

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
KHOA CÔNG NGHỆ THÔNG TIN  
BỘ MÔN HỆ THỐNG THÔNG TIN**

**NGUYỄN THỨC DUY ANH  
NGUYỄN THỊ KHÁNH HÒA**

**NGHIÊN CỨU, TÌM HIỂU VÀ XÂY DỰNG ỨNG  
DỤNG VỚI SEMANTIC WEB**

**KHOÁ LUẬN CỬ NHÂN TIN HỌC**

**TP. HCM, NĂM 2005**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
KHOA CÔNG NGHỆ THÔNG TIN  
BỘ MÔN HỆ THỐNG THÔNG TIN**

**NGUYỄN THỨC DUY ANH – 0112171**

**NGUYỄN THỊ KHÁNH HÒA – 0112219**

**NGHIÊN CỨU, TÌM HIỂU VÀ XÂY DỰNG ỨNG  
DỤNG VỚI SEMANTIC WEB**

**KHÓA LUẬN CỬ NHÂN TIN HỌC**

**GIÁO VIÊN HƯỚNG DẪN**

**TS. NGUYỄN TRẦN MINH THỨ.**

**NIÊN KHÓA 2001 - 2005**



Nhận xét của Giáo viên hướng dẫn

-----  
-----  
-----  
-----  
-----  
-----  
-----  
-----  
-----  
-----  
-----  
-----  
-----  
-----  
-----  
-----  
-----  
-----  
-----  
-----  
-----  
-----  
-----  
-----  
-----  
-----

KHOA CNTT

## LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn khoa Công Nghệ Thông Tin, Trường Đại Học Khoa học Tự nhiên đã hỗ trợ tạo nhiều điều kiện thuận lợi cho chúng em trong quá trình học tập cũng như quá trình thực hiện đề tài tốt nghiệp.

Chúng em xin ghi nhận lòng biết ơn sâu sắc đến ThS Nguyễn Trần Minh Thư đã tận tình hướng dẫn, truyền đạt cho em những kiến thức quý báu cùng với những lời động viên khuyến khích của Cô trong những lúc chúng em gặp khó khăn, trở ngại khi thực hiện đề tài.

Chúng em cũng xin chân thành cảm ơn quý Thầy Cô trong Khoa CNTT đã tận tình giảng dạy, trang bị cho chúng em những kiến thức cần thiết trong suốt quá trình học tập và nghiên cứu tại Khoa.

Con xin ghi nhớ công ơn của ba mẹ đã sinh thành, nuôi dưỡng và dạy dỗ con trưởng thành như ngày hôm nay. Ba mẹ luôn là chỗ dựa tinh thần vững chắc cho con khi con vấp ngã và gặp những khó khăn trong cuộc sống.

Cuối cùng chúng tôi xin gửi lời cảm ơn đến bạn bè đã hỏi thăm, động viên và giúp đỡ chúng tôi trong quá trình thực hiện đề tài, đặc biệt tôi xin gửi lời cảm ơn chân thành đến bạn Vũ Bá Quang đã giúp đỡ chúng tôi rất nhiều khi chúng tôi gặp khó khăn trong lúc thực hiện đề tài.

Mặc dù tôi đã nỗ lực hết sức để hoàn thành tốt đề tài của mình nhưng dù sao những điều sai sót trong đề tài là điều không thể tránh khỏi, kính mong Thầy Cô thông cảm và tận tình chỉ bảo cho chúng em, mong các bạn đóng góp ý kiến để chúng em có thể hoàn thiện đề tài của mình hơn.

TP. Hồ Chí Minh 7/2005

Nhóm thực hiện

Nguyễn Thúc Duy Anh - Nguyễn Thị Khánh Hòa

## MỤC LỤC

|                 |  |           |
|-----------------|--|-----------|
| <b>Chương 1</b> | <b>GIỚI THIỆU SEMANTIC WEB</b>   | <b>5</b>  |
| 1.1             | World Wide Web và những hạn chế của nó   | 5         |
| 1.2             | Sự ra đời của Semantic Web   | 6         |
| 1.2.1           | Semantic Web là gì?  | 7         |
| 1.2.2           | Semantic Web mang lại những gì?  | 9         |
| 1.2.3           | Kiến trúc Semantic Web   | 11        |
| 1.3             | Hoạt động của W3C về Semantic Web  | 21        |
| 1.3.1           | Nhóm <i>quan tâm đến Semantic Web</i> (Semantic Web Interest)                                  | 21        |
| 1.3.2           | Nhóm hoạt động và phát triển Semantic (Web Semantic Web Best Practices and Deployment Working) | 21        |
| 1.3.3           | Nhóm hoạt động truy xuất dữ liệu ( <i>Data Access Working</i> )                                | 22        |
| 1.3.4           | Nhóm RDF Core  | 22        |
| 1.3.5           | Nhóm WebOnt  | 22        |
| <b>Chương 2</b> | <b>CÁC NGÔN NGỮ SEMANTIC WEB</b>   | <b>23</b> |
| 2.1             | XML (eXtensible Markup Language) – Ngôn ngữ đánh dấu mở rộng                                   | 23        |
| 2.2             | DTDs và XML Schema   | 25        |
| 2.3             | RDF - Biểu diễn dữ liệu về dữ liệu   | 26        |
| 2.4             | RDF Schema - Định nghĩa RDF vocabulary   | 28        |
| 2.5             | DAML+OIL (DARPA Agent Markup Language + Ontology Inference Layer)                              | 31        |
| 2.5.1           | Giới thiệu về DAML   | 31        |
| 2.5.2           | Tại sao sử dụng DAML?  | 33        |
| <b>Chương 3</b> | <b>RDF – NỀN TẢNG CỦA SEMANTIC WEB</b>   | <b>35</b> |

|                 |   |           |
|-----------------|---|-----------|
| 3.1             | Gới thiệu.....                                    | 35        |
| 3.2             | Các khái niệm cơ bản.....                         | 35        |
| 3.2.1           | Namespace và cách khai báo.....                   | 35        |
| 3.2.2           | Qualified name(QName) và cách sử dụng.....        | 36        |
| 3.2.3           | Mô hình RDF ( RDF Model).....                     | 37        |
| 3.2.4           | Tripple và cách sử dụng namespace.....            | 42        |
| 3.2.5           | Kiểu dữ liệu có cấu trúc và Blank node.....       | 44        |
| 3.3             | Cấu trúc RDF/XML.....                             | 47        |
| 3.3.1           | Cú pháp RDF/XML cơ bản.....                       | 47        |
| 3.3.2           | RDF Container.....                                | 50        |
| 3.4             | RDF Collection.....                               | 55        |
| 3.5             | RDF Schema.....                                   | 58        |
| 3.5.1           | Gới thiệu.....                                    | 58        |
| 3.5.2           | Định nghĩa class (lớp).....                       | 59        |
| 3.5.3           | Định nghĩa property( thuộc tính).....             | 61        |
| 3.6             | FOAF : Sự mở rộng của RDF.....                    | 67        |
| <b>Chương 4</b> | <b>TRUY VẤN DỮ LIỆU TRONG RDF.....</b>            | <b>75</b> |
| 4.1             | Gới thiệu.....                                    | 75        |
| 4.2             | Tạo một câu truy vấn đơn giản.....                | 76        |
| 4.3             | Cú pháp của câu truy vấn.....                     | 77        |
| 4.4             | Những dạng cú pháp khác.....                      | 79        |
| 4.5             | Ràng buộc dữ liệu.....                            | 81        |
| 4.5.1           | Thêm vào kiểu Optional.....                       | 82        |
| 4.5.2           | Ràng buộc trong khối Optional.....                | 83        |
| 4.5.3           | Nhiều khối Optional.....                          | 84        |
| 4.5.4           | Khối Optional lồng nhau.....                      | 85        |
| 4.6             | Các phép toán và điều kiện trên câu truy vấn..... | 86        |

|                 |   |           |
|-----------------|---|-----------|
| 4.6.1           | Phép hội .....  | 86        |
| 4.6.2           | Phép chiếu.....   | 87        |
| 4.6.3           | DISTINCT .....  | 87        |
| 4.6.4           | ORDER BY .....  | 88        |
| 4.6.5           | LIMIT .....   | 89        |
| 4.7             | Lựa chọn biên được trả về.....  | 89        |
| 4.8             | Các toán tử trong ngôn ngữ SPARQL .....                                       | 92        |
| 4.8.1           | sop:RDFterm-equal.....  | 94        |
| 4.8.2           | sop:isURI .....   | 94        |
| 4.8.3           | sop:isBLANK .....   | 95        |
| 4.8.4           | sop:isLiteral .....   | 96        |
| 4.8.5           | sop:logical-or .....  | 96        |
| 4.8.6           | sop:logical-and.....  | 96        |
| 4.8.7           | sop:str.....  | 96        |
| 4.8.8           | so:lang.....  | 96        |
| 4.8.9           | sop:datatype .....  | 96        |
| <b>Chương 5</b> | <b>CÁC ỨNG DỤNG SEMANTIC WEB.....</b>   | <b>97</b> |
| 5.1             | Các lĩnh vực ứng dụng .....   | 97        |
| 5.1.1           | Search Engine .....   | 97        |
| 5.1.2           | Khung làm việc để quản lý tri thức - Framework for Knowledge Management ..... | 98        |
| 5.1.3           | Internet Agent .....  | 100       |
| 5.1.4           | Push System.....  | 100       |
| 5.1.5           | Relationship .....  | 100       |
| 5.2             | Các ứng dụng & dự án hiện có.....   | 101       |
| 5.3             | Các công cụ phát triển.....   | 101       |
| 5.4             | Giới thiệu Semantic Search Engine .....                                       | 101       |



|   |  |            |
|---|--|------------|
| 5.4.1   | Giới thiệu: .....  | 101        |
| 5.4.2   | So sánh giữa Search Engine truyền thống( Keyword/Free-Text Search Engine) và Semantic Search Engine: ..... | 102        |
| <b>Chương 6 ỨNG DỤNG MINH HỌA : TÌM KIẾM THÔNG TIN CỦA MỘT NGƯỜI, MỘT NHÓM NGƯỜI VÀ KHẢ NĂNG CỦA HỌ TRONG MỘT TỔ CHỨC .....</b> |  | <b>105</b> |
| 6.1   | Giới thiệu ứng dụng .....  | 105        |
| 6.2   | Các yêu cầu của ứng dụng.....  | 107        |
| 6.2.1   | Các yêu cầu lưu trữ .....  | 107        |
| 6.2.2   | Các yêu cầu giao diện .....  | 108        |
| 6.2.3   | Các yêu cầu chức năng .....  | 108        |
| 6.2.4   | Các yêu cầu phi chức năng .....  | 109        |
| 6.3   | Hướng tiếp cận và giải pháp .....  | 110        |
| <b>Chương 7 PHÂN TÍCH ỨNG DỤNG.....</b>   |  | <b>113</b> |
| 7.1   | Mô hình use-case.....  | 113        |
| 7.2   | Đặc tả use case .....  | 115        |
| 7.2.1   | Chức năng Tìm Người .....  | 115        |
| 7.2.2   | Chức năng Tìm Nhóm Người .....   | 116        |
| 7.2.3   | Chức năng Tìm Khả Năng .....   | 117        |
| 7.2.4   | Chức năng Xem Thông Tin Chi Tiết.....  | 118        |
| 7.2.5   | Chức năng Xem Lược Đồ .....  | 120        |
| 7.2.6   | Chức năng Quản Lý Người.....   | 121        |
| 7.2.7   | Chức năng Quản Lý Nhóm Người.....  | 129        |
| 7.2.8   | Chức năng Quản Lý Khả Năng.....  | 134        |
| <b>Chương 8 THIẾT KẾ VÀ CÀI ĐẶT ỨNG DỤNG .....</b>  |  | <b>141</b> |
| 8.1   | Thiết kế dữ liệu .....   | 141        |

|                 |   |            |
|-----------------|---|------------|
| 8.1.1           | Xây dựng ontology PC( Person's Competence).....   | 141        |
| 8.1.2           | Thiết kế dữ liệu mô tả tài nguyên người.....  | 145        |
| 8.1.3           | Thiết kế dữ liệu mô tả tài nguyên nhóm người.....   | 149        |
| 8.1.4           | Thiết kế dữ liệu mô tả tài nguyên khả năng.....   | 153        |
| 8.1.5           | Thiết kế file RDF (persons.rdf, groups.rdf, competences.rdf) lưu trữ<br>toàn bộ các tài nguyên người, nhóm người và khả năng trong hệ thống. .... | 156        |
| 8.1.6           | Thiết kế file personlist.rdf, grouplist.rdf, competencelist.rdf.....  | 160        |
| 8.1.7           | Thiết kế file XSL để chuyển file RDF sang file SVG .....  | 162        |
| 8.2             | Thiết kế xử lý .....  | 174        |
| 8.3             | Thiết kế giao diện.....   | 187        |
| 8.3.1           | Sơ đồ liên kết các màn hình.....  | 187        |
| 8.3.2           | Một số màn hình chính của ứng dụng:.....  | 188        |
| <b>Chương 9</b> | <b>KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....</b>  | <b>199</b> |
| 9.1             | Kết luận .....  | 199        |
| 9.1.1           | Kết quả về mặt nghiên cứu .....   | 200        |
|                 | Qua quá trình nghiên cứu và tìm hiểu về lĩnh vực Semantic Web, chúng em đã<br>hiểu được và ghi nhận lại một số kết quả đạt được như sau: .....    | 200        |
| 9.1.2           | Kết quả về chương trình ứng dụng .....  | 201        |
| 9.2             | Hướng phát triển .....  | 202        |
|                 | <b>TÀI LIỆU THAM KHẢO .....</b>   | <b>203</b> |
|                 | <b>PHỤ LỤC A 211</b>  |            |
|                 | <b>CÁC ỨNG DỤNG VÀ DỰ ÁN VỀ Semantic Web.....</b>   | <b>211</b> |
|                 | <b>PHỤ LỤC B 215</b>  |            |
|                 | <b>MỘT SỐ TOOL VÀ EDITOR.....</b>   | <b>215</b> |
|                 | <b>PHỤ LỤC C 222</b>  |            |
|                 | <b>CÁC SEMANTIC SEARCH ENGINE HIỆN CÓ.....</b>  | <b>222</b> |

## DANH MỤC CÁC HÌNH ẢNH

|                  |  |     |
|------------------|--|-----|
| <i>Hình 0-1</i>  | Semantic Web như một sự mở rộng của World Wide Web .....             | 2   |
| <i>Hình 1-1</i>  | Sự hình thành và phát triển của Semantic Web .....                   | 6   |
| <i>Hình 1-2</i>  | Đề xuất đầu tiên về WWW của Tim Berners-Lee năm 1989. ....           | 7   |
| <i>Hình 1-3</i>  | Liên kết ngữ nghĩa giữa các nguồn khác nhau trong Semantic Web ..... | 9   |
| <i>Hình 1-4</i>  | Kiến trúc Semantic Web.....  | 12  |
| <i>Hình 3-1</i>  | Mối quan hệ giữa các thành phần trong triple.....                    | 39  |
| <i>Hình 3-2</i>  | Mô tả một phát biểu đơn giản bằng đồ thị.....                        | 39  |
| <i>Hình 3-3</i>  | Biểu diễn nhiều phát biểu cho cùng một subject.....                  | 40  |
| <i>Hình 3-4</i>  | Plain Literal biểu diễn cho tuổi của Mr John.....                    | 41  |
| <i>Hình 3-5</i>  | Kiểu Typed Literal biểu diễn cho tuổi của Mr John.....               | 42  |
| <i>Hình 3-6</i>  | Typed Literal biểu diễn cho ngày tháng.....                          | 42  |
| <i>Hình 3-7</i>  | Tạo địa chỉ(address) cho Mr John.....                                | 45  |
| <i>Hình 3-8</i>  | Sử dụng một blank Node.....  | 46  |
| <i>Hình 3-9</i>  | Mô tả 1 bag container đơn giản.....                                  | 52  |
| <i>Hình 3-10</i> | Mô tả 1 Alt container đơn giản.....                                  | 53  |
| <i>Hình 3-11</i> | Mô tả một RDF : Collection.....                                      | 56  |
| <i>Hình 3-12</i> | Mô tả lớp và các lớp con.....  | 60  |
| <i>Hình 3-13</i> | Không gian domain và range của thuộc tính.....                       | 62  |
| <i>Hình 6-1</i>  | Mô hình biểu diễn sự liên kết giữa các loại dữ liệu.....             | 111 |
| <i>Hình 7-1</i>  | Mô hình Use case cho phân hệ người dùng .....                        | 113 |
| <i>Hình 7-2</i>  | Mô hình Use Case cho phân hệ người quản trị (admin) .....            | 114 |
| <i>Hình 7-3</i>  | Mô tả chức năng tìm người.....                                       | 115 |
| <i>Hình 7-4</i>  | Mô tả chức năng tìm nhóm người.....                                  | 116 |
| <i>Hình 7-5</i>  | Mô tả chức năng tìm khả năng.....                                    | 117 |
| <i>Hình 7-6</i>  | Mô tả chức năng xem thông tin chi tiết.....                          | 119 |
| <i>Hình 7-7</i>  | Mô tả chức năng xem lược đồ.....                                     | 120 |
| <i>Hình 7-8</i>  | Mô tả chức năng thêm người.....                                      | 122 |
| <i>Hình 7-9</i>  | Mô tả chức năng xoá người.....                                       | 123 |

|                  |  |     |
|------------------|--|-----|
| <i>Hình 7-10</i> | Mô tả chức năng sửa thông tin người.....   | 125 |
| <i>Hình 7-11</i> | Mô tả chức năng cập nhật một trang web liên quan đến nhiều người .                 | 127 |
| <i>Hình 7-12</i> | Mô tả chức năng cập nhật địa chỉ những trang web liên quan đến một người .         | 129 |
| <i>Hình 7-13</i> | Mô tả chức năng thêm nhóm người.....   | 130 |
| <i>Hình 7-14</i> | Mô tả chức năng xoá nhóm người. ....   | 131 |
| <i>Hình 7-15</i> | Mô tả chức năng sửa thông tin nhóm người.....                                      | 133 |
| <i>Hình 7-16</i> | Mô tả chức năng thêm khả năng.....   | 135 |
| <i>Hình 7-17</i> | Mô tả chức năng xoá khả năng.....  | 137 |
| <i>Hình 7-18</i> | Mô tả chức năng sửa thông tin khả năng.....  | 139 |
| <i>Hình 8-1</i>  | Đồ thị biểu diễn thông tin của một cá nhân.....                                    | 147 |
| <i>Hình 8-2</i>  | Đồ thị biểu diễn thông tin của một nhóm nghiên cứu.....                            | 151 |
| <i>Hình 8-3</i>  | Đồ thị biểu diễn thông tin của một khả năng.....                                   | 154 |
| <i>Hình 8-4</i>  | Mô hình dữ liệu cấp nâng cấp.....  | 157 |
| <i>Hình 8-5</i>  | Mô hình dữ liệu cuối cùng.....   | 161 |
| <i>Hình 8-6</i>  | Sơ đồ Sequence Diagram cho chức năng tìm người.....                                | 175 |
| <i>Hình 8-7</i>  | Sơ đồ Collaboration Diagram cho chức năng tìm người.....                           | 176 |
| <i>Hình 8-8</i>  | Sơ đồ Sequence Diagram cho chức năng hiển thị thông tin.....                       | 177 |
| <i>Hình 8-9</i>  | Sơ đồ Collaboration Diagram cho chức năng hiển thị thông tin.....                  | 178 |
| <i>Hình 8-10</i> | Sơ đồ Sequence Diagram cho chức năng thêm người.....                               | 179 |
| <i>Hình 8-11</i> | Sơ đồ Sequence Diagram cho chức năng thêm người.....                               | 180 |
| <i>Hình 8-12</i> | Sơ đồ Sequence Diagram cho chức năng xóa người.....                                | 181 |
| <i>Hình 8-13</i> | Sơ đồ Collaboration Diagram cho chức năng xóa người.....                           | 182 |
| <i>Hình 8-14</i> | Sơ đồ Sequence Diagram cho chức năng cập nhật người.....                           | 183 |
| <i>Hình 8-15</i> | Sơ đồ Collaboration Diagram cho chức năng cập nhật người.....                      | 184 |
| <i>Hình 8-16</i> | Sơ đồ Sequence Diagram cho chức năng cập nhật trang Web liên quan cho cá nhân..... | 185 |
| <i>Hình 8-17</i> | Sơ đồ Collaboration Diagram cho chức năng cập nhật người trang Web liên quan.....  | 186 |
| <i>Hình 8-18</i> | Sơ đồ liên kết các màn hình.....   | 187 |

|                  |  |     |
|------------------|--|-----|
| <i>Hình 8-19</i> | Giao diện chính của hệ thống người dùng .....                                  | 188 |
| <i>Hình 8-20</i> | Giao diện chính của phân hệ admin.....   | 189 |
| <i>Hình 8-21</i> | Giao diện tìm người.....   | 190 |
| <i>Hình 8-22</i> | Giao diện xem thông tin chi tiết.....  | 191 |
| <i>Hình 8-23</i> | Giao diện thực hiện chức năng xem chi tiết bằng lược đồ.....                   | 192 |
| <i>Hình 8-24</i> | Giao diện cập nhật thông tin. ....   | 193 |
| <i>Hình 8-25</i> | Giao diện thực hiện chức năng thêm người mới vào hệ thống .....                | 194 |
| <i>Hình 8-26</i> | Giao diện thực hiện chức năng xóa người ra khỏi vào hệ thống. ....             | 195 |
| <i>Hình 8-27</i> | Giao diện thực hiện chức năng sửa thông tin một người trong hệ thống.<br>..... | 196 |
| <i>Hình 8-28</i> | Giao diện của chức năng cập nhật thông tin thông qua URL.....                  | 197 |
| <i>Hình 8-29</i> | Giao diện của chức năng cập nhật thông tin thông qua tên. ....                 | 198 |

KHOA CNTT

## ĐỀ CƯƠNG CHI TIẾT

**Đề Tài:** Nghiên cứu, tìm hiểu, và xây dựng ứng dụng với Semantic Web

**Giáo Viên Hướng Dẫn:** ThS NGUYỄN TRẦN MINH THUR.

**Sinh Viên Thực Hiện:**

Nguyễn Thúc Duy Anh-0112171 Nguyễn Thị Khánh Hòa – 0112219.

### I. Mục Đích

Nghiên cứu, tìm hiểu về Semantic Web, lĩnh vực đang được xem là có tiềm năng cực lớn cho việc giải quyết các vấn đề về Web hiện nay.

Áp dụng các kiến thức thu thập được để xây dựng một ứng dụng minh họa.

### II. Nội Dung

Gồm hai phần:

1. Tổng Quan Về Semantic Web
2. Xây Dựng Ứng Dụng Minh Họa EPERSON APPLICATION.

#### Phần I: Tổng Quan Về Semantic Web

- Giới thiệu Semantic Web.
- Các ngôn ngữ Semantic Web .
- RDF - Nền tảng của Semantic Web .
- Ngôn ngữ truy vấn dữ liệu trong RDF .
- Các ứng dụng Semantic Web

#### Phần II: Xây Dựng Ứng Dụng EPerson Application

- Mô tả ứng dụng.
- Phân tích ứng dụng.
- Thiết kế và cài đặt ứng dụng.
- Kết luận và hướng phát triển.

### III. Công Cụ

- Các ngôn ngữ sử dụng: XML, XMLS, RDF/XML, RDF, RDFS, OWL, ...
- Các softwares và tools hỗ trợ việc xây dựng ứng dụng: SVG Viewer, RDF Editor...

### IV. Kết Quả

- Báo cáo Luận Văn: với những nội dung đề cập ở trên.
- Ứng dụng minh họa: EPerson Application.

*TPHCM, Ngày 13 tháng 7 năm 2005*

Giáo Viên Hướng Dẫn

Sinh viên Thực Hiện

ThS Nguyễn Trần Minh Thur

Nguyễn Thúc Duy Anh

Nguyễn Thị Khánh Hòa

## Lời mở đầu

Như John Naisbitt đã nói:

" *Chúng ta đang chìm ngập trong thông tin nhưng lại khát khao tri thức* "

Đúng vậy, World Wide Web chứa một lượng thông tin khổng lồ, chúng được tạo ra từ các tổ chức, cộng đồng và nhiều cá nhân với nhiều lý do khác nhau. Người sử dụng Web có thể dễ dàng truy cập những thông tin này bằng cách chỉ ra địa chỉ **URL – Uniform Resource Locator** và theo các liên kết để tìm ra các tài nguyên liên quan khác.

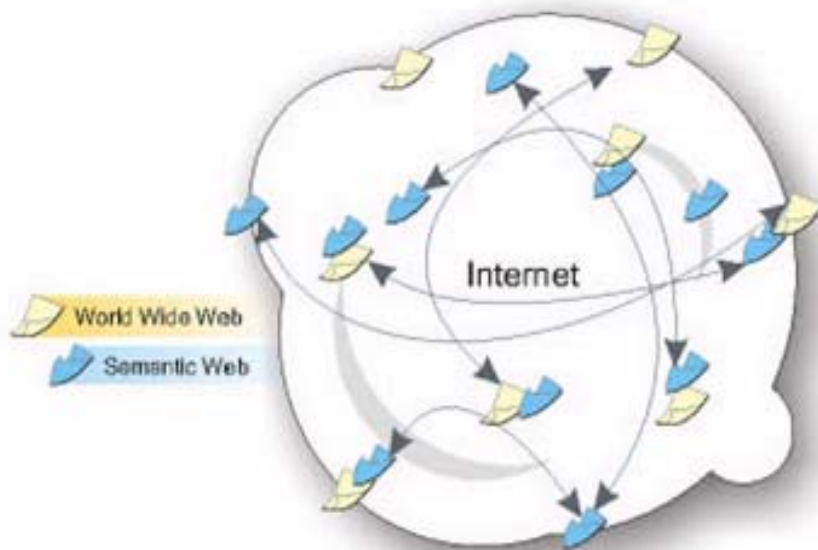
Tính đơn giản của Web hiện nay đã dẫn đến một số mặt hạn chế. Chẳng hạn như chúng ta có thể dễ dàng bị lạc hay phải đối đầu với một lượng thông tin không hợp lý và không liên quan được trả về từ kết quả tìm kiếm trên Web. Một ví dụ minh họa cho vấn đề được đề cập ở trên là giả sử chúng ta muốn tìm toàn bộ các *tài liệu* được viết bởi *Eric Miller*. Nhưng với chức năng tìm kiếm trên Web hiện nay, kết quả trả về có thể là tất cả các thông tin khác trên Web có đề cập đến *Eric* hay/và *Miller* như sổ nhật kí, danh bạ điện thoại, .... . Vấn đề tương tự cũng xuất hiện khi ta tìm kiếm các tài nguyên về *Marja*, kết quả tìm thấy là bất kì thông tin nào liên quan đến *Marja* như tên của một người hoặc là tên của một địa danh.... Hoặc khi chúng ta muốn tìm thông tin của một *chuyên gia về XML*, nhưng với kết quả tìm kiếm hiện tại chỉ cho ra một kết quả khổng lồ như các tài liệu liên quan đến XML, các ứng dụng hay công cụ về XML mà không cho ra chính xác thông tin mà chúng ta cần tìm kiếm. Như vậy câu hỏi đặt ra là làm thế nào để chúng ta có được một kết quả tìm kiếm chính xác và nhanh chóng theo đúng những gì mà chúng ta mong muốn?

Các kĩ thuật Web hiện nay đã cản trở khả năng phát triển thông tin của nó. Tính đơn giản của các kĩ thuật này đã gây ra hiện tượng thất cổ chai, tạo khó khăn

trong việc tìm kiếm, rút trích, bảo trì, và phát sinh thông tin. Máy tính chỉ được dùng như những thiết bị gửi và trả thông tin, chúng không thể truy xuất được những nội dung thật sự cần và do đó chúng chỉ hỗ trợ ở một mức giới hạn nào đó trong việc truy xuất và xử lý thông tin. Kết quả tất yếu là con người (người sử dụng) phải gánh trên vai trách nhiệm không những truy cập và xử lý thông tin mà còn rút trích và thông dịch mọi thông tin.

Để khắc phục các yếu điểm của Web hiện tại khái niệm “*Semantic Web*” đã ra đời. Và khái niệm này đã được Tim Berners-Lee định nghĩa như sau:

*“Semantic Web như một sự mở rộng của web hiện tại mà trong đó thông tin được xử lý một cách tự động bằng máy tính, làm cho máy tính và con người có thể hợp tác với nhau”* .



**Hình 0-1 Semantic Web như một sự mở rộng của World Wide Web**

Như vậy *Semantic Web* là một mạng lưới các thông tin được liên kết theo một cách thức để máy có thể dễ dàng xử lý thông tin ở mức độ toàn cầu. Chúng ta có thể hiểu đơn giản về nó như là một cách trình bày dữ liệu có hiệu quả trên mạng toàn cầu,



hoặc là một cơ sở dữ liệu được kết nối toàn cầu. Kỹ thuật Web hiện tại chỉ cho phép chúng ta lưu trữ thông tin và tìm kiếm thông tin trên những thông tin đã được lưu trữ. Chính vì vậy mà sự ra đời của *Semantic Web* là một bước tiến vượt bậc so với kỹ thuật Web hiện tại dựa vào khả năng làm việc với thông tin của chúng thay vì chỉ đơn thuần là lưu trữ thông tin.

Với sự lớn mạnh và khả năng lưu trữ thông tin ngữ nghĩa, *Semantic Web* sẽ trở thành một thế hệ Web cho tương lai, thế hệ Web mà chúng ta đang chờ đợi. Và đó cũng là lý do đề tài “**Nghiên cứu, tìm hiểu và xây dựng ứng dụng với *Semantic Web***” được chúng em quan tâm và chọn làm khóa luận tốt nghiệp .

Với mục tiêu nghiên cứu, tìm hiểu, và xây dựng ứng dụng với Semantic Web. Luận văn được chia làm hai phần:

**Phần 1:** Tổng Quan Về Semantic Web: Phần này đề cập đến những kiến thức nền tảng về Semantic Web, bao gồm các kiến trúc, khái niệm, ứng dụng điển hình, cũng như các công cụ và ngôn ngữ để xây dựng các ứng dụng Semantic Web. Bao gồm các chương sau:

**Chương 1:** Giới thiệu Semantic Web .

**Chương 2:** Các ngôn ngữ Semantic Web .

**Chương 3:** RDF- Nền tảng của Semantic Web .

**Chương 4:** Truy vấn dữ liệu trong RDF .

**Chương 5:** Các ứng dụng và công cụ phát triển Semantic Web .

**Phần 2:** Xây dựng ứng dụng Tìm Kiếm Thông Tin Của Một Người, Nhóm Người Và Khả Năng Của Họ Trong Một Tổ Chức . Bao gồm các chương sau:

**Chương 6:** Mô tả ứng dụng.

**Chương 7:** Phân tích ứng dụng.

**Chương 8:** Thiết kế và cài đặt ứng dụng.

**Chương 9:** Kết luận và hướng phát triển.

Trong luận văn này nhóm chúng em sẽ cố gắng trình bày các kết quả nghiên cứu một cách có hệ thống và dễ hiểu nhằm đóng một phần kiến thức có giá trị cho những ai quan tâm đến lĩnh vực Semantic Web .

Chúng em cũng đã cố gắng rất nhiều để luận văn đạt được kết quả tốt nhất. Tuy nhiên do kinh nghiệm còn non trẻ và trong một thời gian ngắn nên những sai sót xảy ra là điều không tránh khỏi. Chúng em xin chân thành cảm ơn và trân trọng tiếp thu tất cả những ý kiến đóng góp của quý thầy cô và bạn bè để luận văn được hoàn thiện hơn.

KHOA CNTT

## Chương 1 GIỚI THIỆU SEMANTIC WEB

### 1.1 World Wide Web và những hạn chế của nó

Hệ thống mạng toàn cầu đã trở nên rộng khắp thông qua một loạt các tiêu chuẩn được thiết lập rộng rãi và đảm bảo được các thành phần ở các mức độ khác nhau. Giao thức TCP/IP đảm bảo rằng chúng ta không phải lo lắng về việc chuyển từng bit dữ liệu thông qua hệ thống mạng nữa. Tương tự như vậy, HTTP và HTML đã cung cấp các cách tiêu biểu để có thể nhận thông tin và trình diễn các tài liệu siêu văn bản.

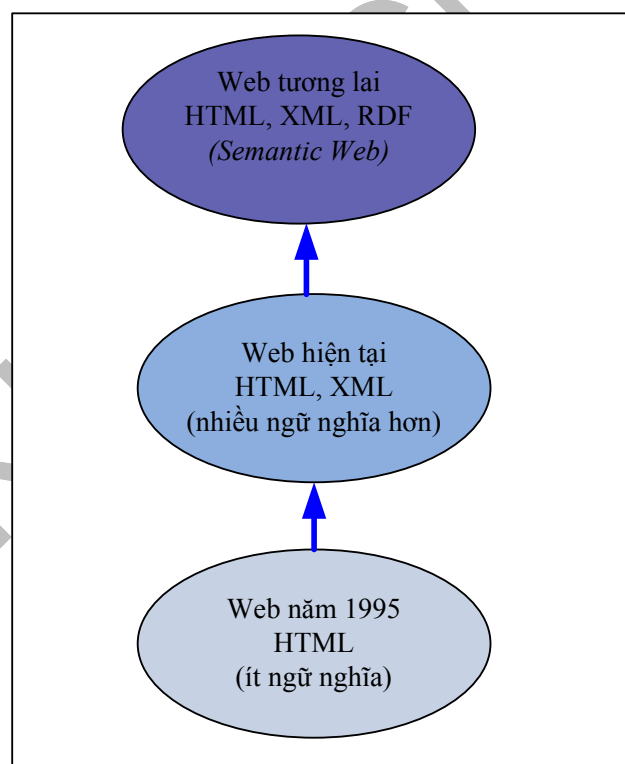
Tuy nhiên, có một khối lượng khổng lồ các tài nguyên trên Web, điều này làm nảy sinh vấn đề nghiêm trọng là làm thế nào để tìm kiếm chính xác tài nguyên mình mong muốn. Dữ liệu trong các file HTML hữu ích trong một vài ngữ cảnh nhưng vô nghĩa đối với những ngữ cảnh khác. Thêm vào đó HTML không thể mô tả về dữ liệu đóng gói trong nó.

Ví dụ, chúng ta biết mã vùng (*PostCode*) và muốn tìm địa chỉ của nó. Vì mỗi quốc gia có tên hệ thống mã vùng khác biệt và vì Web không biểu diễn được mối liên hệ này, nên chúng ta không nhận được điều chúng ta mong đợi. Trái lại, đối với Semantic Web, chúng ta có thể chỉ ra kiểu của mối liên hệ này; ví dụ, *Zip Code* tương đương với *PostCode*.

Vì vậy, nếu như các thành phần chính yếu của dữ liệu trong Web trình bày theo dạng thức thông thường, thì thật khó sử dụng dữ liệu này một cách phổ biến. Một thiếu sót của Web hiện nay là thiếu cơ cấu hiệu quả để chia sẻ dữ liệu khi ứng dụng được phát triển một cách độc lập. Do đó cần phải mở rộng Web để máy có thể hiểu, tích hợp dữ liệu, cũng như tái sử dụng dữ liệu thông qua các ứng dụng khác nhau.

## 1.2 Sự ra đời của Semantic Web

Thế hệ web đầu tiên bắt đầu với những trang HTML thủ công, thế hệ thứ hai đã tạo nên một bước cho máy thực hiện và thường là các trang HTML động. Các thế hệ web này mang ý nghĩa cho con người thao tác trực tiếp (đọc, duyệt, điền vào mẫu). Thế hệ web thứ ba được gọi là “*Semantic Web*”, mang mục đích là thông tin sẽ do máy xử lý. Điều này trùng khớp với quan điểm của Tim Berners-Lee đã mô tả trong cuốn sách gần đây nhất của ông “*Weaving the Web*” [Berners-Lee, 1999]. *Semantic Web* sẽ làm cho các dịch vụ thông minh hơn ví dụ như môi giới thông tin, tác nhân tìm kiếm, bộ lọc thông tin v.v. Những dịch vụ thông minh trên hệ thống web giàu ngữ nghĩa sẽ như thế chắc hẳn sẽ vượt trội hơn những phiên bản sẵn có hiện tại của các dịch vụ này, mà chỉ giới hạn về chức năng.

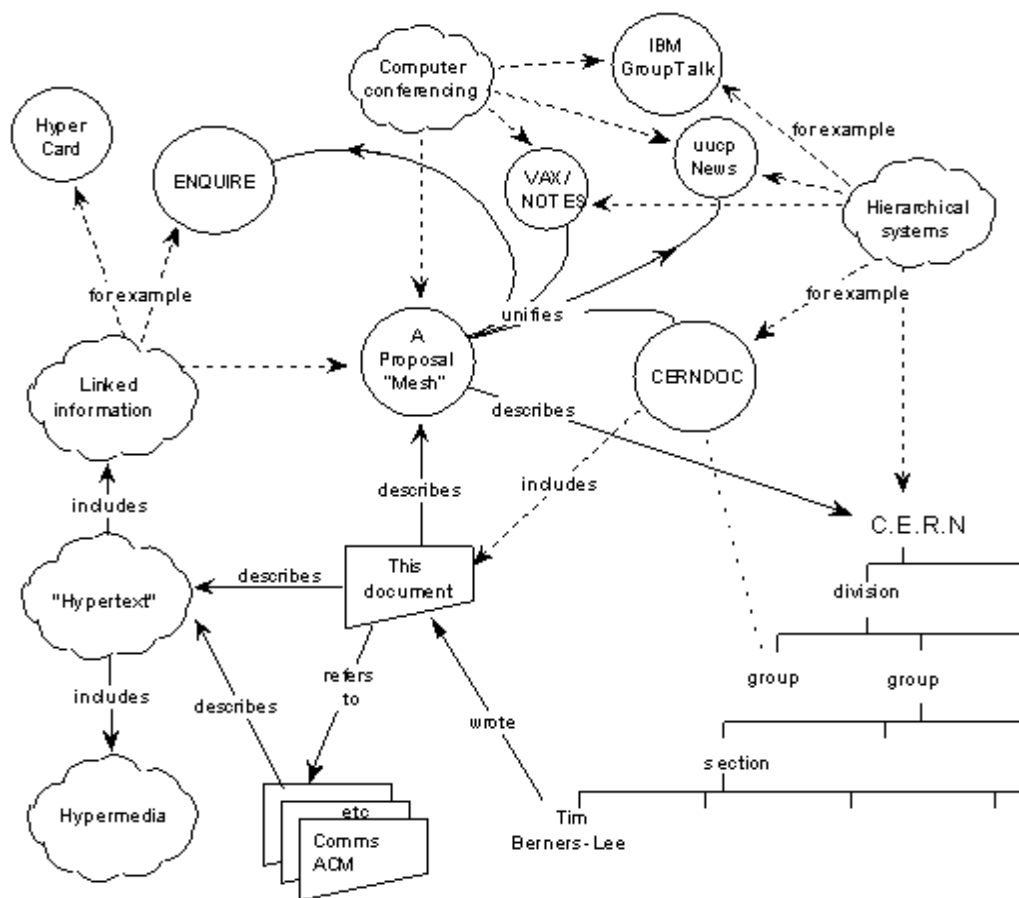


Hình 1-1 Sự hình thành và phát triển của Semantic Web .

### 1.2.1 Semantic Web là gì?

*Semantic Web* không là Web riêng biệt nhưng là một sự mở rộng của Web hiện tại, theo cách thông tin được xác định ý nghĩa tốt hơn, nó cho phép máy tính và người cộng tác với nhau tốt hơn.

*Semantic Web* được hình thành từ ý tưởng của Tim Berners-Lee, người phát minh ra WWW, URI, HTTP, và HTML.



**Hình 1-2 Đề xuất đầu tiên về WWW của Tim Berners-Lee năm 1989.**

*Semantic Web* là một mạng lưới các thông tin được liên kết sao cho chúng có thể được xử lý dễ dàng bởi các máy tính ở phạm vi toàn cầu. Nó được xem là cách mô tả thông tin rất hiệu quả trên World Wide Web, và cũng được xem là một cơ sở dữ liệu có khả năng liên kết toàn cầu.

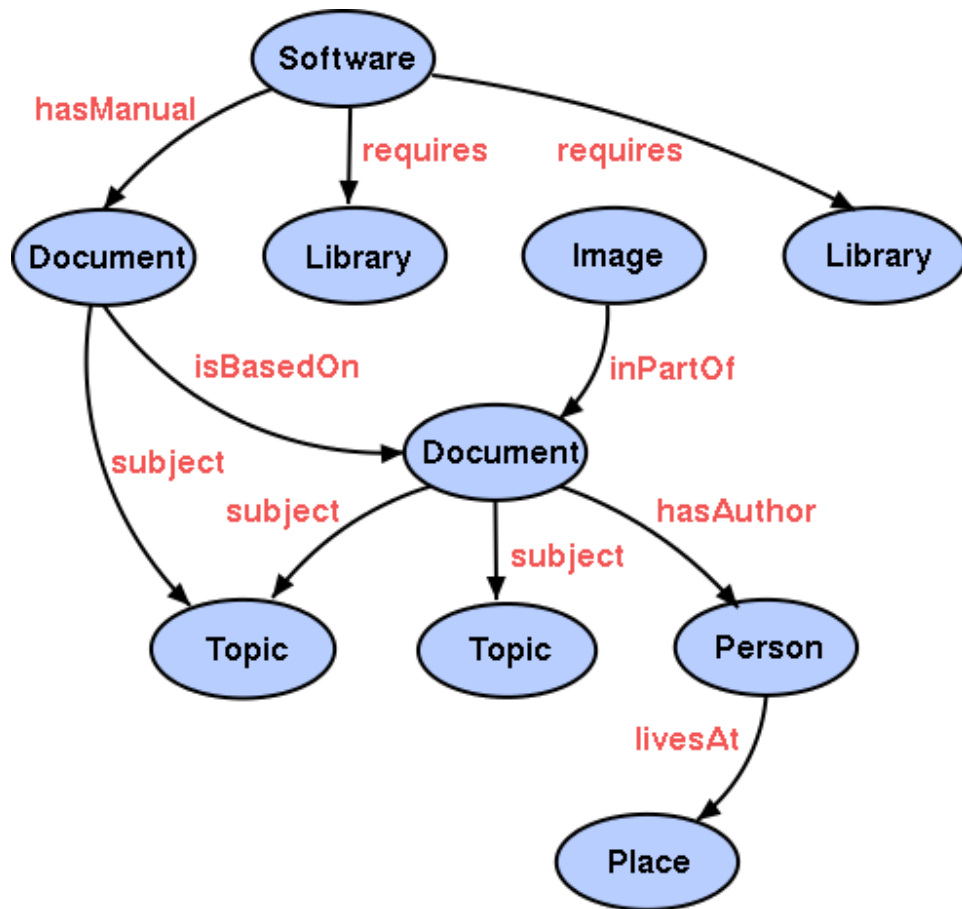
*Semantic Web* là một phương pháp cho phép định nghĩa và liên kết dữ liệu một cách có ngữ nghĩa hơn nhằm phục vụ cho máy tính có thể “hiểu” được. *Semantic Web* còn cung cấp một môi trường chia sẻ và xử lý dữ liệu một cách tự động bằng máy tính.

Ví dụ: Giả sử ta cần so sánh giá để chọn mua một bó hoa hay ta cần tra cứu catalog của các hãng chế tạo xe khác nhau để tìm ra thiết bị thay thế cho các bộ phận bị hư hỏng của xe Volvo 740. Thông tin mà ta thu được trực tiếp trên Web có thể trả lời các câu hỏi này nhưng đòi hỏi con người phân tích ý nghĩa của dữ liệu và sự liên quan của nó với yêu cầu đề ra, không thể xử lý tự động bằng máy tính.

Với *Semantic Web* ta có thể giải quyết vấn đề này bằng 2 cách:

- Thứ nhất, nó sẽ mô tả chi tiết dữ liệu của nó. Do đó một chương trình xử lý không cần quan tâm đến các format, hình ảnh, quảng cáo trên một trang Web để tìm ra sự liên quan của thông tin.
- Thứ hai, *Semantic Web* cho phép chúng ta tạo ra một file mô tả mối liên hệ giữa các tập dữ liệu khác nhau. Ví dụ, ta có thể tạo một liên kết semantic giữa một cột ‘*zip-code*’ trong database với trường ‘*zip*’ ở trên form nhập liệu nếu chúng có chung ý nghĩa. Điều này cho phép máy tính theo các link và tích hợp dữ liệu từ nhiều nguồn khác nhau.

Ý tưởng liên kết các nguồn khác nhau (tài liệu, hình ảnh, con người, khái niệm, ...) cho phép chúng ta mở rộng Web thành một môi trường mới với tập các mối quan hệ mới (như **hasLocation**, **worksFor**, **isAuthorOf**, **hasSubjectOf**, **dependsOn**, ...) giữa các nguồn dữ liệu, tạo ra các mối liên hệ ngữ cảnh (contextual relationship), điều mà Web hiện tại chưa làm được.



Hình 1-3 Liên kết ngữ nghĩa giữa các nguồn khác nhau trong Semantic Web

## 1.2.2 Semantic Web mang lại những gì?

### 1.2.2.1 Máy có thể hiểu được thông tin trên Web

Internet ngày nay dựa hoàn toàn vào nội dung. Web hiện hành chỉ cho con người đọc chứ không dành cho máy hiểu. Semantic Web sẽ cung cấp ý nghĩa cho máy hiểu. Ví dụ như:

- *The Beatles* là một ban nhạc nổi tiếng của *Liverpool*.
- *John Lennon* là một thành viên của *The Beatles*.
- Bản nhạc "*Hey Dude*" do nhóm *The Beatles* trình bày.

Những câu như thế này có thể hiểu bởi con người nhưng làm sao chúng có thể được hiểu bởi máy tính?

*Semantic Web* là tất cả những gì về cách tạo một Web mà cả người và máy có thể hiểu. Người dùng máy tính sẽ vẫn có thông tin trình bày theo cách trước đây, nhưng đối với máy tính, *Semantic Web* là ánh sáng soi rọi vào màn đêm của Web hiện hành. Bây giờ, máy không phải suy luận dựa vào ngữ pháp và các ngôn ngữ đánh dấu (*Markup Language*) nữa vì cấu trúc ngữ nghĩa của văn bản (*text*) thực sự đã chứa nó rồi.

### 1.2.2.2 Thông tin được tìm kiếm nhanh chóng và chính xác hơn

Với *Semantic Web*, việc tìm kiếm sẽ dễ dàng nếu mọi thứ được đặt trong ngữ cảnh. Ý tưởng chính yếu là toàn bộ ngữ cảnh mà người sử dụng được biết đến.

Mục tiêu của *Semantic Web* là phát triển các tiêu chuẩn và kỹ thuật để giúp máy hiểu nhiều thông tin trên Web hơn, để chúng tìm ra các thông tin dồi dào hơn, tích hợp, duyệt dữ liệu, và tự động hóa các thao tác.

Với *Semantic Web*, chúng ta không những nhận được những thông tin chính xác hơn khi tìm kiếm thông tin từ máy tính, mà máy tính còn có thể tích hợp thông tin từ nhiều nguồn khác nhau, biết so sánh các thông tin với nhau.

### 1.2.2.3 Dữ liệu liên kết động

Với *Semantic Web*, chúng ta có thể kết hợp các thông tin đã được mô tả và giàu ngữ nghĩa với bất kì nguồn dữ liệu nào.

Ví dụ, bằng cách thêm các **metadata** (*dữ liệu về dữ liệu*) cho các tài liệu khi tạo ra nó, chúng ta có thể tìm kiếm các tài liệu mà metadata cho biết tác giả là **Eric Miller**. Cũng thế, với metadata chúng ta có thể tìm kiếm chỉ những tài liệu thuộc loại **tài liệu nghiên cứu**.

Với *Semantic Web*, chúng ta không chỉ cung cấp các URI cho tài liệu như đã làm trong quá khứ mà còn cho con người, các khái niệm, các mối liên hệ. Như trong



ví dụ trên, bằng cách cung cấp những định danh duy nhất cho mỗi con người như vai trò của ‘*tác giả*’ và khái niệm ‘*tài liệu nghiên cứu*’, chúng ta đã làm rõ người ở đây là ai và mối quan hệ tương ứng của người này với một tài liệu nào đó. Ngoài ra, bằng cách làm rõ người mà chúng ta đang đề cập chúng ta có thể phân biệt những tài liệu của *Eric Miller* với những tài liệu của những người khác. Chúng ta cũng có thể kết hợp những thông tin đã được mô tả ở nhiều site khác nhau để biết thêm thông tin về người này ở những ngữ cảnh khác nhau ví dụ như vai trò của anh ta ra sao khi anh ta là tác giả, nhà quản lý, nhà phát triển, ...

#### 1.2.2.4 Hỗ trợ công cụ tự động hóa

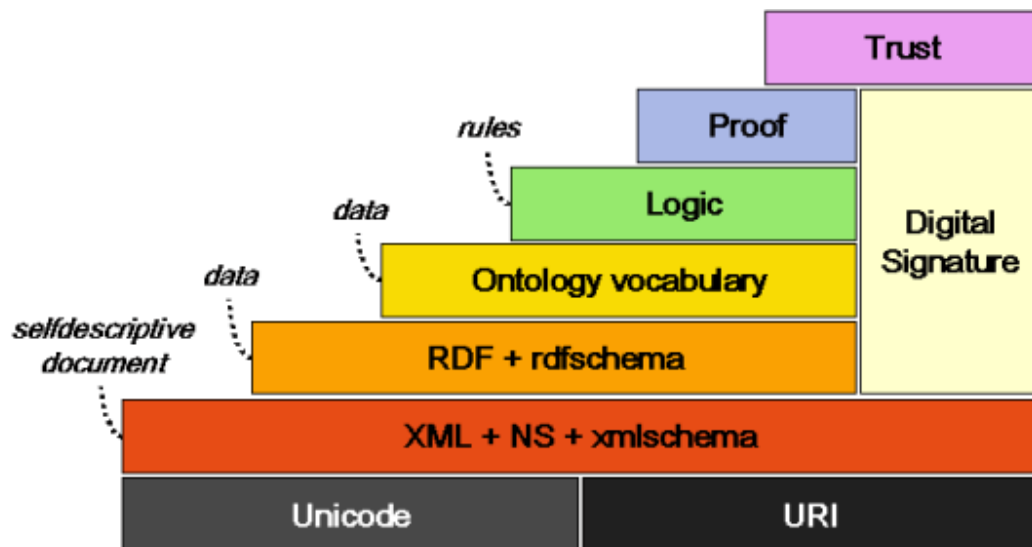
Ngoài ra, chúng còn cung cấp các loại dịch vụ tự động từ nhiều vùng khác nhau: từ gia đình và các thư viện kỹ thuật số cho đến các dịch vụ kinh doanh điện tử và dịch vụ sức khỏe.v.v.

*Semantic Web* còn cung cấp các phương tiện để thêm các thông tin chi tiết lên Web nhằm hỗ trợ sự tự động hóa cho các dịch vụ.

### 1.2.3 Kiến trúc Semantic Web

#### 1.2.3.1 Giới thiệu :

Semantic Web là một tập hợp/ một chồng (stack) các ngôn ngữ. Tất cả các lớp của Semantic Web được sử dụng để đảm bảo độ an toàn và giá trị thông tin trở nên tốt nhất.



Hình 1-4 Kiến trúc Semantic Web

- Lớp **Unicode & URI**: nhằm bảo đảm việc sử dụng tập kí tự quốc tế và cung cấp phương tiện nhằm định danh các đối tượng trong Semantic Web.
- Lớp **XML** cùng với các định nghĩa về *namespace* và *schema* (lược đồ) bảo đảm rằng chúng ta có thể tích hợp các định nghĩa Semantic Web với các chuẩn dựa trên XML khác.
- Lớp **RDF [RDF]** và **RDFSchemat [RDFS]**: ta có thể tạo các phát biểu (*statement*) để mô tả các đối tượng với những từ vựng và định nghĩa của URI, và các đối tượng này có thể được tham chiếu đến bởi những từ vựng và định nghĩa của URI ở trên. Đây cũng là lớp mà chúng ta có thể gán các kiểu (*type*) cho các tài nguyên và liên kết. Và cũng là lớp quan trọng nhất trong kiến trúc Semantic Web .
- Lớp **Ontology**: hỗ trợ sự tiến hóa của từ vựng vì nó có thể định nghĩa mối liên hệ giữa các khái niệm khác nhau.
- Lớp **Digital Signature**: được dùng để xác định chủ thể của tài liệu (vd: tác giả của một tài liệu hay một lời tuyên bố).

Các lớp **Logic, Proof, Trust**: đang trong giai đoạn nghiên cứu và các thể hiện của các ứng dụng giản đơn đang được xây dựng. Lớp **Logic** cho phép viết ra các luật (**rule**) trong khi lớp **Proof** thi hành các luật và cùng với lớp **Trust** đánh giá nhằm quyết định ứng dụng nên hay không nên tin tưởng/chấp nhận (**trust**) chứng cứ (**proof**).

### 1.2.3.2 Vai trò các lớp trong kiến trúc Semantic Web

Chúng ta sẽ xem xét các thành phần khác nhau của Semantic Web và cách các thành phần này thể hiện.

#### 1.2.3.2.1 Lớp định danh tài nguyên-URI

URI - *Uniform Resource Identifier*, URI đơn giản chỉ là một định danh Web giống như các chuỗi bắt đầu bằng “http” hay “ftp” mà bạn thường xuyên thấy trên mạng. Bất kỳ ai cũng có thể tạo một URI, và có quyền sở hữu chúng vì vậy chúng đã hình thành nên một công nghệ nền tảng lý tưởng để xây dựng một hệ thống mạng toàn cầu thông qua đó. Thật sự thì mạng chỉ là một nơi mà bất kỳ thứ nào có một **URI** thì cũng được xem là ở trên mạng.

Khi sử dụng URI, chúng ta có thể dùng cùng một cách đặt tên đơn giản để đề cập đến các tài nguyên dưới các giao thức khác nhau: HTTP, FTP, GOPHER, EMAIL,....

Một dạng thức quen thuộc của URI là URL - *Uniform Resource Locator*. Một URL là một địa chỉ cho phép chúng ta thăm một trang Web, như: <http://www.w3.org/Addressing/>. Nếu click vào nó, URL sẽ bảo máy tính nơi để tìm thấy .

Mặc dù thường được đề cập đến như URL, nhưng URI cũng được đề cập đến như các khái niệm trong Semantic Web để chỉ các resource .

Ví dụ, chúng ta có một quyển sách với tiêu đề “*Machine Learning*”, URI của nó như sau: <http://www.cs.bris.ac.uk/home/pw2538/book/title#machinelearning>.

Dưới đây là vài ví dụ về URI:

- **uuid:04b749bf-3bb2-4dba-934c-c92c56b709df:** là một UUID-*Universal Unique Identifier*. UUID có thể được thiết lập bằng cách kết hợp thời gian và địa chỉ của Ethernet card hay một con số ngẫu nhiên, sau đó nó được xác định là duy nhất.
- **mailto:pw2538@bristol.ac.uk:** xác định địa chỉ mail của một người nào đó.  
***“Mọi thứ trên Web đều có thể có một URI duy nhất.”***

URI là nền tảng của Semantic Web. Trong khi mọi thành phần khác của Web gần như có thể được thay thế nhưng URI thì không. URI liên hệ các thành phần của Web lại với nhau.

Để định danh các thành phần trên Web, chúng ta sử dụng bộ định danh. Bởi vì chúng ta sử dụng một hệ thống đồng bộ về định danh và cũng bởi vì mỗi thành phần được định danh được xem như là một tài nguyên, nên chúng ta gọi những bộ định danh này là “Các Bộ Định Danh Tài Nguyên” hay URIs. Chúng ta có thể gán URI cho bất cứ thứ gì, và bất cứ thứ gì có URI đều có thể biểu diễn trên Web. Ví dụ: con người, quyển sách, con ruồi,... những gì mà chúng ta có thể nghĩ đến, tất cả đều có thể có URI.

Một nguyên tắc chung để tạo các URI là bắt đầu với một trang Web. Trang mô tả đối tượng được định danh và giải thích rằng URL của trang là URI cho đối tượng đó. Ví dụ muốn tạo một URI cho bản sao “Weaving the Web” của Tim Berners-Lee: trước tiên ta tạo một trang Web mô tả bản sao; tiếp theo, ta ghi nhận rằng URI cho bản sao quyển sách tương tự URL của trang chính thức. Làm điều này chúng ta đã kết hợp URI (<http://logicerror.com/myWeavingTheWeb>) với bản sao “WeavingTheWeb”. Việc tạo một URI chỉ làm đơn giản như vậy.

Có thể nhận thấy rằng trong thể hiện này URI: “<http://logicerror.com/myWeavingTheWeb>” đang thực hiện hai nhiệm vụ: nó biểu diễn cả quyển sách vật lý cũng như trang Web nó mô tả. Đây là một lĩnh vực đang được thảo luận, gọi là vấn đề trong

định danh Semantic Web và nó là tâm điểm thảo luận cho người thực hiện Semantic Web.

#### 1.2.3.2.2 Lớp XML và XML Schema

Cho đến bây giờ, XML [Bray et al, 1998] đã được biết rộng rãi trên toàn thế giới và đã nhanh chóng là nền tảng cho sự phát triển phần mềm. XML được thiết kế để trở nên một cách đơn giản để có thể trao đổi các tài liệu (document) qua Web. Nó cho phép mọi người thiết kế định dạng tài liệu và sau đó viết một tài liệu theo định dạng đó.

XML là một mở rộng của ngôn ngữ đánh dấu cho các cấu trúc tài liệu bất kỳ, trái với HTML, là một loại ngôn ngữ đánh dấu chỉ dành cho các loại tài liệu siêu liên kết. Một tài liệu XML bao gồm một tập các thẻ đóng và thẻ mở được lồng vào nhau, ở đó mỗi một thẻ có một cặp các thuộc tính và giá trị. Phần cốt yếu của tài liệu XML là bộ từ vựng của các thẻ và sự kết hợp được cho phép thì không cố định, nhưng có thể được xác định thông qua mỗi ứng dụng XML.

Đây là một ví dụ của một tài liệu dạng văn bản đơn giản:

**I just got a new pet dog.**

```
<sentence>
<person href="http://aaronsw.com/">I</person>just got a new pet
<animal>dog</animal>.
</sentence>
```

Ở đây có một vấn đề, chúng ta đã dùng các từ: “sentence”, “person” và “animal” trong ngôn ngữ định dạng. Nhưng đây là các từ thông dụng. Điều gì xảy ra nếu những người khác cũng dùng những từ này trong ngôn ngữ định dạng của họ? và nếu những từ này có ý nghĩa khác trong những ngôn ngữ đó? Có thể “sentence” trong

ngôn ngữ định dạng khác nói đến số lần mà một tội phạm thụ án phải chịu cho một hình phạt hình sự.

### *Làm thế nào máy tính hiểu đúng?*

Để ngăn ngừa sự rối loạn này, phải xác định duy nhất các element định dạng. Tốt hơn hết là định danh chúng bằng một URI: gán URI cho mỗi element và attribute, thực hiện điều này bằng cách dùng XML Namespace. Theo cách này một người có thể tạo các tag riêng và trộn chúng với những tag của người khác. Một namespace chỉ là một cách nhận diện các phần của Web (space) để biết được ý nghĩa của những tên này. Tạo một Namespace cho ngôn ngữ định dạng bằng cách dùng một URI cho nó. Như đã nói ở trên, ta sẽ tạo một trang Web để mô tả ngôn ngữ định dạng và dùng URL trang Web như URI cho namespace.

Vì các tag của mỗi người có các URI riêng nên chúng ta không lo lắng về mâu thuẫn tên tag. Tất nhiên, XML cho phép chúng ta rút gọn và đặt URI mặc định nên chúng ta không phải gõ chúng ra mỗi lần.

```
<sentence
xmlns="http://example.org/xml/documents/"
xmlns:c="http://animals.example.net/xmlns/">
<c:person c:href="http://aaronsw.com/">I</c:person> just got a new pet
<c:animal>dog</c:animal>.</sentence>
```

Ở đây **http:// example.org/xml/documents/"** là namespace mặc định. Đó là nơi mà tất cả các element và attribute không có “c:” được đặt trước.

XML cho phép người dùng thêm cấu trúc tùy ý cho các tài liệu của họ nhưng không đề cập gì đến ý nghĩa của các cấu trúc. Tên các tag không cung cấp ý nghĩa. Semantic Web chỉ dùng XML cho mục đích cú pháp.

### **1.2.3.2.3 Lớp RDF - RDF Schema**

Ban đầu Web được tạo ra để con người thao tác: đọc, hiểu, .... Mặc dù máy có thể đọc được mọi thứ trên Web, nhưng nó không hiểu được dữ liệu trên Web. Giải pháp được đưa ra là dùng *siêu dữ liệu (metadata)* mô tả dữ liệu trên Web để máy có thể hiểu được chúng. *Siêu dữ liệu* là một dạng dữ liệu dùng để mô tả dữ liệu khác. Hay nói cách khác *siêu dữ liệu* là những thông tin mô tả tài nguyên trên Web. Chẳng hạn như, sách là một loại tài nguyên trên Web (<http://www.amazon.com/CSharp.htm>), khi đó các thông tin mô tả cho tài nguyên này: tên tác giả, tên tựa sách, ngày xuất bản, ... chính là *siêu dữ liệu*.

RDF (Resource Description Framework) là nền tảng của Semantic Web và xử lý *metadata*, được định nghĩa bởi tổ chức W3C. RDF cho phép trao đổi thông tin giữa các ứng dụng trên Web mà máy có thể hiểu được.

Cấu trúc căn bản của một RDF statement rất đơn giản. Gồm 3 thành phần:

- **Subject**: chủ thể - là cái mà chúng ta đề cập, thường được nhận diện bởi một URI.
- **Predicate**: thuộc tính của chủ thể, có kiểu metadata (ví dụ như tiêu đề, tác giả, ...), cũng có thể được xác định bởi một URI.
- **Object**: giá trị của thuộc tính (ví dụ: một người có tên *Eric Miller*).

Tập hợp các RDF statement được lưu dưới dạng cú pháp của XML, còn được gọi là **RDF/XML**.

Hãy xem một khai báo RDF đơn giản:

```
<http://aaronsw.com/>  
<http://love.example.org/terms/reallyLikes>  
<http://www.w3.org/People/Berners-Lee/Weaving/>
```

URI thứ nhất là *subject*. Trong thể hiện này *subject* là “me”. URI thứ 2 là *predicate*. Nó liên kết *subject* và *object*. Trong thể hiện này *predicate* là “reallyLikes”. URI thứ 3 là *object*. Ở đây object là “Tim Berners-Lee’s book ‘Weaving the Web’”. Vì vậy khai báo RDF trên nói rằng “I really like ‘Weaving the Web’”.

Ví dụ trên được viết dưới dạng RDF như sau:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:love="http://love.example.org/terms/">
  <rdf:Description rdf:about="http://aaronsw.com/">
    <love:reallyLikes rdf:resource="http://www.w3.org/People/Berners-
      Lee/Weaving/" />
  </rdf:Description>
</rdf:RDF>
```

Tuy nhiên mô hình dữ liệu RDF không cung cấp những cơ chế cho việc khai báo các thuộc tính, cũng như không cung cấp bất kỳ cơ chế nào để có thể định nghĩa ra những quan hệ giữa các thuộc tính và các tài nguyên. Đó sẽ là vai trò của *RDF schema*, hay nói cách khác *RDF schema* được dùng để định nghĩa các tài nguyên (các lớp trong *RDF schema*) và thuộc tính (thuộc tính trong *RDF schema*) cũng như các quan hệ qua lại giữa tài nguyên với tài nguyên, giữa thuộc tính với thuộc tính, và giữ tài nguyên với thuộc tính.

Tương tự XML schema, RDF schema là một tập những từ khoá mà qua đó RDF schema cho phép người dùng định nghĩa bộ từ vựng (*resource, property*) cụ thể cho dữ liệu RDF (ví dụ như: *hasName, hasPrice, authorOf, ...*) và định nghĩa các quan hệ của nó đến các đối tượng liên quan. Chẳng hạn như từ *hasName* ta định nghĩa quan hệ của nó trên hai đối tượng: '*http://www.w3c.org/employee/id1321*' và "*Jim Lerner*" như sau:

*hasName*

( '*http://www.w3c.org/employee/id1321*' , "*Jim Lerner*" )

#### 1.2.3.2.4 Lớp Ontology



Định nghĩa: *Ontology* là một tập các khái niệm và quan hệ giữa các khái niệm được định nghĩa cho một lĩnh vực nào đó nhằm vào việc biểu diễn và trao đổi thông tin. Đây cũng là một hướng tiếp cận để xây dựng Semantic Web. Tổ chức W3C cũng đã đề ra một ngôn ngữ *ontology* trên Web (OWL: Web Ontology Language) để xây dựng Semantic Web dựa trên nền tảng của *ontology*.

Một *Ontology* định nghĩa một bộ từ vựng mang tính phổ biến & thông thường, nó cho phép các nhà nghiên cứu chia sẻ thông tin trong một/nhiều lĩnh vực. Nó bao gồm các định nghĩa về các khái niệm căn bản trong một lĩnh vực và các mối liên hệ giữa chúng mà máy có thể hiểu được.

Một số lý do cần phát triển một *Ontology*:

- Để chia sẻ những hiểu biết chung về cấu trúc thông tin giữa con người và các software agent.
- Để cho phép tái sử dụng lĩnh vực tri thức (domain knowledge).
- Để làm cho các giả thuyết về lĩnh vực được tường minh.
- Để tách biệt tri thức lĩnh vực (domain knowledge) ra khỏi tri thức thao tác (operational knowledge).
- Để phân tích lĩnh vực tri thức.

#### 1.2.3.2.5 Lớp logic

Từ quan điểm trên, chúng ta sẽ thảo luận các thành phần của Semantic Web chưa được phát triển. Không giống như thảo luận ở trên, chúng ta không bàn về một hệ cụ thể nào mà thay vào đó một khái niệm tổng quát có thể hình thành (hoặc đang hình thành) nhiều hệ khác nhau.

Sẽ là thật tốt nếu có những hệ hiểu những khái niệm cơ bản (subclass, inverse...), sẽ tốt hơn nếu chúng ta có thể khai báo các nguyên tắc logic và cho phép máy tính suy diễn (bằng cách suy luận) bằng cách dùng những nguyên tắc này.

Ở đây là một ví dụ: một công ty quyết định rằng nếu người nào bán hơn 100 sản phẩm, thì họ là thành viên của Super Salesman Club. Một chương trình thông

minh hiện nay có thể hiểu luật này để tạo một diễn dịch đơn giản “Jonh đã bán 102 sản phẩm, vì thế John là thành viên của Super Salesman Club”.

#### **1.2.3.2.6 Lớp Proof**

Chúng ta sẽ xây dựng các hệ hiểu logic và dùng chúng để chứng minh. Mọi người trên thế giới có thể viết các khai báo logic. Sau đó máy tính có thể theo những Semantic link (liên kết ngữ nghĩa) này để kiểm chứng.

Ví dụ: tập hợp các record bán hàng cho thấy rằng Jane đã bán 55 widget và 66 sprocket. Hệ thống kiểm kê cho biết widget và sprocket là những sản phẩm của các công ty khác nhau. Xây dựng luật biểu diễn rằng  $55 + 66 = 121$  và 121 lớn hơn 100 và như chúng ta biết người nào bán hơn 100 sản phẩm và là thành viên của Super Salesman Club. Máy tính kết hợp tất cả các luật logic lại với nhau thành một proof: Jane là một Super Salesman.

Lớp proof cũng cần thiết để cung cấp các thông tin cho các agent tự động. Một cách tự nhiên, chúng ta có thể kiểm tra các kết quả được suy luận bởi các Agent. Tuy nhiên, điều này đòi hỏi sự dịch thuật các cơ cấu suy luận nội tại của agent thành ngôn ngữ thể hiện proof thống nhất.

#### **1.2.3.2.7 Lớp Trust**

Tại thời điểm này có thể nghĩ rằng toàn bộ kế hoạch này quá lớn, nhưng sẽ vô dụng nếu có người nói ‘ai sẽ tin tưởng vào những hệ như thế?’ Sao không cho biết trang Web của bạn?, ai cũng có thể nói rằng mình là vua của thế giới theo nguyên tắc “anything can say anything about anything”, “ai sẽ ngăn điều này?”.

Điều này dẫn đến Digital Signature (chữ ký điện tử) xuất hiện. Chữ ký điện tử làm việc dựa trên toán học và mật mã, chữ ký điện tử cung cấp bằng chứng rằng một người nào đó viết một tài liệu hoặc đưa ra một lời tuyên bố. Khi đánh dấu tất cả khai báo RDF bằng chữ ký điện tử, chúng ta có thể chắc chắn rằng ai đã viết chúng (hoặc ít nhất là bảo đảm tính xác thực của chúng). Bây giờ chúng ta chỉ đơn giản bảo chương

trình những chữ ký nào đáng tin và những chữ ký nào không. Mỗi chữ ký có thể được đặt các mức độ hoặc độ tin cậy (hoặc độ nghi ngờ) mà dựa vào đó máy tính có thể quyết định nên đọc theo độ tin cậy bao nhiêu.

Tim Berners-Lee đã đề xuất một button mà khi click vào máy tính sẽ cố gắng cung cấp những lý do (có sẵn ở Web of Trust) để tin tưởng dữ liệu.

### **1.3 Hoạt động của W3C về Semantic Web**

Mục tiêu của hoạt động là thiết kế các công nghệ hỗ trợ máy tính dễ dàng trao đổi tri thức toàn cầu và công việc thiết yếu là làm cho máy có thể sử dụng và hiểu thông tin. Điều này rất có ý nghĩa, vì giúp con người hiệu quả hơn khi thu nhận kiến thức. Công việc này ủng hộ các dự án triển khai trong thời gian ngắn trong khi vẫn theo dõi các dự án nghiên cứu lâu dài.

#### **1.3.1 Nhóm quan tâm đến Semantic Web (Semantic Web Interest)**

Là một forum dành cho các thành viên và không phải là thành viên của tổ chức W3C để thảo luận sự đổi mới cũng như đưa ra các ý kiến về các ứng dụng của Semantic Web cũng đề xướng ra các cuộc thảo luận về các khả năng tiềm ẩn trong tương lai liên quan đến các công nghệ có thể hỗ trợ cho Semantic Web, và mối quan hệ giữa công việc đó với các hoạt động của tổ chức W3C và với cộng đồng người.

#### **1.3.2 Nhóm hoạt động và phát triển Semantic (Web Semantic Web Best Practices and Deployment Working)**

Tập trung vào các hoạt động thực tập và phát triển công nghệ Semantic Web, nhóm này sẽ cung cấp các hỗ trợ cho những người phát triển ứng dụng Semantic Web.

### **1.3.3 Nhóm hoạt động truy xuất dữ liệu(Data Access Working)**

Mục tiêu của nhóm là đánh giá các yêu cầu cho các ngôn ngữ truy vấn và các giao thức mạng cho Resource Description Framework và định nghĩa các kỹ thuật đơn giản và thử nghiệm để hỗ trợ các yêu cầu trên.

### **1.3.4 Nhóm RDF Core**

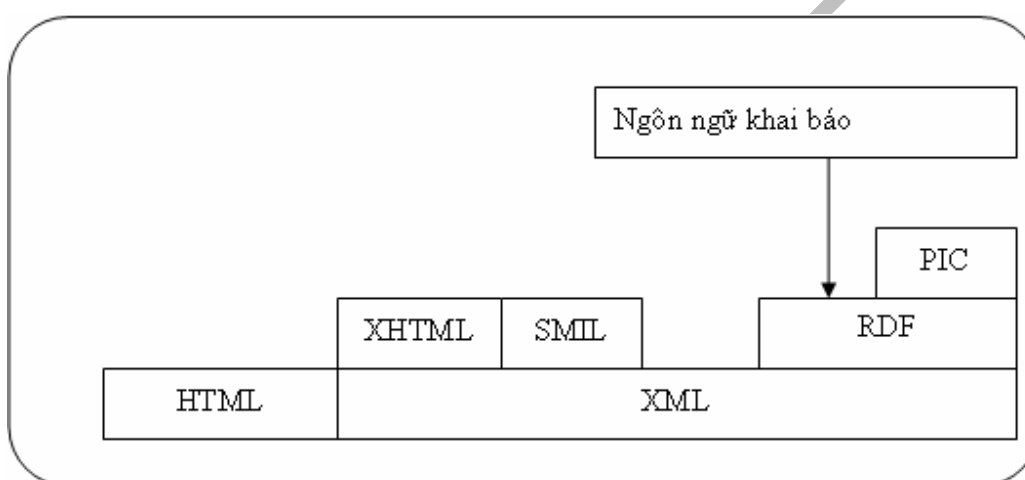
Mục tiêu của nhóm là sàng lọc và thúc đẩy mô hình trừu tượng của RDF và cú pháp XML. Nhóm này được đặc quyền hoàn tất việc mô tả từ vựng RDF trong RDF Schema, giải thích các mối liên hệ giữa các thành phần cơ sở của RDF- mô hình, cú pháp, giản đồ, và các cú pháp của họ XML.

### **1.3.5 Nhóm WebOnt**

Nhóm chuẩn hóa các cách định nghĩa Web Ontology trong đó các định nghĩa này dùng để mô tả cấu trúc của các khái niệm. Đầu vào cho công việc của nhóm là DAML+OIL. WebOnt xây dựng các chuẩn dựa trên RDF Schema (các lớp và các lớp con, các thuộc tính và các thuộc tính con) và mở rộng việc xây dựng để cho phép thêm các mối liên hệ phức tạp giữa các thực thể. Ví dụ, nó có thể giới hạn các kiểu thuộc tính của các lớp. Nó còn cung cấp phương tiện để suy diễn các thành phần có các thuộc tính đa dạng là thành viên của một lớp cụ thể nào đó, cung cấp mô hình được định nghĩa tốt cho thuộc tính kế thừa.

## Chương 2 CÁC NGÔN NGỮ SEMANTIC WEB

Ngôn ngữ biểu diễn dữ liệu và tri thức là một khía cạnh quan trọng của Semantic Web. Có nhiều ngôn ngữ cho Semantic Web, hầu hết những ngôn ngữ này dựa trên XML hay sử dụng XML làm cú pháp. Một số ngôn ngữ sử dụng RDF và RDFSchema.



Hình 2-1 Các tầng ngôn ngữ trên Web.

### 2.1 XML (eXtensible Markup Language) – Ngôn ngữ đánh dấu mở rộng

XML là một đặc tả cho các tài liệu (document) mà máy tính đọc được. Đánh dấu (**Markup**) có nghĩa là các chuỗi ký tự nào đó trong tài liệu có chứa thông tin chỉ ra vai trò nội dung của tài liệu. Markup mô tả sơ đồ (layout) dữ liệu của tài liệu (document) và cấu trúc logic. Markup làm thông tin tự mô tả tùy vào cảm nhận. Markup được mô tả dưới dạng các từ trong dấu ngoặc nhọn hay còn gọi là tag. Ví dụ <name> hay <h1>, về khía cạnh này, XML giống như HTML. Tuy nhiên, tính mở rộng của ngôn ngữ chỉ ra sự khác biệt quan trọng và cũng là đặc điểm chính của XML. XML thực sự là một siêu ngôn ngữ mô tả dữ liệu (metalinguage), là một cơ cấu cho

phép biểu diễn các ngôn ngữ khác một cách chuẩn hóa. Nói cách khác XML chỉ cung cấp định dạng dữ liệu cho tài liệu (document) được cấu trúc, không có đặc tả từ vựng (vocabulary) thực sự. Vì vậy XML phổ biến toàn cầu, chúng ta có thể tạo những ngôn ngữ đánh dấu (markup) mà chúng tùy ý sử dụng cho các kiểu tài liệu không giới hạn - đây là chuẩn đang phổ biến. Bên cạnh nhiều ngôn ngữ riêng biệt, một số ngôn ngữ chuẩn được định nghĩa trong XML (còn gọi là các ứng dụng XML). Ví dụ XHTML là một định nghĩa lại của HTML 4.0 trong XML.

Các thực thể (entity) markup chính trong XML là các đơn vị (element). Chúng thông thường bao gồm một tag mở và một tag đóng. Ví dụ `<person>` và `</person>`. Các element có chứa các element khác hay text. Nếu 1 element không có nội dung, nó có thể được viết ngắn gọn như `<person/>`. Các element nên được xếp lồng nhau, tag mở và tag đóng của element con phải nằm trong tag mở và tag đóng của element cha. Mỗi XML document phải có chính xác một root element.

Các element có thể chứa thuộc tính với giá trị nào đó, có định dạng là: "từ = giá trị" bên trong tag của một element. Ví dụ `<person name="John">`. Dưới đây là một đoạn XML:

```
<?xml version="1.0"?>
<employees>
List of persons in company:
<person name="John">
    <phone>47782</phone>
    On leave for 2001.
</person>
</employees>
```

XML không ngụ ý giải thích rõ ràng về dữ liệu. Thông qua tên của tag mà ý nghĩa của đoạn XML trên trở nên rõ ràng đối với cảm nhận của chúng ta, nhưng nó

không được mô tả một cách hình thức. Chỉ có giải thích hợp lý là mã XML chứa các thực thể (entity), các thực thể con và giá trị được đặt tên. Mỗi XML document thiết lập một cây có nhãn và thứ tự. Tóm lại chúng vừa là điểm mạnh vừa là điểm yếu của XML. Chúng ta có thể mã hóa bất kỳ kiểu cấu trúc dữ liệu nào theo một cú pháp không rõ ràng, nhưng XML không đặc tả sử dụng dữ liệu và ngữ nghĩa (semantic) của chúng. Các nhà phát triển sử dụng XML để trao đổi dữ liệu phải thỏa thuận trước về từ vựng, cách sử dụng và ý nghĩa.

## **2.2 DTDs và XML Schema**

**DTDs** - Document Type Definitions (định nghĩa kiểu tài liệu) và **XML Schemas** (lược đồ XML) mô tả tên của các element, attribute (thuộc tính) và việc sử dụng các element & attribute trong document (tài liệu), dù cả hai không đặc tả ý nghĩa của tài liệu.

Cả hai là cơ cấu để chúng ta đặc tả cấu trúc của các XML document. Chúng ta có thể xác minh tính hợp lệ của các document đối với cấu trúc được mô tả trước bởi một DTD hay XML Schema.

Các **DTD** chỉ cung cấp một luật cấu trúc đơn giản, chúng mô tả các element được phép xếp lồng vào nhau, các giá trị có thể có của element và những nơi text (văn bản) được phép. Ví dụ một DTD có thể ra luật mỗi element (đơn vị) **Person** phải có một attribute (thuộc tính) **name** và có thể có một element con được gọi là **phone**, những nội dung của chúng phải là dạng text (văn bản). Cú pháp của DTD trông khá bất tiện, nhưng nó thực sự rất đơn giản.

Các **XML Schema** là bản nâng cấp của DTD. XML Schema có nhiều thuận lợi hơn DTD. Trước tiên, cơ cấu XML Schema cung cấp ngữ pháp phong phú cho việc mô tả cấu trúc các element. Ví dụ, chúng ta có thể đặc tả chính xác các thể hiện của element con, chúng ta có thể đặc tả giá trị mặc định và đặt các element vào trong một nhóm được lựa chọn, điều này có nghĩa là một trong các element của nhóm được phép ở tại một nơi cụ thể. Kế đến XML Schema cung cấp kiểu dữ liệu.

Trong ví dụ ở đoạn trước, chúng ta có thể ra qui định nội dung của element **phone** là 5 số, có thể đặt trước 5 số vào giữa ngoặc đơn. Sự thuận tiện ở chỗ XML Schema cung cấp cơ cấu kế thừa và gộp vào, cho phép chúng ta tái sử dụng các định nghĩa (definitions) element thông thường và mô phỏng theo các định nghĩa đang tồn tại.

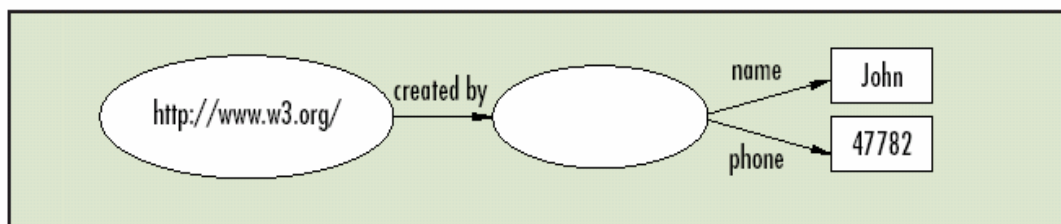
Sự khác biệt cuối cùng đối với DTD là các luật XML Schema dùng XML như cú pháp mã hóa, làm đơn giản hóa các công cụ phát triển vì cả luật cấu trúc (**Structure Prescription**) và đặc tả tài liệu (**prescribed documents**) sử dụng cùng cú pháp. Các nhà phát triển đặc tả XML Schema tìm thấy điểm này bằng cách sử dụng một tài liệu XML Schema để định nghĩa lớp (**class**) của các XML Schema document. Sau cùng, vì một XML Schema rescription (luật) là một ứng dụng XML, nó phải tuân theo các luật cho cấu trúc của nó, mà các luật này được định nghĩa bởi một XML Schema prescription. Tuy nhiên, cách định nghĩa đệ qui này khá rườ.

### **2.3 RDF - Biểu diễn dữ liệu về dữ liệu**

XML cung cấp cú pháp để mã hóa dữ liệu, RDF là một cơ cấu chỉ ra điều gì đó về dữ liệu. Như tên gọi, RDF là một mô hình để biểu diễn dữ liệu về "Mọi thứ trên Web". Mọi thứ ở đây chính là các tài nguyên trong RDF vocabulary (từ vựng). Mô hình dữ liệu cơ sở của RDF thì đơn giản, bên cạnh tài nguyên (resources), RDF còn chứa thuộc tính (properties) và câu phát biểu (statements). Một property là một khía cạnh, tính chất, thuộc tính, hay mối liên hệ mô tả cho một tài nguyên. Một statement bao gồm một tài nguyên riêng biệt, một thuộc tính được đặt tên, và giá trị thuộc tính cho tài nguyên đó. Giá trị này cơ bản có thể là một tài nguyên khác hay một giá trị mang tính nghĩa đen (literal value) hay dạng text tùy ý (free text)



| OBJECT                                     | ATTRIBUTE  | VALUE                |
|--|------------|----------------------|
| http://www.w3.org/<br>#anonymous_resource1 | created_by | #anonymous_resource1 |
| #anonymous_resource1                       | name       | "John"               |
| #anonymous_resource1                       | phone      | "47782"              |



**Hình 2-2 Ví dụ về một RDF**

Nhìn chung một RDF description (mô tả) là một danh sách các triple (bộ ba): object - attribute - value. Ví dụ, hình trên thể hiện 3 triple cần thiết để chỉ ra trang Web được tạo bởi ai đó với **name** là "John" và số **phone** là "47782".

Chúng ta vẽ hình oval cho mỗi tài nguyên, mũi tên cho mỗi thuộc tính và hình chữ nhật cho literal value. **Hình 2-2** biểu diễn một đồ thị về các triple có trong table. Từ hệ thống kí hiệu của ví dụ này, ta nhận thấy RDF không nhận biết cú pháp, nó chỉ cung cấp mô hình để biểu diễn metadata. Việc biểu diễn danh sách các triple giống như một đồ thị được đặt nhãn và các cú pháp khác nếu có. Dĩ nhiên XML là sự chọn lựa rõ ràng đối với các biểu diễn. Đặc tả mô hình dữ liệu (data model) bao gồm mã hóa cho RDF dựa trên XML.

Giống như XML, mô hình RDF không định nghĩa miền giá trị áp dụng hay giả định về một miền giá trị áp dụng cụ thể. Nó chỉ cung cấp một cơ cấu vùng không có tính chất rõ ràng để mô tả metadata. Xác định các thuộc tính có miền giá trị cụ thể và ngữ nghĩa (semantic) của chúng ta cần có các công cụ khác.

Vì RDF là ngôn ngữ chính của Semantic Web cho nên ta sẽ tìm hiểu kỹ nó vào chương sau.

## **2.4 RDF Schema - Định nghĩa RDF vocabulary**

Cơ bản, RDF Schema là một hệ thống kiểu đơn giản cho RDF. Nó cung cấp một cơ cấu để xác định các thuộc tính có miền giá trị cụ thể và các lớp tài nguyên để chúng ta áp dụng cho những thuộc tính này.

Mô hình cơ bản ban đầu trong RDF Schema là các định nghĩa lớp và lớp con (**class & subclass**), thuộc tính và thuộc tính con (**property & subproperty**), domain và range của các statement - để giới hạn các kết hợp có thể có giữa các lớp và thuộc tính, và các trình bày kiểu (để khai báo một tài nguyên như một thực thể của một lớp cụ thể). Ban đầu, chúng ta có thể xây dựng một lược đồ (schema) cho một miền giá trị cụ thể. Ví dụ, chúng ta có thể khai báo 2 lớp tài nguyên **Person** và **Webpage** và 2 thuộc tính name và phone có miền giá trị **Person** và phạm vi **literal**.

Chúng ta có thể dùng lược đồ này để xác định tài nguyên <http://www.w3c.org/> là một thực thể **Webpage** và tài nguyên nặc danh (anonymous resource) là một thực thể của **Person**. Ngoài ra, điều này sẽ giải thích cũng như kiểm định giá trị đối với dữ liệu RDF.

RDF Schema khá đơn giản so với các ngôn ngữ biểu diễn tri thức khác. RDF Schema cũng không cung cấp chính xác ngữ nghĩa (semantic). Tuy nhiên, sự bỏ sót này một phần là do có chủ ý. W3C thấy trước và chủ trương mở rộng xa hơn đối với RDF Schema.

Do đặc tả RDF Schema cũng là một kiểu metadata, chúng ta có thể dùng RDF để mã hóa chúng. Điều này chính xác với những gì thể hiện trong đặc tả RDF Schema document. Hơn thế nữa, đặc tả cung cấp một RDF Schema document để xác định các thuộc tính và các lớp được mô tả. Ví dụ:

```

<rdf:RDF xml:lang="en"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#">
<rdfs:Class rdf:ID="Person">
<rdfs:comment>The class of people.</rdfs:comment>
<rdfs:subClassOf
rdf:resource="http://www.classtypes.org/useful_classes#Animal"/>
</rdfs:Class>
<rdf:Property ID="maritalStatus">
<rdfs:range rdf:resource="#MaritalStatus"/>
<rdfs:domain rdf:resource="#Person"/>
</rdf:Property>
<rdf:Property ID="ssn">
<rdfs:comment>Social Security Number</rdfs:comment>
<rdfs:range
rdf:resource="http://www.datatypes.org/useful_types#Integer"/>
<rdfs:domain rdf:resource="#Person"/>
</rdf:Property>
<rdf:Property ID="age">
<rdfs:range
rdf:resource="http://www.datatypes.org/useful_types#Integer"/>
<rdfs:domain rdf:resource="#Person"/>
</rdf:Property>
<rdfs:Class rdf:ID="MaritalStatus"/>
<MaritalStatus rdf:ID="Married"/>
<MaritalStatus rdf:ID="Divorced"/>
<MaritalStatus rdf:ID="Single"/>
<MaritalStatus rdf:ID="Widowed"/>
</rdf:RDF>

```

Trong ví dụ này, Person là một lớp với mô tả về người. Person là lớp con của lớp Animal (động vật). Một Person có một thuộc tính age (tuổi). Giá trị của age là

kiểu integer. Một Person cũng có một thuộc tính ssn (Social Security Number - số an ninh xã hội). Giá trị của ssn là kiểu integer. Tình trạng marital (hôn nhân) của Person là một trong những tình trạng: Married (lập gia đình), Divorced (ly dị), Single (độc thân), Widowed (góa).

Chúng ta sử dụng hằng số để định nghĩa thuộc tính maritalStatus và lớp MaritalStatus. Sau đó, chúng ta dùng rdfs:range để chỉ rằng một thuộc tính maritalStatus chỉ có ý nghĩa khi nó có giá trị là một thực thể của lớp MaritalStatus. Schema (giản đồ) định nghĩa số lượng các thực thể của lớp này.

XML và RDF có những hình thức, mục đích, và vai trò trong viễn cảnh Semantic Web khác nhau, XML hướng đến cung cấp một cú pháp dễ sử dụng cho dữ liệu Web. Với nó, chúng ta có thể mã hóa tất cả các kiểu dữ liệu và dùng chúng để trao đổi giữa các máy tính, sử dụng XML Schema để tạo ra luật cho cấu trúc dữ liệu. Vì vậy XML trở thành ngôn ngữ nền tảng cho Semantic Web. Ngày nay nhiều kỹ thuật sử dụng XML làm cú pháp thiết yếu.

XML không giải thích trước về dữ liệu, vì vậy nó không đóng góp nhiều về khía cạnh ngữ nghĩa (semantic) của Semantic Web. RDF cung cấp một mô hình chuẩn để mô tả các dữ kiện về tài nguyên Web, cung cấp một số giải thích về dữ liệu. RDF Schema mở rộng những giải thích này thêm.

Tuy nhiên, để thực sự nhận biết được viễn cảnh Semantic Web, ta cần có thêm nhiều ngữ nghĩa về dữ liệu, những sự mở rộng & phát triển cao hơn là cần thiết. Thực sự đã có nhiều bước tiến triển theo khuynh hướng này - ví dụ, ngôn ngữ DAML+OIL (DARPA Agent Markup Language + Ontology Inference Layer) thêm các mô hình nguyên thủy (modeling primitive) mới và ngữ nghĩa hình thức (formal semantic) đối với RDF Schema.

## **2.5 DAML+OIL (DARPA Agent Markup Language + Ontology Inference Layer)**

### **2.5.1 Giới thiệu về DAML :**

DAML+OIL là ngôn ngữ đánh dấu ngữ nghĩa cho tài nguyên Web, được xây dựng dựa trên các chuẩn của W3C như RDF, RDF Schema, và mở rộng những ngôn ngữ này với các mô hình nguyên thủy phong phú hơn.

DAML+OIL cung cấp các mô hình nguyên thủy (primitive) thông thường được tìm thấy trong các hệ thống dựa trên frame (Frame là hệ thống biểu diễn tri thức được giới thiệu bởi Marvin Minsky, được sử dụng làm phương tiện chính nhằm biểu diễn vùng tri thức (domain knowledge), là một cấu trúc để biểu diễn một khái niệm hay tình huống. Gắn liền với frame là vài kiểu thông tin như thông tin định nghĩa, thông tin mô tả và cách sử dụng frame).

DAML+OIL có kế thừa giá trị từ các kiểu dữ liệu XML Schema, có ngữ nghĩa trong sáng và được định nghĩa tốt.

DAML+OIL kế thừa RDF và RDFS bằng cách thêm nhiều sự hỗ trợ cho kiểu dữ liệu và ngữ nghĩa. Những thúc đẩy này có thể thấy thông qua việc bổ sung nhiều thuộc tính và class.

Property (Thuộc tính): DAML thêm một “thuộc tính kiểu dữ liệu” nguyên nhằm giới hạn các kiểu dữ liệu được định nghĩa trong XML Schema hay các kiểu dữ liệu người dùng định nghĩa, ví dụ: float number, integer, ... . Trong DAML, một thuộc tính có thể có nhiều range, điều này mang lại sự uyển chuyển cao. Hơn thế nữa, DAML cho phép khai báo một thuộc tính unique, ví dụ, có hai thực thể với cùng giá trị. Đây là một hàm của một “daml:UniqueProperty” nguyên. Chúng ta cũng khai báo mối liên hệ giữa 2 thuộc tính mà chúng tương đương bởi vừa “daml:samePropertyAs” hay “daml:equivalentTo”.

DAML có nhiều đặc điểm mạnh mẽ trong các thuộc tính: chúng ta có thể diễn tả các mối liên hệ như “hoán đổi”, “bắc cầu”. Nếu A là chủ của B, thì B là nhân viên

của A. Các thuộc tính “employer” và “employee” có thể hoán đổi cho nhau được. Mỗi liên hệ này có thể diễn tả bằng “daml:inverseOf”. Tính “bắc cầu” có nghĩa là: nếu A là một bộ con của B, và B là một bộ con của C, thì A phải là bộ con của C. Thuộc tính “daml:TransitiveProperty” được sử dụng diễn tả mỗi liên hệ này. DAML cung cấp “daml:onProperty”, “daml:hasValue”, “daml:hasClass” và “daml:toClass” để giới hạn các class đối với tập các tài nguyên dựa trên các thuộc tính cụ thể. Sau đó chúng ta ràng buộc các luật cho các class cụ thể để một tài nguyên có thể là một thành viên của class nếu và chỉ nếu thuộc tính của nó thỏa mãn các yêu cầu. “daml:onProperty” nhận diện các thuộc tính được kiểm tra. Chúng ta có thể xác định giới hạn thuộc tính thông qua giá trị của nó “daml:hasValue”, ví dụ, thuộc tính phải có giá trị cụ thể “daml:hasClass” có thể được dùng để xác định giới hạn thuộc tính bởi class của các giá trị của một thuộc tính thay vì giá trị của nó. Ngược lại “daml:toClass” có nhiều giới hạn hơn vì nó cần tất cả giá trị thuộc tính cho một tài nguyên phải là một class cụ thể. Tuy nhiên, một tài nguyên không có thuộc tính cho sẵn “daml:onProperty” có thể cũng thỏa mãn điều kiện. Vì vậy đặc điểm này phải xem xét cẩn thận.

**Class:** “daml:class” là một subclass của “rdfs:Class” và DAML thêm nhiều đặc điểm tuyệt vời vào trong nó. Chúng ta có thể xây dựng nhiều mô tả hơn về các tài nguyên với những đặc điểm này. Chúng ta có thể định nghĩa một liệt kê mà không thể được bổ sung trong RDF. Trong DAML, element “daml:oneOf” định nghĩa một liệt kê. Chúng ta có thể định nghĩa một danh sách bằng cách khai báo “daml:oneOf” là kiểu cú pháp “daml:collection”. Thêm vào đó, chúng ta có thể xây dựng một vài liên hệ như “disjoin”, “union” và “intersection”. Cả hai “daml:disjointWith” và “daml:disjointUnionOf” có thể được sử dụng để chèn. Không có các instance thông thường trong số các class. Các kết hợp boolean không riêng biệt của các class có thể được diễn tả bằng “daml:unionOf”. Thuộc tính “daml:intersectionOf” có thể diễn tả sự giao nhau của các bộ.

**Ví dụ: dưới đây là một statement về lớp Woman**

```
<daml:Class rdf:ID="Woman">
<rdfs:subClassOf rdf:resource="#Person" />
<rdfs:subClassOf rdf:resource="#Female" />
</daml:Class>
```

Chúng ta nhận thấy rằng Woman là lớp con của lớp Person và lớp Female, chúng ta tiếp tục tra cứu các statement sau để tìm hiểu lớp Female

```
<daml:Class rdf:ID="Female">
<rdfs:subClassOf rdf:resource="#Animal" />
<daml:disjointWith rdf:resource="#Male" />
</daml:Class>
```

Chúng ta thấy rằng Female là lớp con của lớp Animal nhưng không phải là lớp Male. Chúng ta cũng quan tâm xem lớp Animal như thế nào

```
<daml:Class rdf:ID="Animal">
<rdfs:label>Animal</rdfs:label>
<rdfs:comment>This class of animals is illustrative of a number of
ontological idioms.</rdfs:comment>
</daml:Class>
```

Chúng ta đã lên tới lớp đỉnh của Ontology này, nghĩa là chúng ta không thể lên được lớp nào cao hơn lớp Animal trong domain (lĩnh vực) này.

### **2.5.2 Tại sao sử dụng DAML?**

RDF thì không phức tạp, nhưng RDF bên cạnh những thuận lợi mà RDF mang lại vẫn còn những khuyết điểm cần khắc phục. RDF không đủ đáp ứng khi chúng ta

muốn nhiều giới hạn kiểu dữ liệu hơn và muốn một biểu thức nhất quán cho enumeration (liệt kê), ...

Ví dụ, chúng ta muốn mô tả một quyển sách được bán bởi Amazon. Dưới đây là dạng RDF và RDFS

```
<rdfs:Class rdf:ID="Book">
<rdfs:label>Book</rdfs:label>
<rdfs:comment>A book sold by Amazon</rdfs:comment>
</rdfs:Class>
<rdfs:Property rdf:ID="pages">
<rdfs:label>Pages</rdfs:label>
<rdfs:domain rdf:resource="#Book"/>
<rdfs:range rdf:resource="http://www.w3.org/2000/01/rdfschema#Literal"/>
</rdfs:Property>
<Book rdf:ID="MachineLearning">
<rdfs:label>Machine Learning</rdfs:label>
<pages>432</pages>
</Book>
```

Điểm bất thuận lợi của dạng thức trên là literal có thể là chuỗi bất kỳ, nhưng chúng ta mong trang (pages) phải là một positive integer (số nguyên dương). So với RDF và RDFS, DAML cho phép chúng ta sử dụng nhiều kiểu dữ liệu chính xác hơn (được định nghĩa trong XSD) để mô tả dữ liệu. Ngoài những thuận lợi này, DAML có nhiều bộ dữ liệu mở đối với mọi người trên Web.

```
<daml:DatatypeProperty rdf:ID="pages">
<rdfs:label>Pages</rdfs:label>
<rdfs:domain rdf:resource="#Book"/>
<rdfs:rangerdf:resource="http://www.w3.org/2000/10/XMLSchema#positiveInteger"/>
</daml:DatatypeProperty>
```



## Chương 3 RDF – NỀN TẢNG CỦA SEMANTIC WEB

### 3.1 Giới thiệu

Ở những chương trước chúng ta đã nghe qua về khái niệm RDF (Resource Description Framework) và vai trò của RDF trong kiến trúc của Semantic Web. Có thể nói, RDF chính là nền tảng của Semantic Web, là linh hồn của Semantic Web. RDF mô tả siêu dữ liệu về các tài nguyên trên Web. RDF dựa trên cú pháp XML tuy nhiên XML chỉ mô tả dữ liệu, RDF còn có khả năng biểu diễn ngữ nghĩa giữa chúng thông qua các tài nguyên được định danh bằng URI. Ở chương này chúng ta sẽ có một cái nhìn cụ thể và bao quát hơn về RDF.

### 3.2 Các khái niệm cơ bản

#### 3.2.1 Namespace và cách khai báo

Namespace là một tập các tên (name), được định danh bởi các URI, được sử dụng trong các tài liệu XML như các element type và attribute name.

Một namespace được khai báo sử dụng một tập các thuộc tính có đã được định nghĩa. Tên của một thuộc tính phải có xmlns hay xmlns: như là một tiếp đầu ngữ.

Cách khai báo một namespace:

| Attribute Names for Namespace Declaration |   |
|---|---|
| [1]                                       | NSAttName ::= <a href="#">PrefixedAttName</a><br>  <a href="#">DefaultAttName</a> |
| [2]                                       | PrefixedAttName ::= 'xmlns:' <a href="#">NCName</a>                               |
| [3]                                       | DefaultAttName ::= 'xmlns'  |

|     |  |
|-----|--|
| [4] | NCName ::= ( <a href="#">Letter</a>   '_' ) ( <a href="#">NCNameChar</a> )*  |
| [5] | NCNameChar ::= <a href="#">Letter</a>   <a href="#">Digit</a>   '.'   '-'   '_'   <a href="#">CombiningChar</a>   <a href="#">Extender</a> |

Ta có một ví dụ khai báo cho một namespace sau:

```
<x xmlns:edi='http://ecommerce.org/schema'>
  <!--Tiếp đầu ngữ “edi” dùng để chỉ URI http://ecommerce.org/schema-->
</x>
```

### 3.2.2 Qualified name(QName) và cách sử dụng

Một QName bao gồm một tiếp đầu ngữ( prefix) mà đã được gán trước đó bởi một URI theo sau là dấu ‘:’ và tên cục bộ. Ví dụ như nếu một QName prefix *foo* được gán cho một URI là <http://example.org/somewhere/> cho nên ta có QName *foo:bar* là cách viết tắt của địa chỉ URI <http://example.org/somewhere/bar> .

Cách định nghĩa một QName:

| Qualified Name |  |
|----------------|--|
| [6]            | QName ::= ( <a href="#">Prefix</a> ':')? <a href="#">LocalPart</a> |
| [7]            | Prefix ::= <a href="#">NCName</a>                                  |
| [8]            | LocalPart ::= <a href="#">NCName</a>                               |

Cách dùng QName đối với kiểu dữ liệu là element

| Element Types |  |
|---------------|--|
| [9]           | S Tag ::= '<' <a href="#">QName</a> ( <a href="#">S Attribute</a> )* <a href="#">S</a> ? [ NSC: <a href="#">Prefix Declared</a> ]        |
| [10]          | E Tag ::= '</' <a href="#">QName</a> <a href="#">S</a> ? '>' [ NSC: <a href="#">Prefix Declared</a> ]                                    |
| [11]          | EmptyElemTag ::= '<' <a href="#">QName</a> ( <a href="#">S Attribute</a> )* <a href="#">S</a> ? [ NSC: <a href="#">Prefix Declared</a> ] |

Ví dụ:

```
<x xmlns:edi='http://ecommerce.org/schema'>
  <!-- the 'price' element's namespace is http://ecommerce.org/schema -->
  <edi:price units='Euro'>32.18</edi:price>
</x>
```

Cách dùng QName đối với kiểu dữ liệu là attribute

#### Attribute

[12] Attribute ::= [NSAttName](#) [Eq](#)  
[AttValue](#)  
 | [QName](#) [Eq](#) [AttValue](#) [ NSC: [Prefix Declared](#) ]

Ví dụ:

```
<x xmlns:edi='http://ecommerce.org/schema'>
  <!-- the 'taxClass' attribute's namespace is http://ecommerce.org/schema -->
  <lineltem edi:taxClass="exempt">Baby food</lineltem>
</x>
```

Một số QName phổ biến hay sử dụng (và cũng được sử dụng nhiều trong tài liệu này) như:

```
prefix rdf:, namespace URI: http://www.w3.org/1999/02/22-rdf-syntax-ns#
prefix rdfs:, namespace URI: http://www.w3.org/2000/01/rdf-schema#
prefix dc:, namespace URI: http://purl.org/dc/elements/1.1/
prefix owl:, namespace URI: http://www.w3.org/2002/07/owl#
prefix ex:, namespace URI: http://www.example.org/ (or
http://www.example.com/)
prefix xsd:, namespace URI: http://www.w3.org/2001/XMLSchema#
```

### 3.2.3 Mô hình RDF ( RDF Model)

Mô hình cơ bản của RDF gồm ba đối tượng sau:

- Tài nguyên (**Resources**): là tất cả những gì được mô tả bằng biểu thức RDF. Nó có thể là một trang Web, ví dụ như trang

<http://www.w3.org/Overview.html>, cũng có thể là một phần của trang Web, hoặc là một tập các trang Web, ...

- Thuộc tính (**Properties**): thuộc tính, đặc tính, hoặc quan hệ dùng để mô tả tính chất của tài nguyên.
- Phát biểu (**Statements**): mỗi phát biểu gồm ba thành phần sau
  - **Subject** (Tài nguyên): địa chỉ hay vị trí tài nguyên muốn mô tả, thông thường thì đây là chuỗi định URL hoặc URI.
  - **Predicate** (Vị ngữ): xác định tính chất của tài nguyên.
  - **Object** (Bổ ngữ): nội dung gán cho thuộc tính.

### 3.2.3.1 Bộ ba RDF ( RDF Tripple)

Mỗi một phát biểu (subject, predicate, object) còn gọi là một bộ ba (triple).

Ví dụ: Xét phát biểu sau

```
http://www.example.org/index.html has a creator whose value is John Smith
```

Phát biểu trên được phân ra thành các phần sau:

|                  |   |
|------------------|---|
| <b>Subject</b>   | http://www.example.org/index.html       |
| <b>Predicate</b> | http://purl.org/dc/elements/1.1/creator |
| <b>Object</b>    | http://www.example.org/staffid/85740    |

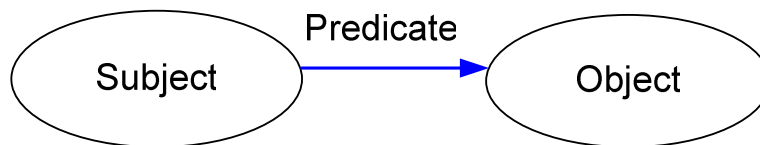
Chúng ta đã biết URI dùng để mô tả bất cứ cái gì trên Web, cho nên ở phát biểu trên thay vì phải dùng từ “creator” và “John Smith”, ta sẽ dùng URI “http://purl.org/dc/elements/1.1/creator” và “http://www.example.org/staffid/85740” để mô tả cho từng khái niệm trên.

Phát biểu trên được biểu diễn dưới dạng một tripple như sau:

```
<http://www.example.org/index.html> <http://purl.org/dc/elements/1.1/creator > <http://www.example.org/staffid/85740>
```

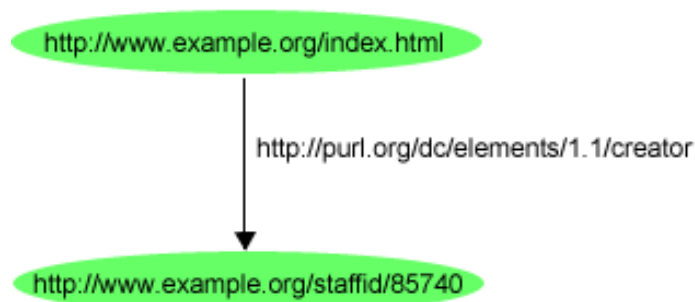
### 3.2.3.2 Đồ thị RDF ( RDF Graph)

Tập các triple hợp lại tạo thành đồ thị RDF( RDF Graph). Các node trong đồ thị có thể là các subject và object trong triple và các cung (arc) trong đồ thị là các predicate. Cho nên một triple còn có thể được mô tả dưới dạng node-arc-node. Hướng của đồ thị rất quan trọng. Cung của đồ thị luôn bắt đầu từ *subject* đến *object*.



**Hình 3-1** Mối quan hệ giữa các thành phần trong triple.

Phát biểu trên được mô hình hóa bằng đồ thị có hướng sau:



**Hình 3-2** Mô tả một phát biểu đơn giản bằng đồ thị

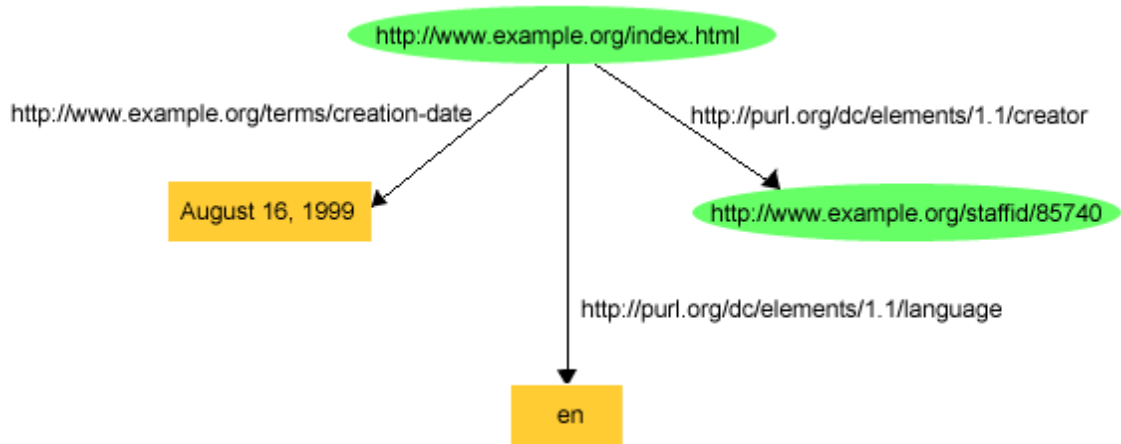
Ví dụ 2: Ta thêm các phát biểu sau cho cùng một subject

**http://www.example.org/index.html** has a **creation-date** whose value is **August 16, 1999**  
**http://www.example.org/index.html** has a **language** whose value is **English**

Được phân ra thành các thành phần sau:

| <b>Subject</b>                    | <b>Predicate</b>                           | <b>Object</b>                        |
|-----------------------------------|--|--------------------------------------|
| http://www.example.org/index.html | http://purl.org/dc/elements/1.1/creator    | http://www.example.org/staffid/85740 |
| http://www.example.org/index.html | http://www.example.org/terms/creation-date | August 16, 1999                      |
| http://www.example.org/index.html | http://purl.org/dc/elements/1.1/language   | En                                   |

Lúc này các phát biểu được biểu diễn dưới cùng một đồ thị như sau:



Hình 3-3 Biểu diễn nhiều phát biểu cho cùng một subject

Mô hình trên chỉ ra rằng các đối tượng trong phát biểu RDF có thể là một **URI** hoặc cũng có thể là một giá trị **Literal**. Ở hình trên ta dùng hình oval để biểu diễn một resource là một URI, và dùng hình chữ nhật để biểu diễn một resource là một giá trị Literal.

### 3.2.3.3 Literal

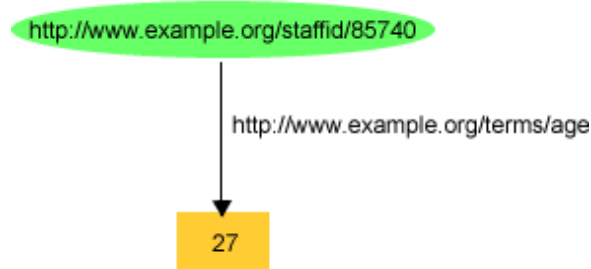
**Literal** được sử dụng để biểu diễn các giá trị như con số, ngày tháng, chuỗi... Bất cứ cái gì có thể biểu diễn bởi một giá trị Literal cũng có thể được biểu diễn dưới dạng một URI.

Một **Literal** có thể là **object** của một phát biểu nhưng không thể là **subject** hay là **predicate**.

**Literal** có hai kiểu sau : *plain literal* hay là *typed literal*:

- Một **plain Literal** là một chuỗi được kết hợp với một tag tùy ý. Trong ngôn ngữ tự nhiên ta có thể gọi nó là một kiểu text.

Ví dụ sau:



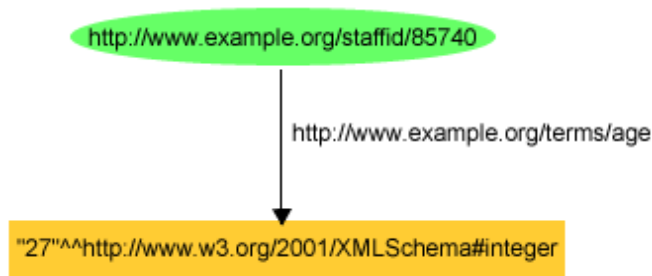
**Hình 3-4 Plain Literal biểu diễn cho tuổi của Mr John**

Ví dụ trên mô tả tuổi của Mr John Smith là 27. Ở đây ta biểu diễn tuổi của Mr John Smith là một kiểu plain literal. Giá trị 27 ở đây chỉ là một chuỗi gồm 2 ký tự '2' và '7'. Nhưng nếu ta muốn biểu diễn nó là một con số, một số nguyên thì phải làm như thế nào đây? Bằng cách dùng kiểu typed literal chúng ta sẽ có được cách định nghĩa mà chúng ta muốn.

- Một typed literal được hình thành bằng cách kết hợp một chuỗi với một định danh URI để biểu diễn một kiểu dữ liệu đặc biệt nào đó. Kết quả trả về là một node trong đồ thị tương tự như kiểu literal. Kiểu giá trị được biểu diễn bằng typed literal sẽ được ánh xạ đến kiểu giá trị đặc biệt mà ta đã khai báo thông qua URI. Ở ví dụ trên, khi muốn biểu diễn tuổi của Mr John là một kiểu số nguyên thì ta sẽ có cách khai báo như sau:

|                  |  |
|------------------|--|
| <b>Subject</b>   | http://www.example.org/staffid/85740             |
| <b>Predicate</b> | http://www.example.org/terms/age                 |
| <b>Object</b>    | "27"^^<http://www.w3.org/2001/XMLSchema#integer> |

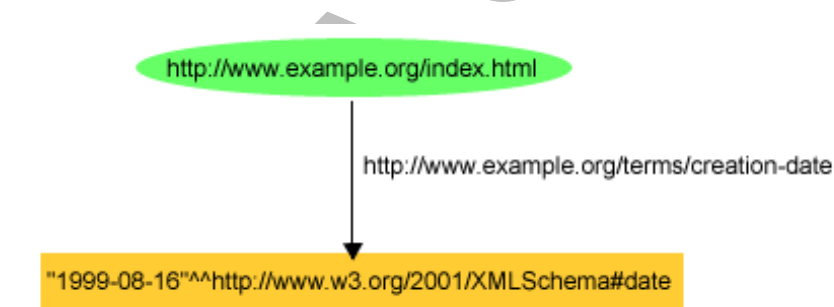
Và được biểu diễn thành đồ thị như sau:



**Hình 3-5** Kiểu Typed Literal biểu diễn cho tuổi của Mr John.

Tương tự như vậy ở ví dụ trên ta có thể dùng kiểu dữ liệu ngày tháng.

|                  |   |
|------------------|---|
| <b>Subject</b>   | http://www.example.org/staffid/85740          |
| <b>Predicate</b> | http://www.example.org/terms/creation-date    |
| <b>Object</b>    | "27"^^<http://www.w3.org/2001/XMLSchema#date> |



**Hình 3-6** Typed Literal biểu diễn cho ngày tháng.

### 3.2.4 Tripple và cách sử dụng namespace

Đôi khi không thuận tiện để biểu diễn các phát biểu dưới dạng đồ thị, ta có thể dùng cách viết dưới dạng tripple. Ta có thể viết lại như sau:

```
<http://www.example.org/index.html> <http://purl.org/dc/elements/1.1/creator>
<http://www.example.org/staffid/85740> .

<http://www.example.org/index.html> <http://www.example.org/terms/creation-date>
```



"August 16, 1999" .

<<http://www.example.org/index.html>> <<http://purl.org/dc/elements/1.1/language>> "en"

Với cách viết này đòi hỏi mỗi URI phải được biết đầy đủ trong một cặp dấu ngoặc nhọn (<>), điều này gây ra một bất lợi là các URI phải viết rất dài như ta đã thấy ở trên. Để thuận lợi hơn, RDF đã cung cấp một cách viết ngắn gọn hơn đó là cách dùng QName( đã được định nghĩa ở trên) mà không cần dùng đến cặp dấu <>.

Ta có thể dùng một số QName sau đây:

prefix `ex:`, namespace URI: <http://www.example.org/> hay

<http://www.example.com/>

prefix `exterm:`, namespace URI: <http://www.example.org/terms/>

prefix `exstaff:`, namespace URI: <http://www.example.org/staffid/>

Và ví dụ trên được viết lại như sau:

```
ex:index.html dc:creator exstaff:85740 .
ex:index.html exterm:creation-date "August 16, 1999" .
ex:index.html dc:language "en" .
```

Và ví dụ trên được viết lại dưới dạng QName như sau:

```
exstaff:85740 exterm:age "27"^^xsd:integer .
ex:index.html exterm:creation-date "1999-08-16"^^xsd:date .
```

Như vậy ta thấy rằng việc khai báo các URI thông qua một QName sẽ rất thuận lợi, nó làm cho các URI này ngắn gọn và dễ hiểu. Hơn nữa khi có một thay đổi về URI thì ta chỉ việc thay đổi ở phần namespace mà chúng ta đã khai báo chứ không cần phải thay đổi tất cả URI trong bộ dữ liệu của mình.

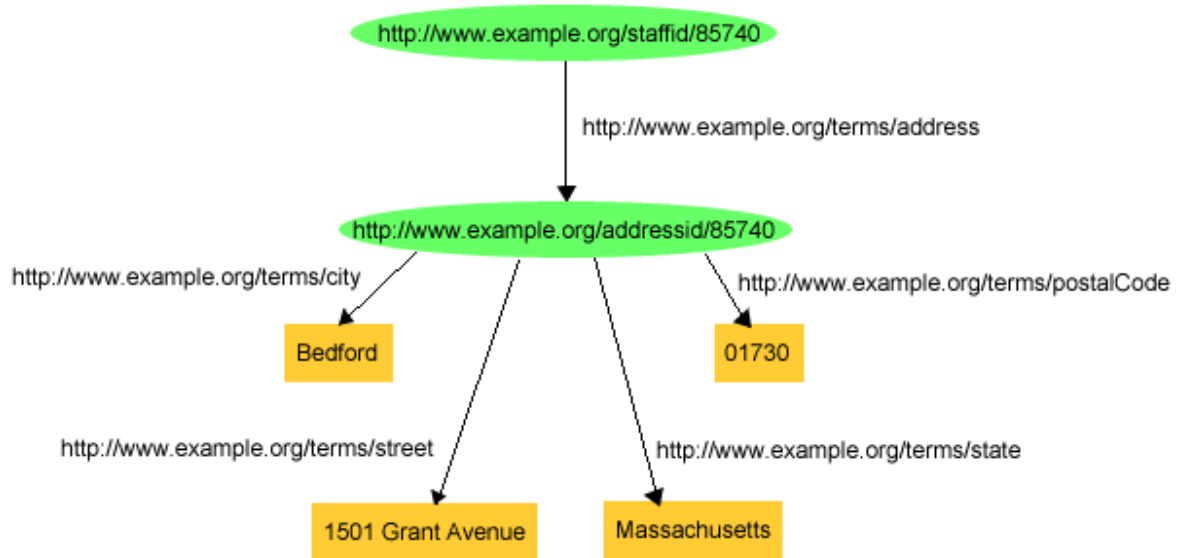
### 3.2.5 Kiểu dữ liệu có cấu trúc và Blank node

Nếu mọi dữ liệu trên Web đều rõ ràng thì thật dễ dàng để khai báo. Tuy nhiên, hầu hết các dữ liệu trên Web đều liên quan đến các cấu trúc phức tạp. Trong ví dụ ... ngày của trang web được tạo ra được khai báo bởi thuộc tính *exterms:creation-date*, với một kiểu dữ liệu là plain literal. Tuy nhiên, giả sử như giá trị của thuộc tính *exterms:creation-date* cần phải lưu trữ thông tin về ngày, tháng, năm độc lập với nhau thì phải biểu diễn vấn đề này như thế nào?. Hay là trong trường hợp thông tin cá nhân của Mr John Smith, giả sử khi mô tả địa chỉ của Mr John thì giá trị của nó sẽ được viết dưới dạng một *plain literal*, như trong tripple sau:

```
exstaff:85740  exterm:s:address  "1501 Grant Avenue, Bedford, Massachusetts 01730" .
```

Tuy nhiên, giả sử rằng địa chỉ này cần phải được chia thành một cấu trúc bao gồm các thành phần riêng biệt như đường, thành phố, bang, và mã vùng. Làm thế nào có thể làm được trong RDF ?

Cấu trúc thông tin như trên sẽ được mô tả trong RDF bằng cách xem thông tin kết hợp ( địa chỉ của Mr John Smith) là một resource, sau đó tạo ra các phát biểu về resource mới này. Vì vậy, trong đồ thị RDF, để tạo ra địa chỉ của Mr John Smith từ các thành phần của nó, một node mới sẽ được tạo ra để biểu diễn khái niệm địa chỉ của Mr John, với một URI để định danh cho nó, ví dụ như <http://www.example.org/addressid/85740> ( viết tắt là *exaddressid:85740*). Các phát biểu RDF ( bao gồm các cung và node) sau đó sẽ được biểu diễn ra trong đồ thị với node mới vừa tạo là subject. Ta hãy xem đồ thị sau:

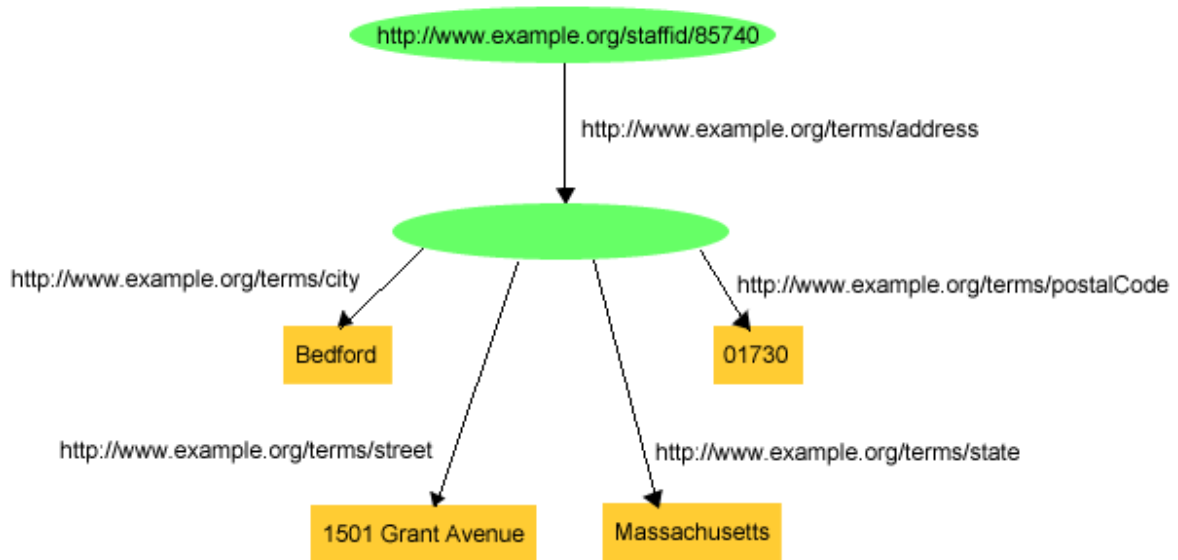


**Hình 3-7 Tạo địa chỉ(address) cho Mr John.**

Hãy viết dưới dạng tripple như sau:

|                   |                    |                       |
|-------------------|--------------------|-----------------------|
| exstaff:85740     | exterms:address    | exaddressid:85740 .   |
| exaddressid:85740 | exterms:street     | "1501 Grant Avenue" . |
| exaddressid:85740 | exterms:city       | "Bedford" .           |
| exaddressid:85740 | exterms:state      | "Massachusetts" .     |
| exaddressid:85740 | exterms:postalCode | "01730" .             |

Cách biểu diễn thông tin có cấu trúc như vậy đòi hỏi phải có một địa chỉ URI *exaddressid:85740* để mô tả khái niệm kết hợp( địa chỉ của Mr John). Những khái niệm như thế này có thể không được sử dụng bởi các thành phần khác trong đồ thị và chính vì vậy định danh này không cần thiết. Chính vì vậy ta nên dùng một node trống (blank node) để biểu diễn node này.



**Hình 3-8 Sử dụng một blank Node.**

Và có thể mô tả dưới dạng tripple như sau:

|               |                    |                     |   |
|---------------|--------------------|---------------------|---|
| exstaff:85740 | externs:address    | ???                 | . |
| ???           | externs:street     | "1501 Grant Avenue" | . |
| ???           | externs:city       | "Bedford"           | . |
| ???           | externs:state      | "Massachusetts"     | . |
| ???           | externs:postalCode | "01730"             | . |

Ở đây ta dùng ??? để chỉ sự hiện diện của blank node. Trong trường hợp một đồ thị sử dụng nhiều blank node, làm thế nào để phân biệt các blank node? Kết quả là bằng cách dùng một định danh (blank node identifiers), có dạng `_:name`, để chỉ sự hiện diện của một blank node. Ví dụ như, trong ví dụ này định danh `_:johnaddress` được sử dụng để chỉ blank node (là địa chỉ của Mr John). Ta có thể viết lại tripple sau:

|               |                 |                     |   |
|---------------|-----------------|---------------------|---|
| exstaff:85740 | externs:address | _:johnaddress       | . |
| _:johnaddress | externs:street  | "1501 Grant Avenue" | . |

```
_:johnaddress exterm:city "Bedford" .
_:johnaddress exterm:state "Massachusetts" .
_:johnaddress exterm:postalCode "01730" .
```

### 3.3 Cấu trúc RDF/XML

#### 3.3.1 Cú pháp RDF/XML cơ bản

Mô hình RDF thể hiện một mô hình ở mức trừu tượng để định nghĩa *metadata*. Cú pháp RDF được dùng để tạo ra và trao đổi *metadata*. Cấu trúc RDF dựa trên cú pháp XML.

Cú pháp cơ bản của RDF có dạng như sau:

```
[1] RDF ::= ['<rdf:RDF>'] description* ['</rdf:RDF>']
[2] description ::= '<rdf:Description' idAboutAttr? '>' propertyElt* '</rdf:Description>'
[3] idAboutAttr ::= idAttr | aboutAttr
[4] aboutAttr ::= 'about="" URI-reference ""'
[5] idAttr ::= 'ID="" IDsymbol ""'
[6] propertyElt ::= '<' propName '>' value '</' propName '>| '<' propName resourceAttr '>'
[7] propName ::= QName
[8] value ::= description | string
[9] resourceAttr ::= 'resource="" tham chiếu URI ""'
[10] QName ::= [ NSprefix ':' ] name
[11] URI-reference ::= string, interpreted per URI
[12] IDsymbol ::= (bất kỳ ID nào hợp lệ nào của XML)
[13] name ::= (bất kỳ tên hợp lệ nào của XML)
[14] NSprefix ::= (bất kỳ tiếp đầu ngữ namespace hợp lệ nào)
[15] string ::= (bất kỳ chuỗi nào)
```

Ví dụ 1: Xét phát biểu

```
ex:index.html exterm:s:creation-date "August 16, 1999" .
```

Một cú pháp RDF/XML để biểu diễn cho phát biểu trên như sau:

```
1. <?xml version="1.0"?>
2. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3.     xmlns:exterm:s="http://www.example.org/terms/">
4. <rdf:Description rdf:about="http://www.example.org/index.html">
5.   <exterm:s:creation-date>August 16, 1999</exterm:s:creation-date>
6. </rdf:Description>
7. </rdf:RDF>
```

Ta có một cách giải thích cụ thể cho cách biểu diễn ở trên như sau:

- Dòng 1: là khai báo XML, cho biết nội dung theo sau dựa trên cú pháp XML và phiên bản XML được dùng.
- Dòng 2 và 3: bắt đầu với thẻ *rdf:RDF*, cho biết rằng nội dung XML tiếp theo (bắt đầu từ đây cho đến *</rdf:RDF>* trong dòng 7) mô tả **RDF**. Từ khóa này xác định tài liệu này được biểu diễn dưới dạng RDF. Tiếp theo là phần khai báo XML namespace được sử dụng trong tài liệu, tùy vào nhu cầu và mục đích sử dụng mà ta có thể dùng các namespace khác nhau cho từng tài liệu.
- Dòng 4,5,6: mô tả những phát biểu RDF. Để mô tả bất kỳ phát biểu nào dạng RDF/XML có thể dùng *rdf:description*, và *rdf:about*, đây chính là subject của phát biểu (about *http://www.example.org/index.html*). Thẻ bắt đầu *rdf:Description* trong dòng 4 cho biết bắt đầu mô tả về một resource, và tiếp tục định danh resource này dùng thuộc tính *rdf:about* để chỉ ra URI của subject resource. Dòng 5 cung cấp 1 phần tử thuộc tính, với QName *exterm:s:creation-date* như là thẻ của nó (phần QName đã được giải thích rõ ở

trên). Nội dung của phần tử thuộc tính này là object của statement, có giá trị là kiểu plain literal “ August 19, 1999 “.

- Dòng 7: cho biết kết thúc của thẻ *rdf:RDF* bắt đầu ở dòng 2 và cũng là thẻ kết thúc của tài liệu RDF .

Như vậy ta đã biết làm thế nào để khai báo một tài liệu RDF. Trong tài liệu RDF, tất cả các phần tử có tiếp đầu ngữ là *rdf* được dùng để mô tả tài nguyên. Chúng sử dụng một không gian tên được tổ chức W3C qui định: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

Ở ví dụ trên ta chỉ mô tả tài liệu cho một phát biểu đơn giản, với một tài liệu gồm nhiều phát biểu , việc mô tả nó cũng thật sự đơn giản:

Ví dụ ta với 2 phát biểu :

|               |                       |                    |
|---------------|-----------------------|--------------------|
| ex:index.html | externs:creation-date | "August 16,1999" . |
| ex:index.html | dc:language           | "en" .             |
| ex:index.html | dc:creator            | exstaff:85740 .    |

Cú pháp RDF được mô tả như sau:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:externs="http://www.example.org/terms/">

  <rdf:Description rdf:about="http://www.example.org/index.html">
    <externs:creation-date>August 16, 1999</externs:creation-date>
  </rdf:Description>

  <rdf:Description rdf:about="http://www.example.org/index.html">
    <dc:language>en</dc:language>
  </rdf:Description>

  <rdf:Description rdf:about="http://www.example.org/index.html">
    <dc:creator rdf:resource="http://www.example.org/staffid/85740"/>
  </rdf:Description>
```

Như vậy với một phát biểu ta có thể dùng từ khóa *rdf:Description* để mô tả cho nó. Tuy nhiên ta thấy 3 phát biểu trên có cùng một subject (<http://www.example.org/index.html>) nên ta có thể gộp 3 phát biểu này lại với nhau dùng một thẻ *rdf:Description* chung như sau:

```
1. <?xml version="1.0"?>
2. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3.     xmlns:dc="http://purl.org/dc/elements/1.1/"
4.     xmlns:exterm="http://www.example.org/terms/">
5.   <rdf:Description rdf:about="http://www.example.org/index.html">
6.     <exterm:creation-date>August 16, 1999</exterm:creation-date>
7.     <dc:language>en</dc:language>
8.     <dc:creator rdf:resource="http://www.example.org/staffid/85740"/>
9.   </rdf:Description>
10. </rdf:RDF>
```

### 3.3.2 RDF Container

Chúng ta thường có nhu cầu mô tả các nhóm (groups) của nhiều đồ vật. Ví dụ như: 1 quyển sách được viết bởi nhiều tác giả, hoặc là danh sách các sinh viên trong 1 khoá học, hoặc là những modules phần mềm trong gói phần mềm v.v... RDF cung cấp nhiều loại (types) và nhiều thuộc tính (properties) tích hợp sẵn giúp chúng ta mô tả được những nhóm như vậy. Tuy nhiên RDF cũng cung cấp một kiểu khai báo là container, dùng để lưu danh sách các tài nguyên hoặc các kiểu giá trị (chuỗi các ký tự, kí số, ...). Những members của 1 container có thể là các resources (gồm cả các blank nodes) hay là các literals.

#### 3.3.2.1 Mô hình Container

RDF định nghĩa 3 loại đối tượng container: *Bag*, *Sequence*, và *Alternative*.



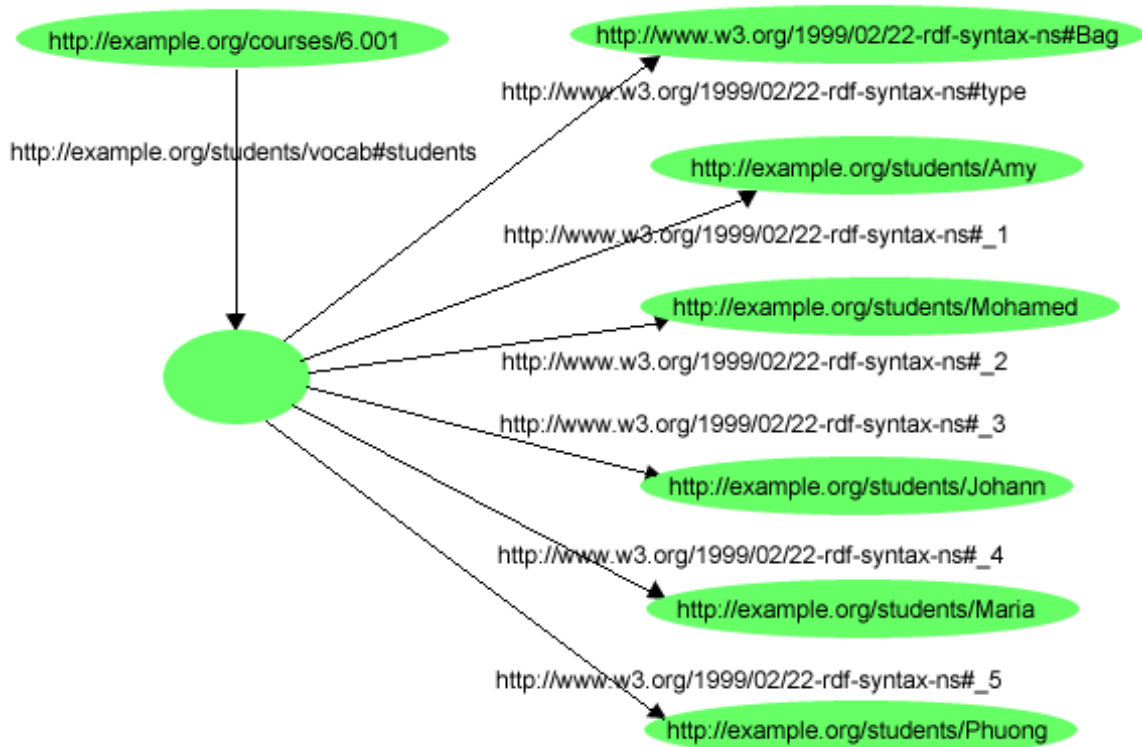
- **Bag** là danh sách không có thứ tự của các tài nguyên hoặc các giá trị. Nó được dùng để khai báo thuộc tính có nhiều giá trị và thứ tự của những giá trị này không cần quan tâm đến. *Bag* cho phép những giá trị có thể trùng lặp nhau.
- **Sequence** là danh sách có thứ tự của các tài nguyên hoặc các giá trị. Nó được dùng để khai báo thuộc tính có nhiều giá trị và thứ tự của những giá trị này cần được quan tâm đến. Chẳng hạn như dùng *Sequence* để lưu trữ các giá trị theo thứ tự bảng chữ cái. *Sequence* cho phép những giá trị có thể trùng lặp nhau.
- **Alternative** là một danh sách các tài nguyên hoặc các giá trị, được dùng để biểu diễn các giá trị lựa chọn của một thuộc tính.

Để biểu diễn một tập hợp các tài nguyên, RDF dùng một tài nguyên mới để xác định một tập tài nguyên cần được biểu diễn. Tài nguyên mới này phải được khai báo như là một thể hiện của một trong các loại đối tượng *container* được đề cập ở trên. Thuộc tính *type* được dùng để khai báo loại đối tượng *container* được sử dụng. Mỗi quan hệ thành viên giữa *container* và các tài nguyên của tập hợp được xác định bằng những tên đơn giản như : “\_1”, “\_2”, “\_3” ...

Ví dụ 1: Xét phát biểu sau:

“Course 6.001 has the students Amy, Mohamed, Johann, Maria, and Phuong”

Có mô hình RDF là:

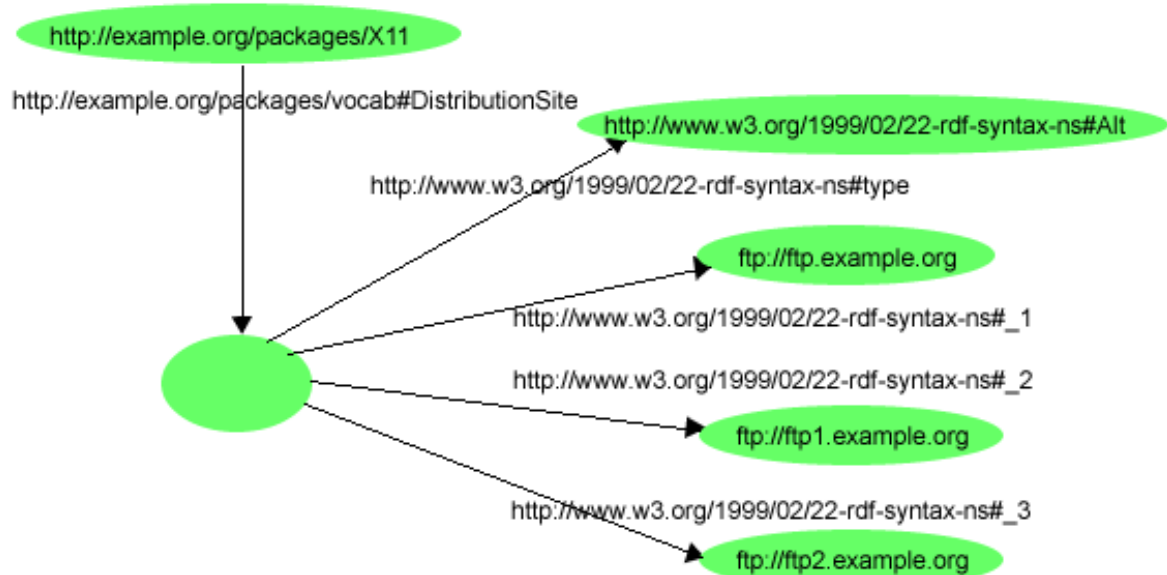


Hình 3-9 Mô tả 1 bag container đơn giản.

Ví dụ 2: Xét phát biểu sau:

"The source code for X11 may be found at ftp.example.org, ftp1.example.org, or ftp2.example.org"

Ta dùng `rdf:Alt` để mô tả như hình sau :



Hình 3-10 Mô tả 1 Alt container đơn giản.

### 3.3.2.2 Cú pháp Container

Cú pháp một RDF *container* có dạng như sau:

```
[16] container ::= sequence | bag | alternative
[17] sequence ::= '<rdf:Seq' idAttr? '>' member* '</rdf:Seq>'
[18] bag ::= '<rdf:Bag' idAttr? '>' member* '</rdf:Bag>'
[19] alternative ::= '<rdf:Alt' idAttr? '>' member+ '</rdf:Alt>'
[20] member ::= referencedItem | inlineltem
[21] referencedItem ::= '<rdf:li' resourceAttr '/>'
[22] inlineltem ::= '<rdf:li>' value '</rdf:li>'
```

*Container* được dùng bất kỳ nơi nào mà một *description* có thể được dùng:

```
[1a] RDF ::= '<rdf:RDF>' obj* '</rdf:RDF>'
[8a] value ::= obj | string
[23] obj ::= description | container
```

vì tài liệu RDF theo như [1a] được dùng để mô tả nhiều *obj*, mà *obj* có thể là *description* hoặc *container*.

Ta biểu diễn 2 ví dụ trên dưới dạng cú pháp RDF/XML như sau:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://example.org/students/vocab#">
  <rdf:Description rdf:about="http://example.org/courses/6.001">
    <s:students>
      <rdf:Bag>
        <rdf:li rdf:resource="http://example.org/students/Amy"/>
        <rdf:li rdf:resource="http://example.org/students/Mohamed"/>
        <rdf:li rdf:resource="http://example.org/students/Johann"/>
        <rdf:li rdf:resource="http://example.org/students/Maria"/>
        <rdf:li rdf:resource="http://example.org/students/Phuong"/>
      </rdf:Bag>
    </s:students>
  </rdf:Description>
</rdf:RDF>
```

Ví dụ trên cho thấy rằng RDF/XML cung cấp thuộc tính `rdf:li` để tránh việc phải đánh từng số cho mỗi thuộc tính thành viên. Những thuộc tính được đánh số `rdf:_1`, `rdf:_2`, ... được phát sinh từ những yếu tố `rdf:li` khi hình thành lược đồ tương ứng. Chú ý rằng việc sử dụng `<rdf:Bag>` được đặt bên trong `<s:students>`. Vì không có URI được ghi rõ nên Bag là 1 blank node. Việc đặt nó trong thuộc tính `<s:students>` là 1 cách viết tắt để cho biết rằng blank node là giá trị của thuộc tính này.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://example.org/packages/vocab#">
```

```
<rdf:Description rdf:about="http://example.org/packages/X11">
  <s:DistributionSite>
    <rdf:Alt>
      <rdf:li rdf:resource="ftp://ftp.example.org"/>
      <rdf:li rdf:resource="ftp://ftp1.example.org"/>
      <rdf:li rdf:resource="ftp://ftp2.example.org"/>
    </rdf:Alt>
  </s:DistributionSite>
</rdf:Description>
</rdf:RDF>
```

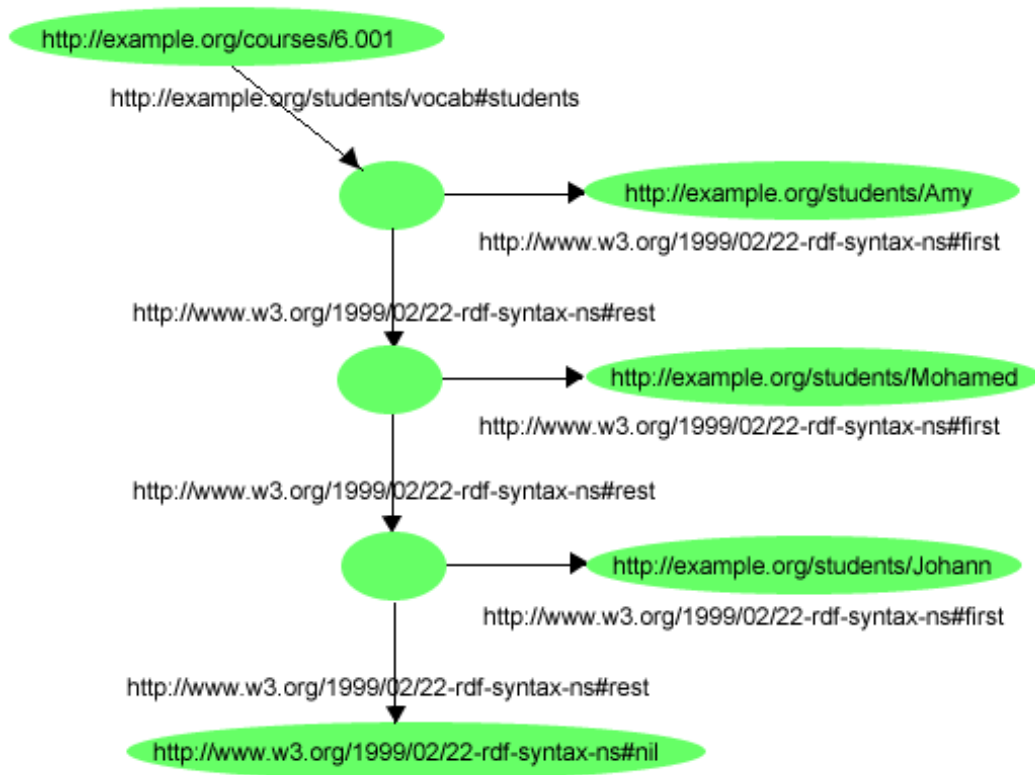
### **3.4 RDF Collection**

Cũng dùng trong mục đích định nghĩa ra những tập hợp. Nhưng RDF container là định nghĩa một tập hợp mở, vì nó không bảo rằng không còn có thêm những thành viên mới nữa. Trong khi đó, RDF collection cho phép khai báo một tập hợp đóng. Cấu trúc của RDF Collection tương tự như một danh sách, có phần tử đầu (*rdf:first*), phần tử kế (*rdf:rest*) và phần tử cuối (*rdf:nil*).

Xét phát biểu sau:

"The students in course 6.001 are Amy, Mohamed, and Johann"

Được biểu diễn dưới đồ thị như sau:



Hình 3-11 Mô tả một RDF : Collection.

Ở đồ thị trên ta thấy với mỗi thành viên của danh sách, ví dụ như *s:Amy*, là subject của thuộc tính *rdf:first* mà subject của nó là một resource( một blank node đối với ví dụ này) mà nó biểu diễn như một danh sách. Danh sách này liên kết với phần còn lại của nó bằng thuộc tính *rdf:rest*. Phần tử cuối cùng của danh sách được chỉ bởi thuộc tính *rdf:rest* có resource là *rdf:nil*( là một danh sách trống).

Ví dụ trên được biểu diễn dưới cú pháp RDF/XML như sau:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://example.org/students/vocab#">

<rdf:Description rdf:about="http://example.org/courses/6.001">
  <s:students rdf:nodeID="sch1"/>
</rdf:Description>
```

```

<rdf:Description rdf:nodeID="sch1">
  <rdf:first rdf:resource="http://example.org/students/Amy"/>
  <rdf:rest rdf:nodeID="sch2"/>
</rdf:Description>

<rdf:Description rdf:nodeID="sch2">
  <rdf:first rdf:resource="http://example.org/students/Mohamed"/>
  <rdf:rest rdf:nodeID="sch3"/>
</rdf:Description>

<rdf:Description rdf:nodeID="sch3">
  <rdf:first rdf:resource="http://example.org/students/Johann"/>
  <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
</rdf:Description>
</rdf:RDF>
    
```

Tuy nhiên RDF / XML cung cấp một cách định nghĩa để mô tả một tập hợp, bằng cách sử dụng một thuộc tính có attribute là `rdf:parseType="Collection"`. Ta có thể viết lại ví dụ trên một cách ngắn gọn hơn như sau:

```

<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://example.org/students/vocab#">

  <rdf:Description rdf:about="http://example.org/courses/6.001">
    <s:students rdf:parseType="Collection">
      <rdf:Description rdf:about="http://example.org/students/Amy"/>
      <rdf:Description rdf:about="http://example.org/students/Mohamed"/>
      <rdf:Description rdf:about="http://example.org/students/Johann"/>
    </s:students>
  </rdf:Description>
</rdf:RDF>
    
```

## 3.5 RDF Schema

### 3.5.1 Giới thiệu

RDF cung cấp một cách để mô tả các phát biểu đơn giản về các resource, sử dụng các thuộc tính và giá trị đã được định nghĩa trước. Tuy nhiên, nhu cầu của con người đòi hỏi phải có một khả năng để tự định nghĩa các thuật ngữ mà họ muốn dùng trong các phát biểu đó. Ví dụ như, công ty *example.com* trong ví dụ.. muốn mô tả các lớp như *exterms:Tent*, và sử dụng thuộc tính *exterms:model*, *exterms:weighInKg* và *exterms:packedSize* để mô tả chúng hoặc là một ứng dụng nào đó muốn mô tả các lớp như *ex3:Person*, *ex3:Company* và các thuộc tính như *ex3:age*, *ex3:jobTitle*, *ex3:stockSymbol*, *ex3:numberOfEmployees*...

Tuy nhiên mô hình dữ liệu RDF không cung cấp những cơ chế cho việc khai báo các thuộc tính, cũng như không cung cấp bất kỳ cơ chế nào để có thể định nghĩa ra những quan hệ giữa các thuộc tính và các tài nguyên. Đó sẽ là vai trò của *RDF schema*, hay nói cách khác *RDF schema* được dùng để định nghĩa các tài nguyên (các lớp trong *RDF schema*) và thuộc tính (thuộc tính trong *RDF schema*) cũng như các quan hệ qua lại giữa tài nguyên với tài nguyên, giữa thuộc tính với thuộc tính, và giữ tài nguyên với thuộc tính.

Tương tự XML schema, RDF schema là một tập những từ khoá mà qua đó RDF schema cho phép người dùng định nghĩa bộ từ vựng (*resource*, *property*) cụ thể cho dữ liệu RDF (ví dụ như: *hasName*, *hasPrice*, *authorOf*, ...) và định nghĩa các quan hệ của nó đến các đối tượng liên quan. Chẳng hạn như từ *hasName* ta định nghĩa quan hệ của nó trên hai đối tượng: '*http://www.w3c.org/employee/id1321*' và "*Jim Lerner*" như sau:

*hasName*('http://www.w3c.org/employee/id1321', "*Jim Lerner*")



### 3.5.2 Định nghĩa class (lớp)

Các resource trên Web có thể chia thành các nhóm gọi là class. Các thành viên ( member) của nhóm được xem như là thể hiện của lớp đó. Class cũng chính là resource. Nó được nhận ra thông qua các định danh URI và có thể được mô tả bằng cách sử dụng các RDF properties. Thuộc tính rdfs:type được sử dụng để chỉ ra một resource là một thể hiện của một class.

Ví dụ như công ty example.org muốn sử dụng RDF để cung cấp thông tin về những loại xe ( motor vehicles) khác nhau. Đầu tiên công ty này phải sử dụng một lớp để biểu diễn lớp xe( motor vehicles). Trong RDF Schema bất kỳ tài nguyên nào có thuộc tính là rdfs:type và có giá trị là resource rdfs:Class gọi là một class. Vì vậy ta định nghĩa lớp motor vehicle như sau:

```
ex:MotorVehicle rdfs:type rdfs:Class .
```

Tương tự như vậy công ty này phải mô tả các lớp xe còn lại là một class

```
ex:PassengerVehicle rdfs:type rdfs:Class .  
ex:Van rdfs:type rdfs:Class .  
ex:Truck rdfs:type rdfs:Class .  
ex:MiniVan rdfs:type rdfs:Class .
```

Tới đây ta chỉ mới có các lớp độc lập, cái công ty này cần đến là mô tả sự liên quan giữa các lớp với nhau. Lớp PassengerVehicle, Van , Truck sẽ là lớp con của lớp MotorVehicle và lớp MiniVan sẽ là lớp con của lớp Van và lớp PassengerVehicle. Chúng ta sẽ dùng thuộc tính rdfs:subClassOf để mô tả thông tin này:

```
ex:PassengerVehicle rdfs:subClassOf ex:MotorVehicle .  
ex:Van rdfs:subClassOf ex:MotorVehicle .  
ex:Truck rdfs:subClassOf ex:MotorVehicle .  
ex:MiniVan rdfs:subClassOf ex:Van .  
ex:MiniVan rdfs:subClassOf ex:PassengerVehicle .
```

Một đồ thị để biểu diễn thông tin này như sau:



**Hình 3-12** Mô tả lớp và các lớp con.

Và cú pháp RDF/XML để mô tả cho ví dụ trên như sau:

```

<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [!ENTITY xsd "http://www.w3.org/2001/XMLSchema#"]>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://example.org/schemas/vehicles">

  <rdf:Description rdf:ID="MotorVehicle">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  </rdf:Description>

  <rdf:Description rdf:ID="PassengerVehicle">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
  </rdf:Description>
  </rdf:RDF>
  
```

```
</rdf:Description>

<rdf:Description rdf:ID="Truck">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdf:Description>

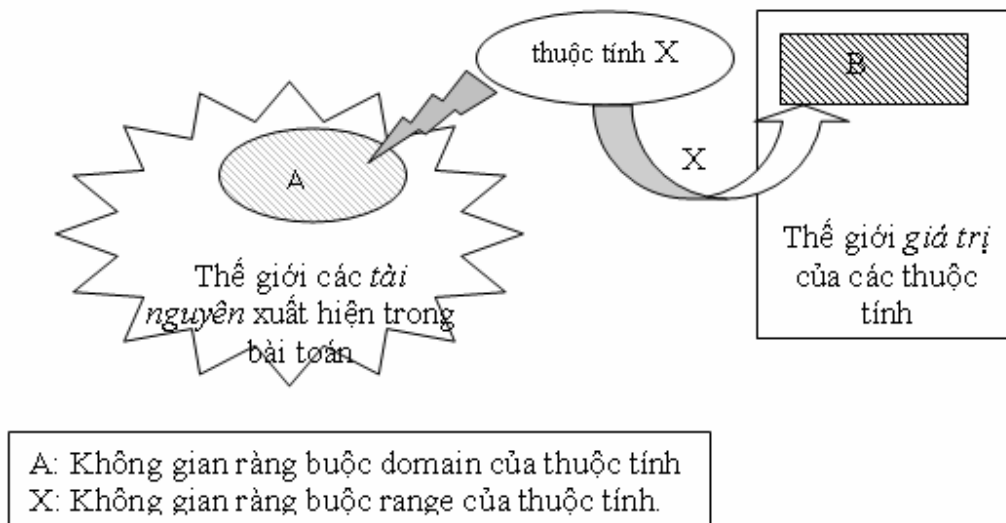
<rdf:Description rdf:ID="Van">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdf:Description>

<rdf:Description rdf:ID="MiniVan">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdfs:subClassOf rdf:resource="#Van"/>
  <rdfs:subClassOf rdf:resource="#PassengerVehicle"/>
</rdf:Description>

</rdf:RDF>
```

### 3.5.3 Định nghĩa property( thuộc tính)

RDF Schema cũng cung cấp một bộ từ vựng để mô tả làm thế nào mà các thuộc tính(property) và lớp(class) có thể được sử dụng cùng với nhau trong dữ liệu RDF . Thuộc tính quan trọng nhất được sử dụng trong trường hợp này là *rdfs:range* và *rdfs:domain*.



**Hình 3-13** Không gian domain và range của thuộc tính.

### 3.5.3.1 Cách sử dụng `rdfs:range`

Thuộc tính `rdfs:range` dùng để chỉ giá trị của một thuộc tính là thể hiện của một lớp. Ví dụ như, nếu công ty `example.org` muốn chỉ rằng thuộc tính `ex:author` có giá trị là thể hiện của lớp `ex:Person`, họ sẽ viết phát biểu RDF như sau:

```
ex:Person    rdfs:type    rdfs:Class .
ex:author    rdfs:type    rdf:Property .
ex:author    rdfs:range   ex:Person .
```

Phát biểu này chỉ rằng `ex:Person` là một lớp, `ex:author` là một thuộc tính, và thuộc tính `ex:author` có object là thể hiện của lớp `ex:Person`.

Tuy nhiên một thuộc tính có thể có nhiều `rdfs:range`, như ví dụ sau:

```
ex:hasMother rdfs:range ex:Female .
ex:hasMother rdfs:range ex:Person .
```

Thuộc tính `rdfs:range` cũng có thể được sử dụng để chỉ ra giá trị của kiểu typed literal

```
ex:age    rdf:type    rdf:Property .
```

```
ex:age rdfs:range xsd:integer .
```

### 3.5.3.2 Cách sử dụng rdfs:domain

Thuộc tính rdfs:domain được sử dụng để chỉ rằng một thuộc tính là thuộc tính của một lớp nào đó. Ví dụ như, công ty *example.org* muốn thuộc tính *ex:author* là thuộc tính của lớp *ex:Book*, ta có phát biểu sau:

```
ex:Book rdfs:type rdfs:Class .
ex:author rdfs:type rdf:Property .
ex:author rdfs:domain ex:Book .
```

Và một thuộc tính cũng có nhiều thuộc tính rdfs:domain

```
exterms:weight rdfs:domain ex:Book .
exterms:weight rdfs:domain ex:MotorVehicle .
```

### 3.5.3.3 Ví dụ

Dưới đây là một RDF Schema đầy đủ mô tả về motor vehicle của công ty *example.org*

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://example.org/schemas/vehicles">

  <rdfs:Class rdf:ID="MotorVehicle"/>

  <rdfs:Class rdf:ID="PassengerVehicle">
    <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
  </rdfs:Class>
```

```
<rdfs:Class rdf:ID="Truck">
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdfs:Class>

<rdfs:Class rdf:ID="Van">
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdfs:Class>

<rdfs:Class rdf:ID="MiniVan">
  <rdfs:subClassOf rdf:resource="#Van"/>
  <rdfs:subClassOf rdf:resource="#PassengerVehicle"/>
</rdfs:Class>

<rdfs:Class rdf:ID="Person"/>

<rdfs:Datatype rdf:about="&xsd;integer"/>

<rdf:Property rdf:ID="registeredTo">
  <rdfs:domain rdf:resource="#MotorVehicle"/>
  <rdfs:range rdf:resource="#Person"/>
</rdf:Property>

<rdf:Property rdf:ID="rearSeatLegRoom">
  <rdfs:domain rdf:resource="#PassengerVehicle"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>

<rdf:Property rdf:ID="driver">
  <rdfs:domain rdf:resource="#MotorVehicle"/>
</rdf:Property>

<rdf:Property rdf:ID="primaryDriver">
  <rdfs:subPropertyOf rdf:resource="#driver"/>
</rdf:Property>

</rdf:RDF>
```

### 3.5.3.4 Tổng kết

óm lại tập từ khóa của RDF schema có thể được phân thành hai tập cơ bản:

- *RDF class*: tập từ khóa liên quan đến các lớp.
- *RDF property*: tập từ khóa liên quan đến các thuộc tính.

**Bảng tóm tắt các từ khóa RDF Class:**

| RDF Class              | Ý nghĩa  |
|------------------------|--|
| <i>rdfs:Resource</i>   | Lớp tài nguyên. Tất cả những gì được mô tả bằng biểu thức RDF đều là tài nguyên và là thể hiện của lớp <i>rdfs:Resource</i> .  |
| <i>rdfs:Literal</i>    | Lớp những giá trị thô (literal), ví dụ: một chuỗi kí tự, hoặc ký số... là những giá trị thô, nó chỉ thể hiện nghĩa đen của nó. |
| <i>rdfs:XMLLiteral</i> | Lớp giá trị thô của XML.   |
| <i>rdfs:Class</i>      | Lớp nói chung, dùng để định nghĩa các khái niệm.   |
| <i>rdf:Property</i>    | Lớp các thuộc tính của RDF, dùng để mô tả đặc điểm của các thể hiện <i>rdf:Resource</i> .                                      |
| <i>rdfs:Datatype</i>   | Lớp các kiểu dữ liệu của RDF.  |
| <i>rdf:Statement</i>   | Lớp các phát biểu của RDF.   |
| <i>rdf:Bag</i>         | Lớp các container không sắp thứ tự.  |
| <i>rdf:Seq</i>         | Lớp các container được sắp thứ tự.   |
| <i>rdf:Alt</i>         | Lớp các container thay thế.  |
| <i>rdfs:Container</i>  | Lớp các RDF container.   |

|   |  |
|---|--|
| <i>rdfs:ContainerMembershipProperty</i> | Lớp các thuộc tính thành viên của một <i>container</i> . Nó được mô tả dưới dạng: <i>rdf:_1</i> , <i>rdf_2</i> , ... |
| <i>rdf:List</i>                         | Lớp các danh sách (list) của RDF.  |

**Bảng tóm tắt các từ khóa RDF Property:**

| <b>RDF Property</b>       | <b>Ý nghĩa</b>  | <b>Domain</b>         | <b>range</b>         |
|---------------------------|---|-----------------------|----------------------|
| <i>rdf:type</i>           | Chỉ mỗi quan hệ là thể hiện của một lớp.                            | <i>rdfs:Resource</i>  | <i>rdfs:Class</i>    |
| <i>rdfs:subClassOf</i>    | Chỉ mỗi quan hệ lớp con của một lớp.                                | <i>rdfs:Class</i>     | <i>rdfs:Class</i>    |
| <i>rdfs:subPropertyOf</i> | Chỉ mỗi quan hệ đặc biệt hoá của một thuộc tính.                    | <i>rdf:Property</i>   | <i>rdf:Property</i>  |
| <i>rdfs:domain</i>        | Không gian giá trị được áp dụng cho một thuộc tính nào đó.          | <i>rdf:Property</i>   | <i>rdfs:Class</i>    |
| <i>rdfs:range</i>         | Miền giá trị cho một thuộc tính nào đó.                             | <i>rdf:Property</i>   | <i>rdfs:Class</i>    |
| <i>rdfs:label</i>         | Tên của resource, mục đích để con người đọc có thể đọc được.        | <i>rdfs:Resource</i>  | <i>rdfs:Literal</i>  |
| <i>rdfs:comment</i>       | Mô tả về tài nguyên.  | <i>rdfs:Resource</i>  | <i>rdfs:Literal</i>  |
| <i>rdfs:member</i>        | Thành viên của một container.                                       | <i>rdfs:Container</i> | <i>rdfs:Resource</i> |
| <i>rdf:first</i>          | Phần tử đầu tiên trong danh sách (list) RDF. Thường được gọi là đầu | <i>rdf:List</i>       | <i>rdfs:Resource</i> |



|                         |  |               |                      |
|-------------------------|--|---------------|----------------------|
| <i>rdf:rest</i>         | Phần còn lại của danh sách RDF sau phần tử đầu tiên.       | rdf:List      | rdf:List             |
| <i>rdfs:seeAlso</i>     | Thông tin cung cấp thêm về tài nguyên cần quan tâm.        | rdfs:Resource | rdfs:Resource        |
| <i>rdfs:isDefinedBy</i> | Chỉ ra <i>namespace</i> của tài nguyên.                    | rdfs:Resource | rdfs:Resource        |
| <i>rdf:value</i>        | Giá trị của thuộc tính khi nó là tài nguyên được cấu trúc. | rdfs:Resource | <i>not specified</i> |
| <i>rdf:subject</i>      | <i>Subject</i> của một phát biểu RDF.                      | rdf:Statement | rdfs:Resource        |
| <i>rdf:predicate</i>    | <i>Predicate</i> của một phát biểu RDF.                    | rdf:Statement | rdf:Property         |
| <i>rdf:object</i>       | <i>Object</i> của một phát biểu RDF.                       | rdf:Statement | <i>not specified</i> |

### 3.6 FOAF : Sự mở rộng của RDF

FOAF(Friend Of A Friend) cung cấp một bộ từ vựng(ontology) cơ bản để mô tả thông tin về con người(people), nhóm(group), tổ chức(organization) và những loại khác và các thông tin liên quan. Ví dụ như để lưu thông tin của một người ta cần có những thông tin sau: tên, tuổi, nghề nghiệp, giới tính, ngày tháng năm sinh, hình ảnh, có những tài liệu nào.v.v. thì trong FOAF đã cung cấp sẵn cho chúng ta một bộ từ vựng cơ bản để có thể mô tả những thông tin này.

Xét một ví dụ sử dụng FOAF

```
<foaf:Person>
  <foaf:name>Dan Brickley</foaf:name>
  <foaf:mbox rdf:resource=mailto:db@yahoo.com/>
  <foaf:homepage rdf:resource="http://rdfweb.org/people/danbri/" />
  <foaf:img rdf:resource="http://rdfweb.org/people/danbri/mugshot/danbri-small.jpeg" />
</foaf:Person>
```

Tài liệu trên mô tả một người (*foaf:Person*) có tên (*foaf:name*) là ‘Dan Brickley’ và có địa chỉ mail (*foaf:mbox*) là [db@yahoo.com](mailto:db@yahoo.com), có một trang web (*foaf:homepage*) homepage là <http://rdfweb.org/people/danbri/> và có một hình ảnh (*foaf:image*) lưu tại địa chỉ <http://rdfweb.org/people/danbri/mugshot/danbri-small.jpeg>.

Bộ từ vựng (ontology) FOAF được định nghĩa thông qua ngôn ngữ OWL, được định danh bởi namespace có URI là 'http://xmlns.com/foaf/0.1/' và có QName là *foaf*:

**Các lớp được định nghĩa trong FOAF**

| FOAF Class                        | Ý nghĩa  |
|-----------------------------------|--|
| <b><i>foaf:Agent</i></b>          | Lớp tác nhân( ví dụ con người, nhóm, tổ chức...)   |
| <b><i>foaf:Document</i></b>       | Lớp tài liệu. Các loại tư liệu trên Web.   |
| <b><i>foaf:Group</i></b>          | Lớp nhóm người. Biểu diễn cho tập một nhóm người.<br>Ví dụ :<br><pre> &lt;foaf:Group&gt;   &lt;foaf:name&gt;ILRT staff&lt;/foaf:name&gt;   &lt;foaf:member&gt;     &lt;foaf:Person&gt;       &lt;foaf:name&gt;Libby Miller&lt;/foaf:name&gt;       &lt;foaf:homepage rdf:resource="http://ilrt.org/people/libby/"&gt;       &lt;foaf:workplaceHomepage rdf:resource="http://www.ilrt.bris.ac.uk/"&gt;     &lt;/foaf:Person&gt;   &lt;/foaf:member&gt; &lt;/foaf:Group&gt; </pre> |
| <b><i>foaf:Image</i></b>          | Lớp hình ảnh   |
| <b><i>foaf:Online Account</i></b> | Lớp tài khoản trên mạng.   |
| <b><i>foaf:OnlineChat</i></b>     | Lớp tài khoản chat trên mạng.  |

|                                     |  |
|-------------------------------------|--|
| <b>Account</b>                      | <p>Ví dụ:</p> <pre>&lt;foaf:Person&gt;   &lt;foaf:name&gt;Dan Brickley&lt;/foaf:name&gt;   &lt;foaf:holdsAccount&gt;     &lt;foaf:OnlineAccount&gt;       &lt;rdf:type rdf:resource="http://xmlns.com/foaf/0.1/OnlineChatAccount"/&gt;       &lt;foaf:accountServiceHomepage rdf:resource="http://www.freenode.net/irc_servers.shtml"/&gt;       &lt;foaf:accountName&gt;danbri&lt;/foaf:accountName&gt;     &lt;/foaf:OnlineAccount&gt;   &lt;/foaf:holdsAccount&gt; &lt;/foaf:Person&gt;</pre> |
| <b>foaf:OnlineEcommerceAccount</b>  | Lớp tài khoản thương mại trên mạng.  |
| <b>foaf:OnlineGamingAccount</b>     | Lớp tài khoản game trên mạng.  |
| <b>foaf:Organization</b>            | Lớp tổ chức.   |
| <b>foaf:Person</b>                  | Lớp con người.   |
| <b>foaf:PersonalProfileDocument</b> | Lớp tài liệu cá nhân.  |
| <b>foaf:Project</b>                 | Lớp dự án.   |

**Các thuộc tính được định nghĩa trong FOAF**

| RDF Property               | Ý nghĩa                                    | Domain             | Range   |
|----------------------------|--|--------------------|---|
| <b>foaf:accountName</b>    | Chỉ tên liên quan đến tài khoản trên mạng. | foaf:OnlineAccount | http://www.w3.org/2000/01/rdf-schema#Literal  |
| <b>foaf:aimChatID</b>      | Chỉ một aim ChatID                         | foaf:Agent         | http://www.w3.org/2000/01/rdf-schema#Literal  |
| <b>foaf:currentProject</b> | Chỉ một dự án hiện tại của một người.      | foaf:Person        | http://www.w3.org/2000/01/rdf-schema#Resource |

|                          |   |   |   |
|--------------------------|---|---|---|
| <b>foaf:depiction</b>    | Chỉ sự miêu tả về một cái gì đó.                    | <a href="http://www.w3.org/2000/01/rdf-schema#Resource">http://www.w3.org/2000/01/rdf-schema#Resource</a> | rdfs:Class  |
| <b>foaf:depicts</b>      | Chỉ một vật được miêu tả .                          | foaf:Image  | <a href="http://www.w3.org/2000/01/rdf-schema#Resource">http://www.w3.org/2000/01/rdf-schema#Resource</a> |
| <b>foaf:family_name</b>  | Chỉ tên gia đình.                                   | foaf:Person   | <a href="http://www.w3.org/2000/01/rdf-schema#Literal">http://www.w3.org/2000/01/rdf-schema#Literal</a>   |
| <b>foaf:firstName</b>    | Chỉ tên first name của một người.                   | foaf:Person   | <a href="http://www.w3.org/2000/01/rdf-schema#Literal">http://www.w3.org/2000/01/rdf-schema#Literal</a>   |
| <b>foaf:fundedBy</b>     | Một tổ chức làm chủ một project hay person          | <a href="http://www.w3.org/2000/01/rdf-schema#Resource">http://www.w3.org/2000/01/rdf-schema#Resource</a> | <a href="http://www.w3.org/2000/01/rdf-schema#Resource">http://www.w3.org/2000/01/rdf-schema#Resource</a> |
| <b>foaf:geekcode</b>     | Chỉ một số geekcode cho một người                   | foaf:Person   | <a href="http://www.w3.org/2000/01/rdf-schema#Literal">http://www.w3.org/2000/01/rdf-schema#Literal</a>   |
| <b>foaf:gender</b>       | Chỉ giới tính của một người                         | foaf:Person   | <a href="http://www.w3.org/2000/01/rdf-schema#Literal">http://www.w3.org/2000/01/rdf-schema#Literal</a>   |
| <b>foaf:givenname</b>    | Chỉ given name của một người.                       | foaf:Person   | <a href="http://www.w3.org/2000/01/rdf-schema#Literal">http://www.w3.org/2000/01/rdf-schema#Literal</a>   |
| <b>foaf:holdsAccount</b> | Chỉ ra một account của một tác nhân.                | foaf:Agent  | foaf:OnlineAccount  |
| <b>foaf:homepage</b>     | Chỉ ra một địa chỉ trang web của tài nguyên nào đó. | <a href="http://www.w3.org/2000/01/rdf-schema#Resource">http://www.w3.org/2000/01/rdf-schema#Resource</a> | foaf:Document   |
| <b>foaf:icqChatID</b>    | Chỉ ra một ICQ chatID.                              | foaf:Agent  | <a href="http://www.w3.org/2000/01/rdf-schema#Literal">http://www.w3.org/2000/01/rdf-schema#Literal</a>   |
| <b>foaf:img</b>          | Hình ảnh dùng để mô tả con người.                   | foaf:Person   | foaf:Image  |
| <b>foaf:interest</b>     | Chỉ ra một trang web mà người nào đó quan tâm đến.  | foaf:Person   | foaf:Document   |
| <b>foaf:jabberID</b>     | Chỉ ra một jabber ID cho một tài nguyên.            | foaf:Agent  | <a href="http://www.w3.org/2000/01/rdf-schema#Literal">http://www.w3.org/2000/01/rdf-schema#Literal</a>   |
| <b>foaf:knows</b>        | Chỉ ra mối quan hệ                                  | foaf:Person   | foaf:Person   |

|                              |   |   |   |
|------------------------------|---|---|---|
|                              | giữa người với người.   |   |   |
| <b>foaf:logo</b>             | Một logo để mô tả cho một tài nguyên.   | <a href="http://www.w3.org/2000/01/rdf-schema#Resource">http://www.w3.org/2000/01/rdf-schema#Resource</a> | <a href="http://www.w3.org/2000/01/rdf-schema#Resource">http://www.w3.org/2000/01/rdf-schema#Resource</a> |
| <b>foaf:made</b>             | Chỉ ra một tài nguyên được làm bởi ai đó.   | foaf:Agent  | <a href="http://www.w3.org/2000/01/rdf-schema#Resource">http://www.w3.org/2000/01/rdf-schema#Resource</a> |
| <b>foaf:maker</b>            | Chỉ ra ai đó đã tạo ra một tài nguyên nào đó.                                     | <a href="http://www.w3.org/2000/01/rdf-schema#Resource">http://www.w3.org/2000/01/rdf-schema#Resource</a> | foaf:Agent  |
| <b>foaf:mbox</b>             | Chỉ ra một địa chỉ mail của một tác nhân.   | foaf:Agent  | <a href="http://www.w3.org/2000/01/rdf-schema#Resource">http://www.w3.org/2000/01/rdf-schema#Resource</a> |
| <b>foaf:mbox_sha1sum</b>     | Chỉ ra số sha1sum của địa chỉ mail( mỗi người chỉ có một số này ứng với một mail) | foaf:Agent  | <a href="http://www.w3.org/2000/01/rdf-schema#Literal">http://www.w3.org/2000/01/rdf-schema#Literal</a>   |
| <b>foaf:member</b>           | Chỉ ra thành viên của một nhóm.   | foaf:Group  | foaf:Agent  |
| <b>foaf:membership Class</b> | Chỉ ra lớp các cá nhân là thành viên của nhóm.                                    | foaf:Group  | <a href="http://www.w3.org/2000/01/rdf-schema#Class">http://www.w3.org/2000/01/rdf-schema#Class.</a>      |
| <b>foaf:msnChatID</b>        | Chỉ một MSN chat ID   | foaf:Agent  | <a href="http://www.w3.org/2000/01/rdf-schema#Literal">http://www.w3.org/2000/01/rdf-schema#Literal</a>   |
| <b>foaf:name</b>             | Chỉ một tên của một tài nguyên.   | <a href="http://www.w3.org/2000/01/rdf-schema#Resource">http://www.w3.org/2000/01/rdf-schema#Resource</a> | <a href="http://www.w3.org/2000/01/rdf-schema#Literal">http://www.w3.org/2000/01/rdf-schema#Literal</a>   |
| <b>foaf:nick</b>             | Chỉ ra một nick của một người nào đó.   | foaf:Person   | <a href="http://www.w3.org/2000/01/rdf-schema#Literal">http://www.w3.org/2000/01/rdf-schema#Literal</a>   |
| <b>foaf:page</b>             | Chỉ ra một trang web của một tài nguyên.  | <a href="http://www.w3.org/2000/01/rdf-schema#Resource">http://www.w3.org/2000/01/rdf-schema#Resource</a> | foaf:Document.  |
| <b>foaf:pastProject</b>      | Chỉ ra một dự án  | foaf:Person   | <a href="http://www.w3.org/2000/">http://www.w3.org/2000/</a>   |

|                            |   |   |   |
|----------------------------|---|---|---|
|                            | trong quá khứ của một người nào đó.           |   | 01/rdf-schema#Resource                        |
| <b>foaf:phone</b>          | Chỉ ra số phone của một tác nhân nào đó.      | foaf:Agent                                    | http://www.w3.org/2000/01/rdf-schema#Literal  |
| <b>foaf:plan</b>           | Chỉ ra một chủ thích cho một kế hoạch.        | foaf:Person                                   | http://www.w3.org/2000/01/rdf-schema#Literal  |
| <b>foaf:primaryTopic</b>   | Chỉ một primary topic của một tài liệu.       | foaf:Document                                 | http://www.w3.org/2000/01/rdf-schema#Resource |
| <b>foaf:publications</b>   | Một link chỉ một tài liệu của một người.      | foaf:Person                                   | foaf:Document                                 |
| <b>foaf:schoolHomepage</b> | Chỉ ra trang web về trường học của một người. | foaf:Person                                   | foaf:Document                                 |
| <b>foaf:sha1</b>           | Chỉ ra một số sha1sum ở dạng hexa             | foaf:Document                                 |   |
| <b>foaf:surname</b>        | Chỉ ra một surname của một người.             | foaf:Person                                   | http://www.w3.org/2000/01/rdf-schema#Literal  |
| <b>foaf:theme</b>          | Chỉ ra một theme( mẫu)                        | http://www.w3.org/2000/01/rdf-schema#Resource | http://www.w3.org/2000/01/rdf-schema#Resource |
| <b>foaf:thumbnail</b>      | Một hình ảnh ở dạng thumbnail.                | foaf:Image                                    | foaf:Image                                    |
| <b>foaf:tipjar</b>         | Chỉ ra một tài liệu tipjar cho một tác nhân.  | foaf:Agent                                    | foaf:Document                                 |
| <b>foaf:title</b>          | Title (Mr, Mrs, Ms, Dr. ..)                   | foaf:Person                                   | http://www.w3.org/2000/01/rdf-schema#Literal  |
| <b>foaf:topic</b>          | Chỉ ra một chủ đề của một tài liệu nào đó.    | foaf:Document                                 | http://www.w3.org/2000/01/rdf-schema#Resource |
| <b>foaf:topic_interest</b> | Chỉ ra một tài nguyên mà một người quan tâm   | foaf:Person                                   | http://www.w3.org/2000/01/rdf-schema#Resource |

|                               |  |             |  |
|-------------------------------|--|-------------|--|
| <b>foaf:weblog</b>            | Chỉ ra một weblog của một tài nguyên.      | foaf:Agent  | foaf:Document                                |
| <b>foaf:workInfoHomepage</b>  | Chỉ ra trang làm việc của một người.       | foaf:Person | foaf:Document                                |
| <b>foaf:workplaceHomepage</b> | Chỉ ra một địa chỉ nơi làm việc của ai đó. | foaf:Person | foaf:Document                                |
| <b>foaf:yahooChatID</b>       | Chỉ ra một Yahoo chat ID.                  | foaf:Agent  | http://www.w3.org/2000/01/rdf-schema#Literal |

Xét một tài liệu mô tả đầy đủ thông tin về một người như sau:

```

<foaf:Person>
  <foaf:mbox rdf:resource=.mailto:ggrimnes@csd.abdn.ac.uk. />
  <foaf:name>Gunnar AAstrand Grimnes</foaf:name>
  <foaf:homepage rdf:resource=.http://www.csd.abdn.ac.uk/_ggrimnes. />
  <foaf:workplaceHomepage rdf:resource=.http://www.csd.abdn.ac.uk./>
  <foaf:projectHomepage
rdf:resource=.http://www.csd.abdn.ac.uk/research/agentcities./>
  <foaf:groupHomepage rdf:resource=.http://www.csd.abdn.ac.uk/research/agentsgroup.
 />
  <foaf:phone rdf:resource=.tel:+441224272835. />
  <foaf:depiction rdf:resource=.http://www.csd.abdn.ac.uk/_ggrimnes/gfx/me.jpg. />
  <foaf:interest rdf:resource=.http://www.w3.org/2001/sw/. />
  <foaf:interest rdf:resource=.http://www.agentcities.net. />
  <foaf:made
rdf:resource=.http://www.csd.abdn.ac.uk/research/AgentCities/GraniteNights. />

  <foaf:knows>
    <foaf:Person>
      <foaf:mbox rdf:resource=.mailto:maym@foobar.lu. />
      <rdfs:seeAlso rdf:resource=.http://martinmay.net/foaf.rdf./>
    </foaf:Person>
  
```

```
</foaf:knows>

<foaf:knows>
  <foaf:Person>
    <foaf:mbox rdf:resource=.mailto:apreece@csd.abdn.ac.uk. />
  </foaf:Person>
</foaf:knows>

<foaf:knows>
  <foaf:Person>
    <foaf:mbox rdf:resource=.mailto:pedwards@csd.abdn.ac.uk. />
  </foaf:Person>
</foaf:knows>

<foaf:knows>
  <foaf:Person foaf:name=.Sonja A Schramm.>
    <foaf:mbox
sha1sum>83276f91273f2900cf0b6657b3708b736276ef81</foaf:mboxsha1sum>
    </foaf:Person>
  </foaf:knows>
  <rdfs:seeAlso rdf:resource=.http://www.csd.abdn.ac.uk/_ggrimnes/codepict.rdf. />
  <rdfs:seeAlso rdf:resource=.http://www.csd.abdn.ac.uk/research/agentsgroup/foaf.rdf.
/>
</foaf:Person>
```



## Chương 4 TRUY VẤN DỮ LIỆU TRONG RDF

### 4.1 Giới thiệu

Như chúng ta đã biết RDF là một cách để mô tả thông tin về các tài nguyên Web một cách linh động. Nó được sử dụng để mô tả nhiều loại thông tin ví dụ như thông tin về cá nhân, về hệ thống mạng toàn cầu, metadata về multimedia như music và hình ảnh cũng như là cung cấp một phương tiện để tích hợp các nguồn tài nguyên khác. Với một lượng thông tin quá lớn như vậy, làm thế nào dữ liệu có thể được truy vấn nhanh chóng và chính xác? Vì vậy tổ chức W3C đã phát triển một ngôn ngữ chuẩn để truy vấn dữ liệu RDF với việc cung cấp nhiều chức năng cho các developer và end users cách để viết và thực thi một câu truy vấn. Chương này mô tả ngôn ngữ truy vấn Protocol And RDF Query Language (SPARQL) để truy cập dữ liệu RDF được phát triển bởi nhóm RDF Data Access Working Group-một phần trong hoạt động của Semantic Web.

Một RDF Graph là một tập các triple, mỗi triple bao gồm bộ ba subject, predicate và object (như đã giới thiệu ở chương 3). Những triple này có thể đến từ nhiều tài nguyên khác nhau. Ví dụ như: chúng có thể đến từ một tài liệu RDF, chúng có thể được suy ra từ các RDF triple khác hoặc là chúng có thể là sự mô tả RDF của các dữ liệu được lưu trữ từ các định dạng khác như XML hay là từ một cơ sở dữ liệu quan hệ.

SPARQL là một ngôn ngữ để truy cập thông tin từ các đồ thị RDF. Nó cung cấp các tính năng sau:

- Trích thông tin trong các dạng của URI, các blank node và các plain hay typed literals.
- Trích thông tin từ các đồ thị con.
- Xây dựng một đồ thị RDF mới dựa trên thông tin trong đồ thị truy vấn.

## **4.2 Tạo một câu truy vấn đơn giản**

Một câu truy vấn bao gồm 2 mệnh đề, mệnh đề SELECT và mệnh đề WHERE. Mệnh đề SELECT định danh các biến mà ứng dụng quan tâm và mệnh đề WHERE bao gồm các triple pattern.

Một triple pattern là một RDF triple nhưng mỗi thành phần (subject, predicate hay object) đều có thể là một biến truy vấn.

Một basic graph pattern là một tập các triple pattern.

Ngôn ngữ SPARQL dựa trên nền tảng so sánh các *graph pattern*. *Graph pattern* đơn giản nhất là các *triple pattern*.

Kết quả của câu truy vấn là tất cả các giải pháp mà một câu truy vấn có thể phù hợp với đồ thị được truy vấn. Một câu truy vấn có thể có không, một hay nhiều kết quả.

Ví dụ dưới đây sẽ chỉ ra một câu truy vấn để tìm ra một tựa đề (*title*) của một quyển sách từ thông tin trong một đồ thị RDF.

Ta có một RDF triple sau:

```
<http://example.org/book/book1> <http://purl.org/dc/elements/1.1/title> "SPARQL Tutorial" .
```

Câu truy vấn:

```
SELECT ?title
WHERE { <http://example.org/book/book1> <http://purl.org/dc/elements/1.1/title> ?title }
```

Kết quả truy vấn:

| Title             |
|-------------------|
| "SPARQL Tutorial" |

### 4.3 Cú pháp của câu truy vấn

Các giá trị được đặt trong dấu ‘<>’ dùng để chỉ một định danh URI. Các giá trị được đặt trong dấu (“”) là các giá trị literal.

Biến trong ngôn ngữ truy vấn có giá trị toàn cục. Biến thường bắt đầu với kí tự ‘?’. Đôi khi người ta cũng sử dụng kí tự ‘\$’ để chỉ một biến, để chỉ rằng nơi mà hệ thống sử dụng ‘?’ như một ký tự thay thế. Trong một câu truy vấn, \$abc và ?abc đều là biến.

Bởi vì các URI có thể được viết rất dài, SPARQL cung cấp một cơ chế viết tắt. Tiếp đầu ngữ (prefix) có thể được định nghĩa và một QName sẽ cung cấp một dạng viết làm cho URI có thể ngắn gọn.

Sau đây là một cách viết tắt cho *triple pattern*

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
SELECT ?title
WHERE { <http://example.org/book/book1> dc:title ?title }
```

**Ta có thể sử dụng một cách viết tắt :**

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX : <http://example.org/book/>
SELECT $title
WHERE { :book1 dc:title $title }
```

#### **Blank node**

Một blank node có thể xuất hiện trong một query pattern. Nó giữ vai trò như một biến, mặc dù nó không được đề cập trong kết quả của câu truy vấn hay bất kỳ nơi nào ở trong graph pattern.

Blank node trong các câu truy vấn thì phân biệt với các blank node trong dữ liệu. Một blank node trong một graph pattern không tương ứng với một blank node trong dữ liệu thông qua một blank node label.

Dữ liệu:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
```

```
_:a foaf:name "Johnny Lee Outlaw" .
```

```
_:a foaf:mbox <mailto:jlow@example.com> .
```

Truy vấn:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
SELECT ?mbox
```

```
WHERE
```

```
{ ?x foaf:name "Johnny Lee Outlaw" .
```

```
?x foaf:mbox ?mbox }
```

Kết quả:

| mbox                      |
|---------------------------|
| <mailto:jlow@example.com> |

Truy vấn với nhiều blank node

Data:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
```

```
_:a foaf:name "Johnny Lee Outlaw" .
```

```
_:a foaf:mbox <mailto:jlow@example.com> .
```

```
_:b foaf:name "Peter Goodguy" .
```

```
_:b foaf:mbox <mailto:peter@example.org> .
```

Truy vấn:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
SELECT ?name ?mbox
```

```
WHERE
```

```
{ ?x foaf:name ?name .
```

```
?x foaf:mbox ?mbox }
```

Kết quả:

| name                | mbox                       |
|---------------------|----------------------------|
| "Johnny Lee Outlaw" | <mailto:jlow@example.com>  |
| "Peter Goodguy"     | <mailto:peter@example.org> |

#### **4.4 Những dạng cú pháp khác**

##### **Danh sách predicate-Object**

Các triple pattern với một subject chung có thể được viết sao cho subject đó chỉ được viết 1 lần, mỗi triple pattern sẽ được cách nhau bởi dấu ‘;’

```
?x foaf:name ?name ;
foaf:mbox ?mbox .
```

Cách viết trên tương tự với cách viết:

```
?x foaf:name ?name .
?x foaf:mbox ?mbox .
```

##### **Danh sách các object:**

Nếu các triple pattern có cùng subject và predicate, ta có thể sử dụng dấu “,” để ngăn cách các object.

```
?x foaf:nick "Alice" , "Peter" .
```

Cách viết trên tương tự với cách viết:

```
?x foaf:nick "Alice" .
?x foaf:nick "Peter" .
```

##### **Blank node:**

Một blank node chỉ được sử dụng một lần trong câu truy vấn có thể được viết tắt bằng cách sử dụng “[ ]”.

```
[ ] :p "v" .
[ :p "v" ] .
```

Một cú pháp blank node được viết tắt có thể được kết hợp với những cách viết tắt khác cho các predicate và object chung.

```
[ foaf:name ?name ;foaf:mbox <alice@example.org> ]
```

Cách viết này tương tự với:

```
_:b18 foaf:name ?name .
_:b18 foaf:mbox <alice@example.org> .
```

### **RDF Collections**

RDF collection có thể được viết trong các triple pattern sử dụng dấu “()”. Dạng “()” là một dạng viết tắt của resource `rdf:nil` hay <http://www.w3.org/1999/02/22-rdf-syntax-ns#nil>.

Ví dụ:

(1 ?x 3)

Là dạng viết tắt của:

```
_:b0 rdf:first 1 .
_:b0 rdf:rest _:b1 .
_:b1 rdf:first ?x .
_:b1 rdf:rest _:b2 .
_:b2 rdf:first 3 .
_:b2 rdf:rest rdf:nil .
```

### **RDF Literal**

Một RDF Literal trong ngôn ngữ SPARQL được viết như một chuỗi trong dấu (“”), theo sau có thể là một tag optional language (được chỉ bởi ‘@’) hay là một kiểu optional datatype(được chỉ bởi ‘^^’). Có một dạng viết thuận lợi đối với các kiểu numeric-typed literal như `xsd:integer`, `xsd:double` hay là `xsd:boolean`.

Xét các ví dụ về literal sau:

- "chat"
- "chat"@fr
- "xyz"^^<http://example.org/ns/userDatatype>
- "abc"^^myNS:myDataType
- 1, tương tự như "1"^^xsd:integer
- 1.0e6, tương tự như "1.0e6"^^xsd:double
- true, tương tự như "true"^^xsd:boolean
- false, tương tự như "false"^^xsd:boolean

Truy vấn RDF Literal:

Dữ liệu:

```
@prefix dt: <http://example.org/datatype#> .
@prefix ns: <http://example.org/ns#> .
@prefix : <http://example.org/ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

:x ns:p "42"^^xsd:integer .
:y ns:p "abc"^^dt:specialDatatype .
:z ns:p "cat"@en .
```

### **Truy vấn số nguyên( Integer):**

```
SELECT ?v WHERE { ?v ?p 42 }
```

Câu truy vấn trên cho kết quả là :x bởi vì 42 là cú pháp cho "42"^^<http://www.w3.org/2001/XMLSchema#integer>.

### **Truy vấn kiểu language tag**

```
SELECT ?x WHERE { ?x ?p "cat" }
```

Câu truy vấn trên không cho ra kết quả bởi vì giá trị “cat” không cùng kiểu dữ liệu với giá trị “cat”@en.

Nhưng câu truy vấn sau lại cho ra kết quả là :z

```
SELECT ?x WHERE { ?x ?p "cat"@en }
```

## **4.5 Ràng buộc dữ liệu**

Ta có thể truy vấn dữ liệu với một kiểu dữ liệu có điều kiện. Ví dụ như:  
Dữ liệu:

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix : <http://example.org/book/> .
@prefix ns: <http://example.org/ns#> .

:book1 dc:title "SPARQL Tutorial" .
:book1 ns:price 42 .
:book2 dc:title "The Semantic Web" .
:book2 ns:price 23 .
```

Truy vấn:

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX ns: <http://example.org/ns#>
SELECT ?title ?price
WHERE { ?x ns:price ?price .
        FILTER ?price < 30 .
        ?x dc:title ?title . }
```

Kết quả :

| title              | price |
|--------------------|-------|
| "The Semantic Web" | 23    |

Bằng cách có một ràng buộc trên biến “price” . chỉ có book2 có thể phù hợp với nội dung của câu truy vấn.

#### 4.5.1 Thêm vào kiểu Optional

Với những ví dụ trên ta thấy rằng mỗi giải pháp của câu truy vấn phải hoàn toàn phù hợp với các thành phần của query pattern. Nhưng với việc thêm vào từ khóa *optional* ta có thể có nhiều hơn giải pháp , mặc dù mỗi giải pháp này có thể chỉ thỏa một phần của query pattern.

Ví dụ:

Dữ liệu:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

_:a rdf:type foaf:Person .
_:a foaf:name "Alice" .
_:a foaf:mbox <mailto:alice@work.example> .

_:b rdf:type foaf:Person .
_:b foaf:name "Bob" .
```

Truy vấn:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
```



```
WHERE { ?x foaf:name ?name .
        OPTIONAL { ?x foaf:mbox ?mbox } }
```

Kết quả:

| name    | mbox                       |
|---------|----------------------------|
| "Alice" | <mailto:alice@example.com> |
| "Bob"   |                            |

Nếu không có optional trong query pattern thứ 2 kết quả sẽ là:

| name    | mbox                       |
|---------|----------------------------|
| "Alice" | <mailto:alice@example.com> |

Như vậy ta thấy nhiệm vụ của optional trong query pattern là chọn tất cả các giá trị có thể có của query pattern, nếu thỏa sẽ cho ra kết quả, nếu không thỏa thì cho ra một kết quả rỗng.

#### 4.5.2 Ràng buộc trong khối Optional

Ràng buộc có thể được sử dụng trong khối option. Ví dụ như:

Dữ liệu:

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix : <http://example.org/book/> .
@prefix ns: <http://example.org/ns#> .

:book1 dc:title "SPARQL Tutorial" .
:book1 ns:price 42 .
:book2 dc:title "The Semantic Web" .
:book2 ns:price 23 .
```

Truy vấn:

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX ns: <http://example.org/ns#>
SELECT ?title ?price
WHERE { ?x dc:title ?title .
        OPTIONAL { ?x ns:price ?price . FILTER ?price < 30 }
        }
```

Kết quả:

| title              | price |
|--------------------|-------|
| "SPARQL Tutorial"  |       |
| "The Semantic Web" | 23    |

Không có price nào cho cuốn sách có tựa đề là ‘SPARQL Tutorial’ bởi vì khối optional không nhận được giá trị nào thỏa mãn biến truy vấn với ràng buộc dữ liệu trên price cho nên sẽ trả về một giá trị Null.

### 4.5.3 Nhiều khối Optional

Graph pattern được định nghĩa đệ quy. Một câu truy vấn có thể có một hay nhiều khối optional và bất kỳ thành phần nào trong query pattern có thể có một phần optional. Trong ví dụ này, có 2 optional block.

Dữ liệu:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

_:a foaf:name "Alice" .
_:a foaf:homepage <http://work.example.org/alice/> .

_:b foaf:name "Bob" .
_:b foaf:mbox <mailto:bob@work.example> .
```

Truy vấn:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox ?hpage
WHERE { ?x foaf:name ?name .
        OPTIONAL { ?x foaf:mbox ?mbox } .
        OPTIONAL { ?x foaf:homepage ?hpage }
}
```

Kết quả:

| name    | mbox | hpage                            |
|---------|------|----------------------------------|
| "Alice" |      | <http://work.example.org/alice/> |

|       |                          |  |
|-------|--------------------------|--|
| "Bob" | <mailto:bob@example.com> |  |
|-------|--------------------------|--|

#### 4.5.4 Khối Optional lồng nhau

Optional pattern có thể xảy ra trong bất kỳ một pattern nào, kể cả trong một nhóm các graph pattern mà chính nó cũng là một optional, tạo nên một tập các optional pattern lồng. Khối optional bên ngoài phải thỏa cho bất kỳ một khối optional bên trong nó. Xét ví dụ sau:

Dữ liệu:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix vcard: <http://www.w3.org/2001/vcard-rdf/3.0#> .

_:a foaf:name "Alice" .
_:a foaf:mbox <mailto:alice@work.example> .
_:a vcard:N _:x .

_:x vcard:Family "Hacker" .
_:x vcard:Given "Alice" .

_:b foaf:name "Bob" .
_:b foaf:mbox <mailto:bob@work.example> .
_:b foaf:N _:z .

_:z vcard:Family "Hacker" .

_:e foaf:name "Ella" .
_:e vcard:N _:y .

_:y vcard:Given "Eleanor" .
```

Truy vấn:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>
SELECT ?foafName ?mbox ?gname ?fname
```

```

WHERE
{
  ?x foaf:name ?foafName .
  OPTIONAL { ?x foaf:mbox ?mbox } .
  OPTIONAL { ?x vcard:N ?vc .
    ?vc vcard:Given ?gname .
    OPTIONAL { ?vc vcard:Family ?fname }
  }
}
    
```

Kết quả:

| foafName | mbox                        | gname     | fname    |
|----------|-----------------------------|-----------|----------|
| "Alice"  | <mailto:alice@work.example> | "Alice"   | "Hacker" |
| "Bob"    | <mailto:bob@example.com>    |           |          |
| "Ella"   |                             | "Eleanor" |          |

## 4.6 Các phép toán và điều kiện trên câu truy vấn

### 4.6.1 Phép hội

Ta sử dụng từ khóa UNION để hội các graph pattern với nhau

Xét ví dụ:

Dữ liệu:

```

@prefix dc10: <http://purl.org/dc/elements/1.0/> .
@prefix dc11: <http://purl.org/dc/elements/1.1/> .

_:a dc10:title "SPARQL Query Language Tutorial" .

_:b dc11:title "SPARQL Protocol Tutorial" .

_:c dc10:title "SPARQL" .
_:c dc11:title "SPARQL (updated)" .
    
```

Truy vấn:

```

PREFIX dc10: <http://purl.org/dc/elements/1.0/>
PREFIX dc11: <http://purl.org/dc/elements/1.1/>
    
```

```
SELECT ?title
WHERE { { ?book dc10:title ?title } UNION { ?book dc11:title ?title } }
```

Kết quả:

| title                            |
|----------------------------------|
| "SPARQL Protocol Tutorial"       |
| "SPARQL"                         |
| "SPARQL (updated)"               |
| "SPARQL Query Language Tutorial" |

#### 4.6.2 Phép chiếu

Tương tự như cơ sở dữ liệu, các giải pháp của kết quả truy vấn có thể được giới hạn bởi tập các biến. Đối với mỗi giải pháp này, ta có thể tạo ra một giải pháp mới thông qua phép chiếu trên biến mà chúng ta có nhu cầu.

#### 4.6.3 DISTINCT

Từ khóa chỉ được sử dụng trong mệnh đề SELECT để mỗi giải pháp trong kết quả là phân biệt (tương tự trong cơ sở dữ liệu, các bộ không được trùng nhau).

Ví dụ như:

Dữ liệu:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

_:a foaf:name "Alice" .
_:a foaf:mbox <mailto:alice@org> .

_:z foaf:name "Alice" .
_:z foaf:mbox <mailto:smith@work> .
```

Truy vấn:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT DISTINCT ?name WHERE { ?x foaf:name ?name }
```

Kết quả:

name

#### 4.6.4 ORDER BY

Tương tự cơ sở dữ liệu, mệnh đề ORDER BY luôn đặt sau mệnh đề WHERE, nó sẽ nhận kết quả từ mệnh đề WHERE và chỉ lọc kết quả theo điều kiện có trong mệnh đề ORDER BY. Một điều kiện trong mệnh đề ORDER BY có thể là một biến hay là một hàm. Hướng của điều kiện là tăng theo mặc định. Chúng ta có thể xét cho điều kiện là tăng hay giảm bằng cách sử dụng khóa ASC[ ] (tăng), DESC[ ] (giảm).

Ta có các ví dụ:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name
WHERE { ?x foaf:name ?name }
ORDER BY ?name
```

→ Sắp xếp theo biến name tăng dần

```
PREFIX : <http://example.org/ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?name
WHERE { ?x foaf:name ?name ; :empld ?emp }
ORDER BY DESC[?emp]
```

→ Sắp xếp theo biến emp giảm dần

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name
WHERE { ?x foaf:name ?name ; :empld ?emp }
ORDER BY ?name DESC[?emp]
```

→ Sắp xếp theo biến name tăng dần và biến emp giảm dần

#### 4.6.5 LIMIT

Số giải pháp của kết quả được trả về trong một giới hạn được sử dụng với từ khóa LIMIT.

Ví dụ:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?name
WHERE { ?x foaf:name ?name }
LIMIT 20
```

Câu truy vấn chỉ cho phép tối đa 20 giải pháp được trả về.

#### 4.7 Lựa chọn biến được trả về

Trong cơ sở dữ liệu quan hệ, kết quả trả về của mỗi câu truy vấn là một bảng trong đó, các cột là các biến, các dòng là các bộ. Với ngôn ngữ SPAQL ta có thể chọn lựa kiểu biến trả về dưới dạng XML hay là RDF.

Xét ví dụ sau:

Dữ liệu:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
_:a foaf:name "Alice" .
_:a foaf:knows _:b .
_:a foaf:knows _:c .

_:b foaf:name "Bob" .

_:c foaf:name "Clare" .
_:c foaf:nick "CT" .
```

Truy vấn:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?nameX ?nameY ?nickY
WHERE
{ ?x foaf:knows ?y ;
  foaf:name ?nameX .
```

```
?y foaf:name ?nameY .
OPTIONAL { ?y foaf:nick ?nickY }
}
```

Kết quả:

| nameX   | nameY   | nickY |
|---------|---------|-------|
| "Alice" | "Bob"   |       |
| "Alice" | "Clare" | "CT"  |

Kết quả là một tài liệu XML

```
<?xml version="1.0"?>
<sparql
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns="http://www.w3.org/2001/sw/DataAccess/rf1/result" >
  <head>
    <variable name="nameX"/>
    <variable name="nameY"/>
    <variable name="nickY"/>
  </head>
  <results>
    <result>
      <nameX>Alice</nameX>
      <nameY>Clare</nameY>
      <nickY>CT</nickY>
    </result>
    <result>
      <nameX>Alice</nameX>
      <nameY>Bob</nameY>
      <nickY bound="false"/>
    </result>
  </results>
</sparql>
```

Kết quả truy vấn là một tài liệu RDF .



```

<rdf:RDF
  xmlns:rs="http://www.w3.org/2001/sw/DataAccess/tests/result-set#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rs:ResultSet>
    <rs:resultVariable>nickY</rs:resultVariable>

    <rs:resultVariable>nameX</rs:resultVariable>
    <rs:resultVariable>nameY</rs:resultVariable>
    <rs:solution rdf:parseType="Resource">
      <rs:binding rdf:parseType="Resource">
        <rs:variable>nameX</rs:variable>
        <rs:value>Alice</rs:value>
      </rs:binding>
      <rs:binding rdf:parseType="Resource">
        <rs:value>CT</rs:value>
        <rs:variable>nickY</rs:variable>
      </rs:binding>
      <rs:binding rdf:parseType="Resource">
        <rs:value>Clare</rs:value>
        <rs:variable>nameY</rs:variable>
      </rs:binding>
    </rs:solution>
    <rs:solution rdf:parseType="Resource">
      <rs:binding rdf:parseType="Resource">
        <rs:variable>nameX</rs:variable>
        <rs:value>Alice</rs:value>
      </rs:binding>
      <rs:binding rdf:parseType="Resource">
        <rs:value>Bob</rs:value>
        <rs:variable>nameY</rs:variable>
      </rs:binding>
    </rs:solution>
  </rs:ResultSet>
</rdf:RDF>

```

#### 4.8 Các toán tử trong ngôn ngữ SPARQL

| Toán tử trong SPARQL  |             |             |  |             |
|---|-------------|-------------|--|-------------|
| Operator  | Type(A)     | Type(B)     | Function   | Result type |
| A    B  | xs:boolean  | xs:boolean  | <a href="#">sop:logical-or</a> (A, B)  | xs:boolean  |
| Trả về kiểu boolean: TRUE nếu cả A hay B là đúng, ngược lại là FALSE. |             |             |  |             |
| A && B  | xs:boolean  | xs:boolean  | <a href="#">sop:logical-and</a> (A, B)   | xs:boolean  |
| Trả về kiểu boolean: TRUE nếu cả A và B là đúng, ngược lại là FALSE.  |             |             |  |             |
| A = B   | xs:string   | xs:string   | <a href="#">op:numeric-equal</a> ( <a href="#">fn:compare</a> (A, B), 0)   | xs:boolean  |
| A != B  | xs:string   | xs:string   | <a href="#">fn:not</a> ( <a href="#">op:numeric-equal</a> ( <a href="#">fn:compare</a> (A, B), 0))                     | xs:boolean  |
| A ≡ B   | numeric     | numeric     | <a href="#">op:numeric-equal</a> (A, B)  | xs:boolean  |
| A ≡ B   | xs:dateTime | xs:dateTime | <a href="#">op:dateTime-equal</a> (A, B)   | xs:boolean  |
| A != B  | numeric     | numeric     | <a href="#">fn:not</a> ( <a href="#">op:numeric-equal</a> (A, B))  | xs:boolean  |
| A != B  | xs:dateTime | xs:dateTime | <a href="#">fn:not</a> ( <a href="#">op:dateTime-equal</a> (A, B))   | xs:boolean  |
| A ≤ B   | numeric     | numeric     | <a href="#">op:numeric-less-than</a> (A, B)  | xs:boolean  |
| A ≤ B   | xs:dateTime | xs:dateTime | <a href="#">op:dateTime-less-than</a> (A, B)   | xs:boolean  |
| A ≥ B   | numeric     | numeric     | <a href="#">op:numeric-greater-than</a> (A, B)   | xs:boolean  |
| A ≥ B   | xs:dateTime | xs:dateTime | <a href="#">op:dateTime-greater-than</a> (A, B)  | xs:boolean  |
| A <= B  | numeric     | numeric     | <a href="#">sop:logical-or</a> ( <a href="#">op:numeric-less-than</a> (A, B), <a href="#">op:numeric-equal</a> (A, B)) | xs:boolean  |

|  |                         |                         |   |   |
|--|-------------------------|-------------------------|---|---|
| A <= B   | xs:dateTime             | xs:dateTime             | <a href="#">fn:not(op:dateTime-greater-than(A, B))</a>  | xs:boolean  |
| A >= B   | numeric                 | numeric                 | <a href="#">sop:logical-or(op:numeric-greater-than(A, B), op:numeric-equal(A, B))</a>             | xs:boolean  |
| A > B  | xs:dateTime             | xs:dateTime             | <a href="#">fn:not(op:dateTime-less-than(A, B))</a>   | xs:boolean  |
| A * B  | numeric                 | numeric                 | <a href="#">op:numeric-multiply(A, B)</a>   | numeric   |
| A / B  | numeric                 | numeric                 | <a href="#">op:numeric-divide(A, B)</a>   | numeric; nhưng là xs:decimal nếu cả hai toán hạng đều là xs:integer |
| A + B  | numeric                 | numeric                 | <a href="#">op:numeric-add(A, B)</a>  | numeric   |
| A - B  | numeric                 | numeric                 | <a href="#">op:numeric-subtract(A, B)</a>   | numeric   |
| A = B  | r:term                  | r:term                  | <a href="#">sop:RDFterm-equal(A, B)</a>   | xs:boolean  |
| A != B   | r:term                  | r:term                  | <a href="#">fn:not(sop:RDFterm-equal(A, B))</a>   | xs:boolean  |
| <a href="#">BOUND(A)</a>                         | variable                | N/A                     | <a href="#">sop:isBound(A)</a>  | xs:boolean  |
| <a href="#">ISURI(A)</a>                         | variable                | N/A                     | <a href="#">sop:isURI(A)</a>  | xs:boolean  |
| <a href="#">ISBLANK(A)</a>                       | variable                | N/A                     | <a href="#">sop:isBlank(A)</a>  | xs:boolean  |
| <a href="#">ISLITERAL(A)</a>                     | variable                | N/A                     | <a href="#">sop:isLiteral(A)</a>  | xs:boolean  |
| <a href="#">REGEX(STRING, PATTERN [, FLAGS])</a> | xs:string               | xs:string [, xs:string] | <a href="#">fn:matches(STRING, PATTERN)</a><br><a href="#">fn:matches(STRING, PATTERN, FLAGS)</a> | xs:boolean  |
| <a href="#">STR(A)</a>                           | rdf:uri hay rdf:literal | N/A                     | <a href="#">sop:str(A)</a>  | xs:string   |
| <a href="#">LANG(A)</a>                          | rdf:literal             | N/A                     | <a href="#">sop:lang(A)</a>   | xs:string   |
| <a href="#">DATATYPE(A)</a>                      | rdf:literal             | N/A                     | <a href="#">sop:datatype(A)</a>   | rdf:uri   |

#### 4.8.1 sop:RDFterm-equal

Đây là phép toán so sánh có dạng ?u=?v. Xét ví dụ:

Dữ liệu:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

_:a foaf:name "Alice".
_:a foaf:mbox <mailto:alice@work.example> .

_:b foaf:name "Ms A.".
_:b foaf:mbox <mailto:alice@work.example> .
```

Truy vấn:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name1 ?name2
WHERE {
  ?x foaf:name ?name1 ;
      foaf:mbox ?mbox1 .
  ?y foaf:name ?name2 ;
      foaf:mbox ?mbox2 .
  FILTER ?mbox1 = ?mbox2 && ?name1 != ?name2
}
```

Kết quả:

| name1   | name2   |
|---------|---------|
| "Alice" | "Ms A." |

#### 4.8.2 sop:isURI

Kiểm tra giá trị của một biến có là URI hay không?

Dữ liệu:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

_:a foaf:name "Alice".
```

```
_:a foaf:mbox <mailto:alice@work.example> .
_:b foaf:name "Bob" .
_:b foaf:mbox "bob@work.example" .
```

Truy vấn:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE { ?x foaf:name ?name ;
        foaf:mbox ?mbox .
        FILTER isUri(?mbox) }
```

Kết quả:

| name    | mbox                        |
|---------|-----------------------------|
| "Alice" | <mailto:alice@work.example> |

#### 4.8.3 sop:isBLANK

Kiểm tra giá trị của một biến có phải là một blank node hay không?

Dữ liệu

```
@prefix a: <http://www.w3.org/2000/10/annotation-ns#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

_:a a:annotates <http://www.w3.org/TR/rdf-sparql-query/> .
_:a dc:creator "Alice B. Toeclips" .

_:b a:annotates <http://www.w3.org/TR/rdf-sparql-query/> .
_:b dc:creator _:c .
_:c foaf:given "Bob" .
_:c foaf:family "Smith" .
```

Truy vấn

```
PREFIX a: <http://www.w3.org/2000/10/annotation-ns#>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?given ?family
WHERE { ?annot a:annotates <http://www.w3.org/TR/rdf-sparql-query/> .
        ?annot dc:creator ?c .
```

```
OPTIONAL { ?c foaf:given ?given ; foaf:family ?family } .  
FILTER isBlank(?c)  
}
```

Kết quả:

| given | family  |
|-------|---------|
| "Bob" | "Smith" |

#### 4.8.4 sop:isLiteral

Kiểm tra giá trị của một biến có phải là một literal hay không?

#### 4.8.5 sop:logical-or

Là phép toán so sánh có kiểu ?u|?v

#### 4.8.6 sop:logical-and

Là phép toán so sánh có kiểu ?u&?v

#### 4.8.7 sop:str

Trả về kiểu xs:string cho một biến có kiểu là URI.

#### 4.8.8 so:lang

Trả về một ngôn ngữ hợp lệ cho một biến.

#### 4.8.9 sop:datatype

Trả về một kiểu dữ liệu cho một biến.

## Chương 5 CÁC ỨNG DỤNG SEMANTIC WEB

### 5.1 Các lĩnh vực ứng dụng

#### 5.1.1 Search Engine

Vấn đề hiện nay là đa số các Search Engines hiện có đều thuộc loại Keyword Search Engine. Cơ chế của chúng là định kì duyệt Web để phát hiện ra những sự thay đổi, rồi lập chỉ mục những thay đổi này. Người sử dụng có thể tạo các câu truy vấn gồm các từ khóa trên các chỉ mục đó để nhận về kết quả mong muốn. Tuy nhiên, phương pháp này gặp hai vấn đề chính sau đây:

- Một từ khóa có thể có một hay nhiều ý nghĩa tùy theo từng ngữ cảnh và Search Engine không thể hiện mối quan hệ giữa các từ khóa với nhau.
- Các trang Web có cùng ý nghĩa với câu truy vấn của người sử dụng sẽ không tồn tại trong kết quả trả về.

Ví dụ: ta cần tìm thông tin về người trưởng bộ môn CNTT của MIT, ta gõ: “*MIT computer science chair*” vào Google, nhưng kết quả thu được là không chính xác.



Hình 5-1 Kết quả tìm kiếm của Google.

Nguyên nhân của việc tìm kiếm thất bại là do: keyword “MIT” có nhiều ý nghĩa. Ngoài ra, Search Engine không thể hiểu mối liên hệ giữa các keyword: MIT, computer science và chair. Nếu Search Engine được tích hợp tri thức để hiểu được ý nghĩa của các từ, thì rất có thể nó cho ta kết quả chính xác hơn, lúc đó việc tìm kiếm sẽ dựa trên khái niệm (concept) chứ không phải theo từ khóa (keyword).

### 5.1.2 Khung làm việc để quản lý tri thức - Framework for Knowledge Management

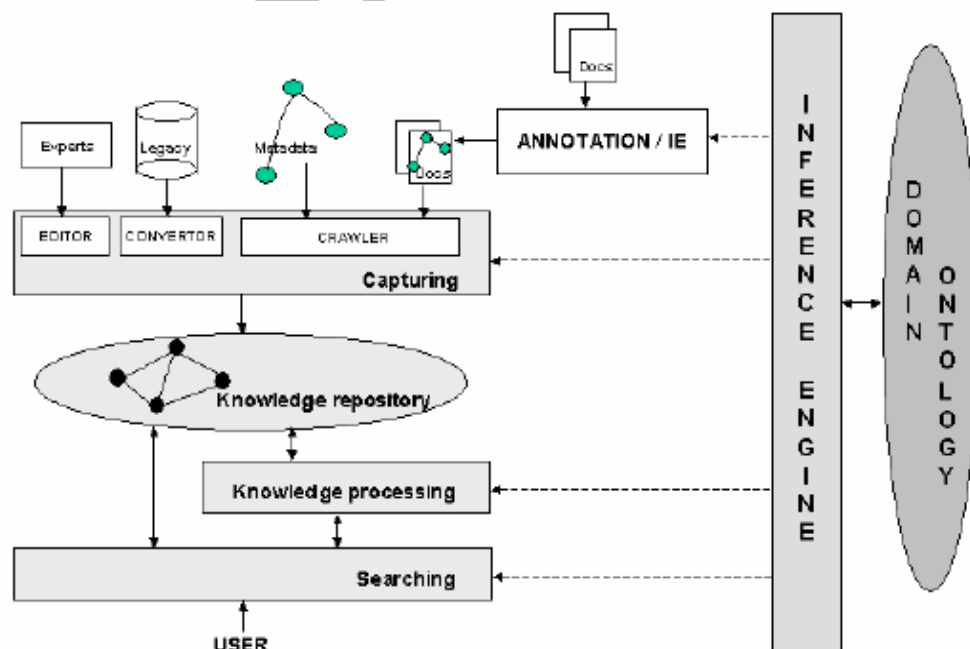
Semantic Web là một hệ nền nhiều hứa hẹn cho việc phát triển các hệ thống quản lý tri thức. Tuy nhiên, vấn đề ở đây là làm thế nào biểu diễn tri thức ở dạng thức máy có thể hiểu được, để tri thức cần thiết có thể được tìm thấy bởi các machine agent.



Chúng ta sử dụng giải pháp quản lý tri thức dựa trên định dạng tương thích RDF để biểu diễn các luật và dựa trên một kỹ thuật mới để chú giải các nguồn tri thức bằng cách sử dụng các câu điều kiện. Giải pháp là dựa trên các công cụ Semantic Web đang tồn tại. Điểm thuận lợi chính là sự thúc đẩy khả năng tìm kiếm tri thức với độ chính xác cao, cũng như khả năng truy cập cấu tạo các nguồn tri thức cần thiết cho việc giải quyết một vấn đề nào đó.

Dạng thức này có thể được biểu diễn bằng cách dùng các statement If-Then, được thiết lập theo cách suy diễn (inference) và ủy quyền (trust) trên Semantic Web.

Các statement điều kiện có thể được dùng để lập chỉ mục nội dung các tài nguyên Web một cách nhiều ý nghĩa hơn so với liên kết các từ khóa / khái niệm hay metadata chung (các mối liên hệ). Điều này có thể sẽ hình thành các truy vấn dựa trên ngữ cảnh hơn, tăng cường độ chính xác trong tìm kiếm tri thức. Ví dụ, trong vấn đề chỉ mục tài liệu, dù có hay không có tài liệu được chỉ mục bởi từ khóa “aspirin” và “headache (đau đầu)”, cách aspirin trị headache hay aspirin gây ra headache đều có thể được giải quyết dễ dàng bằng cách sử dụng các câu điều kiện định nghĩa trước.



Hình 5-2 Khung làm việc quản lý tri thức được đề nghị.

Hình trên phác họa khung làm việc cho quản lý tri thức trên Semantic Web, phản ánh sự chuyển đổi đa dạng trong môi trường phân tán này: tri thức có thể được tập hợp từ các nguồn khác nhau và trong các định dạng khác nhau; sau đó, chúng được lưu trữ trong các hình thức biểu diễn thông thường, được xử lý để tính toán phụ thuộc lẫn nhau giữa các thành phần tri thức, hay để giải quyết các xung đột, chia sẻ / tìm kiếm, và cuối cùng, được sử dụng để giải quyết vấn đề.

### **5.1.3 Internet Agent**

Tiềm năng kế tiếp của Semantic Web có thể kể đến là ứng dụng vào các Internet agents. Hiện nay, để hoàn thành công việc, các internet agents phải yêu cầu các trang Web, thu thập và xử lý chúng. Thông thường công việc của agents là so sánh giá cả, tham gia đấu thầu, sắp xếp, và đăng kí công việc. Ví dụ: để hoàn tất việc đăng kí một tour du lịch, ta phải có một agent đặt chỗ ở tour, một agent đăng kí vé máy bay, đặt phòng dựa trên các giá cả rẻ nhất. Tuy nhiên, các công việc này đều phải làm sẵn trước đối với các trang Web cố định, do đó rất nguy hiểm nếu trang Web có sự thay đổi về cấu trúc. Vì thế, agent cần phải hiểu được ngữ nghĩa của trang Web hơn là sự thể hiện của trang Web để hoạt động một cách tự động.

### **5.1.4 Push System**

Các hệ thống push system hiện nay sẽ phát sinh các thông tin từ Web một cách thường xuyên đến các user dựa vào profile của họ. Tuy nhiên, hiện nay thông tin mà user nhận được thường không phù hợp với yêu cầu của họ. Do đó, cần phải tích hợp semantic để đánh giá chính xác hơn.

### **5.1.5 Relationship**

Semantic Web còn được dùng để tổ chức lại các trang Web có nội dung liên quan nhau, phục vụ công việc chọn lọc và index.

## **5.2 Các ứng dụng & dự án hiện có**

Xem phần phụ lục A trang 211.

## **5.3 Các công cụ phát triển**

Xem phần phụ lục B trang 215.

## **5.4 Giới thiệu Semantic Search Engine**

### **5.4.1 Giới thiệu:**

Search Engine luôn là một ứng dụng rất quan trọng. Thực tế cho thấy có rất nhiều Search Engines đã & đang được xây dựng và chúng đang đóng những vai trò quan trọng như các Internet Search Engines Google, AOL, Yahoo, Altavista, MSN, ... và vô số các Intranet Search Engines ở các trường học, công ty, tổ chức, ... Ngoài ra, còn có các Search Engines khác được tích hợp trong các phần mềm.

Vấn đề hiện nay là đa số các Search Engines hiện có đều thuộc loại *Keyword Search Engine*. Cơ chế của chúng là định kỳ duyệt Web để phát hiện ra những sự thay đổi, rồi lập chỉ mục những thay đổi này. Người sử dụng có thể tạo các câu truy vấn gồm các từ khóa trên các chỉ mục đó để nhận về kết quả mong muốn. Tuy nhiên, phương pháp này gặp hai vấn đề chính sau đây:

- Một từ khóa có thể có một hay nhiều ý nghĩa tùy theo từng ngữ cảnh và Search Engine không thể hiện mối quan hệ giữa các từ khóa với nhau.
- Các trang Web có cùng ý nghĩa với câu truy vấn của người sử dụng sẽ không tồn tại trong kết quả trả về.

Ví dụ: ta cần tìm thông tin về người trưởng bộ môn CNTT của MIT, ta gõ: “*MIT computer science chair*” vào Google, nhưng kết quả thu được không chính xác.

Nguyên nhân của việc tìm kiếm thất bại là do: keyword “*MIT*” có nhiều ý nghĩa. Ngoài ra, Search Engine không thể hiểu mối liên hệ giữa các keyword: *MIT*, *computer science*, và *chair*. Nếu Search Engine được tích hợp tri thức để hiểu được ý nghĩa của

các từ, thì rất có thể nó sẽ cho ta kết quả chính xác hơn, lúc đó việc tìm kiếm sẽ dựa trên khái niệm (concept) chứ không phải theo từ khóa (keyword).

Ví dụ ở trên là một trong rất nhiều trường hợp người dùng thực hiện việc tìm kiếm không phải để tìm thông tin chi tiết mà là để tìm các thông tin liên quan hay các phạm trù, lĩnh vực liên quan.

Mục đích của chương này là đưa ra một cái nhìn tổng thể về Semantic Search Engine (Máy tìm kiếm dựa vào ngữ nghĩa trên quan điểm của Semantic Web), mô tả những ưu điểm của Semantic Search Engine. Semantic Search Engine đã khắc phục những khuyết điểm của các Search Engine truyền thống (Keyword Search Engine).

#### **5.4.2 So sánh giữa Search Engine truyền thống( Keyword/Free-Text Search Engine) và Semantic Search Engine:**

##### **5.4.2.1 Search Engine truyền thống**

Không thể tìm ra các tài nguyên thích hợp một cách hiệu quả vì:

- Những tài liệu người dùng muốn tìm có thể sử dụng những thuật ngữ khác (hay những từ đồng nghĩa khác). Ví dụ, ‘*Myocardial infarction*’ và ‘*coronary thrombosis*’ là tương đương nhau.
- Những lỗi chính tả và các biến thể của từ ngữ được xem là những thuật ngữ khác nhau đối với môi trường máy tính. Ví dụ, ‘*psoriasis*’ (đúng chính tả) và ‘*psoriaisis*’ (lỗi in ấn) được cho là khác nhau; tương tự, ‘*anaemia*’ (đúng chính tả đối với T.Anh) và ‘*anemia*’ (đúng chính tả đối với T.Mỹ) cũng được xem là khác nhau.
- Search Engine không thể xử lý các trang HTML một cách thông minh. Ví dụ, việc tìm kiếm các tài nguyên về ‘*psoriasis*’ (bệnh vẩy nến) sẽ lấy tất cả các tài liệu có chứa từ này. Tuy nhiên, đa số các tài liệu đều không thích hợp/liên quan; nghĩa là, ‘*psoriasis*’ trong các tài liệu này được đề cập theo một cách khác với chủ đề của chúng. Ví dụ, một vài tài liệu sẽ đề cập đến ‘*psoriasis*’ ở

những câu ‘*See also: psoriasis*’; nghĩa là, tại đáy trang (hay tại một phần nào đó của trang) có đề cập đến những bệnh có triệu chứng tương tự.

Vì thế, người dùng phải:

- Học cách sử dụng Search Engine (SE) một cách hiệu quả nhất: ***tìm liên quan (related searches)***: vd: tìm ‘mar’, SE hiển thị những liên kết liên quan khác như ‘mission to mars’, ‘life on mars’, ... ***tìm theo gốc từ (stemming)***: vd: tìm ‘swimming’, SE trả về những kết quả có chứa ‘swim’, ‘swims’, ... ***tìm trên kết quả (search within)***: vd: nếu kết quả trả về quá nhiều, người dùng có thể chọn chức năng *search within* để SE tìm trên kết quả trả về trước đó. Ngoài ra còn có các đặc tính khác như: ***tìm theo phiên bản, tìm theo ngôn ngữ, tìm theo ngày (phạm vi ngày)***, ... Các kỹ năng tìm kiếm khác như: dùng ***dấu ‘+’*** (tìm tài liệu chứa tất cả các từ), ***dấu ‘-’*** (tìm tài liệu không chứa những từ), ***dấu “”*** (tìm cụm từ), các toán tử ***AND, OR, NOT***, ***tìm theo site, tìm trên URL, tìm trên link, tìm trên title, dùng wildcard*** (kí tự thay thế), ***tìm gần kề*** (vd: NEAR/5: các từ phải nằm cách nhau tối đa 5 từ), ***dùng NEST*** (dấu đóng/mở ngoặc), ...
- Dùng nhiều Search Engine để mong có kết quả theo ý muốn.
- Dùng các Search Engine chuyên biệt (theo lĩnh vực muốn tìm).

#### 5.4.2.2 Semantic Search Engine (SSE)

SSE sẽ đáp ứng hầu hết mọi mong đợi của người dùng, nó mang lại:

- Một viễn cảnh về lĩnh vực tri thức (knowledge domain) tốt hơn.
- Cho phép tìm kiếm thông tin dựa trên nội dung tài liệu.
- Thông tin tìm kiếm được trả về chính xác và phù hợp (tìm ra những tài liệu mà ta không thể tìm thấy nếu dùng các Search Engine truyền thống).

SSE có những ưu điểm như trên là do nó kế thừa những ưu điểm của công nghệ Semantic Web:

- Cách tổ chức dữ liệu theo hướng đối tượng (các Knowledge Base: gồm các Ontology và tập các Instance tương ứng. Trong tương lai, nếu các Knowledge Base được tổ chức tốt và được liên kết với nhau nhiều hơn, các SSE sẽ có khả năng hỗ trợ tìm kiếm nhiều lĩnh vực, nhiều file format, đa ngôn ngữ, với độ chính xác cao).
- Được sự hỗ trợ to lớn từ các Agent tự động. Các Agent này hỗ trợ SSE trong việc thi hành các dịch vụ tự động như duyệt thông tin, đánh giá thông tin, tổng hợp thông tin, lập các bảng dữ liệu (thống kê, đánh giá, báo cáo), ghi nhận sở thích cá nhân, crawling, và rất nhiều dịch vụ thông minh khác nhằm hỗ trợ tối đa và tốt nhất đối với nhu cầu người dùng.
- Hiểu ngữ nghĩa của những từ mà người dùng muốn tìm (ngữ nghĩa của các từ sẽ được mô tả trong bộ Ontology tương ứng trong từng lĩnh vực).

#### **5.4.2.3 Các Semantic Search Engine hiện có**

Xem phụ lục C trang 222.

## **Chương 6 ỨNG DỤNG MINH HỌA : TÌM KIẾM THÔNG TIN CỦA MỘT NGƯỜI, MỘT NHÓM NGƯỜI VÀ KHẢ NĂNG CỦA HỌ TRONG MỘT TỔ CHỨC**

### **6.1 Giới thiệu ứng dụng**

Hiện nay, các máy tính đang thay đổi từ những thiết bị đơn lẻ, độc lập thành những điểm truy cập đến một mạng trao đổi thông tin và giao dịch thương mại (gọi là World Wide Web). Tuy nhiên, sự thành công của mạng WWW đã làm cho việc tìm kiếm, truy cập, trình bày và duy trì thông tin cho người sử dụng trở nên ngày càng khó khăn vì số lượng dữ liệu và người dùng quá không lồ. Để đối phó với vấn đề này, nhiều cuộc nghiên cứu mới và các tổ chức kinh doanh thương mại được thành lập để làm phong phú nguồn thông tin có sẵn bằng những ngữ nghĩa mà máy tính có thể xử lý. Công nghệ Semantic Web ra đời đã cung cấp khả năng truy cập thông minh đến những thông tin phân tán và hỗn tạp, cho phép các sản phẩm phần mềm làm cầu nối giữa nhu cầu người dùng với những nguồn thông tin có sẵn.

Cũng dựa trên quan điểm trên cùng với những kiến thức tìm hiểu được trong phần lý thuyết về Semantic Web, nhóm chúng em quyết định xây dựng một ứng dụng minh họa để hiểu rõ hơn về lĩnh vực Semantic Web này. Ứng dụng được xây dựng nhằm phục vụ cho việc lưu trữ và tìm kiếm thông tin cho một tổ chức (ví dụ như trường đại học, bệnh viện, viện nghiên cứu.. hay có thể mở rộng hơn là mạng toàn cầu ). Cụ thể hơn là hệ thống có thể hỗ trợ người dùng tìm kiếm thông tin của một người, nhóm người hay khả năng của họ trong một tổ chức cụ thể nào đó.

Quá trình tìm kiếm thông tin của người dùng được mô tả một cách đơn giản sau: Người dùng vào hệ thống, cung cấp thông tin cần tìm theo ba lựa chọn (hoặc là

thông tin cá nhân, hoặc là thông tin nhóm nghiên cứu, hoặc là khả năng). Hệ thống sẽ hỗ trợ hai tiêu chí tìm kiếm là tìm chính xác và tìm gần đúng. Với cách tìm chính xác hệ thống sẽ yêu cầu nhập đầy đủ và chính xác họ tên, kết quả trả về là thông tin về người này nếu có tồn tại trong hệ thống. Với cách tìm gần đúng, người dùng chỉ cần nhập một phần trong chuỗi tên cần tìm, kết quả trả về là tập các tên trong đó có một phần giống với chuỗi gõ vào (hệ thống hỗ trợ cách tìm có dấu và không dấu). Sau đó người dùng có thể chọn chức năng hiển thị thông tin chi tiết hay hiển thị thông tin dưới dạng lược đồ để xem chi tiết thông tin cần tìm.

Ngoài ra hệ thống còn cung cấp chức năng cập nhật thông tin của con người, về nhóm nghiên cứu hay về khả năng của họ trong tổ chức, các chức năng cập nhật bao gồm: thêm, xóa, sửa thông tin. Quá trình cập nhật có thể mô tả chi tiết như sau: Người dùng (hay là người quản trị hệ thống) sẽ tự động cập nhật thông tin về cá nhân, nhóm người hay khả năng khi có yêu cầu thay đổi, những thay đổi này sẽ được lưu lại trong hệ thống.

Hơn nữa hệ thống còn hỗ trợ các chức năng cập nhật thông tin cá nhân thông qua các URL trên hệ thống mạng toàn cầu. Quá trình cập nhật hệ thống thông qua các URL thực hiện như sau: người quản trị hệ thống nhập đường dẫn của một URL và mức độ tìm kiếm (độ sâu liên kết giữa các trang web). Sau đó hệ thống kiểm tra có thông tin theo địa chỉ URL được nhập vào có liên quan đến các thành viên đang có trong hệ thống hay không? Nếu có người quản trị sẽ chọn những thành viên nào cần cập nhật lại thông tin, lúc đó hệ thống sẽ cập nhật thêm địa chỉ URL này cho các thành viên được lựa chọn. Bên cạnh đó hệ thống cũng hỗ trợ cập nhật các địa chỉ URL được tìm thấy thông qua việc tìm kiếm tên của một người cụ thể bằng công cụ tìm kiếm - Google. Quá trình này được mô tả chi tiết như sau: người quản trị hệ thống sẽ nhập một tên có trong hệ thống và thực hiện tìm kiếm tất cả các URL có liên quan đến người này trên hệ thống mạng toàn cầu (giống như cách tìm của Google). Sau đó người quản trị chọn các URL cần cập nhật và yêu cầu hệ thống cập nhật thêm các URL cho cá nhân này.



## 6.2 Các yêu cầu của ứng dụng

### 6.2.1 Các yêu cầu lưu trữ

Với mục tiêu mang tính ngữ nghĩa vào trong dữ liệu, toàn bộ dữ liệu của ứng dụng sẽ được lưu trữ dưới dạng RDF, những file RDF này sẽ liên kết với nhau thông qua các URIs. Hệ thống kế thừa một bộ từ vựng được định nghĩa sẵn đó là FOAF ( **giới thiệu ở mục 3.6**) để mô tả thông tin về *con người*, *nhóm người* và mối quan hệ giữa chúng. Ngoài ra hệ thống cũng tự định nghĩa một *ontology* riêng nhằm mô tả thông tin cho *khả năng* trong hệ thống và mối quan hệ giữa chúng với *con người* và với *nhóm người*.

- Thông tin mô tả tài nguyên con người bao gồm :
  - Các thông tin cá nhân : danh hiệu (Mr, Mrs, Ms, ...), họ, tên, họ và tên, địa chỉ email, trang chủ, website nơi làm việc, website của trường đã học.
  - Các thông tin mô tả khả năng làm việc.
  - Các thông tin mô tả nhóm mà người đó là thành viên.
  - Các thông tin mô tả những người mà người đó biết.
  - Các địa chỉ URI của các website liên quan đến người đó.
- Thông tin mô tả tài nguyên nhóm người bao gồm:
  - Tên nhóm, mô tả về nhóm (nhóm nghiên cứu lĩnh vực gì, các dự án đã thực hiện, đang thực hiện, kết quả nghiên cứu đạt được ...).
  - Các thông tin mô tả nhóm con của nhóm.
  - Các thông tin mô tả những khả năng cần thiết của nhóm.
  - Các thông tin mô tả các thành viên của nhóm.
- Thông tin mô tả một khả năng bao gồm:
  - Tên khả năng.
  - Các thông tin mô tả khả năng này thuộc nhóm người nào.
  - Các thông tin mô tả những người có khả năng này.

## 6.2.2 Các yêu cầu giao diện

Giao diện đồ họa của ứng dụng mô tả trực quan mối quan hệ giữa một người (hay một nhóm người, hay một khả năng) với những người khác, với những nhóm người khác và với những khả năng khác. Nhiệm vụ của giao diện là chuyển một file RDF mô tả một tài nguyên (một người, một nhóm người hay một khả năng) thành một lược đồ (graph) giúp người dùng hình dung và định hướng được mối quan hệ của tài nguyên này với các tài nguyên khác trong tổ chức.

## 6.2.3 Các yêu cầu chức năng

### 6.2.3.1 Nhóm chức năng tìm kiếm

Cho phép người dùng tìm kiếm theo 3 tiêu chí :

- Tên người
- Tên nhóm người
- Tên khả năng

Kết quả tìm được là tất cả các thông tin liên quan đến người (hay nhóm người hay khả năng) và được thể hiện bằng giao diện đồ họa.

### 6.2.3.2 Nhóm chức năng xem chi tiết

Cho phép người dùng xem thông tin chi tiết (toàn bộ nội dung của một tài nguyên) của một người, một nhóm người hay một khả năng, đây cũng chính là toàn bộ nội dung của một file RDF mô tả tài nguyên con người, nhóm người hay khả năng.

### 6.2.3.3 Nhóm chức năng cập nhật tài nguyên

Ngoài các chức năng tra cứu thông tin hệ thống còn có các chức năng để cập nhật tài nguyên :

- Nhóm chức năng cập nhật tài nguyên con người : Thêm một người mới vào hệ thống, xoá một người khỏi hệ thống, chỉnh sửa thông tin một người trong hệ thống, cập nhật địa chỉ (URI) của các trang web liên quan đến

nhiều người trong hệ thống theo địa chỉ URI của trang web, cập nhật địa chỉ (URI) của các trang web liên quan đến một người trong hệ thống theo tên người.

- Nhóm chức năng cập nhật tài nguyên nhóm người : Thêm một nhóm người mới vào hệ thống, xoá một nhóm người khỏi hệ thống, chỉnh sửa thông tin một nhóm người trong hệ thống.
- Nhóm người chức năng cập nhật tài nguyên khả năng : Thêm một khả năng mới vào hệ thống, xoá một khả năng khỏi hệ thống, chỉnh sửa thông tin một khả năng trong hệ thống.

#### **6.2.4 Các yêu cầu phi chức năng**

- Ràng buộc thiết kế : Hệ thống phải cung cấp toàn bộ giao diện web.
- Tính hiệu quả : Kết quả tìm kiếm phù hợp, chính xác, nhanh chóng theo công nghệ Semantic web.
- Tính tiện dụng : Giao diện thân thiện, dễ dùng, người dùng chỉ gõ một thuật ngữ rồi nhấn Enter là ứng dụng sẽ thực thi.
- Tính tương thích : Người dùng chỉ cần một trình duyệt web và kết nối được với server.
- Tính bảo mật : Do mô hình client-server, người dùng chỉ xem được kết quả tra cứu dưới dạng tĩnh (htm/html).
- Tính dễ bảo trì : Thêm chức năng dễ dàng, phát triển hay thêm các Ontology thuận lợi.
- Tính dễ sửa lỗi : Do các Ontology độc lập nhau, do các tài liệu độc lập nhau, nên việc phát hiện lỗi là đơn giản.
- Tính dễ mở rộng : Hệ thống dễ mở rộng cho các tổ chức lớn hơn hay cho phạm vi toàn cầu.

- Sự hỗ trợ : Hệ thống hỗ trợ người dùng nhập thông tin tìm kiếm dưới dạng chữ tiếng Việt có dấu (theo phong chữ Unicode) hay chữ tiếng Việt không dấu.

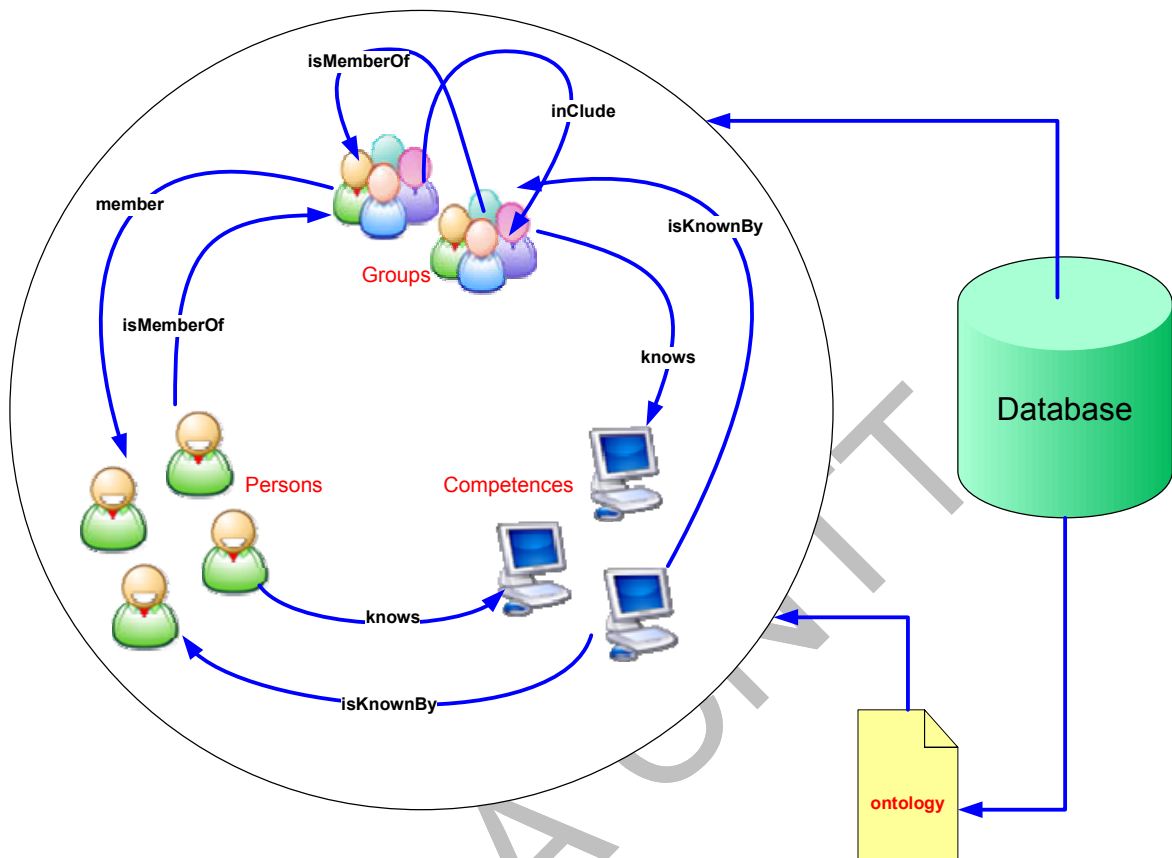
### **6.3 Hướng tiếp cận và giải pháp**

Như đã trình bày ở phần giới thiệu một trong những mục đích của ứng dụng là nhúng ngữ nghĩa cho tài liệu lưu trữ và với những kiến thức nghiên cứu được về công nghệ Semantic, chúng em nhận thấy cần phải áp dụng một trong những kỹ thuật chính của Semantic Web đó là cách lưu trữ thông tin với RDF.

Bên cạnh đó chúng em sẽ xem xét lựa chọn các ontology sẵn có trên mạng để mô tả thông tin cho các tài nguyên của ứng dụng. Vì một trong những tài nguyên cần mô tả là con người và nhóm người nên chúng em chọn bộ từ vựng (ontology) của FOAF (**giới thiệu ở phần 3.6**) để mô tả cho người, nhóm người và mối quan hệ giữa chúng. Đối với tài nguyên khả năng của ứng dụng chúng em sẽ tự xây dựng một từ vựng (*ontology*) riêng để mô tả cho các khả năng, mối quan hệ giữa chúng với người và với nhóm người.

Với những yêu cầu của ứng dụng về giao diện đồ họa : chúng em đã tiếp cận, tìm hiểu thông qua các bài báo, tài liệu và những ứng dụng Semantic Web đã được xây dựng trên thế giới và nhận thấy có thể áp dụng phương pháp vẽ vector *SVG* (Scalable Vector Graphics) [37] để thực hiện.

Riêng việc lưu trữ dữ liệu dưới dạng RDF chúng em đặt ra các mục tiêu lưu trữ như sau: do hệ thống cần lưu trữ thông tin cho *con người*, *nhóm người* và *khả năng* cho nên dữ liệu sẽ bao gồm 3 loại file chính: một file mô tả thông tin cho người, một file mô tả thông tin cho nhóm người và một file mô tả thông tin cho khả năng. Mỗi một người, nhóm người hay một khả năng sẽ được lưu trữ dưới một file riêng biệt và liên kết với nhau thông qua các URIs. Với mục tiêu đề ra như vậy chúng em đưa ra một mô hình nhằm mô tả các file và sự liên kết giữa các file này với nhau như hình



**Hình 6-1** Mô hình biểu diễn sự liên kết giữa các loại dữ liệu.

Vấn đề còn lại là việc truy vấn dữ liệu RDF để phục vụ yêu cầu tìm kiếm thông tin. Với những kiến thức nghiên cứu được về cách truy vấn RDF chúng em sẽ sử dụng ngôn ngữ truy vấn SPARQL để áp dụng vào ứng dụng.

Và sau đây là giải pháp cho ứng dụng :

- Môi trường thực hiện ứng dụng : ASP.NET.
- Các ngôn ngữ được sử dụng để viết ứng dụng:
  - Ngôn ngữ Web: C#, JavaScript.
  - Ngôn ngữ mô tả semantic: RDF (tìm hiểu ở chương Chương 3), RDF/XML [3.3], RDFS [3.5], OWL [2.5].

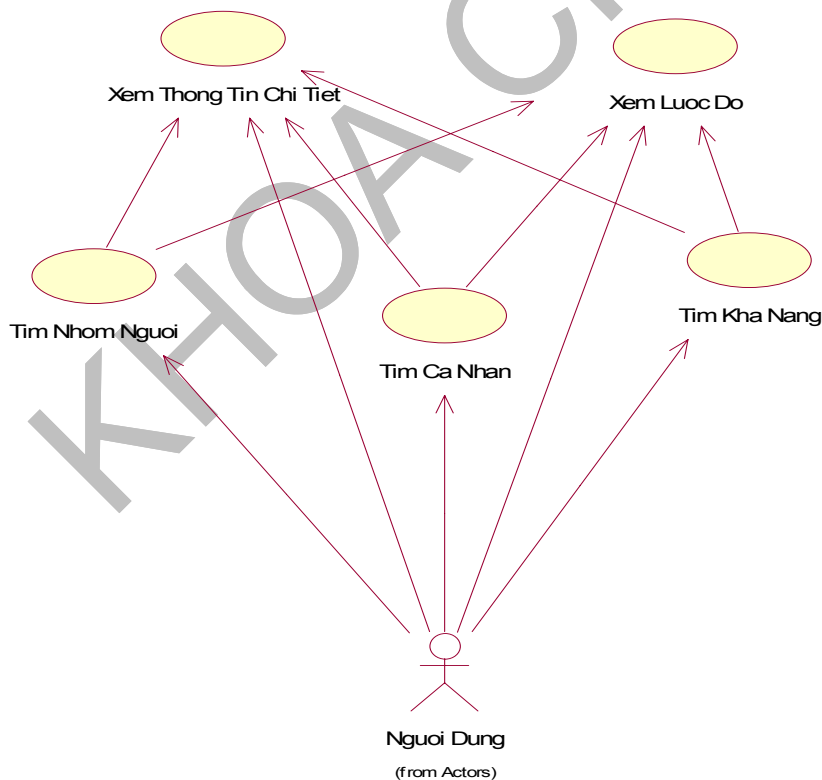
- Ngôn ngữ truy vấn dữ liệu RDF : SPARQL (RDF Query Language) [Chương 4].
- Ngôn ngữ dùng để vẽ lược đồ trên giao diện đồ họa của ứng dụng : SVG (Scalable Vector Graphics) [37]
- Ngôn ngữ dùng để chuyển từ file RDF sang định dạng SVG : XSLT (Extensible Stylesheet Language (XSL) Transformations) [39]
- Các *softwares* và *tools* hỗ trợ việc xây dựng ứng dụng:
  - **SVGView** (Version 0.4a) : là một SVG Software hỗ trợ việc hiển thị file SVG trên trình duyệt web Internet Explorer [38].

## Chương 7 PHÂN TÍCH ỨNG DỤNG

### 7.1 Mô hình use-case

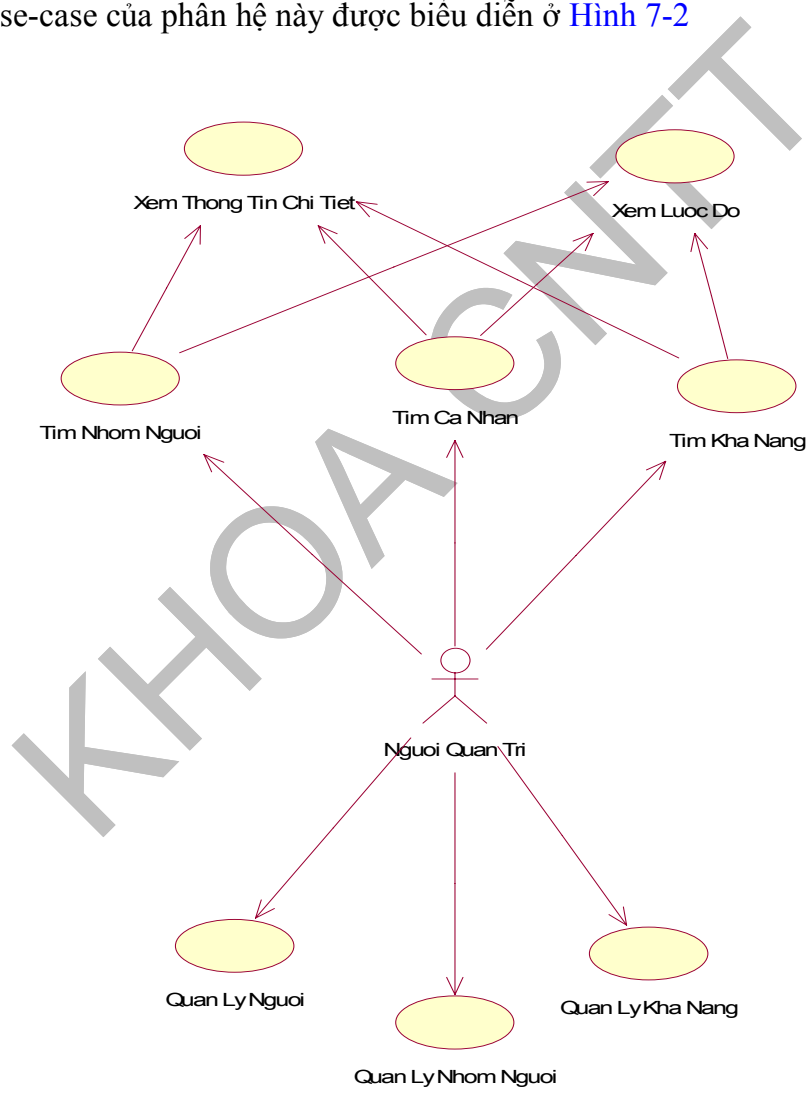
Dựa trên đặc tả các chức năng của ứng dụng được mô tả ở phần 6.2.3 trên, chúng em nhận thấy cần chia hệ thống ra làm hai phân hệ :

- Phân hệ thứ nhất dành cho người sử dụng (user) : phân hệ này sẽ cung cấp cho người dùng các chức năng tìm kiếm thông tin về người, nhóm người, khả năng; xem lược đồ biểu diễn thông tin của người, nhóm người, khả năng; xem thông tin chi tiết của người, nhóm người, khả năng. Mô hình use-case xây dựng cho phân hệ này được thể hiện ở [Hình 7-1](#)



**Hình 7-1** Mô hình Use case cho phân hệ người dùng

- Phân hệ thứ hai là phân hệ dành cho người quản trị (admin) : Ngoài những chức năng giống như các chức năng dành cho phân hệ người dùng, phân hệ này còn cung cấp cho người quản trị các chức năng sau : Quản lý tài nguyên người (thêm, xoá, sửa thông tin người; cập nhật các trang web liên quan đến một người hay nhiều người), quản lý tài nguyên nhóm người (thêm, xoá, sửa thông nhóm người), quản lý tài nguyên khả năng (thêm, xoá, sửa thông tin khả năng). Mô hình use-case của phân hệ này được biểu diễn ở [Hình 7-2](#)



**Hình 7-2** Mô hình Use Case cho phân hệ người quản trị (admin)



## 7.2 Đặc tả use case

Sau đây chúng em sẽ mô tả chi tiết từng chức năng của hệ thống :

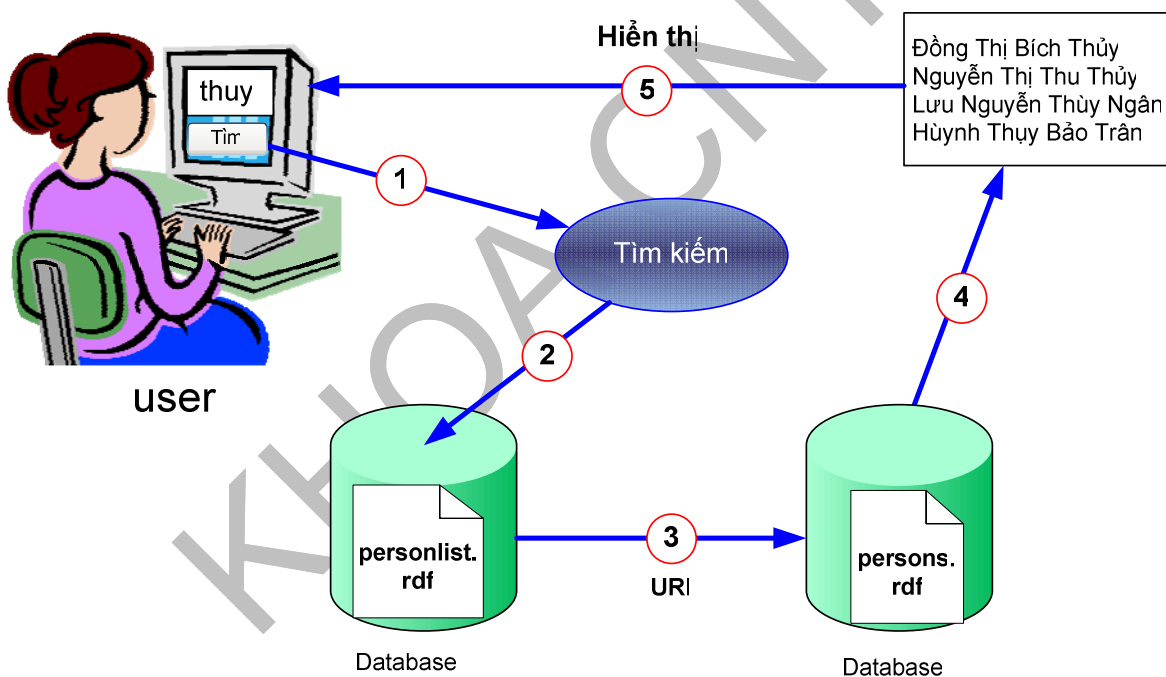
### 7.2.1 Chức năng Tìm Người

- Tóm tắt :

Chức năng này cho phép người dùng tìm một người trong hệ thống.

- Hình minh họa :

**Hình 7-3** sau minh họa cho quá trình người dùng muốn tìm một người trong hệ thống :



**Hình 7-3** Mô tả chức năng tìm người.

- Giải thích cho mô hình trên:

Người dùng nhập tên người muốn tìm (có thể nhập tên không dấu hay chỉ nhập một từ có trong tên) rồi nhấn nút “**Tìm**” để xác nhận chức năng **tìm kiếm** (1). Hệ thống sẽ vào cơ sở dữ liệu (*personlist.rdf*) tìm tất cả các URIs cho cá nhân (2). Dựa vào các

URIs đã tìm được, hệ thống sẽ theo địa chỉ này(*persons.rdf*) để tìm thông tin cá nhân muốn tìm(3). Sau đó trả về tất cả các kết quả tìm được(4)(theo 2 thuật toán tìm chính xác và tìm gần đúng) và hiển thị thông tin tìm được cho người dùng(5).

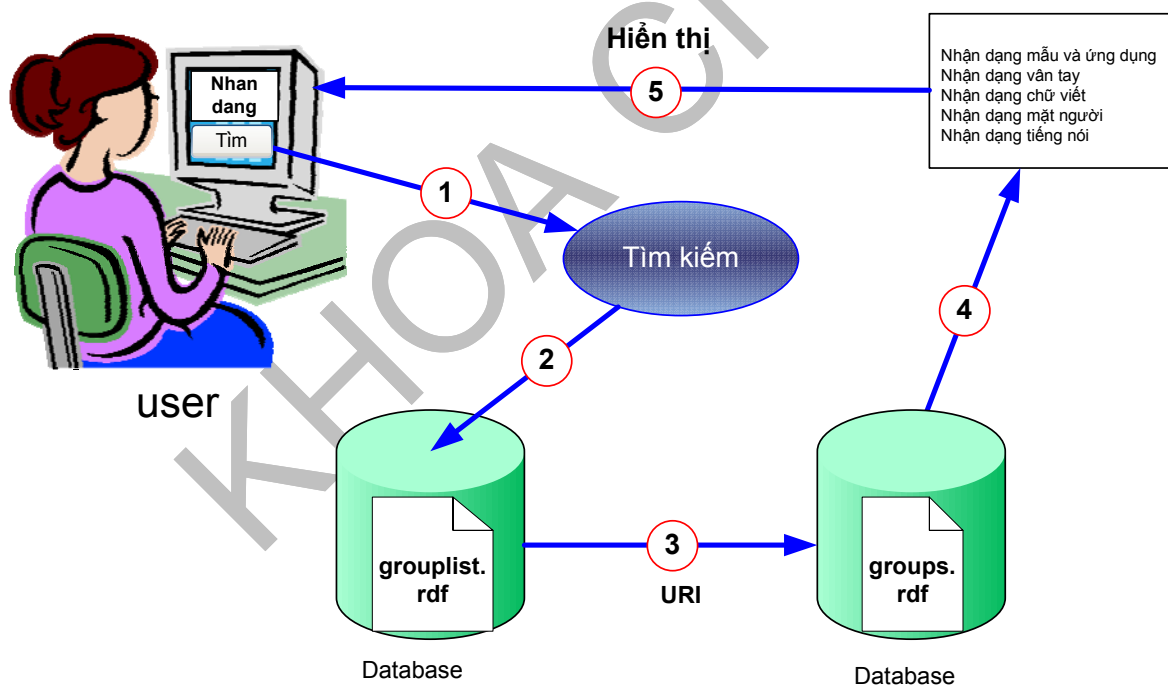
### 7.2.2 Chức năng Tìm Nhóm Người

- Tóm tắt :

Chức năng này cho phép người dùng tìm một nhóm người trong hệ thống.

- Hình minh họa :

*Hình 7-4* sau minh họa cho quá trình người dùng muốn tìm một nhóm người trong hệ thống :



**Hình 7-4** Mô tả chức năng tìm nhóm người.

- Giải thích mô hình trên:

Người dùng nhập tên nhóm người muốn tìm (có thể nhập tên không dấu hay chỉ nhập một từ có trong tên) rồi nhấn nút “**Tìm**” để xác nhận chức năng **tìm kiếm** (1). Hệ

thống sẽ vào cơ sở dữ liệu(*grouplist.rdf*) tìm tất cả các URIs cho nhóm người (2). Dựa vào các URIs đã tìm được, hệ thống sẽ theo địa chỉ này(*groups.rdf*) để tìm thông tin nhóm muốn tìm(3). Sau đó trả về tất cả các kết quả tìm được(4)( theo 2 thuật toán tìm chính xác và tìm gần đúng) và hiển thị thông tin tìm được cho người dùng(5).

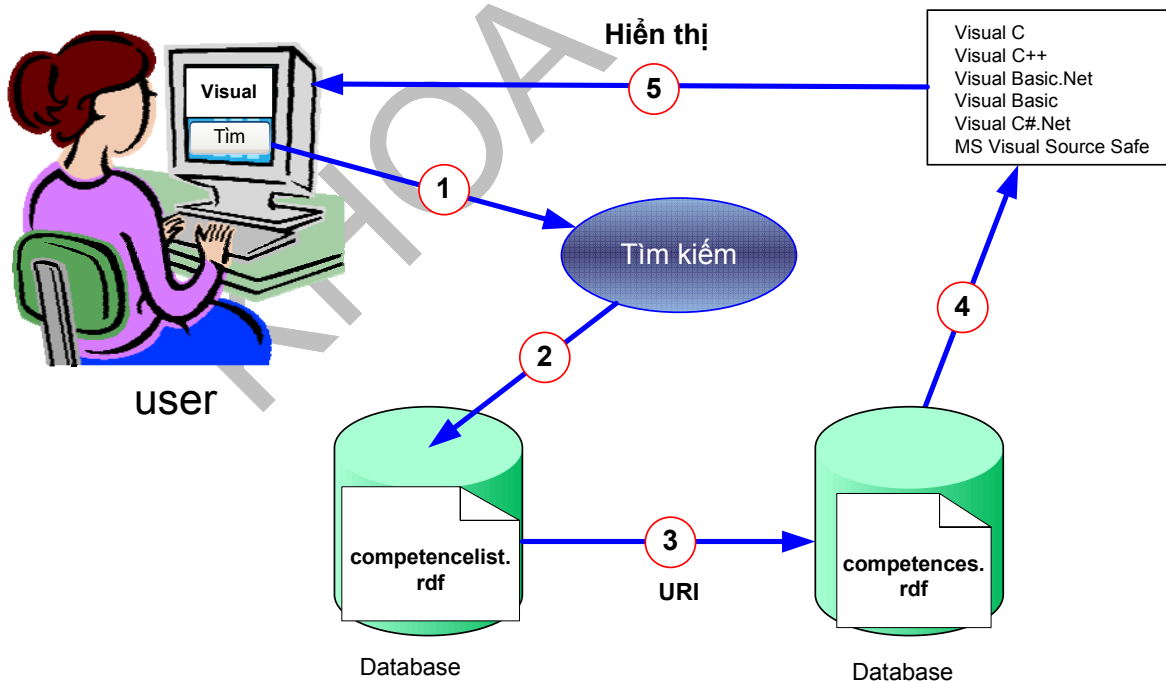
### 7.2.3 Chức năng Tìm Khả Năng

- Tóm tắt :

Chức năng này cho phép người dùng tìm một khả năng trong hệ thống.

- Hình minh họa :

**Hình 7-5** sau minh họa cho quá trình người dùng muốn tìm một khả năng trong hệ thống :



**Hình 7-5** Mô tả chức năng tìm khả năng.

- Giải thích mô hình trên:

Người dùng nhập tên khả năng muốn tìm (có thể nhập tên không dấu hay chỉ nhập một từ có trong tên) rồi nhấn nút “**Tìm**” để xác nhận chức năng **tìm kiếm** (1). Hệ thống sẽ vào cơ sở dữ liệu (*competencelist.rdf*) tìm tất cả các URIs cho nhóm người (2). Dựa vào các URIs đã tìm được, hệ thống sẽ theo địa chỉ này (*competences.rdf*) để tìm thông tin nhóm muốn tìm (3). Sau đó trả về tất cả các kết quả tìm được (4) (theo 2 thuật toán tìm chính xác và tìm gần đúng) và hiển thị thông tin tìm được cho người dùng (5).

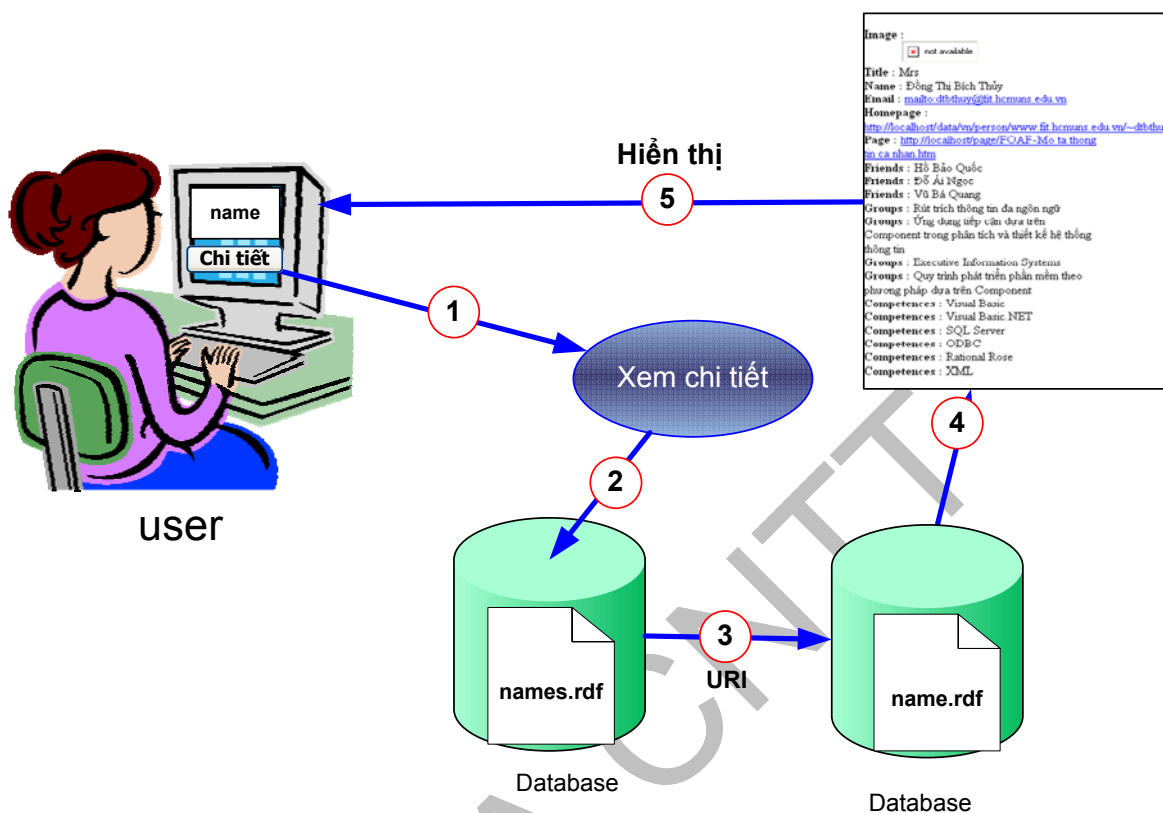
#### 7.2.4 Chức năng Xem Thông Tin Chi Tiết

- Tóm tắt :

Use case thực hiện chức năng tìm xem thông tin chi tiết của một đối tượng trong hệ thống (người, nhóm, khả năng).

- Hình minh họa :

Hình 8.4 sau minh họa cho quá trình người dùng muốn xem thông tin chi tiết của một người, một nhóm hay một khả năng :



**Hình 7-6** Mô tả chức năng xem thông tin chi tiết.

- Giải thích :

Khi người dùng muốn xem thông tin chi tiết của một người, một nhóm hay một khả năng : Người dùng nhấn chọn tên đối tượng (người, nhóm hay khả năng) muốn xem thông tin chi tiết và nhấn chọn nút “**Chi tiết**” để xác nhận việc xem thông tin chi tiết (1). Hệ thống sẽ vào cơ sở dữ liệu tìm trong file *names.rdf* (tùy theo lựa chọn mà name có thể là person, group hay competence) URIs cho tài nguyên cần tìm (2). Dựa vào các URIs đã tìm được, hệ thống sẽ theo địa chỉ này(*name.rdf*) để tìm thông tin chi tiết của tài nguyên muốn tìm(3). Sau đó trả về thông tin chi tiết của tài nguyên (4) và hiển thị thông tin tìm được cho người dùng(5).

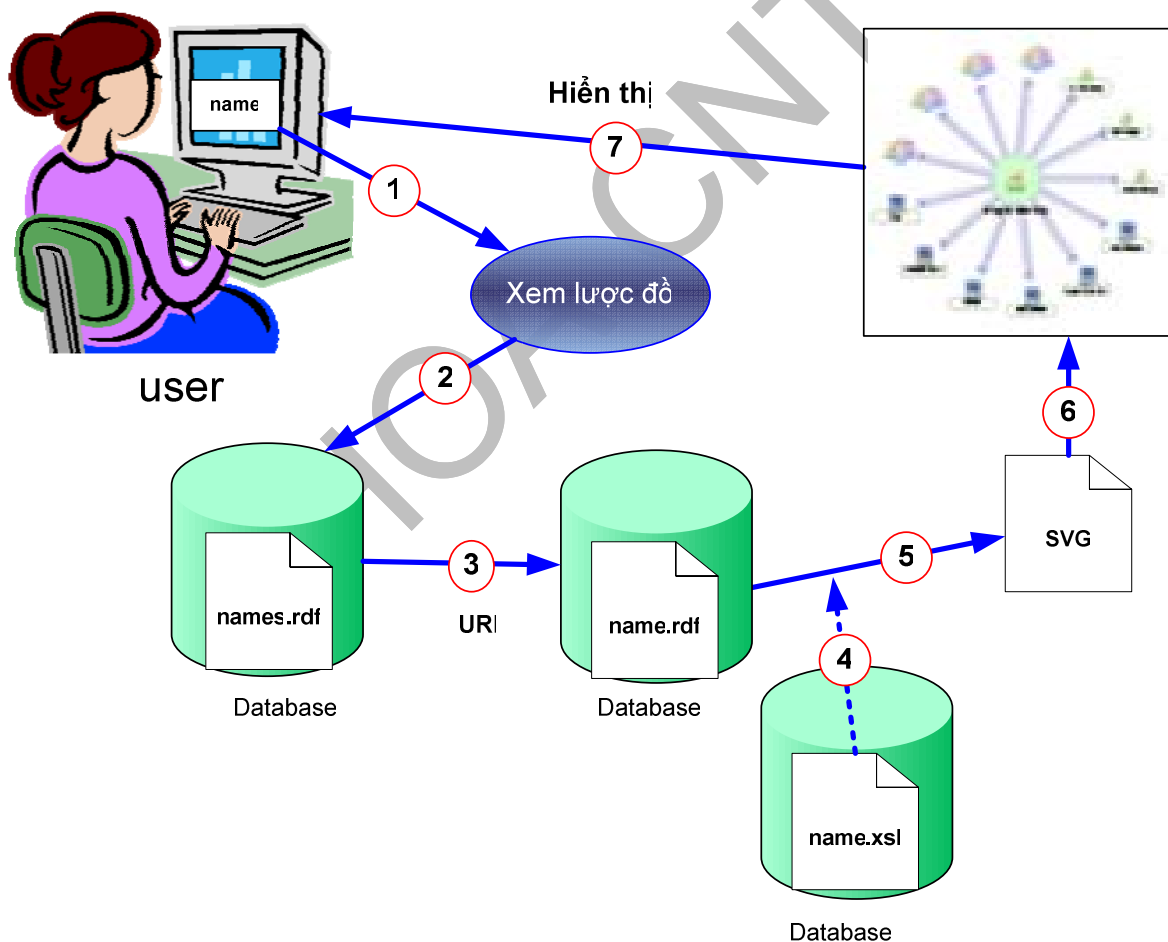
### 7.2.5 Chức năng Xem Lược Đồ

- Tóm tắt :

Use case này mô tả cách thức 1 người dùng muốn xem lược đồ biểu diễn mối quan hệ giữa 1 người, 1 nhóm hay 1 khả năng với những người khác, những nhóm khác và những khả năng khác trong hệ thống.

- Hình minh họa :

*Hình 7-5* sau minh họa cho quá trình người dùng muốn xem lược đồ của một người, một nhóm hay một khả năng :



*Hình 7-7* Mô tả chức năng xem lược đồ.

- Giải thích :

Khi người dùng muốn xem thông tin chi tiết của một người, một nhóm hay một khả năng dưới dạng lược đồ: Người dùng nhấn chọn tên đối tượng (người, nhóm hay khả năng) để xác nhận việc xem thông tin chi tiết (1). Hệ thống sẽ vào cơ sở dữ liệu tìm trong file *names.rdf* (tùy theo lựa chọn mà name có thể là person, group hay competence) URIs cho tài nguyên cần tìm (2). Dựa vào các URIs đã tìm được, hệ thống sẽ theo địa chỉ này(*name.rdf*) để tìm lấy nội dung chi tiết của tài nguyên muốn tìm(3). Tiếp theo hệ thống sẽ kết hợp với file *name.xsl* (4) để chuyển từ định dạng *rdf* sang định dạng *svg*(5). Cuối cùng từ file *svg* hệ thống sẽ hiển thị thông tin chi tiết dưới dạng lược đồ (6) và hiển thị cho người dùng(7).

## 7.2.6 Chức năng Quản Lý Người

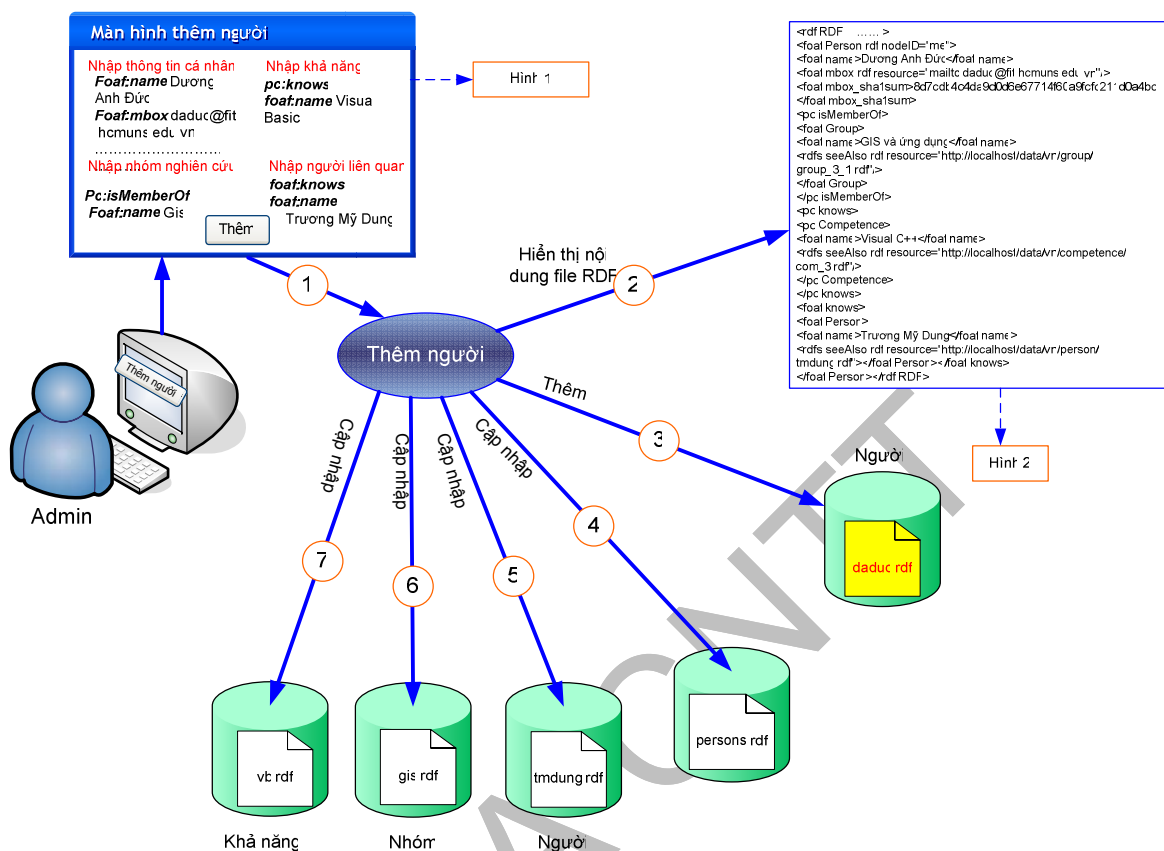
Use case thực hiện chức năng quản lý tài nguyên người. Use case cho phép Admin thực hiện việc thêm, xóa, sửa thông tin của người, cập nhật một trang web liên quan đến nhiều người và cập nhật các trang web liên quan đến một người.

### 7.2.6.1 Thêm người

Chức năng này cho phép admin thêm một người mới vào hệ thống. Xem hình minh họa và giải thích bên dưới.

- Hình minh họa :

**Hình 7-8** sau minh họa cho chức năng thêm người:



**Hình 7-8 Mô tả chức năng thêm người.**

▪ Giải thích :

- Admin nhập thông tin của người mới bao gồm : các thông tin cá nhân, thông tin những người quen, thông tin các nhóm, thông tin các khả năng của người đó ( Hint 1 ) rồi nhấn nút “**Thêm**” để xác nhận việc tạo người mới( 1 ). Hệ thống tự tạo số **mbox\_sha1sum** cho người dựa trên địa chỉ mail của người( mỗi người có một số **mbox\_sha1sum** duy nhất) .
- Hệ thống sẽ hiển thị nội dung file RDF ( Hint 2 ) mô tả thông tin về người mới được tạo ( 2 ), tạo mới file RDF này và lưu file này vào database ( 3 ), thêm thông tin người mới này vào file RDF mô tả danh sách những người có

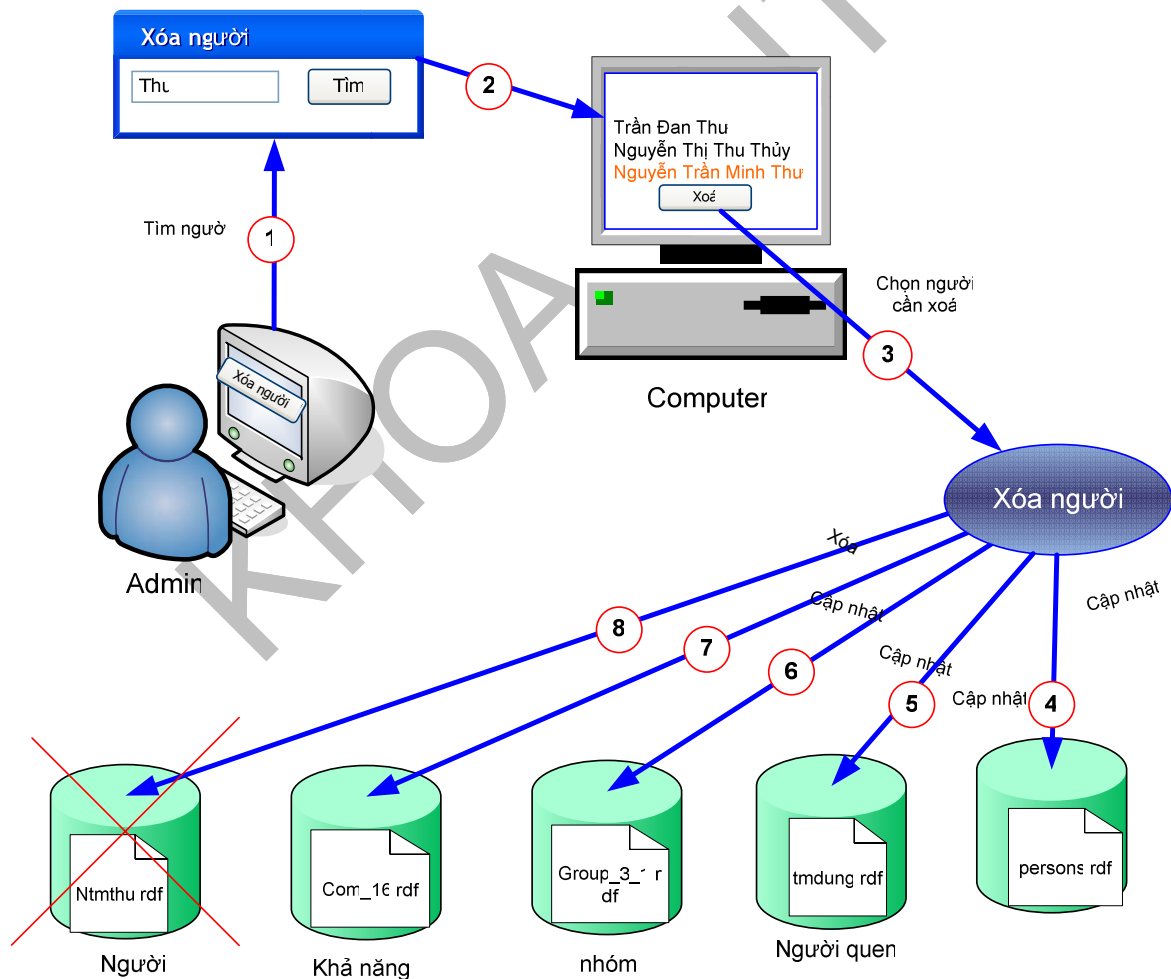


trong hệ thống (4), và cập nhật thông tin của người mới này cho những người có liên quan (5), những nhóm có liên quan (6) và những khả năng có liên quan (7).

### 7.2.6.2 Xóa người

Chức năng này cho phép admin xóa một người ra khỏi hệ thống. Xem hình minh họa và giải thích bên dưới.

- Hình minh họa : **Hình 7-9** mô tả cho chức năng xóa một người ra khỏi hệ thống.



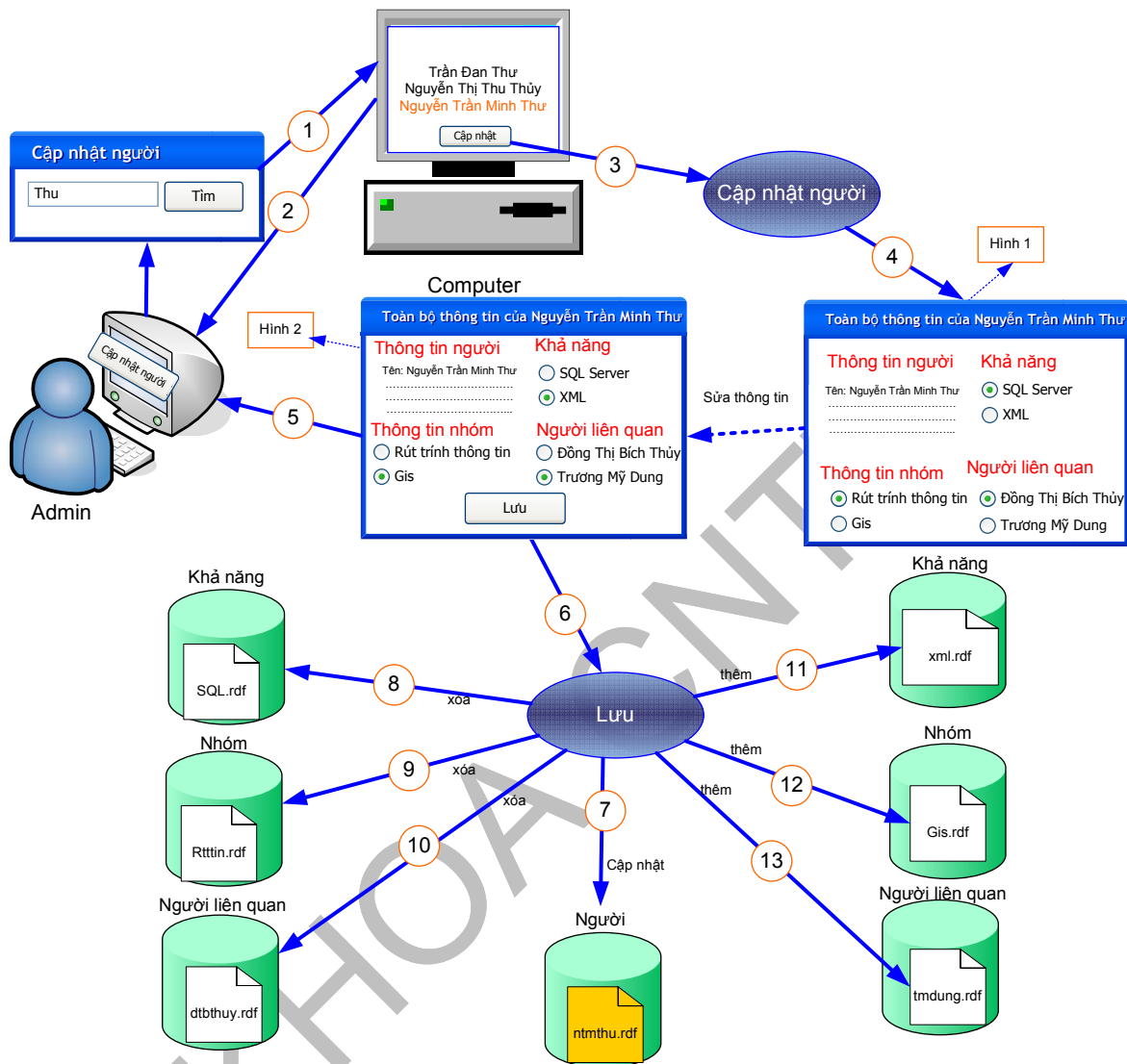
**Hình 7-9** Mô tả chức năng xóa người.

- Giải thích mô hình:
  - Admin nhập tên người muốn xoá và nhấn nút “**Tim**” để tìm xem người muốn xoá có trong hệ thống hay không (1).
  - Hệ thống sẽ vào cơ sở dữ liệu lấy và hiển thị tên người muốn xoá (nếu không tồn tại người này thì hệ thống hiển thị thông tin rỗng) trong danh sách tìm được (2). Admin chọn người muốn xoá trong danh sách tìm được và nhấn nút “**Xoá**” để xác nhận việc xoá người này(3).
  - Hệ thống sẽ xoá thông tin người này khỏi file RDF mô tả danh sách những người có trong hệ thống (4), xoá thông tin của người này khỏi file RDF của những người có liên quan (5), những nhóm có liên quan (6) và những khả năng có liên quan (7), sau cùng hệ thống sẽ xoá file RDF mô tả người này khỏi hệ thống (8).

### 7.2.6.3 Sửa thông tin người

Chức năng này cho phép admin sửa thông tin của một người trong hệ thống. Xem hình minh hoạ và giải thích bên dưới.

- Hình minh hoạ :*Hình 7-10* minh họa cho chức năng sửa thông tin của một người



Hình 7-10 Mô tả chức năng sửa thông tin người.

▪ Giải thích :

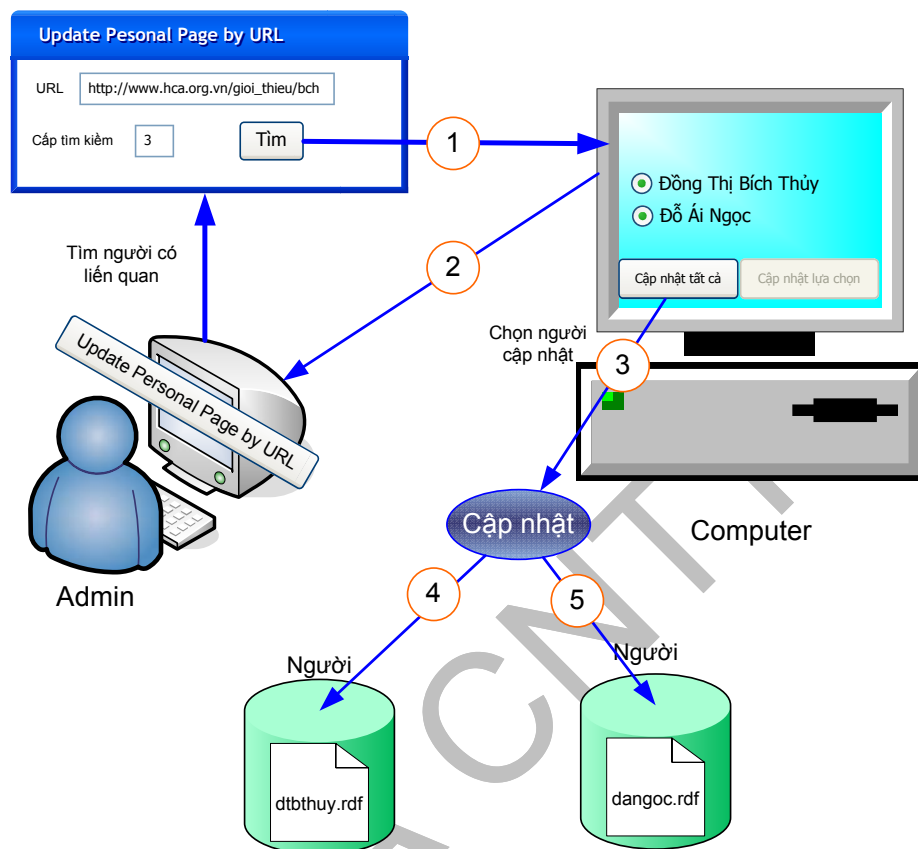
- Admin nhập tên người muốn sửa thông tin và nhấn nút “**Tìm**” để tìm xem người muốn xoá có trong hệ thống hay không (1).
- Hệ thống sẽ vào cơ sở dữ liệu lấy và hiển thị (2) tên người muốn sửa (nếu không tồn tại người này thì hệ thống hiển thị thông tin rỗng) trong danh sách

- tìm được, Admin chọn người muốn sửa trong danh sách tìm được và nhấn nút “**Cập Nhật**” để cập nhật thông tin người này (3).
- Hệ thống sẽ hiển thị (4, 5) tất cả các thông tin về người này để admin chỉnh sửa (Hình 1) đồng thời cho phép admin thêm các thông tin mới về nhóm hay khả năng hay người có liên quan đến người này vào (Hình 2).
  - Admin nhấn nút “**lưu**” để lưu những thông tin đã chỉnh sửa xuống file (6).
  - Hệ thống sẽ hiển thị file RDF mô tả toàn bộ các thông tin mới về người mà admin đã chỉnh sửa, ghi các thông tin mới này vào file RDF mô tả người này trong database (7), cập nhật (xóa) thông tin về người này trong các file RDF của những khả năng, nhóm và những người có liên quan (8, 9, 10) và cập nhật (thêm) thông tin về person này trong các file RDF của những khả năng, nhóm và những người có liên quan (11, 12, 13).

#### 7.2.6.4 Cập nhật trang một trang web liên quan đến nhiều người (*Update Personal Page by URI*)

Chức năng này cho phép admin thêm địa chỉ một trang web vào những người có liên quan trong hệ thống nhằm làm phong phú thêm thông tin cho các cá nhân. Xem hình minh họa và giải thích bên dưới.

- Hình minh họa : **Hình 7-11** minh họa cho quá trình cập nhật một trang web liên quan đến nhiều người



**Hình 7-11** Mô tả chức năng cập nhật một trang web liên quan đến nhiều người

▪ Giải thích :

- Admin nhập vào địa chỉ URL của trang web và cấp tìm kiếm (*level*) trong trang web đó, nhấn nút “**Tìm**” để xác nhận việc tìm kiếm (1).
- Hệ thống sẽ tìm trong trang web và trong tất cả các trang con của trang web (tùy theo độ sâu admin nhập ban đầu) và cho kết quả là tất cả tên của những người trong hệ thống mà trang web có đề cập đến (2).
- Admin kiểm tra lại thông tin của những người được tìm thấy trong trang web và chọn những người nào admin muốn cập nhật địa chỉ URI này vào file RDF mô tả thông của họ. Admin nhấn nút “Cập nhật tất cả” nếu muốn cập nhật co

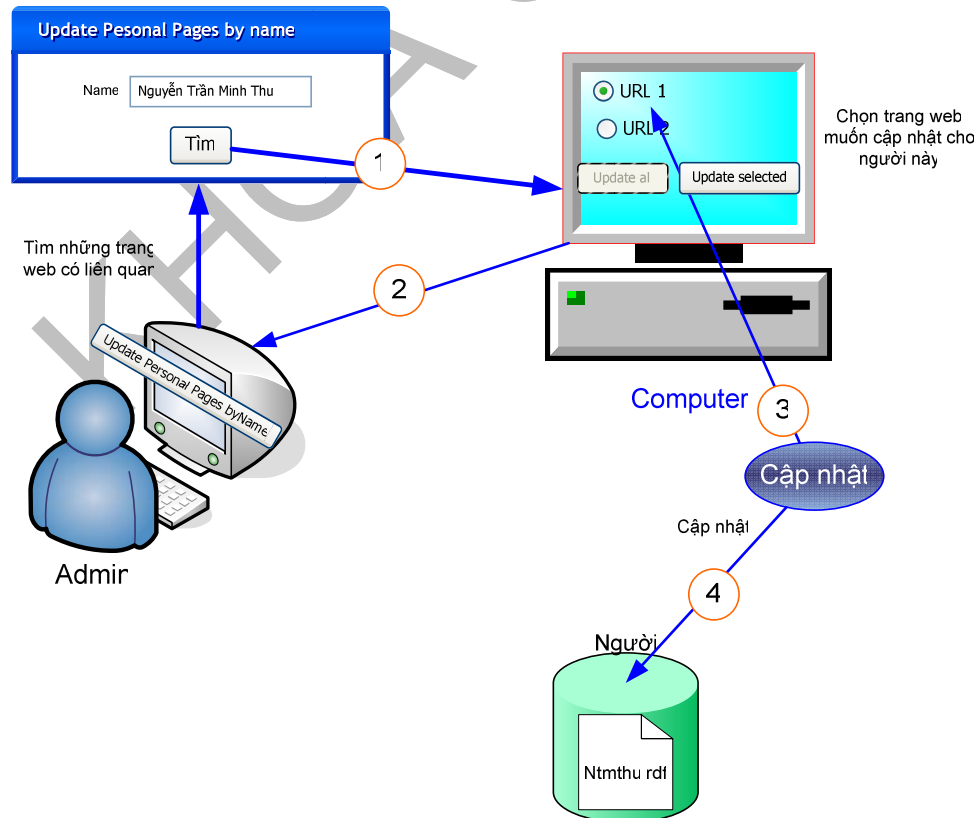
tất cả những người đó hoặc nhấn nút “Cập nhật lựa chọn” để cập nhật cho những người được lựa chọn (3).

- Hệ thống sẽ cập nhật địa chỉ URI này cho những người mà admin đã chọn (4, 5).

### 7.2.6.5 Cập nhật nhiều trang web liên quan đến một người (*Update Personal Pages by Name*)

Chức năng này cho phép admin thêm những địa chỉ web có liên quan cho một người trong hệ thống nhằm làm phong phú thêm thông tin cho người này. Xem hình minh họa và giải thích bên dưới.

- Hình minh họa :**Hình 7-12** Mô tả chức năng cập nhật địa chỉ những trang web liên quan đến một người.



**Hình 7-12** Mô tả chức năng cập nhật địa chỉ những trang web liên quan đến một người .

- Giải thích :
  - Admin nhập tên người muốn tìm các trang web có liên quan, nhấn nút “**Tìm**” để xác nhận việc tìm kiếm (1).
  - Hệ thống sẽ tìm trong tất cả các trang web trên mạng và hiển thị tất cả các địa chỉ URL của các trang web có liên quan đến người này (2).
  - Admin kiểm tra lại và chọn những địa chỉ nào muốn cập nhật cho người này và nhấn nút “Update Selected”, nếu muốn cập nhật tất cả thì nhấn nút “Update All” để xác nhận việc cập nhật (3).
  - Hệ thống sẽ cập nhật các địa chỉ URL admin đã chọn vào file RDF của người này (4).

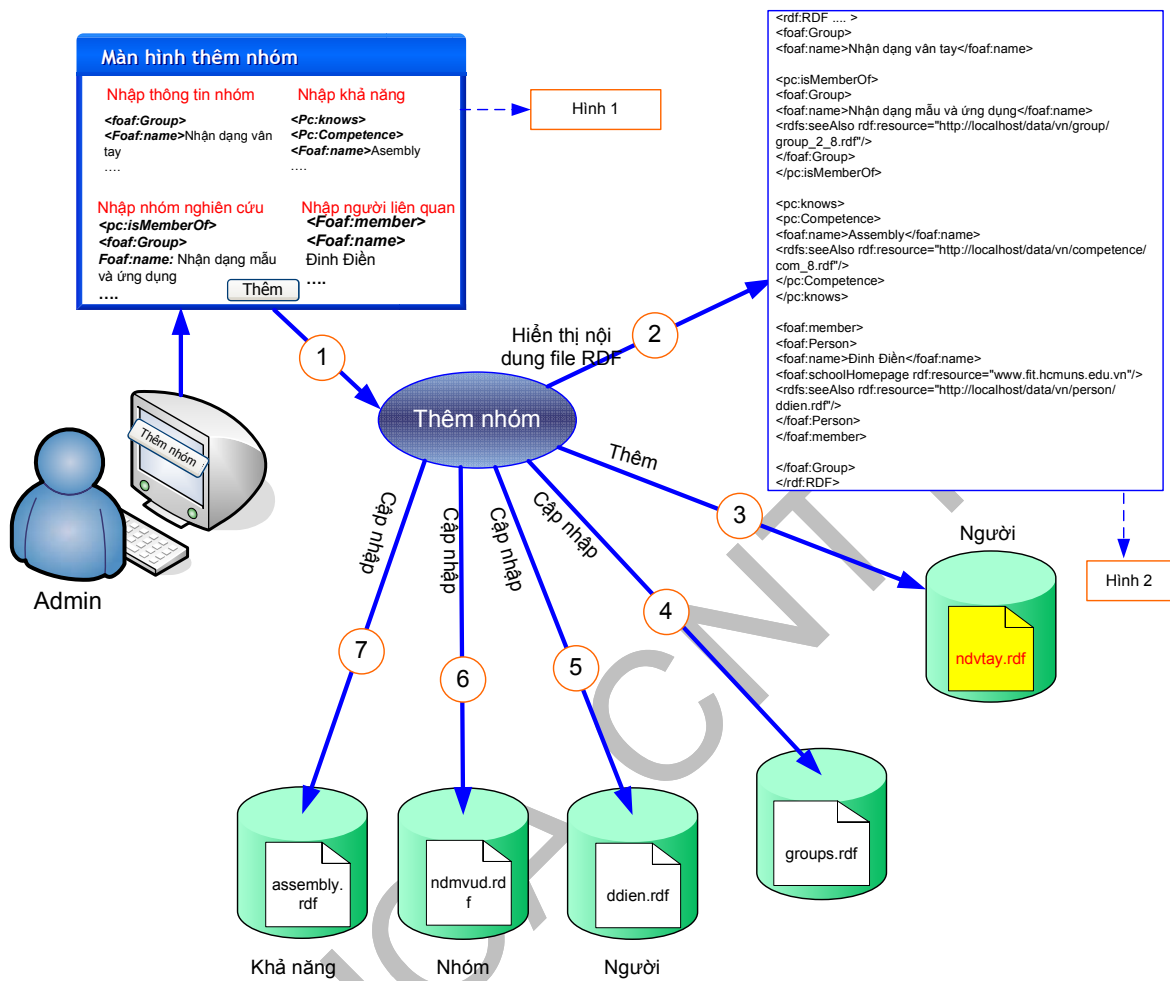
### 7.2.7 Chức năng Quản Lý Nhóm Người

Use case thực hiện chức năng quản lý tài nguyên nhóm người. Use case cho phép Admin thực hiện việc thêm, xoá, sửa thông tin của nhóm người.

#### 7.2.7.1 Thêm nhóm

Chức năng này cho phép admin thêm một nhóm người mới vào hệ thống. Xem hình minh họa và giải thích bên dưới.

- Hình minh họa : **Hình 7-13** minh họa cho quá trình thêm nhóm người :



Hình 7-13 Mô tả chức năng thêm nhóm người.

- Giải thích :
  - Admin nhập thông tin của nhóm mới bao gồm : các thông tin nhóm, thông tin những nhóm con, thông tin các cá nhân liên quan, thông tin các khả năng của nhóm đó ( Hình 1 ) rồi nhấn nút “Thêm” để xác nhận việc tạo nhóm mới( 1 ).
  - Hệ thống sẽ hiển thị nội dung file RDF ( Hình 2 ) mô tả thông tin về nhóm mới được tạo ( 2 ), tạo mới file RDF này và lưu file này vào database ( 3 ),

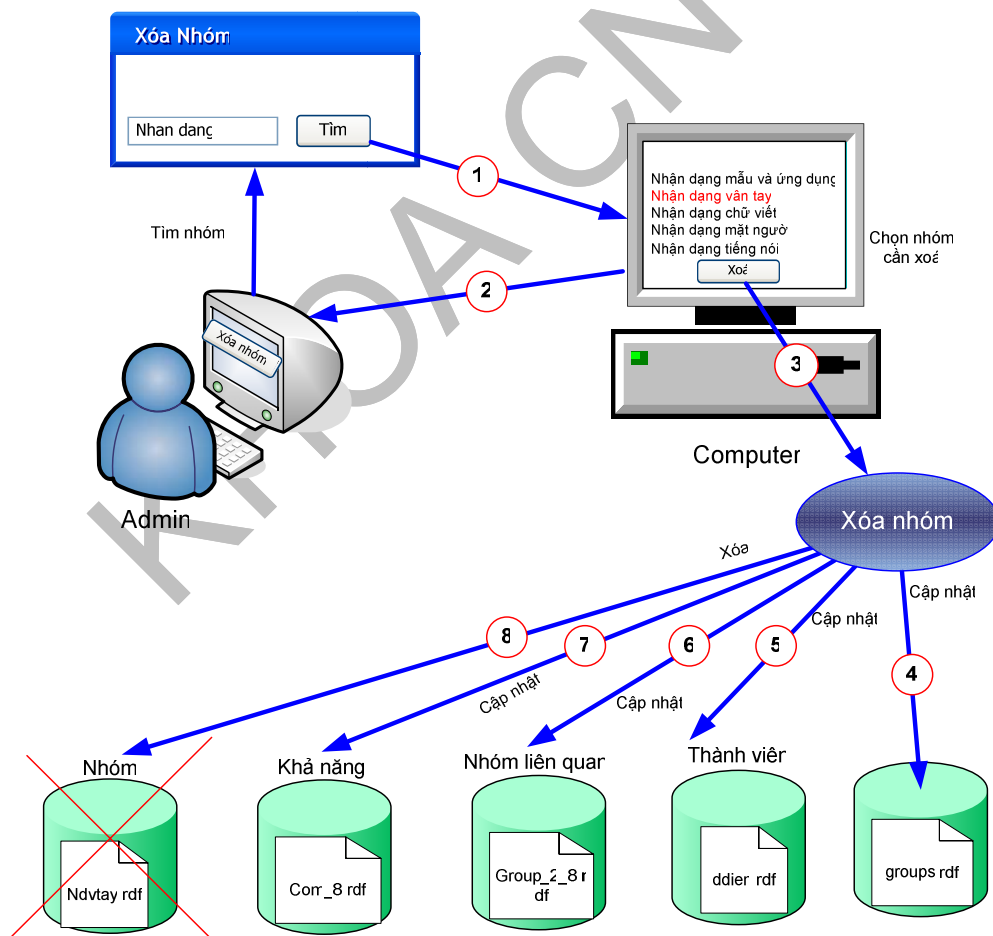


thêm thông tin nhóm mới này vào file RDF mô tả danh sách những nhóm có trong hệ thống (4), và cập nhật thông tin của nhóm mới này cho những người có liên quan (5), những nhóm có liên quan (6) và những khả năng có liên quan (7).

### 7.2.7.2 Xoá nhóm

Chức năng này cho phép admin xoá một nhóm người ra khỏi hệ thống. Xem hình minh họa và giải thích bên dưới.

- Hình minh họa :*Hình 7-14* minh họa cho quá trình xoá nhóm người :



**Hình 7-14** Mô tả chức năng xoá nhóm người.

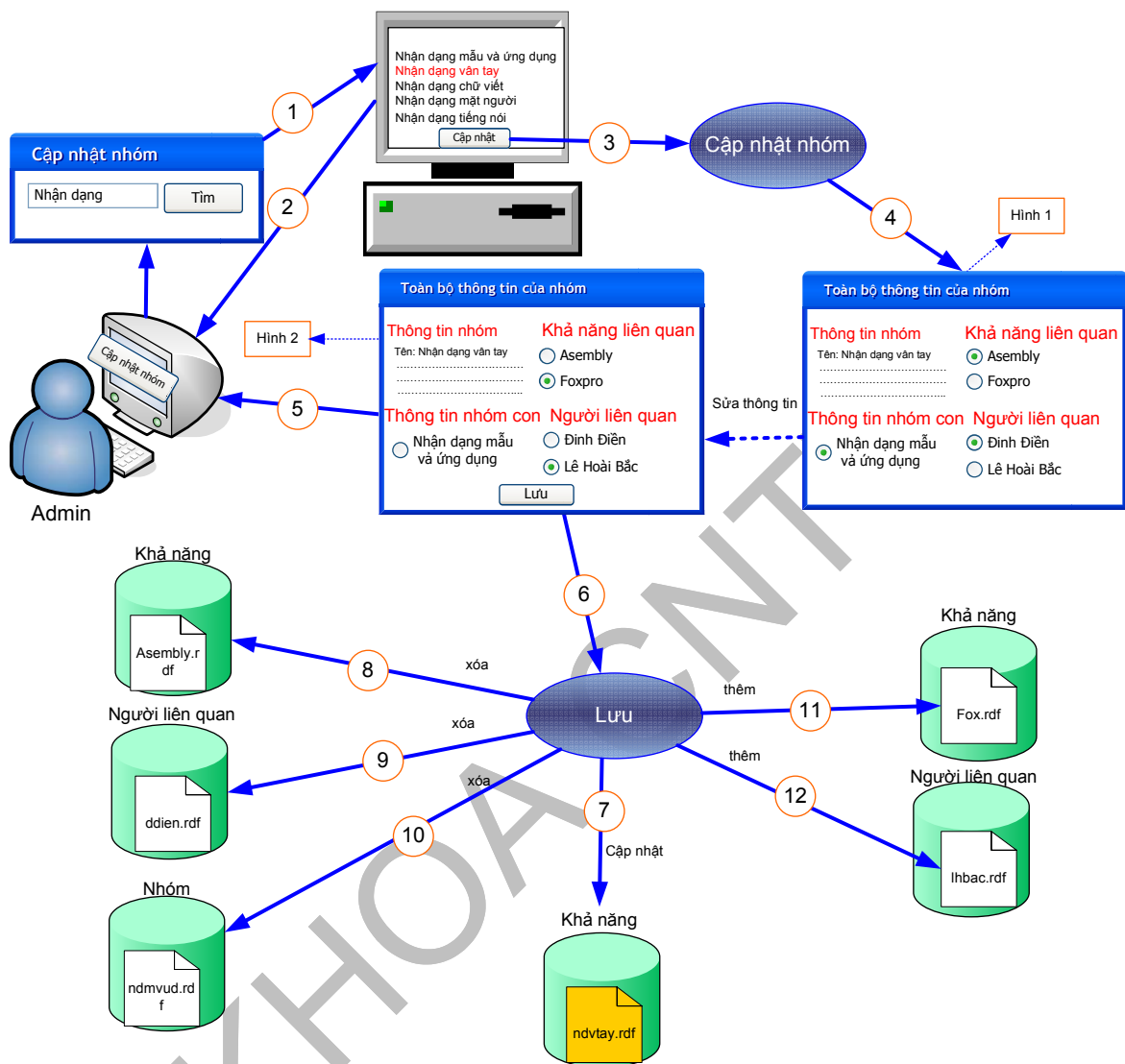
▪ Giải thích :

- Admin nhập tên nhóm người muốn xoá và nhấn nút “**Tìm**” để tìm xem nhóm người muốn xoá có trong hệ thống hay không (1).
- Hệ thống sẽ vào cơ sở dữ liệu lấy và hiển thị tên nhóm người muốn xoá (nếu không tồn tại nhóm người này thì hệ thống hiển thị thông tin rỗng) trong danh sách tìm được (2). Admin chọn nhóm người muốn xoá trong danh sách tìm được và nhấn nút “**Xoá**” để xác nhận việc xoá nhóm người này(3).
- Hệ thống sẽ xoá thông tin nhóm người này khỏi file RDF mô tả danh sách những nhóm người có trong hệ thống (4), xoá thông tin của nhóm người này khỏi file RDF của những thành viên của nhóm (5), những nhóm có liên quan (6) và những khả năng có liên quan (7), sau cùng hệ thống sẽ xoá file RDF mô tả nhóm người này khỏi hệ thống (8).

### 7.2.7.3 Sửa thông tin nhóm

Chức năng này cho phép admin sửa thông tin của một nhóm người trong hệ thống.

- Hình minh họa : *Hình 7-15* minh họa cho chức năng cập nhật thông tin nhóm người.



Hình 7-15 Mô tả chức năng sửa thông tin nhóm người.

▪ Giải thích :

- Admin nhập tên nhóm muốn sửa thông tin và nhấn nút “**Tìm**” để tìm xem nhóm muốn xóa có trong hệ thống hay không (1).
- Hệ thống sẽ vào cơ sở dữ liệu lấy và hiển thị (2) tên người muốn sửa (nếu không tồn tại nhóm này thì hệ thống hiển thị thông tin rỗng) trong danh sách

- tìm được, Admin chọn nhóm muốn sửa trong danh sách tìm được và nhấn nút “**Cập Nhật**” để cập nhật thông tin nhóm này (3).
- Hệ thống sẽ hiển thị (4, 5) tất cả các thông tin về nhóm này để admin chỉnh sửa (Hình 1) đồng thời cho phép admin thêm các thông tin mới về nhóm hay người có liên quan đến người này vào (Hình 2).
  - Admin nhấn nút “**lưu**” để lưu những thông tin đã chỉnh sửa xuống file (6).
  - Hệ thống sẽ hiển thị file RDF mô tả toàn bộ các thông tin mới về nhóm mà admin đã chỉnh sửa, ghi các thông tin mới này vào file RDF mô tả nhóm này trong database (7), cập nhật (xóa) thông tin về nhóm này trong các file RDF của những khả năng, nhóm và những người có liên quan (8, 9, 10) và cập nhật (thêm) thông tin về nhóm này trong các file RDF của những nhóm và những người có liên quan (11, 12).

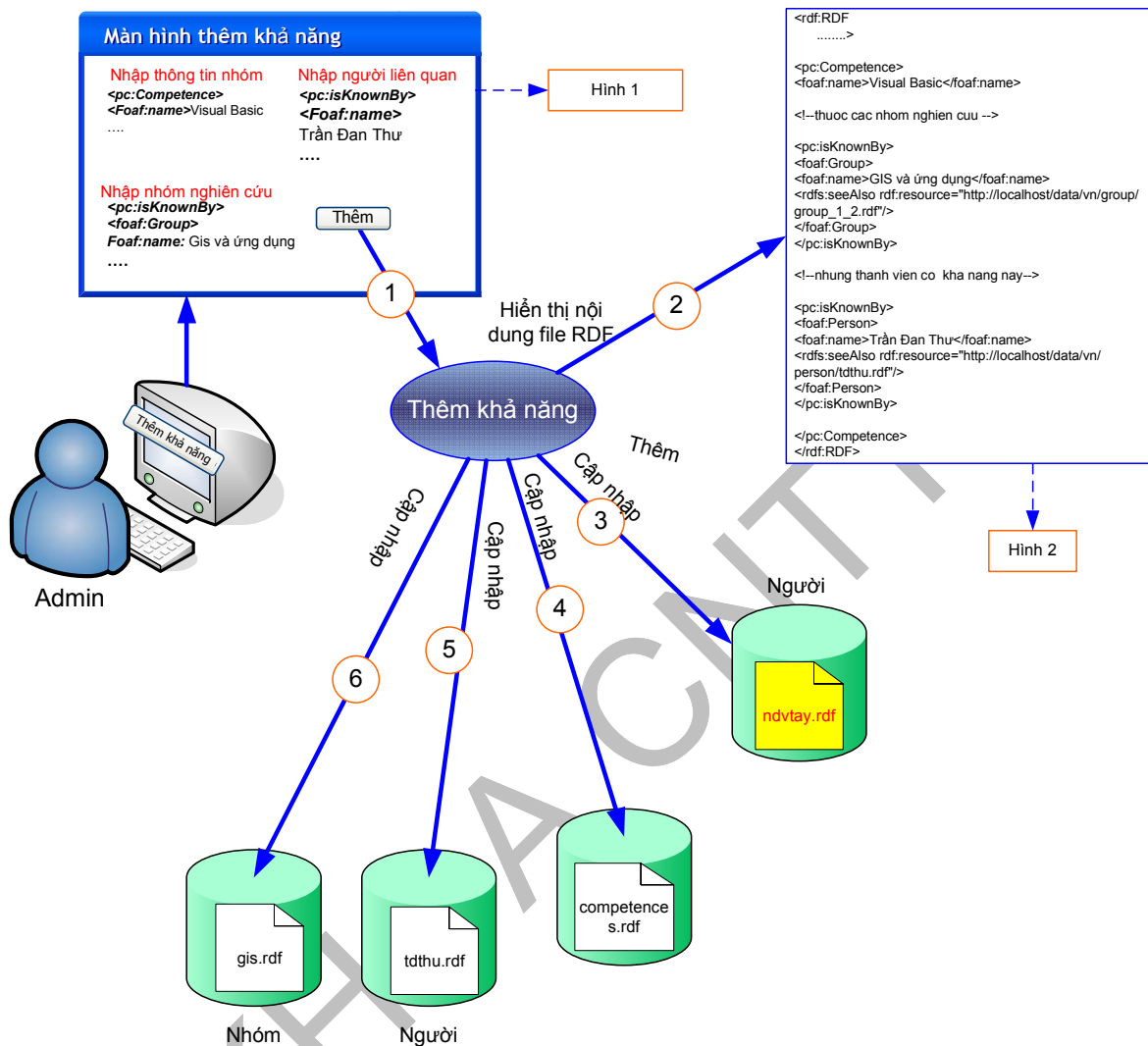
## 7.2.8 Chức năng Quản Lý Khả Năng

Use case thực hiện chức năng quản lý tài nguyên khả năng. Use case cho phép Admin thực hiện việc thêm, xóa, sửa thông tin của khả năng.

### 7.2.8.1 Thêm khả năng

Chức năng này cho phép admin thêm một khả năng mới vào hệ thống. Xem hình minh họa và giải thích bên dưới.

- Hình minh họa : **Hình 7-16** minh họa cho quá trình thêm khả năng :



Hình 7-16 Mô tả chức năng thêm khả năng.

▪ Giải thích :

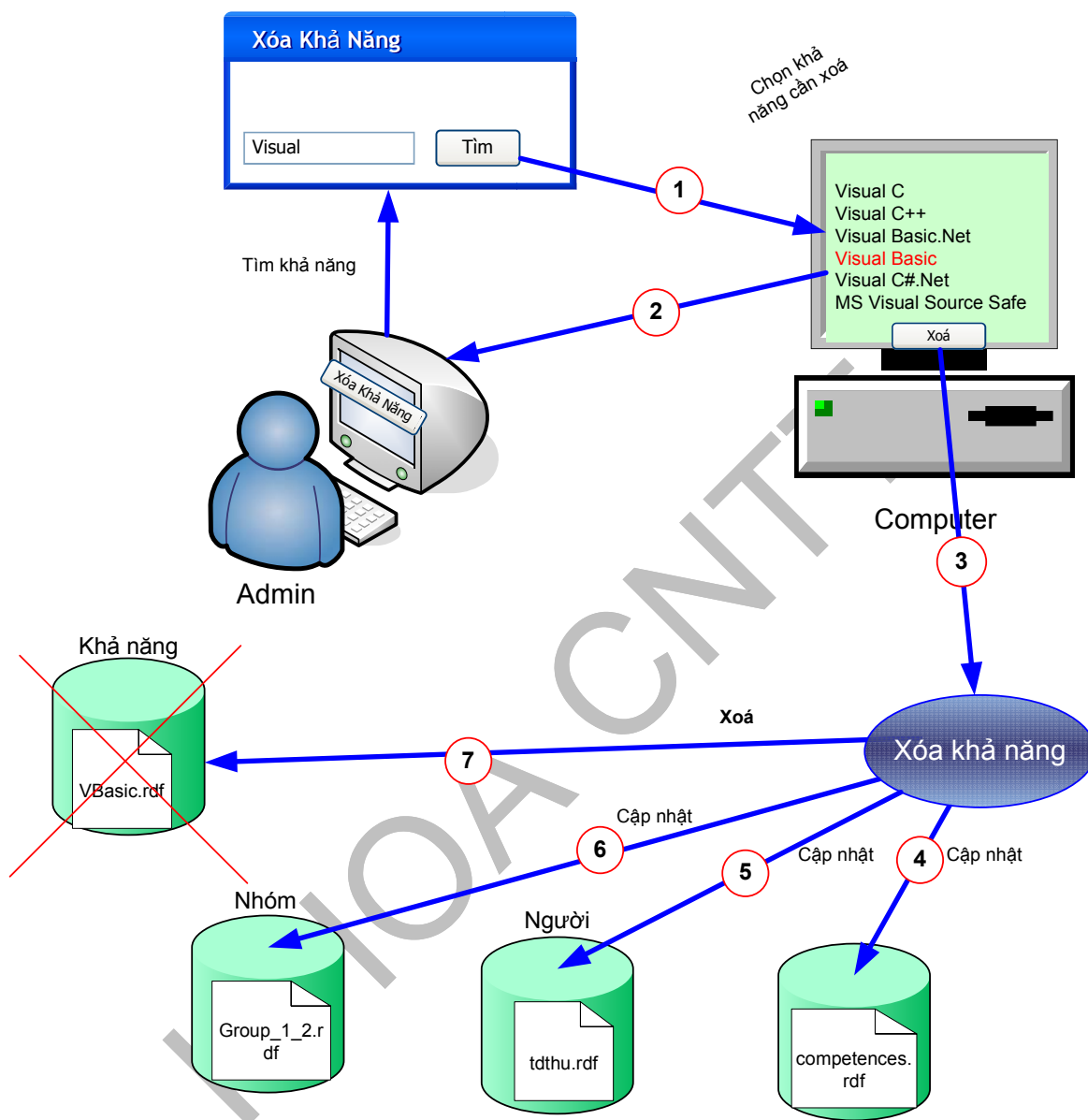
- Admin nhập thông tin của khả năng mới bao gồm : các thông tin khả năng, thông tin các cá nhân liên quan, thông tin các nhóm liên quan ( Hình 7-16 ) rồi nhấn nút “Thêm” để xác nhận việc tạo khả năng mới( 1 ).

- Hệ thống sẽ hiển thị nội dung file RDF ( Hình 2 ) mô tả thông tin về khả năng mới được tạo ( 2 ), tạo mới file RDF này và lưu file này vào database ( 3 ), thêm thông tin khả năng mới này vào file RDF mô tả danh sách những khả năng có trong hệ thống ( 4 ), và cập nhật thông tin của khả năng mới này cho những người có liên quan ( 5 ), những nhóm có liên quan ( 6 )

### 7.2.8.2 Xóa khả năng

Chức năng này cho phép admin xoá một khả năng ra khỏi hệ thống. Xem hình minh hoạ và giải thích bên dưới.

- Hình minh hoạ : *Hình 7-17* minh hoạ cho quá trình xoá khả năng :



Hình 7-17 Mô tả chức năng xóa khả năng.

▪ Giải thích :

- Admin nhập tên khả năng muốn xóa và nhấn nút “**Tìm**” để tìm xem khả năng muốn xóa có trong hệ thống hay không (1).

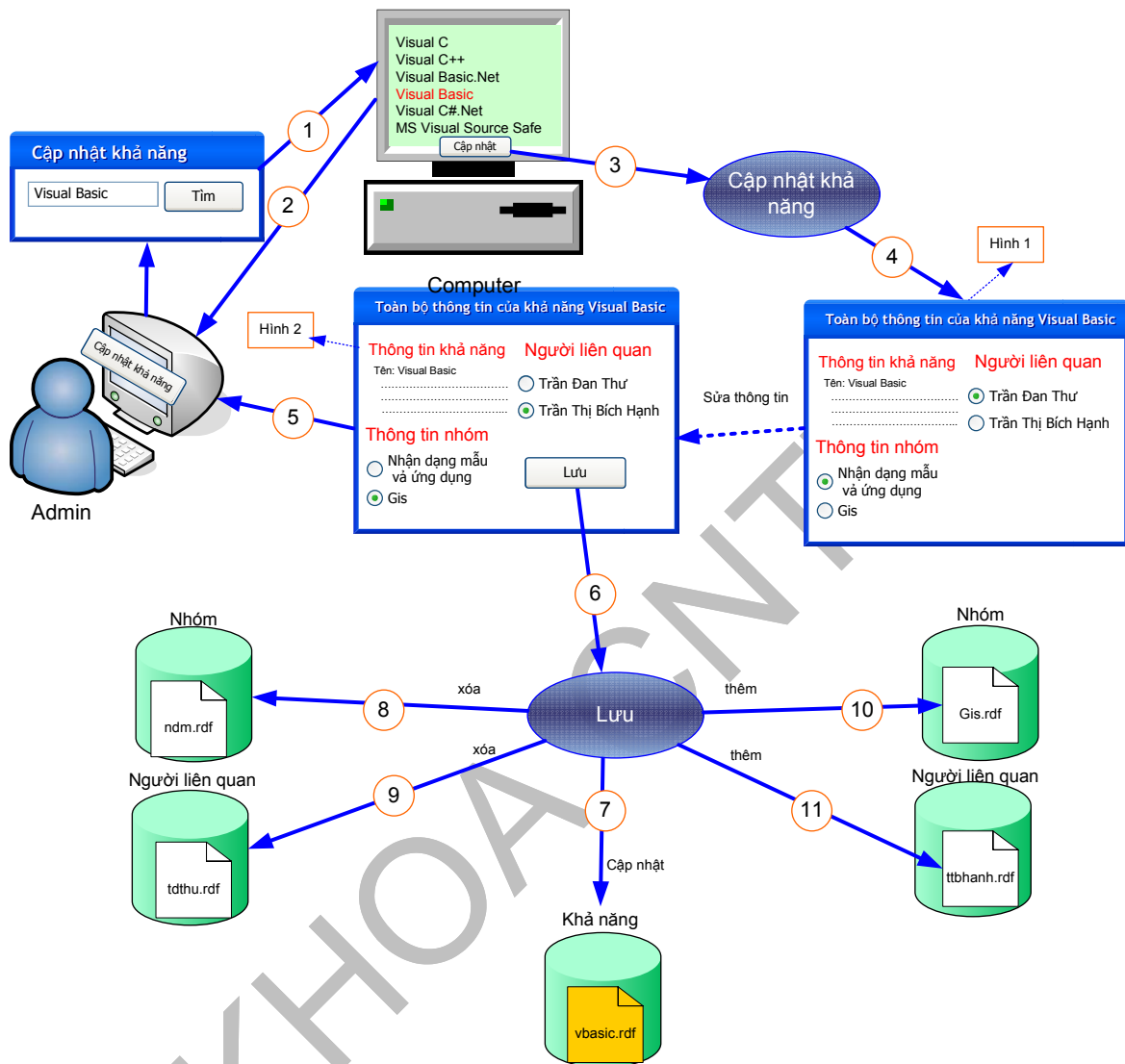
- Hệ thống sẽ vào cơ sở dữ liệu lấy và hiển thị tên khả năng muốn xoá (nếu không tồn tại khả năng này thì hệ thống hiển thị thông tin rỗng) trong danh sách tìm được (2). Admin chọn khả năng muốn xoá trong danh sách tìm được và nhấn nút “**Xoá**” để xác nhận việc xoá khả năng này(3).
- Hệ thống sẽ xoá thông tin khả năng này khỏi file RDF mô tả danh sách những khả năng có trong hệ thống (4), xoá thông tin của khả năng này khỏi file RDF của những người có liên quan (5), những nhóm có liên quan (6), sau cùng hệ thống sẽ xoá file RDF mô tả khả năng này khỏi hệ thống (7).

### 7.2.8.3 Sửa thông tin khả năng

Chức năng này cho phép admin sửa thông tin của một khả năng trong hệ thống. Xem hình minh hoạ và giải thích bên dưới.

- Hình minh họa :**Hình 7-18** minh họa cho quá trình sửa thông tin của một khả năng :





Hình 7-18 Mô tả chức năng sửa thông tin khả năng.

▪ Giải thích :

- Admin nhập tên khả năng muốn sửa thông tin và nhấn nút “**Tim**” để tìm xem khả năng muốn xoá có trong hệ thống hay không (1).
- Hệ thống sẽ vào cơ sở dữ liệu lấy và hiển thị (2) tên người muốn sửa (nếu không tồn tại khả năng này thì hệ thống hiển thị thông tin rỗng) trong danh

- sách tìm được, Admin chọn khả năng muốn sửa trong danh sách tìm được và nhấn nút “**Cập Nhật**” để cập nhật thông tin khả năng này (3).
- Hệ thống sẽ hiển thị (4, 5) tất cả các thông tin về khả năng này để admin chỉnh sửa (Hint 1) đồng thời cho phép admin thêm các thông tin mới về nhóm hay người có liên quan đến người này vào (Hint 2).
  - Admin nhấn nút “**lưu**” để lưu những thông tin đã chỉnh sửa xuống file (6).
  - Hệ thống sẽ hiển thị file RDF mô tả toàn bộ các thông tin mới về khả năng mà admin đã chỉnh sửa, ghi các thông tin mới này vào file RDF mô tả khả năng này trong database (7), cập nhật (xóa) thông tin về khả năng này trong các file RDF của những nhóm và những người có liên quan (8, 9) và cập nhật (thêm) thông tin về khả năng này trong các file RDF của những nhóm và những người có liên quan (10, 11).

## Chương 8 THIẾT KẾ VÀ CÀI ĐẶT ỨNG DỤNG

### 8.1 Thiết kế dữ liệu

#### 8.1.1 Xây dựng ontology PC( Person's Competence)

##### 8.1.1.1 Mục đích

Ontology PC được xây dựng để :

- Định nghĩa cho khái niệm “Competence”.
- Định nghĩa một bộ từ vựng mô tả thông tin cho các competences và mối quan hệ giữa các competences đó với các khái niệm “person”, “group”.

##### 8.1.1.2 Nội dung

Ontology này gồm có lớp khả năng (*pc:Competence*), lớp này có các thuộc tính sau:

- Thuộc tính “*pc:knows*” : cho biết 1 người hay 1 nhóm biết 1 khả năng, thể hiện cho mối quan hệ giữa 1 người với 1 khả năng hay giữa 1 nhóm với 1 khả năng
- Thuộc tính “*pc:isKnownBy*” : cho biết khả năng này được biết bởi 1 người hay 1 nhóm, thể hiện mối quan hệ giữa 1 khả năng với 1 nhóm hay giữa 1 khả năng với 1 người.
- Thuộc tính “*pc:isMemberOf*” : cho biết một người là thành viên của 1 nhóm, thể hiện mối quan hệ giữa 1 người và 1 nhóm.
- Thuộc tính “*pc:include*” : cho biết một nhóm gồm 1 người hay gồm một nhóm con, thể hiện mối quan hệ giữa 1 nhóm với 1 người, giữa 1 nhóm với 1 nhóm con.

Nội dung của file RDF mô tả cho ontology PC :

```

<?xml version="1.0" ?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:vs="http://www.w3.org/2003/06/sw-vocab-status/ns#"
xmlns:foaf="http://xmlns.com/foaf/0.1/"
xmlns:wot="http://xmlns.com/wot/0.1/"
xmlns:dc="http://pURL.org/dc/elements/1.1/"
xmlns:pc="http://localhost/onto/pc/">
<!--The content of the ontology file-->
<owl:Ontology rdf:about="http://localhost/onto/pc/"
  dc:title="Person's Competences Vocabulary"
  dc:description="Person's Competence, RDF vocabulary, described using W3C RDF
Schema and the Web Ontology Language."
  dc:date="$Date: 2005/06/03 07:03:28 $">
  <rdfs:seeAlso rdf:resource="http://localhost/onto/pc/" />
  <owl:imports rdf:resource="http://www.w3.org/2000/01/rdf-schema" />
  <owl:imports rdf:resource="http://www.w3.org/2002/07/owl" />
  <wot:assurance rdf:resource="../foafsig" />
  <wot:src_assurance rdf:resource="../htmlfoafsig" />
</owl:Ontology>
<!--PC classes -->
<rdfs:Class rdf:about="http://localhost/onto/pc/Competence"
  rdfs:label="Competence"

```

```

    rdfs:comment="A competence."
    vs:term_status="testing">
    <rdfs:subClassOf rdf:resource="http://xmlns.com/foaf/0.1/Agent" />
    <rdfs:isDefinedBy rdf:about="http://localhost/onto/pc/" />
</rdfs:Class>
<!--PC property -->
<!--Property knows -->
<rdf:Property rdf:about="http://localhost/onto/pc/knows"
    vs:term_status="testing"
    rdfs:label="knows"
    rdfs:comment="A person or a group know a competence.">
    <rdfs:domain rdf:resource="http://xmlns.com/foaf/0.1/Person" />
    <rdfs:domain rdf:resource="http://xmlns.com/foaf/0.1/Group" />
    <rdfs:range rdf:resource="http://localhost/onto/pc/Competence" />
    <rdfs:isDefinedBy rdf:about="http://localhost/onto/pc/" />
</rdf:Property>
<!--Property isKnownBy -->
<rdf:Property rdf:about="http://localhost/onto/pc/isKnownBy"
    vs:term_status="testing"
    rdfs:label="isKnownBy"
    rdfs:comment="A competence known by a person or a group.">
    <rdfs:domain rdf:resource="http://localhost/onto/pc/Competence" />
    <rdfs:range rdf:resource="http://xmlns.com/foaf/0.1/Person" />
    <rdfs:range rdf:resource="http://xmlns.com/foaf/0.1/Group" />
    <rdfs:isDefinedBy rdf:about="http://localhost/onto/pc/" />

```

```

</rdf:Property>
<!--Property isMemberOf -->
<rdf:Property rdf:about="http://localhost/onto/pc/isMemberOf"
  vs:term_status="testing"
  rdfs:label="isMemberOf"
  rdfs:comment="A group of this person.">
  <rdfs:domain rdf:resource="http://xmlns.com/foaf/0.1/Person" />
  <rdfs:domain rdf:resource="http://xmlns.com/foaf/0.1/Group" />
  <rdfs:range rdf:resource="http://xmlns.com/foaf/0.1/Group" />
  <rdfs:isDefinedBy rdf:about="http://localhost/onto/pc/" />
</rdf:Property>
<!--Property include -->
<rdf:Property rdf:about="http://localhost/onto/pc/include"
  vs:term_status="testing"
  rdfs:label="include"
  rdfs:comment="A group include a person or a sub_group.">
  <rdfs:domain rdf:resource="http://xmlns.com/foaf/0.1/Group" />
  <rdfs:range rdf:resource="http://xmlns.com/foaf/0.1/Person" />
  <rdfs:range rdf:resource="http://xmlns.com/foaf/0.1/Group" />
  <rdfs:isDefinedBy rdf:about="http://localhost/onto/pc/" />
</rdf:Property>
</rdf:RDF>

```

### 8.1.1.3 Sử dụng

Ontology này được khai báo thông qua một URI(cục bộ) <http://localhost/ont/pc/> và có QName là “*pc:*”.

### 8.1.2 Thiết kế dữ liệu mô tả tài nguyên người

Mỗi file RDF mô tả cho một người gồm các phần sau :

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:dc="http://pURL.org/dc/elements/1.1/"
  xmlns:pc="http://localhost/onto/pc/"
  xmlns:admin="http://webns.net/mvcb/">

<!--Thông tin cá nhân-->
<foaf:Person rdf:nodeID="me">
<foaf:name>Tên đầy đủ của một người</foaf:name>
<foaf:title>Mss/Mr/Mrs</foaf:title>
<foaf:givenname>Tên lót</foaf:givenname>
<foaf:family_name>Họ</foaf:family_name>
<foaf:nick>...</foaf:nick>
<foaf:mbox rdf:resource="mailto:Địa chỉ mail"/>
<foaf:mbox_sha1sum>Do hệ thống phát sinh</foaf:mbox_sha1sum>
<foaf:depiction rdf:resource="..."/>

<!--Thông tin nơi làm việc-->
<foaf:homepage rdf:resource="..."/>
<foaf:workplaceHomepage rdf:resource="..."/>
<foaf:workInfoHomepage rdf:resource="..."/>
<foaf:schoolHomepage rdf:resource="..."/>

<!--Thông tin các khả năng-->
<pc:knows>
<pc:Competence>
<foaf:name>Tên khả năng</foaf:name>
<rdfs:seeAlso rdf:resource="..."/>
```

```

</pc:Competence>
</pc:knows>
...
<!--Thông tin nhóm-->
<pc:isMemberOf>
<foaf:Group>
<foaf:name>Tên nhóm</foaf:name>
<rdfs:seeAlso rdf:resource="..."/>
</foaf:Group>
</pc:isMemberOf>
...
<!--Thông tin những người liên quan-->
<foaf:knows>
<foaf:Person>
<foaf:name>Tên người</foaf:name>
<rdfs:seeAlso rdf:resource="..."/>
</foaf:Person>
</foaf:knows>
...
<!--Thông tin những trang web liên quan đến người này-->
<foaf:page rdf:resource="http://localhos/page/FOAF-Mo ta thong tin ca nhan.htm"/>
...
</foaf:Person>
</rdf:RDF>
    
```

**Ví dụ :**

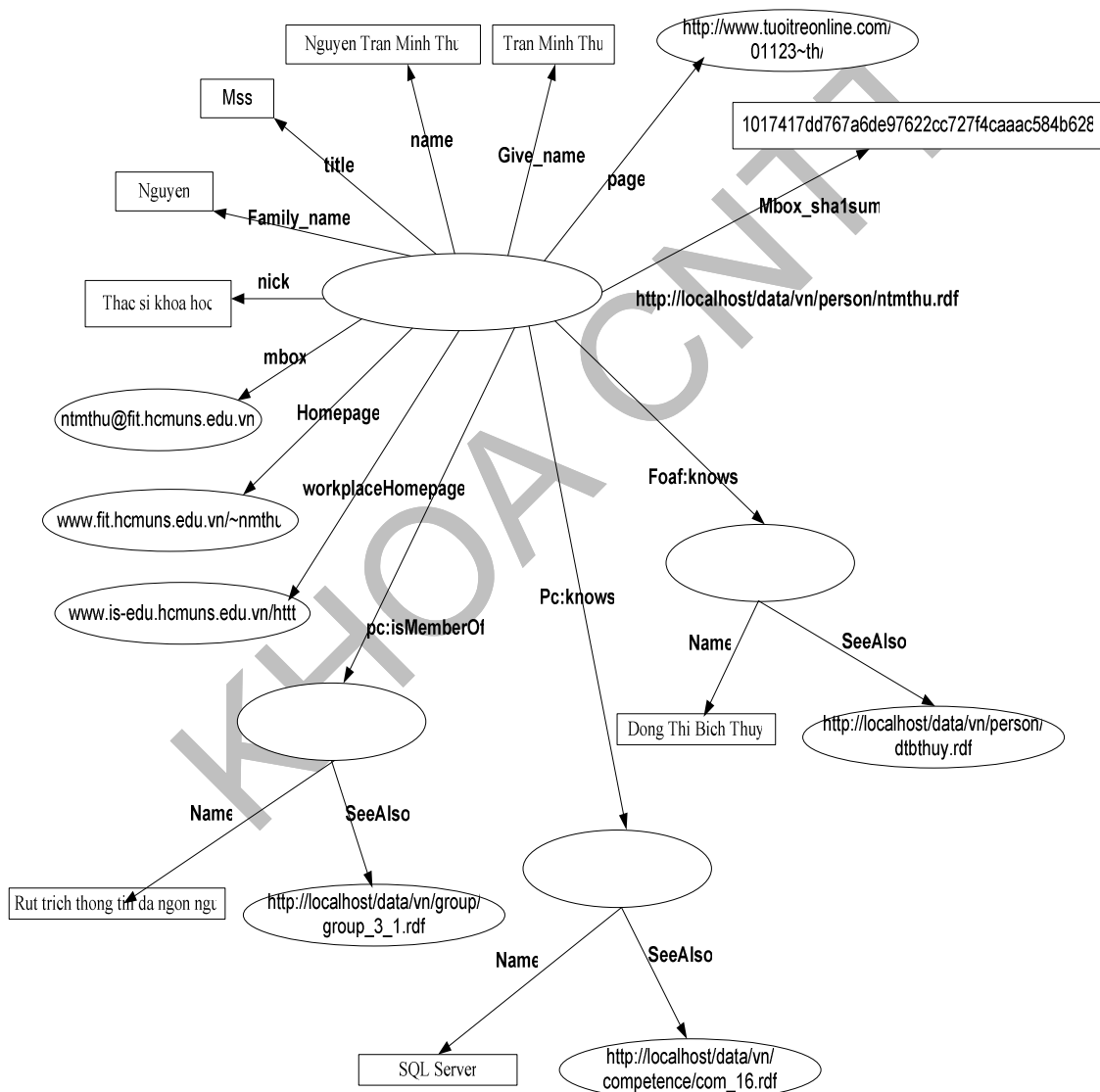
Xét phát biểu sau :

Xét một tài nguyên có định danh URI là “*http://localhost/data/vn/person/ntmthu.rdf*”, tài nguyên này mô tả cho một người trong hệ thống. Người này có tên (**foaf:name**) đầy đủ là *Nguyễn Trần Minh Thu*, họ (**family\_name**) là *Nguyễn*, tên (**given\_name**) là *Trần Minh Thu*, có học vị (**foaf:nick**) là “*thạc sĩ khoa học máy tính*”, địa chỉ mail (**foaf:mbox**) là “*ntmthu@fit.hcmuns.edu.vn*”, có số **mbox\_sha1sum** là “*1017417dd767a6de97622cc727f4caaac584b628*”, có **foaf:homepage** là



“[www.fit.hcmuns.edu.vn/~nmthu](http://www.fit.hcmuns.edu.vn/~nmthu)”, có địa chỉ nơi làm việc(*foaf:workplacehomepage*) là “[www.fit.hcmuns.edu.vn/~ntmthu](http://www.fit.hcmuns.edu.vn/~ntmthu)”, có khả năng(*pc:knows*) về *SQL server*, là thành viên(*pc:isMemberOf*) của nhóm nghiên cứu “*rút trích thông tin đa ngôn ngữ*”, quen (*foaf:knows*) với 1 người có tên(*foaf:name*) là *Đông Thị Bích Thủy*, và có 1 website liên quan(*pc:page*) là “<http://www.tuoitreonline.com/01123~th/>”.

• **Biểu diễn bằng đồ thị :**



**Hình 8-1** Đồ thị biểu diễn thông tin của một cá nhân.

- **Biểu diễn bằng RDF :**

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:dc="http://pURL.org/dc/elements/1.1/"
  xmlns:pc="http://localhost/onto/pc/"
  xmlns:admin="http://webns.net/mvcb/">

  <foaf:Person rdf:nodeID="me">
  <foaf:name>Nguyễn Trần Minh Thư</foaf:name>
  <foaf:title>Mss</foaf:title>
  <foaf:givenname>Nguyễn</foaf:givenname>
  <foaf:family_name>Trần Minh Thư</foaf:family_name>
  <foaf:nick>Thạc sĩ khoa học máy tính</foaf:nick>
  <foaf:mbox rdf:resource="mailto:ntmthu@fit.hcmuns.edu.vn"/>
  <foaf:mbox_sha1sum>1017417dd767a6de97622cc727f4caaac584b628</foaf:mbox_sha1sum>
  <foaf:homepage rdf:resource="www.fit.hcmuns.edu.vn/~nmthu"/>
  <foaf:workplaceHomepage rdf:resource="www.is-edu.hcmuns.edu.vn/http"/>

  <!--Thông tin khả năng-->
  <pc:knows>
  <pc:Competence>
  <foaf:name>SQL Server</foaf:name>
  <rdfs:seeAlso rdf:resource="http://localhost/data/vn/competence/com_16.rdf"/>
  </pc:Competence>
  </pc:knows>

  <!--Thông tin nhóm nghiên cứu -->

  <pc:isMemberOf>
  <foaf:Group>
  <foaf:name>Rút trích thông tin đa ngôn ngữ</foaf:name>
  <rdfs:seeAlso rdf:resource="http://localhost/data/vn/group/group_3_1.rdf"/>
  </foaf:Group>
  
```

```

</pc:isMemberOf>

<!--Thông tin người quen -->
<foaf:knows>
<foaf:Person>
<foaf:name>Đồng Thị Bích Thủy</foaf:name>
<foaf:mbox_sha1sum>74a82e906680b479d030d7a5ebd0196968ddb989</foaf:mbox_sha1s
um>
<rdfs:seeAlso rdf:resource="http://localhost/data/vn/person/dtbthuy.rdf"/>
</foaf:Person>
</foaf:knows>
<!--Thông tin những trang web liên quan đến người này-->
<foaf:page rdf:resource=" http://www.tuoitreonline.com/01123~th"/>

</foaf:Person>
</rdf:RDF>
    
```

### 8.1.3 Thiết kế dữ liệu mô tả tài nguyên nhóm người

Mỗi file RDF mô tả cho một nhóm gồm các phần sau :

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:dc="http://pURL.org/dc/elements/1.1/"
  xmlns:pc="http://localhost/onto/pc/"
  xmlns:admin="http://webns.net/mvcb/">

  <foaf:Group>
  <foaf:name>Tên nhóm</foaf:name>
  <dc:description>Mô tả nhóm</dc:description>

  <!-- Thông tin các khả năng của nhóm -->
  <pc:Competence>
  <foaf:name>Tên khả năng</foaf:name>
  <rdfs:seeAlso rdf:resource="..."/>
  </pc:Competence>
    
```

```

</pc:knows>
...
<!--Thông tin các thành viên tham gia nghiên cứu-->
<foaf:member>
<foaf:Person>
<foaf:name>Tên người</foaf:name>
<rdfs:seeAlso rdf:resource="..."/>
</foaf:Person>
</foaf:member>
...
<!--Thông tin các nhóm con-->
<pc:include>
<foaf:Group>
<foaf:name>Tên nhóm con</foaf:name>
<rdfs:seeAlso rdf:resource="..."/>
</foaf:Group>
</pc:include>
...
</foaf:Group>
</rdf:RDF>
    
```

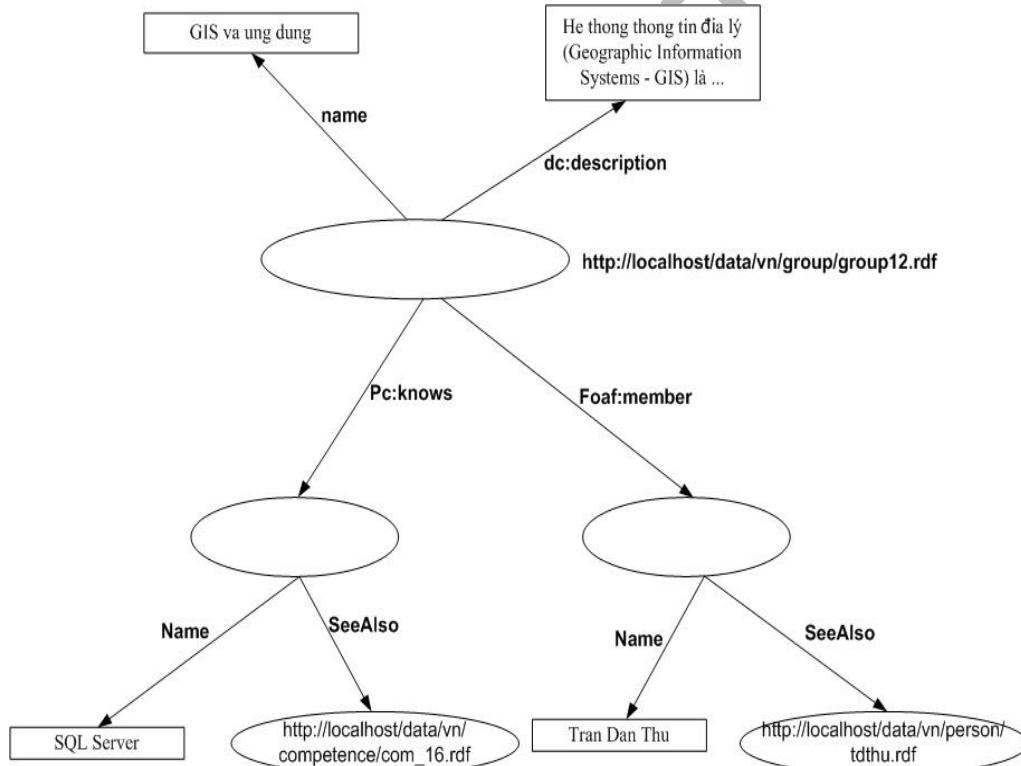
### Ví dụ :

Xét phát biểu sau :

Một tài nguyên có định danh URI là “*http://localhost/data/vn/group/group2.rdf*” mô tả cho một nhóm(**group**) trong hệ thống. Nhóm này có tên(**foaf:name**) là “*GIS và ứng dụng*”, có mô tả(**dc:Description**) là “*Hệ thống thông tin địa lý (Geographic Information Systems - GIS) là một trong những lĩnh vực nghiên cứu chủ đạo tại Khoa. Mục tiêu của chúng tôi là nghiên cứu những công nghệ GIS tiên tiến nhất, xây dựng phần mềm/công cụ GIS, và ứng dụng những công nghệ này trong giao thông, hoạch định môi trường, quản lý, nghiên cứu và giảng dạy. Trong những năm gần đây, với một phòng thí nghiệm mới cùng chương trình nghiên cứu quy mô trong lĩnh vực này, số lượng sinh viên tham gia vào các dự án GIS cũng như chọn GIS làm đề tài tốt nghiệp tăng lên nhanh chóng. Hiện nay,*

chúng tôi đang phát triển nhiều dự án về GIS như: Hệ thống hướng dẫn giao thông (Traffic Guiding System - TGS), Xây dựng bản đồ trên máy Pocket PC 2002... Một số dự án là đề tài nghiên cứu cấp Quốc gia và được hỗ trợ bởi Nhà nước. Cần nói thêm rằng, phần mềm TGS đã được nhận một số giải thưởng như: giải thưởng Sáng tạo Khoa học Kỹ thuật VIFOTEC, giải thưởng của Bộ Giáo dục và Đào tạo ... Phần mềm này đã và đang được bán trên thị trường và nhận được rất nhiều ý kiến tích cực từ phía người tiêu dùng.”, có khả năng(*pc:knows*) là SQL Server và có thành viên tham gia có tên(*foaf:name*) là Trần Đan Thu....

- **Biểu diễn bằng đồ thị :**



**Hình 8-2 Đồ thị biểu diễn thông tin của một nhóm nghiên cứu.**

- **Biểu diễn bằng RDF :**

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  ...
```

```
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:foaf="http://xmlns.com/foaf/0.1/"
xmlns:dc="http://pURL.org/dc/elements/1.1/"
xmlns:pc="http://localhost/onto/pc/"
xmlns:admin="http://webns.net/mvcb/">
<!--
      group 12: GIS và ứng dụng
-->
<foaf:Group>
<foaf:name>GIS và ứng dụng</foaf:name>
<dc:description>Hệ thống thông tin địa lý (Geographic Information Systems - GIS) là một
trong những lĩnh vực nghiên cứu chủ đạo tại Khoa. Mục tiêu của chúng tôi là nghiên cứu
những công nghệ GIS tiên tiến nhất, xây dựng phần mềm/công cụ GIS, và ứng dụng
những công nghệ này trong giao thông, hoạch định môi trường, quản lý, nghiên cứu và
giảng dạy.
Trong những năm gần đây, với một phòng thí nghiệm mới cùng chương trình nghiên cứu
quy mô trong lĩnh vực này, số lượng sinh viên tham gia vào các dự án GIS cũng như chọn
GIS làm đề tài tốt nghiệp tăng lên nhanh chóng.
Hiện nay, chúng tôi đang phát triển nhiều dự án về GIS như: Hệ thống hướng dẫn giao
thông (Traffic Guiding System - TGS), Xây dựng bản đồ trên máy Pocket PC 2002... Một
số dự án là đề tài nghiên cứu cấp Quốc gia và được hỗ trợ bởi Nhà nước. Cần nói thêm
rằng, phần mềm TGS đã được nhận một số giải thưởng như: giải thưởng Sáng tạo Khoa
học Kỹ thuật VIFOTEC, giải thưởng của Bộ Giáo dục và Đào tạo ... Phần mềm này đã và
đang được bán trên thị trường và nhận được rất nhiều ý kiến tích cực từ phía người tiêu
dùng.</dc:description>

<!--Thông tin khả năng-->

<pc:knows>
<pc:Competence>
<foaf:name>SQL Server</foaf:name>
<rdfs:seeAlso rdf:resource="http://localhost/data/vn/competence/com_1.rdf"/>
</pc:Competence>
</pc:knows>

<!--Thông tin thành viên-->

<foaf:member>
```

```
<foaf:Person>
<foaf:name>Trần Đan Thư</foaf:name>
<rdfs:seeAlso rdf:resource="http://localhost/data/vn/person/tdthu.rdf"/>
</foaf:Person>
</foaf:member>

</foaf:Group></rdf:RDF>
```

#### 8.1.4 Thiết kế dữ liệu mô tả tài nguyên khả năng

Mỗi file RDF mô tả cho một khả năng gồm các phần sau :

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:dc="http://pURL.org/dc/elements/1.1/"
  xmlns:pc="http://localhost/onto/pc/"
  xmlns:admin="http://webns.net/mvcb/">

<pc:Competence>
<foaf:name>Tên khả năng</foaf:name>

<!--Thông tin các nhóm liên quan-->
<pc:isKnownBy>
<foaf:Group>
<foaf:name>Tên nhóm</foaf:name>
<rdfs:seeAlso rdf:resource="..."/>
</foaf:Group>
</pc:isKnownBy>
...

<!--Thông tin những người liên quan-->
<pc:isKnownBy>
<foaf:Person>
<foaf:name>Tên người</foaf:name>
<rdfs:seeAlso rdf:resource="..."/>
</foaf:Person>
```

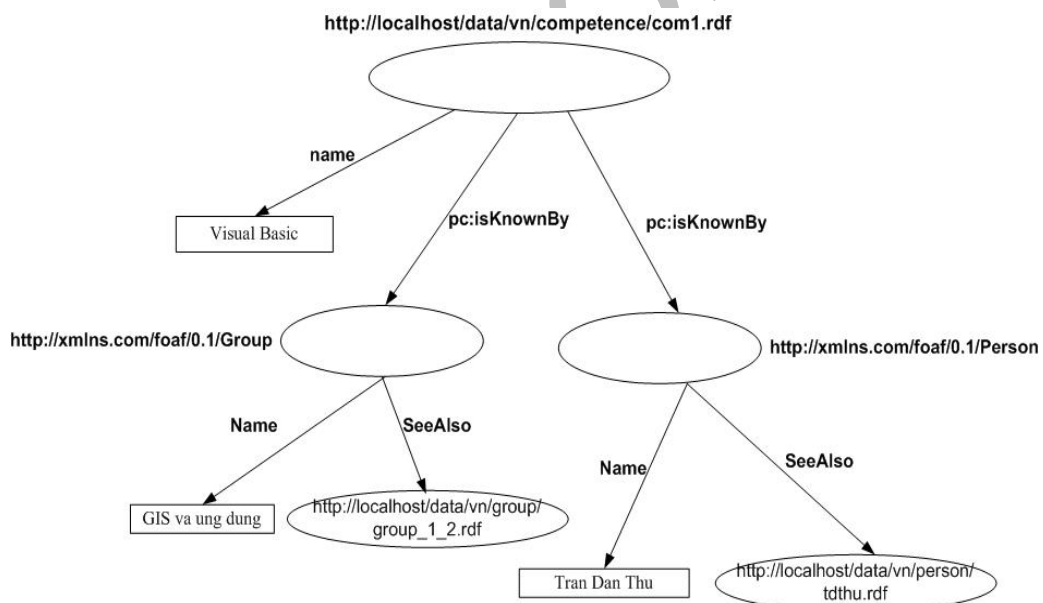
```
</pc:isKnownBy>
...
</pc:Competence>
</rdf:RDF>
```

**Ví dụ :**

Xét phát biểu sau :

Một tài nguyên có định danh URI “*http://localhost/data/vn/competence/com1.rdf*” là một tài nguyên mô tả cho một khả năng (*competence*) trong hệ thống. Khả năng này có tên (*foaf:name*) là *Visual Basic*, được biết (*pc:isKnownBy*) bởi nhóm có tên là (*foaf:name*) là *GIS và ứng dụng*, được biết bởi (*pc:isKnownBy*) cá nhân tên có tên (*foaf:name*) là *Trần Đan Thu*...

- **Biểu diễn bằng đồ thị :**



**Hình 8-3 Đồ thị biểu diễn thông tin của một khả năng.**

- **Biểu diễn bằng RDF :**

```
<rdf:RDF
```



```
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:foaf="http://xmlns.com/foaf/0.1/"
xmlns:dc="http://pURL.org/dc/elements/1.1/"
xmlns:pc="http://localhost/onto/pc/"
xmlns:admin="http://webns.net/mvcb/"
```

```
<pc:Competence>
```

```
<foaf:name>Visual Basic</foaf:name>
```

```
<!--Thông tin nhóm-->
```

```
<pc:isKnownBy>
```

```
<foaf:Group>
```

```
<foaf:name>GIS và ứng dụng</foaf:name>
```

```
<rdfs:seeAlso rdf:resource="http://localhost/data/vn/group/group_1_2.rdf"/>
```

```
</foaf:Group>
```

```
</pc:isKnownBy>
```

```
<!--Thông tin cá nhân-->
```

```
<pc:isKnownBy>
```

```
<foaf:Person>
```

```
<foaf:name>Trần Đan Thư</foaf:name>
```

```
<rdfs:seeAlso rdf:resource="http://localhost/data/vn/person/tdthu.rdf"/>
```

```
</foaf:Person>
```

```
</pc:isKnownBy>
```

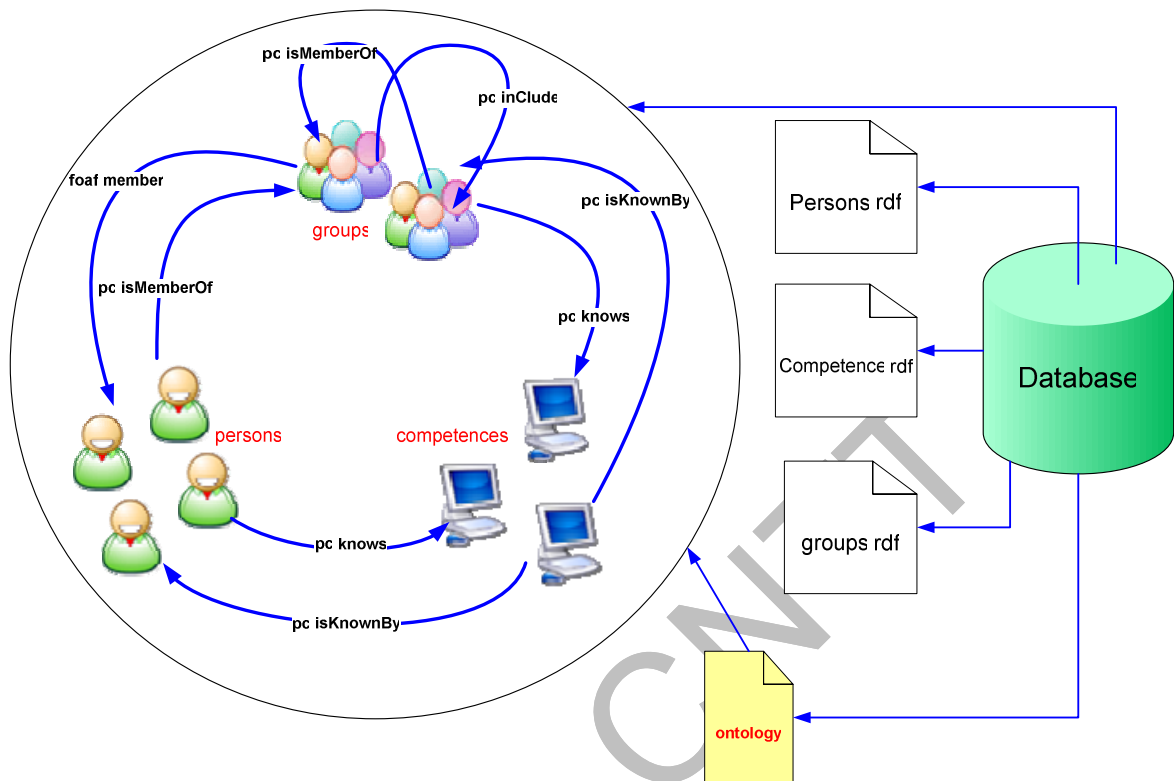
```
</pc:Competence>
```

```
</rdf:RDF>
```

### **8.1.5 Thiết kế file RDF (persons.rdf, groups.rdf, competences.rdf) lưu trữ toàn bộ các tài nguyên người, nhóm người và khả năng trong hệ thống.**

#### **8.1.5.1 Mục đích:**

Với việc thiết kế các file RDF mô tả thông tin cá nhân, nhóm và khả năng như trên hệ thống có thể truy vấn dữ liệu để tìm ra thông tin cần tìm. Tuy nhiên trong quá trình thiết kế hệ thống chúng em nhận thấy rằng với cách lưu trữ như trên thì khi tìm kiếm bất kỳ tài nguyên nào trong hệ thống, hệ thống phải tiến hành tìm kiếm( search) trên tất cả các file có trong hệ thống để tìm ra tài nguyên cần tìm kiếm. Cho nên chúng em đưa ra một ý tưởng là thiết kế các file *persons.rdf*, *groups.rdf* và *competences.rdf* để lưu trữ URIs của tất cả các tài nguyên trong hệ thống nhằm làm cho việc tìm kiếm được nhanh hơn, tiết kiệm được thời gian cũng như chi phí tìm kiếm. Lược đồ mô tả cho giải pháp này như hình sau:



Hình 8-4 Mô hình dữ liệu cấp nâng cấp.

### 8.1.5.2 Thiết kế:

#### 8.1.5.2.1 Thiết kế file "persons.rdf":

**Mục đích:**

File *persons.rdf* này lưu tất cả các persons hiện có trong hệ thống giúp cho việc tìm kiếm, thống kê và hiển thị danh sách các tài nguyên persons được nhanh chóng và thuận lợi.

**Nội dung:**

File *persons.rdf* lưu tên (*foaf:name*), số mbox\_sha1sum (*foaf:mbox\_sha1sum*), email (*foaf:mbox*) và địa chỉ URI (*rdfs:seeAlso*) của các tài nguyên persons.

Nội dung như sau :

```
<?xml version="1.0"?>
```

```

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:foaf="http://xmlns.com/foaf/0.1/" xmlns:dc="http://pURL.org/dc/elements/1.1/"
xmlns:pc="http://localhost/onto/pc/">

<foaf:Person>
<foaf:name>Đồng Thị Bích Thủy</foaf:name>
<foaf:mbox rdf:resource="mailto:dtbthuy@fit.hcmuns.edu.vn"/>
<foaf:mbox_sha1sum>74a82e906680b479d030d7a5ebd0196968ddb989</foaf:mbox_
sha1sum>
<rdfs:seeAlso rdf:resource="http://localhost/data/vn/person/dtbthuy.rdf" />
</foaf:Person>
.
.
.
.
.

<foaf:name>Lê Bảo Thiện</foaf:name>
<foaf:mbox_sha1sum>0a70d07b930fa78ce757a62a1cc998edd263d7a2</foaf:mbox_
sha1sum>
<foaf:mbox rdf:resource="mailto:lbthien@fit.hcmuns.edu.vn" />
<rdfs:seeAlso rdf:resource="http://localhost/data/vn/person/lbthien.rdf" />
</foaf:Person>
</rdf:RDF>
    
```

#### 8.1.5.2.2 Thiết kế file “groups.rdf”:

##### **Mục đích:**

File *groups.rdf* này lưu tất cả các groups hiện có trong hệ thống giúp cho việc tìm kiếm, thống kê và hiển thị danh sách các tài nguyên groups được nhanh chóng và thuận lợi.

##### **Nội dung:**

File *groups.rdf* lưu tên (*foaf:name*) và địa chỉ URI (*rdfs:seeAlso*) của các tài nguyên groups.

Nội dung như sau :

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:foaf="http://xmlns.com/foaf/0.1/">
  <!--các nhóm nghiên cứu hiện đang có trong hệ thống-->
  <foaf:Group>
    <foaf:name>Quy trình phát triển phần mềm theo phương pháp dựa trên
Component</foaf:name>
    <rdfs:seeAlso rdf:resource="http://localhost/data/vn/group/group_1_1.rdf" />
  </foaf:Group>
  .
  .
  .
  .
</foaf:Group>
<foaf:name>GIS và ứng dụng</foaf:name>
<rdfs:seeAlso rdf:resource="http://localhost/data/vn/group/group_1_2.rdf"/>
</foaf:Group>
</rdf:RDF>
```

#### 8.1.5.2.3 Thiết kế file “competences.rdf”:

##### **Mục đích:**

File *competences.rdf* này lưu tất cả các competences hiện có trong hệ thống giúp cho việc tìm kiếm, thống kê và hiển thị danh sách các tài nguyên competences được nhanh chóng và thuận lợi.

##### **Nội dung:**

File *competences.rdf* lưu tên (*pc:name*) và địa chỉ URI (*rdfs:seeAlso*) của các tài nguyên persons.

Nội dung như sau :

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:foaf="http://xmlns.com/foaf/0.1/" xmlns:dc="http://pURL.org/dc/elements/1.1/"
```

```

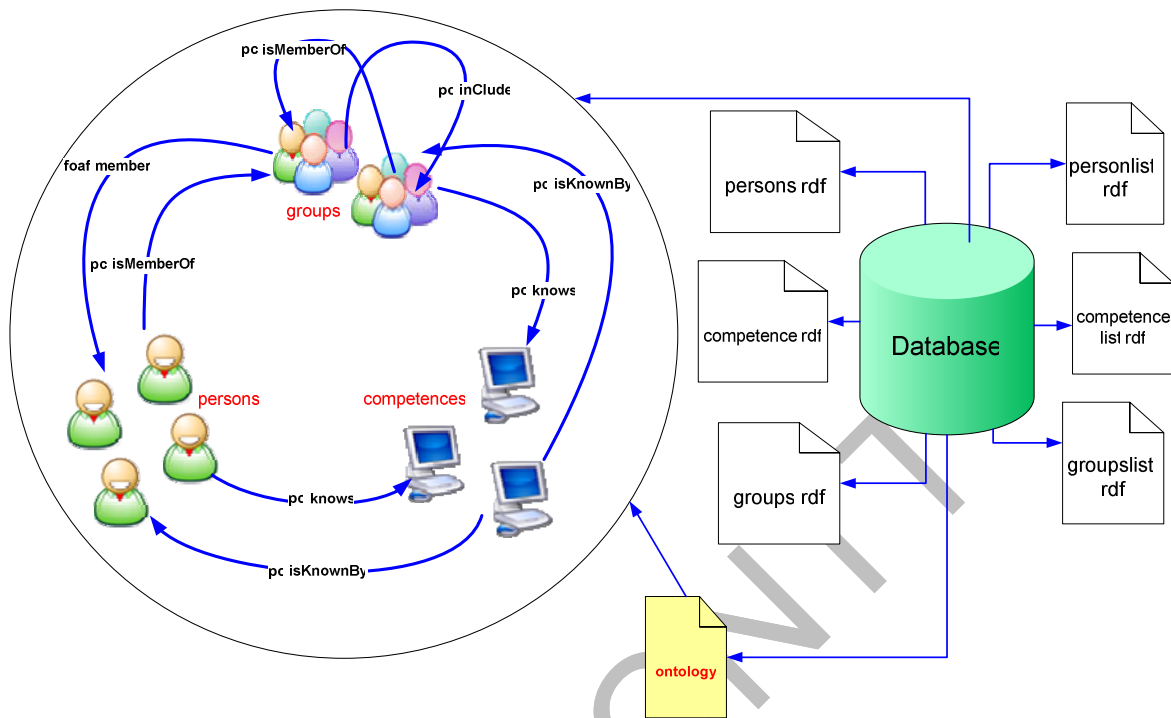
xmlns:pc="http://localhost/onto/pc/">
  <pc:Competence>
    <foaf:name>Visual C</foaf:name>
    <rdfs:seeAlso rdf:resource="http://localhost/data/vn/competence/com_2.rdf" />
  </pc:Competence>
  .
  .
  .
  .
  <pc:Competence>
    <foaf:name>Visual Basic</foaf:name>
    <rdfs:seeAlso rdf:resource="http://localhost/data/vn/competence/com_1.rdf"/>
  </pc:Competence>
</rdf:RDF>

```

## 8.1.6 Thiết kế file personlist.rdf, grouplist.rdf, competencelist.rdf

### 8.1.6.1 Mục đích

Các file này được thiết kế nhằm mục đích lưu các địa chỉ URIs của các tài nguyên về person, group hay competence. Các URIs này không chỉ của riêng những tài nguyên trong hệ thống mà của bất kỳ tài nguyên nào ở bất kỳ đâu trên internet mô tả cho person, group hay competence trong trường hợp muốn mở rộng cơ sở dữ liệu cho hệ thống. Mô hình dữ liệu như hình :



**Hình 8-5** Mô hình dữ liệu cuối cùng.

### 8.1.6.2 Thiết kế:

#### 8.1.6.2.1 Thiết kế file “personlist.rdf” :

Lưu địa chỉ URI của tất cả các tài nguyên persons có trong hệ thống( chính là URI của file *persons.rdf* trong hệ thống này).

```
<!--mô tả các file cơ trong csdl-->
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:admin="http://webns.net/mvcb/">
  <has>http://localhost/data/vn/persons.rdf</has>
</rdf:RDF>
```

#### 8.1.6.2.2 Thiết kế file “ grouplist.rdf” :

Lưu địa chỉ URI của tất cả các tài nguyên groups có trong hệ thống(chính là URI của file *groups.rdf* trong hệ thống).

```
<!--mo ta cac file co trong csdl-->
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:admin="http://webns.net/mvcb/">
  <has>http://localhost/data/vn/groups.rdf</has>
</rdf:RDF>
```

#### 8.1.6.2.3 Thiết kế file “ competencelist.rdf” :

Lưu địa chỉ URI của tất cả các tài nguyên competences có trong hệ thống(chính là URI của file *competences.rdf* trong hệ thống này).

```
<!--mo ta cac file co trong csdl-->
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/" >
  <has>http://localhost/data/vn/competences.rdf</has>
</rdf:RDF>
```

### 8.1.7 Thiết kế file XSL để chuyển file RDF sang file SVG

Ứng dụng phải có giao diện đồ họa thể hiện lược đồ biểu diễn cho các tài nguyên con người, nhóm và khả năng. Và để trình duyệt web hiển thị được lược đồ thì ta phải dùng ngôn ngữ SVG. Công việc mà hệ thống phải làm là chuyển các định dạng RDF sang định dạng SVG. Vì thế chúng em sử dụng ngôn ngữ XSLT xây dựng 3 file XSL sau :

*Person.xsl* : dùng để chuyển các file RDF mô tả người sang định dạng SVG



*Group.xsl* : dùng để chuyển các file RDF mô tả nhóm sang định dạng SVG

*Competence.xsl* : dùng để chuyển các file RDF mô tả khả năng sang định dạng SVG

Nội dung của file Person.xsl :

```
<!--?xml version="1.0" encoding="UTF-8" ?-->
<xsl:stylesheet version="1.0"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:dt="urn:schemas-microsoft-com:datatypes"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:rdfs="http://www.w3.org/1700/01/rdf-schema#"
  xmlns:dc="http://pURL.org/dc/elements/1.1/"
  xmlns:pc="http://localhost/onto/pc/">

<xsl:namespace-alias stylesheet-prefix="dt" result-prefix="xsl"/>
<xsl:namespace-alias stylesheet-prefix="pc" result-prefix="xsl"/>
<xsl:namespace-alias stylesheet-prefix="rdf" result-prefix="xsl"/>
<xsl:namespace-alias stylesheet-prefix="foaf" result-prefix="xsl"/>
<xsl:output method="xml" version="1.0" omit-xml-declaration = "yes"/>

<xsl:template match="/">
  <svg width="750" height="600">
    <xsl:apply-templates/>
  </svg>
</xsl:template>
<xsl:template match="rdf:RDF">
  <xsl:apply-templates/>
</xsl:template>
<xsl:template match="foaf:Person">
  <circle cx="375" cy="350" r="55" style="stroke: green; fill: #cfc; stroke-width:1.5; "/>
  <!--lay hình -->
  <image
    x="337"
    y="320"
    width="75"
    height="65" xlink:href="http://localhost/webapp/EPerson/images/person.png"/>

  <!-- cac nhom nghien cuu-->
```

```

<xsl:for-each select="pc:isMemberOf/foaf:Group">
  <change>
    <xsl:attribute name="pos">
      <xsl:value-of select="position()+count(//*[name()='pc:knows'])"/>
    </xsl:attribute>
  </change>
  <rect x="60" y="60" width="100" height="25" rx="4" ry="4" style="stroke: white; fill:
none;">
  </rect>
  <!--lay ten nhom-->
  <change><!--thay doi tag na`y-->
    <xsl:attribute name="pos">
      <xsl:value-of select="position()+count(//*[name()='pc:knows'])"/>
    </xsl:attribute>
  </change>
  <text x="20" y="20" textLength="100" lengthAdjust="spacingAndGlyphs" style="text-
anchor: middle; stroke:white;" >
  </text>
  <!--in hình chu nhật-->
  <change>
    <xsl:attribute name="pos">
      <xsl:value-of select="position()+count(//*[name()='pc:knows'])"/>
    </xsl:attribute>
  </change>
  <image x="0" y="0" width="75" height="60"
xlink:href="http://localhost/webapp/EPerson/images/group.png">
  </image>
  <!--ve mui ten -->
  <image x="430" y="340" width="170" height="20"
xlink:href="http://localhost/webapp/EPerson/images/muiten.jpg">
    <xsl:attribute name="transform">
      <xsl:value-of
        select="concat('rotate(',
string(((position()+count(//*[name()='pc:knows']))*360)
div
(count(//*[name()='pc:knows'])+count(//*[name()='pc:isMemberOf'])+count(//*[name()='foaf:kn
ows'])), ',375,350)')"/>
    </xsl:attribute>
  </image>
</xsl:for-each>

```

```

<!-- cac kha nang-->
<xsl:for-each select="pc:knows/pc:Competence">

<!-- lay hinh kha nang-->
<change>
  <xsl:attribute name="pos">
    <xsl:value-of select="position()"/>
  </xsl:attribute>
</change>
<image      x="0"      y="0"      width="55"      height="45"
xlink:href="http://localhost/webapp/EPerson/images/comp.png">
</image>
<!--lay ten kha nang -->
<change><!--thay doi tag na`y-->
  <xsl:attribute name="pos">
    <xsl:value-of select="position()"/>
  </xsl:attribute>
</change>
<text x="20" y="20" textLength="100" lengthAdjust="spacingAndGlyphs" style="text-
anchor: middle;" >
  <xsl:value-of select="foaf:name"/>
</text>
<!--ve mui ten -->
<image      x="430"      y="340"      width="170"      height="20"
xlink:href="http://localhost/webapp/EPerson/images/muiten.jpg">
  <xsl:attribute name="transform">
    <xsl:value-of select="concat('rotate(', string((position()*360) div
(count(//*[name()='pc:knows'])+count(//*[name()='pc:isMemberOf'])+count(//*[name()='foaf:kn
ows'])), ',375,350)')"/>
  </xsl:attribute>
</image>
<!--in hinh chu nhat-->
<change>
<xsl:attribute name="pos">
  <xsl:value-of select="position()"/>
</xsl:attribute>
</change>
<rect x="60" y="60" width="100" height="25" rx="4" ry="4" style="stroke: green; fill:

```

```

none;">
  </rect>
</xsl:for-each>

<xsl:for-each select="foaf:knows/foaf:Person">
  <a>
  <!--IN TEXT RA MAN HINH-->
  <change><!--thay doi tag na`y-->
    <xsl:attribute name="pos">
      <xsl:value-of
select="position()+count(//*[name()='pc:knows'])+count(//*[name()='pc:isMemberOf'])"/>

      </xsl:attribute>
    </change>
    <text x="20" y="20" textLength="100" lengthAdjust="spacingAndGlyphs" style="text-
anchor: middle;" >
      <xsl:value-of select="foaf:name"/>
    </text>

    <!--BAT DAU IN HINH PERSON RA-->
    <change>
      <xsl:attribute name="pos">
        <xsl:value-of
select="position()+count(//*[name()='pc:knows'])+count(//*[name()='pc:isMemberOf'])"/>

        </xsl:attribute>
      </change>
      <image x="0" y="0" width="55" height="45"
xlink:href="http://localhost/webapp/EPerson/images/person.png">
      </image>
      <!--VE DUONG THANG QUAY-->
      <image x="430" y="340" width="170" height="20"
xlink:href="http://localhost/webapp/EPerson/images/muiten.jpg">
        <xsl:attribute name="transform">
          <xsl:value-of
select="concat('rotate(',
string(((position()+count(//*[name()='pc:knows'])+count(//*[name()='pc:isMemberOf']))*360)di
v
(count(//*[name()='pc:knows'])+count(//*[name()='pc:isMemberOf'])+count(//*[name()='foaf:kn

```

```

ows'])), ',375,350)')"/>
        </xsl:attribute>
    </image>

    <!--BAT DAU IN HINH CHU NHAT-->
    <change>
        <xsl:attribute name="pos">
            <xsl:value-of
select="position()+count(//*[name()='pc:knows'])+count(//*[name()='pc:isMemberOf'])"/>
            </xsl:attribute>
        </change>
        <rect x="60" y="60" width="100" height="25" rx="4" ry="4" style="stroke: green; fill:
none;">
        </rect>
    </a>
</xsl:for-each>

    <a >
        <xsl:apply-templates select="foaf:homepage"/>
        <text x="375" y="420" style="text-anchor: middle;font-family:Time New Romance;
font-weight: bold; font-size:15;">
            <xsl:value-of select="foaf:name"/>
        </text>
    </a>
</xsl:template>

<xsl:template match="foaf:homepage">
    <xsl:attribute name="xlink:href">
        <xsl:value-of select="@rdf:resource"/>
    </xsl:attribute>
</xsl:template>
</xsl:stylesheet>

```

### Nội dung của file *Group.xsl* :

```

<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

```

```

xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:dt="urn:schemas-microsoft-com:datatypes"
xmlns:foaf="http://xmlns.com/foaf/0.1/"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:dc="http://pURL.org/dc/elements/1.1/"
xmlns:pc="http://localhost/onto/pc/">

<xsl:namespace-alias stylesheet-prefix="dt" result-prefix="xsl"/>
<xsl:namespace-alias stylesheet-prefix="pc" result-prefix="xsl"/>
<xsl:namespace-alias stylesheet-prefix="rdf" result-prefix="xsl"/>
<xsl:namespace-alias stylesheet-prefix="foaf" result-prefix="xsl"/>
<xsl:output method="xml" version="1.0" omit-xml-declaration = "yes"/>

<xsl:template match="/">
    <svg width="1024" height="768">
        <xsl:apply-templates/>
    </svg>
</xsl:template>
<xsl:template match="rdf:RDF">
    <xsl:apply-templates/>
</xsl:template>
<xsl:template match="foaf:Group">
    <circle cx="400" cy="350" r="55" style="stroke: green; fill: #cfc; stroke-width:1.5;
"/>

    <!--lay hình -->
    <image x="363" y="320" width="75" height="60"
xlink:href="http://localhost/webapp/EPerson/images/group.png"/>

    <!-- cac kha nang-->
    <xsl:for-each select="pc:knows/pc:Competence">
        <!--ve mui ten -->
        <image x="456" y="340" width="200" height="20"
xlink:href="http://localhost/webapp/EPerson/images/muiten.jpg">
            <xsl:attribute name="transform">
                <xsl:value-of select="concat('rotate(', string((position()*360) div
(count(//*[name()='pc:knows'])+count(//*[name()='foaf:member'])), ',400,350)')"/>
            </xsl:attribute>

```

```

</image>
<!-- lay hình kha nang-->
<change>
  <xsl:attribute name="pos">
    <xsl:value-of select="position()"/>
  </xsl:attribute>
</change>
<image      x="0"      y="0"      width="55"      height="45"
xlink:href="http://localhost/webapp/EPerson/images/comp.png">
</image>
<!--lay ten kn-->
<change><!--thay doi tag na`y-->
  <xsl:attribute name="pos">
    <xsl:value-of select="position()"/>
  </xsl:attribute>
</change>
<text  x="20"  y="20"  textLength="100"  lengthAdjust="spacingAndGlyphs"
style="text-anchor: middle;" >
  <xsl:value-of select="foaf:name"/>
</text>
<!--in hình chu nhât-->
<change>
<xsl:attribute name="pos">
  <xsl:value-of select="position()"/>
</xsl:attribute>
</change>
<rect x="60" y="60" width="100" height="25" rx="4" ry="4" style="stroke: green; fill:
none;">
</rect>
</xsl:for-each>

<!-- cac thanh vien-->
<xsl:for-each select="foaf:member/foaf:Person">
<!--ve mui ten -->
<image      x="456"      y="340"      width="200"      height="20"
xlink:href="http://localhost/webapp/EPerson/images/muiten.jpg">
  <xsl:attribute name="transform">
    <xsl:value-of
                                                    select="concat('rotate(',

```

```

string(((position()+count(//*[name()='pc:knows'])*360


170


```



```

        </a>
</xsl:template>
<xsl:template match="foaf:homepage">
    <xsl:attribute name="xlink:href">
        <xsl:value-of select="@rdf:resource"/>
    </xsl:attribute>
</xsl:template>
</xsl:stylesheet>

```

Nội dung của file *Competence.xsl* :

```

<!--?xml version="1.0" encoding="UTF-8"?-->
<xsl:stylesheet version="1.0"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:dt="urn:schemas-microsoft-com:datatypes"
    xmlns:foaf="http://xmlns.com/foaf/0.1/"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:dc="http://pURL.org/dc/elements/1.1/"
    xmlns:pc="http://localhost/onto/pc/">

<xsl:namespace-alias stylesheet-prefix="dt" result-prefix="xsl"/>
<xsl:namespace-alias stylesheet-prefix="pc" result-prefix="xsl"/>
<xsl:namespace-alias stylesheet-prefix="rdf" result-prefix="xsl"/>
<xsl:namespace-alias stylesheet-prefix="foaf" result-prefix="xsl"/>
<xsl:output method="xml" version="1.0" omit-xml-declaration = "yes"/>

<xsl:template match="/">
    <svg width="1024" height="768">
        <xsl:apply-templates/>
    </svg>
</xsl:template>
<xsl:template match="rdf:RDF">
    <xsl:apply-templates/>
</xsl:template>
<xsl:template match="pc:Competence">
    <circle cx="400" cy="350" r="55" style="stroke: green; fill: #cfc; stroke-width:1.5;

```

```

"/>
  <!--lay hình -->
  <image      x="365"      y="320"      width="75"      height="65"
  xlink:href="http://localhost/webapp/EPerson/images/comp.png"/>

  <!-- cac nhom-->
  <xsl:for-each select="pc:isKnownBy/foaf:Group">
    <!--ve mui ten -->
    <image      x="456"      y="340"      width="200"      height="20"
    xlink:href="http://localhost/webapp/EPerson/images/muiten.jpg">
      <xsl:attribute name="transform">
        <xsl:value-of select="concat('rotate(', string((position()*360) div
(count(//*[name()='foaf:Person'])+count(//*[name()='foaf:Group'])), ',400,350'))"/>
      </xsl:attribute>
    </image>

    <change>
      <xsl:attribute name="pos">
        <xsl:value-of select="position()"/>
      </xsl:attribute>
    </change>
    <text x="20" y="20" textLength="100" lengthAdjust="spacingAndGlyphs"
    style="text-anchor: middle; stroke:white;" >
    </text>
    <!--in hình chu nhât-->
    <change>
      <xsl:attribute name="pos">
        <xsl:value-of select="position()"/>
      </xsl:attribute>
    </change>
    <rect x="60" y="60" width="100" height="25" rx="4" ry="4" style="stroke: white;
fill: none;">
    </rect>
    <change>
      <xsl:attribute name="pos">
        <xsl:value-of select="position()"/>
      </xsl:attribute>

```

```

</change>
<image      x="0"      y="0"      width="75"      height="60"
xlink:href="http://localhost/webapp/EPerson/images/group.png">
</image>
</xsl:for-each>

<!-- cac thanh vien-->
<xsl:for-each select="pc:isKnownBy/foaf:Person">
<!-- ve mui ten -->
<image      x="456"      y="340"      width="200"      height="20"
xlink:href="http://localhost/webapp/EPerson/images/muiten.jpg">
    <xsl:attribute name="transform">
        <xsl:value-of
            select="concat('rotate(',
string(((position()+count(//*[name()='foaf:Group']))*360)
div
(count(//*[name()='foaf:Person'])+count(//*[name()='foaf:Group'])), ',400,350'))"/>
    </xsl:attribute>
</image>
<!-- lay hinh nhom-->
<change>
    <xsl:attribute name="pos">
        <xsl:value-of select="position()+count(//*[name()='foaf:Group'])"/>
    </xsl:attribute>
</change>
<image      x="0"      y="0"      width="55"      height="45"
xlink:href="http://localhost/webapp/EPerson/images/person.png">
</image>
<!--lay ten nhom-->
<change>
    <xsl:attribute name="pos">
        <xsl:value-of
select="position()+count(//*[name()='foaf:Group'])"/>
    </xsl:attribute>
</change>
<text      x="20"      y="20"      textLength="100"      lengthAdjust="spacingAndGlyphs"
style="text-anchor: middle;" >
    <xsl:value-of select="foaf:name"/>
</text>

```

```

<!--in hình chu nhật-->
<change>
<xsl:attribute name="pos">
    <xsl:value-of select="position()+count(//*[name()='foaf:Group'])"/>

    </xsl:attribute>
</change>
<rect x="60" y="60" width="100" height="25" rx="4" ry="4" style="stroke: green;
fill: none;">
</rect>
</xsl:for-each>
<xsl:apply-templates select="foaf:homepage"/>
<text x="400" y="420" style="text-anchor: middle;font-family:Time New
Romance; font-weight: bold; font-size:20;">
<xsl:value-of select="foaf:name"/>
</text>
</xsl:template>

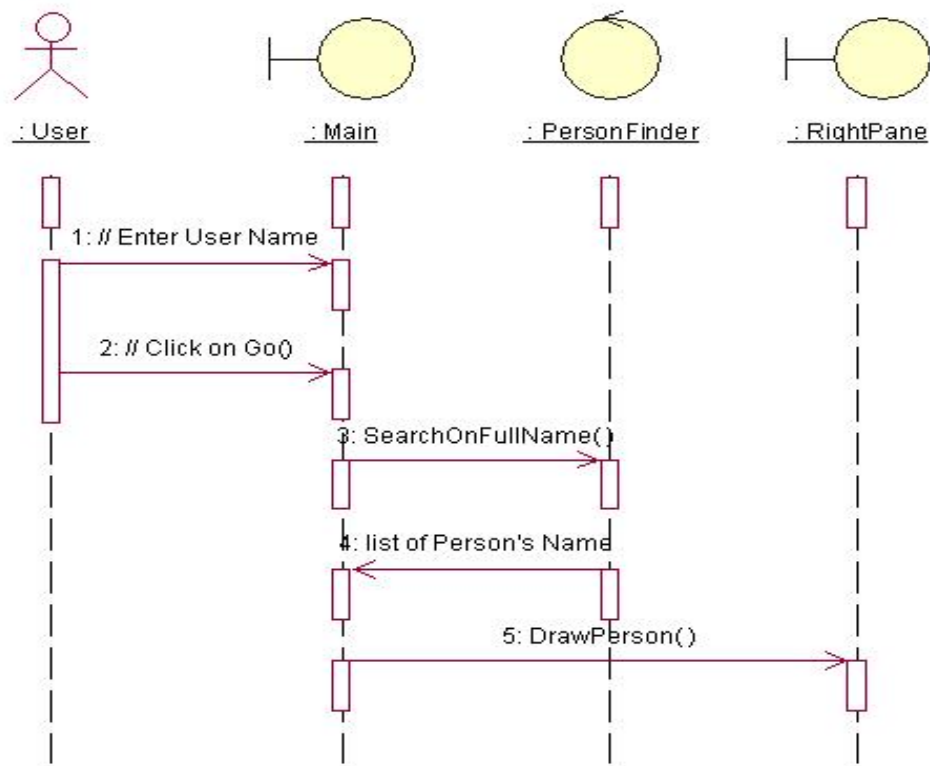
<xsl:template match="foaf:homepage">
    <xsl:attribute name="xlink:href">
        <xsl:value-of select="@rdf:resource"/>
    </xsl:attribute>
</xsl:template>
</xsl:stylesheet>
    
```

## 8.2 Thiết kế xử lý

Với các chức năng đã được đặc tả kỹ trong phần 6.2.3 trên chúng em có các các lược đồ tuần tự( Sequence Diagrams) và các lược đồ cộng tác( Collaboration Diagram) cho một số chức năng chính như sau :

- **Chức năng tìm người (Find Person) :**
- **Sequence Diagram**

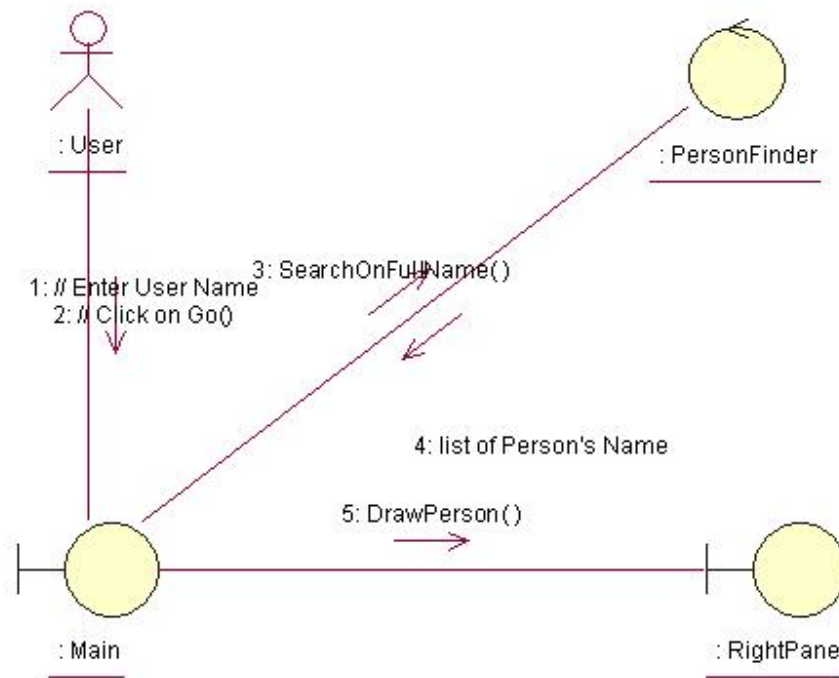
Sơ đồ Sequence Diagram cho chức năng tìm người.



**Hình 8-6** Sơ đồ Sequence Diagram cho chức năng tìm người.

○ **Collaboration Diagram**

Sơ đồ Collaboration Diagram cho chức năng tìm người.

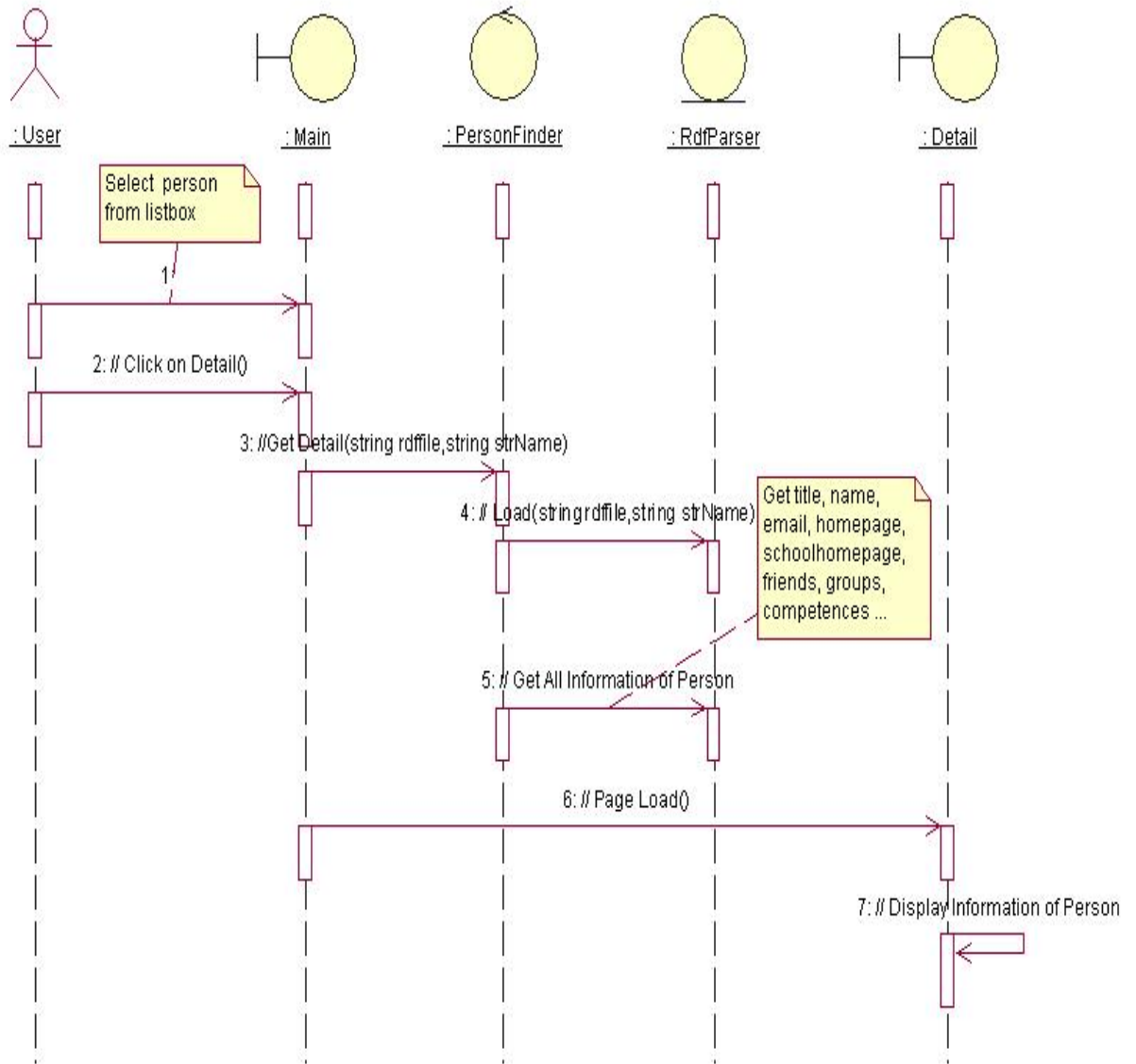


**Hình 8-7** Sơ đồ Collaboration Diagram cho chức năng tìm người.

- Chức năng xem chi tiết thông tin người (View Person's Detail) :

- **Sequence Diagram**

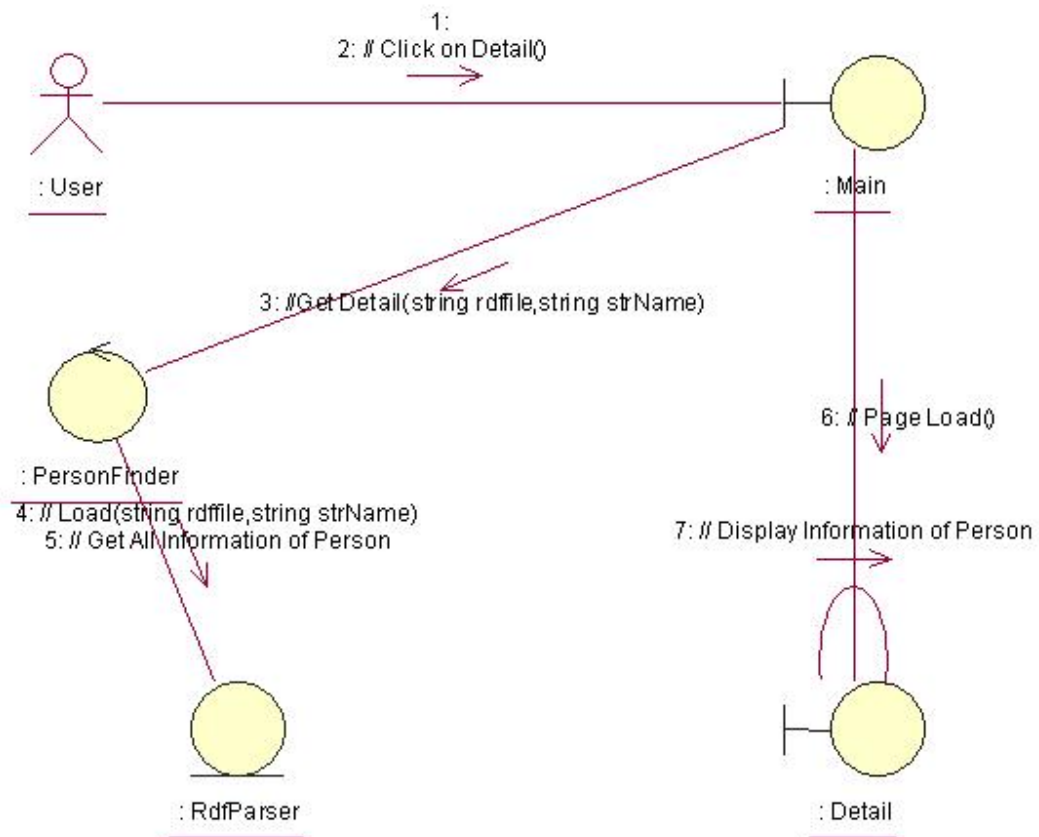
Sơ đồ Sequence Diagram cho chức năng hiển thị thông tin.



**Hình 8-8 Sơ đồ Sequence Diagram cho chức năng hiển thị thông tin.**

○ **Collaboration Diagram**

Sơ đồ Collaboration Diagram cho chức năng hiển thị thông tin.



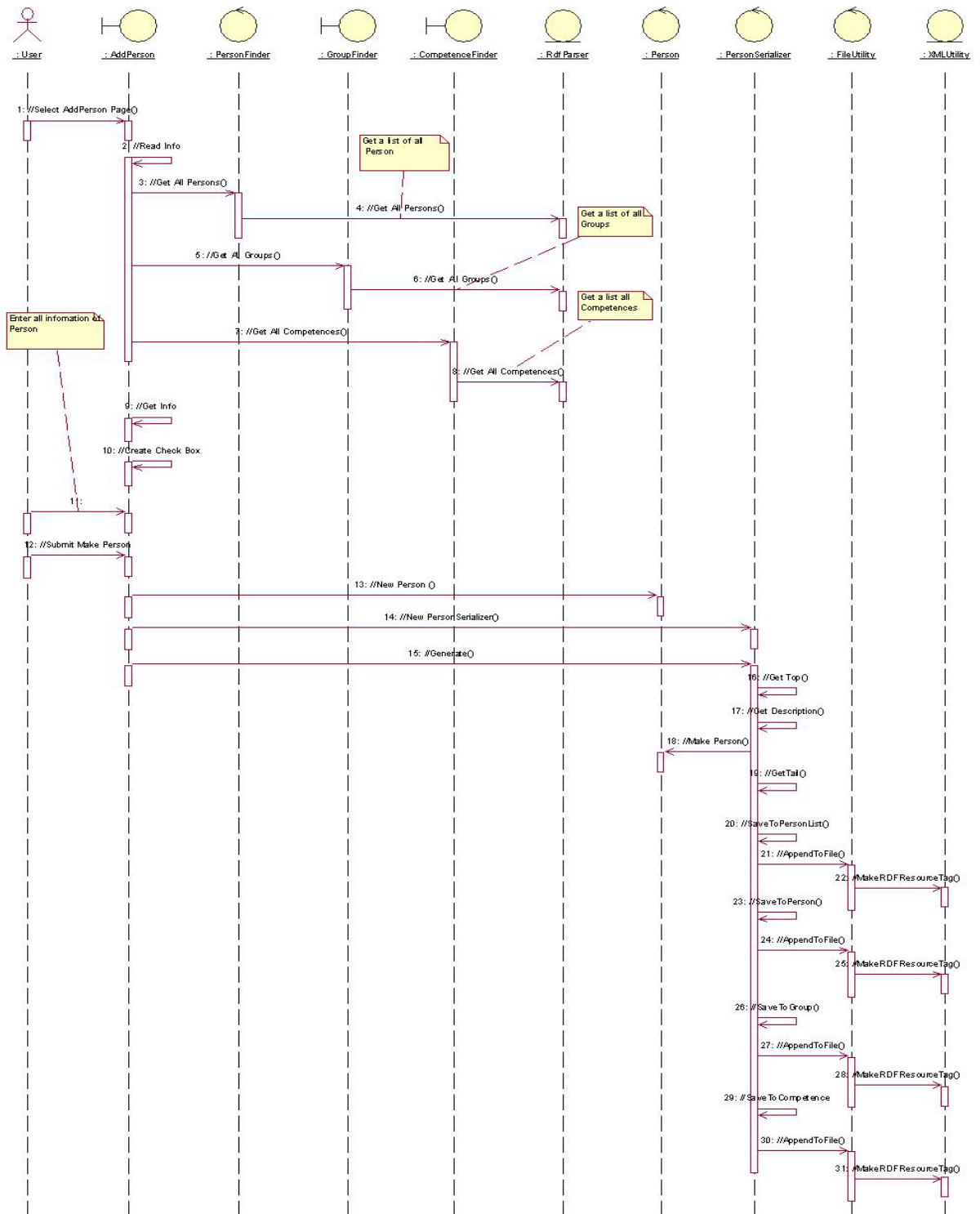
**Hình 8-9** Sơ đồ Collaboration Diagram cho chức năng hiển thị thông tin.

- Chức năng thêm người (Add Person) :

- Sequence Diagram

Sơ đồ Sequence Diagram cho chức năng thêm người.

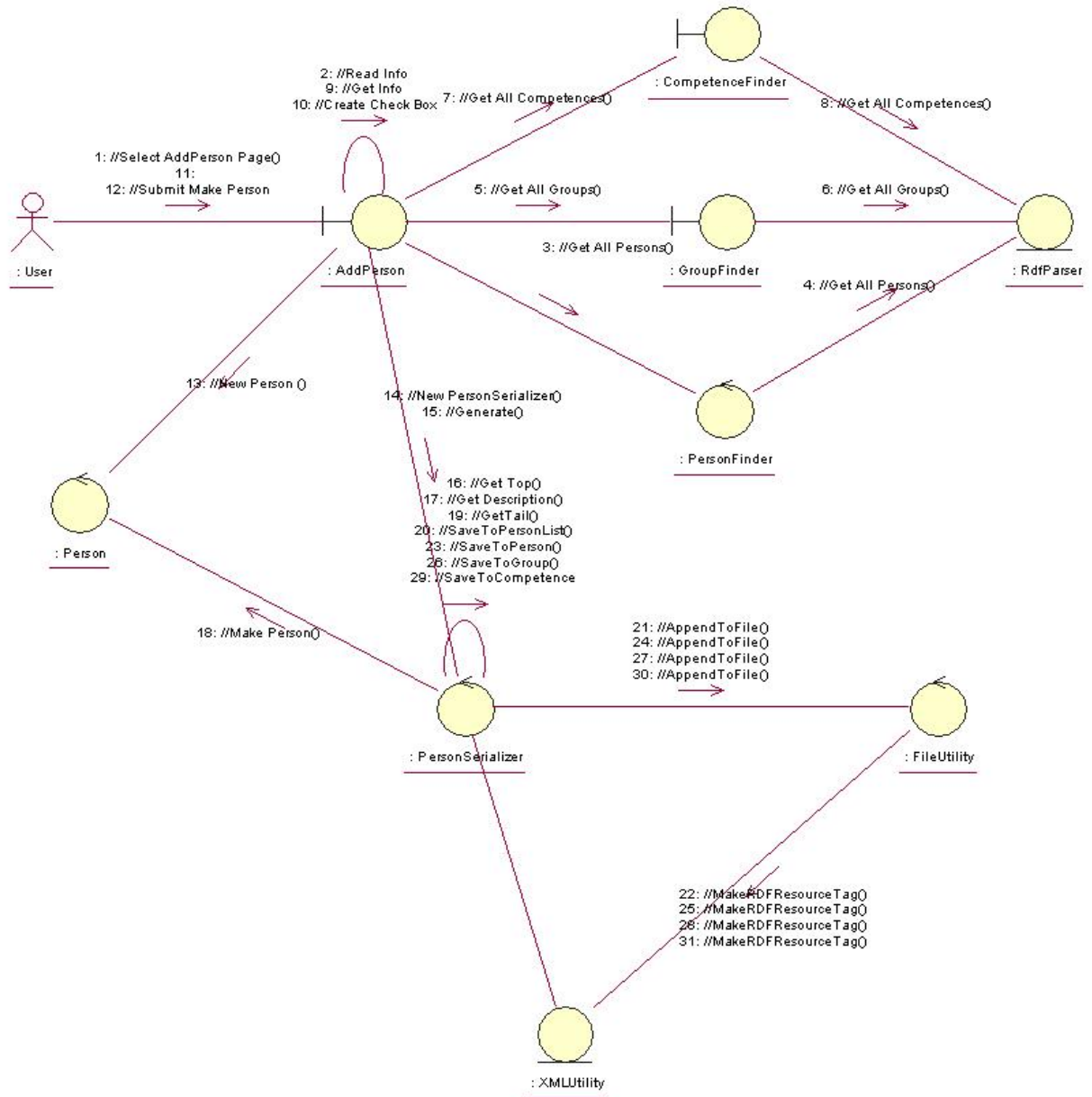




Hình 8-10 Sơ đồ Sequence Diagram cho chức năng thêm người.

o **Collaboration Diagram**

Sơ đồ Sequence Diagram cho chức năng thêm người.

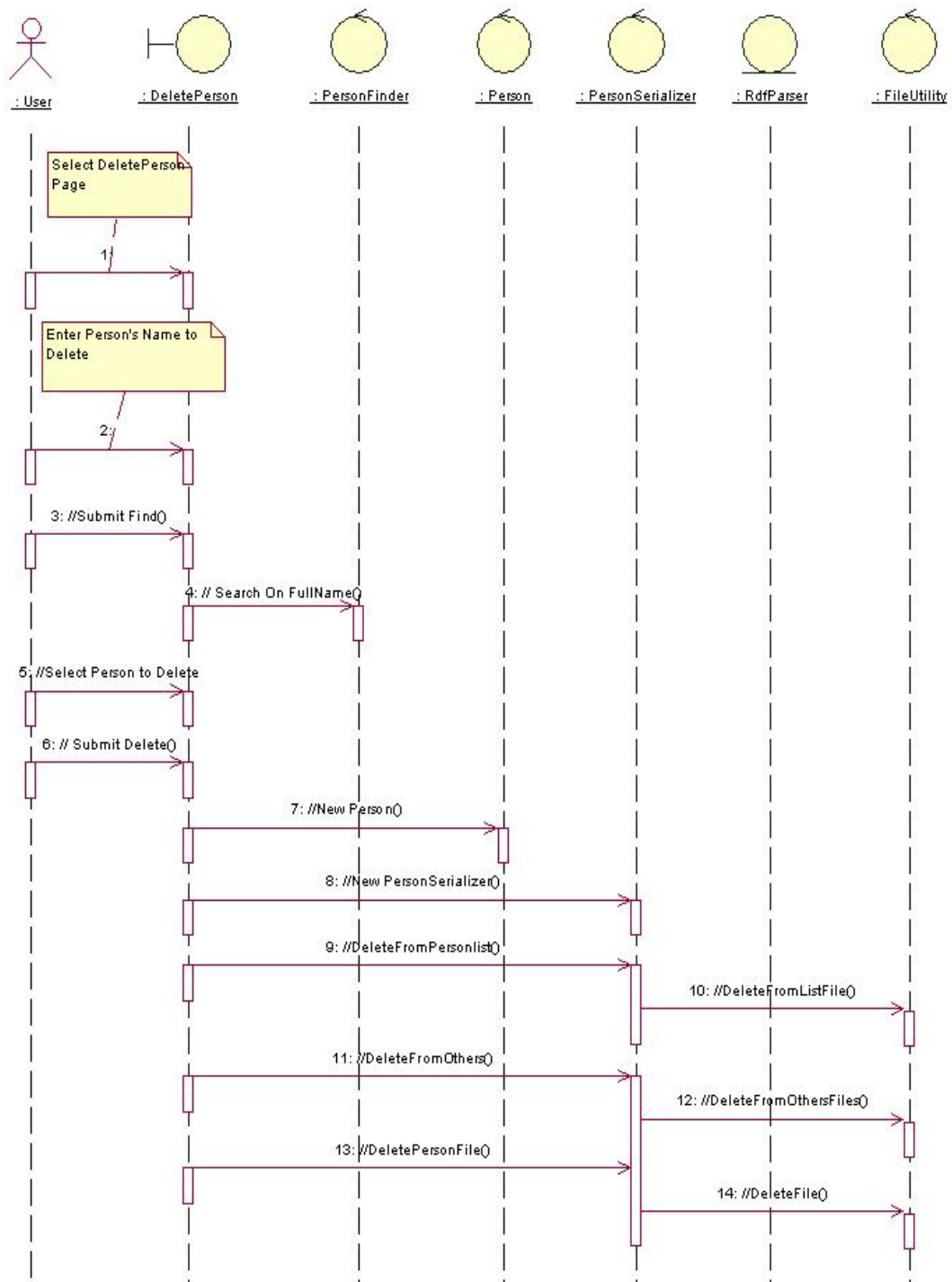


**Hình 8-11** Sơ đồ Sequence Diagram cho chức năng thêm người.

- Chức năng xoá người (Delete Person) :

o **Sequence Diagram**

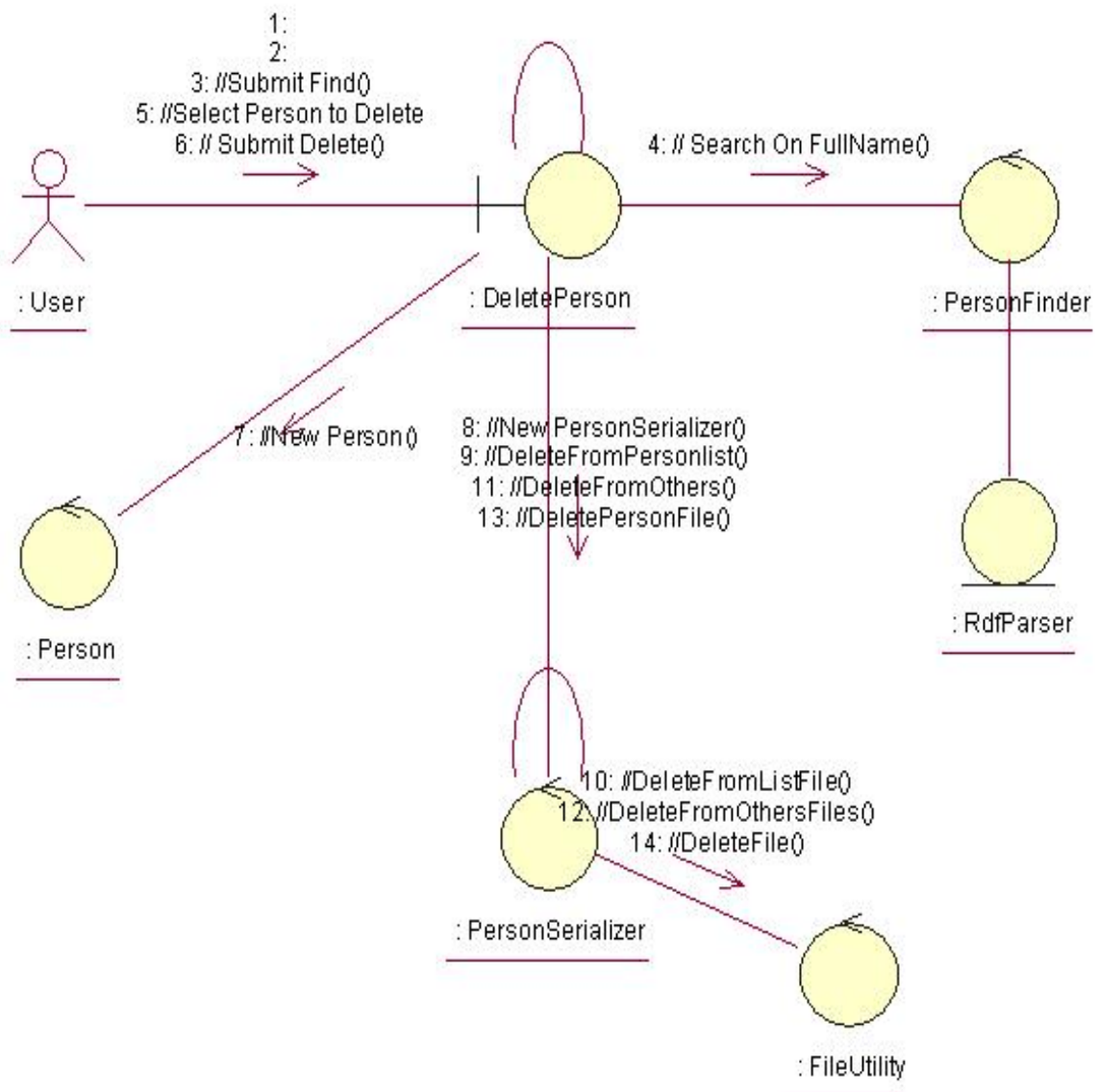
Sơ đồ Sequence Diagram cho chức năng thêm một người một người trong hệ thống



Hình 8-12 Sơ đồ Sequence Diagram cho chức năng xóa người.

o **Collaboration Diagram**

Sơ đồ Collaboration Diagram cho chức năng xóa người trong hệ thống.

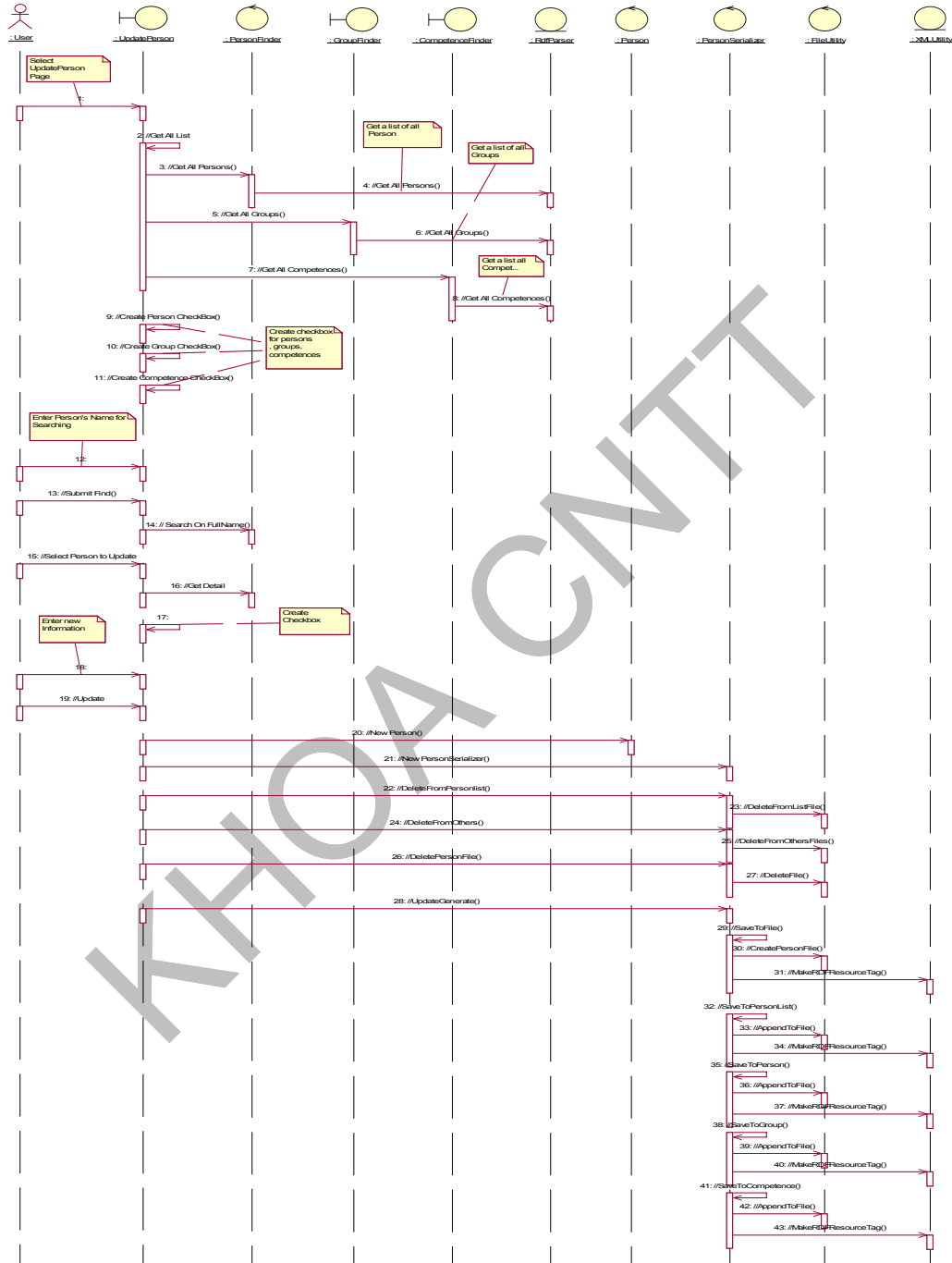


**Hình 8-13** Sơ đồ Collaboration Diagram cho chức năng xóa người.

- Chức năng cập nhật thông tin người (Update Person)

o **Sequence Diagram**

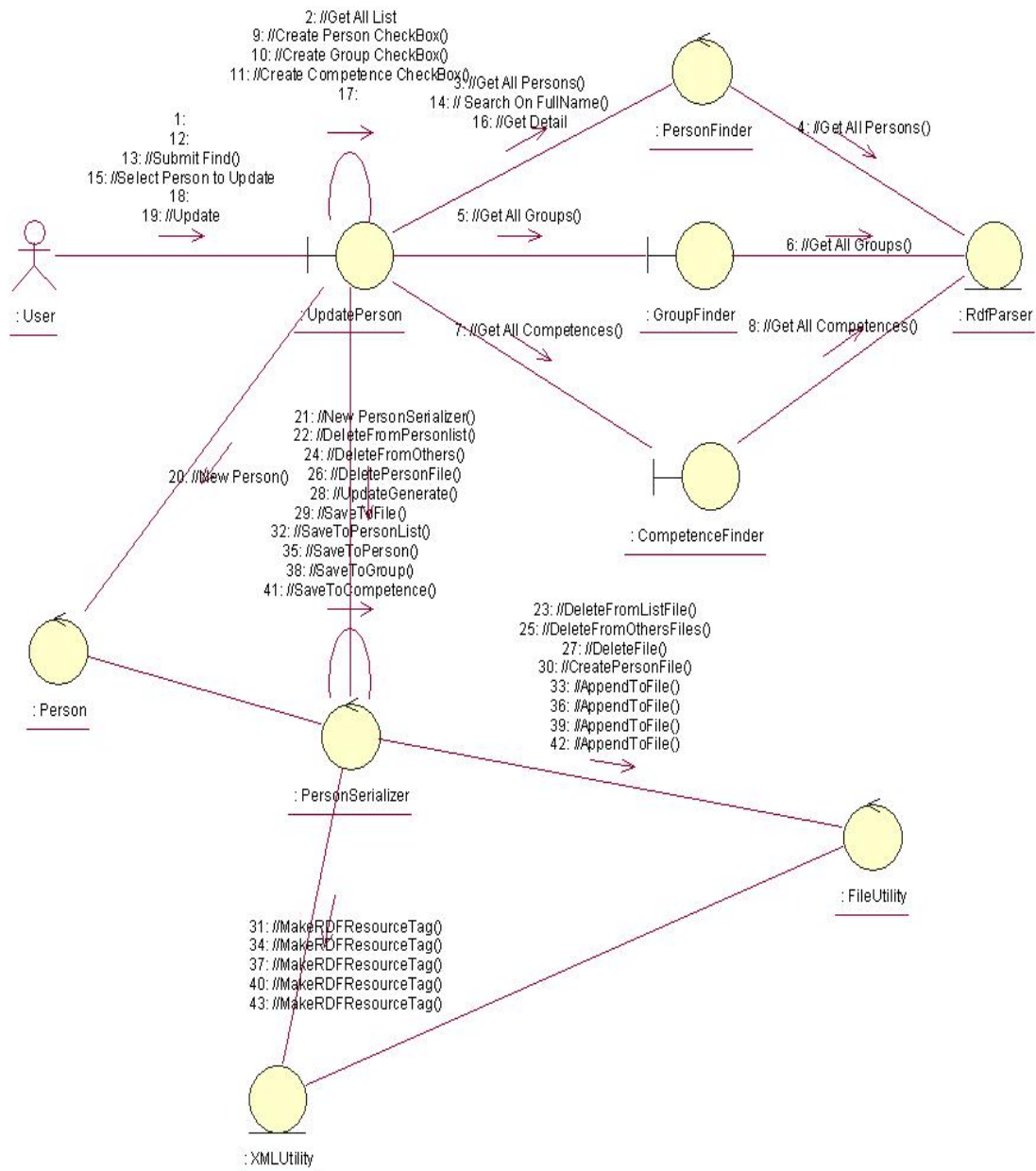
Sơ đồ Sequence Diagram cho chức năng cập nhật người..



Hình 8-14 Sơ đồ Sequence Diagram cho chức năng cập nhật người.

o **Collaboration Diagram**

Sơ đồ Collaboration Diagram cho chức năng cập nhật người.

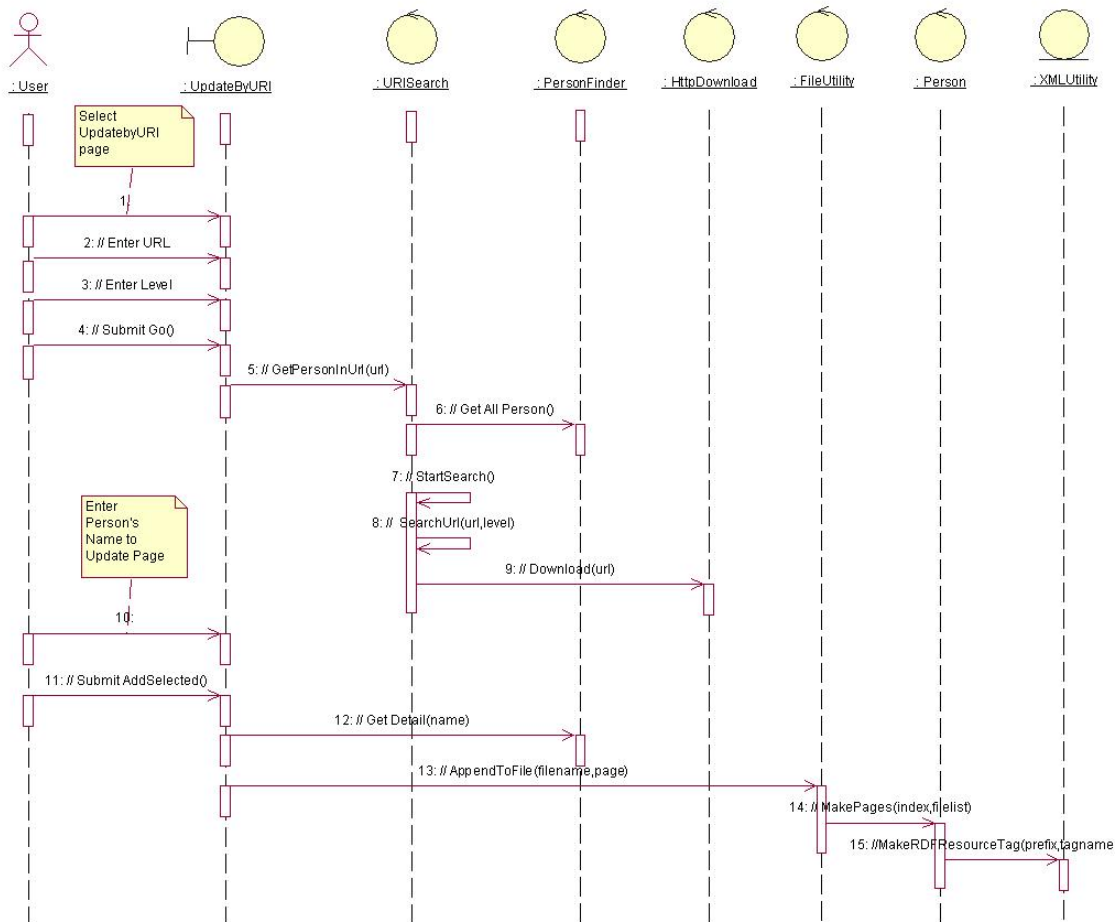


**Hình 8-15** Sơ đồ Collaboration Diagram cho chức năng cập nhật người.

- Chức năng cập nhật địa chỉ trang liên quan đến nhiều người (Update Personal Page by URL)

○ Sequence Diagram

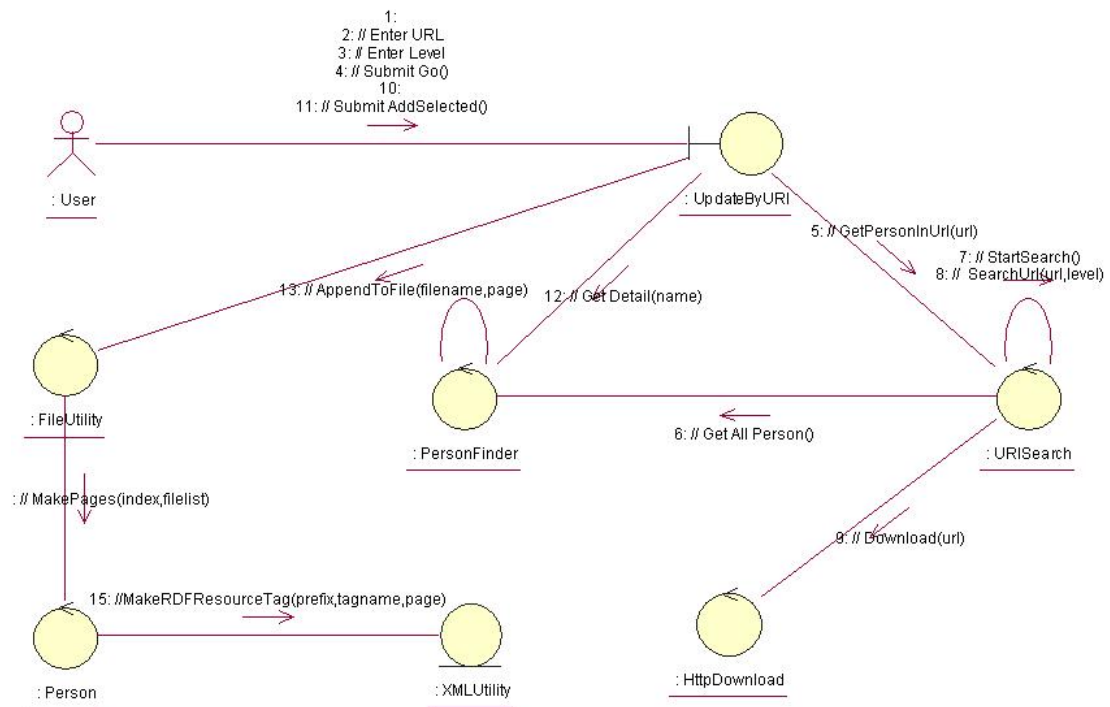
Sơ đồ Sequence Diagram cho chức năng cập nhật trang Web liên quan cho cá nhân.



**Hình 8-16** Sơ đồ Sequence Diagram cho chức năng cập nhật trang Web liên quan cho cá nhân.

○ Collaboration Diagram

Sơ đồ Collaboration Diagram cho chức năng cập nhật người trang Web liên quan.

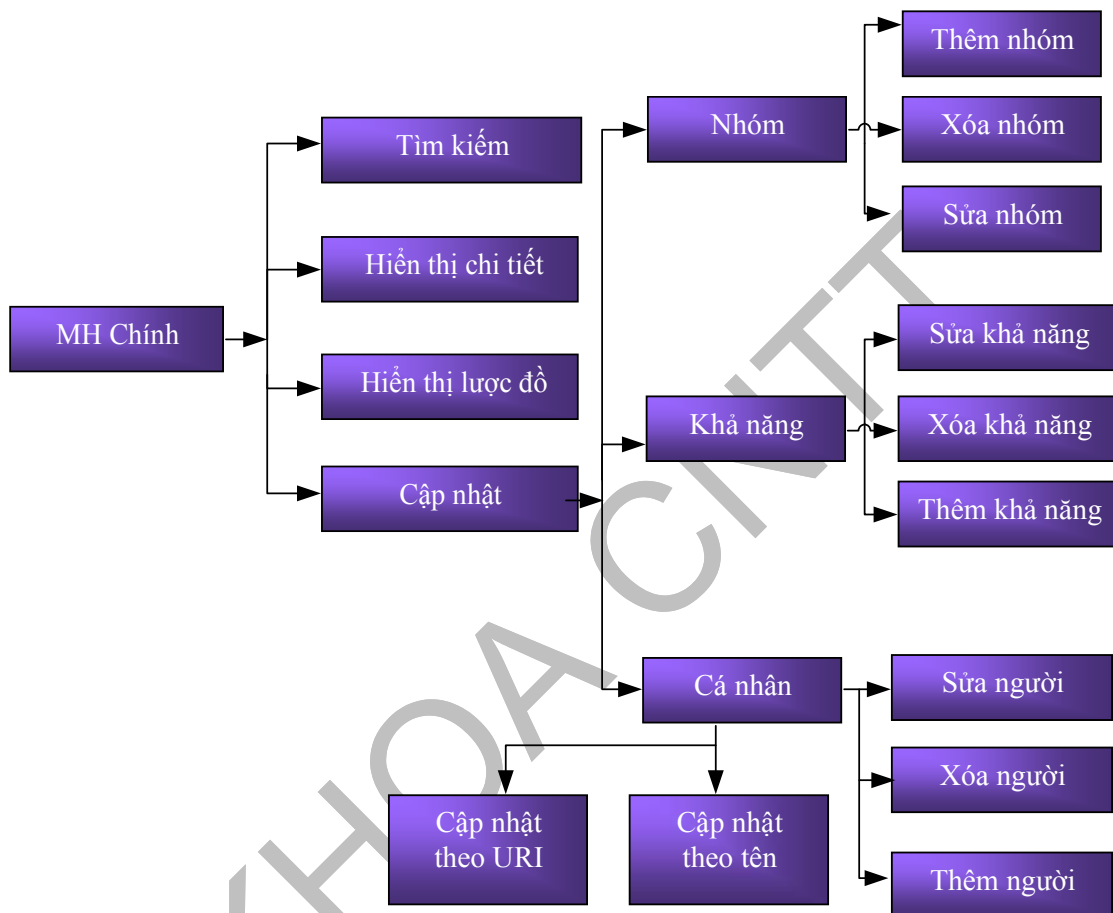


**Hình 8-17 Sơ đồ Collaboration Diagram cho chức năng cập nhật người trang Web liên quan.**



### 8.3 Thiết kế giao diện

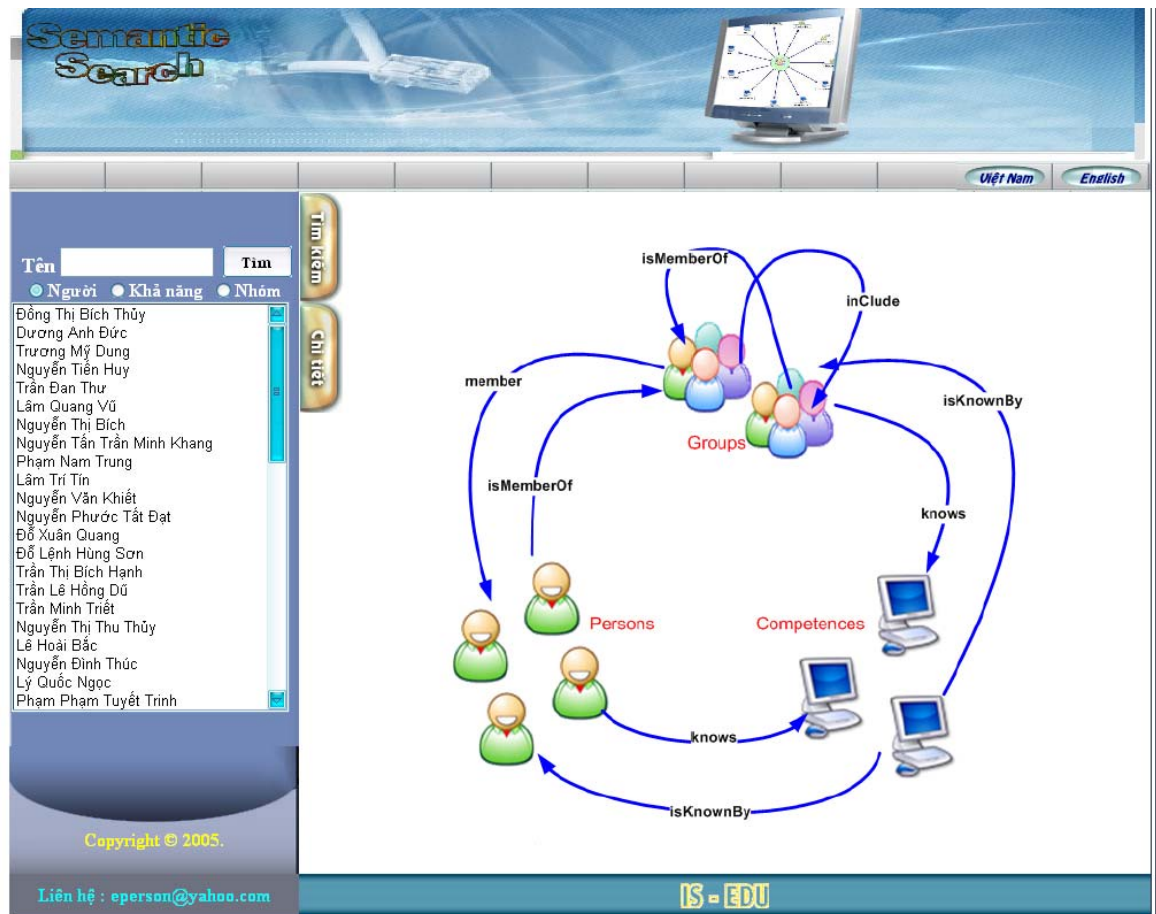
#### 8.3.1 Sơ đồ liên kết các màn hình



Hình 8-18 Sơ đồ liên kết các màn hình.

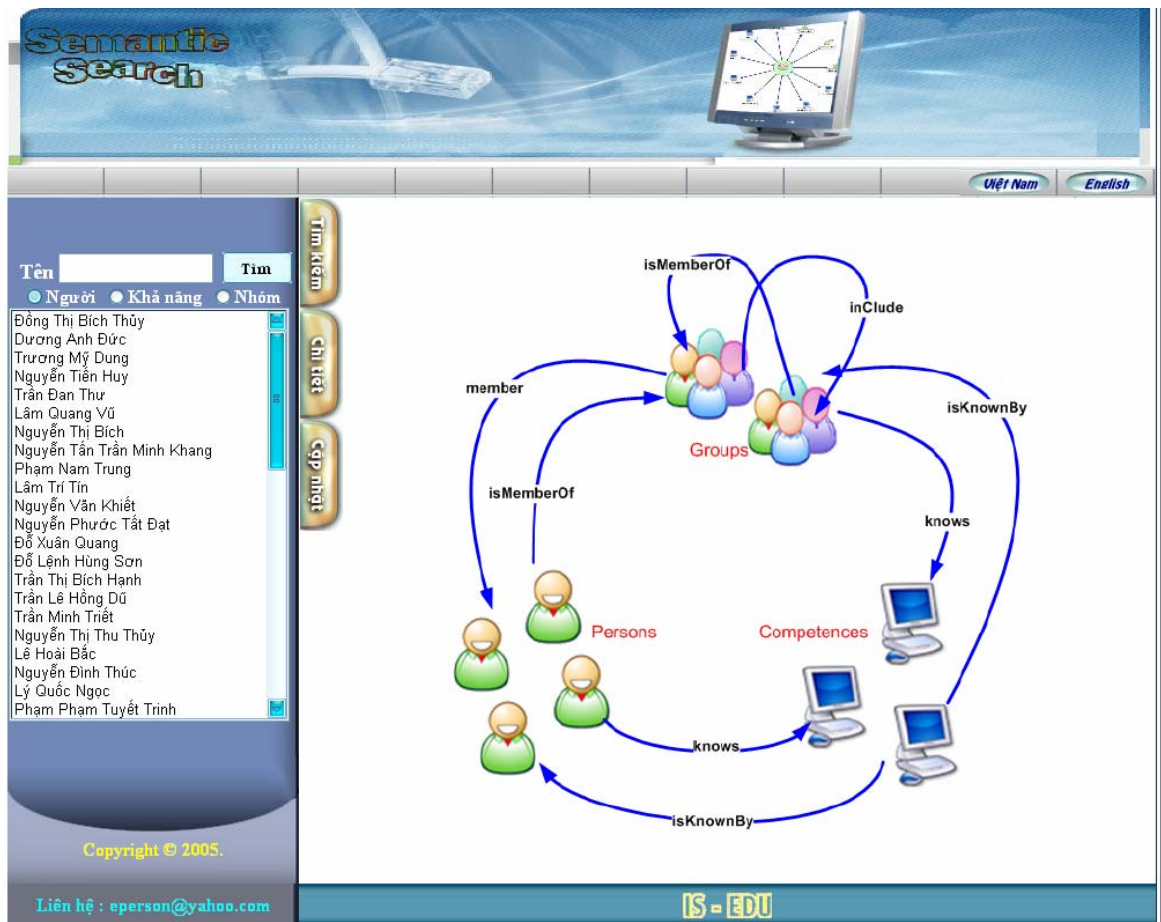
### 8.3.2 Một số màn hình chính của ứng dụng:

#### 8.3.2.1 Giao diện chính của phân hệ người dùng :



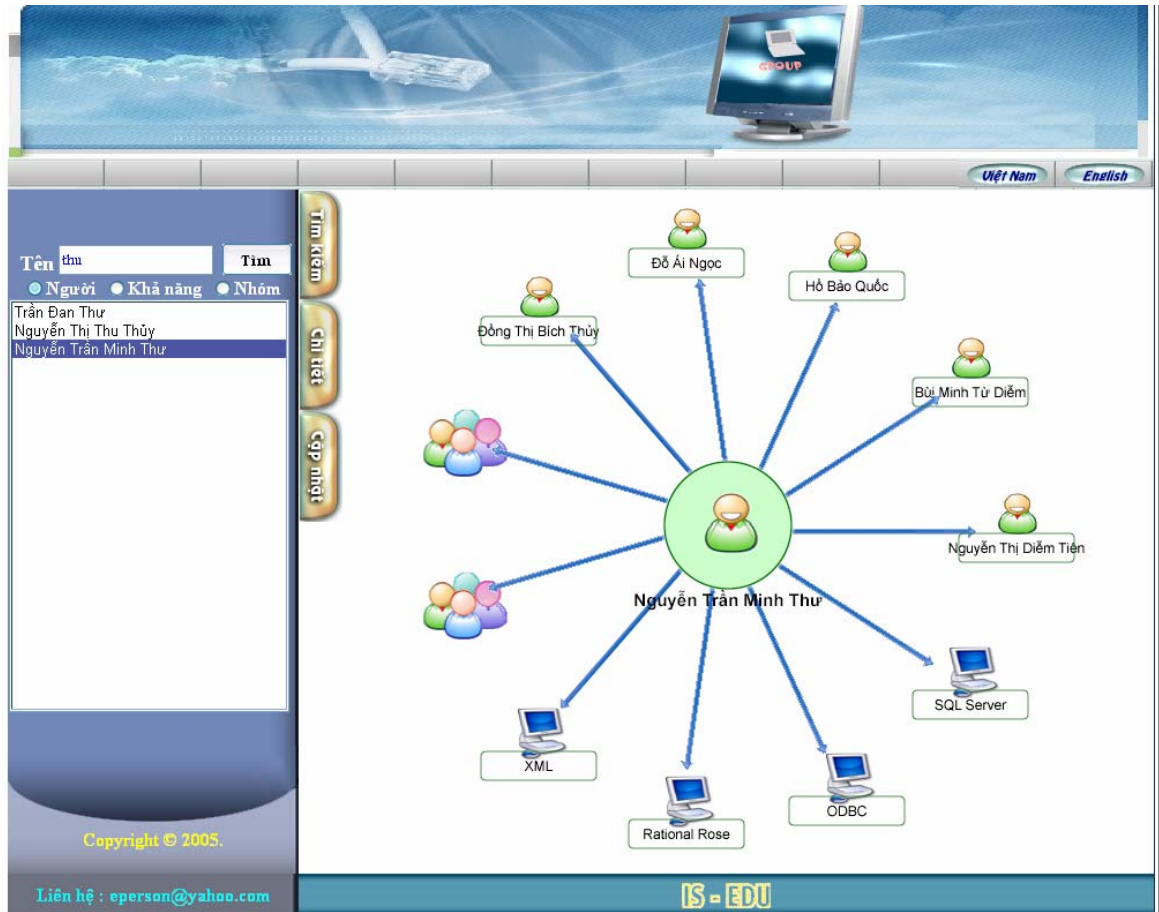
Hình 8-19 Giao diện chính của hệ thống người dùng.

### 8.3.2.2 *Giao diện chính của phân hệ admin :*



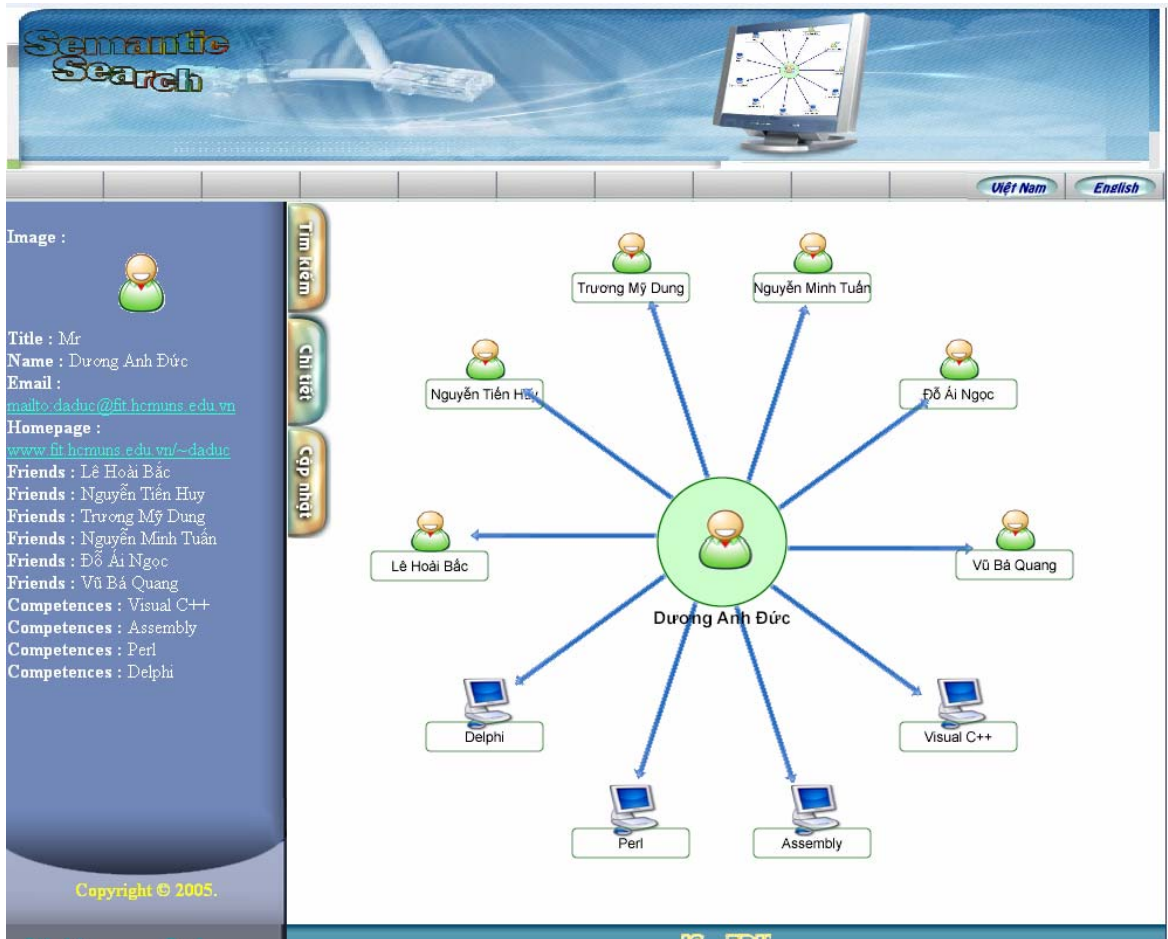
Hình 8-20 Giao diện chính của phân hệ admin.

🚩 **Giao diện của chức năng tìm người:**



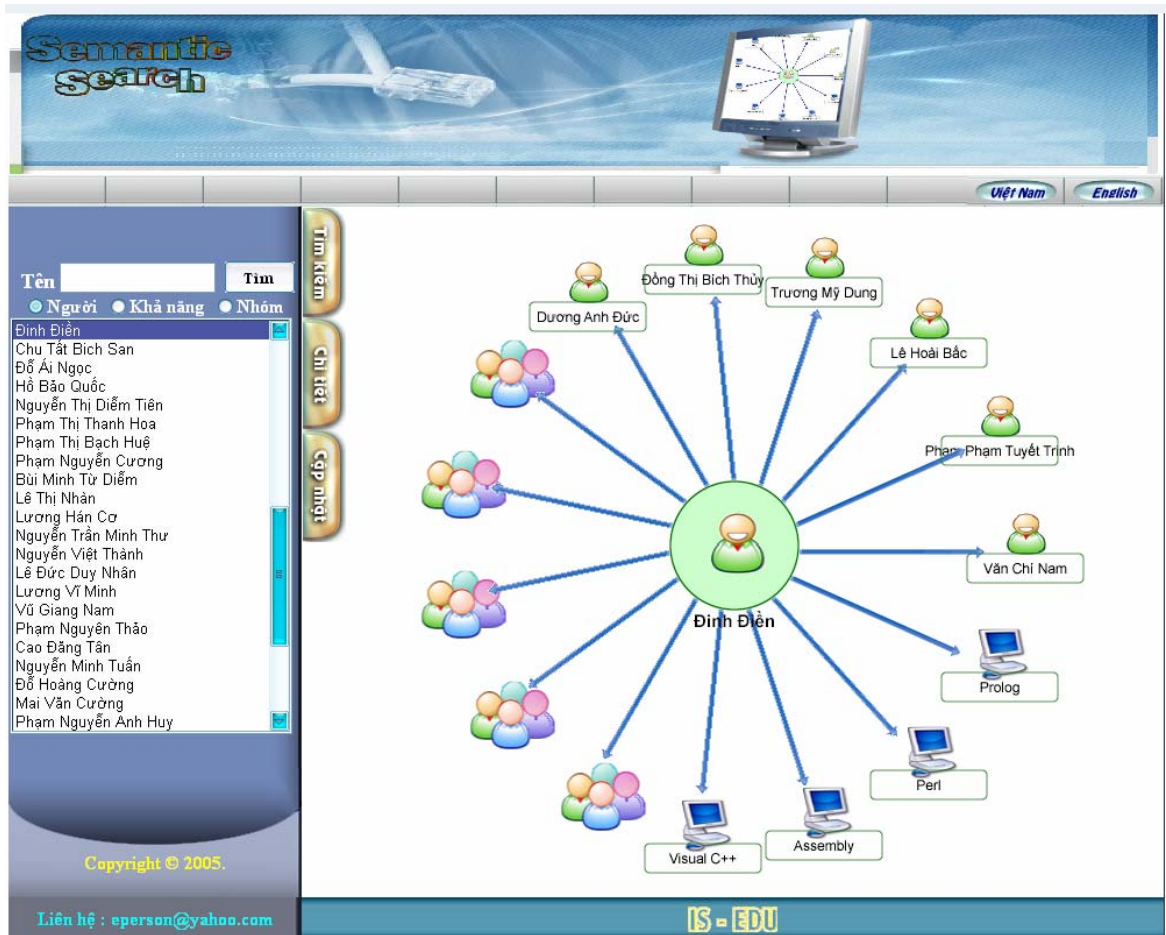
Hình 8-21 Giao diện tìm người

✚ **Giao diện của chức năng xem thông tin chi tiết:**



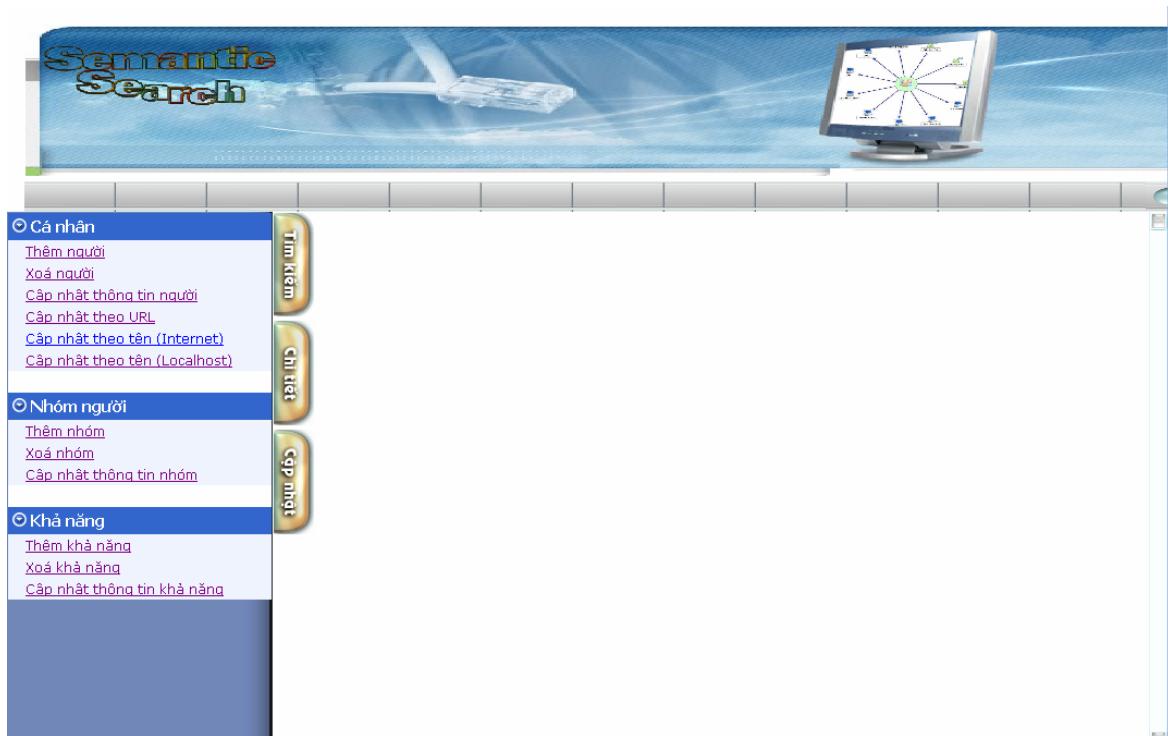
**Hình 8-22** Giao diện xem thông tin chi tiết.

🚩 **Giao diện của chức năng xem thông tin chi tiết bằng lược đồ:**



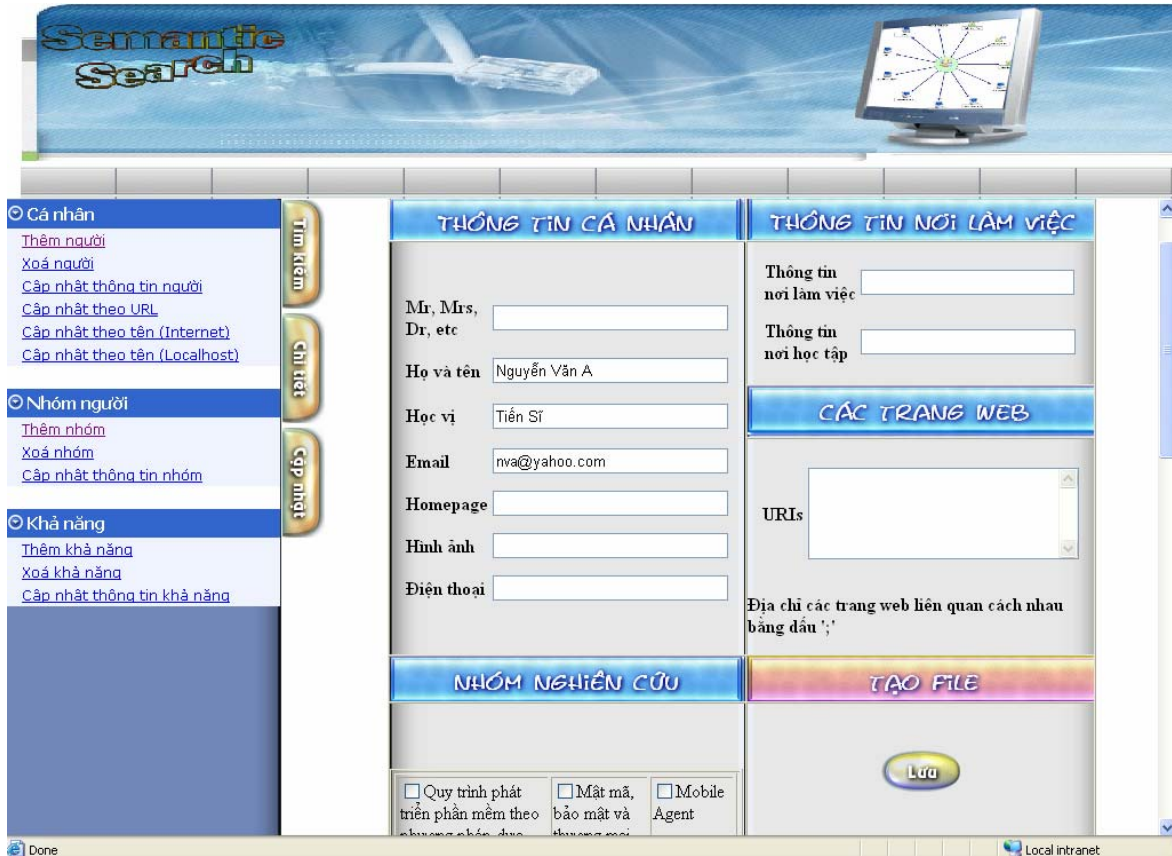
**Hình 8-23** Giao diện thực hiện chức năng xem chi tiết bằng lược đồ

✚ **Giao diện của chức năng cập nhật thông tin:**



Hình 8-24 Giao diện cập nhật thông tin.

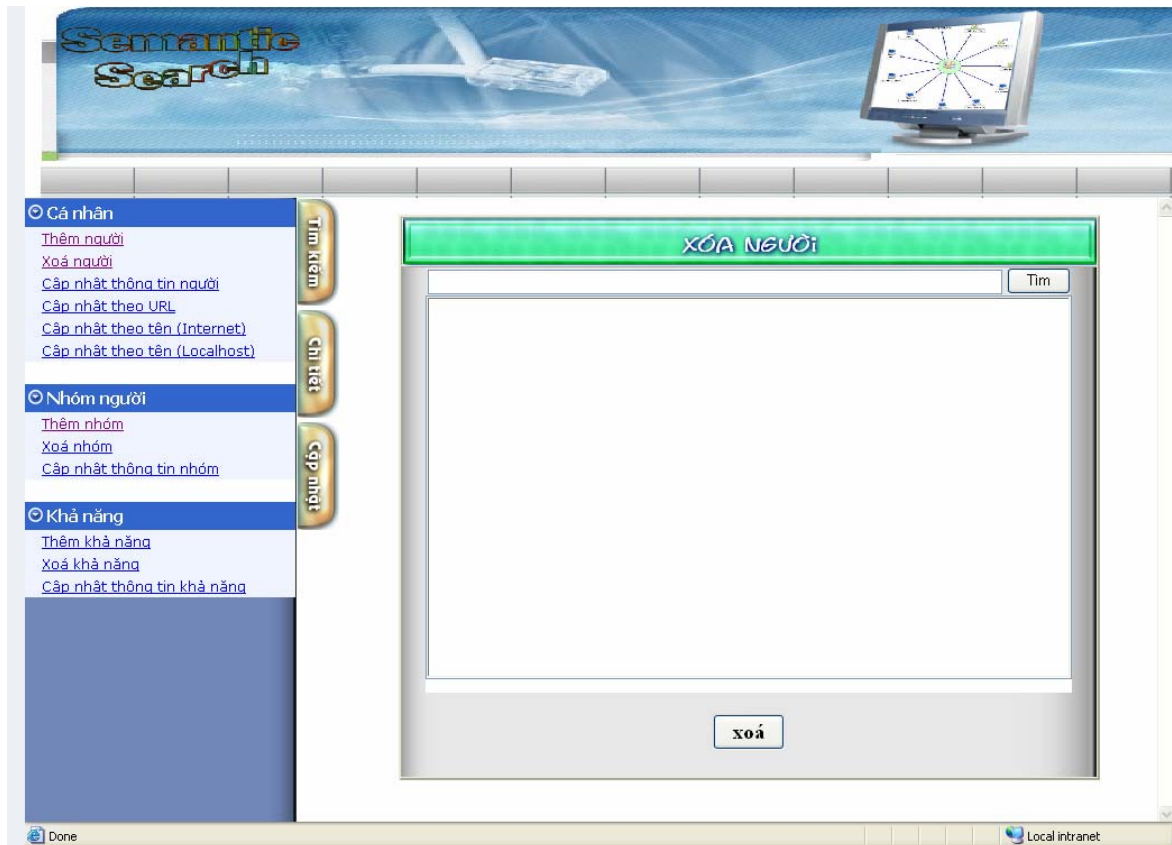
**+** Giao diện của chức năng thêm người:



**Hình 8-25** Giao diện thực hiện chức năng thêm người mới vào hệ thống.

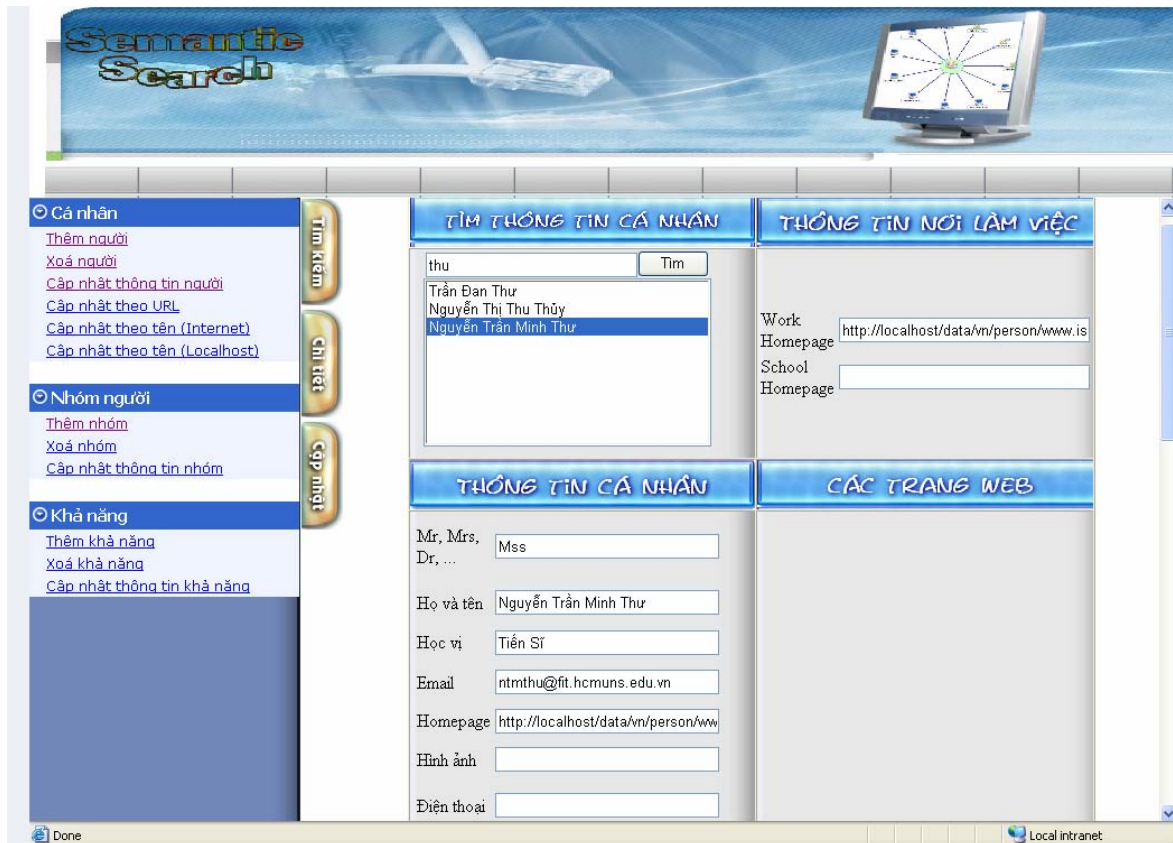


**✚ Giao diện của chức năng xóa một người ra khỏi hệ thống:**



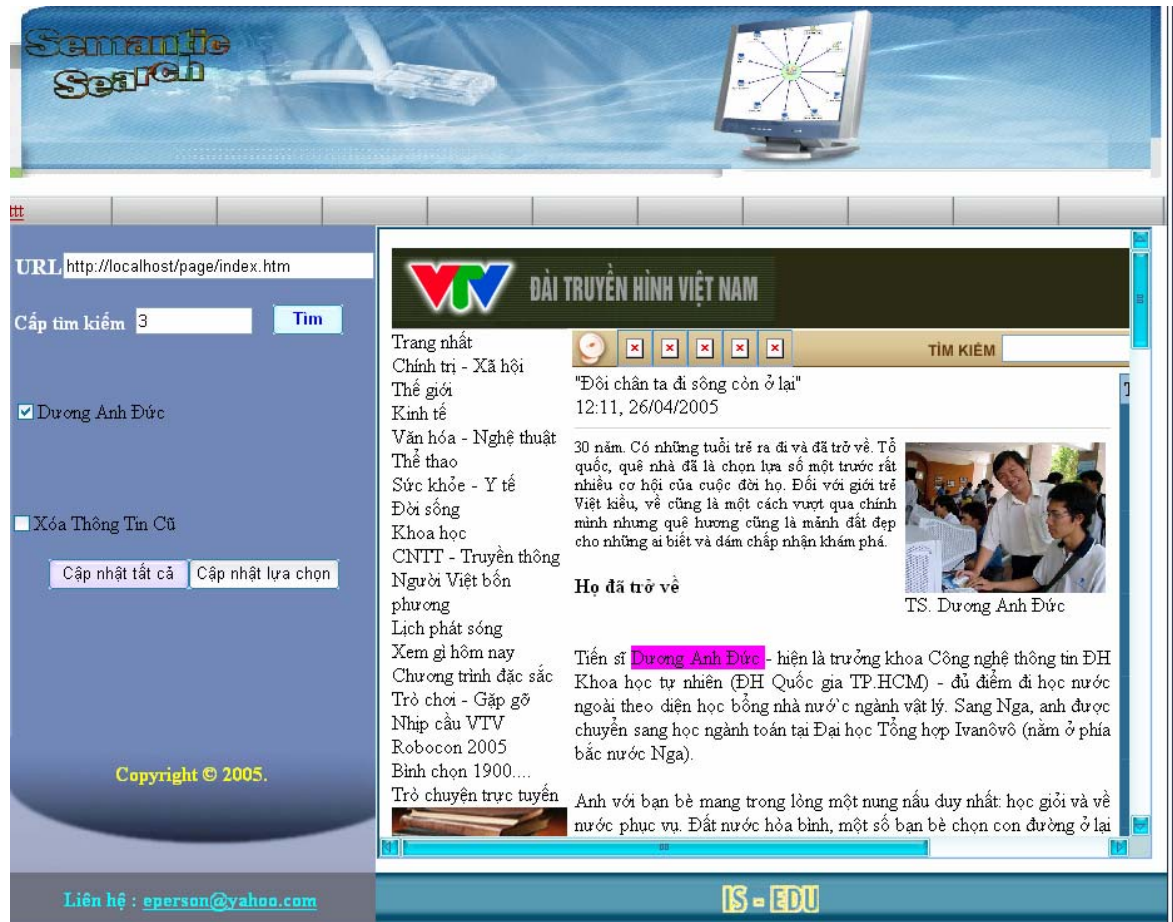
**Hình 8-26** Giao diện thực hiện chức năng xóa người ra khỏi vào hệ thống.

**+** Giao diện của chức năng sửa thông tin cá nhân:



**Hình 8-27** Giao diện thực hiện chức năng sửa thông tin một người trong hệ thống.

**+** Giao diện của chức năng cập nhật thông tin thông qua URL:



**Hình 8-28** Giao diện của chức năng cập nhật thông tin thông qua URL

**✚ Giao diện của chức năng cập nhật thông tin thông qua tên:**

| Chon cập nhật            | Tiêu đề trang web                           | Tóm tắt nội dung  | Tổng kết   | Liên kết  |
|--------------------------|---|---|------------|---|
| <input type="checkbox"/> | Nội dung tin                                | Ngh? các l?p c?a C8 Đ?ng Th? Bích Th? y. - VPK CNTT - Tu?n này, 13/4 đ?n 19/4, C8 Th?y s? đi công tác nên các l?p sau s? ngh?... - H? qu?n tr? CSDL (Pháp), ... | tong ket 1 | <a href="http://www.fit.hcmuns.edu.vn/home/tabid/95/Default.aspx?newsid=110">http://www.fit.hcmuns.edu.vn/home/tabid/95/Default.aspx?newsid=110</a> |
| <input type="checkbox"/> | tiêu đề                                     | giao thieu 1  | tong ket 1 | <a href="http://www.result1.com">www.result1.com</a>  |
| <input type="checkbox"/> | ELib  | Lu?n van khoa CNTT >> H? th?ng thông tin Thesis GVHD: PGSTS. Đ?NG TH? BÍCH TH?Y SVTH: Ph?m Nguyên Th?o S? l?n download: 453 Xem chi ti?t.....                   | tong ket 1 | <a href="http://www.fit.hcmuns.edu.vn/elib/ELib_v.aspx?IDCATEGORY=109">http://www.fit.hcmuns.edu.vn/elib/ELib_v.aspx?IDCATEGORY=109</a>             |
| <input type="checkbox"/> | ELib  | Lu?n van khoa CNTT >> H? th?ng thông tin Thesis GVHD: PGSTS. Đ?NG TH? BÍCH TH?Y SVTH: Ph?m Nguyên Th?o S? l?n download: 453 Xem chi ti?t.....                   | tong ket 1 | <a href="http://www.fit.hcmuns.edu.vn/Elb/ELib_v.aspx?IDCATEGORY=110">http://www.fit.hcmuns.edu.vn/Elb/ELib_v.aspx?IDCATEGORY=110</a>               |
| <input type="checkbox"/> | FAIR-05 Introduction                        | Đ?ng Th? Bích Th?y, Đ?i h?c Qu?c gia Thành ph? H? Chí Minh TS. Đ? Nang Toàn, Vi?n Công ngh? Thông tin. Ban chương trình ...                                     | tong ket 1 | <a href="http://www.hcmut.edu.vn/fair05/introduction.html">http://www.hcmut.edu.vn/fair05/introduction.html</a>                                     |
| <input type="checkbox"/> | Thông báo s? 1                              | Đ?ng Th? Bích Th?y. ĐHQG Tp.HCM TS. Đ? Nang Toàn. Vi?n CNTT. Ban Chương trình. Tru?ng ban. GS. TSKH. B?ch Hưng Khang. Vi?n CNTT. Phó tru?ng ban ...             | tong ket 1 | <a href="http://www.hcmut.edu.vn/fair05/downloads/thongbao.pdf">http://www.hcmut.edu.vn/fair05/downloads/thongbao.pdf</a>                           |
| <input type="checkbox"/> | Bao Binh Club   Tin t?c   Ph?m Th? Bích Th? | Ph?m Th? Bích Th?y t? bóng ném đ?n Billiards. ... Lo?i đ?ng đ?i M?ng Th?y và N?n? Tr?m. Bích Th?y t?c n? đ?i th?  | tong       | <a href="http://www.baobinhclub.com/modules.php?">http://www.baobinhclub.com/modules.php?</a>   |

**Hình 8-29** Giao diện của chức năng cập nhật thông tin thông qua tên.

## Chương 9 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 9.1 Kết luận

Ngày nay với một lượng thông tin khá nhiều trên mạng, đồng thời số lượng người dùng truy cập trên mạng rất nhiều đã làm cho việc tìm kiếm, trình bày và duy trì thông tin trên web ngày càng trở nên khó khăn hơn. Đúng lúc này công nghệ Semantic Web ra đời đã thật sự đem lại niềm hy vọng, một bước phát triển mới cho thế hệ web hiện tại. Công nghệ Semantic Web đã giúp cho con người có thể thêm ngữ nghĩa vào các tài liệu bằng ngôn ngữ mà máy tính có thể hiểu được. Điều này làm cho máy tính hiểu được thông tin trên Web, từ đó giúp cho việc tìm kiếm được nhanh hơn và chính xác hơn. Với công nghệ Semantic Web dữ liệu trên web đã được định nghĩa và liên kết theo cách mà máy tính có thể hiểu được không chỉ cho những mục đích hiển thị mà còn cho mục đích tự động, tích hợp và tái sử dụng dữ liệu qua các ứng dụng khác nhau.

Semantic Web thật sự mang lại nhiều thuận lợi như vậy nhưng để thật sự hiểu rõ và nắm vững Semantic Web là điều không dễ dàng. Do vậy, mục tiêu của đề tài luận văn đặt ra gồm hai phần chính như sau:

**Về lý thuyết:** mục tiêu là tìm hiểu, nghiên cứu về Semantic Web bao gồm các phần như: kiến trúc các tầng trên Semantic Web, các ngôn ngữ được sử dụng trong Semantic Web, RDF nền tảng của Semantic Web cũng như ngôn ngữ truy vấn trong RDF .

**Về phần ứng dụng minh họa:** mục tiêu là nắm vững được các kỹ thuật thao tác trên dữ liệu RDF: cách tạo một tài liệu RDF cũng như việc lưu trữ và truy vấn trên tài liệu này. Bên cạnh đó ứng dụng yêu cầu xây dựng được một ontology được sử dụng như một bộ từ điển cho ứng dụng.

Nội dung đề tài đưa ra khá rộng và chỉ được thực hiện trong vòng 6 tháng do vậy để hoàn tất đề tài này chúng em phải gặp rất nhiều khó khăn về mặt tìm kiếm tài liệu, đọc hiểu tài liệu và một số kỹ thuật để vận dụng kiến thức mình đã tìm hiểu được. Nhưng bù đắp lại những khó khăn vất vả đó là chúng em đã tiếp thu được rất nhiều kiến thức mới lạ và rất bổ ích về Semantic Web - một công nghệ Web tương lai và kết quả xứng đáng đó chính là chúng em đã hoàn thành được những mục tiêu, yêu cầu đưa ra của đề tài.

### 9.1.1 Kết quả về mặt nghiên cứu

Qua quá trình nghiên cứu và tìm hiểu về lĩnh vực Semantic Web, chúng em đã hiểu được và ghi nhận lại một số kết quả đạt được như sau:

- Tìm kiếm được nhiều tài liệu hay và bổ ích về Semantic Web .
- Hiểu được về công nghệ Semantic Web, một lĩnh vực đang được đánh giá đầy tiềm năng, hứa hẹn sẽ mang lại những lợi ích không lồ về mặt tri thức cũng như về mặt kinh tế . Kết quả đạt được khi nghiên cứu về Semantic Web như sau:
  - Nắm rõ về công nghệ Semantic Web, so sánh được Semantic Web với Web hiện tại cũng như những thuận lợi mà Semantic Web mang lại.
  - Nắm rõ về kiến trúc Semantic Web và vai trò các lớp trong kiến trúc Semantic Web .
  - Tìm hiểu về RDF , một ngôn ngữ nền tảng và giữ vai trò khá quan trọng trong kiến trúc Semantic Web bao gồm: lược đồ RDF dùng để mô tả các tài nguyên dưới dạng đồ thị, cú pháp RDF/XML dùng để tạo ra một tài liệu RDF , RDF Schema dùng để định nghĩa các lớp và các thuộc tính , FOAF- một bộ từ vựng mô tả thông tin về con người.
  - Tìm hiểu về ngôn ngữ truy vấn trong RDF đó là ngôn ngữ SPARQL, cách tạo ra một câu truy vấn cũng như việc truy vấn dữ liệu trong tài liệu RDF .

- Ngoài ra, luận văn cũng đã tìm hiểu một số ứng dụng liên quan đến Semantic Web và các công cụ hỗ trợ cho việc phát triển Semantic Web. Trong đó có một ứng dụng hết sức quan trọng trong lĩnh vực quản lý dữ liệu (tri thức), đó là Semantic Search Engine đã và đang được xây dựng trên thế giới.

### 9.1.2 Kết quả về chương trình ứng dụng

Đối với ứng dụng minh họa chúng em đã hoàn tất được mục tiêu của luận văn đề ra là nắm vững các kỹ thuật: tổ chức, lưu trữ và truy vấn trên dữ liệu RDF. Cách sử dụng bộ từ vựng FOAF để mô tả thông tin về con người và nhóm người trong hệ thống. Ngoài ra chúng em cũng đã xây dựng được bộ tự điển (ontology PC) mô tả thông tin của tài nguyên khả năng (*competence*) và mối quan hệ giữa tài nguyên này với tài nguyên con người (*person*) và tài nguyên nhóm người (*group*).

Về mặt chương trình kết quả thu được là hai phân hệ chương trình: một phân hệ dành cho người sử dụng và một phân hệ dành cho người quản trị.

- Đối phân hệ dành cho người dùng: Hỗ trợ việc tìm kiếm thông tin của một người, một nhóm nghiên cứu và khả năng của họ trong một tổ chức cùng với việc hiển thị thông tin liên kết với nhau một cách trực quan sinh động thông qua việc vẽ lược đồ chi tiết.
- Đối với phân hệ dành cho người quản trị: Hỗ trợ việc cập nhật thông tin về các đối tượng trong hệ thống đó là con người, nhóm người, và khả năng của họ trong một tổ chức. Ngoài ra hệ thống còn hỗ trợ việc cập nhật thông qua các URL và thông qua việc tìm kiếm trên mạng toàn cầu.

Sau khi hoàn tất được ứng dụng chúng em còn nắm thêm được các kiến thức về công nghệ: .NET, JavaScript, XSLT, SVG và quan trọng hơn cả là chúng em đã nắm được khả năng lưu trữ ngữ nghĩa của tài liệu RDF bằng cách dùng URI để mô tả các tài nguyên liên quan.

## **9.2 Hướng phát triển**

Luận văn đã được khép lại với những kết quả đạt được theo yêu cầu của đề tài đưa ra. Tuy nhiên qua quá trình thực hiện đề tài chúng em nhận thấy vẫn còn nhiều vấn đề khác liên quan đến lĩnh vực này cần được quan tâm. Do vậy những vấn đề này chúng em xem như hướng phát triển của đề tài để nghiên cứu tiếp sau này. Sau đây chúng em xin trình bày những hướng phát triển của đề tài:

- Đối với phần lý thuyết: Chúng em còn phải nghiên cứu sâu hơn về Semantic Web, đặc biệt là một số tầng trong kiến trúc Semantic Web ( như tầng Logic hay là Proof), từ đó mới có thể mở rộng ứng dụng thông qua việc xử lý ngôn ngữ tự nhiên để đưa tính ngữ nghĩa (semantic) vào trong ứng dụng của mình.
- Đối với ứng dụng đã xây dựng : Vì dữ liệu của ứng dụng chỉ có tính chất minh họa nên để đạt được những kết quả mà công nghệ Semantic mang lại cần tổ chức dữ liệu phong phú hơn, mở rộng phạm vi của tổ chức ra toàn thế giới.
- Phát triển ứng dụng thành một Semantic Search Engine hỗ trợ tìm kiếm đa lĩnh vực, cũng như đa thông tin
- Ứng dụng cũng cần có một bộ từ điển từ đồng nghĩa phong phú và thông minh về các lĩnh vực mà ứng dụng hỗ trợ, để hỗ trợ tìm kiếm có ngữ nghĩa tốt hơn.

Hy vọng những hướng phát triển này sẽ được thực hiện trong một tương lai gần.



## TÀI LIỆU THAM KHẢO



### Semantic Web:

- [1] Luận văn cử nhân tin học 2003 -Đỗ Ngọc Toàn- Đinh Hiếu Nghĩa
- [2] RDF Resource Description Framework, URI:<http://www.w3c.org/RDF/>
- [3] FOAF Vocabulary Specification - Namespace Document 2 Sept 2004 - FOAF Galway Edition, URI:<http://xmlns.com/foaf/0.1/>
- [4] They Rule, URI: <http://www.theyrule.net/>
- [5] The Friend of a Friend (FOAF) project, URI:<http://www.foaf-project.org>
- [6] Foaf explorer, URI: <http://xml.mfdconsult.dk/foaf/explorer/>
- [7] Foafnaut, URI: <http://www.foafnaut.org/>
- [8] Cascading Style Sheets (CSS), URI: <http://www.w3.org/Style/CSS/>
- [9] OWL, Ontology Web Language, W3C Recommendation, Feb. 2004, URI: <http://www.w3.org/2004/OWL/>
- [10] Berners-Lee, Tim (1998). Semantic Web Road map , URI: <http://www.w3.org/DesignIssues/Semantic>
- [11] Berners-Lee, Tim (1999). Web Architecture from 50,000 feet, URI: <http://www.w3.org/DesignIssues/Architecture>
- [12] Bray, Tim (1999). XML Namespaces by Example , URI: <http://www.xml.com/pub/a/1999/01/namespaces.html>
- [13] Berners-Lee, T. What the Semantic Web can represent. URI:<http://www.w3.org/DesignIssues/RDFnot.html>
- [14] Berners-Lee, T., Hendler, J., and Lassila, O. The Semantic Web. *Scientific American*, May 17 2001, 34-43.
- [15] Bieber, M. and Kacmar, C. Designing hypertext support for computational applications. *Communications of the ACM*.

- [16] Hendler, J., Berners-Lee, T., and Miller, E. Integrating Applications on the Semantic Web, 2002, URI: <http://www.w3.org/2002/07/swint> .
- [17] RDF/XML Syntax Specification (Revised), URI: <http://www.w3.org/TR/rdf-syntax-grammar/> , W3C Recommendation, February 10, 2004  
Dave Beckett, ed.
- [18] RDF Vocabulary Description Language 1.0: RDF Schema, URI: <http://www.w3.org/TR/rdf-schema/> , W3C Recommendation, February 10, 2004  
Dan Brickley, R.V. Guha, eds.
- [19] RDF Primer, <http://www.w3.org/TR/rdf-primer/> , W3C Recommendation, February 10, 2004  
Frank Manola, Eric Miller, eds.
- [20] Resource Description Framework (RDF): Concepts and Abstract Syntax, URI: <http://www.w3.org/TR/rdf-concepts/> , W3C Recommendation, February 10, 2004  
Graham Klyne, Jeremy Carroll, eds.
- [21] RDF Semantics, URI: <http://www.w3.org/TR/rdf-mt/> , W3C Recommendation, February 10, 2004  
Patrick Hayes, ed.
- [22] RDF Test Cases, <http://www.w3.org/TR/rdf-testcases/> , W3C Recommendation, February 10, 2004  
Jan Grant, Dave Beckett, eds.
- [23] <http://www.w3c.org/TR/1999/REC-rdf-syntax-19990222/>
- [24] <http://www.ilrt.bris.ac.uk/discovery/rdf/resources/>
- [25] <http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html>
- [26] <http://infomesh.net/2001/swintro/>
- [27] <http://www.semanticweb.org/>
- [28] <http://talad.sis.pitt.edu/marut/soa/>
- [29] <http://www.w3.org/TR/daml+oil-axioms>
- [30] <http://monkeyfist.com/articles/815>
- [31] <http://www.cs.umd.edu/projects/plus/SHOE/pubs/heflin-thesis.pdf>

[32] <http://www.w3.org/2001/12/semweb-fin/w3csw.html>

[33] <http://www.newarchitectmag.com/documents/s=2453/new1020218556549/index.html>

[34] <http://www.swi.psy.uva.nl/ETAIJ/SeWeb/scope.html>

[35] <http://www.netcrucible.com/semantic.html>

[36] [http://ftrain.com/google\\_takes\\_all.html](http://ftrain.com/google_takes_all.html)

### **SVG:**

[37] Scalable Vector Graphics (SVG), W3C Recommendation Jan. 2003, URI:

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

[38] <http://www.adobe.com/>

### **XSLT:**

[39] <http://www.w3.org/TR/xslt>

### **DAML+OIL:**

[40] <http://www.daml.org/>

[41] <http://www.ontoweb.org/index.htm>

[42] <http://www.semanticweb.org/SWWS/program/index.html>

[43] <http://www.xml.com/pub/a/2002/01/30/daml1.html>

[44] <http://www.w3.org/TR/daml+oil-model>

[45] <http://www.w3.org/TR/daml+oil-walkthru/>

[46] <http://ubot.lockheedmartin.com/ubot/>

## Dịch Vụ Web:

- [47] <http://www.daml.org/services/>
- [48] <http://www-106.ibm.com/developerworks/webservices/library/ws-soaprdf/?loc=dwmai>
- [49] <http://www.semanticweb.org/SWWS/program/index.html>
- [50] <http://msdn.microsoft.com/library/default.asp?url=/nhp/Default.asp?contentid=28000523>
- [51] <http://www6.software.ibm.com/developerworks/education/co-soap/index.html>  
(kostenlose Registrierung erforderlich)
- [52] <http://www.cs.vu.nl/~dieter/wsmf/>
- [53] <http://swws.semanticweb.org/>
- [54] <http://jcp.org/aboutJava/communityprocess/review/jsr026/>
- [55] <http://www.w3.org/TR/wsdl>
- [56] <http://www-4.ibm.com/software/solutions/webservices/pdf/WSFL.pdf>
- [57] [http://www.gotdotnet.com/team/xml\\_wsspecs/xlang-c/default.htm](http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm)
- [58] <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnglobspec/html/ws-routing.asp>
- [59] [http://gotdotnet.com/team/xml\\_wsspecs/dime/default.aspx](http://gotdotnet.com/team/xml_wsspecs/dime/default.aspx)
- [60] [http://playground.sun.com/srvloc/slp\\_white\\_paper.html](http://playground.sun.com/srvloc/slp_white_paper.html)
- [61] <http://paul.rutgers.edu/~kashyap/workshop.html>
- [62] <http://www.aifb.uni-karlsruhe.de/AIK/veranstaltungen/aik9/presentations/slides/wsmf-aifb.pdf>
- [63] <http://www-106.ibm.com/developerworks/library/ws-soaprdf/?n-ws-372>

[64] <http://www.ilrt.bris.ac.uk/discovery/2000/08/www9-slides/henrik/soaprdf.html>

### **Hệ Thống Agent:**

[65] <http://www.cs.umd.edu/users/hendler/AgentWeb.html>

[66] <http://www.agentcities.org/>

[67] <http://www.agentlink.org/>

[68] <http://sharon.cselt.it/projects/jade/>

[69] <http://www.insead.fr/CALT/Encyclopedia/ComputerSciences/Agents/>

[70] <http://autonomousAgents.org/2002/oas/>

[71] <http://ubot.lockheedmartin.com/ubot/>

[72] <http://www.csc.liv.ac.uk/~valli/OMAS-CFP.html>

### **Truy Vấn Dữ Liệu RDF:**

[73] <http://triple.semanticweb.org/>

[74] <http://139.91.183.30:9090/RDF/RQL/index.html>

[75] <http://139.91.183.30:9090/RDF/publications/tr308.pdf>

[76] <http://infomesh.net/2001/12/map/data.n3>

[77] <http://www.ninebynine.org/RDFNotes/RDFForLittleLanguages.htm>

[78] rdfDB query language -> <http://guha.com/rdfdb/query.html>

[79] RQL by ICS-FORTH -> <http://139.91.183.30:9090/RDF/RQL/>

a) The RQL User Manual -> <http://139.91.183.30:9090/RDF/RQL/Manual.html>

b) RQL: A Declarative Query Language for RDF\*

<http://139.91.183.30:9090/RDF/publications/www2002/www2002.pdf>

- c) Querying Community Web Portals\*  
<http://139.91.183.30:9090/RDF/publications/sigmod2000.html>
- [80] Sesame RQL by Administrator Nederland -> <http://sesame.administrator.nl/>  
(slightly deviates with original ICS-FORTH RQL)
- a) Query Language Definition ->  
<http://sesame.administrator.nl/publications/del9.pdf>
- b) Sesame: A generic Architecture for Storing and Querying RDF and RDF Schema -> <http://sesame.administrator.nl/publications/del10.pdf>
- c) Babysteps in Sesame RQL -> <http://sesame.administrator.nl/publications/rql-babysteps.pdf>
- [81] RDF Squish query language and Java implementation ->  
<http://ilrt.org/discovery/2001/02/squish/>
- a) Inkling: RDF query using SquishQL -> <http://swordfish.rdfweb.org/rdfquery/>
- b) RDF: Extending and Querying RSS channels ->  
<http://ilrt.org/discovery/2000/11/rss-query/>
- c) “Three Implementations of SquishQL, a Simple RDF Query Language” ->  
<http://freeee.free.fr/iswc/accepted.html>
- [82] RDQL in Jena toolkit -> <http://www.hpl.hp.com/semweb/>
- a) Matching RDF Graphs -> <http://www-uk.hpl.hp.com/people/jjc/tmp/matching.pdf>
- b) Jena: Implementing the RDF Model and Syntax Specification -> <http://www-uk.hpl.hp.com/people/bwm/papers/20001221-paper/>
- [83] TRIPLE <http://www.dfki.uni-kl.de/frodo/triple/>
- [84] TRIPLE-An RDF Query, Inference, and Transformation Language ->  
<http://www.dfki.uni-kl.de/frodo/triple/TripleReport.pdf>

- [85] Intellidimension RDFQL Reference Manual (aka RDFGateway) ->  
<http://www.intellidimension.com/RDFGateway/Docs/rdfqlmanual.asp>
- [86] RDFPath <http://zoe.mathematik.uni-osnabrueck.de/RDFPath/>  
a) “Transforming RDF with RDFPath” . <http://zoe.mathematik.uni-osnabrueck.de/QAT/Transform/RDFTransform.pdf>
- [87] Versa, the RDF query language ->  
<http://uche.ogbuji.net:8080/uche.ogbuji.net/tech/rdf/versa/main.doc?xslt=../../main.xslt>  
a) Versa Specification ->  
[http://uche.ogbuji.net:8080/uche.ogbuji.net/tech/rdf/versa/versa.doc?xslt=/ftss/data/docbook\\_html1.xslt](http://uche.ogbuji.net:8080/uche.ogbuji.net/tech/rdf/versa/versa.doc?xslt=/ftss/data/docbook_html1.xslt)  
b) Versa by Example ->  
<http://uche.ogbuji.net:8080/uche.ogbuji.net/tech/rdf/versa/versa-by-example.txt>
- [88] DAML+OIL Query Language (DQL) Specification ->  
<http://www.daml.org/listarchive/joint-committee/1081.html>
- [89] A RDF Query Language based on DAML ->  
<http://www.langdale.com.au/RDF/DAML-Query.html>
- [90] KAON Tool Suite -> <http://kaon.aifb.uni-karlsruhe.de/>
- [91] Cerebra: RDF Inference Engine -> <http://www.networkinference.com/>

### **CIDOC/ABC:**

[92] <http://metadata.net/harmony/>

[93] <http://cidoc.ics.forth.gr/>

### **Amaya/Annotea:**

[94] <http://www.w3.org/2001/Annotea/>

[95] <http://www.w3.org/Amaya/>

### **UML và RDF:**

[96] <http://jodi.ecs.soton.ac.uk/Articles/v01/i08/Cranefield/>

[97] <http://www-db.stanford.edu/~melnik/rdf/uml/>

[98] <http://maria.grci.com/codipsite/Tools/Tools.html>

[99] <http://www.semanticweb.org/SWWS/program/full/paper1.pdf>

[100] <http://www.interdataworking.com/converter/>

[101] <http://ubot.lockheedmartin.com/ubot/>

KHOA CNTT



## **PHỤ LỤC A**

### **CÁC ỨNG DỤNG VÀ DỰ ÁN VỀ Semantic Web**

- RDF Gateway của Intellidimension - một bộ ứng dụng semantic web trên nền MS Windows NT/2000/XP cho phép quản lý dữ liệu dạng RDF. Nó cũng được xem là một application server, web server. Nó chứa một hệ CSDL RDF suy diễn.
- Catalogue 4.0 file explorer, Soft Experience. Một tiện ích trên Wins nhằm thu thập metadata từ HTML, Adobe XMP, MS Office, Staroffice, Mac, PDF, JPEG/TIFF images, và một số định dạng khác. Nó cũng cho phép phát sinh các báo cáo HTML, XML, hay RDF. Ngoài ra, nó có thể phát sinh RDF hay RDF dạng SVG.
- rdftp server viết bởi Vangelis Vassiliadis. Một PHP server dùng cho RDF, hỗ trợ lưu trữ dữ liệu trên CSDL quan hệ, cho phép truy cập thông qua truy vấn, và cập nhật thao tác thông qua HTTP. Xem online demo.
- RDF Gateway JDBC Driver (RDFJDBC) v0.9.5.1 viết bởi Intellidimension – là một JDBC driver cho phép truy cập RDF Gateway từ Java.
- Semantic Web Server cung cấp bởi Empolis. Một server chạy trên nền J2EE, hỗ trợ lưu trữ & truy vấn mô hình RDF. Nó cũng cho phép thao tác trên Topic Maps & RDF.
- Edurella - Một dự án dựa trên RDF, cung cấp một Cơ sở hạ tầng (infrastructure) hỗ trợ backup, truy vấn, và phân tán cho các hệ thống JXTA peer-to-peer.
- RDFSuite chứa RDF Query Language (RQL) V1.0, RDF Schema Specific DataBase (RSSDB) V1.0, và Validating RDF Parser (VRP) V2.0; được viết bởi Vassilis Christophides và Greg Karvounarakis, Institute of Computer Science.
- Bitzi File Metadata RDF Dump cung cấp bởi Bitzi. Được xem là một nguồn dữ liệu 200.000 files dạng RDF/XML và cho phép xem miễn phí.

- Online RDF-to-XRDF converter tại University of Essen, được chính phủ Đức hỗ trợ phát triển.
- Ngôn ngữ CIM/XML hỗ trợ mô hình hệ thống cao cấp - Một ứng dụng RDF có sử dụng bộ cú pháp con của XML và RDF Schemas.
- CARA Perl RDF API viết bởi Stefan Kokkelink, University of Osnabrueck, Germany - phần mềm miễn phí được phát triển trong dự án CARMEN.
- Xpetal tool cho phép chuyển từ UML sang RDF, viết bởi Arnold deVos - một thành viên của dự án Xmodel. Xpetal tiếp nhận Rational Rose petal model files rồi phát sinh RDF nếu các thuộc tính UML và các role liên quan là ánh xạ tương ứng của các thuộc tính RDF.
- Squish – cho phép lập các truy vấn RDF phức tạp trên Java, viết bởi Libby Miller, ILRT, University of Bristol, UK. Squish là một ngôn ngữ giống SQL, được dùng để thực hiện các truy vấn trên RDF, được viết trên nền Java RDF API, cho phép giao tiếp với CSDL SQL hay mô hình bộ nhớ.
- Universal Learning Format (PDF version) - một tập module các định dạng XML/RDF nhằm mô tả các ‘nội dung học được trực tuyến’ trên các định dạng khác.
- XOTcl RDF, University of Wien, Áo. Hỗ trợ RDF trên ngôn ngữ Extended Object Tcl (XOTCL).
- Dự án Web Resource Application Framework (WRAF) – cho phép xây dựng các ứng dụng - các ứng dụng này hoàn toàn sử dụng mô hình dữ liệu RDF để nhận biết Semantic Web. WRAF được viết bằng ngôn ngữ Perl.
- MusicBrainz Metadata bởi CD Index, một ứng dụng Dublin Core metadata (RDF/XML), hỗ trợ mô tả nội dung đĩa CD như tên đĩa, nghệ sĩ, tên bài hát, ...
- Leveraging UML in RDF bởi Sergey Melnik. Một bản phát thảo các ý tưởng nhằm tạo ra một ‘UML tương thích RDF’ để biểu diễn mô hình quan niệm UML bằng RDF.

- Ontology Interchange Language (OIL) Project - một dự án nhằm phát triển một chuẩn cho phép nối các Ontology lại với nhau. Nó sử dụng các chuẩn Web như RDF, RDF Schemas, XML and XML Schemas.
- RDFPic – một công cụ cho phép nhúng mô tả RDF của một image vào chính nó (ảnh kỹ thuật số).
- Zope Project hỗ trợ RDF, là một phần của dự án Mozilla.
- Non-Governmental Organizations Sector Support Program (NGOSS), Moscow, Nga, dùng RDF để tổ chức site.
- UK Mirror Service một dịch vụ quốc gia của vương quốc Anh, cho phép thu thập và lập bản sao các phần mềm và các nguồn dữ liệu trên thế giới.
- The CERES Thesaurus Effort - CERES (California Environmental Resources Evaluation System) và USGS Biological Resource Division, xây dựng bộ từ điển từ đồng nghĩa kỹ thuật số.
- XWMF - An eXtensible Web Modeling Framework bởi Alexander Block và Reinhold Klapsing, University of Essen, Đức. XWMF sử dụng một mô hình metadata để xác định nội dung và cấu trúc của một web site, và tạo các statements về các thành phần của website bằng RDF. Bộ công cụ này cung cấp RDF parser (cho phép xử lý và truy vấn các mô hình dữ liệu RDF) và trình biên tập đồ họa GramToR (để tương tác với mô hình RDF). Tất cả chúng đều được viết từ các biến thể của ngôn ngữ Tcl và một số thư viện cần thiết khác.
- perl RDF parser / visualiser bởi Eric Prud'Hommeaux ở W3C, một RDF SAX parser và một CSDL các triple được viết bằng Perl.
- RDF Query in Javascript demo bởi Dan Brickley, University of Bristol.
- Một prototype RDF Schema editor bởi Jonas Liljegen - một prototype đang được nghiên cứu thực nghiệm, được viết bằng perl. Nó cho phép xem, chỉnh sửa, và duyệt dữ liệu RDF.
- RDF parser (được viết bằng C) trong XML module tại thư viện của W3C Libwww.

- Cooperative Online Resource Catalog (CORC). CORC là một dự án nhằm chia sẻ metadata giữa các thư viện. Nó dùng RDF/XML, MARC, DC, và Z39.50.
- Dublin Core Metadata Initiative data model, dựa trên mô hình dữ liệu RDF và DC.
- RDF Parser - một giao diện web cho SiRPAC parser, lưu tại UKOLN.
- RDF Browser tại website W3C. Người dùng chỉ cần gửi file RDF đến server, nó sẽ xác nhận cú pháp file và hiển thị file bằng đồ họa.
- Linux Packages Metadata Mirroring Proposal của Daniel Veillard

KHOA CNTT

## **PHỤ LỤC B**

### **MỘT SỐ TOOL VÀ EDITOR**

- Delphi/Kylix Open RDF kit bởi Dieter Köhler. Một tập hợp các thành phần RDF mã nguồn mở để thao tác trên các đồ thị RDF, phân tích cú pháp N-Triples & RDF-aware controls nhằm hiển thị các triple.
- Fenfire Loom - Một trình duyệt RDF bằng đồ họa, được viết bằng Java bởi Benja Fallenstein, một thành viên của Fenfire project.
- 3store persistent RDF knowledgebase bởi Steve Harris, AKT Project, University of Southampton. Một cơ sở tri thức RDF hiệu quả (được viết bằng C), RDQL, và giao diện OKBC. Một công cụ suy diễn nhằm phân loại subclass và subproperty, một công cụ đánh giá độ phức tạp của câu query, và một công cụ thi hành nhanh các truy vấn RDF. Nó dùng MySQL để lưu trữ và cung cấp một Apache module interface.
- IsaViz Visual Authoring Tool for RDF bởi Emmanuel Pietriga (Xerox Research Europe/W3C), thuộc dự án W3C Semantic Web Advanced Development (SWAD) project. Một Java tool dựa trên RDF có sẵn và các hệ thống trực quan nhằm cho phép chỉnh sửa lược đồ RDF bằng cách vẽ & nhập/xuất RDF/XML, N3, và N-Triples.
- Redland RDF Application Framework bởi Dave Beckett, Institute for Learning and Research Technology, University of Bristol - một thư viện C, cung cấp một giao diện RDF cao cấp nhằm tạo lược đồ RDF từ XML, lưu trữ, truy vấn, và thao tác trên chúng. Redland bổ sung các khái niệm RDF vào các class của chúng thông qua API dựa trên đối tượng, rồi ánh xạ thành các ngôn ngữ API khác như Java, Perl, PHP, Python, Ruby and Tcl.
- Raptor RDF Parser Toolkit – một thư viện các RDF parsers - RDF/XML và N-Triples, được thiết kế để tích hợp với Redland.

- ICS-FORTH Validating RDF Parser (VRP) - một RDF parser được viết bằng Java bởi Alexaki Sophia và Karsten Tolle. VRP cho phép chứng thực các câu RDF dựa trên RDF Schema và XML Schema Datatype, nó còn hỗ trợ và quản lý bản cấu trúc nháp về 2003-01-23 RDF/XML.
- Drive RDF Parser for C# bởi Rahul Singh hỗ trợ bản RDF/XML Syntax draft of 23 January 2003.
- Joseki RDF server bởi Andy Seaborne, HP Labs, Bristol. Một ứng dụng nhỏ của bộ Jena RDF framework, nó cho phép server xuất (publish), cập nhật, và truy vấn các mô hình RDF trên web bằng cách dùng HTTP & RDQL.
- DR2 MAP, một ứng dụng Java cho phép xuất RDF từ các CSDL quan hệ, viết bởi Chris Bizer. Nó dùng bất kì RDBMS nào có hỗ trợ JDBC hay ODBC và cho phép người dùng mô tả sự ánh xạ này bằng ngôn ngữ dựa trên XML.
- RDF API for PHP bởi Chris Bizer. Một PHP package thuần túy nhằm thao tác trên các mô hình RDF và phân tích cú pháp RDF/XML.
- Ideagraph bởi Danny Ayers. Một công cụ trực quan, viết bằng Java, cho phép tạo các ánh xạ về tri thức cá nhân (dùng RDF).
- RDF Model Browser bởi Dima Skvortsov. Một ứng dụng web (Servlets 2.2) cho phép duyệt dữ liệu RDF bằng tree view.
- BrownSauce RDF browser bởi Damian Steer, HP Labs. Một ứng dụng hướng server (servlet), cho phép duyệt dữ liệu RDF thông qua giao diện web (HTML và CSS).
- Python RDFLib bởi Dan Krech. Một thư viện RDF, cung cấp một API để thao tác trên các lược đồ RDF, phân tích cú pháp RDF, và lưu trữ dữ liệu dạng triple.
- Semantic Markup, Ontology and RDF Editor (SMORE) bởi Aditya Kalyanpur, MIND-SWAP group, University of Maryland, College Park, USA. Một công cụ cho phép đánh dấu (markup) các tài liệu ở dạng RDF/XML bằng cách dùng các Ontology và các thuật ngữ mà người dùng định nghĩa.

- KARlsruhe ONtology and Semantic Web infrastructure (KAON) Một Java semantic web infrastructure hướng đến các ứng dụng kinh doanh. Nó bao gồm các công cụ để tạo Ontology, cho phép xây dựng các ứng dụng có hỗ trợ các loại CSDL hiện có.
- Tucana KnowledgeStore (TKS) bởi Plugged In Software, Brisbane, Áo - Một CSDL đồ họa, an toàn, phân tán, và khả mở, được tối ưu hóa nhằm quản lý metadata. Nó có thể quản lý các RDF statements từ 40 triệu (32 bit) – 95 triệu triples (64 bit).
- Inkling RDF query using SquishQL bởi Libby Miller, ILRT. Sử dụng ngôn ngữ truy vấn RDF SquishQL, được thực hiện trên nền Java. Nó dùng JDBC API để thao tác CSDL SQL
- RDF Editor (Rensselaer Polytechnic Institute) bởi Orr Bernstein, Mohit Oberoi và John Punin. Một trình biên tập RDF trên nền Java Swing, cho phép tạo RDF/XML hay N-Triples.
- TAP Software Kit bởi R.V. Guha và Rob McCool, TAP Project. Gồm các thư viện được viết bằng Perl & Java, cũng như một TAPache server nhằm sử dụng và cung cấp GetData Interface.
- FOAFBot: IRC Community Support Agent bởi Edd Dumbill. FOAFBot là một IRC bot (agent) - cung cấp khả năng truy cập cơ sở tri thức (cơ sở tri thức này được tạo bằng cách duyệt các FOAF file).
- FOAF-a-matic bởi Leigh Dodds. Một client JavaScript application cho phép tạo mô tả FOAF ("Friend-of-A-Friend") bằng RDF.
- RDQL query language for PHP bởi Garland Foster. Một RDQL engine nhằm truy vấn RDF từ các nguồn dữ liệu khác nhau từ các file trên web, ...
- SMORE: Semantic Markup, Ontology and RDF Editor bởi Aditya Kalyanpur, MIND-SWAP group, ĐH Maryland. Một công cụ Java, cho phép đánh dấu (markup) các trang web hay text ở dạng RDF bằng cách dùng các Ontology & các

thuật ngữ người dùng. Nó cũng chứa một HTML editor and previewer, một ontology browser và một semantic data tree viewer.

- EEP3 Python Semantic Web API của Sean B. Palmer. Một API (viết bằng python), chứa một hệ thống truy vấn, lưu trữ, phân tích triple; một hệ suy diễn CWMclone; một N3 pre-processor; một IRC bot (agent); một Notation3 parser; và một rdfdiff program.
- RDF Web Scraper của Aditya Kalyanpur, ĐH Maryland, cho phép ánh xạ thành RDF markup. Xem bản demo và Zip Download.
- Versa RDF query language bởi Mike Olson và Uche Ogbuji, Fourthought. Một ngôn ngữ truy vấn dựa trên lược đồ RDF.
- TRIPLE RDF-based logic programming language bởi Stefan Decker và Michael Sintek. Một hệ thống chuyển đổi, suy diễn, và truy vấn RDF. Nó cho phép định nghĩa, sử dụng, và phân loại các ngôn ngữ semantic web như RDFS, DAML+OIL, UML, Topic Maps.
- Wilbur RDF Toolkit dùng cho RDF và DAML (và XML) bởi Agent Technology group, Nokia Research Center.
- Profium SIR extractor demonstration: automated metadata extraction. Có thể rút trích RDF từ các định dạng như Adobe XMP documents, Microsoft Word, Excel, PowerPoint.
- OntoEdit 2.0 ontology editor bởi ontoprise. Một môi trường phát triển cho các mô hình tri thức, quản lý các khái niệm và mối liên hệ, xuất/nhập RDF, hỗ trợ xây dựng trên DAML+OIL, và hỗ trợ đa nhãn (label).
- RDFAuthor: a graphical RDF instance data authoring tool dùng cho Mac OSX, bởi Damian Steer. Với mong muốn tạo ra RDF một cách dễ dàng, và cho phép xuất dữ liệu dạng SVG. Nó được viết bằng Java, có dùng Jena.
- RDFStore RDQL Demonstration bởi Alberto Reggiori.



- SWIPT: An RDF Parser/Tool in Python bởi Sean B. Palmer. Một công cụ sử dụng các chương trình Python nhằm đọc/ghi các cú pháp để tạo RDF graphs.
- Blindfold Semantic Web Toolkit V0.1.1 bởi Sandro Hawke - Một bộ công cụ, chứa một hệ thống ngữ pháp nhằm đọc/ghi hầu hết các định dạng và ngôn ngữ như thể chúng là RDF/XML.
- RDF Data Query Language (RDQL), bởi Andy Seaborne, HP Labs. Một ngôn ngữ truy vấn RDF giống SQL, sử dụng hệ thống RDF Java của Jena.
- RDFStore Perl API for RDF Storage bởi Alberto Reggiori. Một package nhằm quản lý các CSDL dạng RDF. Xem demo và work tại perl rdf api.
- FRODO RDFSviz của Michael Sintek, FRODO team, Đức - Một dịch vụ trực quan dùng RDF Schemas để hiển thị class, sub-class, và property relationships.
- DAML Validator bởi Dave Rager, kiểm tra cú pháp & lỗi ngữ nghĩa của DAML markup.
- Redfoot framework for building distributed data-driven web applications with RDF and Python bởi James Tauber và Dan Krech. Cung cấp một CSDL RDF, API truy vấn, ngôn ngữ mẫu, kiến trúc module, editor thông qua giao diện web.
- W3C RDF Validation Service bởi Art Barstow, W3C/HP - Một dịch vụ cho phép phân tích cú pháp RDF/XML, tạo các image của các RDF graphs, và hiển thị kết quả ở dạng triples.
- HyperDAML DAML or RDF/XML schemas to HTML viewer bởi Mike Dean.
- ARP: Another RDF Parser bởi Jeremy Carroll, HP Labs - Một trình phân tích cú pháp RDF/XML, được thiết kế để tích hợp với Jena.
- Euler proof mechanism / RDF logic demonstrator, bởi Jos De Roo, AGFA. Một hệ thống java cho phép phát sinh một proof đối với một câu hỏi về các facts & rules.
- Ontomat DAML annotation tool bởi Siegfried Handschuh.

- Extensible Open RDF Toolkit (EOR) bởi OC LC Office of Research và The Dublin Core Metadata Initiative - Một hệ thống tìm kiếm và quản lý metadata RDF được viết bằng Java.

Fydra - Decentralised and reliable resource discovery using RDF metadata bởi Johan Nagels, CoWorks - Một Java servlet, ‘crawl’ web, rồi lưu file metadata kết quả ở dạng RDF.

RDFedt - RDF Editor dùng cho Windows bởi Jan Winkler. Một trình biên tập các RDF/XML graphs; quản lý namespaces, schemas, và terms.

XSB tabled logic programming and deductive database system dùng cho Unix và Windows, version 2.3 cung cấp một libwww-based RDF parser để thêm các RDF statements.

Representing vCard Objects in RDF/XML, Renato Iannella, IPR Systems.

RDF-Crawler Java-based tool/API, cho phép downloads các phân đoạn (fragment) của RDF từ Internet, rồi xây dựng một knowledgebase dựa trên đó.

Uniform Web Index Maker Program (UWIMP) bởi Sean B. Palmer - Một công cụ đơn giản nhằm thêm RDF vào các file XHTML.

XSLT RDF Extractor for XHTML Files bởi Sean B. Palmer - Một công cụ chuyển XHTML thành RDF (sử dụng XSLT), cho phép lấy dữ liệu RDF từ các trang XHTML.

XSLT RDF Parser bởi Jason Diamond. Hỗ trợ containers, rdf:value, và rdf:parseType.

RDF Parsers (Perl and Java) bởi Profium. Theo kiểu Stanford Java API for RDF (Sergey Melnik).

SVG Linearizer tools - Một công cụ nhằm chuyển SVG thành text (bằng java).

RDFDB: An RDF Database bởi R.V. Guha. Một hệ thống cho phép phân tích cú pháp XML, lưu trữ, lập mô hình. Nó cung cấp một ngôn ngữ truy vấn và một giao diện server.

Dublin Core Extraction Service, Dan Connolly - Một dịch vụ web, sử dụng XSLT và dữ liệu XML để tạo RDF.

Rudolf: RDFViz - Exploring tools for RDF Graph Visualisation, Dan Brickley.  
Một giao diện web nhằm phát sinh một cách trực quan RDF từ bitmap images, VRML, và SVG. Nó dùng bộ công cụ AT&T GraphViz.

DC-Dot Một trình biên tập của Dublin Core, có thể xuất dữ liệu ở dạng RDF và HTML.

KHOA CNTT

## PHỤ LỤC C

### CÁC SEMANTIC SEARCH ENGINE HIỆN CÓ

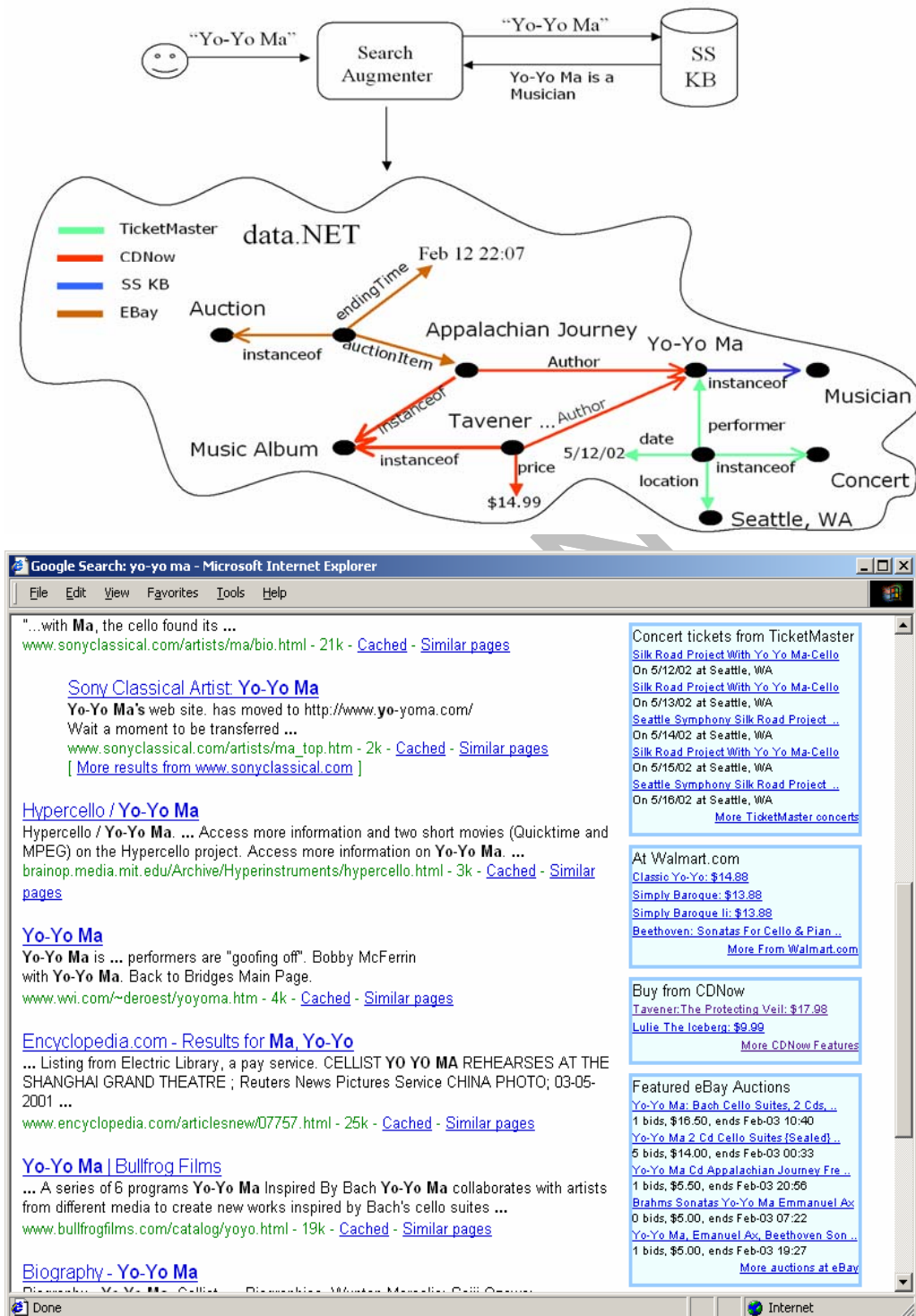
#### 🚩 ABS (Activity Based Search) tại ĐH Stanford

Search Engine này áp dụng công nghệ Semantic Web để thêm các kết quả Semantic vào kết quả của Search Engine truyền thống (cụ thể là Google).

Mục đích của Search Engine này là hiểu được việc: các sự kiện khác nhau của cùng một chuỗi có thể có những ý nghĩa khác nhau. Nó phải lọc, sắp xếp, và hiển thị kết quả tương ứng với ý nghĩa được chọn. Hiện nay, Search Engine này chỉ tập trung vào lĩnh vực con người.

Nguồn dữ liệu rất quan trọng của ABS là Tap Knowledge Base (TAP KB) – dù còn nông (chưa chuyên sâu), nhưng cũng đã bao hàm được nhiều lĩnh vực như *con người* (nhạc sĩ, ca sĩ, vận động viên thể thao, diễn viên, nhà chính trị), *các tổ chức* (công ty, nhóm nhạc, đội thể thao), *nơi chốn* (thành phố, bang, quốc gia), và các *sản phẩm*. Đối với mỗi tài nguyên, hệ thống sẽ cung cấp một *rdf:type* và *rdfs:label* cho các object. Tap Knowledge Base cung cấp cho ABS khoảng 65.000 ‘con người’, ‘tổ chức’, ‘nơi chốn’.

Ví dụ, tìm kiếm Yo-Yo Ma, ngoài kết quả từ Search Engine truyền thống, hệ thống còn cung cấp các thông tin về album, hình ảnh, lịch biểu diễn, ... của ông ta.



Hình 25. Minh họa ABS

Kết quả bên trái lấy từ Google. Các bảng thông tin bên phải (nếu có) có từ Semantic Web, chúng được tìm ra dựa vào ngữ nghĩa (denotation) của các thuật ngữ tìm kiếm (search term). Các denotation được hỗ trợ bởi TAP KB.

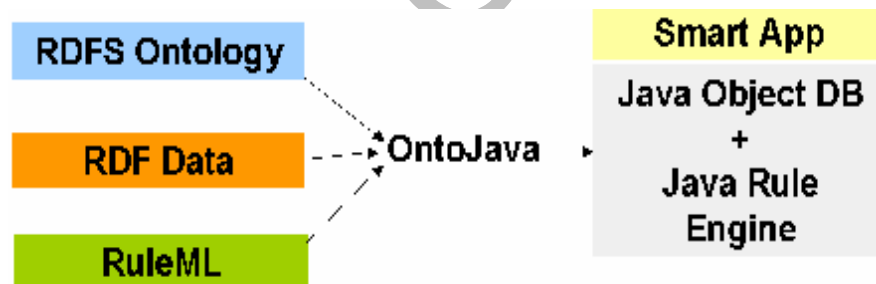
Có thể xem chi tiết và sử dụng Search Engine này tại <http://tap.stanford.edu/ss/> & <http://tap.stanford.edu:8000/wsi-bin/flek.wsp/tap>

### ✚ SPICE

SPICE là một công cụ dựa trên web, nó được xây dựng dựa trên một số bộ công cụ (Protégé, OntoJava, Jakarta Tomcat Apache Server) để cung cấp khả năng tìm kiếm cho LSDIS Library website.

Protégé được dùng để xây dựng knowledge base.

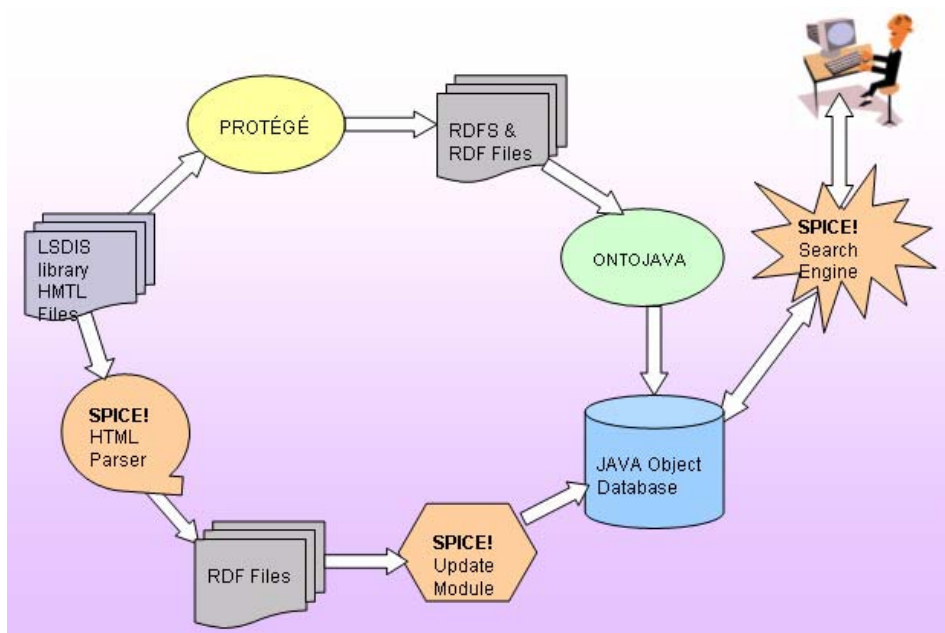
OntoJava là một trình biên dịch, nó tự động ánh xạ các RDF fact, bộ RDFS Ontology, các rule, và các RuleML vào một CSDL bộ nhớ chính (viết bằng Java).



Hình 26. Nguyên tắc hoạt động của SPICE

Tomcat hỗ trợ Java Servlet và kỹ thuật JSP.

Kiến trúc hệ thống:



Hình 27. Kiến trúc của SPICE

Giao diện:

|  |   |
|--|---|
| <b>Author(s)</b><br><small>(seperated by commas)</small> | <input type="text"/> or <input type="radio"/> and |
| <b>Title</b>   | <input type="text"/>                              |
| <b>Keywords</b>  | <input type="text"/>                              |
| <b>Year Published</b>                                    | <input type="text"/>                              |
| <b>Media Type</b>  | All<br>Article<br>Book                            |
| <b>Projects</b>  | All<br>CaTCH<br>InfoHarness                       |
| <input type="button" value="Search"/>                    | <input type="button" value="Reset"/>              |

Hình 28. Giao diện của SPICE

Một số vấn đề còn tồn tại:

- Không hỗ trợ namespace.
- Không hỗ trợ đa thừa kế
- SPICE chỉ được cấu trúc để làm việc với các trang web của LSDIS Library.

Hiện chưa có thông tin URL của SSE này.

#### ✚ Mô hình SSE của Madhan R Arumugam

Input: các keyword (thường là các danh từ), output: các Ontology thích hợp.

Nguyên tắc hoạt động:

- Lấy mỗi keyword rồi tìm chúng trong các subject, object, predicate của Ontology space.
- Lấy tên các Ontology thỏa điều kiện trên và các thông tin chi tiết.
- Nếu mỗi keyword cho kết quả là một tập hợp gồm nhiều Ontology, thì phải xác định tập Ontology tối thiểu. Điều này nhằm loại bỏ các Ontology không cùng ngữ cảnh với tập các từ khóa.

Vì dụ người dùng yêu cầu:

*“Find all **earthquakes** with **epicenter** in a 5000 mile **radius** of the **location** at **latitude** 60.790 North and **longitude** 97.570 East and find all **tsunamis** that they might have caused.”*

Kết quả khi chưa kiểm tra ngữ cảnh:

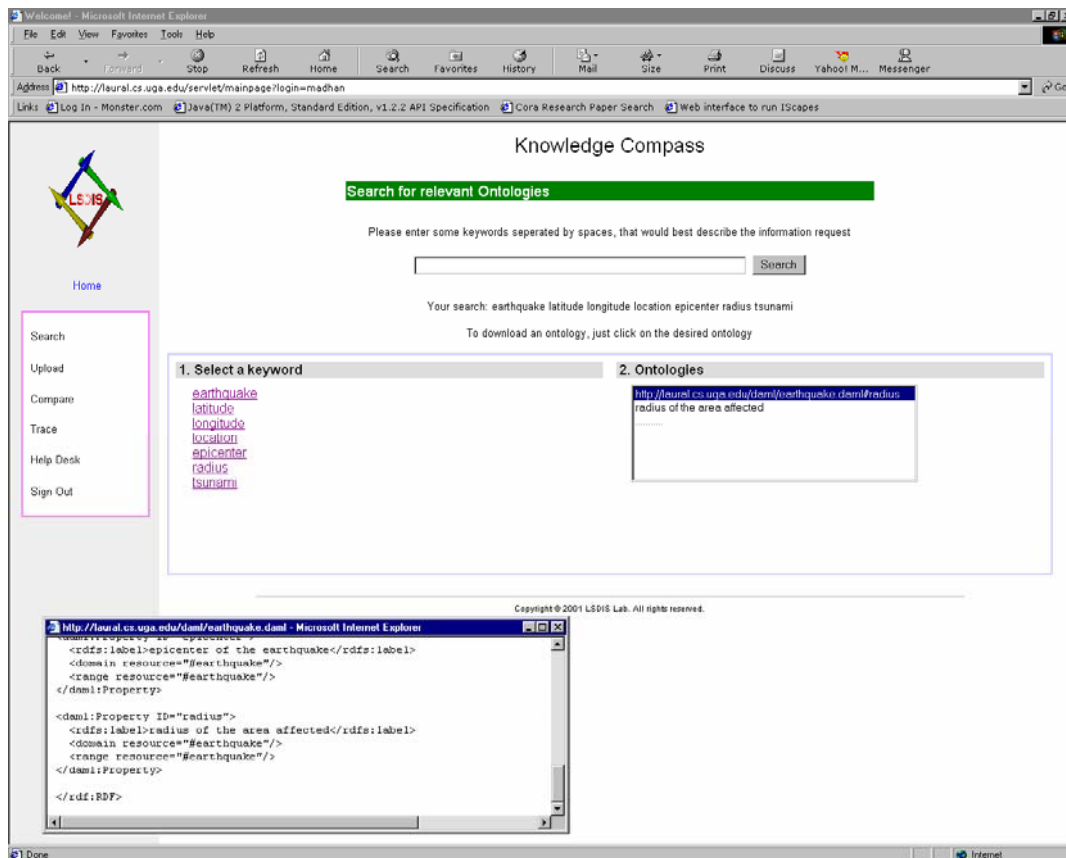
| Keyword                       | Ontologies                                   |
|-------------------------------|--|
| Earthquake                    | earthquake.daml, damage.daml                 |
| latitude, longitude, location | location.daml, weather.daml, earthquake.daml |
| Epicenter, radius             | earthquake.daml, circle.daml                 |
| Tsunami                       | tsunami.daml                                 |

Kết quả sau khi kiểm tra ngữ cảnh:

| Keyword | Ontologies |
|---------|------------|
|---------|------------|



|                               |                                |
|-------------------------------|--------------------------------|
| Earthquake                    | earthquake.daml                |
| latitude, longitude, location | location.daml, earthquake.daml |
| Epicenter, radius             | earthquake.daml                |
| Tsunami                       | tsunami.daml                   |



Hình 29. Giao diện SSE của Madhan R Arumugam

Hiện chưa có thông tin URL của SSE này.

### i-Toolbox

Cho phép gắn SSE vào ứng dụng.

Đặc điểm: tự động lấy nghĩa của các từ T.Anh, T.Pháp; xử lý ngôn ngữ tự nhiên; tự động lọc kết quả tìm kiếm; tăng phạm vi truy vấn; tự động phát sinh các link

‘see also’; có từ điển semantic 400.000 từ; cho phép thêm ngôn ngữ mới, thêm lĩnh vực (domain) mới.

Ưu điểm: truy cập thông tin chính xác, đơn giản; được tổ chức sao cho truy cập được nhiều thông tin nhất, với bất kì ngôn ngữ và định dạng nào.

Chi tiết kĩ thuật: kiến trúc client-server; CGI-BIN, ASP, ISAPI, DBI (perl), PHP, ODBC, Delphi, Lotus Domino, C++, C, JDBC interfaces; nguồn tài liệu đa dạng: XML, SGML, Word, HTML, PDF, www, ODBC sources (Access, Oracle, Sybase, SQL Server, My Sql, PostGreSQL...); môi trường Server: NT, Windows 2000, Sun, Linux, FreeBSD.

Xem thêm thông tin tại <http://www.sinequa.com/html-uk/i-toolbox-en.html>