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

Committee Specification Draft 05

27 October 2012

Specification URIs
This version:
http://docs.oasis-open.org/icom/icom-ics/v1.0/csd05/icom-ics-v1.0-csd05.doc (Authoritative)
http://docs.oasis-open.org/icom/icom-ics/v1.0/csd05/icom-ics-v1.0-csd05.html
http://docs.oasis-open.org/icom/icom-ics/v1.0/csd05/icom-ics-v1.0-csd05.pdf

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

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

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

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

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

Additional artifacts:
This prose specification is one component of a Work Product which also includes:
- XML schemas: http://docs.oasis-open.org/icom/icom-ics/v1.0/csd05/schemas/

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

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

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

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

Status:

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

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

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

Citation format:

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

[ICOM-ics-v1.0]

Integrated Collaboration Object Model (ICOM) for Interoperable Collaboration Services Version 1.0. 27 October 2012. OASIS Committee Specification Draft 05.

http://docs.oasis-open.org/icom/icom-ics/v1.0/csd05/icom-ics-v1.0-csd05.html.
Notices

Copyright © OASIS Open 2012. All Rights Reserved.

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

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

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

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

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

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

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

The name “OASIS” is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see https://www.oasis-open.org/policies-guidelines/trademark for above guidance.
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>1.1 Terminology</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>1.2 Normative References</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>1.3 Non-Normative References</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>Modeling Language</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>2.1 Introduction</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>2.2 Class Definition Grammar</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>2.3 Property Definition Grammar</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>2.4 Namespaces</td>
<td>19</td>
</tr>
<tr>
<td>3</td>
<td>Core Model</td>
<td>20</td>
</tr>
<tr>
<td>3.1</td>
<td>Main Branch</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>3.1.1 Entity and Top-Level Subclasses</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>3.1.2 Identifiable</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>3.1.3 Parental</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>3.1.4 Extent</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>3.1.5 Entity</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>3.1.6 Overview of Scope, Subject, and Artifact Branches</td>
<td>27</td>
</tr>
<tr>
<td>3.2</td>
<td>Scope Branch</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>3.2.1 Scope and Top-Level Subclasses</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>3.2.2 Scope</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>3.2.3 Community</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>3.2.4 Space</td>
<td>33</td>
</tr>
<tr>
<td>3.3</td>
<td>Subject Branch</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>3.3.1 Subject and Top-Level Subclasses</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>3.3.2 Subject</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>3.3.3 Group</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>3.3.4 Actor</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>3.3.5 Person</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>3.3.6 Resource</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>3.3.7 ResourceType</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>3.3.8 ResourceTypeEnum</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>3.3.9 ResourceBookingRule</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>3.3.10 ResourceBookingRuleEnum</td>
<td>50</td>
</tr>
<tr>
<td>3.4</td>
<td>Artifact Branch</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>3.4.1 Artifact and Top-Level Subclasses</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>3.4.2 Item</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>3.4.3 SpaceItem</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>3.4.4 Container</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>3.4.5 FolderContainer</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>3.4.6 Artifact</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>3.4.7 Folder</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>3.4.8 HeterogeneousFolder</td>
<td>59</td>
</tr>
<tr>
<td>3.5</td>
<td>Access Control Model</td>
<td>61</td>
</tr>
</tbody>
</table>
3.5.1 Accessor............................................................................................................. 61
3.5.2 Owner .................................................................................................................. 61
3.5.3 RoleDefinition ..................................................................................................... 62
3.5.4 Role ..................................................................................................................... 63
3.5.5 Privilege .............................................................................................................. 65
3.5.6 PrivilegeEnum ..................................................................................................... 66
3.5.7 AccessControlList ............................................................................................... 67
3.5.8 AccessControlEntry ............................................................................................ 68
3.5.9 AccessType .......................................................................................................... 69
3.5.10 AccessTypeEnum ............................................................................................... 70
3.6 Metadata Model ...................................................................................................... 71
  3.6.1 ClassDefinition .................................................................................................. 71
  3.6.2 Stereotype ......................................................................................................... 74
  3.6.3 StereotypeEnum .................................................................................................. 75
  3.6.4 PropertyDefinition .............................................................................................. 76
  3.6.5 Property ............................................................................................................. 79
  3.6.6 PropertyChoiceType ............................................................................................. 81
  3.6.7 PropertyType ...................................................................................................... 82
  3.6.8 PropertyTypeEnum ............................................................................................... 83
  3.6.9 Updatability ....................................................................................................... 84
  3.6.10 UpdatabilityEnum ............................................................................................. 85
  3.6.11 Cardinality ....................................................................................................... 86
  3.6.12 CardinalityEnum .............................................................................................. 86
  3.6.13 Marker and Subclasses .................................................................................... 87
  3.6.14 Marker ............................................................................................................ 88
  3.6.15 Category ........................................................................................................ 89
  3.6.16 CategoryApplication ....................................................................................... 91
  3.6.17 Tag .................................................................................................................. 92
  3.6.18 TagApplication ................................................................................................ 94
  3.6.19 RelationshipBondable ..................................................................................... 95
  3.6.20 RelationshipDefinition ................................................................................... 96
  3.6.21 Relationship ................................................................................................... 98
3.7 Common Concepts .................................................................................................. 100
  3.7.1 Addressable ..................................................................................................... 100
  3.7.2 EntityAddress ................................................................................................... 101
  3.7.3 Participant ........................................................................................................ 102
  3.7.4 Priority ............................................................................................................ 104
  3.7.5 PriorityEnum .................................................................................................... 105
  3.7.6 DateTimeResolution ......................................................................................... 105
  3.7.7 DateTimeResolutionEnum .................................................................................. 106
  3.7.8 TimeZone .......................................................................................................... 107
  3.7.9 Location .......................................................................................................... 108
  3.7.10 GeoCoordinates ............................................................................................. 110
4 Extension Modules ................................................................................................... 112
  4.1 Overview of Extension Modules ............................................................................ 112
4.2 Content Module ................................................................. 114
  4.2.1 MimeConvertible ....................................................... 114
  4.2.2 Content ................................................................. 114
  4.2.3 MultiContent ........................................................... 116
  4.2.4 SimpleContent ......................................................... 117
  4.2.5 OnlineContent .......................................................... 119
  4.2.6 ContentDispositionType ............................................... 120
  4.2.7 ContentDispositionTypeEnum ....................................... 121
  4.2.8 AttachedItem .......................................................... 122
4.3 Document Module ............................................................ 123
  4.3.1 Versionable ............................................................. 123
  4.3.2 VersionControlMetadata .............................................. 125
  4.3.3 VersionSeries ......................................................... 127
  4.3.4 Version ................................................................. 129
  4.3.5 VersionType ............................................................ 131
  4.3.6 VersionTypeEnum ..................................................... 132
  4.3.7 Document ............................................................... 133
  4.3.8 WikiPage ............................................................... 134
4.4 Message Module .............................................................. 136
  4.4.1 Message ................................................................. 136
  4.4.2 UnifiedMessage ....................................................... 137
  4.4.3 UnifiedMessageParticipant ......................................... 142
  4.4.4 UnifiedMessageFlag .................................................. 143
  4.4.5 UnifiedMessageFlagEnum .......................................... 144
  4.4.6 UnifiedMessageDeliveryStatusNotificationRequest ............ 145
  4.4.7 UnifiedMessageDeliveryStatusNotificationRequestEnum ...... 145
  4.4.8 UnifiedMessageChannel ............................................. 146
  4.4.9 UnifiedMessageChannelEnum ...................................... 147
  4.4.10 UnifiedMessageEditMode .......................................... 148
  4.4.11 UnifiedMessageEditModeEnum .................................... 148
  4.4.12 InstantMessage ...................................................... 150
  4.4.13 InstantMessageType ............................................... 153
  4.4.14 InstantMessageTypeEnum .......................................... 154
  4.4.15 InstantMessageChatStatus ....................................... 155
  4.4.16 InstantMessageChatStatusEnum ................................... 155
  4.4.17 InstantMessageFeed ............................................... 156
  4.4.18 InstantMessageConnection ....................................... 158
4.5 Presence Module ............................................................ 161
  4.5.1 Presence .............................................................. 161
  4.5.2 PresenceEditMode ................................................... 163
  4.5.3 PresenceEditModeEnum ............................................. 164
  4.5.4 ContactMethod ....................................................... 165
  4.5.5 ContactReachabilityStatus ....................................... 167
  4.5.6 ContactReachabilityStatusEnum ................................... 167
  4.5.7 Activity ............................................................. 170
4.5.8 ActivityType ................................................................. 171
4.5.9 ActivityTypeEnum .......................................................... 172
4.6 Address Book Module ..................................................... 173
  4.6.1 AddressBook .............................................................. 173
  4.6.2 PersonContact ............................................................ 174
4.7 Calendar Module .......................................................... 179
  4.7.1 Calendar ..................................................................... 179
  4.7.2 OccurrenceSeries .......................................................... 181
  4.7.3 Occurrence ................................................................. 186
  4.7.4 OccurrenceStatus .......................................................... 192
  4.7.5 OccurrenceStatusEnum .................................................. 192
  4.7.6 OccurrenceType ............................................................ 193
  4.7.7 OccurrenceTypeEnum ...................................................... 194
  4.7.8 OccurrenceParticipant .................................................... 195
  4.7.9 OccurrenceParticipantStatus ........................................... 196
  4.7.10 OccurrenceParticipantStatusEnum ................................... 196
  4.7.11 OccurrenceParticipantTransparency ................................ 197
  4.7.12 OccurrenceParticipantTransparencyEnum ................................ 198
  4.7.13 OccurrenceEditMode ................................................... 199
  4.7.14 OccurrenceEditModeEnum ............................................ 200
4.8 Free Busy Module .......................................................... 200
  4.8.1 FreeBusy .................................................................. 200
  4.8.2 FreeBusyInterval .......................................................... 202
  4.8.3 FreeBusyType .............................................................. 204
  4.8.4 FreeBusyTypeEnum ....................................................... 205
4.9 Task List Module ............................................................ 206
  4.9.1 TaskList ................................................................. 206
  4.9.2 Task ................................................................. 207
  4.9.3 TaskStatus .............................................................. 212
  4.9.4 TaskStatusEnum ......................................................... 212
  4.9.5 TaskParticipantStatus ................................................ 213
  4.9.6 TaskParticipantStatusEnum ......................................... 214
  4.9.7 TaskEditMode ........................................................... 215
  4.9.8 TaskEditModeEnum .................................................... 215
4.10 Forum Module ............................................................... 216
  4.10.1 Discussion ............................................................. 216
  4.10.2 DiscussionContainer .................................................. 217
  4.10.3 DiscussionMessage .................................................... 218
  4.10.4 TopicContainer .......................................................... 219
  4.10.5 Forum ................................................................. 221
  4.10.6 Topic ................................................................. 223
  4.10.7 Announcement .......................................................... 224
  4.10.8 AnnouncementStatus ................................................ 226
  4.10.9 AnnouncementStatusEnum .......................................... 226
4.11 Conference Module .......................................................... 227
4.11.1 Conference .......................................................... 227
4.11.2 ConferenceType ....................................................... 230
4.11.3 ConferenceTypeEnum ................................................ 231
4.11.4 ConferenceStatus .................................................... 232
4.11.5 ConferenceStatusEnum ............................................. 233
4.11.6 ConferenceSession ................................................... 233
4.11.7 ConferenceSessionEndingReason ................................. 236
4.11.8 ConferenceSessionEndingReasonEnum .......................... 236
4.11.9 ConferenceSetting .................................................... 237
4.11.10 ConferenceParticipantRole ...................................... 238

5 Conformance ..................................................................... 241
5.1 Software Architecture or Framework Dependence .................. 241
5.2 Platform Provider Conformance ...................................... 241
  5.2.1 Platform Provider Conformance – No Extension Modules .... 241
  5.2.2 Platform Provider Conformance – One or More Extension Modules .. 241
5.3 Service Provider Conformance ........................................ 241
  5.3.1 ICOM Service Provider – No Extension Modules ............ 241
  5.3.2 ICOM Service Provider – One or More Extension Modules ... 241
5.4 ICOM Producer Conformance ........................................ 242
  5.4.1 ICOM Producer Conformance – No Extension Modules ...... 242
  5.4.2 ICOM Producer Conformance – One or More Extension Modules . 242
5.5 ICOM Consumer Conformance ....................................... 242
  5.5.1 ICOM Consumer Conformance – No Extension Modules ...... 242
  5.5.2 ICOM Consumer Conformance – Extension Modules .......... 242

Appendix A. Acknowledgements ........................................... 243
Appendix B. Revision History ................................................ 244
## Table of Figures

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

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

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

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

The model specified in ICOM is part of existing standards and technologies, several of which are referenced in Section 1.3 Non-Normative References. The model is modular and extensible, with common concepts, metadata concepts, and their relations provided in the Core, while the specific concepts and relations for each area of collaboration activities defined in separate extension modules.

ICOM core model encompasses LDAP Directory Information Models [RFC4512]. The extension modules integrate models from Content Management Interoperability Services [CMIS], Java Content Repository API [JCR 2.0], Web Distributed Authoring and Versioning (WebDAV) [RFC4918], Internet Message Access Protocol (IMAP) [RFC2119], Simple Mail Transfer Protocol (SMTP) [RFC5321], Extensible Messaging and Presence Protocol (XMPP) [RFC3920], IMAP Instant Messaging and Presence [RFC3921], vCard MIME Directory Profile [RFC2426], Internet Calendaring and Scheduling Core Object Specification (iCalendar) [RFC5545], and Calendaring Extensions to WebDAV (CalDAV) [RFC4791].

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

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


1.3 Non-Normative References


[OpenGraph] Facebook Platform Open Graph Core Concepts, (http://developers.facebook.com/docs/coreconcepts/)


## 2 Modeling Language

### 2.1 Introduction

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

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

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

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

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

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

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

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

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

### 2.2 Class Definition Grammar

A class-definition MUST contain the following attributes:

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

**propertyDefinition**

The propertyDefinition attribute defines a set of zero or more property definitions for a class. Property definitions of a class are a union of inherited property definitions from super classes and property definitions explicitly defined on a class. The order of property definitions within a class is not significant.

Property definitions MUST be uniquely named to avoid conflicts from multiple inheritances. Note: It is possible for the same property definition to be inherited through different paths in a super class hierarchy. Duplicate property definitions are eliminated from the set of property definitions of a class.

### 2.3 Property Definition Grammar

A property-definition MUST contain the following attributes:

- **namespace** String
  - The namespace attribute specifies an IRI.

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

- **description** String (optional)
  - The description attribute specifies a description of a property

- **propertyType** Enum
  - The propertyType attribute specifies a property-type for property values.
  - The value of propertyType attribute is one of the property-type names. The property-type names include names for the following data type defined by XML Schema Part 2 [XML SCHEMA]:
    - string (xsd:string)
    - boolean (xsd:boolean)
    - decimal (xsd:decimal)
    - integer (xsd:integer)
    - datetime (xsd:dateTime)
    - duration (xsd:duration)
    - id (an opaque string representing an object id of an identifiable object)
    - html (a document or fragment of Hypertext Markup Language)

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

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.

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.

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

The values of `required` attribute are:

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

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

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

The choices attribute specifies a 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.

A class definition includes the two attributes: namespace and localName. The namespace specifies one of the namespace prefixes in Table 1. The localName specifies an unprefixed name of a class.

Syntactically, the namespace qualifies the local name.

#### Table 1 Namespace prefixes and IRI references.

<table>
<thead>
<tr>
<th>Namespace Prefix</th>
<th>IRI Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>icom_core</td>
<td><a href="http://docs.oasis-open.org/ns/icom/core/201008">http://docs.oasis-open.org/ns/icom/core/201008</a></td>
</tr>
<tr>
<td>icom_ac</td>
<td><a href="http://docs.oasis-open.org/ns/icom/accesscontrol/201008">http://docs.oasis-open.org/ns/icom/accesscontrol/201008</a></td>
</tr>
<tr>
<td>icom_meta</td>
<td><a href="http://docs.oasis-open.org/ns/icom/metadata/201008">http://docs.oasis-open.org/ns/icom/metadata/201008</a></td>
</tr>
<tr>
<td>icom_content</td>
<td><a href="http://docs.oasis-open.org/ns/icom/content/201008">http://docs.oasis-open.org/ns/icom/content/201008</a></td>
</tr>
<tr>
<td>icom_doc</td>
<td><a href="http://docs.oasis-open.org/ns/icom/document/201008">http://docs.oasis-open.org/ns/icom/document/201008</a></td>
</tr>
<tr>
<td>icom_msg</td>
<td><a href="http://docs.oasis-open.org/ns/icom/message/201008">http://docs.oasis-open.org/ns/icom/message/201008</a></td>
</tr>
<tr>
<td>icom_card</td>
<td><a href="http://docs.oasis-open.org/ns/icom/contact/201008">http://docs.oasis-open.org/ns/icom/contact/201008</a></td>
</tr>
<tr>
<td>icom_presence</td>
<td><a href="http://docs.oasis-open.org/ns/icom/presence/201008">http://docs.oasis-open.org/ns/icom/presence/201008</a></td>
</tr>
<tr>
<td>icom_cal</td>
<td><a href="http://docs.oasis-open.org/ns/icom/calendar/201008">http://docs.oasis-open.org/ns/icom/calendar/201008</a></td>
</tr>
<tr>
<td>icom_task</td>
<td><a href="http://docs.oasis-open.org/ns/icom/task/201008">http://docs.oasis-open.org/ns/icom/task/201008</a></td>
</tr>
<tr>
<td>icom_forum</td>
<td><a href="http://docs.oasis-open.org/ns/icom/forum/201008">http://docs.oasis-open.org/ns/icom/forum/201008</a></td>
</tr>
<tr>
<td>icom_conf</td>
<td><a href="http://docs.oasis-open.org/ns/icom/conference/201008">http://docs.oasis-open.org/ns/icom/conference/201008</a></td>
</tr>
</tbody>
</table>

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

3.1 Main Branch

3.1.1 Entity and Top-Level Subclasses

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

3.1.2 Identifiable

3.1.2.1 Description

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

3.1.2.2 Class Definition

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

The Identifiable class has attribute values:

```
localNamespace
  Value: icom_core

localName
  Value: Identifiable
```
extendsFrom

Value:

stereotype

Value: mixin

description

Value: Identifiable is a mixin class which defines the characteristics of all entities and some non-entities that enables unique identification.

propertyDefinitions

The values for this attribute are defined in Section 3.1.2.3.

3.1.2.3 Property Definitions

The Identifiable class MUST have the property definitions:

icom_core:objectId

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

icom_core: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 has attribute values:

- **localNamespace**
  - Value: icom_core

- **localName**
  - Value: Parental

- **extendsFrom**
  - Value: icom_core:Identifiable

- **stereotype**
  - Value: mixin

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

The values for this attribute are defined in Section 3.1.3.3.

### 3.1.3.3 Property Definitions

The Parental class inherits property definitions from super classes.

The Parental class MUST have the property definition:

- **icom_core:parent**
  - Description: Parent of an object.
  - Required: False
  - Inherited: False
  - Property Type: icom_core:Parental
  - Cardinality: Single
  - Updatability: Read Only

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

### 3.1.4 Extent

#### 3.1.4.1 Description

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

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

```plaintext
localNamespace
  Value: icom_core

localName
  Value: Extent

extendsFrom
  Value: icom_core:Parental

stereotype
  Value: mixin

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

3.1.4.3 Property Definitions

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

```plaintext
icom_core:parent
  Description: Parent of an extent.
  Required: False
  Inherited: True
  Property Type: icom_core:Extent
  Cardinality: Single
  Updatability: Read Only
```

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

3.1.5 Entity

3.1.5.1 Description

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

### 3.1.5.2 Class Definition

The Entity class has attribute values:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>localNamespace</td>
<td>icom_core</td>
</tr>
<tr>
<td>localName</td>
<td>Entity</td>
</tr>
<tr>
<td>extendsFrom</td>
<td>icom_core:Identifiable</td>
</tr>
<tr>
<td>stereotype</td>
<td>primary</td>
</tr>
<tr>
<td>isAbstract</td>
<td>TRUE</td>
</tr>
<tr>
<td>description</td>
<td>An entity is an object with an immutable id and individual access control.</td>
</tr>
</tbody>
</table>

### 3.1.5.3 Property Definitions

The Entity class inherits property definitions from super classes.

The Entity class MUST have the property definitions:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Required</th>
<th>Inherited</th>
<th>Property Type</th>
<th>Cardinality</th>
<th>Updatability</th>
</tr>
</thead>
<tbody>
<tr>
<td>icom_core:name</td>
<td>Name of an entity.</td>
<td>False</td>
<td>False</td>
<td>String</td>
<td>Single</td>
<td>Read Write</td>
</tr>
<tr>
<td>icom_core:createdBy</td>
<td>An actor who created an entity.</td>
<td>False</td>
<td>False</td>
<td>String</td>
<td>Single</td>
<td>Read Write</td>
</tr>
</tbody>
</table>
property_type: icom_core:Actor
cardinality: Single
updatability: Read Only

icom_core:creationDate
  description: Date and time when an entity is created. It is immutable.
  required: False
  inherited: False

icom_core:lastModifiedBy
  description: An actor who last modified an entity.
  required: False
  inherited: False

icom_core:lastModificationDate
  description: Date and time of last modification.
  required: False
  inherited: False

icom_core:parent
  description: A parental entity which contains an entity.
  required: False
  inherited: False

icom_ac:owner
  description: A subject who owns an entity.
  required: False
  inherited: False

property_type: icom_ac:Owner
cardinality: Single
Updatability: Read Write

**icom_ac:accessControlList**

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

**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

The Entity class MAY include additional property definitions which are implementation-defined.
3.1.6 Overview of Scope, Subject, and Artifact Branches

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

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

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

Figure 2: Entity Class Diagram.
3.2 Scope Branch

3.2.1 Scope and Top-Level Subclasses

Figure 3: Scope, Subject, and Artifact Branches.

Figure 4: Scope Branch.

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

3.2.2.1 Description
A scope is an extent of an administrative policy.

3.2.2.2 Class Definition
The Scope class has attribute values:

localNamespace
Value: icom_core

localName
Value: Scope

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

stereotype
Value: primary

isAbstract
Value: TRUE

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

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

3.2.2.3 Property Definitions
The Scope class inherits property definitions from super classes.
The Scope class MUST have the property definitions:

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

icom_core:parent
Description: A community which contains a scope.
<table>
<thead>
<tr>
<th>Required</th>
<th>Inherited</th>
<th>Property Type</th>
<th>Cardinality</th>
<th>Updatability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>True</td>
<td>icom_core:Community</td>
<td>Single</td>
<td>Read Only</td>
<td>Zero or more groups defined in a scope.</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>icom_core:Group</td>
<td>Multi</td>
<td>Read Only</td>
<td>Member groups of a scope, i.e. groups whose assigned scopes include this scope.</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>icom_core:Group</td>
<td>Multi</td>
<td>Read Only</td>
<td>Member groups of a scope, i.e. groups whose assigned scopes include this scope.</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>icom_ac:RoleDefinition</td>
<td>Multi</td>
<td>Read Only</td>
<td>Zero or more role definitions defined in a scope.</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>icom_ac:Role</td>
<td>Multi</td>
<td>Read Only</td>
<td>Zero or more roles defined in a scope.</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>icom_meta:relationship</td>
<td>Multi</td>
<td>Read Only</td>
<td>Zero or more relationships associated with a scope.</td>
</tr>
</tbody>
</table>
The Scope class MAY include additional property definitions which are implementation-defined.

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 has attribute values:

```plaintext
localNamespace
  Value: icom_core

localName
  Value: Community
```
extendsFrom
Value: icom_core:Scope

stereotype
Value: primary

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

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

3.2.3.3 Property Definitions
The Community class inherits property definitions from super classes.
The Community class MUST have the property definitions:

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

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

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

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

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

3.2.4 Space

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

3.2.4.2 Class Definition
The Space class has attribute values:

localNamespace
Value: icom_core

localName
Value: Space

extendsFrom
Value: icom_core:Scope, icom_core:FolderContainer
808    stereotype
809         Value: primary
810
812    description
813         Value: A space is a scope that defines a durable context and place for actors to work or
collaborate.
815
816    propertyDefinitions
817         The values for this attribute are defined in Section 3.2.4.3.

818    3.2.4.3 Property Definitions
819    The Space class inherits property definitions from super classes.
820    The Space class MUST have the property definition:
821
822    icom_core:element
823         Description: Elements of a space.
824         Required: False
825         Inherited: True
826         Property Type: icom_core:SpaceItem
827         Cardinality: Multi
828         Updatability: Read Only
829
830    The Space class MAY include additional property definitions which are implementation-defined.
831
832
833    Figure 7: Space Class Diagram.
3.3 Subject Branch

3.3.1 Subject and Top-Level Subclasses

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

3.3.2 Subject

3.3.2.1 Description

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

3.3.2.2 Class Definition

The Subject class has attribute values:

```
localNamespace
  Value: icom_core

localName
  Value: Subject

extendsFrom
  Value: icom_core:Entity, icom_meta:RelationshipBondable

 stereotype
  Value: primary

isAbstract
  Value: TRUE

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

<table>
<thead>
<tr>
<th>Description:</th>
<th>A description of a subject.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required:</td>
<td>False</td>
</tr>
<tr>
<td>Inherited:</td>
<td>False</td>
</tr>
<tr>
<td>Property Type:</td>
<td>icom_core:Scope</td>
</tr>
<tr>
<td>Cardinality:</td>
<td>Single</td>
</tr>
<tr>
<td>Updatability:</td>
<td>Read Write</td>
</tr>
</tbody>
</table>

### icom_core:parent

<table>
<thead>
<tr>
<th>Description:</th>
<th>A scope which contains a subject.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required:</td>
<td>False</td>
</tr>
<tr>
<td>Inherited:</td>
<td>True</td>
</tr>
<tr>
<td>Property Type:</td>
<td>icom_core:Scope</td>
</tr>
<tr>
<td>Cardinality:</td>
<td>Single</td>
</tr>
<tr>
<td>Updatability:</td>
<td>Read Only</td>
</tr>
</tbody>
</table>

### icom_meta:relationship

<table>
<thead>
<tr>
<th>Description:</th>
<th>Zero or more relationships associated with a subject.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required:</td>
<td>False</td>
</tr>
<tr>
<td>Inherited:</td>
<td>False</td>
</tr>
<tr>
<td>Property Type:</td>
<td>icom_meta:Relationship</td>
</tr>
<tr>
<td>Cardinality:</td>
<td>Multi</td>
</tr>
<tr>
<td>Updatability:</td>
<td>Read Only</td>
</tr>
</tbody>
</table>

### icom_meta:property

<table>
<thead>
<tr>
<th>Description:</th>
<th>Zero or more extended properties of a subject.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required:</td>
<td>False</td>
</tr>
<tr>
<td>Inherited:</td>
<td>False</td>
</tr>
<tr>
<td>Property Type:</td>
<td>icom_meta:Property</td>
</tr>
<tr>
<td>Cardinality:</td>
<td>Multi</td>
</tr>
<tr>
<td>Updatability:</td>
<td>Read Write</td>
</tr>
</tbody>
</table>

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

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

3.3.3.2 Class Definition
The Group class has attribute values:

```plaintext
localNamespace
Value: icom_core

localName
Value: Group

extendsFrom
Value: icom_core:Subject, icom_core:Addressable, icom_ac:Accessor
Optional Value: icom_ac:Owner

stereotype
Value: primary

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

propertyDefinitions
The values for this attribute are defined in Section 3.3.3.3.
```

Figure 9: Subject Class Diagram.
3.3.3.3 Property Definitions

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

icom_core:assignedGroup

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

icom_core:assignedScope

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

icom_core:memberGroup

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

icom_core:memberActor

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

icom_ac:assignedRole

- Description: A group's roles.
- Required: False
- Inherited: False
- Property Type: icom_ac:Role
- Cardinality: Multi
976     Updatability: Read Write
977
978     The Group class MAY include additional property definitions which are implementation-defined.
979
980
981     Figure 10: Group and Actor Class Diagram.
982
983     3.3.4 Actor
984
985     3.3.4.1 Description
986     An actor is a subject that can perform actions on objects.
987     It can be an owner of entities.
988
989     3.3.4.2 Class Definition
990     The Actor class has attribute values:
991
992     localNamespace
993     Value: icom_core
994
995     localName
996     Value: Actor
extendsFrom

Value: icom_core:Subject, icom_core: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_core:parent

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

icom_core:assignedGroup

Description: An actor’s groups.
Required: False
Inherited: False
Property Type: icom_core:Group
Cardinality: Multi
Updatability: Read Write

icom_core:assignedCommunity

Description: An actor’s communities.
Required: False
Inherited: False
Property Type: icom_core:Community
Cardinality: Multi
Updatability: Read Write
The Actor class MAY include additional property definitions which are implementation-defined.

### 3.3.5 Person

#### 3.3.5.1 Description

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

A person has a personal space.

#### 3.3.5.2 Class Definition

The Person class has attribute values:

- **localNamespace**
  - Value: icom_core

- **localName**
  - Value: Person

- **extendsFrom**
  - Value: icom_core:Actor

- **stereotype**
  - Value: primary

- **description**
  - Value: A person is an individual human who may be an actor.

- **propertyDefinitions**
  - The values for this attribute are defined in Section 3.3.5.3.

#### 3.3.5.3 Property Definitions

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

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

**icom_core:middleName**
- **Description**: Middle name of a person. Can include multiple names concatenated.
- **Required**: False
- **Inherited**: False
- **Property Type**: String
- **Cardinality**: Single
- **Updatability**: Read Write

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

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

**icom_core:suffix**
- **Description**: Suffix of a person’s name.
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Required</th>
<th>Inherited</th>
<th>Property Type</th>
<th>Cardinality</th>
<th>Updatability</th>
</tr>
</thead>
<tbody>
<tr>
<td>nickname</td>
<td>Nickname of a person.</td>
<td>False</td>
<td>False</td>
<td>String</td>
<td>Multi</td>
<td>Read Write</td>
</tr>
<tr>
<td>jobTitle</td>
<td>Job title of a person.</td>
<td>False</td>
<td>False</td>
<td>String</td>
<td>Single</td>
<td>Read Write</td>
</tr>
<tr>
<td>department</td>
<td>A person’s affiliated department.</td>
<td>False</td>
<td>False</td>
<td>String</td>
<td>Single</td>
<td>Read Write</td>
</tr>
<tr>
<td>officeLocation</td>
<td>Location of a person’s department.</td>
<td>False</td>
<td>False</td>
<td>String</td>
<td>Single</td>
<td>Read Write</td>
</tr>
<tr>
<td>company</td>
<td>A person’s affiliated company.</td>
<td>False</td>
<td>False</td>
<td>String</td>
<td>Single</td>
<td>Read Write</td>
</tr>
<tr>
<td>profession</td>
<td>A person’s profession.</td>
<td>False</td>
<td>False</td>
<td>String</td>
<td>Single</td>
<td>Read Write</td>
</tr>
</tbody>
</table>
Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Write

icom_core:personalSpace
Description: Personal space of a person.
Required: False
Inherited: False
Property Type: icom_core: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 Person class MAY include additional property definitions which are implementation-defined.
3.3.6 Resource

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

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

3.3.6.2 Class Definition
The Resource class has attribute values:

```plaintext
localNamespace
  Value: icom_core

localName
  Value: Resource

extendsFrom
  Value: icom_core:Actor

stereotype
  Value: primary

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

propertyDefinitions
  The values for this attribute are defined in Section 3.3.6.3.
```
3.3.6.3 Property Definitions

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

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

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

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

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

icom_core:bookingRule
- Description: Resource booking rule.
- Required: False
- Inherited: False
- Property Type: icom_core:ResourceBookingRule
- Cardinality: Single
The Resource class MAY include additional property definitions which are implementation-defined.

Figure 12: Resource Class Diagram.

### 3.3.7 ResourceType

#### 3.3.7.1 Description

A resource type is a category of resources.

#### 3.3.7.2 Class Definition

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

The ResourceType class has attribute values:

- **localNamespace**
  - Value: icom_core
localName
   Value: ResourceType
extendsFrom
   Value:
stereotype
   Value: mixin
description
   Value: ResourceType is a mixin class which defines a type of resources.
propertyDefinitions
   The values for this attribute are defined in Section 3.3.7.3.

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

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

localNamespace
   Value: icom_core

localName
   Value: ResourceTypeEnum
extendsFrom
   Value: icom_core:ResourceType
stereotype
   Value: primary
isEnumeration
   Value: TRUE
description
   Value: A type of resources.
ICOM defines four resource types:

- `icom_core:Room` a resource represents a room.
- `icom_core:Equipment` a resource represents an equipment.
- `icom_core:OnlineConference` a resource represents an online conference.
- `icom_core:OtherResourceType` a resource represents other things.

### 3.3.9 ResourceBookingRule

#### 3.3.9.1 Description

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

#### 3.3.9.2 Class Definition

The `ResourceBookingRule` class is a mixin class which defines a resource booking rule. The `ResourceBookingRule` class has attribute values:

- **localNamespace**
  - Value: `icom_core`

- **localName**
  - Value: `ResourceBookingRule`

- **extendsFrom**
  - Value: `ResourceBookingRule`

- **stereotype**
  - Value: `mixin`

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

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

#### 3.3.9.3 Property Definitions

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

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

The ResourceBookingRuleEnum class has attribute values:

```plaintext
localNamespace
  Value: icom_core

localName
  Value: ResourceBookingRuleEnum

extendsFrom
  Value: icom_core:ResourceBookingRule

stereotype
  Value: primary

isEnumeration
  Value: TRUE

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

instances
  Value: <icom_core:Open, icom_core:FirstComeFirstServed>
```

ICOM defines two resource booking rules:

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

3.4 Artifact Branch

3.4.1 Artifact and Top-Level Subclasses

Figure 13: Artifact Branch.

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

3.4.2 Item

3.4.2.1 Description

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

3.4.2.2 Class Definition

The Item class is a mixin class which defines the characteristics of entities that can be elements of a Container. The Item class has attribute values:
localNamespace
Value: icom_core

localName
Value: Item

extendsFrom
Value: icom_core:Identifiable

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

The Item class MAY have the optional property definition:

icom_core:container
Description: Zero, one, or more containers of an item, including the parent container.
Required: False
Inherited: False
Property Type: icom_core:Container
Cardinality: Multi
Updatability: Read Write

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

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

3.4.3.2 Class Definition
The Spaceltem class is a mixin class which defines the characteristics of items that can be elements of a Space.
The Spaceltem class has attribute values:

```plaintext
localNamespace
  Value: icom_core

localName
  Value: Spaceltem

extendsFrom
  Value: icom_core:Item

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

propertyDefinitions
  The values for this attribute are defined in Section 3.4.3.3.
```

3.4.3.3 Property Definitions
The Spaceltem class inherits property definitions from super classes.
The Spaceltem class MAY include additional property definitions which are implementation-defined.

3.4.4 Container

3.4.4.1 Description
A container is an extent that contains items.

3.4.4.2 Class Definition
The Container class is a mixin class which defines the characteristics of extents that contain items.
The Container class has attribute values:

- **localNamespace**
  - Value: icom_core

- **localName**
  - Value: Container

- **extendsFrom**
  - Value: icom_core: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 definition:

- **icom_core: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_core: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 has attribute values:

```plaintext
localNamespace
  Value: icom_core

localName
  Value: FolderContainer

eextendsFrom
  Value: icom_core:Container

stereotype
  Value: mixin

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

propertyDefinitions
  The values for this attribute are defined in Section 3.4.5.3.
```

### 3.4.5.3 Property Definitions

The FolderContainer class inherits property definitions from super classes.

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

### 3.4.6 Artifact

#### 3.4.6.1 Description

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

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

#### 3.4.6.2 Class Definition

The Artifact class has attribute values:

```plaintext
localNamespace
  Value: icom_core

localName
  Value: Artifact
```
extendsFrom
Value: icom_core:Entity, icom_core:Item, icom_meta:RelationshipBondable
Optional Value: icom_core: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_core:description
Description: A description of an artifact.
Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Write

icom_core:userCreationDate
Description: Date and time when an artifact was created.
Required: False
Inherited: False
Property Type: DateTime
Cardinality: Single
Updatability: Read Write

icom_core:userLastModificationDate
Description: Date and time when an artifact was last modified.
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:viewerProperty
Description: Zero or more extended properties of an artifact visible to a viewer.
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.
3.4.7 Folder

3.4.7.1 Description

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

3.4.7.2 Class Definition

The Folder class has attribute values:

```
localNamespace
  Value: icom_core

localName
  Value: Folder

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

stereotype
  Value: primary

isAbstract
  Value: TRUE
```
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 definition:

icom_core:parent
Description: A parent container of a folder.
Required: False
Inherited: True
Property Type: icom_core: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.
Note: It is typically used for document folders, inbox, outbox, and trash folder of a space.

3.4.8.2 Class Definition
The HeterogeneousFolder class has attribute values:

localNamespace
Value: icom_core

localName
Value: HeterogeneousFolder

extendsFrom
Value: icom_core:Folder, icom_core: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 definition:

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

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

Figure 15: Heterogeneous Folder Class Diagram.
3.5 Access Control Model

3.5.1 Accessor

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

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

The Accessor class has attribute values:

```
localNamespace
    Value: com_ac

localName
    Value: Accessor

extendsFrom
    Value: com_core:Identifiable

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 privileges in role assignments.
```

3.5.1.3 Property Definitions
The Accessor class inherits property definitions from super classes.
The Accessor class MAY include additional property definitions which are implementation-defined.

3.5.2 Owner

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

An owner of an entity MAY always have rights to update the access control list for the entity.
3.5.2.2 Class Definition
The Owner class is a mixin class which defines the characteristics of subjects such as groups and actors that can own entities.
The Owner class has attribute values:

```
localNamespace
    Value: icom_ac

localName
    Value: Owner

extendsFrom
    Value: icom_ac:Accessor

stereotype
    Value: mixin

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

3.5.2.3 Property Definitions
The Owner class inherits property definitions from super classes.
The Owner class MAY include additional property definitions which are implementation-defined.

3.5.3 RoleDefinition

3.5.3.1 Description
A role definition is a named set of privileges.

3.5.3.2 Class Definition
The RoleDefinition class has attribute values:

```
localNamespace
    Value: icom_ac

localName
    Value: RoleDefinition
```
extendsFrom
    Value: icom_core:Entity, icom_meta:RelationshipBondable

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 definition:

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

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 has attribute values:
localNamespace
Value: icom.ac

localName
Value: Role

extendsFrom
Value: icom.core:Subject

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.core:Scope
Cardinality: Single
Updatability: Read Write

icom.ac:memberAccessor
Description: Accessors (actors and groups) assigned to a role.
Required: False
Inherited: False
The Role class MAY include additional property definitions which are implementation-defined.

**Figure 16: Role Definition and Role Class Diagram.**

### 3.5.5 Privilege

#### 3.5.5.1 Description
A privilege is an access right granted through roles.

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

```
localNamespace
  Value: icom_ac

localName
  Value: Privilege
```
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 has attribute values:

localNamespace
Value: icom_ac

localName
Value: PrivilegeEnum

extendsFrom
Value: icom_ac:Privilege

stereotype
Value: primary

isEnumeration
Value: TRUE

description
Value: Privilege that can be assigned to a role.

instances
Value: <icom_ac:Archive, icom_ac:Audit>
ICOM defines two privileges:

- **icom_ac:Archive**: A right to archive contents in a scope.
- **icom_ac:Audit**: 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 has 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.

#### 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_core: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 has attribute values:

localNamespace
Value: icom_ac

localName
Value: AccessControlEntry

extendsFrom
Value:

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 types 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 has attribute values:

```
localNamespace
    Value: icom_ac

localName
    Value: AccessType

extendsFrom
    Value:
```
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 has attribute values:

```plaintext
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 has attribute values:

```plaintext
localNamespace
  Value: icom_ac

localName
  Value: AccessTypeEnum

extendsFrom
  Value: icom_ac:AccessType

stereotype
  Value: primary

isEnumeration
  Value: TRUE

description
  Value: Access type that can be granted or denied in an access control entry.

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

ICOM defines three access types:
  • icom_ac:Read a right to retrieve an entity.
```
• **icom_ac:Write** a right to update an entity.

• **icom_ac:Delete** a right to delete an entity.

![Access Control List Class Diagram](image)

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

### 3.6 Metadata Model

#### 3.6.1 ClassDefinition

**3.6.1.1 Description**

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

**3.6.1.2 Class Definition**

The ClassDefinition class has attribute values:

- **localNamespace**
  - Value: `icom_meta`

- **localName**
  - Value: `ClassDefinition`
extendsFrom
    Value: icom_core:Entity, icom_meta:RelationshipBondable

stereotype
    Value: primary

isAbstract
    Value: FALSE

description
    Value: A class definition defines a type of entities.

propertyDefinitions
    The values for this attribute are defined in Section 3.6.1.3.

3.6.1.3 Property Definitions
The ClassDefinition class inherits property definitions from super classes.
The ClassDefinition class MUST have the property definition:

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

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

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

- **Description**: Stereo type of a class.
- **Required**: True
- **Inherited**: False
- **Property Type**: icom_meta:StereoType
- **Cardinality**: Single
- **Updatability**: Read Write

**icom_meta:abstract**

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

**icom_meta:enumeration**

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

**icom_meta:instances**

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

**icom_meta:propertyDefinition**

- **Description**: One or more property definitions of a class definition.
- **Required**: False
- **Inherited**: False
- **Property Type**: icom_meta:PropertyDefinition
- **Cardinality**: Multi
- **Updatability**: Read Write
The ClassDefinition class MAY include additional property definitions which are implementation-defined.

3.6.2 StereoType

3.6.2.1 Description

A stereo type of a class definition.
3.6.2.2 Class Definition

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

The StereoType class has attribute values:

- **localNamespace**
  
  Value: icom_meta

- **localName**
  
  Value: StereoType

- **extendsFrom**
  
  Value:

- **stereotype**
  
  Value: mixin

- **description**
  
  Value: StereoType is a mixin class which defines a stereo type of a class definition.

- **propertyDefinitions**
  
  The values for this attribute are defined in Section 3.6.2.3.

3.6.2.3 Property Definitions

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

3.6.3 StereotypeEnum

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

The StereotypeEnum has attribute values:

- **localNamespace**
  
  Value: icom_meta

- **localName**
  
  Value: StereotypeEnum

- **extendsFrom**
  
  Value: icom_meta:StereoType

- **stereotype**
  
  Value: primary
ICOM defines two stereo types:

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

### 3.6.4 PropertyDefinition

#### 3.6.4.1 Description

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

#### 3.6.4.2 Class Definition

The PropertyDefinition class has attribute values:

- **localNamespace**
  - Value: icom_meta

- **localName**
  - Value: PropertyDefinition

- **extendsFrom**
  - Value: icom_core: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 3.6.4.3.

### 3.6.4.3 Property Definitions

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

**icom_core:namespace**
- **Description:** Namespace for a property name.
- **Required:** False
- **Inherited:** False
- **Property Type:** String
- **Cardinality:** Single
- **Updatability:** Read Write

**icom_core:name**
- **Description:** Name for a property.
- **Required:** True
- **Inherited:** False
- **Property Type:** String
- **Cardinality:** Single
- **Updatability:** Read Write

**icom_core: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:** False
- **Inherited:** False
- **Property Type:** icom_meta:PropertyType
- **Cardinality:** Single
- **Updatability:** On Create
- **Choices:** {PropertyChoiceType}
- **Open Choice:** False

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

**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:choice**
Description: An allowed value for a property.
Required: False
Inherited: False
Property Type: `icom_meta:PropertyChoiceType`
Cardinality: Multi
Updatability: Read Write

**icom_meta:openChoice**
Description: Indicates whether value of the property must be listed among the choices.
Required: False
Inherited: False
Property Type: Boolean
Cardinality: Single
Updatability: Read Write

**icom_meta:inherited**
Description: Indicates whether a property definition is inherited from a super class.
Required: False
Inherited: False
Property Type: Boolean
Cardinality: Single
Updatability: Read Write

**icom_meta:required**
Description: Indicates whether a property value must be provided. It is applicable only when the updatability of the property is read-write or on-create.
Required: True
Inherited: False
Property Type: Boolean
Cardinality: Single
Updatability: Read Write

**icom_meta:updatability**
Description: Updatability of a property specifying under what circumstances the property value can be updated.
Required: True
Inherited: False
Property Type: icom_meta:Updatability
Cardinality: Single
Updatability: On Create

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:Cardinality
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.

3.6.5 Property

3.6.5.1 Description
The property holds a property value.

3.6.5.2 Class Definition
The Property class has attribute values:

localNamespace
Value: icom_meta
localName
  Value: Property

extendsFrom
  Value: 

description
  Value: A property value.

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

3.6.5.3 Property Definitions
The Property class MUST have the property definitions:

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

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

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

3.6.6.1 Description

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

3.6.6.2 Class Definition

The PropertyChoiceType class has attribute values:

- `localNamespace`
  - Value: `icom_meta`

- `localName`
  - Value: `PropertyChoiceType`

- `extendsFrom`
  - Value:
stereotype
    Value: primary

description
    Value: A choice for a property value.

propertyDefinitions
    The values for this attribute are defined Section 3.6.6.3.

3.6.6.3 Property Definitions
The PropertyChoiceType class MUST have the property definitions:

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

icom_meta:displayName
    Description: Display name of a property choice.
    Required: True
    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

3.6.7 PropertyType
A PropertyType expresses a name of a property-type.
3.6.7.1 Class Definition

The PropertyType class is a mixin class which expresses a name of a property-type. The PropertyType class has attribute values:

```
localNamespace
  Value: icom_meta

localName
  Value: PropertyType

extendsFrom
  Value:

stereotype
  Value: mixin

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

propertyDefinitions
  The values for this attribute are defined in Section 3.6.7.2.
```

3.6.7.2 Property Definitions

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

3.6.8 PropertyTypeEnum

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

```
localNamespace
  Value: icom_meta

localName
  Value: PropertyTypeEnum

extendsFrom
  Value: icom_meta:PropertyType

stereotype
  Value: primary
```
isEnumeration
Value: TRUE

description
Value: 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:Duration, icom_meta:IRI, icom_meta:ID, icom_meta:HTML>

ICOM defines nine data types:

- `icom_meta:String` is equivalent to XML schema type `xsd:string`.
- `icom_meta:Boolean` is equivalent to XML schema type `xsd:boolean`.
- `icom_meta:Decimal` is equivalent to XML schema type `xsd:decimal`.
- `icom_meta:Integer` is equivalent to XML schema type `xsd:integer`.
- `icom_meta:Datetime` is equivalent to XML schema type `xsd:dateTime`.
- `icom_meta:Duration` is equivalent to XML schema type `xsd:duration`.
- `icom_meta:IRI` is equivalent to XML schema type `xsd:anyURI`.
- `icom_meta:ID` opaque object identifiers.
- `icom_meta:HTML` documents or fragments of Hypertext Markup Language (HTML) content


3.6.9 Updatability

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

3.6.9.2 Class Definition
The Updatability class is a mixin class which specifies under what circumstances a property value can be updated.

The Updatability class has attribute values:

```
localNamespace
Value: icom_meta

localName
Value: Updatability

extendsFrom
Value:
```
The Updatability class specifies under what circumstances a property value can be updated.

The values for this attribute are defined in Section 3.6.9.3.

3.6.9.3 Property Definitions

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

3.6.10 UpdatabilityEnum

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

The UpdatabilityEnum has attribute values:

- **localNamespace**
  - Value: icom_meta

- **localName**
  - Value: UpdatabilityEnum

- **extendsFrom**
  - Value: icom_meta:Updatability

- **stereotype**
  - Value: primary

- **isEnumeration**
  - Value: TRUE

- **description**
  - Value: Updatability of a property.

- **instances**
  - Value: `<icom_meta:ReadOnly, icom_meta:WriteOnly, icom_meta:ReadWrite, icom_meta:OnCreate>`

ICOM defines four updatability types:

- **icom_meta:ReadOnly** a property must not be updated directly by application.
- **icom_meta:WriteOnly** a property can be updated but cannot be read by application.
•  icom_meta:ReadWrite a property can be updated by application.
•  icom_meta:WriteOnly a property can be updated during the creation by application.

3.6.11 Cardinality

3.6.11.1 Description
Cardinality specifies whether a property is single or multi valued.

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

The Cardinality class has attribute values:

```plaintext
localNamespace
  Value: icom_meta

localName
  Value: Cardinality

extendsFrom
  Value: 

stereotype
  Value: mixin

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

propertyDefinitions
  The values for this attribute are defined in Section 3.6.11.3.
```

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

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

The CardinalityEnum has attribute values:

```plaintext
localNamespace
  Value: icom_meta
```
localName
Value: CardinalityEnum

extendsFrom
Value: icom_meta:Cardinality

stereotype
Value: primary

isEnumeration
Value: TRUE

description
Value: Cardinality of a property.

instances
Value: <icom_meta:Single, icom_meta:Multi>

ICOM defines two cardinality types:

- **icom_meta:Single** a property can have zero or one value (if property is not required), or exactly one value (if property is required).

- **icom_meta:Multi** a property can have zero or more values (if property is not required), or one or more values (if property is required).

3.6.13 Marker and Subclasses

Figure 20: Marker Branch.

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

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

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

3.6.14.2 Class Definition
The Marker class has attribute values:

```plaintext
localNamespace
   Value: icom_meta

localName
   Value: Marker

extendsFrom
   Value: icom_core:Artifact

stereotype
   Value: primary

isAbstract
   Value: TRUE

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

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

3.6.14.3 Property Definitions
The Marker class inherits property definitions from super classes.
The Marker class MUST have the property definition:

```plaintext
icom_meta:markedEntity
   Description: A marked entity.
   Required: False
   Inherited: False
   Property Type: icom_core:Entity
   Cardinality: Multi
   Updatability: Read Only
```
The Marker class MAY include additional property definitions which are implementation-defined.

**Figure 21**: Marker Class Diagram.

### 3.6.15 Category

#### 3.6.15.1 Description

A category is a marker that classifies entities.

#### 3.6.15.2 Class Definition

The Category class has attribute values:

- `%localNamespace`  
  Value:  `icom_meta`

- `%localName`  
  Value:  `Category`

- `%extendsFrom`  
  Value:  `icom_meta:Marker, icom_core:Container`

- `%stereotype`  
  Value:  `primary`

- `%description`  
  Value:  A category is a marker that classifies entities.

- `%propertyDefinitions`  
  The values for this attribute are defined in Section 3.6.15.3.

### 3.6.15.3 Property Definitions

The Category class inherits property definitions from super classes.

The Category class MUST have the property definitions:

- `%icom_meta:superCategory`  
  Description:  Zero or more super categories.
  Required:  False
The Category class MAY include additional property definitions which are implementation-defined.
3.6.16 CategoryApplication

3.6.16.1 Description

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

3.6.16.2 Class Definition

The CategoryApplication class has attribute values:

- **localNamespace**
  
  Value: icom_meta

- **localName**
  
  Value: CategoryApplication

- **extendsFrom**
  
  Value: icom_core:Identifiable

- **stereotype**
  
  Value: primary

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

- **propertyDefinitions**
  
  The values for this attribute are defined in Section 3.6.16.3.
3.6.16.3 Property Definitions

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

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

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

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

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

3.6.17 Tag

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

3.6.17.2 Class Definition
The Tag class has 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 3.6.17.3.

### 3.6.17.3 Property Definitions

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

**icom_meta:applicationCount**

Description: An estimate of the number of times a tag is applied on entities.

Required: False

Inherited: False

Property Type: Integer

Cardinality: Single

Updatability: Read Only

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

3.6.18.1 Description

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

3.6.18.2 Class Definition

The TagApplication class has attribute values:

localNamespace
Value: icom_meta

localName
Value: TagApplication

extendsFrom
Value: icom_core:Identifiable

stereotype
Value: primary

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

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

3.6.18.3 Property Definitions

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

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

### icom_meta:tag
- **Description:** A tag which is applied to 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 to an entity.
- **Required:** False
- **Inherited:** False
- **Property Type:** icom_core:Actor
- **Cardinality:** Single
- **Updatability:** Read Only

### icom_meta:applicationDate
- **Description:** A date and time when a tag is applied to 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.

### 3.6.19 RelationshipBondable

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

Note: A relationship can exist among entities that are not relationships.
3.6.19.2 Class Definition

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

The RelationshipBondable class has attribute values:

```
localNamespace
  Value: icom_meta

localName
  Value: RelationshipBondable

extendsFrom
  Value: icom_core:Identifiable

stereotype
  Value: mixin

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

propertyDefinitions
  The values for this attribute are defined in Section 3.6.19.3.
```

3.6.19.3 Property Definitions

The RelationshipBondable class inherits property definitions from super classes.

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

3.6.20 RelationshipDefinition

3.6.20.1 Description

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

3.6.20.2 Class Definition

The RelationshipDefinition class has attribute values:

```
localNamespace
  Value: icom_meta
```
localName
Value: RelationshipDefinition

extendsFrom
Value: icom_core:Entity, icom_meta:RelationshipBondable

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

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

3.6.20.3 Property Definitions
The RelationshipDefinition class inherits property definitions from super classes.
The RelationshipDefinition class MUST have the property definitions:

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

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.

3.6.21 Relationship

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

3.6.21.2 Class Definition
The Relationship class has attribute values:

```
localNamespace
  Value: icom_meta

localName
  Value: Relationship

extendsFrom
  Value: icom_core: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 3.6.21.3.
```

3.6.21.3 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.
3.7 Common Concepts

3.7.1 Addressable

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

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

The Addressable class has attribute values:

localNamespace
Value: icom_core

localName
Value: Addressable
extendsFrom
Value: icom_core:Identifiable

stereotype
Value: mixin

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

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

3.7.1.3 Property Definitions
The Addressable class inherits property definitions from super classes.
The Addressable class MUST have the property definitions:

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

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

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

3.7.2 EntityAddress

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

3.7.2.2 Class Definition
The EntityAddress class has attribute values:
localNamespace
  Value: icom_core

localName
  Value: EntityAddress

extendsFrom
  Value:

stereotype
  Value: primary

description
  Value: An entity address object represents an address which is defined by type and IRI.

propertyDefinitions
  The values for this attribute are defined in Section 3.7.2.3.

3.7.2.3 Property Definitions
The EntityAddress class MUST have the property definitions:

icom_core:addressType
  Description: Type of an address.
  Required: False
  Inherited: False
  Property Type: String
  Cardinality: Single
  Updatability: Read Write

icom_core:address
  Description: A IRI representing an address.
  Required: False
  Inherited: False
  Property Type: IRI
  Cardinality: Single
  Updatability: Read Write

3.7.3 Participant

3.7.3.1 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.

### 3.7.3.2 Class Definition

The Participant class has attribute values:

```plaintext
localNamespace
  Value:  icom_core

localName
  Value:  Participant

extendsFrom
  Value:

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 3.7.3.3.
```

### 3.7.3.3 Property Definitions

The Participant class inherits property definitions from super classes.

The Participant class MUST have the property definitions:

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

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

### 3.7.4 Priority

#### 3.7.4.1 Description

A priority level for delivery of information.

#### 3.7.4.2 Class Definition

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

The Priority class has attribute values:

- **localNamespace**
  - Value: icom_core

- **localName**
  - Value: Priority

- **extendsFrom**
  - Value: 

- **stereotype**
  - Value: mixin

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

- **propertyDefinitions**
  - The values for this attribute are defined in Section 3.7.4.3.

#### 3.7.4.3 Property Definitions

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

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

The PriorityEnum has attribute values:

- **localNamespace**
  - Value: icom_core

- **localName**
  - Value: PriorityEnum

- **extendsFrom**
  - Value: icom_core:Priority

- **stereotype**
  - Value: primary

- **isEnumeration**
  - Value: TRUE

- **description**
  - Value: Priority level for delivery of information.

- **instances**
  - Value: <icom_core:Normal, icom_core:Low, icom_core:Medium, icom_core:High>

ICOM defines four priorities:

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

3.7.6 DateTimeResolution

3.7.6.1 Description

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

3.7.6.2 Class Definition

The DateTimeResolution class is a mixin class which defines a resolution of date time value.
The DateTimeResolution class has attribute values:

```
localNamespace
   Value: icom_core

localName
   Value: DateTimeResolution

extendsFrom
   Value:

stereotype
   Value: mixin

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

propertyDefinitions
   The values for this attribute are defined in Section 3.7.6.3.
```

### 3.7.6.3 Property Definitions

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

### 3.7.7 DateTimeResolutionEnum

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

The DateTimeResolutionEnum has attribute values:

```
localNamespace
   Value: icom_core

localName
   Value: DateTimeResolutionEnum

extendsFrom
   Value: icom_core:DateTimeResolution

stereotype
   Value: primary
```
isEnumeration
Value: TRUE

description
Value: Resolution of a date time value.

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

ICOM defines three date time resolutions:

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

3.7.8 TimeZone

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

3.7.8.2 Class Definition
The TimeZone class has attribute values:

localNamespace
Value: icom_core

localName
Value: TimeZone

extendsFrom
Value: 

stereotype
Value: primary

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

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

3.7.8.3 Property Definitions
The TimeZone class inherits property definitions from super classes.
The TimeZone class MUST have the property definitions:

icom_core:ID
  Description: Identifier of a time zone.
  Required: False
  Inherited: False
  Property Type: String
  Cardinality: Single
  Updatability: On Create

icom_core:rawOffset
  Description: An offset to add to Universal Coordinated Time (UTC) to get local time. If Daylight Saving Time is in effect at the specified date, the offset value is adjusted with the amount of daylight saving.
  Required: False
  Inherited: False
  Property Type: Integer
  Cardinality: Single
  Updatability: On Create

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

3.7.9 Location

3.7.9.1 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.

3.7.9.2 Class Definition

The Location class has attribute values:

localNamespace
  Value: icom_core

localName
  Value: Location

extendsFrom
  Value:


3.7.9.3 Property Definitions

The Location class MUST have the property definitions:

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

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

**icom_core:timeZone**
- **Description:** Time zone of a location.
- **Required:** False
- **Inherited:** False
- **Property Type:** icom_core:TimeZone
- **Cardinality:** Single
- **Updatability:** Read Write

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

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

3.7.10 GeoCoordinates

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

3.7.10.2 Class Definition
The GeoCoordinates class has attribute values:

localNamespace
Value: icom_core

localName
Value: GeoCoordinates

extendsFrom
Value:

type
Value: primary

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

3.7.10.3 Property Definitions
The GeoCoordinates class MUST have the property definitions:

icom_core:latitude
Description: Latitude of a location.
Required: False
Inherited: False
Property Type: Float
Cardinality: Single
Updatability: Read Write
icom_core:longitude
Description: Longitude of a location.
Required: False
Inherited: False
Property Type: Float
Cardinality: Single
Updatability: Read Write

icom_core:altitude
Description: Altitude of a location.
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 Overview of Extension Modules

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

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

Figure 25: Containers of Collaboration Activities.

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

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

- a library of documents and wiki pages to support content sharing and co-creation,
• an inbox or outbox for communication, or
• a trash folder to archive all types of artifacts deleted from a space.

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

**Calendar** is a specialized container to support time management.

**TaskList** is a specialized container to support task coordination.

**Forum** is a specialized container to support

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

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

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

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

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

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

4. **Presence Module** (in Section 4.5) defines Presence, Activity, and Contact Method. Presence represents a watchable state of a presentity (which is usually a person). Presence state is derived using an actor’s subscriptions.

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

5. **Address Book Module** (in Section 4.6) defines AddressBook and PersonContact. A person contact can bookmark a reference to a person in an ICOM community as well as store addresses, phone numbers, and other entries about a person who may not be in any ICOM community.

6. **Calendar Module** (in Section 4.7) defines Calendar, Occurrence, and OccurrenceSeries. Occurrence artifacts are used to resolve the free-busy times of participants for scheduling of meetings and booking of rooms and other resources.

7. **Free Busy Module** (in Section 4.8) defines FreeBusy. FreeBusy is a view derived from occurrences in a calendar or a set of calendars using an actor’s privileges to determine the free or busy states of calendar occurrences.

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

8. **Task List Module** (in Section 4.9) defines TaskList and Task. Tasks are used to coordinate the assignment of tasks and to track the progress of task activities.

9. **Forum Module** (in Section 4.10) defines Forum, Topic, Announcement, and DiscussionMessage. Topics, announcements, and discussions are used for threaded discussions. Moderators of a forum can prune, merge, or fork the discussion threads.

10. **Conference Module** (in Section 4.11) defines Conference and related classes. A conference can contain visual, audio, and chat transcripts of the conference sessions. It also contains the current status, conference settings, past sessions, active session, and activity logs.
4.2 Content Module

4.2.1 MimeConvertible

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

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

The MimeConvertible class has attribute values:

```plaintext
localNamespace
  Value:  icom_content

localName
  Value:  MimeConvertible

extendsFrom
  Value:  icom_core:Identifiable

stereotype
  Value:  mixin

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

propertyDefinitions
  The values for this attribute are defined in Section 4.2.1.3.
```

4.2.1.3 Property Definitions
The MimeConvertible class inherits property definitions from super classes.

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

4.2.2 Content

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

The Content class has attribute values:

```plaintext
localNamespace
  Value: icom_content

localName
  Value: Content

declaredExtendsFrom
  Value: icom_core:Identifiable, icom_content:MimeConvertible

declaredStereotype
  Value: primary

isAbstract
  Value: TRUE

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

propertyDefinitions
  The values for this attribute are defined in Section 4.2.2.3.
```

4.2.2.3 Property Definitions

The Content class inherits property definitions from super classes.

The Content class MUST have the property definitions:

```plaintext
icom_content:contentId
  Description: A content id is a unique identifier for a part of content in multi-part contents.
  Required: False
  Inherited: False
  Property Type: String
  Cardinality: Single
  Updatability: Read Write

icom_content:mediaType
  Description: Media type is a two-part identifier for Internet file formats.
  Required: False
  Inherited: False
  Property Type: String
  Cardinality: Single
```
The Content class MAY include additional property definitions which are implementation-defined.

**4.2.3 MultiContent**

**4.2.3.1 Description**

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

The MultiContent class has attribute values:

```
localNamespace
  Value: icom_content
```

```
localName
  Value: MultiContent
```

```
extendsFrom
  Value: icom_content:Content
```

```
stereotype
  Value: primary
```

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

```
propertyDefinitions
  The values for this attribute are defined in Section 4.2.3.3.
```

4.2.3.3 Property Definitions

The MultiContent class inherits property definitions from super classes.

The MultiContent class MUST have the property definitions:

```
icom_content:part
```

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

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

4.2.4 SimpleContent

4.2.4.1 Description

A simple content holds a single piece of data.
4.2.4.2 Class Definition

The SimpleContent class has attribute values:

\[\text{localNamespace}\]
Value: icom_content

\[\text{localName}\]
Value: SimpleContent

\[\text{extendsFrom}\]
Value: icom_content:Content

\[\text{stereotype}\]
Value: primary

\[\text{description}\]
Value: A simple content holds a single piece of data.

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

4.2.4.3 Property Definitions

The SimpleContent class inherits property definitions from super classes.

The SimpleContent class MUST have the property definitions:

\[\text{icom_content:characterEncoding}\]
Description: Character encoding specifies character set of a content (a missing value means that a piece of content should be treated as binary or raw).
Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Write

\[\text{icom_content:contentEncoding}\]
Description: Content encoding specifies encoding of a piece of content.
Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Write
Content language specifies language for a piece of content (a missing value means non-natural language content).

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

Description: Length of a piece of 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.

4.2.5 OnlineContent

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

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

4.2.5.2 Class Definition
The OnlineContent class has attribute values:

localNamespace
Value: icom_content

localName
Value: OnlineContent
extendsFrom
   Value: icom_content:Content

class stereotype
   Value: primary

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

propertyDefinitions
   The values for this attribute are defined in Section 4.2.5.3.

4.2.5.3 Property Definitions
The OnlineContent class inherits property definitions from super classes.
The OnlineContent class MUST have the property definition:

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

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

4.2.6 ContentDispositionType

4.2.6.1 Description
A content disposition type is a presentation style of content.

4.2.6.2 Class Definition
The ContentDispositionType class is a mixin class which defines a presentation style of content.
The ContentDispositionType class has attribute values:

localNamespace
   Value: icom_content

localName
   Value: ContentDispositionType
extendsFrom
  Value:

stereotype
  Value: mixin

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

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

4.2.6.3 Property Definitions

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

4.2.7 ContentDispositionTypeEnum

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

The ContentDispositionTypeEnum class has attribute values:

  localNamespace
    Value: icom_content

  localName
    Value: ContentDispositionType

  extendsFrom
    Value: icom_content:ContentDispositionType

  stereotype
    Value: primary

  isEnumeration
    Value: TRUE

  description
    Value: A presentation style of content.

  instances
    Value: <icom_content:Inline, icom_content:Attachment>
ICOM defines two content disposition types:

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

### 4.2.8 AttachedItem

#### 4.2.8.1 Description

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

#### 4.2.8.2 Class Definition

The AttachedItem class has attribute values:

```plaintext
localNamespace
  Value:  icom_content

localName
  Value:  AttachedItem

extendsFrom
  Value:  

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

propertyDefinitions
  The values for this attribute are defined in Section 4.2.8.3.
```

#### 4.2.8.3 Property Definitions

The AttachedItem class MUST have the property definitions:

```plaintext
icom\_core:name
  Description: Name of a content attachment.
  Required: True
  Inherited: False
  Property Type: String
  Cardinality: Single
  Updatability: Read Write
```
icom_content:content

Description: A content attached to an occurrence, task, or contact artifact.
Required: True
Inherited: False
Property Type: icom_content:Content
Cardinality: Single
Updatability: Read Write

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

4.3 Document Module

4.3.1 Versionable

4.3.1.1 Description

A versionable artifact is
1. a non-version-controlled copy,
2. a specific versioned copy,
3. a private working copy, or
4. a representative copy (optional)
of an artifact version series.

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

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

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

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

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

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

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

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

- a representative copy MUST have its own object identifier that is different from the object identifier of any versioned copy or private working copy;
assignment of an object identifier to a representative copy is implementation-dependent:
  o a representative copy MAY retain the object identifier of a non-version-controlled copy; if
    so the version type of a versionable artifact MUST change from
    NonVersionControlledCopy to RepresentativeCopy;
  o a representative copy MAY be assigned a new object identifier that is different from the
    object identifier of a non-version-controlled copy;

content and state of a representative copy is implementation-dependent:
  o a representative copy MAY be a copy of the content and state of the latest versioned
    copy or the latest major versioned copy in a version series;
  o a representative copy MAY be a copy of the content and state of a private working copy if
    the current user loading the representative copy is the same user who checks out a
    version series.

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

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

A private working copy MAY be saved in a 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.

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

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

Note: 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].

4.3.1.2 Class Definition

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

```xml
localNamespace
    Value: icom_doc

localName
    Value: Versionable

extendsFrom
    Value: icom_core: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 4.3.1.3.

4.3.1.3 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.

4.3.2 VersionControlMetadata

4.3.2.1 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:NonVersionControlledCopy then metadata is optional; if metadata is present, it MUST be a version series object.
- If the version type is icom_doc:RepresentativeCopy, 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.
4.3.2.2 Class Definition

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

The VersionControlMetadata class has attribute values:

```plaintext
localNamespace
  Value: icom_doc

localName
  Value: VersionControlMetadata

extendsFrom
  Value: icom_core:Identifiable

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

propertyDefinitions
  The values for this attribute are defined in Section 4.3.2.3.
```

4.3.2.3 Property Definitions

The VersionControlMetadata class inherits property definitions from super classes.

The VersionControlMetadata class MUST have the property definition:

```plaintext
icom_doc:representativeCopy
  Description: A representative copy of a versionable artifact.
  Required: False
  Inherited: False
  Property Type: icom_doc:Versionable
  Cardinality: Single
  Updatability: Read Only
```

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

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

4.3.3.2 Class Definition
The VersionSeries class has attribute values:

```plaintext
localNamespace
   Value: icom_doc

localName
   Value: VersionSeries

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

stereotype
   Value: primary

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

propertyDefinitions
   The values for this attribute are defined in Section 4.3.3.3.
```

4.3.3.3 Property Definitions
The VersionSeries class inherits property definitions from super classes.
The VersionSeries class MUST have the property definitions:

```plaintext
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.
```
**icom_doc:latestVersionedCopy**

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

**icom_doc:versionSeriesCheckedOutOn**

- **Description:** The time when a version series is checked out.
- **Required:** False
- **Inherited:** False
4231 Property Type: DateTime
4232 Cardinality: Single
4233 Updatability: Read Only
4234
4235 icom_doc:versionSeriesCheckoutComment
4236 Description: A check out comment of a version series.
4237 Required: False
4238 Inherited: False
4239 Property Type: String
4240 Cardinality: Single
4241 Updatability: Read Only
4242
4243 icom_doc:totalSize
4244 Description: Total size of all versioned copies of a versionable artifact in a version series.
4245 Required: False
4246 Inherited: False
4247 Property Type: Integer
4248 Cardinality: Single
4249 Updatability: Read Only
4250
4251 The VersionSeries class MAY include additional property definitions which are implementation-defined.
4252
4253
4254 4.3.4 Version
4255
4256 4.3.4.1 Description
4257 A version is a version control metadata that contains a version number, label, and description.
4258 A version object is a version control metadata of a versioned copy or a private working copy of a versionable artifact.
4259
4260 4.3.4.2 Class Definition
4261 The Version class has attribute values:
4262
4263 localNamespace
4264 Value: icom_doc
4265
4266 localName
4267 Value: Version
4268
4269 extendsFrom
4270 Value: icom_core:Entity, icom_doc:VersionControlMetadata, icom_meta:RelationshipBondable
### stereotype
Value: primary

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

### propertyDefinitions
The values for this attribute are defined in Section 4.3.4.3.

## 4.3.4.3 Property Definitions

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

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Required</th>
<th>Inherited</th>
<th>Property Type</th>
<th>Cardinality</th>
<th>Updatability</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>icom_doc:checkinComment</code></td>
<td>A check in comment of a versioned copy.</td>
<td>False</td>
<td>False</td>
<td>String</td>
<td>Single</td>
<td>Read Write</td>
</tr>
<tr>
<td><code>icom_doc:versionNumber</code></td>
<td>A version number of a versioned copy.</td>
<td>True</td>
<td>False</td>
<td>Integer</td>
<td>Single</td>
<td>Read Write</td>
</tr>
<tr>
<td><code>icom_doc:versionLabel</code></td>
<td>A version label of a versioned copy.</td>
<td>True</td>
<td>False</td>
<td>String</td>
<td>Single</td>
<td>Read Write</td>
</tr>
<tr>
<td><code>icom_doc:minorVersion</code></td>
<td>Indicates whether a versioned copy is a major version.</td>
<td>True</td>
<td>False</td>
<td>Boolean</td>
<td>Single</td>
<td>Read Write</td>
</tr>
</tbody>
</table>

icom-ics-v1.0-csd05 Standards Track Work Product Copyright © OASIS Open 2012. All Rights Reserved.
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
Cardinality: Single
Updatability: Read Only

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

4.3.5 VersionType

4.3.5.1 Description
A version type is a version state of a copy of versionable document.

4.3.5.2 Class Definition
The VersionType class is a mixin class which defines a version state of a copy of versionable document.
The VersionType class has attribute values:

localNamespace
Value: icom_doc

localName
Value: VersionType

extendsFrom
Value:

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

propertyDefinitions
The values for this attribute are defined in Section 4.3.5.3.
4.3.5.3 Property Definitions

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

4.3.6 VersionTypeEnum

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

The VersionTypeEnum class has attribute values:

```plaintext
localNamespace
  Value: icom_doc

localName
  Value: VersionTypeEnum

extendsFrom
  Value: icom_doc:VersionType

stereotype
  Value: primary

isEnumeration
  Value: TRUE

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

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

ICOM defines four version types:

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

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

4.3.7.2 Class Definition
The Document class has attribute values:

localNamespace
Value: icom_doc

localName
Value: Document

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

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

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

4.3.7.3 Property Definitions
The Document class inherits property definitions from super classes.
The Document class MUST have the property definitions:

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

icom_doc:size
Description: The size of a copy of a document.
Required: False
Inherited: False
Property Type: Integer
Cardinality: Single
Updatability: Read Only

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

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

4.3.8 WikiPage

4.3.8.1 Description
A wiki page is a document that contains a wiki content and that provides an html page generated from the wiki content.
4.3.8.2 Class Definition

The WikiPage class has attribute values:

localNamespace
Value: icom_doc

localName
Value: WikiPage

extendsFrom
Value: icom_doc:Document

stereotype
Value: primary

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

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

4.3.8.3 Property Definitions

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

icom_doc:renderedPage
Description: An html page generated from a wiki content.
Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Only

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

4.4.1 Message

4.4.1.1 Description

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

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

4.4.1.2 Class Definition

The Message class has attribute values:

- **localNamespace**
  - Value: icom_msg

- **localName**
  - Value: Message

- **extendsFrom**
  - Value: icom_core:Artifact

- **stereotype**
  - Value: primary

- **isAbstract**
  - Value: TRUE
Value: A message is a unit of conversation.

The values for this attribute are defined in Section 4.4.1.3.

### 4.4.1.3 Property Definitions

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

#### icom_content:content

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

#### icom_msg:sender

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

#### icom_msg:deliveredTime

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

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

### 4.4.2 UnifiedMessage

#### 4.4.2.1 Description

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

### 4.4.2.2 Class Definition

The UnifiedMessage class has attribute values:

- `localNamespace`
  - Value: `icom_msg`

- `localName`
  - Value: `UnifiedMessage`

- `extendsFrom`
  - Value: `icom_msg:Message, icom_content:MimeConvertible`

- `stereotype`
  - Value: `primary`

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

- `propertyDefinitions`
  - The values for this attribute are defined in Section 4.4.2.3.

### 4.4.2.3 Property Definitions

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

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

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

**icom_msg:bccReceivers**
Description: A list of participants to receive blind-carbon-copies of a message.
Required: False
Inherited: False
Property Type: icom_core:Participant
Cardinality: Multi
Updatability: Read Write

**icom_msg:replyTo**
Description: A list of participants to receive a reply message.
Required: False
Inherited: False
Property Type: icom_core:Participant
Cardinality: Multi
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 requested for a message.
Required: False
Inherited: False
Property Type: icom_msg:UnifiedMessageFlag
Cardinality: Multi
Updatability: Read Write

**icom_msg:messageDeliveryStatusNotificationRequest**
Description: Indicates the types of delivery status notifications requested for a message. Default is icom_msg:Failure.
Required: False
Inherited: False
Property Type: icom_msg: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:editMode**

**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. Each header is represented by a multi-valued property.

**Required:** False

**Inherited:** False

**Property Type:** icom_meta:Property

**Cardinality:** Multi

**Updatability:** Read Write

**icom_msg:size**

**Description:** The size of a unified message.

**Required:** False

**Inherited:** False

**Property Type:** Integer

**Cardinality:** Single

**Updatability:** Read Only

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

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

4.4.3.2 Class Definition
The UnifiedMessageParticipant class has attribute values:

- **localNamespace**
  - Value: icom_msg

- **localName**
  - Value: UnifiedMessageParticipant

- **extendsFrom**
  - Value: icom_core:Participant

- **stereotype**
  - Value: primary

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

- **propertyDefinitions**
  - The values for this attribute are defined in Section 4.4.3.3.

4.4.3.3 Property Definitions
The UnifiedMessageParticipant class inherits property definitions from super classes.

The UnifiedMessageParticipant class MUST have the property definitions:

- **icom_msg:fullAddress**
  - Description: Full address of a participant.
  - Required: False
  - Inherited: False
  - Property Type: IRI
  - Cardinality: Single
  - Updatability: Read Write

- **icom_msg:localPart**
  - Description: Local part of a full address.
  - Required: False
The UnifiedMessageParticipant class MAY include additional property definitions which are implementation-defined.

### 4.4.4 UnifiedMessageFlag

#### 4.4.4.1 Description

A unified message flag is a flag on a message.

#### 4.4.4.2 Class Definition

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

```
localNamespace
  Value: icom_msg

localName
  Value: UnifiedMessageFlag

extendsFrom
  Value: 

stereotype
  Value: mixin

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

propertyDefinitions
  The values for this attribute are defined in Section 4.4.4.3.
```
4.4.4.3 Property Definitions
The UnifiedMessageFlag class MAY include additional property definitions which are implementation-defined.

4.4.5 UnifiedMessageFlagEnum
The UnifiedMessageFlagEnum class is an enum class that enumerates the instances each of which expresses a flag on a message.
The UnifiedMessageFlagEnum class has attribute values:

```plaintext
type localNamespace
  Value: icom_msg

type localName
  Value: UnifiedMessageFlagEnum

type extendsFrom
  Value: icom_msg:UnifiedMessageFlag

type stereotype
  Value: primary

type isEnumeration
  Value: TRUE

type description
  Value: A flag on a message.

instances
```

ICOM defines eight flags:
- `icom_msg:Answered` a message is answered.
- `icom_msg:Forwarded` a message is forwarded.
- `icom_msg:Redirected` a message is redirected.
- `icom_msg:Hidden` a message is hidden.
- `icom_msg:MarkedForDelete` a message is marked for delete.
- `icom_msg:MarkedForFollowUp` a message is marked for follow up.
- `icom_msg:MarkedForDraft` a message is marked for draft.
- `icom_msg:MessageDispositionNotificationProcessed` a message disposition notification is processed.
4.4.6 UnifiedMessageDeliveryStatusNotificationRequest

4.4.6.1 Description

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

4.4.6.2 Class Definition

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

The UnifiedMessageDeliveryStatusNotificationRequest class has attribute values:

```plaintext
localNamespace
  Value: icom_msg

localName
  Value: UnifiedMessageDeliveryStatusNotificationRequest

extendsFrom
  Value: 

stereotype
  Value: mixin

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

propertyDefinitions
  The values for this attribute are defined in Section 4.4.6.3.
```

4.4.6.3 Property Definitions

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

4.4.7 UnifiedMessageDeliveryStatusNotificationRequestEnum

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

The UnifiedMessageDeliveryStatusNotificationRequestEnum class has attribute values:

```plaintext
localNamespace
  Value: icom_msg
```
ICOM defines four delivery status notification requests:

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

### 4.4.8 UnifiedMessageChannel

#### 4.4.8.1 Description

A message channel used to deliver a unified message.

#### 4.4.8.2 Class Definition

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

The UnifiedMessageChannel class has attribute values:

- **localName**
  
  Value: UnifiedMessageDeliveryStatusNotificationEnum

- **extendsFrom**
  
  Value: icom_msg:UnifiedMessageDeliveryStatusNotificationRequest

- **stereotype**
  
  Value: primary

- **isEnumeration**
  
  Value: TRUE

- **description**
  
  Value: 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>
extendsFrom
Value:

stereotype
Value: mixin

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

propertyDefinitions
The values for this attribute are defined in Section 4.4.8.3.

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

4.4.9 UnifiedMessageChannelEnum
The UnifiedMessageChannelEnum class is an enum class that enumerates the instances each of which expresses a type of delivery channel.
The UnifiedMessageChannelEnum class has attribute values:

localNamespace
Value: icom_msg

localName
Value: UnifiedMessageChannelEnum

extendsFrom
Value: icom_msg:UnifiedMessageChannel

stereotype
Value: primary

isEnumeration
Value: TRUE

description
Value: A delivery channel.

instances
Value: <icom_msg:Email, icom_msg:Voice, icom_msg:Fax, icom_msg:Notification>
ICOM defines four channel types:

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

### 4.4.10 UnifiedMessageEditMode

#### 4.4.10.1 Description

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

#### 4.4.10.2 Class Definition

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

The UnifiedMessageEditMode class has attribute values:

- **localNamespace**
  - Value: icom_msg

- **localName**
  - Value: UnifiedMessageEditMode

- **extendsFrom**
  - Value:

- **stereotype**
  - Value: mixin

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

- **propertyDefinitions**
  - The values for this attribute are defined in Section 4.4.10.3.

### 4.4.11 UnifiedMessageEditModeEnum

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

```plaintext
localNamespace
    Value: icom_msg

localName
    Value: UnifiedMessageEditModeEnum

extendsFrom
    Value: icom_msg:UnifiedMessageEditMode

stereotype
    Value: primary

isEnumeration
    Value: TRUE

description
    Value: A message is a new copy, a saved draft copy, or a delivered copy. New or draft copies are usually editable while delivered copies are usually not editable.

instances
    Value: <icom_msg:NewCopy, icom_msg:DraftCopy, icom_msg:DeliveredCopy>
```

ICOM defines three modes:

- **icom_msg:NewCopy** a message is a new message.
- **icom_msg:DraftCopy** a message is saved as a draft.
- **icom_msg:DeliveredCopy** a message is a sent or received message.
4.4.12 InstantMessage

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

4.4.12.2 Class Definition
The InstantMessage class has attribute values:

```plaintext
localNamespace
    Value: icom_msg

localName
    Value: InstantMessage

extendsFrom
    Value: icom_msg:Message
```
stereotype
Value: primary

isAbstract
Value: FALSE

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

propertyDefinitions
The values for this attribute are defined in Section 4.4.12.3.

4.4.12.3 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 receive a message.
Required: False
Inherited: False
Property Type: icom_core: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:formattingStyle
Description: A style for formatting a rich text message.
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
4.4.13 InstantMessageType

4.4.13.1 Description

An instant message type.

4.4.13.2 Class Definition

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

The InstantMessageType class has attribute values:

```java
localNamespace
    Value: icon_msg

localName
    Value: InstantMessageType

extendsFrom
    Value:
```

Figure 30: Instant Message Class Diagram.
```text
5127 stereotype
5128    Value: mixin
5130
description
5131    Value: InstantMessageType is a mixin class which defines a type of instant message.
5133
propertyDefinitions
5134    The values for this attribute are defined in Section 4.4.13.3.
5135
4.4.13.3 Property Definitions
5136    The InstantMessageType class MAY include additional property definitions which are implementation-
5137    defined.
5138
4.4.14 InstantMessageTypeEnum
5139    The InstantMessageTypeEnum class is an enum class that enumerates the instances each of which
5140    expresses a type of instant message.
5141    The InstantMessageTypeEnum class has attribute values:
5142
      localNamespace
5143        Value: icom_msg
5144
      localName
5145        Value: InstantMessageTypeEnum
5146
      extendsFrom
5147        Value: icom_msg:InstantMessageType
5148
      stereotype
5149        Value: primary
5150
      isEnumeration
5151        Value: TRUE
5152
      description
5153        Value: A type of instant message.
5154
      instances
5155        Value: <icom_msg:System, icom_msg:Chat, icom_msg:Broadcast, icom_msg:Gone,
5156             icom_msg:OtherInstantMessageType>
5157
ICOM defines five instant message types:
5158    • icom_msg:System an instant message is a system message.
```
• **icom_msg:Chat** an instant message is a chat message.
• **icom_msg:Broadcast** an instant message is a broadcast message.
• **icom_msg:Gone** an instant message is a message indicating that a user is gone.
• **icom_msg:OtherInstantMessageType** an instant message is of other type.

### 4.4.15 InstantMessageChatStatus

#### 4.4.15.1 Description
An instant message chat status defines a vocabulary of chat status.

#### 4.4.15.2 Class Definition
The InstantMessageChatStatus class is a mixin class which defines a chat status.

The InstantMessageChatStatus class has attribute values:

```plaintext
localNamespace
  Value: icom_msg

localName
  Value: InstantMessageChatStatus

extendsFrom
  Value:

stereotype
  Value: mixin

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

propertyDefinitions
  The values for this attribute are defined in Section 4.4.15.3.
```

#### 4.4.15.3 Property Definitions
The InstantMessageChatStatus class MAY include additional property definitions which are implementation-defined.

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

The InstantMessageChatStatusEnum class has attribute values:
ICOM defines five chat status:

- **icom_msg:Active** a user is active.
- **icom_msg:Typing** a user is typing.
- **icom_msg:Paused** a user has paused.
- **icom_msg:Inactive** a user is inactive.
- **icom_msg:Gone** a user is gone.

### 4.4.17 InstantMessageFeed

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

#### 4.4.17.2 Class Definition
The InstantMessageFeed class has attribute values:

```
<table>
<thead>
<tr>
<th>localNamespace</th>
<th>Value: icom_msg</th>
</tr>
</thead>
</table>

localName
Value: InstantMessageChatStatusEnum

extendsFrom
Value: icom_msg:InstantMessageChatStatus

stereotype
Value: primary

isEnumeration
Value: TRUE

description
Value: A chat status of a user.

instances
Value: <icom_msg:Active, icom_msg:Typing, icom_msg:Paused, icom_msg:Inactive, icom_msg:Gone>
```

...
extendsFrom

      Value: icom_core:Entity

stereotype

      Value: primary

description

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

propertyDefinitions

      The values for this attribute are defined in Section 4.4.17.3.

4.4.17.3 Property Definitions

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

icom_msg:connection

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

icom_msg:outboundInstantMessage

      Description: Outbound instant messages.
      Required: False
      Inherited: False
      Property Type: icom_msg:InstantMessage
      Cardinality: Multi
      Updatability: Write Only
4.4.18 InstantMessageConnection

4.4.18.1 Description

An instant message connection contains queues for inbound instant messages. A presentity can update the contact status, contact priority, and note for a contact method associated with a connection.
4.4.18.2 Class Definition

The InstantMessageConnection class has attribute values:

localNamespace
   Value: icom_msg

localName
   Value: InstantMessageConnection

extendsFrom
   Value: icom_core:Entity

stereotype
   Value: primary

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

propertyDefinitions
   The values for this attribute are defined in Section 4.4.18.3.

4.4.18.3 Property Definitions

The InstantMessageConnection class inherits property definitions from super classes.

The InstantMessageConnection class MUST have the property definitions:

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

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

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

icom_msg:inboundInstantMessage

- **Description:** Inbound instant messages.
- **Required:** False
- **Inherited:** False
- **Property Type:** icom_msg:InstantMessage
- **Cardinality:** Multi
- **Updatability:** Read Only

icom_presence:contactStatus

- **Description:** Reachability status to be propagated to an associated contact method in presence.
- **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 in presence.
- **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 in presence.
- **Required:** False
- **Inherited:** False
- **Property Type:** String
- **Cardinality:** Single
- **Updatability:** Write Only
4.5 Presence Module

4.5.1 Presence

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

4.5.1.2 Class Definition
The Presence class has attribute values:

```plaintext
localNamespace
Value: icom_presence
```

```plaintext
localName
Value: Presence
```

```plaintext
extendsFrom
Value: icom_core:Identifiable
```

```plaintext
stereotype
Value: primary
```

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

```plaintext
propertyDefinitions
The values for this attribute are defined in Section 4.5.1.3.
```

4.5.1.3 Property Definitions
The Presence class inherits property definitions from super classes. The Presence class MUST have the property definitions:

```plaintext
icom_core:lastModificationDate
```

<table>
<thead>
<tr>
<th>Description</th>
<th>Last modification date and time of information in a presence.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required</td>
<td>False</td>
</tr>
<tr>
<td>Inherited</td>
<td>False</td>
</tr>
<tr>
<td>Property Type</td>
<td>DateTime</td>
</tr>
<tr>
<td>Cardinality</td>
<td>Single</td>
</tr>
<tr>
<td>Updatability</td>
<td>Read Only</td>
</tr>
</tbody>
</table>
icom_core:location

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

icom_presence:EditMode

Description: Indicates a mode which determines whether a presence is editable.
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.
4.5.2 PresenceEditMode

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

4.5.2.2 Class Definition
The PresenceEditMode class is a mixin class which defines a mode that indicates whether a presence is editable.
The PresenceEditMode class has attribute values:

```plaintext
localNamespace
Value: icom_presence

localName
Value: PresenceEditMode

extendsFrom
Value: 
```
5467 stereotype
5468     Value: mixin
5470
description
5472     Value: PresenceEditMode is a mixin class which defines a mode that indicates whether a
5473         presence is editable.
5474
5475 propertyDefinitions
5476     The values for this attribute are defined in Section 4.5.2.3.
5477
5478 4.5.2.3 Property Definitions
5479 The PresenceEditMode class MAY include additional property definitions which are implementation-defined.
5480
5481 4.5.3 PresenceEditModeEnum
5482 The PresenceEditModeEnum class is an enum class that enumerates the instances each of which
5483 expresses a mode that indicates whether a presence is editable.
5484 The PresenceEditModeEnum class has attribute values:
5485
5486     localNamespace
5487         Value: icom_presence
5488
5489     localName
5490         Value: PresenceEditMode
5491
5492     extendsFrom
5493         Value: icom_presence:PresenceEditMode
5494
5495     stereotype
5496         Value: primary
5497
5498     isEnumeration
5499         Value: TRUE
5500
5501     description
5502         Value: A mode that indicates whether a presence is editable.
5503
5504     instances
5505         Value: <icom_presence:PresentityCopy, icom_presence:ViewerCopy>
5506
5507 ICOM defines two presence editable modes:
4.5.4 ContactMethod

4.5.4.1 Description
A contact method object describes reachability circumstances of a presentity.

4.5.4.2 Class Definition
The ContactMethod class has attribute values:

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

4.5.4.3 Property Definitions
The ContactMethod class MUST have the property definitions:

**icom_core:creationDate**
- Description: Creation date and time of information in a contact method.
- Required: False
- Inherited: False
- Property Type: DateTime
- Cardinality: Single
- Updatability: Read Only
icom_core: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 IRIs 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_prese:note

Description: A note about contacting a presentity.
Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Only

4.5.5 ContactReachabilityStatus

4.5.5.1 Description

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

4.5.5.2 Class Definition

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

The ContactReachabilityStatus class has attribute values:

- **localNamespace**
  - Value: icom_presence

- **localName**
  - Value: ContactReachabilityStatus

- **extendsFrom**
  - Value: 

- **stereotype**
  - Value: mixin

- **description**
  - Value: ContactReachabilityStatus is a mixin class which defines a status of a contact method.

- **propertyDefinitions**
  - The values for this attribute are defined in Section 4.5.5.3.

4.5.5.3 Property Definitions

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

4.5.6 ContactReachabilityStatusEnum

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

- **localNamespace**
  - Value: `icom_presence`

- **localName**
  - Value: `ContactReachabilityStatusEnum`

- **extendsFrom**
  - Value: `icom_presence:ContactReachabilityStatus`

- **stereotype**
  - Value: `primary`

- **isEnumeration**
  - Value: `TRUE`

- **description**
  - Value: A reachability status of a contact method.

- **instances**

ICOM defines six reachability status:

- **icom_presence:Reachable** a presentity is reachable through a contact method.
- **icom_presence:NotReachable** a presentity is not reachable through a contact method.
- **icom_presence:Chatty** a presentity is chatty.
- **icom_presence:Away** a presentity is away.
- **icom_presence:ExtendedAway** a presentity is away for an extended period.
- **icom_presence:DoNotDisturb** a presentity prefers not to be disturbed.
Figure 33: Presence Contact Method and Instant Message Connection Class Diagram.
4.5.7 Activity

4.5.7.1 Description
An activity object describes what a presentity is doing.

4.5.7.2 Class Definition
The Activity class has attribute values:

  localNamespace
      Value:  icom_presence

  localName
      Value:  Activity

  extendsFrom
      Value:

  stereotype
      Value:  primary

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

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

4.5.7.3 Property Definitions
The Activity class MUST have the property definitions:

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

  icom_core:endDate
      Description:  End date and time of an activity.
      Required:  True
      Inherited:  False
      Property Type:  DateTime
      Cardinality:  Single
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_core:Entity
- **Cardinality**: Single
- **Updatability**: Read Write

### 4.5.8 ActivityType

#### 4.5.8.1 Description

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

#### 4.5.8.2 Class Definition

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

The ActivityType class has attribute values:

- **localNamespace**
  - **Value**: icom_presence

- **localName**
  - **Value**: ActivityType
extendsFrom
    Value:

stereotype
    Value: mixin

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

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

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

4.5.9 ActivityTypeEnum
The ActivityTypeEnum class is an enum class that enumerates the instances each of which expresses a type of activity.
The ActivityTypeEnum class has attribute values:

    localNamespace
        Value: icom_presence

    localName
        Value: ActivityTypeEnum

    extendsFrom
        Value: icom_presence:ActivityType

    stereotype
        Value: primary

    isEnumeration
        Value: TRUE

    description
        Value: 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:OtherActivityType>
ICOM defines eleven activity types:

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

### 4.6 Address Book Module

#### 4.6.1 AddressBook

#### 4.6.1.1 Description

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

#### 4.6.1.2 Class Definition

The AddressBook class has attribute values:

```plaintext
localNamespace
  Value: icom_card

localName
  Value: AddressBook

extendsFrom
  Value: icom_core:Folder

stereotype
  Value: primary

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

propertyDefinitions
  The values for this attribute are defined in Section 4.6.1.3.
```
4.6.1.3 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:PersonContact
- Cardinality: Multi
- Updatability: Read Only

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

Figure 34: Address Book Class Diagram.

4.6.2 PersonContact

4.6.2.1 Description

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

The PersonContact class has attribute values:

    localNamespace
    Value: icom_card

    localName
    Value: PersonContact

extendsFrom
    Value: icom_core:Artifact, icom_core:Addressable

stereotype
    Value: primary

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

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

4.6.2.3 Property Definitions

The PersonContact class inherits property definitions from super classes.

The PersonContact class MUST have the property definitions:

icom_core:timeZone
    Description: Time zone of a person.
    Required: False
    Inherited: False
    Property Type: icom_core:TimeZone
    Cardinality: Single
    Updatability: Read Write

icom_core:givenName
    Description: Given name of a person.
    Required: False
    Inherited: False
    Property Type: String
    Cardinality: Single
    Updatability: Read Write
icom_core:middleName
Description: Middle name of a person. Can include multiple names concatenated.
Required: False
Inherited: False
Property Type: String
Cardinality: Single
Updatability: Read Write

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

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

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

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

icom_core:jobTitle
Description: Job title of a person.
icom_core:department

5941 Description: A person’s affiliated department.
5942 Required: False
5943 Inherited: False
5944 Property Type: String
5945 Cardinality: Single
5946 Updatability: Read Write

icom_core:officeLocation

5949 Description: Location of a person’s department.
5950 Required: False
5951 Inherited: False
5952 Property Type: String
5953 Cardinality: Single
5954 Updatability: Read Write

icom_core:company

5965 Description: A person’s affiliated company.
5966 Required: False
5967 Inherited: False
5968 Property Type: String
5969 Cardinality: Single
5970 Updatability: Read Write

icom_core:profession

5973 Description: A person’s profession.
5974 Required: False
5975 Inherited: False

icom_content:attachment

5976 Description: One or more content attachments in a contact.
5977 Required: False
5978 Inherited: False
Property Type: icom_content:AttachedItem
Cardinality: Multi
Updatability: Read Write

icom_card:bookmark
Description: A person which is bookmarked by a contact.
Required: False
Inherited: False
Property Type: icom_core:Person
Cardinality: Single
Updatability: On Create

The PersonContact class MAY include additional property definitions which are implementation-defined.
4.7 Calendar Module

4.7.1 Calendar

4.7.1.1 Description

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

4.7.1.2 Class Definition

The Calendar class has attribute values:
localNamespace
  Value: icom_cal

localName
  Value: Calendar

declaresFrom
  Value: icom_core:Folder

stereotype
  Value: primary

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

propertyDefinitions
  The values for this attribute are defined in 4.7.1.3.

4.7.1.3 Property Definitions

The Calendar class inherits property definitions from super classes.

The Calendar class MUST have the property definitions:

icom_core:timeZone
  Description: Time zone setting for a calendar.
  Required: True
  Inherited: False
  Property Type: icom_core:TimeZone
  Cardinality: Single
  Updatability: Read Write

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

icom_cal:recurrence
  Description: Occurrence series of a calendar.
  Required: False
  Inherited: False
Property Type:  icom_cal:OccurrenceSeries
Cardinality:  Multi
Updatability:  Read Only

Figure 36: Calendar Class Diagram.

4.7.2 OccurrenceSeries
4.7.2.1 Description
An occurrence series represents a series of occurrences associated with the same calendar event.

4.7.2.2 Class Definition
The OccurrenceSeries class has attribute values:

localNamespace
Value:  icom_cal

localName
Value:  OccurrenceSeries

extendsFrom
Value:  icom_core:Artifact

stereotype
Value:  primary

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

propertyDefinitions
The values for this attribute are defined in 4.7.2.3.
4.7.2.3 Property Definitions

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

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Required</th>
<th>Inherited</th>
<th>Property Type</th>
<th>Cardinality</th>
<th>Updatability</th>
</tr>
</thead>
<tbody>
<tr>
<td>com_core:location</td>
<td>Location of an occurrence series.</td>
<td>False</td>
<td>False</td>
<td>com_core:Location</td>
<td>Single</td>
<td>Read Write</td>
</tr>
<tr>
<td>com_core:organizer</td>
<td>Organizer of an occurrence series.</td>
<td>True</td>
<td>False</td>
<td>com_core:Participant</td>
<td>Single</td>
<td>On Create</td>
</tr>
<tr>
<td>com_core:participant</td>
<td>Participants in an occurrence series.</td>
<td>False</td>
<td>False</td>
<td>com_cal:OccurrenceParticipant</td>
<td>Multi</td>
<td>Read Write</td>
</tr>
<tr>
<td>com_core:priority</td>
<td>Priority for an attendee of an occurrence series.</td>
<td>False</td>
<td>False</td>
<td>com_core:Priority</td>
<td>Single</td>
<td>Read Write</td>
</tr>
<tr>
<td>com_content:attachment</td>
<td>One or more content attachments in an occurrence series.</td>
<td>False</td>
<td>False</td>
<td>com_content:AttachedItem</td>
<td>Multi</td>
<td></td>
</tr>
</tbody>
</table>
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_core: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_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

<table>
<thead>
<tr>
<th>Description</th>
<th>Type of an occurrence series.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required</td>
<td>True</td>
</tr>
<tr>
<td>Inherited</td>
<td>False</td>
</tr>
<tr>
<td>Property Type</td>
<td>icom_cal:OccurrenceType</td>
</tr>
<tr>
<td>Cardinality</td>
<td>Single</td>
</tr>
<tr>
<td>Updatability</td>
<td>Read Write</td>
</tr>
</tbody>
</table>

### icom_cal:editMode

<table>
<thead>
<tr>
<th>Description</th>
<th>Indicates a mode which determines whether an occurrence series is editable.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required</td>
<td>False</td>
</tr>
<tr>
<td>Inherited</td>
<td>False</td>
</tr>
<tr>
<td>Property Type</td>
<td>icom_cal:OccurrenceEditMode</td>
</tr>
<tr>
<td>Cardinality</td>
<td>Single</td>
</tr>
<tr>
<td>Updatability</td>
<td>Read Only</td>
</tr>
</tbody>
</table>

### icom_cal:occurrence

<table>
<thead>
<tr>
<th>Description</th>
<th>Occurrences in an occurrence series.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required</td>
<td>False</td>
</tr>
<tr>
<td>Inherited</td>
<td>False</td>
</tr>
<tr>
<td>Property Type</td>
<td>icom_cal:Occurrence</td>
</tr>
<tr>
<td>Cardinality</td>
<td>Multi</td>
</tr>
<tr>
<td>Updatability</td>
<td>Read Only</td>
</tr>
</tbody>
</table>

### icom_cal:attendee

<table>
<thead>
<tr>
<th>Description</th>
<th>An attendee of an occurrence series.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required</td>
<td>False</td>
</tr>
<tr>
<td>Inherited</td>
<td>False</td>
</tr>
<tr>
<td>Property Type</td>
<td>icom_core:Participant</td>
</tr>
<tr>
<td>Cardinality</td>
<td>Single</td>
</tr>
<tr>
<td>Updatability</td>
<td>Read Only</td>
</tr>
</tbody>
</table>

### icom_cal:attendeeParticipantStatus

<table>
<thead>
<tr>
<th>Description</th>
<th>Participation status for an attendee of an occurrence series.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required</td>
<td>False</td>
</tr>
<tr>
<td>Inherited</td>
<td>False</td>
</tr>
<tr>
<td>Property Type</td>
<td>icom_cal:OccurrenceParticipantStatus</td>
</tr>
<tr>
<td>Cardinality</td>
<td>Single</td>
</tr>
<tr>
<td>Updatability</td>
<td>Read Write</td>
</tr>
</tbody>
</table>
icom_cal:transparency

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

icom_conf:conference

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

4.7.3.1 Description
An occurrence represents an event in a calendar.

4.7.3.2 Class Definition
The Occurrence class has attribute values:

```
localNamespace
  Value: icom_cal

localName
  Value: Occurrence
```
extendsFrom
  Value:  icom_core:Artifact

stereotype
  Value:  primary

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

propertyDefinitions
  The values for this attribute are defined in 4.7.3.3.

4.7.3.3 Property Definitions

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

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

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

icom_core:participant
  Description:  Participants of an occurrence.
  Required:  False
  Inherited:  False
  Property Type:  icom_cal:OccurrenceParticipant
  Cardinality:  Multi
  Updatability:  Read Write
<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
<th>Required</th>
<th>Inherited</th>
<th>Property Type</th>
<th>Cardinality</th>
<th>Updatability</th>
</tr>
</thead>
<tbody>
<tr>
<td>icoicm_core:priority</td>
<td>Priority for an attendee of an occurrence.</td>
<td>False</td>
<td>False</td>
<td>icoicm_core:Priority</td>
<td>Single</td>
<td>Read Write</td>
</tr>
<tr>
<td>icoicm_core:startDate</td>
<td>Start date and time of an occurrence.</td>
<td>True</td>
<td>False</td>
<td>DateTime</td>
<td>Single</td>
<td>On Create</td>
</tr>
<tr>
<td>icoicm_core:startDateResolution</td>
<td>Resolution of start date and time of an occurrence.</td>
<td>True</td>
<td>False</td>
<td>icoicm_core:DateTimeResolution</td>
<td>Single</td>
<td>On Create</td>
</tr>
<tr>
<td>icoicm_core:endDate</td>
<td>End date and time of an occurrence.</td>
<td>True</td>
<td>False</td>
<td>DateTime</td>
<td>Single</td>
<td>On Create</td>
</tr>
<tr>
<td>icoicm_core:endDateResolution</td>
<td>Resolution of end date and time of an occurrence.</td>
<td>True</td>
<td>False</td>
<td>icoicm_core:DateTimeResolution</td>
<td>Single</td>
<td>On Create</td>
</tr>
<tr>
<td>icoicm_content:attachment</td>
<td>One or more content attachments in an occurrence.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>icom_cal:occurrenceSeries</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Required</td>
<td>False</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inherited</td>
<td>False</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property Type</td>
<td>icom_cal:OccurrenceSeries</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cardinality</td>
<td>Single</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Updatability</td>
<td>Read Only</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>An occurrence is part of this occurrence series.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>icom_cal:fromRecurringOccurrenceSeries</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Required</td>
<td>False</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inherited</td>
<td>False</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property Type</td>
<td>Boolean</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cardinality</td>
<td>Single</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Updatability</td>
<td>Read Only</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Occurrence is part of a recurring occurrence series.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>icom_cal:exceptionToOccurrenceSeries</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Required</td>
<td>False</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inherited</td>
<td>False</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property Type</td>
<td>Boolean</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cardinality</td>
<td>Single</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Updatability</td>
<td>Read Only</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Occurrence is an exception to an occurrence series.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>icom_cal:occurrenceStatus</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Required</td>
<td>True</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inherited</td>
<td>False</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property Type</td>
<td>icom_cal:OccurrenceStatus</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cardinality</td>
<td>Single</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Updatability</td>
<td>Read Write</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Status of an occurrence.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>icom_cal:occurrenceType</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Required</td>
<td>True</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inherited</td>
<td>False</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property Type</td>
<td>icom_cal:OccurrenceType</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cardinality</td>
<td>Single</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Type of an occurrence.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Property Type: icom_cal:OccurrenceType
Cardinality: Single
Updatability: Read Write

**icom_cal:editMode**

Description: Indicates a mode which determines whether an occurrence is editable.
Required: False
Inherited: False
Property Type: icom_cal:OccurrenceEditMode
Cardinality: Single
Updatability: Read Only

**icom_cal:attendee**

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

**icom_cal:attendeeParticipantStatus**

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

**icom_cal:transparency**

Description: Transparency for an attendee of an occurrence.
Required: False
Inherited: False
Property Type: icom_cal:OccurrenceParticipantTransparency
Cardinality: Single
Updatability: Read Write

**icom_cal:attendeeProperty**

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

**icom_conf:conference**

**Description:** One or more conferences for an occurrence.

**Required:** False

**Inherited:** False

**Property Type:** icom_conf:Conference

**Cardinality:** Multi

**Updatability:** Read Write

---

**Figure 38:** Occurrence Class Diagram.
4.7.4 OccurrenceStatus

4.7.4.1 Description
An occurrence status is a status of a calendar occurrence.

4.7.4.2 Class Definition
The OccurrenceStatus class is a mixin class which defines status of a calendar occurrence.
The OccurrenceStatus class has attribute values:

localNamespace
Value: icom_cal

localName
Value: OccurrenceStatus

extendsFrom
Value: 

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

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

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

4.7.5 OccurrenceStatusEnum
The OccurrenceStatusEnum class is an enum class that enumerates the instances each of which expresses a status of an occurrence or occurrence series.
The OccurrenceStatusEnum class has attribute values:

localNamespace
Value: icom_cal

localName
Value: OccurrenceStatusEnum
extendsFrom
  Value: icom_cal:OccurrenceStatus

stereotype
  Value: primary

isEnumeration
  Value: TRUE

description
  Value: Status of an occurrence or occurrence series.

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

ICOM defines three occurrence status:

- icom_cal:Cancelled an occurrence or occurrence series is cancelled.
- icom_cal:Tentative an occurrence or occurrence series is tentative.
- icom_cal:Confirmed an occurrence or occurrence series is confirmed.

4.7.6 OccurrenceType

4.7.6.1 Description
An occurrence type is a category of calendar occurrences.

4.7.6.2 Class Definition
The OccurrenceType class is a mixin class which defines a type of occurrence.
The OccurrenceType class has attribute values:

localNamespace
  Value: icom_cal

localName
  Value: OccurrenceType

extendsFrom
  Value:

stereotype
  Value: mixin
description
Value: OccurrenceType is a mixin class which defines a type of occurrence.

propertyDefinitions
The values for this attribute are defined in Section 4.7.6.3.

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

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

The OccurrenceTypeEnum class has attribute values:

localNamespace
Value: icom_cal

localName
Value: OccurrenceTypeEnum

extendsFrom
Value: icom_cal:OccurrenceType

stereotype
Value: primary

isEnumeration
Value: TRUE

description
Value: Type of an occurrence or occurrence series.

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

ICOM defines five occurrence types:

- icom_cal:Meeting an occurrence or occurrence series is a meeting.
- icom_cal:DayEvent an occurrence or occurrence series is a day event.
- icom_cal:Holiday an occurrence or occurrence series is a holiday.
- icom_cal:JournalEntry an occurrence or occurrence series is a journal entry.
- icom_cal:OtherOccurrenceType an occurrence or occurrence series is of other type.
4.7.8 OccurrenceParticipant

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

4.7.8.2 Class Definition
The OccurrenceParticipant class has attribute values:

- **localNamespace**
  - Value: icom_cal

- **localName**
  - Value: OccurrenceParticipant

- **extendsFrom**
  - Value: icom_core:Participant

- **stereotype**
  - Value: primary

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

**propertyDefinitions**
The values for this attribute are defined in Section 4.7.8.3.

4.7.8.3 Property Definitions
The OccurrenceParticipant class inherits property definitions from super classes.
The OccurrenceParticipant class MUST have the property definition:

- **icom_cal:participantStatus**
  - Description: Status of an occurrence participant.
  - Required: False
  - Inherited: False
  - Property Type: icom_cal:OccurrenceParticipantStatus
  - Cardinality: Single
  - Updatability: Read Write
4.7.9 OccurrenceParticipantStatus

4.7.9.1 Description
An occurrence participant status is a participant’s response status for an occurrence or occurrence series.

4.7.9.2 Class Definition
The OccurrenceParticipantStatus class is a mixin class which defines a participant’s response status for an occurrence or occurrence series.
The OccurrenceParticipantStatus class has attribute values:

```
localNamespace
   Value: icom_cal

localName
   Value: OccurrenceParticipantStatus

extendsFrom
   Value:

stereotype
   Value: mixin

description
   Value: OccurrenceParticipantStatus is a mixin class which defines a participant’s response status for an occurrence or occurrence series.

propertyDefinitions
   The values for this attribute are defined in Section 4.7.9.3.
```

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

4.7.10 OccurrenceParticipantStatusEnum
The OccurrenceParticipantStatusEnum 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 OccurrenceParticipantStatusEnum class has attribute values:

```
localNamespace
   Value: icom_cal

localName
   Value: OccurrenceParticipantStatusEnum
```
ICOM defines four occurrence participant’s status:

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

### 4.7.11 OccurrenceParticipantTransparency

#### 4.7.11.1 Description

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

The OccurrenceParticipantTransparency class has attribute values:

```plaintext
localNamespace
  Value: icom_cal

localName
  Value: OccurrenceParticipantTransparency

detributesFrom
  Value:
```

```plaintext
```
stereotype
Value: mixin

description
Value: OccurrenceParticipantTransparency is a mixin class which defines visibility of an occurrence or occurrence series in a participant’s calendar or free busy.

propertyDefinitions
The values for this attribute are defined in Section 4.7.11.3.

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

4.7.12 OccurrenceParticipantTransparencyEnum
The OccurrenceParticipantTransparencyEnum 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 OccurrenceParticipantTransparencyEnum class has attribute values:

  stereotype
  Value: primary

  isEnumeration
  Value: TRUE

  instances
ICOM defines five participant transparencies:

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

### 4.7.13 OccurrenceEditMode

#### 4.7.13.1 Description

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

#### 4.7.13.2 Class Definition

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

The OccurrenceEditMode class has attribute values:

```
  localNamespace
    Value: icom_cal

  localName
    Value: OccurrenceEditMode

  extendsFrom
    Value: 

  stereotype
    Value: mixin

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

  propertyDefinitions
    The values for this attribute are defined in Section 4.7.13.3.
```

#### 4.7.13.3 Property Definitions

The OccurrenceEditMode class MAY include additional property definitions which are implementation-defined.
4.7.14 OccurrenceEditModeEnum

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

The OccurrenceEditModeEnum class has attribute values:

```plaintext
6738
localNamespace
Value: icom_cal

6740
localName
Value: OccurrenceEditMode

6742
extendsFrom
Value: icom_cal:OccurrenceEditMode

6744
stereotype
Value: primary

6746
isEnumeration
Value: TRUE

6748
description
Value: A mode that indicates whether an occurrence or occurrence series is editable.

6750
instances
Value: <icom_cal:OrganizerCopy, icom_cal:AttendeeCopy>
```

ICOM defines two occurrence editable modes:

- **icom_cal:OrganizerCopy** 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** an occurrence or occurrence series is a copy delivered to an attendee who may only update the attendee properties such as priority, transparency, etc.

4.8 Free Busy Module

4.8.1 FreeBusy

4.8.1.1 Description

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

The FreeBusy class has attribute values:

localNamespace

   Value: icom_cal

localName

   Value: FreeBusy

declaration

   Value: primary

description

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

propertyDefinitions

   The values for this attribute are defined in Section 4.8.1.3.

4.8.1.3 Property Definitions

The FreeBusy class MUST have the property definitions:

icom_core:participant

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

icom_core:creationDate

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

icom_core:startDate

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

### 4.8.2 FreeBusyInterval

#### 4.8.2.1 Description

A free busy interval 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.

#### 4.8.2.2 Class Definition

The FreeBusyInterval class has attribute values:

- **localNamespace**
  - Value: icom_cal

- **localName**
  - Value: FreeBusyInterval

- **extendsFrom**
  - Value:
6855  stereotype
6856       Value: primary
6857
6858  description
6859       Value: A free busy interval object specifies an interval of free or busy time.
6860
6861  propertyDefinitions
6862       The values for this attribute are defined in Section 4.8.2.3
6863
6864  4.8.2.3 Property Definitions
6865       The FreeBusyInterval class MUST have the property definitions:
6866
6867  icom_core:startDate
6868       Description: Start date and time of a free busy interval.
6869       Required: False
6870       Inherited: False
6871       Property Type: DateTime
6872       Cardinality: Single
6873       Updatability: Read Only
6874
6875  icom_core:endDate
6876       Description: End date and time of a free busy interval.
6877       Required: False
6878       Inherited: False
6879       Property Type: DateTime
6880       Cardinality: Single
6881       Updatability: Read Only
6882
6883  icom_cal:freeBusyType
6884       Description: A type of free busy interval.
6885       Required: False
6886       Inherited: False
6887       Property Type: icom_cal:FreeBusyType
6888       Cardinality: Single
6889       Updatability: Read Only
4.8.3 FreeBusyType

4.8.3.1 Description

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

4.8.3.2 Class Definition

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

The FreeBusyType class has attribute values:

- **localNamespace**
  Value: icom_cal

- **localName**
  Value: FreeBusyType

- **extendsFrom**
  Value:

- **stereotype**
  Value: mixin

- **description**
  Value: FreeBusyType is a mixin class which defines different types to classify a time interval.
The values for this attribute are defined in Section 4.8.3.3.

4.8.3.3 Property Definitions

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

4.8.4 FreeBusyTypeEnum

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

The FreeBusyTypeEnum class has attribute values:

```plaintext
localNamespace
  Value: icom_cal

localName
  Value: FreeBusyTypeEnum

extendsFrom
  Value: icom_cal:FreeBusyType

stereotype
  Value: primary

isEnumeration
  Value: TRUE

description
  Value: 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:OtherFreeBusyType>

ICOM defines six free busy types:
```
- `icom_cal:Free` a free busy interval is free.
- `icom_cal:Busy` a free busy interval is busy.
- `icom_cal:Tentative` a free busy interval is tentative.
- `icom_cal:OutsideAvailableHours` a free busy interval is outside available hours.
- `icom_cal:OutOfOffice` a free busy interval is within out of office hours.
- `icom_cal:OtherFreeBusyType` a free busy interval is of other type.
```
4.9 Task List Module

4.9.1 TaskList

4.9.1.1 Description
A task list contains task management artifacts.

4.9.1.2 Class Definition
The TaskList class has attribute values:

```plaintext
localNamespace
  Value:  icom_task

localName
  Value:  TaskList

extendsFrom
  Value:  icom_core:Folder

stereotype
  Value:  primary

description
  Value:  A task list contains task management artifacts.

propertyDefinitions
  The values for this attribute are defined in 4.9.1.3.
```

4.9.1.3 Property Definitions
The TaskList class inherits property definitions from super classes.
The TaskList class MUST have the property definitions:

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

icom_core:element
  Description:  Elements of a task list.
  Required:  False
```
4.9.2 Task

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

4.9.2.2 Class Definition
The Task class has attribute values:

- **localNamespace**
  - Value: `icom_task`

- **localName**
  - Value: Task

- **extendsFrom**
  - Value: `icom_core: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.

Figure 40: Task List Class Diagram.
The values for this attribute are defined in 4.9.2.3.

### 4.9.2.3 Property Definitions

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

#### `icom_core:location`

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

#### `icom_core:organizer`

- Description: Organizer of a task.
- Required: False
- Inherited: False
- Property Type: `icom_core:Participant`
- Cardinality: Single
- Updatability: On Create

#### `icom_core:priority`

- Description: Priority of a task.
- Required: False
- Inherited: False
- Property Type: `icom_core:Priority`
- Cardinality: Single
- Updatability: Read Write

#### `icom_core:startDate`

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

#### `icom_core:startDateResolution`

- Description: Resolution of start date and time of a task.
- Required: True
7064  Inherited: False
7065  Property Type: icom_core:DateTimeResolution
7066  Cardinality: Single
7067  Updatability: On Create
7068
7069  **icom_content:attachment**
7070  Description: One or more content attachments in a task.
7071  Required: False
7072  Inherited: False
7073  Property Type: icom_content:AttachedItem
7074  Cardinality: Multi
7075  Updatability: Read Write
7076
7077  **icom_task:dueDate**
7078  Description: Due date and time of a task.
7079  Required: True
7080  Inherited: False
7081  Property Type: DateTime
7082  Cardinality: Single
7083  Updatability: On Create
7084
7085  **icom_task:dueDateResolution**
7086  Description: Resolution of due date and time of a task.
7087  Required: True
7088  Inherited: False
7089  Property Type: icom_core:DateTimeResolution
7090  Cardinality: Single
7091  Updatability: On Create
7092
7093  **icom_task:editMode**
7094  Description: Indicates a mode which determines whether a task is editable.
7095  Required: False
7096  Inherited: False
7097  Property Type: icom_task:TaskEditMode
7098  Cardinality: Single
7099  Updatability: Read Only
7100
7101  **icom_task:taskStatus**
7102  Description: Status of a task.
7103  Required: True
7104  Inherited: False
7105  Property Type: icom_task:TaskStatus
7106
Cardinality: Single
Updatability: Read Write

**icom_task:assignee**

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

**icom_task:participantStatus**

Description: Participation status of a task.
Required: False
Inherited: False
Property Type: icom_task:TaskParticipantStatus
Cardinality: Single
Updatability: Read Write

**icom_task:completionDate**

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

**icom_task:completionDateResolution**

Description: Resolution of completion date and time of a task.
Required: False
Inherited: False
Property Type: icom_core:DateTimeResolution
Cardinality: Single
Updatability: Read Write

**icom_task:percentComplete**

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

Description: Extensible properties for an assignee of a task.

Required: False

Inherited: False

Property Type: icom_meta:Property

Cardinality: Multi

Updatability: Read Write

**Figure 41: Task Class Diagram.**
4.9.3 TaskStatus

4.9.3.1 Description
A task status is a status of a task.

4.9.3.2 Class Definition
The TaskStatus class is a mixin class which defines status of a task.
The TaskStatus class has attribute values:

    localNamespace
    Value:  icom_task

    localName
    Value:  TaskStatus

    extendsFrom
    Value:

    stereotype
    Value:  mixin

    description
    Value:  TaskStatus is a mixin class which defines status of a task.

    propertyDefinitions
    The values for this attribute are defined in Section 4.9.3.3.

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

4.9.4 TaskStatusEnum
The TaskStatusEnum class is an enum class that enumerates the instances each of which expresses a status of task.
The TaskStatusEnum class has attribute values:

    localNamespace
    Value:  icom_task

    localName
    Value:  TaskStatusEnum

    extendsFrom
    Value:  icom_task:TaskStatus
7201 stereotype
7202     Value: primary
7204
isEnumeration
7206     Value: TRUE
7207
description
7208     Value: Status of a task.
7210
7211 instances
7212     Value: <icom_task:NeedsAction, icom_task:InProgress, icom_task:Completed, icom_task:Cancelled>
7214
7215 ICOM defines four task statuses:
7216     • icom_task:NeedsAction a task needs action.
7217     • icom_task:InProgress a task is in progress.
7218     • icom_task:Completed a task is completed.
7219     • icom_task:Cancelled a task is cancelled.
7220
4.9.5 TaskParticipantStatus

4.9.5.1 Description
7222 A task participant status is a participant’s response status for a task assignment.

4.9.5.2 Class Definition
7224 The TaskParticipantStatus class is a mixin class which defines a participant’s response status for a task assignment.
7225 The TaskParticipantStatus class has attribute values:
7226
class
7227     localNamespace
7228     Value: icom_task
7230
class
7232     localName
7233     Value: TaskParticipantStatus
7235
class
7237     extendsFrom
7238     Value:
7240
class
7242     stereotype
7243     Value: mixin
7241   description
7242       TaskParticipantStatus is a mixin class which defines a participant’s response status for a task assignment.
7244
7245   propertyDefinitions
7246       The values for this attribute are defined in Section 4.9.5.3.
7247
7248   4.9.5.3 Property Definitions
7249       The TaskParticipantStatus class MAY include additional property definitions which are implementation-defined.
7250
7251   4.9.6 TaskParticipantStatusEnum
7252       The TaskParticipantStatusEnum class is an enum class that enumerates the instances each of which expresses a participant's response status for a task.
7253       The TaskParticipantStatusEnum class has attribute values:
7254
7255     localNamespace
7256         Value:  icom_task
7258
7259     localName
7260         Value:  TaskParticipantStatusEnum
7261
7262     extendsFrom
7263         Value:  icom_task:TaskParticipantStatus
7264
7265     stereotype
7266         Value:  primary
7267
7268     isEnumeration
7269         Value:  TRUE
7270
7271     description
7272         Value:  Participant’s response status for a task.
7273
7274     instances
7275         Value:  <icom_task:NeedsAction, icom_task:Accepted, icom_task:Declined, icom_task:InProgress, icom_task:Completed, icom_task:WaitingOnOther, icom_task:Tentative, icom_task:Deferred>
7277
7278
7279   ICOM defines eight task participant's status:
7280     •  icom_task:NeedsAction an assignee needs to act on a task.
7281     •  icom_task:Accepted an assignee accepted a task.
7282     •  icom_task:Declined an assignee declined a task.
- **icom_task:InProgress** a task is in progress.
- **icom_task:Completed** a task is completed.
- **icom_task:WaitingOnOther** an assignee is waiting on other.
- **icom_task:Tentative** an assignee is tentative about a task.
- **icom_task:Deferred** an assignee deferred a task.

### 4.9.7 TaskEditMode

#### 4.9.7.1 Description

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

#### 4.9.7.2 Class Definition

The TaskEditMode class is a mixin class which defines a mode that indicates whether a task is editable.

The TaskEditMode class has attribute values:

```plaintext
    localNamespace
      Value:  icom_task

    localName
      Value:  TaskEditMode

    extendsFrom
      Value:  [Value:  ]

    stereotype
      Value:  mixin

    description
      Value:  TaskEditMode is a mixin class which defines a mode that indicates whether task is editable.

    propertyDefinitions
      The values for this attribute are defined in Section 4.9.7.3.
```

#### 4.9.7.3 Property Definitions

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

#### 4.9.8 TaskEditModeEnum

The TaskEditModeEnum class is an enum class that enumerates the instances each of which expresses an editable mode of a task.
The TaskEditModeEnum class has attribute values:

- **localNamespace**
  - Value: icom_task

- **localName**
  - Value: TaskEditModeEnum

- **extendsFrom**
  - Value: icom_task:TaskEditMode

- **stereotype**
  - Value: primary

- **isEnumeration**
  - Value: TRUE

- **description**
  - Value: A mode that indicates whether a task is editable.

- **instances**
  - Value: <icom_task:OrganizerCopy, icom_task:AssigneeCopy>

ICOM defines two task editable modes:

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

### 4.10 Forum Module

#### 4.10.1 Discussion

##### 4.10.1.1 Description

A discussion is an item in a discussion container.

#### 4.10.1.2 Class Definition

The Discussion class is a mixin class that defines the characteristics of artifacts that can be elements of discussion containers.

The Discussion class has attribute values:

- **localNamespace**
  - Value: icom_forum
localName
Value: Discussion

extendsFrom
Value: icom_core:Item

stereotype
Value: mixin

description
Value: Discussion is a mixin class that defines the characteristics of artifacts that can be placed in a discussion container.

propertyDefinitions
The values for this attribute are defined in Section 4.10.1.3.

4.10.1.3 Property Definitions
The Discussion class inherits property definitions from super classes.
The Discussion class MUST have the property definition:

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.

4.10.2 DiscussionContainer

4.10.2.1 Description
A discussion container contains discussion items.

4.10.2.2 Class Definition
The DiscussionContainer class is a mixin class that defines the characteristics of folders that contain Discussion items.
The DiscussionContainer class has attribute values:

```plaintext
localNamespace
   Value: icom_forum

localName
   Value: DiscussionContainer

extendsFrom
   Value: icom_core:Container

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

propertyDefinitions
   The values for this attribute are defined in Section 4.10.2.3.
```

### 4.10.2.3 Property Definitions

The DiscussionContainer class inherits property definitions from super classes.

The DiscussionContainer class MUST have the property definition:

```plaintext
icom_core: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.

### 4.10.3 DiscussionMessage

#### 4.10.3.1 Description

A discussion message is a message in a forum discussion thread.
4.10.3.2 Class Definition

The DiscussionMessage class has attribute values:

- **localNamespace**
  - Value: icom_forum
- **localName**
  - Value: DiscussionMessage
- **extendsFrom**
  - Value: icom_msg:Message, icom_forum:Discussion
- **stereotype**
  - Value: primary
- **description**
  - Value: Discussion message is a message in a forum discussion thread.

**propertyDefinitions**
The values for this attribute are defined in Section 4.10.3.3.

4.10.3.3 Property Definitions

The DiscussionMessage class inherits property definitions from super classes.

The DiscussionMessage class MUST have the property definition:

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

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

4.10.4 TopicContainer

4.10.4.1 Description

A topic container contains topics.
4.10.4.2 Class Definition

The TopicContainer class is a mixin class which defines the characteristics of folders that contain Topics.

The TopicContainer class has attribute values:

```
localNamespace
    Value: icom_forum
```

```
localName
    Value: TopicContainer
```

```
extendsFrom
    Value: icom_core:Container
```

```
stereotype
    Value: mixin
```

```
description
    Value: TopicContainer is a mixin class that defines the characteristics of folders that contain topics.
```

```
propertyDefinitions
    The values for this attribute are defined in Section 4.10.4.3.
```

4.10.4.3 Property Definitions

The TopicContainer class inherits property definitions from super classes.

The TopicContainer class MUST have the property definitions:

```
icom_core:element
```

<table>
<thead>
<tr>
<th>Description</th>
<th>Property Type</th>
<th>Cardinality</th>
<th>Required</th>
<th>Updatability</th>
<th>Inherited</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elements of a topic container.</td>
<td>icom_forum:Topic</td>
<td>Multi</td>
<td>False</td>
<td>Read Only</td>
<td>True</td>
</tr>
</tbody>
</table>

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

4.10.5.1 Description
A forum contains sub-forums, topics, and announcements.

4.10.5.2 Class Definition
The Forum class has attribute values:

```plaintext
localNamespace
Value: icom_forum
```
localName
Value: Forum

extendsFrom
Value: icom_core:Folder, icom_forum:TopicContainer

description
Value: A forum contains sub-forums, topics, and announcements.

propertyDefinitions
The values for this attribute are defined in Section 4.10.5.3.

4.10.5.3 Property Definitions
The Forum class inherits property definitions from super classes.
The Forum class MUST have the property definitions:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Required</th>
<th>Inherited</th>
<th>Property Type</th>
<th>Cardinality</th>
<th>Updatability</th>
</tr>
</thead>
<tbody>
<tr>
<td>icom_forum:lastPost</td>
<td>The last posted discussion in a forum.</td>
<td>False</td>
<td>False</td>
<td>icom_forum:Discussion</td>
<td>Single</td>
<td>Read Only</td>
</tr>
<tr>
<td>icom_forum:forum</td>
<td>Sub-forums of a forum.</td>
<td>False</td>
<td>False</td>
<td>icom_forum:Forum</td>
<td>Multi</td>
<td>Read Only</td>
</tr>
<tr>
<td>icom_forum:topic</td>
<td>Topics of a forum.</td>
<td>False</td>
<td>False</td>
<td>icom_forum:Topic</td>
<td>Multi</td>
<td>Read Only</td>
</tr>
</tbody>
</table>
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.

4.10.6 Topic

4.10.6.1 Description
A topic contains conversations among forum participants. The discussions in a topic may be sorted in chronological order or threaded by reply.

4.10.6.2 Class Definition
The Topic class has attribute values:

description
Value: A topic contains discussion threads.

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

4.10.6.3 Property Definitions
The Topic class inherits property definitions from super classes.
The Topic class MUST have the property definitions:

icom_core:element
Description: Elements of a topic.
4.10.7 Announcement

4.10.7.1 Description

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

4.10.7.2 Class Definition

The Announcement class has attribute values:

```plaintext
localNamespace
    Value:  icom_forum

localName
    Value:  Announcement

extendsFrom
    Value:  icom_forum:Topic
```
The values for this attribute are defined in Section 4.10.7.3.

4.10.7.3 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.
4.10.8 AnnouncementStatus

4.10.8.1 Description
An announcement status is status of an announcement.

4.10.8.2 Class Definition
The AnnouncementStatus class is a mixin class which defines status of an announcement.

The AnnouncementStatus class has attribute values:

```
   localNamespace
       Value:  icom_forum

   localName
       Value:  AnnouncementStatus

   extendsFrom
       Value:

   stereotype
       Value:  mixin

   description
       Value:  AnnouncementStatus is a mixin class which defines status of an announcement.

   propertyDefinitions
       The values for this attribute are defined in Section 4.10.8.3.
```

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

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

The AnnouncementStatusEnum class has attribute values:

```
   localNamespace
       Value:  icom_forum

   localName
       Value:  AnnouncementStatusEnum
```
ICOM defines three announcement status:

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

### 4.11 Conference Module

#### 4.11.1 Conference

#### 4.11.1.1 Description

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

#### 4.11.1.2 Class Definition

The Conference class has attribute values:

```xml
localNamespace
    Value: icom_conf

localName
    Value: Conference

extendsFrom
    Value: icom_core:Folder

stereotype
    Value: primary
```
description
Value: A conference represents a durable context for online conference sessions.

propertyDefinitions
The values for this attribute are defined in Section 4.11.1.3.

4.11.1.3 Property Definitions
The Conference class inherits property definitions from super classes.
The Conference class MUST have the property definitions:

icom_core:organizer
Description: Organizer of a conference.
Required: False
Inherited: False
Property Type: icom_core: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:conferenceStatus
Description: Status of a conference.
Required: False
Inherited: False
Property Type: icom_conf:ConferenceStatus
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
<table>
<thead>
<tr>
<th>Property Type</th>
<th>Description</th>
<th>Required</th>
<th>Inherited</th>
<th>Cardinality</th>
<th>Updatability</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>icom_conf:conferenceSetting</code></td>
<td>Configurable settings of a conference.</td>
<td>False</td>
<td>False</td>
<td>Single</td>
<td>Read Only</td>
</tr>
<tr>
<td><code>icom_conf:transcript</code></td>
<td>Transcripts from ended sessions of a conference.</td>
<td>False</td>
<td>False</td>
<td>Multi</td>
<td>Read Write</td>
</tr>
<tr>
<td><code>icom_conf:scheduledStartDate</code></td>
<td>Scheduled start date and time of a conference session.</td>
<td>False</td>
<td>False</td>
<td>Single</td>
<td>Read Write</td>
</tr>
<tr>
<td><code>icom_conf:scheduledEndDate</code></td>
<td>Scheduled end date and time of a conference session.</td>
<td>False</td>
<td>False</td>
<td>Single</td>
<td>Read Write</td>
</tr>
</tbody>
</table>

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

4.11.2.1 Description

A conference type represents a category of conferences.

4.11.2.2 Class Definition

The ConferenceType class is a mixin class which defines a type of conference.

The ConferenceType class has attribute values:

```
localNamespace
Value: icom_conf
```
localName
  Value: ConferenceType

extendsFrom
  Value:

stereotype
  Value: mixin

description
  Value: ConferenceType is a mixin class which defines type of conference.

propertyDefinitions
  The values for this attribute are defined in Section 4.11.2.3.

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

4.11.3 ConferenceTypeEnum
  The ConferenceTypeEnum class is an enum class that enumerates the instances each of which expresses a type of a conference.
  The ConferenceTypeEnum class has attribute values:

  localNamespace
    Value: icom_conf

  localName
    Value: ConferenceTypeEnum

  extendsFrom
    Value: icom_conf:ConferenceType

  stereotype
    Value: primary

  isEnumeration
    Value: TRUE

  description
    Value: A type of a conference.
ICOM defines four conference types:

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

### 4.11.4 ConferenceStatus

#### 4.11.4.1 Description

A conference status is status of an online conference.

#### 4.11.4.2 Class Definition

The ConferenceStatus class is a mixin class which defines status of an online conference. The ConferenceStatus class has attribute values:

- **localNamespace**
  - Value: icom_conf

- **localName**
  - Value: ConferenceStatus

- **extendsFrom**
  - Value:

- **stereotype**
  - Value: mixin

- **description**
  - Value: ConferenceStatus is a mixin class which defines status of an online conference.

- **propertyDefinitions**
  - The values for this attribute are defined in Section 4.11.4.3.

#### 4.11.4.3 Property Definitions

The ConferenceStatus class MAY include additional property definitions which are implementation-defined.
### 4.11.5 ConferenceStatusEnum

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

The ConferenceStateEnum class has attribute values:

```plaintext
localNamespace
Value: icom_conf

localName
Value: ConferenceStatusEnum

extendsFrom
Value: icom_conf:ConferenceStatus

stereotype
Value: primary

isEnumeration
Value: TRUE

description
Value: Status of a conference.

instances
Value: <icom_conf:NotStarted, icom_conf:WaitingForHost, icom_conf:Running, icom_conf:Hibernating, icom_conf:Ended>
```

ICOM defines five conference status:

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

### 4.11.6 ConferenceSession

#### 4.11.6.1 Description

A conference session represents the metadata for a session of a conference.
4.11.6.2 Class Definition

The ConferenceSession class has attribute values:

- **localNamespace**
  - Value: icom_conf

- **localName**
  - Value: ConferenceSession

- **extendsFrom**
  - Value: icom_core:Identifiable

- **stereotype**
  - Value: primary

- **description**
  - Value: A conference session represents the metadata for a session of a conference.

- **propertyDefinitions**
  - The values for this attribute are defined in Section 4.11.6.3.

4.11.6.3 Property Definitions

The ConferenceSession class inherits property definitions from super classes.

The ConferenceSession class MUST have the property definitions:

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

- **icom_core:startDate**
  - Description: Start date and time of a conference session.
  - Required: False
  - Inherited: False
  - Property Type: DateTime
  - Cardinality: Single
  - Updatability: Read Only
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Required</th>
<th>Inherited</th>
<th>Property Type</th>
<th>Cardinality</th>
<th>Updatability</th>
</tr>
</thead>
<tbody>
<tr>
<td>icom_core:EndDate</td>
<td>End date and time of a conference session.</td>
<td>False</td>
<td>False</td>
<td>DateTime</td>
<td>Single</td>
<td>Read Only</td>
</tr>
<tr>
<td>icom_conf:comment</td>
<td>Comment on a conference session.</td>
<td>False</td>
<td>False</td>
<td>String</td>
<td>Single</td>
<td>Read Write</td>
</tr>
<tr>
<td>icom_conf:rating</td>
<td>Rating of a conference session.</td>
<td>False</td>
<td>False</td>
<td>String</td>
<td>Single</td>
<td>Read Write</td>
</tr>
<tr>
<td>icom_conf:serverAddress</td>
<td>Address of a server that hosts a conference session.</td>
<td>False</td>
<td>False</td>
<td>String</td>
<td>Single</td>
<td>Read Only</td>
</tr>
<tr>
<td>icom_conf:endingReason</td>
<td>Reason for ending a conference session.</td>
<td>False</td>
<td>False</td>
<td>icom_conf:ConferenceSessionEndingReason</td>
<td>Single</td>
<td>Read Only</td>
</tr>
</tbody>
</table>

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

4.11.7.1 Description
A conference session ending reason is an indication of how a conference session ended.

4.11.7.2 Class Definition
The ConferenceSessionEndingReason class is a mixin class which defines an indication of how a conference session ended.

The ConferenceSessionEndingReason class has attribute values:

```
localNamespace
   Value: icom_conf

localName
   Value: ConferenceSessionEndingReason

extendsFrom
   Value: 

stereotype
   Value: mixin

description
   Value: ConferenceSessionEndingReason is a mixin class which defines an indication of how a conference session ended.

propertyDefinitions
   The values for this attribute are defined in Section 4.11.7.3.
```

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

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

The ConferenceSessionEndingReasonEnum class has attribute values:

```
localNamespace
   Value: icom_conf

localName
   Value: ConferenceSessionEndingReasonEnum
ICOM defines four conference session states:

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

### 4.11.9 ConferenceSetting

#### 4.11.9.1 Description

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

#### 4.11.9.2 Class Definition

The ConferenceSetting class has attribute values:

```java
localNamespace
  Value: icom_conf

localName
  Value: ConferenceSetting

extendsFrom
  Value: ConferenceSessionEndingReason

stereotype
  Value: primary

isEnumeration
  Value: TRUE

description
  Value: Reason for ending a conference session.

instances
  Value: <icom_conf:HostLeft, icom_conf:HostAborted, icom_conf:NoHost, icom_conf:Hibernating>
```
8127     description
8128         Value: A conference setting represents configuration settings for sessions of a conference.
8129
8130     propertyDefinitions
8131         The values for this attribute are defined in Section 4.11.9.3.
8132
8133     4.11.9.3 Property Definitions
8134         The ConferenceSetting class inherits property definitions from super classes.
8135         The ConferenceSetting class MUST have the property definitions:
8136
8137     icom_meta:property
8138         Description: Configurable properties for a conference.
8139         Required: False
8140         Inherited: False
8141         Property Type: icom_meta:property
8142         Cardinality: Multi
8143         Updatability: Read Write
8144
8145     icom_conf:participantRole
8146         Description: Role settings for conference participants.
8147         Required: False
8148         Inherited: False
8149         Property Type: icom_conf:ConferenceParticipantRole
8150         Cardinality: Multi
8151         Updatability: Read Write
8152
8153         The ConferenceSetting class MAY include additional property definitions which are implementation-defined.
8154
8155     4.11.10 ConferenceParticipantRole
8156
8157     4.11.10.1 Description
8158         A conference participant role defines roles settings for a conference participant.
8159
8160     4.11.10.2 Class Definition
8161         The ConferenceParticipantRole class has attribute values:
8162
8163     localNamespace
8164         Value: icom_conf
8165
8166     localName
8167         Value: ConferenceParticipantRole
extendsFrom
    Value:

stereotype
    Value: primary

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

propertyDefinitions
    The values for this attribute are defined in Section 4.11.10.3.

4.11.10.3 Property Definitions
The ConferenceParticipantRole class MUST have the property definitions:

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

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

icom_meta:property
    Description: Configurable properties for a role setting.
    Required: False
    Inherited: False
    Property Type: icom_meta:Property
    Cardinality: Multi
    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

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

5.1 Software Architecture or Framework Dependence

The ICOM specification does not presume a particular software architecture or framework for use of the ICOM model. Fulfillment of ICOM use case roles and accompanying responsibilities is implementation dependent.

5.2 Platform Provider Conformance

5.2.1 Platform Provider Conformance – No Extension Modules

An ICOM platform provider with no extension modules (Section 4):

a. SHALL conform to all mandatory statements and
b. MAY conform to optional statements

of the core ICOM model as defined in Section 3 of this standard.

5.2.2 Platform Provider Conformance – One or More Extension Modules

An ICOM platform provider with extension modules (Section 4):

a. SHALL conform to Section 5.2.1 and
b. SHALL conform to all mandatory statements and
c. MAY conform to optional statements

as defined in Section 4 for each extension module.

5.3 Service Provider Conformance

5.3.1 ICOM Service Provider – No Extension Modules

An ICOM service provider may provide one or more services defined in Section 3. For each such service provided, an ICOM service provider:

a. SHALL conform to all mandatory statements and
b. MAY conform to optional statements

for the classes, super classes, and related classes defined in Section 3 of this standard.

5.3.2 ICOM Service Provider – One or More Extension Modules

An ICOM service provider MAY support one or more extension modules as defined in Section 4 of this standard. For each service provided, an ICOM service provider:

a. SHALL conform to Section 5.3.1 (if an offered service is defined in Section 3) and
b. SHALL conform to all mandatory statements and
c. MAY conform to optional statements

as defined in Section 4 for that extension module.
5.4 ICOM Producer Conformance

5.4.1 ICOM Producer Conformance – No Extension Modules

An ICOM producer that produces no objects of a class conforming to Section 4:

- a. SHALL conform to all mandatory statements and
- b. MAY conform to optional statements

for the class and super classes thereof in Section 3 of this standard, for any object produced.

5.4.2 ICOM Producer Conformance – One or More Extension Modules

An ICOM producer that produces objects of a class conforming to Section 4:

- a. SHALL conform to Section 5.4.1 and
- b. SHALL conform to all mandatory statements and
- c. MAY conform to optional statements

as defined in Section 4 for that extension module.

5.5 ICOM Consumer Conformance

5.5.1 ICOM Consumer Conformance – No Extension Modules

An ICOM consumer that consumes no objects of a class conforming to Section 4:

- a. SHALL conform to all mandatory statements and
- b. MAY conform to optional statements

for the class and super classes thereof in Section 3 of this standard, for any object consumed.

5.5.2 ICOM Consumer Conformance – Extension Modules

An ICOM consumer that consumes objects of a class conforming to Section 4:

- a. SHALL conform to Section 5.5.1 and
- b. SHALL conform to all mandatory statements and
- c. MAY conform to optional statements

as defined in Section 4 for that extension module.
Appendix A. Acknowledgements

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

Participants:

- Rafiul Ahad, Oracle Corporation
- Kenneth P. Baclawski, Northeastern University
- Eric S. Chan, Oracle Corporation
- Martin Chapman, Oracle Corporation
- Scott Conroy, Individual
- Stefan Decker, Digital Enterprise Research Institute (DERI)
- Laura Dragan, Digital Enterprise Research Institute (DERI)
- Patrick Durusau, Individual
- Siegfried Handschuh, Digital Enterprise Research Institute (DERI)
- Deirdre Lee, Digital Enterprise Research Institute (DERI)
- Marc Pallot, ESoCE-NET
- Chancellor Pascale, Johns Hopkins University Applied Physics Laboratory
- Vassilios Peristeras, Digital Enterprise Research Institute (DERI)
- Peter Saint-Andre, Cisco Systems, Inc.
- Ramesh Vasudevan, Oracle Corporation
- Peter Yim, Individual
## Appendix B. Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Editor</th>
<th>Changes Made</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSPRD 01</td>
<td>March 16, 2011</td>
<td>Eric S. Chan Patrick Durusau</td>
<td>Committee Specification Draft for Public Review</td>
</tr>
<tr>
<td>CSPRD 02</td>
<td>November 8, 2011</td>
<td>Eric S. Chan Patrick Durusau</td>
<td>Changes in response to public review comments.</td>
</tr>
<tr>
<td>CSPRD 03</td>
<td>March 20, 2012</td>
<td>Eric S. Chan Patrick Durusau Laura Dragan</td>
<td>Changes in response to TC members review comments.</td>
</tr>
<tr>
<td>CSPRD 04</td>
<td>June 26, 2012</td>
<td>Ken Baclawski</td>
<td>Add 4 additional attributes from grammar to PropertyDefinition metadata model, corrected spelling of Cardinality, renamed the address property of Addressable to entityAddress to avoid clashing with the address properties of EntityAddress and Participant, and specified the omitted namespaces of the superCategories of some of the enumerations.</td>
</tr>
<tr>
<td>CSPRD 05</td>
<td>October 15, 2012</td>
<td>Ken Baclawski Eric S. Chan Patrick Durusau</td>
<td>Change InstantMessage isAbstract to false, change PropertyType to optional in PropertyDefinition, change cardinality of superCategory property in Category to multi, add ClassDefinition, StereoType, StereoTypeEnum in icom_meta, add Figure 18 ClassDefinition UML diagram, remove EntityDefinition in icom_core. Updated the conformance clauses in Section 5.</td>
</tr>
</tbody>
</table>