OpenICOM: A Framework for Integrated Collaboration Environments

Kenneth Baclawski[1], Eric Chan[2] Laura Dragan[3], Patrick Durusau[4], Deirdre Lee[3], and Peter Yim[5]

[1] Northeastern University, Boston, MA 02115, USA,
kenb@ccs.neu.edu,
WWW home page: http://www.ccs.neu.edu/home/kenb
[2] Oracle Corporation
[3] Digital Enterprise Research Institute, Galway, Ireland
[4] Individual
[5] Individual

Abstract. Collaboration is an important activity that is increasingly using technology to improve the productivity of the participants. The Integrated Collaboration Object Model (ICOM) is a proposed OASIS standard for interoperation among collaboration services. The ICOM is intended to be a framework for integrating a broad range of domain models for collaboration environments. The intention is to encourage independent software vendors and open source communities to create common collaboration clients that interoperate with integrated collaboration platforms and standalone collaboration services across enterprise boundaries. The article provides an overview of ICOM that covers the high-level concepts, directory, space, access control, metadata, content management, and unified message models. ICOM has been expressed in a number of formats, including the Java Persistence API, XML Schema, RDF and OWL. Some examples of applications based on ICOM will also be described.

Keywords: JPA, collaboration environments, Semantic Web

1 Introduction

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.

In Section 2, we define the Core model of ICOM. The Core defines the classes that bring together the model of directory management, identity management, and content management in a framework with a common access control model and metadata model. The extension modules in Section 3 extend the artifact and folder model of the Core model to define the specialized model for different collaboration activities. The range of collaboration models include 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. The 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. 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 [21]. The extension modules integrate models from Content Management Interoperability Services [14], Java Content Repository API [8], Web Distributed Authoring and Versioning (WebDAV) [6], Internet Message Access Protocol (IMAP) [1], Simple Mail Transfer Protocol (SMTP) [9], Extensible Messaging and Presence Protocol (XMPP) [18], XMPP Instant Messaging and Presence [19], vCard MIME Directory Profile [4], Internet Calendaring and Scheduling Core Object Specification (iCalendar) [5], and Calendaring Extensions to WebDAV (CalDAV) [3]. 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 [16], Web Services Business Process Execution Language [11], WS-BPEL Extension for People [13], and Web Services for Human Task [12]; in social networking domain, which includes Friend of a Friend [2], Semantically-Interlinked Online Communities [20], Open Social [17], and Facebook Platform Open Graph [7]. The OASIS ICOM TC Wiki [10] 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.

As with any standard, ICOM has an authoritative specification as well as informative specifications. The authoritative specification of ICOM uses the Content Management Interoperability Services (CMIS) [14] grammar to define classes and properties. The ICOM specification adapts the CMIS grammar to introduce mixed-in types, enumeration types, and other base types which are not part of the domain model of the CMIS Version 1 specification. The informative specifications were derived from the authoritative specification as follows:

Language	**Basis for Specification**	**Section**
CMIS	Authoritative	1
Java (POJO classes)	CMIS	2, 3
UML	Java (POJO classes)	2, 3
JPA	CMIS	4
RDF	CMIS	6
OWL	CMIS	6
XML Schema	Java (POJO classes)	5

Note that the UML and XML Schema are derived from the Java classes, while the other informative specifications are directly derived from the authoritative specification.

2 Core Model

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. All objects in the ICOM framework must be instances of at least one class. The class and property definitions correspond to the UML meta-model, which is an OMG Meta Object Facility (MOF) M2-model. The UML diagrams for ICOM were generated from plain Java classes (usually called POJO classes) which where directly translated from the authoritative ICOM specification. Some of these diagrams are shown below. The full set of UML class diagrams is in [15].

The ICOM Core model has five branches in its class hierarchy: Scope, Subject, Artifact, Metadata and Access. The Scope branch includes the model of communities and spaces which are containers of subjects and artifacts. This branch is concerned with directory (also called folder) management. The Subject branch includes the model of actors, groups, and roles. This branch is concerned with the identity of actors in collaboration. The Artifact branch includes the model of content produced by actors. The Metadata branch is concerned with metadata annotations. 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 the 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.

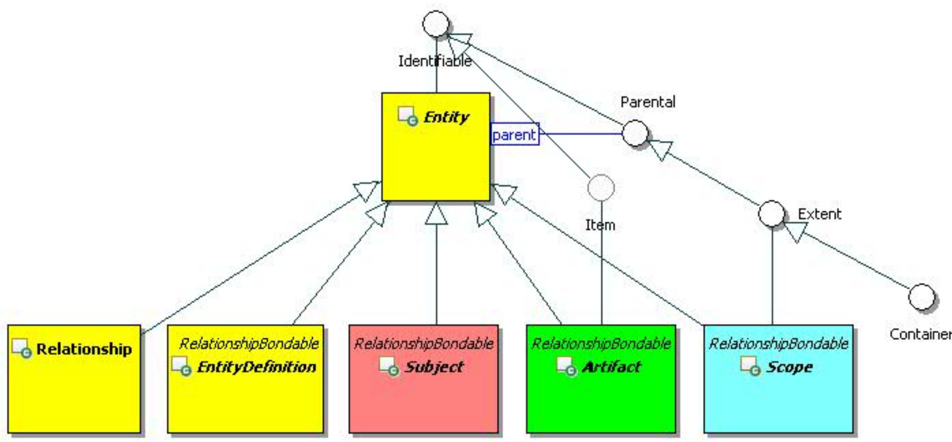

Fig. 1. ICOM Top-Level Classes

Figure 1 depicts the 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 the ICOM class hierarchy. To deal with the fact that major programming languages, such as Java, do not support multiple inheritance, it is necessary to separate classes into two kinds: “mixin” and ordinary. The mixin classes are represented in Java as interfaces rather than classes. In Java a class can be a subclass of at most one immediate superclass, but it may implement any number of interfaces. The mixin classes are shown as circles rather than rectangles in the class diagram.

The UML diagram in Figure 2 depicts the core classes in the Scope, Subject, and Artifact branches of ICOM class hierarchy. This figure only shows the subclass relationships, not the attributes or the associations.

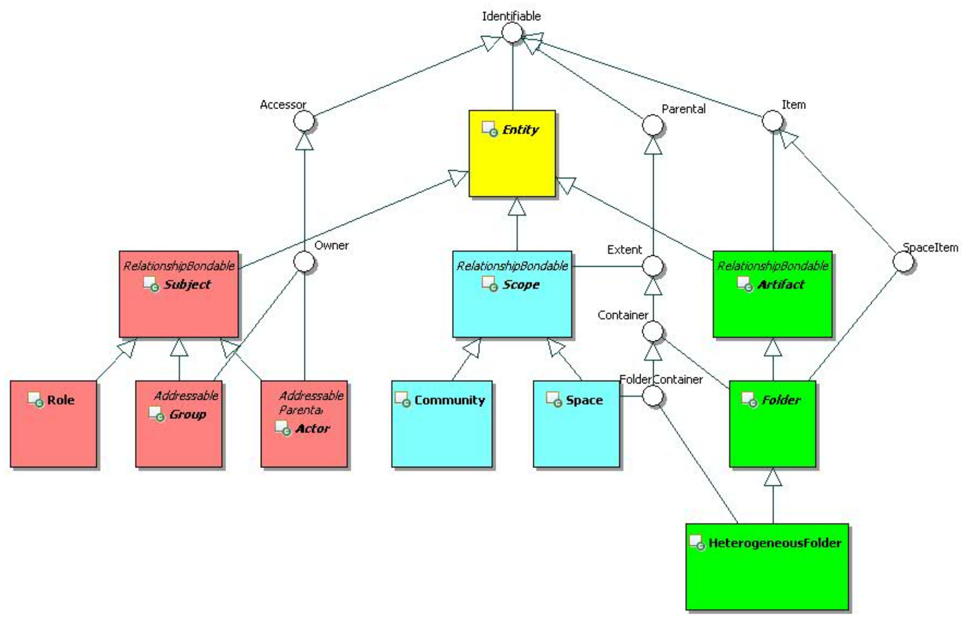

Fig. 2. The Main Branches of the ICOM Core Model

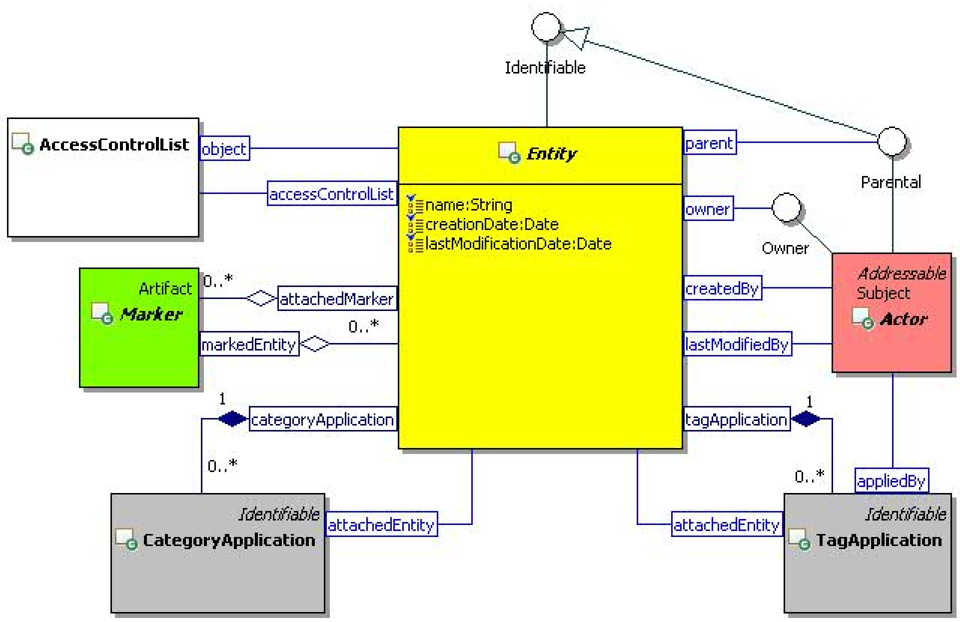

Fig. 3. ICOM Entity Classes

The UML diagram in Figure 3 depicts the entity class in more detail, showing both its attributes and the associations with other classes.

3 Extensions

Each ICOM 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. In addition to the extension modules described here, the ICOM framework allows additional extension modules. For example, applications can adopt a model for the 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, a social network model may be imported from [7] or [17].

ICOM defines containers that provide contexts and structures for specific areas of collaborative activities. For example, 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 defines Content, MultiContent, and SimpleContent. A content represents a piece of data in a document or message. Content, multicontent, simple content, and online content form a composite design pattern.
2. Document Module defines Document, WikiPage, and version control model. A document can contain a composite content. Documents are typically contained by heterogeneous folders.
3. Message Module defines Message, UnifiedMessage, InstantMessage, and related classes. A message can contain a composite content. Unified messages are typically contained by heterogeneous folders.
4. Presence Module 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. 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 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 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 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. 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 defines TaskList and Task. Tasks are used to coordinate the assignment of tasks and to track the progress of task activities.
9. Forum Module 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 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 Persistence

Maintaining data persistently is a necessity for nearly all software applications. Since the predominant storage technology is the relational model but the predominant software languages are object-oriented, it is necessary to have an object-relational mapping. An increasing number of applications are using the Java Persistence API (JPA) for specifying their object-relational mapping. Accordingly, one of the first mappings for ICOM was a mapping to JPA. This mapping consists of Java annotations added to the POJO classes.

The ICOM specification defines a class called Entity which is the superclass of any class that supports a persistent identifier, a change token for optimistic locking, and an access control list. The object identifier and change token are annotated, respectively, by javax.persistence.Id and javax.persistence.Version, matching the ICOM concept of Entity with the JPA concept of Entity. ICOM Entity has another fundamental dimension for access control list, which together with JPA Id and Version, defines a unit of persistent information for concurrency and access control. The generation of object identifiers is implementation dependent; however, ICOM recommends that the object identifiers should be globally unique to support permanent references to the entities that may migrate amongst interoperable ICOM repositories. An object identifier is read-only (immutable) once it is assigned and should never be duplicated or re-used for more than one object. The UML diagram in Figure 3 depicts the Entity class, properties, and cardinality of the properties. Entity's properties include name, created by, creation date, last modified by, last modification date, owner, parent, attached markers, category applications, tag applications, and access control list.

5 Interoperability

Since one of the purposes of ICOM is seamless interoperability between different collaboration environments, it is necessary for there to be a interchange mechanism between such environments. Since these environments may use different programming languages, the interchange format must be language-independent. The most commonly-used interchange format today is XML, and ICOM has been expressed in terms of XML Schema. This allows one to exchange data via SOAP/REST web services. The XML Schema specification was derived from the POJO classes using the JAXB schemagen processor.

6 Semantic Web

ICOM can further accelerate the removal of the walls between the collaboration tools and also expose the data from behind the wall of applications. Exposing the data in machine readable form is what Semantic Web is about. OASIS ICOM TC includes representations from Digital Enterprise Research Institute (DERI) and Ontolog Forum, whose focus areas are in semantics technologies. ICOM ontology is defined from the outset for concomitant representation in UML and RDF. ICOM TC wiki page discusses the mappings between UML and RDF representations. ICOM can bridge the object-oriented software engineering world with the semantic web world by providing bi-directional transformations between UML and RDF. Linked Data Community is advancing the use of web, URI, and RDF to connect distributed data. There is a popular saying "a little semantics goes a long way" about enriching the data with inference capability. ICOM data with a seamless programming model like JPA and a concomitant RDF representation will lower the barrier for applying inference engines. Figuratively speaking, a rich vocabulary of "nouns" in ICOM makes up for the strong "verbs" in service interfaces. A well-defined set of classes of ICOM makes the API amenable for rule-based applications and declarative inference. ICOM containers are active or reactive entities, for example conference and chat rooms are highly active while outbox, calendar, and task list are reactive. Their behavior can be augmented by applications.

The RDF and OWL specifications consists of two parts: TBox and ABox. The TBox part represents the specifications of RDF/OWL classes and properties. This part of the specification allows one to perform conceptual querying and reasoning about ICOM. This is especially important when ICOM to check consistency when ICOM is used with other ontologies. The ABox part represents the metadata for ICOM. This allows one to perform querying and reasoning on the RDF/OWL classes and properties as instances. This is important because RDF and OWL reasoners cannot perform reasoning on classes and properties beyond the built-in axioms for them. Instance reasoning, on the other hand, is much more flexible and richer.

The RDF and OWL specifications for ICOM could have been derived from the UML diagrams by using the Object Definition Metamodel (ODM). However, a direct translation from the authoritative specification was employed. This was done so that the metadata for ICOM would be available for querying and reasoning.

7 Conclusion and Future Work

We have described the ICOM framework for Interoperable Collaboration Services. Like most standards, the ICOM is specified in an authoritative document using a specification language (CMIS) that is designed to be easily readable by humans. However, ICOM is intended to be mainly used as an embedded framework within a collaboration environment for such tasks as storing data, sending data to other collaboration services as well as interacting with human collaborators. As a result, the ICOM framework has been translated to a number of other languages to facilitate these many purposes. Inasmuch as many other kinds of applications have these same requirements, it would be useful for their standards to be translated to other languages as well. Accordingly, we plan to use our translation software with other standards.

References

1. S. Bradner. Key words for use in RFCs to indicate requirement levels BCP 14, RFC 2119, March 1997. `http://www.ietf.org/rfc/rfc2119.txt`.
2. D. Brickley and L. Miller. FOAF vocabulary specification, August 2009. `http://xmlns.com/foaf/spec/`.
3. C. Daboo and B. Desruisseaux. Calendaring Extensions to WebDAV (CalDAV) RFC 4791, March 2007. `http://tools.ietf.org/html/rfc4791`.
4. F. Dawson and T. Howes. vCard MIME Directory Profile RFC 2426, September 1998. `http://tools.ietf.org/html/rfc2426`.
5. B. Desruisseaux. Internet Calendaring and Scheduling Core Object Specification (iCalendar) RFC 5545, September 2009. `http://tools.ietf.org/html/rfc5545`.
6. L. Dusseault. HTTP extensions for Web Distributed Authoring and Versioning (WebDAV) RFC 4918, June 2007. `http://tools.ietf.org/html/rfc4918`.
7. Facebook Platform Open Graph Core Concepts. `http://developers.facebook.com/docs/coreconcepts/`.
8. Java Specification Request (JSR) 283, content repository for Java^{TM} Technology API 2.0 Specification, August 2009. `http://jcp.org/en/jsr/detail?id=283`.
9. J. Klensin. Simple Mail Transfer Protocol, Draft Standard RFC 5321, October 2008. `http://tools.ietf.org/html/rfc5321`.

10. OASIS ICOM TC Wiki. http://wiki.oasis-open.org/icom.
11. OASIS Standard, Web Services Business Process Execution Language Version 2.0, April 2007. http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html.
12. OASIS Committee Specification, Web Services Human Task (WS-HumanTask) Specification Version 1.1 CS-01, August 2010. http://docs.oasis-open.org/bpel4people/ws-humantask-1.1-spec-cs-01.html.
13. OASIS Committee Specification, WS-BPEL extension for people (BPEL4People) Specification Version 1.1, August 2010. http://docs.oasis-open.org/bpel4people/bpel4people-1.1.html.
14. OASIS standard, content management interoperability services (CMIS) Version 1.0, May 2010. http://docs.oasis-open.org/cmis/CMIS/v1.0/os/cmis-spec-v1.0.doc.
15. OASIS Integrated Collaboration Object Model (ICOM) for Interoperable Collaboration Services Version 1.0, August 2012. http://docs.oasis-open.org/icom/icom-ics/v1.0/csprd03/icom-ics-v1.0-csprd03.doc.
16. Business Process Model and Notation (BPMN) Version 2.0, January 2011. http://www.omg.org/spec/BPMN/2.0/PDF.
17. OpenSocial and gadgets specification group, "Social Data Specification", November 2010. http://opensocial-resources.googlecode.com/svn/spec/2.0/Social-Data.xml.
18. P. Saint-Andre. Extensible Messaging and Presence Protocol (XMPP): Core RFC 3920, October 2004. http://tools.ietf.org/html/rfc3920.
19. P. Saint-Andre. Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence RFC 3921, October 2004. http://tools.ietf.org/html/rfc3921.
20. W3C member submission, "SIOC Core Ontology Specification", June 2007. http://www.w3.org/Submission/2007/SUBM-sioc-spec-20070612/.
21. K. Zeilenga. Lightweight Directory Access Protocol (LDAP): Directory information models RFC 4512, June 2006. http://tools.ietf.org/html/rfc4512.