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

Committee Specification Draft 01 / Public Review Draft 01

16 March 2011

Specification URIs:

This Version:

http://docs.oasis-open.org/icom/icom-ics/v1.0/csprd01/icom-ics-v1.0-csprd01.doc (Authoritative)
http://docs.oasis-open.org/icom/icom-ics/v1.0/csprd01/icom-ics-v1.0-csprd01.html
http://docs.oasis-open.org/icom/icom-ics/v1.0/csprd01/icom-ics-v1.0-csprd01.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 / Public Review Draft 01. http://docs.oasis-open.org/icom/icom-ics/v1.0/csprd01/icom-ics-v1.0-csprd01.doc.

Notices

Table of Contents

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, *Content Management Interoperability Services (CMIS) Version 1.0,* May 2010. (http://docs.oasis-open.org/cmis/CMIS/v1.0/os/cmis-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.

- **ReadWrite:** The property value can be modified.
- **OnCreate:** The property value MUST only be update-able during the creation (a create operation) of an object.

inherited Boolean

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

The values of inherited attribute are:

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

required Boolean

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

The values of required attribute are:

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

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

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

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

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

Each value of choices attribute is an instance of **property-choice-type** that specifies a display name and a value to be stored in a property when selected.

If the value of cardinatity attribute is **Single** and the value of openChoice attribute is **FALSE**, then a property value MUST be at most one of the values listed in choices attribute.

If the value of cardinatity attribute is **Single** and the value of openChoice attribute is **TRUE**, then a property value MAY be one of the values listed in choices attribute.

If the value of cardinatity attribute is **Multi** and the value of openChoice attribute is **FALSE**, then a property value MUST be zero, one, or more than one of the values listed in choices attribute.

If the value of cardinatity attribute is **Multi** and the value of openChoice attribute is **TRUE**, then a property value MAY be zero, one, or more than one of the values listed in choices attribute.

If choices attribute is "not set", then a property value MAY be an instance of the **property-type** specified by the propertyType attribute of a property definition.

`openChoice` Boolean

The `openChoice` attribute specifies whether the value of a property must be listed in `choices` attribute. It is applicable only when `choices` attribute is set.

The values of `openChoice` attribute are:

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

`defaultValue` **property-type**

The `defaultValue` attribute specifies a value that a service provider MUST set for a property if a value is not provided by application when an object is created.

If no default value is specified and application creates an object of this class without setting a value for a property of this property definition, a service provider MUST attempt to store a “not set” state for the property value. If this occurs for a property that is defined to be required, then a service provider MUST throw an exception.

The value of the `defaultValue` attribute is an instance of the **property-type** specified by the `propertyType` attribute of a property definition.

`minValue` Integer | Decimal

The minimum value allowed for a property. It is applicable only when the `propertyType` attribute of a property definition specifies the property types Integer or Decimal.

`maxValue` Integer | Decimal

The maximum value allowed for a property. It is applicable only when the `propertyType` attribute of a property definition specifies the property types Integer or Decimal.

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

`displayName` String

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

`value` **property-type**

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

2.4 Namespaces

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

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

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

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

3 Core Model

3.1 Main Branch

3.1.1 UML Diagram of Entity and Top-Level Subclasses

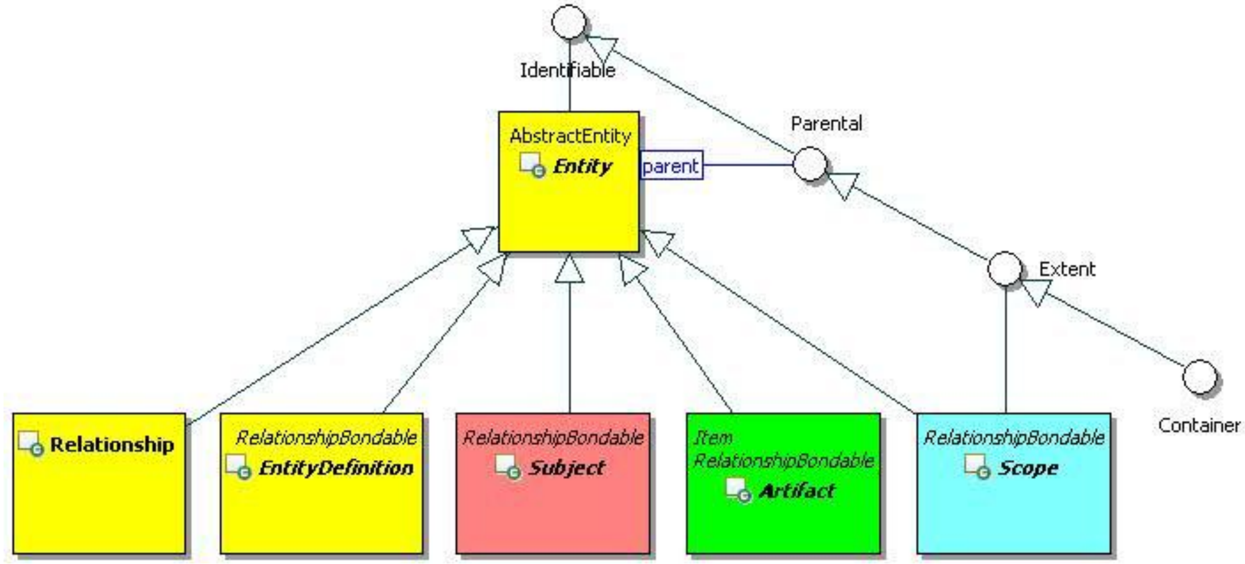

Figure 1 Entity and Top-Level Subclasses

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 mixin 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:

stereotype

Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.1.2.3.

3.1.2.3 Property Definitions

The Identifiable class MUST have the property definitions:

icom:objectId

Description:	A persistent identifier of an object.
Required:	False
Inherited:	False
Property Type:	ID
Cardinality:	Single
Updatability:	Read Only

icom:changeToken

Description:	An opaque token used for optimistic locking & concurrency checking.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Only

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

3.1.3 Parental

3.1.3.1 Description

A parental object may be a parent of other objects.

3.1.3.2 Class Definition

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

The Parental class is defined by the attribute values:

localNamespace

Value: icom

localName

Value: Parental

extendsFrom

Value: icom:Identifiable

stereotype

Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.1.3.3.

3.1.3.3 Property Definitions

The Parental class inherits property definitions from super classes.

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

3.1.4 Extent

3.1.4.1 Description

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

3.1.4.2 Class Definition

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

The Extent class is defined by the attribute values:

localNamespace

Value: icom

localName

Value: Extent

extendsFrom

Value: icom:Parental

stereotype

Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.1.4.3.

3.1.4.3 Property Definitions

The Extent class inherits property definitions from super classes.

The Extent class MUST have the property definitions:

icom:parent

Description:	Parent of an extent.
Required:	False
Inherited:	False
Property Type:	icom:Extent
Cardinality:	Single
Updatability:	Read Only

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

3.1.5 Entity

3.1.5.1 Description

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

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

3.1.5.2 Class Definition

The Entity class is defined by the attribute values:

localNamespace

Value: icom

localName

Value: Entity

extendsFrom

Value: icom:Identifiable

stereotype

Value: primary

isAbstract

Value: TRUE

description

Value: An entity is an object that has an immutable id and can be individually access controlled.

propertyDefinitions

The values for this attribute are defined in Section 3.1.5.3.

3.1.5.3 Property Definitions

The Entity class inherits property definitions from super classes.

The Entity class MUST have the property definitions:

icom:name

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

icom:createdBy

Description:	An actor who creates an entity.
Required:	False
Inherited:	False
Property Type:	icom:Actor
Cardinality:	Single
Updatability:	Read Only

icom:creationDate

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

icom:lastModifiedBy

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

Cardinality:	Single
Updatability:	Read Only

icom:lastModificationDate

Description:	Date and time when an entity is last modified.
Required:	False
Inherited:	False
Property Type:	DateTime
Cardinality:	Single
Updatability:	Read Only

icom_ac:owner

Description:	A subject who owns an entity.
Required:	True
Inherited:	False
Property Type:	icom_ac:Owner
Cardinality:	Single
Updatability:	Read Write

icom:parent

Description:	A parental entity which contains an entity.
Required:	False
Inherited:	False
Property Type:	icom:Parental
Cardinality:	Single
Updatability:	Read Only

icom_meta:attachedMarker

Description:	Zero or more markers applied on an entity.
Required:	False
Inherited:	False
Property Type:	icom_meta:Marker
Cardinality:	Multi
Updatability:	Read Only

icom_meta:categoryApplication

Description:	Zero or more category applications on an entity.
Required:	False
Inherited:	False
Property Type:	icom_meta:CategoryApplication
Cardinality:	Multi
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.

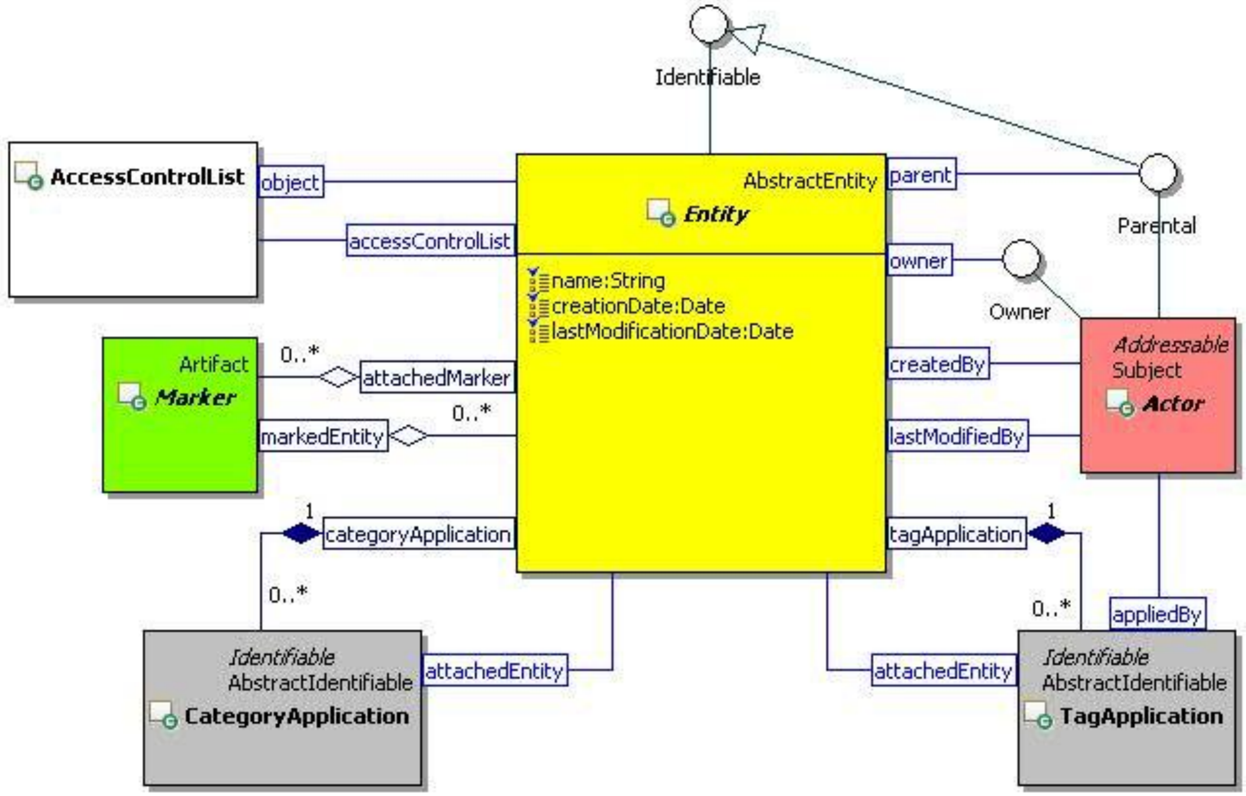

Figure 2 Entity Class Diagram

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

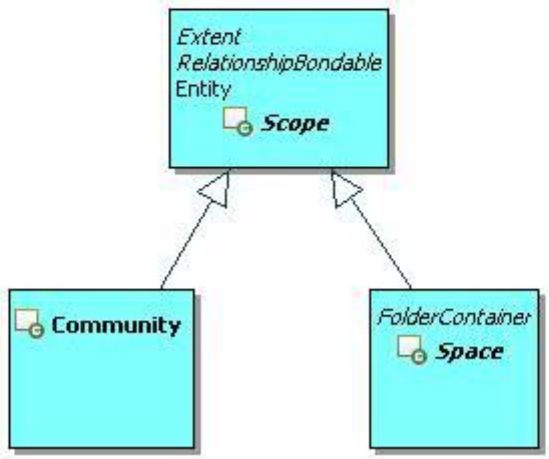

Figure 3 Scope Branch

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.

`propertyDefinitions`

The values for this attribute are defined in Section 3.2.2.3.

3.2.2.3 Property Definitions

The Scope class inherits property definitions from super classes.

The Scope class MUST have the property definitions:

icom:description

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

icom:parent

Description:	A community which contains a scope.
Required:	False
Inherited:	True
Property Type:	icom:Community
Cardinality:	Single
Updatability:	Read Only

icom:roleDefinition

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

icom:role

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

icom:group

Description:	Zero or more groups defined in a scope.
Required:	False

Inherited:	False
Property Type:	icom:Group
Cardinality:	Multi
Updatability:	Read Only

icom:memberGroup

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

icom_meta:relationship

Description:	Zero or more relationships associated with a scope.
Required:	False
Inherited:	False
Property Type:	icom_meta:Relationship
Cardinality:	Multi
Updatability:	Read Only

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

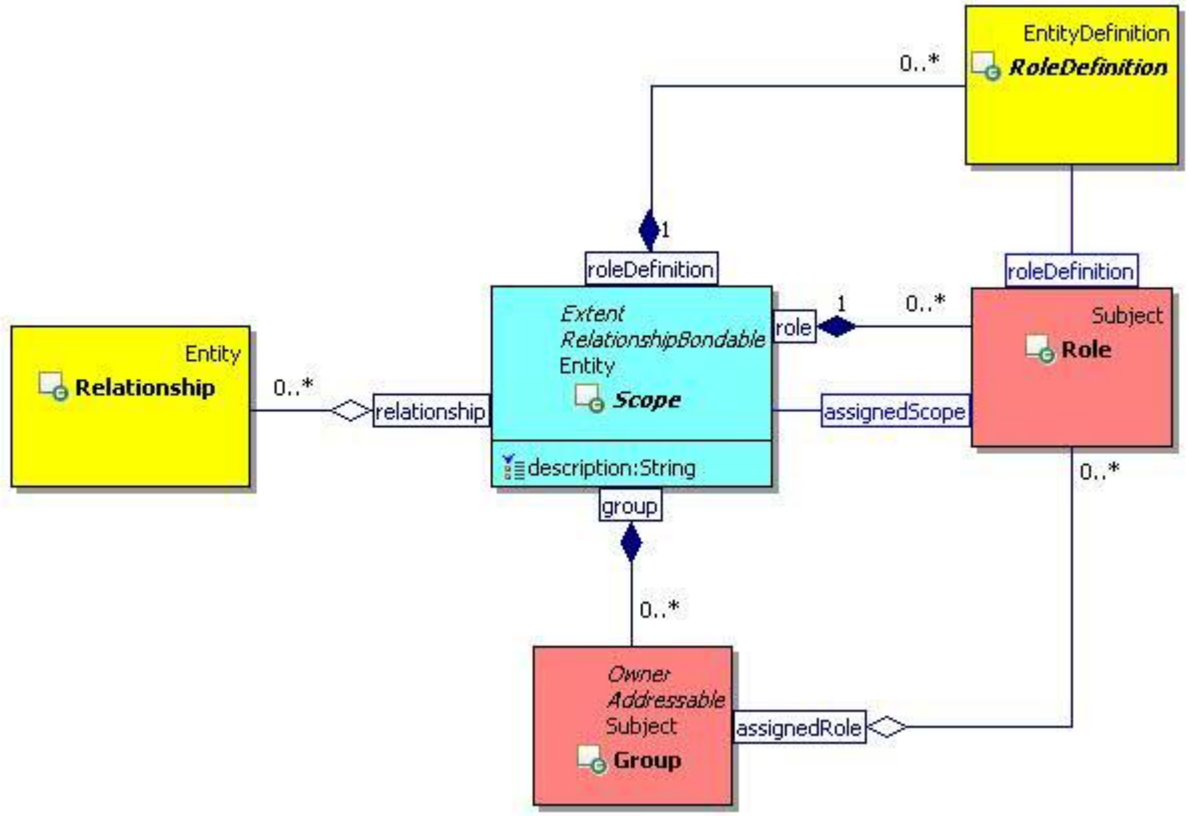

Figure 4 Scope Class Diagram

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

Value: primary

`description`

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

`propertyDefinitions`

The values for this attribute are defined in Section 3.2.3.3.

3.2.3.3 Property Definitions

The Community class inherits property definitions from super classes.

The Community class MUST have the property definitions:

icom:community

Description:	Sub-communities of a community.
Required:	False
Inherited:	False
Property Type:	icom:Community
Cardinality:	Multi
Updatability:	Read Only

icom:space

Description:	Spaces of a community.
Required:	False
Inherited:	False
Property Type:	icom:Space
Cardinality:	Multi
Updatability:	Read Only

icom:actor

Description:	Managed actors of a community, i.e. actors whose parent community is this community.
Required:	False
Inherited:	False
Property Type:	icom:Actor
Cardinality:	Multi
Updatability:	Read Only

icom:memberActor

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

Property Type:	icom:Actor
Cardinality:	Multi
Updatability:	Read Only

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

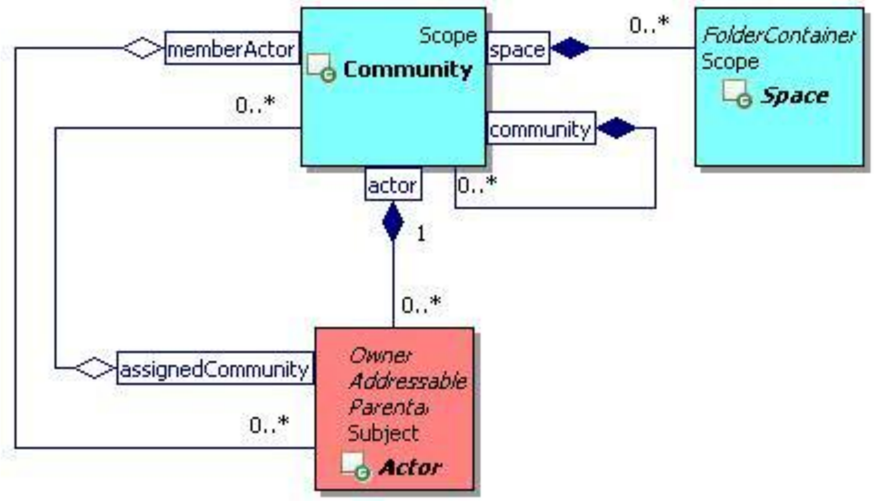

Figure 5 Community Class Diagram

3.2.4 Space

3.2.4.1 Description

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

3.2.4.2 Class Definition

The Space class is defined by the attribute values:

`localNamespace`

Value: icom

`localName`

Value: Space

`extendsFrom`

Value: icom:Scope, icom:FolderContainer

`stereotype`

Value: primary

`description`

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

`propertyDefinitions`

The values for this attribute are defined in Section 3.2.4.3.

3.2.4.3 Property Definitions

The Space class inherits property definitions from super classes.

The Space class MUST have the property 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.

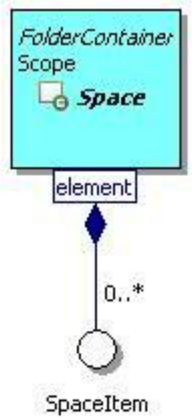

Figure 6 Space Class Diagram

3.3 Subject Branch

3.3.1 UML Diagram of Subject and Top-Level Subclasses

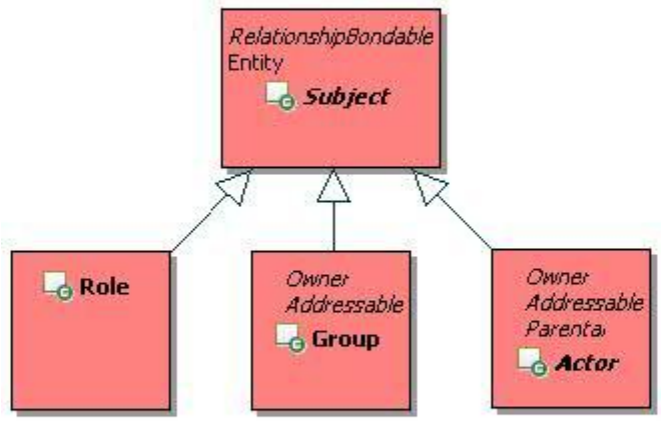

Figure 7 Subject Branch

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.

`propertyDefinitions`

The values for this attribute are defined in Section 3.3.2.3.

3.3.2.3 Property Definitions

The Subject class inherits property definitions from super classes.

The Subject class MUST have the property definitions:

icom:description

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

icom:parent

Description:	A scope which contains a subject.
Required:	False
Inherited:	True
Property Type:	icom:Scope
Cardinality:	Single
Updatability:	Read Only

icom_meta:relationship

Description:	Zero or more relationships associated with a subject.
Required:	False
Inherited:	False
Property Type:	icom_meta:Relationship
Cardinality:	Multi
Updatability:	Read Only

icom_meta:property

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

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

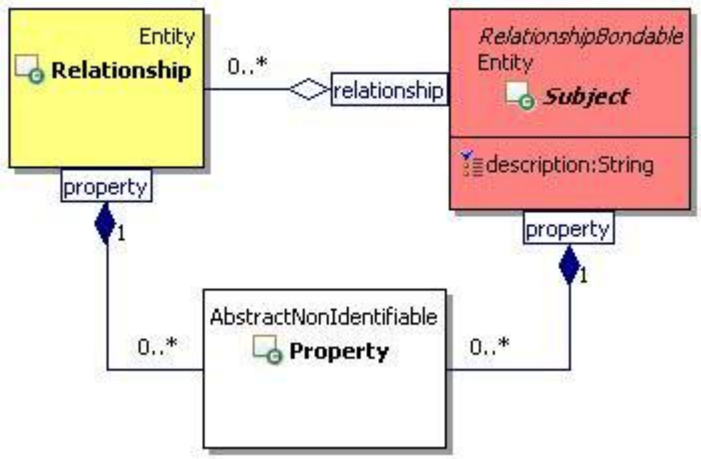

Figure 8 Subject Class Diagram

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.

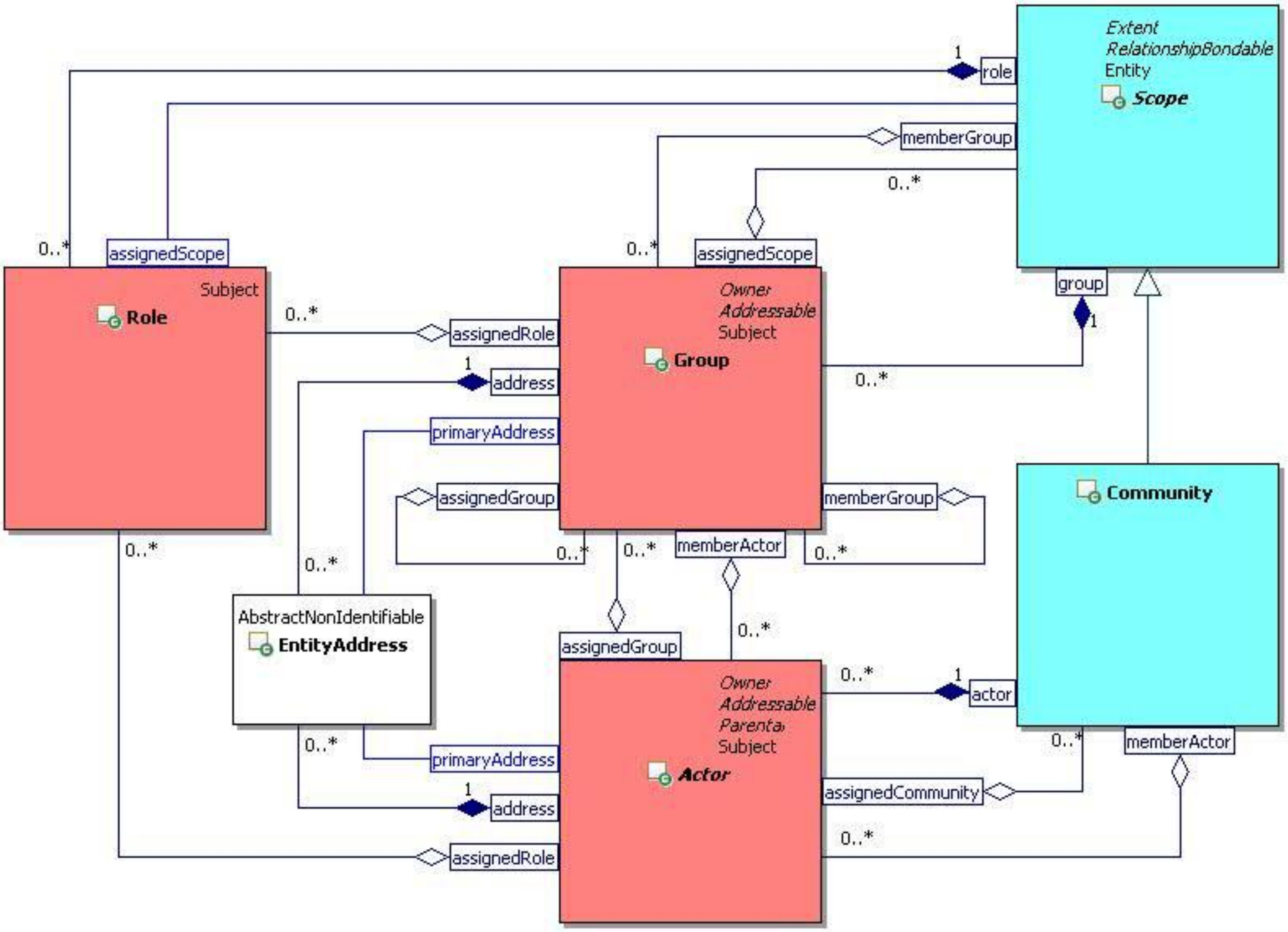

Figure 9 Group and Actor Class Diagram

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

localName

Value: Actor

extendsFrom

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

stereotype

Value: primary

isAbstract

Value: TRUE

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.3.4.3.

3.3.4.3 Property Definitions

The Actor class inherits property definitions from super classes.

The Actor class MUST have the property definitions:

icom:parent

Description:	A community which contains an actor.
Required:	False
Inherited:	True
Property Type:	icom:Community
Cardinality:	Single
Updatability:	Read Only

icom:assignedRole

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

icom:assignedGroup

Description:	Groups to which an actor is assigned.
Required:	False
Inherited:	False

Property Type: icom:Group
Cardinality: Multi
Updatability: Read Write

icom:assignedCommunity

Description: Communities to which an actor is assigned.
Required: False
Inherited: False
Property Type: icom:Community
Cardinality: Multi
Updatability: Read Write

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

3.3.5 User

3.3.5.1 Description

A user is an actor who is also a person.

A user has a personal space.

3.3.5.2 Class Definition

The User class is defined by the attribute values:

`localNamespace`

Value: icom

`localName`

Value: User

`extendsFrom`

Value: icom:Actor, icom_card:Person

`stereotype`

Value: primary

`description`

Value: A user is an actor who is also a person.

`propertyDefinitions`

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.

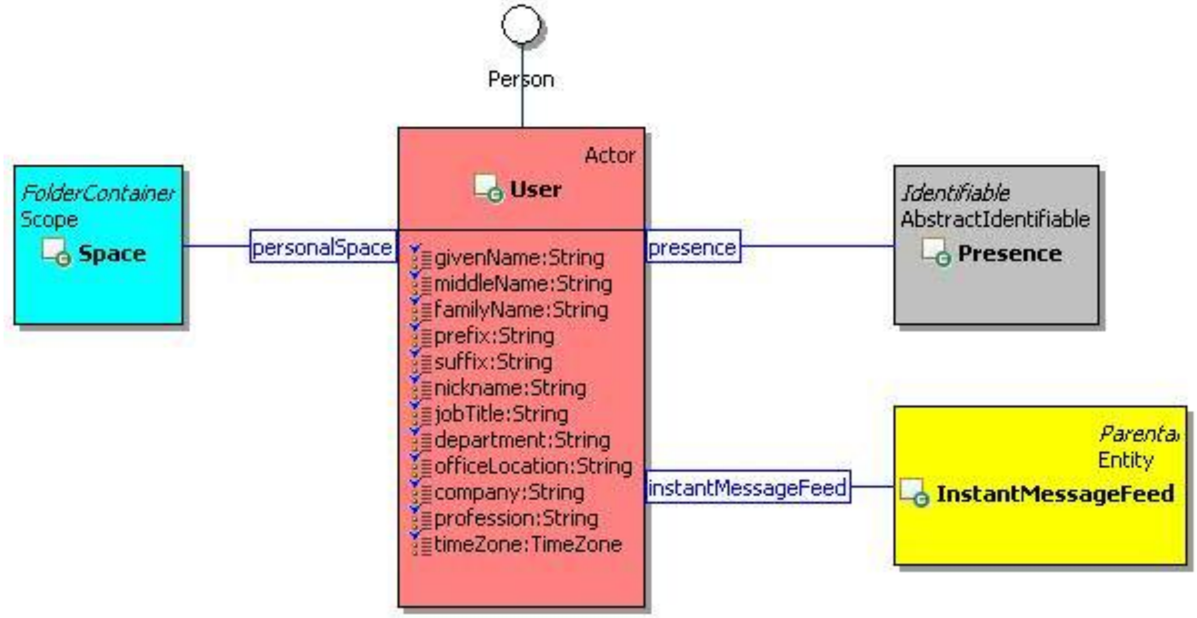

Figure 10 User Class Diagram.

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`

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

`propertyDefinitions`

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

3.3.6.3 Property Definitions

The ResourceActor class inherits property definitions from super classes.

The ResourceActor class MUST have the property definitions:

icom:resourceSpace

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

icom:description

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

icom:location

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

icom:capacity

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

icom:resourceType

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

icom:bookingRule

Description:	Resource booking rule.
Required:	False
Inherited:	False
Property Type:	icom:ResourceBookingRule
Cardinality:	Single
Updatability:	Read Write

icom:bookingApprover

Description:	One or more users who approve the booking of a resource.
Required:	False
Inherited:	False
Property Type:	icom:User
Cardinality:	Single
Updatability:	Read Write

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

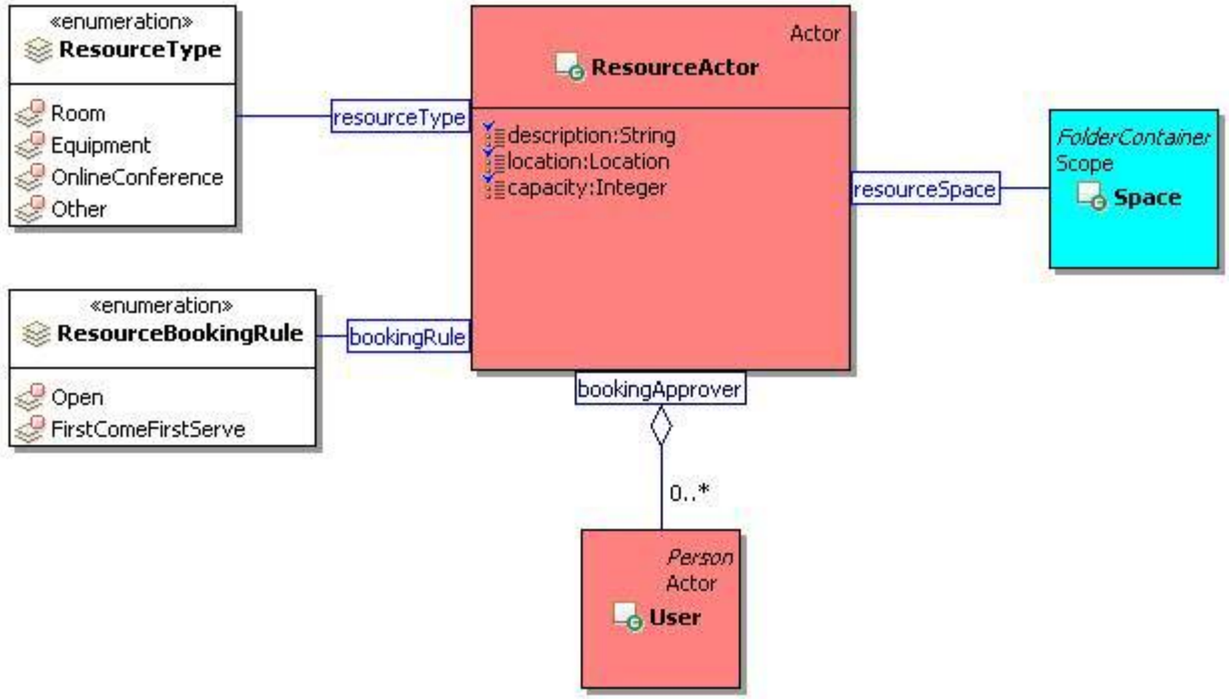

Figure 11 ResourceActor Class Diagram.

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.

instances

Value: <icom:Room, icom:Equipment, icom:OnlineConference, icom:Other>

The following resource types are defined by ICOM:

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

3.3.8 ResourceBookingRule

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

The ResourceBookingRule class is defined by the attribute values:

localNamespace

Value: icom

localName

Value: ResourceBookingRule

extendsFrom

Value:

stereotype

Value: primary

isEnumeration

Value: TRUE

description

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

instances

Value: <icom:Open, icom:FirstComeFirstServe>

The following resource booking rules are defined by ICOM:

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

3.4 Artifact Branch

3.4.1 UML Diagram of Artifact and Top-Level Subclasses

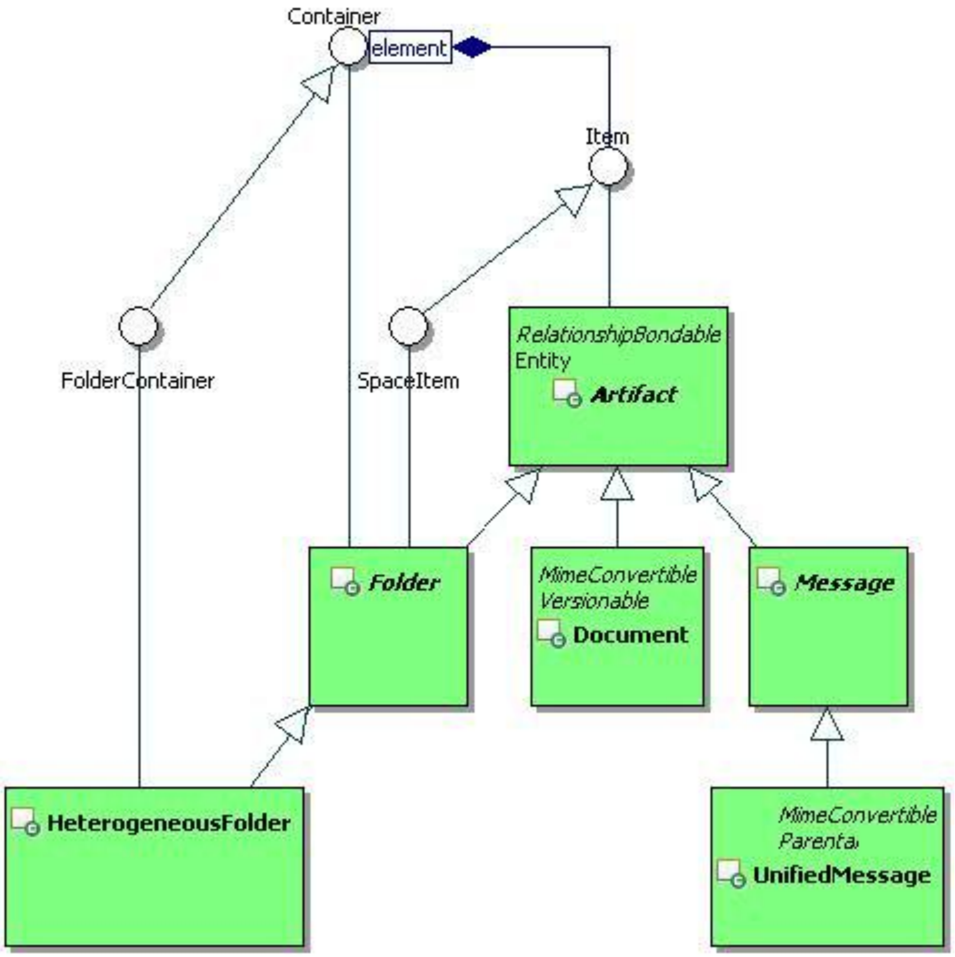

Figure 12 Artifact Branch

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

localName

Value: Item

extendsFrom

Value: icom:Identifiable

stereotype

Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.4.2.3.

3.4.2.3 Property Definitions

The Item class inherits property definitions from super classes.

The Item class MUST have the property definition:

icom:parent

Description:	A parent container of an item.
Required:	False
Inherited:	True
Property Type:	icom:Container
Cardinality:	Single
Updatability:	Read Only

The Item class MAY have the optional property definition:

icom:container

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

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

3.4.3 SpaceItem

3.4.3.1 Description

A space item is an element of a space.

3.4.3.2 Class Definition

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

The SpaceItem class is defined by the attribute values:

`localNamespace`

Value: icom

`localName`

Value: SpaceItem

`extendsFrom`

Value: icom:Item

`stereotype`

Value: mixin

`description`

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

`propertyDefinitions`

The values for this attribute are defined in Section 3.4.3.3.

3.4.3.3 Property Definitions

The SpaceItem class inherits property definitions from super classes.

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

3.4.4 Container

3.4.4.1 Description

A container is an extent that contains items.

3.4.4.2 Class Definition

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

The Container class is defined by the attribute values:

`localNamespace`

Value: icom

localName

Value: Container

extendsFrom

Value: icom:Extent

stereotype

Value: mixin

description

Value: A container is an extent that contains items.

propertyDefinitions

The values for this attribute are defined in Section 3.4.4.3.

3.4.4.3 Property Definitions

The Container class inherits property definitions from super classes.

The Container class MUST have the property definitions:

icom:element

Description:	Elements of a container, i.e. items whose parent container is the container or whose containers include the container.
Required:	False
Inherited:	False
Property Type:	icom:Item
Cardinality:	Multi
Updatability:	Read Only

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

3.4.5 FolderContainer

3.4.5.1 Description

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

3.4.5.2 Class Definition

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

The FolderContainer class is defined by the attribute values:

localNamespace

Value: icom

localName

Value: FolderContainer

extendsFrom

Value: icom:Container

stereotype

Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.4.5.3.

3.4.5.3 Property Definitions

The FolderContainer class inherits property definitions from super classes.

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

3.4.6 Artifact

3.4.6.1 Description

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

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

3.4.6.2 Class Definition

The Artifact class is defined by the attribute values:

localNamespace

Value: icom

localName

Value: Artifact

extendsFrom

Value: icom:Entity, icom:Item, icom_meta:RelationshipBondable

Optional Value: icom:SpaceItem

stereotype

Value: primary

isAbstract

Value: TRUE

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.4.6.3.

3.4.6.3 Property Definitions

The Artifact class inherits property definitions from super classes.

The Artifact class MUST have the property definitions:

icom:description

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

icom:userCreationDate

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

icom:userLastModificationDate

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

icom_meta:property

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

icom_meta:relationship

Description:	Zero or more relationships associated with an artifact.
Required:	False
Inherited:	False
Property Type:	icom_meta:Relationship
Cardinality:	Multi
Updatability:	Read Only

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

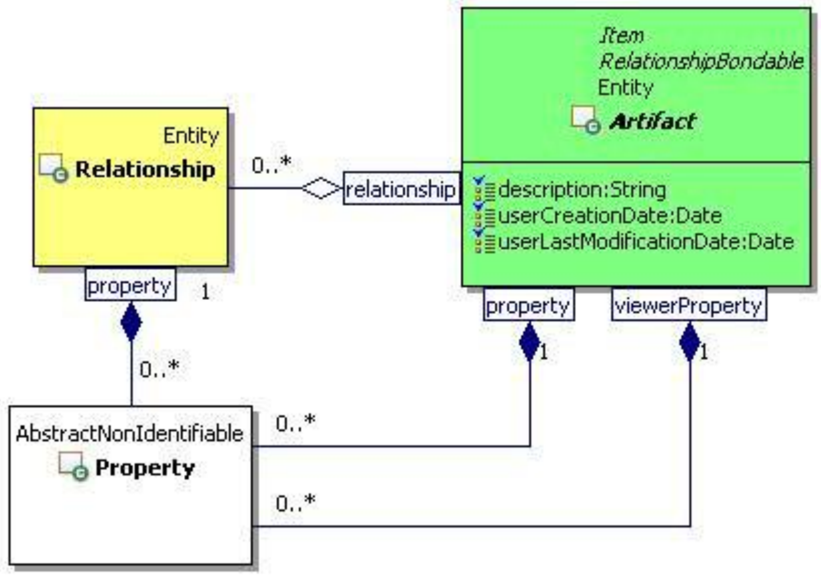

Figure 13 Artifact Class Diagram

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:

localNamespace

Value: icom

localName

Value: Folder

extendsFrom

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

stereotype

Value: primary

isAbstract

Value: TRUE

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.4.7.3.

3.4.7.3 Property Definitions

The Folder class inherits property definitions from super classes.

The Folder class MUST have the property definitions:

icom:parent

Description:	A parent container of a folder.
Required:	False
Inherited:	True
Property Type:	icom:FolderContainer
Cardinality:	Single
Updatability:	Read Only

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

3.4.8 HeterogeneousFolder

3.4.8.1 Description

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

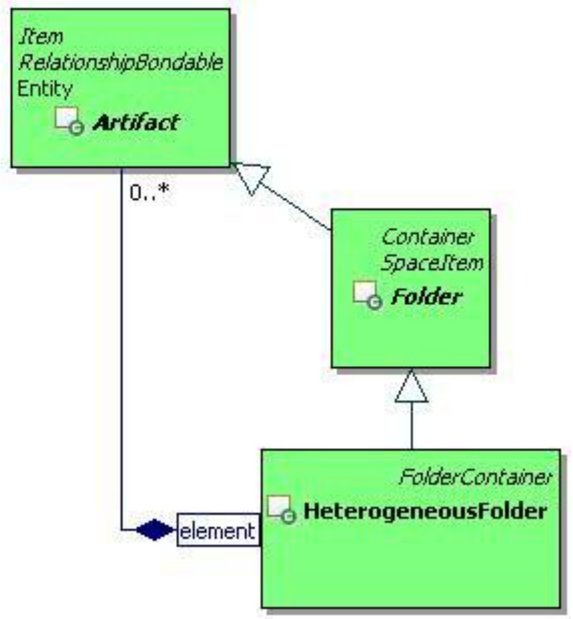

Figure 14 Heterogeneous Folder Class Diagram

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

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.5.1.3.

3.5.1.3 Property Definitions

The Accessor class inherits property definitions from super classes.

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

3.5.2 Owner

3.5.2.1 Description

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

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

3.5.2.2 Class Definition

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

The Owner class is defined by the attribute values:

localNamespace

Value: icom_ac

localName

Value: Owner

extendsFrom

Value: icom_ac:Accessor

stereotype

Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.5.2.3.

3.5.2.3 Property Definitions

The Owner class inherits property definitions from super classes.

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

3.5.3 RoleDefinition

3.5.3.1 Description

A role definition is a named set of privileges.

3.5.3.2 Class Definition

The RoleDefinition class is defined by the attribute values:

localNamespace

Value: icom_ac

localName

Value: RoleDefinition

extendsFrom

Value: icom:EntityDefinition

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.5.3.3.

3.5.3.3 Property Definitions

The RoleDefinition class inherits property definitions from super classes.

The RoleDefinition class MUST have the property definitions:

icom_ac:privilege

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

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

3.5.4 Role

3.5.4.1 Description

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

3.5.4.2 Class Definition

The Role class is defined by the 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

Property Type:	icom:Scope
Cardinality:	Single
Updatability:	Read Write

icom_ac:memberAccessor

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

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

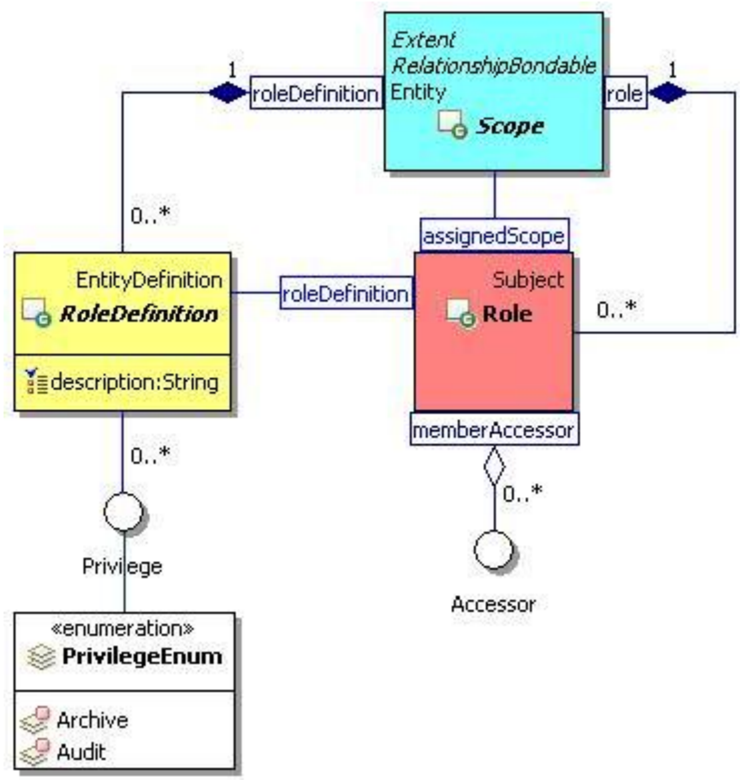

Figure 15 Role Definition and Role Class Diagram

3.5.5 Privilege

3.5.5.1 Description

A privilege is an access right granted through roles.

3.5.5.2 Class Definition

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

The Privilege class is defined by the attribute values:

`localNamespace`

Value: icom_ac

`localName`

Value: Privilege

`extendsFrom`

Value:

`stereotype`

Value: mixin

`description`

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

`propertyDefinitions`

The values for this attribute are defined in Section 3.5.5.3.

3.5.5.3 Property Definitions

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

3.5.6 PrivilegeEnum

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

The PrivilegeEnum class is defined by the attribute values:

`localNamespace`

Value: icom_ac

`localName`

Value: PrivilegeEnum

`extendsFrom`

Value: icom_ac:Privilege

`stereotype`

Value: primary

`isEnumeration`

Value: TRUE

description

Value: An enumeration of instances each of which expresses a privilege that can be assigned to a role.

instances

Value: <icom_ac:Archive, icom_ac:Audit>

The following privileges are defined by ICOM:

- **icom_ac:Archive** to express a right to archive contents in a scope.
- **icom_ac:Audit** to express a right to audit activities in a scope.

3.5.7 AccessControlList

3.5.7.1 Description

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

3.5.7.2 Class Definition

The AccessControlList class is defined by the attribute values:

localNamespace

Value: icom_ac

localName

Value: AccessControlList

extendsFrom

Value:

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.5.7.3.

3.5.7.3 Property Definitions

The AccessControlList class MUST have the property definitions:

icom_ac:object

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

icom_ac:accessControlEntry

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

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

3.5.8 AccessControlEntry

3.5.8.1 Description

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

3.5.8.2 Class Definition

The AccessControlEntry class is defined by the attribute values:

localNamespace

Value: icom_ac

localName

Value: AccessControlEntry

extendsFrom

Value:

stereotype

Value: primary

description

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

Value: icom_ac

localName

Value: AccessType

extendsFrom

Value:

stereotype

Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.5.9.2.

3.5.9.2 Property Definitions

The AccessType class inherits property definitions from super classes.

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

3.5.10 AccessTypeEnum

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

The AccessTypeEnum class is defined by the attribute values:

localNamespace

Value: icom_ac

localName

Value: AccessTypeEnum

extendsFrom

Value: icom_ac:AccessType

stereotype

Value: primary

isEnumeration

Value: TRUE

`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.

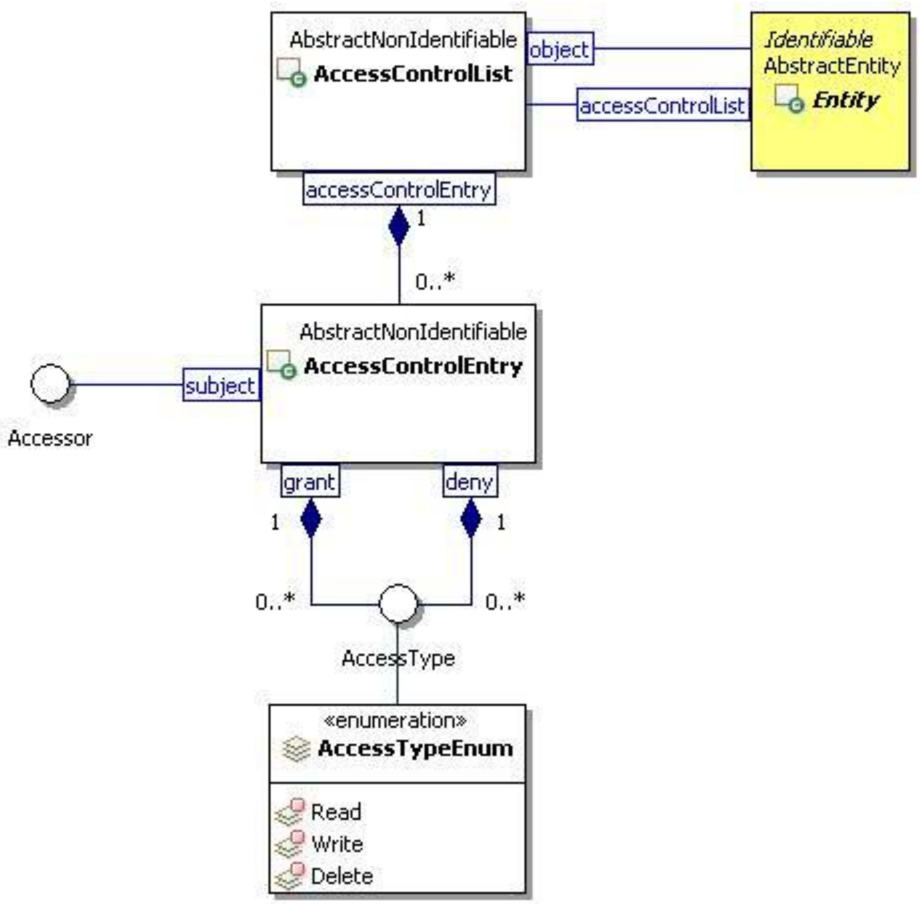

Figure 16 Access Control List Class Diagram

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.

Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

icom_meta:propertyType

Description:	Type of a property.
Required:	True
Inherited:	False
Property Type:	icom_meta:PropertyType
Cardinality:	Single
Updatability:	On Create
Choices:	PropertyChoiceType
Open Choice:	False

icom_meta:defaultValue

Description:	A default value for a property.
Required:	False
Inherited:	False
Property Type:	**property-type**
Cardinality:	Single
Updatability:	Read Write

icom_meta:choices

Description:	An explicit ordered set of single values allowed for a property.
Required:	False
Inherited:	False
Property Type:	List< icom_meta:PropertyChoiceType>
Cardinality:	Single
Updatability:	Read Write

icom_meta:cardinality

Description:	Cardinality of a property specifying whether the property can have “zero or one” or “zero or more” values.
Required:	True
Inherited:	False
Property Type:	icom_meta:Cardinaility
Cardinality:	Single
Updatability:	On Create

icom_meta:minValue

Description:	Minimum value for an integer or decimal property.

Required:	False
Inherited:	False
Property Type:	Integer | Decimal
Cardinality:	Single
Updatability:	Read Write

icom_meta:maxValue

Description:	Maximum value for an integer or decimal property.
Required:	False
Inherited:	False
Property Type:	Integer | Decimal
Cardinality:	Single
Updatability:	Read Write

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

3.6.2 Property

3.6.2.1 Description

The property holds a property value.

3.6.2.2 Class Definition

The Property class is defined by the attribute values:

localNamespace

Value: icom_meta

localName

Value: Property

extendsFrom

Value:

stereotype

Value: primary

description

Value: A property value.

propertyDefinitions

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.

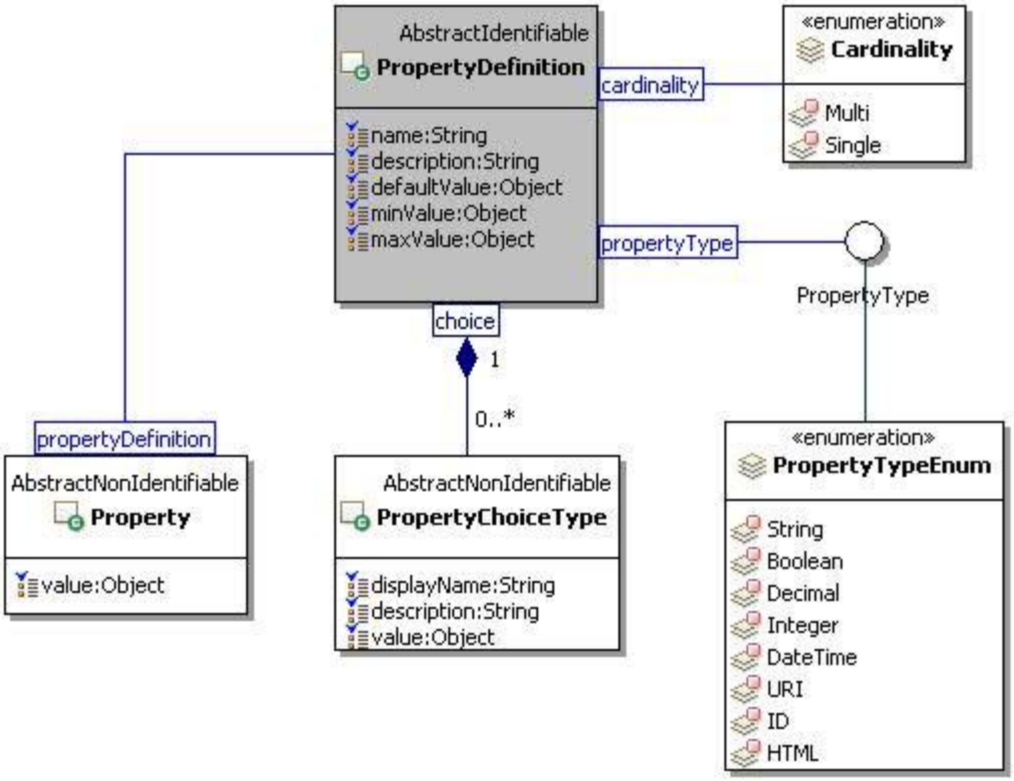

Figure 17 Property Definition and Property Class Diagram

3.6.3 PropertyChoiceType

3.6.3.1 Description

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

3.6.3.2 Class Definition

The PropertyChoiceType class is defined by the attribute values:

`localNamespace`

Value: icom_meta

`localName`

Value: PropertyChoiceType

`extendsFrom`

Value:

`stereotype`

Value: primary

`description`

Value: A choice for a property value.

`propertyDefinitions`

The values for this attribute are defined Section 3.6.3.3.

3.6.3.3 Property Definitions

The PropertyChoiceType class MUST have the property definitions:

icom_meta:displayName

Description:	Display name of a property choice.
Required:	True
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

icom:description

Description:	A description of a property choice.
Required:	False
Inherited:	False

Property Type:	String
Cardinality:	Single
Updatability:	Read Write

icom_meta:value

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

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

3.6.4 PropertyType

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

3.6.4.1 Class Definition

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

The PropertyType class is defined by the attribute values:

`localNamespace`

Value: icom_meta

`localName`

Value: PropertyType

`extendsFrom`

Value:

`stereotype`

Value: mixin

`description`

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

`propertyDefinitions`

The values for this attribute are defined in Section 3.6.4.2.

3.6.4.2 Property Definitions

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

3.6.5 PropertyTypeEnum

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

localNamespace

Value: icom_meta

localName

Value: PropertyTypeEnum

extendsFrom

Value: PropertyType

stereotype

Value: primary

isEnumeration

Value: TRUE

description

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

instances

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

The following names of data types are defined by ICOM:

- **icom_meta:String** to express **xsd:string**.
- **icom_meta:Boolean** to express **xsd:boolean**.
- **icom_meta:Decimal** to express **xsd:decimal**.
- **icom_meta:Integer** to express **xsd:integer**.
- **icom_meta:Datetime** to express **xsd:dateTime**.
- **icom_meta:URI** to express **xsd:anyURI**.
- **icom_meta:ID** to express opaque object identifiers.
- **icom_meta:HTML** to express documents or fragments of Hypertext Markup Language (HTML) content

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

3.6.6 Cardinality

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

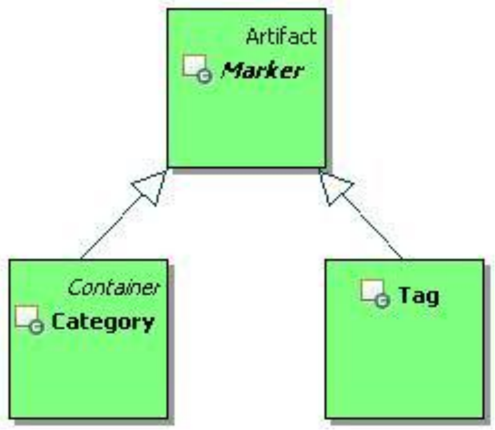

Figure 18 Marker Branch

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.

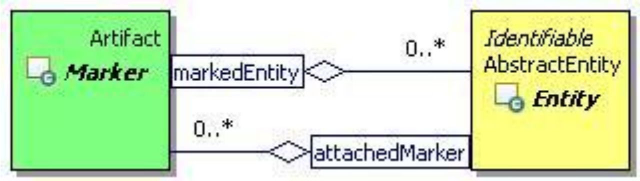

Figure 19 Marker Class Diagram

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

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.6.9.3.

3.6.9.3 Property Definitions

The Category class inherits property definitions from super classes.

The Category class MUST have the property definitions:

icom_meta:superCategory

Description:	A super category.
Required:	False
Inherited:	False
Property Type:	icom_meta:Category
Cardinality:	Single
Updatability:	Read Only

icom_meta:subcategory

Description:	Zero or more sub categories.
Required:	False
Inherited:	False
Property Type:	icom_meta:Category
Cardinality:	Multi
Updatability:	Read Only

icom_meta:isAbstract

Description:	Indicates whether a category is abstract or concrete.
Required:	False
Inherited:	False
Property Type:	Boolean
Cardinality:	Single
Updatability:	Read Write

icom_meta:propertyDefinition

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

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

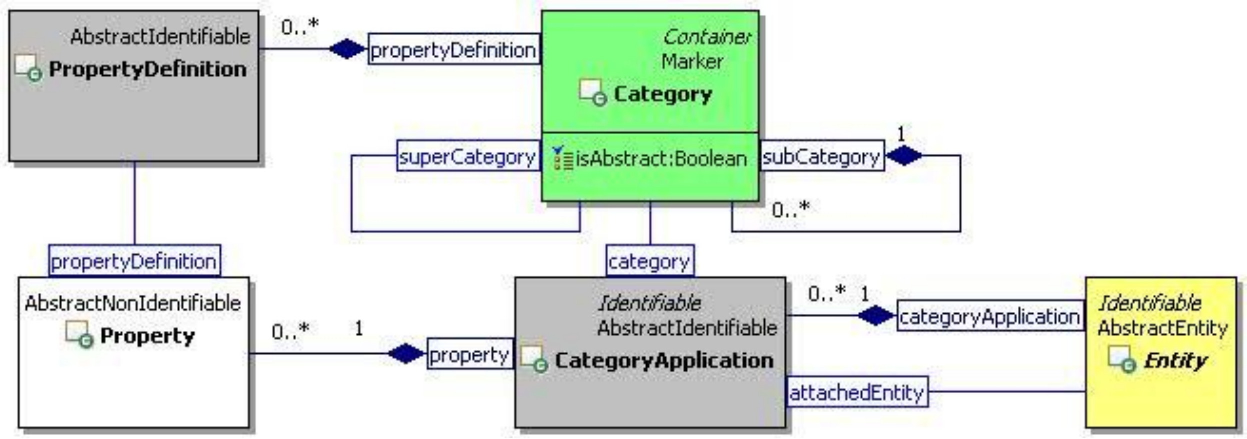

Figure 20 Category and Category Application Class Diagram

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

localName

Value: Tag

extendsFrom

Value: icom_meta:Marker

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.6.11.3.

3.6.11.3 Property Definitions

The Tag class inherits property definitions from super classes.

The Tag class MUST have the property definitions:

icom_meta:applicationCount

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

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

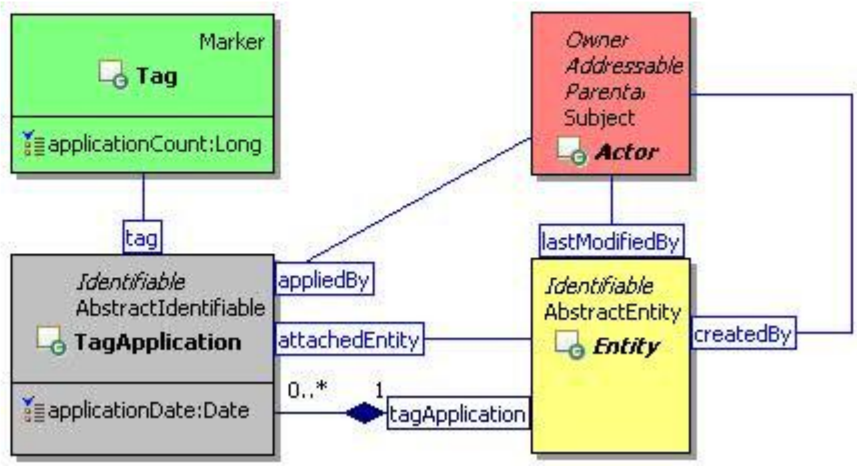

Figure 21 Tag and Tag Application Class Diagram

3.6.12 TagApplication

3.6.12.1 Description

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

3.6.12.2 Class Definition

The TagApplication class is defined by the attribute values:

localNamespace

Value: icom_meta

localName

Value: TagApplication

extendsFrom

Value: icom:Identifiable

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.6.12.3.

3.6.12.3 Property Definitions

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

icom_meta:attachedEntity

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

icom_meta:tag

Description:	A tag which is applied on an entity.
Required:	True
Inherited:	False
Property Type:	icom_meta:Tag

Cardinality:	Single
Updatability:	On Create

icom_meta:appliedBy

Description:	A user who applies a tag on an entity.
Required:	False
Inherited:	False
Property Type:	icom:Actor
Cardinality:	Single
Updatability:	Read Only

icom_meta:applicationDate

Description:	A date and time when a tag is applied on an entity.
Required:	False
Inherited:	False
Property Type:	DateTime
Cardinality:	Single
Updatability:	Read Write

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

3.6.13 RelationshipBondable

3.6.13.1 Description

A relationship bondable entity is an entity which may be relationship bonded.

Note: a relationship cannot be relationship bonded by other relationships, i.e. relationships are entities that are not relationship bondable.

3.6.13.2 Class Definition

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

The RelationshipBondable class is defined by the attribute values:

`localNamespace`

Value: icom_meta

`localName`

Value: RelationshipBondable

`extendsFrom`

Value: icom:Identifiable

`stereotype`

Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.6.13.3.

3.6.13.3 Property Definitions

The RelationshipBondable class inherits property definitions from super classes.

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

3.6.14 RelationshipDefinition

3.6.14.1 Description

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

3.6.14.2 Class Definition

The RelationshipDefinition class is defined by the attribute values:

localNamespace

Value: icom_meta

localName

Value: RelationshipDefinition

extendsFrom

Value: icom:EntityDefinition

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.6.14.3.

3.6.14.3 Property Definitions

The RelationshipDefinition class inherits property definitions from super classes.

The RelationshipDefinition class MUST have the property definitions:

icom_meta:propertyDefinition

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

icom_meta:allowedSourceType

Description:	A list of expanded names of relationship bondable classes, indicating that the source entity of a relationship MUST be an instance of a class in the list.
Required:	False
Inherited:	False
Property Type:	IRI
Cardinality:	Multi
Updatability:	Read Write

icom_meta:allowedTargetType

Description:	A list of expanded names of relationship bondable classes, indicating that the target entity of a relationship MUST be an instance of a class in the list.
Required:	False
Inherited:	False
Property Type:	IRI
Cardinality:	Multi
Updatability:	Read Write

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

3.6.15 Relationship

3.6.15.1 Description

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

3.6.15.2 Class Definition

The Relationship class is defined by the attribute values:

`localNamespace`

Value: icom_meta

localName

Value: Relationship

extendsFrom

Value: icom:Entity

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.6.15.3.

3.6.15.3 Property Definitions

The Relationship class inherits property definitions from super classes.

The Relationship class MUST have the property definitions:

icom_meta:relationshipDefinition

Description:	A definition of relationships.
Required:	True
Inherited:	False
Property Type:	icom_meta:RelationshipDefinition
Cardinality:	Single
Updatability:	On Create

icom_meta:sourceEntity

Description:	A source entity of a relationship.
Required:	True
Inherited:	False
Property Type:	icom_meta:RelationshipBondable
Cardinality:	Single
Updatability:	On Create

icom_meta:targetEntity

Description:	One or more target entities of a relationship.
Required:	True
Inherited:	False
Property Type:	icom_meta:RelationshipBondable
Cardinality:	Multi
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.

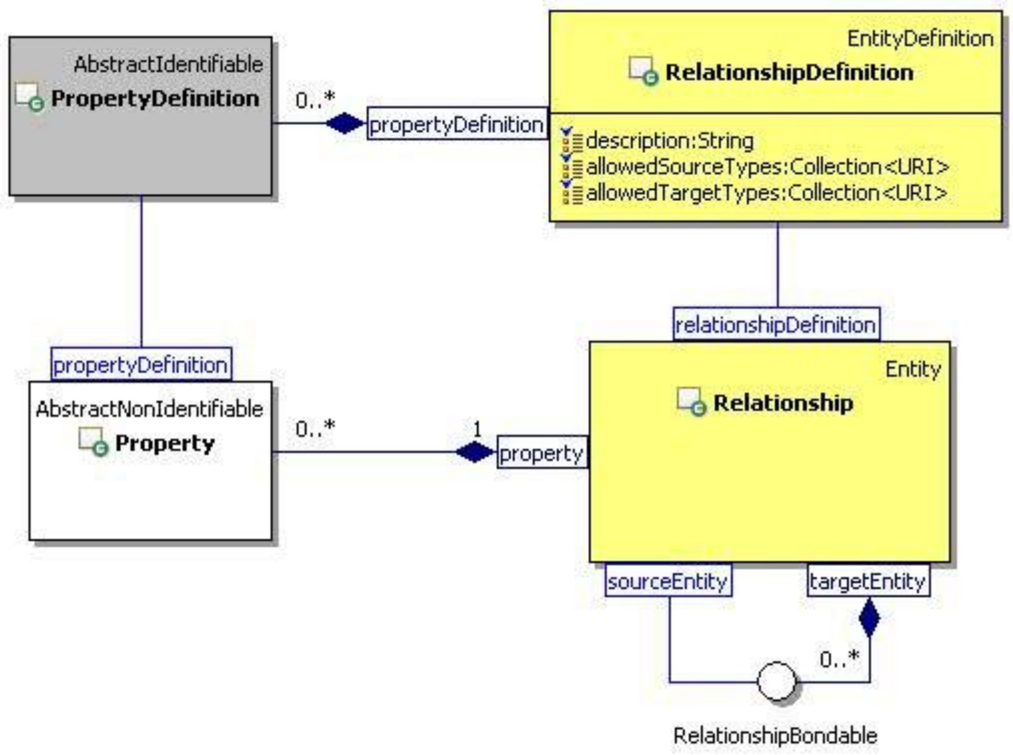

Figure 22 Relationship Class Diagram

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:

localNamespace

Value: icom

localName

Value: Participant

extendsFrom

Value:

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.7.1.3.

3.7.1.3 Property Definitions

The Participant class inherits property definitions from super classes.

The Participant class MUST have the property definitions:

icom:participant

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

icom:address

Description:	An address of a participant in a collaboration activity.
Required:	False
Inherited:	False
Property Type:	URI
Cardinality:	Single
Updatability:	On Create

icom:name

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

Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	On Create

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

3.7.2 Priority

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

The Priority is defined by the attribute values:

localNamespace

Value: icom

localName

Value: Priority

extendsFrom

Value:

stereotype

Value: primary

isEnumeration

Value: TRUE

description

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

instances

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

The following priorities are defined by ICOM:

- **icom:None** to express a normal priority.
- **icom:Low** to express a low priority.
- **icom:Medium** to express a medium priority.
- **icom:High** to express a high priority.

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:

localNamespace

Value: icom

localName

Value: Location

extendsFrom

Value:

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.7.4.3.

3.7.4.3 Property Definitions

The Location class inherits property definitions from super classes.

The Location class MUST have the property definitions:

icom:name

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

icom:description

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

icom:timeZone

Description:	Time zone of a location.
Required:	False
Inherited:	False

Property Type:	TimeZone
Cardinality:	Single
Updatability:	Read Write

icom:locationMark

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

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

3.7.5 GeoCoordinates

3.7.5.1 Description

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

3.7.5.2 Class Definition

The GeoCoordinates class is defined by the attribute values:

localNamespace

Value: icom

localName

Value: GeoCoordinates

extendsFrom

Value:

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 3.7.5.3.

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

Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

icom_ content:mediaType

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

icom_ content:contentDisposition

Description:	Content disposition is defined in RFC 2183 to specify a presentation style.
Required:	False
Inherited:	False
Property Type:	ContentDispositionType
Cardinality:	Single
Updatability:	Read Write

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

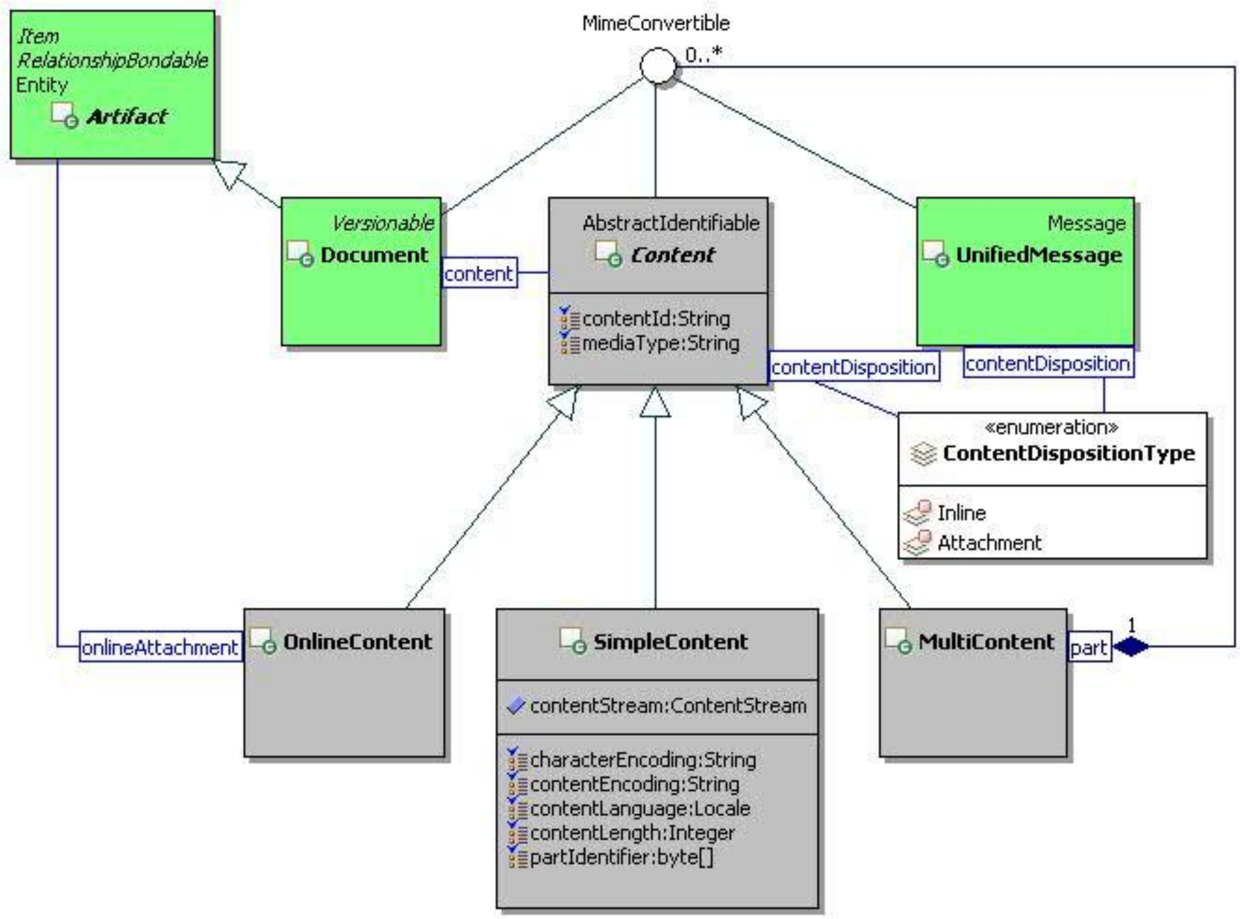

Figure 23 Composite Content Class Diagram

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

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.1.2.3.

4.1.2.3 Property Definitions

The MultiContent class inherits property definitions from super classes.

The MultiContent class MUST have the property definitions:

icom_content:part

Description:	Zero or more parts of a hierarchical composite content.
Required:	False
Inherited:	False
Property Type:	icom_msg:MimeConvertible
Cardinality:	Multi
Updatability:	Read Write

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

4.1.3 SimpleContent

4.1.3.1 Description

A simple content holds a single piece of data.

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

4.1.3.2 Class Definition

The SimpleContent class is defined by the attribute values:

localNamespace

Value: icom_content

localName

Value: SimpleContent

extendsFrom

Value: icom_content:Content

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.1.3.3.

4.1.3.3 Property Definitions

The SimpleContent class inherits property definitions from super classes.

The SimpleContent class MUST have the property definitions:

icom_content:characterEncoding

Description:	Character encoding specifies RFC 2616 character set of a content (a missing value means that a content should be treated as binary or raw).
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

icom_content:contentEncoding

Description:	Content encoding specifies RFC 2616 content encoding applied to a content.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

icom_content:contentLangauge

Description:	Content language specifies RFC 2616 content language for a content (a missing value means non-natural language content).
Required:	False
Inherited:	False
Property Type:	Locale
Cardinality:	Single
Updatability:	Read Write

icom_content:contentLength

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

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

4.1.4 OnlineContent

4.1.4.1 Description

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

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

4.1.4.2 Class Definition

The OnlineContent class is defined by the attribute values:

`localNamespace`

Value: icom_content

`localName`

Value: OnlineContent

`extendsFrom`

Value: icom_content:Content

`stereotype`

Value: primary

`description`

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

`propertyDefinitions`

The values for this attribute are defined in Section 4.1.4.3.

4.1.4.3 Property Definitions

The OnlineContent class inherits property definitions from super classes.

The OnlineContent class MUST have the property definitions:

icom_content:onlineAttachment

Description:	An online artifact attached to a message.
Required:	True
Inherited:	False
Property Type:	Artifact
Cardinality:	Single
Updatability:	Read Write

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

4.1.5 ContentDispositionType

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

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

The ContentDispositionType class is defined by the attribute values:

localNamespace

Value: icom_content

localName

Value: ContentDispositionType

extendsFrom

Value:

stereotype

Value: primary

isEnumeration

Value: TRUE

description

Value: An enumeration of instances each of which expresses a presentation style of content defined in RFC 2183.

instances

Value: <icom_content:Inline, icom_content:Attachment>

There are two content disposition types defined by ICOM:

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

4.1.6 Attachment

4.1.6.1 Description

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

4.1.6.2 Class Definition

The Attachment class is defined by the attribute values:

localNamespace

Value: icom_content

localName

Value: Attachment

extendsFrom

Value:

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.1.6.3.

4.1.6.3 Property Definitions

The Attachment class MUST have the property definitions:

icom:name

Description:	Name of a simple content attachment.
Required:	True
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

icom_content:content

Description:	A simple content attached to an occurrence, task, or contact artifact.
Required:	True
Inherited:	False
Property Type:	SimpleContent

Cardinality: Single
Updatability: Read Write

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

4.2 Document Module

4.2.1 Versionable

4.2.1.1 Description

A versionable artifact is

1. a representative copy,
2. a specific versioned copy, or
3. a private working copy

of an artifact version series.

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

When a versionable artifact is under version control:

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

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

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

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

ICOM version control model is based on the CMIS version control model specified in Section 2.1.9 of 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

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

4.2.2 VersionControlMetadata

4.2.2.1 Description

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

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

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

4.2.2.2 Class Definition

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

The VersionControlMetadata class is defined by the attribute values:

localNamespace

Value: icom_doc

localName

Value: VersionControlMetadata

extendsFrom

Value: icom:Identifiable

stereotype

Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.2.2.3.

4.2.2.3 Property Definitions

The VersionControlMetadata class inherits property definitions from super classes.

The VersionControlMetadata class MUST have the property definitions:

icom_doc:representativeCopy

Description: A representative copy of a versionable artifact.

Required:	False
Inherited:	False
Property Type:	icom_doc:Versionable
Cardinality:	Single
Updatability:	Read Only

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

4.2.3 VersionSeries

4.2.3.1 Description

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

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

4.2.3.2 Class Definition

The VersionSeries class is defined by the attribute values:

localNamespace

Value: icom_doc

localName

Value: VersionSeries

extendsFrom

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

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.2.3.3.

4.2.3.3 Property Definitions

The VersionSeries class inherits property definitions from super classes.

The VersionSeries class MUST have the property definitions:

icom_doc:versionHistory

Description: A history of version nodes of a versionable artifact.
Required: False
Inherited: False
Property Type: icom_doc:Version
Cardinality: Multi
Updatability: Read Only

icom_doc:versionableHistory

Description: A history of the versioned copies of a versionable artifact.
Required: False
Inherited: False
Property Type: icom_doc:Versionable
Cardinality: Multi
Updatability: Read Only

icom_doc:latestVersionedCopy

Description: Latest versioned copy of a versionable artifact.
Required: False
Inherited: False
Property Type: icom_doc:Versionable
Cardinality: Single
Updatability: Read Only

icom_doc:privateWorkingCopy

Description: A private working copy of a versionable artifact.
Required: False
Inherited: False
Property Type: icom_doc:Versionable
Cardinality: Single
Updatability: Read Only

icom_doc:versionSeriesCheckedOut

Description: Indicates whether a version series is checked out.
Required: False
Inherited: False
Property Type: Boolean
Cardinality: Single
Updatability: Read Only

icom_doc:versionSeriesCheckedOutBy

Description: An actor who checks out a version series.
Required: False

Inherited:	False
Property Type:	icom:Actor
Cardinality:	Single
Updatability:	Read Only

icom_doc:versionSeriesCheckedOutOn

Description:	The time when a version series is checked out.
Required:	False
Inherited:	False
Property Type:	DateTime
Cardinality:	Single
Updatability:	Read Only

icom_doc:versionSeriesCheckoutComment

Description:	A checked out comment of a version series.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Only

icom_doc:totalSize

Description:	Total size of all versioned copies of a versionable artifact in a version series.
Required:	False
Inherited:	False
Property Type:	Integer
Cardinality:	Single
Updatability:	Read Only

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

4.2.4 Version

4.2.4.1 Description

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

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

4.2.4.2 Class Definition

The Version class is defined by the attribute values:

`localNamespace`

Value: icom_doc

localName

Value: Version

extendsFrom

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

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.2.4.3.

4.2.4.3 Property Definitions

The Version class inherits property definitions from super classes.

The Version class MUST have the property definitions;

icom_doc:checkinComment

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

icom_doc:versionNumber

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

icom_doc:versionLabel

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

Cardinality:	Single
Updatability:	Read Write

icom_doc:majorVersion

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

icom_doc:versionedOrPrivateWorkingCopy

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

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

4.2.5 VersionType

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

The VersionType class is defined by the attribute values:

`localNamespace`

Value: icom_doc

`localName`

Value: VersionType

`extendsFrom`

Value:

`stereotype`

Value: primary

`isEnumeration`

Value: TRUE

description

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

instances

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

There are four version types defined by ICOM:

- **icom_doc:RepresentativeCopy** to express that a versionable artifact is a representative copy of an artifact.
- **icom_doc:ViewOnlyRepresentativeCopy** to express that a versionable artifact is a view only representative copy of an artifact version series.
- **icom_doc:VersionedCopy** to express that a versionable artifact is a versioned copy of an artifact version series.
- **icom_doc:PrivateWorkingCopy** to express that a versionable artifact is a private working copy of an artifact version series.

4.2.6 Document

4.2.6.1 Description

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

4.2.6.2 Class Definition

The Document class is defined by the attribute values:

localNamespace

Value: icom_doc

localName

Value: Document

extendsFrom

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

stereotype

Value: primary

description

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

propertyDefinitions

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.

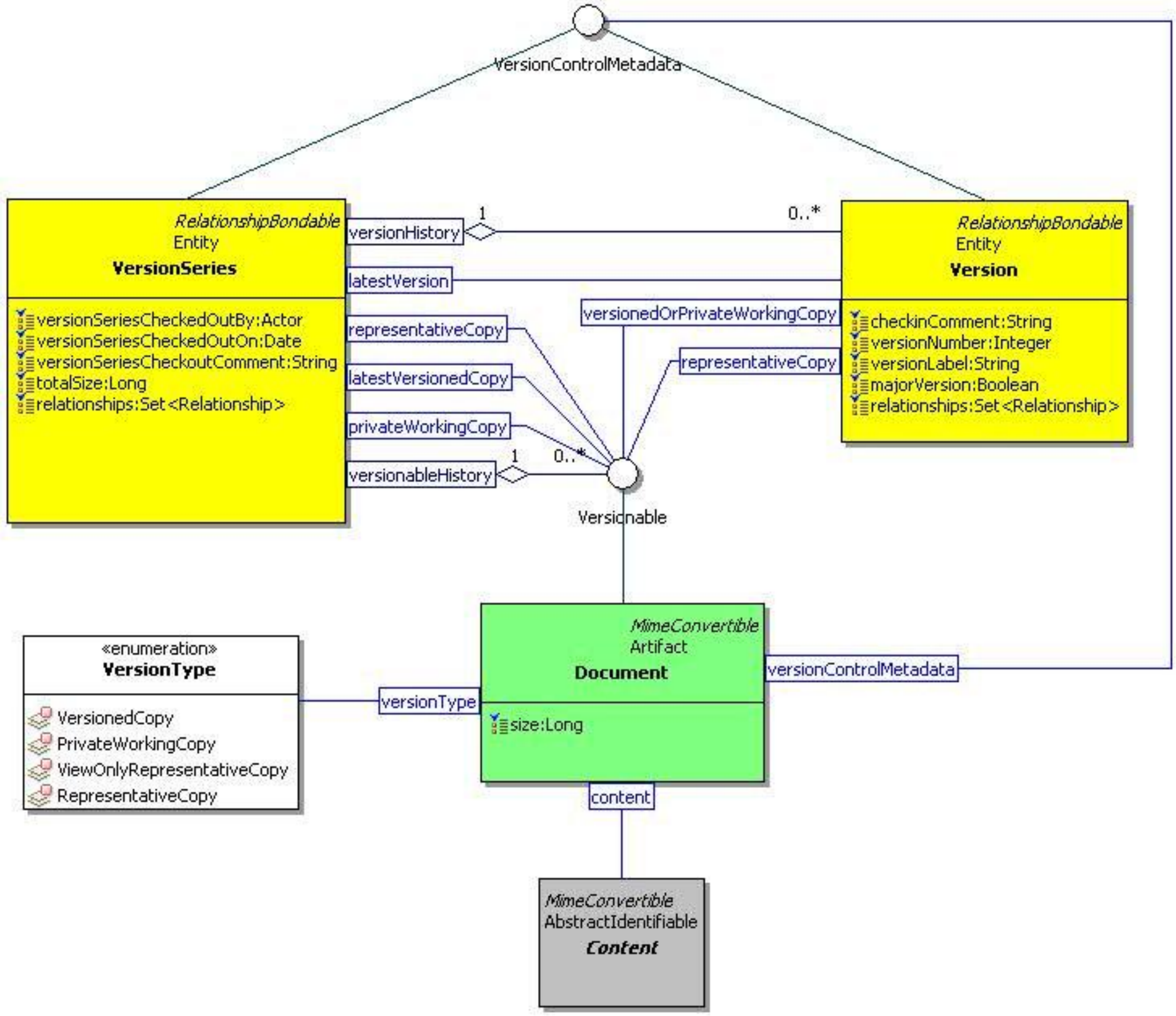

Figure 24 Document, Version Series, and Version Class Diagram

4.2.7 WikiPage

4.2.7.1 Description

A wiki page is a document that contains a rendered page and a rendered content.

4.2.7.2 Class Definition

The WikiPage class is defined by the attribute values:

`localNamespace`

Value: icom_doc

`localName`

Value: WikiPage

extendsFrom

Value: icom_doc:Document

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.2.7.3.

4.2.7.3 Property Definitions

The WikiPage class inherits property definitions from super classes.

The WikiPage class MUST have the property definitions:

icom_doc:renderedPage

Description:	A page rendered from wiki page content.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Only

icom_doc:renderedContent

Description:	An object rendered from wiki page content.
Required:	False
Inherited:	False
Property Type:	Object
Cardinality:	Single
Updatability:	Read Only

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

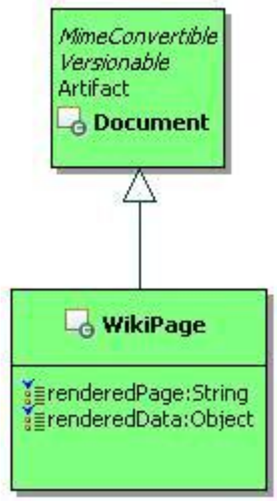

Figure 25 WikiPage Class Diagram.

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

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

propertyDefinitions

The values for this attribute are defined in Section 4.3.1.3.

4.3.1.3 Property Definitions

The MimeConvertible class inherits property definitions from super classes.

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

4.3.2 Message

4.3.2.1 Description

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

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

4.3.2.2 Class Definition

The Message class is defined by the attribute values:

localNamespace

Value: icom_msg

localName

Value: Message

extendsFrom

Value: icom:Artifact

stereotype

Value: primary

isAbstract

Value: TRUE

description

Value: A message is a unit of conversation.

propertyDefinitions

The values for this attribute are defined in Section 4.3.2.3.

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

icom_msg:ccReceivers

Description:	A list of participants to whom a message is carbon-copied or to be carbon-copied.
Required:	False
Inherited:	False
Property Type:	icom:Participant
Cardinality:	Multi
Updatability:	Read Write

icom_msg:bccReceivers

Description:	A list of participants to whom a message is blind-carbon-copied or to be blind-carbon-copied.
Required:	False
Inherited:	False
Property Type:	icom:Participant
Cardinality:	Multi
Updatability:	Read Write

icom_msg:replyTo

Description:	A list of participants to whom reply messages should be sent.
Required:	False
Inherited:	False
Property Type:	icom:Participant
Cardinality:	Multi
Updatability:	Read Write

icom_ content:contentId

Description:	Content id is a unique identifier for a message part in multi-part messages.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

icom_ content:mediaType

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

icom_ content:contentDisposition

Description:	Content disposition is defined in RFC 2183 to specify a presentation style.
Required:	False
Inherited:	False
Property Type:	icom_content:ContentDispositionType
Cardinality:	Single
Updatability:	Read Write

icom:priority

Description:	The priority of a message.
Required:	False
Inherited:	False
Property Type:	icom:Priority
Cardinality:	Single
Updatability:	Read Write

icom_msg:flag

Description:	Zero or more flags on a message.
Required:	False
Inherited:	False
Property Type:	icom_msg:UnifiedMessageFlag
Cardinality:	Multi
Updatability:	Read Write

icom_msg:messageDispositionNotificationRequested

Description:	A message disposition notification (RFC 2298) is requested for a message.
Required:	False
Inherited:	False
Property Type:	Boolean
Cardinality:	Single
Updatability:	Read Write

icom_msg:messageDeliveryStatusNotificationRequest

Description:	Indicates the types of delivery status notifications (RFC 1891) requested for a message. Default is icom_msg:Failure.
Required:	False
Inherited:	False
Property Type:	UnifiedMessageDeliveryStatusNotificationRequest
Cardinality:	Multi
Updatability:	Read Write

icom_msg:channel

Description:	Indicates the delivery channel of a message.
Required:	False
Inherited:	False
Property Type:	icom_msg:UnifiedMessageChannel
Cardinality:	Single
Updatability:	Read Write

icom_msg:mode

Description:	Indicates an editable mode (new, draft, or delivered) of a message.
Required:	False
Inherited:	False
Property Type:	icom_msg:UnifiedMessageEditMode
Cardinality:	Single
Updatability:	Read Only

icom_msg:mimeHeader

Description:	A list of headers including those defined in RFC 822 and other custom headers. Each header is represented by a multi-valued property. The name of a property is a printable header name. The value of a property is a collection of ascii or non-ascii strings.
Required:	False
Inherited:	False
Property Type:	icom_meta:Property
Cardinality:	Multi
Updatability:	Read Write

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

4.3.4 UnifiedMessageFlag

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

The UnifiedMessageFlag class is defined by the attribute values:

localNamespace

Value: icom_msg

localName

Value: UnifiedMessageFlag

extendsFrom

Value:

stereotype

Value: primary

isEnumeration

Value: TRUE

description

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

instances

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

There are eight flags defined by ICOM:

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

4.3.5 UnifiedMessageDeliveryStatusNotificationRequest

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

The UnifiedMessageDeliveryStatusNotificationRequest class is defined by the attribute values:

localNamespace

Value: icom_msg

localName

Value: UnifiedMessageDeliveryStatusNotificationRequest

extendsFrom

Value:

stereotype

Value: primary

isEnumeration

Value: TRUE

description

Value: An enumeration of the instances each of which expresses a request for one of several types of delivery status notification.

instances

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

There are four delivery status notification requests defined by ICOM:

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

4.3.6 UnifiedMessageChannel

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

The UnifiedMessageChannel class is defined by the attribute values:

localNamespace

Value: icom_msg

localName

Value: UnifiedMessageChannel

extendsFrom

Value:

stereotype

Value: primary

isEnumeration

Value: TRUE

description

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

instances

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

There are four channel types defined by ICOM:

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

4.3.7 UnifiedMessageEditMode

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

The UnifiedMessageEditMode class is defined by the attribute values:

localNamespace

Value: icom_msg

localName

Value: UnifiedMessageEditMode

extendsFrom

Value:

stereotype

Value: primary

isEnumeration

Value: TRUE

description

Value: An enumeration of the instances each of which expresses whether a message is a draft copy, delivered copy, or other.

instances

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

There are three modes defined by ICOM:

- **icom_msg:DraftCopy** to express that a message is saved as a draft.
- **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.

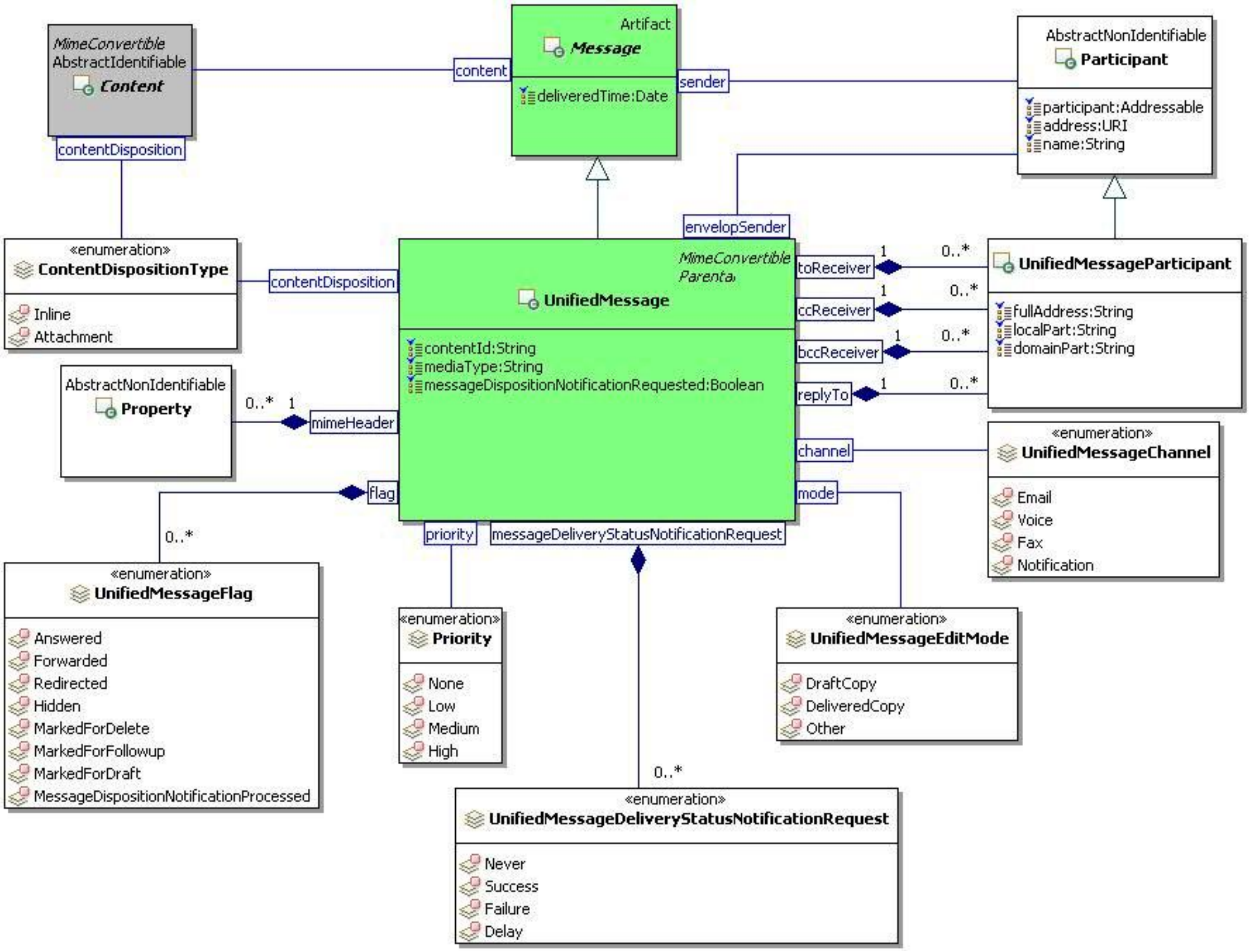

Figure 26 Unified Message Class Diagram

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

localName

Value: InstantMessage

extendsFrom

Value: icom_msg:Message

stereotype

Value: primary

isAbstract

Value: TRUE

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.3.8.3.

4.3.8.3 Property Definitions

The InstantMessage class inherits property definitions from super classes.

The InstantMessage class MUST have the property definitions:

icom_msg:toReceivers

Description:	A list of participants to whom a message is sent or to be sent.
Required:	False
Inherited:	False
Property Type:	icom:Participant
Cardinality:	Multi
Updatability:	Read Write

icom_msg:conversationId

Description:	An identifier of a conversation involving one or more instant messages.
Required:	False
Inherited:	False
Property Type:	Integer
Cardinality:	Single
Updatability:	Read Write

icom_msg:clientSideId

Description:	An identifier of a client.
Required:	False

Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Write

icom_msg:formatingStyle

Description: A formatting style of rich text message in xhtml.
Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Write

icom_msg:instantMessageType

Description: A type of instant message.
Required: False
Inherited: False
Property Type: icom_msg:InstantMessageType
Cardinality: Single
Updatability: Read Write

icom_msg:chatStatus

Description: A chat status of a user.
Required: False
Inherited: False
Property Type: icom_msg:InstantMessageChatStatus
Cardinality: Single
Updatability: Read Write

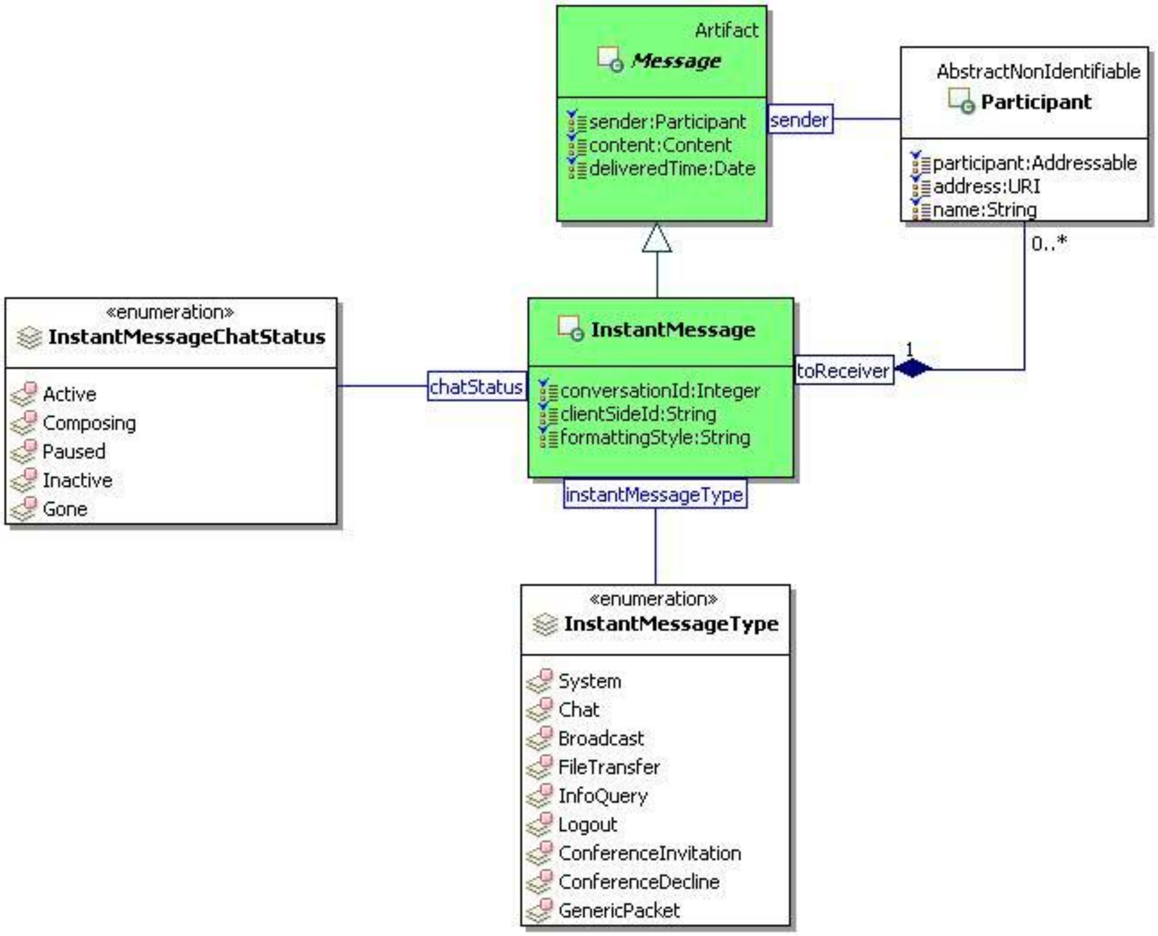

Figure 27 InstantMessage Class Diagram.

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

isEnumeration

Value: TRUE

description

Value: An enumeration of the instances each of which expresses a type of instant message.

instances

Value: <icom_msg:System, icom_ msg:Chat, icom_ msg:Broadcast, icom_msg:FileTransfer, icom_ msg:InfoQuery, icom_ msg:Logout, icom_ msg:ConferenceInvitation, icom_ msg:ConferenceDecline, icom_ msg:GenericPacket>

There are nine instant message types defined by ICOM:

- **icom_msg:System** to express that an instant message is a system message.
- **icom_msg:Chat** to express that an instant message is a chat message.
- **icom_msg:Broadcast** to express that an instant message is a broadcast message.
- **icom_msg:FileTransfer**: to express that an instant message is a file transfer message.
- **icom_msg:InfoQuery** to express that an instant message is a info query message.
- **icom_msg:Logout** to express that an instant message is a logout message.
- **icom_msg:ConferenceInvitation** to express that an instant message is a conference invitation message.
- **icom_msg:ConferenceDecline** to express that an instant message is a decline message to a conference invitation.
- **icom_msg:GenericGone** to express that an instant message is a generic message indicating that a user is gone.

4.3.10 InstantMessageChatStatus

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

The InstantMessageChatStatus class is defined by the attribute values:

localNamespace

Value: icom_msg

localName

Value: InstantMessageChatStatus

extendsFrom

Value:

stereotype

Value: primary

isEnumeration

Value: TRUE

description

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

instances

Value: <icom_msg:Active, icom_ msg:Composing, icom_ msg:Paused, icom_msg:Inactive, icom_ msg:Gone>

There are five chat status defined by ICOM:

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

4.3.11 InstantMessageFeed

4.3.11.1 Description

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

4.3.11.2 Class Definition

The InstantMessageFeed class is defined by the attribute values:

localNamespace

Value: icom_msg

localName

Value: InstantMessageFeed

extendsFrom

Value: icom:Entity

stereotype

Value: primary

description

Value: An instant message feed contains a set of instant message connections.

propertyDefinitions

The values for this attribute are defined in Section 4.3.11.3.

4.3.11.3 Property Definitions

The InstantMessageFeed class inherits property definitions from super classes.

The InstantMessageFeed class MUST have the property definitions:

icom_msg:connection

Description:	One or more instant messaging connections.
Required:	False
Inherited:	False
Property Type:	icom_msg:InstantMessageConnection
Cardinality:	Multi
Updatability:	Read Only

icom_msg:outboundInstantMessage

Description:	A queue for outbound instant messages.
Required:	False
Inherited:	False
Property Type:	icom_msg:InstantMessage
Cardinality:	Multi
Updatability:	Write Only

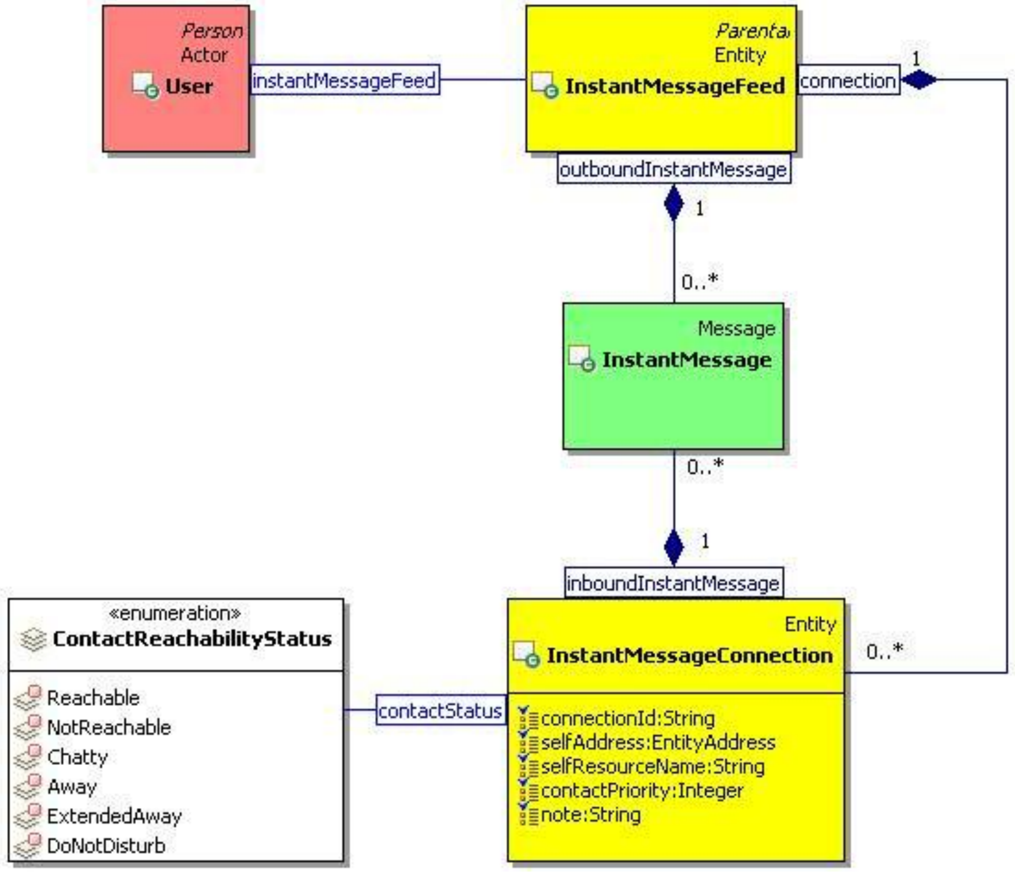

Figure 28 Instant Message Feed and Connection Class Diagram.

4.3.12 InstantMessageConnection

4.3.12.1 Description

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

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

4.3.12.2 Class Definition

The InstantMessageConnection class is defined by the attribute values:

`localNamespace`

Value: icom_msg

`localName`

Value: InstantMessageConnection

`extendsFrom`

Value: icom:Entity

`stereotype`

Value: primary

`description`

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

`propertyDefinitions`

The values for this attribute are defined in Section 4.3.12.3.

4.3.12.3 Property Definitions

The InstantMessageConnection class inherits property definitions from super classes.

The InstantMessageConnection class MUST have the property definitions:

icom_msg:connectionId

Description:	An identifier of a connection.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Only

icom_msg:selfAddress

Description:	Address of a presentity who opens a connection.
Required:	True
Inherited:	False
Property Type:	URI
Cardinality:	Single
Updatability:	On Create

icom_msg:selfResourceName

Description:	Resource name associated with a connection.
Required:	True
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	On Create

icom_presence:contactStatus

Description:	Reachability status to be propagated to an associated contact method.
Required:	False
Inherited:	False
Property Type:	icom_presence:ContactReachabilityStatus
Cardinality:	Single
Updatability:	Write Only

icom_presence:contactPriority

Description:	Priority to be propagated to an associated contact method.
Required:	False
Inherited:	False
Property Type:	Integer
Cardinality:	Single
Updatability:	Write Only

icom_presence:note

Description:	Note to be propagated to an associated contact method.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Write Only

icom_msg:inboundInstantMessage

Description:	A queue for inbound instant messages.
Required:	False

Inherited:	False
Property Type:	icom_msg:InstantMessage
Cardinality:	Multi
Updatability:	Read Only

4.4 Presence Module

4.4.1 Presence

4.4.1.1 Description

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

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

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

4.4.1.2 Class Definition

The Presence class is defined by the attribute values:

`localNamespace`

Value: icom_presence

`localName`

Value: Presence

`extendsFrom`

Value: icom:Identifiable

`stereotype`

Value: primary

`description`

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

`propertyDefinitions`

The values for this attribute are defined in Section 4.4.1.3.

4.4.1.3 Property Definitions

The Presence class inherits property definitions from super classes.

The Presence class MUST have the property definitions:

icom:lastModificationDate

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

Inherited:	False
Property Type:	DateTime
Cardinality:	Single
Updatability:	Read Only

icom_presence:mode

Description:	Editable mode of a presence.
Required:	False
Inherited:	False
Property Type:	icom_presence:PresenceEditMode
Cardinality:	Single
Updatability:	Read Only

icom_presence:contactMethod

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

icom_presence:activity

Description:	A collection of activities describing what a presentity is doing.
Required:	False
Inherited:	False
Property Type:	icom_presence:Activity
Cardinality:	Multi
Updatability:	Read Only

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

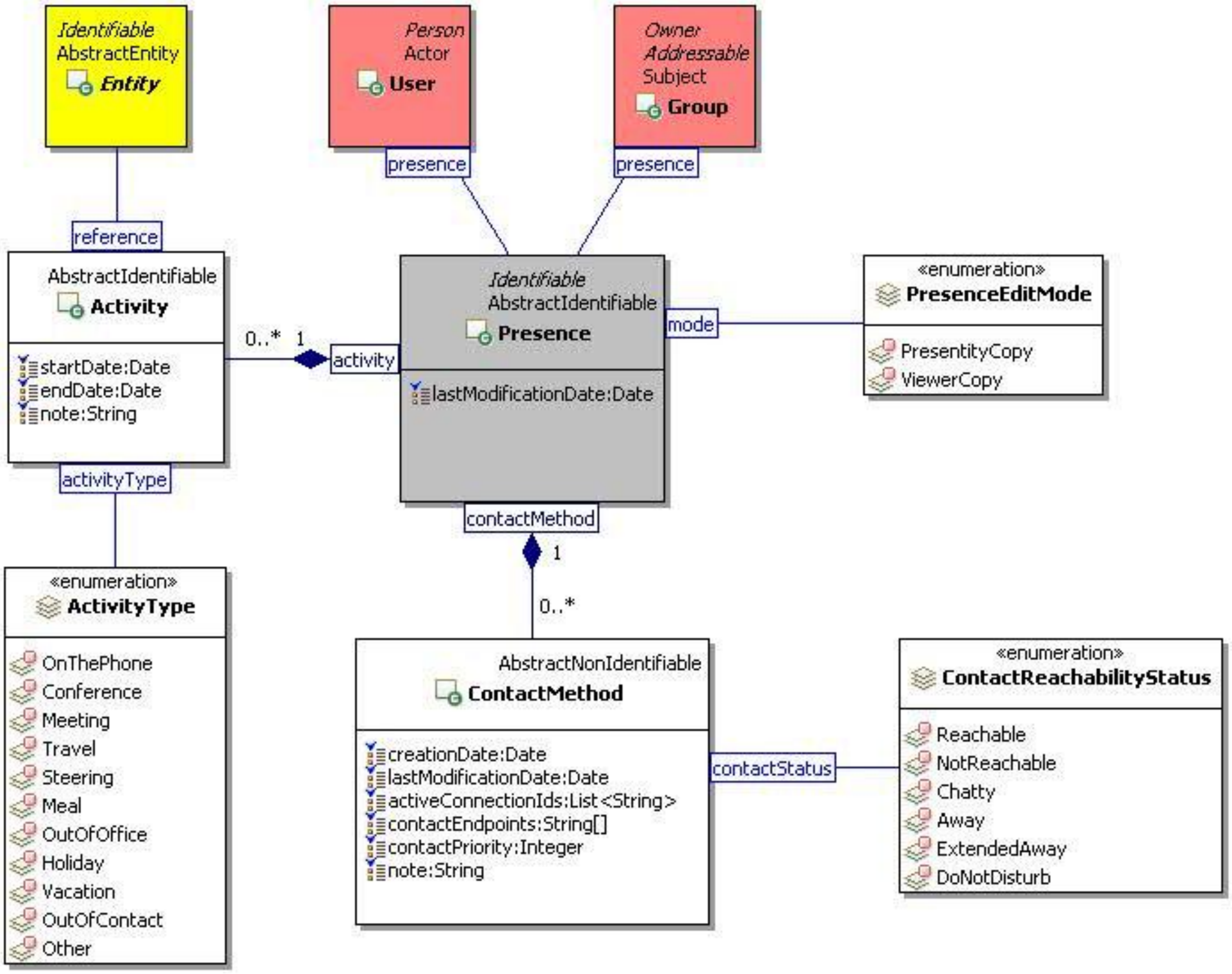

Figure 29 Presence Class Diagram.

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

isEnumeration

Value: TRUE

description

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

instances

Value: <icom_presence:PresentityCopy, icom_presence:ViewerCopy>

There are two presence editable modes defined by ICOM:

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

4.4.3 ContactMethod

4.4.3.1 Description

A contact method object describes reachability circumstances of a presentity.

4.4.3.2 Class Definition

The ContactMethod class is defined by the attribute values:

localNamespace

Value: icom_presence

localName

Value: ContactMethod

extendsFrom

Value:

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.4.3.3

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

Updatability:	Read Only

icom_presence:contactStatus

Description:	Status of a contact method in a presence.
Required:	False
Inherited:	False
Property Type:	icom_presence:ContactReachabilityStatus
Cardinality:	Single
Updatability:	Read Only

icom_presence:note

Description:	A note about a contact method in a presence.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Only

4.4.4 ContactReachabilityStatus

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

The ContactReachabilityStatus class is defined by the attribute values:

localNamespace

Value: icom_presence

localName

Value: ContactReachabilityStatus

extendsFrom

Value:

stereotype

Value: primary

isEnumeration

Value: TRUE

description

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

`instances`

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

There are six reachability status defined by ICOM:

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

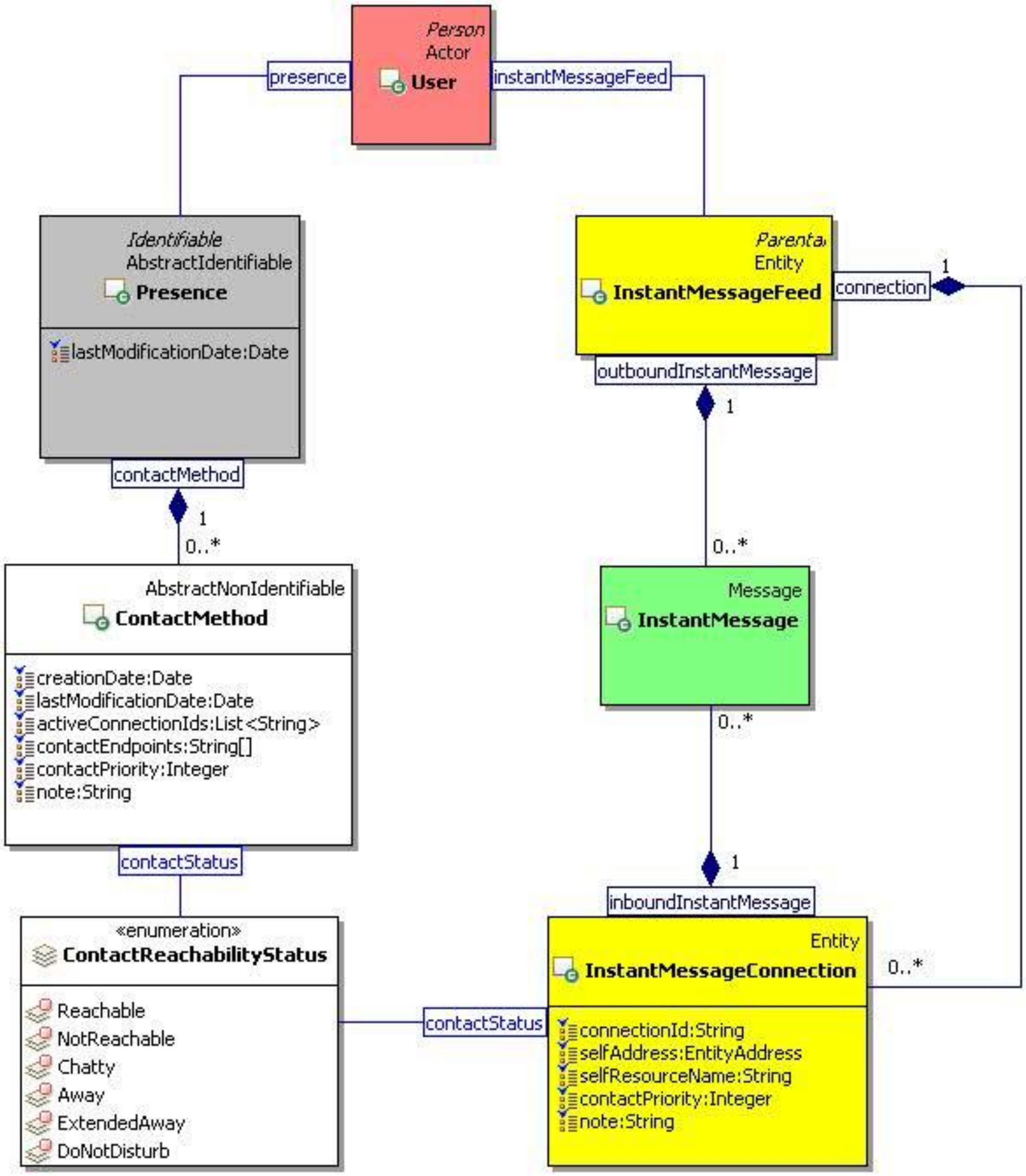

Figure 30 Presence Contact Method and Instant Message Connection Class Diagram.

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

localName

Value: Activity

extendsFrom

Value:

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.4.5.3.

4.4.5.3 Property Definitions

The Activity class MUST have the property definitions:

icom:startDate

Description:	Start date and time of an activity.
Required:	True
Inherited:	False
Property Type:	DateTime
Cardinality:	Single
Updatability:	Read Write

icom:endDate

Description:	End date and time of an activity.
Required:	True
Inherited:	False
Property Type:	DateTime
Cardinality:	Single
Updatability:	Read Write

icom_presence:activityType

Description:	Type of an activity.
Required:	true
Inherited:	False
Property Type:	icom_presence:ActivityType
Cardinality:	Single
Updatability:	Read Write

icom_presence:note

Description:	A note describing an activity.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

icom_presence:reference

Description:	An entity, such as occurrence, task, conference, etc., which is the source of or reference for an activity.
Required:	False
Inherited:	False
Property Type:	icom:Entity
Cardinality:	Single
Updatability:	Read Write

4.4.6 ActivityType

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

The ActivityType class is defined by the attribute values:

localNamespace

Value: icom_presence

localName

Value: ActivityType

extendsFrom

Value:

stereotype

Value: primary

isEnumeration

Value: TRUE

description

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

instances

Value: <icom_presence:OnThePhone, icom_ presence:Conference, icom_ presence:Meeting, icom_presence:Travel, icom_ presence:Steering, icom_ presence:Meal,

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

There are eleven activity types defined by ICOM:

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

4.5 Address Book Module

4.5.1 Addressable

4.5.1.1 Description

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

4.5.1.2 Class Definition

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

The Addressable class is defined by the attribute values:

`localNamespace`

Value: icom_card

`localName`

Value: Addressable

`extendsFrom`

Value: icom:Identifiable

`stereotype`

Value: mixin

`description`

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

propertyDefinitions

The values for this attribute are defined in 4.5.1.3.

4.5.1.3 Property Definitions

The Addressable class inherits property definitions from super classes.

The Addressable class MUST have the property definitions:

icom_card:address

Description:	Zero or more addresses of an addressable object.
Required:	False
Inherited:	False
Property Type:	icom_card:EntityAddress
Cardinality:	Multi
Updatability:	Read Write

icom_card:primaryAddress

Description:	The primary address of an addressable object.
Required:	False
Inherited:	False
Property Type:	icom_card:EntityAddress
Cardinality:	Single
Updatability:	Read Write

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

4.5.2 Person

4.5.2.1 Description

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

4.5.2.2 Class Definition

The Person class is a mixin class which defines the characteristics of persons.

The Person class is defined by the attribute values:

localNamespace

Value: icom_card

localName

Value: Person

extendsFrom

Value: icom_card:Person

stereotype

Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.5.2.3.

4.5.2.3 Property Definitions

The Person class inherits property definitions from super classes.

The Person class MUST have the property definitions:

icom_card:givenName

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

icom_card:middleName

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

icom_card:familyName

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

icom_card:prefix

Description:	Prefix of a person’s name.
Required:	False

Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

icom_card:suffix

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

icom_card:nickname

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

icom_card:jobTitle

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

icom_card:department

Description:	A person’s affiliated department.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

icom_card:officeLocation

Description:	Location of a person’s department.
Required:	False
Inherited:	False
Property Type:	String

Cardinality:	Single
Updatability:	Read Write

icom_card:company

Description:	A person’s affiliated company.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

icom_card:profession

Description:	A person’s profession.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

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

4.5.3 AddressBook

4.5.3.1 Description

An address book is a folder that contains addressable contacts.

4.5.3.2 Class Definition

The AddressBook class is defined by the attribute values:

localNamespace

Value: icom_card

localName

Value: AddressBook

extendsFrom

Value: icom:Folder

stereotype

Value: primary

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.

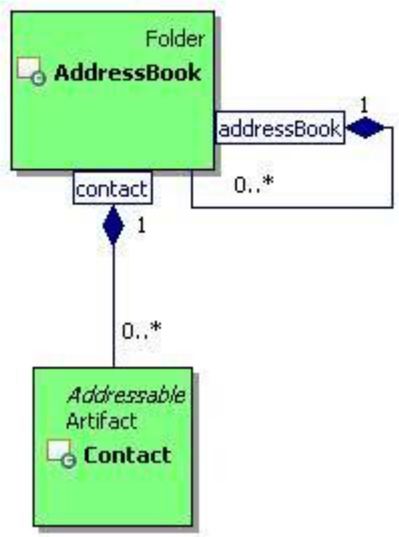

Figure 31 AddressBook Class Diagram.

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

Cardinality:	Multi
Updatability:	Read Write

icom_content:attachment

Description:	One or more simple content attachments in a contact.
Required:	False
Inherited:	False
Property Type:	icom_content:Attachment
Cardinality:	Multi
Updatability:	Read Write

icom_card:givenName

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

icom_card:middleName

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

icom_card:familyName

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

icom_card:prefix

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

icom_card:suffix

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

icom_card:nickname

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

icom_card:jobTitle

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

icom_card:department

Description:	A person’s affiliated department.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

icom_card:officeLocation

Description:	Location of a person’s department.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

icom_card:company

Description:	A person’s affiliated company.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

icom_card:profession

Description:	A person’s profession.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

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

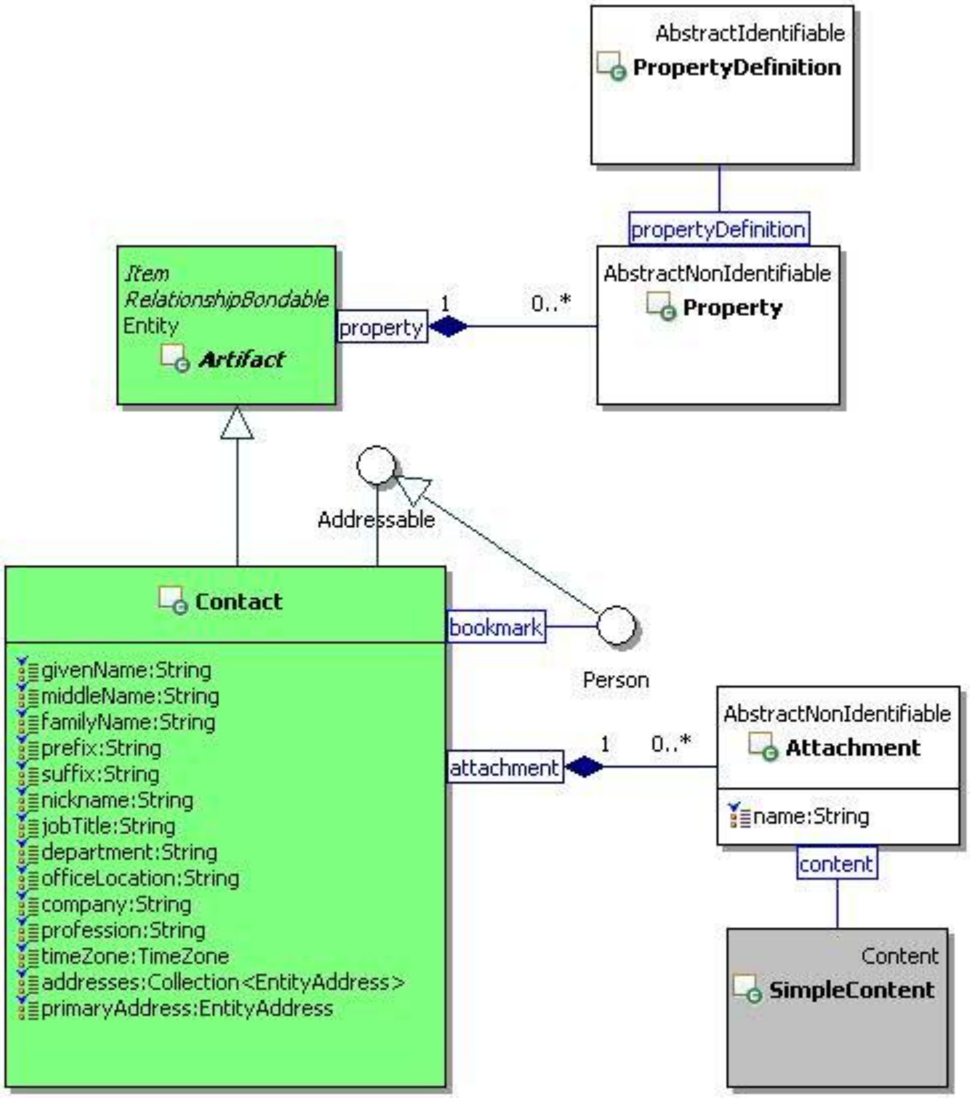

Figure 32 Contact Class Diagram.

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

localName

Value: Calendar

extendsFrom

Value: icom:Folder

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in 4.6.1.3.

4.6.1.3 Property Definitions

The Calendar class inherits property definitions from super classes.

The Calendar class MUST have the property definitions:

icom_cal:timeZone

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

icom:element

Description:	Elements of a calendar.
Required:	False
Inherited:	True
Property Type:	icom_cal:Occurrence
Cardinality:	Multi
Updatability:	Read Only

icom_cal:recurrence

Description:	Recurrence elements of a calendar.
Required:	False
Inherited:	True
Property Type:	icom_cal:OccurrenceSeries
Cardinality:	Multi

Updatability: Read Only

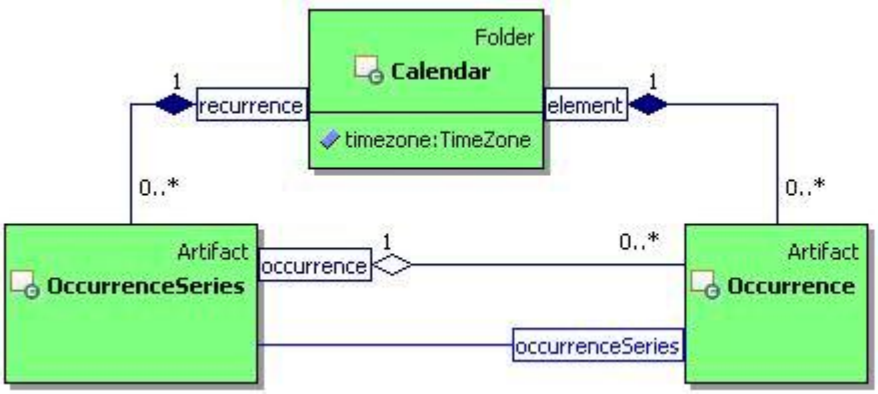

Figure 33 Calendar Class Diagram.

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

Updatability:	Read Write

icom:organizer

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

icom:participant

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

icom_cal:occurrenceStatus

Description:	Status of an occurrence series.
Required:	True
Inherited:	False
Property Type:	icom_cal:OccurrenceStatus
Cardinality:	Single
Updatability:	Read Write

icom_cal:occurrenceType

Description:	Type of an occurrence series.
Required:	True
Inherited:	False
Property Type:	icom_cal:OccurrenceType
Cardinality:	Single
Updatability:	Read Write

icom_cal:mode

Description:	Mutability mode of an occurrence series.
Required:	False
Inherited:	False
Property Type:	icom_cal:OccurrenceEditMode
Cardinality:	Single
Updatability:	Read Only

icom_cal:occurrence

Description:	Occurrences in an occurrence series.
Required:	False
Inherited:	False
Property Type:	icom_cal:Occurrence
Cardinality:	Multi
Updatability:	Read Only

icom_content:attachment

Description:	One or more simple content attachments in an occurrence series.
Required:	False
Inherited:	False
Property Type:	icom_content:Attachment
Cardinality:	Multi
Updatability:	Read Write

icom_cal:attendee

Description:	An attendee of an occurrence series.
Required:	False
Inherited:	False
Property Type:	icom:Participant
Cardinality:	Single
Updatability:	Read Only

icom_cal:attendeePriority

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

icom_cal:attendeeParticipantStatus

Description:	Participation status for an attendee of an occurrence series.
Required:	False
Inherited:	False
Property Type:	icom_cal:OccurrenceParticipantStatus
Cardinality:	Single
Updatability:	Read Write

icom_cal:attendeeTransparency

Description:	Participant transparency for an attendee of an occurrence series.
Required:	False
Inherited:	False
Property Type:	icom_cal:ParticipantTransparency
Cardinality:	Single
Updatability:	Read Write

icom_cal:attendeeProperty

Description:	Extensible properties for an attendee of an occurrence series.
Required:	False
Inherited:	False
Property Type:	icom:Property
Cardinality:	Multi
Updatability:	Read Write

icom_cal:conference

Description:	One or more conferences for an occurrence series.
Required:	False
Inherited:	False
Property Type:	icom_conf:Conference
Cardinality:	Multi
Updatability:	Read Write

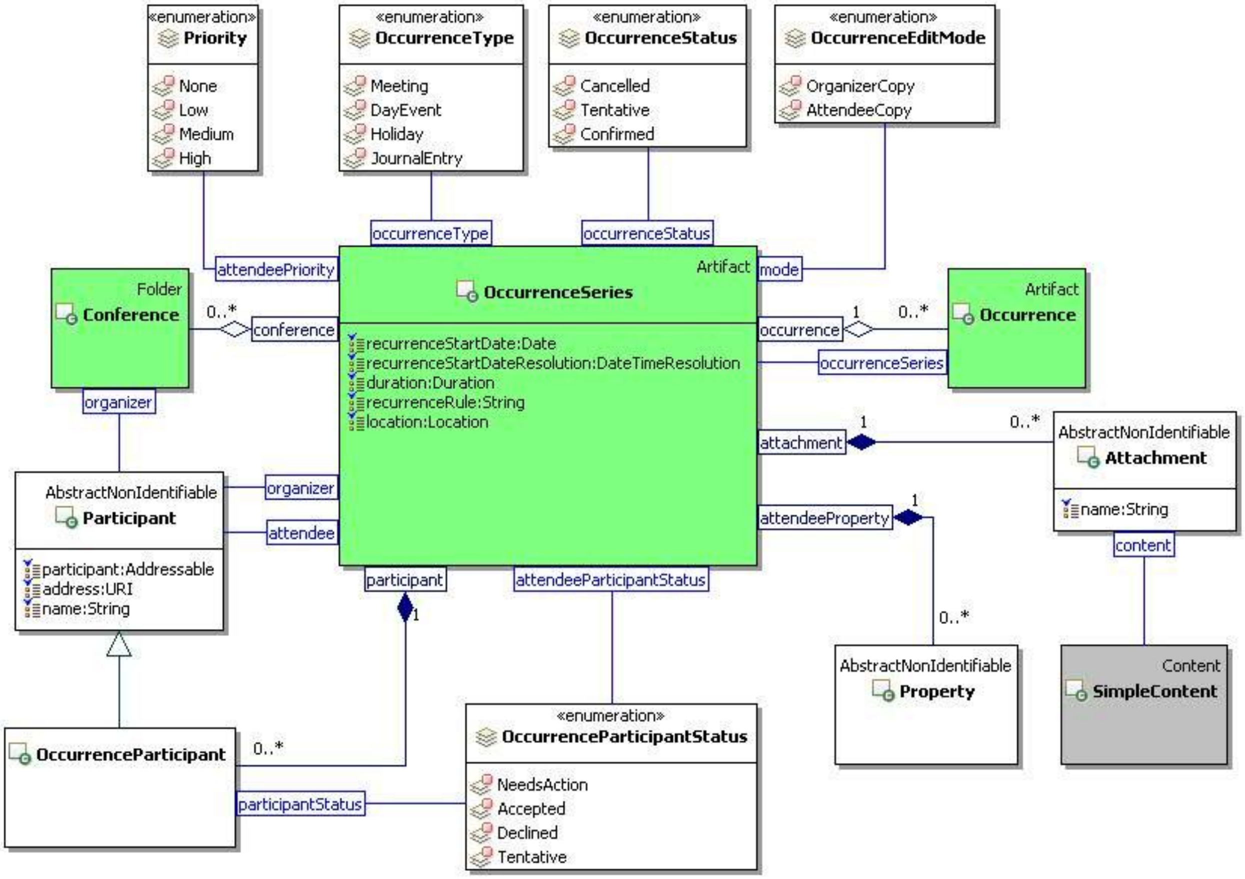

Figure 34 OccurrenceSeries Class Diagram.

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`

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in 4.6.3.3.

4.6.3.3 Property Definitions

The Occurrence class inherits property definitions from super classes.

The Occurrence class MUST have the property definitions:

icom_cal:startDate

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

icom_cal:startDateResolution

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

icom_cal:endDate

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

icom_cal:endDateResolution

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

Updatability:	On Create

icom:location

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

icom_cal:occurrenceSeries

Description:	An occurrence series that includes an occurrence.
Required:	False
Inherited:	False
Property Type:	icom_cal:OccurrenceSeries
Cardinality:	Single
Updatability:	Read Only

icom_cal:fromRecurringOccurrenceSeries

Description:	Occurrence is part of a recurring occurrence series.
Required:	False
Inherited:	False
Property Type:	Boolean
Cardinality:	Single
Updatability:	Read Only

icom_cal:exceptionToOccurrenceSeries

Description:	Occurrence is an exception to an occurrence series.
Required:	False
Inherited:	False
Property Type:	Boolean
Cardinality:	Single
Updatability:	Read Only

icom:organizer

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

icom:participant

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

icom_cal:occurrenceStatus

Description:	Status of an occurrence.
Required:	True
Inherited:	False
Property Type:	icom_cal:OccurrenceStatus
Cardinality:	Single
Updatability:	Read Write

icom_cal:occurrenceType

Description:	Type of an occurrence.
Required:	True
Inherited:	False
Property Type:	icom_cal:OccurrenceType
Cardinality:	Single
Updatability:	Read Write

icom_cal:mode

Description:	Mutability mode of an occurrence.
Required:	False
Inherited:	False
Property Type:	icom_cal:OccurrenceEditMode
Cardinality:	Single
Updatability:	Read Only

icom_content:attachment

Description:	One or more simple content attachments in an occurrence.
Required:	False
Inherited:	False
Property Type:	icom_content:Attachment
Cardinality:	Multi
Updatability:	Read Write

icom_cal:attendee

Description:	An attendee of an occurrence.

Required:	False
Inherited:	False
Property Type:	icom:Participant
Cardinality:	Single
Updatability:	Read Only

icom_cal:attendeePriority

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

icom_cal:attendeeParticipantStatus

Description:	Participation status for an attendee of an occurrence.
Required:	False
Inherited:	False
Property Type:	icom_cal:OccurrenceParticipantStatus
Cardinality:	Single
Updatability:	Read Write

icom_cal:attendeeTransparency

Description:	Participant transparency for an attendee of an occurrence.
Required:	False
Inherited:	False
Property Type:	icom_cal:ParticipantTransparency
Cardinality:	Single
Updatability:	Read Write

icom_cal:attendeeProperty

Description:	Extensible properties for an attendee of an occurrence.
Required:	False
Inherited:	False
Property Type:	icom:Property
Cardinality:	Multi
Updatability:	Read Write

icom_cal:conference

Description:	One or more conferences for an occurrence.
Required:	False
Inherited:	False

Property Type: icom_conf:Conference
Cardinality: Multi
Updatability: Read Write

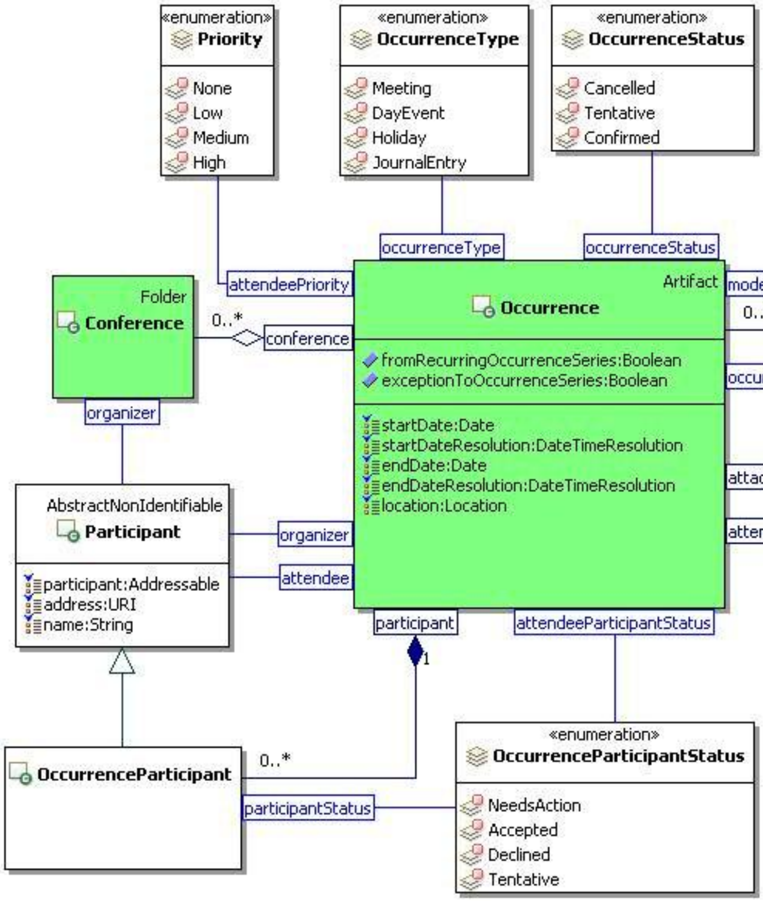

Figure 35 Occurrence Class Diagram.

4.6.4 OccurrenceStatus

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

The OccurrenceStatus class is defined by the attribute values:

localNamespace

Value: icom_cal

localName

Value: OccurrenceStatus

extendsFrom

Value:

stereotype

Value: primary

isEnumeration

Value: TRUE

description

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

instances

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

There are three occurrence status defined by ICOM:

- **icom_cal:Cancelled** to express that an occurrence or occurrence series is cancelled.
- **icom_cal:Tentative** to express that an occurrence or occurrence series is tentative.
- **icom_cal:Confirmed** to express that an occurrence or occurrence series is confirmed.

4.6.5 OccurrenceType

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

The OccurrenceType class is defined by the attribute values:

localNamespace

Value: icom_cal

localName

Value: OccurrenceType

extendsFrom

Value:

stereotype

Value: primary

isEnumeration

Value: TRUE

description

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

instances

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

There are four occurrence types defined by ICOM:

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

4.6.6 OccurrenceEditMode

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

The OccurrenceEditMode class is defined by the attribute values:

localNamespace

Value: icom_cal

localName

Value: OccurrenceEditMode

extendsFrom

Value:

stereotype

Value: primary

isEnumeration

Value: TRUE

description

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

instances

Value: <icom_cal:OrganizerCopy, icom_cal:AttendeeCopy>

There are two occurrence editable modes defined by ICOM:

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

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

localName

Value: OccurrenceParticipantStatus

extendsFrom

Value:

stereotype

Value: primary

isEnumeration

Value: TRUE

description

Value: An enumeration of the instances each of which expresses a participant's response status for an occurrence or occurrence series.

instances

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

There are four occurrence participant's status defined by ICOM:

- **icom_cal:NeedsAction** to express that an attendee needs to act on an occurrence or occurrence series.
- **icom_cal:Accepted** to express that an attendee accepted an occurrence or occurrence series.
- **icom_cal:Declined** to express that an attendee declined an occurrence or occurrence series.
- **icom_cal:Tentative** to express that an attendee is tentative about attending an occurrence or occurrence series.

4.7 FreeBusy Module

4.7.1 FreeBusy

4.7.1.1 Description

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

4.7.1.2 Class Definition

The FreeBusy class is defined by the attribute values:

localNamespace

Value: icom_cal

localName

Value: FreeBusy

`extendsFrom`

Value:

`stereotype`

Value: primary

`description`

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

`propertyDefinitions`

The values for this attribute are defined in Section 4.7.1.3.

4.7.1.3 Property Definitions

The FreeBusy class MUST have the property definitions:

icom_cal:creationDate

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

icom_cal:startDate

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

icom_cal:endDate

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

icom_cal:interval

Description:	A list of free busy intervals.
Required:	False

Inherited:	False
Property Type:	icom_cal:FreeBusyInterval
Cardinality:	Multi
Updatability:	Read Only

icom_cal:participant

Description:	A list of participants whose free busy intervals are merged.
Required:	False
Inherited:	False
Property Type:	icom:Participant
Cardinality:	Multi
Updatability:	Read Only

4.7.2 FreeBusyInterval

4.7.2.1 Description

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

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

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

4.7.2.2 Class Definition

The FreeBusyInterval class is defined by the attribute values:

`localNamespace`

Value: icom_cal

`localName`

Value: FreeBusy

`extendsFrom`

Value:

`stereotype`

Value: primary

`description`

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

`propertyDefinitions`

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

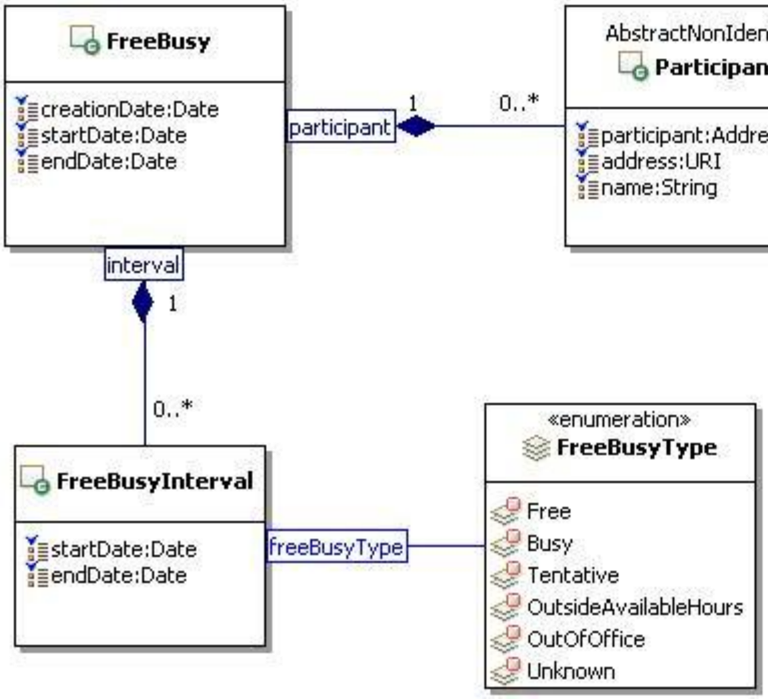

Figure 36 FreeBusy Class Diagram.

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.

instances

Value: <icom_cal:Free, icom_cal:Busy, icom_cal:Tentative, icom_cal:OutsideAvailableHours, icom_cal:OutOfOffice, icom_cal:Unknown>

There are six free busy types defined by ICOM:

- **icom_cal:Free** to express that a free busy interval is free.
- **icom_cal:Busy** to express that a free busy interval is busy.
- **icom_cal:Tentative** to express that a free busy interval is tentative.
- **icom_cal:OutsideAvailableHours** to express that a free busy interval is outside available hours.
- **icom_cal:OutOfOffice** to express that a free busy interval is coincides with out of office hours.
- **icom_cal:Unknown** to express that a free busy interval is unknown.

4.8 TaskList Module

4.8.1 TaskList

4.8.1.1 Description

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

4.8.1.2 Class Definition

The TaskList class is defined by the attribute values:

localNamespace

Value: icom_cal

localName

Value: TaskList

extendsFrom

Value: icom:Folder

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in 4.8.1.3.

4.8.1.3 Property Definitions

The TaskList class inherits property definitions from super classes.

The TaskList class MUST have the property definitions:

icom_cal:timeZone

Description:	Time zone of a task list.
Required:	True
Inherited:	False
Property Type:	TimeZone
Cardinality:	Single
Updatability:	Read Write

icom:element

Description:	Elements of a task list.
Required:	False
Inherited:	True
Property Type:	icom_cal:Task
Cardinality:	Multi
Updatability:	Read Only

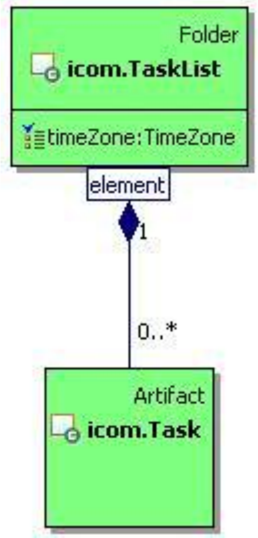

Figure 37 TaskList Class Diagram.

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

localName

Value: Task

extendsFrom

Value: icom:Artifact

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in 4.8.2.3.

4.8.2.3 Property Definitions

The Task class inherits property definitions from super classes.

The Task class MUST have the property definitions:

icom_cal:startDate

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

icom_cal:startDateResolution

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

icom_cal:dueDate

Description:	Due date and time of a task.
Required:	True
Inherited:	False
Property Type:	DateTime
Cardinality:	Single
Updatability:	On Create

icom_cal:dueDateResolution

Description:	Resolution of due date and time of a task.
Required:	True
Inherited:	False
Property Type:	icom:DateTimeResolution
Cardinality:	Single
Updatability:	On Create

icom:location

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

icom:organizer

Description:	Organizer of a task.
Required:	True
Inherited:	False
Property Type:	icom:Participant
Cardinality:	Single
Updatability:	On Create

icom_cal:taskStatus

Description:	Status of a task.
Required:	True
Inherited:	False
Property Type:	icom_cal:TaskStatus
Cardinality:	Single
Updatability:	Read Write

icom_cal:mode

Description:	Mutability mode of a task.
Required:	False
Inherited:	False
Property Type:	icom_cal:TaskEditMode
Cardinality:	Single
Updatability:	Read Only

icom_content:attachment

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

icom_cal:assignee

Description: An assignee of a task.
Required: False
Inherited: False
Property Type: icom:Participant
Cardinality: Single
Updatability: Read Only

icom_cal:assigneePriority

Description: Priority for an assignee of a task.
Required: False
Inherited: False
Property Type: icom:Priority
Cardinality: Single
Updatability: Read Write

icom_cal:assigneeParticipantStatus

Description: Participation status for an assignee of a task.
Required: False
Inherited: False
Property Type: icom_cal:TaskParticipantStatus
Cardinality: Single
Updatability: Read Write

icom_cal:assigneeCompletionDate

Description: Completion date and time of a task.
Required: False
Inherited: False
Property Type: DateTime
Cardinality: Single
Updatability: Read Write

icom_cal:assigneeCompletionDateResolution

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

Inherited:	False
Property Type:	icom:DateTimeResolution
Cardinality:	Single
Updatability:	Read Write

icom_cal:assigneePercentComplete

Description:	Percentage of task completed.
Required:	False
Inherited:	False
Property Type:	Integer
Cardinality:	Single
Updatability:	Read Write

icom_cal:assigneeProperty

Description:	Extensible properties for an assignee of a task.
Required:	False
Inherited:	False
Property Type:	icom:Property
Cardinality:	Multi
Updatability:	Read Write

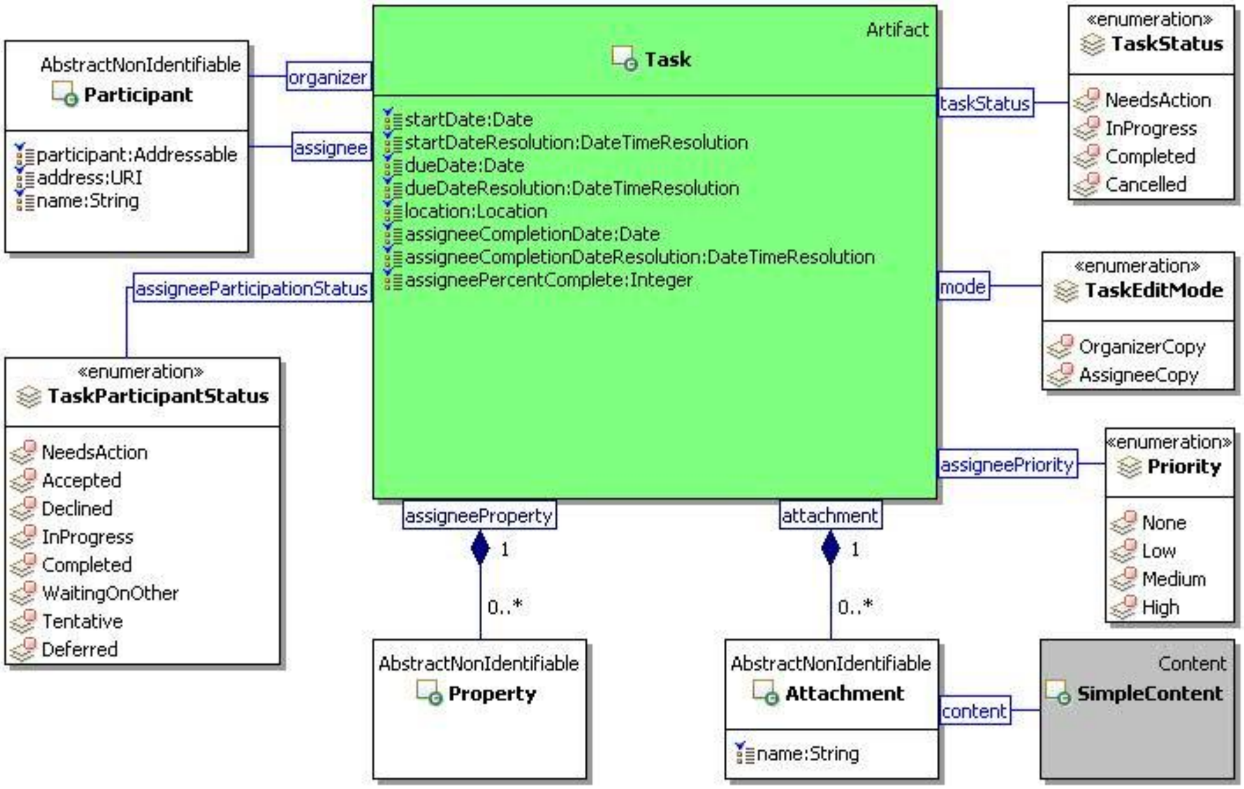

Figure 38 Task Class Diagram.

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

extendsFrom

Value:

stereotype

Value: primary

isEnumeration

Value: TRUE

description

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

instances

Value: <icom_cal:OrganizerCopy, icom_cal:AssigneeCopy>

There are two task editable modes defined by ICOM:

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

4.8.5 TaskParticipantStatus

The TaskParticipantStatus class is an enum class that enumerates the instances each of which expresses a participant's response status for a task.

The TaskParticipantStatus class is defined by the attribute values:

localNamespace

Value: icom_cal

localName

Value: TaskParticipantStatus

extendsFrom

Value:

stereotype

Value: primary

isEnumeration

Value: TRUE

description

Value: An enumeration of the instances each of which expresses a participant's response status for a task.

instances

Value: <icom_cal:NeedsAction, icom_cal:Accepted, icom_cal:Declined, icom_cal:InProgress, icom_cal:Completed, icom_cal:WaitingOnOther, icom_cal:Tentative, icom_cal:Deferred>

There are eight task participant's status defined by ICOM:

- **icom_cal:NeedsAction** to express that an assignee needs to act on a task.
- **icom_cal:Accepted** to express that an assignee accepted a task.
- **icom_cal:Declined** to express that an assignee declined a task.
- **icom_cal:InProgress** to express that a task is in progress.
- **icom_cal:Completed** to express that a task is completed.
- **icom_cal:WaitingOnOther** to express that an assignee is waiting on other.
- **icom_cal: Tentative** to express that an assignee is tentative about a task status.
- **icom_cal:Deferred** to express that an assignee deferred a task.

4.9 Forum Module

4.9.1 Discussion

4.9.1.1 Description

A discussion is an item in a discussion container.

4.9.1.2 Class Definition

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

The Discussion class is defined by the attribute values:

localNamespace

Value: icom_forum

localName

Value: Discussion

extendsFrom

Value: icom:Item

stereotype

Value: mixin

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.9.1.3.

4.9.1.3 Property Definitions

The Discussion class inherits property definitions from super classes.

The Discussion class MUST have the property definitions:

icom_forum:inReplyTo

Description:	Another discussion object that a discussion object is replying to.
Required:	False
Inherited:	False
Property Type:	icom_forum:Discussion
Cardinality:	Single
Updatability:	Read Write

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

4.9.2 DiscussionContainer

4.9.2.1 Description

A discussion container is a container of discussions.

4.9.2.2 Class Definition

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

The DiscussionContainer class is defined by the attribute values:

localNamespace

Value: icom_forum

localName

Value: DiscussionContainer

extendsFrom

Value: icom:Container

stereotype

Value: mixin

`description`

Value: DiscussionContainer is a mixin class which defines the characteristics of entities that contain Discussion items.

`propertyDefinitions`

The values for this attribute are defined in Section 4.9.2.3.

4.9.2.3 Property Definitions

The DiscussionContainer class inherits property definitions from super classes.

The DiscussionContainer class MUST have the property definitions:

icom_forum:element

Description:	Elements of a discussion container.
Required:	False
Inherited:	True
Property Type:	icom_forum:Discussion
Cardinality:	Multi
Updatability:	Read Only

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

4.9.3 DiscussionMessage

4.9.3.1 Description

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

4.9.3.2 Class Definition

The DiscussionMessage class is defined by the attribute values:

`localNamespace`

Value: icom_forum

`localName`

Value: DiscussionMessage

`extendsFrom`

Value: icom_msg:Message, icom_forum:Discussion

`stereotype`

Value: primary

`description`

Value: Discussion message is a message in a forum discussion thread.

propertyDefinitions

The values for this attribute are defined in Section 4.9.3.3.

4.9.3.3 Property Definitions

The DiscussionMessage class inherits property definitions from super classes.

The DiscussionMessage class MUST have the property definitions:

icom_forum:inReplyTo

Description:	Another discussion message that a discussion message is replying to.
Required:	False
Inherited:	True
Property Type:	icom_forum:DiscussionMessage
Cardinality:	Single
Updatability:	Read Write

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

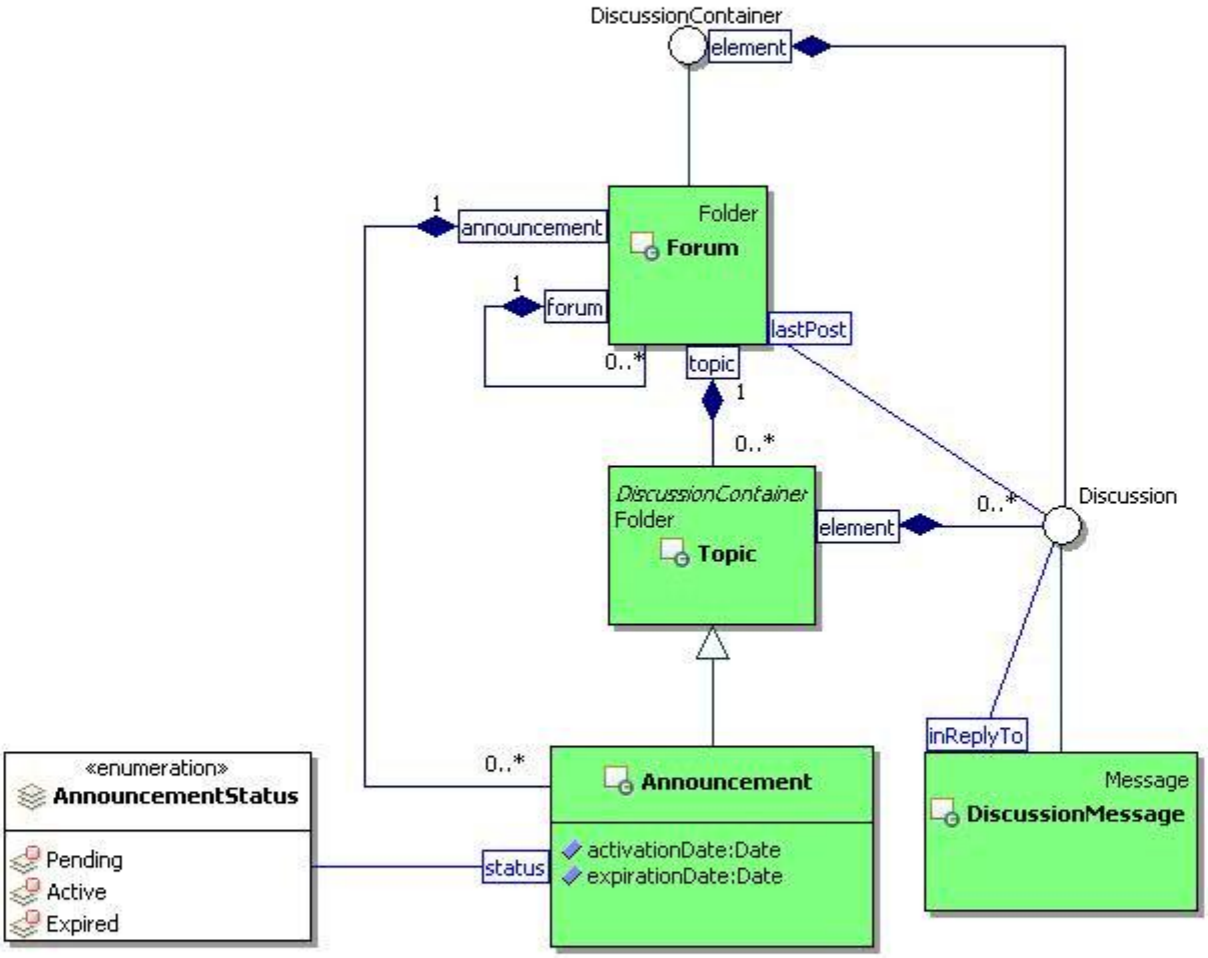

Figure 39 Forum Class Diagram

4.9.4 Forum

4.9.4.1 Description

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

4.9.4.2 Class Definition

The Forum class is defined by the attribute values:

localNamespace

Value: icom_forum

localName

Value: Forum

extendsFrom

Value: icom:Folder

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.9.4.3.

4.9.4.3 Property Definitions

The Forum class inherits property definitions from super classes.

The Forum class MUST have the property definitions:

icom_forum:lastPost

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

icom_forum:forum

Description:	Sub-forums of a forum.
Required:	False
Inherited:	False
Property Type:	icom_forum:Forum
Cardinality:	Multi
Updatability:	Read Only

icom_forum:topic

Description:	Topics of a forum.
Required:	False
Inherited:	False
Property Type:	icom_forum:Topic
Cardinality:	Multi
Updatability:	Read Only

icom_forum:announcement

Description:	Announcements of a forum.
Required:	False
Inherited:	False
Property Type:	icom_forum:Announcement

Cardinality:	Multi
Updatability:	Read Only

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

4.9.5 Topic

4.9.5.1 Description

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

4.9.5.2 Class Definition

The Topic class is defined by the attribute values:

`localNamespace`

Value: icom_forum

`localName`

Value: Topic

`extendsFrom`

Value: icom:Folder, icom_forum:DiscussionContainer

`stereotype`

Value: primary

`description`

Value: A topic is a folder that contains discussion threads.

`propertyDefinitions`

The values for this attribute are defined in Section 4.9.5.3.

4.9.5.3 Property Definitions

The Topic class inherits property definitions from super classes.

The Topic class MUST have the property definitions:

icom:element

Description:	Elements of a topic.
Required:	False
Inherited:	True
Property Type:	icom_forum:Discussion
Cardinality:	Multi
Updatability:	Read Only

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.

`propertyDefinitions`

The values for this attribute are defined in Section 4.9.6.3.

4.9.6.3 Property Definitions

The Announcement class inherits property definitions from super classes.

The Announcement class MUST have the property definitions:

icom_forum:activationDate

Description:	Date and time when an announcement becomes active.
Required:	False
Inherited:	False
Property Type:	DateTime
Cardinality:	Single
Updatability:	Read Write

icom_forum:expirationDate

Description:	Date and time when an announcement expires.
Required:	False
Inherited:	False
Property Type:	DateTime
Cardinality:	Single
Updatability:	Read Write

icom_forum:announcementStatus

Description:	Status of an announcement.
Required:	True
Inherited:	False
Property Type:	icom_forum:AnnouncementStatus
Cardinality:	Single
Updatability:	Read Write

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

4.9.7 AnnouncementStatus

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

The AnnouncementStatus class is defined by the attribute values:

`localNamespace`

Value: icom_forum

localName

Value: AnnouncementStatus

extendsFrom

Value:

stereotype

Value: primary

isEnumeration

Value: TRUE

description

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

instances

Value: <icom_forum:Pending, icom_forum:Active, icom_forum:Expired>

There are three announcement status defined by ICOM:

- **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.

4.10 Conference Module

4.10.1 Conference

4.10.1.1 Description

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

4.10.1.2 Class Definition

The Conference class is defined by the attribute values:

localNamespace

Value: icom_conf

localName

Value: Conference

extendsFrom

Value: icom:Folder

`stereotype`

Value: primary

`description`

Value: A conference is a folder that represents a durable context for online conference sessions.

`propertyDefinitions`

The values for this attribute are defined in Section 4.10.1.3.

4.10.1.3 Property Definitions

The Conference class inherits property definitions from super classes.

The Conference class MUST have the property definitions:

icom_conf:organizer

Description:	Organizer of a conference.
Required:	False
Inherited:	False
Property Type:	icom_cond:Participant
Cardinality:	Single
Updatability:	On Create

icom_conf:conferenceType

Description:	Type of a conference.
Required:	False
Inherited:	False
Property Type:	icom_conf:ConferenceType
Cardinality:	Single
Updatability:	Read Write

icom_conf:conferenceState

Description:	Session state of a conference.
Required:	False
Inherited:	False
Property Type:	icom_conf:ConferenceState
Cardinality:	Single
Updatability:	Read Only

icom_conf:runningSession

Description:	Current session of a conference.
Required:	False
Inherited:	False
Property Type:	icom_conf:ConferenceSession

Cardinality:	Single
Updatability:	Read Only

icom_conf:conferenceSetting

Description:	Configurable settings of a conference.
Required:	False
Inherited:	False
Property Type:	icom_conf:ConferenceSetting
Cardinality:	Single
Updatability:	Read Only

icom_conf:transcript

Description:	Transcripts from ended sessions of a conference.
Required:	False
Inherited:	False
Property Type:	icom_doc:Document
Cardinality:	Multi
Updatability:	Read Write

icom_conf:scheduledStartTime

Description:	Scheduled start time of the current or next session of a conference.
Required:	False
Inherited:	False
Property Type:	Date
Cardinality:	Single
Updatability:	Read Write

icom_conf:scheduledEndTime

Description:	Scheduled end time of the current or next session of a conference.
Required:	False
Inherited:	False
Property Type:	Date
Cardinality:	Single
Updatability:	Read Write

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

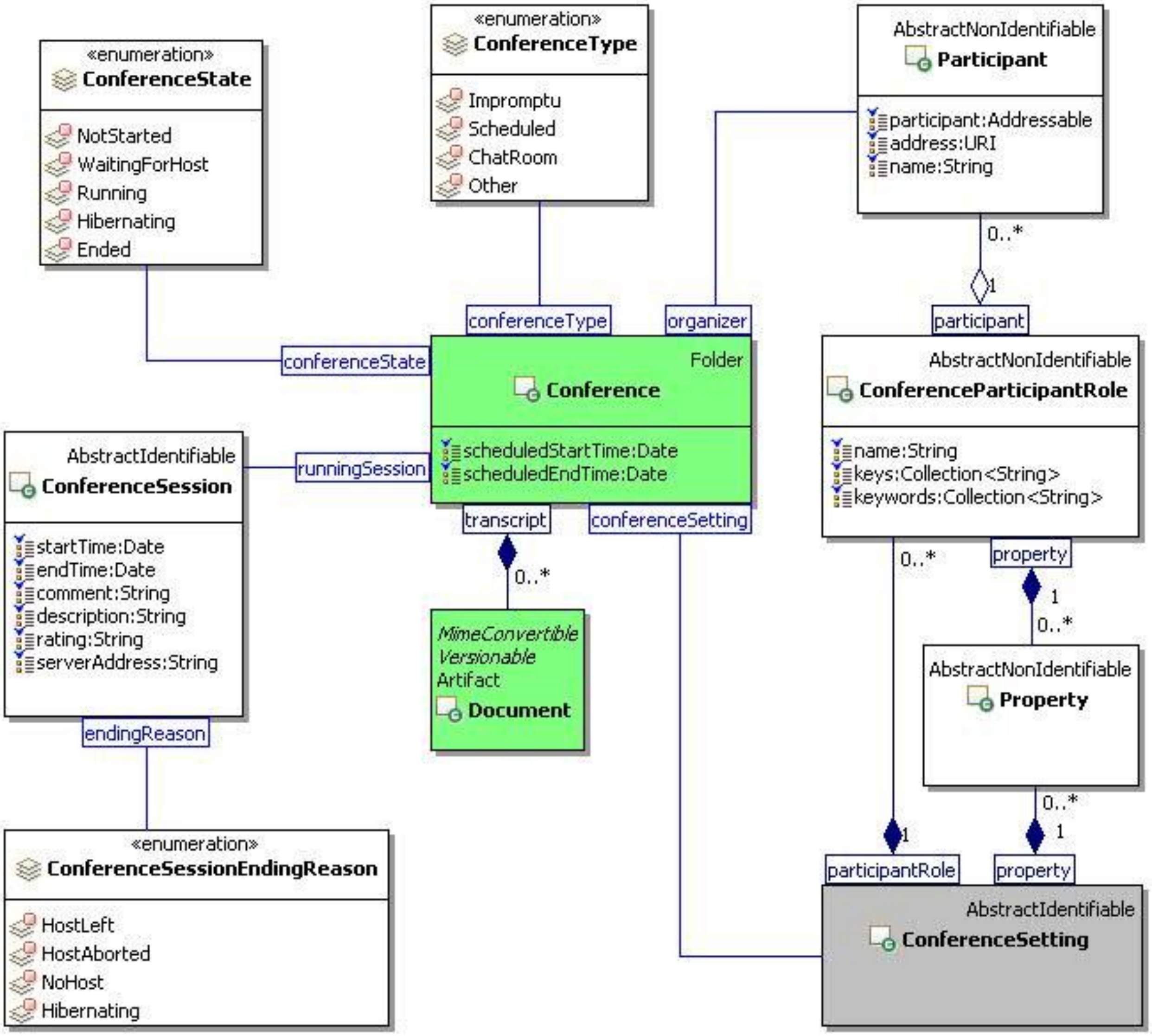

Figure 40 Conference Class Diagram.

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:

stereotype

Value: primary

isEnumeration

Value: TRUE

description

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

instances

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

There are four conference types defined by ICOM:

- **icom_conf:Impromptu** to express that a conference session is started impromptu .
- **icom_conf:Scheduled** to express that a conference session is scheduled.
- **icom_conf:ChatRoom** to express that a conference is used for a chat room.
- **icom_conf:Other** to express that a conference is of other type.

4.10.3 ConferenceState

The ConferenceState class is an enum class that enumerates the instances each of which expresses a session state of a conference.

The ConferenceState class is defined by the attribute values:

localNamespace

Value: icom_conf

localName

Value: ConferenceState

extendsFrom

Value:

stereotype

Value: primary

isEnumeration

Value: TRUE

description

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

instances

Value: <icom_conf:NotStarted, icom_conf:WaitingForHost, icom_conf:Running, icom_conf:Hibernatimg, icom_conf:Ended>

There are five conference session states defined by ICOM:

- **icom_conf:NotStarted** to express that a conference session is not started .
- **icom_conf:WaitingForHost** to express that a conference session is waiting for a host.
- **icom_conf:Running** to express that a conference session is running.
- **icom_conf:Hibernating** to express that a conference session is hibernating.
- **icom_conf:Ended** to express that a conference session is ended.

4.10.4 ConferenceSession

4.10.4.1 Description

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

4.10.4.2 Class Definition

The ConferenceSession class is defined by the attribute values:

localNamespace

Value: icom_conf

localName

Value: ConferenceSession

extendsFrom

Value: icom:Identifiable

stereotype

Value: primary

description

Value: A conference session represents the metadata for a runtime session of a conference.

propertyDefinitions

The values for this attribute are defined in Section 4.10.4.3.

4.10.4.3 Property Definitions

The ConferenceSession class inherits property definitions from super classes.

The ConferenceSession class MUST have the property definitions:

icom_conf:startTime

Description:	Start time of a conference session.
Required:	False
Inherited:	False
Property Type:	Date
Cardinality:	Single
Updatability:	Read Only

icom_conf:endTime

Description:	End time of a conference session.
Required:	False
Inherited:	False
Property Type:	Date
Cardinality:	Single
Updatability:	Read Only

icom_conf:comment

Description:	Comment on a conference session.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

icom_conf:description

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

icom_conf:rating

Description:	Rating of a conference session.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

icom_conf:serverAddress

Description:	Address of a server that hosts a conference session.

Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Only

icom_conf:endingReason

Description:	Reason for ending a conference session.
Required:	False
Inherited:	False
Property Type:	icom_conf:ConferenceSessionEndingReason
Cardinality:	Single
Updatability:	Read Only

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

4.10.5 ConferenceSessionEndingReason

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

The ConferenceSessionEndingReason class is defined by the attribute values:

localNamespace

Value: icom_conf

localName

Value: ConferenceSessionEndingReason

extendsFrom

Value:

stereotype

Value: primary

isEnumeration

Value: TRUE

description

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

instances

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

There are four conference session states defined by ICOM:

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

4.10.6 ConferenceSetting

4.10.6.1 Description

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

4.10.6.2 Class Definition

The ConferenceSetting class is defined by the attribute values:

localNamespace

Value: icom_conf

localName

Value: ConferenceSetting

extendsFrom

Value:

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.10.6.3.

4.10.6.3 Property Definitions

The ConferenceSetting class inherits property definitions from super classes.

The ConferenceSetting class MUST have the property definitions:

icom_conf:participantRole

Description:	Role settings for conference participants.
Required:	False
Inherited:	False
Property Type:	icom_conf:ConferenceParticipantRole

Cardinality:	Multi
Updatability:	Read Write

icom_conf:property

Description:	Configurable properties for a conference.
Required:	False
Inherited:	False
Property Type:	icom:property
Cardinality:	Multi
Updatability:	Read Write

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

4.10.7 ConferenceParticipantRole

4.10.7.1 Description

A conference participant role contains roles settings for a conference.

4.10.7.2 Class Definition

The ConferenceParticipantRole class is defined by the attribute values:

localNamespace

Value: icom_conf

localName

Value: ConferenceParticipantRole

extendsFrom

Value:

stereotype

Value: primary

description

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

propertyDefinitions

The values for this attribute are defined in Section 4.10.7.3.

4.10.7.3 Property Definitions

The ConferenceParticipantRole class MUST have the property definitions:

icom_conf:name

Description:	Name of a role setting in a conference.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Single
Updatability:	Read Write

icom_conf:key

Description:	One or more sign on keys to activate a role setting.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Multi
Updatability:	Read Write

icom_conf:keyword

Description:	One or more key words to activate a role setting.
Required:	False
Inherited:	False
Property Type:	String
Cardinality:	Multi
Updatability:	Read Write

icom_conf:participant

Description:	One or more participants in a role setting.
Required:	False
Inherited:	False
Property Type:	icom:Participant
Cardinality:	Multi
Updatability:	Read Write

icom_conf:property

Description:	Configurable properties for a role setting.
Required:	False
Inherited:	False
Property Type:	icom:Property
Cardinality:	Multi
Updatability:	Read Write

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

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

B. Non-Normative Text

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