

TCVN

TIÊU CHUẨN QUỐC GIA

TCVN ISO/TS 15000-3 : 2007

ISO/TS 15000-3 : 2004

Xuất bản lần 1

**NGÔN NGỮ ĐÁNH DẤU MỞ RỘNG KINH DOANH ĐIỆN TỬ (ebXML)
PHẦN 3: QUY ĐỊNH KỸ THUẬT VỀ MÔ HÌNH
THÔNG TIN ĐĂNG KÝ (ebRIM)**

*Electronic business eXtensible Markup Language (ebXML)
Part 3: Registry information model specification (ebRIM)*

HÀ NỘI – 2007

Mục lục	Trang
1 Phạm vi áp dụng	7
2 Ban kỹ thuật về đăng ký ebXML của OASIS/ebXML.....	7
2.1 Người đóng góp	8
3 Giới thiệu	8
3.1 Tóm tắt nội dung	8
3.2 Quy ước chung.....	8
3.3 Người đọc	9
3.4 Tài liệu liên quan.....	9
4 Mục tiêu thiết kế	9
4.1 Mục đích	9
5 Tổng quan hệ thống	10
5.1 Vai trò của sổ đăng ký ebXML	10
5.2 Dịch vụ đăng ký	10
5.3 Việc thực hiện của mô hình thông tin đăng ký	10
5.4 Cách làm việc của mô hình thông tin đăng ký	10
5.5 Mô hình thông tin đăng ký có thể được thực thi	10
5.6 Phù hợp với một sổ đăng ký ebXML.....	10
6 Mô hình thông tin đăng ký: Mô tả chung mức cao	11
6.1 RegistryObject (Đối tượng đăng ký)	12
6.2 Slot (Khe bổ sung)	12
6.3 Association (Liên kết).....	12
6.4 ExternalIdentifier (Định danh ngoài).....	12
6.5 ExternalLink (Liên kết ngoài).....	12
6.6 ClassificationScheme (Giản đồ phân loại).....	13
6.7 ClassificationNode (Nút phân loại).....	13
6.8 Classification (Phân loại).....	13
6.9 RegistryPackage (Gói đăng ký).....	13
6.10 AuditableEvent (Sự kiện kiểm tra)	13
6.11 User (Người sử dụng).....	13
6.12 PostalAddress (Địa chỉ bưu điện)	14
6.13 EmailAddress (Địa chỉ thư điện tử)	14
6.14 Organization (Tổ chức).....	14
6.15 Service (Dịch vụ).....	14
6.16 ServiceBinding (Quy định dịch vụ)	14
6.17 SpecificationLink (Liên kết quy định kỹ thuật).....	14

7	Mô hình thông tin đăng ký: Chi tiết.....	14
7.1	Thuộc tính và phương pháp của các lớp mô hình thông tin	16
7.2	Các kiểu dữ liệu.....	16
7.3	Hỗ trợ quốc tế (I18N).....	16
7.4	Lớp RegistryObject (Đối tượng đăng ký).....	17
7.5	Lớp RegistryEntry (Lớp mục nhập đăng ký).....	20
7.6	Lớp Slot (Khe bổ sung).....	23
7.7	Lớp ExtrinsicObject (Đối tượng ngoại lai).....	24
7.8	Lớp RegistryPackage	25
7.9	Lớp ExternalIdentifier (Định danh ngoài)	25
7.10	Lớp ExternalLink (Liên kết ngoài).....	26
8	Vết kiểm tra sổ đăng ký	27
8.1	Lớp AuditableEvent (Sự kiện có thể kiểm tra).....	27
8.2	Lớp User (Người sử dụng)	29
8.3	Lớp Organization (Tổ chức).....	30
8.4	Lớp PostalAddress (Địa chỉ bưu điện)	31
8.5	Lớp telephoneNumbers (Số điện thoại)	32
8.6	Lớp EmailAddress (Địa chỉ hòm thư điện tử).....	33
8.7	Lớp personName (Tên cá nhân).....	33
8.8	Lớp Service (Dịch vụ).....	34
8.9	Lớp Service (dịch vụ) Bindings.....	34
8.10	Lớp SpecificationLinks (Liên kết quy định).....	35
9	Phương thức liên kết của RegistryObject (Đối tượng đăng ký)	36
9.1	Ví dụ về Association (Liên kết)	36
9.2	Đối tượng Nguồn và đối tượng Đích.....	37
9.3	Các kiểu liên kết.....	37
9.4	Liên kết trong (Intramural).....	37
9.5	ExternalLink (Liên kết ngoài).....	38
9.6	Xác nhận một liên kết.....	38
9.7	Khả năng hiển thị các liên kết không được xác nhận	39
9.8	Các trạng thái xác nhận có thể xảy ra	39
9.9	Lớp Association (Liên kết).....	39
10	Phân loại RegistryObject (Đối tượng đăng ký)	41
10.1	Lớp classificationScheme (Giản đồ phân loại).....	43
10.2	Lớp classificationNode (Nút phân loại)	44
10.3	Phân loại lớp	46
10.4	Ví dụ về biểu mẫu phân loại	49

11	Mô hình thông tin: Quan niệm về bảo mật	50
11.1	Lớp AccessControlPolicy (Chính sách kiểm soát truy cập)	50
11.2	Lớp Permission (Cho phép)	51
11.3	Lớp Privilege (Đặc quyền)	51
11.4	Lớp PrivilegeAttribute (Thuộc tính đặc quyền)	52
11.5	Lớp Role (Vai trò)	52
11.6	Lớp Group (Nhóm)	52
11.7	Lớp Identity (Đặc tính)	52
11.8	Lớp Principal (Thực thể)	53
	Tài liệu tham khảo	54

Lời nói đầu

TCVN ISO/TS 15000-3 : 2007 hoàn toàn tương đương với tiêu chuẩn **ISO/TS 15000-3 : 2004**.

TCVN ISO/TS 15000-3 : 2007 do Ban kỹ thuật tiêu chuẩn TCVN/TC 154 "*Quá trình, các yếu tố dữ liệu và tài liệu trong thương mại, công nghiệp và hành chính*" biên soạn, Tổng cục Tiêu chuẩn Đo lường Chất lượng đề nghị, Bộ Khoa học và Công nghệ công bố.

Ngôn ngữ đánh dấu mở rộng kinh doanh điện tử (ebXML)

Phần 3: Quy định kỹ thuật về mô hình thông tin đăng ký

Electronic business eXtensible Markup Language (ebXML)

Part 3: Registry information model specification (ebRIM)

1 Phạm vi áp dụng

Phiên bản hiện tại của quy định kỹ thuật về mô hình thông tin đăng ký của OASIS:

<http://www.oasis-open.org/committees/regrep/documents/2.0/specs/ebRIM.pdf>

Phiên bản mới nhất của quy định kỹ thuật về mô hình thông tin đăng ký của OASIS:

<http://www.oasis-open.org/committees/regrep/documents/2.0/specs/ebRIM.pdf>

2 Ban kỹ thuật về đăng ký ebXML của OASIS/ebXML

Tiêu chuẩn này đã được chấp nhận như một tiêu chuẩn OASIS, dưới dạng trình bày hiện tại, đã được chấp nhận là quy định kỹ thuật của Ban Kỹ thuật về Đăng ký ebXML của OASIS. Tiêu chuẩn này dựa trên cơ sở Dự thảo quy định phiên bản 1.0 của Ban Kỹ thuật về Đăng ký ebXML của OASIS.

Ban Kỹ thuật về Đăng ký ebXML của OASIS xây dựng phiên bản 2.0 hiện tại bao gồm:

Kathryn Breininger, Boeing

Lisa Carnahan, US NIST (TC Chair)

Joseph M. Chiusano, LMI

Suresh Damodaran, Sterling Commerce

Mike DeNicola Fujitsu

Anne Fischer, Drummond Group

Sally Fuger, AIAG

Jong Kim InnoDigital

Kyu-Chul Lee, Chungnam National University

Joel Munter, Intel

Farrukh Najmi, Sun Microsystems

Joel Neu, Vitria Technologies

Sanjay Patil, IONA

Neal Smith, ChevronTexaco

Nikola Stojanovic, Encoda Systems, Inc.

Prasad Yendluri, webMethods

Yutaka Yoshida, Sun Microsystems

2.1 Người đóng góp

Những cá nhân sau đã tích cực tham gia đóng góp vào nội dung của tiêu chuẩn này, nhưng không phải là thành viên đã bỏ phiếu của Ban Kỹ thuật về Đăng ký ebXML của OASIS:

Len Gallagher, NIST

Sekhar Vajjhala, Sun Microsystems

3 Giới thiệu

3.1 Tóm tắt nội dung

Tiêu chuẩn này quy định kỹ thuật về mô hình thông tin đăng ký ebXML.

Tài liệu khác quy định dịch vụ đăng ký ebXML (ebRS) mô tả cách thức xây dựng các dịch vụ đăng ký để truy cập nội dung thông tin trong sổ đăng ký ebXML.

3.2 Quy ước chung

Các quy ước được sử dụng trong tiêu chuẩn này như sau:

Giản đồ *UML* mô tả ngắn gọn các khái niệm, không thể hiện các yêu cầu phương pháp luận cũng như *sự thực thi* cụ thể.

Thuật ngữ "*hạng mục kho*" đề cập đến một đối tượng lưu giữ an toàn (ví dụ: Một tài liệu XML hoặc DTT). Tất cả các *hạng mục kho* này được trình bày trong sổ đăng ký bởi *trường hợp* RegistryObject (đối tượng đăng ký).

Thuật ngữ "mục nhập đăng ký" đề cập đến một đối tượng cung cấp siêu dữ liệu về một *hạng mục kho*.

Mô hình thông tin không xem xét đến nội dung cụ thể của kho. Tất cả *phần tử* của mô hình thông tin thể hiện siêu dữ liệu về nội dung và không chứa nội dung.

Các từ viết *hoa nghiêng* được liệt kê trong bảng chú giải thuật ngữ ebXML.

Các từ khóa: MUST (PHẢI), MUST NOT (KHÔNG PHẢI), REQUIRED (YÊU CẦU), SHALL (PHẢI), SHALL NOT (KHÔNG PHẢI), SHOULD (NÊN), SHOULD NOT (KHÔNG NÊN), RECOMMENDED (KHUYẾN CÁO), MAY (CÓ THỂ) và OPTIONAL (LỰA CHỌN), trong tiêu chuẩn này được định nghĩa và mô tả trong RFC 2119 [Bra97].

Người ứng dụng phần mềm CÓ THỂ sử dụng tiêu chuẩn này kết hợp với các tài liệu quy định ebXML khác khi xây dựng phần mềm ebXML.

3.2.1 Quy định về đặt tên

Để thống nhất về chữ hoa và quy định đặt tên trong tiêu chuẩn này, các kiểu "chữ hoa" - *UCC* và "chữ thường" - *LCC* được quy định như sau:

- tên phần tử sử dụng *UCC* (ví dụ: <UpperCamelCaseElement/>);
- tên thuộc tính sử dụng *LCC* (ví dụ: <LowerCamelCaseAttribute = "whatEver">);
- tên lớp, giao diện sử dụng *UCC* (ví dụ: ClassificationNode, Versionable);
- tên phương pháp sử dụng *LCC* (ví dụ: getName(), setName()).

Các từ *hoa nghiêng* được quy định trong bảng chú giải thuật ngữ [ebGLOSS].

3.3 Người đọc

Đối tượng người đọc tiêu chuẩn này là người phát triển phần mềm:

- người thực thi dịch vụ đăng ký ebXML;
- người thực thi máy client đăng ký ebXML.

3.4 Tài liệu liên quan

Các tài liệu sau cung cấp cho người đọc một số cơ sở và thông tin liên quan:

- a. quy định dịch vụ đăng ký ebXML (ebRS) - Chỉ rõ các dịch vụ đăng ký thực dựa trên mô hình thông tin trong tiêu chuẩn này;
- b. quy định thoả thuận và giao thức cộng tác ebXML [ebCPP] - Chỉ rõ các cách thức hồ sơ có thể được xác định đối với một bên tham gia và cách thức các hồ sơ của hai bên tham gia có thể được xác định thành một thoả thuận hợp tác.

4 Mục tiêu thiết kế

4.1 Mục đích

Mục đích của tiêu chuẩn này:

- truyền đạt thông tin trong sổ đăng ký và cách thức thông tin được tổ chức;
- thúc đẩy công việc được thực thi trong OASIS [OAS] và ISO 11170 [ISSO] - Các mô hình đăng ký;
- hài hòa với các công việc liên quan trong các nhóm làm việc ebXML khác;
- có khả năng phát triển để hỗ trợ các yêu cầu đăng ký ebXML trong tương lai;
- tương thích với các quy định về ebXML khác.

5 Tổng quan hệ thống

5.1 Vai trò của sổ đăng ký ebXML

Sổ đăng ký cung cấp một kho lưu trữ ổn định cho thông tin do một *tổ chức đệ trình*. Thông tin như vậy được sử dụng để tạo điều kiện thuận lợi cho các bên tham gia thương mại và giao dịch thương mại dựa trên ebXML (B2B). Nội dung đệ trình có thể là giản đồ và tài liệu XML, mô tả quá trình, *thành phần* ebXML chính, mô tả ngữ cảnh, UMLModel, thông tin về các bên tham gia, thậm chí cả thành phần của phần mềm.

5.2 Dịch vụ đăng ký

Một bộ các dịch vụ đăng ký để truy cập vào nội dung *sổ đăng ký* tới các máy client của *sổ đăng ký* được thể hiện trong quy định dịch vụ đăng ký ebXML [ebRS]. Tiêu chuẩn này không cung cấp chi tiết về các dịch vụ nhưng đôi khi có thể đề cập đến nó.

5.3 Việc thực hiện của mô hình thông tin đăng ký

Mô hình thông tin đăng ký cung cấp giản đồ thiết kế hoặc bậc cao cho *sổ đăng ký* ebXML. Lợi ích chủ yếu của mô hình thông tin đăng ký dành cho người thực thi các *sổ đăng ký* ebXML. Nó cung cấp cho người thực thi này các thông tin về kiểu siêu dữ liệu được lưu trong *sổ đăng ký* cũng như mối liên hệ giữa các *lớp* siêu dữ liệu.

Mô hình thông tin đăng ký:

- xác định kiểu đối tượng được lưu trong *sổ đăng ký*;
- xác định cách tổ chức của các đối tượng đã lưu trong *sổ đăng ký*.

5.4 Cách làm việc của mô hình thông tin đăng ký

Người thực thi *sổ đăng ký* ebXML CÓ THỂ sử dụng mô hình thông tin đăng ký để xác định các *lớp* bao gồm trong *việc thực thi sổ đăng ký* và thuộc tính, phương pháp các *lớp* này có thể có. Người thực thi cũng CÓ THỂ sử dụng mô hình thông tin đăng ký để xác định loại giản đồ cơ sở dữ liệu cần thiết.

CHÚ THÍCH: Mô hình thông tin là để minh họa và không mô tả bất cứ sự lựa chọn thực thi cụ thể nào.

5.5 Mô hình thông tin đăng ký có thể được thực thi

Mô hình thông tin đăng ký CÓ THỂ được thực thi trong một *sổ đăng ký ebXML* dưới dạng một giản đồ cơ sở dữ liệu quan hệ, giản đồ cơ sở dữ liệu đối tượng hoặc một số giản đồ vật lý khác. Nó cũng có thể được thực thi như các *lớp* và giao diện thuộc *việc thực thi sổ đăng ký*.

5.6 Phù hợp với một sổ đăng ký ebXML

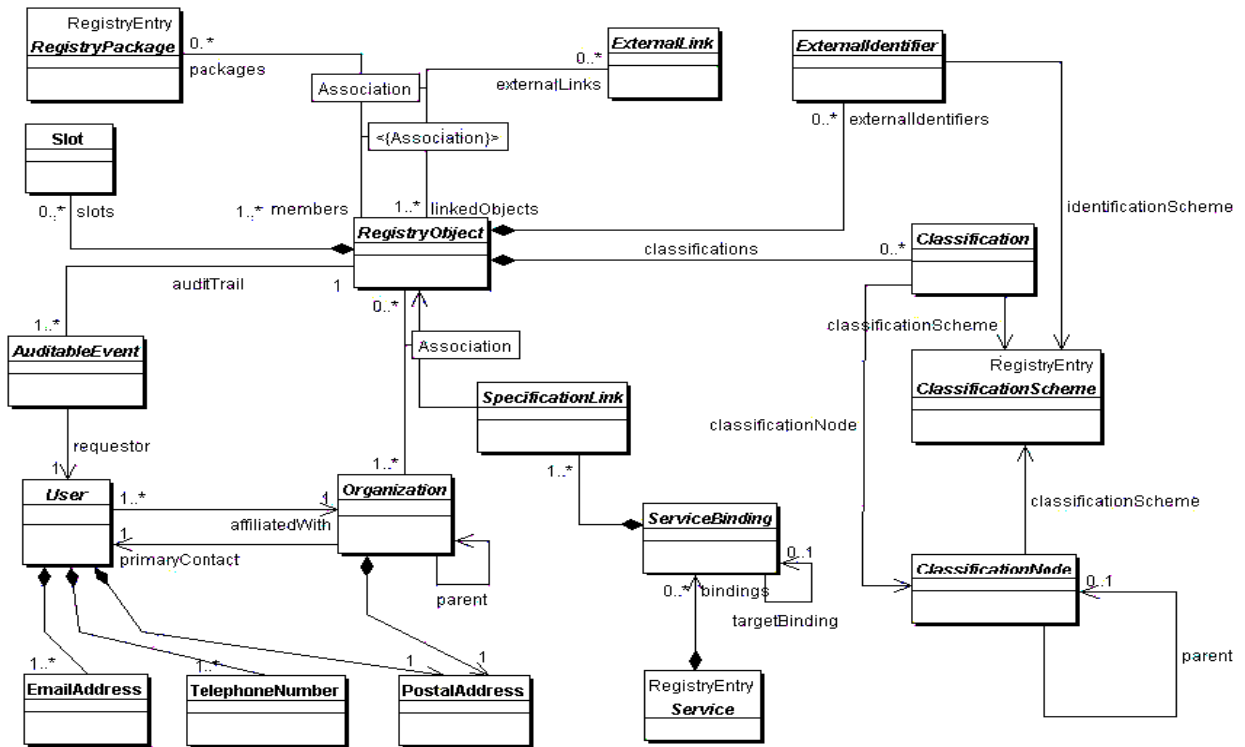
Nếu một *phần tử* yêu cầu sự phù hợp với quy định của tiêu chuẩn này thì nó sẽ hỗ trợ tất cả các giao diện và *lớp* mô hình thông tin được yêu cầu, thuộc tính của chúng và xác định định nghĩa của chúng là cụ thể thông qua các dịch vụ đăng ký ebXML.

6 Mô hình thông tin đăng ký: Mô tả chung mức cao

Phần này đưa ra một quan điểm chung mức cao của các đối tượng cụ thể nhất trong *sổ đăng ký*.

Sơ đồ 1: Thể hiện quan điểm chung mức cao trong *sổ đăng ký* và các mối quan hệ của chúng như sơ đồ lớp UML. Nó không chỉ rõ các thuộc tính hoặc phương pháp lớp, kế thừa.

Lưu ý mô hình thông tin không mô phỏng các *hạng mục kho* thực.



Hình 1 – Mô tả chung mức cao của mô hình thông tin

CHÚ GIẢI:

- RegistryEntry: Mục nhập đăng ký
- RegistryPackage: Gói đăng ký
- Slot: Khe bổ sung
- Package: Gói
- Association: Sự liên kết
- externalLink: Liên kết ngoài
- ExternalIdentifier: Định danh ngoài
- Member: Thành viên
- linkedObject: Đối tượng được liên kết
- Classification: Phân loại
- ClassificationScheme: Giản đồ phân loại
- AudibleEvent: Sự kiện có thể kiểm tra
- User: Người sử dụng
- auditTrail: Vết kiểm tra
- identificationScheme: Giản đồ định danh
- ClassificationNode: Nút phân loại
- targetBinding: Quy định đích
- Binding: Quy định
- Service: Dịch vụ
- Parent: Gốc
- EmailAddress: Địa chỉ thư điện tử

TCVN ISO/TS 15000-3 : 2007

TelephoneNumber:	Số điện thoại
PostalAddress:	Địa chỉ Bưu điện
affiliatedWith:	được nhập (liên kết) với
Organization:	Tổ chức
primaryContact:	Điểm liên hệ chính
requestor	Người yêu cầu

6.1 RegistryObject (Đối tượng đăng ký)

Lớp RegistryObject là một *lớp* trừu tượng cơ sở được sử dụng bởi phần lớn các *lớp* trong mô hình này. Nó cung cấp siêu dữ liệu tối thiểu cho các đối tượng đăng ký. Bên cạnh đó, nó cung cấp các phương pháp cho việc truy cập các đối tượng khác cũng có liên quan đến siêu dữ liệu động cho đối tượng đăng ký.

6.2 Slot (Khe bổ sung)

Trường hợp Slot (khe bổ sung) cung cấp cách thức động để bổ sung các thuộc tính tùy ý cho các *trường hợp* RegistryObject (đối tượng đăng ký). Khả năng này để bổ sung động các thuộc tính vào các *trường hợp* RegistryObject (đối tượng đăng ký) cho phép mở rộng trong mô hình thông tin đăng ký. Ví dụ: Nếu một công ty muốn bổ sung thuộc tính có nội dung là "quyền sao chép" vào mỗi *trường hợp* đối tượng được yêu cầu, họ có thể thực thi bằng cách tạo ra thêm một vị trí có tên là "quyền sao chép" và có giá trị bao gồm tuyên bố về quyền sao chép.

6.3 Association (Liên kết)

Trường hợp Association (liên kết) là các *trường hợp* RegistryObject (đối tượng đăng ký) được sử dụng để xác định càng nhiều các kết hợp giữa các đối tượng trong mô hình thông tin. Các Association này được mô tả chi tiết trong phần 9.

6.4 ExternalIdentifier (Định danh ngoài)

Trường hợp ExternalIdentifier (định danh ngoài) cung cấp thông tin định danh phụ đối với một *trường hợp* RegistryObject (đối tượng đăng ký) bao gồm số DUNS, số bảo mật chung, hoặc một tên bí danh của một tổ chức.

6.5 ExternalLink (Liên kết ngoài)

ExternalLink (liên kết ngoài) là *trường hợp* RegistryObject (đối tượng đăng ký) mô phỏng một URI đã được đặt tên cho một nội dung nhưng không được quản lý bởi *sổ đăng ký*. Không giống như nội dung được quản lý, nội dung ngoài như vậy có thể thay đổi hoặc bị xoá bất cứ khi nào, không cần kiến thức am hiểu về *sổ đăng ký*. Một trường hợp RegistryObject (đối tượng đăng ký) có thể được kết hợp với một số ExternalLink nào đó.

Trong *trường hợp* một *tổ chức* *đệ trình* đưa ra một *hạng mục kho* giữ (ví dụ: một DTD) và muốn kết hợp một vài nội dung ngoài vào đối tượng đó (ví dụ: trang Web chủ của *tổ chức* *đệ trình*). ExternalLink có khả năng thực thi việc này. Khả năng sử dụng của ExternalLink có thể có trong một công cụ GUI được hiển thị ExternalLink đối với một RegistryObject (đối tượng đăng ký). Người sử dụng đó có thể nhấn chuột vào một liên kết và truy cập vào được một trang Web bên ngoài do đã được tạo liên kết.

6.6 ClassificationScheme (Giản đồ phân loại)

ClassificationScheme (giản đồ phân loại) là các *trường hợp* RegistryEntry (mục nhập đăng ký) mô tả một cách đã được cấu trúc để phân loại hoặc chia loại các *trường hợp* RegistryObject (đối tượng đăng ký).

Cấu trúc của ClassificationScheme (giản đồ phân loại) có thể được xác định cho số đăng ký trong và ngoài, dẫn đến kết quả là sự phân biệt giữa các giản đồ phân loại trong và ngoài. Một ví dụ đơn giản của ClassificationScheme (giản đồ phân loại) trong khoa học là *việc phân loại các vấn đề sự sống* trong đó các các vấn đề về sự sống được phân chia theo cấu trúc hình cây. Một ví dụ khác là hệ thống thập phân Dewey sử dụng trong thư viện để phân loại giữa sách và tài liệu khác. ClassificationScheme (giản đồ phân loại) được mô tả chi tiết tại phần 10 của tiêu chuẩn này.

6.7 ClassificationNode (Nút phân loại)

Các *trường hợp* ClassificationNode (nút phân loại) là các *trường hợp* RegistryObject (đối tượng đăng ký) được sử dụng để xác định cấu trúc hình cây trong ClassificationScheme (giản đồ phân loại), trong đó mỗi nút trong hình cây là một ClassificationNode và là gốc của ClassificationScheme (giản đồ phân loại). Cây *phân loại* được xây dựng với các ClassificationNode sử dụng để xác định cấu trúc của các giản đồ phân loại hoặc bản thể học. ClassificationNode được mô tả chi tiết tại phần 10 của tiêu chuẩn này.

6.8 Classification (Phân loại)

Các *trường hợp* classification (phân loại) là các *trường hợp* RegistryObject (đối tượng đăng ký) sử dụng để phân loại với các *trường hợp* RegistryObject (đối tượng đăng ký) khác. Một *trường hợp* Classification chỉ ra một ClassificationScheme (giản đồ phân loại) và giá trị của sự phân loại được xác định trong ClassificationScheme (giản đồ phân loại) đó. Các Classification có thể là bên trong và bên ngoài, phụ thuộc vào bất kể ClassificationScheme (giản đồ phân loại) đã được tham chiếu trong và ngoài. Classification được mô tả chi tiết trong phần 10 tiêu chuẩn này.

6.9 RegistryPackage (Gói đăng ký)

Các *trường hợp* RegistryPackage (gói đăng ký) là các *trường hợp* RegistryObject (đối tượng đăng ký) để nhóm các *trường hợp* RegistryObject (đối tượng đăng ký) có cùng quan hệ logic.

6.10 AuditableEvent (Sự kiện kiểm tra)

Các *trường hợp* AuditableEvent (sự kiện kiểm tra) là các *trường hợp* RegistryObject (đối tượng đăng ký) sử dụng để cung cấp một vết kiểm tra đối với các *trường hợp* RegistryObject (đối tượng đăng ký). Sự kiện này được mô tả chi tiết trong phần 8 của tiêu chuẩn này.

6.11 User (Người sử dụng)

Các *trường hợp* User (người sử dụng) là các *trường hợp* RegistryObject (đối tượng đăng ký) sử dụng để cung cấp thông tin về người sử dụng đã được đăng ký trong *sổ đăng ký*. Đối tượng của người sử dụng

TCVN ISO/TS 15000-3 : 2007

được sử dụng trong quá trình kiểm soát đối với các *trường hợp* RegistryObject (đối tượng đăng ký). User (người sử dụng) được mô tả chi tiết trong phần 8 tiêu chuẩn này.

6.12 PostalAddress (Địa chỉ bưu điện)

PostalAddress (địa chỉ bưu điện) là một *lớp* thực đơn lẻ có thể tái sử dụng. Chúng xác định thuộc tính của một địa chỉ cổng.

6.13 EmailAddress (Địa chỉ thư điện tử)

EmailAddress (địa chỉ thư điện tử) là một *lớp thực thể* đơn giản có thể tái sử dụng. Chúng xác định thuộc tính của một địa chỉ thư điện tử.

6.14 Organization (Tổ chức)

Các *trường hợp* Organization (tổ chức) là các *trường hợp* RegistryObject (đối tượng đăng ký) cung cấp thông tin về tổ chức đó, ví dụ như một *tổ chức đệ trình*. Mỗi Organization có thể có một sự tham chiếu đến Organization mẹ.

6.15 Service (Dịch vụ)

Các *trường hợp* Service là các *trường hợp* RegistryObject (đối tượng đăng ký) cung cấp thông tin về các dịch vụ (ví dụ: Các dịch vụ về Web).

6.16 ServiceBinding (Quy định dịch vụ)

ServiceBinding (quy định dịch vụ) là các *trường hợp* RegistryObject (đối tượng đăng ký) thể hiện thông tin kỹ thuật về cách thức cụ thể để truy cập một giao diện cụ thể được cung cấp bởi một dịch vụ. Một dịch vụ có một tập hợp các kết nối.

6.17 SpecificationLink (Liên kết quy định kỹ thuật)

Một SpecificationLink cung cấp sự liên kết giữa một ServiceBinding với một trong các quy định kỹ thuật của nó. Nó mô tả làm thế nào để sử dụng dịch vụ với kết nối của dịch vụ đó. Ví dụ: Một ServiceBinding có thể có một SpecificationLink mô tả cách làm thế nào để truy cập vào dịch vụ đó bằng cách sử dụng một quy định kỹ thuật trong tài liệu WSDL hoặc tài liệu CORBA IDL.

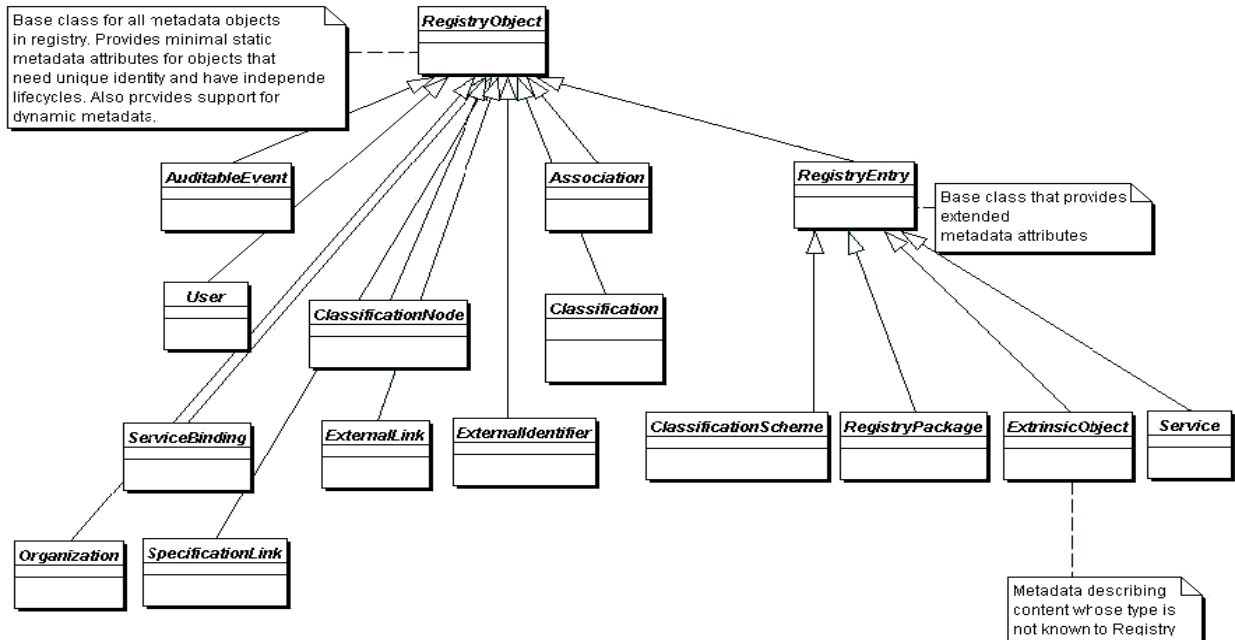
7 Mô hình thông tin đăng ký: Chi tiết

Phần này mô tả chi tiết về các *lớp* mô hình thông tin, giới thiệu một số *lớp* bổ sung trong mô hình không được đề cập trong nội dung về mô hình thông tin đã trình bày ở trên.

Hình 2 cho thấy các *lớp* kế thừa hoặc là các mối quan hệ giữa các *lớp* trong mô hình thông tin. Chú ý rằng nó không chỉ rõ các kiểu quan hệ khác như một tổng hợp các quan hệ được chỉ ra trong Hình 1. Hình 2 cũng không chỉ rõ thuộc tính của các *lớp* và các phương pháp *lớp*. Mô tả chi tiết các phương pháp và thuộc tính của phần lớn các giao diện và các *lớp* sẽ được thể hiện trong bảng biểu và mô tả mỗi *lớp* trong kiểu.

Lớp Association sẽ được đề cập chi tiết trong phần 9. Các Lớp classificationScheme (giản đồ phân loại), Classification và ClassificationNode sẽ được đề cập chi tiết trong phần 10.

Như đã đề cập, người đọc lưu ý là mô hình thông tin này chỉ để tham khảo, không phải là mẫu cho các mục lưu trữ thực.



Hình 2 – Mô tả mô hình thông tin

CHÚ GIẢI:

- RegistryEntry: Mục nhập đăng ký
- RegistryPackage: Gói đăng ký
- Slot: Khe bổ sung
- Package: Gói
- Association: Sự liên kết
- externalLink: Liên kết ngoài
- ExternalIdentifier: Định danh ngoài
- Member: Thành viên
- linkedObject: Đối tượng được liên kết
- Classification: Phân loại
- ClassificationScheme: Giản đồ phân loại
- AudibleEvent: Sự kiện có thể kiểm tra
- User: Người sử dụng
- auditTrail: Vết kiểm tra
- identificationScheme: Giản đồ định danh
- ClassificationNode: Nút phân loại
- targetBinding: Quy định đích
- Binding: Quy định
- Service: Dịch vụ
- Parent: Gốc
- EmailAddress: Địa chỉ thư điện tử
- TelephoneNumber: Số điện thoại
- PostalAddress: Địa chỉ Bưu điện
- affiliatedWith: được nhập (liên kết) với
- Organization: Tổ chức
- primaryContact: Điểm liên hệ chính
- requestor: Người yêu cầu

7.1 Thuộc tính và phương pháp của các lớp mô hình thông tin

Các lớp mô hình thông tin được xác định chủ yếu theo các thuộc tính mà nó thực thi. Các thuộc tính này cung cấp thông tin minh họa cho các lớp. Thực thi một đăng ký thường được gắn các thuộc tính của lớp với các thuộc tính trong một kho lưu trữ XML hoặc các cột trong kho lưu trữ liên kết.

Các lớp mô hình thông tin cũng có thể có nhiều phương pháp được xác định cho chúng. Các phương pháp này cung cấp hành vi bổ sung đối với các lớp mà các phương pháp được xác định trong đó. Các phương pháp này được sử dụng để ánh xạ tới các khả năng truy vấn SQL và truy vấn lọc được xác định trong [ebRS].

Khi mô hình hỗ trợ quá trình thông báo giữa các lớp, thường là đối với trường hợp một lớp trong kiểu đó thừa hưởng các thuộc tính và phương pháp từ các lớp cơ sở, bổ sung xác định các thuộc tính và phương pháp đặc biệt của nó.

7.2 Các kiểu dữ liệu

Bảng dưới liệt kê các kiểu dữ liệu khác nhau được sử dụng bởi các thuộc tính trong lớp mô hình thông tin:

Kiểu dữ liệu	Kiểu dữ liệu giản đồ XML	Mô tả	Độ dài
Boolean	boolean	Được sử dụng cho một giá trị đúng hoặc sai	
String 4	String	Được sử dụng cho chuỗi dài 4 ký tự	4 ký tự
String 8	string	Được sử dụng cho chuỗi dài 8 ký tự	8 ký tự
String 16	String	Được sử dụng cho chuỗi dài 16 ký tự	16 ký tự
String 32	String	Được sử dụng cho chuỗi dài 32 ký tự	32 ký tự
ShortName	String	Một chuỗi văn bản ngắn	64 ký tự
LongName	String	Một chuỗi văn bản dài	128 ký tự
FreeFormText	String	Một chuỗi văn bản dài cho biểu để trống	256 ký tự
UUID	String	Các chỉ danh duy nhất tổng thể DCE 128 bit sử dụng cho tham chiếu đối tượng khác	64 ký tự
URI	String	Được sử dụng cho các giá trị URL và URN	256 ký tự
Integer	Integer	Được sử dụng cho các giá trị Integer	4 byte
DateTime	DateTime	Được sử dụng cho một giá trị nhãn DateTime, ví dụ như ngày	

7.3 Hỗ trợ quốc tế (I18N)

Một số lớp mô hình thông tin có các thuộc tính chuỗi. Đó là khả năng chuẩn I18N và có thể được khoanh vùng trong các ngôn ngữ khác nhau. Ví dụ: Bao gồm tên và các thuộc tính mô tả của lớp RegistryObject trong mục 7.4.

Mô hình thông tin xác định InternationalString và giao diện LocalizedString để hỗ trợ các thuộc tính có khả năng I18N bên trong các lớp mô hình thông tin. Các lớp này được cụ thể bên dưới.

7.3.1 Lớp **InternationalString** (Chuỗi ký tự quốc tế)

Lớp này được sử dụng như sự thay thế cho kiểu chuỗi kể cả một thuộc tính chuỗi cần tới khả năng I18N. một *lớp* kiểu **InternationalString** bao gồm các **LocalizedString** nơi mà mỗi một chuỗi đặc trưng cho một sự việc riêng. Lớp **InternationalString** cung cấp các phương pháp thiết lập/ tiếp nhận cho việc bổ sung hoặc lấy các giá trị chuỗi cụ thể đối với **InternationalString**.

7.3.2 Lớp **LocalizedString** (Chuỗi ký tự vùng)

Lớp này sử dụng như một *lớp* bao bọc đơn mà nó kết hợp một chuỗi với nơi diễn ra. Lớp này cần được nằm trong *lớp* **InternationalString** mà một tập hợp các **LocalizedString** được cất trữ. Mỗi **LocalizedString** có một bộ ký tự và thuộc tính ngôn ngữ cũng như một thuộc tính giá trị của kiểu chuỗi.

7.4 Lớp **RegistryObject** (Đối tượng đăng ký)

Các *lớp* phụ trực tiếp:

Association (liên kết); AuditableEvent (sự kiện có thể kiểm tra); Classification (phân loại); ClassificationNode (nút phân loại); ExternalIdentifier (định danh ngoài); ExternalLink (liên kết ngoài); Organization (tổ chức); RegistryEntry (mục nhập đăng ký); User (người sử dụng); Service (dịch vụ); ServiceBinding (quy định dịch vụ); SpecificationLink (liên kết quy định).

RegistryObject cung cấp một *lớp* cơ sở bình thường cho hầu hết các đối tượng trong mô hình thông tin. Các *lớp* mô hình thông tin có một định danh duy nhất là *trường hợp lớp* này được có sau *lớp* **RegistryObject** (đối tượng đăng ký).

CHÚ THÍCH: Khe, địa chỉ, và một vài *lớp* khác không phải là *trường hợp lớp* **RegistryObject** bởi vì chúng không có sự tồn tại độc lập và định danh duy nhất. Chúng luôn là một phần của một số *trường hợp lớp* khác (ví dụ: Một **Organization** (tổ chức) đều có **PostalAddress**).

7.4.1 Tóm tắt thuộc tính

Bảng sau là một trong các bảng được tóm tắt các thuộc tính của một *lớp*.

Cột	Mô tả
Thuộc tính (Attribute)	Tên của thuộc tính
Kiểu dữ liệu (Data Type)	Kiểu dữ liệu của thuộc tính
Yêu cầu (Required)	Đặc trưng kể cả thuộc tính được yêu cầu thể hiện
Mặc định (Default)	Thể hiện giá trị mặc định trong <i>trường hợp</i> thuộc tính bị bỏ sót
quy định bởi (Specified By)	Thuộc tính được thể hiện bởi máy client hoặc chỉ ra bởi đăng ký. Trong một số <i>trường hợp</i> có thể bởi cả hai.
Khả năng thay đổi (Mutable)	Thể hiện một thuộc tính có thể bị thay đổi mỗi khi nó được thiết lập một giá trị nào đó.

Thuộc tính	Kiểu dữ liệu	Yêu cầu	Giá trị mặc định	Quy định bởi	Khả năng thay đổi
AccessControlPolicy	UUID	Không		Số đăng ký	Không
description	InternationalString	Không		Khách	Có
id	UUID	Có		Khách hoặc đăng ký	Không
name	InternationalString	Không		Khách	Có
objectType	LongName	Có		Số đăng ký	Không

7.4.2 Thuộc tính AccessControlPolicy (Chính sách kiểm soát truy cập)

Mỗi trường hợp RegistryObject (đối tượng đăng ký) có thể có một accessControlPolicy được liên kết với nó. một accessControlPolicy xác định kiểu bảo mật được liên kết với RegistryObject được hiểu là "ai được phép làm gì" với RegistryObject đó.

7.4.3 Thuộc tính description (Mô tả)

Mỗi trường hợp đối tượng đăng ký có thể có sự mô tả nguyên văn để dễ đọc và thân thiện với người sử dụng. Thuộc tính này là khả năng của kiểu I18N và kiểu InternationalString.

7.4.4 Thuộc tính id (Định danh)

Một trường hợp RegistryObject (đối tượng đăng ký) phải có một ID chung duy nhất. Các RegistryObject sử dụng id của các trường hợp RegistryObject (đối tượng đăng ký) khác đối với mục đích tham chiếu các đối tượng.

CHÚ THÍCH: Một số lớp trong mô hình thông tin không có nhu cầu đối với định danh duy nhất. Các lớp không được lấy từ lớp RegistryObject, ví dụ: Các lớp thực thể như số điện thoại, địa chỉ liên lạc, địa chỉ thư điện tử và tên người.

Tất cả các lớp đều được lấy từ RegistryObject có một định danh, định danh này được xác định bởi [UUID]. UUID dựa trên thuộc tính của định danh có thể được thể hiện bởi đối tượng khách. Nếu UUID này dựa trên định danh không được cụ thể, thì nó phải được sinh ra bởi đăng ký khi một trường hợp RegistryObject (đối tượng đăng ký) mới được gửi tới đăng ký trước tiên

7.4.5 Thuộc tính name (Tên)

Mỗi trường hợp RegistryObject (đối tượng đăng ký) có thể có tên người có thể đọc được. Tên này không cần phải là duy nhất đối với các trường hợp RegistryObject (đối tượng đăng ký) khác. Thuộc tính này là I18N và cũng là kiểu InternationalString (chuỗi ký tự quốc tế).

7.4.6 Thuộc tính objectType (Kiểu đối tượng)

Mỗi trường hợp RegistryObject (đối tượng đăng ký) có một objectType (kiểu đối tượng). Kiểu này đối với hầu hết các đối tượng trong mô hình thông tin là tên lớp của nó. Ví dụ: objectType (kiểu đối tượng) đối với một Classification (phân loại) là "Classification". Trường hợp ngoại lệ là objectType (kiểu đối tượng) đối với một đối tượng ngoại lai mà người sử dụng xác định và chỉ ra kiểu của loại kho chứa được kết hợp với đối tượng ngoại lai đó.

7.4.6.1 Các kiểu đối tượng xác định trước

Bảng sau liệt kê các objectType (kiểu đối tượng). Lưu ý rằng một đối tượng ngoại lai có nhiều kiểu được xác định dựa trên kiểu của kho. Thêm vào đó, có nhiều kiểu được xác định đối với tất cả các lớp phụ của RegistryObject (đối tượng đăng ký).

Các kiểu đối tượng xác định trước được xác định như một ClassificationScheme (giản đồ phân loại). Khi cơ chế này có thể dễ dàng mở ra, một đăng ký phải hỗ trợ các objectType (kiểu đối tượng) được liệt kê như sau:

Tên	Mô tả
Unkown	Một đối tượng phân chia các nội dung có kiểu không đặc trưng hoặc không được nhận biết
CPA	Đối tượng ngoại lai của kiểu phân chia một tài liệu XML Thỏa thuận giao thức cộng tác (CPA) đại diện cho một thỏa thuận kỹ thuật giữa hai bên tham gia về làm thế nào để hoạch định cách giao tiếp với nhau bằng cách sử dụng một giao thức cụ thể
CPP	Đối tượng ngoại lai của kiểu phân chia một tài liệu được gọi là Hồ sơ giao thức hợp tác (CPP), nó cung cấp thông tin về một bên tham gia tham gia vào một giao dịch kinh doanh. Xem thêm phần [ebCPP]
Process	Đối tượng ngoại lai phân chia thành một tài liệu mô tả quá trình
SoftwareComponent	Đối tượng ngoại lai phân chia thành các thành phần của phần mềm (ví dụ: Một bên của thư viện lớp).
UMLModel	Đối tượng ngoại lai phân chia thành một UMLModel (Mô hình UML)
Giản đồ XML	Đối tượng ngoại lai phân chia thành một giản đồ XML (DTD, giản đồ XML, ngữ pháp RELAX, v.v.)
RegistryPackage	Một đối tượng RegistryPackage
ExternalLink	Đối tượng ExternalLink (liên kết ngoài)
ExternalIdentifier	Đối tượng ExternalIdentifier (định danh ngoài)
Association	Đối tượng Association (kết hợp)
ClassificationScheme	Đối tượng Classification Scheme (giản đồ phân loại)
Classification	Đối tượng Classification (phân loại)
ClassificationNode	Đối tượng ClassificationNode (nút phân loại)
AuditableEvent	Đối tượng AuditableEvent (sự kiện có thể kiểm tra)
User	Đối tượng User (người sử dụng)
Organization	Đối tượng Organization (tổ chức)
Service	Đối tượng Service (dịch vụ)
ServiceBinding	Đối tượng ServiceBinding (quy định dịch vụ)
SpecificationLink	Đối tượng SpecificationLink (liên kết quy định)

7.4.7 Tóm tắt phương pháp

Ngoài các thuộc tính của nó, *lớp* RegistryObject cũng xác định các phương pháp sau đây. Các phương pháp này được sử dụng để điều hướng các liên kết quan hệ từ một *trường hợp* RegistryObject (đối tượng đăng ký) tới các đối tượng khác.

Bảng tóm tắt phương pháp đối với RegistryObject	
Tập hợp	getAssociations() Lấy tất cả các liên kết tại nơi đối tượng này là gốc của liên kết đó.
Tập hợp	getAuditTrail() Lấy vết kiểm tra đầy đủ của tất cả các yêu cầu ảnh hưởng thay đổi trạng thái trong đối tượng đó như một tập hợp có thứ tự của các đối tượng AuditableEvent (sự kiện có thể kiểm tra).
Tập hợp	getClassifications() Lấy Classification để phân loại kiểu đối tượng đó.
Tập hợp	getExternalIdentifiers() Lấy tập hợp của các ExternalIdentifier kèm theo đối tượng này.
Tập hợp	getExternalLinks() Lấy ExternalLinks kèm theo đối tượng này.
Tập hợp	getOrganizations(String type) Lấy Organizations kèm theo đối tượng này. Nếu quy định kiểu nonnull thì nó được sử dụng như một bộ lọc phù hợp chỉ quy định kiểu của các Organization (tổ chức) như được chỉ ra bởi thuộc tính associationType (kiểu liên kết) trong <i>trường hợp</i> Association liên kết đối tượng với Organization (tổ chức) đó.
Tập hợp	getRegistryPackages() Lấy RegistryPackages chứa đối tượng này.
Tập hợp	getSlots() Lấy các Slot kèm theo đối tượng này.

7.5 Lớp RegistryEntry (Lớp mục nhập đăng ký)

Các lớp chuẩn:

RegistryObject

Các phân lớp:

ClassificationScheme (giản đồ phân loại), ExtrinsicObject, RegistryPackage

RegistryEntry là một *lớp* cơ sở chung cho các *lớp* trong mô hình thông tin để yêu cầu siêu dữ liệu nằm ngoài siêu dữ liệu tối thiểu được cung cấp bởi *lớp* RegistryObject (đối tượng đăng ký). RegistryEntry được sử dụng như một *lớp* cơ sở cho các đối tượng thô cấp cao trong sổ đăng ký đó. Chu kỳ điển hình của chúng yêu cầu nhiều quản lý hơn (nghĩa là; có thể yêu cầu sự chấp nhận, phản đối). Điển hình chúng có tương đối ít *trường hợp* trừ *trường hợp* sử dụng như một gốc hệ đẳng cấp *thành phần* cấu tạo bao gồm các đối tượng số là các Classification (phân loại) của RegistryObject (đối tượng đăng ký) nhưng không của RegistryEntry (mục nhập đăng ký).

Siêu dữ liệu bổ sung được mô tả bởi các thuộc tính của *Lớp* RegistryEntry (*lớp* mục nhập đăng ký) sau:

7.5.1 Tóm tắt thuộc tính

Thuộc tính	Kiểu dữ liệu	Yêu cầu	Giá trị mặc định	Được quy định bởi	Khả năng thay đổi
expiration	DateTime	Không		Máy client	Có
majorVersion	Integer	Có	1	Sổ đăng ký	Có
minorVersion	Integer	Có	0	Sổ đăng ký	Có
stability	LongName	Không		Máy client	Có
status	LongName	Có		Sổ đăng ký	Có
UserVersion (phiên bản người sử dụng)	ShortName	Không		Máy client	Có

Chú ý rằng các thuộc tính kế thừa của *Lớp RegistryEntry* (lớp mục nhập đăng ký) từ *lớp RegistryObject* không được chỉ ra trong bảng trên.

7.5.2 Thuộc tính expiration (Thời gian hết hạn)

Mỗi *trường hợp RegistryEntry* (mục nhập đăng ký) có thể có một thuộc tính `expirationDate` (ngày hết hạn). Thuộc tính này xác định một giới hạn `DateTime` trên chỉ dẫn về trạng thái ổn định được cung cấp bởi Thuộc tính `stability` (tính ổn định). Ngay khi đạt tới `expirationDate` thuộc tính `stability` (tính ổn định) tác động trở thành `STABILITY_DYNAMIC` hàm ý rằng hạng mục trong kho có thể thay đổi bất kỳ thời điểm nào dưới bất kỳ hình thức nào. Một giá trị `null` hàm ý rằng không có `DateTime` hết hạn của Thuộc tính `stability` (tính ổn định).

7.5.3 Thuộc tính majorVersion (Phiên bản chính)

Mỗi *trường hợp RegistryEntry* (mục nhập đăng ký) phải có một số hiệu soát xét chính đối với phiên bản hiện thời của *trường hợp RegistryEntry*. Số hiệu này được ấn định bởi sổ đăng ký khi đối tượng được tạo ra. Số hiệu này có thể được cập nhật bởi sổ đăng ký khi một đối tượng được cập nhật.

7.5.4 Thuộc tính minorVersion (Phiên bản phụ)

Mỗi *trường hợp RegistryEntry* (mục nhập đăng ký) phải có một số hiệu soát xét thứ yếu đối với phiên bản hiện tại của *trường hợp RegistryEntry* (mục nhập đăng ký) đó. Số hiệu này được ấn định bởi sổ đăng ký khi đối tượng được tạo ra. Số hiệu này có thể được cập nhật bởi sổ đăng ký khi một đối tượng được cập nhật.

7.5.5 Thuộc tính stability (Tính ổn định)

Mỗi *trường hợp RegistryEntry* (mục nhập đăng ký) có thể có một bộ chỉ báo tính. Bộ chỉ báo tính được cung cấp bởi người đệ trình như một sự chỉ dẫn của mức tính đối với hạng mục kho.

7.5.5.1 Các tập đếm tính của RegistryEntry xác định trước

Bảng sau đây liệt kê các lựa chọn đã xác định đối với thuộc tính tính của RegistryEntry. Các kiểu tính xác định trước này được định nghĩa như một *ClassificationScheme* (giản đồ phân loại). Trong khi giản đồ này có thể được mở rộng để dành, một số đăng ký có thể hỗ trợ các kiểu tính được liệt kê như sau:

Tên	Mô tả
Dynamic	Tính ổn định của một RegistryEntry để chỉ ra rằng nội dung đó là động và có thể được thay đổi tùy ý bởi người đệ trình tại bất kỳ thời điểm nào.
DynamicCompatible	Tính ổn định của một RegistryEntry để chỉ ra rằng nội dung đó là động và có thể được thay đổi theo một cách thức tương thích ngược bởi người đệ trình tại bất kỳ thời điểm nào.
Static	Tính ổn định của một RegistryEntry để chỉ ra rằng nội dung đó là ổn định và không bị thay đổi bởi người đệ trình.

7.5.6 Thuộc tính status (Trạng thái)

Mỗi *trường hợp* RegistryEntry (mục nhập đăng ký) phải có một bộ chỉ thị trạng thái chu kỳ công tác. Các trạng thái được ấn định bởi số đăng ký.

7.5.6.1 Các kiểu trạng thái của RegistryObject (đối tượng đăng ký) xác định trước

Bảng sau đây liệt kê các lựa chọn đã xác định đối với Thuộc tính status (trạng thái) của RegistryObject (đối tượng đăng ký). Các kiểu trạng thái xác định trước này được định nghĩa như một *ClassificationScheme* (giản đồ phân loại).

Tên	Mô tả
Submitted	Trạng thái của một RegistryObject mà nội dung danh mục liệt kê đã được đệ trình trong sổ đăng ký.
Approved	Trạng thái của một RegistryObject mà nội dung danh mục liệt kê đã được đệ trình trong sổ đăng ký và đã được phê chuẩn sau đó.
Deprecated	Trạng thái của một RegistryObject mà nội dung danh mục liệt kê đã được đệ trình trong sổ đăng ký và đã được phủ quyết sau đó.
Withdrawn	Trạng thái của một RegistryObject mà nội dung danh mục liệt kê đã được rút khỏi sổ đăng ký.

7.5.7 Thuộc tính userVersion (Phiên bản người sử dụng)

Mỗi *trường hợp* RegistryEntry (mục nhập đăng ký) có thể có một userVersion (phiên bản người sử dụng). UserVersion (phiên bản người sử dụng) tương tự như cặp majorVersion-minorVersion. Cả hai phiên bản cung cấp một chỉ dẫn về phiên bản của đối tượng đó. Cặp majorVersion-minorVersion được cung cấp bởi số đăng ký trong khi userVersion (phiên bản người sử dụng) đưa ra một người sử dụng quy định phiên bản cho đối tượng đó.

7.5.8 Tóm tắt phương pháp

Ngoài các thuộc tính của nó, Lớp RegistryEntry (lớp mục nhập đăng ký) cũng xác định các phương pháp sau đây:

Tóm tắt phương pháp for RegistryEntry	
Organization	<p>getSubmittingOrganization()</p> <p>Lấy trường hợp Organization (tổ chức) của tổ chức đó để đệ trình trường hợp RegistryEntry (mục nhập đăng ký) cho trước. Phương pháp này trả lại một kết quả non-null đối với toàn bộ các RegistryEntry. Đối với việc chỉ định đặc quyền, tổ chức đó được trả lại bởi phương pháp này liên quan đến chủ sở hữu trường hợp RegistryEntry (mục nhập đăng ký) đó.</p>
Organization	<p>getResponsibleOrganization()</p> <p>Lấy trường hợp Organization (tổ chức) của tổ chức có trách nhiệm đối với định nghĩa, phê chuẩn, và/hoặc duy trì hạng mục kho được tham chiếu bởi trường hợp RegistryEntry (mục nhập đăng ký) cho trước. Phương pháp này có thể trả lại một kết quả nếu việc đệ trình tổ chức RegistryEntry đó Unknown một tổ chức có trách nhiệm hoặc nếu cơ quan có thẩm quyền đăng ký không ấn định một tổ chức có thẩm quyền.</p>

7.6 Lớp Slot (Khe bổ sung)

Các trường hợp Slot (khe bổ sung) cung cấp một phương pháp động để bổ sung các thuộc tính tùy ý cho các trường hợp RegistryObject (đối tượng đăng ký). Khả năng bổ sung các thuộc tính một cách linh động này cho các trường hợp RegistryObject (đối tượng đăng ký) cho phép các tính năng mở rộng trong mô hình thông tin.

Một trường hợp RegistryObject (đối tượng đăng ký) có thể có không hay nhiều Slot (khe bổ sung). Một Slot (khe bổ sung) bao gồm một tên, một thuộc tính slotType (kiểu khe bổ sung) và một tập hợp các giá trị.

7.6.1 Tóm tắt thuộc tính

Thuộc tính	Kiểu dữ liệu	Yêu cầu	Giá trị mặc định	Được quy định bởi	Khả năng thay đổi
Name	LongName	Có		Máy client	Không
SlotType	LongName	Không		Máy client	Không
Values	Collection of ShortName	Có		Máy client	Không

7.6.2 Thuộc tính name (Tên)

Mỗi trường hợp Slot (khe bổ sung) phải có một tên. Tên này là phương tiện chính để xác định một trường hợp Slot (khe bổ sung) trong một RegistryObject (đối tượng đăng ký). Do đó, tên của một trường hợp Slot (khe bổ sung) phải là mang tính cục bộ duy nhất trong trường hợp RegistryObject (đối tượng đăng ký).

7.6.3 Thuộc tính slotType (Kiểu khe bổ sung)

Mỗi trường hợp Slot (khe bổ sung) có thể có một thuộc tính slotType (kiểu khe bổ sung) để cho phép các slot khác nhau được nhóm lại cùng nhau.

7.6.4 Thuộc tính values (Giá trị)

Một *trường hợp* Slot (khe bổ sung) phải có một tập hợp các giá trị. Tập hợp các giá trị này có thể là rỗng. Do một Slot (khe bổ sung) biểu diễn một thuộc tính có thể mở rộng mà giá trị của nó có thể là một tập hợp, vì vậy một Slot (khe bổ sung) được phép có tập hợp các giá trị hơn là một giá trị đơn.

7.7 Lớp ExtrinsicObject (Đối tượng ngoại lai)

Các lớp chuẩn:

RegistryEntry (mục nhập đăng ký), **RegistryObject** (đối tượng đăng ký)

Các Lớp ExtrinsicObject (đối tượng ngoại lai) cung cấp siêu dữ liệu để mô tả nội dung được đệ trình mà kiểu của nó không được hiểu về bản chất trong *sổ đăng ký* và vì vậy PHẢI được mô tả bởi các phương pháp bổ sung các thuộc tính (như là : kiểu MIME).

Do sổ đăng ký có thể bao gồm nội dung tùy ý không yêu cầu kiến thức về bản chất nội dung đó, nên các Lớp ExtrinsicObject (đối tượng ngoại lai), các thuộc tính metadata (siêu dữ liệu) đặc biệt để cung cấp một số kiến thức về đối tượng đó (như là; kiểu mime).

Các ví dụ về nội dung được mô tả bởi ExtrinsicObject (đối tượng ngoại lai) bao gồm *các hồ sơ về thủ tục hợp tác [ebCPP]*, *Các mô tả quá trình kinh doanh*, và các giản đồ.

7.7.1 Tóm tắt thuộc tính

Thuộc tính	Kiểu dữ liệu	Yêu cầu	Giá trị mặc định	Được quy định bởi	Khả năng thay đổi
IsOpaque	Boolean	Không		Máy client	Không
MimeType	LongName	Không		Máy client	Không

Chú ý rằng các thuộc tính được kế thừa từ RegistryEntry (mục nhập đăng ký) và RegistryObject (đối tượng đăng ký) không được chỉ ra trong bảng trên.

7.7.2 Thuộc tính isOpaque (Đục)

Mỗi *trường hợp* ExtrinsicObject (đối tượng ngoại lai) có thể có một thuộc tính isOpaque (đục) đã xác định. Thuộc tính này xác định nội dung đó có được ghi vào mục lục bởi ExtrinsicObject (đối tượng ngoại lai) này là không rõ ràng đối với sổ đăng ký đó. Trong một số *trường hợp*, một *Tổ chức đệ trình* có thể đệ trình nội dung được mật mã hóa và thậm chí không thể đọc được bởi sổ đăng ký.

7.7.3 Thuộc tính mimeType (Kiểu MIME)

Mỗi *trường hợp* ExtrinsicObject (đối tượng ngoại lai) có thể có một thuộc tính mimeType (kiểu MIME) đã xác định. Thuộc tính mimeType (kiểu MIME) cung cấp thông tin về kiểu hạng mục kho được ghi vào danh mục bởi *trường hợp* ExtrinsicObject (đối tượng ngoại lai).

7.8 Lớp RegistryPackage

Các lớp chuẩn:

RegistryEntry (mục nhập đăng ký), **RegistryObject** (đối tượng đăng ký)

Các *trường hợp* RegistryPackage (gói đăng ký) cho phép việc nhóm các *trường hợp* RegistryObject (đối tượng đăng ký) có quan hệ logic ngay cả khi các đối tượng của thành phần riêng thuộc các tổ chức đệ trình khác nhau.

7.8.1 Tóm tắt thuộc tính

Lớp RegistryPackage (gói đăng ký) không biết các thuộc tính mới ngoài những thuộc tính được kế thừa từ các lớp cơ sở của RegistryEntry (mục nhập đăng ký) và RegistryObject (đối tượng đăng ký). Các thuộc tính kế thừa không được chỉ ra ở đây.

7.8.2 Tóm tắt phương pháp

Ngoài các thuộc tính của nó, *Lớp RegistryPackage* (gói đăng ký) cũng xác định các phương pháp sau đây :

Tóm tắt phương pháp of RegistryPackage	
Collection	<p>getMemberObjects()</p> <p>Lấy tập hợp các <i>trường hợp</i> RegistryObject (đối tượng đăng ký) là thành phần của RegistryPackage (gói đăng ký) này.</p>

7.9 Lớp ExternalIdentifier (Định danh ngoài)

Các lớp chuẩn:

RegistryObject

Các *trường hợp* ExternalIdentifier (định danh ngoài) cung cấp thông tin định danh bổ sung cho RegistryObject (đối tượng đăng ký) như là Số hiệu DUNS, Số an ninh xã hội, hoặc một bí danh tham chiếu gián tiếp định danh (như là, "DUNS", "Social Security #"), và giá trị thuộc tính đó bao gồm thông tin thực (như là, Số hiệu DUNS, số an ninh xã hội). Mỗi RegistryObject (đối tượng đăng ký) có thể bao gồm không hoặc nhiều các *trường hợp* ExternalIdentifier (định danh ngoài).

7.9.1 Tóm tắt thuộc tính

Thuộc tính	Kiểu dữ liệu	Yêu cầu	Giá trị mặc định	Được quy định bởi	Khả năng thay đổi
IdentificationScheme	UUID	Có		Máy client	Có
RegistryObject	UUID	Có		Máy client	Không
Value	ShortName	Có		Máy client	Có

Chú ý rằng các thuộc tính được kế thừa từ các lớp cơ sở của lớp này không được chỉ ra.

TCVN ISO/TS 15000-3 : 2007

7.9.2 Thuộc tính **identificationScheme** (Giản đồ định danh)

Mỗi *trường hợp ExternalIdentifier* (định danh ngoài) phải có một thuộc tính **identificationScheme** (giản đồ định danh) để tham chiếu tới **ClassificationScheme** (giản đồ phân loại). **ClassificationScheme** (giản đồ phân loại) này xác định tên miền trong đó một định danh được xác định có sử dụng Thuộc tính **value** (giá trị) đối với **RegistryObject** đó được tham chiếu bởi Thuộc tính **registryObject** (đối tượng đăng ký).

7.9.3 Thuộc tính **registryObject** (Đối tượng đăng ký)

Mỗi *trường hợp ExternalIdentifier* (định danh ngoài) phải có một thuộc tính **registryObject** (đối tượng đăng ký) để tham chiếu **RegistryObject** gốc đối với **ExternalIdentifier** đó.

7.9.4 Thuộc tính **value** (Giá trị)

Mỗi *trường hợp ExternalIdentifier* (định danh ngoài) phải có một thuộc tính **value** (giá trị) để cung cấp giá trị định danh đối với **ExternalIdentifier** đó (như là, số an ninh xã hội thực).

7.10 Lớp **ExternalLink** (Liên kết ngoài)

Các lớp chuẩn:

RegistryObject

ExternalLink sử dụng URI làm một loại liên kết sử dụng để liên kết nội dung trong *Registry* với nội dung nằm ngoài *sổ đăng ký*. Chẳng hạn như, một **Organization** (tổ chức) muốn trình bày hoặc đệ trình DTD có thể sử dụng **ExternalLink** để kết nối DTD với trang chủ của **Organization** (tổ chức) đó.

7.10.1 Bảng tóm tắt các thuộc tính

Thuộc tính	Kiểu dữ liệu	Yêu cầu	Giá trị mặc định	Nơi xác định	Khả năng thay đổi
ExternalURI	URI	Có		Máy client	Có

7.10.2 Thuộc tính **ExternalURI** (uri ngoài)

Mỗi *trường hợp của ExternalURI* phải có một thuộc tính **ExternalURI** (uri ngoài) được xác định.

Thuộc tính **ExternalURI** (uri ngoài) cung cấp một URI cho các nguồn thông tin ngoại trú mà *trường hợp của ExternalURI* này hướng tới.

Nếu URI đó là một URL (địa chỉ một trang mạng internet hay intranet) thì chắc chắn *Registry* phải kiểm tra tính hợp lệ của đường liên kết URL là có thể giải quyết được tại thời điểm đệ trình trước khi quyết định chấp nhận **ExternalLink** (liên kết ngoài) vào *Registry* (sổ đăng ký).

7.10.3 Bảng tóm tắt phương pháp thực thi

Bên cạnh các thuộc tính cố định của mình, *Lớp ExternalLink (liên kết ngoài)* cũng xác định các phương pháp được ghi ở bảng sau đây:

Tóm tắt phương pháp của <i>ExternalLink</i> (liên kết ngoài)	
Tập hợp	<p><u>getlinkobject ()</u></p> <p>Lấy tập hợp các <i>RegistryObject</i> được liên kết bởi <i>Lớp ExternalLink</i> (liên kết ngoài) với nội dung và cấu trúc bên ngoài <i>Registry</i>.</p>

8 Vết kiểm tra số đăng ký

Phần này mô tả các phần tử của mô hình thông tin được sử dụng để hỗ trợ khả năng truy tìm vết kiểm tra của *Registry*. Có một vài *lớp* trong phần này là các *Lớp Entity* (thực thể) được sử dụng như các trình bao bọc với mục đích xây dựng mô hình cho tổ hợp các thuộc tính có liên quan. Các trình bao bọc này cũng tương tự như cấu trúc "struct" trong ngôn ngữ lập trình C.

Phương pháp *GetAuditTrail ()* của *RegistryObject* (đối tượng đăng ký) trả về tập hợp có thứ tự của các *AuditableEvent* (sự kiện có thể kiểm tra). Các *AuditableEvent* (sự kiện có thể kiểm tra) này cấu thành nên vết kiểm tra cho *RegistryObject* (đối tượng đăng ký). *AuditableEvent* (sự kiện có thể kiểm tra) bao gồm tem thời gian cho *Event* (sự kiện) đó. Mỗi *AuditableEvent* (sự kiện có thể kiểm tra) đều có danh mục mẫu đối chiếu dành cho một người sử dụng trong đó xác định và nhận dạng một người sử dụng cụ thể thực thi một hành động góp phần làm nên *AuditableEvent* (sự kiện có thể kiểm tra). Mỗi người sử dụng được gắn với một *Organization* (tổ chức) và thường thì các *Organization* (tổ chức) này là *Organization* (tổ chức) đệ trình *RegistryObject* (đối tượng đăng ký).

8.1 Lớp *AuditableEvent* (Sự kiện có thể kiểm tra)

Các lớp chuẩn:

RegistryObject

Các trường hợp *AuditableEvent* (sự kiện có thể kiểm tra) cung cấp một bản ghi thời gian dài các *Event* (sự kiện) tạo ra sự thay đổi trong một *RegistryObject* (đối tượng đăng ký). Một *RegistryObject* (đối tượng đăng ký) bao giờ cũng có liên quan đến một tập hợp có thứ tự của các trường hợp *AuditableEvent* (sự kiện có thể kiểm tra), cung cấp đầy đủ vết kiểm tra hoàn chỉnh cho *RegistryObject* (đối tượng đăng ký).

Thường thì các *AuditableEvent* (sự kiện có thể kiểm tra) là kết quả của một yêu cầu từ máy client. Các trường hợp *AuditableEvent* (sự kiện có thể kiểm tra) thường do số đăng ký *Service* tạo ra để ghi lại các *Event* (sự kiện) như thế này.

Các *Event* (sự kiện) như thế này thường tạo ra sự thay đổi trong chu kỳ tồn tại của một *RegistryObject* nhất định. Chẳng hạn như một máy client có thể yêu cầu tạo lập, cập nhật, từ chối hoặc huỷ bỏ một *RegistryObject* (đối tượng đăng ký) nào đó theo đúng chỉ dẫn của máy đó. một *AuditableEvent* (sự kiện có thể kiểm tra) được tạo ra nếu và chỉ khi nào có một yêu cầu tạo lập hoặc sửa đổi nội dung hoặc quyền sở hữu đối với *RegistryObject* (đối tượng đăng ký).

TCVN ISO/TS 15000-3 : 2007

Các yêu cầu dạng *Read-only* (chỉ có thể đọc) không tạo ra một *AuditableEvent* (sự kiện có thể kiểm tra). Không một *AuditableEvent* (sự kiện có thể kiểm tra) nào được tạo ra dành cho một *RegistryObject* (đối tượng đăng ký) khi nó được phân loại, gán cho một *RegistryPackage* (gói đăng ký) hoặc được kết hợp với các *RegistryObject* (đối tượng đăng ký) khác. Đây là một nguyên tắc cơ bản mà các lập trình viên cần nắm rõ khi tiến hành thực thi công việc của mình.

8.1.1 Bảng tóm tắt các thuộc tính

Thuộc tính	Kiểu dữ liệu	Yêu cầu	Giá trị mặc định	Được quy định bởi	Khả năng thay đổi
<i>eventType</i>	LongName	Có		Số đăng ký	Không
<i>registryObject</i>	UUID	Có		Số đăng ký	Không
<i>timestamp</i>	DateTime	Có		Số đăng ký	Không
<i>user</i>	UUID	Có		Số đăng ký	Không

8.1.2 Thuộc tính *eventType* (Kiểu sự kiện)

Mỗi *AuditableEvent* (sự kiện có thể kiểm tra) đều chứa trong mình một thuộc tính *eventType* (kiểu sự kiện) nhất định giúp phân biệt và nhận dạng kiểu sự kiện được ghi lại bởi *AuditableEvent* (sự kiện có thể kiểm tra).

8.1.2.1 Các kiểu *AuditableEvent* (sự kiện có thể kiểm tra) xác định trước

Bảng sau đây liệt kê các kiểu *AuditableEvent* (sự kiện có thể kiểm tra) xác định trước. Các *eventType* xác định trước được định nghĩa là *ClassificationScheme* (giản đồ phân loại) xác định trước và được đặt tên là "*EventType*". Một số đăng ký phải hỗ trợ các *eventType* được liệt kê dưới đây.

Tên	Mô tả
Created	Một <i>Event</i> góp phần tạo ra một <i>RegistryObject</i>
Deleted	Một <i>Event</i> huỷ bỏ <i>RegistryObject</i>
Deprecated	Một <i>Event</i> phản đối <i>RegistryObject</i>
Updated	Một <i>Event</i> cập nhật tình trạng của một <i>RegistryObject</i>
Versioned	Một <i>Event</i> gắn phiên bản cho <i>RegistryObject</i> (đối tượng đăng ký)

8.1.3 Thuộc tính *registryObject* (Đối tượng đăng ký)

Mỗi *AuditableEvent* (sự kiện có thể kiểm tra) phải có một thuộc tính *registryObject* (đối tượng đăng ký) để xác định trường hợp *RegistryObject* (đối tượng đăng ký) bị ảnh hưởng bởi sự kiện này.

8.1.4 Thuộc tính *TimeStamp* (Tem thời gian)

Mỗi *AuditableEvent* (sự kiện có thể kiểm tra) phải có thuộc tính *timestamp* (tem thời gian) có ghi lại ngày, tháng, năm và thời gian cụ thể mà sự kiện này xảy ra.

8.1.5 Thuộc tính user (Người sử dụng)

Mỗi AuditableEvent (sự kiện có thể kiểm tra) phải có thuộc tính user (người sử dụng) giúp phân biệt và xác định User (user) - người đã gửi yêu cầu góp phần tạo ra sự kiện này ảnh hưởng đến trường hợp RegistryObject (đối tượng đăng ký).

8.2 Lớp User (Người sử dụng)

Các lớp chuẩn:

RegistryObject

Các trường hợp của User (người sử dụng) được sử dụng trong một AuditableEvent (sự kiện có thể kiểm tra) để theo dõi và nhận dạng người yêu cầu - Người đã gửi yêu cầu góp phần tạo nên AuditableEvent (sự kiện có thể kiểm tra).

8.2.1 Bảng tóm tắt các thuộc tính

Thuộc tính	Kiểu dữ liệu	Yêu cầu	Giá trị mặc định	Được quy định bởi	Khả năng thay đổi
address	PostalAddress	Có		Máy client	Có
emailAddresses	Collection of EmailAddress	Có		Máy client	Có
organization	UUID	Có		Máy client	Không
personName	PersonName	Có		Máy client	Không
telephoneNumbers	Collection of TelephoneNumber	Có		Máy client	Có
url	URI	Không		Máy client	Có

8.2.2 Thuộc tính address (Địa chỉ)

Mỗi trường hợp User (người sử dụng) phải có thuộc tính address (địa chỉ) có tác dụng cung cấp cổng địa chỉ cho User (người sử dụng) đó.

8.2.3 Thuộc tính emailAddresses (Địa chỉ thư điện tử)

Mỗi trường hợp User (người sử dụng) có một thuộc tính emailAddresses (địa chỉ thư điện tử). Đây là một tập hợp bao gồm các địa chỉ hòm thư điện tử. Mỗi địa chỉ thư mục chỉ hòm thư điện tử cung cấp một địa chỉ hòm thư điện tử cho User (người sử dụng) đó. một User (người sử dụng) phải có ít nhất là một hòm thư điện tử, như vậy người sử dụng đó mới có thể truy cập mạng và đăng ký được.

8.2.4 Thuộc tính organization (Tổ chức)

Mỗi trường hợp User (người sử dụng) phải có một thuộc tính organization (tổ chức) được coi là nhân tố tham chiếu một trường hợp của Organization (tổ chức) cho một Organization (tổ chức) của User (người sử dụng) đó.

TCVN ISO/TS 15000-3 : 2007

8.2.5 Thuộc tính `personName` (Tên riêng)

Mỗi *trường hợp User* (người sử dụng) phải có thuộc tính `personName` (tên riêng) có tác dụng cung cấp tên cho *User* (người sử dụng) đó.

8.2.6 Thuộc tính `telephoneNumbers` (Số điện thoại)

Mỗi *trường hợp User* (người sử dụng) phải có một thuộc tính `telephoneNumbers` (số điện thoại) trong đó bao gồm tập hợp các *trường hợp* của số điện thoại cho mỗi số điện thoại được xác định dành cho *User* (người sử dụng) đó. Một *User* (người sử dụng) phải có ít nhất một số điện thoại.

8.2.7 Thuộc tính `url`

Mỗi *trường hợp User* (người sử dụng) có thể có một thuộc tính URL có tác dụng cung cấp địa chỉ URL cho trang web có liên hệ với *User* (người sử dụng) đó.

8.3 Lớp `Organization` (Tổ chức)

Các lớp chuẩn:

`RegistryObject`

Các *trường hợp Organization* (tổ chức) cung cấp thông tin về các `Organization` (tổ chức) chẳng hạn như `Organization` (tổ chức) đệ trình. Mỗi *trường hợp Organization* có thể có một tham chiếu tới một `Organization` (tổ chức) gốc.

8.3.1 Bảng tổng kết thuộc tính

Thuộc tính	Kiểu dữ liệu	Yêu cầu	Giá trị mặc định	Được quy định bởi	Khả năng thay đổi
<code>address</code>	<code>PostalAddress</code>	Có		<i>Máy client</i>	Có
<code>parent</code>	<code>UUID</code>	Không		<i>Máy client</i>	Có
<code>primaryContact</code>	<code>UUID</code>	Có		<i>Máy client</i>	Không
<code>telephoneNumbers</code>	Collection of <code>TelephoneNumber</code>	Có		<i>Máy client</i>	Có

8.3.2 Thuộc tính `address` (Địa chỉ)

Mỗi *trường hợp* của `Organization` (tổ chức) phải có một thuộc tính `address` (địa chỉ) cung cấp địa chỉ cổng cho `Organization` (tổ chức) đó.

8.3.3 Thuộc tính `parent` (Gốc)

Mỗi *trường hợp* của `Organization` (tổ chức) có thể có một thuộc tính `parent` (gốc) có vai trò như nhân tố tham chiếu *trường hợp* của `Organization` (tổ chức) (nếu có) cho `Organization` (tổ chức) đó.

8.3.4 Thuộc tính **primaryContact** (Điểm liên lạc chính)

Mỗi *trường hợp* của Organization (tổ chức) phải có một thuộc tính **primaryContact** (điểm liên lạc chính) tham chiếu *trường hợp* User (người sử dụng) cho User và đây chính là *primaryContact* (điểm liên lạc chính) cho Organization (tổ chức) đó.

8.3.5 Thuộc tính **telephoneNumbers** (Số điện thoại)

Mỗi *trường hợp* của Organization (tổ chức) phải có một thuộc tính **telephoneNumbers** (số điện thoại) trong đó bao gồm tập hợp của các *trường hợp* số điện thoại đối với mỗi số điện thoại cố định được xác định là dành cho Organization (tổ chức) đó. Một Organization (tổ chức) phải có ít nhất một số điện thoại.

8.4 Lớp **PostalAddress** (Địa chỉ bưu điện)

PostalAddress là một *lớp thực thể* đơn giản, có thể sử dụng đi sử dụng lại, có tác dụng xác định các thuộc tính của địa chỉ cổng.

Bảng tổng kết các thuộc tính

Thuộc tính	Kiểu dữ liệu	Yêu cầu	Giá trị mặc định	Được quy định bởi	Khả năng thay đổi
city	ShortName	Không		<i>Máy client</i>	Có
country	ShortName	Không		<i>Máy client</i>	Có
postalCode	ShortName	Không		<i>Máy client</i>	Có
state	ShortName	Không		<i>Máy client</i>	Có
street	ShortName	Không		<i>Máy client</i>	Có
streetNumber	string 32	Không		<i>Máy client</i>	Có

8.4.2 Thuộc tính **city** (Thành phố)

Mỗi *PostalAddress* (địa chỉ bưu điện) có thể có một thuộc tính **city** (thành phố) xác định và định dạng địa chỉ.

8.4.3 Thuộc tính **country** (Quốc gia)

Mỗi *PostalAddress* (địa chỉ bưu điện) có thể có một thuộc tính **country** (quốc gia) xác định địa chỉ trong quốc gia.

8.4.4 Thuộc tính **postalCode** (Mã bưu điện)

Mỗi *PostalAddress* (địa chỉ bưu điện) có thể có một thuộc tính **postalCode** (mã bưu điện) xác định mã số bưu điện (chẳng hạn như mã số zip) dành cho địa chỉ đó.

8.4.5 Thuộc tính **state** (Bang)

Mỗi *PostalAddress* (địa chỉ bưu điện) có thể có một thuộc tính **state** (bang) xác định bang, tỉnh hoặc khu vực của địa chỉ đó.

TCVN ISO/TS 15000-3 : 2007

8.4.6 Thuộc tính **street** (Đường phố)

Mỗi **PostalAddress** (địa chỉ bưu điện) có thể có một thuộc tính **street** (đường phố) xác định tên đường phố cho địa chỉ đó.

8.4.7 Thuộc tính **streetNumber** (Số hiệu đường phố)

Mỗi **PostalAddress** (địa chỉ bưu điện) có thể có một thuộc tính **streetNumber** (số hiệu đường phố) xác định số đường phố (ví dụ: 65) cho địa chỉ đường phố.

8.4.8 Bảng tóm tắt phương pháp

Ngoài các thuộc tính trên, lớp **PostalAddress** (địa chỉ bưu điện) cũng xác định các phương pháp sau.

Tóm tắt phương pháp thực thi của ExternalLink (liên kết ngoài)	
Tập hợp	<u>getSlots ()</u> Lấy tập hợp của các khe bổ sung cho đối tượng này. Mỗi PostalAddress (địa chỉ bưu điện) có thể có nhiều <i>trường hợp</i> khe bổ sung, trong đó mỗi khe bổ sung là một thuộc tính được xác định. Việc sử dụng khe bổ sung cho phép <i>Máy client</i> có thể mở rộng <i>Lớp PostalAddress (địa chỉ bưu điện)</i> bằng cách xác định các thuộc tính bổ sung động có sử dụng khe bổ sung để xử lý các nhu cầu cụ thể.

8.5 Lớp **telephoneNumbers** (Số điện thoại)

Đây là một lớp *thực thể* đơn giản, có thể sử dụng đi sử dụng lại, nó xác định các thuộc tính của một số điện thoại.

8.5.1 Bảng tổng kết thuộc tính

Thuộc tính	Kiểu dữ liệu	Yêu cầu	Giá trị mặc định	Được quy định bởi	Khả năng thay đổi
areaCode	Chuỗi 4	Không		<i>Máy client</i>	Có
countryCode	Chuỗi 4	Không		<i>Máy client</i>	Có
extension	Chuỗi 8	Không		<i>Máy client</i>	Có
number	Chuỗi 19	Không		<i>Máy client</i>	Có
phonetype	Chuỗi 32	Không		<i>Máy client</i>	Có
url	URI	Không		<i>Máy client</i>	Có

8.5.2 Thuộc tính **areaCode** (Mã vùng)

Mỗi số điện thoại có thể có một thuộc tính mã vùng cung cấp mã vùng cho số điện thoại.

8.5.3 Thuộc tính **countryCode** (Mã quốc gia)

Mỗi số điện thoại có thể có thuộc tính **countryCode** (mã quốc gia) cung cấp mã đất nước cho số điện thoại.

8.5.4 Thuộc tính **extension** (Số máy lẻ)

Mỗi số thuê bao có thể có thuộc tính **extension** (số máy lẻ) cung cấp số máy lẻ, nếu có, cho số điện thoại thuê bao.

8.5.5 Thuộc tính number (Số)

Mỗi *trường hợp* số điện thoại thuê bao có thể có một thuộc tính số máy cung cấp số nội bộ (mà không có mã khu vực, mã đất nước và số máy lẻ) cho số điện thoại thuê bao đó.

8.5.6 Thuộc tính phonetype (Loại điện thoại)

Mỗi *trường hợp* số điện thoại thuê bao có thể có thuộc tính phoneType cung cấp số điện thoại thuê bao. Một số ví dụ về loại điện thoại là "điện thoại gia đình", "điện thoại văn phòng".

8.6 Lớp EmailAddress (Địa chỉ hòm thư điện tử)

Là một lớp *thực thể* đơn giản, có thể sử dụng đi sử dụng lại, xác định các thuộc tính của một địa chỉ hòm thư điện tử.

8.6.1 Bảng tổng kết các thuộc tính

Thuộc tính	Kiểu dữ liệu	Yêu cầu	Giá trị mặc định	Được quy định bởi	Có thể thay đổi
address	Tên viết tắt	Có		<i>Máy client</i>	Có
type	Chuỗi 32	Không		<i>Máy client</i>	Có

8.6.2 Thuộc tính address (Địa chỉ)

Mỗi địa chỉ hòm thư điện tử phải có một thuộc tính address (địa chỉ) cung cấp địa chỉ hòm thư điện tử thực sự.

8.6.3 Thuộc tính type (Kiểu)

Mỗi *trường hợp* địa chỉ hòm thư điện tử có thể có một thuộc tính type (kiểu) cung cấp loại địa chỉ hòm thư điện tử. Đây là giá trị rời rạc, ngẫu nhiên. Các ví dụ về các kiểu hòm thư là "Gia đình", "Công việc".

8.7 Lớp personName (Tên cá nhân)

Là một lớp *thực thể* dành cho tên của một người nào đó.

8.7.1 Bảng tóm tắt thuộc tính

Thuộc tính	Kiểu dữ liệu	Yêu cầu	Giá trị mặc định	Được quy định bởi	Có thể thay đổi
firstName	ShortName	Không		<i>Máy client</i>	Có
lastName	ShortName	Không		<i>Máy client</i>	Có
middleName	ShortName	Không		<i>Máy client</i>	Có

8.7.2 Thuộc tính firstName (Tên gọi)

Mỗi PersonName có thể có một thuộc tính firstName (tên gọi) là tên thường gọi của một người nào đó.

8.7.3 Thuộc tính lastName (Tên họ)

Mỗi PersonName có thể có một thuộc tính lastName (tên họ) của một người nào đó.

8.7.4 Thuộc tính middleName (Tên đệm)

Mỗi PersonName có thuộc tính middleName là tên đệm của một người.

8.8 Lớp Service (Dịch vụ)

Các lớp chuẩn:

RegistryEntry (mục nhập đăng ký), **RegistryObject** (đối tượng đăng ký)

Các trường hợp của Service (dịch vụ) cung cấp thông tin về các service (dịch vụ) chẳng hạn như các web service (các dịch vụ web – Ứng dụng web).

8.8.1 Tóm tắt các thuộc tính

Lớp Service (dịch vụ) chưa được biết bất cứ thuộc tính cụ thể nào ngoài các thuộc tính cố định của nó.

8.8.2 Tóm tắt phương pháp

Bên cạnh các thuộc tính của nó, lớp Service (dịch vụ) cũng xác định các phương pháp sau đây.

Tóm tắt phương pháp thực thi của dịch vụ	
Tập hợp	<u>getServiceBindings ()</u> Lấy tập hợp các service cố định được xác định đối với dịch vụ này.

8.9 Lớp Service (dịch vụ) Bindings

Các lớp chuẩn:

RegistryObject

Các trường hợp của ServiceBindings (quy định dịch vụ) là các RegistryObject đại diện cho thông tin kỹ thuật về cách xác định truy cập một giao diện cụ thể do một trường hợp của lớp Service (dịch vụ) cung cấp. Một Service (dịch vụ) có một tập hợp các ServiceBinding (quy định dịch vụ).

Thuộc tính mô tả ServiceBinding cung cấp các chi tiết về mối quan hệ giữa một vài đường liên kết xác định tạo ra ServiceBinding. Sự mô tả có thể sẽ rất hữu ích đối với sự hiểu biết của con người. Có khả năng đẩy mạnh một cấu trúc về quá trình mô tả này để cho phép máy móc xử lý quá trình ServiceBinding, tuy nhiên quá trình này lại không được giải quyết trong tiêu chuẩn này.

8.9.1 Bảng tóm tắt các thuộc tính

Thuộc tính	Kiểu dữ liệu	Yêu cầu	Giá trị mặc định	Được quy định bởi	Có thể thay đổi
accessURI	URI	Không		Máy client	Có
targetBinding	UUID	Không		Máy client	Có

8.9.2 Thuộc tính accessURI (Uri truy cập)

ServiceBinding có thể có một Thuộc tính accessURI (uri truy cập) để có thể truy cập dịch vụ kết nối đó. Thuộc tính này sẽ bị bỏ qua nếu một thuộc tính mục tiêu kết nối là dành riêng cho *ServiceBinding*. Nếu URI là một URL thì RegistryObject phải xác định rõ URL vào thời điểm đệ trình trước khi chấp nhận *ServiceBinding* vào sổ đăng ký.

8.9.3 Thuộc tính targetBinding (Quy định bên đích)

Một *ServiceBinding* có thể có một Thuộc tính targetBinding (quy định bên đích) được xác định và đây được coi là *phần tử* tham chiếu một *ServiceBinding* khác. Một targetBinding (quy định bên đích) có thể được xác định khi một *service* (dịch vụ) được chuyển hướng sang một *service* (dịch vụ) khác. Điều này cho phép thay đổi máy chủ của dịch vụ bằng một nhà cung cấp dịch vụ khác.

8.9.4 Bảng tóm tắt các phương pháp

Bên cạnh các thuộc tính của nó, tuyến *ServiceBinding* cũng xác định các phương pháp sau đây

Tóm tắt phương pháp thực thi của <i>ServiceBinding</i>	
Tập hợp	<u>getSpecificationLinks ()</u> Tiếp nhận tập hợp của các đường liên kết được xác định rõ cho quá trình kết nối dịch vụ này.

8.10 Lớp SpecificationLinks (Liên kết quy định)

Các lớp chuẩn:

RegistryObject (đối tượng đăng ký)

SpecificationLinks cung cấp sự liên kết giữa dịch vụ kết nối và một trong các thông số kỹ thuật cụ thể của nó trong đó mô tả phương pháp cũng như cách thức sử dụng dịch vụ kết nối cố định. Chẳng hạn như, *ServiceBinding* có thể có một *SpecificationLink* trong đó mô tả cách thức truy cập dịch vụ có sử dụng các thông số kỹ thuật được nêu dưới dạng văn bản WSDL hoặc văn bản CORBA IDL.

8.10.1 Bảng tóm tắt các thuộc tính

Thuộc tính	Kiểu dữ liệu	Yêu cầu	Giá trị mặc định	Được quy định bởi	Có thể thay đổi
specificationObject	UUID	Có		<i>Máy client</i>	Có
usageDiscription	InternationalString	Không		<i>Máy client</i>	Có
usageParrameters	Tập hợp các văn bản dưới dạng tự do	Không		<i>Máy client</i>	Có

8.10.2 Thuộc tính specificationObject (Quy định đối tượng)

Trường hợp *SpecificationLink* phải có một Thuộc tính *specificationObject* (quy định đối tượng) có vai trò cung cấp 4 tham chiếu cho *trường hợp RegistryObject* (đối tượng đăng ký) và cung cấp các thông số quy định kỹ thuật cho dịch vụ kết nối chủ. Thông thường đây là *trường hợp* đối tượng ngoại lai đại diện cho các thông số kỹ thuật (chẳng hạn như một văn bản dưới dạng WSDL).

8.10.3 Thuộc tính usageDescription (Mô tả sử dụng)

Một *trường hợp* của *SpecificationLink* có thể có một thuộc tính *usageDescription* có vai trò cung cấp sự mô tả dưới dạng văn bản về cách thức, phương pháp sử dụng thuộc tính *usageParameters* (tham số sử dụng) lựa chọn tiếp theo. *UsageDescription* là một dạng *internationalString*, chính vì vậy cho phép có thể mô tả cách sử dụng ở dạng nhiều ngôn ngữ khác nhau.

8.10.4 Thuộc tính usageParameters (Tham số sử dụng)

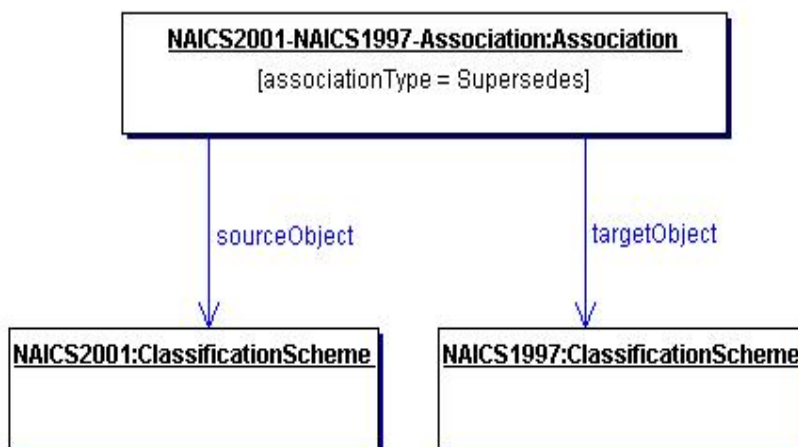
Trường hợp *SpecificationLink* có thể có một thuộc tính *usageParameters* (tham số sử dụng) cung cấp một tập hợp các đường đại diện cho *usageParameters* (tham số sử dụng) cần thiết cho việc sử dụng các thông số quy định kỹ thuật (chẳng hạn như văn bản dưới dạng WSDL) do đối tượng *SpecificationLink* này xác định.

9 Phương thức liên kết của RegistryObject (Đối tượng đăng ký)

Một *RegistryObject* (RO) có thể không liên kết hoặc có liên kết với nhiều *RegistryObject* khác. Mẫu dữ liệu sẽ xác định một *lớp Association* (liên kết) để liên kết hai RO bất kỳ.

9.1 Ví dụ về Association (Liên kết)

Một ví dụ phương thức liên kết xảy ra giữa hai giản đồ phân loại (*ClassificationScheme* (giản đồ phân loại)). Trong ví dụ một *ClassificationScheme* (giản đồ phân loại) sẽ thay thế một CS khác như trong hình minh hoạ số 3. *trường hợp* này có thể rơi vào khi đưa ra một phiên bản mới của SC. Trong hình 3, chúng ta thấy được cách xác định một *Association* giữa một phiên bản CS mới và một phiên bản cũ hơn của NAICS.



Hình 3 – Ví dụ về liên kết của RO

9.2 Đối tượng Nguồn và đối tượng Đích

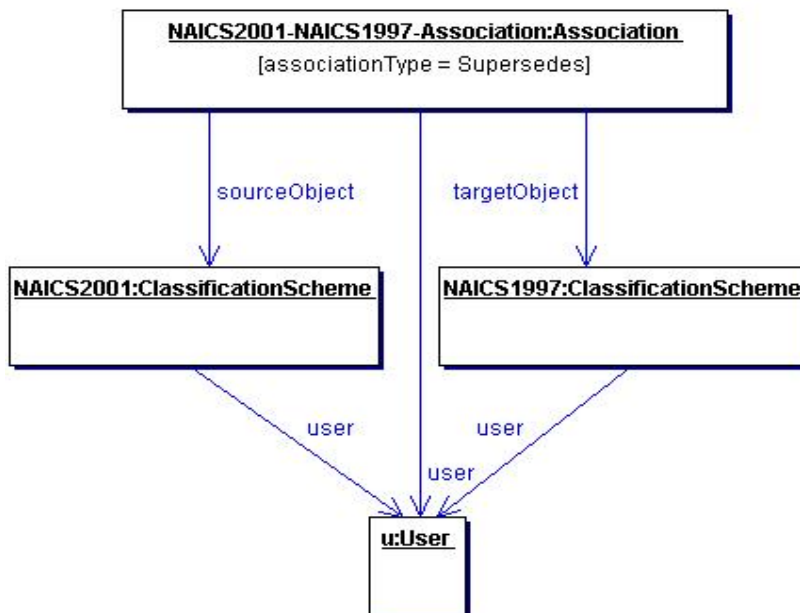
Một Association biểu diễn liên kết giữa một *RO nguồn* và một *RO đích* và được gọi là một *sourceObject* và *targetObject* của một *trường hợp Association*. Điều quan trọng là phải phân biệt được đâu là đối tượng nguồn và đâu là đối tượng đích bởi vì điều này sẽ mang tính chất quyết định đến ngữ nghĩa định hướng của một liên kết. Ví dụ ở Hình 3, một phiên bản mới CS của NAICS trở thành đối tượng nguồn còn phiên bản CS cũ của NAICS thành đối tượng đích. Chính Type (loại) liên kết đã chỉ định rằng cần phải thay thế đối tượng nguồn bằng đối tượng đích chứ không phải ngược lại.

9.3 Các kiểu liên kết

Mỗi Association (liên kết) phải mang một thuộc tính *associationType* (kiểu liên kết) để xác định loại của liên kết đó.

9.4 Liên kết trong (Intramural)

Đối với *trường hợp* sử dụng cho *lớp* liên kết thông thường thì khi một User (người sử dụng) “u” tạo ra một Association (liên kết) “a” giữa 2 RO “o1” và “o2”. Trong đó “a” và “o1” “o2” là hai objectType (kiểu đối tượng) do cùng một User (người sử dụng) tạo ra. *trường hợp* liên kết này là cho mục đích sử dụng đơn giản nhất được thực thi giữa 2 đối tượng bởi một User (người sử dụng) – Đây chính là định nghĩa về liên kết RegistryObject (đối tượng đăng ký). Các liên kết tương tự như vậy đều được gọi là liên kết Trong hay liên kết Nội vùng. Hình số 4 dưới đây là phần mở rộng của liên kết nội vùng của ví dụ hình số trước đó.

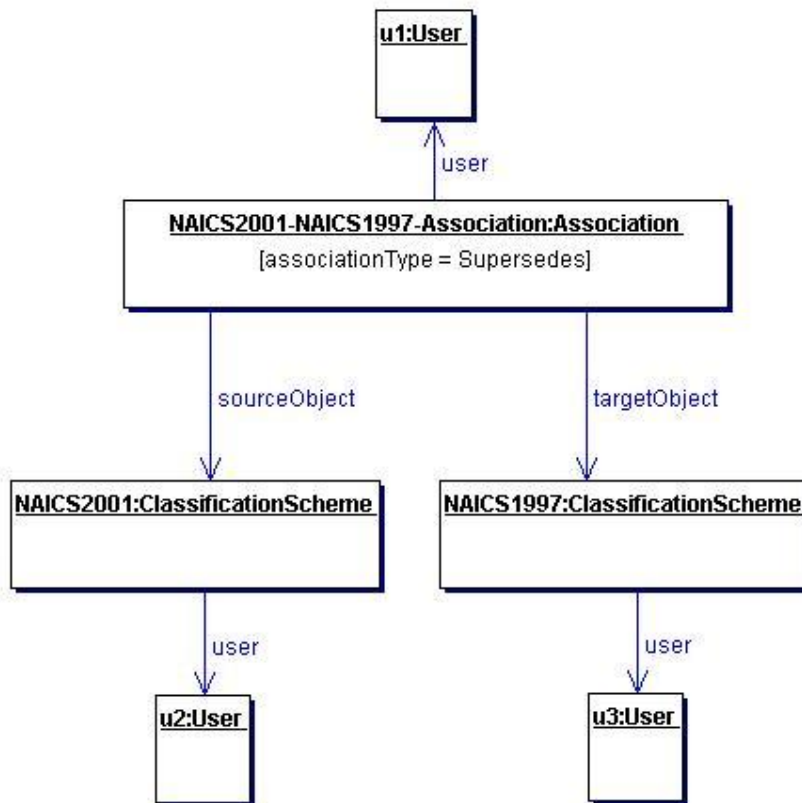


Hình 4 – Đuôi mở rộng của liên kết nội vùng trong ví dụ hình 3

9.5 ExternalLink (Liên kết ngoại)

Mô hình thông tin cũng cho phép thực thi các liên kết tinh vi hơn.

Ví dụ một User (người sử dụng) “u1” tạo ra một liên kết “a” giữa 2 RegistryObject là “o1” và “o2” trong đó liên kết “a” thuộc về User “u1” nhưng RegistryObject “o1” và “o2” lại thuộc về User “u2” và User “u3”. Trong trường hợp sử dụng này một liên kết được xác lập tới từng đối tượng hoặc cả hai mà các liên kết này lại thuộc quyền sở hữu của một User (người sử dụng) khác chính là User đã tạo ra liên kết ban đầu. Các liên kết như vậy được gọi là ExternalLink. Lớp liên kết đưa ra một phương pháp thuyết phục là *isExtranual()*. Phương pháp này sẽ trả lại giá trị đúng nếu trường hợp liên kết này là một ExternalLink. Hình số 5 dưới đây là phần mở rộng cho ví dụ ở hình vẽ số 3 về trường hợp ExternalLink. Lưu ý rằng một ExternalLink có thể có 2 User khác nhau chứ không phải là 3 User như trong hình số 5. Trong trường hợp như vậy, một trong hai User sẽ có 2 trong 3 đối tượng tham gia bao gồm Association, SourceObject và TargetObject



Hình 5 – Ví dụ về ExternalLink (liên kết ngoại)

9.6 Xác nhận một liên kết

Một liên kết cần phải được một nhóm đối tượng tham gia vào liên kết đó xác nhận như là đối tượng nguồn hoặc đối tượng đích. Mục tiếp theo sẽ mô tả ý nghĩa của một liên kết của nhóm đối tượng tham gia.

9.6.1 Xác nhận liên kết trong

Các liên kết trong được công bố bởi chứng thực và không yêu cầu thêm bất kỳ bước giải thích rõ ràng nào để xác nhận rằng liên kết đang ở giá trị thực (có liên kết). Điều đó có nghĩa các liên kết trong đã hoàn toàn được xác nhận.

9.6.2 Xác nhận các ExternalLink (Liên kết ngoài)

ExternalLink được cho là đã xác nhận đơn phương mà không được xem như giá trị “đúng” cho tới khi nó được các nhóm đối tượng khác tham gia xác nhận (ví dụ, User “u2” và “u3” ở ví dụ mục 9.5)

Để xác nhận một ExternalLink mỗi một nhóm đối tượng ExternalLink (nhóm đối tượng chứa đối tượng đích hoặc đối tượng nguồn nhưng không chứa liên kết) Phải được đưa ra một liên kết xác định (liên kết nhân bản) như là một liên kết mà chúng cần để xác nhận một yêu cầu nộp đối tượng (SubmitterObjectRequest). Liên kết nhân bản phải có cùng Id với liên kết gốc.

9.7 Khả năng hiển thị các liên kết không được xác nhận

Các ExternalLink đòi hỏi mỗi bên phải xác nhận việc chèn thông tin vào thông qua ExternalLink trước khi các bên thứ 3 không tham gia trong liên kết có thể quan sát được liên kết. Điều này đảm bảo rằng các liên kết không được xác nhận sẽ không hiển thị cho máy client đăng ký thuộc bên thứ 3.

9.8 Các trạng thái xác nhận có thể xảy ra

Giả sử có *trường hợp* thông dụng nhất là có 3 *trường hợp* User (người sử dụng) cụ thể như thể hiện ở hình 5. ExternalLink cần phải được xác nhận bởi cả hai bên sử dụng còn lại là u2 và u3 để có được sự xác nhận đầy đủ. Phương pháp *isConfirmedBySourceOwner* và *isConfirmedByTargetOwner* trong lớp liên kết cho phép truy nhập tới tình trạng xác nhận của cả sourceObject và targetObject. Phương pháp thứ ba gọi là *isConfirmed* cho phép thêm một cách để xác định liên kết đã được xác nhận đầy đủ hay chưa. Vì vậy, có bốn khả năng liên quan tới tình trạng khẳng định của một ExternalLink.

- liên kết không được xác nhận bởi người chủ của sourceObject hoặc targetObject;
- liên kết được xác nhận bởi người chủ của sourceObject nhưng không được xác nhận bởi targetObject;
- liên kết không được xác nhận bởi người chủ của sourceObject nhưng được xác nhận bởi người chủ của targetObject;
- liên kết được xác nhận bởi cả người chủ của sourceObject và targetObject. Đây là trạng thái duy nhất khi mà liên kết được xác nhận đầy đủ.

9.9 Lớp Association (Liên kết)

Các lớp chuẩn:

RegistryObject

TCVN ISO/TS 15000-3 : 2007

Các trường hợp Association được sử dụng để xác định các đa liên kết giữa các RO với nhau trong một mô hình thông tin.

Một trường hợp của lớp Association thể hiện một liên kết giữa hai RO.

9.9.1 Tóm tắt các thuộc tính

Thuộc tính	Kiểu dữ liệu	Có được yêu cầu	Giá trị mặc định	Được xác định bởi	Khả biến
associationType	LongName	Có		Máy khách	Không
sourceObject	UUID	Có		Máy client	Không
targetObject	UUID	Có		Máy client	Không

9.9.2 Thuộc tính associationType (Kiểu liên kết)

Mỗi Association cần phải có một thuộc tính associationType (kiểu liên kết) xác định loại cho liên kết đó.

9.9.2.1 Các kiểu liên kết tiên xác định

Bảng dưới đây liệt kê các kiểu liên kết tiên xác định. Các kiểu liên kết tiên xác định này được định nghĩa là một ClassificationScheme (giản đồ phân loại). Có thể một scheme (giản đồ) dễ dàng được mở rộng nhưng một Registry cần phải hỗ trợ các kiểu liên kết dưới đây:

Tên	Mô tả
<i>RelatedTo</i>	Xác định rằng RO nguồn có liên quan tới RO đích
<i>HasMember</i>	Xác định rằng đối tượng RegistryPackage nguồn lấy đối tượng RO đích làm thành viên. Được sử dụng trong việc đóng gói (Packaging) các RegistryEntries
<i>ExternallyLinks</i>	Xác định rằng đối tượng ExternalLink nguồn có ExternalLink với đối tượng RO đích. Sử dụng trong việc liên kết các ExternalLinks với RegistryEntries.
<i>Contains</i>	Xác định rằng RO nguồn có chứa RO đích. Chi tiết quan hệ bao hàm này xác định cụ thể giá trị sử dụng. Ví dụ, một catalog các phụ kiện có thể xác định một đối tượng Bộ máy có mối quan hệ với đối tượng bộ truyền lực
<i>EquivalentTo</i>	Xác định rằng RO nguồn tương đương với RO đích
<i>Extends</i>	Xác định rằng RO nguồn thừa kế từ hoặc là đặc biệt hóa RO đích
<i>Implements</i>	Xác định rằng RO nguồn thực thi chức năng được xác định bởi RO đích
<i>InstanceOf</i>	Xác định rằng RO nguồn là một trường hợp của RO đích
<i>Supersedes</i>	Xác định rằng RO nguồn vượt qua RO đích
<i>Uses</i>	Xác định rằng RO nguồn sử dụng RO đích ở một mức độ nào đó
<i>Replaces</i>	Xác định rằng RO nguồn thay thế RO đích ở một mức độ nào đó
<i>SubmitterOf</i>	Xác định rằng Organization nguồn là nơi đệ trình RO đích
<i>ResponsibleFor</i>	Xác định rằng Organization nguồn có trách nhiệm duy trì RO đích

9.9.3 Thuộc tính sourceObject (Đối tượng nguồn)

Mỗi Association cần phải có một thuộc tính sourceObject (đối tượng nguồn) tham chiếu tới trường hợp RO là nguồn của liên kết.

9.9.4 Thuộc tính `targetObject` (Đối tượng đích)

Mỗi Association cần phải có một thuộc tính `targetObject` (đối tượng đích) tham chiếu tới *trường hợp* RO là đích của liên kết.

Tóm tắt phương pháp của Association	
boolean	<code>isConfirmed()</code> trả về True nếu cả <code>isConfirmedBySourceOwner</code> và <code>isConfirmedByTargetOwner</code> đều trả về giá trị True. Đối với các liên nội, luôn trả về giá trị là True. một liên kết chỉ nên hiển thị cho bên thứ 3 (không liên quan tới Association đó) nếu <code>isConfirmed</code> trả về giá trị True
boolean	<code>isConfirmedBySourceOwner()</code> Trả về giá trị True nếu liên kết được xác nhận bởi người chủ của <code>sourceObject</code> . Với các Association nội bộ, luôn trả về giá trị True
boolean	<code>isConfirmedByTargetOwner()</code> Trả về giá trị True nếu liên kết được xác nhận bởi <code>targetObject</code> . Với các Association nội, luôn trả về giá trị True
boolean	<code>isExtramural ()</code> Trả về giá trị True nếu <code>sourceObject</code> và/hoặc <code>targetObject</code> có một User (người sử dụng) sở hữu khác với User tạo ra Association

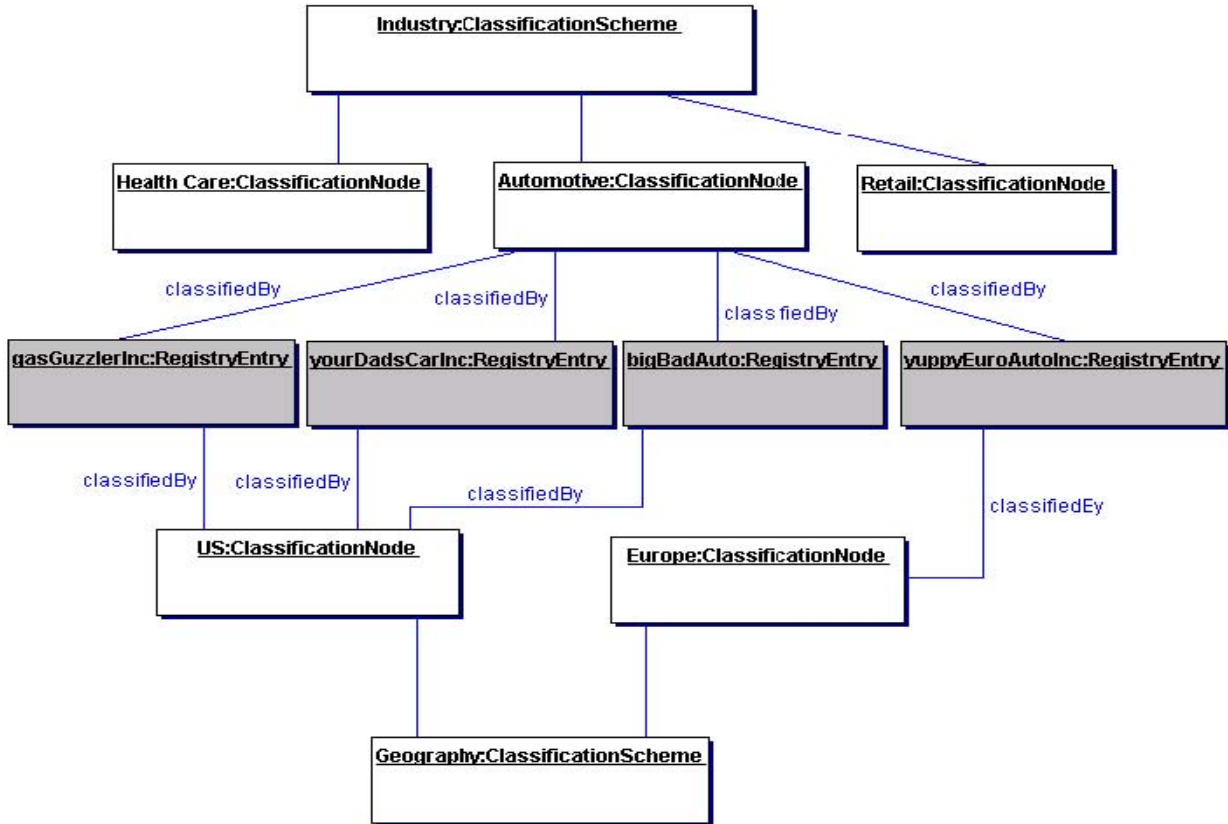
10 Phân loại RegistryObject (Đối tượng đăng ký)

Phần này mô tả cách thức mô hình thông tin hỗ trợ việc Phân loại RegistryObject (đối tượng đăng ký). Nó là phiên bản đơn giản hóa của mô hình phân loại OASIS (OAS).

Một RO có thể được phân loại theo nhiều cách. ví dụ, RO cho hồ sơ giao thức hợp tác (Collaboration Protocol Profile (CPP)) có thể được phân loại theo ngành công nghiệp của nó, theo các sản phẩm và theo vị trí địa lý của nó.

Một ClassificationScheme (giản đồ phân loại) chung có thể được xem là một cây phân loại. Trong ví dụ ở hình 6, các *trường hợp* RO thể hiện CPP được thể hiện dạng các hộp mờ. Mỗi CPP thể hiện một nhà sản xuất ô tô. Mỗi CPP được phân loại bởi ClassificationNode gọi là “Automotive” dưới ClassificationScheme (giản đồ phân loại) tên là Geography (địa lý). Tương tự, một nhà sản xuất ô tô ở Châu Âu được phân loại bởi ClassificationNode “Europe” dưới cái ClassificationNode mang tên Geography (địa lý).

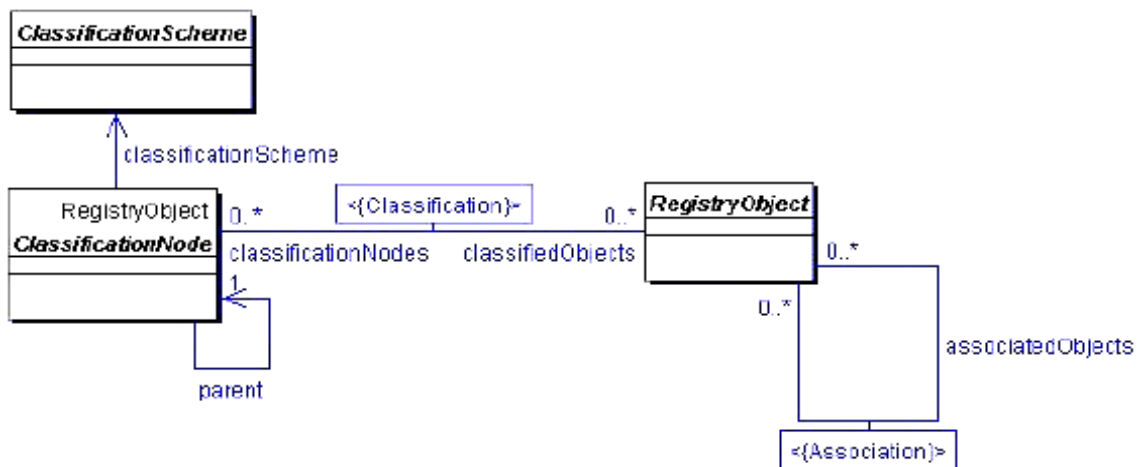
Ví dụ thể hiện RO được phân loại thế nào bởi nhiều các *trường hợp* ClassificationNode (nút phân loại) khác nhau dưới nhiều *trường hợp* ClassificationScheme khác nhau (Chẳng hạn như ngành địa lý).



Hình 6 – Ví dụ minh họa về cây phân loại

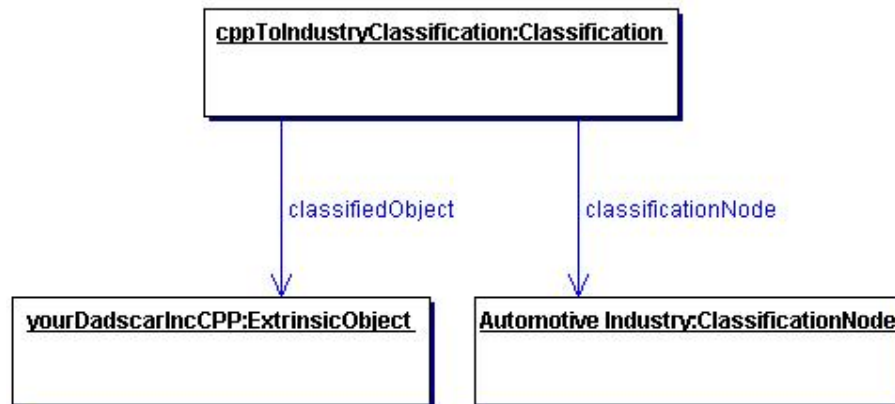
Chú ý: Các nút đậm (gas GuzzlerInc, YourDadCarInc, etc) không phải là một phần của cây phân loại. Các nút của cây phân loại bao gồm: ngành y tế, ngành tự động hoá, ngành bán lẻ, Mĩ, và Châu Âu. Các nút đậm kết hợp với cây phân loại thông qua ví dụ phân loại không được minh họa trong hình vẽ.

Để hiểu được giản đồ phân loại nói chung hỗ trợ cho quá trình phân loại cấp độ đơn và đa cấp độ, mô hình thông tin về các kiểu phân loại và mối quan hệ giữa chúng được minh họa trong hình 7



Hình 7 – Hình ảnh mô hình thông tin về phân loại

Phân loại là một dạng đặc biệt của quá trình liên kết. Hình 8 minh họa ví dụ về đối tượng khác với đối tượng trong bản báo cáo dự thảo hợp tác được phân loại dựa vào *classificationNode* mà đối tượng đó thuộc về.



Hình 8 – Sơ đồ ví dụ phân loại

10.1 Lớp classificationScheme (Giản đồ phân loại)

Các lớp chuẩn: **RegistryEntry** (nơi đăng ký), **RegistryObject** (đối tượng đăng ký).

Ví dụ: Giản đồ minh họa bao gồm nguồn dữ liệu mô tả khoa phân loại đã được đăng ký. Hệ thống khoa phân loại được xác định ở bên trong tới nơi đăng ký thông qua ví dụ về các *classificationNode* hoặc có thể được xác định bên ngoài nơi đăng ký trong trường hợp nơi đăng ký không biết được cấu trúc cũng như giá trị của các phần tử khoa phân loại.

10.1.1 Tóm tắt các thuộc tính

Thuộc tính	Kiểu dữ liệu	Yêu cầu	Giá trị mặc định	Được quy định bởi	Khả năng thay đổi
isinternal	Boolean	Có		Máy client	Không
ClassificationNode	String 32	Có		Máy client	Không

Chú ý: Các thuộc tính của giản đồ phân loại theo lớp từ nơi đăng kí không được trình bày

10.1.2 Thuộc tính isInternal (Nội bộ)

Khi đệ trình một giản đồ phân loại, Organization tham gia vào quá trình phân loại phải báo cáo giản đồ phân loại là thuộc dạng phân loại bên trong hay bên ngoài. Điều này cho phép nơi đăng kí có thể phê chuẩn các đệ trình tiếp theo của classificationNode và ví dụ phân loại nhằm mục đích duy trì tính thống nhất của giản đồ phân loại trong suốt quá trình phân loại.

10.1.3 Thuộc tính classificationNode (Nút phân loại)

Khi đệ trình một giản đồ phân loại, Organization tham gia vào quá trình phân loại phải báo cáo cấu trúc classificationNode mà giản đồ phân loại sẽ áp dụng. Dưới đây là bản liệt kê giá trị của thuộc tính này:

- mật mã duy nhất: Giá trị này cho biết mỗi classificationNode đều có một mật mã riêng;
- đường dẫn được gắn vào: Giá trị này cho biết mỗi mật mã tương ứng với một classificationNode sẽ giải mã đường đi của nó. Đây chính là thuộc tính của khoa phân loại NAICS;
- nonuniquecode: Trong một số trường hợp ClassificationNode (nút phân loại) không phải là duy nhất và cần thiết phải định vị đường dẫn đầy đủ để xác định classificationNode. Chẳng hạn như: Trong khoa học phân loại địa lý Matxcova nằm dưới Nga và Mỹ nơi có năm thành phố với tên gọi các bang khác nhau.

Dữ liệu liệt kê này có thể sẽ mở rộng với một vài giá trị mới chẳng hạn như: Giá trị có thể của dữ liệu này có thể là NamedPathElements (các thành phần đường dẫn được đặt tên).

10.2 Lớp classificationNode (Nút phân loại)

Các lớp chuẩn:

RegistryObject

Các trường hợp ClassificationNode (nút phân loại) được sử dụng để xác định cấu trúc dạng cây mà mỗi nút ở cấu trúc cây này là một ClassificationNode (nút phân loại). Các cây Classification (phân loại) được cấu trúc bởi các trường hợp ClassificationNode (nút phân loại) dưới dạng một ClassificationScheme (giản đồ phân loại) (lược đồ phân loại) và được sử dụng để xác định các giản đồ Classification (phân loại) hoặc bản thể học.

10.2.1 Tóm tắt thuộc tính

Thuộc tính	kiểu dữ liệu	yêu cầu	Giá trị mặc định	Quy định bởi	Khả năng thay đổi
Parent	UUID	No		Máy client	No
Code	ShortName	No		Máy client	No

10.2.2 Thuộc tính parent (Gốc)

Mỗi *classificationNode* có thể có một thuộc tính parent (gốc). Thuộc tính parent (gốc) có thể tham chiếu một *classificationNode* gốc hoặc một trường hợp giản đồ phân loại đối với *classificationNode* cấp độ đầu tiên.

10.2.3 Thuộc tính code (Mã)

Mỗi *classificationNode* có thể có một thuộc tính code (mã). một Thuộc tính code (mã) bao gồm một mã với một giản đồ mã hóa chuẩn.

10.2.4 Tóm tắt phương pháp

Ngoài các thuộc tính, *classificationNode* cũng có thể xác định theo các phương pháp sau đây:

Tóm tắt phương pháp *classificationNode*

ClassificationScheme	getClassificationScheme () Dùng ClassificationScheme (giản đồ phân loại) mà có <i>classificationNode</i>
Collection	getClassifiedObjects() Dùng tập hợp RegistryObjects được phân loại bởi <i>classificationNode</i> này
String	getPath() Dùng đường dẫn chuẩn từ classificationScheme của <i>classificationNode</i> này. Cú pháp đường dẫn được xác định ở trong mục 10.2.5
Integer	getLevelNumber() Dùng cấp độ số của <i>classificationNode</i> trong giản đồ phân cấp phân loại. Phương pháp này trả về một dãy Integer dương và được xác nhận cho mỗi trường hợp ClassificationNode (nút phân loại)

Trong Hình 6, một số trường hợp ClassificationNode (nút phân loại) được xác định(tất cả các hộp màu sáng). Một *classificationNode* thì có giá trị bằng 0 hoặc giá trị gốc hoặc có nhiều giá trị con tức thời. Gốc của *classificationNode* có thể là *classificationNode* khác hoặc classificationSchem trong trường hợp ClassificationNode (nút phân loại) ở cấp độ I.

10.2.5 Cú pháp đường dẫn chuẩn

Phương pháp getPart của Lớp *classificationNode* (nút phân loại) trả về đường dẫn tuyệt đối trong biểu diễn chuẩn xác định đường dẫn từ giản đồ phân loại đến classificationNode.

Biểu diễn đường dẫn chuẩn được xác định bởi các lệnh BNF dưới đây:

```
canonicalPath ::= '/' schemeId nodePath
nodePath ::= '/' nodeCode
| '/' nodeCode ( nodePath )?
```

TCVN ISO/TS 15000-3 : 2007

Trong các lệnh trên, tạo giản đồ chính là thuộc tính của giản đồ phân loại và nút mã hoá được xác định bởi sản phẩm NCName từ trang Web:

<http://www.w3.org/TR/REC-xml-names/#NT-NCName>

10.3 Phân loại lớp

Các lớp chuẩn:

RegistryObject

Với một cách thức phân loại cụ thể sẽ phân loại một RegistryObject cụ thể dựa vào các hệ số tham chiếu của nút được xác định theo một giản đồ phân loại đặc thù nào đó.

Một hệ phân loại nội tuyến thường trực tiếp chỉ ra các thông số của nút nhờ vào mã id của nó trong khi phân loại ngoại tuyến chỉ gián tiếp đưa ra thông số của nút nhờ vào việc xác định tập biểu diễn các giá trị (là cách thức duy nhất để xác định tham số ngoại tuyến) của nút đó.

Các thuộc tính và phương pháp phân loại lớp cho phép biểu diễn cả phân loại nội tuyến và ngoại tuyến nhờ đó có thể giản hóa bước chờ xét hay còn gọi là xếp hàng chờ phân biệt đâu là nội tuyến và ngoại tuyến.

Trong giản đồ 6, các trường hợp phân loại không chỉ ra các tham số ngoại tuyến mà chỉ hàm ẩn như một tham số liên hợp giữa các RegistryObject cụ thể (nút tối) và Nút phân loại liên kết.

10.3.1 Tổng hợp thuộc tính

Thuộc tính	Kiểu dữ liệu	Đối tượng	Giá trị mặc định	Xác định	Dao động
ClassificationScheme (giản đồ phân loại)	UUID	Phân loại ngoài	null	Máy client	Không
ClassificationNode	UUID	Phân loại trong	null	Máy client	Không
ClassifiedObject	UUID	Có		Máy client	Không
nodeRepresentation	(LongName)	Phân loại ngoài	null	Máy client	Không

Cần lưu ý bảng này không liệt kê đối với các thuộc tính kế thừa từ Các lớp chuẩn theo tiêu chí phân loại này.

10.3.2 Thuộc tính classificationScheme (Giản đồ phân loại)

Nếu một phân loại đối với một trường hợp cụ thể nào đó là phân loại ngoại tuyến thì cần có thêm bảng phân loại thuộc tính. Giá trị phân loại thuộc tính này phải đưa ra được các thông số tham chiếu đối với một mẫu phân loại cụ thể.

10.3.3 Thuộc tính classificationNode (Nút phân loại)

Nếu một phân loại đối với một trường hợp cụ thể nào đó là phân loại nội tuyến thì cần có thêm Thuộc tính classificationNode (nút phân loại). Giá trị của classificationNode này phải đưa ra được các thông số tham chiếu đối với một mẫu phân loại cụ thể.

10.3.4 Thuộc tính `classifiedObject` (Đối tượng được phân loại)

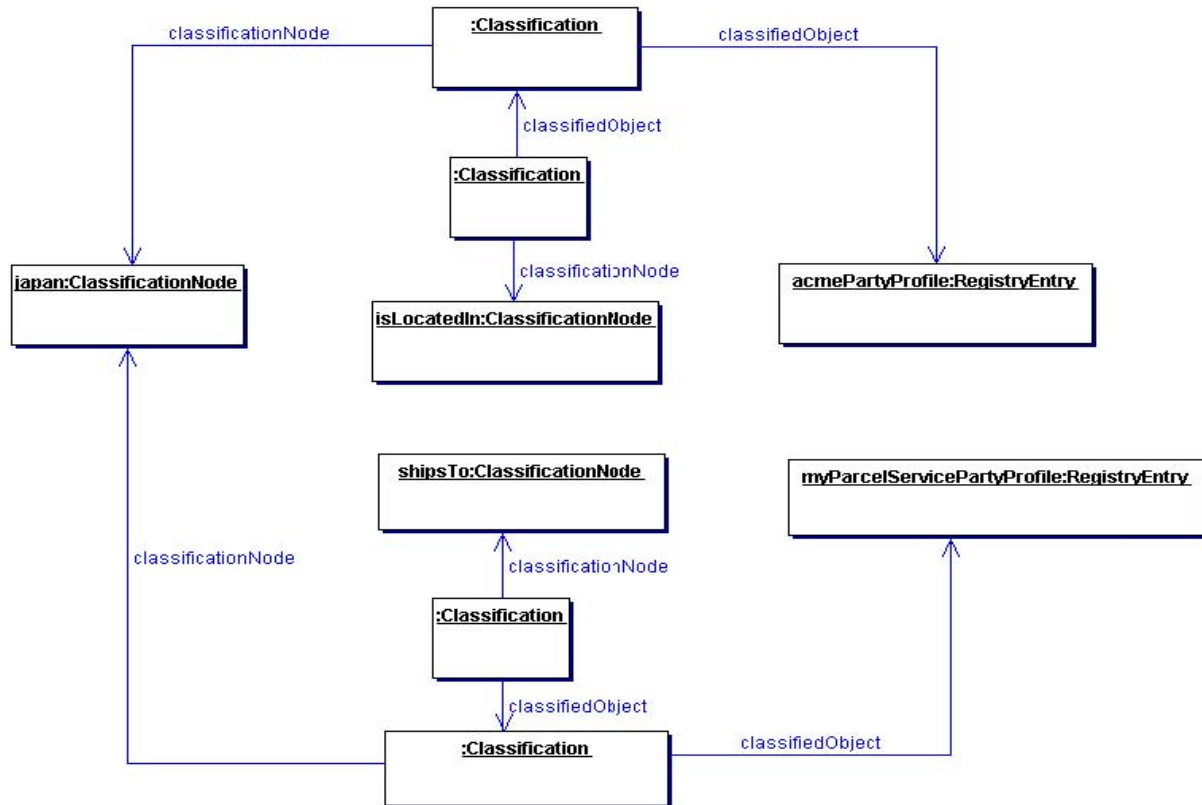
Cả kiểu phân loại nội tuyến và ngoại tuyến đều cần đến thuộc tính của mẫu được phân loại và thuộc tính này chính là *phần tử* biểu diễn các tham số của `RegistryObject` (đối tượng đăng ký) được phân loại theo tiêu chí phân loại này.

10.3.5 Thuộc tính `nodeRepresentation` (Biểu diễn nút)

Nếu một *trường hợp* của `Classification` là phân loại ngoại tuyến thì cần có thêm bảng phân loại thuộc tính. Đây chính là bảng biểu diễn các tiêu chí phân loại của một kiểu phân loại nhất định. Phân biệt sự khác biệt giữa các `nodeRepresentation` khác nhau là tác vụ của `Registry`, cũng giống như việc phân biệt mã `classificationNode` với đường chuẩn của `classificationNode` đó. Chính điều này cho phép *Máy client* được chính thức sử dụng các câu lệnh cú pháp khác nhau khi biểu diễn nút.

10.3.6 Phân loại theo ngữ cảnh

Chú ý *trường hợp* mô tả trong giản đồ 9 khi một hồ sơ dự thảo hợp tác với tập đoàn ACME Inc, được phân loại theo nút phân loại của Nhật Bản theo giản đồ phân loại khu vực địa lý. Trong *trường hợp* không đưa ra tiêu chí cụ thể của sự phân loại này thì rõ ràng ý nghĩa của việc phân loại này sẽ trở nên khó hiểu. Người ta có thể hiểu rằng hãng ACME này đóng ở Nhật Bản, hay hãng ACME đóng hàng sang Nhật Bản, hay phân loại như thế còn mang một ý nghĩa nào khác? Để tránh tình trạng hiểu lầm như vậy, một cách phân loại nào đó cần đi kèm với một `classificationNode` khác nữa (trong ví dụ này đó chính là tên địa danh của mẫu phân loại) làm nhiệm vụ bổ khuyết ngữ cảnh còn thiếu cho việc phân loại. Một hồ sơ dự thảo hợp tác khác của hãng MyParcelService (Dịch vụ vận chuyển bưu kiện) cũng có thể được phân loại theo `classificationNode` của Nhật Bản khi nó được phân loại có kèm theo một `classificationNode` khác (ví dụ như tên địa danh hàng được chuyển tới) để chỉ ra một thông tin khác thông tin mà ACME đã có.



Hình 9 – Phân loại theo ngữ cảnh

Do đó, để đưa ra được phương án phân loại khả dĩ trong nhiều trường hợp, một tiêu chí phân loại mà bản thân nó phải được phân loại theo một vài tiêu chí phân loại nào đó để ràng buộc tiêu chí phân loại thứ nhất với các *classificationNode* có nhiệm vụ làm rõ nghĩa ngữ cảnh còn khuyết. Nói tóm lại, đối với một mô hình thông tin, tiêu chí bổ sung tổng quan cho các mẫu phân loại sẽ cho phép:

- phân loại một RegistryObject theo sự phân loại nội tuyến kết hợp với một *classificationNode* trong một biểu mẫu phân loại;
- phân loại một RegistryObject theo sự phân loại ngoại tuyến kết hợp với một giá trị trong một biểu mẫu phân loại ngoại tuyến;
- phân loại một RegistryObject theo các khía cạnh khác nhau khi sử dụng các tiêu chí phân loại đa chiều kết hợp cùng các *classificationNode* đa chiều hay giá trị trong một biểu mẫu phân loại;
- một tiêu chí phân loại xác định đối với một RegistryObject cần đáp ứng được các điều kiện mà nó sẽ tiến hành phân loại.

10.3.7 Tổng kết phương pháp

Bên cạnh vấn đề về thuộc tính, lớp phân loại cũng cần xác định rõ tính chất của các phương pháp sau:

Kiểu trả về	Phương pháp
UUID	getClassificationScheme (giản đồ phân loại) () Đối với phân loại ngoại tuyến, cần sử dụng đến biểu mẫu đã xác định theo thuộc tính của biểu mẫu phân loại. Đối với phân loại nội tuyến, cần sử dụng đến biểu mẫu đã xác định theo cùng một phương pháp sử dụng ở nút phân loại cụ thể.
String	getPath() Đối với phân loại ngoại tuyến, cần sử dụng đến chuỗi có cấu trúc thích hợp được xác định theo kết quả từ phương pháp xác định đường dẫn sử dụng trong <i>classificationNode</i> cụ thể được xác định theo thuộc tính của nút phân loại.
ShortName	getCode() Đối với phân loại ngoại tuyến, cần sử dụng đến chuỗi biểu đạt giá trị của các tiêu chí phân loại. Do đó mà không cần thiết phải xác định nút đó nữa. Đối với phân loại nội tuyến, cần sử dụng giá trị thuộc tính mã lệnh của <i>classificationNode</i> được xác định theo thuộc tính của <i>classificationNode</i> đó.
Organization	getSubmittingOrganization() Xác định điều kiện của tập đoàn đệ trình mẫu nhập đăng ký. Phương pháp này sẽ đưa ra một kết quả khác 0 đối với mọi mẫu nhập đăng ký. Đối với một dự án được ưu tiên nào đó thì tập đoàn được bồi hoàn nhờ phương pháp này sẽ được xem là chủ sở hữu đối với trường hợp phân loại đó.

10.4 Ví dụ về biểu mẫu phân loại

Bảng sau đây sẽ đưa ra các ví dụ về các biểu mẫu phân loại được sử dụng cho một mô hình thông tin. Các biểu mẫu này được kiến thiết dựa trên một tập con trong toàn ngữ cảnh được Đội dự án nòng cốt và quá trình kinh doanh ebXML xác định. Bản thống kê này được minh họa không kèm diễn giải nào khác.

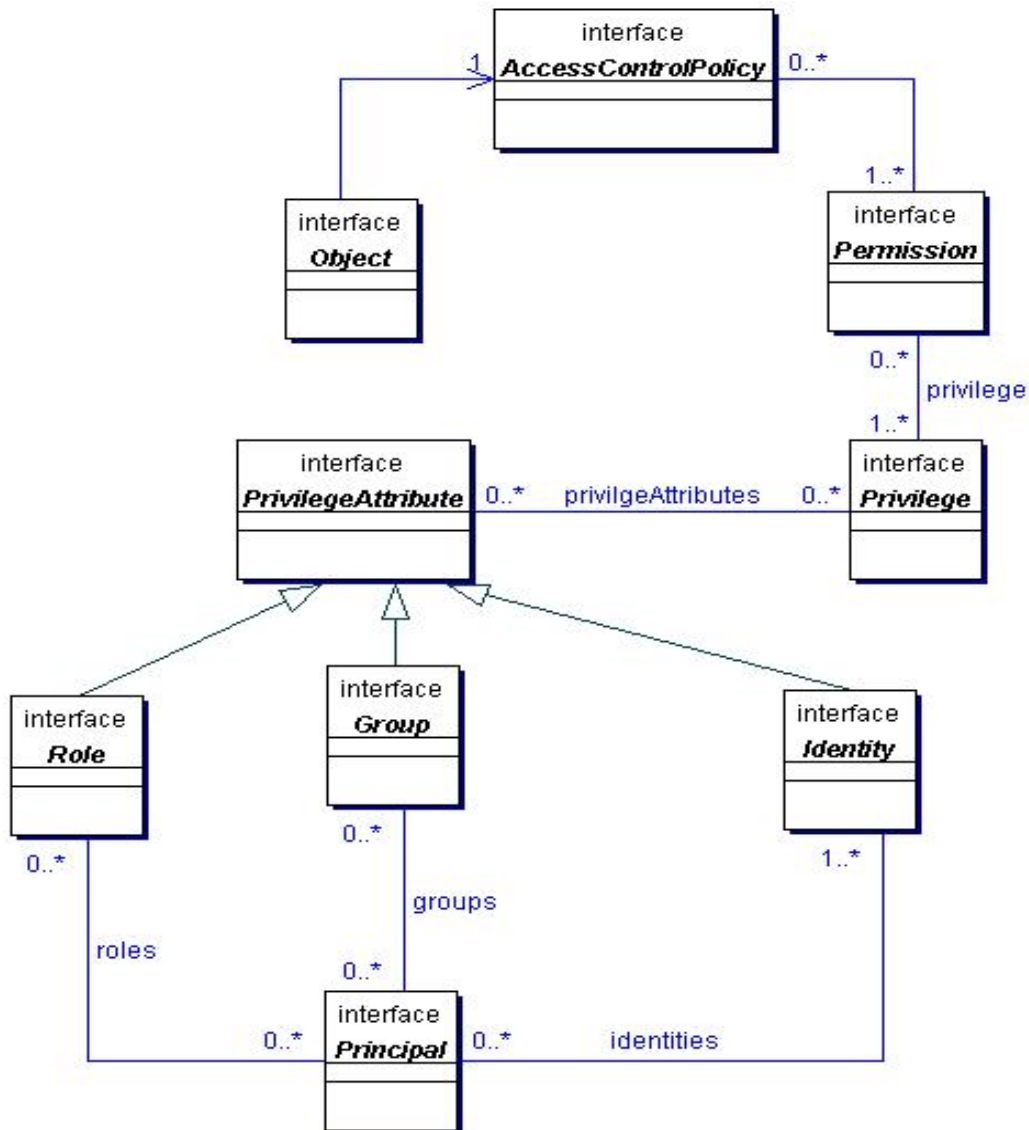
Mẫu phân loại	Ví dụ sử dụng	Mẫu phân Loại chuẩn
Công nghiệp	Tìm tất cả các hãng trong ngành công nghiệp tự động	NAICS
Quy trình	Tìm một giao diện dịch vụ thực thi quy trình	
Sản phẩm/ dịch vụ	Tìm một doanh nghiệp đang bán một sản phẩm hay chào một loại hình dịch vụ nào đó	UNSPSC
Địa danh	Tìm một nhà cung ứng đóng ở Nhật Bản	ISO 3166
DateTime	Tìm một hãng vận chuyển làm việc trong 24h	
Vai trò	Tìm tất cả các nhà cung ứng cùng đóng một vai trò là 'người bán'	

Bảng 1 – Các dạng phân loại mẫu

11 Mô hình thông tin: Quan niệm về bảo mật

Mục này mô tả các khía cạnh của mô hình thông tin gắn liền với các đặc điểm về bảo mật của Registry.

Hình 10 diễn tả quan điểm về các đối tượng của Registry từ khía cạnh bảo mật. Hình 10 cũng chỉ ra mối quan hệ giữa các đối tượng như một sơ đồ Lớp UML. Nó không chỉ ra các thuộc tính Lớp hay các phương thức Lớp. Các vấn đề này sẽ được làm rõ trong mục tiếp theo. Sơ đồ sau chỉ có tính minh họa, không có tính qui tắc.



Hình 10 – Mô hình thông tin: Cơ cấu an ninh

11.1 Lớp AccessControlPolicy (Chính sách kiểm soát truy cập)

Mỗi RegistryObject (đối tượng đăng ký) có thể liên kết với chỉ một AccessControlPolicy. AccessControlPolicy giúp xác định qui tắc kiểm soát truy cập đối với phép toán hay phương thức được

thực thi trên RegistryObject (đối tượng đăng ký) đó. Các qui tắc này được định nghĩa giống như một tập hợp các quyền (Permission)

Phương thức Tổng của lớp <i>AccessControlPolicy</i> (chính sách kiểm soát truy cập)	
collection (tập hợp)	<u>getPermissions()</u> Lấy các quyền (Permission) được định nghĩa cho <i>AccessControlPolicy</i> . ánh xạ tới thuộc tính permissions (cho phép).

11.2 Lớp Permission (Cho phép)

Đối tượng của lớp *Permission* (cho phép) được sử dụng để cấp quyền và điều khiển truy cập tới RegistryObject (đối tượng đăng ký) trong Registry. Quyền của các RegistryObject (đối tượng đăng ký) được định nghĩa trong đối tượng của lớp *AccessControlPolicy* (chính sách kiểm soát truy cập). Đối tượng của lớp *Permission* (cho phép) cấp quyền truy cập cho một phương thức trong RegistryObject (đối tượng đăng ký) nếu đối tượng của lớp *Principal* (đang gọi) có bất kì đặc quyền nào được xác định trong lớp *Permission* (cho phép) .

Xem: **Privilege** , **AccessControlPolicy**

Phương thức Tổng của lớp <i>Permission</i> (cho phép)	
string (chuỗi)	<u>getMethodname()</u> Lấy tên phương thức mà có thể truy cập tới đối tượng của lớp <i>Principal</i> với đặc quyền nhất định. ánh xạ tới thuộc tính methodName
Collection (Tập hợp)	<u>getPrivileges()</u> Lấy các đặc quyền liên quan đến Quyền này. ánh xạ tới thuộc tính privileges.

11.3 Lớp Privilege (Đặc quyền)

Đối tượng của lớp *Privilege* (đặc quyền) không hoặc có chứa *PrivilegeAttributes* (thuộc tính đặc quyền). *PrivilegeAttribute* (thuộc tính đặc quyền) có thể là *Nhóm (Group)*, *Vai trò (Role)* hay *Đặc tính (Identity)*.

Đối tượng của lớp *Principal* phải có tất cả các thuộc tính Đặc quyền (*PrivilegeAttributes*) trong lớp *Privilege* để có thể truy cập tới một phương thức trong RegistryObject (đối tượng đăng ký) đang được bảo mật.

Các quyền được định nghĩa trong lớp *AccessControlPolicy* (chính sách kiểm soát truy cập) của RegistryObject (đối tượng đăng ký) xác định các đặc quyền (*Privileges*) có quyền truy cập tới các phương thức nhất định.

Cơ chế này tạo ra sự linh hoạt trong các qui tắc kiểm soát truy cập. Các qui tắc này dựa vào Association bất kì nào giữa Vai trò (*Role*), Đặc tính (*Identity*) và Nhóm (*Group*).

Xem: PrivilegeAttribute (Thuộc tính đặc quyền), Permission (cho phép)

Phương thức Tổng của lớp Privilege (Đặc quyền)	
Tập hợp	<u>getPrivilegeAttributes()</u> Lấy các thuộc tính Đặc quyền (PrivilegeAttributes) liên quan đến Đặc quyền (Privilege). ánh xạ tới thuộc tính privilegeCác thuộc tính.

11.4 Lớp PrivilegeAttribute (Thuộc tính đặc quyền)

Các lớp con đã biết: Group (Nhóm), Identity (Đặc tính), Role (Vai trò).

PrivilegeAttribute (Thuộc tính đặc quyền) là lớp cơ sở của tất cả các thuộc tính về bảo mật được sử dụng để cấp.

các đặc quyền kiểm soát truy cập cho lớp Principal.

Lớp Principal có thể có một vài loại Thuộc tính Đặc quyền khác nhau. Association riêng của các thuộc tính Đặc quyền được định nghĩa như một Đối tượng của lớp Privilege.

Xem: Principal, Privilege.

11.5 Lớp Role (Vai trò)

Tất cả Các lớp chuẩn: PrivilegeAttribute (thuộc tính đặc quyền).

11.5.1 Vai trò bảo mật của PrivilegeAttribute (Thuộc tính đặc quyền)

Ví dụ như một bệnh viện có thể có các vai trò như: Y tá, Bác sĩ, Quản lí v..v. Vai trò sử dụng để gán các đặc quyền (privilege) cho đối tượng của lớp Principal. Như vai trò *Bác sĩ* có quyền viết đơn thuốc nhưng *Y tá* thì không.

11.6 Lớp Group (Nhóm)

Tất cả Các lớp chuẩn: PrivilegeAttribute (Thuộc tính đặc quyền).

11.6.1 Nhóm bảo mật của PrivilegeAttribute (Thuộc tính đặc quyền)

Một nhóm là một tập hợp người sử dụng có các vai trò khác nhau. Ví dụ như: bệnh viện có các nhóm như Y tá và Bác sĩ tham gia vào các việc khám và điều trị bệnh khác nhau. Nhóm được sử dụng để gán Đặc quyền cho các đối tượng của lớp Principal. Ví dụ các thành viên của nhóm Aspirin được phép viết đơn thuốc Aspirin (mặc dù Vai trò *Y tá* không được viết đơn thuốc).

11.7 Lớp Identity (Đặc tính)

Tất cả Các lớp chuẩn: PrivilegeAttribute (thuộc tính đặc quyền).

11.7.1 Đặc tính bảo mật của PrivilegeAttribute (Thuộc tính đặc quyền)

Thường sử dụng để nhận dạng một người, Organization hay một phần mềm. Thuộc tính nhận dạng (Identity thuộc tính) có thể ở dạng giấy xác nhận dạng số.

11.8 Lớp Principal (Thực thể)

Principal là một thuật ngữ được sử dụng bởi cộng đồng an ninh bao gồm cả người và các hệ thống phần mềm. Đối tượng của *lớp* Principal là các thực thể có một tập hợp các thuộc tính Đặc quyền (PrivilegeAttribute (thuộc tính đặc quyền)). Các PrivilegeAttribute (thuộc tính đặc quyền) này có ít nhất là một đặc tính nhận dạng và có thể có tập hợp các *thành phần* vai trò, *thành phần* nhóm hay quyền về bảo mật.

Lớp Principal được sử dụng để xác nhận yêu cầu và cấp quyền thực thi hành động được yêu cầu dựa trên các PrivilegeAttribute (thuộc tính đặc quyền) liên quan tới *lớp* Principal.

Xem: PrivilegeAttributes, [Privilege](#), [Permission](#)

Phương thức Tổng của <i>lớp</i> Principal (Thực thể)	
Tập hợp	<u>getGroups()</u> Lấy các Nhóm có liên quan đến Thực thể (principal). ánh xạ tới thuộc tính groups.
Tập hợp	<u>getIdentities()</u> Lấy các đặc tính liên quan đến Thực thể (principal). ánh xạ tới Thuộc tính id (định danh)entities.
Tập hợp	<u>getRoles()</u> Lấy các vai trò liên quan đến Thực thể (principal). ánh xạ tới thuộc tính roles.

Tài liệu tham khảo

[ebGLOSS] ebXML Glossary,

http://www.ebxml.org/documents/199909/terms_of_reference.htm

[OAS] OASIS Information Model

<http://xsun.sdct.itl.nist.gov/regrep/OasisRegrepSpec.pdf>

[ISO] ISO 11179 Information Model

<http://208.226.167.205/SC32/jtc1sc32.nsf/576871ad2f11bba785256621005419d7/b83fc7816a6064c68525690e0065f913?OpenDocument>

[BRA97] IETF (Internet Engineering Task Force). RFC 2119: Key words for use in RFCs to Indicate Requirement Levels

<http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2119.html>

[ebRS] ebXML Registry Services Specification

<http://www.oasisopen.org/committees/regrep/documents/2.0/specs/ebRS.pdf>

[ebCPP] ebXML Collaboration-Protocol Profile và Agreement Specification

<http://www.ebxml.org/specrafts/>

[UUID] DCE 128 bit Universal Unique Identifier

http://www.opengroup.org/onlinepubs/009629399/apdxa.htm#tagcjh_20

<http://www.opengroup.org/publications/catalog/c706.htm><http://www.w3.org/TR/REC-xml>

[XPath] XML Path Language (XPath) Version 1.0

<http://www.w3.org/TR/xpath>

[NCName] Namespaces in XML 19990114

<http://www.w3.org/TR/REC-xml-names/#NT-NCName>

