom

Value: icom:Subject

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.5.4.3.

3.5.4.3 Property Definitions

The Role class inherits property definitions from super classes.

The Role class MUST have the property definitions:

icom_ac:roleDefinition

Description: A role definition containing a set of privileges.

Required: True

Inherited: False

Property Type: icom_ac:RoleDefinition

Cardinality: Single

Updatability: On Create

icom_ac:assignedScope

Description: A scope in which a role is assigned.

Required: True

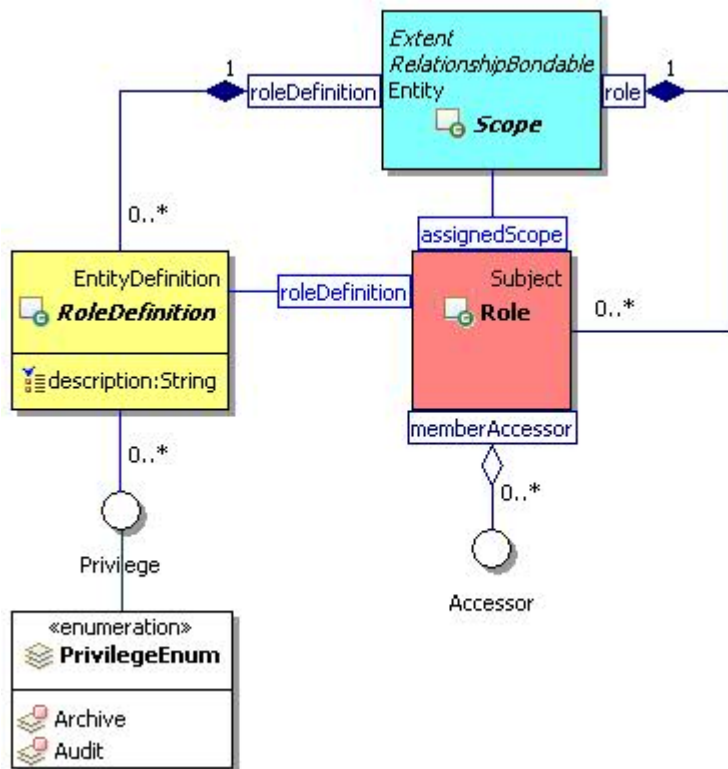
Inherited: False

1689 Property Type: icom:Scope
1690 Cardinality: Single
1691 Updatability: Read Write

1692
1693 **icom_ac:memberAccessor**

1694 Description: Accessors (actors and groups) assigned to a role.
1695 Required: False
1696 Inherited: False
1697 Property Type: icom_ac:Accessor
1698 Cardinality: Multi
1699 Updatability: Read Write

1700
1701 The Role class MAY include additional property definitions which are implementation-defined.
1702



1703
1704 Figure 15 Role Definition and Role Class Diagram
1705

1706 **3.5.5 Privilege**

1707 **3.5.5.1 Description**

1708 A privilege is an access right granted through roles.

1709 **3.5.5.2 Class Definition**

1710 The Privilege class is a mixin class which defines access rights that can be included in role definitions.

1711 The Privilege class is defined by the attribute values:
1712
1713 **localNamespace**
1714 Value: icom_ac
1715
1716 **localName**
1717 Value: Privilege
1718
1719 **extendsFrom**
1720 Value:
1721
1722 **stereotype**
1723 Value: mixin
1724
1725 **description**
1726 Value: Privilege is a mixin class which defines access rights that can be included in role
1727 definitions.
1728
1729 **propertyDefinitions**
1730 The values for this attribute are defined in Section 3.5.5.3.

1731 3.5.5.3 Property Definitions

1732 The Privilege class MAY include additional property definitions which are implementation-defined.
1733

1734 3.5.6 PrivilegeEnum

1735 The PrivilegeEnum class is an enum class that enumerates the instances each of which expresses a
1736 privilege that can be assigned to a role.

1737 The PrivilegeEnum class is defined by the attribute values:

1738
1739 **localNamespace**
1740 Value: icom_ac
1741
1742 **localName**
1743 Value: PrivilegeEnum
1744
1745 **extendsFrom**
1746 Value: icom_ac:Privilege
1747
1748 **stereotype**
1749 Value: primary
1750
1751 **isEnumeration**

1752 Value: TRUE
1753
1754 **description**
1755 Value: An enumeration of instances each of which expresses a privilege that can be assigned to
1756 a role.
1757
1758 **instances**
1759 Value: <icom_ac:Archive, icom_ac:Audit>
1760
1761 The following privileges are defined by ICOM:
1762 • **icom_ac:Archive** to express a right to archive contents in a scope.
1763 • **icom_ac:Audit** to express a right to audit activities in a scope.
1764

1765 3.5.7 AccessControlList

1766 3.5.7.1 Description

1767 An access control list (ACL) is an object attached to an entity to specify a list of permissions to access the
1768 entity.

1769 3.5.7.2 Class Definition

1770 The AccessControlList class is defined by the attribute values:

1771
1772 **localNamespace**
1773 Value: icom_ac
1774
1775 **localName**
1776 Value: AccessControlList
1777
1778 **extendsFrom**
1779 Value:
1780
1781 **stereotype**
1782 Value: primary
1783
1784 **description**
1785 Value: An access control list (ACL) is an object attached to an entity to specify a list of
1786 permissions to access the entity.
1787
1788 **propertyDefinitions**
1789 The values for this attribute are defined in Section 3.5.7.3.

1790 3.5.7.3 Property Definitions

1791 The AccessControlList class MUST have the property definitions:

1792

1793 **icom_ac:object**

1794 Description: Associated object.

1795 Required: True

1796 Inherited: False

1797 Property Type: icom:Entity

1798 Cardinality: Single

1799 Updatability: On Create

1800

1801 **icom_ac:accessControlEntry**

1802 Description: One or more access control entries.

1803 Required: True

1804 Inherited: False

1805 Property Type: icom_ac:AccessControlEntry

1806 Cardinality: Multi

1807 Updatability: Read Write

1808

1809 AccessControllist class MAY include additional property definitions which are implementation-defined.

1810

1811 **3.5.8 AccessControlEntry**

1812 **3.5.8.1 Description**

1813 An access control entry specifies access types granted to or denied for an accessor.

1814 **3.5.8.2 Class Definition**

1815 The AccessControlEntry class is defined by the attribute values:

1816

1817 **localNamespace**

1818 Value: icom_ac

1819

1820 **localName**

1821 Value: AccessControlEntry

1822

1823 **extendsFrom**

1824 Value:

1825

1826 **stereotype**

1827 Value: primary

1828

1829 **description**

1830 Value: An access control entry is associated with an accessor and contains a list of access

1831 types (permissions) granted to or denied from the accessor.

propertyDefinitions

The values for this attribute are defined in Section 3.5.8.3.

3.5.8.3 Property Definitions

The AccessControlEntry class MUST have the property definitions:

icom_ac:subject

Description:	Associated subject.
Required:	True
Inherited:	False
Property Type:	icom_ac:Accessor
Cardinality:	Single
Updatability:	On Create

icom_ac:grant

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

icom_ac:deny

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

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

3.5.9 AccessType

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

3.5.9.1 Class Definition

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

The AccessType class is defined by the attribute values:

localNamespace

1873 Value: icom_ac
1874
1875 **localName**
1876 Value: AccessType
1877
1878 **extendsFrom**
1879 Value:
1880
1881 **stereotype**
1882 Value: mixin
1883
1884 **description**
1885 Value: AccessType is a mixin class which defines access rights that can be granted or denied in
1886 an access control entry.
1887
1888 **propertyDefinitions**
1889 The values for this attribute are defined in Section 3.5.9.2.

1890 **3.5.9.2 Property Definitions**

1891 The AccessType class inherits property definitions from super classes.
1892 The AccessType class MAY include additional property definitions which are implementation-defined.
1893

1894 **3.5.10 AccessTypeEnum**

1895 The AccessTypeEnum class is an enum class that enumerates the instances each of which expresses an
1896 access type that can be granted or denied in an access control entry.
1897 The AccessTypeEnum class is defined by the attribute values:

1898
1899 **localNamespace**
1900 Value: icom_ac
1901
1902 **localName**
1903 Value: AccessTypeEnum
1904
1905 **extendsFrom**
1906 Value: icom_ac:AccessType
1907
1908 **stereotype**
1909 Value: primary
1910
1911 **isEnumeration**
1912 Value: TRUE
1913

description

Value: An enumeration of instances each of which expresses an access type that can be granted or denied in an access control entry.

instances

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

The following access types are defined by ICOM:

- **icom_ac:Read** to express a right to retrieve an entity.
- **icom_ac:Write** to express a right to update an entity.
- **icom_ac>Delete** to express a right to delete an entity.

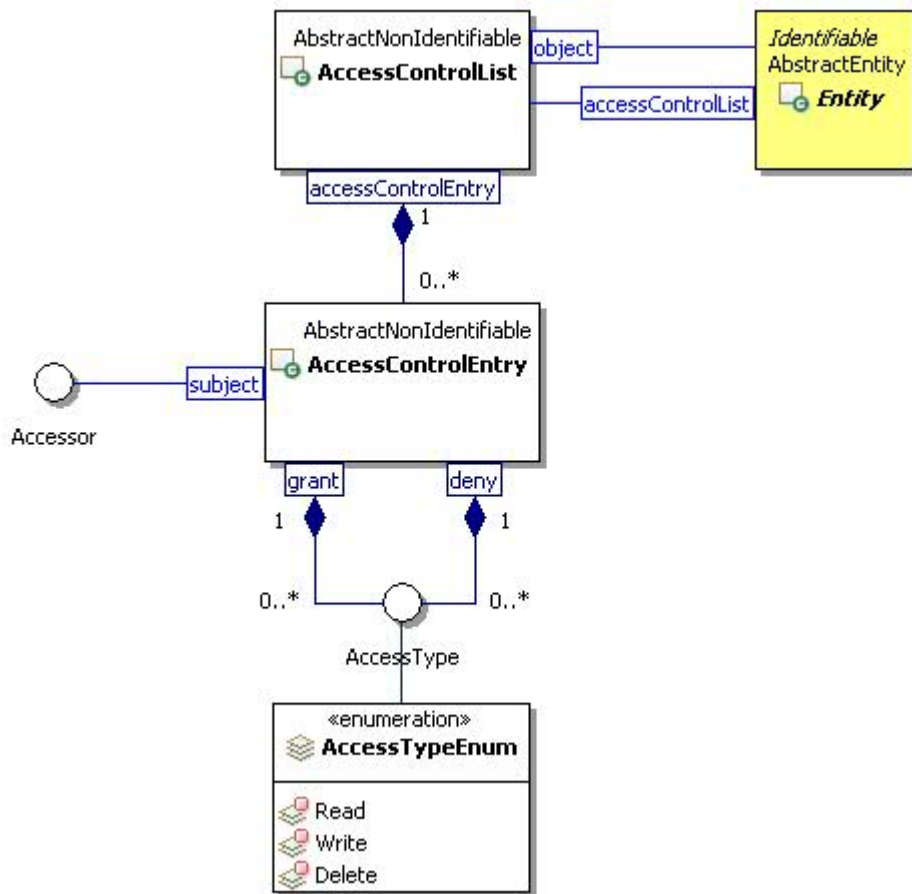


Figure 16 Access Control List Class Diagram

3.6 Metadata Model

3.6.1 PropertyDefinition

3.6.1.1 Description

A property definition specifies the name, type, choice, and cardinality of values for properties. A property type includes string, boolean, decimal, integer, datetime, etc.

3.6.1.2 Class Definition

The PropertyDefinition class is defined by the attribute values:

localNamespace

Value: icom_meta

localName

Value: PropertyDefinition

extendsFrom

Value: icom:Identifiable

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.6.1.3.

3.6.1.3 Property Definitions

The PropertyDefinition class inherits property definitions from super classes.

The PropertyDefinition class MUST have the property definitions:

icom:name

Description: Name for a property.

Required: True

Inherited: False

Property Type: String

Cardinality: Single

Updatability: Read Write

icom:description

Description: A description of a property definition.

1969	Required:	False
1970	Inherited:	False
1971	Property Type:	String
1972	Cardinality:	Single
1973	Updatability:	Read Write
1974		
1975	icom_meta:propertyType	
1976	Description:	Type of a property.
1977	Required:	True
1978	Inherited:	False
1979	Property Type:	icom_meta:PropertyType
1980	Cardinality:	Single
1981	Updatability:	On Create
1982	Choices:	PropertyChoiceType
1983	Open Choice:	False
1984		
1985	icom_meta:defaultValue	
1986	Description:	A default value for a property.
1987	Required:	False
1988	Inherited:	False
1989	Property Type:	property-type
1990	Cardinality:	Single
1991	Updatability:	Read Write
1992		
1993	icom_meta:choices	
1994	Description:	An explicit ordered set of single values allowed for a property.
1995	Required:	False
1996	Inherited:	False
1997	Property Type:	List< icom_meta:PropertyChoiceType>
1998	Cardinality:	Single
1999	Updatability:	Read Write
2000		
2001	icom_meta:cardinality	
2002	Description:	Cardinality of a property specifying whether the property can have “zero or one” or “zero or more” values.
2003		
2004	Required:	True
2005	Inherited:	False
2006	Property Type:	icom_meta:Cardinality
2007	Cardinality:	Single
2008	Updatability:	On Create
2009		
2010	icom_meta:minValue	
2011	Description:	Minimum value for an integer or decimal property.

2012	Required:	False
2013	Inherited:	False
2014	Property Type:	Integer Decimal
2015	Cardinality:	Single
2016	Updatability:	Read Write
2017		
2018	icom_meta:maxValue	
2019	Description:	Maximum value for an integer or decimal property.
2020	Required:	False
2021	Inherited:	False
2022	Property Type:	Integer Decimal
2023	Cardinality:	Single
2024	Updatability:	Read Write
2025		
2026	The PropertyDefinition class MAY include additional property definitions which are implementation-	
2027	defined.	
2028		

2029 3.6.2 Property

2030 3.6.2.1 Description

2031 The property holds a property value.

2032 3.6.2.2 Class Definition

2033 The Property class is defined by the attribute values:

2034	
2035	localNamespace
2036	Value: icom_meta
2037	
2038	localName
2039	Value: Property
2040	
2041	extendsFrom
2042	Value:
2043	
2044	stereotype
2045	Value: primary
2046	
2047	description
2048	Value: A property value.
2049	
2050	propertyDefinitions
2051	The values for this attribute are defined in Section 3.6.2.3.

3.6.2.3 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.

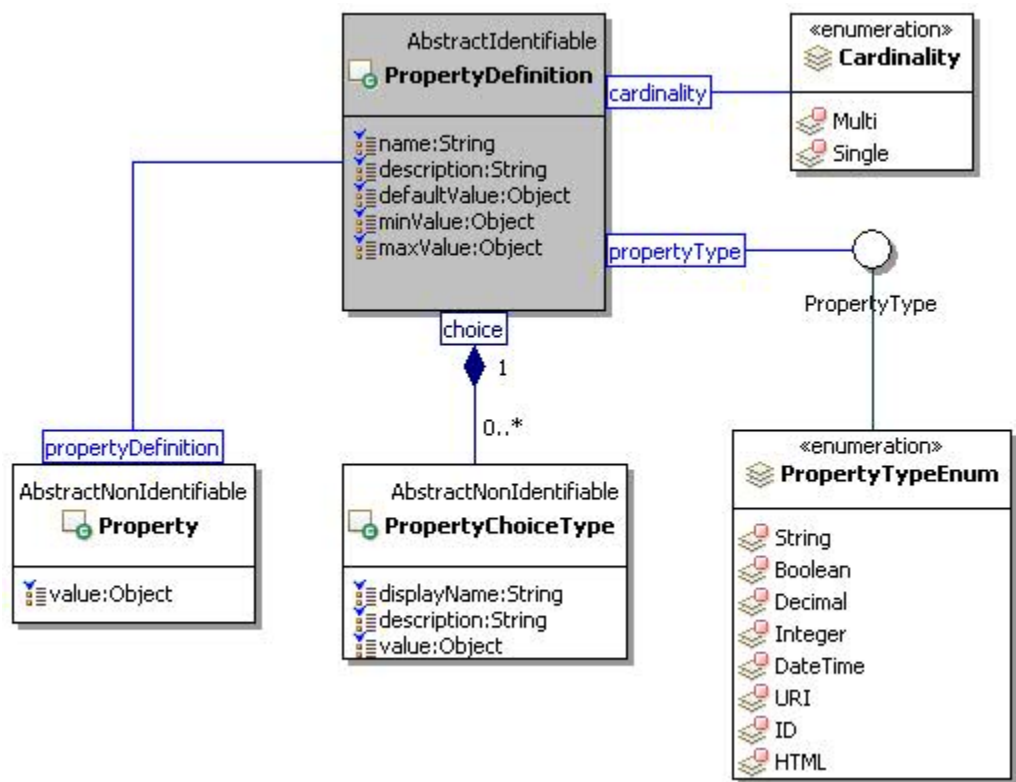


Figure 17 Property Definition and Property Class Diagram

2076

2077 **3.6.3 PropertyChoiceType**

2078 **3.6.3.1 Description**

2079 The property choice type represents a value choice for a property. Each choice includes a display name
2080 to be used for presentation purpose and a value to be stored in a property when a choice is selected.

2081 **3.6.3.2 Class Definition**

2082 The PropertyChoiceType class is defined by the attribute values:

2083

2084 **localNamespace**

2085 Value: icom_meta

2086

2087 **localName**

2088 Value: PropertyChoiceType

2089

2090 **extendsFrom**

2091 Value:

2092

2093 **stereotype**

2094 Value: primary

2095

2096 **description**

2097 Value: A choice for a property value.

2098

2099 **propertyDefinitions**

2100 The values for this attribute are defined Section 3.6.3.3.

2101 **3.6.3.3 Property Definitions**

2102 The PropertyChoiceType class MUST have the property definitions:

2103

2104 **icom_meta:displayName**

2105 Description: Display name of a property choice.

2106 Required: True

2107 Inherited: False

2108 Property Type: String

2109 Cardinality: Single

2110 Updatability: Read Write

2111

2112 **icom:description**

2113 Description: A description of a property choice.

2114 Required: False

2115 Inherited: False

2116	Property Type:	String
2117	Cardinality:	Single
2118	Updatability:	Read Write

2119

2120 **icom_meta:value**

2121	Description:	A value of a property choice.
2122	Required:	True
2123	Inherited:	False
2124	Property Type:	property-type
2125	Cardinality:	Single
2126	Updatability:	Read Write

2127

2128 The PropertyChoiceType class MAY include additional property definitions which are implementation-
2129 defined.

2130

2131 3.6.4 PropertyType

2132 A PropertyType expresses a name of a **property-type**.

2133 3.6.4.1 Class Definition

2134 The PropertyType class is a mixin class which expresses a name of a **property-type**.

2135 The PropertyType class is defined by the attribute values:

2136

2137 **localNamespace**

2138 Value: icom_meta

2139

2140 **localName**

2141 Value: PropertyType

2142

2143 **extendsFrom**

2144 Value:

2145

2146 **stereotype**

2147 Value: mixin

2148

2149 **description**

2150 Value: PropertyType is a mixin class which expresses a name of a **property-type**.

2151

2152 **propertyDefinitions**

2153 The values for this attribute are defined in Section 3.6.4.2.

2154 3.6.4.2 Property Definitions

2155 The PropertyType class MAY include additional property definitions which are implementation-defined.

2156

2157 3.6.5 PropertyTypeEnum

2158 The PropertyTypeEnum class is an enum class that enumerates the instances each of which expresses
2159 the name of a **property-type**.

2160

2161 **localNamespace**

2162 Value: icom_meta

2163

2164 **localName**

2165 Value: PropertyTypeEnum

2166

2167 **extendsFrom**

2168 Value: PropertyType

2169

2170 **stereotype**

2171 Value: primary

2172

2173 **isEnumeration**

2174 Value: TRUE

2175

2176 **description**

2177 Value: An enumeration of instances each of which expresses the name of a basic data type.

2178

2179 **instances**

2180 Value: <icom_meta:String, icom_meta:Boolean, icom_meta:Decimal, icom_meta:Integer,
2181 icom_meta:Datetime, icom_meta:URI, icom_meta:ID, icom_meta:HTML>

2182

2183 The following names of data types are defined by ICOM:

- 2184 • **icom_meta:String** to express **xsd:string**.
- 2185 • **icom_meta:Boolean** to express **xsd:boolean**.
- 2186 • **icom_meta:Decimal** to express **xsd:decimal**.
- 2187 • **icom_meta:Integer** to express **xsd:integer**.
- 2188 • **icom_meta:Datetime** to express **xsd:dateTime**.
- 2189 • **icom_meta:URI** to express **xsd:anyURI**.
- 2190 • **icom_meta:ID** to express opaque object identifiers.
- 2191 • **icom_meta:HTML** to express documents or fragments of Hypertext Markup Language (HTML)
2192 content

2193

2194 Note: ICOM uses several basic data types defined by “XML Schema Part 2: Datatypes Second Edition”
2195 (W3C Recommendation, 28 October 2004, <http://www.w3.org/TR/xmlschema-2/>).

2196

3.6.6 Cardinality

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

The Cardinality is defined by the attribute values:

localNamespace

Value: icom_meta

localName

Value: Cardinality

extendsFrom

Value:

stereotype

Value: primary

isEnumeration

Value: TRUE

description

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

instances

Value: <icom_meta:Single, icom_meta:Multi>

The following cardinality types are defined by ICOM:

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

3.6.7 UML Diagram of Marker and Subclasses

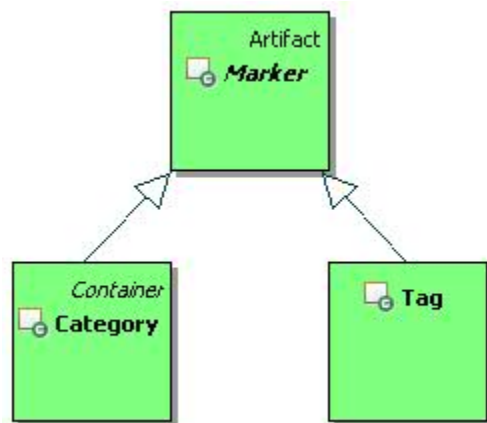


Figure 18 Marker Branch

3.6.8 Marker

3.6.8.1 Description

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

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

3.6.8.2 Class Definition

The Marker class is defined by the attribute values:

localNamespace

Value: icom-meta

localName

Value: Marker

extendsFrom

Value: icom: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 3.6.8.3.

3.6.8.3 Property Definitions

The Marker class inherits property definitions from super classes.

The Marker class MUST have the property definitions:

icom_meta:markedEntity

Description: A marked entity.

Required: False

Inherited: False

Property Type: icom:Entity

Cardinality: Multi

Updatability: Read Only

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

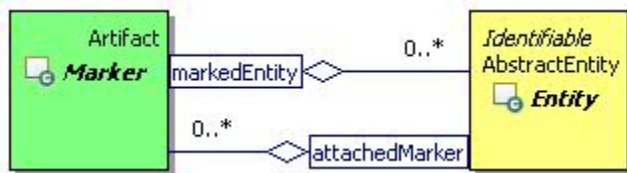


Figure 19 Marker Class Diagram

3.6.9 Category

3.6.9.1 Description

A category is a marker that classifies entities by taxonomy.

3.6.9.2 Class Definition

The Category class is defined by the attribute values:

localNamespace

Value: icom_meta

localName

Value: Category

extendsFrom

Value: icom_meta:Marker

stereotype

2296 Value: primary
 2297
 2298 **description**
 2299 Value: A category is a marker that classifies entities by taxonomy.
 2300
 2301 **propertyDefinitions**
 2302 The values for this attribute are defined in Section 3.6.9.3.

2303 3.6.9.3 Property Definitions

2304 The Category class inherits property definitions from super classes.
 2305 The Category class MUST have the property definitions:
 2306

2307 **icom_meta:superCategory**

2308	Description:	A super category.
2309	Required:	False
2310	Inherited:	False
2311	Property Type:	icom_meta:Category
2312	Cardinality:	Single
2313	Updatability:	Read Only

2314

2315 **icom_meta:subcategory**

2316	Description:	Zero or more sub categories.
2317	Required:	False
2318	Inherited:	False
2319	Property Type:	icom_meta:Category
2320	Cardinality:	Multi
2321	Updatability:	Read Only

2322

2323 **icom_meta:isAbstract**

2324	Description:	Indicates whether a category is abstract or concrete.
2325	Required:	False
2326	Inherited:	False
2327	Property Type:	Boolean
2328	Cardinality:	Single
2329	Updatability:	Read Write

2330

2331 **icom_meta:propertyDefinition**

2332	Description:	Optional or mandatory properties for a category application.
2333	Required:	False
2334	Inherited:	False
2335	Property Type:	icom_meta:PropertyDefinition
2336	Cardinality:	Multi
2337	Updatability:	Read Write

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

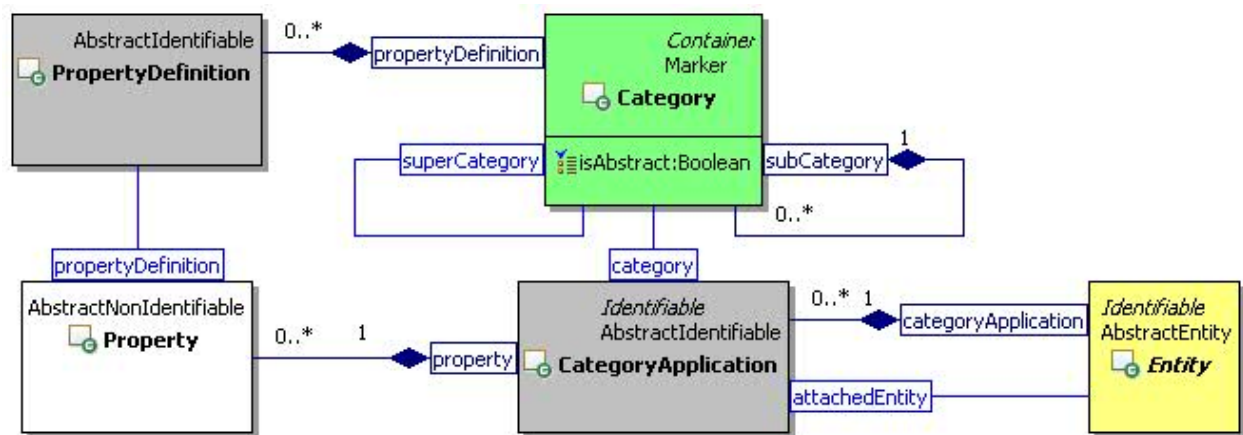


Figure 20 Category and Category Application Class Diagram

3.6.10 CategoryApplication

3.6.10.1 Description

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

3.6.10.2 Class Definition

The CategoryApplication class is defined by the attribute values:

localNamespace

Value: icom_meta

localName

Value: CategoryApplication

extendsFrom

Value: icom: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 3.6.10.3.

3.6.10.3 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 on which a category is applied.
Required:	True
Inherited:	False
Property Type:	icom: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.

3.6.11 Tag

3.6.11.1 Description

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

3.6.11.2 Class Definition

The Tag class is defined by the attribute values:

localNamespace

Value: icom_meta

2408 **localName**
2409 Value: Tag
2410
2411 **extendsFrom**
2412 Value: icom_meta:Marker
2413
2414 **stereotype**
2415 Value: primary
2416
2417 **description**
2418 Value: A tag is a marker that labels entities by a keyword.
2419
2420 **propertyDefinitions**
2421 The values for this attribute are defined in Section 3.6.11.3.

2422 **3.6.11.3 Property Definitions**

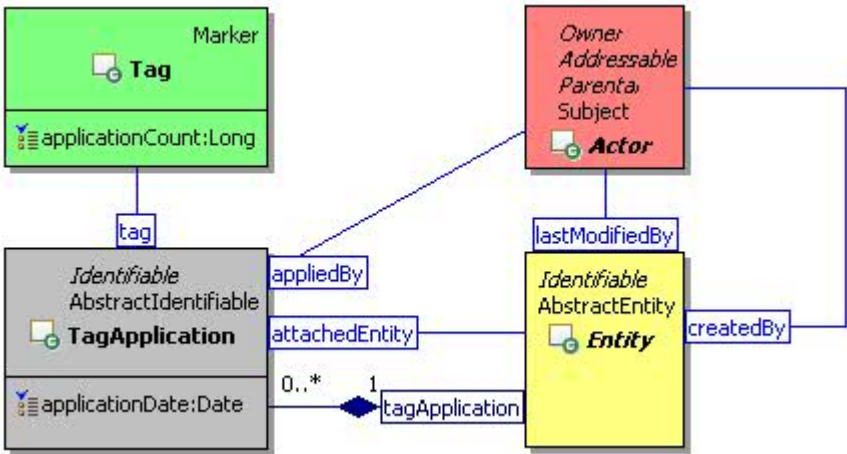
2423 The Tag class inherits property definitions from super classes.
2424 The Tag class **MUST** have the property definitions:

2425

2426	icom_meta:applicationCount	
2427	Description:	The number of times a tag is applied on entities.
2428	Required:	False
2429	Inherited:	False
2430	Property Type:	Integer
2431	Cardinality:	Single
2432	Updatability:	Read Only

2433

2434 The Tag class **MAY** include additional property definitions which are implementation-defined.
2435



2436
2437 Figure 21 Tag and Tag Application Class Diagram
2438

2439 3.6.12 TagApplication

2440 3.6.12.1 Description

2441 A tag application is an instance of association between a tag and a specific entity.

2442 3.6.12.2 Class Definition

2443 The TagApplication class is defined by the attribute values:

2444

2445 **localNamespace**

2446 Value: icom_meta

2447

2448 **localName**

2449 Value: TagApplication

2450

2451 **extendsFrom**

2452 Value: icom:Identifiable

2453

2454 **stereotype**

2455 Value: primary

2456

2457 **description**

2458 Value: A tag application is an instance of association between a tag and a specific entity.

2459

2460 **propertyDefinitions**

2461 The values for this attribute are defined in Section 3.6.12.3.

2462 3.6.12.3 Property Definitions

2463 The TagApplication class inherits property definitions from super classes.

2464 The TagApplication class MUST have the property definitions:

2465

2466 **icom_meta:attachedEntity**

2467 Description: An entity on which a tag is applied.

2468 Required: True

2469 Inherited: False

2470 Property Type: icom:Entity

2471 Cardinality: Single

2472 Updatability: On Create

2473

2474 **icom_meta:tag**

2475 Description: A tag which is applied on an entity.

2476 Required: True

2477 Inherited: False

2478 Property Type: icom_meta:Tag

2479	Cardinality:	Single
2480	Updatability:	On Create
2481		
2482	icom_meta:appliedBy	
2483	Description:	A user who applies a tag on an entity.
2484	Required:	False
2485	Inherited:	False
2486	Property Type:	icom:Actor
2487	Cardinality:	Single
2488	Updatability:	Read Only

2490	icom_meta:applicationDate	
2491	Description:	A date and time when a tag is applied on an entity.
2492	Required:	False
2493	Inherited:	False
2494	Property Type:	DateTime
2495	Cardinality:	Single
2496	Updatability:	Read Write

2497
2498 The TagApplication class MAY include additional property definitions which are implementation-defined.
2499

2500 **3.6.13 RelationshipBondable**

2501 **3.6.13.1 Description**

2502 A relationship bondable entity is an entity which may be relationship bonded.
2503 Note: a relationship cannot be relationship bonded by other relationships, i.e. relationships are entities
2504 that are not relationship bondable.

2505 **3.6.13.2 Class Definition**

2506 The RelationshipBondable class is a mixin class which defines the characteristics of entities that may be
2507 relationship bonded. It includes almost every subclass of Entity except Relationship.

2508 The RelationshipBondable class is defined by the attribute values:

2509		
2510	localNamespace	
2511	Value:	icom_meta
2512		
2513	localName	
2514	Value:	RelationshipBondable
2515		
2516	extendsFrom	
2517	Value:	icom:Identifiable
2518		
2519	stereotype	

2520 Value: mixin
2521
2522 **description**
2523 Value: RelationshipBondable is a mixin class which defines the characteristics of entities that
2524 can be relationship bonded.
2525
2526 **propertyDefinitions**
2527 The values for this attribute are defined in Section 3.6.13.3.

2528 3.6.13.3 Property Definitions

2529 The RelationshipBondable class inherits property definitions from super classes.
2530 The RelationshipBondable class MAY include additional property definitions which are implementation-
2531 defined.
2532

2533 3.6.14 RelationshipDefinition

2534 3.6.14.1 Description

2535 A relationship definition is an entity that defines a type of relationship, including a name and a description
2536 of the relationship type, types of source entity and target entities of a relationship, and definition of
2537 properties in a relationship.

2538 3.6.14.2 Class Definition

2539 The RelationshipDefinition class is defined by the attribute values:

2540
2541 **localNamespace**
2542 Value: icom_meta
2543
2544 **localName**
2545 Value: RelationshipDefinition
2546
2547 **extendsFrom**
2548 Value: icom:EntityDefinition
2549
2550 **stereotype**
2551 Value: primary
2552
2553 **description**
2554 Value: A relationship definition is an entity that defines a type of relationship.
2555
2556 **propertyDefinitions**
2557 The values for this attribute are defined in Section 3.6.14.3.

2558 3.6.14.3 Property Definitions

2559 The RelationshipDefinition class inherits property definitions from super classes.

2560 The RelationshipDefinition class MUST have the property definitions:

2561

2562 **icom_meta:propertyDefinition**

2563 Description: Optional or mandatory properties for a relationship.

2564 Required: False

2565 Inherited: False

2566 Property Type: icom_meta:PropertyDefinition

2567 Cardinality: Multi

2568 Updatability: Read Write

2569

2570 **icom_meta:allowedSourceType**

2571 Description: A list of expanded names of relationship bondable classes,
2572 indicating that the source entity of a relationship MUST be an
2573 instance of a class in the list.

2574 Required: False

2575 Inherited: False

2576 Property Type: IRI

2577 Cardinality: Multi

2578 Updatability: Read Write

2579

2580 **icom_meta:allowedTargetType**

2581 Description: A list of expanded names of relationship bondable classes,
2582 indicating that the target entity of a relationship MUST be an
2583 instance of a class in the list.

2584 Required: False

2585 Inherited: False

2586 Property Type: IRI

2587 Cardinality: Multi

2588 Updatability: Read Write

2589

2590 The RelationshipDefinition class MAY include additional property definitions which are implementation-
2591 defined.

2592

2593 **3.6.15 Relationship**

2594 **3.6.15.1 Description**

2595 A relationship is an entity that relates a set of entities by a predicate.

2596 **3.6.15.2 Class Definition**

2597 The Relationship class is defined by the attribute values:

2598

2599 **localNamespace**

2600 Value: icom_meta

2601

2602 **localName**

2603 Value: Relationship

2604

2605 **extendsFrom**

2606 Value: icom:Entity

2607

2608 **stereotype**

2609 Value: primary

2610

2611 **description**

2612 Value: A relationship is an entity that relates a set of entities by a predicate.

2613

2614 **propertyDefinitions**

2615 The values for this attribute are defined in Section 3.6.15.3.

2616 **3.6.15.3 Property Definitions**

2617 The Relationship class inherits property definitions from super classes.

2618 The Relationship class **MUST** have the property definitions:

2619

2620 **icom_meta:relationshipDefinition**

2621 Description: A definition of relationships.

2622 Required: True

2623 Inherited: False

2624 Property Type: icom_meta:RelationshipDefinition

2625 Cardinality: Single

2626 Updatability: On Create

2627

2628 **icom_meta:sourceEntity**

2629 Description: A source entity of a relationship.

2630 Required: True

2631 Inherited: False

2632 Property Type: icom_meta:RelationshipBondable

2633 Cardinality: Single

2634 Updatability: On Create

2635

2636 **icom_meta:targetEntity**

2637 Description: One or more target entities of a relationship.

2638 Required: True

2639 Inherited: False

2640 Property Type: icom_meta:RelationshipBondable

2641 Cardinality: Multi

2642 Updatability: Read Write

icom_meta:property

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

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

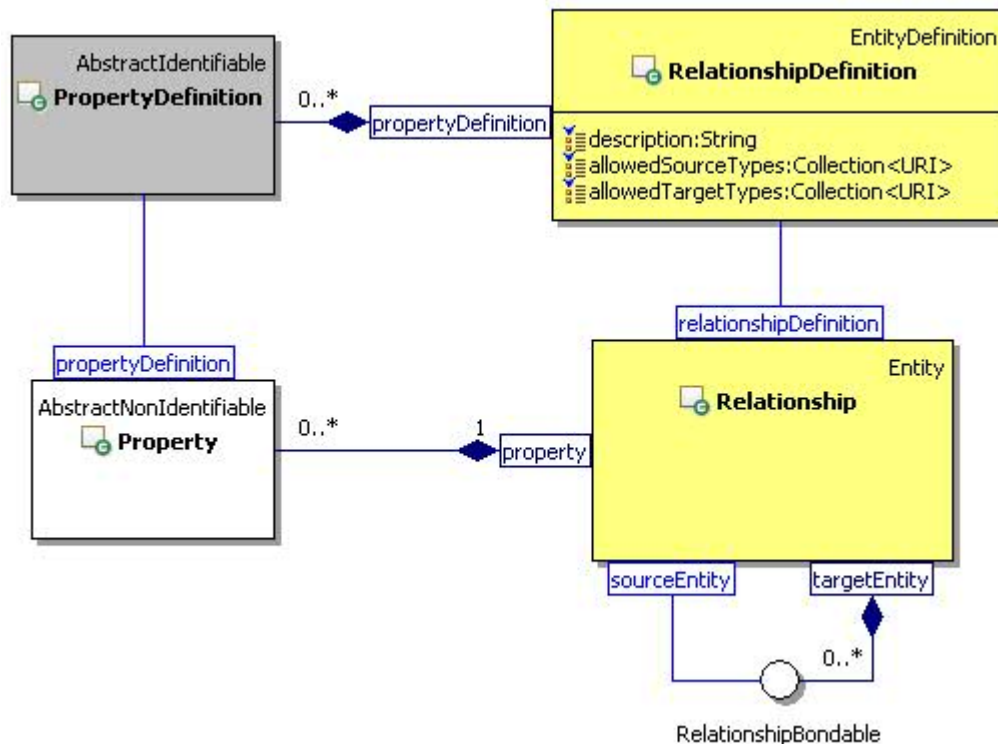


Figure 22 Relationship Class Diagram

3.7 Common Concepts

3.7.1 Participant

3.7.1.1 Description

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

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

3.7.1.2 Class Definition

The Participant class is defined by the attribute values:

2665

2666 **localNamespace**

2667 Value: icom

2668

2669 **localName**

2670 Value: Participant

2671

2672 **extendsFrom**

2673 Value:

2674

2675 **stereotype**

2676 Value: primary

2677

2678 **description**

2679 Value: A participant object represents the participation of any addressable entity in a

2680 collaboration activity such as an occurrence, task, conference, discussion, and message.

2681

2682 **propertyDefinitions**

2683 The values for this attribute are defined in Section 3.7.1.3.

2684 **3.7.1.3 Property Definitions**

2685 The Participant class inherits property definitions from super classes.

2686 The Participant class **MUST** have the property definitions:

2687

2688 **icom:participant**

2689	Description:	An addressable entity to participate in a collaboration activity.
2690	Required:	False
2691	Inherited:	False
2692	Property Type:	icom_card:Addressable
2693	Cardinality:	Single
2694	Updatability:	On Create

2695

2696 **icom:address**

2697	Description:	An address of a participant in a collaboration activity.
2698	Required:	False
2699	Inherited:	False
2700	Property Type:	URI
2701	Cardinality:	Single
2702	Updatability:	On Create

2703

2704 **icom:name**

2705	Description:	Name of a participant in a collaboration activity.
2706	Required:	False

2707 Inherited: False
2708 Property Type: String
2709 Cardinality: Single
2710 Updatability: On Create

2711

2712 The Participant class MAY include additional property definitions which are implementation-defined.

2713

2714 **3.7.2 Priority**

2715 The Priority class is an enum class that enumerates the instances each of which expresses a precedence
2716 ordering.

2717 The Priority is defined by the attribute values:

2718

2719 **localNamespace**

2720 Value: icom

2721

2722 **localName**

2723 Value: Priority

2724

2725 **extendsFrom**

2726 Value:

2727

2728 **stereotype**

2729 Value: primary

2730

2731 **isEnumeration**

2732 Value: TRUE

2733

2734 **description**

2735 Value: An enumeration of the instances each of expresses a precedence ordering.

2736

2737 **instances**

2738 Value: <icom:None, icom:Low, icom:Medium, icom:High>

2739

2740 The following priorities are defined by ICOM:

- 2741 • **icom:None** to express a normal priority.
- 2742 • **icom:Low** to express a low priority.
- 2743 • **icom:Medium** to express a medium priority.
- 2744 • **icom:High** to express a high priority.

2745

3.7.3 DateTimeResolution

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

The DateTimeResolution is defined by the attribute values:

localNamespace

Value: icom

localName

Value: DateTimeResolution

extendsFrom

Value:

stereotype

Value: primary

isEnumeration

Value: TRUE

description

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

instances

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

The following three date time resolutions are defined by ICOM:

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

3.7.4 Location

3.7.4.1 Description

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

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

3.7.4.2 Class Definition

The Location class is defined by the attribute values:

2787 **localNamespace**
 2788 Value: icom
 2789
 2790 **localName**
 2791 Value: Location
 2792
 2793 **extendsFrom**
 2794 Value:
 2795
 2796 **stereotype**
 2797 Value: primary
 2798
 2799 **description**
 2800 Value: A location object represents a physical location which is defined by name, description, or
 2801 geo coordinates.
 2802
 2803 **propertyDefinitions**
 2804 The values for this attribute are defined in Section 3.7.4.3.

2805 **3.7.4.3 Property Definitions**

2806 The Location class inherits property definitions from super classes.

2807 The Location class MUST have the property definitions:

2808
 2809 **icom:name**
 2810 Description: Name of a location.
 2811 Required: False
 2812 Inherited: False
 2813 Property Type: String
 2814 Cardinality: Single
 2815 Updatability: Read Write
 2816
 2817 **icom:description**
 2818 Description: A description of a location.
 2819 Required: False
 2820 Inherited: False
 2821 Property Type: String
 2822 Cardinality: Single
 2823 Updatability: Read Write
 2824
 2825 **icom:timeZone**
 2826 Description: Time zone of a location.
 2827 Required: False
 2828 Inherited: False

2829	Property Type:	TimeZone
2830	Cardinality:	Single
2831	Updatability:	Read Write
2832		
2833	icom:locationMark	
2834	Description:	A list of geo coordinates marking a point, path, or area of a physical location.
2835		
2836	Required:	False
2837	Inherited:	False
2838	Property Type:	icom:GeoCoordinates
2839	Cardinality:	Multi
2840	Updatability:	Read Write

2841

2842 The Location class MAY include additional property definitions which are implementation-defined.

2843

2844 3.7.5 GeoCoordinates

2845 3.7.5.1 Description

2846 A geo coordinates object specifies the latitude, longitude, and altitude of a physical location.

2847 3.7.5.2 Class Definition

2848 The GeoCoordinates class is defined by the attribute values:

2849

2850 **localNamespace**

2851 Value: icom

2852

2853 **localName**

2854 Value: GeoCoordinates

2855

2856 **extendsFrom**

2857 Value:

2858

2859 **stereotype**

2860 Value: primary

2861

2862 **description**

2863 Value: A geo coordinates object specifies the latitude, longitude, and altitude of a physical location.

2864

2865

2866 **propertyDefinitions**

2867 The values for this attribute are defined in Section 3.7.5.3.

3.7.5.3 Property Definitions

The GeoCoordinates class MUST have the property definitions:

icom:latitude

Description:	Latitude of coordinates.
Required:	False
Inherited:	False
Property Type:	Float
Cardinality:	Single
Updatability:	Read Write

icom:longitude

Description:	Longitude of coordinates.
Required:	False
Inherited:	False
Property Type:	Float
Cardinality:	Single
Updatability:	Read Write

icom:altitude

Description:	Altitude of coordinates.
Required:	False
Inherited:	False
Property Type:	Float
Cardinality:	Single
Updatability:	Read Write

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

4 Extension Modules

4.1 Content Module

4.1.1 Content

4.1.1.1 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.

4.1.1.2 Class Definition

The Content class is defined by the attribute values:

localNamespace

Value: icom_content

localName

Value: Content

extendsFrom

Value: icom:Identifiable, icom_msg: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 4.1.1.3.

4.1.1.3 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

2934	Inherited:	False
2935	Property Type:	String
2936	Cardinality:	Single
2937	Updatability:	Read Write
2938		
2939	icom_content:mediaType	
2940	Description:	Media type is a two-part identifier for Internet file formats as defined in RFC 2046 and additional RFCs including RFC 3236, RFC 1847, etc.
2941		
2942		
2943	Required:	
2944	Inherited:	False
2945	Property Type:	String
2946	Cardinality:	Single
2947	Updatability:	Read Write
2948		
2949	icom_content:contentDisposition	
2950	Description:	Content disposition is defined in RFC 2183 to specify a presentation style.
2951		
2952	Required:	False
2953	Inherited:	False
2954	Property Type:	ContentDispositionType
2955	Cardinality:	Single
2956	Updatability:	Read Write
2957		
2958	The Content class MAY include additional property definitions which are implementation-defined.	
2959		

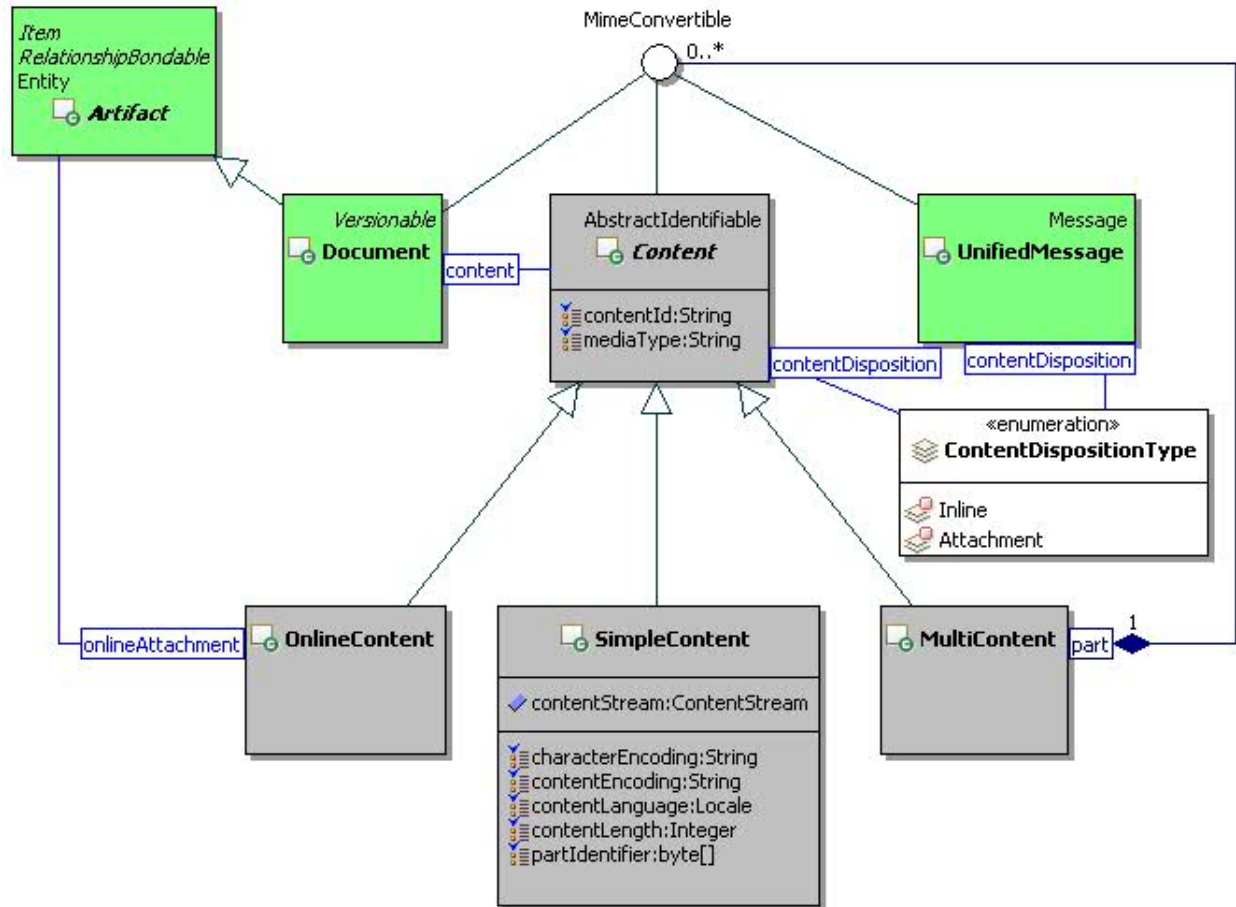


Figure 23 Composite Content Class Diagram

4.1.2 MultiContent

4.1.2.1 Description

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

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

4.1.2.2 Class Definition

The MultiContent class is defined by the attribute values:

localNamespace

Value: icom content

localName

Value: MultiContent

extendsFrom

Value: icom content:Content

2979
2980 **stereotype**
2981 Value: primary
2982
2983 **description**
2984 Value: A multi-content object represents the multiple parts of a message or document.
2985
2986 **propertyDefinitions**
2987 The values for this attribute are defined in Section 4.1.2.3.

2988 **4.1.2.3 Property Definitions**

2989 The MultiContent class inherits property definitions from super classes.
2990 The MultiContent class MUST have the property definitions:

2991
2992 **icom_content:part**
2993 Description: Zero or more parts of a hierarchical composite content.
2994 Required: False
2995 Inherited: False
2996 Property Type: icom_msg:MimeConvertible
2997 Cardinality: Multi
2998 Updatability: Read Write
2999

3000 The MultiContent class MAY include additional property definitions which are implementation-defined.
3001

3002 **4.1.3 SimpleContent**

3003 **4.1.3.1 Description**

3004 A simple content holds a single piece of data.
3005 Note: A media type is an official RFC 2046 type. Content encoding specifies RFC 2616 content encoding
3006 applied to a content. Character encoding specifies RFC 2616 character set of a content (a missing value
3007 means that a content should be treated as binary or raw). Content language specifies RFC 2616 content
3008 language for a content (a missing value means non-natural language content).

3009 **4.1.3.2 Class Definition**

3010 The SimpleContent class is defined by the attribute values:

3011
3012 **localNamespace**
3013 Value: icom_content
3014
3015 **localName**
3016 Value: SimpleContent
3017
3018 **extendsFrom**

3019 Value: icom_content:Content

3020

3021 **stereotype**

3022 Value: primary

3023

3024 **description**

3025 Value: A simple content holds a single piece of data.

3026

3027 **propertyDefinitions**

3028 The values for this attribute are defined in Section 4.1.3.3.

3029 **4.1.3.3 Property Definitions**

3030 The SimpleContent class inherits property definitions from super classes.

3031 The SimpleContent class MUST have the property definitions:

3032

3033 **icom_content:characterEncoding**

3034 Description: Character encoding specifies RFC 2616 character set of a
3035 content (a missing value means that a content should be
3036 treated as binary or raw).

3037 Required: False

3038 Inherited: False

3039 Property Type: String

3040 Cardinality: Single

3041 Updatability: Read Write

3042

3043 **icom_content:contentEncoding**

3044 Description: Content encoding specifies RFC 2616 content encoding
3045 applied to a content.

3046 Required: False

3047 Inherited: False

3048 Property Type: String

3049 Cardinality: Single

3050 Updatability: Read Write

3051

3052 **icom_content:contentLangauge**

3053 Description: Content language specifies RFC 2616 content language for a
3054 content (a missing value means non-natural language
3055 content).

3056 Required: False

3057 Inherited: False

3058 Property Type: Locale

3059 Cardinality: Single

3060 Updatability: Read Write

3061

3062	icom_content:contentLength	
3063	Description:	Length of a content.
3064	Required:	False
3065	Inherited:	False
3066	Property Type:	Integer
3067	Cardinality:	Single
3068	Updatability:	Read Write

3069

3070 The SimpleContent class MAY include additional property definitions which are implementation-defined.

3071

3072 4.1.4 OnlineContent

3073 4.1.4.1 Description

3074 An online content holds an online artifact attached to a document, message, or invitation.

3075 Note: An online artifact must be rendered as a URL when a message or invitation is delivered to external

3076 recipients.

3077 4.1.4.2 Class Definition

3078 The OnlineContent class is defined by the attribute values:

3079

3080	localNamespace	
3081	Value:	icom_content
3082		
3083	localName	
3084	Value:	OnlineContent
3085		
3086	extendsFrom	
3087	Value:	icom_content:Content
3088		
3089	stereotype	
3090	Value:	primary
3091		
3092	description	
3093	Value:	An online content holds an online artifact attached to a message or invitation.
3094		
3095	propertyDefinitions	
3096	The values for this attribute are defined in	Section 4.1.4.3.

3097 4.1.4.3 Property Definitions

3098 The OnlineContent class inherits property definitions from super classes.

3099 The OnlineContent class MUST have the property definitions:

3100

3101 **icom_content:onlineAttachment**

3102	Description:	An online artifact attached to a message.
3103	Required:	True
3104	Inherited:	False
3105	Property Type:	Artifact
3106	Cardinality:	Single
3107	Updatability:	Read Write

3108
3109 The OnlineContent class MAY include additional property definitions which are implementation-defined.
3110

3111 4.1.5 ContentDispositionType

3112 The ContentDispositionType class is an enum class that enumerates the instances each of which
3113 expresses a presentation style of content.

3114 Note: The enumerated instances for this class are content disposition types defined in RFC 2183.

3115 The ContentDispositionType class is defined by the attribute values:

3116
3117 **localNamespace**
3118 Value: icom_content
3119
3120 **localName**
3121 Value: ContentDispositionType
3122
3123 **extendsFrom**
3124 Value:
3125
3126 **stereotype**
3127 Value: primary
3128
3129 **isEnumeration**
3130 Value: TRUE
3131
3132 **description**
3133 Value: An enumeration of instances each of which expresses a presentation style of content
3134 defined in RFC 2183.
3135
3136 **instances**
3137 Value: <icom_content:Inline, icom_content:Attachment>
3138

3139 There are two content disposition types defined by ICOM:

- 3140 • **icom_content:Inline** to express that content is to be displayed automatically upon display of the
- 3141 main body of an artifact.
- 3142 • **icom_content:Attachment** to express that content is separate from the main body of an artifact,
- 3143 and that its display should not be automatic, but contingent upon some further action of a user.

3144

3145 **4.1.6 Attachment**

3146 **4.1.6.1 Description**

3147 An attachment holds a simple content for an occurrence, task, and contact artifact.

3148 **4.1.6.2 Class Definition**

3149 The Attachment class is defined by the attribute values:

3150

3151 **localNamespace**

3152 Value: icom_content

3153

3154 **localName**

3155 Value: Attachment

3156

3157 **extendsFrom**

3158 Value:

3159

3160 **stereotype**

3161 Value: primary

3162

3163 **description**

3164 Value: An attachment holds a simple content for an occurrence, task, and contact artifact.

3165

3166 **propertyDefinitions**

3167 The values for this attribute are defined in Section 4.1.6.3.

3168 **4.1.6.3 Property Definitions**

3169 The Attachment class MUST have the property definitions:

3170

3171 **icom:name**

3172 Description: Name of a simple content attachment.

3173 Required: True

3174 Inherited: False

3175 Property Type: String

3176 Cardinality: Single

3177 Updatability: Read Write

3178

3179 **icom_content:content**

3180 Description: A simple content attached to an occurrence, task, or contact artifact.

3181

3182 Required: True

3183 Inherited: False

3184 Property Type: SimpleContent

3185 Cardinality: Single
3186 Updatability: Read Write

3187

3188 The Attachment class MAY include additional property definitions which are implementation-defined.

3189

3190 **4.2 Document Module**

3191 **4.2.1 Versionable**

3192 **4.2.1.1 Description**

3193 A versionable artifact is

- 3194 1. a representative copy,
3195 2. a specific versioned copy, or
3196 3. a private working copy

3197 of an artifact version series.

3198 When a versionable artifact is not under version control, a representative copy of a versionable artifact is
3199 the only version of a version series and represents the versionable artifact itself, i.e. there is only one
3200 *objectId* so far.

3201 When a versionable artifact is under version control:

- 3202 • a representative copy of a versionable artifact is a versionable artifact which has its own object
3203 identifier that is different from the object identifier of any versioned copy or private working copy
3204 of the versionable artifact. It retains the object identifier it has when the artifact is created. Its
3205 version type changes from RepresentativeCopy to ViewOnlyRepresentativeCopy.
- 3206 • a representative copy of a versionable artifact provides a view of a version series, depending on
3207 the check out state of the version series and the user loading the artifact. If the current user
3208 loading a representative copy is the same user who checks out from a version series, the
3209 representative copy is a copy of the content and state of a private working copy. Otherwise, the
3210 representative copy is a copy of the content and state of the latest versioned copy in a version
3211 series.

3212 A specific versioned copy of a versionable artifact is an explicit "deep" copy of the content and state of a
3213 versionable artifact, preserving its content and state at a certain point in time. Each versioned copy of a
3214 versionable artifact is itself a versionable artifact, i.e. it has its own *objectId*. A versioned copy has a
3215 version number, label, and check in comment.

3216 A private working copy of a versionable artifact is a versionable artifact created by an explicit checkout
3217 operation on a versionable artifact under version control. The properties for a private working copy
3218 SHOULD be identical to the properties of a versioned copy of a versionable artifact on which a checkout
3219 operation was performed. Certain properties such as *objectId* and *creationDate* SHALL be different from
3220 a versioned copy. The content of a private working copy MAY be identical to the content of a versioned
3221 copy. Its object identifier must be different from that of the representative copy or any versioned copy. A
3222 private working copy can be saved to the version series for sharing and co-editing, however, it needs not
3223 be visible to users who may only have permissions to view other versioned copies in a version series.
3224 Until it is checked in using an explicit checkin operation, a private working copy MUST NOT be
3225 considered the LatestMajorVersion in a version series.

3226 Note: A container of a versionable artifact can contain a representative copy of a version series so that it
3227 provides an artifact view of the latest state of the version series. Starting from a representative copy in a
3228 container, an actor can traverse a version series to retrieve any versioned copy or private working copy.

3229 ICOM version control model is based on the CMIS version control model specified in Section 2.1.9 of
3230 Content Management Interoperability Services Version 1.0 [CMIS].

4.2.1.2 Class Definition

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

The Versionable class is defined by the attribute values:

localNamespace

Value: icom_doc

localName

Value: Versionable

extendsFrom

Value: icom:Identifiable

stereotype

Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.2.1.3.

4.2.1.3 Property Definitions

The Versionable class inherits property definitions from super classes.

The Versionable class MUST have the property definitions:

icom_doc:versionControlMetadata

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

Required: False

Inherited: False

Property Type: icom_doc:VersionControlMetadata

Cardinality: Single

Updatability: Read Only

icom_doc:versionType

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

Required: False

Inherited: False

Property Type: icom_doc:VersionType

Cardinality: Single

Updatability: Read Only

3273
3274 The Versionable class MAY include additional property definitions which are implementation-defined.
3275

3276 4.2.2 VersionControlMetadata

3277 4.2.2.1 Description

3278 A version control metadata is an object that contains version control information.
3279 There are two classes of version control metadata: version series and version. A version control metadata
3280 of a versionable artifact is either a version series or a version depending on the version type.
3281

- If the version type is icom_doc:RepresentativeCopy or icom_doc:ViewOnlyRepresentativeCopy,
3282 then metadata MUST be a version series object.
- If the version type is icom_doc:VersionedCopy or icom_doc:PrivateWorkingCopy, then metadata
3283 MUST be a version object.
3284

3285 4.2.2.2 Class Definition

3286 The VersionControlMetadata class is a mixin class that defines the characteristics of entities that serve as
3287 metadata for version control.

3288 The VersionControlMetadata class is defined by the attribute values:

3289
3290 **localNamespace**
3291 Value: icom_doc
3292
3293 **localName**
3294 Value: VersionControlMetadata
3295
3296 **extendsFrom**
3297 Value: icom:Identifiable
3298
3299 **stereotype**
3300 Value: mixin
3301
3302 **description**
3303 Value: VersionControlMetadata is a mixin class that defines the characteristics of entities that
3304 serve as metadata for version control.
3305
3306 **propertyDefinitions**
3307 The values for this attribute are defined in Section 4.2.2.3.

3308 4.2.2.3 Property Definitions

3309 The VersionControlMetadata class inherits property definitions from super classes.

3310 The VersionControlMetadata class MUST have the property definitions:

3311
3312 **icom_doc:representativeCopy**
3313 Description: A representative copy of a versionable artifact.

3314	Required:	False
3315	Inherited:	False
3316	Property Type:	icom_doc:Versionable
3317	Cardinality:	Single
3318	Updatability:	Read Only

3319

3320 The VersionControlMetadata class MAY include additional property definitions which are implementation-
3321 defined.

3322

3323 4.2.3 VersionSeries

3324 4.2.3.1 Description

3325 A version series is a version control metadata that contains a version history and check in/out states of a
3326 versionable artifact.

3327 A version series object is a version control metadata of a representative copy of a versionable artifact.

3328 4.2.3.2 Class Definition

3329 The VersionSeries class is defined by the attribute values:

3330

3331 **localNamespace**

3332 Value: icom_doc

3333

3334 **localName**

3335 Value: VersionSeries

3336

3337 **extendsFrom**

3338 Value: icom:Entity, icom_doc:VersionControlMetadata, icom_meta:RelationshipBondable

3339

3340 **stereotype**

3341 Value: primary

3342

3343 **description**

3344 Value: A version series is a version control metadata that contains a version history and check
3345 in/out states of a versionable artifact.

3346

3347 **propertyDefinitions**

3348 The values for this attribute are defined in Section 4.2.3.3.

3349 4.2.3.3 Property Definitions

3350 The VersionSeries class inherits property definitions from super classes.

3351 The VersionSeries class MUST have the property definitions:

3352

3353 **icom_doc:versionHistory**

3354	Description:	A history of version nodes of a versionable artifact.
3355	Required:	False
3356	Inherited:	False
3357	Property Type:	icom_doc:Version
3358	Cardinality:	Multi
3359	Updatability:	Read Only
3360		
3361	icom_doc:versionableHistory	
3362	Description:	A history of the versioned copies of a versionable artifact.
3363	Required:	False
3364	Inherited:	False
3365	Property Type:	icom_doc:Versionable
3366	Cardinality:	Multi
3367	Updatability:	Read Only
3368		
3369	icom_doc:latestVersionedCopy	
3370	Description:	Latest versioned copy of a versionable artifact.
3371	Required:	False
3372	Inherited:	False
3373	Property Type:	icom_doc:Versionable
3374	Cardinality:	Single
3375	Updatability:	Read Only
3376		
3377	icom_doc:privateWorkingCopy	
3378	Description:	A private working copy of a versionable artifact.
3379	Required:	False
3380	Inherited:	False
3381	Property Type:	icom_doc:Versionable
3382	Cardinality:	Single
3383	Updatability:	Read Only
3384		
3385	icom_doc:versionSeriesCheckedOut	
3386	Description:	Indicates whether a version series is checked out.
3387	Required:	False
3388	Inherited:	False
3389	Property Type:	Boolean
3390	Cardinality:	Single
3391	Updatability:	Read Only
3392		
3393	icom_doc:versionSeriesCheckedOutBy	
3394	Description:	An actor who checks out a version series.
3395	Required:	False

3396 Inherited: False
3397 Property Type: icom:Actor
3398 Cardinality: Single
3399 Updatability: Read Only

3400

3401 **icom_doc:versionSeriesCheckedOutOn**

3402 Description: The time when a version series is checked out.
3403 Required: False
3404 Inherited: False
3405 Property Type: DateTime
3406 Cardinality: Single
3407 Updatability: Read Only

3408

3409 **icom_doc:versionSeriesCheckoutComment**

3410 Description: A checked out comment of a version series.
3411 Required: False
3412 Inherited: False
3413 Property Type: String
3414 Cardinality: Single
3415 Updatability: Read Only

3416

3417 **icom_doc:totalSize**

3418 Description: Total size of all versioned copies of a versionable artifact in a
3419 version series.
3420 Required: False
3421 Inherited: False
3422 Property Type: Integer
3423 Cardinality: Single
3424 Updatability: Read Only

3425

3426 The VersionSeries class MAY include additional property definitions which are implementation-defined.

3427

3428 **4.2.4 Version**

3429 **4.2.4.1 Description**

3430 A version is a version control metadata that contains a version number, label, and description.
3431 A version object is a version control metadata of a versioned copy or a private working copy of a
3432 versionable artifact.

3433 **4.2.4.2 Class Definition**

3434 The Version class is defined by the attribute values:

3435

3436 **localNamespace**

3437 Value: icom_doc
 3438
 3439 **localName**
 3440 Value: Version
 3441
 3442 **extendsFrom**
 3443 Value: icom:Entity, icom_doc:VersionControlMetadata, icom_meta:RelationshipBondable
 3444
 3445 **stereotype**
 3446 Value: primary
 3447
 3448 **description**
 3449 Value: A version is a version control metadata that contains a version number, label, and
 3450 description.
 3451
 3452 **propertyDefinitions**
 3453 The values for this attribute are defined in Section 4.2.4.3.

3454 4.2.4.3 Property Definitions

3455 The Version class inherits property definitions from super classes.

3456 The Version class MUST have the property definitions;

3457

3458 **icom_doc:checkinComment**

3459	Description:	A check in comment of a versioned copy.
3460	Required:	False
3461	Inherited:	False
3462	Property Type:	String
3463	Cardinality:	Single
3464	Updatability:	Read Write

3465

3466 **icom_doc:versionNumber**

3467	Description:	A version number of a versioned copy.
3468	Required:	True
3469	Inherited:	False
3470	Property Type:	Integer
3471	Cardinality:	Single
3472	Updatability:	Read Write

3473

3474 **icom_doc:versionLabel**

3475	Description:	A version label of a versioned copy.
3476	Required:	True
3477	Inherited:	False
3478	Property Type:	String

3479	Cardinality:	Single
3480	Updatability:	Read Write
3481		
3482	icom_doc:majorVersion	
3483	Description:	Indicates whether a versioned copy is a major version.
3484	Required:	True
3485	Inherited:	False
3486	Property Type:	Boolean
3487	Cardinality:	Single
3488	Updatability:	Read Write
3489		
3490	icom_doc:versionedOrPrivateWorkingCopy	
3491	Description:	A versioned copy or private working copy corresponding to a
3492		version of a versionable artifact.
3493	Required:	False
3494	Inherited:	False
3495	Property Type:	icom_doc:Versionable
3496	Cardinality:	Multi
3497	Updatability:	Read Only

3498

3499 The Version class MAY include additional property definitions which are implementation-defined.

3500

3501 4.2.5 VersionType

3502 The VersionType class is an enum class that enumerates the instances each of which expresses a

3503 version type.

3504 The VersionType class is defined by the attribute values:

3505		
3506	localNamespace	
3507	Value:	icom_doc
3508		
3509	localName	
3510	Value:	VersionType
3511		
3512	extendsFrom	
3513	Value:	
3514		
3515	stereotype	
3516	Value:	primary
3517		
3518	isEnumeration	
3519	Value:	TRUE
3520		

3521 **description**
3522 Value: An enumeration of the instances each of which expresses a version type.

3523
3524 **instances**
3525 Value: <icom_doc:RepresentativeCopy, icom_doc:ViewOnlyRepresentativeCopy,
3526 icom_doc:VersionedCopy, icom_doc:PrivateWorkingCopy>

3527
3528 There are four version types defined by ICOM:

- 3529 • **icom_doc:RepresentativeCopy** to express that a versionable artifact is a representative copy of
3530 an artifact.
- 3531 • **icom_doc:ViewOnlyRepresentativeCopy** to express that a versionable artifact is a view only
3532 representative copy of an artifact version series.
- 3533 • **icom_doc:VersionedCopy** to express that a versionable artifact is a versioned copy of an
3534 artifact version series.
- 3535 • **icom_doc:PrivateWorkingCopy** to express that a versionable artifact is a private working copy
3536 of an artifact version series.

3537

3538 4.2.6 Document

3539 4.2.6.1 Description

3540 A document is a versionable artifact that can contain a single content of a media type or composite
3541 contents of any assortment of media types.

3542 4.2.6.2 Class Definition

3543 The Document class is defined by the attribute values:

3544

3545 **localNamespace**
3546 Value: icom_doc

3547

3548 **localName**
3549 Value: Document

3550

3551 **extendsFrom**
3552 Value: icom:Artifact, icom_doc:Versionable, icom_content:MimeConvertible

3553

3554 **stereotype**
3555 Value: primary

3556

3557 **description**
3558 Value: A document is a versionable artifact that can contain a single content of a media type or
3559 composite contents of any assortment of media types.

3560

3561 **propertyDefinitions**

3562 The values for this attribute are defined in Section 4.2.6.3.

4.2.6.3 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
Inherited:	False
Property Type:	Integer
Cardinality:	Single
Updatability:	Read Only

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

3600 **extendsFrom**
3601 Value: icom_doc:Document
3602
3603 **stereotype**
3604 Value: primary
3605
3606 **description**
3607 Value: A wiki page is a document that contains a rendered page and a rendered content.
3608
3609 **propertyDefinitions**
3610 The values for this attribute are defined in Section 4.2.7.3.

3611 4.2.7.3 Property Definitions

3612 The WikiPage class inherits property definitions from super classes.
3613 The WikiPage class MUST have the property definitions:

3614
3615 **icom_doc:renderedPage**
3616 Description: A page rendered from wiki page content.
3617 Required: False
3618 Inherited: False
3619 Property Type: String
3620 Cardinality: Single
3621 Updatability: Read Only
3622
3623 **icom_doc:renderedContent**
3624 Description: An object rendered from wiki page content.
3625 Required: False
3626 Inherited: False
3627 Property Type: Object
3628 Cardinality: Single
3629 Updatability: Read Only

3630
3631 The WikiPage class MAY include additional property definitions which are implementation-defined.
3632

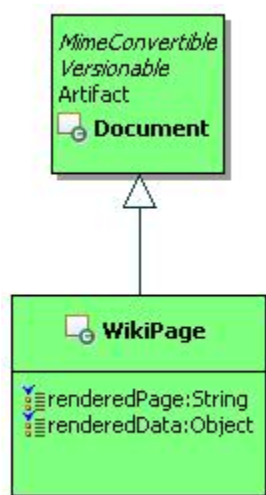


Figure 25 WikiPage Class Diagram.

4.3 Message Module

4.3.1 MimeConvertible

4.3.1.1 Description

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

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

4.3.1.2 Class Definition

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

The MimeConvertible class is defined by the attribute values:

```

localNamespace
    Value: icom_msg

localName
    Value: MimeConvertible

extendsFrom
    Value: icom:Identifiable

stereotype
    Value: mixin

description
  
```

3661 Value: MimeConvertible class is a mixin class that defines the characteristics of objects that can
3662 be represented in MIME format.

3663

3664 **propertyDefinitions**

3665 The values for this attribute are defined in Section 4.3.1.3.

3666 4.3.1.3 Property Definitions

3667 The MimeConvertible class inherits property definitions from super classes.

3668 The MimeConvertible class MAY include additional property definitions which are implementation-defined.

3669

3670 4.3.2 Message

3671 4.3.2.1 Description

3672 A message is a unit of conversation. It holds a simple content or multipart message contents in a content
3673 property. It has a single sender.

3674 Note: The delivered time is the time when a message is delivered to a given recipient. The sent time of a
3675 message is represented by a user creation date and time of the message. The name property holds the
3676 subject of a message.

3677 4.3.2.2 Class Definition

3678 The Message class is defined by the attribute values:

3679

3680 **localNamespace**

3681 Value: icom_msg

3682

3683 **localName**

3684 Value: Message

3685

3686 **extendsFrom**

3687 Value: icom:Artifact

3688

3689 **stereotype**

3690 Value: primary

3691

3692 **isAbstract**

3693 Value: TRUE

3694

3695 **description**

3696 Value: A message is a unit of conversation.

3697

3698 **propertyDefinitions**

3699 The values for this attribute are defined in Section 4.3.2.3.

4.3.2.3 Property Definitions

The Message class inherits property definitions from super classes.

The Message class MUST have the property definitions:

icom_msg:sender

Description:	Sender of a message.
Required:	False
Inherited:	False
Property Type:	icom:Participant
Cardinality:	Single
Updatability:	Read Write

icom_msg:deliveredTime

Description:	The time when a message is delivered to a given recipient.
Required:	False
Inherited:	False
Property Type:	DateTime
Cardinality:	Single
Updatability:	Read Only

icom_content:content

Description:	Content of a message
Required:	False
Inherited:	False
Property Type:	icom_content:Content
Cardinality:	Single
Updatability:	Read Write

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

4.3.3 UnifiedMessage

4.3.3.1 Description

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

A unified message can be one of these types:

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

4.3.3.2 Class Definition

The UnifiedMessage class is defined by the attribute values:

localNamespace

Value: icom_msg

localName

Value: UnifiedMessage

extendsFrom

Value: icom_msg:Message, icom_msg:MimeConvertible

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.3.3.3.

4.3.3.3 Property Definitions

The UnifiedMessage class inherits property definitions from super classes.

The UnifiedMessage class MUST have the property definitions:

icom_msg:envelopSender

Description: An envelop sender (sometimes called return path) of a message.

Required: False

Inherited: False

Property Type: icom:Participant

Cardinality: Single

Updatability: Read Write

icom_msg:toReceivers

Description: A list of participants to whom a message is sent or to be sent.

Required: False

Inherited: False

Property Type: icom:Participant

Cardinality: Multi

Updatability: Read Write

3781	icom_msg:ccReceivers	
3782	Description:	A list of participants to whom a message is carbon-copied or
3783		to be carbon-copied.
3784	Required:	False
3785	Inherited:	False
3786	Property Type:	icom:Participant
3787	Cardinality:	Multi
3788	Updatability:	Read Write
3789		
3790	icom_msg:bccReceivers	
3791	Description:	A list of participants to whom a message is blind-carbon-
3792		copied or to be blind-carbon-copied.
3793	Required:	False
3794	Inherited:	False
3795	Property Type:	icom:Participant
3796	Cardinality:	Multi
3797	Updatability:	Read Write
3798		
3799	icom_msg:replyTo	
3800	Description:	A list of participants to whom reply messages should be sent.
3801	Required:	False
3802	Inherited:	False
3803	Property Type:	icom:Participant
3804	Cardinality:	Multi
3805	Updatability:	Read Write
3806		
3807	icom_content:contentId	
3808	Description:	Content id is a unique identifier for a message part in multi-
3809		part messages.
3810	Required:	False
3811	Inherited:	False
3812	Property Type:	String
3813	Cardinality:	Single
3814	Updatability:	Read Write
3815		
3816	icom_content:mediaType	
3817	Description:	Media type is a two-part identifier for Internet file formats as
3818		defined in RFC 2046 and additional RFCs including RFC
3819		3236, RFC 1847, etc.
3820	Required:	False
3821	Inherited:	False
3822	Property Type:	String
3823	Cardinality:	Single
3824	Updatability:	Read Write

3825		
3826	icom_content:contentDisposition	
3827	Description:	Content disposition is defined in RFC 2183 to specify a
3828		presentation style.
3829	Required:	False
3830	Inherited:	False
3831	Property Type:	icom_content:ContentDispositionType
3832	Cardinality:	Single
3833	Updatability:	Read Write
3834		
3835	icom:priority	
3836	Description:	The priority of a message.
3837	Required:	False
3838	Inherited:	False
3839	Property Type:	icom:Priority
3840	Cardinality:	Single
3841	Updatability:	Read Write
3842		
3843	icom_msg:flag	
3844	Description:	Zero or more flags on a message.
3845	Required:	False
3846	Inherited:	False
3847	Property Type:	icom_msg:UnifiedMessageFlag
3848	Cardinality:	Multi
3849	Updatability:	Read Write
3850		
3851	icom_msg:messageDispositionNotificationRequested	
3852	Description:	A message disposition notification (RFC 2298) is requested
3853		for a message.
3854	Required:	False
3855	Inherited:	False
3856	Property Type:	Boolean
3857	Cardinality:	Single
3858	Updatability:	Read Write
3859		
3860	icom_msg:messageDeliveryStatusNotificationRequest	
3861	Description:	Indicates the types of delivery status notifications (RFC 1891)
3862		requested for a message. Default is icom_msg:Failure.
3863	Required:	False
3864	Inherited:	False
3865	Property Type:	UnifiedMessageDeliveryStatusNotificationRequest
3866	Cardinality:	Multi
3867	Updatability:	Read Write

3868		
3869	icom_msg:channel	
3870	Description:	Indicates the delivery channel of a message.
3871	Required:	False
3872	Inherited:	False
3873	Property Type:	icom_msg:UnifiedMessageChannel
3874	Cardinality:	Single
3875	Updatability:	Read Write
3876		
3877	icom_msg:mode	
3878	Description:	Indicates an editable mode (new, draft, or delivered) of a
3879		message.
3880	Required:	False
3881	Inherited:	False
3882	Property Type:	icom_msg:UnifiedMessageEditMode
3883	Cardinality:	Single
3884	Updatability:	Read Only
3885		
3886	icom_msg:mimeHeader	
3887	Description:	A list of headers including those defined in RFC 822 and
3888		other custom headers. Each header is represented by a multi-
3889		valued property. The name of a property is a printable header
3890		name. The value of a property is a collection of ascii or non-
3891		ascii strings.
3892	Required:	False
3893	Inherited:	False
3894	Property Type:	icom_meta:Property
3895	Cardinality:	Multi
3896	Updatability:	Read Write
3897		
3898	The UnifiedMessage class MAY include additional property definitions which are implementation-defined.	
3899		

3900 **4.3.4 UnifiedMessageFlag**

3901 The UnifiedMessageFlag class is an enum class that enumerates the instances each of which expresses
3902 a type of flag.

3903 The UnifiedMessageFlag class is defined by the attribute values:

3904		
3905	localNamespace	
3906	Value:	icom_msg
3907		
3908	localName	
3909	Value:	UnifiedMessageFlag
3910		

3911 **extendsFrom**
3912 Value:
3913
3914 **stereotype**
3915 Value: primary
3916
3917 **isEnumeration**
3918 Value: TRUE
3919
3920 **description**
3921 Value: An enumeration of the instances each of which expresses a type of flag.
3922
3923 **instances**
3924 Value: <icom_msg:Answered, icom_msg:Forwarded, icom_msg:Redirected, icom_msg:Hidden,
3925 icom_msg:MarkedForDelete, icom_msg:MarkedForFollowUp, icom_msg:MarkedForDraft,
3926 icom_msg:MessageDispositionNotificationProcessed>
3927

3928 There are eight flags defined by ICOM:

- 3929 • **icom_msg:Answered** to express that a message is answered.
- 3930 • **icom_msg:Forwarded** to express that a message is forwarded.
- 3931 • **icom_msg:Redirected** to express that a message is redirected.
- 3932 • **icom_msg:Hidden** to express that a message is hidden.
- 3933 • **icom_msg:MarkedForDelete** to express that a message is marked for delete.
- 3934 • **icom_msg:MarkedForFollowUp** to express that a message is marked for follow up.
- 3935 • **icom_msg:MarkedForDraft** to express that a message is marked for draft.
- 3936 • **icom_msg:MessageDispositionNotificationProcessed** to express that a message disposition
3937 notification (RFC 2298) is processed.

3938

3939 **4.3.5 UnifiedMessageDeliveryStatusNotificationRequest**

3940 The UnifiedMessageDeliveryStatusNotificationRequest class is an enum class that enumerates the
3941 instances each of which expresses a request for one of several types of delivery status notification
3942 defined in RFC 1891.

3943 The UnifiedMessageDeliveryStatusNotificationRequest class is defined by the attribute values:

3944
3945 **localNamespace**
3946 Value: icom_msg
3947
3948 **localName**
3949 Value: UnifiedMessageDeliveryStatusNotificationRequest
3950
3951 **extendsFrom**
3952 Value:
3953

3954 **stereotype**
3955 Value: primary
3956
3957 **isEnumeration**
3958 Value: TRUE
3959
3960 **description**
3961 Value: An enumeration of the instances each of which expresses a request for one of several
3962 types of delivery status notification.
3963
3964 **instances**
3965 Value: <icom_msg:Never, icom_msg:Success, icom_msg:Failure, icom_msg:Delay>
3966
3967 There are four delivery status notification requests defined by ICOM:

- 3968 • **icom_msg:Never** to express that a sender requests status notification not be returned to the
- 3969 sender under any condition.
- 3970 • **icom_msg:Success** to express that a sender requests a status notification for successful
- 3971 delivery of a message.
- 3972 • **icom_msg:Failure** to express that a sender requests a status notification for delivery failure of a
- 3973 message.
- 3974 • **icom_msg:Delay** to express that a sender requests a status notification when delivery of a
- 3975 message has been delayed for an unusual length of time.
- 3976

3977 **4.3.6 UnifiedMessageChannel**

3978 The UnifiedMessageChannel class is an enum class that enumerates the instances each of which
3979 expresses a type of delivery channel.

3980 The UnifiedMessageChannel class is defined by the attribute values:

3981
3982 **localNamespace**
3983 Value: icom_msg
3984
3985 **localName**
3986 Value: UnifiedMessageChannel
3987
3988 **extendsFrom**
3989 Value:
3990
3991 **stereotype**
3992 Value: primary
3993
3994 **isEnumeration**
3995 Value: TRUE
3996

3997 **description**
3998 Value: An enumeration of the instances each of which expresses a requested receipt type.

3999
4000 **instances**
4001 Value: <icom_msg:Email, icom_msg:Voice, icom_msg:Fax, icom_msg:Notification>

4002
4003 There are four channel types defined by ICOM:

- 4004 • **icom_msg:Email** to express that delivery channel is email.
- 4005 • **icom_msg:Voice** to express that delivery channel is voice.
- 4006 • **icom_msg:Fax** to express that delivery channel is fax.
- 4007 • **icom_msg:Notification** to express that delivery channel is notification.

4008

4009 **4.3.7 UnifiedMessageEditMode**

4010 The UnifiedMessageEditMode class is an enum class that enumerates the instances each of which
4011 expresses whether a message is a draft copy, delivered copy, or other.

4012 The UnifiedMessageEditMode class is defined by the attribute values:

4013
4014 **localNamespace**
4015 Value: icom_msg

4016
4017 **localName**
4018 Value: UnifiedMessageEditMode

4019
4020 **extendsFrom**
4021 Value:

4022
4023 **stereotype**
4024 Value: primary

4025
4026 **isEnumeration**
4027 Value: TRUE

4028
4029 **description**
4030 Value: An enumeration of the instances each of which expresses whether a message is a draft
4031 copy, delivered copy, or other.

4032
4033 **instances**
4034 Value: <icom_msg:DraftCopy, icom_msg:DeliveredCopy, icom_msg:Other>

4035
4036 There are three modes defined by ICOM:

- 4037 • **icom_msg:DraftCopy** to express that a message is saved as a draft.
- 4038 • **icom_msg:DeliveredCopy** to express that a message is a sent or received message.

- icom_msg:Other to express that a message is other than draft or delivered.

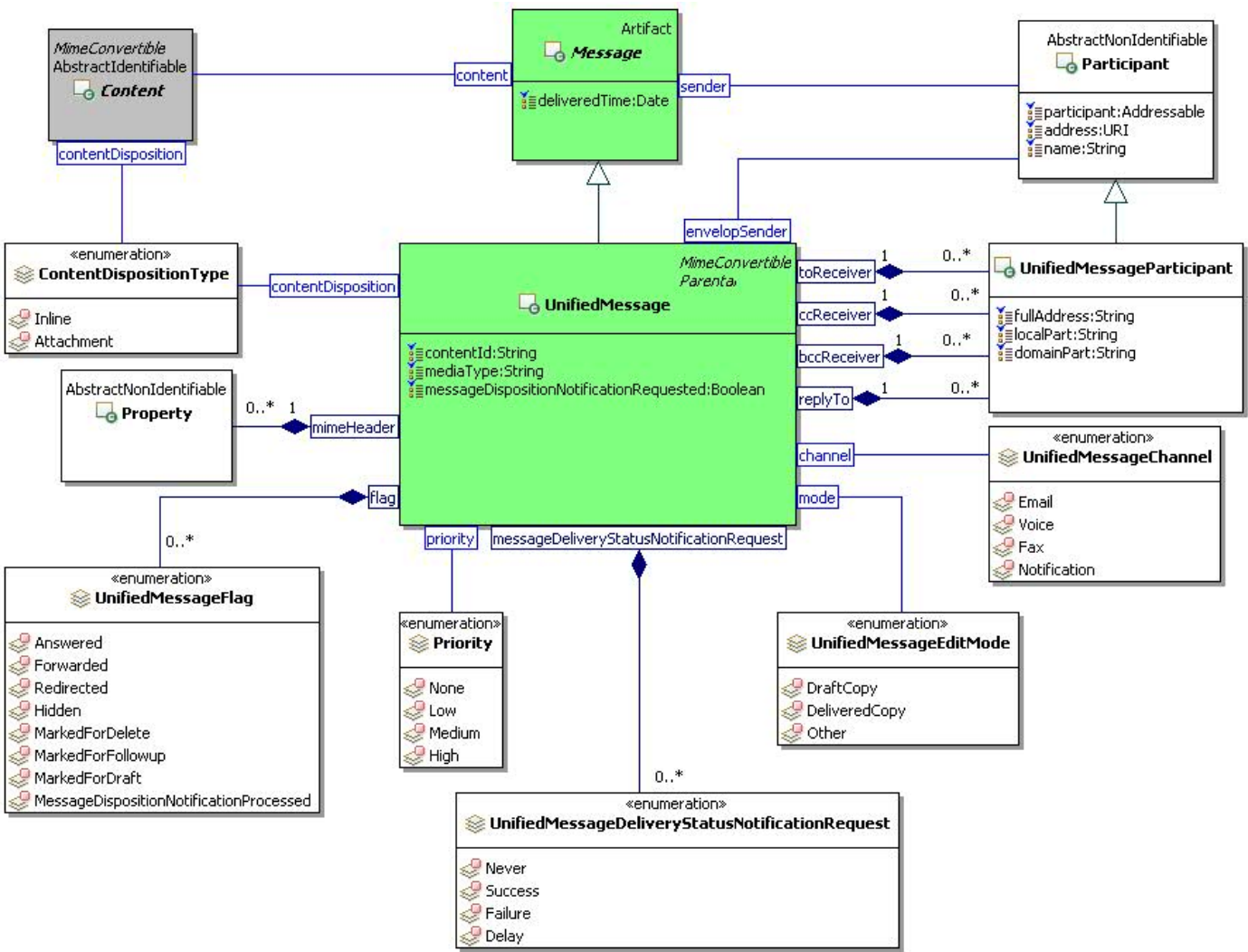


Figure 26 Unified Message Class Diagram

4.3.8 InstantMessage

4.3.8.1 Description

An instant message is a special type of message for one-on-one, synchronous, usually text based, conversation.

4.3.8.2 Class Definition

The InstantMessage class is defined by the attribute values:

```
localNamespace
  Value: icom_msg
```


4054 **localName**
4055 Value: InstantMessage
4056
4057 **extendsFrom**
4058 Value: icom_msg:Message
4059
4060 **stereotype**
4061 Value: primary
4062
4063 **isAbstract**
4064 Value: TRUE
4065
4066 **description**
4067 Value: An instant message is a special type of message for one-on-one, synchronous, usually
4068 text based, conversation.
4069
4070 **propertyDefinitions**
4071 The values for this attribute are defined in Section 4.3.8.3.

4072 **4.3.8.3 Property Definitions**

4073 The InstantMessage class inherits property definitions from super classes.

4074 The InstantMessage class MUST have the property definitions:

4075
4076 **icom_msg:toReceivers**
4077 Description: A list of participants to whom a message is sent or to be sent.
4078 Required: False
4079 Inherited: False
4080 Property Type: icom:Participant
4081 Cardinality: Multi
4082 Updatability: Read Write

4083
4084 **icom_msg:conversationId**
4085 Description: An identifier of a conversation involving one or more instant
4086 messages.
4087 Required: False
4088 Inherited: False
4089 Property Type: Integer
4090 Cardinality: Single
4091 Updatability: Read Write

4092
4093 **icom_msg:clientId**
4094 Description: An identifier of a client.
4095 Required: False

4096	Inherited:	False
4097	Property Type:	String
4098	Cardinality:	Single
4099	Updatability:	Read Write
4100		
4101	icom_msg:formattingStyle	
4102	Description:	A formatting style of rich text message in xhtml.
4103	Required:	False
4104	Inherited:	False
4105	Property Type:	String
4106	Cardinality:	Single
4107	Updatability:	Read Write
4108		
4109	icom_msg:instantMessageType	
4110	Description:	A type of instant message.
4111	Required:	False
4112	Inherited:	False
4113	Property Type:	icom_msg:InstantMessageType
4114	Cardinality:	Single
4115	Updatability:	Read Write
4116		
4117	icom_msg:chatStatus	
4118	Description:	A chat status of a user.
4119	Required:	False
4120	Inherited:	False
4121	Property Type:	icom_msg:InstantMessageChatStatus
4122	Cardinality:	Single
4123	Updatability:	Read Write
4124		

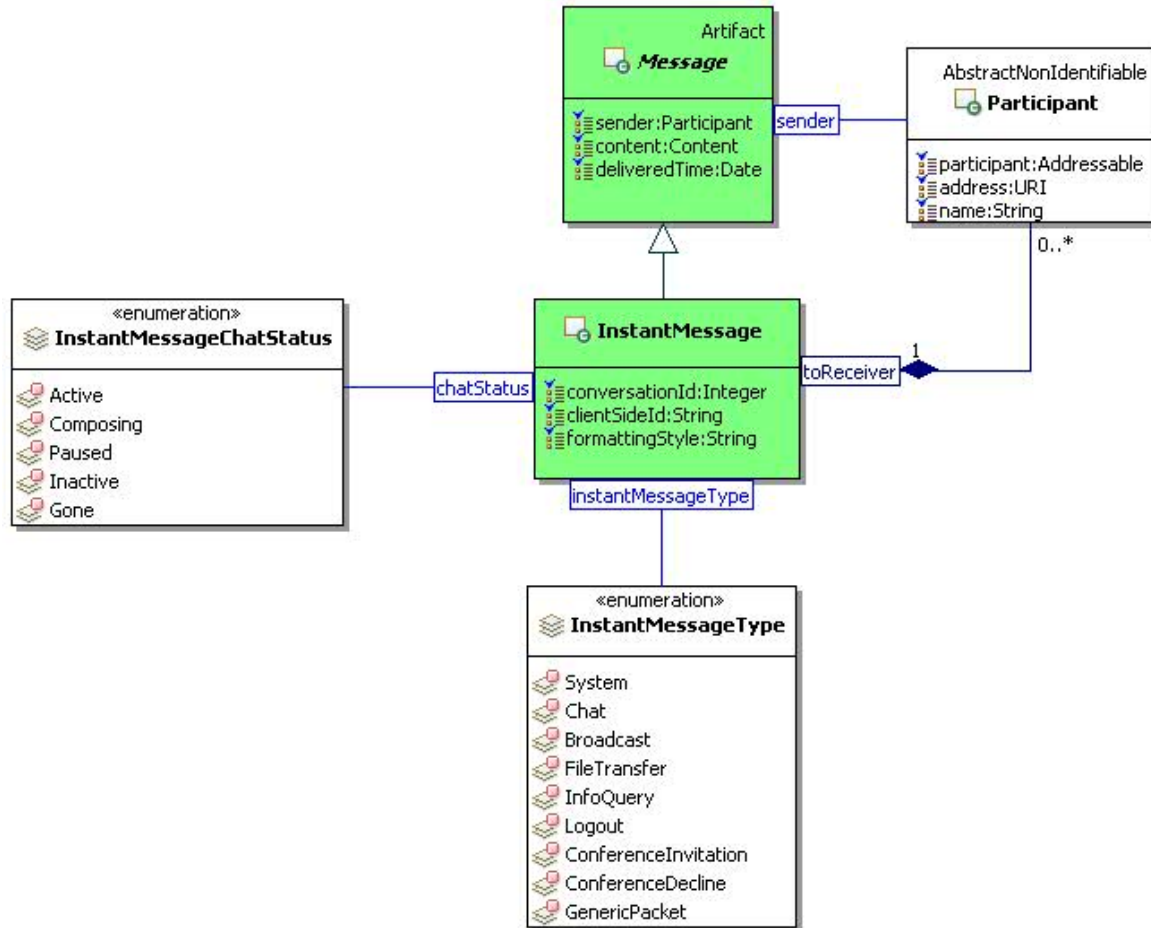


Figure 27 InstantMessage Class Diagram.

4.3.9 InstantMessageType

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

The InstantMessageType class is defined by the attribute values:

```

localNamespace
    Value: icom_msg

localName
    Value: InstantMessageType

extendsFrom
    Value:

stereotype
    Value: primary
  
```

4145 **isEnumeration**
4146 Value: TRUE
4147
4148 **description**
4149 Value: An enumeration of the instances each of which expresses a type of instant message.
4150
4151 **instances**
4152 Value: <icom_msg:System, icom_msg:Chat, icom_msg:Broadcast, icom_msg:FileTransfer,
4153 icom_msg:InfoQuery, icom_msg:Logout, icom_msg:ConferenceInvitation, icom_
4154 msg:ConferenceDecline, icom_msg:GenericPacket>
4155

4156 There are nine instant message types defined by ICOM:

- 4157 • **icom_msg:System** to express that an instant message is a system message.
- 4158 • **icom_msg:Chat** to express that an instant message is a chat message.
- 4159 • **icom_msg:Broadcast** to express that an instant message is a broadcast message.
- 4160 • **icom_msg:FileTransfer**: to express that an instant message is a file transfer message.
- 4161 • **icom_msg:InfoQuery** to express that an instant message is a info query message.
- 4162 • **icom_msg:Logout** to express that an instant message is a logout message.
- 4163 • **icom_msg:ConferenceInvitation** to express that an instant message is a conference invitation
4164 message.
- 4165 • **icom_msg:ConferenceDecline** to express that an instant message is a decline message to a
4166 conference invitation.
- 4167 • **icom_msg:GenericGone** to express that an instant message is a generic message indicating
4168 that a user is gone.

4169

4170 **4.3.10 InstantMessageChatStatus**

4171 The InstantMessageChatStatus class is an enum class that enumerates the instances each of which
4172 expresses a chat status of a user.

4173 The InstantMessageChatStatus class is defined by the attribute values:

4174
4175 **localNamespace**
4176 Value: icom_msg
4177
4178 **localName**
4179 Value: InstantMessageChatStatus
4180
4181 **extendsFrom**
4182 Value:
4183
4184 **stereotype**
4185 Value: primary
4186
4187 **isEnumeration**

4188 Value: TRUE
4189
4190 **description**
4191 Value: An enumeration of the instances each of which expresses a chat status of a user.
4192
4193 **instances**
4194 Value: <icom_msg:Active, icom_msg:Composing, icom_msg:Paused, icom_msg:Inactive, icom_
4195 msg:Gone>
4196

4197 There are five chat status defined by ICOM:

- 4198 • **icom_msg:Active** to express that a user is active.
 - 4199 • **icom_msg:Composing** to express that a user is composing a message.
 - 4200 • **icom_msg:Paused** to express that a user has paused.
 - 4201 • **icom_msg:Inactive**: to express that a user is inactive.
 - 4202 • **icom_msg:Gone** to express that a user is gone.
- 4203

4204 4.3.11 InstantMessageFeed

4205 4.3.11.1 Description

4206 An instant message feed contains a set of instant message connections.

4207 4.3.11.2 Class Definition

4208 The InstantMessageFeed class is defined by the attribute values:

4209
4210 **localNamespace**
4211 Value: icom_msg
4212
4213 **localName**
4214 Value: InstantMessageFeed
4215
4216 **extendsFrom**
4217 Value: icom:Entity
4218
4219 **stereotype**
4220 Value: primary
4221
4222 **description**
4223 Value: An instant message feed contains a set of instant message connections.
4224
4225 **propertyDefinitions**
4226 The values for this attribute are defined in Section 4.3.11.3.

4227 **4.3.11.3 Property Definitions**

4228 The InstantMessageFeed class inherits property definitions from super classes.

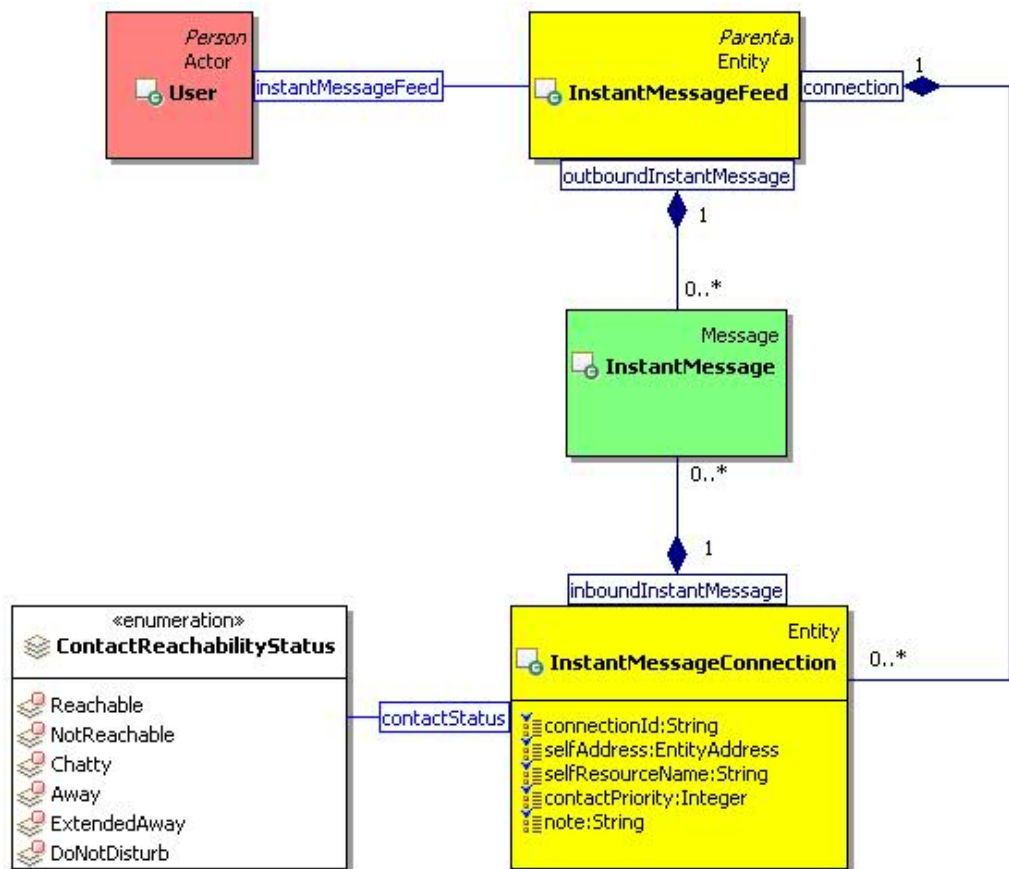
4229 The InstantMessageFeed class MUST have the property definitions:

4231 **icom_msg:connection**

4232	Description:	One or more instant messaging connections.
4233	Required:	False
4234	Inherited:	False
4235	Property Type:	icom_msg:InstantMessageConnection
4236	Cardinality:	Multi
4237	Updatability:	Read Only

4239 **icom_msg:outboundInstantMessage**

4240	Description:	A queue for outbound instant messages.
4241	Required:	False
4242	Inherited:	False
4243	Property Type:	icom_msg:InstantMessage
4244	Cardinality:	Multi
4245	Updatability:	Write Only



4247 **Figure 28 Instant Message Feed and Connection Class Diagram.**

4249

4250 **4.3.12 InstantMessageConnection**

4251 **4.3.12.1 Description**

4252 An instant message connection contains the queues for inbound and outbound instant messages.

4253 A presentity can update the contact status, contact priority, and note for a contact method associated with
4254 a connection.

4255 **4.3.12.2 Class Definition**

4256 The InstantMessageConnection class is defined by the attribute values:

4257

4258 **localNamespace**

4259 Value: icom_msg

4260

4261 **localName**

4262 Value: InstantMessageConnection

4263

4264 **extendsFrom**

4265 Value: icom:Entity

4266

4267 **stereotype**

4268 Value: primary

4269

4270 **description**

4271 Value: An instant message connection contains the queues for inbound and outbound instant
4272 messages.

4273

4274 **propertyDefinitions**

4275 The values for this attribute are defined in Section 4.3.12.3.

4276 **4.3.12.3 Property Definitions**

4277 The InstantMessageConnection class inherits property definitions from super classes.

4278 The InstantMessageConnection class MUST have the property definitions:

4279

4280 **icom_msg:connectionId**

4281 Description: An identifier of a connection.

4282 Required: False

4283 Inherited: False

4284 Property Type: String

4285 Cardinality: Single

4286 Updatability: Read Only

4287

4288 **icom_msg:selfAddress**

4289	Description:	Address of a presentity who opens a connection.
4290	Required:	True
4291	Inherited:	False
4292	Property Type:	URI
4293	Cardinality:	Single
4294	Updatability:	On Create
4295		
4296	icom_msg:selfResourceName	
4297	Description:	Resource name associated with a connection.
4298	Required:	True
4299	Inherited:	False
4300	Property Type:	String
4301	Cardinality:	Single
4302	Updatability:	On Create
4303		
4304	icom_presence:contactStatus	
4305	Description:	Reachability status to be propagated to an associated contact method.
4306		
4307	Required:	False
4308	Inherited:	False
4309	Property Type:	icom_presence:ContactReachabilityStatus
4310	Cardinality:	Single
4311	Updatability:	Write Only
4312		
4313	icom_presence:contactPriority	
4314	Description:	Priority to be propagated to an associated contact method.
4315	Required:	False
4316	Inherited:	False
4317	Property Type:	Integer
4318	Cardinality:	Single
4319	Updatability:	Write Only
4320		
4321	icom_presence:note	
4322	Description:	Note to be propagated to an associated contact method.
4323	Required:	False
4324	Inherited:	False
4325	Property Type:	String
4326	Cardinality:	Single
4327	Updatability:	Write Only
4328		
4329	icom_msg:inboundInstantMessage	
4330	Description:	A queue for inbound instant messages.
4331	Required:	False

4332	Inherited:	False
4333	Property Type:	icom_msg:InstantMessage
4334	Cardinality:	Multi
4335	Updatability:	Read Only
4336		

4337 4.4 Presence Module

4338 4.4.1 Presence

4339 4.4.1.1 Description

4340 A presence describes the contact methods and activities of a presentity.
 4341 It provides a list of contact methods describing how to contact a presentity. A viewer may choose any one
 4342 of the contact methods based on circumstances.
 4343 It includes a list of activities describing what a presentity is doing.

4344 4.4.1.2 Class Definition

4345 The Presence class is defined by the attribute values:

4346

4347 **localNamespace**
 4348 Value: icom_presence

4349

4350 **localName**
 4351 Value: Presence

4352

4353 **extendsFrom**
 4354 Value: icom:Identifiable

4355

4356 **stereotype**
 4357 Value: primary

4358

4359 **description**
 4360 Value: A presence describes the contact methods and activities of a presentity.

4361

4362 **propertyDefinitions**
 4363 The values for this attribute are defined in Section 4.4.1.3.

4364 4.4.1.3 Property Definitions

4365 The Presence class inherits property definitions from super classes.
 4366 The Presence class MUST have the property definitions:

4367

4368 **icom:lastModificationDate**

4369	Description:	Last modification date and time of information in a presence.
4370	Required:	False

4371	Inherited:	False
4372	Property Type:	DateTime
4373	Cardinality:	Single
4374	Updatability:	Read Only
4375		
4376	icom_presence:mode	
4377	Description:	Editable mode of a presence.
4378	Required:	False
4379	Inherited:	False
4380	Property Type:	icom_presence:PresenceEditMode
4381	Cardinality:	Single
4382	Updatability:	Read Only
4383		
4384	icom_presence:contactMethod	
4385	Description:	A collection of contact methods describing how to contact a
4386		presentity. A viewer may choose any one of the contact
4387		methods based on circumstances.
4388	Required:	False
4389	Inherited:	False
4390	Property Type:	icom_presence:ContactMethod
4391	Cardinality:	Multi
4392	Updatability:	Read Only
4393		
4394	icom_presence:activity	
4395	Description:	A collection of activities describing what a presentity is doing.
4396	Required:	False
4397	Inherited:	False
4398	Property Type:	icom_presence:Activity
4399	Cardinality:	Multi
4400	Updatability:	Read Only
4401		
4402	The Presence class MAY include additional property definitions which are implementation-defined.	
4403		

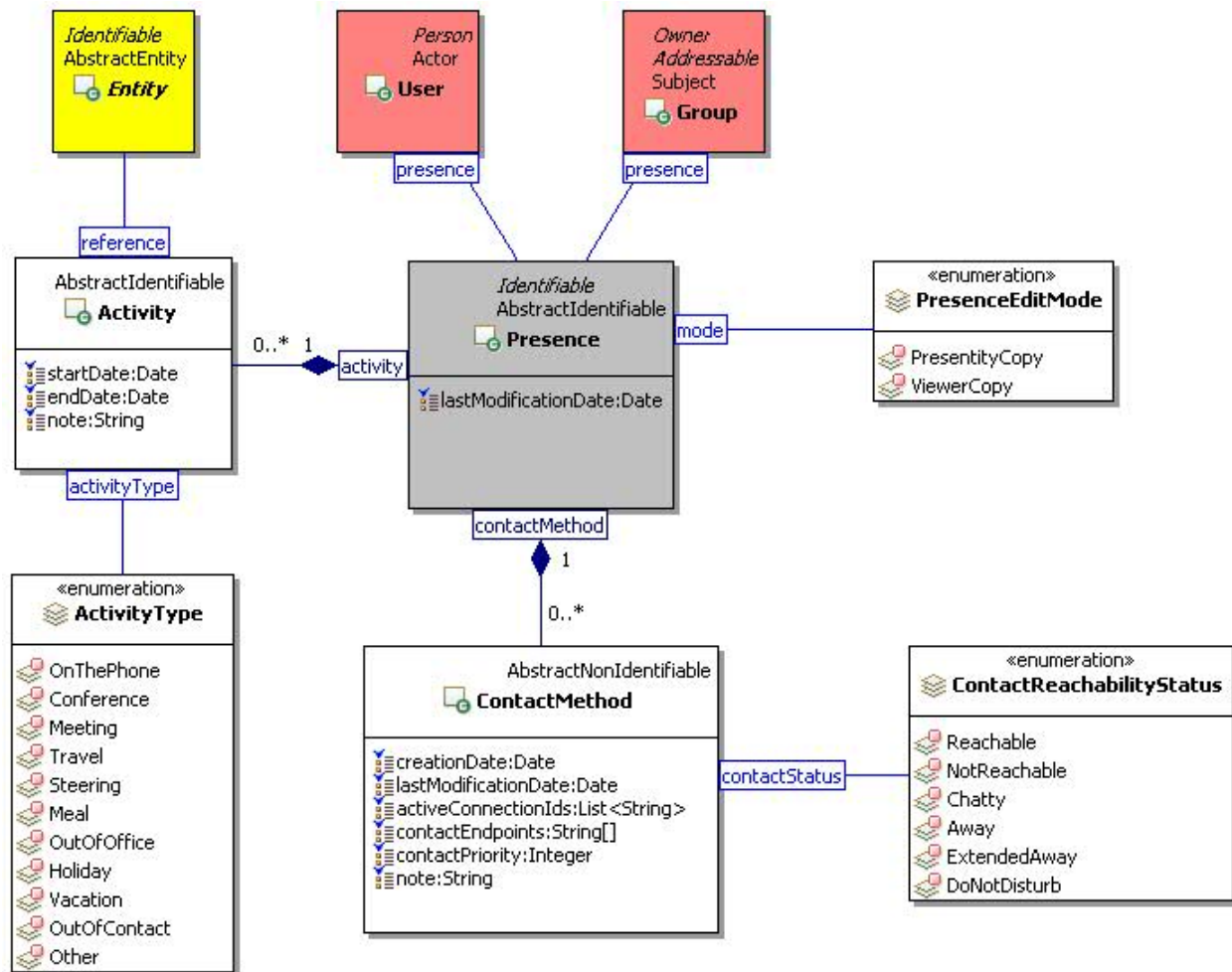


Figure 29 Presence Class Diagram.

4.4.2 PresenceEditMode

The PresenceEditMode class is an enum class that enumerates the instances each of which expresses an editable mode of a presence.

The PresenceEditMode class is defined by the attribute values:

```

localNamespace
    Value: icom_presence

localName
    Value: PresenceEditMode

extendsFrom
    Value:

stereotype
    Value: primary
  
```

4423

4424 **isEnumeration**

4425 Value: TRUE

4426

4427 **description**

4428 Value: An enumeration of the instances each of which expresses an editable mode of a

4429 presence.

4430

4431 **instances**

4432 Value: <icom_presence:PresentityCopy, icom_presence:ViewerCopy>

4433

4434 There are two presence editable modes defined by ICOM:

- 4435 • **icom_presence:PresentityCopy** to express that a presence is a copy belonging to a presentity
- 4436 who may update the properties such activities.
- 4437 • **icom_presence:ViewerCopy** to express that a presence is a copy visible to a subscriber who
- 4438 may not update the properties.

4439

4440 **4.4.3 ContactMethod**

4441 **4.4.3.1 Description**

4442 A contact method object describes reachability circumstances of a presentity.

4443 **4.4.3.2 Class Definition**

4444 The ContactMethod class is defined by the attribute values:

4445

4446 **localNamespace**

4447 Value: icom_presence

4448

4449 **localName**

4450 Value: ContactMethod

4451

4452 **extendsFrom**

4453 Value:

4454

4455 **stereotype**

4456 Value: primary

4457

4458 **description**

4459 Value: A contact method object describes reachability circumstances of a presentity.

4460

4461 **propertyDefinitions**

4462 The values for this attribute are defined in Section 4.4.3.3

4.4.3.3 Property Definitions

The ContactMethod class MUST have the property definitions:

icom:creationDate

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

icom:lastModificationDate

Description:	Last modification date and time of information in a contact method.
Required:	False
Inherited:	False
Property Type:	DateTime
Cardinality:	Single
Updatability:	Read Only

icom_presence:activeConnectionId

Description:	A list of active connection ids of a presentity.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Multi
Updatability:	Read Only

icom_presence:contactEndpoint

Description:	A list of endpoints or URIs for contacting a presentity.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Multi
Updatability:	Read Only

icom_presence:contactPriority

Description:	Priority of a contact method relative to other contact methods in a presence.
Required:	False
Inherited:	False
Property Type:	Integer
Cardinality:	Single

4506	Updatability:	Read Only
4507		
4508	icom_presence:contactStatus	
4509	Description:	Status of a contact method in a presence.
4510	Required:	False
4511	Inherited:	False
4512	Property Type:	icom_presence:ContactReachabilityStatus
4513	Cardinality:	Single
4514	Updatability:	Read Only

4515		
4516	icom_presence:note	
4517	Description:	A note about a contact method in a presence.
4518	Required:	False
4519	Inherited:	False
4520	Property Type:	String
4521	Cardinality:	Single
4522	Updatability:	Read Only

4523

4524 **4.4.4 ContactReachabilityStatus**

4525 The ContactReachabilityStatus class is an enum class that enumerates the instances each of which
 4526 expresses a reachability status of a contact method.

4527 The ContactReachabilityStatus class is defined by the attribute values:

4528

4529	localNamespace	
4530	Value:	icom_presence
4531		
4532	localName	
4533	Value:	ContactReachabilityStatus

4534

4535	extendsFrom	
4536	Value:	

4537

4538	stereotype	
4539	Value:	primary

4540

4541	isEnumeration	
4542	Value:	TRUE

4543

4544	description	
4545	Value:	An enumeration of the instances each of which expresses a reachability status of a
4546		contact method.

4547

4548 **instances**

4549 Value: <icom_presence:Reachable, icom_presence:NotReachable, icom_presence:Chatty,
4550 icom_presence:Away, icom_presence:ExtendedAway, icom_presence:DoNotDisturb>

4551

4552 There are six reachability status defined by ICOM:

- 4553 • **icom_presence:Reachable** to express that a presentity is reachable through a contact method.
- 4554 • **icom_presence:NotReachable** to express that a presentity is not reachable through a contact
4555 method.
- 4556 • **icom_presence:Chatty** to express that a presentity is chatty.
- 4557 • **icom_presence:Away** to express that a presentity is away.
- 4558 • **icom_presence:ExtendedAway** to express that a presentity is away for an extended period.
- 4559 • **icom_presence:DoNotDisturb** to express that a presentity prefers not to be disturbed.

4560

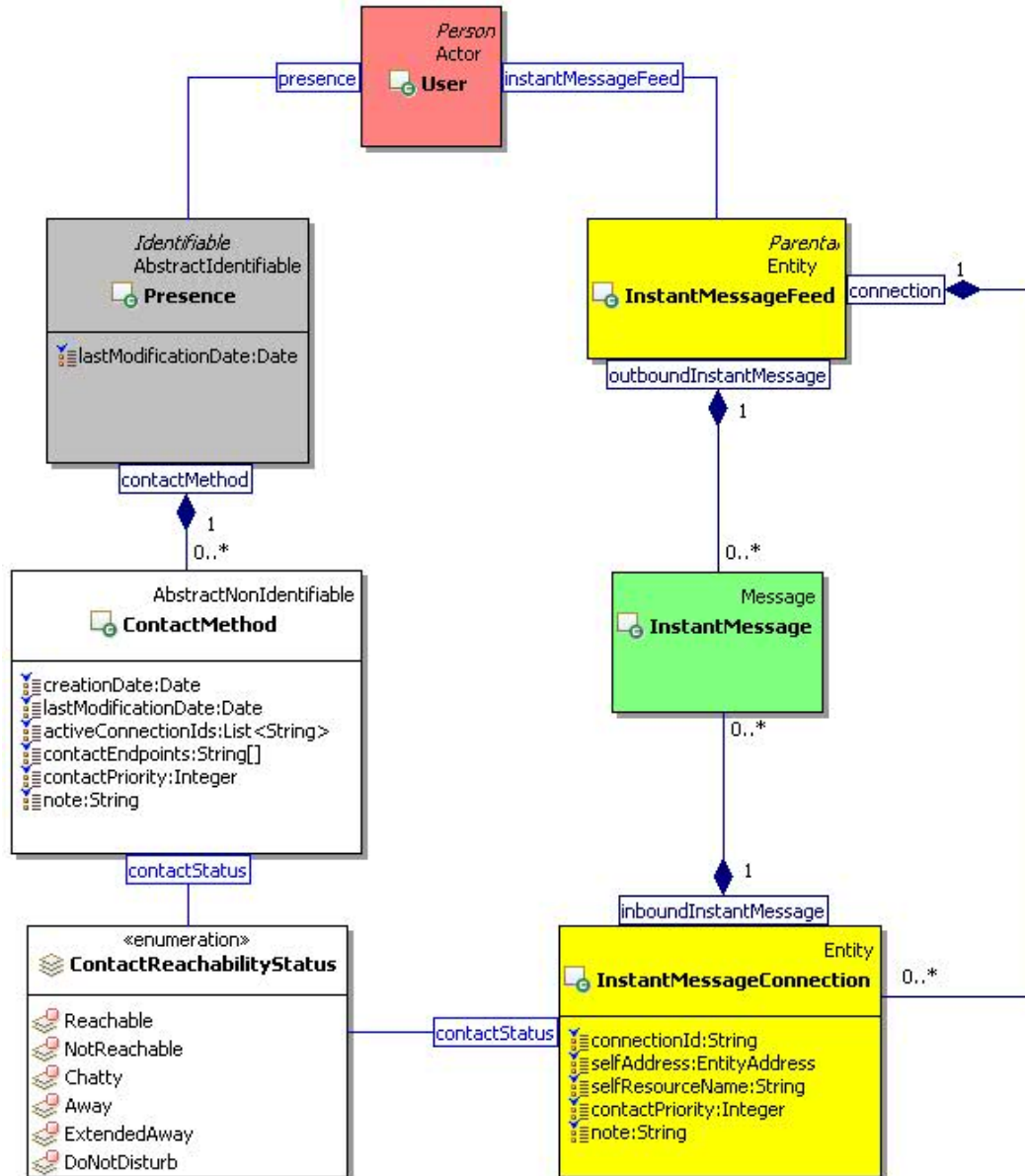


Figure 30 Presence Contact Method and Instant Message Connection Class Diagram.

4.4.5 Activity

4.4.5.1 Description

An activity object describes what a presentity is currently doing.

4.4.5.2 Class Definition

The Activity class is defined by the attribute values:

localNamespace

Value: icom_presence

4573 **localName**
4574 Value: Activity
4575
4576 **extendsFrom**
4577 Value:
4578
4579 **stereotype**
4580 Value: primary
4581
4582 **description**
4583 Value: An activity object describes what a presentity is currently doing.
4584
4585 **propertyDefinitions**
4586 The values for this attribute are defined in Section 4.4.5.3.

4587 **4.4.5.3 Property Definitions**

4588 The Activity class MUST have the property definitions:

4589
4590 **icom:startDate**
4591 Description: Start date and time of an activity.
4592 Required: True
4593 Inherited: False
4594 Property Type: DateTime
4595 Cardinality: Single
4596 Updatability: Read Write

4597
4598 **icom:endDate**
4599 Description: End date and time of an activity.
4600 Required: True
4601 Inherited: False
4602 Property Type: DateTime
4603 Cardinality: Single
4604 Updatability: Read Write

4605
4606 **icom_presence:activityType**
4607 Description: Type of an activity.
4608 Required: true
4609 Inherited: False
4610 Property Type: icom_presence:ActivityType
4611 Cardinality: Single
4612 Updatability: Read Write

4613
4614 **icom_presence:note**

4615	Description:	A note describing an activity.
4616	Required:	False
4617	Inherited:	False
4618	Property Type:	String
4619	Cardinality:	Single
4620	Updatability:	Read Write
4621		
4622	icom_presence:reference	
4623	Description:	An entity, such as occurrence, task, conference, etc., which is the source of or reference for an activity.
4624		
4625	Required:	False
4626	Inherited:	False
4627	Property Type:	icom:Entity
4628	Cardinality:	Single
4629	Updatability:	Read Write
4630		

4631 4.4.6 ActivityType

4632 The ActivityType class is an enum class that enumerates the instances each of which expresses a type of
4633 activity.

4634 The ActivityType class is defined by the attribute values:

4635		
4636	localNamespace	
4637	Value:	icom_presence
4638		
4639	localName	
4640	Value:	ActivityType
4641		
4642	extendsFrom	
4643	Value:	
4644		
4645	stereotype	
4646	Value:	primary
4647		
4648	isEnumeration	
4649	Value:	TRUE
4650		
4651	description	
4652	Value:	An enumeration of the instances each of which expresses a type of activity.
4653		
4654	instances	
4655	Value:	<icom_presence:OnThePhone, icom_ presence:Conference, icom_ presence:Meeting,
4656		icom_presence:Travel, icom_ presence:Steering, icom_ presence:Meal,

4657 icom_presence:OutOfOffice, icom_ presence:Holiday, icom_ presence:Vacation, icom_
4658 presence:OutOfContact, icom_ presence:Other>

4659

4660 There are eleven activity types defined by ICOM:

- 4661 • **icom_presence:OnThePhone** to express that a presentity is on the phone.
- 4662 • **icom_presence:Conference** to express that a presentity is in a conference.
- 4663 • **icom_presence:Meeting** to express that a presentity is in a meeting.
- 4664 • **icom_presence:Travel** to express that a presentity is traveling.
- 4665 • **icom_presence:Steering** to express that a presentity is steering a vehicle.
- 4666 • **icom_presence:Meal** to express that a presentity is having a meal.
- 4667 • **icom_presence:OutOfOffice** to express that a presentity is out of office.
- 4668 • **icom_presence:Holiday** to express that a presentity is on holiday.
- 4669 • **icom_presence:Vacation** to express that a presentity is on vacation.
- 4670 • **icom_presence:OutOfContact** to express that a presentity is out of contact.
- 4671 • **icom_presence:Other** to express that a presentity is involved in an unspecified activity.

4672

4673 4.5 Address Book Module

4674 4.5.1 Addressable

4675 4.5.1.1 Description

4676 An addressable object is an identifiable object which has email and other addresses.

4677 4.5.1.2 Class Definition

4678 The Addressable class is a mixin class which defines the characteristics of entities that has email and
4679 other addresses.

4680 The Addressable class is defined by the attribute values:

4681

4682 **localNamespace**

4683 Value: icom_card

4684

4685 **localName**

4686 Value: Addressable

4687

4688 **extendsFrom**

4689 Value: icom:Identifiable

4690

4691 **stereotype**

4692 Value: mixin

4693

4694 **description**

4695 Value: Addressable is a mixin class which defines the characteristics of entities that has email
4696 and other addresses.

4697
4698 **propertyDefinitions**
4699 The values for this attribute are defined in 4.5.1.3.

4700 **4.5.1.3 Property Definitions**

4701 The Addressable class inherits property definitions from super classes.
4702 The Addressable class **MUST** have the property definitions:

4703
4704 **icom_card:address**
4705 Description: Zero or more addresses of an addressable object.
4706 Required: False
4707 Inherited: False
4708 Property Type: icom_card:EntityAddress
4709 Cardinality: Multi
4710 Updatability: Read Write

4711
4712 **icom_card:primaryAddress**
4713 Description: The primary address of an addressable object.
4714 Required: False
4715 Inherited: False
4716 Property Type: icom_card:EntityAddress
4717 Cardinality: Single
4718 Updatability: Read Write

4719
4720 The Addressable class **MAY** include additional property definitions which are implementation-defined.
4721

4722 **4.5.2 Person**

4723 **4.5.2.1 Description**

4724 A person object is an addressable object which has a given name, middle name, family name, nickname,
4725 etc.

4726 **4.5.2.2 Class Definition**

4727 The Person class is a mixin class which defines the characteristics of persons.
4728 The Person class is defined by the attribute values:

4729
4730 **localNamespace**
4731 Value: icom_card
4732
4733 **localName**
4734 Value: Person
4735
4736 **extendsFrom**

4737 Value: icom_card:Person
 4738
 4739 **stereotype**
 4740 Value: mixin
 4741
 4742 **description**
 4743 Value: A person object is an addressable object which has a given name, middle name, family
 4744 name, nickname, etc.
 4745
 4746 **propertyDefinitions**
 4747 The values for this attribute are defined in Section 4.5.2.3.

4748 4.5.2.3 Property Definitions

4749 The Person class inherits property definitions from super classes.
 4750 The Person class MUST have the property definitions:

4751
 4752 **icom_card:givenName**
 4753 Description: Given name of a person.
 4754 Required: False
 4755 Inherited: False
 4756 Property Type: String
 4757 Cardinality: Single
 4758 Updatability: Read Write

4759
 4760 **icom_card:middleName**
 4761 Description: Middle name of a person.
 4762 Required: False
 4763 Inherited: False
 4764 Property Type: String
 4765 Cardinality: Single
 4766 Updatability: Read Write

4767
 4768 **icom_card:familyName**
 4769 Description: Family name of a person.
 4770 Required: False
 4771 Inherited: False
 4772 Property Type: String
 4773 Cardinality: Single
 4774 Updatability: Read Write

4775
 4776 **icom_card:prefix**
 4777 Description: Prefix of a person's name.
 4778 Required: False

4779	Inherited:	False
4780	Property Type:	String
4781	Cardinality:	Single
4782	Updatability:	Read Write
4783		
4784	icom_card:suffix	
4785	Description:	Suffix of a person's name.
4786	Required:	False
4787	Inherited:	False
4788	Property Type:	String
4789	Cardinality:	Single
4790	Updatability:	Read Write
4791		
4792	icom_card:nickname	
4793	Description:	Nickname of a person.
4794	Required:	False
4795	Inherited:	False
4796	Property Type:	String
4797	Cardinality:	Single
4798	Updatability:	Read Write
4799		
4800	icom_card:jobTitle	
4801	Description:	Job title of a person.
4802	Required:	False
4803	Inherited:	False
4804	Property Type:	String
4805	Cardinality:	Single
4806	Updatability:	Read Write
4807		
4808	icom_card:department	
4809	Description:	A person's affiliated department.
4810	Required:	False
4811	Inherited:	False
4812	Property Type:	String
4813	Cardinality:	Single
4814	Updatability:	Read Write
4815		
4816	icom_card:officeLocation	
4817	Description:	Location of a person's department.
4818	Required:	False
4819	Inherited:	False
4820	Property Type:	String

4821	Cardinality:	Single
4822	Updatability:	Read Write
4823		
4824	icom_card:company	
4825	Description:	A person's affiliated company.
4826	Required:	False
4827	Inherited:	False
4828	Property Type:	String
4829	Cardinality:	Single
4830	Updatability:	Read Write

4831		
4832	icom_card:profession	
4833	Description:	A person's profession.
4834	Required:	False
4835	Inherited:	False
4836	Property Type:	String
4837	Cardinality:	Single
4838	Updatability:	Read Write

4839

4840 The Person class MAY include additional property definitions which are implementation-defined.

4841

4842 4.5.3 AddressBook

4843 4.5.3.1 Description

4844 An address book is a folder that contains addressable contacts.

4845 4.5.3.2 Class Definition

4846 The AddressBook class is defined by the attribute values:

4847

4848	localNamespace
4849	Value: icom_card
4850	
4851	localName
4852	Value: AddressBook
4853	
4854	extendsFrom
4855	Value: icom:Folder
4856	
4857	stereotype
4858	Value: primary
4859	
4860	description

Value: An address book is a folder that contains addressable contacts.

propertyDefinitions

The values for this attribute are defined in Section 4.5.3.3.

4.5.3.3 Property Definitions

The AddressBook class inherits property definitions from super classes.

The AddressBook class MUST have the property definitions:

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

icom_card:contact

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

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

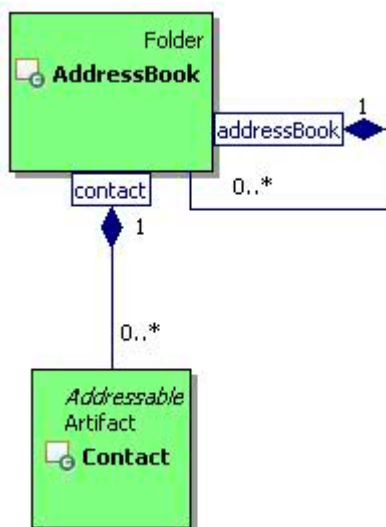


Figure 31 AddressBook Class Diagram.

4.5.4 Contact

4.5.4.1 Description

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

4.5.4.2 Class Definition

The Contact class is defined by the attribute values:

localNamespace

Value: icom_card

localName

Value: Contact

extendsFrom

Value: icom:Artifact, icom_card:Addressable

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.5.4.3.

4.5.4.3 Property Definitions

The Contact class inherits property definitions from super classes.

The Contact class MUST have the property definitions:

icom_card:bookmark

Description: A person which is bookmarked by a contact.

Required: False

Inherited: False

Property Type: icom_card:Person

Cardinality: Single

Updatability: On Create

icom_card:timeZone

Description: Time zone of a person.

Required: False

Inherited: False

Property Type: TimeZone

4930	Cardinality:	Multi
4931	Updatability:	Read Write
4932		
4933	icom_content:attachment	
4934	Description:	One or more simple content attachments in a contact.
4935	Required:	False
4936	Inherited:	False
4937	Property Type:	icom_content:Attachment
4938	Cardinality:	Multi
4939	Updatability:	Read Write
4940		
4941	icom_card:givenName	
4942	Description:	Given name of a person.
4943	Required:	False
4944	Inherited:	False
4945	Property Type:	String
4946	Cardinality:	Single
4947	Updatability:	Read Write
4948		
4949	icom_card:middleName	
4950	Description:	Middle name of a person.
4951	Required:	False
4952	Inherited:	False
4953	Property Type:	String
4954	Cardinality:	Single
4955	Updatability:	Read Write
4956		
4957	icom_card:familyName	
4958	Description:	Family name of a person.
4959	Required:	False
4960	Inherited:	False
4961	Property Type:	String
4962	Cardinality:	Single
4963	Updatability:	Read Write
4964		
4965	icom_card:prefix	
4966	Description:	Prefix of a person's name.
4967	Required:	False
4968	Inherited:	False
4969	Property Type:	String
4970	Cardinality:	Single
4971	Updatability:	Read Write

4972		
4973	icom_card:suffix	
4974	Description:	Suffix of a person's name.
4975	Required:	False
4976	Inherited:	False
4977	Property Type:	String
4978	Cardinality:	Single
4979	Updatability:	Read Write
4980		
4981	icom_card:nickname	
4982	Description:	Nickname of a person.
4983	Required:	False
4984	Inherited:	False
4985	Property Type:	String
4986	Cardinality:	Single
4987	Updatability:	Read Write
4988		
4989	icom_card:jobTitle	
4990	Description:	Job title of a person.
4991	Required:	False
4992	Inherited:	False
4993	Property Type:	String
4994	Cardinality:	Single
4995	Updatability:	Read Write
4996		
4997	icom_card:department	
4998	Description:	A person's affiliated department.
4999	Required:	False
5000	Inherited:	False
5001	Property Type:	String
5002	Cardinality:	Single
5003	Updatability:	Read Write
5004		
5005	icom_card:officeLocation	
5006	Description:	Location of a person's department.
5007	Required:	False
5008	Inherited:	False
5009	Property Type:	String
5010	Cardinality:	Single
5011	Updatability:	Read Write
5012		
5013	icom_card:company	

5014	Description:	A person's affiliated company.
5015	Required:	False
5016	Inherited:	False
5017	Property Type:	String
5018	Cardinality:	Single
5019	Updatability:	Read Write

5020

5021 **icom_card:profession**

5022	Description:	A person's profession.
5023	Required:	False
5024	Inherited:	False
5025	Property Type:	String
5026	Cardinality:	Single
5027	Updatability:	Read Write

5028

5029 The Contact class MAY include additional property definitions which are implementation-defined.

5030

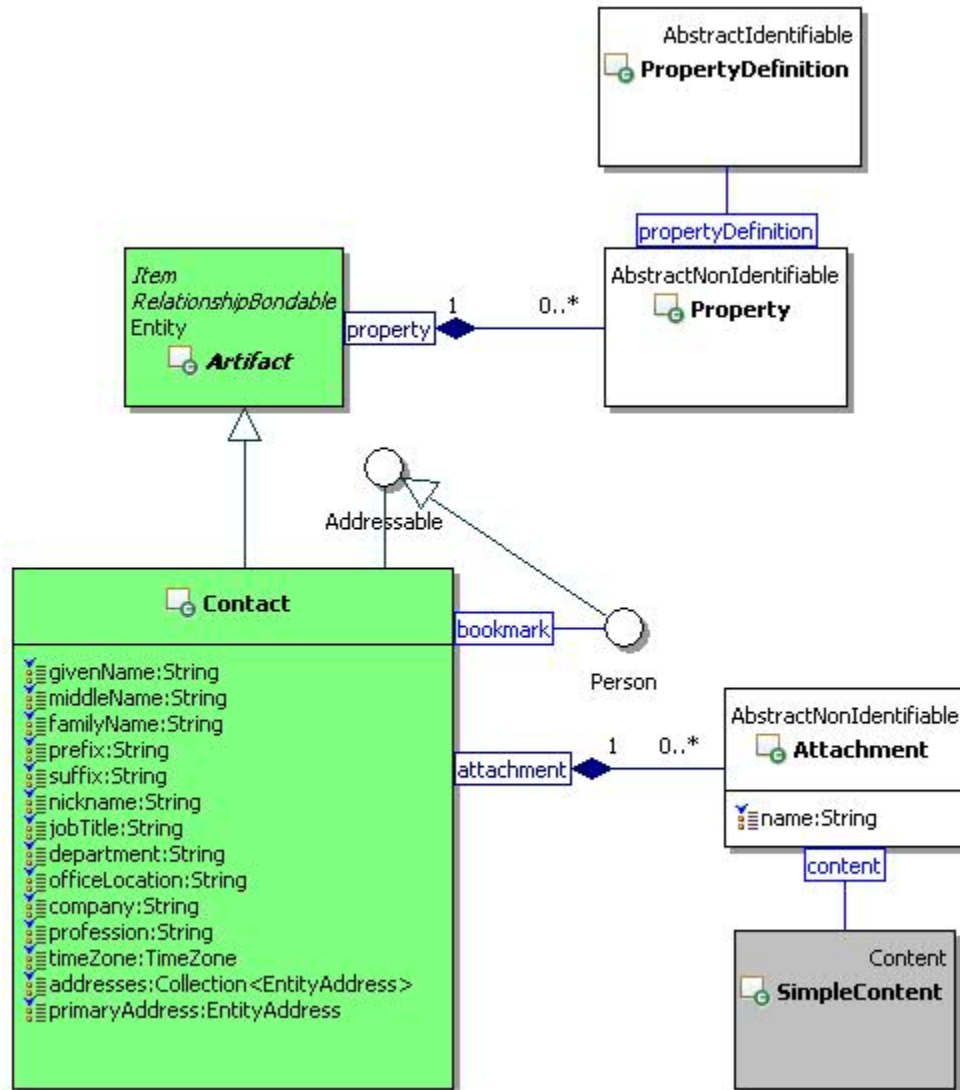


Figure 32 Contact Class Diagram.

4.6 Calendar Module

4.6.1 Calendar

4.6.1.1 Description

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

4.6.1.2 Class Definition

The Calendar class is defined by the attribute values:

```

localNamespace
  Value: icom_cal

```

5044

5045 **localName**

5046 Value: Calendar

5047

5048 **extendsFrom**

5049 Value: icom:Folder

5050

5051 **stereotype**

5052 Value: primary

5053

5054 **description**

5055 Value: A forum is a folder that contains time management artifacts such as occurrences and

5056 occurrence series.

5057

5058 **propertyDefinitions**

5059 The values for this attribute are defined in 4.6.1.3.

5060 **4.6.1.3 Property Definitions**

5061 The Calendar class inherits property definitions from super classes.

5062 The Calendar class **MUST** have the property definitions:

5063

5064 **icom_cal:timeZone**

5065 Description:	Time zone of a calendar.
5066 Required:	True
5067 Inherited:	False
5068 Property Type:	TimeZone
5069 Cardinality:	Single
5070 Updatability:	Read Write

5071

5072 **icom:element**

5073 Description:	Elements of a calendar.
5074 Required:	False
5075 Inherited:	True
5076 Property Type:	icom_cal:Occurrence
5077 Cardinality:	Multi
5078 Updatability:	Read Only

5079

5080 **icom_cal:recurrence**

5081 Description:	Recurrence elements of a calendar.
5082 Required:	False
5083 Inherited:	True
5084 Property Type:	icom_cal:OccurrenceSeries
5085 Cardinality:	Multi

Updatability: Read Only

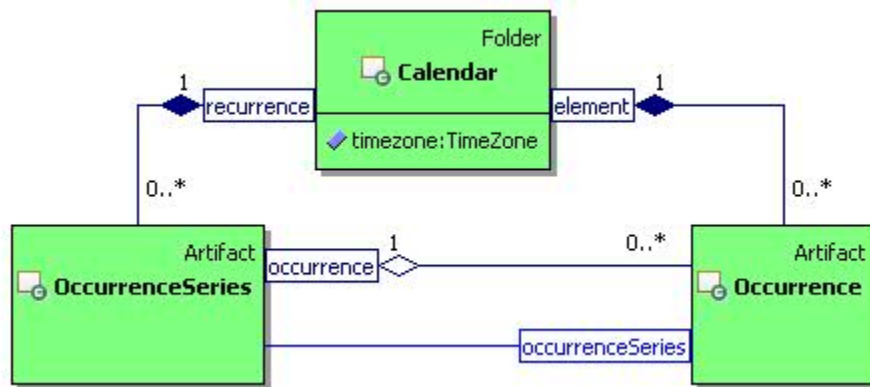


Figure 33 Calendar Class Diagram.

4.6.2 OccurrenceSeries

4.6.2.1 Description

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

4.6.2.2 Class Definition

The OccurrenceSeries class is defined by the attribute values:

localNamespace

Value: icom_cal

localName

Value: OccurrenceSeries

extendsFrom

Value: icom:Artifact

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in 4.6.2.3.

4.6.2.3 Property Definitions

The OccurrenceSeries class inherits property definitions from super classes.

The OccurrenceSeries class MUST have the property definitions:

icom_cal:recurrenceStartDate

Description:	Start date and time of an occurrence series.
Required:	True
Inherited:	False
Property Type:	DateTime
Cardinality:	Single
Updatability:	On Create

icom_cal:recurrenceStartDateResolution

Description:	Resolution of start date and time of an occurrence series.
Required:	True
Inherited:	False
Property Type:	icom:DateTimeResolution
Cardinality:	Single
Updatability:	On Create

icom_cal:duration

Description:	Duration of each occurrence in an occurrence series.
Required:	True
Inherited:	False
Property Type:	Duration
Cardinality:	Single
Updatability:	On Create

icom_cal:recurrenceRule

Description:	A recurrence rule of an occurrence series.
Required:	True
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	On Create

icom:location

Description:	Location of an occurrence series.
Required:	False
Inherited:	False
Property Type:	icom:Location
Cardinality:	Single

5158	Updatability:	Read Write
5159		
5160	icom:organizer	
5161	Description:	Organizer of an occurrence series.
5162	Required:	True
5163	Inherited:	False
5164	Property Type:	icom:Participant
5165	Cardinality:	Single
5166	Updatability:	On Create
5167		
5168	icom:participant	
5169	Description:	Participants of an occurrence series.
5170	Required:	False
5171	Inherited:	False
5172	Property Type:	icom_cal:OccurrenceParticipant
5173	Cardinality:	Multi
5174	Updatability:	Read Write
5175		
5176	icom_cal:occurrenceStatus	
5177	Description:	Status of an occurrence series.
5178	Required:	True
5179	Inherited:	False
5180	Property Type:	icom_cal:OccurrenceStatus
5181	Cardinality:	Single
5182	Updatability:	Read Write
5183		
5184	icom_cal:occurrenceType	
5185	Description:	Type of an occurrence series.
5186	Required:	True
5187	Inherited:	False
5188	Property Type:	icom_cal:OccurrenceType
5189	Cardinality:	Single
5190	Updatability:	Read Write
5191		
5192	icom_cal:mode	
5193	Description:	Mutability mode of an occurrence series.
5194	Required:	False
5195	Inherited:	False
5196	Property Type:	icom_cal:OccurrenceEditMode
5197	Cardinality:	Single
5198	Updatability:	Read Only
5199		

5200	icom_cal:occurrence	
5201	Description:	Occurrences in an occurrence series.
5202	Required:	False
5203	Inherited:	False
5204	Property Type:	icom_cal:Occurrence
5205	Cardinality:	Multi
5206	Updatability:	Read Only
5207		
5208	icom_content:attachment	
5209	Description:	One or more simple content attachments in an occurrence series.
5210		
5211	Required:	False
5212	Inherited:	False
5213	Property Type:	icom_content:Attachment
5214	Cardinality:	Multi
5215	Updatability:	Read Write
5216		
5217	icom_cal:attendee	
5218	Description:	An attendee of an occurrence series.
5219	Required:	False
5220	Inherited:	False
5221	Property Type:	icom:Participant
5222	Cardinality:	Single
5223	Updatability:	Read Only
5224		
5225	icom_cal:attendeePriority	
5226	Description:	Priority for an attendee of an occurrence series.
5227	Required:	False
5228	Inherited:	False
5229	Property Type:	icom:Priority
5230	Cardinality:	Single
5231	Updatability:	Read Write
5232		
5233	icom_cal:attendeeParticipantStatus	
5234	Description:	Participation status for an attendee of an occurrence series.
5235	Required:	False
5236	Inherited:	False
5237	Property Type:	icom_cal:OccurrenceParticipantStatus
5238	Cardinality:	Single
5239	Updatability:	Read Write
5240		
5241	icom_cal:attendeeTransparency	

5242	Description:	Participant transparency for an attendee of an occurrence series.
5243		
5244	Required:	False
5245	Inherited:	False
5246	Property Type:	icom_cal:ParticipantTransparency
5247	Cardinality:	Single
5248	Updatability:	Read Write
5249		
5250	icom_cal:attendeeProperty	
5251	Description:	Extensible properties for an attendee of an occurrence series.
5252	Required:	False
5253	Inherited:	False
5254	Property Type:	icom:Property
5255	Cardinality:	Multi
5256	Updatability:	Read Write
5257		
5258	icom_cal:conference	
5259	Description:	One or more conferences for an occurrence series.
5260	Required:	False
5261	Inherited:	False
5262	Property Type:	icom_conf:Conference
5263	Cardinality:	Multi
5264	Updatability:	Read Write
5265		

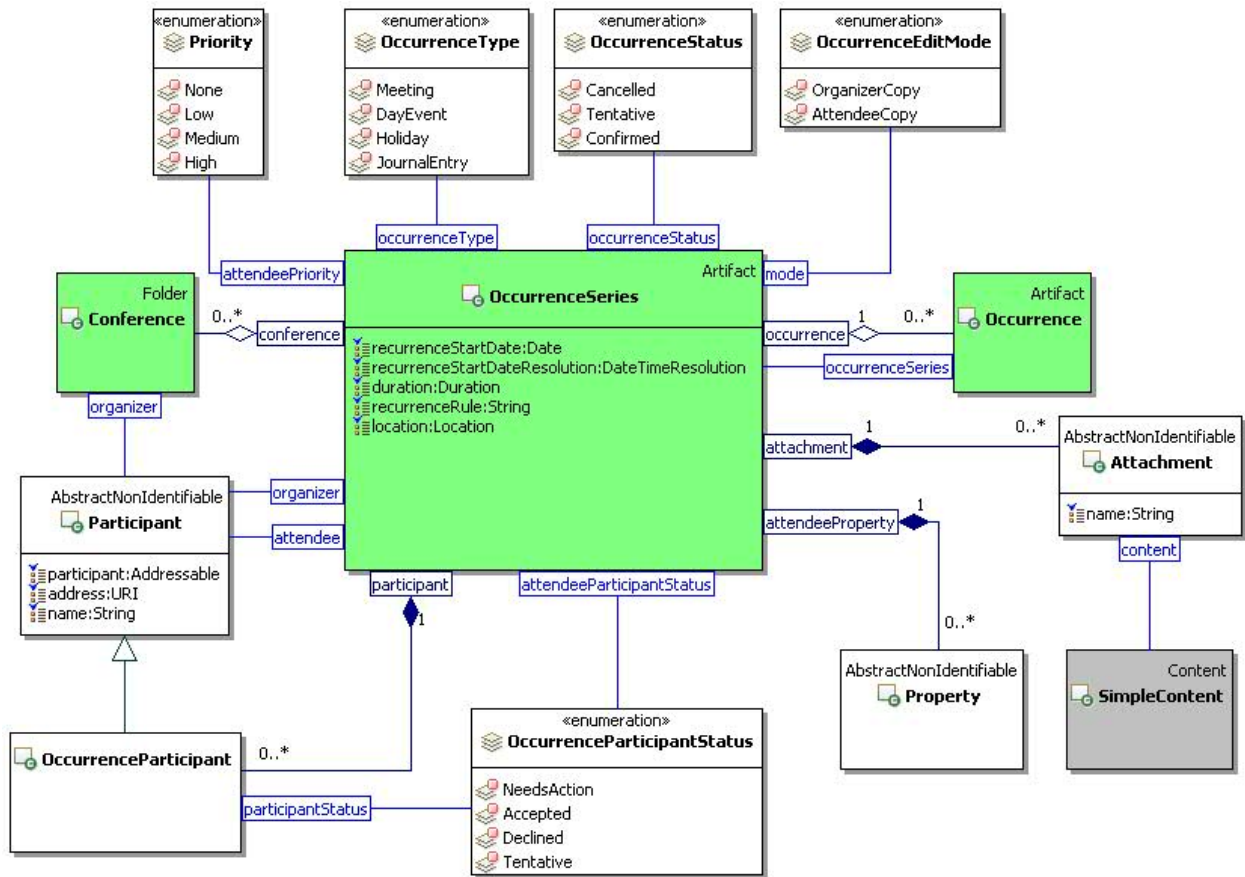


Figure 34 OccurrenceSeries Class Diagram.

4.6.3 Occurrence

4.6.3.1 Description

An occurrence is an artifact that represents an event in a calendar.

4.6.3.2 Class Definition

The Occurrence class is defined by the attribute values:

```

localNamespace
    Value: icom_cal

localName
    Value: Occurrence

extendsFrom
    Value: icom:Artifact

stereotype

```

5285 Value: primary

5286

5287 **description**

5288 Value: An occurrence is an artifact that represents an event in a calendar.

5289

5290 **propertyDefinitions**

5291 The values for this attribute are defined in 4.6.3.3.

5292

5293 4.6.3.3 Property Definitions

5294 The Occurrence class inherits property definitions from super classes.

5295 The Occurrence class MUST have the property definitions:

5296

5297 **icom_cal:startDate**

5298	Description:	Start date and time of an occurrence.
5299	Required:	True
5300	Inherited:	False
5301	Property Type:	DateTime
5302	Cardinality:	Single
5303	Updatability:	On Create

5304

5305 **icom_cal:startDateResolution**

5306	Description:	Resolution of start date and time of an occurrence.
5307	Required:	True
5308	Inherited:	False
5309	Property Type:	icom:DateTimeResolution
5310	Cardinality:	Single
5311	Updatability:	On Create

5312

5313 **icom_cal:endDate**

5314	Description:	End date and time of an occurrence.
5315	Required:	True
5316	Inherited:	False
5317	Property Type:	DateTime
5318	Cardinality:	Single
5319	Updatability:	On Create

5320

5321 **icom_cal:endDateResolution**

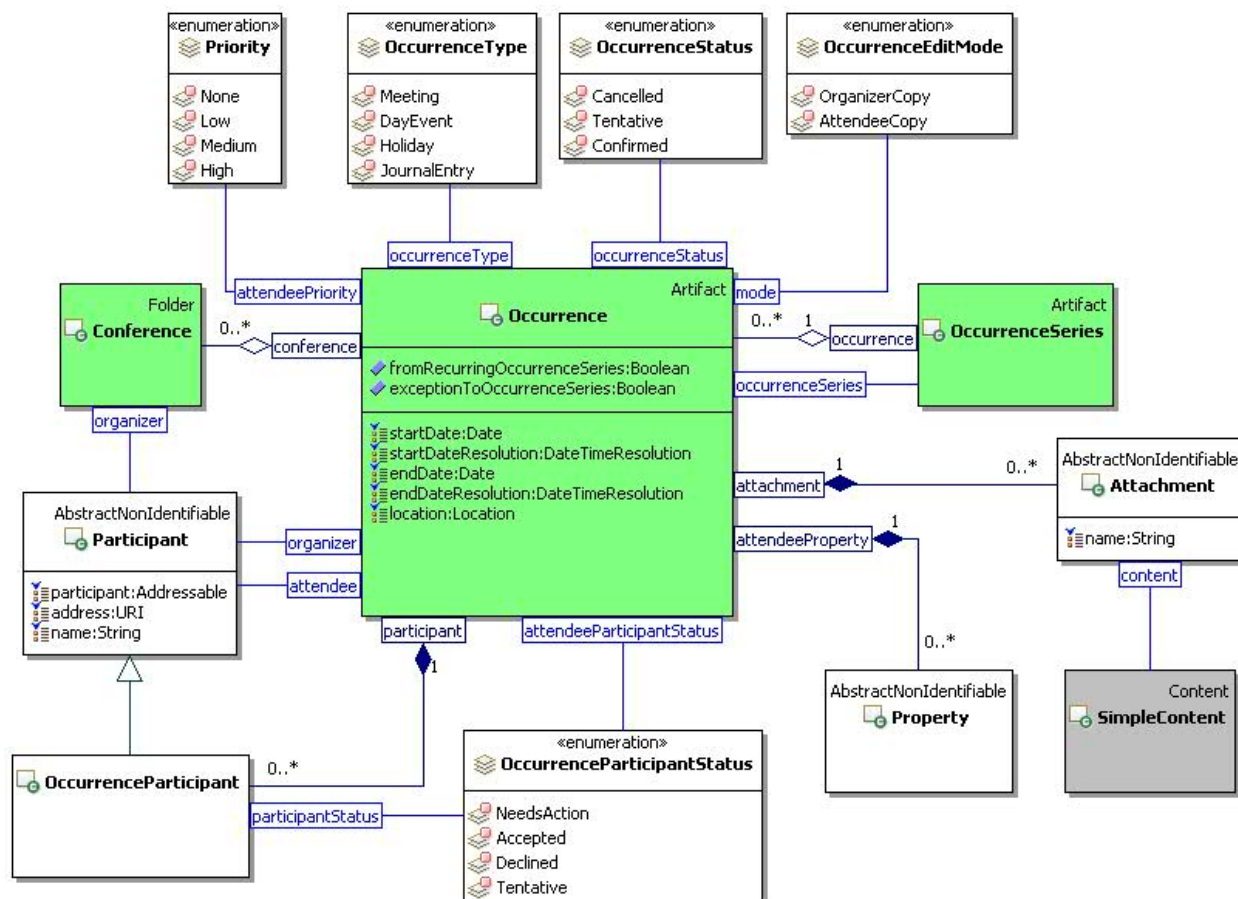
5322	Description:	Resolution of end date and time of an occurrence.
5323	Required:	True
5324	Inherited:	False
5325	Property Type:	icom:DateTimeResolution
5326	Cardinality:	Single

5327	Updatability:	On Create
5328		
5329	icom:location	
5330	Description:	Location of an occurrence.
5331	Required:	False
5332	Inherited:	False
5333	Property Type:	icom:Location
5334	Cardinality:	Single
5335	Updatability:	Read Write
5336		
5337	icom_cal:occurrenceSeries	
5338	Description:	An occurrence series that includes an occurrence.
5339	Required:	False
5340	Inherited:	False
5341	Property Type:	icom_cal:OccurrenceSeries
5342	Cardinality:	Single
5343	Updatability:	Read Only
5344		
5345	icom_cal:fromRecurringOccurrenceSeries	
5346	Description:	Occurrence is part of a recurring occurrence series.
5347	Required:	False
5348	Inherited:	False
5349	Property Type:	Boolean
5350	Cardinality:	Single
5351	Updatability:	Read Only
5352		
5353	icom_cal:exceptionToOccurrenceSeries	
5354	Description:	Occurrence is an exception to an occurrence series.
5355	Required:	False
5356	Inherited:	False
5357	Property Type:	Boolean
5358	Cardinality:	Single
5359	Updatability:	Read Only
5360		
5361	icom:organizer	
5362	Description:	Organizer of an occurrence.
5363	Required:	True
5364	Inherited:	False
5365	Property Type:	icom:Participant
5366	Cardinality:	Single
5367	Updatability:	On Create
5368		

5369	icom:participant	
5370	Description:	Participants of an occurrence.
5371	Required:	False
5372	Inherited:	False
5373	Property Type:	icom_cal:OccurrenceParticipant
5374	Cardinality:	Multi
5375	Updatability:	Read Write
5376		
5377	icom_cal:occurrenceStatus	
5378	Description:	Status of an occurrence.
5379	Required:	True
5380	Inherited:	False
5381	Property Type:	icom_cal:OccurrenceStatus
5382	Cardinality:	Single
5383	Updatability:	Read Write
5384		
5385	icom_cal:occurrenceType	
5386	Description:	Type of an occurrence.
5387	Required:	True
5388	Inherited:	False
5389	Property Type:	icom_cal:OccurrenceType
5390	Cardinality:	Single
5391	Updatability:	Read Write
5392		
5393	icom_cal:mode	
5394	Description:	Mutability mode of an occurrence.
5395	Required:	False
5396	Inherited:	False
5397	Property Type:	icom_cal:OccurrenceEditMode
5398	Cardinality:	Single
5399	Updatability:	Read Only
5400		
5401	icom_content:attachment	
5402	Description:	One or more simple content attachments in an occurrence.
5403	Required:	False
5404	Inherited:	False
5405	Property Type:	icom_content:Attachment
5406	Cardinality:	Multi
5407	Updatability:	Read Write
5408		
5409	icom_cal:attendee	
5410	Description:	An attendee of an occurrence.

5411	Required:	False
5412	Inherited:	False
5413	Property Type:	icom:Participant
5414	Cardinality:	Single
5415	Updatability:	Read Only
5416		
5417	icom_cal:attendeePriority	
5418	Description:	Priority for an attendee of an occurrence.
5419	Required:	False
5420	Inherited:	False
5421	Property Type:	icom:Priority
5422	Cardinality:	Single
5423	Updatability:	Read Write
5424		
5425	icom_cal:attendeeParticipantStatus	
5426	Description:	Participation status for an attendee of an occurrence.
5427	Required:	False
5428	Inherited:	False
5429	Property Type:	icom_cal:OccurrenceParticipantStatus
5430	Cardinality:	Single
5431	Updatability:	Read Write
5432		
5433	icom_cal:attendeeTransparency	
5434	Description:	Participant transparency for an attendee of an occurrence.
5435	Required:	False
5436	Inherited:	False
5437	Property Type:	icom_cal:ParticipantTransparency
5438	Cardinality:	Single
5439	Updatability:	Read Write
5440		
5441	icom_cal:attendeeProperty	
5442	Description:	Extensible properties for an attendee of an occurrence.
5443	Required:	False
5444	Inherited:	False
5445	Property Type:	icom:Property
5446	Cardinality:	Multi
5447	Updatability:	Read Write
5448		
5449	icom_cal:conference	
5450	Description:	One or more conferences for an occurrence.
5451	Required:	False
5452	Inherited:	False

5453 Property Type: icom_conf:Conference
 5454 Cardinality: Multi
 5455 Updatability: Read Write
 5456



5457
 5458 Figure 35 Occurrence Class Diagram.
 5459

5460 4.6.4 OccurrenceStatus

5461 The OccurrenceStatus class is an enum class that enumerates the instances each of which expresses a
 5462 status of an occurrence or occurrence series.

5463 The OccurrenceStatus class is defined by the attribute values:

5464
 5465 **localNamespace**
 5466 Value: icom_cal
 5467
 5468 **localName**
 5469 Value: OccurrenceStatus
 5470
 5471 **extendsFrom**
 5472 Value:
 5473

5474 **stereotype**
5475 Value: primary
5476
5477 **isEnumeration**
5478 Value: TRUE
5479
5480 **description**
5481 Value: An enumeration of the instances each of which expresses a status of an occurrence or
5482 occurrence series.
5483
5484 **instances**
5485 Value: <icom_cal:Cancelled, icom_cal:Tentative, icom_cal:Confirmed>
5486
5487 There are three occurrence status defined by ICOM:
5488 • **icom_cal:Cancelled** to express that an occurrence or occurrence series is cancelled.
5489 • **icom_cal:Tentative** to express that an occurrence or occurrence series is tentative.
5490 • **icom_cal:Confirmed** to express that an occurrence or occurrence series is confirmed.
5491

5492 **4.6.5 OccurrenceType**

5493 The OccurrenceType class is an enum class that enumerates the instances each of which expresses a
5494 type of an occurrence or occurrence series.
5495 The OccurrenceType class is defined by the attribute values:

5496
5497 **localNamespace**
5498 Value: icom_cal
5499
5500 **localName**
5501 Value: OccurrenceType
5502
5503 **extendsFrom**
5504 Value:
5505
5506 **stereotype**
5507 Value: primary
5508
5509 **isEnumeration**
5510 Value: TRUE
5511
5512 **description**
5513 Value: An enumeration of the instances each of which expresses a type of an occurrence or
5514 occurrence series.
5515
5516 **instances**

5517 Value: <icom_cal:Meeting, icom_cal:DayEvent, icom_cal:Holiday, icom_cal:JournalEntry>

5518

5519 There are four occurrence types defined by ICOM:

- 5520 • **icom_cal:Meeting** to express that an occurrence or occurrence series is a meeting.
- 5521 • **icom_cal:DayEvent** to express that an occurrence or occurrence series is a day event.
- 5522 • **icom_cal:Holiday** to express that an occurrence or occurrence series is a holiday.
- 5523 • **icom_cal:JournalEntry** to express that an occurrence or occurrence series is a journal entry.

5524

5525 4.6.6 OccurrenceEditMode

5526 The OccurrenceEditMode class is an enum class that enumerates the instances each of which expresses
5527 an editable mode of an occurrence or occurrence series.

5528 The OccurrenceEditMode class is defined by the attribute values:

5529

5530 **localNamespace**

5531 Value: icom_cal

5532

5533 **localName**

5534 Value: OccurrenceEditMode

5535

5536 **extendsFrom**

5537 Value:

5538

5539 **stereotype**

5540 Value: primary

5541

5542 **isEnumeration**

5543 Value: TRUE

5544

5545 **description**

5546 Value: An enumeration of the instances each of which expresses an editable mode of an
5547 occurrence or occurrence series.

5548

5549 **instances**

5550 Value: <icom_cal:OrganizerCopy, icom_cal:AttendeeCopy>

5551

5552 There are two occurrence editable modes defined by ICOM:

- 5553 • **icom_cal:OrganizerCopy** to express that an occurrence or occurrence series is a copy created
5554 by an organizer who may update the properties such as occurrence type, occurrence status, etc.
- 5555 • **icom_cal:AttendeeCopy** to express that an occurrence or occurrence series is a copy delivered
5556 to an attendee who may only update the attendee properties such as attendee priority, attendee
5557 transparency, etc .

5558

4.6.7 ParticipantTransparency

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

The ParticipantTransparency class is defined by the attribute values:

localNamespace

Value: icom_cal

localName

Value: ParticipantTransparency

extendsFrom

Value:

stereotype

Value: primary

isEnumeration

Value: TRUE

description

Value: An enumeration of the instances each of which expresses an occurrence or occurrence series transparency in a participant's calendar or free busy.

instances

Value: <icom_cal:Opaque, icom_cal:Transparent, icom_cal:Tentative, icom_cal:OutOfOffice, icom_cal:DefaultTransparency >

There are five participant transparencies defined by ICOM:

- **icom_cal:Opaque** to express that an occurrence or occurrence series is opaque in a participant's calendar or free busy.
- **icom_cal:Transparent** to express that an occurrence or occurrence series is transparent in a participant's calendar or free busy.
- **icom_cal:Tentative** to express that an occurrence or occurrence series has a tentative transparency in a participant's calendar or free busy.
- **icom_cal:OutOfOffice** to express that an occurrence or occurrence series has out of office transparency in a participant's calendar or free busy.
- **icom_cal:DefaultTransparency** to express that an occurrence or occurrence series has default transparency in a participant's calendar or free busy.

4.6.8 OccurrenceParticipant

4.6.8.1 Description

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

4.6.8.2 Class Definition

The OccurrenceParticipant class is defined by the attribute values:

localNamespace

Value: icom_cal

localName

Value: OccurrenceParticipant

extendsFrom

Value: icom:Participant

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.6.8.3.

4.6.8.3 Property Definitions

The OccurrenceParticipant class inherits property definitions from super classes.

The OccurrenceParticipant class MUST have the property definitions:

icom_cal:participantStatus

Description: Status of an occurrence participant.

Required: False

Inherited: False

Property Type: icom_cal:OccurrenceParticipantStatus

Cardinality: Single

Updatability: Read Write

4.6.9 OccurrenceParticipantStatus

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

The OccurrenceParticipantStatus class is defined by the attribute values:

localNamespace

Value: icom_cal

5643 **localName**
5644 Value: OccurrenceParticipantStatus
5645
5646 **extendsFrom**
5647 Value:
5648
5649 **stereotype**
5650 Value: primary
5651
5652 **isEnumeration**
5653 Value: TRUE
5654
5655 **description**
5656 Value: An enumeration of the instances each of which expresses a participant's response status
5657 for an occurrence or occurrence series.
5658
5659 **instances**
5660 Value: <icom_cal:NeedsAction, icom_cal:Accepted, icom_cal:Declined, icom_cal:Tentative>
5661
5662 There are four occurrence participant's status defined by ICOM:
5663

- 5663 • **icom_cal:NeedsAction** to express that an attendee needs to act on an occurrence or
5664 occurrence series.
- 5665 • **icom_cal:Accepted** to express that an attendee accepted an occurrence or occurrence series.
- 5666 • **icom_cal:Declined** to express that an attendee declined an occurrence or occurrence series.
- 5667 • **icom_cal:Tentative** to express that an attendee is tentative about attending an occurrence or
5668 occurrence series.

5669

5670 4.7 FreeBusy Module

5671 4.7.1 FreeBusy

5672 4.7.1.1 Description

5673 A free busy object specifies the free time and busy time intervals of one or more participants.

5674 4.7.1.2 Class Definition

5675 The FreeBusy class is defined by the attribute values:

5676
5677 **localNamespace**
5678 Value: icom_cal
5679
5680 **localName**
5681 Value: FreeBusy
5682

```

5683      extendsFrom
5684          Value:
5685
5686      stereotype
5687          Value: primary
5688
5689      description
5690          Value: A free busy object specifies the free time and busy time intervals of one or more
5691          participants.
5692
5693      propertyDefinitions
5694          The values for this attribute are defined in Section 4.7.1.3.

```

5695 4.7.1.3 Property Definitions

5696 The FreeBusy class **MUST** have the property definitions:

5698	icom_cal:creationDate	
5699	Description:	Creation date and time of a free busy object.
5700	Required:	False
5701	Inherited:	False
5702	Property Type:	DateTime
5703	Cardinality:	Single
5704	Updatability:	Read Only
5705		
5706	icom_cal:startDate	
5707	Description:	Start date and time of a list of free busy intervals.
5708	Required:	False
5709	Inherited:	False
5710	Property Type:	DateTime
5711	Cardinality:	Single
5712	Updatability:	Read Only
5713		
5714	icom_cal:endDate	
5715	Description:	End date and time of a list of free busy intervals.
5716	Required:	False
5717	Inherited:	False
5718	Property Type:	DateTime
5719	Cardinality:	Single
5720	Updatability:	Read Only
5721		
5722	icom_cal:interval	
5723	Description:	A list of free busy intervals.
5724	Required:	False

5725	Inherited:	False
5726	Property Type:	icom_cal:FreeBusyInterval
5727	Cardinality:	Multi
5728	Updatability:	Read Only
5729		
5730	icom_cal:participant	
5731	Description:	A list of participants whose free busy intervals are merged.
5732	Required:	False
5733	Inherited:	False
5734	Property Type:	icom:Participant
5735	Cardinality:	Multi
5736	Updatability:	Read Only

5737

5738 4.7.2 FreeBusyInterval

5739 4.7.2.1 Description

- 5740 A free busy interval object specifies an interval of free or busy time.
- 5741 If a free busy type is icom_cal:Free, then a time interval is free for scheduling.
- 5742 If a free busy type is icom_cal:Busy, then a time interval is busy because one or more events have been
- 5743 scheduled for the interval.

5744 4.7.2.2 Class Definition

5745 The FreeBusyInterval class is defined by the attribute values:

5746

5747 **localNamespace**

5748 Value: icom_cal

5749

5750 **localName**

5751 Value: FreeBusy

5752

5753 **extendsFrom**

5754 Value:

5755

5756 **stereotype**

5757 Value: primary

5758

5759 **description**

5760 Value: A free busy interval object specifies an interval of free or busy time.

5761

5762 **propertyDefinitions**

5763 The values for this attribute are defined in Section 4.7.2.3

4.7.2.3 Property Definitions

The FreeBusyInterval class MUST have the property definitions:

icom_cal:startDate

Description:	Start date and time of a free busy interval.
Required:	False
Inherited:	False
Property Type:	DateTime
Cardinality:	Single
Updatability:	Read Only

icom_cal:endDate

Description:	End date and time of a free busy interval.
Required:	False
Inherited:	False
Property Type:	DateTime
Cardinality:	Single
Updatability:	Read Only

icom_cal:freeBusyType

Description:	A type of free busy interval.
Required:	False
Inherited:	False
Property Type:	icom_cal:FreeBusyType
Cardinality:	Single
Updatability:	Read Only

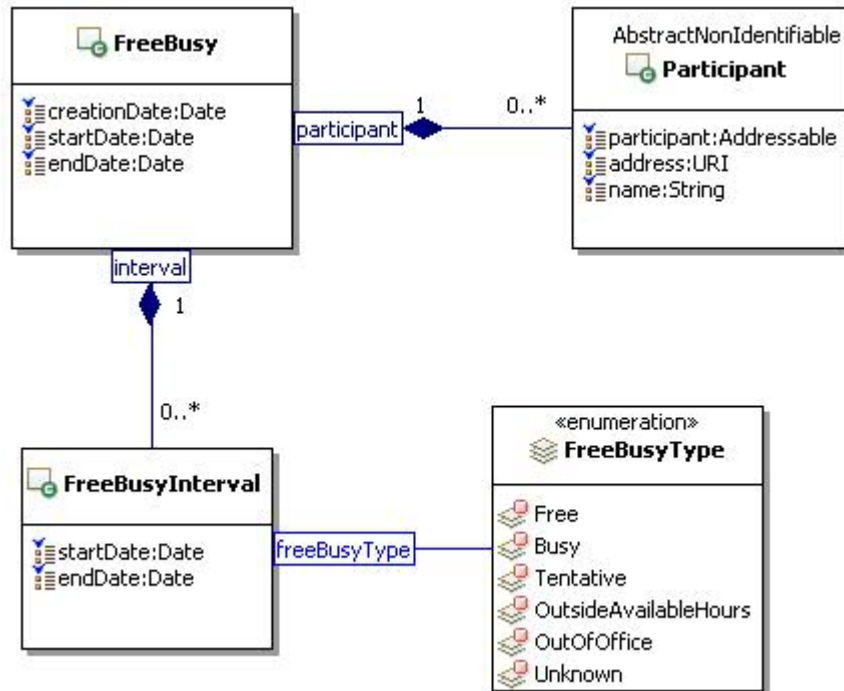


Figure 36 FreeBusy Class Diagram.

4.7.3 FreeBusyType

The FreeBusyType class is an enum class that enumerates the instances each of which expresses a type of free busy interval.

The FreeBusyType class is defined by the attribute values:

```

localNamespace
  Value: icom_cal

localName
  Value: FreeBusyType

extendsFrom
  Value:

stereotype
  Value: primary

isEnumeration
  Value: TRUE

description
  Value: An enumeration of the instances each of which expresses a type of free busy interval.
  
```

5817 **instances**

5818 Value: <icom_cal:Free, icom_cal:Busy, icom_cal:Tentative, icom_cal:OutsideAvailableHours,

5819 icom_cal:OutOfOffice, icom_cal:Unknown>

5820

5821 There are six free busy types defined by ICOM:

5822 • **icom_cal:Free** to express that a free busy interval is free.

5823 • **icom_cal:Busy** to express that a free busy interval is busy.

5824 • **icom_cal:Tentative** to express that a free busy interval is tentative.

5825 • **icom_cal:OutsideAvailableHours** to express that a free busy interval is outside available hours.

5826 • **icom_cal:OutOfOffice** to express that a free busy interval is coincides with out of office hours.

5827 • **icom_cal:Unknown** to express that a free busy interval is unknown.

5828

5829 4.8 TaskList Module

5830 4.8.1 TaskList

5831 4.8.1.1 Description

5832 A task list is a folder that contains task management artifacts.

5833 4.8.1.2 Class Definition

5834 The TaskList class is defined by the attribute values:

5835

5836 **localNamespace**

5837 Value: icom_cal

5838

5839 **localName**

5840 Value: TaskList

5841

5842 **extendsFrom**

5843 Value: icom:Folder

5844

5845 **stereotype**

5846 Value: primary

5847

5848 **description**

5849 Value: A task list is a folder that contains task management artifacts.

5850

5851 **propertyDefinitions**

5852 The values for this attribute are defined in 4.8.1.3.

5853 4.8.1.3 Property Definitions

5854 The TaskList class inherits property definitions from super classes.

5855 The TaskList class **MUST** have the property definitions:

5856		
5857	icom_cal:timeZone	
5858	Description:	Time zone of a task list.
5859	Required:	True
5860	Inherited:	False
5861	Property Type:	TimeZone
5862	Cardinality:	Single
5863	Updatability:	Read Write
5864		
5865	icom:element	
5866	Description:	Elements of a task list.
5867	Required:	False
5868	Inherited:	True
5869	Property Type:	icom_cal:Task
5870	Cardinality:	Multi
5871	Updatability:	Read Only
5872		

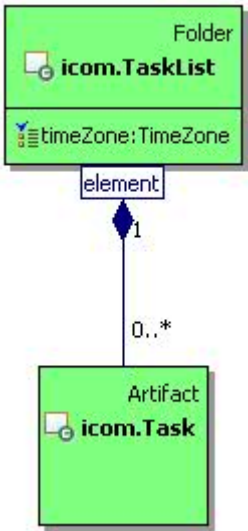


Figure 37 TaskList Class Diagram.

4.8.2 Task

4.8.2.1 Description

A task is an artifact that represents a task to do or a task assignment in a task list.

4.8.2.2 Class Definition

The Task class is defined by the attribute values:

localNamespace

Value: icom_cal

5884

5885 **localName**

5886 Value: Task

5887

5888 **extendsFrom**

5889 Value: icom:Artifact

5890

5891 **stereotype**

5892 Value: primary

5893

5894 **description**

5895 Value: A task is an artifact that represents a task to do or a task assignment in a task list.

5896

5897 **propertyDefinitions**

5898 The values for this attribute are defined in 4.8.2.3.

5899 **4.8.2.3 Property Definitions**

5900 The Task class inherits property definitions from super classes.

5901 The Task class MUST have the property definitions:

5902

5903 **icom_cal:startDate**

5904 Description:	Start date and time of a task.
5905 Required:	True
5906 Inherited:	False
5907 Property Type:	DateTime
5908 Cardinality:	Single
5909 Updatability:	On Create

5910

5911 **icom_cal:startDateResolution**

5912 Description:	Resolution of start date and time of a task.
5913 Required:	True
5914 Inherited:	False
5915 Property Type:	icom:DateTimeResolution
5916 Cardinality:	Single
5917 Updatability:	On Create

5918

5919 **icom_cal:dueDate**

5920 Description:	Due date and time of a task.
5921 Required:	True
5922 Inherited:	False
5923 Property Type:	DateTime
5924 Cardinality:	Single
5925 Updatability:	On Create

5926

5927 **icom_cal:dueDateResolution**

5928 Description: Resolution of due date and time of a task.

5929 Required: True

5930 Inherited: False

5931 Property Type: icom:DateTimeResolution

5932 Cardinality: Single

5933 Updatability: On Create

5934

5935 **icom:location**

5936 Description: Location of a task.

5937 Required: False

5938 Inherited: False

5939 Property Type: icom:Location

5940 Cardinality: Single

5941 Updatability: Read Write

5942

5943 **icom:organizer**

5944 Description: Organizer of a task.

5945 Required: True

5946 Inherited: False

5947 Property Type: icom:Participant

5948 Cardinality: Single

5949 Updatability: On Create

5950

5951 **icom_cal:taskStatus**

5952 Description: Status of a task.

5953 Required: True

5954 Inherited: False

5955 Property Type: icom_cal:TaskStatus

5956 Cardinality: Single

5957 Updatability: Read Write

5958

5959 **icom_cal:mode**

5960 Description: Mutability mode of a task.

5961 Required: False

5962 Inherited: False

5963 Property Type: icom_cal:TaskEditMode

5964 Cardinality: Single

5965 Updatability: Read Only

5966

5967 **icom_content:attachment**

5968	Description:	One or more simple content attachments in a task.
5969	Required:	False
5970	Inherited:	False
5971	Property Type:	icom_content:Attachment
5972	Cardinality:	Multi
5973	Updatability:	Read Write

5974

5975 **icom_cal:assignee**

5976	Description:	An assignee of a task.
5977	Required:	False
5978	Inherited:	False
5979	Property Type:	icom:Participant
5980	Cardinality:	Single
5981	Updatability:	Read Only

5982

5983 **icom_cal:assigneePriority**

5984	Description:	Priority for an assignee of a task.
5985	Required:	False
5986	Inherited:	False
5987	Property Type:	icom:Priority
5988	Cardinality:	Single
5989	Updatability:	Read Write

5990

5991 **icom_cal:assigneeParticipantStatus**

5992	Description:	Participation status for an assignee of a task.
5993	Required:	False
5994	Inherited:	False
5995	Property Type:	icom_cal:TaskParticipantStatus
5996	Cardinality:	Single
5997	Updatability:	Read Write

5998

5999 **icom_cal:assigneeCompletionDate**

6000	Description:	Completion date and time of a task.
6001	Required:	False
6002	Inherited:	False
6003	Property Type:	DateTime
6004	Cardinality:	Single
6005	Updatability:	Read Write

6006

6007 **icom_cal:assigneeCompletionDateResolution**

6008	Description:	Resolution of completion date and time of a task.
6009	Required:	False

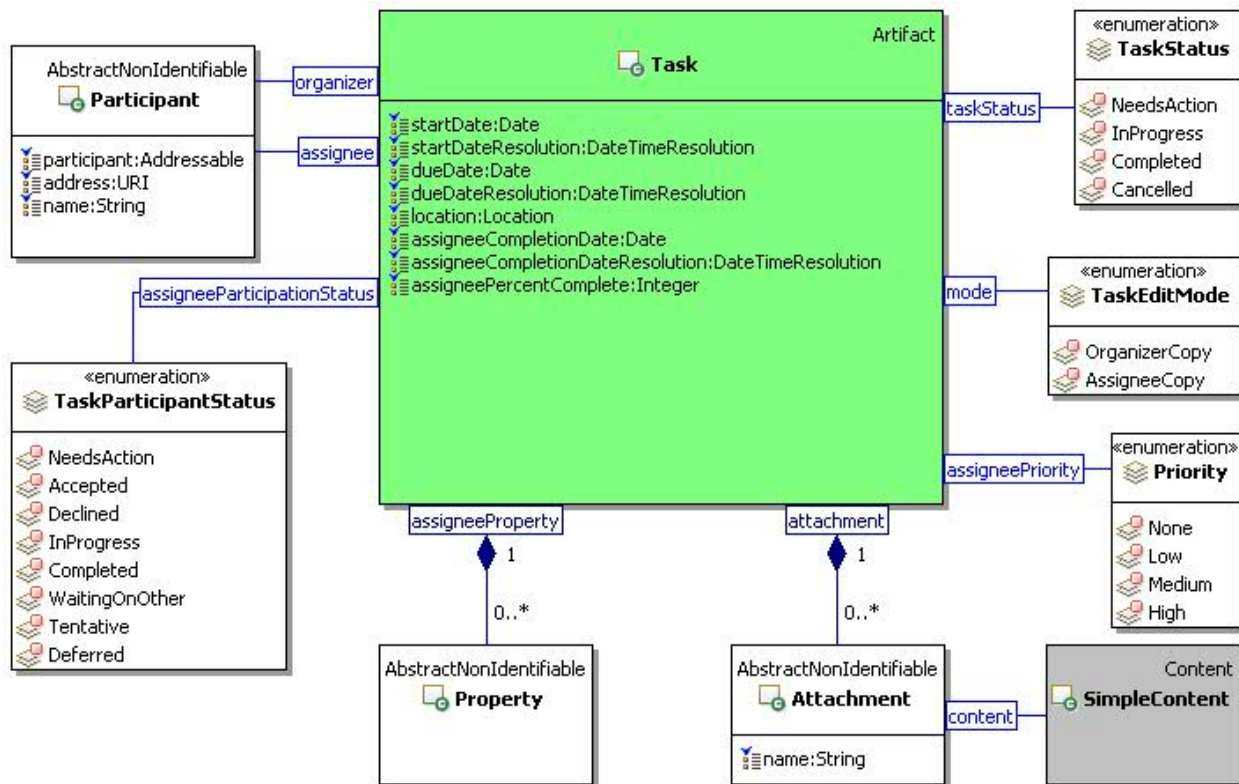
6010 Inherited: False
6011 Property Type: icom:DateTimeResolution
6012 Cardinality: Single
6013 Updatability: Read Write
6014

6015 **icom_cal:assigneePercentComplete**

6016 Description: Percentage of task completed.
6017 Required: False
6018 Inherited: False
6019 Property Type: Integer
6020 Cardinality: Single
6021 Updatability: Read Write
6022

6023 **icom_cal:assigneeProperty**

6024 Description: Extensible properties for an assignee of a task.
6025 Required: False
6026 Inherited: False
6027 Property Type: icom:Property
6028 Cardinality: Multi
6029 Updatability: Read Write
6030



6031
6032 *Figure 38 Task Class Diagram.*
6033

4.8.3 TaskStatus

The TaskStatus class is an enum class that enumerates the instances each of which expresses a status of task.

The TaskStatus class is defined by the attribute values:

localNamespace

Value: icom_cal

localName

Value: TaskStatus

extendsFrom

Value:

stereotype

Value: primary

isEnumeration

Value: TRUE

description

Value: An enumeration of the instances each of which expresses a status of a task.

instances

Value: <icom_cal:NeedsAction, icom_cal:InProgress, icom_cal:Completed, icom_cal:Cancelled>

There are four task status defined by ICOM:

- **icom_cal:NeedsAction** to express that a task needs action.
- **icom_cal:InProgress** to express that a task is in progress.
- **icom_cal:Completed** to express that a task is completed.
- **icom_cal:Cancelled** to express that a task is cancelled.

4.8.4 TaskEditMode

The TaskEditMode class is an enum class that enumerates the instances each of which expresses an editable mode of a task.

The TaskEditMode class is defined by the attribute values:

localNamespace

Value: icom_cal

localName

Value: TaskEditMode

6076

6077 **extendsFrom**

6078 Value:

6079

6080 **stereotype**

6081 Value: primary

6082

6083 **isEnumeration**

6084 Value: TRUE

6085

6086 **description**

6087 Value: An enumeration of the instances each of which expresses an editable mode of a task.

6088

6089 **instances**

6090 Value: <icom_cal:OrganizerCopy, icom_cal:AssigneeCopy>

6091

6092 There are two task editable modes defined by ICOM:

- 6093 • **icom_cal:OrganizerCopy**: to express that a task is a copy created by an organizer who may
- 6094 update the properties such as start time, due time, etc.
- 6095 • **icom_cal:AssigneeCopy**: to express that task is a copy delivered to an assignee who may only
- 6096 update the assignee properties such as assignee completion time, assignee participant status,
- 6097 assignee percent completed, etc .

6098

6099 **4.8.5 TaskParticipantStatus**

6100 The TaskParticipantStatus class is an enum class that enumerates the instances each of which

6101 expresses a participant's response status for a task.

6102 The TaskParticipantStatus class is defined by the attribute values:

6103

6104 **localNamespace**

6105 Value: icom_cal

6106

6107 **localName**

6108 Value: TaskParticipantStatus

6109

6110 **extendsFrom**

6111 Value:

6112

6113 **stereotype**

6114 Value: primary

6115

6116 **isEnumeration**

6117 Value: TRUE

6118

6119 **description**
6120 Value: An enumeration of the instances each of which expresses a participant's response status
6121 for a task.
6122
6123 **instances**
6124 Value: <icom_cal:NeedsAction, icom_cal:Accepted, icom_cal:Declined, icom_cal:InProgress,
6125 icom_cal:Completed, icom_cal:WaitingOnOther, icom_cal:Tentative, icom_cal:Deferred>
6126
6127 There are eight task participant's status defined by ICOM:
6128 • **icom_cal:NeedsAction** to express that an assignee needs to act on a task.
6129 • **icom_cal:Accepted** to express that an assignee accepted a task.
6130 • **icom_cal:Declined** to express that an assignee declined a task.
6131 • **icom_cal:InProgress** to express that a task is in progress.
6132 • **icom_cal:Completed** to express that a task is completed.
6133 • **icom_cal:WaitingOnOther** to express that an assignee is waiting on other.
6134 • **icom_cal: Tentative** to express that an assignee is tentative about a task status.
6135 • **icom_cal:Deferred** to express that an assignee deferred a task.
6136

6137 4.9 Forum Module

6138 4.9.1 Discussion

6139 4.9.1.1 Description

6140 A discussion is an item in a discussion container.

6141 4.9.1.2 Class Definition

6142 The Discussion class is a mixin class which defines the characteristics of entities that can be elements of
6143 a DiscussionContainer.

6144 The Discussion class is defined by the attribute values:

6145
6146 **localNamespace**
6147 Value: icom_forum
6148
6149 **localName**
6150 Value: Discussion
6151
6152 **extendsFrom**
6153 Value: icom:Item
6154
6155 **stereotype**
6156 Value: mixin
6157
6158 **description**

6159 Value: Discussion is a mixin class which defines the characteristics of entities that can be placed
6160 in a DiscussionContainer.

6161

6162 **propertyDefinitions**

6163 The values for this attribute are defined in Section 4.9.1.3.

6164 4.9.1.3 Property Definitions

6165 The Discussion class inherits property definitions from super classes.

6166 The Discussion class MUST have the property definitions:

6167

6168 **icom_forum:inReplyTo**

6169 Description: Another discussion object that a discussion object is replying
6170 to.

6171 Required: False

6172 Inherited: False

6173 Property Type: icom_forum:Discussion

6174 Cardinality: Single

6175 Updatability: Read Write

6176

6177 The Discussion class MAY include additional property definitions which are implementation-defined.

6178

6179 4.9.2 DiscussionContainer

6180 4.9.2.1 Description

6181 A discussion container is a container of discussions.

6182 4.9.2.2 Class Definition

6183 The DiscussionContainer class is a mixin class which defines the characteristics of entities that contain
6184 Discussion items.

6185 The DiscussionContainer class is defined by the attribute values:

6186

6187 **localNamespace**

6188 Value: icom_forum

6189

6190 **localName**

6191 Value: DiscussionContainer

6192

6193 **extendsFrom**

6194 Value: icom:Container

6195

6196 **stereotype**

6197 Value: mixin

6198

6199 **description**
6200 Value: DiscussionContainer is a mixin class which defines the characteristics of entities that
6201 contain Discussion items.

6202
6203 **propertyDefinitions**
6204 The values for this attribute are defined in Section 4.9.2.3.

6205 **4.9.2.3 Property Definitions**

6206 The DiscussionContainer class inherits property definitions from super classes.
6207 The DiscussionContainer class MUST have the property definitions:

6208
6209 **icom_forum:element**
6210 Description: Elements of a discussion container.
6211 Required: False
6212 Inherited: True
6213 Property Type: icom_forum:Discussion
6214 Cardinality: Multi
6215 Updatability: Read Only

6216
6217 The DiscussionContainer class MAY include additional property definitions which are implementation-
6218 defined.

6219

6220 **4.9.3 DiscussionMessage**

6221 **4.9.3.1 Description**

6222 A discussion message is a message in a forum discussion thread.

6223 **4.9.3.2 Class Definition**

6224 The DiscussionMessage class is defined by the attribute values:

6225
6226 **localNamespace**
6227 Value: icom_forum
6228
6229 **localName**
6230 Value: DiscussionMessage
6231
6232 **extendsFrom**
6233 Value: icom_msg:Message, icom_forum:Discussion
6234
6235 **stereotype**
6236 Value: primary
6237
6238 **description**

6239 Value: Discussion message is a message in a forum discussion thread.
6240
6241 **propertyDefinitions**
6242 The values for this attribute are defined in Section 4.9.3.3.

6243 **4.9.3.3 Property Definitions**

6244 The DiscussionMessage class inherits property definitions from super classes.
6245 The DiscussionMessage class MUST have the property definitions:
6246

6247 **icom_forum:inReplyTo**

6248 Description:	Another discussion message that a discussion message is replying to.
6249	
6250 Required:	False
6251 Inherited:	True
6252 Property Type:	icom_forum:DiscussionMessage
6253 Cardinality:	Single
6254 Updatability:	Read Write
6255	

6256 The DiscussionMessage class MAY include additional property definitions which are implementation-defined.
6257
6258

6276

6277 **stereotype**

6278 Value: primary

6279

6280 **description**

6281 Value: A forum is a folder that contains sub-forums, topics, and announcements.

6282

6283 **propertyDefinitions**

6284 The values for this attribute are defined in Section 4.9.4.3.

6285 **4.9.4.3 Property Definitions**

6286 The Forum class inherits property definitions from super classes.

6287 The Forum class MUST have the property definitions:

6288

6289 **icom_forum:lastPost**

6290 Description: The last posted discussion in a forum.

6291 Required: False

6292 Inherited: False

6293 Property Type: icom_forum:Discussion

6294 Cardinality: Single

6295 Updatability: Read Only

6296

6297 **icom_forum:forum**

6298 Description: Sub-forums of a forum.

6299 Required: False

6300 Inherited: False

6301 Property Type: icom_forum:Forum

6302 Cardinality: Multi

6303 Updatability: Read Only

6304

6305 **icom_forum:topic**

6306 Description: Topics of a forum.

6307 Required: False

6308 Inherited: False

6309 Property Type: icom_forum:Topic

6310 Cardinality: Multi

6311 Updatability: Read Only

6312

6313 **icom_forum:announcement**

6314 Description: Announcements of a forum.

6315 Required: False

6316 Inherited: False

6317 Property Type: icom_forum:Announcement

6318 Cardinality: Multi
6319 Updatability: Read Only

6320
6321 The Forum class MAY include additional property definitions which are implementation-defined.
6322

6323 4.9.5 Topic

6324 4.9.5.1 Description

6325 A topic is a folder that contains a conversation among forum participants. The discussions in a topic may
6326 be sorted in chronological order or threaded by reply.

6327 4.9.5.2 Class Definition

6328 The Topic class is defined by the attribute values:

6329
6330 **localNamespace**
6331 Value: icom_forum
6332
6333 **localName**
6334 Value: Topic
6335
6336 **extendsFrom**
6337 Value: icom:Folder, icom_forum:DiscussionContainer
6338
6339 **stereotype**
6340 Value: primary
6341
6342 **description**
6343 Value: A topic is a folder that contains discussion threads.
6344
6345 **propertyDefinitions**
6346 The values for this attribute are defined in Section 4.9.5.3.

6347 4.9.5.3 Property Definitions

6348 The Topic class inherits property definitions from super classes.

6349 The Topic class MUST have the property definitions:

6350
6351 **icom:element**
6352 Description: Elements of a topic.
6353 Required: False
6354 Inherited: True
6355 Property Type: icom_forum:Discussion
6356 Cardinality: Multi
6357 Updatability: Read Only

6358
6359
6360
6361
6362
6363
6364
6365
6366
6367
6368
6369
6370
6371
6372
6373
6374
6375
6376

6377

6378
6379
6380

6381
6382
6383
6384
6385
6386
6387
6388
6389
6390
6391
6392
6393
6394
6395
6396
6397
6398

icom_forum:firstPost

Description: The first posted discussion in a topic.
Required: False
Inherited: False
Property Type: icom_forum:Discussion
Cardinality: Single
Updatability: Read Only

icom_forum:lastPost

Description: The last posted discussion in a topic.
Required: False
Inherited: False
Property Type: icom_forum:Discussion
Cardinality: Single
Updatability: Read Only

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

4.9.6 Announcement

4.9.6.1 Description

An announcement is a special topic for time-sensitive discussion posts that are valid for a specified period of time, depending on activation and expiration times.

4.9.6.2 Class Definition

The Announcement class is defined by the attribute values:

localNamespace

Value: icom_forum

localName

Value: Announcement

extendsFrom

Value: icom_forum:Topic

stereotype

Value: primary

description

Value: An announcement is a special topic for discussions that are valid for a specified period of time.

6399

6400 **propertyDefinitions**

6401 The values for this attribute are defined in Section 4.9.6.3.

6402 **4.9.6.3 Property Definitions**

6403 The Announcement class inherits property definitions from super classes.

6404 The Announcement class MUST have the property definitions:

6405

6406 **icom_forum:activationDate**

6407 Description: Date and time when an announcement becomes active.

6408 Required: False

6409 Inherited: False

6410 Property Type: DateTime

6411 Cardinality: Single

6412 Updatability: Read Write

6413

6414 **icom_forum:expirationDate**

6415 Description: Date and time when an announcement expires.

6416 Required: False

6417 Inherited: False

6418 Property Type: DateTime

6419 Cardinality: Single

6420 Updatability: Read Write

6421

6422 **icom_forum:announcementStatus**

6423 Description: Status of an announcement.

6424 Required: True

6425 Inherited: False

6426 Property Type: icom_forum:AnnouncementStatus

6427 Cardinality: Single

6428 Updatability: Read Write

6429

6430 The Announcement class MAY include additional property definitions which are implementation-defined.

6431

6432 **4.9.7 AnnouncementStatus**

6433 The AnnouncementStatus class is an enum class that enumerates the instances each of which expresses
6434 a status of announcement.

6435 The AnnouncementStatus class is defined by the attribute values:

6436

6437 **localNamespace**

6438 Value: icom_forum

6439

6440 **localName**
6441 Value: AnnouncementStatus
6442
6443 **extendsFrom**
6444 Value:
6445
6446 **stereotype**
6447 Value: primary
6448
6449 **isEnumeration**
6450 Value: TRUE
6451
6452 **description**
6453 Value: An enumeration of the instances each of which expresses a status of announcement.
6454
6455 **instances**
6456 Value: <icom_forum:Pending, icom_forum:Active, icom_forum:Expired>
6457
6458 There are three announcement status defined by ICOM:
6459

- **icom_forum:Pending** to express that an announcement is pending.
- **icom_forum:Active** to express that an announcement is active.
- **icom_forum:Expired** to express that an announcement is expired.

6462

6463 **4.10 Conference Module**

6464 **4.10.1 Conference**

6465 **4.10.1.1 Description**

6466 A conference is a folder that represents a durable context for conference sessions.
6467 It contains conference metadata, settings, and transcripts.

6468 **4.10.1.2 Class Definition**

6469 The Conference class is defined by the attribute values:

6470
6471 **localNamespace**
6472 Value: icom_conf
6473
6474 **localName**
6475 Value: Conference
6476
6477 **extendsFrom**
6478 Value: icom:Folder

6479
6480 **stereotype**
6481 Value: primary
6482
6483 **description**
6484 Value: A conference is a folder that represents a durable context for online conference sessions.
6485
6486 **propertyDefinitions**
6487 The values for this attribute are defined in Section 4.10.1.3.

6488 **4.10.1.3 Property Definitions**

6489 The Conference class inherits property definitions from super classes.
6490 The Conference class MUST have the property definitions:

6491
6492 **icom_conf:organizer**
6493 Description: Organizer of a conference.
6494 Required: False
6495 Inherited: False
6496 Property Type: icom_cond:Participant
6497 Cardinality: Single
6498 Updatability: On Create
6499
6500 **icom_conf:conferenceType**
6501 Description: Type of a conference.
6502 Required: False
6503 Inherited: False
6504 Property Type: icom_conf:ConferenceType
6505 Cardinality: Single
6506 Updatability: Read Write
6507
6508 **icom_conf:conferenceState**
6509 Description: Session state of a conference.
6510 Required: False
6511 Inherited: False
6512 Property Type: icom_conf:ConferenceState
6513 Cardinality: Single
6514 Updatability: Read Only
6515
6516 **icom_conf:runningSession**
6517 Description: Current session of a conference.
6518 Required: False
6519 Inherited: False
6520 Property Type: icom_conf:ConferenceSession

6521	Cardinality:	Single
6522	Updatability:	Read Only
6523		
6524	icom_conf:conferenceSetting	
6525	Description:	Configurable settings of a conference.
6526	Required:	False
6527	Inherited:	False
6528	Property Type:	icom_conf:ConferenceSetting
6529	Cardinality:	Single
6530	Updatability:	Read Only
6531		
6532	icom_conf:transcript	
6533	Description:	Transcripts from ended sessions of a conference.
6534	Required:	False
6535	Inherited:	False
6536	Property Type:	icom_doc:Document
6537	Cardinality:	Multi
6538	Updatability:	Read Write
6539		
6540	icom_conf:scheduledStartTime	
6541	Description:	Scheduled start time of the current or next session of a
6542		conference.
6543	Required:	False
6544	Inherited:	False
6545	Property Type:	Date
6546	Cardinality:	Single
6547	Updatability:	Read Write
6548		
6549	icom_conf:scheduledEndTime	
6550	Description:	Scheduled end time of the current or next session of a
6551		conference.
6552	Required:	False
6553	Inherited:	False
6554	Property Type:	Date
6555	Cardinality:	Single
6556	Updatability:	Read Write
6557		
6558	The Conference class MAY include additional property definitions which are implementation-defined.	
6559		

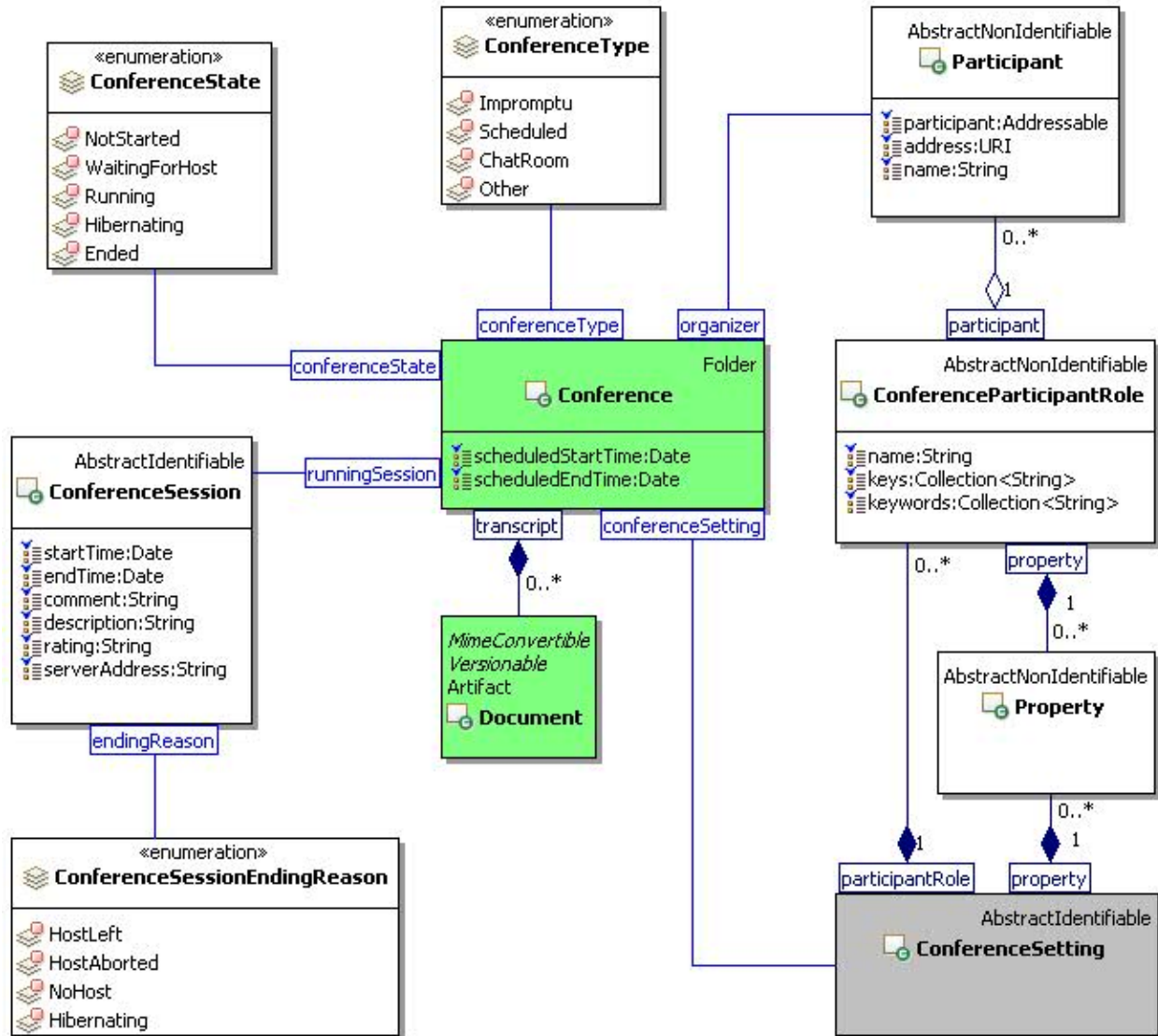


Figure 40 Conference Class Diagram.

4.10.2 ConferenceType

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

The ConferenceType class is defined by the attribute values:

```

localNamespace
    Value: icom_conf

localName
    Value: ConferenceType

extendsFrom
    Value:

```

6576

6577 **stereotype**

6578 Value: primary

6579

6580 **isEnumeration**

6581 Value: TRUE

6582

6583 **description**

6584 Value: An enumeration of the instances each of which expresses a type of a conference.

6585

6586 **instances**

6587 Value: <icom_conf:Impromptu, icom_conf:Scheduled, icom_conf:ChatRoom, icom_conf:Other>

6588

6589 There are four conference types defined by ICOM:

6590 • **icom_conf:Impromptu** to express that a conference session is started impromptu .

6591 • **icom_conf:Scheduled** to express that a conference session is scheduled.

6592 • **icom_conf:ChatRoom** to express that a conference is used for a chat room.

6593 • **icom_conf:Other** to express that a conference is of other type.

6594

6595 **4.10.3 ConferenceState**

6596 The ConferenceState class is an enum class that enumerates the instances each of which expresses a

6597 session state of a conference.

6598 The ConferenceState class is defined by the attribute values:

6599

6600 **localNamespace**

6601 Value: icom_conf

6602

6603 **localName**

6604 Value: ConferenceState

6605

6606 **extendsFrom**

6607 Value:

6608

6609 **stereotype**

6610 Value: primary

6611

6612 **isEnumeration**

6613 Value: TRUE

6614

6615 **description**

6616 Value: An enumeration of the instances each of which expresses a session state of a

6617 conference.

6618
6619 **instances**
6620 Value: <icom_conf:NotStarted, icom_conf:WaitingForHost, icom_conf:Running,
6621 icom_conf:Hibernating, icom_conf:Ended>
6622

6623 There are five conference session states defined by ICOM:

- 6624 • **icom_conf:NotStarted** to express that a conference session is not started .
6625 • **icom_conf:WaitingForHost** to express that a conference session is waiting for a host.
6626 • **icom_conf:Running** to express that a conference session is running.
6627 • **icom_conf:Hibernating** to express that a conference session is hibernating.
6628 • **icom_conf:Ended** to express that a conference session is ended.
6629

6630 **4.10.4 ConferenceSession**

6631 **4.10.4.1 Description**

6632 A conference session represents the metadata for a runtime session of a conference.

6633 **4.10.4.2 Class Definition**

6634 The ConferenceSession class is defined by the attribute values:

6635
6636 **localNamespace**
6637 Value: icom_conf
6638
6639 **localName**
6640 Value: ConferenceSession
6641
6642 **extendsFrom**
6643 Value: icom:Identifiable
6644
6645 **stereotype**
6646 Value: primary
6647
6648 **description**
6649 Value: A conference session represents the metadata for a runtime session of a conference.
6650
6651 **propertyDefinitions**
6652 The values for this attribute are defined in Section 4.10.4.3.

6653 **4.10.4.3 Property Definitions**

6654 The ConferenceSession class inherits property definitions from super classes.

6655 The ConferenceSession class **MUST** have the property definitions:
6656

6657	icom_conf:startTime	
6658	Description:	Start time of a conference session.
6659	Required:	False
6660	Inherited:	False
6661	Property Type:	Date
6662	Cardinality:	Single
6663	Updatability:	Read Only
6664		
6665	icom_conf:endTime	
6666	Description:	End time of a conference session.
6667	Required:	False
6668	Inherited:	False
6669	Property Type:	Date
6670	Cardinality:	Single
6671	Updatability:	Read Only
6672		
6673	icom_conf:comment	
6674	Description:	Comment on a conference session.
6675	Required:	False
6676	Inherited:	False
6677	Property Type:	String
6678	Cardinality:	Single
6679	Updatability:	Read Write
6680		
6681	icom_conf:description	
6682	Description:	Description of a conference session.
6683	Required:	False
6684	Inherited:	False
6685	Property Type:	String
6686	Cardinality:	Single
6687	Updatability:	Read Write
6688		
6689	icom_conf:rating	
6690	Description:	Rating of a conference session.
6691	Required:	False
6692	Inherited:	False
6693	Property Type:	String
6694	Cardinality:	Single
6695	Updatability:	Read Write
6696		
6697	icom_conf:serverAddress	
6698	Description:	Address of a server that hosts a conference session.

6699 Required: False
6700 Inherited: False
6701 Property Type: String
6702 Cardinality: Single
6703 Updatability: Read Only

6704

6705 **icom_conf:endingReason**

6706 Description: Reason for ending a conference session.
6707 Required: False
6708 Inherited: False
6709 Property Type: icom_conf:ConferenceSessionEndingReason
6710 Cardinality: Single
6711 Updatability: Read Only

6712

6713 The ConferenceSession class MAY include additional property definitions which are implementation-
6714 defined.

6715

6716 **4.10.5 ConferenceSessionEndingReason**

6717 The ConferenceSessionEndingReason class is an enum class that enumerates the instances each of
6718 which expresses a reason for ending a conference session.

6719 The ConferenceSessionEndingReason class is defined by the attribute values:

6720

6721 **localNamespace**

6722 Value: icom_conf

6723

6724 **localName**

6725 Value: ConferenceSessionEndingReason

6726

6727 **extendsFrom**

6728 Value:

6729

6730 **stereotype**

6731 Value: primary

6732

6733 **isEnumeration**

6734 Value: TRUE

6735

6736 **description**

6737 Value: An enumeration of the instances each of which expresses a reason for ending a
6738 conference session.

6739

6740 **instances**

6741 Value: <icom_conf:HostLeft, icom_conf:HostAborted, icom_conf:NoHost, icom_conf:Hibernating>

6742

6743 There are four conference session states defined by ICOM:

- 6744 • **icom_conf:HostLeft** to express that a conference session ended after the host left.
- 6745 • **icom_conf:HostAborted** to express that a conference session ended after the host aborted it.
- 6746 • **icom_conf:NoHost** to express that a conference session ended due to no one hosting.
- 6747 • **icom_conf:Hibernating** to express that a conference session is hibernating.

6748

6749 4.10.6 ConferenceSetting

6750 4.10.6.1 Description

6751 A conference setting represents configuration settings for sessions of a conference.

6752 4.10.6.2 Class Definition

6753 The ConferenceSetting class is defined by the attribute values:

6754

6755 **localNamespace**

6756 Value: icom_conf

6757

6758 **localName**

6759 Value: ConferenceSetting

6760

6761 **extendsFrom**

6762 Value:

6763

6764 **stereotype**

6765 Value: primary

6766

6767 **description**

6768 Value: A conference setting represents configuration settings for sessions of a conference.

6769

6770 **propertyDefinitions**

6771 The values for this attribute are defined in Section 4.10.6.3.

6772 4.10.6.3 Property Definitions

6773 The ConferenceSetting class inherits property definitions from super classes.

6774 The ConferenceSetting class MUST have the property definitions:

6775

6776 **icom_conf:participantRole**

6777 Description: Role settings for conference participants.

6778 Required: False

6779 Inherited: False

6780 Property Type: icom_conf:ConferenceParticipantRole

6781 Cardinality: Multi
6782 Updatability: Read Write

6783

6784 **icom_conf:property**

6785 Description: Configurable properties for a conference.
6786 Required: False
6787 Inherited: False
6788 Property Type: icom:property
6789 Cardinality: Multi
6790 Updatability: Read Write

6791

6792 The ConferenceSetting class MAY include additional property definitions which are implementation-
6793 defined.

6794

6795 **4.10.7 ConferenceParticipantRole**

6796 **4.10.7.1 Description**

6797 A conference participant role contains roles settings for a conference.

6798 **4.10.7.2 Class Definition**

6799 The ConferenceParticipantRole class is defined by the attribute values:

6800

6801 **localNamespace**

6802 Value: icom_conf

6803

6804 **localName**

6805 Value: ConferenceParticipantRole

6806

6807 **extendsFrom**

6808 Value:

6809

6810 **stereotype**

6811 Value: primary

6812

6813 **description**

6814 Value: A conference participant role contains roles settings for a conference.

6815

6816 **propertyDefinitions**

6817 The values for this attribute are defined in Section 4.10.7.3.

6818 **4.10.7.3 Property Definitions**

6819 The ConferenceParticipantRole class MUST have the property definitions:

6820

6821	icom_conf:name	
6822	Description:	Name of a role setting in a conference.
6823	Required:	False
6824	Inherited:	False
6825	Property Type:	String
6826	Cardinality:	Single
6827	Updatability:	Read Write
6828		
6829	icom_conf:key	
6830	Description:	One or more sign on keys to activate a role setting.
6831	Required:	False
6832	Inherited:	False
6833	Property Type:	String
6834	Cardinality:	Multi
6835	Updatability:	Read Write
6836		
6837	icom_conf:keyword	
6838	Description:	One or more key words to activate a role setting.
6839	Required:	False
6840	Inherited:	False
6841	Property Type:	String
6842	Cardinality:	Multi
6843	Updatability:	Read Write
6844		
6845	icom_conf:participant	
6846	Description:	One or more participants in a role setting.
6847	Required:	False
6848	Inherited:	False
6849	Property Type:	icom:Participant
6850	Cardinality:	Multi
6851	Updatability:	Read Write
6852		
6853	icom_conf:property	
6854	Description:	Configurable properties for a role setting.
6855	Required:	False
6856	Inherited:	False
6857	Property Type:	icom:Property
6858	Cardinality:	Multi
6859	Updatability:	Read Write
6860		

6861 The ConferenceParticipantRole class MAY include additional property definitions which are
6862 implementation-defined.

6863

5 Conformance

The ICOM specification does not presume a particular software architecture or arrangement for use of the ICOM model.

Conformance to the ICOM model is defined by the roles played in any software architecture or arrangement; however a particular implementation chooses to allocate these roles and accompanying responsibilities.

Conformance to the ICOM model is defined as:

1. Service provider role: An ICOM service provider shall conform to all mandatory and optional statements of the core ICOM model as defined in Section 3 of this standard.
2. Service provider role: An ICOM service provider shall conform to all mandatory and optional statements for one or more extension models as defined in Section 4 of this standard, but no application fulfilling the service provider role is required to conform to any particular extension model.

Note: Implementations may choose one or more extension modules to support in an ICOM application.

3. ICOM producer role: An ICOM producer, which is an application that produces artifacts to be managed by an ICOM service provider, shall produce artifacts that conform to both mandatory and optional statements in Section 3 and 4 of this standard, for any artifact produced.

Note: Implementations in the ICOM producer role are not required to produce any particular ICOM artifacts, but any which are produced to be managed by an ICOM service provider, must conform to this standard.

4. ICOM consumer role: An ICOM consumer, which is an application that consumes artifacts managed by an ICOM service provider, shall accept ICOM artifacts that conform to both mandatory and optional statement in Section 3 and 4 of this standard.

A. Acknowledgements

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:

Rafiul Ahad, Oracle Corporation
Eric S. Chan, Oracle Corporation
Martin Chapman, Oracle Corporation
Scott Conroy, Individual
Stefan Decker, Digital Enterprise Research Institute (DERI)
Laura Dragan, Digital Enterprise Research Institute (DERI)
Patrick Durusau, Individual
Siegfried Handschuh, Digital Enterprise Research Institute (DERI)
Deirdre Lee, Digital Enterprise Research Institute (DERI)
Marc Pallot, Individual
Chancellor Pascale, Johns Hopkins University Applied Physics Laboratory
Vassilios Peristeras, Digital Enterprise Research Institute (DERI)
Peter Saint-Andre, Cisco Systems, Inc.
Peter Yim, CIM Engineering Inc. (CIM3)
Ramesh Vasudevan, Oracle Corporation

C. Revision History

Revision	Date	Editor	Changes Made
cd01	Aug 23, 2010	Eric S. Chan	Created
cd01a	Nov 1, 2010	Eric, Patrick	Co-edited for TC Meeting Review
cd01b	Nov 9, 2010	Eric, Patrick	Co-edited
cd01c	Nov 22, 2010	Eric	Added UML diagrams
cd01d	Feb 9, 2011	Eric	Added Presence, Calendar, Task, FreeBusy
cd01e	Feb 15, 2011	Eric	Added AddressBook and Contact
cd01f	Feb 18, 2011	Eric	Added User, ResourceActor, InstantMessage, WikiPage, Conference
cd01g	Feb 22, 2011	Eric	Changed Conference accessor to participant
cd01h	Mar 6, 2011	Eric	Moved Instant Message Feed from Presence Module to Message Module. Change Occurrence-Conference relations from "many-to-one" to "many-to-many" relationship. Add references in the introduction to other standards and technologies that ICOM reuses. First pass typo corrections across the document.
CD1	March 09, 2011	Eric S. Chan Patrick Durusau	Candidate for Committee Specification Draft: added conformance clauses