

TCVN

TIÊU CHUẨN QUỐC GIA

**TCVN ISO/TS 15000-4 : 2007
ISO/TS 15000-4 : 2004**

Xuất bản lần 1

**NGÔN NGỮ ĐÁNH DẤU MỞ RỘNG KINH DOANH ĐIỆN TỬ (ebXML)
PHẦN 4: QUY ĐỊNH KỸ THUẬT VỀ DỊCH VỤ ĐĂNG KÝ (ebRS)**

*Electronic business eXtensible Markup Language (ebXML)
Part 4: Registry services specification (ebRS)*

HÀ NỘI – 2007

Mục lục	Trang
Lời nói đầu.....	8
1 Phạm vi áp dụng	9
2 Ban kỹ thuật đăng ký của OASIS/ebXML.....	9
2.1 Cá nhân đóng góp.....	10
3 Giới thiệu	10
3.1 Tóm tắt các nội dung của tiêu chuẩn này.....	10
3.2 Các quy ước chung.....	10
3.3 Người sử dụng tài liệu	10
4 Mục đích thiết kế	11
4.1 Mục đích	11
4.2 Dự báo và giả định	11
5 Tổng quan hệ thống	11
5.1 Sổ đăng ký ebXML	11
5.2 Cách thức làm việc của sổ đăng ký ebXML	11
5.2.1 Các tài liệu giản đồ được đệ trình.....	12
5.2.2 Các tài liệu quy trình kinh doanh được đệ trình	12
5.2.3 Hồ sơ giao thức hợp tác của người bán được đệ trình	12
5.2.4 Người mua tìm người bán	12
5.2.5 CPA được thiết lập.....	12
5.3 Người sử dụng sổ đăng ký	12
5.4 Thực hiện dịch vụ đăng ký.....	14
5.5 Thẻ thức tiến hành	14
5.5.1 Sổ đăng ký (Registry) ebXML	14
5.5.2 Khách hàng đăng ký (Registry Client) ebXML phù hợp.....	15
6 Cấu trúc đăng ký ebXML.....	15
6.1 Mô tả dịch vụ đăng ký	15
6.2 Dịch vụ đăng ký trừu tượng	16
6.3 Các dịch vụ đăng ký thực tế	16
6.3.1 Quy định SOAP	17
6.3.2 Quy định dịch vụ thông điệp ebXML.....	18
6.4 Giao diện LifeCycleManager (Quản lý chu kỳ hoạt động).....	19
6.5 Giao diện LifeCycleManager (Quản lý truy vấn).....	20
6.6 Khách hàng đăng ký (Registry Client)	20
6.6.1 Mô tả khách hàng đăng ký (Registry Client).....	20
6.6.2 Tự khởi động truyền thông đăng ký.....	21
6.6.3 Giao diện khách hàng đăng ký (Registry Client)	22
6.6.4 Registry Response (Phản hồi đăng ký)	23
6.7 Các yêu cầu khả năng hoạt động tương tác	23
6.7.1 Khả năng hoạt động tương tác của khách hàng	23
6.7.2 Hợp tác đăng ký bên trong.....	23
7 Dịch vụ LifeCycleManager (Quản lý chu kỳ hoạt động)	23
7.1 Chu trình của một Hạng mục kho	24
7.2 Thuộc tính registryObject (đối tượng đăng ký).....	24
7.3 Giao thức đệ trình đối tượng.....	25
7.3.1 Tạo ID toàn cầu duy nhất.....	25
7.3.2 Thuộc tính id và các tham chiếu đối tượng	26
7.3.3 Chuỗi đánh giá	26
7.3.4 Tổ chức đệ trình	26
7.3.5 Xử lý lỗi.....	26
7.3.6 Ví dụ về SubmitObjectsRequest (yêu cầu đệ trình đối tượng)	27
7.4 Giao thức cập nhật đối tượng.....	30
7.4.1 Chuỗi đánh giá	30
7.4.2 Tổ chức đệ trình	30
7.4.3 Xử lý lỗi.....	30

7.5	Giao thức thêm các khe dữ liệu.....	31
7.6	Giao thức gỡ bỏ các khe dữ liệu.....	32
7.7	Giao thức phê chuẩn các đối tượng.....	32
7.7.1	Chuỗi đánh giá.....	33
7.7.2	Tổ chức đệ trình.....	33
7.7.3	Xử lý lỗi.....	33
7.8	Giao thức không tán thành đối tượng.....	33
7.8.1	Chuỗi đánh giá.....	34
7.8.2	Tổ chức đệ trình.....	34
7.8.3	Xử lý lỗi.....	34
7.9	Giao thức gỡ bỏ các đối tượng.....	35
7.9.1	Phạm vi xoá bỏ DeleteRepositoryItemOnly (chỉ xoá hạng mục kho).....	35
7.9.2	Phạm vi xoá bỏ DeleteAll (xoá tất cả).....	35
7.9.3	Xử lý lỗi.....	36
8	Dịch vụ quản lý truy vấn.....	36
8.1	Yêu cầu/Phản hồi đối với các truy vấn đặc biệt.....	37
8.1.1	Các tùy chọn phản hồi truy vấn.....	38
8.2	Hỗ trợ truy vấn theo bộ lọc (Filter Query Support).....	39
8.2.1	Bộ lọc truy vấn.....	42
8.2.2	RegistryObjectQuery (đối tượng truy vấn số đăng ký).....	43
8.2.3	RegistryEntryQuery (truy vấn mục nhập đăng ký).....	55
8.2.4	Truy vấn Association (liên kết).....	58
8.2.5	AuditableEventQuery (truy vấn sự kiện có thể kiểm tra).....	60
8.2.6	ClassificationQuery (truy vấn phân loại).....	63
8.2.7	ClassificationNodeQuery (truy vấn nút phân loại).....	65
8.2.8	ClassificationSchemeQuery (truy vấn giản đồ phân loại).....	70
8.2.9	RegistryPackageQuery (truy vấn gói đăng ký).....	71
8.2.10	ExtrinsicObjectQuery (truy vấn đối tượng ngoại lai).....	74
8.2.11	OrganizationQuery (truy vấn tổ chức).....	75
8.2.12	OrganizationQuery (truy vấn dịch vụ).....	79
8.2.13	Các bộ lọc của số đăng ký.....	81
8.2.14	Cách biểu diễn bắt buộc trong mệnh đề XML.....	85
8.3	Hỗ trợ truy vấn SQL.....	88
8.3.1	Quy định cú pháp truy vấn SQL đến [eRIM].....	89
8.3.2	Quy định ngữ nghĩa dựa trên cú pháp truy vấn.....	91
8.3.3	Kết quả truy vấn SQL.....	91
8.3.4	Truy vấn được dựa trên siêu dữ liệu riêng lẻ.....	91
8.3.5	Truy vấn RegistryObject (đối tượng đăng ký).....	92
8.3.6	Truy vấn RegistryEntry (mục nhập đăng ký).....	92
8.3.7	Truy vấn Classification (phân loại).....	92
8.3.8	Truy vấn Association (liên kết).....	93
8.3.9	Truy vấn gói.....	94
8.3.10	Truy vấn ExternalLink (Liên kết ngoài).....	94
8.3.11	Truy vấn vết kiểm tra.....	95
8.4	Truy cập nội dung.....	95
8.4.1	Nhận dạng dung lượng nội dung.....	95
8.4.2	Cấu trúc của thông báo "trả lời tiếp nhận nội dung".....	95
9	An ninh đăng ký.....	96
9.1	Các vấn đề liên quan an ninh.....	96
9.2	Tính toàn vẹn của nội dung đăng ký.....	96
9.2.1	Chữ ký vùng mang thông tin của thông điệp.....	97
9.2.2	Yêu cầu chữ ký cho vùng mang thông tin.....	98
9.3	Xác thực.....	99
9.3.1	Chữ ký tiêu đề của thông điệp.....	99
9.4	Phần tử KeyInfo (thông tin chính) và Distribution (phân phối) chính.....	101
9.5	Tính bảo mật.....	101
9.5.1	Tính bảo mật thông điệp điện tín.....	101

TCVN ISO/TS 15000-4 : 2007

9.5.2	Tính bảo mật của nội dung đăng ký	101
9.6	Sự cấp phép	102
9.6.1	Hoạt động	102
9.7	Kiểm soát truy cập	102
Phụ lục A	104
A.1	Quy định lý thuyết dịch vụ đăng ký	104
A.2	Liên kết dịch vụ đăng ký SOAP	104
Phụ lục B	105
Định nghĩa giản đồ đăng ký ebXML	105
B.1	Giản đồ RIM	105
B.2	Giản đồ truy vấn	105
B.3	Giản đồ giao diện dịch vụ đăng ký	105
B.4	Những ví dụ của tài liệu mẫu	105
Phụ lục C	106
C.1	Biểu đồ phân loại UML	106
C.2	Biểu đồ chuỗi UML	106
Phụ lục D	107
D.1	Đặc điểm kỹ thuật của cú pháp truy vấn SQL	107
D.2	BNF không có tính quy phạm cho ngữ pháp cú pháp truy vấn	107
D.3	Giản đồ quan hệ truy vấn SQL	108
Phụ lục E	109
E.1	Phân loại tự động nội dung XML	109
E.2	Định nghĩa bảng chỉ dẫn	109
E.3	Ví dụ định nghĩa bảng chỉ dẫn	110
E.4	Định nghĩa mục đích XML	110
E.5	Ví dụ của sự phân loại tự động	110
Phụ lục F	111
Hướng dẫn thực thi an ninh	111
F.1	Các vấn đề an ninh	111
F.2	Sự xác nhận	112
F.3	Sự cấp phép	112
F.4	Tự đăng ký	113
F.5	Đệ trình nội dung – Trách nhiệm của khách hàng	113
F.6	Đệ trình nội dung – Trách nhiệm của sổ đăng ký	113
F.7	Xoá/ Không tán thành nội dung – Trách nhiệm của khách hàng	113
F.8	Xoá/ Không tán thành nội dung – Trách nhiệm của sổ đăng ký	113
F.9	Sử dụng ds:KeyInfo (Thông tin chính)	114
Phụ lục G	115
G.1	Định nghĩa	115
G.1.1	Bộ ký tự mã hoá (CCS):	115
G.1.2	Hệ thống ký tự mã hoá (CES)	115
G.1.3	Bộ ký tự (charset)	115
G.2	NLS và thông điệp yêu cầu/ phản hồi	115
G.3	NLS và lưu trữ RegistryObject (đối tượng đăng ký)	115
G.3.1	Bộ ký tự của LocalizedString (Chuỗi ký tự được địa phương hóa)	116
G.3.2	Thông tin ngôn ngữ của LocalizedString (Chuỗi ký tự được địa phương hóa)	116
G.4	NLS và lưu trữ các hạng mục kho	116
G.4.1	Bộ ký tự của các hạng mục kho	116
G.4.2	Thông tin ngôn ngữ của hạng mục kho	116
Phụ lục H	117
Tài liệu tham khảo	118

	Trang
Hình vẽ	
Hình 1 - Mối quan hệ các chủ thể	14
Hình 2 - Cấu trúc dịch vụ đăng ký ebXML	15
Hình 3 - Dịch vụ đăng ký ebXML trừu tượng	16

Hình 4 - Dịch vụ đăng ký ebXML thực tế.....	17
Hình 5 - Cấu trúc đăng ký hỗ trợ các mạng tô-pô linh hoạt	21
Hình 6 - Chu trình của một hạng mục kho.....	24
Hình 7 - Biểu đồ thứ tự các đối tượng đệ trình	25
Hình 8 - Sơ đồ thứ tự các đối tượng cập nhật	30
Hình 9 - Sơ đồ thứ tự các khe bổ sung	31
Hình 10 - Sơ đồ thứ tự các thông tin gỡ bỏ.....	32
Hình 11 - Sơ đồ thứ tự các đối tượng phê chuẩn	32
Hình 12 - Sơ đồ thứ tự các đối tượng không tán thành	34
Hình 13 - Sơ đồ thứ tự các đối tượng gỡ bỏ.....	36
Hình 14 - Sơ đồ thứ tự truy vấn đặc biệt đệ trình.....	37
Hình 15 - Ví dụ quy định ebRIM.....	40
Hình 16 - Quy định ebRIM đối với RegistryObjectQuery	43
Hình 17 - Quy định ebRIM đối với RegistryEntryQuery	56
Hình 18 - Quy định ebRIM đối với AssociationQuery.....	59
Hình 19 - Quy định ebRIM đối với AuditableEventQuery.....	60
Hình 20 - Quy định ebRIM đối với ClassificationQuery.....	64
Hình 21 - Quy định ebRIM cho ClassificationNodeQuery (truy vấn nút phân loại).....	66
Hình 22 - Quy định ebRIM cho ClassificationSchemeQuery (truy vấn giản đồ phân loại).....	71
Hình 23 - Quy định ebRIM cho RegistryPackageQuery (truy vấn gói đăng ký)	72
Hình 24 - Quy định ebRIM cho ExtrinsicObjectQuery (truy vấn đối tượng ngoại lai).....	74
Hình 25 - Quy định ebRIM cho OrganizationQuery (truy vấn tổ chức).....	75
Hình 26 - Quy định ebRIM cho OrganizationQuery (truy vấn dịch vụ)	79
Hình 27 - Cấu trúc mệnh đề.....	85

Bảng	Trang
Bảng 1 - Người sử dụng đăng ký	13
Bảng 2 - Tóm tắt LifeCycleManager (quản lý chu kỳ hoạt động).....	20
Bảng 3 - QueryManager (quản lý truy vấn)	20
Bảng 4 - Tóm tắt khách hàng đăng ký (Registry Client).....	23
Bảng 5 - Xử lý lỗi các đối tượng đệ trình.....	27
Bảng 6 - Xử lý lỗi các đối tượng cập nhật	31
Bảng 7 - Xử lý lỗi các đối tượng phê chuẩn	33
Bảng 8 - Xử lý lỗi các đối tượng không tán thành.....	35
Bảng 9 - Xử lý lỗi các đối tượng gỡ bỏ.....	36
Bảng 10 - Các biểu thức bộ lọc hướng cho các trường hợp được sử dụng.....	69
Bảng 11 - Chính sách kiểm soát truy cập mặc định.....	102

TCVN ISO/TS 15000-4 : 2007

Lời nói đầu

TCVN ISO/TS 15000-4 : 2007 hoàn toàn tương đương với tiêu chuẩn **ISO/TS 15000-4 : 2004**.

TCVN ISO/TS 15000-4 : 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 4: Quy định kỹ thuật về dịch vụ đăng ký (ebRS)

*Electronic business eXtensible Markup Language (ebXML) —
Part 4: Registry services specification (ebRS)*

1 Phạm vi áp dụng

Tiêu chuẩn này dựa trên cơ sở dạng thức RFC của Internet Society (Cộng đồng người sử dụng Internet).

Phiên bản hiện tại của quy định kỹ thuật về dịch vụ đăng ký của OASIS/ebXML:

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

Phiên bản mới nhất của quy định kỹ thuật về dịch vụ đăng ký của OASIS/ebXML:

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

2 Ban kỹ thuật đăng ký của OASIS/ebXML

Kathryn Breininger, Boeing

Lisa Carnahan, NIST

Joseph M. Chiusano, LMI

Suresh Damodaran, Sterling Commerce

Mike DeNicola, Fujitsu

Anne Fischer, Drummond Group, Inc.

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, Chevron

Nikola Stojanovic, Encoda Systems, Inc.

Prasad Yendluri, webmethods

Yutaka Yoshida, Sun Microsystems.

2.1 Cá nhân đóng góp

Sau đây là các cá nhân đã đóng góp vào nội dung tiêu chuẩn nhưng không phải là thành viên được bỏ phiếu của ban kỹ thuật về đăng ký OASIS/ebXML.

Len Gallagher, NIST

Sekhar Vajjhala, Sun Microsystems.

3 Giới thiệu

3.1 Tóm tắt các nội dung của tiêu chuẩn này

Tài liệu này định nghĩa giao diện cho các dịch vụ đăng ký ebXML cũng như những giao thức liên quan, các định nghĩa thông điệp và giản đồ XML.

Một tài liệu khác; mô hình thông tin đăng ký ebXML (ebRIM), cung cấp thông tin về các kiểu siêu dữ liệu (metadata) được lưu giữ tại Sổ đăng ký (Registry) cũng như các quan hệ giữa các lớp siêu dữ liệu (metadata) khác nhau.

3.2 Các quy ước chung

Các quy ước sau đây được áp dụng trong toàn bộ tiêu chuẩn này:

Các biểu đồ UML được sử dụng như là một phương pháp mô tả chính xác các khái niệm. Chúng không nhằm để truyền đạt bất kỳ một yêu cầu thực hiện hay phương pháp luận nào.

Thuật ngữ "hạng mục kho" (repository item) được sử dụng để nói đến một đối tượng dạng kho được lưu giữ và bảo vệ (ví dụ tài liệu XML hoặc DTD). Mỗi hạng mục kho được mô tả trong sổ đăng ký (Registry) bởi một trường hợp RegistryObject (đối tượng đăng ký).

Thuật ngữ "mục nhập đăng ký" (RegistryEntry) được dùng để nói đến một đối tượng cung cấp siêu dữ liệu (metadata) về một hạng mục kho.

Những từ viết hoa in nghiêng được định nghĩa trong bảng chú giải thuật ngữ ebXML.

Các từ **PHẢI**, **KHÔNG PHẢI**, **ĐƯỢC YÊU CẦU**, **SẼ**, **SẼ KHÔNG**, **NÊN**, **KHÔNG NÊN**, **ĐƯỢC KHUYẾN NGHỊ**, **CÓ THỂ** và **LỰA CHỌN** khi xuất hiện trong tiêu chuẩn này sẽ phải giải thích như đã nêu trong Phần tham khảo 2119 [Bra97].

3.3 Người sử dụng tài liệu

Người sử dụng mục tiêu của tiêu chuẩn này là cộng đồng những người phát triển phần mềm, bao gồm:

- những người đảm nhận các dịch vụ đăng ký ebXML;
- các khách hàng đăng ký (Registry Client) ebXML.

Các tài liệu liên quan:

Những tài liệu dưới đây cung cấp cơ sở và thông tin liên quan cho người đọc:

- a) mô hình thông tin đăng ký ebXML [ebRIM];
- b) quy định dịch vụ thông điệp ebXML [ebMS];
- c) giản đồ quy định quá trình kinh doanh ebXML [ebBPSS];
- d) quy định thỏa thuận và hồ sơ thủ tục hợp tác ebXML [ebCPP].

4 Mục đích thiết kế

4.1 Mục đích

Mục đích của tiêu chuẩn này là:

- thông tin định kỳ các dịch vụ đăng ký cho người phát triển phần mềm;
- quy định giao diện cho khách hàng đăng ký (Registry Client) và sổ đăng ký (Registry);
- cung cấp cơ sở cho việc hỗ trợ các yêu cầu đăng ký ebXML hoàn thiện hơn trong tương lai;
- tương thích với các tiêu chuẩn ebXML khác.

4.2 Dự báo và giả định

Các yêu cầu khả năng vận hành tương tác chỉ ra rằng ít nhất một trong những giao diện có tính quy chuẩn như đã đề cập trong định mức này phải được hỗ trợ.

1. Toàn bộ truy cập vào nội dung sổ đăng ký phải được thể hiện qua những giao diện được xác định đối với các dịch vụ đăng ký;
2. Sổ đăng ký (Registry) sử dụng một kho để lưu trữ và truy cập thông tin liên tục theo yêu cầu của dịch vụ đăng ký. Đây là một chi tiết thực hiện và không được đưa ra trong tiêu chuẩn này.

5 Tổng quan hệ thống

5.1 Sổ đăng ký ebXML

Sổ đăng ký (Registry) ebXML cung cấp một bộ dịch vụ để có thể chia sẻ thông tin giữa các bên quan tâm với mục đích tạo nên sự hợp nhất quá trình kinh doanh giữa các bên tham gia đó dựa trên các tiêu chuẩn ebXML. Các thông tin được chia sẻ được duy trì như các đối tượng trong kho dữ liệu và được điều hành bởi dịch vụ đăng ký ebXML được định nghĩa trong tiêu chuẩn này.

5.2 Cách thức làm việc của sổ đăng ký ebXML

Phần này mô tả một vài trường hợp mức cao để minh họa cách thức khách hàng đăng ký (Registry Client) có thể sử dụng các dịch vụ đăng ký để thực hiện các trao đổi B2B. Đây chỉ là minh họa, không quy định.

Kịch bản kinh doanh dưới đây cung cấp một ví dụ nguyên bản ở mức cao mà các trường hợp sử dụng dưới dạng tương tác giữa các khách hàng đăng ký (Registry Client) và sổ đăng ký (Registry). Nó không phải là một danh sách đầy đủ các trường hợp sử dụng có thể mượn tương được. Ví dụ, Một người mua và một người bán muốn thực hiện các trao đổi B2B sử dụng tập quán kinh doanh Purchase Order (đơn đặt hàng mua bán) RosettaNet PIP3A4. Giả thuyết rằng cả người mua và người bán sử dụng cùng một dịch vụ đăng ký như nhau được cung cấp bởi một bên thứ ba. Chú ý rằng cấu trúc này hỗ trợ các khả năng khác (ví dụ mỗi bên sử dụng sổ đăng ký (Registry) riêng của mình).

5.2.1 Các tài liệu gián đồ được đệ trình

Một bên thứ ba ví dụ một tập đoàn công nghiệp hay một nhóm tiêu chuẩn đệ trình các tài liệu gián đồ cần thiết theo yêu cầu của tập quán kinh doanh Purchase Order (đơn đặt hàng mua bán) RosettaNet PIP3A4 với sổ đăng ký (Registry) sử dụng dịch vụ LifeCycleManager (quản lý chu kỳ hoạt động) của sổ đăng ký (Registry) được mô tả trong Mục 7.3.

5.2.2 Các tài liệu quy trình kinh doanh được đệ trình

Một bên thứ ba ví dụ một tập đoàn công nghiệp hay một nhóm xây dựng tiêu chuẩn đệ trình các tài liệu quy trình kinh doanh cần thiết theo yêu cầu của tập quán kinh doanh Purchase Order (đơn đặt hàng mua bán) RosettaNet PIP3A4 với sổ đăng ký (Registry) sử dụng dịch vụ LifeCycleManager (quản lý chu kỳ hoạt động) của sổ đăng ký (Registry) được mô tả trong Mục 7.3.

5.2.3 Hồ sơ giao thức hợp tác của người bán được đệ trình

Người bán phát hành hồ sơ giao thức hợp tác (Collaboration Protocol Profile) hay CPP của mình được ebCPP xác định trong sổ đăng ký (Registry). CPP mô tả về người bán, vai trò của người bán, các dịch vụ mà người bán chào bán và các chi tiết kỹ thuật về cách mà các dịch vụ đó có thể được truy cập. Người bán phân loại hồ sơ giao thức hợp tác (Collaboration Protocol Profile) của mình sử dụng các khả năng phân loại linh hoạt của sổ đăng ký (Registry).

5.2.4 Người mua tìm người bán

Người mua duyệt sổ đăng ký (Registry) sử dụng các gián đồ phân loại (Classification scheme) được xác định trong sổ đăng ký (Registry) sử dụng một công cụ Registry Browser GUI (Trình duyệt sổ đăng ký) để tìm người bán phù hợp. Ví dụ người mua có thể tìm tất cả các đối tác trong ngành công nghiệp ô tô, là người bán, hỗ trợ quy trình RosettaNet PIP3A4 và bán máy radio của ô tô.

Người mua tìm CPP của người bán và quyết định trở thành đối tác với người bán.

5.2.5 CPA được thiết lập

Người mua đơn phương tạo một thỏa thuận giao thức hợp tác (Collaboration Protocol Agreement) hay CPA như được ebCPP xác định cho người bán sử dụng CPP của người bán và CPP của chính họ như là đầu vào. Người mua đề nghị một quan hệ trao đổi mua bán với người bán sử dụng CPA đơn phương này. Người bán chấp nhận CPA được yêu cầu này và quan hệ trao đổi mua bán được thiết lập.

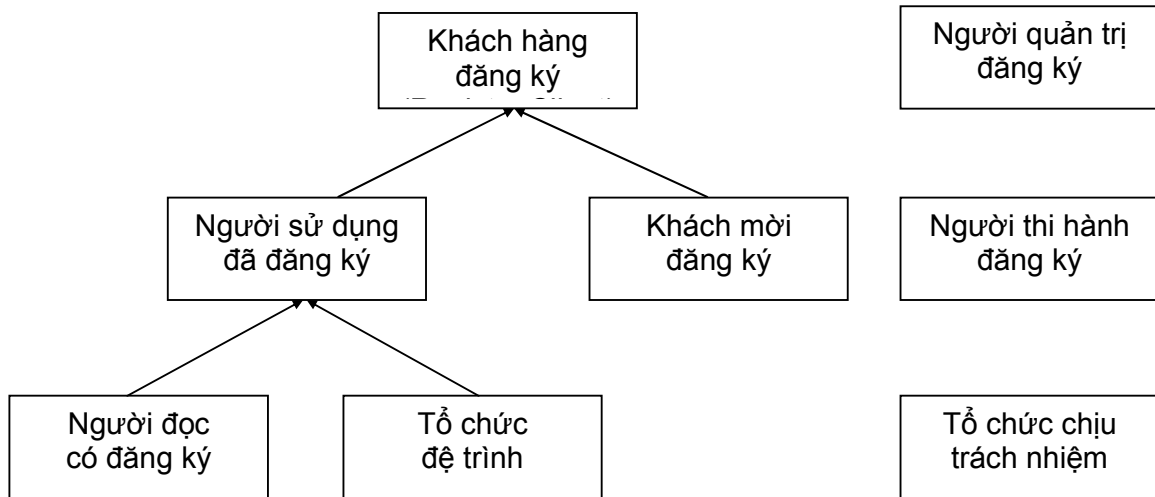
Khi người bán chấp nhận CPA, các bên có thể bắt đầu tiến hành các giao dịch B2B như đã được ebMS xác định.

5.3 Người sử dụng sổ đăng ký

Chủ thể sử dụng đăng ký từ quan điểm an ninh và phân tích các mối lo ngại an ninh của việc đăng ký ở dưới đây. Bản phân tích này hướng tới những yêu cầu an ninh cho phiên bản 2.0. Một vài chủ thể được định nghĩa trong Mục 9.7. Chú ý rằng cũng một chủ thể có thể thực hiện các vai trò khác nhau. Ví dụ, tổ chức có thẩm quyền đăng ký và người quản trị đăng ký có thể là cùng một chủ thể.

Bảng 1 - Những người sử dụng đăng ký

Chủ thể	Chức năng	ISO/IEC 11179	Ghi chú
Tổ chức có thẩm quyền đăng ký	Tổ chức các đối tượng đăng ký	Tổ chức có thẩm quyền đăng ký (RA)	
Người quản trị đăng ký	Đánh giá và làm cho các chính sách an ninh đăng ký có hiệu lực. Thúc đẩy sự xác minh chính sách an ninh đăng ký.		Có thể cũng là tổ chức có thẩm quyền đăng ký
Người sử dụng đã đăng ký	Có thỏa thuận với tổ chức có thẩm quyền đăng ký và phải được Tổ chức có thẩm quyền đăng ký xác nhận		Thỏa thuận có thể là một CPA ebXML hoặc một dạng thỏa thuận khác
Khách mời đăng ký	Không có thỏa thuận với tổ chức có thẩm quyền đăng ký. Không được xác nhận để truy cập đăng ký. Không thể thay đổi nội dung đăng ký (có thể được phép đọc một vài đối tượng đăng ký)		Lưu ý rằng một khách mời đăng ký không phải là một người đọc sổ đăng ký
Tổ chức đệ trình	Một người sử dụng đã đăng ký thực hiện các hoạt động vòng đời của các đối tượng đăng ký được phép	Tổ chức đệ trình (SO)	
Người đọc có đăng ký	Người sử dụng đã đăng ký chỉ được phép đọc		
Tổ chức chịu trách nhiệm	Tạo lập các đối tượng đăng ký	Tổ chức chịu trách nhiệm (RO)	RO có thể cũng là SO
Khách hàng đăng ký (Registry Client)	Người sử dụng hoặc khách mời đã đăng ký		



Hình 1 - Mối quan hệ các chủ thể

CHÚ THÍCH:

Trong phiên bản hiện tại của tiêu chuẩn này, những điều sau là đúng.

Tổ chức đệ trình và tổ chức chịu trách nhiệm là một.

Việc đăng ký của người sử dụng diễn ra ngoài bảng trên, nghĩa là không được phân loại trong bảng phân loại này.

Người quản trị đăng ký và tổ chức có thẩm quyền đăng ký là một.

5.4 Thực hiện dịch vụ đăng ký

Dịch vụ đăng ký có thể được thực hiện bằng một số cách, bao gồm địa chỉ web công cộng, địa chỉ web cá nhân, được thực hiện bởi một ASP hoặc bởi một nhà cung cấp VPN.

5.5 Thẻ thức tiến hành

Sự thực thi là làm theo sổ đăng ký ebXML nếu việc tiến hành thoả mãn các điều kiện trong Mục 5.5.1. Sự thực thi là làm theo khách hàng đăng ký (Registry Client) ebXML nếu việc tiến hành thoả mãn các điều kiện trong Mục 5.5.2. Sự thực thi là làm theo sổ đăng ký (Registry) ebXML và theo khách hàng đăng ký (Registry Client) ebXML nếu việc tiến hành tuân theo các điều kiện của Mục 5.5.1 và 5.5.2. Việc thi hành có thể là làm theo sổ đăng ký (Registry) ebXML, hoặc làm theo khách hàng đăng ký (Registry Client) ebXML, hoặc làm theo sổ đăng ký (Registry) và khách hàng đăng ký (Registry Client) ebXML.

5.5.1 Sổ đăng ký (Registry) ebXML

Sự thực thi tuân theo tiêu chuẩn này là một sổ đăng ký (Registry) ebXML nếu thoả mãn các điều kiện sau:

1. tuân theo mô hình thông tin đăng ký ebXML [ebRIM];
2. hỗ trợ cú pháp và ngữ nghĩa của mô hình an ninh và các giao diện đăng ký;
3. hỗ trợ giản đồ đăng ký ebXML đã được xác định (phụ lục B);

4. hỗ trợ có tính lựa chọn cú pháp và ngữ nghĩa của Mục 8.3, hỗ trợ truy vấn SQL.

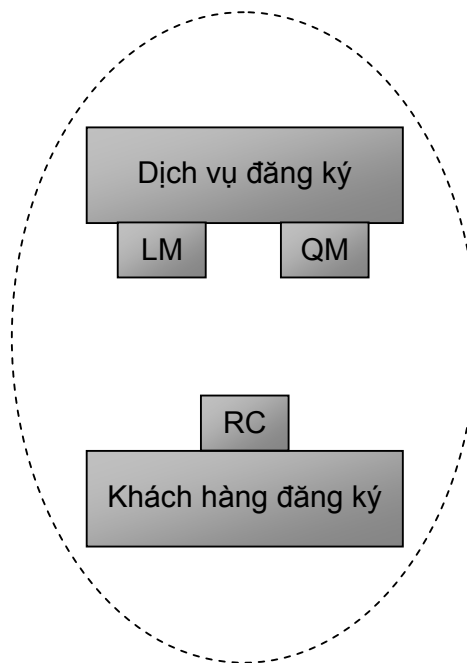
5.5.2 Khách hàng đăng ký (Registry Client) ebXML phù hợp

Việc thực thi tuân theo tiêu chuẩn này là một khách hàng đăng ký (Registry Client) ebXML nếu thoả mãn các điều kiện sau:

1. hỗ trợ CPA ebXML và quy trình tự khởi động;
2. hỗ trợ cú pháp và ngữ nghĩa của giao diện khách hàng đăng ký (Registry Client Interfaces);
3. hỗ trợ thông điệp lỗi ebXML DTD đã được xác định;
4. hỗ trợ giản đồ đăng ký ebXML đã được xác định (phụ lục B).

6 Cấu trúc đăng ký ebXML

Cấu trúc đăng ký ebXML bao gồm một dịch vụ đăng ký ebXML và các khách hàng đăng ký (Registry Client) ebXML. dịch vụ đăng ký ebXML cung cấp các biện pháp để quản lý kho dữ liệu. Khách hàng đăng ký (Registry Client) ebXML là một trình ứng dụng được dùng để truy cập sổ đăng ký (Registry).



Hình 2 - Cấu trúc dịch vụ đăng ký ebXML

6.1 Mô tả dịch vụ đăng ký

Dịch vụ đăng ký ebXML gồm một bộ các giao diện mạnh được thiết kế để quản lý về cơ bản các đối tượng và các yêu cầu có liên quan đến đăng ký ebXML. Hai giao diện chính của dịch vụ đăng ký bao gồm:

- giao diện LifeCycleManager (quản lý chu kỳ hoạt động) cung cấp một bộ các biện pháp quản lý các đối tượng trong sổ đăng ký (Registry);
- giao diện LifeCycleManager (quản lý truy vấn) kiểm soát việc phát hiện và thu hồi thông tin từ sổ đăng ký (Registry).

TCVN ISO/TS 15000-4 : 2007

Một chương trình khách đăng ký tận dụng các dịch vụ đăng ký bằng cách viện dẫn các cách thức trên một trong các giao diện nói trên đã được dịch vụ đăng ký xác định. Tiêu chuẩn này xác định các giao diện được dịch vụ đăng ký đưa ra (Mục 6.4 và 6.5) cũng như giao diện cho khách hàng đăng ký (Registry Client) (Mục 6.6).

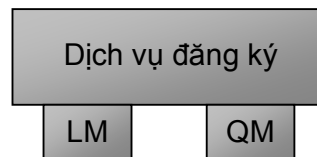
6.2 Dịch vụ đăng ký trừu tượng

Cấu trúc này quy định sổ đăng ký (Registry) ebXML là dịch vụ đăng ký trừu tượng được xác định như sau:

1. một bộ các giao diện được hỗ trợ bởi sổ đăng ký;
2. một bộ cách thức được hỗ trợ bởi mỗi giao diện;
3. các thông số và phản hồi được hỗ trợ bởi mỗi cách thức.

Dịch vụ đăng ký trừu tượng không xác định bất kỳ một tiến hành cụ thể nào đối với sổ đăng ký (Registry) ebXML cũng như không định rõ các giao thức cụ thể mà tổ chức đăng ký sử dụng. Các chi tiết tiến hành này được mô tả bởi các dịch vụ đăng ký thực tế để thực hiện dịch vụ đăng ký ảo.

Dịch vụ đăng ký trừu tượng (Mô hình 3) chỉ ra cách thức một sổ đăng ký (Registry) ebXML trừu tượng cung cấp hai giao diện chức năng chính gọi là QueryManager (quản lý truy vấn) (QM) và LifeCycleManager (quản lý chu kỳ hoạt động) (LM)



Hình 3 - Dịch vụ đăng ký ebXML trừu tượng

Phụ lục A cung cấp các siêu liên kết cho việc xác định dịch vụ trừu tượng trong cú pháp WSDL (Ngôn ngữ mô tả dịch vụ web).

6.3 Các dịch vụ đăng ký thực tế

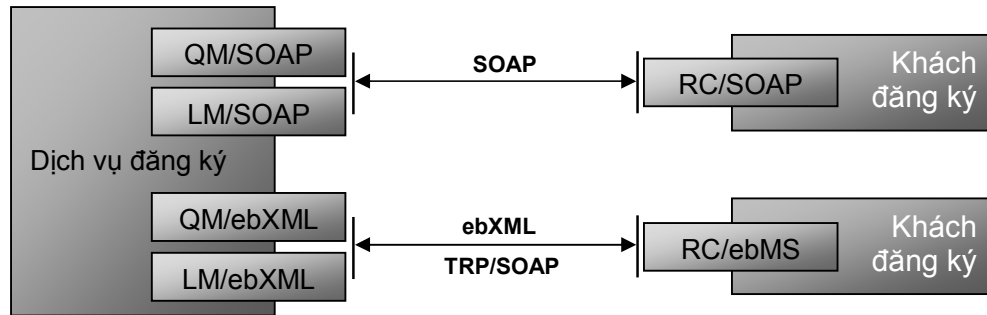
Cơ cấu trên cho phép dịch vụ đăng ký trừu tượng được ghép với một hay nhiều dịch vụ đăng ký thực tế được xác định như:

- các tiến hành của các giao diện được xác định bởi dịch vụ đăng ký ảo;
- các quy định về những giao diện thực tế này với các giao thức truyền thông cụ thể.

Tiêu chuẩn này mô tả hai quy định thực tế cho dịch vụ đăng ký ảo:

- quy định SOAP sử dụng giao thức HTTP;
- quy định dịch vụ thông điệp (ebMS) ebXML.

Một đăng ký có thể tiến hành một hoặc cả hai quy định thực tế cho dịch vụ đăng ký trừu tượng như được thể hiện trong Mô hình 4.



Hình 4 - Dịch vụ đăng ký ebXML thực tế

Mô hình 4 thể hiện một tiến hành thực tế của Registry ebXML (Sổ đăng ký ebXML) trừu tượng (dịch vụ đăng ký) ở bên trái. Dịch vụ đăng ký cung cấp các giao diện LifeCycleManager (quản lý truy vấn) (quản lý truy vấn) và LifeCycleManager (quản lý chu kỳ hoạt động) luôn có với các quy định đa giao thức (SOAP và ebMS).

Mô hình 4 cũng thể hiện hai khách hàng khác nhau của sổ đăng ký ebXML ở bên phải. Khách hàng phía trên dùng giao diện SOAP để truy cập đăng ký trong khi khách hàng phía dưới dùng giao diện ebMS. Các khách hàng dùng giao diện thực tế thích hợp trong dịch vụ đăng ký trên cơ sở tham chiếu giao thức của họ.

6.3.1 Quy định SOAP

6.3.1.1 Thuật ngữ WSDL

Phần này cung cấp một giới thiệu vắn tắt về ngôn ngữ mô tả dịch vụ web (Web Service Description Language) (WSDL) khi quy định SOAP được định rõ là sử dụng cú pháp WSDL. WSDL cung cấp khả năng mô tả một dịch vụ web một cách trừu tượng cũng như là với các quy định thực tế đối với các giao thức cụ thể. Trong WSDL, một dịch vụ trừu tượng gồm một hay nhiều kiểu port (cổng) hay end-points (điểm cuối). Mỗi cổng gồm một bộ các thao tác (operations). Mỗi thao tác được xác định ở dạng các thông điệp (message) trong đó định ra dữ liệu nào được trao đổi như một phần của hoạt động đó. Mỗi thông điệp được xác định một cách đặc trưng ở dạng các yếu tố trong một định nghĩa Giản đồ XML.

Một dịch vụ trừu tượng không bị quy định với bất kỳ giao thức cụ thể nào (ví dụ SOAP). Trong WSDL, một dịch vụ trừu tượng có thể được dùng để xác định một dịch vụ thực tế bằng cách quy định nó với một giao thức cụ thể. Quy định này được thực hiện bằng cách cung cấp một định nghĩa quy định cho một kiểu cổng ảo để xác định các chi tiết cụ thể của các giao thức thêm. Cuối cùng, một định nghĩa dịch vụ thực tế được xác định như là một tập hợp các cổng, ở mỗi cổng chỉ đơn giản thêm thông tin địa chỉ là một URL cho mỗi cổng thực tế.

6.3.1.2 Quy định cụ thể đối với SOAP

Phần này giả thiết rằng người đọc có quen thuộc phần nào với SOAP và WSDL. Quy định SOAP với sổ đăng ký (Registry) ebXML được xác định như một mô tả dịch vụ web trong WSDL như sau:

- một phần tử Service (dịch vụ) độc lập với tên gọi là "RegistryService" (dịch vụ đăng ký) xác định quy định SOAP thực tế cho dịch vụ đăng ký;
- phần tử Service (dịch vụ) gồm 2 định nghĩa cổng, ở mỗi cổng tương ứng với một trong các giao diện được xác định cho dịch vụ đăng ký ảo. Mỗi cổng bao gồm một HTTP URL để truy nhập cổng đó;
- mỗi định nghĩa cổng cũng tham chiếu một yếu tố quy định, một yếu tố cho mỗi giao diện được xác định trong WSDL đối với dịch vụ đăng ký ảo.

```
<service name = "RegistryService">
  <port name = "QueryManagerSOAPBinding" binding =
"tns:QueryManagerSOAPBinding">
    <soap:address location = "http://your_URL_to_your_QueryManager"/>
  </port>
  <port name = "LifecycleManagerSOAPBinding" binding =
"tns:LifecycleManagerSOAPBinding">
    <soap:address location = "http://your_URL_to_your_QueryManager"/>
  </port>
</service>
```

Mô tả WSDL đầy đủ cho quy định SOAP có thể thu được qua một siêu liên kết trong Phụ lục A.

6.3.2 Quy định dịch vụ thông điệp ebXML

6.3.2.1 Các phần tử Action (hành động) và Service (dịch vụ)

Khi sử dụng Tiêu chuẩn Dịch vụ thông điệp ebXML, các phần tử Service (dịch vụ) đăng ký ebXML tương ứng với các phần tử Service (dịch vụ) thông điệp như sau:

- giá trị của phần tử Service (dịch vụ) trong MessageHeader (tiêu đề thông điệp) là một tên giao diện của dịch vụ đăng ký ebXML (ví dụ "LifecycleManager" ("quản lý chu kỳ hoạt động")). Thuộc tính type (kiểu) của phần tử Service (dịch vụ) nên có giá trị là "ebXMLRegistry" ("Đăng ký ebXML");
- giá trị của phần tử Action (hành động) trong MessageHeader (tiêu đề thông điệp) là tên phương pháp của dịch vụ đăng ký ebXML (ví dụ "submitObjects" ("Đối tượng đệ trình")).

```
<eb:Service    eb:type="ebXMLRegistry">LifecycleManager</eb:Service >
<eb:Action> submitObjects </eb:Action>
```

Chú ý rằng những điều trên cho phép Khách hàng đăng ký (Registry Client) chỉ một cặp giao diện/ cách thức cho một thông điệp. Điều này có ngụ ý rằng một khách hàng đăng ký (Registry Client) chỉ có thể viện dẫn một phương pháp trong một giao diện cụ thể cho một yêu cầu đăng ký được đưa ra.

6.3.2.1 Các phản hồi đồng bộ và không đồng bộ

Tất cả các cách thức trên các giao diện được đưa ra để phản hồi một thông điệp bởi tổ chức đăng ký.

Phản hồi không đồng bộ

Khi một thông điệp được gửi đi không đồng bộ, sổ đăng ký (Registry) sẽ gửi 2 thông điệp phản hồi. Thông điệp đầu sẽ là một phản hồi ngay lập tức tới lời yêu cầu và không thể hiện phản hồi thực sự cho yêu cầu. Thông điệp này sẽ bao gồm:

- MessageHeader (tiêu đề thông điệp);
- RegistryResponse (phản hồi đăng ký) không có nội dung (ví dụ: KHÔNG AdHocQueryResponse).

- thuộc tính status ứng với giá trị Unavailable (không có sẵn).

Sổ đăng ký (Registry) gửi RegistryResponse (phản hồi đăng ký) thực tế với nội dung không đồng bộ sau đó. Việc gửi đi này được hoàn thành bởi sổ đăng ký (Registry) viện dẫn cách thức Phản hồi trực tiếp trên giao diện khách hàng đăng ký (Registry Client) như được thực hiện bởi trình ứng dụng khách hàng đăng ký (Registry Client). Cách thức Phản hồi trực tiếp bao gồm một RegistryResponse (phản hồi đăng ký) như dưới đây:

- messageHeader (tiêu đề thông điệp);
- registryResponse (phản hồi đăng ký) bao gồm:
 - thuộc tính status (Success, Failure (thành công, thất bại));
 - registryErrorList (danh sách lỗi đăng ký) tùy chọn.

Phản hồi đồng bộ

Khi một thông điệp gửi đi một cách đồng bộ, người Xử lý dịch vụ thông điệp sẽ duy trì cơ chế liên lạc mở cho đến khi Sổ đăng ký (Registry) gửi lại phản hồi. Thông điệp này sẽ bao gồm:

- messageHeader (tiêu đề thông điệp);
- registryResponse (phản hồi đăng ký) bao gồm:
 - thuộc tính status (Success, Failure (thành công, thất bại));
 - registryErrorList (danh sách lỗi đăng ký) tùy chọn.

6.3.2.3 Các thỏa thuận và hồ sơ hợp tác của sổ đăng ký ebXML

Tiêu chuẩn CPP ebXML [ebCPP] xác định một hồ sơ giao thức hợp tác (Collaboration Protocol Profile) (CPP) và một thỏa thuận giao thức hợp tác (Collaboration Protocol Agreement) (CPA) như là cơ chế cho cả hai phía để chia sẻ thông tin liên quan tới các quy trình kinh doanh tương ứng. Tiêu chuẩn đó giả định rằng một CPA được thống nhất bởi cả 2 bên để họ tham gia vào các giao dịch B2B.

Tiêu chuẩn này không áp đặt việc sử dụng CPA giữa sổ đăng ký (Registry) và khách hàng đăng ký (Registry Client). Tuy nhiên nếu sổ đăng ký (Registry) không sử dụng một CPP, sổ đăng ký (Registry) sẽ cung cấp một cơ chế khác cho khách hàng đăng ký (Registry Client) để nhận các dịch vụ và thông tin khác được cung cấp bởi một CPP. Cơ chế khác có thể là một URL đơn giản.

CPA giữa các khách hàng và sổ đăng ký (Registry) nên mô tả các giao diện mà sổ đăng ký (Registry) và khách hàng đưa ra cho nhau đối với các giao dịch đăng ký. Việc định rõ mẫu CPP của tổ chức đăng ký và mẫu CPP của khách hàng đăng ký (Registry Client) nằm ngoài phạm vi của tiêu chuẩn này.

6.4 Giao diện LifeCycleManager (Quản lý chu kỳ hoạt động)

Đây là giao diện được đưa ra bởi dịch vụ đăng ký để thực hiện chức năng quản lý chu kỳ hoạt động đối tượng của sổ đăng ký (Registry). Các cách thức của nó được viện dẫn bởi khách hàng đăng ký

(Registry Client). Ví dụ, khách hàng có thể sử dụng giao diện này cho các đối tượng đệ trình, để phân loại và liên kết các đối tượng và để phản đối hay gỡ bỏ các đối tượng. Đối với tiêu chuẩn này, nghĩa của các từ đệ trình, phân loại, liên kết, phản đối và gỡ bỏ được tra trong [ebRIM].

Bảng 2 - Tổng kết LifeCycleManager (Quản lý chu kỳ hoạt động)

Tổng kết cách thức của LifeCycleManager (Quản lý chu kỳ hoạt động)	
RegistryResponse (phản hồi đăng ký)	ApproveObjects (ApproveObjectsRequest) Phê chuẩn một hay nhiều đối tượng được đệ trình trước đó.
RegistryResponse (phản hồi đăng ký)	DeprecateObjects (DeprecateObjectsRequest) Phản đối một hay nhiều các đối tượng được đệ trình trước đó.
RegistryResponse (phản hồi đăng ký)	RemoveObjects (RemoveObjectsRequest) Gỡ bỏ một hay nhiều các đối tượng được đệ trình trước đó.
RegistryResponse (phản hồi đăng ký)	SubmitObjects (SubmitObjectsRequest) Đệ trình một hay nhiều đối tượng và siêu dữ liệu (metadata) có thể liên quan như là các Association (liên kết) và Classification (phân loại).
RegistryResponse (phản hồi đăng ký)	UpdateObjects (UpdateObjectsRequest) Cập nhật một hay nhiều các đối tượng được đệ trình trước đó.
RegistryResponse (phản hồi đăng ký)	AddSlots (AddSlotsRequest) AddSlots vào một hay nhiều RegistryEntrys (mục đăng ký).
RegistryResponse (phản hồi đăng ký)	RemoveSlots (RemoveSlotsRequest) RemoveSlots cụ thể khỏi một hay nhiều RegistryEntrys (mục đăng ký).

6.5 Giao diện LifeCycleManager (Quản lý truy vấn)

Đây là giao diện được đưa ra bởi sổ đăng ký (Registry) để thực hiện dịch vụ QueryManager của sổ đăng ký (Registry). Các cách thức của nó được viện dẫn bởi khách hàng đăng ký (Registry Client). Ví dụ, khách hàng có thể sử dụng giao diện này để thể hiện các truy vấn duyệt qua và chuyên sâu hoặc các truy vấn đặc biệt về thông tin đăng ký.

Bảng 3 - QueryManager (Quản lý truy vấn)

Tổng kết cách thức QueryManager (Quản lý truy vấn)	
RegistryResponse (phản hồi đăng ký)	SubmitAdhocQuery (AdhocQueryRequest) Đệ trình một yêu cầu truy vấn đặc biệt.

6.6 Khách hàng đăng ký (Registry Client)

6.6.1 Mô tả khách hàng đăng ký (Registry Client)

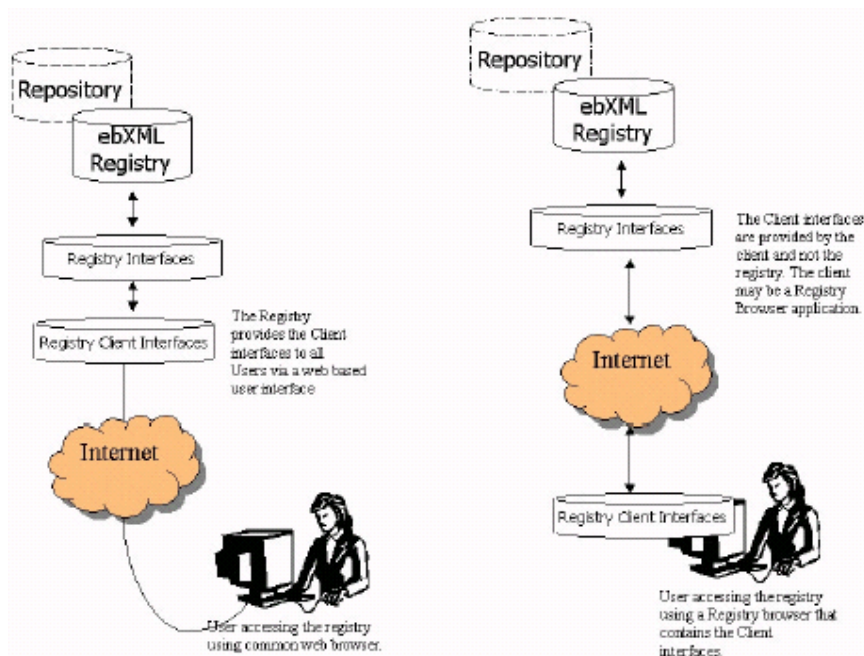
Các giao diện khách hàng đăng ký (Registry Client) có thể dành cho tổ chức đăng ký hoặc cho người sử dụng. Mô hình 5 mô tả hai tô-pô mạng được hỗ trợ bởi cấu trúc đăng ký đối với sổ đăng ký (Registry) và khách hàng đăng ký (Registry Client). Hình bên trái là trường hợp sổ đăng ký (Registry) cung cấp một trang web dựa trên ứng dụng “khách hàng nhỏ” để truy cập sổ đăng ký (Registry) mà luôn sẵn sàng đối với người sử dụng dùng một trình duyệt web thông thường. Trong trường hợp này,

các giao diện khách hàng đăng ký (Registry Client) nằm phía bên kia của Internet và dành cho sổ đăng ký (Registry), thể hiện quan điểm của người sử dụng.

Hình bên phải là trường hợp người sử dụng đang sử dụng một ứng dụng trình duyệt đăng ký “khách hàng lớn” để truy cập sổ đăng ký (Registry).

Trong trường hợp này, các giao diện khách hàng đăng ký (Registry Client) nằm bên trong công cụ Trình duyệt đăng ký và dành cho sổ đăng ký (Registry), thể hiện quan điểm của người sử dụng. Các giao diện khách hàng đăng ký (Registry Client) liên kết với sổ đăng ký (Registry) qua Internet trong trường hợp này.

Một mạng tô-pô thứ ba có khả năng được tạo ra bởi cấu trúc đăng ký là nơi các giao diện khách hàng đăng ký (Registry Client) nằm trong một bộ phận kinh doanh bên máy chủ như bộ phận Kinh doanh thu mua. Trong tô-pô mạng này, có thể không có giao diện người sử dụng trực tiếp hoặc sự can thiệp của người sử dụng. Thay vào đó, bộ phận kinh doanh thu mua có thể truy cập sổ đăng ký (Registry) một cách tự động để lựa chọn những người bán hoặc những người cung cấp dịch vụ có khả năng dựa trên nhu cầu kinh doanh hiện tại.



Hình 5 - Cấu trúc đăng ký hỗ trợ các mạng tô-pô linh hoạt

6.6.2 Tự khởi động truyền thông đăng ký

Trước khi một khách hàng có thể truy cập các dịch vụ của một sổ đăng ký (Registry), cần phải có vài thông tin tự khởi động giữa khách hàng và tổ chức đăng ký. Điểm cần thiết nhất của quy trình tự khởi động này là về phía khách hàng tìm ra thông tin địa chỉ (ví dụ một HTTP URL) tới mỗi giao diện dịch vụ cụ thể của sổ đăng ký (Registry). Khách hàng có thể có được thông tin địa chỉ bằng cách tìm ra sổ đăng ký (Registry) ebXML trong một tổ chức đăng ký công cộng như là UDDI hoặc trong một sổ đăng ký (Registry) ebXML khác.

- trong trường hợp quy định SOAP, tất cả thông tin khách hàng cần (ví dụ các URL của sổ đăng ký (Registry)) sẵn có trong một mô tả WSDL về tổ chức đăng ký. WSDL này tuân theo mô tả WSDL mẫu trong Phụ lục A.1. Mô tả WSDL này có thể được tìm thấy trong một tổ chức đăng ký công cộng như là UDDI;
- trong trường hợp quy định ebMS, thông tin trao đổi giữa khách hàng và tổ chức đăng ký có thể được hoàn thành theo một cách đăng ký cụ thể, mà bao gồm việc tạo ra một CPA giữa khách hàng và tổ chức đăng ký. Khi việc trao đổi thông tin diễn ra sổ đăng ký (Registry) và khách hàng sẽ có thông tin địa chỉ (ví dụ các URL) cho mỗi bên.

6.6.2.1 Tự khởi động thông tin đối với quy định SOAP

Mỗi sổ đăng ký (Registry) ebXML phải cung cấp một mô tả WSDL đối với dịch vụ đăng ký của mình như được nêu ở Phụ lục A.1. Một khách hàng sử dụng mô tả WSDL đó để xác định thông tin địa chỉ của dịch vụ đăng ký bằng một giao thức xác định. Ví dụ các cổng dựa trên SOAP/HTTP của dịch vụ đăng ký có thể được truy cập qua một URL được xác định trong WSDL đối với tổ chức đăng ký.

Việc sử dụng WSDL cho phép khách hàng có thể sử dụng các công cụ tự động như là trình biên dịch WSDL để thu được các bản gốc mà cho phép truy cập tới tổ chức đăng ký bằng một ngôn ngữ xác định.

Ít nhất, mọi khách hàng có thể truy cập tổ chức đăng ký thông qua SOAP/HTTP sử dụng thông tin địa chỉ trong WSDL, với các yêu cầu cơ sở hạ tầng ít nhất hơn là khả năng tạo ra SOAP đồng bộ cho các cổng dựa trên SOAP trong dịch vụ đăng ký.

6.6.2.2 Tự khởi động thông tin cho dịch vụ thông điệp ebXML

Vì không có CPA thiết lập trước giữa sổ đăng ký (Registry) và khách hàng đăng ký (Registry Client) nên khách hàng phải biết ít nhất là 1 địa chỉ liên lạc chuyển tải xác định của sổ đăng ký (Registry). Địa chỉ liên lạc này đặc trưng là một URL đối với sổ đăng ký (Registry), mặc dù nó có thể là một vài dạng địa chỉ khác như là một địa chỉ email. Ví dụ, nếu sổ đăng ký (Registry) sử dụng giao thức truyền thông là HTTP thì địa chỉ liên lạc là URL. Trong ví dụ này, khách hàng sử dụng URL công cộng của sổ đăng ký (Registry) để tạo một CPA ngầm với sổ đăng ký (Registry). Khi khách hàng gửi yêu cầu tới Sổ đăng ký (Registry), thì sẽ cung cấp một URL cho chính mình. Sổ đăng ký (Registry) sử dụng URL của khách hàng để định dạng phiên bản CPA ngầm với khách hàng. Lúc này một tọa đàm được thiết lập trong sổ đăng ký (Registry). Trong suốt quá trình tọa đàm của khách hàng với sổ đăng ký (Registry), các thông điệp có thể được trao đổi trực tiếp theo yêu cầu của các giao thức tương tác được quy định trong tiêu chuẩn này.

6.6.3 Giao diện khách hàng đăng ký (Registry Client)

Đây là giao diện mang tính quy tắc được thực hiện bởi khách hàng đăng ký (Registry Client). Khách hàng cung cấp giao diện này khi tạo một kết nối tới Sổ đăng ký (Registry). Nó cung cấp các cách thức mà sổ đăng ký (Registry) sử dụng để chuyển các phản hồi không đồng bộ tới khách hàng. Lưu ý rằng

một khách hàng không cần cung cấp một giao diện RegistryClient (khách hàng đăng ký) nếu [CPA] giữa khách hàng và tổ chức đăng ký không có hỗ trợ các phức đáp không đồng bộ.

Sổ đăng ký (Registry) gửi tất cả các phức đáp không đồng bộ tới các hoạt động thông qua cách thức phức đáp trực tiếp.

Bảng 4 - Bảng tóm tắt RegistryClient (khách hàng đăng ký)

Tóm tắt phương pháp RegistryClient (khách hàng đăng ký)	
Tránh	onResponse (phản hồi trực tiếp) (RegistryResponse (phản hồi đăng ký)) Lưu ý khách hàng phản hồi được tổ chức đăng ký gửi tới bản yêu cầu được đệ trình trước đó.

6.6.4 Registry Response (Phản hồi đăng ký)

RegistryResponse (phản hồi đăng ký) là một lớp giao diện chung được sổ đăng ký (Registry) xác định và được sổ đăng ký (Registry) sử dụng để gửi các phản hồi tới các bản yêu cầu của khách hàng.

6.7 Các yêu cầu khả năng hoạt động tương tác

6.7.1 Khả năng hoạt động tương tác của khách hàng

Cấu trúc yêu cầu mọi ebXML theo khách hàng đăng ký (Registry Client) có thể truy cập mọi ebXML theo dịch vụ đăng ký bằng cách có thể hoạt động tương tác được. Một sổ đăng ký (Registry) ebXML có thể thực hiện một số quy định giao thức từ nhóm các quy định có tính quy phạm (ebXML TRP và SOAP/HTTP hiện tại) được xác định trong đề xuất này. Hỗ trợ các quy định giao thức thêm là chọn lựa.

6.7.2 Hợp tác đăng ký bên trong

Phiên tiêu chuẩn này không ngăn ngừa các Đăng ký ebXML từ việc hợp tác với nhau tới chia sẻ thông tin, không ngăn ngừa các chủ sở hữu của các Đăng ký ebXML từ việc đăng ký các Sổ đăng ký (Registry) ebXML của họ với các hệ thống, các danh mục hoặc các thư mục đăng ký khác.

Các ví dụ bao gồm:

- một sổ đăng ký (Registry) ebXML phục vụ như một tổ chức đăng ký của các Đăng ký ebXML;
- một sổ đăng ký (Registry) không ebXML phục vụ như một tổ chức đăng ký của các Đăng ký ebXML;
- hợp tác của các Sổ đăng ký (Registry) ebXML, nơi mà các tổ chức đăng ký ebXML đa phương đăng ký với nhau để lập ra một liên đoàn.

7 Dịch vụ LifeCycleManager (Quản lý chu kỳ hoạt động)

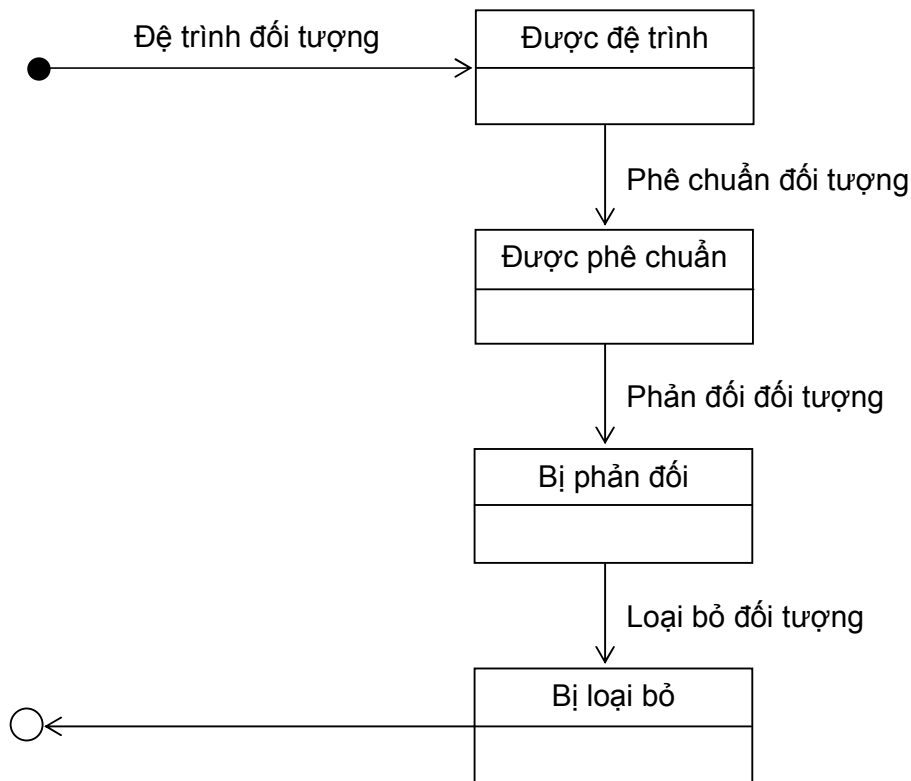
Phần này quy định Dịch vụ LifeCycleManager (quản lý chu kỳ hoạt động) của sổ đăng ký (Registry). Dịch vụ LifeCycleManager (quản lý chu kỳ hoạt động) là một dịch vụ phụ của dịch vụ đăng ký. Nó cung cấp chức năng theo yêu cầu của khách hàng đăng ký (Registry Client) để quản lý chu kỳ hoạt động của các hạng mục kho (ví dụ các tài liệu XML được yêu cầu cho các quy trình kinh doanh ebXML). Dịch vụ LifeCycleManager (quản lý chu kỳ hoạt động) có thể được sử dụng với tất cả các hạng mục kho cũng

như các đối tượng siêu dữ liệu (metadata) được xác định trong [ebRIM] như là Sự phân loại và Sự liên kết.

Chính sách an ninh tối thiểu cho một tổ chức đăng ký ebXML là để chấp nhận nội dung từ bất kỳ khách hàng nào nếu một giấy chứng nhận do một Cơ quan chứng nhận phát hành được thừa nhận bởi tổ chức đăng ký ebXML ký hiệu số hoá nội dung.

7.1 Chu trình của một Hạng mục kho

Mục đích chính của Dịch vụ LifeCycleManager (quản lý chu kỳ hoạt động) là để quản lý chu kỳ hoạt động của các hạng mục kho. Mô hình 6 cho thấy vòng đời đặc trưng của một hạng mục kho. Lưu ý rằng phiên bản hiện thời của tiêu chuẩn này không hỗ trợ đối tượng xác định phiên bản. Đối tượng xác định phiên bản sẽ được thêm vào trong một phiên bản trong tương lai của tiêu chuẩn này.



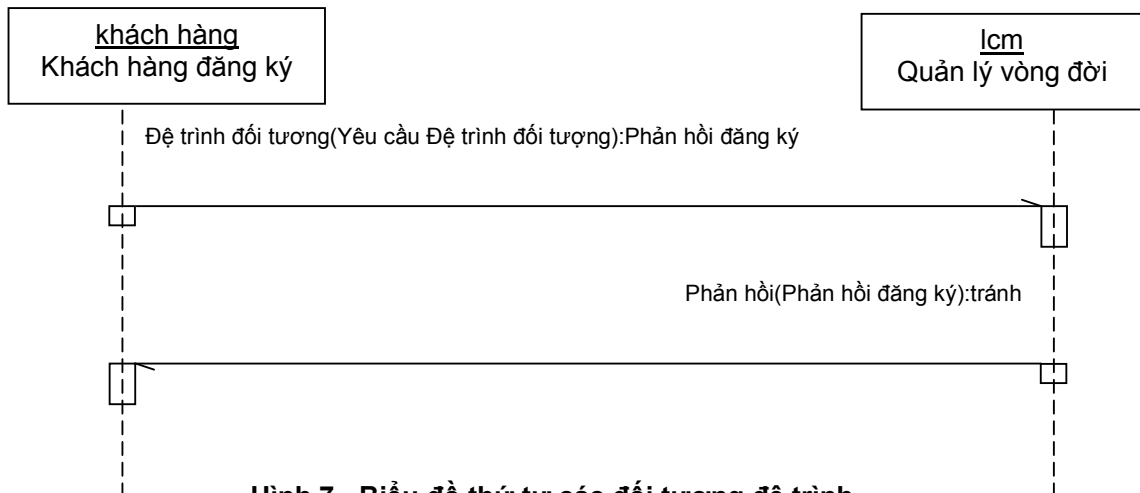
Hình 6 - Chu trình của một hạng mục kho

7.2 Thuộc tính registryObject (đối tượng đăng ký)

Một hạng mục kho được liên kết với một bộ siêu dữ liệu chuẩn được xác định như các thuộc tính của lớp RegistryObject (đối tượng đăng ký) và các lớp phụ của nó được mô tả trong [ebRIM]. Các thuộc tính này cư trú bên ngoài của hạng mục kho thực tế và thông tin mô tả danh mục về hạng mục kho. Các yếu tố XML được gọi là ExtrinsicObject (đối tượng ngoại lai) và các các phần tử khác (xem chi tiết ở Phụ lục B.1) gói gọn tất cả các thuộc tính metadata của đối tượng được quy định trong [ebRIM] như các thuộc tính XML.

7.3 Giao thức đệ trình đối tượng

Phần này mô tả giao thức của dịch vụ đăng ký cho phép một khách hàng đăng ký (Registry Client) đệ trình một hay nhiều hạng mục kho tới kho dữ liệu sử dụng LifecycleManager với tư cách một Tổ chức đệ trình. Nó được thể hiện ở dạng ký hiệu UML được mô tả trong Phụ lục C.



Hình 7 - Biểu đồ thứ tự các đối tượng đệ trình

Chi tiết về giản đồ cho các tài liệu kinh doanh được đưa ra trong quy trình này xem thêm Phụ lục B.

Thông điệp SubmitObjectsRequest bao gồm một phần tử LeafRegistryObjectList.

Phần tử LeafRegistryObjectList xác định một hay nhiều ExtrinsicObjects hay các RegistryEntry khác như Classification, Association, ExternalLink hoặc Packages. .

Một yếu tố ExtrinsicObjects cung cấp siêu dữ liệu (metadata) được yêu cầu về nội dung được đệ trình tới Sổ đăng ký (Registry) được quy định bởi [ebRIM]. Lưu ý rằng các thuộc tính ExtrinsicObjects theo tiêu chuẩn này được phân chia từ mục đối lưu, điều này cho phép Sổ đăng ký (Registry) ebXML ghi danh mục các đối tượng ở mọi dạng đối tượng.

7.3.1 Tạo ID toàn cầu duy nhất

Như được quy định bởi [ebRIM], tất cả các đối tượng trong tổ chức đăng ký có một id duy nhất. id này phải là một nhận dạng duy nhất trên toàn cầu (UUID) và phải tuân theo định dạng của một URN mà xác định một UUID 128 bit DCE như quy định trong [UUID].

(ví dụ: urn:uuid:a2345678-1234-1234-123456789012)

Sổ đăng ký (Registry) thường phát ra id này. Khách hàng có thể cung cấp đặc tính id cho các đối tượng được đệ trình. Nếu khách hàng cung cấp id này và nó phù hợp với nhận dạng của URN mà xác định một UUID 128 bit DCE, sau đó tổ chức đăng ký giả định rằng khách hàng muốn xác định id cho đối tượng. Trong trường hợp này, tổ chức đăng ký phải tôn trọng id khách hàng cung cấp và sử dụng nó như thuộc tính id của đối tượng trong tổ chức đăng ký. Nếu id được tổ chức đăng ký tạo ra không phải là duy nhất trên toàn cầu, tổ chức đăng ký phải đưa ra điều kiện lỗi: Lỗi Id Unavailable.

Nếu khách hàng không cung cấp một id cho một đối tượng được đệ trình thì sau đó tổ chức đăng ký phải phát ra một id duy nhất trên toàn cầu. Dù khách hàng phát ra id hay tổ chức đăng ký phát ra id thì nó cũng phải được sử dụng thuật toán phát ra UUID 128 bit DCE như được quy định trong [UUID].

7.3.2 Thuộc tính id và các tham chiếu đối tượng

Thuộc tính id của một đối tượng có thể được sử dụng bởi các đối tượng khác để tham chiếu đối tượng đầu tiên. Các tham chiếu này thường có trong cả SubmitObjectsRequest cũng như trong sổ đăng ký (Registry). Trong SubmitObjectsRequest, thuộc tính id có thể được dùng để dẫn chiếu tới một đối tượng trong SubmitObjectsRequest cũng như dẫn chiếu tới một đối tượng trong sổ đăng ký (Registry). Một đối tượng trong SubmitObjectsRequest mà cần được dẫn chiếu tới trong tài liệu yêu cầu có thể được người đệ trình gán một id để nó có thể được dẫn chiếu trong bản yêu cầu. Người đệ trình có thể cho đối tượng một URN uuid thích hợp, trong trường đó id được gán vĩnh viễn cho đối tượng trong tổ chức đăng ký. Hoặc là người đệ trình có thể gán một id tùy ý (không phải một URN uuid thích hợp) miễn là id đó là duy nhất trong tài liệu yêu cầu. Trong trường hợp này id có vai trò như một cơ chế nối kết trong tài liệu yêu cầu nhưng phải được tổ chức đăng ký lờ đi và thay thế với một id được phát ra bởi tổ chức đăng ký theo việc đệ trình.

Khi một đối tượng trong SubmitObjectsRequest cần tham chiếu một đối tượng đã tồn tại trong tổ chức đăng ký, bản yêu cầu phải bao gồm một phần tử ObjectRef (tham chiếu đối tượng) mà thuộc tính id của nó là id của đối tượng trong tổ chức đăng ký đó. Id này là quy định một URN uuid thích hợp. Một Tham chiếu đối tượng có thể được xem xét như một sự uỷ quyền trong bản yêu cầu cho một đối tượng có trong tổ chức đăng ký.

7.3.3 Chuỗi đánh giá

RS phải tạo ra đối tượng AuditableEvents (các sự kiện có thể đánh giá được) với kiểu sự kiện được tạo ra cho mỗi RegistryObject (đối tượng đăng ký) có được qua một SubmitObjectsRequest.

7.3.4 Tổ chức đệ trình

RS phải tạo ra một Association những Người đệ trình trong tổ chức đệ trình và mỗi RegistryObject (đối tượng đăng ký) được tạo ra qua một SubmitObjectsRequest. (Tổ chức đệ trình được xác định từ thuộc tính tổ chức của người sử dụng là người đệ trình một SubmitObjectsRequest.)

7.3.5 Xử lý lỗi

Một SubmitObjectsRequest là hạt nhân của cả thành công và thất bại. Trong trường hợp thành công, tổ chức đăng ký gửi một RegistryResponse (phản hồi đăng ký) với nội dung "Success" ("thành công") cho khách hàng. Trong trường hợp thất bại, tổ chức đăng ký gửi một RegistryResponse (phản hồi đăng ký) với nội dung "Failure" ("thất bại") cho khách hàng. Trong trường hợp phản hồi ngay lập tức cho một yêu cầu không đồng bộ, tổ chức đăng ký gửi một RegistryResponse (phản hồi đăng ký) với nội dung "Unavailable" cho khách hàng. Thất bại xảy ra khi một hay nhiều điều kiện lỗi phát sinh trong quá trình đối tượng được đệ trình. Các thông điệp cảnh báo không có nghĩa là yêu cầu bị thất bại. Các quy luật kinh doanh sau áp dụng:

Bảng 5 - Xử lý lỗi các đối tượng đệ trình

Quy tắc kinh doanh	Áp dụng cho	Lỗi/Cảnh báo
ID không phải là duy nhất	Tất cả các lớp	Lỗi
Không được cho phép	Tất cả các lớp	Lỗi
Không tìm thấy đối tượng được tham chiếu	Association, Phân loại, Nút phân loại, Tổ chức	Lỗi
Các liên kết không cho phép kết nối với các đối tượng bị phản đối	Association	Lỗi
Tình trạng đối tượng, phiên bản chính và phiên bản phụ được gắn kết bởi RS, và bị lờ đi nếu được cung cấp	Tất cả các loại	Cảnh báo

7.3.6 Ví dụ về SubmitObjectsRequest (yêu cầu đệ trình đối tượng)

Ví dụ sau chỉ ra một vài trường hợp sử dụng khác nhau trong một SubmitObjectsRequest độc lập. Nó không cho thấy thông điệp SOAP hoặc thông điệp [ebMS] đầy đủ với MessageHeader (tiêu đề thông điệp) và các phần thêm trong thông điệp cho các hạng mục kho.

Một SubmitObjectsRequest bao gồm một Danh sách RegistryObject (đối tượng đăng ký) mà có chứa một số đối tượng được đệ trình. Nó cũng có thể chứa một số Đối tượng tham chiếu để nối kết các đối tượng được đệ trình với các đối tượng đã có trong tổ chức đăng ký.

```
<?xml version = "1.0" encoding = "UTF-8"?>
<SubmitObjectsRequest
xmlns = "urn:oasis:names:tc:ebxml-regrep:registry:xsd:2.0"
xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation = "urn:oasis:names:tc:ebxml-regrep:rim:xsd:2.0 file:///C:/osws/ebxmlrr-
spec/misc/schema/rim.xsd urn:oasis:names:tc:ebxml-regrep:registry:xsd:2.0
file:///C:/osws/ebxmlrr-spec/misc/schema/rs.xsd"
xmlns:rim = "urn:oasis:names:tc:ebxml-regrep:rim:xsd:2.0"
xmlns:rs = "urn:oasis:names:tc:ebxml-regrep:registry:xsd:2.0"
>
<rim:LeafRegistryObjectList>
<!--
The following 3 objects package specified ExtrinsicObject in specified
RegistryPackage, where both the RegistryPackage and the ExtrinsicObject are
being submitted
-->
<rim:RegistryPackage id = "acmePackage1" >
<rim:Name>
<rim:LocalizedString value = "RegistryPackage #1"/>
</rim:Name>
<rim:Description>
<rim:LocalizedString value = "ACME's package #1"/>
</rim:Description>
</rim:RegistryPackage>
<rim:ExtrinsicObject id = "acmeCPPI" >
<rim:Name>
<rim:LocalizedString value = "Widget Profile" />
</rim:Name>
<rim:Description>
<rim:LocalizedString value = "ACME's profile for selling widgets" />
</rim:Description>
</rim:ExtrinsicObject>
<rim:Association id = "acmePackage1-acmeCPPI-Assoc" associationType = "Packages" sourceObject
= "acmePackage1" targetObject = "acmeCPPI" />
<!--
The following 3 objects package specified ExtrinsicObject in specified RegistryPackage,
Where the RegistryPackage is being submitted and the ExtrinsicObject is
already in registry
-->
<rim:RegistryPackage id = "acmePackage2" >
<rim:Name>
```

TCVN ISO/TS 15000-4 : 2007

```
<rim:LocalizedString value = "RegistryPackage #2"/>
</rim:Name>
<rim:Description>
<rim:LocalizedString value = "ACME's package #2"/>
</rim:Description>
</rim:RegistryPackage>
<rim:ObjectRef id = "urn:uuid:a2345678-1234-1234-123456789012"/>
<rim:Association id = "acmePackage2-alreadySubmittedCPP-Assoc" associationType = "Packages"
sourceObject = "acmePackage2" targetObject = "urn:uuid:a2345678-1234-1234-123456789012"/>
<!--
The following 3 objects package specified ExtrinsicObject in specified RegistryPackage,
where the RegistryPackage and the ExtrinsicObject are already in registry
-->
<rim:ObjectRef id = "urn:uuid:b2345678-1234-1234-123456789012"/>
<rim:ObjectRef id = "urn:uuid:c2345678-1234-1234-123456789012"/>
<!-- id is unspecified implying that registry must create a uuid for this object -->
<rim:Association associationType = "Packages" sourceObject = "urn:uuid:b2345678-1234-1234-
123456789012" targetObject = "urn:uuid:c2345678-1234-1234-123456789012"/>
<!--
The following 3 objects externally link specified ExtrinsicObject using
specified ExternalLink, where both the ExternalLink and the ExtrinsicObject
are being submitted
-->
<rim:ExternalLink id = "acmeLink1" >
<rim:Name>
<rim:LocalizedString value = "Link #1"/>
</rim:Name>
<rim:Description>
<rim:LocalizedString value = "ACME's Link #1"/>
</rim:Description>
</rim:ExternalLink>
<rim:ExtrinsicObject id = "acmeCPP2" >
<rim:Name>
<rim:LocalizedString value = "Sprockets Profile" />
</rim:Name>
<rim:Description>
<rim:LocalizedString value = "ACME's profile for selling sprockets"/>
</rim:Description>
</rim:ExtrinsicObject>
<rim:Association id = "acmeLink1-acmeCPP2-Assoc" associationType = "ExternallyLinks"
sourceObject = "acmeLink1" targetObject = "acmeCPP2"/>
<!--
The following 2 objects externally link specified ExtrinsicObject using specified
ExternalLink, where the ExternalLink is being submitted and the ExtrinsicObject
is already in registry. Note that the targetObject points to an ObjectRef in a
previous line
-->
<rim:ExternalLink id = "acmeLink2">
<rim:Name>
<rim:LocalizedString value = "Link #2"/>
</rim:Name>
<rim:Description>
<rim:LocalizedString value = "ACME's Link #2"/>
</rim:Description>
</rim:ExternalLink>
<rim:Association id = "acmeLink2-alreadySubmittedCPP-Assoc" associationType =
"ExternallyLinks" sourceObject = "acmeLink2" targetObject = "urn:uuid:a2345678-1234-1234-
123456789012"/>
<!--
The following 3 objects externally identify specified ExtrinsicObject using specified
ExternalIdentifier, where the ExternalIdentifier is being submitted and the
ExtrinsicObject is already in registry. Note that the targetObject points to an
ObjectRef in a previous line
-->
<rim:ClassificationScheme id = "DUNS-id" isInternal="false" nodeType="UniqueCode" >
<rim:Name>
<rim:LocalizedString value = "DUNS"/>
</rim:Name>
<rim:Description>
<rim:LocalizedString value = "This is the DUNS scheme"/>
</rim:Description>
</rim:ClassificationScheme>
<rim:ExternalIdentifier id = "acmeDUNSID" identificationScheme="DUNS-id" value =
"13456789012">
<rim:Name>
<rim:LocalizedString value = "DUNS" />
</rim:Name>
<rim:Description>
<rim:LocalizedString value = "DUNS ID for ACME"/>
```

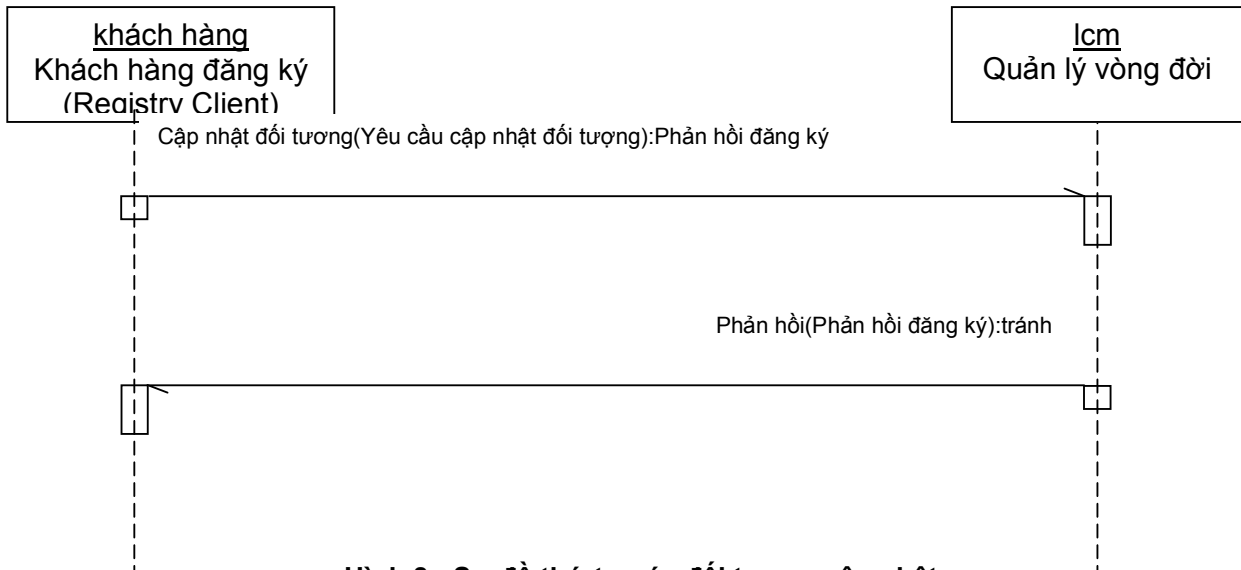
```

</rim:Description>
</rim:ExternalIdentifier>
<rim:Association id = "acmeDUNSID-alreadySubmittedCPP-Assoc" associationType =
"ExternallyIdentifies" sourceObject = "acmeDUNSID" targetObject = "urn:uuid:a2345678-1234-1234-
123456789012" />
<!--
The following show submission of a brand new classification scheme in its entirety
-->
<rim:ClassificationScheme id = "Geography-id" isInternal="true" nodeType="UniqueCode" >
<rim:Name>
<rim:LocalizedString value = "Geography" />
</rim:Name>
<rim:Description>
<rim:LocalizedString value = "This is a sample Geography scheme" />
</rim:Description>
<rim:ClassificationNode id = "NorthAmerica-id" parent = "Geography-id" code =
"NorthAmerica" >
<rim:ClassificationNode id = "UnitedStates-id" parent = "NorthAmerica-id" code =
"UnitedStates" />
<rim:ClassificationNode id = "Canada-id" parent = "NorthAmerica-id" code = "Canada" />
</rim:ClassificationNode>
<rim:ClassificationNode id = "Asia-id" parent = "Geography-id" code = "Asia" >
<rim:ClassificationNode id = "Japan-id" parent = "Asia-id" code = "Japan" >
<rim:ClassificationNode id = "Tokyo-id" parent = "Japan-id" code = "Tokyo" />
</rim:ClassificationNode>
</rim:ClassificationNode>
</rim:ClassificationScheme>
<!--
The following show submission of a Automotive sub-tree of ClassificationNodes that
gets added to an existing classification scheme named 'Industry'
that is already in the registry
-->
<rim:ObjectRef id = "urn:uuid:d2345678-1234-1234-123456789012" />
<rim:ClassificationNode id = "automotiveNode" parent = "urn:uuid:d2345678-1234-1234-
123456789012">
<rim:Name>
<rim:LocalizedString value = "Automotive" />
</rim:Name>
<rim:Description>
<rim:LocalizedString value = "The Automotive sub-tree under Industry scheme" />
</rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode id = "partSuppliersNode" parent = "automotiveNode">
<rim:Name>
<rim:LocalizedString value = "Parts Supplier" />
</rim:Name>
<rim:Description>
<rim:LocalizedString value = "The Parts Supplier node under the Automotive node" />
</rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode id = "engineSuppliersNode" parent = "automotiveNode">
<rim:Name>
<rim:LocalizedString value = "Engine Supplier" />
</rim:Name>
<rim:Description>
<rim:LocalizedString value = "The Engine Supplier node under the Automotive node" />
</rim:Description>
</rim:ClassificationNode>
<!--
The following show submission of 2 Classifications of an object that is already in
the registry using 2 ClassificationNodes. One ClassificationNode
is being submitted in this request (Japan) while the other is already in the registry.
-->
<rim:Classification id = "japanClassification" classifiedObject = "urn:uuid:a2345678-1234-
1234-123456789012" classificationNode = "Japan-id">
<rim:Description>
<rim:LocalizedString value = "Classifies object by /Geography/Asia/Japan node" />
</rim:Description>
</rim:Classification>
<rim:Classification id = "classificationUsingExistingNode" classifiedObject =
"urn:uuid:a2345678-1234-1234-123456789012" classificationNode = "urn:uuid:e2345678-1234-1234-
123456789012">
<rim:Description>
<rim:LocalizedString value = "Classifies object using a node in the registry" />
</rim:Description>
</rim:Classification>
<rim:ObjectRef id = "urn:uuid:e2345678-1234-1234-123456789012" />
</rim:LeafRegistryObjectList>
</SubmitObjectsRequest>

```

7.4 Giao thức cập nhật đối tượng

Phần này mô tả giao thức của dịch vụ đăng ký cho phép một khách hàng đăng ký (Registry Client) cập nhật một hay nhiều mục đăng ký đã có trong bản đăng ký với tư cách một Tổ chức đệ trình. Nó được thể hiện trong ký hiệu UML được mô tả trong Phụ lục C.



Hình 8 - Sơ đồ thứ tự các đối tượng cập nhật

Chi tiết về giản đồ cho các tài liệu kinh doanh được chỉ ra trong quá trình này xem Phụ lục B. Thông điệp UpdateObjectsRequest bao gồm một yếu tố Danh sách đăng ký đối tượng. Danh sách đăng ký đối tượng xác định một hay nhiều RegistryObject (đối tượng đăng ký). Mỗi đối tượng trong danh sách phải là một RegistryObject (đối tượng đăng ký) hiện hành. Các RegistryObject (đối tượng đăng ký) phải bao gồm tất cả các thuộc tính, thậm chí người sử dụng không thể thay đổi. Một thuộc tính bị mất được lý giải như một yêu cầu để gắn kết thuộc tính này với NULL.

7.4.1 Chuỗi đánh giá

RS phải tạo ra đối tượng AuditableEvents (các sự kiện có thể đánh giá được) với kiểu sự kiện được cập nhật cho mỗi RegistryObject (đối tượng đăng ký) được cập nhật qua một UpdateObjectsRequest.

7.4.2 Tổ chức đệ trình

RS phải duy trì một Association những Người đệ trình giữa tổ chức đệ trình và mỗi RegistryObject (đối tượng đăng ký) được cập nhật thông qua một UpdateObjectsRequest. Nếu một UpdateObjectsRequest được chấp nhận từ một tổ chức đệ trình khác, sau đó RS phải gỡ bỏ đối tượng tương ứng cũ và tạo một đối tượng mới. Tất nhiên, Chính sách Kiểm soát truy cập có thể ngăn chặn kiểu cập nhật này tại vị trí đầu tiên. (Tổ chức đệ trình được xác định từ thuộc tính tổ chức của người sử dụng, là người đệ trình một UpdateObjectsRequest)

7.4.3 Xử lý lỗi

Một UpdateObjectsRequest là hạt nhân của cả thành công và thất bại. Trong trường hợp thành công, tổ chức đăng ký gửi lại một RegistryResponse (phản hồi đăng ký) với nội dung "Success" ("thành công") cho khách hàng.

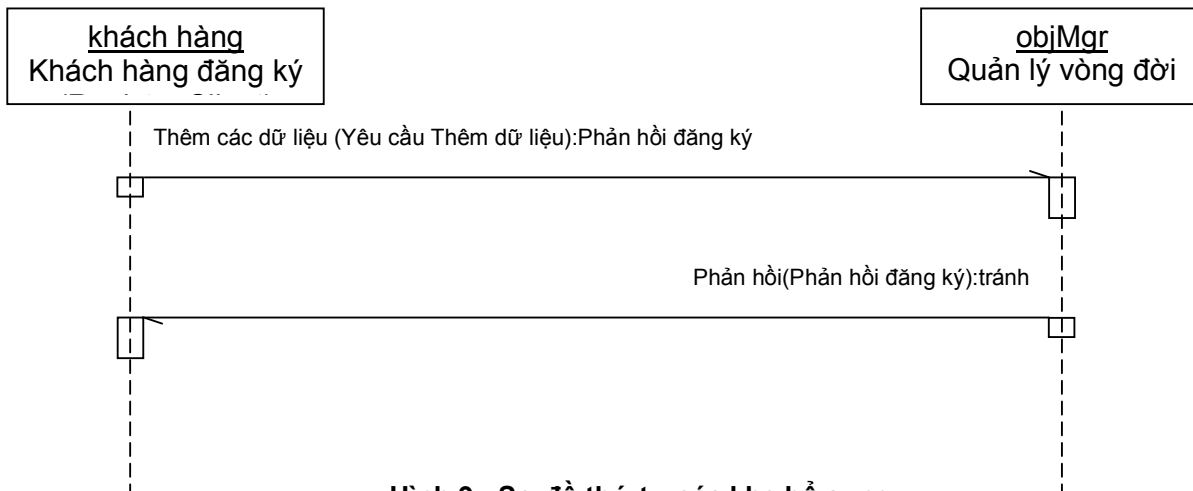
Trong trường hợp thất bại, tổ chức đăng ký gửi lại một RegistryResponse (phản hồi đăng ký) với nội dung "Failure" ("thất bại") cho khách hàng. Trong trường hợp có phản hồi ngay lập tức đối với một yêu cầu không đồng bộ, tổ chức đăng ký gửi lại một RegistryResponse (phản hồi đăng ký) với nội dung "Unavailable" cho khách hàng. Thất bại xảy ra khi một hay nhiều điều kiện lỗi phát sinh trong quá trình các đối tượng được cập nhật. Các thông điệp cảnh báo không có nghĩa là yêu cầu bị thất bại. Các quy tắc kinh doanh sau áp dụng:

Bảng 6 - Xử lý lỗi các đối tượng cập nhật

Quy tắc kinh doanh	Áp dụng cho	Lỗi/Cảnh báo
Không tìm thấy đối tượng	Tất cả các lớp	Lỗi
Không được cho phép	Tất cả các lớp	Lỗi
Không tìm thấy đối tượng tham chiếu	Association, Phân loại, Nút phân loại, Tổ chức	Lỗi
Các liên kết không được phép kết nối với các đối tượng bị phản đối	Association	Lỗi
Tình trạng đối tượng, phiên bản chính và phiên bản phụ không thể thay đổi thông qua Giao thức cập nhật đối tượng, lờ đi nếu được cung cấp	Tất cả các lớp	Cảnh báo
RegistryEntry với tính ổn định = "Ổn định" không nên được cập nhật	Tất cả các lớp	Cảnh báo

7.5 Giao thức thêm các khe dữ liệu

Phần này mô tả giao thức của dịch vụ đăng ký cho phép một khách hàng thêm các dữ liệu vào một RegistryEntry được đệ trình trước đó sử dụng LifeCycleManager. Các dữ liệu tạo ra một cơ chế linh hoạt cho việc mở rộng các RegistryEntry như được quy định bởi [ebRIM].

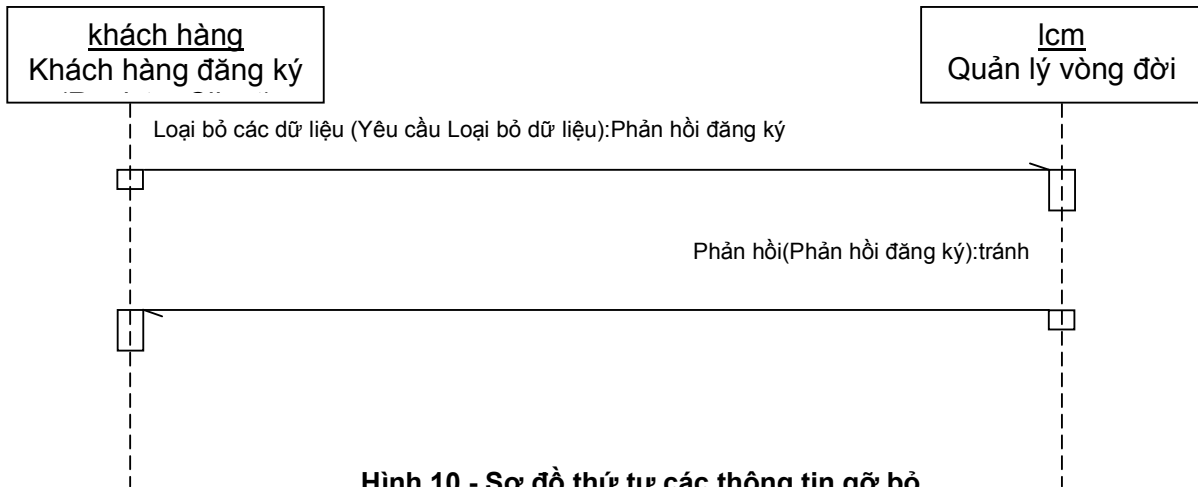


Hình 9 - Sơ đồ thứ tự các khe bổ sung

Trong trường hợp thành công, tổ chức đăng ký gửi một RegistryResponse (phản hồi đăng ký) với nội dung "Success" ("thành công") cho khách hàng. Trong trường hợp thất bại, tổ chức đăng ký gửi RegistryResponse (phản hồi đăng ký) với nội dung "Failure" ("thất bại") cho khách hàng.

7.6 Giao thức gỡ bỏ các khe dữ liệu

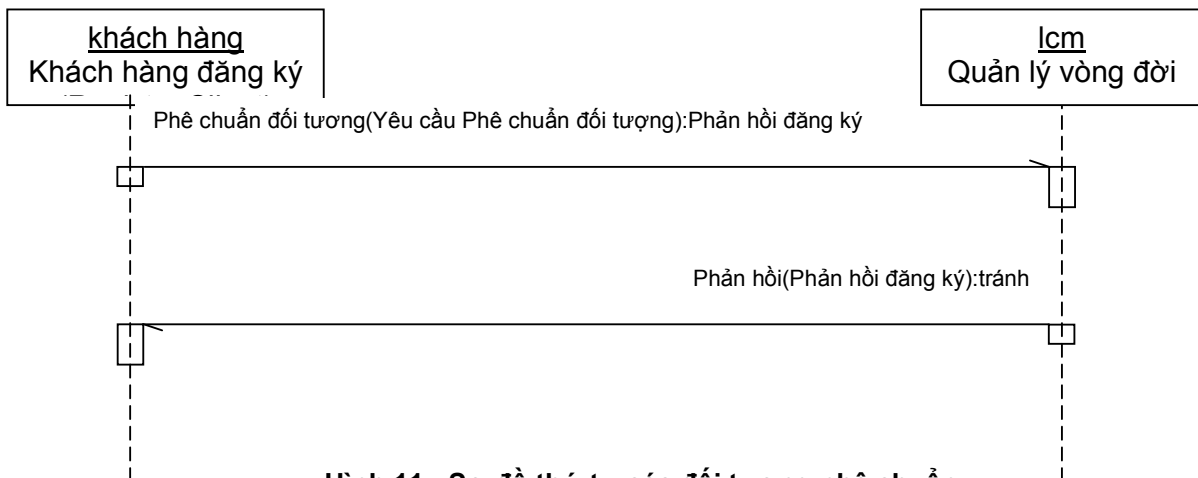
Phần này mô tả giao thức của dịch vụ đăng ký cho phép một khách hàng gỡ bỏ các khe dữ liệu đối với Mục nhập số đăng ký được đệ trình trước đó sử dụng LifecycleManager.



Hình 10 - Sơ đồ thứ tự các thông tin gỡ bỏ

7.7 Giao thức phê chuẩn các đối tượng

Phần này mô tả giao thức của dịch vụ đăng ký cho phép một khách hàng được phê chuẩn một hay nhiều hạng mục kho được đệ trình trước đó sử dụng LifecycleManager. Khi một hạng mục kho được phê chuẩn nó sẽ trở nên có giá trị sử dụng cho các bên kinh doanh (ví dụ trong việc lắp ráp các CPA mới và các hồ sơ giao thức hợp tác (Collaboration Protocol Profile)).



Hình 11 - Sơ đồ thứ tự các đối tượng phê chuẩn

Giản đồ chi tiết cho các tài liệu kinh doanh được chỉ ra trong quá trình này xem thêm Phụ lục B.

7.7.1 Chuỗi đánh giá

RS phải tạo ra các đối tượng AuditableEvents (các sự kiện có thể đánh giá được) với kiểu sự kiện được phê chuẩn cho mỗi RegistryObject (đối tượng đăng ký) được phê chuẩn qua một ApproveObjectsRequest.

7.7.2 Tổ chức đệ trình

RS phải duy trì một Association những Người đệ trình giữa tổ chức đệ trình và mỗi RegistryObject (đối tượng đăng ký) được cập nhật thông qua một ApproveObjectsRequest.

Nếu một ApproveObjectsRequest được chấp nhận từ một tổ chức đệ trình khác, sau đó RS phải gỡ bỏ đối tượng tương ứng cũ và tạo một đối tượng mới. Tất nhiên, Chính sách Kiểm soát truy cập có thể ngăn chặn ApproveObjectsRequest này tại vị trí đầu tiên. (Tổ chức đệ trình được xác định từ thuộc tính tổ chức của người sử dụng, là người đệ trình một ApproveObjectsRequest).

7.7.3 Xử lý lỗi

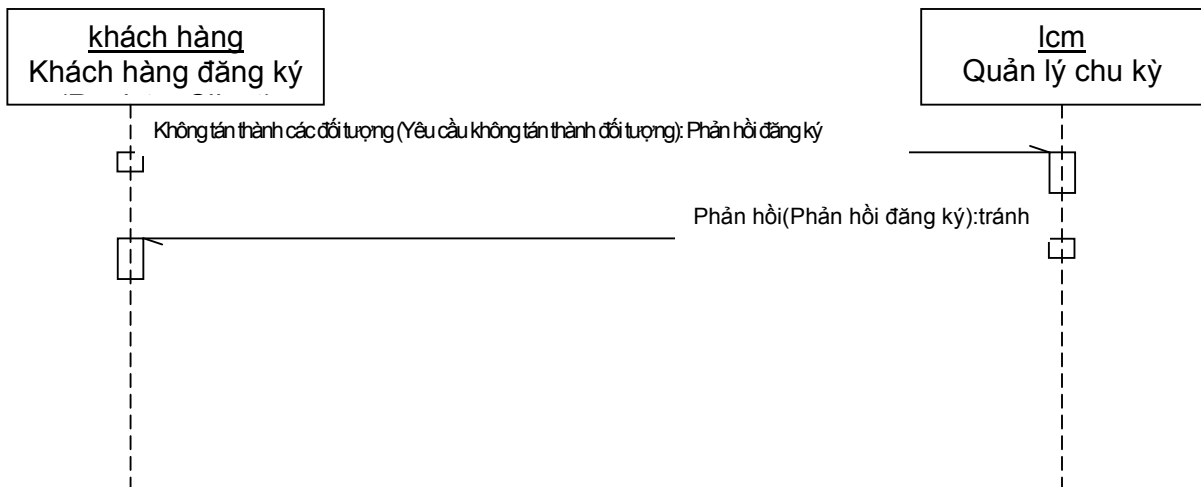
Một ApproveObjectsRequest (yêu cầu phê chuẩn đối tượng) là hạt nhân của cả thành công và thất bại. Trong trường hợp thành công, tổ chức đăng ký gửi lại một RegistryResponse (phản hồi đăng ký) với nội dung "Success" ("thành công") cho khách hàng. Trong trường hợp thất bại, tổ chức đăng ký gửi lại một RegistryResponse (phản hồi đăng ký) với nội dung "Failure" ("thất bại") cho khách hàng. Trong trường hợp có phản hồi ngay lập tức đối với một yêu cầu không đồng bộ, tổ chức đăng ký gửi lại một RegistryResponse (phản hồi đăng ký) với nội dung "Unavailable" ("không có sẵn") cho khách hàng. Thất bại xảy ra khi một hay nhiều điều kiện lỗi phát sinh trong quá trình liệt kê tham chiếu đối tượng. Các thông điệp cảnh báo không có nghĩa là yêu cầu bị thất bại. Các quy tắc kinh doanh sau áp dụng:

Bảng 7 - Xử lý lỗi các đối tượng phê chuẩn

Quy tắc kinh doanh	Áp dụng cho	Lỗi/Cảnh báo
Không tìm thấy đối tượng	Tất cả các lớp	Lỗi
Không được cho phép	Các lớp RegistryEntry	Lỗi
Chỉ RegistryEntry có thể "được phê chuẩn"	Tất cả các lớp khác lớp RegistryEntry	Lỗi
Tình trạng đối tượng đã "Được phê chuẩn"	Các lớp RegistryEntry	Cảnh báo

7.8 Giao thức không tán thành đối tượng

Phần này mô tả giao thức của dịch vụ đăng ký cho phép một khách hàng phản đối một hay nhiều hạng mục kho được đệ trình trước đó sử dụng LifeCycleManager (quản lý chu kỳ hoạt động). Khi một đối tượng bị phản đối, không có các tham chiếu mới (ví dụ các Association, Classifications và ExternalLink mới) để đối tượng này có thể được đệ trình. Tuy nhiên, các tham chiếu tới một đối tượng bị phản đối đang tồn tại tiếp tục thực hiện chức năng bình thường.



Hình 12 - Sơ đồ thứ tự các đối tượng không tán thành

Chi tiết về giản đồ cho các tài liệu kinh doanh được chỉ ra trong quá trình này xem Phụ lục B.

7.8.1 Chuỗi đánh giá

RS phải tạo ra đối tượng AuditableEvents (các sự kiện có thể đánh giá được) với kiểu sự kiện bị phản đối cho mỗi RegistryObject (đối tượng đăng ký) bị phản đối qua một DeprecateObjectsRequest.

7.8.2 Tổ chức đệ trình

RS phải duy trì một Association những Người đệ trình giữa tổ chức đệ trình và mỗi RegistryObject (đối tượng đăng ký) được cập nhật thông qua một DeprecateObjectsRequest. Nếu một DeprecateObjectsRequest được chấp nhận từ một tổ chức đệ trình khác, sau đó RS phải gỡ bỏ đối tượng tương ứng cũ và tạo một đối tượng mới. Tất nhiên, Chính sách Kiểm soát truy cập có thể ngăn chặn DeprecateObjectsRequest này tại vị trí đầu tiên. (Tổ chức đệ trình được xác định từ thuộc tính tổ chức của người sử dụng, là người đệ trình một DeprecateObjectsRequest).

7.8.3 Xử lý lỗi

Một DeprecateObjectsRequest là hạt nhân của cả thành công và thất bại. Trong trường hợp thành công, tổ chức đăng ký gửi lại một RegistryResponse (phản hồi đăng ký) với nội dung "Success" ("thành công") cho khách hàng. Trong trường hợp thất bại, tổ chức đăng ký gửi lại một RegistryResponse (phản hồi đăng ký) với nội dung "Failure" ("thất bại") cho khách hàng. Trong trường hợp có phản hồi ngay lập tức đối với một yêu cầu không đồng bộ, tổ chức đăng ký gửi lại một RegistryResponse (phản hồi đăng ký) với nội dung "Unavailable" cho khách hàng. Thất bại xảy ra khi một hay nhiều điều kiện lỗi phát sinh trong quá trình liệt kê tham chiếu đối tượng. Các thông điệp cảnh báo không có nghĩa là yêu cầu bị thất bại. Các quy tắc kinh doanh sau áp dụng:

Bảng 8 - Xử lý lỗi các đối tượng không tán thành

Quy tắc kinh doanh	Áp dụng cho	Lỗi/Cảnh báo
Không tìm thấy đối tượng	Tất cả các lớp	Lỗi
Không được cho phép	Các lớp RegistryEntry	Lỗi
Chỉ RegistryEntry có thể “bị phản đối”	Tất cả các lớp khác các lớp RegistryEntry	Lỗi
Tình trạng đối tượng đã “bị phản đối”	Các lớp RegistryEntry	Cảnh báo

7.9 Giao thức gỡ bỏ các đối tượng

Phần này mô tả giao thức của dịch vụ đăng ký cho phép một khách hàng gỡ bỏ một hay nhiều trường hợp và/hoặc hạng mục kho RegistryObject (đối tượng đăng ký) sử dụng LifecycleManager (quản lý chu kỳ hoạt động).

Thông điệp RemoveObjectsRequest (yêu cầu gỡ bỏ đối tượng) được khách hàng gửi để gỡ bỏ các trường hợp RegistryObject (đối tượng đăng ký) và/hoặc hạng mục kho. Phần tử RemoveObjectsRequest (yêu cầu gỡ bỏ đối tượng) bao gồm một thuộc tính XML gọi là DeletionScope (phạm vi xoá bỏ) là một bảng liệt kê có thể có các giá trị như được xác định bởi các phần tiếp theo.

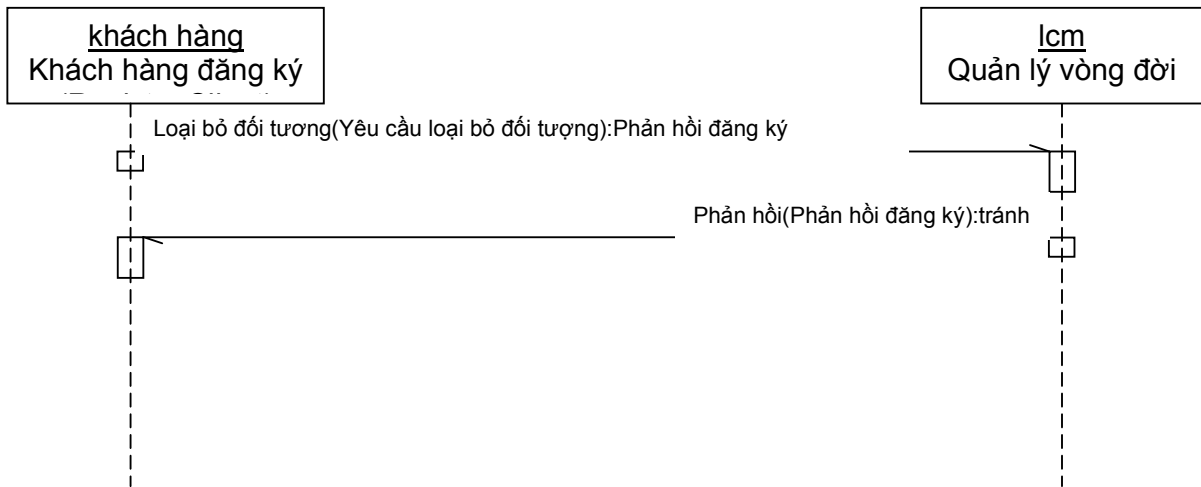
7.9.1 Phạm vi xoá bỏ DeleteRepositoryItemOnly (chỉ xoá hạng mục kho)

DeletionScope (phạm vi xoá bỏ) xác định yêu cầu nên xoá bỏ các hạng mục kho đối với các RegistryEntry cụ thể chứ không xoá bỏ các RegistryEntry cụ thể. Điều này hữu ích cho việc giữ lại các tham chiếu tới các RegistryEntry hợp lệ.

7.9.2 Phạm vi xoá bỏ DeleteAll (xoá tất cả)

DeletionScope (phạm vi xoá bỏ) xác định yêu cầu nên xoá bỏ cả RegistryObject (đối tượng đăng ký) và hạng mục kho đối với các RegistryEntry cụ thể. Nếu tất cả các tham chiếu (ví dụ các Association, Phân loại, Nối kết ngoài) tới một RegistryObject (đối tượng đăng ký) bị gỡ bỏ, có thể sau đó RegistryObject (đối tượng đăng ký) bị gỡ bỏ sử dụng một RemoveObjectsRequest (yêu cầu gỡ bỏ đối tượng) với phạm vi xoá bỏ DeleteAll (xoá tất cả). Cố gắng gỡ bỏ RegistryObject (đối tượng đăng ký) trong khi vẫn có các tham chiếu phát sinh một điều kiện lỗi: Lỗi yêu cầu không hợp lệ.

Giao thức gỡ bỏ đối tượng được thể hiện bằng ký hiệu UML như được mô tả trong Phụ lục C.



Hình 13 - Sơ đồ thứ tự các đối tượng gỡ bỏ

Chi tiết về giản đồ cho các tài liệu kinh doanh được chỉ ra trong quá trình này xem Phụ lục B.

7.9.3 Xử lý lỗi

Một RemoveObjectsRequest (yêu cầu gỡ bỏ đối tượng) là hạt nhân của cả thành công và thất bại. Trong trường hợp thành công, tổ chức đăng ký gửi lại một RegistryResponse (phản hồi đăng ký) với nội dung "Success" ("thành công") cho khách hàng. Trong trường hợp thất bại, tổ chức đăng ký gửi lại một RegistryResponse (phản hồi đăng ký) với nội dung "Failure" ("thất bại") cho khách hàng. Trong trường hợp có phản hồi ngay lập tức đối với một yêu cầu không đồng bộ, tổ chức đăng ký gửi lại một RegistryResponse (phản hồi đăng ký) với nội dung "Unavailable" cho khách hàng. Thất bại xảy ra khi một hay nhiều điều kiện lỗi phát sinh trong quá trình liệt kê tham chiếu đối tượng. Các thông điệp cảnh báo không có nghĩa là yêu cầu bị thất bại. Các quy tắc kinh doanh sau áp dụng:

Bảng 9 - Xử lý lỗi các đối tượng gỡ bỏ

Quy tắc kinh doanh	Áp dụng cho	Lỗi/Cảnh báo
Không tìm thấy đối tượng	Tất cả các lớp	Lỗi
Không được cho phép	Các lớp RegistryObject	Lỗi

8 Dịch vụ quản lý truy vấn

Phần này mô tả khả năng của dịch vụ đăng ký (Registry Service), dịch vụ này cho phép khách hàng (quản lý truy vấn của khách hàng (QueryManagerClient)) tìm kiếm hoặc đưa ra các câu truy vấn đối với các kiểu khác nhau về các đối tượng đăng ký của sổ đăng ký ebXML khi sử dụng giao diện quản lý truy vấn (QueryManager) trong sổ đăng ký. Sổ đăng ký ebXML hỗ trợ những khả năng truy vấn sau:

- truy vấn theo bộ lọc (Filter Query);
- truy vấn SQL (SQL Query).

Cơ chế truy vấn theo bộ lọc (Filter Query) trong phần 8.2 sẽ được hỗ trợ thông qua mọi thực thi đăng kí. Cơ chế truy vấn SQL là một đặc điểm không bắt buộc và có thể được cung cấp bởi một thực thi đăng kí. Tuy nhiên nếu một người nào đó đưa ra một khả năng về truy vấn SQL đến đăng kí ebXML, điều này sẽ chiếu theo cứ liệu này. Những khả năng này là hết sức bình thường và ngẫu nhiên.

Giả định trong tương lai xuất hiện tình huống cụ thể, thì cú pháp W3C XQuery được xem như là một cú pháp về truy vấn khác.

Sổ đăng kí ebXML (ebXML Registry) sẽ giữ lại khả năng tự mô tả giúp xác định tất cả những lựa chọn truy vấn đặc biệt được hỗ trợ. Những thông tin mô tả này được mô tả trong phần phụ lục H.

8.1 Yêu cầu/Phản hồi đối với các truy vấn đặc biệt

Một khách hàng đưa ra một thác mắc đặt biệt tới người quản lý truy vấn thông qua việc gửi AdhocQueryRequest. AdhocQueryRequest này bao gồm một phần tử phụ xác định một truy vấn trong các cơ chế truy vấn sổ đăng kí được hỗ trợ.

Người quản lý truy vấn (QueryManager) gửi AdHocQueryResponse (phản hồi truy vấn đặc biệt) hoặc đồng bộ hoặc thiếu đồng bộ tới khách hàng. AdHocQueryResponse (phản hồi truy vấn đặc biệt) đưa ra một tập hợp các đối tượng gồm kiểu phần tử phụ thuộc vào các thuộc tính responseOption (lựa chọn phản hồi) của AdhocQueryRequest. Đây có thể là các đối tượng đại diện cho lớp phân cấp (leaf class) trong ebRIM hoặc có thể là sự tham chiếu đến các đối tượng trong sổ đăng kí cũng như các lớp trung gian trong ebRIM, ví dụ như RegistryObject (đối tượng đăng kí) và RegistryEntry (mục nhập đăng kí).

Mọi sai sót trong các thông điệp về yêu cầu truy vấn được chỉ ra tương ứng với thông điệp phản hồi truy vấn.



Hình 14 - Sơ đồ thứ tự truy vấn đặc biệt đệ trình

Chi tiết về lược đồ đối với các tài liệu kinh doanh trong quá trình này được đưa ra ở Phụ lục B.2.

Định Nghĩa

```

<element name="AdhocQueryRequest">
  <complexType>
    <sequence>
      <element ref="tns:ResponseOption" minOccurs="1" maxOccurs="1" />
      <choice minOccurs="1" maxOccurs="1">
        <element ref="tns:FilterQuery" />
        <element ref="tns:SQLQuery" />
      </choice>
    </sequence>
  </complexType>
</element>
<element name="AdhocQueryResponse">
  <complexType>
    <choice minOccurs="1" maxOccurs="1">
      <element ref="tns:FilterQueryResult" />
      <element ref="tns:SQLQueryResult" />
    </choice>
  </complexType>
</element>

```

8.1.1 Các tùy chọn phản hồi truy vấn**Mục đích:**

Người quản lý truy vấn (QueryManager) của khách hàng có thể xác định rõ các truy vấn đặc biệt nào có thể trả lời trong phạm vi của AdHocQueryResponse (phản hồi truy vấn đặc biệt) khi sử dụng phần tử ResponseOption (tùy chọn phản hồi) nằm trong AdHocQueryRequest (yêu cầu phản hồi đặc biệt). Phần tử ResponseOption (tùy chọn phản hồi) có một thuộc tính "returnType" ("kiểu phản hồi") và giá trị của nó là:

- ObjectRef (tham chiếu đối tượng) – Tùy chọn này xác định rằng AdHocQueryResponse (phản hồi truy vấn đặc biệt) có thể bao gồm một tập hợp các phần tử ObjectRef (tham chiếu đối tượng) của XML khi được xác định trong [lược đồ ebRIM]. Mục đích của tùy chọn này là phản hồi lại các thẻ định danh của các đối tượng trong sổ đăng ký;
- RegistryObject (đối tượng đăng ký) – Tùy chọn này xác định rằng AdHocQueryResponse (phản hồi truy vấn đặc biệt) có thể bao gồm một tập hợp các phần tử RegistryObject (đối tượng đăng ký) XML như được xác định trong [lược đồ ebRIM].

Trong trường hợp này tất cả các thuộc tính của đối tượng đăng ký được phản hồi lại (ObjectType (kiểu đối tượng), name (tên), description (mô tả)....) bổ sung với các thuộc tính id.

- RegistryEntry (mục nhập đăng ký) - Tùy chọn này xác định rằng AdHocQueryResponse (phản hồi truy vấn đặc biệt) có thể bao gồm một tập hợp RegistryEntry hoặc các phần tử RegistryObject (đối tượng đăng ký) XML như được xác định trong [lược đồ ebRIM], điều này tương ứng với RegistryEntry hoặc các thuộc tính RegistryObject (đối tượng đăng ký);
- leafClass (lớp phân cấp) – Tùy chọn này xác định rằng AdHocQueryResponse (phản hồi truy vấn đặc biệt) có thể bao gồm một tập hợp các phần tử ExtrinsicObjects XML như được xác định trong [lược đồ ebRIM];

- LeafClassWithRepositoryItem (lớp phân cấp cùng với hạng mục kho) – Tùy chọn này xác định rằng AdHocQueryResponse (phản hồi truy vấn đặc biệt) có thể bao gồm một tập hợp các phần tử ExtrinsicObjects XML như được xác định trong [lược đồ ebRIM], nó kèm theo bởi các hạng mục kho (repository items) hoặc RegistryEntry hoặc RegistryObject (đối tượng đăng ký) và các thuộc tính của các phần tử này. Sự kết nối ExtrinsicObjects và hạng mục kho được thực hiện thông qua nội dung URI (content URI) như được giải thích tại phần 8.4 - Khôi phục nội dung (Content Retrieval).

Phần tử ResponseOption (tùy chọn phản hồi) cũng có thuộc tính là “returnComposedObject (phản hồi lại đối tượng được soạn)”. Điều này xác định rõ liệu toàn bộ hệ thống phân cấp về đối tượng được soạn được phản hồi lại thông qua các đối tượng sổ đăng ký.

Nếu “returnType” (kiểu phản hồi) ở mức cao hơn lựa chọn RegistryObject (đối tượng đăng ký), thì sự lựa chọn tốt nhất làm thoả mãn truy vấn được phản hồi. Điều này có thể được mô tả thông qua trường hợp khi OrganizationQuery (truy vấn tổ chức) (truy vấn tổ chức) được yêu cầu phản hồi lại LeafClassWithRepositoryItem (lớp phân cấp cùng với hạng mục kho).

Khi điều này không thể thực hiện, thì người quản lý truy vấn (QueryManager) lựa chọn Lớp phân cấp (leaf class) để thay thế. Nếu OrganizationQuery (truy vấn tổ chức) (truy vấn tổ chức) được yêu cầu khôi phục RegistryEntry như là một kiểu hình phản hồi thì những siêu dữ liệu về RegistryObject (đối tượng đăng ký) sẽ được phản hồi.

Định nghĩa

```
<complexType name="ResponseOptionType">
<attribute name="returnType" default="RegistryObject">
<simpleType>
<restriction base="NMTOKEN">
<enumeration value="ObjectRef" />
<enumeration value="RegistryObject" />
<enumeration value="RegistryEntry" />
<enumeration value="LeafClass" />
<enumeration value="LeafClassWithRepositoryItem" />
</restriction>
</simpleType>
</attribute>
<attribute name="returnComposedObjects" type="boolean" default="false" />
</complexType>
<element name="ResponseOption" type="tns:ResponseOptionType" />
```

8.2 Hỗ trợ truy vấn theo bộ lọc (Filter Query Support)

FilterQuery (truy vấn theo bộ lọc) là một cú pháp XML giúp cung cấp những khả năng truy vấn đơn giản đối với bất kỳ ebXML làm thoả mãn việc thực hiện đăng ký (Registry). Mỗi lựa chọn truy vấn được định hướng dựa vào các kiểu hình giản đơn được xác định bởi Mô hình đăng ký thông tin ebXML (ebRIM: eb XML Registry Information Model). Có hai kiểu bộ lọc truy vấn dựa trên mỗi kiểu hình được truy vấn.

- trước tiên, có Mục tiêu truy vấn sổ đăng ký (RegistryObjectQuery) và RegistryEntryQuery (mục nhập đăng ký) truy vấn (RegistryEntryQuery). Chúng cho phép các truy vấn có đặc điểm chung có

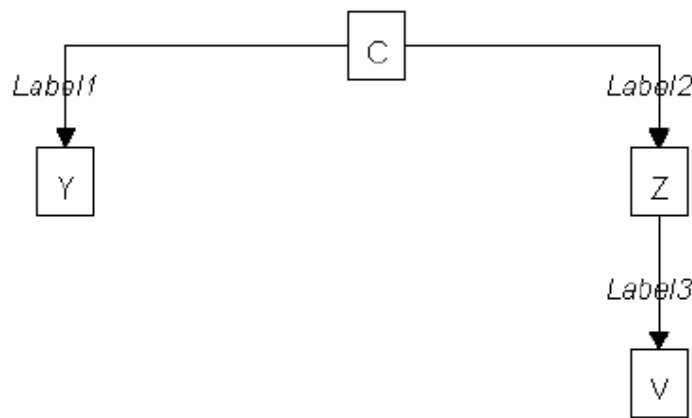
thể phản hồi lại những tiểu kiểu hình khác biệt trong một kiểu hình đang được yêu cầu giải quyết truy vấn. Kết quả của các truy vấn như vậy là một tập hợp của các phần tử XML mà phù hợp với những trường hợp trong bất cứ hạng nào nào (class), làm thoả mãn sự Lựa chọn phản hồi (Option) được xác định trước đó trong phần 8.1.1. Một ví dụ có thể là Mục tiêu đăng kí truy vấn với Lựa chọn phản hồi Lớp phân cấp (leaf class) sẽ phản hồi lại tất cả các thuộc tính của tất cả các trường hợp mà làm thoả mãn truy vấn. Điều này ngụ ý rằng phản hồi có thể đáp lại những phần tử XML mà phù hợp với các kiểu hình như Kế hoạch phân loại (Classification Scheme), Gói đăng ký (Registry Package), Tổ chức và Dịch vụ;

- thứ hai, Bộ lọc truy vấn hỗ trợ các truy vấn về các kiểu hình ebRIM đã được lựa chọn nhằm xác định chính xác những trở ngại của các kiểu kiểu hình này. Phản hồi tới các truy vấn này là bị cản trở hợp lý.

Một khách hàng đệ trình Bộ lọc truy vấn như là một phần của AdhocQueryRequest. Người quản lý truy vấn (QueryManager) gửi một AdHocQueryResponse (phản hồi truy vấn đặc biệt) tới khách hàng, đính kèm với Kết quả bộ lọc truy vấn phù hợp, điều này được cụ thể hoá trong phần 8.1.

Mỗi lựa chọn Bộ lọc truy vấn được liên hệ với một Quy định ebRIM (ebRIM Binding) sẽ xác định một hệ thống phân cấp của các kiểu hình bắt nguồn từ một kiểu hình đơn và sự liên hệ của nó với các kiểu hình khác như được xác định bởi ebRIM.

Mỗi sự lựa chọn về một kiểu hình mà mang tính tiên xác định một văn bản XLM mà có thể được truy vấn như dạng cây. Ví dụ, để C là một kiểu hình, để Y và Z là các kiểu hình có sự liên hệ trực tiếp tới C, và để V là một kiểu hình có thể liên hệ tới Z. Association ebRIM cho C có thể được mô tả trong Hình 15.



Hình 15 - Ví dụ quy định ebRIM

Nhãn 1 xác định sự liên kết từ C đến Y, Nhãn 2 xác định sự liên kết từ C tới Z và Nhãn 3 xác định sự liên kết từ Z tới V. Các nhãn có thể bị bỏ sót nếu không có bất cứ sự mơ hồ nào về bất cứ sự liên kết ebRIM được dự định. Tên của truy vấn được xác định bởi gốc rễ của kiểu hình vv...

Đây là một liên kết ebRIM cho truy vấn C. Giao điểm Y in cây trên bị giới hạn bởi một tập hợp các trường Y mà tập hợp này được liên kết tới C thông qua chuỗi kết hợp được xác định bởi nhãn 1

(Label 1). Tương tự như vậy, giao điểm Z và giao điểm V bị giới hạn bởi các trường mà được lên kết tới các giao điểm gốc thông qua chuỗi liên kết đã được xác định.

Mỗi lựa chọn Bộ lọc truy vấn phụ thuộc vào một hay nhiều lớp lọc khác nhau, tại điểm một lớp lọc là một mệnh đề xác định có giới hạn đối với các thuộc tính của một hạng đơn. Các phương pháp hạng được xác định trong ebRIM và điều này phản hồi lại các kiểu hình cấu hành “các thuộc tính hữu hình”, đó là những lực chọn có giá trị cho những mệnh đề xác định. Tên của các thuộc tính này sẽ tương tự như tên của những phương pháp tương ứng chỉ khác là không có tiền tố ‘get’ (tìm ra). Ví dụ, trường hợp của phương pháp “Cấp độ số tìm ra (getLevelNumber), thì các thuộc tính hữu hình tương thích là “cấp độ số (level number)”. Những lớp lọc được hỗ trợ được làm rõ trong Phần 8.2.13 và những mệnh đề xác định được hỗ trợ được làm rõ trong phần 8.2.14. Một bộ lọc truy vấn sẽ được cấu thành từ những phần tử xuyên suốt cây ebRIM Binding để xác định nhánh nào mà làm thoả mãn những lớp lọc xác định, và kết quả truy vấn sẽ là một chuỗi các trường giúp hỗ trợ những nhánh tương tự.

Trong ví dụ trên, phần tử truy vấn C sẽ bao gồm ba tiểu phần tử, một là Bộ lọc C đối với lớp C với mục đích xoá những trường C mà không làm thoả mãn sự xác định của bộ lọc C, kế đến là bộ lọc Y đối với lớp Y với mục đích những nhánh từ C đến Y, nơi mà mục đích của chuỗi liên kết không làm thoả mãn bộ lọc Y, và thứ ba là để xoá những nhánh dọc theo con đường từ C qua Z để tới V. Phần tử thứ ba được gọi là nhánh phần tử bởi nó cho phép lớp lọc trong mỗi lớp dọc theo con đường từ C tới Y.

Một cách tổng quan, phần tử nhánh sẽ có những tiểu phần tử đó là chính những lớp lọc, những phần tử nhánh khác hoặc truy vấn đã bị hoá giải hoàn toàn đối với lớp ở trên con đường.

Nếu một liên kết từ hạng C tới hạng Y là một tới không hoặc từ một tới một, sau đó tại nhánh một nhiều nhất, bộ lọc hoặc phần tử truy vấn đối với Y được cho phép.

Tuy nhiên, nếu liên kết là một tới nhiều, thì nhánh đa dạng, những phần tử bộ lọc hay những phần tử truy vấn được cho phép. Điều này cho phép một làm rõ rằng một trường của C phải có sự kết nối với những trường đa dạng của Y trước khi trường C được cho là làm thoả mãn phần tử nhánh này.

Cú pháp Bộ lọc truy vấn được gắn chặt với sự kết nối các cấu trúc được xác định trong ebRIM. Vì ebRIM được cho là ổn định, cú pháp Lọc truy vấn mang tính ổn định. Tuy nhiên, nếu một cấu trúc mới được thêm vào ebRIM, thì cú pháp Lọc truy vấn và ngữ nghĩa của nó có thể được mở rộng vào cùng thời điểm. Cũng vậy, của pháp Bộ lọc truy vấn tiếp theo sau sự kế thừa có thức bậc của ebRIM, điều này có nghĩa là các truy vấn hạng phụ thừa hưởng từ các truy vấn liên hạng tương ứng. Những cấu trúc của các phần tử XML mà phù hợp với bởi những hạng ebRIM được lý giải tại {ebRIM scheme}. Tên của những Bộ lọc, Các truy vấn và những Nhánh tương ứng với các tên trong ebRIM bất cứ khi nào có thể.

Những đoạn nói về Kết nối ebRIM được trình bày trong phần 8.2.2 tới 8.2.12 dưới đây xác định thứ bậc thực sự đối với mỗi lựa chọn truy vấn. Những nguyên tắc mang tính ngữ nghĩa đối với mỗi lựa chọn truy vấn sẽ xác định tính hiệu quả của sự kết nối đối với các truy vấn mang tính ngữ nghĩa.

8.2.1 Bộ lọc truy vấn

Mục đích:

Nhằm xác định một tập hợp các truy vấn mà traverse (làm rõ, ngăn cản) hạng đăng ký cụ thể. Mỗi sự lựa chọn sẽ đảm nhận một sự liên kết cụ thể đối với ebRIM. Trạng thái này có thể là một chỉ báo về sự thành công hoặc nó là một tập hợp về sự cảnh báo hay sự gỡ bỏ.

Định nghĩa

```
<element name="FilterQuery">  
<complexType>  
<choice minOccurs="1" maxOccurs="1">  
<element ref="tns:RegistryObjectQuery" />  
<element ref="tns:RegistryEntryQuery" />  
<element ref="tns:AssociationQuery" />  
<element ref="tns:AuditableEventQuery" />  
<element ref="tns:ClassificationQuery" />  
<element ref="tns:ClassificationNodeQuery (truy vấn nút phân loại)" />  
<element ref="tns:ClassificationSchemeQuery (truy vấn giản đồ phân loại)" />  
<element ref="tns:RegistryPackageQuery (truy vấn gói đăng ký)" />  
<element ref="tns:ExtrinsicObjectQuery (truy vấn đối tượng ngoại lai)" />  
<element ref="tns:OrganizationQuery (truy vấn tổ chức)" />  
<element ref="tns:OrganizationQuery (truy vấn dịch vụ)" />
```

Nguyên tắc ngữ nghĩa

1. nguyên tắc ngữ nghĩa đối với mỗi lựa chọn Bộ lọc truy vấn được quy định tại thứ tự các tiểu phần (tiểu khu);
2. nguyên tắc ngữ nghĩa làm rõ thủ tục cho sự thực thi lượng giá về những bộ lọc truy vấn. Việc thực thi không cần thiết phải tuân theo những thủ tục giống nhau đã được định sẵn mà là những hiệu lực tương tự cần được đạt tới;
3. mỗi Kết quả bộ lọc truy vấn (Each FilterQueryResult) là một tập hợp những phần tử XML nhằm xác định mỗi trường của tập hợp các kết quả. Mỗi thuộc tính XML mang một giá trị xuất phát từ giá trị thuộc tính được xác định trong Mô hình đăng ký thông tin (Registry Information Model [[lược đồ ebRIM]]);
4. đối với mỗi phần tử phụ của Bộ lọc truy vấn chỉ có một phần tử phụ Kết quả bộ lọc truy vấn tương ứng mà sẽ được chuyển lại như là một phản hồi. Tên Hạng của phần tử phụ của Kết quả bộ lọc truy vấn phải phù hợp với tên hạng của phần tử phụ của Bộ lọc truy vấn;
5. nếu một Bộ lọc, Nhánh hay phần tử Truy vấn có một hạng không có các phần tử phụ thì những trường kiện định của hạng sẽ làm thoả mãn Bộ lọc, Nhánh hay Truy vấn;
6. nếu một điều kiện lỗi xuất hiện trong suốt quá trình thực hiện của Bộ lọc truy vấn, thì trạng thái thuộc tính của Kết quả đăng ký XML sẽ đi vào "Failure" ("thất bại") và không có phần tử Kết quả truy vấn đặc biệt được phản hồi lại, thay vào đó, phần tử Danh Mục lỗi đăng ký (RegistryErrorList element) sẽ được phản hồi lại với phần tử Tin cậy nhất (highestSeverity element) báo "lỗi". ít nhất một trong những phần tử Đăng ký lỗi (RegistryError) trong danh sách Đăng ký lỗi sẽ có thuộc tính tin cậy để báo "lỗi";
7. nếu không có điều kiện lỗi nào xuất hiện trong suốt quá trình thực hiện Bộ lọc truy vấn, thì trạng thái thuộc tính của Kết quả đăng ký XML sẽ báo "Success" ("thành công") và phần tử về Kết quả bộ lọc truy vấn mang tính hợp lý sẽ bao gồm trong đó. Nếu Danh sách lỗi đăng ký cũng được phản

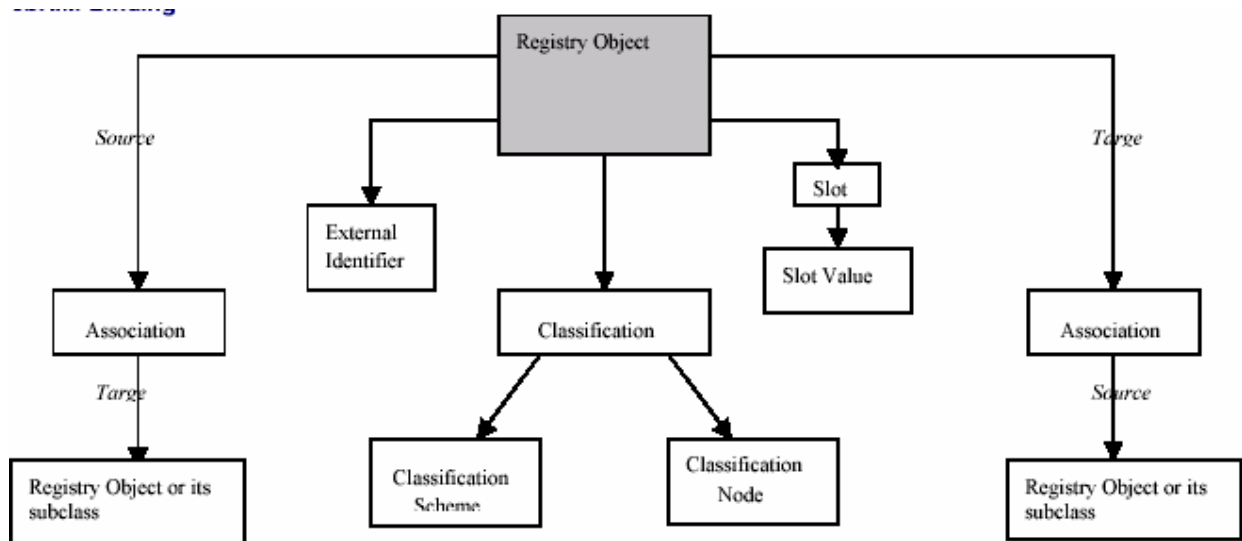
hồi, thì thuộc tính Tin cậy tốt nhất (highestSeverity) của Danh mục Đăng ký lỗi sẽ hiển thị “cảnh báo” và thuộc tính tin cậy của Lỗi đăng ký sẽ hiển thị “cảnh báo”.

8.2.2 RegistryObjectQuery (đối tượng truy vấn sổ đăng ký)

Mục đích:

Nhằm xác định một tập hợp những trường của mục đích đăng ký như là kết quả của thả mắc đối với những dữ liệu đăng ký được lựa chọn.

Quy định ebRIM (ebRIM Binding)



Định nghĩa

```

<complexType name="RegistryObjectQueryType">
  <sequence>
    <element ref="tns:RegistryObjectFilter" minOccurs="0" maxOccurs="1" />
    <element ref="tns:ExternalIdentifierFilter" minOccurs="0" maxOccurs="unbounded" />
    <element ref="tns:AuditableEventQuery" minOccurs="0" maxOccurs="unbounded" />
    <element ref="tns:NameBranch" minOccurs="0" maxOccurs="1" />
    <element ref="tns:DescriptionBranch" minOccurs="0" maxOccurs="1" />
    <element ref="tns:ClassifiedByBranch" minOccurs="0" maxOccurs="unbounded" />
    <element ref="tns:SlotBranch" minOccurs="0" maxOccurs="unbounded" />
    <element ref="tns:SourceAssociationBranch" minOccurs="0" maxOccurs="unbounded" />
    <element ref="tns:TargetAssociationBranch" minOccurs="0" maxOccurs="unbounded" />
  </sequence>
</complexType>
<element name="RegistryObjectQuery" type="tns:RegistryObjectQueryType" />
<complexType name="LeafRegistryObjectListType">
  <choice minOccurs="0" maxOccurs="unbounded">
    <element ref="tns:ObjectRef" />
    <element ref="tns:Association" />
    <element ref="tns:AuditableEvent" />
    <element ref="tns:Classification" />
    <element ref="tns:ClassificationNode" />
    <element ref="tns:ClassificationScheme" />
    <element ref="tns:ExternalIdentifier" />
    <element ref="tns:ExternalLink" />
    <element ref="tns:ExtrinsicObject" />
    <element ref="tns:Organization" />
    <element ref="tns:RegistryPackage" />
    <element ref="tns:Service" />
  </choice>
</complexType>
  
```

```

<element ref="tns:ServiceBinding" />
<element ref="tns:SpecificationLink" />
<element ref="tns:User" />
</choice>
</complexType>
<complexType name="RegistryObjectListType">
<complexContent>
<extension base="tns:LeafRegistryObjectListType">
<choice minOccurs="0" maxOccurs="unbounded">
<element ref="tns:RegistryEntry" />
<element ref="tns:RegistryObject" />
</choice>
</extension>
</complexContent>
</complexType>
<element name="RegistryObjectQueryResult" type="rim:RegistryObjectListType" />
<complexType name="InternationalStringBranchType">
<sequence>
<element ref="tns:LocalizedStringFilter" minOccurs="0" maxOccurs="unbounded" />
</sequence>
</complexType>
<complexType name="AssociationBranchType">
<sequence>
<element ref="tns:AssociationFilter" minOccurs="0" maxOccurs="1" />
<choice minOccurs="0" maxOccurs="1">
<element ref="tns:ExternalLinkFilter" minOccurs="0" maxOccurs="1" />
<element ref="tns:ExternalIdentifierFilter" minOccurs="0" maxOccurs="1" />
<element ref="tns:RegistryObjectQuery" minOccurs="0" maxOccurs="1" />
<element ref="tns:RegistryEntryQuery" minOccurs="0" maxOccurs="1" />
<element ref="tns:AssociationQuery" minOccurs="0" maxOccurs="1" />
<element ref="tns:ClassificationQuery" minOccurs="0" maxOccurs="1" />
<element ref="tns:ClassificationSchemeQuery (truy vấn giản đồ phân loại)"
minOccurs="0" maxOccurs="1" />
<element ref="tns:ClassificationNodeQuery (truy vấn nút phân loại)" minOccurs="0"
maxOccurs="1" />
<element ref="tns:OrganizationQuery (truy vấn tổ chức)" minOccurs="0" maxOccurs="1"
/>
<element ref="tns:AuditableEventQuery" minOccurs="0" maxOccurs="1" />
<element ref="tns:RegistryPackageQuery (truy vấn gói đăng ký)" minOccurs="0"
maxOccurs="1" />
<element ref="tns:ExtrinsicObjectQuery (truy vấn đối tượng ngoại lai)" minOccurs="0"
maxOccurs="1" />
<element ref="tns:OrganizationQuery (truy vấn dịch vụ)" minOccurs="0" maxOccurs="1"
/>
<element ref="tns:UserBranch" minOccurs="0" maxOccurs="1" />
<element ref="tns:ServiceBindingBranch" minOccurs="0" maxOccurs="1" />
<element ref="tns:SpecificationLinkBranch" minOccurs="0" maxOccurs="1" />
</choice>
</sequence>
</complexType>
<element name="SourceAssociationBranch" type="tns:AssociationBranchType" />
<element name="TargetAssociationBranch" type="tns:AssociationBranchType" />
<element name="ClassifiedByBranch">
<complexType>
<sequence>
<element ref="tns:ClassificationFilter" minOccurs="0" maxOccurs="1" />
<element ref="tns:ClassificationSchemeQuery (truy vấn giản đồ phân loại)"
minOccurs="0" maxOccurs="1" />
<element ref="tns:ClassificationNodeQuery (truy vấn nút phân loại)" minOccurs="0"
maxOccurs="1" />
</sequence>
</complexType>
</element>

```

```

<element name="SlotBranch">
<complexType>
<sequence>
<element ref="tns:SlotFilter" minOccurs="0" maxOccurs="1" />
<element ref="tns:SlotValueFilter" minOccurs="0" maxOccurs="unbounded" />
</sequence>
</complexType>
</element>
<element name = "UserBranch">
<complexType>
<sequence>
<element ref = "tns:UserFilter" minOccurs = "0" maxOccurs="1"/>
<element ref = "tns:PostalAddressFilter" minOccurs = "0" maxOccurs="1"/>
<element ref = "tns:TelephoneNumberFilter" minOccurs = "0" maxOccurs="unbounded"/>
<element ref = "tns:EmailAddressFilter" minOccurs = "0" maxOccurs="unbounded"/>
<element ref = "tns:OrganizationQuery (truy vấn tổ chức)" minOccurs = "0"
maxOccurs="1"/>
</sequence>
</complexType>
</element>
<complexType name="ServiceBindingBranchType">
<sequence>
<element ref="tns:ServiceBindingFilter" minOccurs="0" maxOccurs="1" />
<element ref="tns:SpecificationLinkBranch" minOccurs="0" maxOccurs="unbounded" />
<element ref="tns:ServiceBindingTargetBranch" minOccurs="0" maxOccurs="1" />
</sequence>
</complexType>
<element name="ServiceBindingBranch" type="tns:ServiceBindingBranchType" />
<element name="ServiceBindingTargetBranch" type="tns:ServiceBindingBranchType" />
<element name="SpecificationLinkBranch">
<complexType>
<sequence>
<element ref="tns:SpecificationLinkFilter" minOccurs="0" maxOccurs="1" />
<element ref="tns:RegistryObjectQuery" minOccurs="0" maxOccurs="1" />
<element ref="tns:RegistryEntryQuery" minOccurs="0" maxOccurs="1" />
</sequence>
</complexType>
</element>

```

Nguyên tắc ngữ nghĩa

1. để Mục đích đăng ký (RO) biểu thị một tập hợp tất cả trường RegistryObject (đối tượng đăng ký)
 - kiến định trong mục Đăng ký. Các bước tiếp theo sẽ xoá các trường trong RO mà nó không thảo
 mãn những điều kiện của bộ lọc xác định.
 - a) nếu RO trống rỗng thì hãy chuyển đến số 2 dưới đây;
 - b) nếu Bộ lọc đối tượng đăng ký không được xác định thì chuyển tới bước tiếp theo, mặt khác, để x
 làm đối tượng đăng ký trong RO. Nếu x không thoả mãn Bộ lọc đối tượng đăng ký, thì hãy kiểu x
 khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề;
 - c) nếu phần tử Bộ lọc định danh ngoài (ExternalIdentifierFilter) không được xác định, thì chuyển tới
 bước tiếp theo, mặt khác, để x là đối tượng đăng ký đang duy trì trong RO. Nếu x không được liên
 kết tới ít nhất một trường định danh ngoài, thì kiểu x khỏi RO, mặt khác, xử lý mỗi phần tử Bộ lọc
 định danh ngoài tách biệt khỏi nhau như sau: Để EI là một tập hợp của những trường định danh
 bên ngoài mà làm thoả mãn Bộ lọc định danh ngoài và được liên kết tới x. Nếu EI là trống rỗng, thì
 kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề;

d) nếu Truy vấn về sự kiện có thể kiểm tra (AuditableEventQuery) không được xác định thì chuyển tới bước tiếp theo; mặt khác, để x là đối tượng đăng ký được duy trì trong RO. Nếu x không có sự kiện có thể kiểm tra mà làm thoả mãn AuditableEventQuery (truy vấn sự kiện có thể kiểm tra) như được xác định trong phần 8.2.5 thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề;

e) nếu Tên Nhánh (NameBranch) không được xác định thì chuyển tới bước tiếp theo; mặt khác, để x là đối tượng đăng ký được duy trì trong RO. Nếu x không có tên thì kiểu x khỏi RO.

Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề; mặt khác xử lý Tên Nhánh như sau: Nếu bất cứ Bộ lọc Chuỗi được khu biệt (LocalizedStringFilter) mà được xác định là không thoả mãn ít nhất một trong các Chuỗi được khu biệt mà có tác dụng cấu thành tên của mục đích đăng kí thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề;

f) nếu một Mô tả nhánh (DescriptionBranch) không được xác định thì chuyển tới bước tiếp theo; mặt khác, để x là đối tượng được duy trì trong RO. Nếu x không có tên thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề; mặt khác xử lý Mô tả nhánh như sau: Nếu bất cứ Bộ lọc LocalizedString (chuỗi ký tự được địa phương hóa) nào mà được xác định nhưng không thoả mãn bởi một số trong LocalizedString (chuỗi ký tự được địa phương hóa) có tác dụng cấu thành sự mô tả mục đích đăng kí thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề;

g) nếu một phần tử Nhánh được phân loại (ClassifiedByBranch) không được xác định, thì chuyển tới bước tiếp theo; mặt khác, để x là một đối tượng đăng ký được duy trì trong RO. Nếu x không là đối tượng được phân loại của ít nhất một trường phân loại, thì kiểu x khỏi RO; mặt khác, xử lý mỗi phần tử Nhánh được phân loại riêng rẽ như sau: Nếu không có Bộ lọc phân loại (ClassificationFilter) được xác định trong phạm vi Nhánh được phân loại thì để CL là tập hợp của tất cả những trường Phân loại mà làm thoả mãn Bộ lọc Phân loại và có x như là đối tượng được phân loại. Nếu CL trống rỗng thì kiểu x khỏi RO và tiếp tục số nguyên tắc liền kề. Mặt khác, nếu CL không trống rỗng, và nếu Kế hoạch phân loại truy vấn (ClassificationSchemeQuery (truy vấn giản đồ phân loại)) được xác định, thì thay thế CL bằng tập hợp những trường Phân loại được duy trì trong CL bao gồm kế hoạch phân loại xác định thoả mãn Kế hoạch phân loại truy vấn. Nếu CL mới trống rỗng thì kiểu x khỏi RO và tiếp tục số nguyên tắc liền kề. Mặt khác, nếu những duy trì CL trống rỗng, và nếu Giao điểm phân loại truy vấn (ClassificationNodeQuery (truy vấn nút phân loại)) được xác định thì thay CL bằng tập hợp trường phân loại được duy trì nằm trong CL để cho điểm giao phân loại được tồn tại và để cho điểm giao phân loại thoả mãn Giao điểm phân loại truy vấn. Nếu CL mới trống rỗng, thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề;

h) nếu phần tử Vị trí Nhánh (SlotBranch) không được xác định thì chuyển tới bước tiếp theo; mặt khác để x là đối tượng đăng ký được duy trì trong RO. Nếu x được liên kết với ít nhất một trường Vị trí thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề; mặt khác, xử lý mỗi phần tử Nhánh vị trí riêng biệt như sau: Nếu Bộ lọc vị trí không được xác định trong phạm vi Nhánh vị trí thì để SL là tập hợp của tất cả trường Vị trí cho x; mặt khác, để SL là tập hợp trường Vị trí mà thoả mãn Bộ lọc vị trí và là trường Vị trí cho x. nếu SL trống rỗng thì kiểu x khỏi RO và tiếp tục số

nguyên tắc liền kề. Mặt khác, nếu việc duy trì SL là trống rỗng, và nếu Bộ lọc Giá trị Vị trí được xác định, thì thay thế SL bằng tập hợp trường Vị trí được duy trì trong SL để mọi Bộ lọc giá trị vị trí được xác định có hiệu lực. Nếu SL trống rỗng thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề;

- i) nếu Nhánh liên kết nguồn (SourceAssociationBranch) không được xác định thì chuyển tới bước tiếp theo; mặt khác để x là đối tượng đăng ký được duy trì trong RO. N

Nếu x không là đối tượng nguồn cầu ít nhất một trường Association thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề; mặt khác, xử lý phần tử Nhánh liên kết nguồn tách biệt như sau:

Nếu không có Bộ lọc liên kết (AssociationFilter) được xác định cụ thể trong khuôn khổ Nhánh liên kết nguồn (SourceAssociationBranch), thì để AR như là tập hợp của tất cả các trường Association mà có x như là đối tượng nguồn; mặt khác, để AF như là tập hợp trường Association mà làm thoả mãn Bộ lọc liên kết và có x như là đối tượng nguồn. Nếu AR trống rỗng, thì kiểu x khỏi RO.

Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu Bộ lọc liên kết bên ngoài (ExternalLinkFilter) được xác định cụ thể trong khuôn khổ Nhánh liên kết nguồn, thì để ROT như là tập hợp của những trường Association bên ngoài mà làm thoả mãn Bộ lọc liên kết ngoài và là đối tượng đích của một số phần tử AF. Nếu ROT trống rỗng, thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu Bộ lọc xác định bên ngoài (ExternalIdentifierFilter) được xác định cụ thể trong khuôn khổ Nhánh nguồn liên kết, thì để ROT như là tập hợp của những trường Xác định bên ngoài mà làm thoả mãn Bộ lọc xác định bên ngoài và là đối tượng đích của một số phần tử AR. Nếu ROT trống rỗng, thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu một truy vấn đối tượng đăng ký (RegistryObjectQuery) được quy định trong khuôn khổ Nhánh nguồn liên kết, thì để ROT như là tập hợp những trường RegistryObject (đối tượng đăng ký) mà làm thoả mãn Truy vấn đối tượng đăng ký và là đối tượng đích của một số phần tử AF. Nếu ROT trống rỗng, thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu một Kế hoạch phân loại truy vấn (ClassificationSchemeQuery (truy vấn giản đồ phân loại)) được quy định trong khuôn khổ Nhánh nguồn liên kết, thì để ROT như là tập hợp của những trường Kế hoạch phân loại mà làm thoả mãn Kế hoạch phân loại truy vấn và là đối tượng đích của một số phần tử AR. Nếu ROT trống rỗng thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu một Truy vấn Nut phân loại được quy định trong khuôn khổ Nhánh liên kết nguồn, thì để ROT như là tập hợp của những trường ClassificationNode (nút phân loại) mà làm thoả mãn Truy vấn Nut phân loại và là đối tượng đích của một số phần tử của AF. Nếu ROT trống rỗng, thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu một truy vấn của tổ chức được quy định trong khuôn khổ Nhánh liên kết nguồn, thì để ROT như là tập hợp của trường Tổ chức mà làm thoả mãn Truy vấn của tổ chức và là đối tượng đích của một số phần tử AR. Nếu ROT trống rỗng, thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu một AuditableEventQuery (truy vấn sự kiện có thể kiểm tra) được quy định trong Nhánh liên kết nguồn, thì để ROT như là tập hợp của những trường Sự kiện có thể kiểm tra được mà làm thoả mãn AuditableEventQuery (truy vấn sự kiện có thể kiểm tra) và là đối tượng đích của một số phần tử AF. Nếu ROT trống rỗng, thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu một RegistryPackageQuery (truy vấn gói đăng ký) được quy định trong Nhánh liên kết nguồn, thì để ROT như là tập hợp của những trường Gói đăng ký mà làm thoả mãn RegistryPackageQuery (truy vấn gói đăng ký) và là đối tượng đích của một số phần tử AR. Nếu ROT trống rỗng, thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu một ExtrinsicObjectQuery (truy vấn đối tượng ngoại lai) được quy định trong Nhánh liên kết nguồn, thì để ROT như là tập hợp của những trường ExtrinsicObjects mà làm thoả mãn ExtrinsicObjectQuery (truy vấn đối tượng ngoại lai) và là đối tượng đích của một số phần tử AF. Nếu ROT trống rỗng, thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu một OrganizationQuery (truy vấn dịch vụ) được quy định trong Nhánh liên kết nguồn, thì để ROT như là tập hợp của những trường Dịch vụ mà làm thoả mãn OrganizationQuery (truy vấn dịch vụ) và là đối tượng đích của một số phần tử AF. Nếu ROT trống rỗng, thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu một Nhánh người sử dụng được quy định trong Nhánh liên kết nguồn, thì để ROT như là tập hợp của những trường Người sử dụng và là đối tượng đích của một số phần tử AF. Nếu ROT trống rỗng, thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề. Để u là thành viên của ROT. Nếu phần tử Bộ lọc người sử dụng được quy định trong Nhánh người sử dụng, và nếu u không làm thoả mãn bộ lọc, thì kiểu u khỏi ROT. Nếu ROT trống rỗng, thì kiểu x khỏi ROT. Nếu ROT trống rỗng thì tiếp tục số nguyên tắc liền kề. Nếu một phần tử Bộ lọc địa chỉ thư (Postal AddressFilter) được quy định trong Nhánh người sử dụng, và nếu địa chỉ thư của u không thoả mãn bộ lọc thì kiểu u khỏi ROT. Nếu ROT trống rỗng, thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề. Nếu Bộ lọc số điện thoại (TelephoneNumberFilter(s)) được quy định trong Nhánh người sử dụng và nếu bất cứ số Bộ lọc số điện thoại nào không được thoả mãn bởi ít nhất một trong các số điện thoại của u, thì kiểu u khỏi ROT. Nếu ROT trống rỗng, thì kiểu x khỏi RO. Nếu RO trống rỗng, thì tiếp tục số nguyên tắc liền kề. Nếu một phần tử Truy vấn của tổ chức được quy định trong Nhánh người sử dụng, thì để o như là trường của Tổ chức mà được xác định bởi tổ chức và được u liên kết cùng. Nếu o không o làm thoả mãn trường Tổ chức như được xác định trong phần 8.2.11, thì kiểu u khỏi

ROT. Nếu ROT trống rỗng, thì kiểu x khởi RO. Nếu RO trống rỗng, thì tiếp tục số nguyên tắc liền kề.

Nếu một ClassificationQuery (truy vấn phân loại) được quy định trong Nhánh liên kết nguồn, thì để ROT như là tập hợp của những trường Phân loại mà làm thoả mãn ClassificationQuery (truy vấn phân loại) và là đối tượng đích của một số phần tử AF. Nếu ROT trống rỗng, thì kiểu x khởi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề (Nguyên tắc 2).

Nếu một Nhánh Quy định dịch vụ được quy định trong Nhánh liên kết nguồn, thì để ROT như là tập hợp của những trường Quy định dịch vụ mà là đối tượng đích của một số phần tử AF. Nếu ROT trống rỗng, thì kiểu x khởi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề. Để sb như là thành viên của ROT. Nếu phần tử Bộ lọc Quy định dịch vụ được quy định trong Nhánh Quy định dịch vụ và nếu sb không thoả mãn bộ lọc này, thì kiểu sb khởi ROT. Nếu ROT trống rỗng thì kiểu x khởi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề. Nếu một nhánh liên kết quy định kỹ thuật được quy định trong Nhánh dịch vụ Quy định thì xem xét mỗi phần tử của Nhánh liên kết quy định kỹ thuật một cách riêng rẽ như sau:

Để sb như là Quy định dịch vụ được duy trì trong ROT. Để SL như là tập hợp của tất cả các trường liên kết quy định sl mà mô tả những liên kết quy định của sb. Nếu một phần tử Bộ lọc liên kết quy định được quy định trong Nhánh liên kết quy định kỹ thuật, và nếu sl không thoả mãn bộ lọc, thì kiểu sl khởi SL. Nếu SL trống rỗng thì kiểu sb khởi ROT. Nếu ROT trống rỗng thì kiểu x khởi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề. Nếu một phần tử Truy vấn đối tượng đăng ký được quy định trong Nhánh liên kết quy định kỹ thuật, thì để sl là một liên kết quy định đang được duy trì trong SL. Xử lý phần tử Truy vấn đối tượng liên kết như sau:

Để RO là tập hợp kết quả của Truy vấn đối tượng đăng ký như được làm rõ tại phần 8.2.2. Nếu sl không phải là liên kết quy định cho ít nhất một đối tượng đăng ký trong RO, thì kiểu sl từ SL. Nếu SL trống rỗng thì kiểu sb khởi ROT. Nếu ROT trống rỗng thì kiểu x khởi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề. Nếu một phần tử RegistryEntryQuery (truy vấn mục nhập đăng ký) được quy định trong Nhánh liên kết quy định kỹ thuật, thì để sl như là một liên kết quy định được duy trì trong SL. Xử lý phần tử RegistryEntryQuery (truy vấn mục nhập đăng ký) như sau: Để RE là tập hợp kết quả của RegistryEntryQuery (truy vấn mục nhập đăng ký) như được làm rõ tại phần 8.2.3. Nếu sl không phải là liên kết quy định cho ít nhất một RegistryEntryQuery (mục nhập đăng ký) trong RE, thì kiểu sl khởi SL. Nếu SL trống rỗng thì kiểu sb khởi ROT. Nếu ROT trống rỗng thì kiểu x khởi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề. Nếu Nhánh Quy định dịch vụ hợp đích (ServiceBindingTargetBranch) được quy định trong Nhánh Quy định dịch vụ, thì để SBT như là tập hợp những trường Quy định dịch vụ mà làm thoả mãn Nhánh Quy định dịch vụ hợp đích và là Quy định dịch vụ đích của một số phần tử của ROT. Nếu SBT trống rỗng thì kiểu sb khởi ROT. Nếu ROT trống rỗng, thì kiểu x khởi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu một Nhánh liên kết quy định kỹ thuật được quy định trong Nhánh liên kết nguồn, thì để ROT như là tập hợp những trường Association quy định mà là đối tượng đích của một số phần tử AF. Nếu ROT trống rỗng, thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề. Để si như là thành viên của ROT. Nếu một phần tử của Bộ lọc liên kết quy định được quy định trong Nhánh liên kết quy định kỹ thuật, và nếu si không làm thoả mãn bộ lọc đó, thì kiểu si khỏi ROT. Nếu ROT trống rỗng thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề. Nếu một phần tử Truy vấn mục tiêu đăng ký được quy định trong Nhánh liên kết quy định kỹ thuật, thì để si như là một liên kết quy định được duy trì trong ROT.

Xử lý phần tử Truy vấn đối tượng đăng ký như sau: Để RO như là tập hợp kết quả của Truy vấn đối tượng đăng ký như được xác định trong phần 8.2.2. Nếu si không phải là liên kết quy định cho một số đối tượng đăng ký trong RO, thì kiểu si khỏi ROT.

Nếu ROT trống rỗng thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề. Nếu một yếu tố RegistryEntryQuery (truy vấn mục nhập đăng ký) được quy định trong Nhánh liên kết quy định, thì để si như là một liên kết quy định được duy trì trong ROT. Xử lý phần tử RegistryEntryQuery (truy vấn mục nhập đăng ký) như sau: Để RE như là tập hợp kết quả của RegistryEntryQuery (truy vấn mục nhập đăng ký) như được xác định trong phần 8.2.3. Nếu si không phải là liên kết quy định cho đối tượng đăng ký trong RO, thì kiểu si khỏi ROT. Nếu ROT trống rỗng thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề. Nếu một phần tử RegistryEntryQuery (truy vấn mục nhập đăng ký) được quy định trong Nhánh liên kết quy định kỹ thuật thì để si như là một liên kết quy định được duy trì trong ROT. Xử lý phần tử RegistryEntryQuery (truy vấn mục nhập đăng ký) như sau: Để RE như là tập hợp kết quả của RegistryEntryQuery (truy vấn mục nhập đăng ký) như được xác định trong phần 8.2.3. Nếu si không phải là liên kết quy định cho ít nhất một RegistryEntryQuery (mục nhập đăng ký) trong RE, thì kiểu si từ ROT. Nếu ROT là trống rỗng thì kiểu x khỏi RO. Nếu RO là trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu một Truy vấn Association (liên kết) được quy định trong Nhánh liên kết nguồn, thì để ROT như là tập hợp của những trường Association mà làm thoả mãn Truy vấn Association (liên kết) và là đối tượng đích của một số phần tử AF. Nếu ROT trống rỗng, thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề (Nguyên tắc 2).

- j) nếu một phần tử Nhánh liên kết hợp đích không được quy định thì chuyển tới bước tiếp sau; mặt khác, để x như là một đối tượng đăng ký được duy trì trong RO. Nếu x không phải là đối tượng đích của một số trường Association, thì kiểu x khỏi RO. Nếu RO trống rỗng thì chuyển tới số nguyên tắc liền kề; mặt khác, xử lý mỗi phần tử Nhánh đăng ký hợp đích một cách riêng rẽ như sau:

Nếu không có Bộ lọc liên kết được quy định trong Nhánh liên kết hợp đích, thì để AR như là một tập hợp của tất cả các trường Association trong Nhánh liên kết hợp đích, thì để AF như là tập hợp của những trường Association mà làm thoả mãn Bộ lọc Association và xem x như là

đối tượng đích. Nếu AF trống rỗng, thì kiểu x khởi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu một Bộ lọc liên kết ngoài (ExternalLinkFilter) được quy định trong Nhánh liên kết hợp đích, thì xem ROS như là tập hợp của những trường Association ngoài mà làm thoả mãn Bộ lọc liên kết ngoài và là đối tượng nguồn của một số phần tử của AF. Nếu ROS trống rỗng, thì kiểu x khởi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu một Bộ lọc định danh ngoài được quy định trong Nhánh liên kết hợp đích, thì để ROS như là tập hợp những trường định danh ngoài mà làm thoả mãn.

Bộ lọc định danh ngoài và là đối tượng nguồn của một số phần tử AF. Nếu ROS trống rỗng, thì kiểu x khởi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu một Truy vấn đối tượng nghiên cứu được quy định trong Nhánh liên kết hợp đích, thì để ROS như là tập hợp những trường RegistryObject (đối tượng đăng ký) mà làm thoả mãn Truy vấn đối tượng định danh và là đối tượng nguồn của một số phần tử AF. Nếu ROS trống rỗng, thì kiểu x khởi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu một RegistryEntryQuery (truy vấn mục nhập đăng ký) được quy định trong Nhánh liên kết hợp đích, thì để ROS như là tập hợp những trường RegistryEntryQuery (mục nhập đăng ký) mà làm thoả mãn Truy vấn đường vào nghiên cứu và là đối tượng nguồn của một số phần tử AF. Nếu ROS trống rỗng, thì kiểu x khởi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu một Truy vấn kế hoạch phân loại được quy định trong Nhánh liên kết hợp đích, thì để ROS như là tập hợp những trường kế hoạch phân loại mà làm thoả mãn Truy vấn kế hoạch phân loại và là đối tượng nguồn của một số phần tử AF. Nếu ROS trống rỗng, thì kiểu x khởi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu một Truy vấn Nut phân loại được quy định trong Nhánh liên kết hợp đích, thì để ROS như là tập hợp những trường Nut phân loại mà làm thoả mãn Truy vấn Nut phân loại và là đối tượng nguồn của một số phần tử AF. Nếu ROS trống rỗng, thì kiểu x khởi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu một Truy vấn của tổ chức được quy định trong Nhánh liên kết hợp đích, thì để ROS như là tập hợp những trường Tổ chức mà làm thoả mãn Truy vấn của tổ chức và là đối tượng nguồn của một số phần tử AF. Nếu ROS trống rỗng, thì kiểu x khởi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu một AuditableEventQuery (truy vấn sự kiện có thể kiểm tra) được quy định trong Nhánh liên kết hợp đích, thì để ROS như là tập hợp những trường Sự kiện có thể kiểm tra được mà làm thoả mãn AuditableEventQuery (truy vấn sự kiện có thể kiểm tra) được và là đối tượng nguồn của một số phần tử AF. Nếu ROS trống rỗng, thì kiểu x khởi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu một RegistryPackageQuery (truy vấn gói đăng ký) được quy định trong Nhánh liên kết hợp đích, thì để ROS như là tập hợp những trường gói đăng ký mà làm thoả mãn RegistryPackageQuery (truy vấn gói đăng ký) và là đối tượng nguồn của một số phần tử AF. Nếu ROS trống rỗng, thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu một ExtrinsicObjectQuery (truy vấn đối tượng ngoại lai) (ExtrinsicObjectQuery (truy vấn đối tượng ngoại lai)) được quy định trong Nhánh liên kết hợp đích, thì để ROS như là tập hợp những trường ExtrinsicObjects mà làm thoả mãn ExtrinsicObjectQuery (truy vấn đối tượng ngoại lai) và là đối tượng nguồn của một số phần tử AF. Nếu ROS trống rỗng, thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu một OrganizationQuery (truy vấn dịch vụ) được quy định trong Nhánh liên kết hợp đích, thì để ROS như là tập hợp những trường dịch vụ mà làm thoả mãn OrganizationQuery (truy vấn dịch vụ) và là đối tượng nguồn của một số phần tử AF. Nếu ROS trống rỗng, thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu một Nhánh người sử dụng được quy định trong Nhánh liên kết hợp đích, thì để ROS như là tập hợp những trường người sử dụng mà làm thoả mãn Nhánh người sử dụng và là đối tượng nguồn của một số phần tử AF. Nếu ROS trống rỗng, thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề. Để u như là thành viên của ROS. Nếu một phần tử Bộ lọc người sử dụng được quy định trong Nhánh người sử dụng, và nếu u không làm thoả mãn bộ lọc, thì kiểu u khỏi ROS. Nếu ROS trống rỗng thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề. Nếu phần tử Bộ lọc địa chỉ thư được quy định trong Nhánh người sử dụng, và nếu địa chỉ thư của u không làm thoả mãn bộ lọc, thì kiểu u khỏi ROS. Nếu ROS trống rỗng thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề. Nếu Bộ lọc số điện thoại được quy định trong Nhánh người sử dụng và nếu bất cứ Bộ lọc số điện thoại nào không được làm thoả mãn bởi một số số điện thoại của u, thì kiểu u khỏi ROS. Nếu ROS trống rỗng thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề. Nếu một phần tử Truy vấn của tổ chức được quy định trong Nhánh người sử dụng, thì để o như là trường của Tổ chức mà được định danh bởi tổ chức và u sẽ là được liên kết. Nếu o không làm thoả mãn Truy vấn của tổ chức như được xác định tại phần 8.2.11 thì kiểu u khỏi ROS. Nếu ROS trống rỗng thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu một ClassificationQuery (truy vấn phân loại) được quy định trong Nhánh liên kết hợp đích, thì để ROS như là tập hợp những trường Phân loại mà làm thoả mãn ClassificationQuery (truy vấn phân loại) và là đối tượng nguồn của một số phần tử AF. Nếu ROS trống rỗng, thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề (Nguyên tắc 2).

Nếu một Nhánh Quy định dịch vụ được quy định trong Nhánh liên kết hợp đích, thì để ROS như là tập hợp những trường Quy định dịch vụ mà làm thoả mãn Nhánh Quy định dịch vụ và là đối tượng nguồn của một số phần tử AF. Nếu ROS trống rỗng, thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp

tục số nguyên tắc liền kề. Để sb như là thành viên của ROS. Nếu một phần tử của Bộ lọc Quy định dịch vụ được quy định trong Nhánh Quy định dịch vụ, và nếu sb không làm thỏa mãn bộ lọc này, thì kiểu sb khỏi ROS. Nếu ROS trống rỗng thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề. Nếu một Nhánh liên kết quy định kỹ thuật được quy định trong Nhánh Quy định dịch vụ thì xem xét mỗi phần tử Nhánh liên kết quy định kỹ thuật như sau:

Để sb như là Quy định dịch vụ được duy trì trong ROS. Để SL như là tập hợp của tất cả những trường liên kết quyết định si mà mô tả những liên kết quyết định của sb. Nếu một phần tử Bộ lọc liên kết quyết định được quy định trong Nhánh liên kết quyết định, và nếu si không làm thỏa mãn bộ lọc này thì kiểu si khỏi SL. Nếu SL trống rỗng thì kiểu sb khỏi ROS. Nếu ROS trống rỗng thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số quy định liền kề. Nếu một phần tử Truy vấn đối tượng đăng ký được quy định trong Nhánh liên kết quy định kỹ thuật thì để si như là liên kết quy định đang được duy trì trong SL. Xử lý phần tử Truy vấn đối tượng đăng ký như sau: Để RO như là tập hợp kết quả của Truy vấn đối tượng đăng ký như được xác định tại phần 8.2.2. Nếu si không phải là liên kết quyết định kỹ thuật đối với một số đối tượng đăng ký trong RO, thì kiểu si khỏi SL.

Nếu SL trống rỗng thì kiểu sb khỏi ROS. Nếu ROS trống rỗng thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề. Nếu một phần tử RegistryEntryQuery (truy vấn mục nhập đăng ký) được quy định trong Nhánh liên kết quy định thì để si như là một liên kết quy định kỹ thuật được duy trì trong SL. Xử lý phần tử RegistryEntryQuery (truy vấn mục nhập đăng ký) như sau: Để RE như là tập hợp kết quả của RegistryEntryQuery (truy vấn mục nhập đăng ký) như được xác định trong phần 8.2.3. Nếu si không phải là một liên kết quyết định kỹ thuật đối với một số RegistryEntryQuery (mục nhập đăng ký) trong RE, thì kiểu si khỏi SL. Nếu SL trống rỗng thì kiểu sb khỏi ROS. Nếu ROS trống rỗng thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu một Nhánh liên kết quy định kỹ thuật được quy định trong Nhánh liên kết hợp đích, thì để ROS như là tập hợp của những trường Association quy định kỹ thuật mà là đối tượng nguồn của một số phần tử AF. Nếu ROS trống rỗng, thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề. Để si như là thành viên của ROS. Nếu một phần tử Bộ lọc liên kết quyết định kỹ thuật được quy định trong Nhánh liên kết quyết định kỹ thuật, và nếu si không thỏa mãn bộ lọc này, thì kiểu si khỏi ROS. Nếu ROS trống rỗng thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề. Nếu một phần tử Truy vấn đối tượng đăng ký được quy định trong Nhánh liên kết quy định kỹ thuật thì để si như là liên kết quy định kỹ thuật được duy trì trong ROS. Xử lý phần tử Truy vấn đối tượng đăng ký như sau: Để RO như là tập hợp kết quả của Truy vấn đối tượng đăng ký như được xác định trong phần 8.2.2. Nếu si không phải là liên kết quy định liền kề đối với một số đối tượng đăng ký trong RO, thì kiểu si khỏi ROS. Nếu ROS trống rỗng thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề. Nếu một phần tử RegistryEntryQuery (truy vấn mục nhập đăng ký) được quy định trong Nhánh liên kết quy định kỹ thuật, thì để si như là một liên kết quy định kỹ thuật trong ROS. Xử lý những phần tử RegistryEntryQuery (truy vấn mục nhập đăng

ký) như sau: Để RE như là tập hợp kết quả của RegistryEntryQuery (truy vấn mục nhập đăng ký) như được xác định trong phần 8.2.3. Nếu si không phải là một liên kết quy định kỹ thuật đối với một số RegistryEntryQuery (mục nhập đăng ký) trong RE, thì kiểu si khỏi ROS. Nếu ROS trống rỗng thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề. Nếu một Nhánh Quy định dịch vụ hợp đích được quy định trong Nhánh Quy định dịch vụ, thì để SBT như là tập hợp của những trường Quy định dịch vụ mà làm thỏa mãn Nhánh Quy định dịch vụ hợp đích và là Quy định dịch vụ hợp đích của một số phần tử của ROT. Nếu SBT trống rỗng thì kiểu sb khỏi ROT. Nếu ROT trống rỗng thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề.

Nếu một Truy vấn Association (liên kết) được quy định trong Nhách Association hợp đích, thì để ROS như là tập hợp của những trường Association mà làm thỏa mãn Truy vấn Association (liên kết) và là đối tượng nguồn của một số phần tử AF. Nếu ROS trống rỗng, thì kiểu x khỏi RO. Nếu RO trống rỗng thì tiếp tục số nguyên tắc liền kề (Nguyên tắc 2).

2. nếu RO trống rỗng, thì đưa ra cảnh báo: kết quả truy vấn đối tượng đăng ký trống rỗng; mặt khác, cài đặt RO như là kết quả của Truy vấn đối tượng đăng ký;
3. hồi trả kết quả và những cảnh báo lũy tích hoặc gỡ bỏ (trong danh sách lỗi đăng ký) trong Phản hồi đăng ký.

Ví dụ:

Một khách hàng cần tất cả các mục khoản được phân loại bởi hai kế hoạch phân loại khác nhau, một kế hoạch dựa trên cơ sở “Công nghiệp” và còn lại dựa trên “Mặt địa lý”. Cả hai kết hoạch này được xác định một cách riêng biệt bởi ebXML và được đăng ký như là “công nghiệp: urn:ebxml:cs” và “địa lý: urn:ebxml:cs”. Truy vấn tiếp theo xác định RegistryEntryQuery (mục nhập đăng ký) cho tất cả những mục khoản được đăng ký mà được phân loại bởi Công nghiệp như là Nút phụ của “Tự động” và bởi Địa lý như là Nút phụ của “Châu á/Nhật bản”

```
<AdhocQueryRequest>
<ResponseOption returnType = "RegistryEntry"/>
<FilterQuery>
<RegistryObjectQuery>
<ClassifiedByBranch>
<ClassificationFilter>
<Clause>
<SimpleClause leftArgument = "path">
<StringClause stringPredicate = "Equal">//Automotive</StringClause>
</SimpleClause>
</Clause>
</ClassificationFilter>
<ClassificationSchemeQuery (truy vấn giản đồ phân loại)>
<NameBranch>
<LocalizedStringFilter>
<Clause>
<SimpleClause leftArgument = "value">
<StringClause stringPredicate = "Equal">urn:ebxml:cs:industry</StringClause>
</SimpleClause>
</Clause>
</LocalizedStringFilter>
</NameBranch>
</ClassificationSchemeQuery (truy vấn giản đồ phân loại)>
```

```

</ClassifiedByBranch>
<ClassifiedByBranch>
<ClassificationFilter>
<Clause>
<SimpleClause leftArgument = "path">
<StringClause stringPredicate = "StartsWith">/Geography-id/Asia/Japan</StringClause>
</SimpleClause>
</Clause>
</ClassificationFilter>
<ClassificationSchemeQuery (truy vấn giản đồ phân loại)>
<NameBranch>
<LocalizedStringFilter>
<Clause>
<SimpleClause leftArgument = "value">
<StringClause stringPredicate = "Equal">urn:ebxml:cs:geography</StringClause>
</SimpleClause>
</Clause>
</LocalizedStringFilter>
</NameBranch>
</ClassificationSchemeQuery (truy vấn giản đồ phân loại)>
</ClassifiedByBranch>
</RegistryObjectQuery>
</FilterQuery>
</AdhocQueryRequest>

```

Một khách hàng mong muốn xác định tất cả những trường RegistryObject (đối tượng đăng ký) mà được phân loại bởi một số kế hoạch phân loại bên trong và có một số từ chính quan trọng được cho trước như là một phần của mô tả về nút phân loại của kế hoạch phân loại. Truy vấn tiếp theo xác định tất cả những trường RegistryObject (đối tượng đăng ký).

Truy vấn có lợi điểm về kiến thức mà kế hoạch phân loại mang tính bên trong, và vì tất cả nút được mô tả đầy đủ như là những trường ClassificationNode (nút phân loại).

```

<AdhocQueryRequest>
<ResponseOption returnType = "RegistryObject"/>
<FilterQuery>
<RegistryObjectQuery>
<ClassifiedByBranch>
<ClassificationNodeQuery (truy vấn nút phân loại)>
<DescriptionBranch>
<LocalizedStringFilter>
<Clause>
<SimpleClause leftArgument = "value">
<StringClause stringPredicate = "Equal">transistor</StringClause>
</SimpleClause>
</Clause>
</LocalizedStringFilter>
</DescriptionBranch>
</ClassificationNodeQuery (truy vấn nút phân loại)>
</ClassifiedByBranch>
</RegistryObjectQuery>
</FilterQuery>
</AdhocQueryRequest>

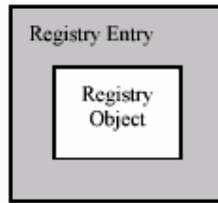
```

8.2.3 RegistryEntryQuery (truy vấn mục nhập đăng ký)

Mục đích:

Nhằm xác định tập hợp những trường RegistryEntryQuery (mục nhập đăng ký) như là kết quả của một truy vấn đối với siêu dữ liệu đăng ký được lựa chọn.

Quy định ebRIM



Hình 17 - Quy định ebRIM đối với RegistryEntryQuery

Định nghĩa

```
<complexType name="RegistryEntryQueryType">
  <complexContent>
    <extension base="tns:RegistryObjectQueryType">
      <sequence>
        <element ref="tns:RegistryEntryFilter" minOccurs="0" maxOccurs="1" />
      </sequence>
    </extension>
  </complexContent>
</complexType>
<element name="RegistryEntryQuery" type="tns:RegistryEntryQueryType" />
<element name="RegistryEntryQueryResult">
  <complexType>
    <choice minOccurs="0" maxOccurs="unbounded">
      <element ref="rim:ObjectRef" />
      <element ref="rim:ClassificationScheme" />
      <element ref="rim:ExtrinsicObject" />
      <element ref="rim:RegistryEntry" />
      <element ref="rim:RegistryObject" />
      <element ref="rim:RegistryPackage" />
    </choice>
  </complexType>
</element>
```

Nguyên tắc ngữ nghĩa

1. để RE bao hàm tập hợp của tất cả những trường RegistryEntryQuery (mục nhập đăng ký) bất biến trong sổ đăng ký. Những bước dưới đây sẽ xoá những trường trong RE mà không làm thoả mãn điều kiện của bộ lọc được xác định.
 - a) nếu RE trống rỗng thì tiếp tục số nguyên tắc liền kề;
 - b) nếu một Bộ lọc RegistryEntryQuery (mục nhập đăng ký) không được xác định thì chuyển tới bước liền kề, để x như là một RegistryEntryQuery (mục nhập đăng ký) trong RE. Nếu x không thoả mãn Bộ lọc RegistryEntryQuery (mục nhập đăng ký), thì kiểu x khỏi RE. Nếu RE trống rỗng thì tiếp tục số nguyên tắc liền kề;
 - c) để RE như là tập hợp của những trường RegistryEntryQuery (mục nhập đăng ký) được duy trì. lượng giá Truy vấn đối tượng đăng ký được thừa hưởng đối với RE như được giảng giải trong phần 8.2.2.
2. nếu RE trống rỗng, thì đưa ra cảnh báo: truy vấn đường đăng ký trống rỗng; mặt khác, cài đặt RE là kết quả của RegistryEntryQuery (truy vấn mục nhập đăng ký);

3. hỏi trả kết quả và những cảnh báo lũy tích hoặc gỡ bỏ (trong Danh sách lỗi đăng ký) trong Phản hồi đăng ký.

Ví dụ

Một khách hàng muốn thiết lập mối quan hệ thương mại với Tập đoàn XYZ và muốn biết liệu họ đã đăng ký bất cứ tài liệu thương mại nào trong Mục Đăng ký. Truy vấn tiếp theo hỏi trả kiểu một tập hợp RegistryEntryQuery (mục nhập đăng ký) định danh về bất cứ điều khoản được đăng ký hiện hành mà đã được đề trình bởi những tổ chức, những tổ chức có tên bao hàm "XYZ". Nó không hỏi trả bất cứ RegistryEntryQuery (mục nhập đăng ký) định danh đối với những điều khoản bị thay thế, không tán thành, rút khỏi và không dùng.

```
<AdhocQueryRequest>
<ResponseOption returnType = "ObjectRef"/>
<FilterQuery>
<RegistryEntryQuery>
<TargetAssociationBranch>
<AssociationFilter>
<Clause>
<SimpleClause leftArgument = "associationType">
<StringClause stringPredicate = "Equal">SubmitterOf</StringClause>
</SimpleClause>
</Clause>
</AssociationFilter>
<OrganizationQuery (truy vấn tổ chức)>
<NameBranch>
<LocalizedStringFilter>
<Clause>
<SimpleClause leftArgument = "value">
<StringClause stringPredicate = "Contains">XYZ</StringClause>
</SimpleClause>
</Clause>
</LocalizedStringFilter>
</NameBranch>
</OrganizationQuery (truy vấn tổ chức)>
</TargetAssociationBranch>
<RegistryEntryFilter>
<Clause>
<SimpleClause leftArgument = "status">
<StringClause stringPredicate = "Equal">Approved</StringClause>
</SimpleClause>
</Clause>
</RegistryEntryFilter>
</RegistryEntryQuery>
</FilterQuery>
</AdhocQueryRequest>
```

Một khách hàng sử dụng Sản phẩm tiêu chuẩn Liên hợp quốc và kế hoạch Phân loại các dịch vụ (UNSPSC) và muốn xác định tất cả các công ty mà buôn bán những sản phẩm được xác định như là "Những phần quay vòng hợp nhất (Integrated circuit components)", nghĩa là mã UNSPSC code "321118". Khách hàng biết rằng các công ty đã đăng ký tài liệu Collaboration Protocol Profile (CPP) trong mục Đăng ký, và rằng mỗi profile như vậy đã được xác định bởi UNSPSC trên cơ sở những sản phẩm mà công ty buôn bán. Tuy nhiên, khách hàng không biết liệu kế hoạch phân loại UNSPSC là bên trong hay bên ngoài đối với đăng ký này. Truy vấn tiếp theo hỏi trả một tập hợp những trường

RegistryEntryQuery (mục nhập đăng ký) được thông qua đối với những công ty trong CPP mà buôn bán những phần quay vòng hợp nhất.

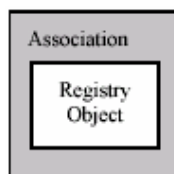
```
<AdhocQueryRequest>
<ResponseOption returnType = "RegistryEntry"/>
<FilterQuery>
<RegistryEntryQuery>
<ClassifiedByBranch>
<ClassificationFilter>
<Clause>
<SimpleClause leftArgument = "code">
<StringClause stringPredicate = "Equal">321118</StringClause>
</SimpleClause>
</Clause>
</ClassificationFilter>
<ClassificationSchemeQuery (truy vấn giản đồ phân loại)>
<NameBranch>
<LocalizedStringFilter>
<Clause>
<SimpleClause leftArgument = "value">
<StringClause stringPredicate = "Equal">urn:org:un:spsc:cs2001</StringClause>
</SimpleClause>
</Clause>
</LocalizedStringFilter>
</NameBranch>
</ClassificationSchemeQuery (truy vấn giản đồ phân loại)>
</ClassifiedByBranch>
<RegistryEntryFilter>
<Clause>
<CompoundClause connectivePredicate = "And">
<Clause>
<SimpleClause leftArgument = "objectType">
<StringClause stringPredicate = "Equal">CPP</StringClause>
</SimpleClause>
</Clause>
<Clause>
<SimpleClause leftArgument = "status">
<StringClause stringPredicate = "Equal">Approved</StringClause>
</SimpleClause>
</Clause>
</CompoundClause>
</Clause>
</RegistryEntryFilter>
</RegistryEntryQuery>
</FilterQuery>
</AdhocQueryRequest>
```

8.2.4 Truy vấn Association (liên kết)

Mục đích:

Nhằm xác định một tập hợp trường liên kết như là kết quả của một truy vấn đối với siêu dữ liệu đăng ký được lựa chọn.

Quy định ebRIM



Hình 18 - Quy định ebRIM đối với AssociationQuery**Định nghĩa**

```

<complexType name = "AssociationQueryType">
<complexContent>
<extension base = "tns:RegistryObjectQueryType">
<sequence>
<element ref = "tns:AssociationFilter" minOccurs = "0" maxOccurs = "1"/>
</sequence>
</extension>
</complexContent>
</complexType>
<element name = "AssociationQuery" type = "tns:AssociationQueryType"/>
<element name="AssociationQueryResult">
<complexType>
<choice minOccurs="0" maxOccurs="unbounded">
<element ref="rim:ObjectRef" />
<element ref="rim:RegistryObject" />
<element ref="rim:Association" />
</choice>
</complexType>
</element>

```

Nguyên tắc ngữ nghĩa

1. để A bao hàm tập hợp của tất cả những trường Association bất biến trong sổ đăng ký. Những bước dưới đây sẽ xoá những trường trong A mà không làm thoả mãn điều kiện của bộ lọc được xác định.
 - a) nếu A trống rỗng thì tiếp tục số nguyên tắc liền kề;
 - b) nếu một phần tử Bộ lọc liên kết không được xác định thì chuyển tới bước liền kề, để x như là một RegistryEntryQuery (mục nhập đăng ký) trong RE. Nếu x không thoả mãn Bộ lọc liên kết, thì kiểu x khỏi A. Nếu A trống rỗng thì tiếp tục số nguyên tắc liền kề;
 - c) để A như là tập hợp của những trường Association được duy trì. Lượng giá Truy vấn đối tượng đăng ký được thừa hưởng đối với RE như được giảng giải trong phần 8.2.2.
2. nếu A trống rỗng, thì đưa ra cảnh báo: Truy vấn Association (liên kết) trống rỗng; mặt khác, cài đặt A là kết quả của Truy vấn Association (liên kết);
3. hồi trả kết quả và những cảnh báo lũy tích hoặc gỡ bỏ (trong Danh sách lỗi đăng ký) trong Phản hồi đăng ký.

Vi dụ

Một khách hàng mong muốn xác định một tập hợp liên kết mà 'tương đương với' một tập các liên kết khác

```

<AdhocQueryRequest">
<ResponseOption returnType="LeafClass" />
<FilterQuery>
<AssociationQuery>
<SourceAssociationBranch>
<AssociationFilter>
<Clause>
<SimpleClause leftArgument="associationType">
<StringClause stringPredicate="Equal">EquivalentTo</StringClause>

```

```

</SimpleClause>
</Clause>
</AssociationFilter>
<AssociationQuery>
<AssociationFilter>
<Clause>
<SimpleClause leftArgument="associationType">
<StringClause stringPredicate="StartsWith">Sin</StringClause>
</SimpleClause>
</Clause>
</AssociationFilter>
</AssociationQuery>
</SourceAssociationBranch>
<AssociationFilter>
<Clause>
<SimpleClause leftArgument="associationType">
<StringClause stringPredicate="StartsWith">Son</StringClause>
</SimpleClause>
</Clause>
</AssociationFilter>
</AssociationQuery>
</FilterQuery>
</AdhocQueryRequest>

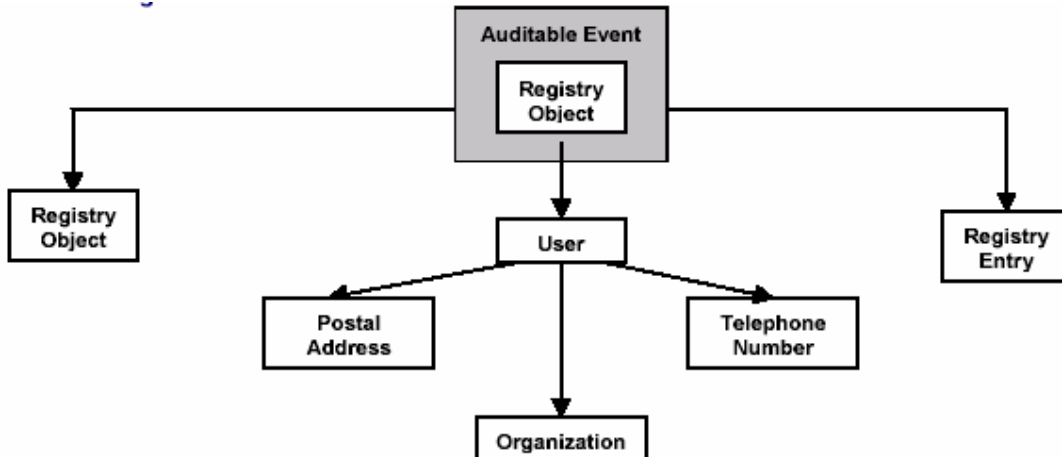
```

8.2.5 AuditableEventQuery (truy vấn sự kiện có thể kiểm tra)

Mục đích:

Nhằm xác định một tập hợp những trường sự kiện có thể được kiểm tra như là kết quả của truy vấn đối với siêu dữ liệu được đăng ký.

Quy định ebRIM



Hình 19 - Quy định ebRIM đối với AuditableEventQuery

Định nghĩa

```

<complexType name="AuditableEventQueryType">
<complexContent>
<extension base="tns:RegistryObjectQueryType">
<sequence>
<element ref="tns:AuditableEventFilter" minOccurs="0" />
<element ref="tns:RegistryObjectQuery" minOccurs="0" maxOccurs="1" />
<element ref="tns:RegistryEntryQuery" minOccurs="0" maxOccurs="1" />
<element ref="tns:UserBranch" minOccurs="0" maxOccurs="1" />
</sequence>
</extension>
</complexContent>

```

```

</complexType>
<element name="AuditableEventQuery" type="tns:AuditableEventQueryType" />
<element name="AuditableEventQueryResult">
<complexType>
<choice minOccurs="0" maxOccurs="unbounded">
<element ref="rim:ObjectRef" />
<element ref="rim:RegistryObject" />
<element ref="rim:AuditableEvent" />
</choice>
</complexType>
</element>

```

Nguyên tắc ngữ nghĩa

1. để AE bao hàm tập hợp của tất cả những trường Sự kiện có thể kiểm tra trong mục Đăng ký. Những bước tiếp theo sẽ xoá những trường trong AE mà không làm thoả mãn điều kiện của những bộ lọc xác định.
 - a) nếu AE trống rỗng thì tiếp tục số nguyên tắc liền kề;
 - b) nếu Bộ lọc sự kiện có thể kiểm tra không được quy định thì chuyển tới bước tiếp theo; mặt khác, để x như là một sự kiện có thể được kiểm tra trong AE. Nếu x không làm thoả mãn Bộ lọc sự kiện có thể kiểm tra, thì kiểu x khỏi AE. Nếu AE trống rỗng thì tiếp tục tới số nguyên tắc liền kề;
 - c) nếu một phần tử Truy vấn đối tượng đăng ký không được quy định thì chuyển tới bước tiếp theo; mặt khác, để x như là một sự kiện có thể kiểm tra được duy trì trong AE. Xử lý phần tử Truy vấn đối tượng đăng ký như sau: Để RO như là tập hợp kết quả của Truy vấn đối tượng đăng ký như được xác định trong phần 8.2.2. Nếu x không phải là sự kiện có thể kiểm tra đối với một số đối tượng đăng ký trong RO, thì kiểu x khỏi AE. Nếu AE trống rỗng thì tiếp tục số nguyên tắc liền kề;
 - d) nếu một phần tử RegistryEntryQuery (truy vấn mục nhập đăng ký) không được quy định thì chuyển tới bước tiếp theo; mặt khác, để x như là một sự kiện có thể kiểm tra được duy trì trong AE. Xử lý phần tử RegistryEntryQuery (truy vấn mục nhập đăng ký) như sau: Để RE như là tập hợp kết quả của RegistryEntryQuery (truy vấn mục nhập đăng ký) như được xác định trong phần 8.2.3. Nếu x không phải là sự kiện có thể kiểm tra đối với một số đối tượng đăng ký trong RE, thì kiểu x khỏi AE. Nếu AE trống rỗng thì tiếp tục số nguyên tắc liền kề;
 - e) nếu một phần tử Nhánh người sử dụng không được quy định thì chuyển tới bước tiếp theo; mặt khác, để x như là một sự kiện có thể kiểm tra được duy trì trong AE. Để u như là trường người sử dụng mà liên quan tới x. Nếu một phần tử Bộ lọc người sử dụng được quy định trong Nhánh người sử dụng, và nếu u không làm thoả mãn bộ lọc này, thì kiểu x khỏi AE. Nếu một phần tử Bộ lọc địa chỉ thư (PostalAddressFilter) được quy định trong Nhánh người sử dụng, và nếu địa chỉ thư của u không làm thoả mãn bộ lọc này, thì kiểu x khỏi AE. Nếu Bộ lọc số điện thoại được quy định trong Nhánh người sử dụng và nếu bất cứ Bộ lọc số điện thoại nào không làm thoả mãn bởi một số số điện thoại của u thì kiểu x khỏi AE. Nếu Bộ lọc địa chỉ Email không được quy

định trong Nhánh người sử dụng và nếu bất cứ Bộ lọc địa chỉ Email nào không làm thoả mãn số số địa chỉ chỉ Email thì kiểu x khỏi AE.

Nếu một phần tử OrganizationQuery (truy vấn tổ chức) được quy định trong Nhánh người sử dụng, thì để o như là trường của Tổ chức mà được Tổ chức xác định rằng u được thành một nhánh liên kết. Nếu o không làm thoả mãn OrganizationQuery (truy vấn tổ chức) như được xác định trong phần 8.2.11, thì kiểu x khỏi AE. Nếu AE trống rỗng thì tiếp tục số nguyên tắc liền kề;

- f) Để AE như là tập hợp của những trường Sự kiện có thể kiểm tra. Lượng giá Truy vấn đối tượng đăng ký được thừa hưởng đối với AE như được trình bày trong phần 8.2.2.
2. nếu AE trống rỗng, thì đưa ra cảnh báo: AuditableEventQuery (truy vấn sự kiện có thể kiểm tra) trống rỗng; mặt khác tập hợp AE là kết quả của Truy vấn sự kiện có thể được kiểm tra;
 3. hồi trả kết quả và những cảnh báo lũy tích hoặc gỡ bỏ (trong Danh sách lỗi đăng ký) trong Phản hồi đăng ký.

Ví dụ

Một khách hàng đăng ký (Registry Client) đã đăng ký một mục khoảng và nó đã được đặt một cái tên "urn:path:myitem". Khách hàng hiện tài hứng thú với tất cả các sự kiện vì ngay từ đầu năm khách hàng đã có sự tác động tới mục khoản này. Truy vấn tiếp theo sẽ hồi trả một tập hợp những trường Sự kiện có thể kiểm tra cho tất cả những sự kiện tương tự.

```
<AdhocQueryRequest>
<ResponseOption returnType = "LeafClass"/>
<FilterQuery>
<AuditableEventQuery>
<AuditableEventFilter>
<Clause>
<SimpleClause leftArgument = "timestamp">
<RationalClause logicalPredicate = "GE">
DateTimeClause>2000-01-01T00:00:00-05:00</DateTimeClause>
</RationalClause>
</SimpleClause>
</Clause>
</AuditableEventFilter>
<RegistryEntryQuery>
<NameBranch>
<LocalizedStringFilter>
<Clause>
<SimpleClause leftArgument = "value">
<StringClause stringPredicate = "Equal">urn:path:myitem</StringClause>
</SimpleClause>
</Clause>
</LocalizedStringFilter>
</NameBranch>
</RegistryEntryQuery>
</AuditableEventQuery>
</FilterQuery>
</AdhocQueryRequest
```

Công ty của khách hàng có nhiều đối tượng được đăng ký trong mục Đăng ký. Mục Đăng ký cho phép các sự kiện được đề nghị từ những tổ chức khác nhau để có một tác động đối với những mục khoản được đăng ký tại công ty bạn, chẳng hạn, một sự phân loại mới và một liên kết mới. Truy vấn tiếp sau

sẽ hồi trả một tập hợp những định danh cho tất cả những sự kiện có thể kiểm tra, được yêu cầu bởi một số tổ chức khác mà có tác động lên mục khoản được đề nghị bởi “myorg”.

```

<AdhocQueryRequest>
<ResponseOption returnType = "LeafClass"/>
<FilterQuery>
<AuditableEventQuery>
<RegistryEntryQuery>
<TargetAssociationBranch>
<AssociationFilter>
<Clause>
<SimpleClause leftArgument = "associationType">
<StringClause stringPredicate = "Equal">SubmitterOf</StringClause>
</SimpleClause>
</Clause>
</AssociationFilter>
<OrganizationQuery (truy vấn tổ chức)>
<NameBranch>
<LocalizedStringFilter>
<Clause>
<SimpleClause leftArgument = "value">
<StringClause stringPredicate = "Equal">myorg</StringClause>
</SimpleClause>
</Clause>
</LocalizedStringFilter>
</NameBranch>
</OrganizationQuery (truy vấn tổ chức)>
</TargetAssociationBranch>
</RegistryEntryQuery>
<UserBranch>
<OrganizationQuery (truy vấn tổ chức)>
<NameBranch>
<LocalizedStringFilter>
<Clause>
<SimpleClause leftArgument = "value">
<StringClause stringPredicate = "-Equal">myorg</StringClause>
</SimpleClause>
</Clause>
</LocalizedStringFilter>
</NameBranch>
</OrganizationQuery (truy vấn tổ chức)>
</UserBranch>
</AuditableEventQuery>
</FilterQuery>
</AdhocQueryRequest>

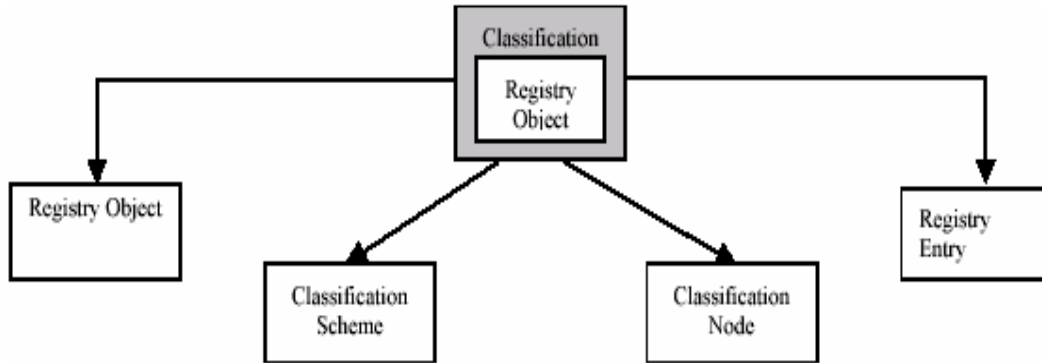
```

8.2.6 ClassificationQuery (truy vấn phân loại)

Mục đích:

Nhằm xác định một tập hợp những trường phân loại như là kết quả của truy vấn đối với siêu dữ liệu đăng ký được lựa chọn.

Quy định ebRIM



Hình 20 - Quy định ebRIM đối với ClassificationQuery

Định nghĩa

```

<complexType name = "ClassificationQueryType">
<complexContent>
<extension base = "tns:RegistryObjectQueryType">
<sequence>
<element ref = "tns:ClassificationFilter" minOccurs = "0" maxOccurs="1"/>
<element ref = "tns:ClassificationSchemeQuery (truy vấn giản đồ phân loại)"
minOccurs = "0" maxOccurs="1"/>
<element ref = "tns:ClassificationNodeQuery (truy vấn nút phân loại)" minOccurs =
"0" maxOccurs="1"/>
<element ref = "tns:RegistryObjectQuery" minOccurs = "0" maxOccurs="1"/>
<element ref = "tns:RegistryEntryQuery" minOccurs = "0" maxOccurs="1"/>
</sequence>
</extension>
</complexContent>
</complexType>
<element name = "ClassificationQuery" type = "tns:ClassificationQueryType"/>
<element name="ClassificationQueryResult">
<complexType>
<choice minOccurs="0" maxOccurs="unbounded">
<element ref="rim:ObjectRef" />
<element ref="rim:RegistryObject" />
<element ref="rim:Classification" />
</choice>
</complexType>
</element>
    
```

Các quy tắc ngữ nghĩa

1. để C biểu thị cho tập hợp của tất cả các trường hợp phân loại liên tục trong sổ đăng ký. Các bước dưới đây sẽ kiểm tra các trường hợp trong C mà không thoả mãn các điều kiện trong các bộ lọc đã được xác định;
 - a) nếu C trống rỗng thì sau đó tiếp tục đến quy tắc kế tiếp;
 - b) nếu một nhân tố Bộ lọc phân loại không được bao hàm trực tiếp trong nhân tố ClassificationQuery (truy vấn phân loại), thì sau đó chuyển đến bước kế tiếp; nếu không thì đặt x là một trường hợp phân loại trong C. Nếu x không thoả mãn Bộ lọc phân loại thì xoá x khỏi C. Nếu C trống rỗng thì tiếp tục chuyển đến quy tắc đã được tính đến;

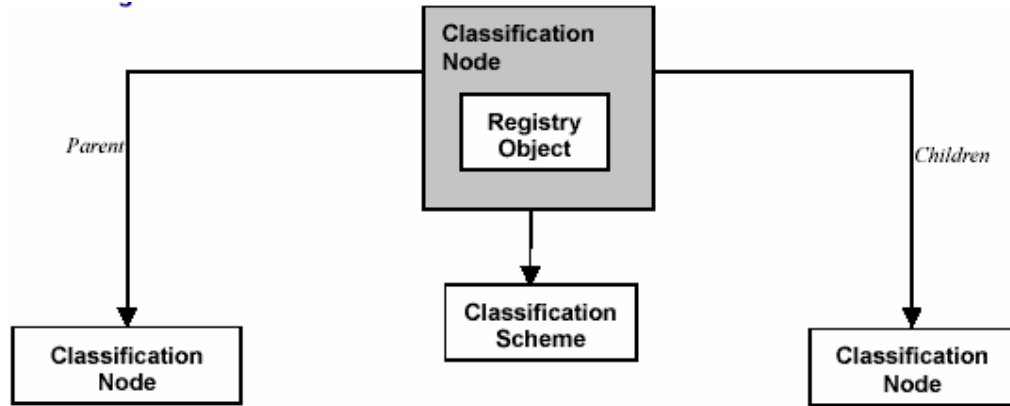
- c) nếu một ClassificationSchemeQuery (truy vấn giản đồ phân loại) là không được định rõ thì chuyển đến bước kế tiếp; nếu không thì, đặt x là một phân loại còn lại trong C. Nếu giản đồ phân loại xác định của x không thoả mãn ClassificationSchemeQuery (truy vấn giản đồ phân loại) như được xác định trong Phần 8.2.8, thì xoá x khỏi C. Nếu C trống rỗng thì tiếp tục đến quy tắc đã được tính đến;
 - d) nếu một ClassificationNodeQuery (truy vấn nút phân loại) không được định rõ thì chuyển đến bước kế tiếp; nếu không thì, đặt x là một phân loại còn lại trong C. Nếu Nút phân kiểu của x không thoả mãn ClassificationNodeQuery (truy vấn nút phân loại) như được xác định trong Phần 8.2.7, thì xoá x khỏi C. Nếu C trống rỗng thì tiếp tục chuyển đến đã được tính đến;
 - e) nếu một nhân tố Truy vấn đối tượng đăng ký không được định rõ thì sau đó chuyển đến bước kế tiếp; nếu không thì, đặt x là một phân loại còn lại trong C. Nhân tố Truy vấn đối tượng đăng ký là như sau: đặt RO là tập hợp kết quả của Truy vấn đối tượng đăng ký như đã được xác định trong Phần 8.2.2. Nếu x không phải là một phân loại của ít nhất một đối tượng đăng ký trong RO, thì xoá x khỏi X. Nếu C trống rỗng thì tiếp tục chuyển đến quy tắc đã được tính đến;
 - f) nếu một nhân tố Truy vấn mục nhập đăng ký không được định rõ x thì chuyển đến bước kế tiếp; nếu không thì, đặt x là một phân loại còn lại trong C. nhân tố Truy vấn mục nhập đăng ký là như sau: đặt RE là kết quả của Truy vấn mục nhập đăng ký được xác định trong Phần 8.2.3. Nếu x không phải là một phân loại của ít nhất một mục nhập đăng ký trong RE, thì xoá x khỏi C. Nếu C trống rỗng thì tiếp tục chuyển đến quy tắc đã được tính đến kế tiếp.
2. nếu C trống rỗng, thì xây dựng cảnh báo: kết quả ClassificationQuery (truy vấn phân loại) trống rỗng; nếu không thì, đặt C là kết quả của ClassificationQuery (truy vấn phân loại);
 3. trở lại kết quả và bất kỳ cảnh báo gia tăng nào hoặc sự kiểu trừ (trong Danh sách lỗi đăng ký) trong RegistryResponse (phản hồi đăng ký) .

8.2.7 ClassificationNodeQuery (truy vấn nút phân loại)

Mục đích:

Để xác định một tập hợp các trường hợp nút phân loại là kết quả của một truy vấn với siêu số liệu đăng ký được lựa chọn.

Quy định ebRIM



Hình 21 - Quy định ebRIM cho ClassificationNodeQuery (truy vấn nút phân loại)

Định nghĩa

```

<complexType name="ClassificationNodeQuery (truy vấn nút phân loại)Type">
<complexContent>
<extension base="tns:RegistryObjectQueryType">
<sequence>
<element ref="tns:ClassificationNodeFilter" minOccurs="0" maxOccurs="1" />
<element ref="tns:ClassificationSchemeQuery (truy vấn giản đồ phân loại)"
minOccurs="0" maxOccurs="1" />
<element name="ClassificationNodeParentBranch" type="ClassificationNodeQuery (truy
vấn nút phân loại)Type" minOccurs="0"
maxOccurs="1" />
<element name="ClassificationNodeChildrenBranch" type="ClassificationNodeQuery (truy
vấn nút phân loại)Type"
minOccurs="0" maxOccurs="unbounded" />
</sequence>
</extension>
</complexContent>
</complexType>
<element name="ClassificationNodeQuery (truy vấn nút phân loại)"
type="tns:ClassificationNodeQuery (truy vấn nút phân loại)Type" />
<element name="ClassificationNodeQuery (truy vấn nút phân loại)Result">
<complexType>
<choice minOccurs="0" maxOccurs="unbounded">
<element ref="rim:ObjectRef" />
<element ref="rim:RegistryObject" />
<element ref="rim:ClassificationNode" />
</choice>
</complexType>
</element>
  
```

Các quy tắc ngữ nghĩa

1. đặt CN biểu thị cho một tập hợp của tất cả các trường hợp Nút phân loại liên tục. Các bước dưới đây sẽ gỡ bỏ các trường hợp trong CN mà không thoả mãn các điều kiện của các bộ lọc đã được xác định;
 - a) nếu CN trống rỗng thì chuyển đến các quy tắc đã được tính đến kế tiếp;
 - b) nếu một Bộ lọc nút phân loại không được xác định thì chuyển đến bước kế tiếp; nếu không thì, đặt x là một nút phân loại trong CN. Nếu x không thoả mãn Bộ lọc nút phân loại thì gỡ bỏ x khỏi CN. Nếu CN trống rỗng thì tiếp tục chuyển đến quy tắc đã được tính đến kế tiếp;

- c) nếu một ClassificationSchemeQuery (truy vấn giản đồ phân loại) không được xác định thì chuyển đến bước kế tiếp; nếu không thì, đặt x là một nút phân loại còn lại trong CN. Nếu giản đồ phân loại xác định của x không thoả mãn ClassificationSchemeQuery (truy vấn giản đồ phân loại) như đã được xác định trong Phần 8.2.8, thì gỡ bỏ x khỏi CN. Nếu CN trống rỗng thì tiếp tục chuyển đến quy tắc đã được tính đến kế tiếp;
- d) nếu một nhân tố nhánh gốc nút phân loại không được xác định, thì chuyển đến bước kế tiếp; nếu không thì, đặt x là một nút phân loại còn lại trong CN và thực hiện theo đoạn dưới đây với $n=x$.

Đặt n là một trường hợp nút phân loại. Nếu n không có một nút gốc (ví dụ nếu n là một nút cấp độ cơ sở), thì gỡ bỏ x khỏi CN và chuyển đến bước tiếp theo; nếu không thì đặt p là nút gốc của n. Nếu một nhân tố Bộ lọc nút phân loại là được bao gồm thẳng Nhánh gốc nút phân loại và nếu p không thoả mãn Bộ lọc nút phân loại, thì sau đó gỡ bỏ x khỏi CN. Nếu CN trống rỗng thì tiếp tục chuyển đến quy tắc đã được tính đến kế tiếp. Nếu một nhân tố ClassificationSchemeQuery (truy vấn giản đồ phân loại) được bao gồm thẳng trong Nhánh gốc nút phân loại và nếu giản đồ phân loại xác định của p không thoả mãn ClassificationSchemeQuery (truy vấn giản đồ phân loại), thì gỡ bỏ x khỏi CN. Nếu CN trống rỗng thì tiếp tục chuyển đến quy tắc đã được tính đến kế tiếp.

Nếu một nhân tố Nhánh gốc nút phân loại khác được bao gồm thẳng với nhân tố Nhánh gốc nút phân loại này, thì lặp lại đoạn trên với $n=p$;

- e) nếu một nhân tố Nhánh con nút phân loại không được xác định thì tiếp tục đến quy tắc đã được tính đến kế tiếp; nếu không thì, đặt x là một nút phân loại còn lại trong CN. Nếu x không phải là nút gốc của một vài trường hợp Nút phân loại, thì gỡ bỏ x khỏi CN và nếu CN trống rỗng thì tiếp tục chuyển đến quy tắc đã được tính đến kế tiếp; nếu không thì coi như mỗi nhân tố Nhánh con nút phân loại tách riêng và thực hiện đoạn dưới đây với $n=x$;

Đặt n là một trường hợp nút phân loại. Nếu một nhân tố Bộ lọc nút phân loại không được xác định với nhân tố Nhánh con nút phân loại thì sau đó đặt CNC là tập hợp của tất cả các nút phân loại mà có n như là nút gốc của chúng; nếu không thì đặt CNC là tập hợp của tất cả các nút phân loại mà thoả mãn Bộ lọc nút phân loại và có n như là nút gốc của chúng. Nếu CNC trống rỗng, thì gỡ bỏ x khỏi CN và nếu CN trống rỗng thì tiếp tục quy tắc đã được tính đến kế tiếp; nếu không thì đặt c là bất kỳ thành viên nào của CNC. Nếu một nhân tố ClassificationSchemeQuery (truy vấn giản đồ phân loại) được bao gồm thẳng với Nhánh con nút phân loại và nếu giản đồ phân loại bao gồm của c không thoả mãn Truy vấn giản đồ phân loại sau đó gỡ bỏ c khỏi CNC. Nếu CNC trống rỗng thì gỡ bỏ x khỏi CN. Nếu CN trống rỗng thì tiếp tục chuyển đến các quy tắc đã được tính đến kế tiếp; nếu không thì để y là một nhân tố của CNC và tiếp tục với các đoạn kế tiếp.

Nếu nhân tố Nhánh con nút phân loại là cuối cùng, ví dụ nếu không bao gồm thẳng một nhân tố Nhánh con nút phân loại khác, sau đây tiếp tục chuyển đến quy tắc đã được tính đến kế tiếp; nếu không thì lặp lại đoạn trước với nhân tố con nút phân loại và với $n=y$.

- f) đặt CN là tập hợp của các trường hợp Nút phân loại. Đánh giá Truy vấn đối tượng đăng nhập với CN như là giải thích trong phần 8.2.2.
2. nếu CN trống rỗng, thì sẽ xây dựng nên loại cảnh báo: **kết quả ClassificationNodeQuery (truy vấn nút phân loại) trống rỗng**; nếu không thì đặt CN là kết quả của ClassificationNodeQuery (truy vấn nút phân loại);
 3. trở lại kết quả và bất kỳ những lời cảnh báo gia tăng hoặc sự kiểm tra nào (trong Danh sách lỗi đăng ký) trong RegistryResponse (phản hồi đăng ký).

Biểu thức bộ lọc theo hướng được sử dụng trong Bộ lọc nút phân loại

Biểu thức bộ lọc hướng được sử dụng để kết nối các nút phân loại trong các nhân tố Bộ lọc nút phân loại mà có liên quan đến thuộc tính hướng của lớp Nút phân loại mà đã bị gây khó khăn bởi phương pháp Hướng trong [ebRIM].

Biểu thức bộ lọc hướng được dựa trên một tập hợp rất nhỏ và thích hợp với cú pháp hướng định vị của Hướng X

Cú pháp biểu thức hướng bao gồm sự hỗ trợ cho việc kết nối các nút phức tạp bằng việc sử dụng cú pháp thẻ tự do như dưới đây:

- việc sử dụng '*' như một ký tự liên kết thay cho bất kỳ nhân tố hướng nào trong Bộ lọc hướng;
- việc sử dụng cú pháp '/' biểu thị cho bất kỳ sự đi xuống nào của một nút trong Bộ lọc hướng.

Nó được xác định bởi ngữ pháp BNF dưới đây:

```
pathFilter ::= '/' schemeld nodePath
nodePath ::= slashes nodeCode
           | slashes '*'
           | slashes nodeCode ( nodePath )?
Slashes ::= '/' | '/'
```

Trong phần ngữ pháp ở trên, giản đồ Id là thuộc tính Id của trường hợp giản đồ phân loại. Trong phần ngữ pháp trên Mã nút được xác định bởi việc đưa ra Tên NC như được xác định bởi :

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

Các quy tắc ngữ nghĩa cho nhân tố Bộ lọc nút phân loại cho phép sử dụng các thuộc tính hướng như là một bộ lọc được dựa trên mệnh đề EQUAL. Mẫu đã được định rõ cho việc liên kết mệnh đề EQUAL là một biểu thức Bộ lọc hướng (PATH).

Điều này được minh họa trong ví dụ dưới đây mà đã liên kết tất cả các nút cấp độ thứ hai trong Giản đồ phân loại với id 'Geograpy-id' và với mã 'Japan':

```

<ClassificationNodeQuery (truy vấn nút phân loại)>
<ClassificationNodeFilter>
<Clause>
<SimpleClause leftArgument = "path">
<StringClause stringPredicate = "Equal">/Geography-id/*/Japan</StringClause>
</SimpleClause>
</Clause>
</ClassificationNodeFilter>
</ClassificationNodeQuery (truy vấn nút phân loại)>

```

Việc sử dụng các trường hợp và ví dụ của Biểu thức bộ lọc hướng

Bảng dưới đây liệt kê danh sách các trường hợp khác nhau được sử dụng và các ví dụ sử dụng giản đồ Địa lý giống nhau:

```

<ClassificationScheme id='Geography-id' name="Geography"/>
<ClassificationNode id="NorthAmerica-id" parent="Geography-id" code="NorthAmerica" />
<ClassificationNode id="UnitedStates-id" parent="NorthAmerica-id"
code="UnitedStates" />
<ClassificationNode id="Asia-id" parent="Geography-id" code="Asia" />
<ClassificationNode id="Japan-id" parent="Asia-id" code="Japan" />
<ClassificationNode id="Tokyo-id" parent="Japan-id" code="Tokyo" />

```

Bảng 10 - Các biểu thức bộ lọc hướng cho các trường hợp được sử dụng

Trường hợp sử dụng	Biểu thức PATH	Mô tả
Match all nodes in first level that have a specified value	/Geography-id/NorthAmerica	Find all first level nodes whose code is 'NorthAmerica'
Find all children of first level node whose code is "NorthAmerica"	/Geography-id/NorthAmerica/*	Match all nodes whose first level path element has code "NorthAmerica"
Match all nodes that have a specified value regardless of level	/Geography-id//Japan	Find all nodes with code "Japan"
Match all nodes in the second level that have a specified value	/Geography-id/*/Japan	Find all second level nodes with code 'Japan'
Match all nodes in the 3rd level that have a specified value	/Geography-id/*/*/Tokyo	Find all third level nodes with code 'Tokyo'

Các ví dụ

Một khách hàng mong muốn xác định tất cả các nút phân loại ở tất cả ba cấp đầu tiên của hệ thống cấp bậc giản đồ phân loại. Người khác đó biết rằng tên của giản đồ phân loại cơ sở là "urn:ebxml:cs:myscheme". Truy vấn dưới đây sẽ xác định tất cả các nút nằm trong ba cấp đầu tiên.

```

<AdhocQueryRequest>
<ResponseOption returnType = "LeafClass"/>
<FilterQuery>
<ClassificationNodeQuery (truy vấn nút phân loại)>
<ClassificationNodeFilter>
<Clause>
<SimpleClause leftArgument = "levelNumber">
<RationalClause logicalPredicate = "LE">
<IntClause>3</IntClause>

```

```
</RationalClause>
</SimpleClause>
</Clause>
</ClassificationNodeFilter>
<ClassificationSchemeQuery (truy vấn giản đồ phân loại)>
  <NameBranch>
    <LocalizedStringFilter>
      <Clause>
        <SimpleClause leftArgument = "value">
          <StringClause stringPredicate = "Equal">urn:ebxml:cs:myscheme</StringClause>
        </SimpleClause>
      </Clause>
    </LocalizedStringFilter>
  </NameBranch>
</ClassificationSchemeQuery (truy vấn giản đồ phân loại)>
</ClassificationNodeQuery (truy vấn nút phân loại)>
</FilterQuery>
</AdhocQueryRequest>
```

Thay vào đó, nếu vị khách hàng này muốn quay lại tất cả các mức, thì đơn giản chỉ cần xoá bỏ nhân tố Bộ lọc nút phân loại từ truy vấn.

Truy vấn dưới đây tìm tất cả nút con của nút mức đầu tiên mà mã của nó là Bắc Mỹ (North America)

```
<AdhocQueryRequest>
  <ResponseOption returnType = "LeafClass"/>
  <FilterQuery>
    <ClassificationNodeQuery (truy vấn nút phân loại)>
      <ClassificationNodeFilter>
        <Clause>
          <SimpleClause leftArgument = "path">
            <StringClause stringPredicate = "Equal">/Geography-id/NorthAmerica/*</StringClause>
          </SimpleClause>
        </Clause>
      </ClassificationNodeFilter>
    </ClassificationNodeQuery (truy vấn nút phân loại)>
  </FilterQuery>
</AdhocQueryRequest>
```

Truy vấn dưới đây tìm ra tất cả các nút mức thứ ba với mã là Tokyo

```
<AdhocQueryRequest>
  <ResponseOption returnType = "LeafClass" returnComposedObjects = "True"/>
  <FilterQuery>
    <ClassificationNodeQuery (truy vấn nút phân loại)>
      <ClassificationNodeFilter>
        <Clause>
          <SimpleClause leftArgument = "path">
            <StringClause stringPredicate = "Equal">/Geography-id/*/*Tokyo</StringClause>
          </SimpleClause>
        </Clause>
      </ClassificationNodeFilter>
    </ClassificationNodeQuery (truy vấn nút phân loại)>
  </FilterQuery>
</AdhocQueryRequest>
```

8.2.8 ClassificationSchemeQuery (truy vấn giản đồ phân loại)

Mục đích:

Để xác định một tập hợp các trường hợp giản đồ phân loại là kết quả của một truy vấn với siêu số liệu đăng ký được lựa chọn.

Quy định ebRIM



Hình 22 - Quy định ebRIM cho ClassificationSchemeQuery (truy vấn gián đồ phân loại)

Định nghĩa

```

<complexType name="ClassificationSchemeQuery (truy vấn gián đồ phân loại)Type">
<complexContent>
<extension base="tns:RegistryEntryQueryType">
<sequence>
<element ref="tns:ClassificationSchemeFilter" minOccurs="0" maxOccurs="1" />
</sequence>
</extension>
</complexContent>
</complexType>
<element name="ClassificationSchemeQuery (truy vấn gián đồ phân loại)"
type="tns:ClassificationSchemeQuery (truy vấn gián đồ phân loại)Type" />
  
```

Các quy tắc ngữ nghĩa

- đặt CS biểu thị cho tập hợp của tất cả các trường hợp Giản đồ phân loại liên tục trong sổ đăng ký. Các bước dưới đây sẽ gỡ bỏ những trường hợp không thoả mãn các điều kiện của các bộ lọc đã được xác định;
 - nếu CS trống rỗng thì tiếp tục đến quy tắc đã được tính đến kế tiếp;
 - nếu một Bộ lọc gián đồ phân loại không được xác định thì chuyển đến bước kế tiếp; nếu không thì đặt x là một gián đồ phân loại trong CS. Nếu x không thoả mãn Bộ lọc gián đồ phân loại thì gỡ bỏ x khỏi CS. Nếu CS trống rỗng thì tiếp tục chuyển đến quy tắc kế tiếp đã được tính đến;
 - đặt CS là một tập hợp của các trường hợp Giản đồ phân loại còn lại. Đánh giá Truy vấn mục nhập đăng ký được kế thừa với khắp CS đã được giải thích trong Phần 8.2.3.
- nếu CS trống rỗng, khi đó xây dựng cảnh báo: kết quả ClassificationSchemeQuery (truy vấn gián đồ phân loại) trống rỗng; nếu không thì tập hợp CS là kết quả của ClassificationSchemeQuery (truy vấn gián đồ phân loại);
- quay lại kết quả và bất kỳ cảnh báo gia tăng nào hoặc các kiểu trừ (trong Danh sách lỗi đăng ký) trong RegistryResponse (phản hồi đăng ký) .

Các ví dụ

Một khách hàng mong muốn xác định tất cả các trường hợp gián đồ phân loại trong sổ đăng ký.

```

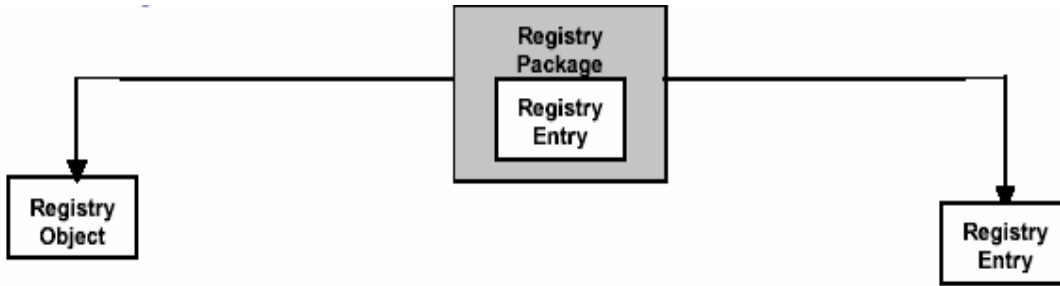
<AdhocQueryRequest>
<ResponseOption returnType = "LeafClass"/>
<FilterQuery>
<ClassificationSchemeQuery (truy vấn gián đồ phân loại)/>
</FilterQuery>
</AdhocQueryRequest>
  
```

8.2.9 RegistryPackageQuery (truy vấn gói đăng ký)

Mục đích:

Để xác định một tập hợp các trường hợp gói đăng ký là kết quả của một truy vấn với siêu số liệu đăng ký đã được lựa chọn.

Quy định ebRIM



Hình 23 - Quy định ebRIM cho RegistryPackageQuery (truy vấn gói đăng ký)

Định nghĩa

```

<complexType name="RegistryPackageQuery (truy vấn gói đăng ký)Type">
<complexContent>
<extension base="tns:RegistryEntryQueryType">
<sequence>
<element ref="tns:RegistryPackageFilter" minOccurs="0" maxOccurs="1" />
<element ref="tns:RegistryObjectQuery" minOccurs="0" maxOccurs="unbounded" />
<element ref="tns:RegistryEntryQuery" minOccurs="0" maxOccurs="unbounded" />
</sequence>
</extension>
</complexContent>
</complexType>
<element name="RegistryPackageQuery (truy vấn gói đăng ký)"
type="tns:RegistryPackageQuery (truy vấn gói đăng ký)Type" />
<element name="RegistryPackageQuery (truy vấn gói đăng ký)Result">
<complexType>
<choice minOccurs="0" maxOccurs="unbounded">
<element ref="rim:ObjectRef" />
<element ref="rim:RegistryEntry" />
<element ref="rim:RegistryObject" />
<element ref="rim:RegistryPackage" />
</choice>
</complexType>
</element>
  
```

Các quy tắc ngữ nghĩa

1. đặt RP biểu thị cho tập hợp của tất cả các trường hợp Gói đăng ký trong sổ đăng ký. Các bước dưới đây sẽ gỡ bỏ các trường hợp trong RP mà không thoả mãn các điều kiện của các bộ lọc đã được xác định;
 - a) nếu RP trống rỗng thì tiếp tục đến các quy tắc đã được xác định kế tiếp;
 - b) nếu một Bộ lọc gói đăng ký không được xác định, thì chuyển đến quy tắc đã được xác định kế tiếp; nếu không thì đặt x là một trường hợp gói đăng ký trong RP. Nếu x không thoả mãn Bộ lọc gói đăng ký thì xoá x khỏi RP. Nếu RP trống rỗng thì tiếp tục quy tắc đã được xác định kế tiếp;
 - c) nếu một nhân tố Truy vấn đối tượng đăng ký được bao gồm trực tiếp trong nhân tố RegistryPackageQuery (truy vấn gói đăng ký) thì sau đó xem xét mỗi Truy vấn đối tượng đăng ký như sau: để RO là tập hợp của các trường hợp RegistryObject (đối tượng đăng ký) được trả

lại bởi Truy vấn đối tượng đăng ký mà được xác định trong Phần 8.2.2 và đặt PO là tập hợp con của RO mà là các thành viên của gói x. Nếu PO trống rỗng, thì xoá x khỏi RP. Nếu RP trống rỗng thì tiếp tục chuyển đến các quy tắc đã được xác định kế tiếp. Nếu một nhân tố Truy vấn mục nhập đăng ký được bao gồm trực tiếp trong nhân tố RegistryPackageQuery (truy vấn gói đăng ký) thì xem xét mỗi Truy vấn mục nhập đăng ký như sau: đặt RE là tập hợp của các trường hợp Mục nhập đăng ký được trả lại bởi Truy vấn mục nhập đăng ký đã được xác định trong Phần 8.2.3 và đặt PE là tập hợp con của RE mà là thành viên của gói x. Nếu PE trống rỗng, thì xoá x khỏi RP. Nếu RP trống rỗng thì tiếp tục chuyển đến quy tắc đã được tính đến kế tiếp;

- d) đặt RP là tập hợp của các trường hợp Gói đăng ký còn lại. Đánh giá Truy vấn mục nhập đăng ký kế thừa khắp RP như đã được giải thích trong Phần 8.2.3.
2. nếu RP trống rỗng, thì xây dựng cảnh báo: kết quả RegistryPackageQuery (truy vấn gói đăng ký) trống rỗng; nói cách khác tập hợp RP là kết quả của RegistryPackageQuery (truy vấn gói đăng ký);
 3. quay lại kết quả và bất kỳ các cảnh báo gia tăng nào hoặc các kiểu trừ (trong Danh sách lỗi đăng ký) trong RegistryResponse (phản hồi đăng ký).

Các ví dụ

Một khách hàng mong muốn xác định tất cả các trường hợp gói trong sổ đăng ký mà bao gồm một ExtrinsicObjects (đối tượng ngoại lai) mang hoá đơn như là một thành viên của gói.

```
<AdhocQueryRequest>
<ResponseOption returnType = "LeafClass"/>
<FilterQuery>
<RegistryPackageQuery (truy vấn gói đăng ký)>
<RegistryEntryQuery>
<RegistryEntryFilter>
<Clause>
<SimpleClause leftArgument = "objectType">
<StringClause stringPredicate = "Equal">Invoice</StringClause>
</SimpleClause>
</Clause>
</RegistryEntryFilter>
</RegistryEntryQuery>
</RegistryPackageQuery (truy vấn gói đăng ký)>
</FilterQuery>
</AdhocQueryRequest>
```

Một khách hàng mong muốn xác định tất cả các trường hợp gói trong sổ đăng ký mà không trống rỗng.

```
<AdhocQueryRequest>
<ResponseOption returnType = "LeafClass"/>
<FilterQuery>
<RegistryPackageQuery (truy vấn gói đăng ký)>
<RegistryObjectQuery/>
</RegistryPackageQuery (truy vấn gói đăng ký)>
</FilterQuery>
</AdhocQueryRequest>
```

Một khách hàng mong muốn xác định tất cả các gói trong sổ đăng ký mà trống rỗng. Vì RegistryPackageQuery (truy vấn gói đăng ký) không được cài đặt cho thao tác phủ định, vì vậy khách hàng này phải thực hiện hai đề nghị RegistryPackageQuery (truy vấn gói đăng ký) riêng biệt, một là tìm ra tất cả các gói và thứ hai là tìm ra tất cả các gói không trống rỗng và sau đó sẽ tìm ra sự khác biệt từ

kết quả. Như vậy, họ có thể thực hiện nhiều Truy vấn mục nhập đăng ký phức tạp hơn và kiểm tra sự liên kết gói giữa gói và các thành viên của nó mà không tồn tại

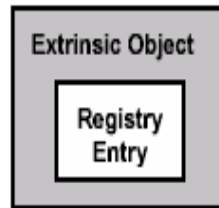
Ghi nhớ: một gói đăng ký là một trường hợp Mục nhập đăng ký bên trong mà hoàn toàn được xác định bởi những liên kết với những thành viên của nó. Vì vậy một RegistryPackageQuery (truy vấn gói đăng ký) có thể luôn luôn được xác định lại như một Truy vấn mục nhập đăng ký tương đương sử dụng các liên kết “Nguồn” và “Mục tiêu”. Tuy nhiên, truy vấn mục nhập đăng ký tương đương thường khá phức tạp khi viết.

8.2.10 ExtrinsicObjectQuery (truy vấn đối tượng ngoại lai)

Mục đích

Để xác định một tập hợp các trường hợp ExtrinsicObjects là kết quả của một truy vấn với siêu dữ liệu đăng ký đã được lựa chọn.

Quy định ebRIM



Hình 24 - Quy định ebRIM cho ExtrinsicObjectQuery (truy vấn đối tượng ngoại lai)

Định nghĩa

```
<complexType name="ExtrinsicObjectQuery (truy vấn đối tượng ngoại lai)Type">
<complexContent>
<extension base="tns:RegistryEntryQueryType">
<sequence>
<element ref="tns:ExtrinsicObjectFilter" minOccurs="0" maxOccurs="1" />
</sequence>
</extension>
</complexContent>
</complexType>
<element name="ExtrinsicObjectQuery (truy vấn đối tượng ngoại lai)"
type="tns:ExtrinsicObjectQuery (truy vấn đối tượng ngoại lai)Type" />
<element name="ExtrinsicObjectQuery (truy vấn đối tượng ngoại lai)Result">
<complexType>
<choice minOccurs="0" maxOccurs="unbounded">
<element ref="rim:ObjectRef" />
<element ref="rim:RegistryEntry" />
<element ref="rim:RegistryObject" />
<element ref="rim:ExtrinsicObject" />
</choice>
</complexType>
</element>
```

Các quy tắc ngữ nghĩa

1. đặt EO biểu thị cho tập hợp của tất cả các trường hợp ExtrinsicObjects trong sổ đăng ký. Các bước tiếp theo sẽ gỡ bỏ các trường hợp trong EO mà không thoả mãn các điều kiện của các bộ lọc được xác định;
 - a) nếu EO trống rỗng thì chuyển tiếp đến các quy tắc đã được tính đến kế tiếp;

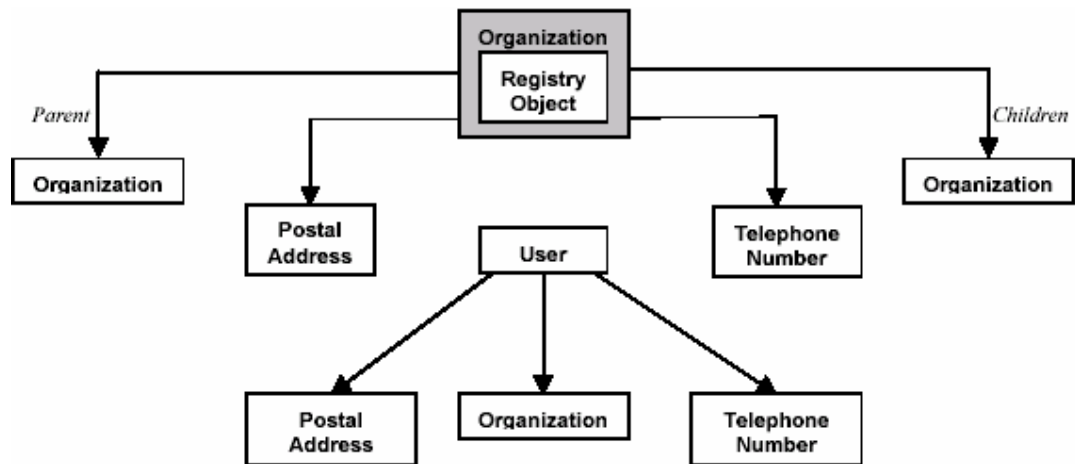
- b) nếu một Bộ lọc ExtrinsicObjects không được xác định thì chuyển đến bước kế tiếp; nếu không thì đặt x là một ExtrinsicObjects (đối tượng ngoại lai) trong EO. Nếu EO trống rỗng thì chuyển đến bước đã được tính đến kế tiếp;
- e) đặt EO là tập hợp của các trường hợp ExtrinsicObjects còn lại. Đánh giá Truy vấn mục nhập đăng ký kế thừa với EO như đã được giải thích trong Phần 8.2.3.
- if EO is empty, then raise the warning: extrinsic object query result is empty; otherwise, set EO to be the result of the ExtrinsicObjectQuery (truy vấn đối tượng ngoại lai);
 - return the result and any accumulated warnings or exceptions (in the RegistryErrorList) within the RegistryResponse.

8.2.11 OrganizationQuery (truy vấn tổ chức)

Mục đích:

Để xác định một tập hợp của các trường hợp tổ chức là kết quả của một truy vấn đối với siêu dữ liệu đăng ký đã được lựa chọn.

Quy định ebRIM



Định nghĩa

```

<complexType name="OrganizationQuery (truy vấn tổ chức)Type">
  <complexContent>
    <extension base="tns:RegistryObjectQueryType">
      <sequence>
        <element ref="tns:OrganizationFilter" minOccurs="0" maxOccurs="1" />
        <element ref="tns:PostalAddressFilter" minOccurs="0" maxOccurs="1" />
        <element ref="tns:TelephoneNumberFilter" minOccurs="0" maxOccurs="unbounded" />
        <element ref="tns:UserBranch" minOccurs="0" maxOccurs="1" />
        <element name="OrganizationParentBranch" type="tns:OrganizationQuery (truy vấn tổ chức)Type" minOccurs="0" maxOccurs="1" />
        <element name="OrganizationChildrenBranch" type="tns:OrganizationQuery (truy vấn tổ chức)Type" minOccurs="0" maxOccurs="unbounded" />
      </sequence>
    </extension>
  </complexContent>
</complexType>
  
```

```
<element name="OrganizationQuery (truy vấn tổ chức)" type="tns:OrganizationQuery
(truy vấn tổ chức)Type" />
<element name="OrganizationQuery (truy vấn tổ chức)Result">
<complexType> 2728
<choice minOccurs="0" maxOccurs="unbounded">
<element ref="rim:ObjectRef" />
<element ref="rim:RegistryObject" />
<element ref="rim:Organization" />
</choice>
</complexType>
</element>
```

Các quy tắc ngữ nghĩa

1. đặt ORG biểu thị cho tập hợp của tất cả các trường hợp Tổ chức liên tục trong sổ đăng ký. Các bước dưới đây sẽ gỡ bỏ những trường hợp trong ORG mà không thoả mãn các điều kiện của các bộ lọc đã được xác định;
 - a) nếu ORG trống rỗng thì tiếp tục chuyển đến quy tắc đã được tính đến kế tiếp;
 - b) nếu một nhân tố Bộ lọc tổ chức không được bao gồm trực tiếp trong nhân tố OrganizationQuery (truy vấn tổ chức) thì chuyển tiếp đến bước kế tiếp; nếu không thì đặt x là một trường hợp tổ chức trong ORG. Nếu x không thoả mãn Bộ lọc tổ chức thì gỡ bỏ x khỏi ORG. Nếu ORG trống rỗng thì tiếp tục đến quy tắc đã được tính đến kế tiếp;
 - c) nếu một nhân tố Bộ lọc địa chỉ bưu điện không được bao gồm trực tiếp trong nhân tố OrganizationQuery (truy vấn tổ chức) thì chuyển đến bước kế tiếp; nếu không thì đặt x là một ExtrinsicObjects (đối tượng ngoại lai) trong ORG. Nếu địa chỉ bưu điện của x không thoả mãn điều kiện của Bộ lọc địa chỉ bưu điện thì gỡ bỏ x khỏi ORG. Nếu ORG trống rỗng thì tiếp tục đến quy tắc đã được tính đến kế tiếp;
 - d) nếu không có nhân tố Bộ lọc số điện thoại được bao gồm trực tiếp trong nhân tố OrganizationQuery (truy vấn tổ chức) thì chuyển đến bước kế tiếp; nếu không thì đặt x là một ExtrinsicObjects (đối tượng ngoại lai) trong ORG. Nếu bất kỳ Bộ lọc số điện thoại nào không được thoả mãn bởi một vài số điện thoại của x thì gỡ bỏ x khỏi ORG. Nếu ORG trống rỗng thì chuyển đến quy tắc đã được tính đến kế tiếp;
 - e) nếu một nhân tố Nhánh người sử dụng không được bao gồm trực tiếp trong nhân tố OrganizationQuery (truy vấn tổ chức) thì chuyển đến bước kế tiếp; nếu không thì đặt x là một ExtrinsicObjects (đối tượng ngoại lai) trong ORG. Đặt u là một trường hợp người sử dụng mà được nhập vào với x. Nếu một nhân tố Bộ lọc người sử dụng được xác định trong phạm vi Nhánh người sử dụng, và nếu u không thoả mãn bộ lọc này thì gỡ bỏ x khỏi ORG. Nếu một nhân tố Bộ lọc địa chỉ bưu điện được xác định trong phạm vi Nhánh người sử dụng và nếu địa chỉ bưu điện của u không thoả mãn bộ lọc này thì gỡ bỏ x khỏi ORG. Nếu (các) Bộ lọc số điện thoại được xác định trong phạm vi Nhánh người sử dụng và nếu bất kỳ Bộ lọc số điện thoại nào không được thoả mãn bởi một vài số điện thoại của x thì gỡ bỏ x khỏi ORG. Nếu (các) Bộ lọc địa chỉ Email được xác định trong phạm vi Nhánh người sử dụng và nếu bất kỳ Bộ lọc địa chỉ Email nào không được thoả mãn bởi một vài địa chỉ Email của x thì gỡ bỏ x khỏi ORG. Nếu một nhân

tổ OrganizationQuery (truy vấn tổ chức) được xác định trong phạm vi Nhánh người sử dụng thì đặt o là trường hợp Tổ chức mà được xác định tổ chức mà u được nhập vào. Nếu o không thoả mãn OrganizationQuery (truy vấn tổ chức) như được xác định trong Phần 8.2.11 thì gỡ bỏ x khỏi ORG. Nếu ORG trống rỗng thì tiếp tục chuyển đến quy tắc đã được tính đến kế tiếp;

- f) nếu một nhân tố Nhánh gốc tổ chức không được xác định trong phạm vi OrganizationQuery (truy vấn tổ chức) thì chuyển đến bước kế tiếp; nếu không thì đặt x là một ExtrinsicObjects (đối tượng ngoại lai) trong ORG. thực hiện theo đoạn dưới đây với $o=x$:

Đặt o là một trường hợp tổ chức. Nếu một Bộ lọc tổ chức không được xác định trong phạm vi Nhánh gốc tổ chức và nếu o không có nhánh gốc (ví dụ nếu o là một tổ chức gốc trong hệ thống cấp bậc của tổ chức) thì kiểu x khỏi ORG; nếu không thì đặt p là nhánh gốc của o. Nếu p không thoả mãn Bộ lọc tổ chức thì kiểu x khỏi ORG.

Nếu ORG trống rỗng thì tiếp tục quy tắc đã được tính đến kế tiếp.

Nếu một nhân tố Nhánh gốc tổ chức khác được bao gồm trực tiếp trong phạm vi nhân tố Nhánh gốc tổ chức này thì lặp lại đoạn trên với $o=p$;

- g) nếu một nhân tố Nhánh con tổ chức không được xác định thì tiếp tục quy tắc đã được tính đến kế tiếp; nếu không thì đặt x là tổ chức còn lại trong ORG. Nếu x không phải là nút gốc của một vài trường hợp tổ chức thì kiểu x khỏi ORG và nếu ORG trống rỗng thì tiếp tục quy tắc đã được tính đến kế tiếp; nếu không thì xem xét mỗi nhân tố Nhánh con tổ chức riêng biệt và thực hiện đoạn dưới đây với $n=x$.

Đặt n là một trường hợp tổ chức. Nếu một nhân tố Bộ lọc tổ chức không được xác định trong phạm vi nhân tố Nhánh con tổ chức thì đặt ORGC là tập hợp của tất cả các tổ chức mà có n là nút gốc của chúng; nếu không thì đặt ORGC là tập hợp của tất cả các tổ chức mà thoả mãn Bộ lọc tổ chức và có n như là nút gốc của chúng. Nếu ORGC trống rỗng thì kiểu x khỏi ORG và nếu ORG trống rỗng thì tiếp tục quy tắc đã được tính đến kế tiếp; nếu không thì đặt c là bất kỳ thành viên nào của ORGC. Nếu một nhân tố Bộ lọc địa chỉ bưu điện bao gồm trực tiếp trong nhánh con tổ chức và nếu địa chỉ bưu điện của c không thoả mãn Bộ lọc địa chỉ bưu điện thì gỡ bỏ c khỏi ORGC. Nếu ORGC trống rỗng thì kiểu x khỏi ORG. Nếu ORG trống rỗng thì tiếp tục quy tắc đã được tính đến kế tiếp. Nếu không có nhân tố Bộ lọc số điện thoại được bao gồm trực tiếp trong Nhánh con tổ chức và nếu bất kỳ Bộ lọc số điện thoại nào không được thoả mãn bởi một vài số điện thoại của c thì gỡ bỏ x khỏi ORGC.

Nếu ORGC trống rỗng thì gỡ bỏ x khỏi ORG. Nếu ORG trống rỗng thì tiếp tục đến quy tắc đã được tính đến kế tiếp; nếu không thì đặt y là một nhân tố của ORGC và tiếp tục đến đoạn kế tiếp.

Nếu nhân tố Nhánh con tổ chức là tận cùng ví dụ nếu nó không bao gồm trực tiếp một nhân tố Nhánh con tổ chức khác thì tiếp tục đến quy tắc đã được tính đến kế tiếp; nếu không thì lặp lại đoạn trước với nhân tố Nhánh con tổ chức mới và với $n=y$;

- h) đặt ORG là tập hợp của các trường hợp Tổ chức còn lại. Đánh giá Truy vấn đối tượng đăng ký kế thừa đối với ORG như đã được giải thích trong Phần 8.2.2.
2. nếu ORG trống rỗng thì xây dựng cảnh báo: kết quả OrganizationQuery (truy vấn tổ chức) trống rỗng; nếu không thì tập hợp ORG là kết quả của OrganizationQuery (truy vấn tổ chức);
 3. trở lại kết quả và bất kỳ cảnh báo gia tăng hoặc kiểu trừ nào (trong Danh sách lỗi đăng ký) trong RegistryResponse (phản hồi đăng ký).

Các ví dụ

Một khách hàng muốn xác định một tập hợp các tổ chức được đặt cơ sở ở France, vì vậy đã đệ trình một ExtrinsicObjects (đối tượng ngoại lai) Hồ sơ tổ chức năm nay.

```
<AdhocQueryRequest>
<ResponseOption returnType = "LeafClass" returnComposedObjects = "True"/>
<FilterQuery>
<OrganizationQuery (truy vấn tổ chức)>
<SourceAssociationBranch>
<AssociationFilter>
<Clause>
<SimpleClause leftArgument = "associationType">
<StringClause stringPredicate = "Equal">SubmitterOf</StringClause>
</SimpleClause>
</Clause>
</AssociationFilter>
<RegistryObjectQuery>
<RegistryObjectFilter>
<Clause>
<SimpleClause leftArgument = "objectType">
<StringClause stringPredicate = "Equal">CPP</StringClause>
</SimpleClause>
</Clause>
</RegistryObjectFilter>
<AuditableEventQuery>
<AuditableEventFilter>
<Clause>
<SimpleClause leftArgument = "timestamp">
<RationalClause logicalPredicate = "GE">
<DateTimeClause>2000-01-01T00:00:00-05:00</DateTimeClause>
</RationalClause>
</SimpleClause>
</Clause>
</AuditableEventFilter>
</AuditableEventQuery>
</RegistryObjectQuery>
</SourceAssociationBranch>
<PostalAddressFilter>
<Clause>
<SimpleClause leftArgument = "country">
<StringClause stringPredicate = "Equal">France</StringClause>
</SimpleClause>
</Clause>
</PostalAddressFilter>
</OrganizationQuery (truy vấn tổ chức)>
</FilterQuery>
</AdhocQueryRequest>
```

Một khách hàng muốn xác định tất cả các tổ chức mà có Đoàn thể (Corporation) được gọi là XYZ như là một gốc.

```

<AdhocQueryRequest>
<ResponseOption returnType = "LeafClass"/>
<FilterQuery>
<OrganizationQuery (truy vấn tổ chức)>
<OrganizationParentBranch>
<NameBranch>
<LocalizedStringFilter>
<Clause>
<SimpleClause leftArgument = "value">
<StringClause stringPredicate = "Equal">XYZ</StringClause>
</SimpleClause>
</Clause>
</LocalizedStringFilter>
</NameBranch>
</OrganizationParentBranch>
</OrganizationQuery (truy vấn tổ chức)>
</FilterQuery>
</AdhocQueryRequest>

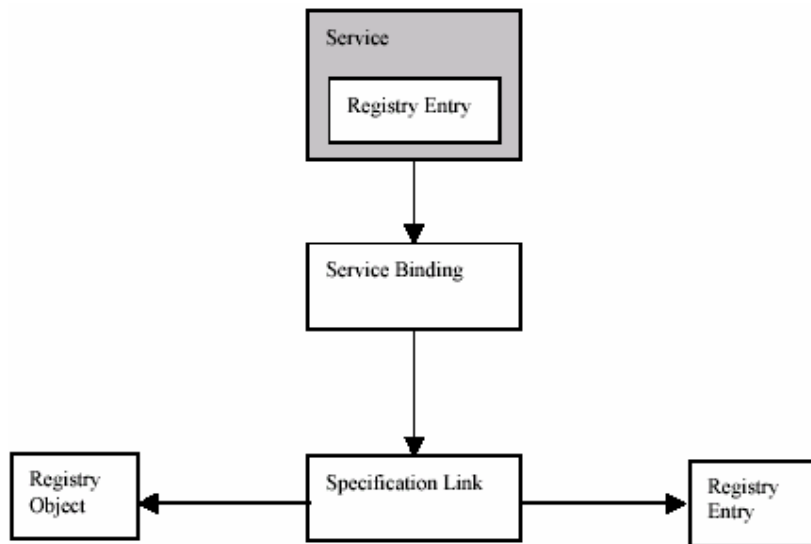
```

8.2.12 OrganizationQuery (truy vấn dịch vụ)

Mục đích:

Để xác định một tập hợp các trường hợp dịch vụ là kết quả của một truy vấn đối với siêu dữ liệu đăng ký đã được lựa chọn.

Quy định ebRIM



Hình 26 - Quy định ebRIM cho OrganizationQuery (truy vấn dịch vụ)

Định nghĩa

```

<complexType name="OrganizationQuery (truy vấn dịch vụ)Type">
<complexContent>
<extension base="tns:RegistryEntryQueryType">
<sequence>
<element ref="tns:ServiceFilter" minOccurs="0"
maxOccurs="1" />
<element ref="tns:ServiceBindingBranch" minOccurs="0"
maxOccurs="unbounded" />

```

```
</sequence>
</extension>
</complexContent>
</complexType>
<element name="OrganizationQuery (truy vấn dịch vụ)" type="tns:OrganizationQuery
(truy vấn dịch vụ)Type" />
<element name="OrganizationQuery (truy vấn dịch vụ)Result">
<complexType>
<choice minOccurs="0" maxOccurs="unbounded">
<element ref="rim:ObjectRef" />
<element ref="rim:RegistryObject" />
<element ref="rim:Service" />
</choice>
</complexType>
</element>
```

Các quy tắc ngữ nghĩa

1. đặt S biểu thị cho tập hợp của tất cả các trường hợp Dịch vụ liên tục trong sổ đăng ký. Các bước tiếp theo sẽ gỡ bỏ các trường hợp không thoả mãn các điều kiện của các bộ lọc đã được xác định;
 - a) nếu S trống rỗng thì tiếp tục chuyển đến quy tắc đã được tính đến kế tiếp. Nếu một Bộ lọc dịch vụ không được xác định thì chuyển đến bước kế tiếp; nếu không thì đặt x là một dịch vụ trong S. Nếu x không thoả mãn Bộ lọc dịch vụ, thì gỡ bỏ x khỏi S. Nếu S trống rỗng thì tiếp tục đến quy tắc đã được tính đến kế tiếp;

- b) nếu một Nhánh Quy định dịch vụ không được xác định thì tiếp tục quy tắc đã được tính đến kế tiếp; nếu không thì cân nhắc mỗi nhân tố Nhánh Quy định dịch vụ riêng biệt như dưới đây:

Đặt SB là tập hợp của tất cả các trường hợp Quy định dịch vụ mà mô tả Quy định của x. Đặt sb là thành viên của SB. Nếu một nhân tố Bộ lọc Quy định dịch vụ được xác định trong phạm vi Nhánh Quy định dịch vụ, và nếu sb không thoả mãn bộ lọc này, thì kiểu sb khỏi SB. Nếu SB trống rỗng thì kiểu x khỏi S. Nếu S trống rỗng thì tiếp tục quy tắc đã được tính đến kế tiếp. Nếu một Nhánh liên kết quy định kỹ thuật không được xác định trong phạm vi Nhánh Quy định dịch vụ thì tiếp tục quy tắc đã được tính đến kế tiếp; nếu không thì cân nhắc mỗi nhân tố Nhánh liên kết quy định kỹ thuật riêng biệt dưới đây:

Đặt sb là một Quy định dịch vụ còn lại trong SB. Đặt SL là tập hợp của tất cả các trường hợp liên kết quy định kỹ thuật sl mà mô tả các liên kết quy định kỹ thuật của sb. Nếu một nhân tố Bộ lọc liên kết quy định kỹ thuật được xác định trong phạm vi Nhánh liên kết quy định kỹ thuật, và nếu sl không thoả mãn bộ lọc này, thì gỡ bỏ sl khỏi SL. Nếu SL trống rỗng thì gỡ bỏ sb khỏi SB. Nếu SB trống rỗng thì gỡ bỏ x khỏi S. Nếu S trống rỗng thì tiếp tục quy tắc đã được tính toán kế tiếp. Nếu một nhân tố Truy vấn đối tượng đăng ký được xác định trong phạm vi Nhánh liên kết quy định kỹ thuật thì đặt sl là một liên kết quy định kỹ thuật còn lại trong SL. Xem xét nhân tố Truy vấn đối tượng đăng ký như sau: Đặt RO là tập hợp kết quả của Truy vấn đối tượng đăng ký như đã được xác định trong Phần 8.2.2. Nếu sl không phải là một liên kết quy định kỹ thuật cho một số đối tượng đăng ký trong RO, thì gỡ bỏ sl khỏi SL. Nếu SL trống rỗng thì tiếp tục gỡ bỏ sb khỏi SB. Nếu SB trống rỗng thì gỡ bỏ x khỏi S. Nếu S trống rỗng thì tiếp tục đến quy tắc đã được tính đến kế tiếp. Nếu một nhân tố Truy vấn mục nhập đăng ký được xác định trong

phạm vi Nhánh liên kết quy định kỹ thuật thì đặt *sl* là một liên kết quy định kỹ thuật còn lại trong *SL*. Xem xét nhân tố Truy vấn mục nhập đăng ký như sau: Đặt *RE* là tập hợp kết quả của Truy vấn mục nhập đăng ký như đã được xác định trong Phần 8.2.3. Nếu *SL* không phải là một liên kết quy định kỹ thuật cho một số mục nhập đăng ký trong *RE* thì gỡ bỏ *sl* khỏi *SL*. Nếu *SL* trống rỗng thì gỡ bỏ *sb* khỏi *SB*. Nếu *SB* trống rỗng thì gỡ bỏ *x* khỏi *S*. Nếu *S* trống rỗng thì tiếp tục đến quy tắc đã được tính đến kế tiếp;

- c) đặt *S* là tập hợp của các trường hợp Dịch vụ còn lại. Đánh giá Truy vấn mục nhập đăng ký kế thừa đối với *AE* như đã được giải thích trong Phần 8.2.3.
2. nếu *S* trống rỗng thì xây dựng cảnh báo: kết quả *OrganizationQuery* (truy vấn dịch vụ) trống rỗng; nếu không thì đặt *S* là kết quả của *OrganizationQuery* (truy vấn dịch vụ);
3. trở lại kết quả và bất kỳ các cảnh báo gia tăng hoặc kiểu trừ nào (trong Danh sách lỗi đăng ký) trong *RegistryResponse* (phản hồi đăng ký) .

Các ví dụ

8.2.13 Các bộ lọc của số đăng ký

Mục đích:

Để xác định một tập hợp con của tập hợp tất cả các trường hợp liên tục của một lớp đăng ký đã cho

Định nghĩa

```
<complexType name="FilterType">
<sequence>
<element ref="tns:Clause" />
</sequence>
</complexType>
<element name="RegistryObjectFilter" type="tns:FilterType" />
<element name="RegistryEntryFilter" type="tns:FilterType" />
<element name="ExtrinsicObjectFilter" type="tns:FilterType" />
<element name="RegistryPackageFilter" type="tns:FilterType" />
<element name="OrganizationFilter" type="tns:FilterType" />
<element name="ClassificationNodeFilter" type="tns:FilterType" />
<element name="AssociationFilter" type="tns:FilterType" />
<element name="ClassificationFilter" type="tns:FilterType" />
<element name="ClassificationSchemeFilter" type="tns:FilterType" />
<element name="ExternalLinkFilter" type="tns:FilterType" />
<element name="ExternalIdentifierFilter" type="tns:FilterType" />
<element name="SlotFilter" type="tns:FilterType" />
<element name="AuditableEventFilter" type="tns:FilterType" />
<element name="UserFilter" type="tns:FilterType" />
<element name="SlotValueFilter" type="tns:FilterType" />
<element name="PostalAddressFilter" type="tns:FilterType" />
<element name="TelephoneNumberFilter" type="tns:FilterType" />
<element name="ServiceFilter" type="tns:FilterType" />
<element name="ServiceBindingFilter" type="tns:FilterType" />
<element name="SpecificationLinkFilter" type="tns:FilterType" />
<element name="LocalizedStringFilter" type="tns:FilterType" />
```

Các quy tắc ngữ nghĩa

1. nhân tố mệnh đề được định nghĩa trong Phần 8.2.14;

2. với mọi Bộ lọc đối tượng đăng ký nhân tố XML, thuộc tính đối số trái của bất kỳ Mệnh đề đơn bao gồm sẽ xác định một thuộc tính công cộng của hạng UML đối tượng đăng ký được xác định trong [ebRIM]. Nếu không, sẽ xây dựng sự kiểu trừ: lỗi thuộc tính đối tượng. Bộ lọc đối tượng đăng ký quay lại một tập hợp định danh cho các trường hợp RegistryObject (đối tượng đăng ký) mà các giá trị thuộc tính đánh giá Đúng cho tính chất mệnh đề;
3. với tất cả các nhân tố XML Bộ lọc mục nhập đăng ký, thuộc tính đối số trái của bất kỳ mệnh đề đơn bao gồm sẽ xác định một thuộc tính công cộng cho hạng UML mục nhập đăng ký được xác định trong [ebRIM]. Nếu không, xây dựng sự kiểu trừ: lỗi thuộc tính mục nhập đăng ký.

Bộ lọc mục nhập đăng ký quay lại một tập hợp định danh cho các trường hợp mục nhập đăng ký mà các giá trị thuộc tính đánh giá Đúng cho tính chất mệnh đề;

4. với tất cả nhân tố XML Bộ lọc ExtrinsicObjects, thuộc tính đối số trái của bất kỳ mệnh đề đơn bao gồm sẽ xác định một thuộc tính công cộng của hạng UML ExtrinsicObjects được xác định trong [ebRIM]. Nếu không, xây dựng sự kiểu trừ: Lỗi thuộc tính ExtrinsicObjects. Bộ lọc ExtrinsicObjects quay lại một tập hợp định danh cho các trường hợp ExtrinsicObjects mà các giá trị thuộc tính đánh giá Đúng cho tính chất mệnh đề;
5. với nhân tố XML Bộ lọc gói đăng ký, thuộc tính đối số trái của bất kỳ mệnh đề đơn bao gồm sẽ xác định một thuộc tính công cộng của hạng UML gói đăng ký được xác định trong [ebRIM]. Nếu không, xây dựng sự kiểu trừ: lỗi thuộc tính gói. Bộ lọc gói thuộc tính trả lại một tập hợp định danh cho các trường hợp gói đăng ký mà các giá trị thuộc tính đánh giá đúng cho các tính chất mệnh đề;
6. với mọi nhân tố XML Bộ lọc tổ chức, thuộc tính đối số trái của bất kỳ Mệnh đề đơn nào sẽ xác định một thuộc tính công cộng cho tổ chức hoặc các hạng UML địa chỉ bưu điện được xác định trong [ebRIM]. Nếu không, xây dựng sự kiểu trừ: lỗi thuộc tính tổ chức. Bộ lọc tổ chức trả lại một tập hợp định danh cho các trường hợp Tổ chức mà các giá trị thuộc tính đánh giá Đúng cho các tính chất mệnh đề;
7. với mọi nhân tố XML Bộ lọc nút phân loại, thuộc tính đối số trái của bất kỳ mệnh đề đơn bao gồm nào sẽ xác định một thuộc tính công cộng của hạng UML Nút phân loại được xác định trong [ebRIM]. Nếu không, xây dựng sự kiểu trừ: lỗi thuộc tính nút phân loại. Nếu thuộc tính còn lại là “hướng” thuộc tính hữu hình sau đó nếu thuộc tính chữ của mệnh đề chữ là không “Bằng” sau đó xây dựng sự kiểu trừ lỗi thuộc tính hướng nút phân loại. Bộ lọc nút phân loại trả lại một tập hợp định danh cho các trường hợp nút phân loại mà các giá trị thuộc tính đánh giá Đúng cho tính chất mệnh đề;
8. với mọi nhân tố XML Bộ lọc liên kết, thuộc tính đối số trái của bất kỳ mệnh đề đơn bao gồm sẽ xác định một thuộc tính công cộng của hạng UML liên kết được xác định trong [ebRIM]. Nếu không, xây dựng sự kiểu trừ: Lỗi thuộc tính liên kết. Bộ lọc liên kết trả lại một tập hợp đồng nhất cho các trường hợp liên kết mà các giá trị thuộc tính đánh giá Đúng cho tính chất mệnh đề;

9. với mọi nhân tố XML Bộ lọc phân loại, thuộc tính đối số trái của bất kỳ mệnh đề đơn nào sẽ xác định một thuộc tính công cộng của hạng UML liên kết được xác định trong [ebRIM]. Nếu không xây dựng sự kiểu trừ: lỗi thuộc tính phân loại. Bộ lọc phân loại trả lại một tập hợp định danh cho các trường hợp phân loại mà các giá trị thuộc tính đánh giá Đúng cho tính chất mệnh đề.
10. với mọi nhân tố XML Bộ lọc giản đồ phân loại, thuộc tính đối số trái của Mệnh đề đơn bao gồm sẽ xác định một thuộc tính công cộng của lớp UML Nút phân loại được xác định trong [ebRIM]. Nếu không, xây dựng sự kiểu trừ: lỗi thuộc tính giản đồ phân loại. Bộ lọc giản đồ phân loại trả lại một tập hợp định danh cho các trường hợp giản đồ phân loại mà các giá trị thuộc tính đánh giá Đúng cho tính chất mệnh đề;
11. với mọi nhân tố XML Bộ lọc liên kết ngoài, thuộc tính đối số trái của bất kỳ mệnh đề đơn bao gồm nào sẽ xác định một thuộc tính công cộng của lớp UML liên kết ngoài được xác định trong [ebRIM]. Nếu không, xây dựng sự kiểu trừ: lỗi thuộc tính liên kết ngoài. Bộ lọc liên kết ngoài trả lại một tập hợp định danh cho các trường hợp liên kết ngoài mà các giá trị thuộc tính đánh giá đúng cho tính chất mệnh đề;
12. với mọi nhân tố XML định danh ngoài, thuộc tính đối số ngoài của bất kỳ mệnh đề đơn bao gồm nào sẽ xác định một thuộc tính công cộng cho lớp UML định danh ngoài được xác định trong [ebRIM]. Nếu không, xây dựng sự kiểu trừ: lỗi thuộc tính định danh ngoài. Bộ lọc định danh ngoài trả lại một tập hợp định danh cho các trường hợp Định danh ngoài mà các giá trị thuộc tính đánh giá Đúng cho tính chất mệnh đề;
13. với mọi nhân tố XML Bộ lọc khe cấm, thuộc tính đối số trái của bất kỳ Mệnh đề đơn bao gồm nào sẽ xác định một thuộc tính công cộng của lớp UML khe cấm được xác định trong [ebRIM]. Nếu không, xây dựng sự kiểu trừ: lỗi thuộc tính khe cấm. Bộ lọc khe cấm trả lại một tập hợp định danh cho các trường hợp định danh mà các giá trị thuộc tính đánh giá Đúng cho tính chất mệnh đề;
14. với mọi nhân tố XML Bộ lọc sự kiện có thể kiểm tra, thuộc tính đối số trái của bất kỳ Mệnh đề đơn bao gồm nào sẽ xác định một thuộc tính công cộng của lớp UML sự kiện có thể kiểm tra được xác định trong [ebRIM]. Nếu không, xây dựng sự kiểu trừ: lỗi thuộc tính sự kiện có thể kiểm tra. Bộ lọc sự kiện có thể kiểm tra trả lại một tập hợp định danh cho các trường hợp Sự kiện có thể kiểm tra mà các giá trị thuộc tính đánh giá Đúng cho tính chất mệnh đề;
15. với mọi nhân tố XML Bộ lọc người sử dụng, thuộc tính đối số trái của bất kỳ Mệnh đề đơn bao gồm nào sẽ xác định một thuộc tính công cộng của lớp UML Người sử dụng được xác định trong [ebRIM]. Nếu không, xây dựng sự kiểu trừ: lỗi thuộc tính người sử dụng. Bộ lọc người sử dụng trả lại một tập hợp định danh cho các trường hợp người sử dụng mà các giá trị thuộc tính đánh giá Đúng cho tính chất mệnh đề;
16. giá trị khe cấm là một lớp được bắt nguồn, không liên tục dựa trên lớp khe cấm từ ebRIM. Chỉ có một trường hợp Giá trị khe cấm cho mỗi “giá trị” trong danh sách “các giá trị” của một trường hợp khe cấm. Thuộc tính hữu hình của Giá trị khe cấm là “giá trị”. Đó là một đặc tính kiểu chữ. Các

trường hợp động lực của Giá trị khe cắm được bắt nguồn từ thuộc tính “các giá trị” được xác định trong ebRIM cho một trường hợp khe cắm. Với mọi nhân tố XML Bộ lọc giá trị khe cắm, thuộc tính đối số trái của bất kỳ Mệnh đề đơn bao gồm nào sẽ xác định thuộc tính “giá trị” của lớp Giá trị khe cắm vừa được xác định. Nếu không, xây dựng sự kiểu trừ: lỗi thuộc tính nhân tố khe cắm. Bộ lọc giá trị khe cắm trả lại một tập hợp các trường hợp Khe cắm mà thuộc tính “giá trị” đánh giá Đúng cho tính chất mệnh đề;

17. với mọi nhân tố XML Bộ lọc địa chỉ bưu điện, thuộc tính đối số trái của bất kỳ Mệnh đề đơn bao gồm nào sẽ xác định một thuộc tính công cộng của lớp UML Địa chỉ bưu điện được xác định trong [ebRIM]. Nếu không, xây dựng sự kiểu trừ: lỗi thuộc tính địa chỉ bưu điện.

Bộ lọc địa chỉ bưu điện trả lại một tập hợp định danh cho các trường hợp địa chỉ bưu điện mà các giá trị thuộc tính đánh giá Đúng cho tính chất mệnh đề;

18. với mọi nhân tố XML Bộ lọc số điện thoại, thuộc tính đối số trái của bất kỳ Mệnh đề đơn bao gồm nào sẽ xác định một thuộc tính công cộng của lớp UML số điện thoại được xác định trong [ebRIM]. Nếu không, xây dựng sự kiểu trừ: lỗi thuộc tính đồng nhất số điện thoại. Bộ lọc số điện thoại trả lại một tập hợp định danh cho các trường hợp Số điện thoại mà các giá trị thuộc tính đánh giá Đúng cho tính chất mệnh đề;

19. với mọi nhân tố XML Bộ lọc Quy định dịch vụ, thuộc tính đối số trái của bất kỳ Mệnh đề đơn bao gồm nào sẽ xác định một thuộc tính công cộng của lớp UML Dịch vụ được xác định trong [ebRIM]. Nếu không, xây dựng sự kiểu trừ: lỗi thuộc tính dịch vụ. Bộ lọc dịch vụ trả lại một tập hợp định danh cho các trường hợp Dịch vụ mà các giá trị thuộc tính đánh giá Đúng cho tính chất mệnh đề;

20. với mọi nhân tố XML Bộ lọc Quy định dịch vụ, thuộc tính đối số trái của bất kỳ Mệnh đề đơn bao gồm nào sẽ xác định một thuộc tính công cộng của lớp UML Quy định dịch vụ được xác định trong [ebRIM]. Nếu không, xây dựng sự kiểu trừ: lỗi thuộc tính Quy định dịch vụ. Bộ lọc Quy định dịch vụ trả lại một tập hợp định danh cho các trường hợp Quy định dịch vụ mà các giá trị thuộc tính đánh giá Đúng cho tính chất mệnh đề;

21. với mọi nhân tố XML Bộ lọc liên kết quy định kỹ thuật, thuộc tính đối số trái của bất kỳ Mệnh đề đơn bao gồm nào sẽ xác định một thuộc tính công cộng của lớp UML liên kết quy định kỹ thuật được xác định trong [ebRIM]. Nếu không, xây dựng sự kiểu trừ: lỗi thuộc tính liên kết quy định kỹ thuật. Bộ lọc liên kết quy định kỹ thuật trả lại một tập hợp định danh cho các trường hợp liên kết quy định kỹ thuật mà các giá trị thuộc tính đánh giá Đúng cho tính chất mệnh đề;

22. với mọi nhân tố XML Bộ lọc kiểu chữ được khu biệt, thuộc tính đối số trái của bất kỳ mệnh đề đơn bao gồm nào sẽ xác định một thuộc tính công cộng của lớp UML kiểu chữ khu biệt được xác định trong [ebRIM]. Nếu không, xây dựng sự kiểu trừ: lỗi Thuộc tính type chữ khu biệt. Bộ lọc kiểu chữ khu biệt trả lại một tập hợp định danh cho các trường hợp Kiểu chữ khu biệt mà các giá trị thuộc tính đánh giá Đúng cho tính chất mệnh đề.

8.2.14 Cách biểu diễn bắt buộc trong mệnh đề XML

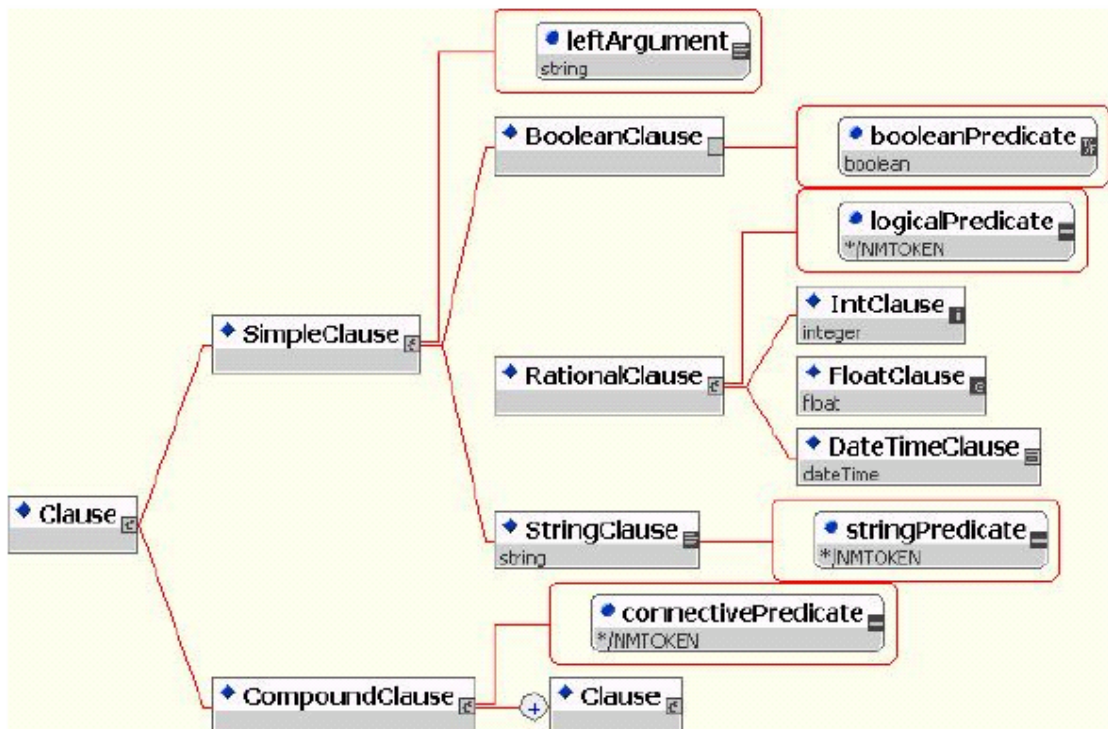
Mục đích:

Truy vấn bộ lọc XML đơn sử dụng một cấu trúc XML hình thức được dựa trên các Mệnh đề tính chất.

Các mệnh đề tính chất được sử dụng để chính thức xác định cơ chế bắt buộc, và được chuyển đổi một cách dễ dàng như các mệnh đề trong quy định này.

Biểu đồ khái niệm

Dưới đây là một biểu đồ khái niệm phác thảo cấu trúc mệnh đề.



Hình 27 - Cấu trúc mệnh đề

Các quy tắc ngữ nghĩa

Các tính chất và Đối số được kết hợp trong một mẫu “Đối số trái_ Tính chất- Đối số phải” để thiết lập một mệnh đề. Có hai kiểu mệnh đề: Mệnh đề đơn và Mệnh đề hỗn hợp.

Mệnh đề đơn

Một Mệnh đề đơn luôn luôn xác định Đối số trái bằng kiểu chữ, đôi khi được quy cho là chủ đề của một mệnh đề. Bản thân mệnh đề đơn là chưa đầy đủ (tóm tắt) và phải được mở rộng. Mệnh đề đơn được mở rộng để hỗ trợ Mệnh đề Đúng- Sai (Boolean), Mệnh đề chữ và Mệnh đề hữu tỷ (tóm tắt).

Mệnh đề Đúng- Sai xác định tuyệt đối tính chất như là “ngang bằng với”, với đối số phải là một Đúng- Sai. Mệnh đề chữ xác định tính chất như là một thuộc tính đếm của những thao tác đếm chữ và một mệnh đề phải là số liệu chữ của nhân tố. Hỗ trợ số hữu tỷ được cung cấp thông qua một Mệnh đề hữu

tỷ chung cung cấp một bảng liệt kê các hoạt động so sánh số hữu tỷ phù hợp, mà sự mở rộng thêm đến Mệnh đề Int và Mệnh đề động, chỉ với chữ ký phù hợp cho mệnh đề phải.

Mệnh đề hỗn hợp

Một mệnh đề hỗn hợp bao gồm hai hoặc nhiều hơn (đơn hoặc hỗn hợp) và một tính chất liên kết. Điều này giúp mệnh đề hỗn hợp tự động được hình thành.

Định nghĩa

```

<element name = "Clause">
  <annotation>
    <documentation xml:lang = "en">
      The following lines define the XML syntax for Clause.
    </documentation>
  </annotation>
  <complexType>
    <choice>
      <element ref = "tns:SimpleClause"/>
      <element ref = "tns:CompoundClause"/>
    </choice>
  </complexType>
</element>
<element name = "SimpleClause">
  <complexType>
    <choice>
      <element ref = "tns:BooleanClause"/>
      <element ref = "tns:RationalClause"/>
      <element ref = "tns:StringClause"/>
    </choice>
    <attribute name = "leftArgument" use = "required" type =
      "string"/>
  </complexType>
</element>
<element name = "CompoundClause">
  <complexType>
    <sequence>
      <element ref = "tns:Clause" maxOccurs = "unbounded"/>
    </sequence>
    <attribute name = "connectivePredicate" use = "required">
      <simpleType>
        <restriction base = "NMTOKEN">
          <enumeration value = "And"/>
          <enumeration value = "Or"/>
        </restriction>
      </simpleType>
    </attribute>
  </complexType>
</element>
<element name = "BooleanClause">
  <complexType>
    <attribute name = "booleanPredicate" use = "required" type =
      "boolean"/>
  </complexType>
</element>
<element name = "RationalClause">
  <complexType>
    <choice>
      <element ref = "tns:IntClause"/>
      <element ref = "tns:FloatClause"/>
      <element ref = "tns:DateTimeClause"/>
    </choice>
  </complexType>

```

```

<attribute name = "logicalPredicate" use = "required">
<simpleType>
<restriction base = "NMTOKEN">
<enumeration value = "LE"/>
<enumeration value = "LT"/>
<enumeration value = "GE"/>
<enumeration value = "GT"/>
<enumeration value = "EQ"/>
<enumeration value = "NE"/>
</restriction>
</simpleType>
</attribute>
</complexType>
</element>
<element name = "IntClause" type = "integer"/>
<element name = "FloatClause" type = "float"/>
<element name = "DateTimeClause" type = "dateTime"/>
<element name = "StringClause">
<complexType>
<simpleContent>
<extension base = "string">
<attribute name = "stringPredicate" use = "required">
<simpleType>
<restriction base = "NMTOKEN">
<enumeration value = "Contains"/>
<enumeration value = "-Contains"/>
<enumeration value = "StartsWith"/>
<enumeration value = "-StartsWith"/>
<enumeration value = "Equal"/>
<enumeration value = "-Equal"/>
<enumeration value = "EndsWith"/>
<enumeration value = "-EndsWith"/>
</restriction>
</simpleType>
</attribute>
</extension>
</simpleContent>
</complexType>
</element>

```

Ví dụ

Mệnh đề Đúng- Sai đơn: "Smoker" = True

```

<Clause>
<SimpleClause leftArgument="Smoker">
<BooleanClause booleanPredicate="True"/>
</SimpleClause>
</Clause>

```

Mệnh đề chữ đơn: "Smoker" contains "mo"

```

<Clause>
<SimpleClause leftArgument = "Smoker">
<StringClause stringPredicate = "Contains">mo</StringClause>
</SimpleClause>
</Clause>

```

Mệnh đề Int đơn: "Age" >= 7

```

<Clause>
<SimpleClause leftArgument="Age">
<RationalClause logicalPredicate="GE">
<IntClause>7</IntClause>
</RationalClause>
</SimpleClause>

```

</Clause>

Mệnh đề đơn: "Size" = 4.3

```
<Clause>
<SimpleClause leftArgument="Size">
<RationalClause logicalPredicate="Equal">
<FloatClause>4.3</FloatClause>
</RationalClause>
</SimpleClause>
</Clause>
```

Hỗn hợp với hai mệnh đề đơn (("Smoker" = False)AND("Age" =< 45))

```
<Clause>
<CompoundClause connectivePredicate="And">
<Clause>
<SimpleClause leftArgument="Smoker">
<BooleanClause booleanPredicate="False"/>
</SimpleClause>
</Clause>
<Clause>
<SimpleClause leftArgument="Age">
<RationalClause logicalPredicate="LE">
<IntClause>45</IntClause>
</RationalClause>
</SimpleClause>
</Clause>
</CompoundClause>
</Clause>
```

Hỗn hợp với một mệnh đề đơn và một mệnh đề hỗn hợp

(("Smoker" = False)And(("Age" =< 45)Or("American"=True)))

```
<Clause>
<CompoundClause connectivePredicate="And">
<Clause>
<SimpleClause leftArgument="Smoker">
<BooleanClause booleanPredicate="False"/>
</SimpleClause>
</Clause>
<Clause>
<CompoundClause connectivePredicate="Or">
<Clause>
<SimpleClause leftArgument="Age">
<RationalClause logicalPredicate="LE">
<IntClause>45</IntClause>
</RationalClause>
</SimpleClause>
</Clause>
<Clause>
<SimpleClause leftArgument="American">
<BooleanClause booleanPredicate="True"/>
</SimpleClause>
</Clause>
</CompoundClause>
</Clause>
</CompoundClause>
<Clause>
```

8.3 Hỗ trợ truy vấn SQL

Đăng ký có thể hỗ trợ tùy chọn một SQL dựa trên khả năng truy vấn được thiết kế cho các khách hàng truy vấn mà đề nghị thêm nhiều khả năng truy vấn cao cấp hơn. Nhân tố truy vấn SQL tùy chọn trong

Đề nghị truy vấn đặc biệt cho phép một khách hàng đề nghị các truy vấn SQL phức hợp sử dụng một ngôn ngữ truy vấn tuyên bố.

Cú pháp cho truy vấn SQL của sổ đăng ký được xác định bởi việc sử dụng được cách điệu hoá của một tập hợp con thích hợp với sự trình bày “Lựa chọn” của SQL mức mục nhập được xác định bởi ISO/IEC 9075:1992, Ngôn ngữ dữ liệu SQL [SQL], phần mở rộng bao gồm <sql invoked routines> (cũng được xem như là một quy trình bảo quản) được chỉ rõ trong IEC 9075-4 [SQL-PSM] và thủ tục đã xác định trước được xác định trong mẫu ở Phụ lục D.3. Cú pháp của ngôn ngữ truy vấn Đăng nhập được xác định bởi ngữ pháp BNF trong D.1.

Ghi nhớ rằng việc sử dụng một tập hợp con của cú pháp SQL cho truy vấn SQL không bao hàm một việc đề nghị sử dụng cơ sở dữ liệu liên quan trong việc thi hành Đăng ký.

8.3.1 Quy định cú pháp truy vấn SQL đến [ebRIM]

Các truy vấn SQL được xác định dựa trên cú pháp truy vấn trong phụ lục D.1. và giản đồ quan hệ cố định trong phụ lục D.3. Giản đồ quan hệ là một gói thuật toán tới [ebRIM] được mô tả trong các phần dưới đây.

8.3.1.1 Quy định lớp

Một tập hợp con của các tên lớp được xác định trong bản đồ [ebRIM] đến các tên bảng mà có thể được truy vấn bởi một truy vấn SQL. Phụ lục D.3. xác định các tên của các lớp ebRIM mà có thể được truy vấn bởi một truy vấn SQL.

Thuật toán này được sử dụng để xác định gói của các lớp [ebRIM] đến các định nghĩa bảng trong phụ lục D.3. như sau:

- các lớp mà có các trường hợp cụ thể được sắp xếp trong các bảng mối quan hệ. Thêm nữa, các lớp thực thể (ví dụ: địa chỉ bưu điện và số điện thoại) cũng được sắp xếp vào các bảng mối quan hệ;
- các lớp trung gian trong hệ thống cấp bậc, đây là RegistryObject (đối tượng đăng ký) và Mục nhập đăng ký, bản đồ để xem xét các mối quan hệ;
- tên của các bảng mối quan hệ và việc xem xét giống như là tên các lớp [ebRIM] tương ứng. Tuy nhiên, gói tên là trường hợp nhạy cảm;
- mỗi lớp [ebRIM] mà sắp xếp một bảng trong Phụ lục D.3. bao gồm các định nghĩa cột trong Phụ lục D.3. ở đó các định nghĩa cột được dựa trên một tập hợp con của các thuộc tính được xác định cho lớp này trong [ebRIM]. Các thuộc tính mà sắp xếp thành cột bao gồm các thuộc tính kế thừa cho lớp [ebRIM]. Những giải thích trong phụ lục D.3. cho biết các lớp trước đã đóng góp cho các định nghĩa cột.

Một truy vấn SQL tương phản với một bảng không được xác định trong phụ lục D.3. có thể xây dựng một lỗi điều kiện: Sự kiểu trừ truy vấn vô giá trị.

Các phần dưới đây mô tả thuật toán cho các thuộc tính đồ thị của [ebRIM] cho các định nghĩa cột SQL.

8.3.1.2 Quy định thuộc tính gốc

Các thuộc tính được xác định bởi [ebRIM] đây là các kiểu gốc (ví dụ kiểu chữ) được sử dụng cùng theo cách các tên cột trong SQL. Mặt khác các tên thuộc tính chính xác được xác định trong các định nghĩa lớp trong [ebRIM]. Cần nhớ rằng trong khi các tên là các trường hợp hỗn tạp, SQL-92 là trường hợp nhạy cảm. Vì vậy các truy vấn có giá trị là bao gồm các tên thuộc tính mà không phù hợp chính xác trường hợp được xác định trong [ebRIM].

8.3.1.3 Quy định thuộc tính tham khảo

Một vài trong số thuộc tính lớp [ebRIM] là kiểu UUID và là một tham khảo cho trường hợp của một lớp được xác định bởi [ebRIM]. Ví dụ, thuộc tính Cảnh sát kiểm soát của lớp Đối tượng đăng nhập trả lại một tham khảo cho một trường hợp của một đối tượng Cảnh sát kiểm soát truy nhập.

Trong những trường hợp như vậy, việc tham khảo chỉ dẫn đến thuộc tính id cho đối tượng được tham khảo. Tên của cột kết quả giống như tên thuộc tính trong [ebRIM] là được xác định trong 8.3.12. Kiểu số liệu cho cột là VARCHAR(64) được xác định trong Phụ lục D.3.

Khi một giá trị thuộc tính tham khảo nắm giữ một tham khảo không tồn tại trong gói SQL và có thể được kiểm tra với <null specification> (“IS [NOT] NULL” syntax) được xác định bởi [SQL].

Quy định thuộc tính tham khảo là một trường hợp đặc biệt của bản đồ thuộc tính gốc.

8.3.1.4 Quy định thuộc tính phức hợp

Một vài trong số giao diện [ebRIM] xác định các thuộc tính mà không phải là kiểu gốc. Thay cho đấy, chúng là của một kiểu phức hợp được xác định bởi lớp thực thể trong [ebRIM]. Các ví dụ bao gồm các thuộc tính của kiểu Số điện thoại, Liên lạc, Tên người ... trong lớp Tổ chức và Lớp người sử dụng.

Giản đồ truy vấn SQL không vạch ra các thuộc tính phức hợp như cột trong bảng cho các lớp để các thuộc tính được xác định. Thay vì đấy, các thuộc tính phức hợp được sắp xếp trong cột trong bảng cho lớp miền mà biểu thị kiểu số liệu cho thuộc tính phức hợp (ví dụ: số điện thoại). Một cột liên kết hàng trong bảng miền đến hàng trong bảng gốc (ví dụ: Người sử dụng).

Một cột thêm vào được đặt tên “tên_thuộc tính” định danh tên thuộc tính trong lớp gốc, trong trường hợp có nhiều thuộc tính phức tạp với kiểu thuộc tính phức hợp giống như vậy.

Bản đồ này cũng cho phép dễ dàng cho các thuộc tính mà là tập hợp của một kiểu phức tạp. Ví dụ, một người sử dụng có thể có một tập hợp các số điện thoại. Điều này sắp xếp thành các hàng phức tạp trong bảng số điện thoại (một cho mỗi một số điện thoại) ở mỗi một hàng có một định danh gốc và một tên_thuộc tính.

8.3.1.5 Quy định các phương pháp trả lại tập hợp

Một vài lớp [ebRIM] xác định các phương pháp để thêm vào các thuộc tính, khi đó các phương pháp này trả lại các tập hợp tham khảo cho các trường hợp các lớp được xác định bởi [ebRIM]. Ví dụ,

phương pháp Gói nhận được của lớp Đối tượng được quản lý trả lại một tập hợp các tham khảo cho các trường hợp Gói mà đối tượng là một thành viên trong đó.

Các phương pháp trả lại tập hợp như thế này trong các lớp [ebRIM] đã được sắp xếp cho các quy trình bảo lưu trong Phụ lục D.3. như vậy các quy trình bảo lưu này trả lại một tập hợp các giá trị thuộc tính id. Giá trị được trả lại này của các quy trình bảo lưu này có thể được xem xét như là kết quả của một bảng truy vấn nhỏ trong SQL.

Các quy trình được bảo lưu này có thể được sử dụng như là mặt bàn tay phải của một mệnh đề SQLIN để kiểm tra tư cách thành viên của một đối tượng trong các tập hợp tham khảo như vậy.

8.3.2 Quy định ngữ nghĩa dựa trên cú pháp truy vấn

Phần này xác định sự thúc ép đơn với cú pháp truy vấn mà không thể được diễn giải trong BNF cho cú pháp truy vấn. Những sự thúc ép này phải được ứng dụng trong phân tích ngữ nghĩa của truy vấn.

1. các tên lớp và tên thuộc tính phải được xử lý theo một cách không nhạy cảm;
2. cú pháp này được sử dụng cho sự dẫn chứng quy trình lưu trữ phải phù hợp với cú pháp của một sự dẫn chứng quy trình SQL được xác định bởi ISO/IEC 9075-4 [SQL/PSM];
3. với phiên bản này của quy định kỹ thuật, danh sách cột lựa chọn SQL chính xác bao gồm một cột, và phải luôn luôn là t.id, nơi mà một bảng tham khảo trong mệnh đề FROM;
4. các hoạt động ghép nối phải được giới hạn để đơn giản các ghép nối chỉ có liên quan các cột mà có danh mục được xác định với giản đồ SQL quy chuẩn. Sự thúc ép này để ngăn ngừa các truy vấn mà được tính toán là quá đắt.

8.3.3 Kết quả truy vấn SQL

Kết quả của một truy vấn SQL để giải tập hợp các đối tượng trong phạm vi đăng ký. Nó không bao giờ giải các thuộc tính bộ phận. Các đối tượng có liên quan đến tập hợp kết quả có thể được trả lại như một Ref đối tượng, RegistryObject (đối tượng đăng ký), Mục nhập đăng ký hoặc lớp ebRIM là phụ thuộc vào thông số Lựa chọn phản hồi được xác định bởi khách hàng với Đề nghị truy vấn đặc biệt.

Toàn bộ tập hợp kết quả được trả lại nha một Kết quả truy vấn SQL mà đã được xác định bởi AdHocQueryResponse (phản hồi truy vấn đặc biệt) trong Phần 8.1.

8.3.4 Truy vấn được dựa trên siêu dữ liệu riêng lẻ

Mẫu đơn giản nhất của một truy vấn SQL được dựa trên các thuộc tính metadata (siêu dữ liệu) được xác định cho mọi lớp [ebRIM]. Phần này đưa ra một vài ví dụ về siêu dữ liệu đơn dựa trên các truy vấn.

Ví dụ như, để có tập hợp các ExtrinsicObject (đối tượng ngoại lai) mà tên của chúng có từ "Acme" và chúng có một phiên bản lớn hơn 1.3, truy vấn dưới đây phải được xem xét:

```
SELECT eo.id from ExtrinsicObject eo, Name nm where nm.value LIKE '%Acme%' AND
    eo.id = nm.parent AND
    eo.majorVersion >= 1 AND
    (eo.majorVersion >= 2 OR eo.minorVersion > 3);
```

Nhớ rằng cú pháp truy vấn cho phép sự nối kết các tính chất đơn giản hơn thành các truy vấn phức tạp hơn như đã được giới thiệu trong ví dụ trên.

8.3.5 Truy vấn RegistryObject (đối tượng đăng ký)

Giản đồ cho truy vấn SQL xác định một cái nhìn đặc biệt được gọi là RegistryObject (đối tượng đăng ký) mà cho phép thực hiện một truy vấn nhiều dạng tương phản với tất cả các trường hợp Đối tượng đăng ký mà không tính đến kiểu cụ thể thực hoặc tên bảng.

Ví dụ dưới đây là tương tự với những gì trong phần 8.3.4 trừ việc nó được ứng dụng tương phản với tất cả các trường hợp RegistryObject (đối tượng đăng ký) hơn là chỉ với các trường hợp ExtrinsicObjects. Tập hợp kết quả sẽ bao gồm id cho tất cả các trường hợp RegistryObject (đối tượng đăng ký) thoả mãn mà tên của chúng có từ “Acme” và sự mô tả của chúng có chứa từ “bicycle”.

```
SELECT ro.id from RegistryObject ro, Name nm, Description d where nm.value LIKE '%Acme%' AND
    d.value LIKE '%bicycle%' AND
    ro.id = nm.parent AND ro.id = d.parent;
```

8.3.6 Truy vấn RegistryEntry (mục nhập đăng ký)

Giản đồ cho truy vấn SQL xác định một cái nhìn đặc biệt được gọi là Mục nhập đăng ký mà cho phép thực hiện một truy vấn nhiều dạng tương phản với tất cả các trường hợp Mục nhập đăng ký mà không tính đến kiểu cụ thể thực tế của chúng hoặc tên bảng.

Ví dụ dưới đây là giống với Phần 8.3.4 ngoại trừ rằng nó được ứng dụng tương phản với tất cả các trường hợp Mục nhập đăng ký hơn là chỉ với các trường hợp ExtrinsicObjects. Tập hợp kết quả sẽ bao gồm id cho tất cả các trường hợp Mục nhập đăng ký mà tên của chúng có từ “Acme:” và chúng có phiên bản lớn hơn 1.3.

```
SELECT re.id from RegistryEntry re, Name nm where nm.value LIKE '%Acme%' AND
    re.id = nm.parent AND
    re.majorVersion >= 1 AND
    (re.majorVersion >= 2 OR re.minorVersion > 3);
```

8.3.7 truy vấn Classification (phân loại)

Phần này mô tả các phân loại khác nhau có liên quan đến các truy vấn mà cần được hỗ trợ.

8.3.7.1 Nhận dạng nút phân loại

như tất cả các đối tượng trong {ebRIM}, nút phân loại được nhận dạng bởi ID của chúng. Tuy nhiên chúng cũng có thể được nhận dạng như là một thuộc tính đường dẫn mà định rõ một biểu thức (sự diễn tả)XPATh[XPT] từ một nút phân loại gốc tới nút phân loại đặc trưng trong tư liệu XML mà đại diện cho cây nút phân loại bao gồm nút phân loại đã nói.

8.3.7.2 Tiếp nhận giản đồ phân loại

Để có được sự quy tụ của giản đồ phân loại thuộc tính truy vấn dưới đây phải được hỗ trợ :

```
SELECT scheme.id FROM classificationScheme scheme;
```

Truy vấn ở trên quay lại tất cả những hệ thống phân loại. Chú ý rằng Truy vấn ở trên cũng có thể được định rõ các thuộc tính phụ (e.g.tên, sự mô tả ...)nếu muốn.

8.3.7.3 Tiếp nhận con của nút phân loại lý thuyết

Để tiếp nhận con của nút phân loại lý thuyết dựa trên ID của nút đó thì kiểu truy vấn sau phải được hỗ trợ :

```
SELECT cn.id FROM ClassificationNode cn WHERE parent = <id>
```

Truy vấn ở trên quay lại tất cả những nút phân loại mà có nút được định rõ bởi <id> như thuộc tính gốc của chúng.

8.3.7.4 Tiếp nhận các đối tượng được phân loại bởi một nút phân loại

Để tiếp nhận sự quy tụ của những ExtrinsicObjects được phân loại bởi nút phân loại lý thuyết kiểu Truy vấn dưới đây phải được hỗ trợ :

```
SELECT id FROM ExtrinsicObject
WHERE
    id IN (SELECT classifiedObject FROM Classification
           WHERE .
           ClassificationNode IN (SELECT id FROM
           ClassificationNode
                               WHERE
           path = '/Geography/Asia/Japan'))
AND
    id IN (SELECT classifiedObject FROM Classification
           WHERE .
           ClassificationNode IN (SELECT id FROM
           ClassificationNode
                               WHERE
           path = '/Industry/Automotive'))
```

Truy vấn trên nhận sự quy tụ của những ExtrinsicObjects được phân loại bởi ngành công nghiệp xe hơi và địa lý Nhật Bản. Chú ý rằng về mặt ngữ nghĩa định nghĩa cho GetClassifiedObjectRequest (truy vấn tiến nhận đối tượng phân loại) Truy vấn cũng sẽ bao gồm bất cứ đối tượng nào được phân loại bởi con cháu của nút phân loại lý thuyết.

8.3.7.5 Tiếp nhận sự phân loại của một đối tượng

Để nhận sự quy tụ của sự phân loại mà có thể phân loại một đối tượng lý thuyết kiểu truy vấn dưới đây phải được hỗ trợ :

```
SELECT id FROM Classification c
       WHERE c.classifiedObject = <id>;
```

8.3.8 Truy vấn Association (liên kết)

Phần này mô tả mối quan hệ của nhiều kiểu truy vấn Association (liên kết) khác nhau mà chúng phải được hỗ trợ.

8.3.8.1 Tiếp nhận tất cả sự liên kết với đối tượng lý thuyết như nguồn của nó

Để nhận sự quy tụ của các liên kết mà đối tượng lý thuyết như nguồn của nó truy vấn sau phải được hỗ trợ :

```
SELECT id FROM Association WHERE sourceObject = <id>
```

8.3.8.2 Tiếp nhận tất cả sự liên kết với đối tượng lý thuyết như đích của nó

Để nhận sự quy tụ của các liên kết mà đối tượng lý thuyết như đích của nó Truy vấn sau phải được hỗ trợ :

TCVN ISO/TS 15000-4 : 2007

```
SELECT id FROM Association WHERE targetObject = <id>
```

8.3.8.3 Tiếp nhận đối tượng liên kết dựa trên các thuộc tính liên kết

Để nhận sự quy tụ của những liên kết mà có thuộc tính liên kết ảo Truy vấn sau phải được hỗ trợ:

Lựa chọn những liên kết có chứa tên lý thuyết.

```
SELECT id FROM Association WHERE name = <name>
```

Lựa chọn những liên kết có kiểu liên kết lý thuyết, nơi kiểu liên kết này là một chuỗi có chứa trường tên tương ứng được mô tả trong [ebRIM].

```
SELECT id FROM Association WHERE  
    associationType = <associationType>
```

8.3.8.4 Truy vấn Association (liên kết) phức tạp

những hình thức đa dạng của các truy vấn Association (liên kết) có thể được kết hợp vào các thuộc tính phức tạp, truy vấn dưới đây lựa chọn các liên kết có một đối tượng nguồn, đối tượng đích rõ ràng và kiểu liên kết:

```
SELECT id FROM Association WHERE  
    sourceObject = <id1> AND  
    targetObject = <id2> AND  
    associationType = <associationType>;
```

8.3.9 Truy vấn gói

Tìm tất cả các gói mà một đối tượng đăng ký lý thuyết thuộc vào, truy vấn dưới đây phải được định rõ:

```
SELECT id FROM Package WHERE id IN (RegistryObject_packages (<id>));
```

8.3.9.1 Truy vấn gói phức tạp

Truy vấn dưới đây nhận tất cả gói mà một đối tượng lý thuyết thuộc về, không chống lại và tên có chứa "RosettaNet."

```
SELECT id FROM package p, Name n WHERE  
    p . id IN (RegistryObject_packages (<id>) AND  
    nm . value LIKE '%RosettaNet%' AND nm.parent = p.id AND  
    p . status < > 'Deprecated'
```

8.3.10 Truy vấn ExternalLink (Liên kết ngoài)

Tìm tất cả các ExternalLink (liên kết ngoài) mà một ExtrinsicObject (đối tượng ngoại lai) cụ thể được liên kết, quy định truy vấn sau:

```
SELECT id From ExternalLink WHERE id IN (RegistryObject_externalLinks(<id>))
```

Tìm tất cả các ExtrinsicObject (đối tượng ngoại lai) được liên kết bởi ExternalLink (liên kết ngoài) cụ thể, quy định truy vấn sau:

```
SELECT id From ExtrinsicObject WHERE id IN (RegistryObject_linkedObjects(<id>))
```

8.3.10.1 Truy vấn Association (liên kết) ngoài phức tạp

Truy vấn dưới đây nhận tất cả những liên kết ngoài mà một ExtrinsicObjects (đối tượng ngoại lai) đặc trưng trực thuộc, có chứa từ 'legal' trong sự mô tả của chúng và có một URL cho URI ngoài của chúng.

```
SELECT id FROM ExternalLink WHERE
    id IN (RegistryObject_externalLinks (<id>)) AND
    description LIKE '%legal%' AND
    externalURI LIKE '%http://%'
```

8.3.11 Truy vấn vết kiểm tra

Để nhận tập hợp đầy đủ các đối tượng AuditableEvent (sự kiện có thể kiểm tra) cho một ManagedObject cụ thể, truy vấn dưới đây phải được xác định :

```
SELECT id FROM AuditableEvent WHERE registryObject = <id>
```

8.4 Truy cập nội dung

người sử dụng gọi nội dung thông qua bộ đăng ký bằng cách gửi “yêu cầu tiếp nhận nội dung” tới Truy vấn chủ, “yêu cầu tiếp nhận nội dung” xác định một danh sách các đối tượng tham khảo cho các đối tượng cần được truy cập. Truy vấn chủ quay lại nội dung đặc trưng bằng cách gửi thông báo “trả lời tiếp nhận nội dung” giao diện đăng ký sử dụng của máy người sử dụng. Nếu không gặp phải lỗi gì, thông báo “trả lời tiếp nhận nội dung” bao gồm nội dung xác định như dung lượng phụ trong phạm vi của thông báo. Ngoài ra dung lượng trả lời tiếp nhận nội dung, có một dung lượng phụ cho mỗi nội dung được yêu cầu. Nếu không có lỗi gì xảy ra, dung lượng đăng ký trả lời bao gồm một lỗi và không có phần mở rộng có chứa dung lượng xác định.

8.4.1 Nhận dạng dung lượng nội dung

từ thông báo 'trả lời tiếp nhận nội dung' có thể bao gồm nhiều gói tin lưu trữ như dung lượng phụ, điều này rất cần thiết để có một phương pháp xác định mỗi dung lượng trong thông báo. Để dễ dàng cho sự nhận dạng này, trình đăng ký phải thực hiện những việc sau:

- dùng ID của ExtrinsicObjects như giá trị của vùng chứa ID chủ cho vai trò ẩn có chứa gói tin lưu trữ tương ứng với ExtrinsicObjects;
- trong trường hợp sự vận chuyển ebMS, dùng ID cho mỗi trường hợp RegistryObject (đối tượng đăng ký) mà mô tả gói tin lưu trữ trong phần tử tham khảo cho đối tượng đó trong phần tử hiển nhiên của ebXML header.

8.4.2 Cấu trúc của thông báo "trả lời tiếp nhận nội dung"

đoạn thông điệp dưới đây minh họa cho cấu trúc của thông báo "trả lời tiếp nhận nội dung" mà đang lặp lại một sự quy tụ của CPPs như kết quả của một yêu cầu tiếp nhận nội dung xác định Ids cho đối tượng yêu cầu.

```
Content-type: multipart / related; boundary = "boundary" ; type = "text/xml";
--Boundary
content-ID:
content-Type : text/xml
<?xml version = "1.0" encoding = "UTF-8"?>
```

TCVN ISO/TS 15000-4 : 2007

```
<SOAP - ENV: Envelope xmlns: SOAP-ENV = 'http://schemas .
xmlsoap.org/soap/envelope/'
    xmlns : eb= 'http://www.oasis-open.org/committees/ebxml-msg/schema/draft-msg-
header-03.xsd'> <SOAP-ENV: Header>
...ebMS header goes here if using ebMS
<SOAP-ENV :Header>
<SOAP-ENV : Body>
...ebMS header goes here if using ebMS
<SOAP-ENV :Header>
<SOAP-ENV : Body>
...ebMS manifest goes here if using ebMS
<?xml version = "1.0" encoding = "UTF-8"?>
<GetContentRequest>
    <ObjectReList>
        <ObjectRef id = "d8163dfb-f45 a-4798-81d9-88aca29c24ff".../>
        <ObjectRef id = "212c3a78-1368-45d7-acc9-a935197e1e4f".../>
    </ObjectReList>
</GetContentRequest>
</SOAP-ENV :Body>
</SOAP-ENV : Envelope>
--Boundary
content-ID: d8163dfb-f45 a-4798-81d9-88aca29c24ff"
conten-Type : text/xml
<?xml version = "1.0" encoding = "UTF-8"?>
<CPP>
.....
</CPP>
--Boundary--
content-ID: 212c3a78-1368-45d7-acc9-a935197e1e4f"
conten-Type : text/xml
<CPP>
.....
</CPP>
--Bondary-
```

9 An ninh đăng ký

Mục này mô tả những đặc tính bảo mật của trình đăng ký ebXML. Nó được thừa nhận rằng người đọc quen thuộc với mức độ bảo mật liên quan trong kiểu thông tin đăng ký cũng như được mô tả trong[ebRIM]. Những thuật ngữ chuyên ngành về bảo mật có thể tham khảo từ RFC 2828.

9.1 Các vấn đề liên quan an ninh

Trong tiêu chuẩn này, chúng tôi sử dụng nguyên vẹn dữ liệu và nguồn của nó (thuật ngữ 1 trong phần phụ lục F.1) sử dụng cách tiếp cận tối thiểu trực tiếp tới vùng điều khiển truy cập có ích như trong thuật ngữ 2 của phần phụ lục F1. Về cơ bản, “bất cứ một thực thể được biết đến nào (một tổ chức chính thức) đều có thể ban hành nội dung và bất cứ ai cũng có thể xem nội dung được ban hành đó.” Kiểu thông tin đăng ký được thiết kế để cho phép giải quyết vấn đề bảo mật cao hơn trong những phiên bản tương lai của tiêu chuẩn này.

9.2 Tính toàn vẹn của nội dung đăng ký

Phải thừa nhận rằng hầu hết những sổ đăng ký không có khả năng đánh giá tính xác thực của những nội dung được đệ trình cho họ. “Cơ chế được mô tả trong phần này có thể được dùng để bảo đảm rằng bất cứ sự mua chuộc hay giả mạo, làm thay đổi nội dung được đăng ký bởi một tổ chức đăng ký đệ trình đều có thể được nhận biết.” Hơn nữa hỗ trợ cho cơ quan có thẩm quyền tránh được sự mập

mờ không rõ ràng trong những nội dung đăng ký. Khách hàng đăng ký (Registry Client) phải đóng dấu vào nội dung muốn đăng ký trước khi đệ trình cho cơ quan có chức năng đăng ký, nếu không nội dung ấy sẽ bị từ chối.

Chú ý rằng trong những thảo luận ở phần này chúng ta thừa nhận rằng tổ chức muốn đăng ký cũng có thể là tổ chức có thẩm quyền đăng ký. Phiên bản sau của tiêu chuẩn này có thể cung cấp thêm những ví dụ và tình huống nơi một cơ quan muốn đăng ký và cơ quan có thẩm quyền đăng ký là khác nhau.

9.2.1 Chữ ký vùng mang thông tin của thông điệp

Sự nguyên vẹn của nội dung đăng ký đòi hỏi tất cả những nội dung đệ trình phải được đóng dấu (hoặc có chữ ký) của khách hàng muốn đăng ký. Chữ ký trên nội dung đệ trình bảo đảm rằng:

- bất kỳ sự can thiệp nào làm thay đổi nội dung cần phải được nhận biết;
- tính trung thực của nội dung đăng ký có thể xác định chắc chắn bởi sự liên quan của nó với cụ thể tổ chức muốn đăng ký.

Phần này định rõ những yêu cầu cho sự phát sinh, phổ biến và tính hiệu lực của chữ ký cho vùng mang thông tin. chữ ký cho vùng mang thông tin được Quy định với the vùng mang thông tin.

A chữ ký cho vùng mang thông tin được trình bày bằng vùng mang thông tin. Do đó, những yêu cầu được áp dụng bất kể việc khách hàng đăng ký (Registry Client) và Cơ quan cấp đăng ký có giao thiệp qua lại với các tài liệu đính kèm hoặc dịch vụ thông báo ebXML [ebMS] hay không. Hiện nay, dịch vụ thông báo eb XML không chỉ việc phát sinh, ph ỏ bi ến và tính hiệu lực các chữ ký cho vùng mang thông tin. Chi tiết về chữ ký cho vùng mang thông tin ở bên trái phía trên của đơn (như đăng ký). Vì thế, những yêu cầu về chữ ký cho vùng mang thông tin làm tăng lên sự xác định [ebMS].

Trường hợp sử dụng:

Trường hợp sử dụng này minh họa cho việc sử dụng chữ ký tiêu đề và vùng mang thông tin (chúng ta sẽ bàn luận về chữ ký cho tiêu đề sau):

- RC1 (Khách hàng đăng ký (Registry Client) 1) ký nội dung (thường là 1 chữ ký cho vùng mang thông tin) và công bố nội dung cùng chữ ký cho vùng mang thông tin với sổ đăng ký;
- RC2 (Khách hàng đăng ký (Registry Client) 2) lấy nội dung của RC1 từ sổ đăng ký;
- RC2 muốn xác minh rằng RC1 đã công bố nội dung. Để làm điều này, khi RC2 lấy nội dung đó, câu trả lời từ Cơ quan thẩm quyền đăng ký bao gồm những nội dung sau:
 - vùng mang thông tin bao hàm nội dung đã được công bố bởi RC1;
 - chữ ký cho vùng mang thông tin của RC1 (đại diện bởi một ds: phần tử chữ ký) dựa trên nội dung đã công bố của RC1;
 - điểm mấu chốt công khai cho việc phê chuẩn chữ ký cho vùng mang thông tin của RC1 ở ds: phần tử chữ ký (sử dụng phần tử thông tin chủ chốt như đã được quy định trong [XMLDSIG]. Do đó RC2 có thể giành được điểm mấu chốt công khai cho chữ ký (ví dụ lấy một chứng chỉ bao hàm điểm mấu chốt công khai cho RC1);

TCVN ISO/TS 15000-4 : 2007

- 1 ds: Phần tử chữ ký bao hàm chữ ký tiêu đề. Lưu ý rằng Cơ quan thẩm quyền đăng ký (không phải RC1) phát sinh chữ ký này.

9.2.2 Yêu cầu chữ ký cho vùng mang thông tin

9.2.2.1 Yêu cầu về sự phổ biến của chữ ký cho vùng mang thông tin

Một chữ ký cho vùng mang thông tin được đại diện bởi một ds:Signature. Một chữ ký cho vùng mang thông tin phải được phổ biến với kiểu vùng mang thông tin đặc thù ở đây. Việc phổ biến này cho rằng vùng mang thông tin phải luôn được phê duyệt.

- payload và chữ ký của nó phải được gửi kèm trong một thông điệp MIME liên quan với một kiểu nội dung liên quan/ thuộc vào;
- phần đầu phải bao hàm chữ ký XML như đã được chỉ rõ trong phần 9.2.2.2 “Yêu cầu về sự phổ biến chữ ký cho vùng mang thông tin”;
- phần hai cho đến phần thứ n phải là nội dung.

Việc giới thiệu chữ ký vùng mang thông tin với 1 trọng tải như sau:

9.2.2.2 Yêu cầu tạo chữ ký cho vùng mang thông tin

ds:Signature [XMLDSIG] đối với một chữ ký cho vùng mang thông tin phải được phát sinh rõ ràng trong phần này. Lưu ý : tham khảo thêm tên ‘ds’ trong

- ds:SignatureMethod phải được đưa ra. [XMLDSIG] yêu cầu thuật toán không xác định sử dụng thuộc tính của thuật toán ấy. [XMLDSIG] cho phép nhiều hơn 1 thuộc tính thuật toán và khách hàng có thể sử dụng bất kỳ cái nào trong số này. Tuy nhiên, việc ký sử dụng thuộc tính thuật toán sau sẽ cho phép khả năng thực hiện liên hợp dễ dàng, vì XMLDSIG đòi hỏi sự thực hiện thuật toán này.

Phần tử ds:SignatureMethod phải bao hàm một phần tử ds:CanonicalizationMethod (phương pháp chuẩn). Thuật toán chuẩn (được chỉ rõ trong [XMLDSIG]) phải được hỗ trợ

- một phần tử ds:Reference để tham khảo mỗi một vùng mang thông tin cần được ký phải được tạo ra. ds:Reference:
 - phải xác định vùng mang thông tin được ký sử dụng thuộc tính URL của ds:Reference;
 - phải bao hàm <ds:DigestMethod> như đã định trong [XMLDSIG]). Một khách hàng phải hỗ trợ cho thuật toán phân loại sau ;
 - phải bao hàm một <ds:DigestValue> đã được vi tính hoá như trong [XMLDSIG].

ds:SignedValue phải được phát ra như trong [XMLDSIG].

ds: Phần tử KeyInfo (thông tin chính) sẽ được đưa ra. Tuy nhiên, khi đưa ra, trường ds:KeyInfo (thông tin chính) là một chủ đề đối với những yêu cầu trong phần 9.4, “Phần tử Distribution chính và phần tử KeyInfo (thông tin chính)”.

9.2.2.3 Tính hiệu lực của message chữ ký cho vùng mang thông tin

ds: phần tử chữ ký phải được phê chuẩn bởi số đăng ký định rõ trong [XMLDSIG].

9.2.2.4 Ví dụ chữ ký cho vùng mang thông tin

Ví dụ dưới đây cho thấy hình thức của chữ ký cho vùng mang thông tin.

9.3 Xác thực

Số đăng ký phải có khả năng xác nhận nhân dạng của người uỷ nhiệm liên quan với những yêu cầu của khách hàng. Nhân dạng của người uỷ nhiệm đó có thể được xác định thông qua việc kiểm tra chữ ký tiêu đề của thông điệp với giấy chứng nhận của người uỷ nhiệm. Giấy chứng nhận này có thể ở trong bản thân thông điệp hoặc được cung cấp cho số đăng ký thông qua các phương tiện không được định rõ trong phần này. Nếu không được cung cấp trong thông điệp, phần này không chỉ rõ số đăng ký tương quan như thế nào giữa một thông điệp với một giấy chứng nhận. Xác thực cũng được đòi hỏi để xác nhận “những đặc quyền” mà một người uỷ nhiệm có được (“sự cấp phép”) để có các đối tượng cụ thể trong số đăng ký.

Số đăng ký phải thực hiện xác thực trên nền tảng mỗi một thông điệp. Từ quan điểm bảo mật, tất cả các thông điệp đều độc lập và không có ý niệm chung về phiên họp các thông điệp đa phương diện hay các buổi thảo luận. Buổi họp hỗ trợ có thể được bổ sung như là một dấu hiệu khả quan trong những phiên bản tương lai của lĩnh vực này.

Cần lưu ý rằng chữ ký tiêu đề của thông điệp chỉ có thể bảo đảm tính nguyên vẹn của dữ liệu và nó có thể được sử dụng cho việc chứng thực biết rằng thật dễ tổn thương khi chiếu lại các kiểu công kích. Sự hỗ trợ thực tế cho việc chứng thực đòi hỏi thời gian để đóng dấu hoặc nonce (những dãy số tuần hoàn để nhận dạng mỗi kiểu thông điệp) được ký duyệt.

9.3.1 Chữ ký tiêu đề của thông điệp

Thông điệp headers được ký để cung cấp dữ liệu toàn vẹn trong khi thông điệp đang được truyền đi. Lưu ý rằng chữ ký trong khuôn khổ thông điệp header cũng ký những bộ phận loại của vùng mang thông tin

Yêu cầu về chữ ký cho tiêu đề

Thông điệp headers có thể được ký duyệt và tham khám như là một chữ ký tiêu đề. Phần này chỉ ra những yêu cầu cho việc phát sinh, phổ biến và tính hiệu lực của một chữ ký cho tiêu đề. Những yêu cầu này áp dụng khi khác hàng đăng ký và cơ quan chủ quản đăng ký giao dịch với nhau sử dụng vanilla SOAP với các tài liệu đính kèm. Khi ebXML được sử dụng để giao dịch, rồi đến người điều khiển các thông điệp đó (chẳng hạn [ebMS]) xác định sự phát sinh, phổ biến và tính hiệu lực của các chữ ký XML trong SOAP header. Do đó, những yêu cầu đối với chữ ký cho tiêu đề không áp dụng khi ebXML MS được sử dụng cho giao dịch. Tuy nhiên, những yêu cầu phát sinh chữ ký vùng mang thông tin (được chỉ rõ ở phần khác trong tiêu chuẩn này) được áp dụng bất kể là vanilla SOAP với các tài liệu đính kèm hay ebXML MS được dùng để giao dịch.

9.3.1.1 Yêu cầu phổ biến

Một chữ ký cho tiêu đề được đại diện bởi một ds: phần tử chữ ký. Ds: Phần tử chữ ký được sinh ra phải được phổ biến trong một phần tử <SOAP-EVN: Header>. Sự phổ biến của ds: phần tử chữ ký trong phạm vi tiêu đề của SOAP được chỉ ra dưới đây.

9.3.1.2 Yêu cầu về việc phát sinh chữ ký tiêu đề

ds:Signature [XMLDSIG] dành cho một chữ ký cho tiêu đề phải được sinh ra như đã chỉ rõ trong phần này. Một ds:Signature bao gồm :

- ds : Thông tin được ký;
- ds: Giá trị chữ ký;
- ds: Thông tin chủ chốt.

Ds: Phần tử thông tin được ký phải được sinh ra như sau:

1. ds: Phương pháp ký phải được thực hiện. [XMLDSIG] đòi hỏi thuật toán được xác định, sử dụng thuộc tính thuật toán. Trong khi [XMLDSIG] cho phép nhiều hơn một thuộc tính thuật toán, một khách hàng phải có khả năng ký mà chỉ sử dụng một thuộc tính thuật toán dưới đây: . Thuật toán này được chọn vì tất cả các thành phần của XMLDSIG tuân theo [XMLDSIG] sự định rõ hỗ trợ nó;
2. ds: phần tử phương pháp chữ ký phải bao gồm một ds: phần tử phương pháp hợp chuẩn. Thuật toán hợp chuẩn dưới đây (đã chỉ rõ trong [XMLDSIG]) phải được hỗ trợ;
3. 1 ds: phần tử tham chiếu bao gồm <SOAP-EVN: phong bì> trong sự tính toán chữ ký. Điều này ghi dấu một ds toàn vẹn: Phần tử tham chiếu và:
 - phải bao gồm ds sau: Chuyển đổi.

Điều này đảm bảo rằng chữ ký (được gắn vào trong phần tử <SOAP-EVN: header>) không nằm trong sự tính toán chữ ký

- phải xác định rằng phần tử <SOAP-EVN: phong bì> sử dụng thuộc tính URI của ds: phần tử tham chiếu (Thuộc tính URI chính thức trong sự xác định [XMLDSIG]). Thuộc tính URI phải là ""
- phải bao hàm <ds:DigestMethod> được cụ thể hoá trong [XMLDSIG]. Khách hàng phải hỗ trợ thuật toán phân loại sau;
- phải bao hàm một <ds:DigestValue> đã được vi tính hoá cụ thể trong [XMLDSIG].

ds:SignedValue phải được phát sinh cụ thể như trong [XMLDSIG].

ds: phần tử KeyInfo (thông tin chính) được đưa ra. Khi đưa ra, nó nhắm vào những yêu cầu được ghi trong phần 9.4, "Phần tử phân phối chính và thông tin chủ chốt".

9.3.1.3 Yêu cầu về tính hiệu lực

ds: phần tử chữ ký đối với ebXML thông điệp header phải được phê chuẩn bởi người nhận như đã định rõ bởi [XMLDSIG].

9.3.1.4 Ví dụ chữ ký tiêu đề

Ví dụ dưới đây cho thấy hình thức của một chữ ký tiêu đề:

9.4 Phân tử KeyInfo (thông tin chính) và Distribution (phân phối) chính

Để phê chuẩn một chữ ký, người nhận chữ ký cần khoá công bố tương ứng với khoá công bố của người ký. Người tham gia có thể sử dụng trường KeyInfo (thông tin chính) của ds:Signature hay phân phối các khoá công bố ngoài các nhóm. Trong phần này, trường hợp khi khoá công bố được gửi trong trường KeyInfo (thông tin chính). Những trường hợp sử dụng sau cần được xét đến:

- cơ quan có thẩm quyền đăng ký cần khoá công bố của khách hàng đăng ký (Registry Client) để kiểm tra tính hợp lệ chữ ký;
- khách hàng đăng ký (Registry Client) cần khoá công bố của cơ quan có thẩm quyền đăng ký để kiểm tra tính hợp lệ chữ ký của sổ đăng ký;
- khách hàng đăng ký (Registry Client) RC1 cần khoá công bố của khách hàng đăng ký (Registry Client) (RC2) để kiểm tra tính hợp lệ nội dung đã ký bởi RC1;
- [XMLDSIG] cung cấp một ds: phân tử KeyInfo (thông tin chính) là một phần tử khả quan cụ thể trong [XMLDSIG]. Lĩnh vực này cùng với các thủ tục bên ngoài trong phần này được sử dụng để chắc chắn chuyển khoá công bố cho một người nhận. ds: KeyInfo có thể được sử dụng để chuyển thông tin như các khoá, giấy chứng nhận, tên, v.v... Việc sử dụng của trường KeyInfo (thông tin chính) là gửi giấy chứng nhận X509 và sau đó rút ra khoá công bố từ giấy chứng nhận. Do đó, trường KeyInfo (thông tin chính) phải bao gồm giấy chứng nhận X509 cụ thể trong [XMLDSIG] nếu trường KeyInfo (thông tin chính) được đưa ra.

Những giả định sau cũng được tạo ra:

1. giấy chứng nhận liên quan đến cả cơ quan có thẩm quyền đăng ký và khách hàng đăng ký (Registry Client);
2. khách hàng đăng ký (Registry Client) đăng ký giấy chứng nhận của mình với cơ quan có thẩm quyền đăng ký. Cơ chế được dùng cho vấn đề này không được làm rõ ở đây;
3. khách hàng đăng ký (Registry Client) có được chứng nhận của cơ quan có thẩm quyền và lưu giữ nó trong kho khoá địa phương của mình. Cơ chế này không được làm rõ ở đây.

Hai ví dụ về việc sử dụng trường KeyInfo (thông tin chính) trong phụ lục F.8.

9.5 Tính bảo mật

9.5.1 Tính bảo mật thông điệp điện tín

Người ta gợi ý nhưng không bắt buộc rằng vùng mang thông tin thông điệp trao đổi giữa khách hàng và sổ đăng ký phải được viết lại thành mật mã qua quá trình truyền tin. Phần này không chỉ ra làm cách thức việc mã hoá vùng mang thông tin được thực hiện.

9.5.2 Tính bảo mật của nội dung đăng ký

Trong phiên bản hiện hành của tiêu chuẩn này, không có điều khoản nào cho tính bảo mật của nội dung đăng ký.

Tất cả nội dung được đệ trình cho sổ đăng ký có thể bị phát hiện và đọc bởi bất kỳ khách hàng nào. Điều này hàm ý rằng sổ đăng ký và khách hàng phải có một thoả thuận trước liên quan đến thuật toán mã hoá, những thoả thuận về việc trao đổi khoá, v.v... Dịch vụ này không được nhấn mạnh trong tiêu chuẩn này.

9.6 Sự cấp phép

Sổ đăng ký phải đưa ra một cơ chế cấp phép dựa trên mẫu thông tin được xác định trong [ebRIM]. Trong phiên bản của tiêu chuẩn này, cơ chế cấp phép dựa trên chính sách kiểm soát truy cập mặc định được xác định cho tập hợp rõ ràng các vai trò của người sử dụng đăng ký. Phiên bản tương lai của tài liệu sẽ cho phép những chính sách kiểm soát truy cập theo thói quen được xác định bởi tổ chức đệ trình. Việc cấp phép sẽ được áp dụng trong một tập hợp đặc quyền cụ thể. Đặc quyền là khả năng tiến hành một hoạt động đặc thù.

9.6.1 Hoạt động

Hoạt động chu kỳ

- submitObjects (đối tượng đệ trình)
- updateObjects (đối tượng cập nhật)
- addSlots (khe bổ sung)
- removeSlots (khe gỡ bỏ)
- approveObjects (đối tượng phê chuẩn)
- deprecateObjects (đối tượng không tán thành)
- removeObjects (đối tượng gỡ bỏ)

Hoạt động đọc

Các phương pháp getXXX khác nhau trong dịch vụ QueryManager (quản lý truy vấn)

9.7 Kiểm soát truy cập

Sổ đăng ký phải tạo một đối tượng kiểm soát truy cập mặc định để mặc định cho phép người sử dụng đăng ký dựa trên vai trò được phân công của họ. Bảng dưới đây định nghĩa “sự cấp phép” được cấp bởi sổ đăng ký với những vai trò cơ bản khác nhau cho người sử dụng dựa trên phương pháp kiểm soát truy cập mặc định. Lưu ý rằng chúng ta có thuật ngữ “contentOwner” (người sở hữu nội dung) như là một vai trò. Vai trò này vạch ra tổ chức đệ trình đăng ký trong phiên bản hiện hành của tiêu chuẩn.

Bảng 11 - Chính sách kiểm soát truy cập mặc định

Vai trò	Được phép
Người sở hữu nội dung	Truy cập tới tất cả các thuộc tính sở hữu bởi họ
Người quản lý đăng ký	Truy cập tới tất cả thuộc tính trên tất cả các đối tượng đăng ký
Khách đăng ký	Truy cập tất cả nhưng thuộc tính chỉ đọc(get XXX) trên tất cả đối tượng đăng ký(truy cập chỉ đọc tới tất cả nội dung)

Cơ quan tiếp nhận đăng ký phải thi hành phương pháp kiểm soát truy cập mặc định và liên kết nó với tất cả các đối tượng trong hồ sơ đăng ký. Danh sách dưới đây tổng hợp phương pháp kiểm soát truy cập mặc định dựa trên vai trò:

- bất cứ ai cũng có thể công bố nội dung nhưng cần phải đăng ký là người sử dụng;
- bất cứ ai cũng có thể truy cập vào nội dung mà không đòi hỏi xác nhận;
- người chủ sở hữu có thể truy cập vào tất cả các thuộc tính của đối tượng đăng ký được tạo bởi họ;
- người quản lý đăng ký có thể truy cập tới tất cả thuộc tính của các đối tượng đăng ký;
- khách hàng không đăng ký người sử dụng có thể truy cập tất cả các thuộc tính chỉ đọc;
- các thời điểm xem xét nội dung, cơ quan tiếp nhận đăng ký phải chỉ định vai trò chủ sở hữu tới tổ chức đệ trình để thẩm định tính xác thực trong thông báo đăng ký. Trong phiên bản hiện hành của tiêu chuẩn này, tổ chức xin đăng ký sẽ là DN như được nhận dạng bởi chứng chỉ;
- những khách hàng nhỏ, cần thiết đăng ký không dùng chứng chỉ, sổ đăng ký phải chỉ định vai trò khách đăng ký.

Phụ lục A

Cấu trúc dịch vụ web

A.1 Quy định lý thuyết dịch vụ đăng ký

Định nghĩa quy phạm của dịch vụ đăng ký trừu tượng trong WSDL được định nghĩa tại địa chỉ website sau :

<http://www.oasis-open.org/committees/regrep/documents/2.0/services/Registry.wsd>

A.2 Liên kết dịch vụ đăng ký SOAP

Định nghĩa quy phạm của sự kết hợp dịch vụ đăng ký thực thể với SOAP trong WSDL được định nghĩa ở website dưới đây:

<http://www.oasis-open.org/committees/regrep/documents/2.0/services/SOAPBinding.wsd>

Phụ lục B

Định nghĩa giản đồ đăng ký ebXML

B.1 Giản đồ RIM

Định nghĩa quy phạm của giản đồ XML để ánh xạ [ebRIM] phân loại tới XML có thể được tìm thấy ở trang web.

<http://www.oasis-open.org/committees/regrep/documents/2.0/schema/rim.xsd>

B.2 Giản đồ truy vấn

Định nghĩa quy phạm giản đồ XML cho cú pháp truy vấn XML của giao diện dịch vụ đăng ký có thể tìm thấy ở trang web dưới đây:

<http://www.oasis-open.org/committees/regrep/documents/2.0/schema/query.xsd>

B.3 Giản đồ giao diện dịch vụ đăng ký

Định nghĩa quy phạm giản đồ XML mà xác định rõ những yêu cầu và trả lời hỗ trợ bởi giao diện dịch vụ đăng ký trong tiêu chuẩn này có thể được tìm thấy ở trang web dưới đây

<http://www.oasis-open.org/committees/regrep/documents/2.0/schema/rs.xsd>

B.4 Những ví dụ của tài liệu mẫu

Một số lượng ngày càng lớn của những tài liệu mẫu XML không có tính quy phạm mà làm cho thích ứng với định nghĩa giản đồ quy phạm được mô tả sớm hơn cũng có thể được tìm thấy ở trang web sau:

<http://cvs.sourceforge.net/cgi-bin/viewcvs.cgi/ebxmlrr/ebxmlrr-spec/misc/samples/>

Phụ lục C

Sự diễn giải của biểu đồ UML

Phần này mô tả bằng ngôn ngữ trừu tượng quy ước thường dùng để định nghĩa sự mô tả quy trình thương mại ebXML trong UML

C.1 Biểu đồ phân loại UML

Một giản đồ phân loại (Classification scheme) UML được dùng để mô tả giao diện dịch vụ đòi hỏi để thực thi dịch vụ đăng ký ebXML và khách hàng. Bản đồ phân loại UML bao gồm :

1. một tập hợp của những giao diện UML, nơi mỗi giao diện đại diện một giao diện dịch vụ cho một dịch vụ đăng ký;
2. bảng mô tả sự sắp xếp trên mỗi giao diện nơi mỗi sự sắp xếp đại diện một hành động (như được định nghĩa bởi [ebCPP]) trong khi đó giao diện dịch vụ đại diện giao diện UML;
3. mỗi sự sắp xếp trong vòng giao diện UML xác định một hay nhiều tham số nơi mà kiểu của mỗi chủ đề sắp xếp đại diện cho kiểu thông báo ebXML mà được thay đổi như một phần của hành động phù hợp với sự sắp xếp. Một số lượng nhiều đối số dẫn đến số lượng nhiều tài liệu kéo theo trong vòng nội dung của thông báo ebXML tương ứng.

C.2 Biểu đồ chuỗi UML

Một biểu đồ chuỗi UML được dùng để xác định giao thức thông mai đại diện cho những sự tương tác giữa giao diện UML cho một quá trình thương mại đăng ký ebXML cụ thể. Một biểu đồ chuỗi UML cung cấp những thông tin cần thiết để xác định chuỗi thông báo, sự liên kết từ đòi hỏi tới trả lời cũng như từ đòi hỏi tới lỗi trả lời.

Mỗi biểu đồ chuỗi trình bày chuỗi của một giao thức hội thoại xác định.như một phương pháp gọi từ người truy vấn tới người trả lời.mỗi cách thức dẫn chứng có thể đồng bộ hay không đồng bộ dựa trên dấu hiệu UML dùng trên đầu mũi tên cho chỗ liên kết. Một nửa mũi tên đại diện giao tiếp không đồng bộ, toàn bộ mũi tên đại diện cho giao tiếp đồng bộ.

Mỗi phương pháp dẫn chứng có thể được theo sau bởi một phương pháp trả lời dẫn chứng từ người trả lời tới người truy vấn để chỉ ra tên trả lời cho truy vấn trước. Những trả lời sai sót có thể xảy ra thì được chỉ ra bởi phương pháp việ dẫn trả lời có điều kiện từ người trả lời tới người truy vấn.(xem ví dụ minh họa ở hình 7).

Phụ lục D

Truy vấn SQL

D.1 Đặc điểm kỹ thuật của cú pháp truy vấn SQL

Chương này xác định những quy tắc nhằm định nghĩa cú pháp truy vấn SQL như là một tập hợp con của SQL-92. Ngôn ngữ định kèm trong dấu ngoặc đơn , nghiêng được định nghĩa trong [SQL] hoặc trong [SQL/PSM]. Cú pháp truy vấn SQL thích ứng với <sự phân loại truy vấn>, sự hạn chế được nhận dạng bên dưới.

1. Một <**select list**> có thể bao gồm tối thiểu một <**select sublist**>;
2. Trong một <**select list**> phải là một cột đơn của kiểu dữ liệu là UUID, từ bảng trong <**from clause**>;
3. Một <**derived column**> có thể không có một <**as clause**>;
4. <**table expression**> không chứa mệnh đề không bắt buộc <**group by clause**> và <**having clause**>;
5. Một <**table reference**> có thể chỉ gồm <**table name**> và <**correlation name**>;
6. Một <**table reference**> không có AS không bắt buộc giữa <**table name**> và <**correlation name**>
7. chỉ có thể có một <**table reference**> trong <**from clause**>;
8. giới hạn sử dụng của truy vấn phụ được cho phép bởi cú pháp sau : <**in predicate**> cho phép mặt tay phải của <**in predicate**> bị giới hạn với một sự phân loại giới hạn <**query specification**> như phần định nghĩa trên;
9. một <**search condition**> trong vòng <**where clause**> có thể không bao gồm một <**query expression**>;
10. Những điểm nối đơn giản chỉ được phép nếu chúng dựa trên cột chỉ dẫn nằm trong giản đồ liên hệ;
11. cấu trúc truy vấn SQL cho phép sự sử dụng <**sql invoked routines**> của sự dẫn chứng từ [SQL/PSM] như là RHS của <**in predicate**>.

D.2 BNF không có tính quy phạm cho ngữ pháp cú pháp truy vấn.

BNF dưới đây minh họa ngữ pháp cho cú pháp truy vấn đăng ký. Ở đây nó được cung cấp như một sự giúp đỡ cho các đối tượng thực thi. Từ đó BNF này không dựa trên [SQL] nó được cung cấp như cú pháp không có tính quy phạm. Với những quy tắc cú pháp có tính quy phạm, xem ở phần phụ lục D.1.

```

/*****
* The Registry Query (Subset of SQL-92) grammar starts here
*****/
RegistryQuery = SQLSelect [";"]
SQLSelect = "SELECT" ["DISTINCT"] SQLSelectCols "FROM" SQLTableList [ SQLWhere ]
SQLSelectCols = ID

```

TCVN ISO/TS 15000-4 : 2007

```
SQLTableList = SQLTableRef
SQLTableRef = ID
SQLWhere = "WHERE" SQLOrExpr
SQLOrExpr = SQLAndExpr ( "OR" SQLAndExpr)*
SQLAndExpr = SQLNotExpr ("AND" SQLNotExpr)*
SQLNotExpr = [ "NOT" ] SQLCompareExpr
SQLCompareExpr =
(SQLColRef "IS") SQLIsClause
| SQLSumExpr [ SQLCompareExprRight ]
SQLCompareExprRight =
SQLLikeClause
| SQLInClause
| SQLCompareOp SQLSumExpr
SQLCompareOp =
"="
| "<>"
| ">"
| ">="
| "<"
| "<="
SQLInClause = [ "NOT" ] "IN" "(" SQLLValueList ")"
SQLLValueList = SQLLValueElement ( "," SQLLValueElement )*
SQLLValueElement = "NULL" | SQLSelect
SQLIsClause = SQLColRef "IS" [ "NOT" ] "NULL"
SQLLikeClause = [ "NOT" ] "LIKE" SQLPattern
SQLPattern = STRING_LITERAL
SQLLiteral =
STRING_LITERAL
| INTEGER_LITERAL
| FLOATING_POINT_LITERAL
SQLColRef = SQLLvalue
SQLLvalue = SQLLvalueTerm
SQLLvalueTerm = ID ( "." ID )*
SQLSumExpr = SQLProductExpr (( "+" | "-" ) SQLProductExpr )*
SQLProductExpr = SQLUnaryExpr (( "*" | "/" ) SQLUnaryExpr )*
SQLUnaryExpr = [ ( "+" | "-" ) ] SQLTerm
SQLTerm = "(" SQLOrExpr ")"
| SQLColRef
| SQLLiteral
INTEGER_LITERAL = ([ "0"-"9" ])+
FLOATING_POINT_LITERAL =
([ "0"-"9" ])+ "." ([ "0"-"9" ])+ ( EXPONENT)?
| "." ([ "0"-"9" ])+ ( EXPONENT)?
| ([ "0"-"9" ])+ EXPONENT
| ([ "0"-"9" ])+ ( EXPONENT)?
EXPONENT = [ "e", "E" ] ([ "+", "-" ])? ([ "0"-"9" ])+
STRING_LITERAL: "'" (~[ "'" ])* ( "'" (~[ "'" ])* )* "'"
ID = ( <LETTER> )+ ( "_" | "$" | "#" | <DIGIT> | <LETTER> )*
LETTER = [ "A"-"Z", "a"-"z" ]
DIGIT = [ "0"-"9" ]
```

D.3 Giản đồ quan hệ truy vấn SQL

Định nghĩa quy phạm của giản đồ quan hệ truy vấn SQL có thể tìm thấy ở trang web dưới đây.

<http://www.oasis-open.org/committees/regrep/documents/2.0/sql/database.sql>

Thủ tục lưu trữ mà phải được hỗ trợ bởi đặc tính truy vấn SQL được định nghĩa ở trang web dưới đây.

<http://www.oasis-open.org/committees/regrep/documents/2.0/sql/storedProcedures.sql>

Phụ lục E

Truy vấn đặc biệt dựa trên nội dung tham khảo

Khả năng truy vấn đăng ký SQL hỗ trợ khả năng tìm kiếm nội dung không những dựa trên siêu dữ liệu được ghi mục lục trong nội dung mà còn bao gồm ngay trong chính nội dung đó. Ví dụ một khách hàng có thể đệ trình một truy vấn mà tìm kiếm tất cả những bản liệt kê cộng tác một phần mà định nghĩa một đối tượng có tên "seller" bên trong một phần tử tên đối tượng trong chính tài liệu CPP của nó. Hiện nay khả năng truy vấn dựa trên nội dung thì bị hạn chế tới nội dung XML

E.1 Phân loại tự động nội dung XML

Truy vấn dựa trên nội dung thì được hỗ trợ trực tiếp thông qua cơ chế phân loại sẵn có hỗ trợ bởi sổ đăng ký.

Một tổ chức đệ trình đăng ký có thể định nghĩa bảng chú dẫn logic trên bất kì giản đồ XML hoặc DTD khi nó được đệ trình. Một ví dụ bảng chỉ dẫn logic xác định một liên kết giữa một thuộc tính xác định hay một nút cơ bản trong một cây tài liệu XML và một nút phân loại trong giản đồ phân loại thuộc phạm vi sổ đăng ký.

Sổ đăng ký tận dụng bảng chỉ dẫn này để tự động phân loại tài liệu được làm mẫu của giản đồ vào thời điểm mà tài liệu mẫu được đệ trình. Từ đó tiêu chuẩn này được phân loại theo dữ liệu có chứa trong phạm vi chính nó.

Như vậy nội dung được phân loại tự động sau đó có thể được khám phá bởi khách hàng dùng cơ chế khám phá dựa trên sự phân loại sẵn có của sổ đăng ký và những tiện ích truy vấn của hệ thống QueryManager.

CHÚ Ý: Sự tiếp cận này dựa trên các khái niệm tương tự cách mà cơ sở dữ liệu hỗ trợ sự phục hồi được chỉ dẫn. DBAs xác định bảng chỉ dẫn trên bảng trong giản đồ. Khi dữ liệu được cho vào bảng nó sẽ tự động nhận bảng chỉ dẫn.

E.2 Định nghĩa bảng chỉ dẫn

Phần này mô tả bảng chỉ dẫn logic được định nghĩa như thế nào trong nguyên lý đối tượng đăng ký định nghĩa trong giản đồ đăng ký. Giản đồ đăng ký hoàn chỉnh thì có sẵn thông qua liên kết trong phụ lục B

Một nguyên lý đối tượng đăng ký cho một giản đồ hoặc DTD có thể xác định một tập hợp của các bảng chỉ dẫn phân loại trong danh sách bảng chỉ dẫn phân loại nguyên lý không bắt buộc. Danh sách bảng chỉ dẫn phân loại được bỏ qua nếu nội dung đang được đệ trình không thuộc kiểu đối tượng của giản đồ.

Nguyên lý bảng chỉ dẫn phân loại thừa hưởng thuộc tính của đối tượng đăng ký cơ bản trong [ebRIM]. Sau đó nó định nghĩa thuộc tính chuyên biệt như sau :

TCVN ISO/TS 15000-4 : 2007

1. nút phân loại: Thuộc tính này tham khảo một nút phân loại đặc trưng bằng ID của nó ;
2. nhận dạng nội dung: Thuộc tính này xác định phần tử dữ liệu đặc trưng trong phạm vi tài liệu mẫu của giản đồ dùng một sự diễn tả XPATH như định nghĩa bởi [XPT].

E.3 Ví dụ định nghĩa bảng chỉ dẫn.

Để định nghĩa một bảng chỉ dẫn mà tự động phân loại một CPP dựa trên các đối tượng xác định trong phạm vi nguyên lý tên đối tượng của nó. Bảng chỉ dẫn sau đây phải được xác định trên giản đồ CPP hoặc DTD:

```
<ClassificationIndex
  classificationNode=' id-for-role-classification-scheme'
  contentIdentifier='/Role//RoleName'
/>
```

E.4 Định nghĩa mục đích XML

```
<!--
A ClassificationIndexList is specified on ExtrinsicObjects of objectType
'Schema' to define an automatic Classification of instance objects of the
schema using the specified classificationNode as parent and a
ClassificationNode created or selected by the object content as selected
by the contentIdentifier
-->
<!ELEMENT ClassificationIndex EMPTY>
<!ATTLIST ClassificationIndex
  %ObjectAttributes;
  classificationNode IDREF #REQUIRED
  contentIdentifier CDATA #REQUIRED
>
<!-- ClassificationIndexList contains new ClassificationIndexes -->
<!ELEMENT ClassificationIndexList (ClassificationIndex)*>
```

E.5 Ví dụ của sự phân loại tự động

Thừa nhận rằng một CPP được đệ trình mà xác định hai vai trò : “seller và “buyer” . Khi CPP được đệ trình nó sẽ tự động được phân loại bởi hai nút phân loại có tên “seller” và “buyer” cả hai con của nút phân loại định rõ trong thuộc tính nút phân loại của bảng chỉ dẫn phân loại. Nếu cả hai nút phân loại “buyer” và “seller” đều không tồn tại trước, hệ thống QueryManager sẽ tự động tạo hai nút phân loại này.

Phụ lục F

Hướng dẫn thực thi an ninh

Phần này cung cấp một bản đề nghị chi tiết cho việc làm thế nào để quá trình bảo mật được thực thi trong hệ thống đăng ký. Nó có nghĩa rằng để minh họa chứ không phải để đưa ra chỉ thị. Hệ thống đăng ký có thể chọn để có sự thực thi khác nhau với điều kiện là chúng hỗ trợ các đối tượng bảo mật đặc trưng và những quy tắc cấp phép trong tiêu chuẩn này.

F.1 Các vấn đề an ninh

Những rủi ro trong an ninh xuất phát rộng rãi từ những mối quan tâm sau: sau sự mô tả của những mối quan tâm và giải pháp tiềm năng, chúng ta xác định những lo lắng mà chúng ta vạch ra trong tiêu chuẩn hiện hành này.

1. Nội dung của dữ liệu đăng ký có đáng tin cậy?

- a) làm thế nào bảo đảm được rằng “những gì trong hệ thống đăng ký là những gì đúng là phải đặt ở đó” bởi một tổ chức đệ trình đăng ký? mối quan tâm này có thể được chú tâm tới bằng cách bảo đảm rằng người công bố được xác thực dùng chữ ký tay (nguồn nguyên vẹn) thông báo không bị làm sai lệch trong suốt sự lưu chuyển dùng chữ ký tay (dữ liệu nguyên vẹn) và dữ liệu không bị thay đổi bởi những mục đích không được phép dựa trên chính sách kiểm soát truy cập.(sự cấp phép);
- b) làm thế nào bảo vệ dữ liệu trong khi lưu chuyển?

Sự giao tiếp đáng tin cậy có hai thành phần: Dữ liệu nguyên vẹn (mục 1a) và dữ liệu tin cậy có thể được lưu trữ bằng cách mã hóa trong lưu chuyển. Làm thế nào để chống lại một sự tấn công lặp lại;
- c) nội dung có được cập nhật?kiểu cũng như bất cứ thời gian nào của quá trình niêm phong nào khi đã làm chắc chắn sẽ bảo đảm nội dung sau cùng tới nội dung sau cùng;
- d) làm thế nào bảo đảm rằng chỉ những tổ chức có trách nhiệm không gian đối mới đưa nội dung để đăng ký? Bảo đảm nguồn nguyên vẹn (như trong phần 1a);
- e) làm thế nào bảo đảm rằng người công bố đưa nội dung để đăng ký chỉ ở nơi được phép? (hệ thống nguyên vẹn);
- f) chuyện gì sẽ xảy ra nếu người công bố phủ nhận việc nội dung chắc chắn bị thay đổi sau khi họ vi phạm, để ngăn chặn điều này (không từ chối) sự kiểm tra theo dấu vết có thể phải duy trì để những bằng chứng không bị tiêu hủy;
- g) chuyện gì sẽ xảy ra nếu người đọc phủ nhận những thông tin nhận được từ sổ đăng ký?.

2. Làm thế nào để cung cấp sự truy cập có lựa chọn vào nội dung đăng ký?câu trả lời chung là dùng một chính sách kiểm soát truy cập – ứng dụng trực tiếp phần a; b ; c;

TCVN ISO/TS 15000-4 : 2007

- a) làm thế nào hạn chế tổ chức đệ trình đăng ký truy cập tới nội dung mà chỉ người đọc đăng ký xác định;
 - b) làm thế nào một tổ chức đệ trình đăng ký cho phép vài “partner” (thành viên những người công bố) để sửa đổi nội dung;
 - c) làm thế nào cung cấp sự truy cập có lựa chọn cho những “partner” dữ liệu đăng ký thông thường?;
 - d) làm thế nào ngăn cản sự truy cập ngẫu nhiên tới dữ liệu bởi những người dùng không được phép? Đặc biệt với những hw/sw không thích hợp của những thành phần đăng ký an ninh. Giải pháp cho vấn đề này là có một hệ thống đáng nguyên vẹn;
 - e) sự tin cậy dữ liệu của đối tượng đăng ký.
3. Làm thế nào chúng ta tạo ra chính sách “ai có thể nhìn thấy những gì” bản thân nó hiện diện để hạn chế những phần khác, thậm chí không bao gồm người quản lý hệ thống (tự tin duy trì chính sách kiểm soát truy cập). Bằng cách bảo đảm chắc chắn có một chính sách kiểm soát truy cập cho sự truy cập chính sách của chúng;
4. Làm thế nào để lưu chuyển tin cậy? Giải pháp chung là sử dụng sự khẳng định độ tin cậy (như được làm việc trong SAML). Hiện nay, sổ đăng ký không hỗ trợ ý niệm về phiên làm việc này. Do đó, một số công ty này không được đề cập đến trong tài liệu hiện hành:
- a) làm cách nào để lưu chuyển tài liệu đáng tin cậy (sự cấp phép/ sự xác nhận) đối với các đăng ký được tập hợp với nhau?;
 - b) làm thế nào để người tổng hợp có được tài liệu đáng tin cậy (sự cấp phép/ sự xác nhận) đã được chuyển cho họ?;
 - c) làm thế nào để lưu trữ các tiêu chuẩn này qua một buổi làm việc?.

F.2 Sự xác nhận

1. Ngay sau khi thông điệp được tiếp nhận, công việc đầu tiên là xác nhận. Một đối tượng uỷ nhiệm được tạo ra;
2. Nếu thông điệp đó được ký duyệt, nó được thẩm tra lại (bao gồm giá trị của giấy chứng nhận) và DN của giấy chứng nhận trở thành nhận dạng của đối tượng uỷ nhiệm. Sau đó, sổ đăng ký được tìm kiếm cho đối tượng uỷ nhiệm và nếu tìm được, vai trò và nhóm sẽ được điền vào;
3. Nếu thông điệp không được ký duyệt, một đối tượng uỷ nhiệm trống sẽ được tạo ra với vai trò khách đăng ký. Bước này là dành cho sự cân đối và để tách riêng phần còn lại của quá trình ra;
4. Sau đó, thông điệp được xử lý theo lệnh và các đối tượng mà nó hành động theo.

F.3 Sự cấp phép

Đối với mọi đối tượng, người kiểm soát truy cập sẽ làm lại thông qua tất cả các đối tượng thuộc chính sách kiểm soát truy cập với đối tượng đó và xem liệu có sự xâm chuỗi nào qua các đối tượng được phép để thẩm tra lại rằng phương pháp được yêu cầu có được cấp cho đối tượng uỷ nhiệm không.

Nếu đối tượng liên quan đến bất kỳ đối tượng được phép nào có một vai trò hoặc nhân dạng hoặc nhóm chung với đối tượng uỷ nhiệm thì hoạt động được cấp phép.

F.4 Tự đăng ký

Khi một đăng ký được tạo mới, đối tượng uỷ nhiệm mặc định sẽ được tạo ra với nhân dạng của giấy chứng nhận DN của quản trị sở đăng ký với vai trò quản trị đăng ký. Theo cách này, bất kỳ thông điệp nào được ký duyệt bởi quản trị đăng ký sẽ nhận được tất cả đặc quyền.

Khi một đăng ký được tạo mới, một trường hợp đơn của chính sách kiểm soát truy cập được tạo ra như một chính sách kiểm soát truy cập mặc định. Điều này bao gồm việc tạo ra sự cho phép cần thiết làm ví dụ cũng như những đặc quyền và thuộc tính của đặc quyền.

F.5 Độ trình nội dung – Trách nhiệm của khách hàng

Khách hàng đăng ký (Registry Client) phải ký duyệt các nội dung trước khi đệ trình - nếu không nội dung đó sẽ bị từ chối.

F.6 Độ trình nội dung – Trách nhiệm của sở đăng ký

1. Với bất kỳ yêu cầu nào khác, khách hàng trước tiên phải được xác nhận. Trong trường hợp này, đối tượng uỷ nhiệm sẽ nhận được DN từ giấy chứng nhận;
2. Với mỗi yêu cầu trong thông điệp, Sự tiếp nhận đăng ký sẽ được tạo ra;
3. Sự tiếp nhận đăng ký được quy cho chính sách kiểm soát truy cập đơn mặc định;
4. Nếu một đối tượng uỷ nhiệm với nhân dạng của SO không có sẵn, một đối tượng xác định với DN của SO được tạo ra;
5. Đối tượng uỷ nhiệm với nhân dạng này sẽ được tạo ra.

F.7 Xoá/ Không tán thành nội dung – Trách nhiệm của khách hàng

Khách hàng đăng ký (Registry Client) phải ký vào header trước khi đệ trình, cho các mục đích xác nhận hoặc yêu cầu sẽ bị từ chối.

F.8 Xoá/ Không tán thành nội dung – Trách nhiệm của sở đăng ký

1. Với bất kỳ yêu cầu nào khác, khách hàng trước tiên sẽ phải được chấp nhận. Trong trường hợp này, đối tượng uỷ nhiệm sẽ nhận được DN từ giấy chứng nhận. Vì đó sẽ là một đối tượng uỷ nhiệm với nhân dạng này trong sở đăng ký, đối tượng uỷ nhiệm sẽ nhận được tất cả các vai trò từ đối tượng đó;
2. Như từng yêu cầu trong thông điệp (xoá hoặc phản đối), phương pháp thích hợp trong phân loại đối tượng đăng ký sẽ được truy cập;
3. Người kiểm soát truy cập thực hiện việc cấp phép bằng việc làm lại qua các đối tượng được phép liên quan với đối tượng này theo đường chính sách kiểm soát truy cập đơn mặc định;
4. Nếu việc cấp phép thành công, sau đó hoạt động được cho phép. Nếu không sự trả lời lỗi được gửi ngược lại với một thông điệp báo lỗi chấp nhận cấp phép phù hợp.

F.9 Sử dụng ds:KeyInfo (Thông tin chính)

Hai viễn cảnh sử dụng điển hình đối với ds: KeyInfo được mô tả dưới đây.

Viễn cảnh 1:

1. Khách hàng đăng ký (Registry Client) (RC) ký vào vùng mang thông tin và phong bì SOAP bằng việc dùng khoá riêng của mình;
2. Giấy chứng nhận của RC được chuyển tới sổ đăng ký trong trường KeyInfo (thông tin chính) của chữ ký tiêu đề;
3. Giấy chứng nhận của RC được chuyển tới sổ đăng ký trong trường KeyInfo (thông tin chính) của chữ ký vùng mang thông tin;
4. Cơ quan cấp phép lấy giấy chứng nhận từ trường KeyInfo (thông tin chính) trong chữ ký tiêu đề;
5. Cơ quan cấp phép phê chuẩn chữ ký cho tiêu đề sử dụng khoá công bố từ giấy chứng nhận;
6. Cơ quan cấp phép phê chuẩn chữ ký vùng mang thông tin bằng việc lặp lại bước 4 và 5 sử dụng giấy chứng nhận từ trường KeyInfo (thông tin chính) của chữ ký vùng mang thông tin. Lưu ý rằng bước này không phải là bước cần thiết nếu nhiệm vụ của việc phê chuẩn là trách nhiệm của người sử dụng nội dung cuối cùng, khách hàng đăng ký khác.

Viễn cảnh 2

1. RC1 ký vào vùng mang thông tin và phong bì SOAP sử dụng khoá cá nhân của mình và công bố với sổ đăng ký;
2. Giấy chứng nhận của RC1 được chuyển cho sổ đăng ký trong trường KeyInfo (thông tin chính) của chữ ký tiêu đề;
3. Giấy chứng nhận của RC1 được chuyển cho sổ đăng ký trong trường KeyInfo (thông tin chính) của chữ ký vùng mang thông tin. Bước này được yêu cầu để bổ sung cho bước 2 vì khi RC2 lấy lại được nội dung, nó sẽ thấy chữ ký của RC1 với vùng mang thông tin;
4. RC2 nhận lại nội dung từ sổ đăng ký;
5. Cơ quan thẩm quyền đăng ký ký duyệt phong bì SOAP sử dụng chữ khoá riêng. Cơ quan này gửi nội dung của RC1 và chữ ký của RC1 (được ký bởi RC1);
6. Cơ quan thẩm quyền đăng ký không cần gửi giấy chứng nhận trong trường KeyInfo (thông tin chính) vì RC2 được giả định rằng đạt được giấy chứng nhận của cơ quan thẩm quyền đăng ký ra khỏi nhóm và cài đặt nó vào trong kho khoá địa phương;
7. RC2 có được giấy chứng nhận của cơ quan thẩm quyền đăng ký ngoài kho khoá địa phương của nó và thẩm tra lại chữ ký của cơ quan thẩm quyền đăng ký;
8. RC2 có được giấy chứng nhận của RC1 từ trường KeyInfo (thông tin chính) của chữ ký vùng mang thông tin và phê chuẩn chữ ký trên vùng mang thông tin.

Phụ lục G

Hỗ trợ ngôn ngữ quốc gia (Native language support – NLS)

G.1 Định nghĩa

Mặc dù phần này chỉ bàn luận những bộ ký tự và ngôn ngữ nhưng những thuật ngữ dưới đây phải được định nghĩa rõ ràng.

G.1.1 Bộ ký tự mã hoá (CCS):

CCS là sự sắp xếp từ bộ những ký tự trừu tượng đến bộ trọn vẹn. [RFC 2130]. Những ví dụ của CCS là ISO-10646, US-ASCII, ISO-8859-1, v.v...

G.1.2 Hệ thống ký tự mã hoá (CES)

CES là sự sắp xếp từ CCS (hoặc một vài) đến bộ tám. [RFC 2130]. Ví dụ của CES là ISO-2002, UTF-8.

G.1.3 Bộ ký tự (charset)

- Charset là tập hợp các nguyên tắc cho việc sắp xếp từ chuỗi của bộ tám đến chuỗi ký tự. [RFC 2277], [RFC 2278]. Ví dụ của bộ ký tự là ISO-2022-JP. EUC-KR ;
- Danh sách bộ các ký tự được đăng ký có thể tìm thấy tại [IANA].

G.2 NLS và thông điệp yêu cầu/ phản hồi

Đối với quá trình chính xác của dữ liệu đối với cả khách hàng đăng ký (Registry Client) và các dịch vụ đăng ký, việc biết bộ ký tự nào được sử dụng rất cần thiết. Mặc dù phần thân của sự giao dịch này có lẽ bao gồm cả charset trong bản công bố mã xml, khách hàng đăng ký (Registry Client) và dịch vụ đăng ký sẽ xác nhận tham số charset trong tiêu đề MIME khi họ sử dụng text/xml. Vì như đã định nghĩa trong [RFC 3023], nếu một thực thể text/xml được nhận với tham số charset bị bỏ sót, bộ xử lý MIME và XML phải sử dụng giá trị charset mặc định của « us-ascii ». Ví dụ :

```
Content-Type: text/xml; charset=ISO-2022-JP
```

Cũng như vậy, khi một lá đơn/thực thể xml được sử dụng, tham số charset là không bắt buộc và khách hàng đăng ký (Registry Client) và dịch vụ đăng ký phải làm theo những yêu cầu trong phần 4.3.3 của [REC-XML] nhấn mạnh trực tiếp đến tính ngẫu nhiên này.

Nếu kiểu nội dung khác được chọn sử dụng, việc sử dụng charset phải tuân theo [RFC 3023].

G.3 NLS và lưu trữ RegistryObject (đối tượng đăng ký)

Phần này cung cấp những hướng dẫn NLS về việc một số đăng ký nên lưu trữ các trường hợp RegistryObject (đối tượng đăng ký) như thế nào.

Một ví dụ độc lập về kiểu con cụ thể của đối tượng đăng ký là khả năng hỗ trợ đa địa điểm. Do đó, không có ngôn ngữ hay bộ ký tự nào liên quan với một trường hợp RegistryObject (đối tượng đăng ký) đặc thù.

Một ví dụ độc lập về kiểu con cụ thể hỗ trợ đa địa điểm như sau. Mỗi thuộc tính của RegistryObject (đối tượng đăng ký) là khả năng I18N (ví dụ các thuộc tính về tên và mô tả trong kiểu đối tượng đăng ký) như đã xác định bởi [ebRIM], có thể có đa vị trí các giá trị đặc thù rõ ràng như các phần tử phụ LocalizedString xác định giá trị của thuộc tính khả năng I18N trong một vị trí cụ thể. Mỗi phần tử LocalizedString (chuỗi ký tự được địa phương hóa) có một thuộc tính charset và lang (ngôn ngữ) cũng như một thuộc tính value (giá trị) của chuỗi phân loại.

G.3.1 Bộ ký tự của LocalizedString (Chuỗi ký tự được địa phương hóa)

Bộ ký tự này được sử dụng bởi LocalizedString (chuỗi ký tự được địa phương hóa) đặc thù (LocalizedString) được xác định bằng thuộc tính charset. Người ta khuyên dùng UTF-8 hoặc UTF-16 cho khả năng thực hiện liên kết tối đa.

G.3.2 Thông tin ngôn ngữ của LocalizedString (Chuỗi ký tự được địa phương hóa)

Ngôn ngữ này được chỉ rõ trong xml: thuộc tính hoá học (phần 2.12 [REC-XML]).

G.4 NLS và lưu trữ các hạng mục kho

Phần này cung cấp hướng dẫn NLS về cách thức một số đăng ký nên lưu trữ các hạng mục kho

Trong khi ví dụ độc lập của một ExtrinsicObjects (đối tượng ngoại lai) có thể trợ giúp đa khu, nó thường liên kết với một hạng mục kho độc lập. Hạng mục này có thể nằm trong một khu độc lập hoặc nằm trong nhiều khu. Tài liệu này không chỉ rõ hạng mục này.

G.4.1 Bộ ký tự của các hạng mục kho

Loại nội dung MIME điển đạt header đối với nhiều phần mime chứa trong hạng mục kho chứa có thể bao gồm một thuộc tính "charset" xác định bộ ký tự được sử dụng bằng hạng mục kho chứa. Ví dụ:

```
Content-Type: text/xml; charset="UTF-8"
```

Người ta khuyên dùng UTF-16 hoặc UTF-8 cho khả năng thực hiện liên kết tối đa. Charset của hạng mục kho phải được bảo tồn như định dạng của nó trong giao dịch.

G.4.2 Thông tin ngôn ngữ của hạng mục kho

Header điều bộ của ngôn ngữ nội dung đối với phần thân điều bộ chứa trong hạng mục kho có thể xác định ngôn ngữ cho hạng mục kho định khu cụ thể. Giá trị của các đặc tính header điều bộ của ngôn ngữ nội dung phải tuân thủ [RFC 1766].

Tài liệu này hiện nay chỉ xác định phương pháp gửi thông tin của bộ ký tự và ngôn ngữ và cách thức nó được lưu giữ trong một đăng ký. Tuy nhiên, thông tin ngôn ngữ có thể được sử dụng là một tiêu chuẩn truy vấn như việc tách chỉ DTD viết bằng tiếng Pháp. Hơn nữa, một thủ tục đàm phán ngôn ngữ, giống khách hàng đăng ký (Registry Client) đang đặt ra ngôn ngữ thứ vị cho các thông điệp từ các dịch vụ đăng ký, có thể có chức năng khác cho việc chỉnh sửa tương lai của tiêu chuẩn này.

Phụ lục H

Hồ sơ đăng ký

Tất cả sổ đăng ký (Registry) phải hỗ trợ một cách chính xác một hồ sơ đăng ký. Hồ sơ đăng ký đó là một tài liệu XML để mô tả các khả năng của sổ đăng ký (Registry). Tài liệu hồ sơ đó phải phù hợp với phần tử RegistryProfile như đã mô tả trong giản đồ giao diện các dịch vụ đăng ký được đề cập trong phụ lục B. Sổ đăng ký (Registry) đó phải có thể sử dụng được RegistryProfile qua giao thức HTTP qua một URL. URL đó phải phù hợp với mẫu: <http://<base url>/registryProfile>

Tài liệu tham khảo

- [Bra97] Keywords for use in RFCs to Indicate Requirement Levels.
- [ebRIM] sổ đăng ký ebXML (ebXML Registry) Information Model version 2.0
<http://www.oasis-open.org/committees/regrep/documents/2.0/specs/ebRIM.pdf>
- [[lược đồ ebRIM]] sổ đăng ký ebXML (ebXML Registry) Information Model Schema
<http://www.oasis-open.org/committees/regrep/documents/2.0/schema/rim.xsd>
- [ebBPSS] ebXML Business Process Specification Schema
<http://www.ebxml.org/specs>
- [ebCPP] ebXML Collaboration-Protocol Profile and Agreement Specification
<http://www.ebxml.org/specs/>
- [ebMS] ebXML Messaging Service Specification, Version 1.0
<http://www.ebxml.org/specs/>
- [XPT] XML Path Language (XPath) Version 1.0
<http://www.w3.org/TR/xpath>
- [SQL] Structured Query Language (FIPS PUB 127-2)
<http://www.itl.nist.gov/fipspubs/fip127-2.htm>
- [SQL/PSM] Database Language SQL — Part 4: Persistent Stored Modules (SQL/PSM) [ISO/IEC 9075-4:1996]
- [IANA] IANA (Internet Assigned Numbers Authority).
Official Names for Character Sets, ed. Keld Simonsen et al.
<ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets>
- [RFC 1766] IETF (Internet Engineering Task Force). RFC 1766:
Tags for the Identification of Languages, ed. H. Alvestrand. 1995.
<http://www.cis.ohio-state.edu/htbin/rfc/rfc1766.html>
- [RFC 2119] IETF (Internet Engineering Task Force). RFC 2119
- [RFC 2130] IETF (Internet Engineering Task Force). RFC 2130
- [RFC 2277] IETF (Internet Engineering Task Force). RFC 2277:
IETF policy on character sets and languages, ed. H. Alvestrand. 1998.
<http://www.cis.ohio-state.edu/htbin/rfc/rfc2277.html>
- [RFC 2278] IETF (Internet Engineering Task Force). RFC 2278:
IANA Charset Registration Procedures, ed. N. Freed and J. Postel. 1998.
<http://www.cis.ohio-state.edu/htbin/rfc/rfc2278.html>
- [RFC 2828] IETF (Internet Engineering Task Force). RFC 2828:
Internet Security Glossary, ed. R. Shirey. May 2000.
<http://www.cis.ohio-state.edu/htbin/rfc/rfc2828.html>
- [RFC 3023] IETF (Internet Engineering Task Force). RFC 3023:
XML Media Types, ed. M. Murata. 2001.
<ftp://ftp.isi.edu/in-notes/rfc3023.txt>

[REC-XML] W3C Recommendation. Extensible Markup language(XML)1.0(Second Edition)

<http://www.w3.org/TR/REC-xml>

[UUID] DCE 128 bit Universal Unique Identifier

http://www.opengroup.org/onlinepubs/009629399/apdx.htm - tagcjh_20

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

[WSDL] W3C Note. Web Services Description Language (WSDL) 1.1

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

[SOAP11] W3C Note. Simple Object Access Protocol, May 2000,

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

[SOAPAt] W3C Note: SOAP with Attachments, Dec 2000,

<http://www.w3.org/TR/SOAP-attachments>

[XMLDSIG] XML-Signature Syntax and Processing,

<http://www.w3.org/TR/2001/PR-xmlsig-core-20010820/>
