

Học viện công nghệ bưu chính viễn thông
Khoa công nghệ thông tin

Tiểu luận Agent

Ứng dụng bookstore trong semantic web

Giáo viên hướng dẫn : Trần Đình Quế

Sinh viên : Đới Thanh Thịnh

Lớp : D07CNPM1

Mục Lục :

| | |
|--|-----------|
| I. Giới Thiệu | 3 |
| II. Web ngữ nghĩa..... | 4 |
| 1. Tổng quan về web ngữ nghĩa | 4 |
| 2. Sự ra đời của semantic web | 4 |
| 3. Web ngữ nghĩa là gì? | 5 |
| 4. Kiến trúc web ngữ nghĩa..... | 9 |
| 5. Vai trò các lớp trong kiến trúc Semantic Web..... | 10 |
| 5.1 URI (Uniform Resource Identifier) | 10 |
| 5.2 RDFS (RDF schema) | 12 |
| 5.3 Ontology Vocabulary..... | 12 |
| 5.4 Tầng Logic | 13 |
| 5.5 Tầng Proof..... | 13 |
| 5.6 Tầng Trust..... | 13 |
| 6. Nội dung xây dựng Web ngữ nghĩa..... | 14 |
| 6.1 Chuẩn hoá các ngôn ngữ biểu diễn dữ liệu (XML) và các siêu dữ liệu (RDF) trên Web..... | 14 |
| 6.2 Chuẩn hoá các ngôn ngữ biểu diễn Ontology cho Web có ngữ nghĩa..... | 15 |
| 6.3 Phát triển nâng cao Web ngữ nghĩa (Semantic Web Advanced Development - SWAD) | 15 |
| 7. Tổng kết chương..... | 16 |
| 1. Khái niệm về ontology..... | 16 |
| 2. Các phần tử trong ontology..... | 18 |
| 3. Ngôn ngữ OWL DL..... | 20 |
| 4. Tổng kết chương..... | 23 |
| IV. Ứng dụng bookstore với semantic web | 23 |
| 1. Dữ liệu bookstore của cơ sở dữ liệu A | 23 |
| 2. Dữ liệu bookstore của cơ sở dữ liệu F..... | 24 |
| 3. Bắt đầu việc trộn dữ liệu | 25 |
| 4. Trộn dữ liệu giữa cơ sở dữ liệu A và F | 26 |
| 5. Hoàn thiện | 27 |
| V. Tài liệu tham khảo..... | 28 |

I. Giới Thiệu

Bạn quyết định đi xem một bộ phim và muốn ăn một thứ gì đó sau khi xem. Trong trường hợp này bạn sẽ khởi động PC , mở trình duyệt web và truy cập vào Google để tìm kiếm các thông tin về rạp hát , rạp chiếu phim và quán ăn.

Tiếp đó bạn cần biết bộ phim nào sẽ được trình chiếu trong các rạp chiếu ở gần bạn, khi đó bạn sẽ tốn khá nhiều thời gian cho việc đọc những chỉ dẫn vắn tắt cho mỗi bộ phim trước khi thực hiện sự lựa chọn của mình. Cũng thêm vào đó, bạn lại muốn xem các quán ăn ngon nào gần với rạp chiếu nhất. Do đó bạn sẽ muốn kiểm tra các bản đánh giá của khách hàng về các quán ăn. Tóm lại, bạn sẽ phải ghé thăm đến hàng nửa tá website trước khi ra khỏi cửa nhà.

Một số chuyên gia Internet tin tưởng rằng thế hệ web kế tiếp – web 3.0 sẽ thực hiện các nhiệm vụ giống như việc bạn tìm kiếm các bộ phim, các món ăn tại các quán gần đó được nhanh hơn và dễ dàng hơn. Thay vì phải thực hiện nhiều tìm kiếm, bạn chỉ cần đánh vào đó một hoặc hai câu phức tạp hơn trong trình duyệt web 3.0, còn lại web sẽ thực hiện toàn bộ các công việc còn lại cho bạn. Cho ví dụ, bạn có thể đánh “Tôi muốn xem một bộ phim hay và sau đó đi ăn tại một nhà hàng ăn ngon. Lựa chọn của tôi là gì?”. Trình duyệt web 3.0 sẽ phân tích sự đáp trả của bạn, tìm kiếm trên Internet những câu trả lời có thể và sau đó trả về kết quả cho bạn.

Tuy nhiên đó không phải là tất cả. Các chuyên gia còn tin tưởng rằng trình duyệt web 3.0 còn có thể thực hiện những công việc giống như một người trợ lý. Khi bạn tìm kiếm trên web, trình duyệt sẽ biết được những gì bạn quan tâm. Càng sử dụng web, trình duyệt càng biết nhiều về bạn và càng ngày bạn cần phải đưa ra các câu hỏi hơn. Cuối cùng là bạn có thể hỏi trình duyệt mở các câu hỏi giống như “Tôi nên đi đâu cho bữa trưa?”. Khi đó trình duyệt sẽ cố vấn cho bạn những gì bạn thích và không thích, lấy địa điểm bạn đang ở và sau đó gợi ý một danh sách các nhà hàng.

Khái niệm “Semantic Web” 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 “

Với sự lớn mạnh và khả năng lưu trữ thông tin ngữ nghĩa , Semantic Web sẽ trở thành một thể hệ web cho tương lai , thể hệ web mà chúng ta đang chờ đợi . Và đó cũng là lí do mà em chọn đề tài “Ứng dụng semantic web trong bookstore “ trong bài tiểu luận này .

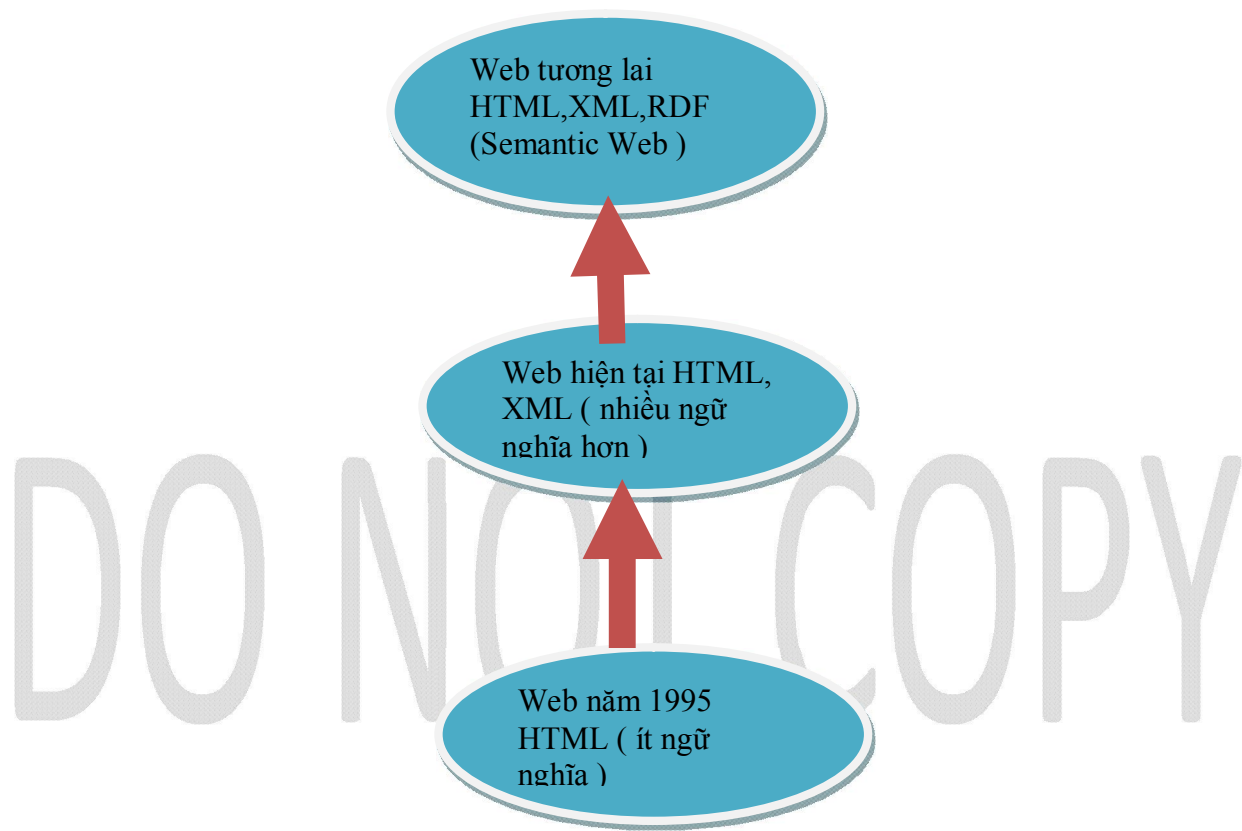
II. Web ngữ nghĩa

1. Tổng quan về web ngữ nghĩa

Web ngữ nghĩa là một thể hệ Web mới, đang được phát triển và sẽ xuất hiện trong tương lai gần. Hệ thống Web mới này sẽ thay thế hệ thống Web hiện tại song không có nghĩa là một hệ thống hoàn toàn khác hệ thống Web hiện tại. Web ngữ nghĩa được phát triển trên hệ thống Web hiện tại bằng cách bổ sung thêm ngữ nghĩa cho các tài nguyên Web mà máy có thể hiểu và tăng khả năng xử lý tự động.

2. Sự ra đời của semantic web

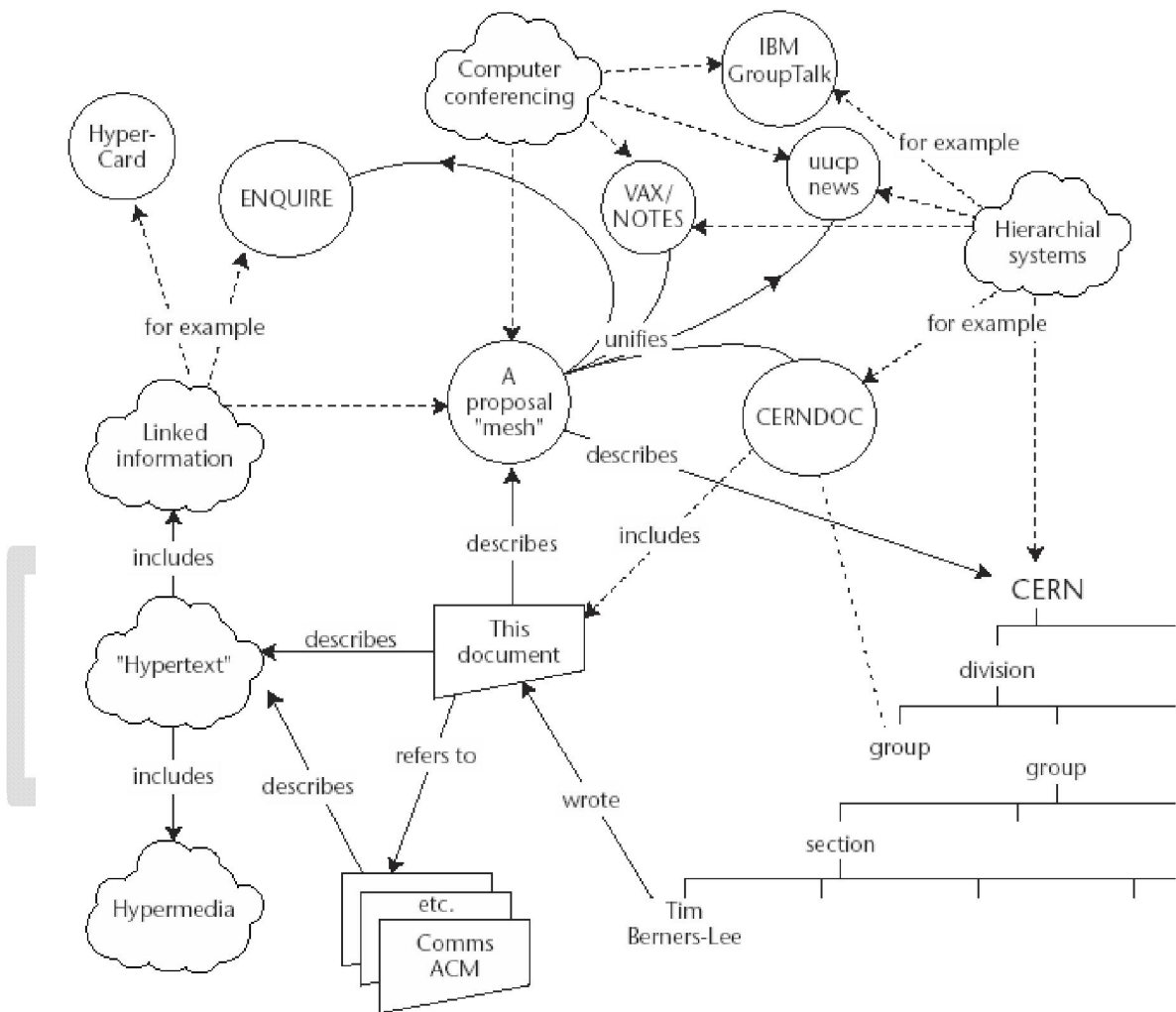
Thế hệ web đầu tiên bắt đầu với những trang HTML thủ công , thế hệ thứ hai đã tạo nên một bước cho máy thực hiện và thường là các trang HTML động . Các thế hệ web này mang ý nghĩa cho con người thao tác trực tiếp (đọc , duyệt , điền vào mẫu) . Thế hệ web thứ ba được gọi là “Semantic Web” , mang mục đích là thông tin se 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” . 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 ...Những dịch vụ thông minh trên hệ thống web giàu ngữ nghĩa sẽ như thế chắc hẳn sẽ vượt trội hơn những bản sẵn có hiện tại của các dịch vụ này , mà chỉ giới hạn về chắc năng .



3. Web ngữ nghĩa là gì?

Tim Berners-Lee đã đưa ra hai vấn đề của web ngữ nghĩa, đó là tạo cho Web một môi trường cộng tác tốt hơn và vấn đề thứ hai là máy có thể hiểu và xử lý tự động các thông tin trên Web. Hình 1.1. biểu diễn cái nhìn bản chất nguồn gốc Web của Tim Berners-Lee. Cái nhìn này rất phức tạp và phức tạp hơn nhiều so với việc lọc ra các trang HTML từ các dịch vụ Web hiện tại. Nhưng trong hình này, có các mối quan hệ giữa các thông tin như: "includes", "describes", "rote", đây là các mối quan hệ giữa các tài nguyên mà trong web hiện tại không có. Và chính các quan hệ này đưa vào web ngữ nghĩa là cần thiết, để máy có thể hiểu và xử lý thông tin tự

động trên web. Các quan hệ này được gọi là các siêu dữ liệu. Công nghệ để có được các siêu dữ liệu này là công nghệ RDF (Resource Description Framework)

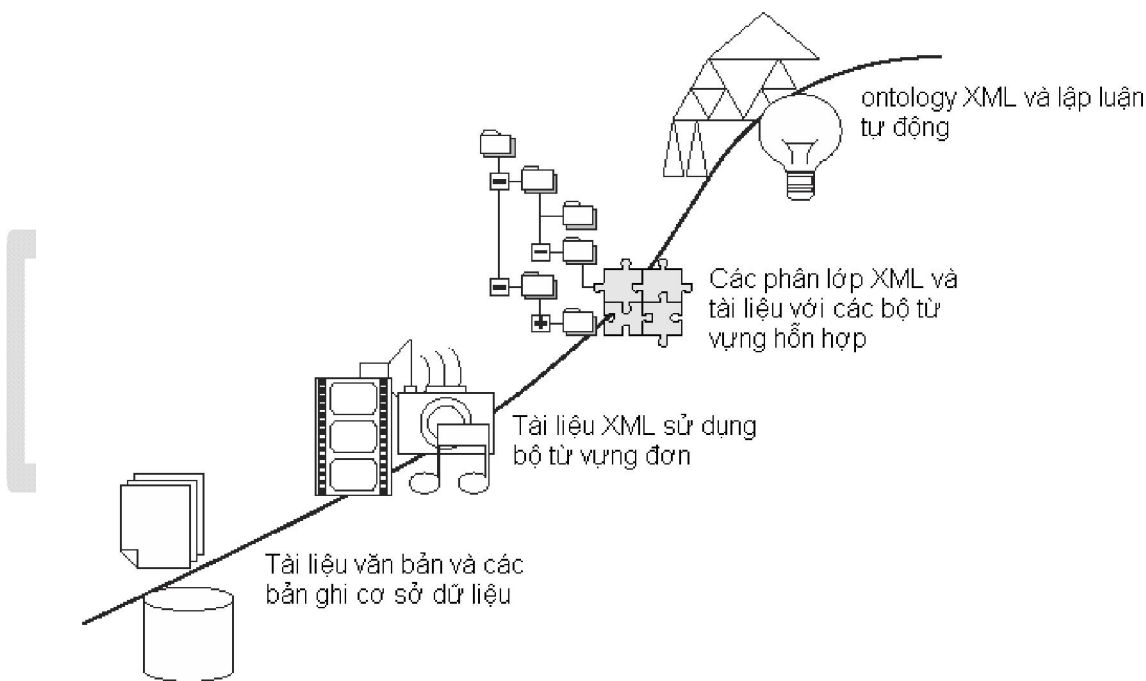


Hình 1.1. Một đề xuất nguồn gốc Web với CERN

(CERN-Hội đồng Châu Âu nghiên cứu hạt nhân)

Để có thể tạo ra web có dữ liệu mà máy có thể xử lý được, trước hết phải thay đổi mô hình trong cách chúng ta nghĩ về dữ liệu. Từ trước đến nay, dữ liệu bị khoá ngay trong các ứng dụng độc quyền. Dữ liệu được coi như thứ yếu để xử lý dữ liệu. Thái độ không đúng này bộc lộ những sai sót cơ bản trong luận cứ gốc bằng việc thiết lập sự phụ thuộc giữa xử lý và dữ liệu. Hay nói cách khác, phần mềm tốt thì hoàn toàn phụ thuộc vào dữ liệu tốt. Với các hệ tính toán chuyên

nghiệp thì ngay khi bắt đầu thực hiện, vấn đề dữ liệu là rất quan trọng, nó phải được xác minh và bảo vệ. Với Web, XML và xu thế Web ngữ nghĩa đang nổi lên thì có sự thay đổi của sức mạnh đang di chuyển từ các ứng dụng sang dữ liệu. Vấn đề này cũng cho chúng ta mấu chốt để hiểu Web ngữ nghĩa. Con đường để máy có thể xử lý dữ liệu chủ yếu là tạo ra dữ liệu thông minh hơn chứ không phải là các xử lý thông minh. Hình 1.2 biểu diễn sự phát triển tính thông minh của dữ liệu theo thời gian.



Hình 1.2. Sơ đồ phát triển tính thông minh của dữ liệu

Hình 1.2 chỉ ra sự phát triển liên tục của tính thông minh dữ liệu có bốn bậc quan trọng. Bốn bậc này biểu diễn từ dữ liệu có tính thông minh thấp nhất đến dữ liệu đã có thông tin đủ ngữ nghĩa để máy thực hiện suy luận về nó.

Văn bản và cơ sở dữ liệu (bậc này là tiền XML)

Bậc đầu tiên, hầu hết dữ liệu ở bậc này được sở hữu độc quyền cho một ứng dụng. Do vậy tính thông minh là nằm trong ứng dụng đó chứ không phải là trong dữ liệu.

Tài liệu XML với miền đơn

Ở bậc này, dữ liệu đạt được là: ứng dụng độc lập với miền riêng. Bây giờ, dữ liệu đủ thông minh để có thể di chuyển giữa các ứng dụng trong một miền đơn. Ví dụ: các chuẩn XML trong công nghiệp chăm sóc sức khỏe, công nghiệp bảo hiểm, hoặc công nghiệp bất động sản.

Các phân loại và tài liệu với các bộ từ vựng hỗn hợp

Trong bậc này, dữ liệu có thể được soạn từ đa miền và được phân lớp chính xác theo một nguyên tắc phân loại có thứ bậc. Thực tế, sự phân lớp đó có thể được sử dụng để khám phá dữ liệu. Các quan hệ đơn giản giữ các lớp có thể được sử dụng để quan hệ và do vậy, ta có dữ liệu kết hợp. Như vậy, dữ liệu bây giờ thông minh đủ để khám phá dễ dàng và được kết hợp một cách hợp lý với dữ liệu khác.

Các ontology và các luật lập luận

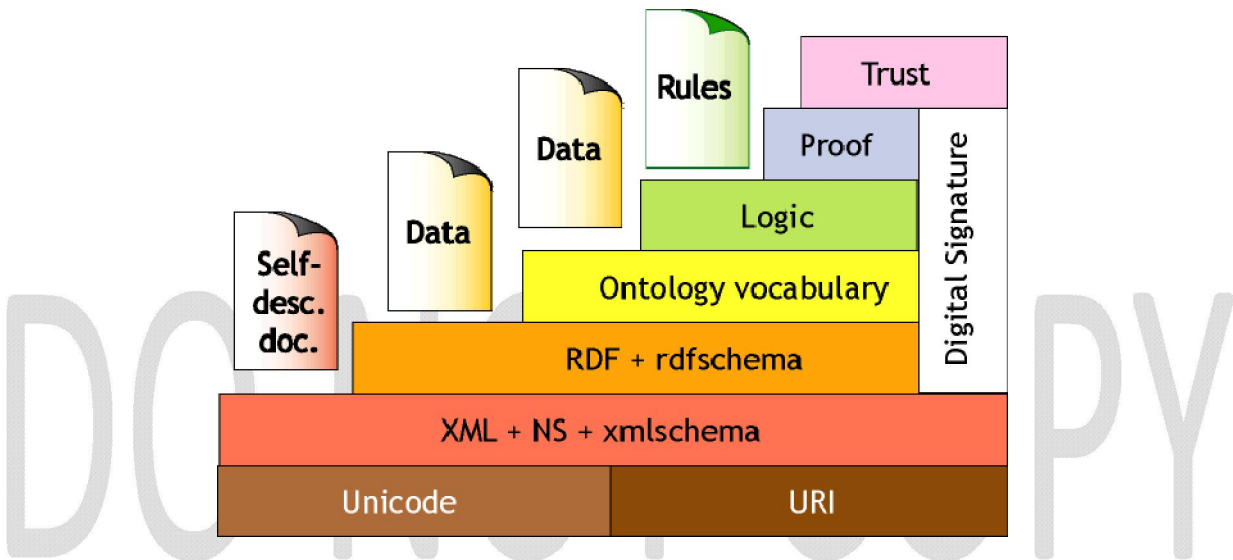
Trong mức này, dữ liệu mới có thể được suy ra từ những dữ liệu đã có theo các luật logic. Thực chất, dữ liệu bây giờ thông minh đủ để được mô tả với các quan hệ cụ thể và các dạng hình thức phức tạp. Điều này cho phép kết hợp hoặc tái kết hợp dữ liệu nguyên thủy nhiều hơn và sự phân tích về dữ liệu mịn hơn. Một ví dụ về sự phức tạp của dữ liệu là việc chuyển tự động tài liệu trong một miền này sang một tài liệu tương đương trong một miền khác. Ở mức này, ta có thể soạn một định nghĩa mới trên Web ngữ nghĩa, máy có thể xử lý các tài liệu web có dữ liệu thông minh. Hơn nữa, ta có thể định nghĩa dữ liệu thông minh vì dữ liệu nó độc lập với ứng dụng, có thể soạn, phân lớp và lấy một phần của các hệ thống tin lớn hơn.

Như vậy, việc xây dựng web ngữ nghĩa chủ yếu tập trung xây dựng hệ thống web có dữ liệu thông minh. Dữ liệu thông minh ở đây không có nghĩa là gắn liền với trí tuệ nhân tạo, không phải là cho máy học hiểu các từ ngữ của con người hay máy xử lý được ngôn ngữ tự nhiên mà nó thông minh bởi việc tổ chức lựa chọn dữ

liệu biểu diễn trong ngôn ngữ được cấu trúc để máy có thể hiểu. Và sau đây chúng ta tìm hiểu về kiến trúc của web ngữ nghĩa.

4. Kiến trúc web ngữ nghĩa

Web ngữ nghĩa được xây dựng trên nền hệ thống web hiện tại. Web ngữ nghĩa được coi là sự mở rộng của Web hiện tại có bổ sung thêm ngữ nghĩa vào dữ liệu trên web. Hình 1.3 chỉ ra sơ đồ kiến trúc của Web ngữ nghĩa.



Hình 1.3. Kiến trúc của web ngữ nghĩa

Từ sơ đồ kiến trúc của Web ngữ nghĩa, ta thấy có bảy tầng kiến trúc. Trong đó, với hệ thống Web hiện tại (World Wide Web) là đang ở tầng thứ hai.

Lớp Unicode , URI : nhằm đảm bảo sử dụng tập kí tự quốc tế và cung cấp phương tiện để định danh các đối tượng trong semantic web .

Lớp XML : cùng với các định nghĩa về namespace , schema , đảm bảo rằng chúng ta có thể tích hợp các định nghĩa semantic web với các chuẩn dựa trên XML khác .

Lớp RDF , RDFS : Có thể tạo ra các phát biểu để mô tả các đối tượng với với những từ vựng và định nghĩa URI , các đối tượng này có thể tham chiếu đến bởi những từ vựng và định nghĩa URI ở trên . Đây là lớp mà chúng ta có thể gán các

kiểu 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 từ vựng khác nhau .

Lớp Digital Signature : được dùng để xác định chủ thể của tài liệu (vd: tác giả của tài liệu)

Các lớp Logic , Proof , Trust : Lớp logic cho phép viết ra các luật , trong khi lớp proof thực hiện các luật , và cùng với lớp Trust đánh giá nhằm quyết định ứng dụng nên hay không nên tin tưởng .

5. Vai trò các lớp trong kiến trúc Semantic Web

Unicode: chỉ đơn thuần là một bảng mã chuẩn chung có đủ các ký tự để thông nhất sự giao tiếp trên tất cả các quốc gia, đáp ứng tính nhất quán toàn cầu của web.

5.1 URI (Uniform Resource Identifier)

Một URI (Uniform Resource Identifier) là một kí hiệu nhận dạng Web đơn giản. Cụ thể, nó là một xâu ngắn cho phép nhận dạng tài nguyên Web như: với các xâu bắt đầu với "http:" hoặc "ftp:" mà chúng ta thường thấy trên World Wide Web. Bất kỳ một người nào cũng có thể tạo một URI, và sở hữu chúng và chúng là một công nghệ cơ sở để xây dựng một hệ thống Web toàn cầu. Hệ thống World Wide Web được xây dựng trên chúng và bất kỳ cái gì mà có một URI thì được coi là "trên Web".

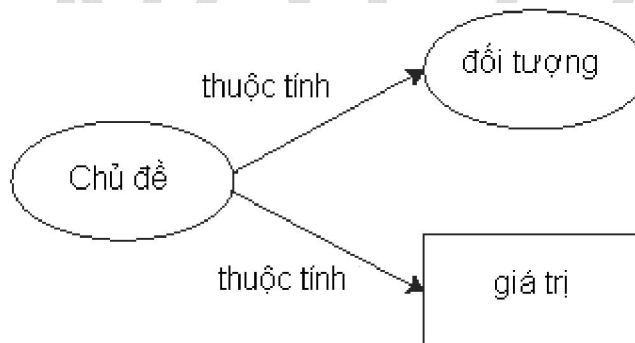
URL (Uniform Resource Locator) là một dạng đặc biệt của URI, cụ thể nó là một địa chỉ trên mạng.

URIref (URI reference) là một URI cùng với một phần nhận dạng tùy ý ở cuối. Ví dụ, ta có một URIref : "http://www.example.org/Books#Ontology" bao gồm một URI: "http://www.example.org/Books" và một phần nhận dạng "Ontology" được cách nhau bởi kí hiệu #.

Theo như quy ước, các không gian tên là những tài nguyên mà tạo ra các đa tài nguyên, thường là những URI được kết thúc bởi kí hiệu #. Ví dụ: "http://www.example.org/Books#" là một không gian tên. Các tài nguyên không có URIsref thì được gọi là các nút trắng; một nút trắng chỉ ra sự tồn tại của tài nguyên không có sự đề cập rõ ràng về tham chiếu URIsref của tài nguyên.

RDF (Resource Description Framework): Khung mô tả tài nguyên

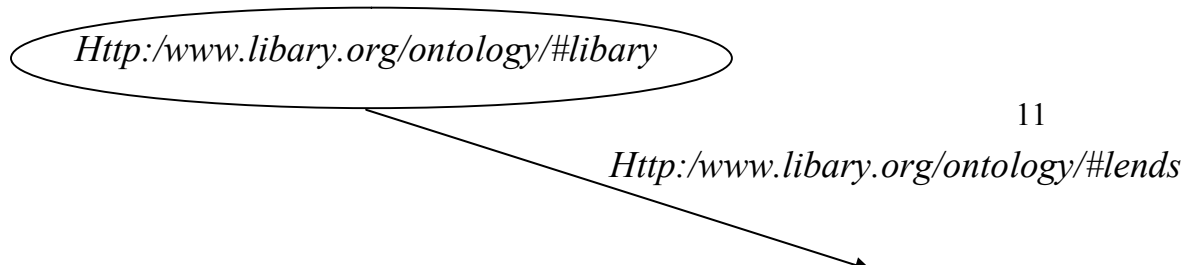
RDF được W3C giới thiệu để cung cấp một cú pháp chuẩn để tạo, thay đổi và sử dụng các chủ thích trong Web ngữ nghĩa. Một mệnh đề RDF là một bộ ba có dạng: [*chủ đề thuộc tính đối tượng*]. Trong đó, *chủ đề* là tài nguyên mà được mô tả bằng *thuộc tính* và *đối tượng*. *Thuộc tính* thể hiện mối quan hệ giữa *chủ đề* và *đối tượng*. Còn *đối tượng* ở đây có thể là một tài nguyên hoặc một giá trị. Ba thành phần trên trong RDF đều là các URI.



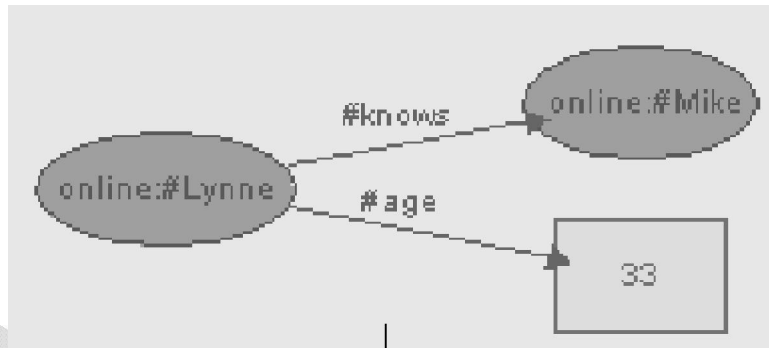
Hình 1.4. Bộ ba RDF

Ví dụ : về mệnh đề RDF

Giả sử biểu diễn mệnh đề "*The library lends books*", ta có chủ đề là "*The library*", thuộc tính là "*lends*", đối tượng là "*books*" và các thông tin này có các URI tương ứng là "*Http://www.library.org/ontology/#library*", "*Http://www.library.org/ontology/#lends*" và "*Http://www.library.org/ontology/#books*". Khi đó mệnh đề được biểu diễn dạng đồ thị như sau:



Một tập các mệnh đề RDF được gọi là đồ thị RDF.



Hình 1.5. Đồ thị RDF gồm hai mệnh đề

Để biểu diễn các mệnh đề RDF mà máy có thể hiểu, RDF định nghĩa ra một số cú pháp như: cú pháp Notation 3 (hay còn gọi là N3), cú pháp ngôn ngữ RDF/XML (mở rộng từ XML), hay đồ thị của các bộ ba như ví dụ Hình 2.5.

5.2 RDFS (RDF schema)

RDFS là một ngôn ngữ ontology đơn giản của web ngữ nghĩa, được coi là một ngôn ngữ cơ sở của web ngữ nghĩa. RDFS là ngôn ngữ mô tả bộ từ vựng trên các bộ ba RDF. Nó cung cấp các công việc sau:

- Định nghĩa các lớp tài nguyên
- Định nghĩa các quan hệ giữa các lớp
- Định nghĩa các loại thuộc tính mà các lớp trên có
- Định nghĩa các mối quan hệ giữa các thuộc tính.

5.3 Ontology Vocabulary

Bộ từ vựng ontology được xây dựng trên cơ sở tầng RDF và RDFS, cung cấp biểu diễn ngữ nghĩa mềm dẻo cho tài nguyên web và có khả năng hỗ trợ lập luận. Để xây dựng được các bộ từ vựng này, người ta đã sử dụng các ngôn ngữ ontology để biểu diễn chúng như: RDFS, OIL, DAML, DAML+OIL, OWL, ... Các ngôn ngữ này cung cấp khả năng biểu diễn và hỗ trợ lập luận khác nhau và chúng dựa trên nền tảng là các ngôn ngữ logic mô tả tương ứng khác nhau.

5.4 Tầng Logic

Việc biểu diễn các tài nguyên dưới dạng các bộ từ vựng ontology có mục đích là để máy có thể lập luận được. Mà cơ sở lập luận chủ yếu dựa vào logic. Chính vì vậy mà các ontology được ánh xạ sang logic, cụ thể là logic mô tả để có thể hỗ trợ lập luận. Vì logic mô tả có biểu diễn ngữ nghĩa hình thức (đặc trưng của lý thuyết mô hình), và cung cấp các dịch vụ lập luận, là cơ sở để hỗ trợ máy có thể lập luận và hiểu tài nguyên.

5.5 Tầng Proof

Tầng này đưa ra các luật để suy luận. Cụ thể từ các thông tin đã có ta có thể suy ra các thông tin mới. Ví dụ: *A là cha của B, A là em trai C* thì khi đó ta có thông tin mới là *C là bác của B*. Để có được các suy luận này thì cơ sở là FOL (First-Order-Logic). Và tầng này hiện nay các nhà nghiên cứu đang xây dựng các ngôn ngữ luật cho nó như: SWRL, RuleML.

5.6 Tầng Trust

Đảm bảo tính tin cậy của các ứng dụng trên Web ngữ nghĩa. Ví dụ: có một người bảo *x là xanh*, một người khác lại nói *x không xanh*, như thế Web ngữ nghĩa là không đáng tin cậy? Câu trả lời ở đây được xem xét trong các ngữ cảnh. Mỗi ứng dụng trên web ngữ nghĩa sẽ có một ngữ cảnh cụ thể, chính vì thế các mệnh đề trên có thể nằm trong các ngữ cảnh khác nhau khi đó ngữ nghĩa tương ứng khác nhau nên các mệnh đề đó vẫn đúng, đáng tin cậy trong ngữ cảnh của nó. Để có được sự chứng minh về độ tin cậy thì các lập luận được áp dụng là không đơn điệu

và có các cơ chế kiểm tra chứng minh kết hợp với công nghệ chữ ký điện tử để xác nhận độ tin cậy. Các ngôn ngữ chứng minh là ngôn ngữ cho ta chứng minh một mệnh đề là đúng hay sai.

6. Nội dung xây dựng Web ngữ nghĩa

Để xây dựng hệ thống Web ngữ nghĩa thay thế cho World Wide Web hiện tại, các nhà nghiên cứu đang nỗ lực và tập trung nghiên cứu với ba hướng chính sau:

- Chuẩn hoá các ngôn ngữ biểu diễn dữ liệu (XML) và siêu dữ liệu (RDF) trên Web.
- Chuẩn hoá các ngôn ngữ biểu diễn Ontology cho Web có ngữ nghĩa.
- Phát triển nâng cao Web có ngữ nghĩa (Semantic Web Advanced Development - SWAD).

6.1 Chuẩn hoá các ngôn ngữ biểu diễn dữ liệu (XML) và các siêu dữ liệu (RDF) trên Web

Như chúng ta đã biết, World Wide Web đã rất thành công bởi ngôn ngữ XML. XML đã làm cho công nghệ Web phát triển mạnh mẽ như hiện nay. Và Web ngữ nghĩa được phát triển trên nền Web hiện tại, trên nền XML, chúng ta có thể thấy rất rõ ngay trong sơ đồ kiến trúc của Web ngữ nghĩa. Web ngữ nghĩa được bổ sung các chú thích ngữ nghĩa cho các tài nguyên Web, và W3C đã giới thiệu RDF là chuẩn cú pháp để tạo, thay đổi và sử dụng chú thích trong Web ngữ nghĩa. Do vậy việc xây dựng ngôn ngữ chuẩn cho XML và RDF là thiết yếu để biểu diễn dữ liệu cho Web ngữ nghĩa. Trong quá trình này, đã có một số cú pháp được đưa ra để biểu diễn RDF như: Notation 3 (hay N3), XML/RDF, hay đồ thị các mệnh đề của RDF như đã trình bày ở trên. Các cú pháp này hỗ trợ biểu diễn dữ liệu cho máy có thể hiểu được.

6.2 Chuẩn hoá các ngôn ngữ biểu diễn Ontology cho Web có ngữ nghĩa

Ontology đóng vai trò then chốt trong việc cung cấp tài nguyên có thể truy nhập để xử lý tự động bằng cách cung cấp bộ từ vựng cho việc đánh dấu ngữ nghĩa của Web ngữ nghĩa. Để biểu diễn Ontology cho phát triển Web có ngữ nghĩa cần thiết đưa ra một ngôn ngữ chuẩn cho sự biểu diễn mềm dẻo, linh hoạt và đa dạng của tài nguyên Web. Trong quá trình chuẩn hoá ngôn ngữ biểu diễn Ontology, một số ngôn ngữ được đề xuất với các khả năng biểu diễn tăng dần như: RDFS, DAML+OIL, OWL, ... và tiếp tục được mở rộng trong tương lai. Các ngôn ngữ này có khả năng biểu diễn tăng dần nhưng tương ứng là khả năng quyết định giảm dần (độ phức tạp tăng dần). Và các ngôn ngữ biểu diễn Ontology được xây dựng phải cân bằng được khả năng biểu diễn và độ phức tạp tính toán. Các ngôn ngữ này được xây dựng trên nền các chuẩn XML, RDF, và thường sử dụng cơ sở logic là logic mô tả để biểu diễn ngữ nghĩa và hỗ trợ lập luận.

6.3 Phát triển nâng cao Web ngữ nghĩa (Semantic Web Advanced Development - SWAD)

Các công việc chuẩn hoá các ngôn ngữ biểu diễn dữ liệu XML, siêu dữ liệu RDF hay ngôn ngữ biểu diễn Ontology là các công việc nền tảng, cơ sở tạo ra chuẩn chung cơ sở để phát triển hệ thống và các ứng dụng trên Web ngữ nghĩa. Việc đưa các chuẩn đó thành các thể hiện, các sản phẩm của hệ thống Web ngữ nghĩa là công việc của SWAD. SWAD thực hiện nghiên cứu và đưa ra các thể hiện của Web ngữ nghĩa trên công nghệ cơ sở và nó cũng bổ sung, hoàn thiện các công nghệ cơ sở này. Nó chỉ ra các công nghệ Web ngữ nghĩa được sử dụng như thế nào? Mục tiêu cụ thể của SWAD là sử dụng các công nghệ, kỹ thuật và các công cụ hiện có của Web ngữ nghĩa để tạo ra các chương trình vượt xa hơn tầm hiện có, phát triển các thể hiện thực tiễn và cung cấp các công cụ và các chuẩn mới cho Web ngữ nghĩa. Để phát triển SWAD hiện tại đã có một số dự án của các tổ chức thực hiện nghiên cứu và phát triển như: SWAD-Europe, SWAD-DAML, SWAD-Simile, SWAD-Oxygen. Hiện tại, ở Việt Nam, một dự án cấp Nhà nước đang nghiên cứu thực hiện trích rút tự động thông tin từ các trang Web về tin tức

để xây dựng các ontology cho lĩnh vực này nhằm "tiếp nhận" công nghệ Web ngữ nghĩa khi có hệ thống mạng mới này.

7. Tổng kết chương

Qua phần II này chúng ta có thể hiểu một cách khái quát về Web ngữ nghĩa như động cơ, mục đích và các công việc để phát triển Web ngữ nghĩa. Trong đó, ta thấy được Ontology có một vai trò quan trọng trong Web ngữ nghĩa.

Trong chương tiếp theo chúng ta sẽ tìm hiểu chi tiết hơn về Ontology

III. Ontology và Web ngữ nghĩa

Chương này sẽ trình bày chi tiết về Ontology và cơ sở cũng như cú pháp và ngữ nghĩa của ngôn ngữ OWL DL, một ngôn ngữ phát triển Ontology Web

1. Khái niệm về ontology

Các Ontology đóng vai trò then chốt trong việc cung cấp ngữ nghĩa mà máy có thể hiểu được cho các tài nguyên của Web ngữ nghĩa. Ontology là một thuật ngữ mượn từ triết học nhằm chỉ khoa học mô tả các loại thực thể trong thế giới thực và cách chúng liên kết với nhau. Trong khoa học máy tính, một cách khái quát, Ontology là "một biểu diễn của sự khái niệm hoá chung được chia sẻ" của một miền nhất định. Nó cung cấp một bộ từ vựng chung bao gồm các khái niệm, các thuộc tính quan trọng và các định nghĩa về các khái niệm và các thuộc tính này. Ngoài bộ từ vựng, Ontology còn cung cấp các ràng buộc, đôi khi các ràng buộc này được coi như các giả định cơ sở về ý nghĩa mong muốn của bộ từ vựng, nó được sử dụng trong một miền mà có thể được giao tiếp giữa người và các hệ thống ứng dụng phân tán khác. Việc tiếp cận ngữ nghĩa tài nguyên Web thông qua các Ontology thì mềm dẻo hơn vì người sử dụng có thể lựa chọn bộ từ vựng và các ràng buộc trong các Ontology. Ví dụ, các ứng dụng trong các miền khác nhau có thể sử dụng các Ontology khác nhau. Đặc biệt, các Ontology có thể được sử dụng để đặc tả ý nghĩa của các tài nguyên Web (thông qua các chú thích) bằng cách xác nhận các tài nguyên như các trường hợp cụ thể của một số khái niệm quan trọng và hay hoặc khẳng định

các tài nguyên có quan hệ với các tài nguyên khác thông qua một số thuộc tính quan trọng đã định nghĩa trong các Ontology.

Từ vựng trong một Ontology có thể được biểu diễn bằng các khái niệm và các quan hệ được đặt tên và các định nghĩa khái niệm có thể được biểu diễn bằng các giới thiệu tương đương. Các giả định cơ sở có thể được biểu diễn bằng các tiên đề khái niệm và quan hệ khái quát. Đôi khi một Ontology tương ứng với một cơ sở tri thức logic mô tả. Một Ontology cũng chứa các trường hợp của các khái niệm và các mối quan hệ quan trọng của các cá thể này, nó được biểu diễn bằng các khẳng định của logic mô tả.

Ví dụ sau chỉ ra một Ontology được biểu diễn trong logic mô tả thông qua các tiên đề và các khẳng định.

Ví dụ 3.1. Ví dụ về một Ontology có cơ sở logic mô tả

Một Ontology đơn giản về thư viện có tên là *ThuVien* có thể bao gồm 3 phần:

Phần thứ nhất: là tập các khái niệm và các thuộc tính quan trọng, có thể bao gồm:

- Các khái niệm như: *Sach*, *TacGia*, *DocGia*, *NhaXuatBan*, *Nguoi*
- Các thuộc tính: *doc*, *viet*, *motphanCua*, *namxuatban*, *theloai*.
- Khái niệm định nghĩa: ví dụ khái niệm “*Tác giả*” bao gồm tất cả những người viết sách hoặc một phần cuốn sách.

$$TacGia = Nguoi \sqcap \forall viet. (Sach \sqcup \exists motphanCua. Sach)$$

Thành phần thứ hai của Ontology *ThuVien* được đưa ra bằng các giả định cơ sở của miền và có thể bao gồm:

-Nguyễn Duy, Hồ Chí Minh, Phan Thị Tươi là tác giả:

$$NguyenDuy \sqsubseteq TacGia, HoChiMinh \sqsubseteq TacGia, PhanThiTuoi \sqsubseteq TacGia$$

Thành phần thứ ba của Ontology *ThuVien* là về các cá thể và mối quan hệ của chúng, có thể bao gồm:

- “VN-KIM CHO WEB VIỆT CÓ NGỮ NGHĨA” là một quồn sách: $\langle VN-KIMChoWebVietCoNguNghia \rangle : sach$

- “VN-KIM CHO WEB VIỆT CÓ NGỮ NGHĨA” được viết bởi Cao Hoàn Tru.

$\langle VN-KIMChoWebVietCoNguNghia, CaoHoangTru \rangle : viet$

Các Ontology dựa trên logic mô tả có thể khai thác các công cụ lập luận mạnh của logic mô tả, do vậy máy có thể hiểu được các tài nguyên Web dễ hơn. Hỗ trợ lập luận Logic mô tả có thể rất hữu ích để đảm bảo chất lượng của các Ontology mà Ontology là nòng cốt của Web ngữ nghĩa.

Giả sử ta muốn kiểm tra xem *CaoHoangTru* có phải là *TacGia* không?

Dựa vào 3 tri thức sau:

$\langle VN-KIMChoWebVietCoNguNghia \rangle : sach$

$TacGia = \text{Nguoi} \sqcap \forall viet. (Sach \sqcup \exists motphanCua. Sach)$

$\langle VN-KIMChoWebVietCoNguNghia, CaoHoangTru \rangle : viet$

Bộ lập luận DL sẽ đưa ra kết quả là thỏa. Do đó có thể có một đề nghị bổ sung tri thức cơ sở này vào trong Ontology *ThuVien*

Như vậy, các Ontology biểu diễn dựa trên logic mô tả đã khai thác được khả năng biểu diễn tri thức cũng như khả năng lập luận hiệu quả của logic mô tả để máy có thể hiểu được tài nguyên Web.

2. Các phần tử trong ontology

Các cá thể (Individuals) - Thể hiện

Các cá thể là các thành phần cơ bản, nền tảng của một ontology. Các cá thể trong một ontology có thể bao gồm các đối tượng cụ thể như con người, động vật, cái bàn... cũng như các cá thể trừu tượng như các thành viên hay các từ. Một ontology có thể không cần bất kỳ một cá thể nào, nhưng một trong những lý do chính của một ontology là để cung cấp một ngữ nghĩa của việc phân lớp các cá thể, mặc dù các cá thể này không thực sự là một phần của ontology.

Các lớp (Classes) - Khái niệm

Các lớp là các nhóm, tập hợp các đối tượng trừu tượng. Chúng có thể chứa các cá thể, các lớp khác, hay là sự phối hợp của cả hai.

Các ontology biến đổi tùy thuộc vào cấu trúc và nội dung của nó: Một lớp có thể chứa các lớp con, có thể là một lớp tổng quan (chứa tất cả mọi thứ), có thể là lớp chỉ chứa những cá thể riêng lẻ, Một lớp có thể xếp gộp vào hoặc bị xếp gộp vào bởi các lớp khác. Mỗi quan hệ xếp gộp này được sử dụng để tạo ra một cấu trúc có thứ bậc các lớp, thường là với một lớp thông dụng nhất kiểu Thing ở trên đỉnh và các lớp rất rõ ràng kiểu 2002, Ford ở phía dưới cùng.

Các thuộc tính (Properties)

Các đối tượng trong ontology có thể được mô tả thông qua việc khai báo các thuộc tính của chúng. Mỗi một thuộc tính đều có tên và giá trị của thuộc tính đó. Các thuộc tính được sử dụng để lưu trữ các thông tin mà đối tượng có thể có. Ví dụ, đối với một cá nhân có thể có các thuộc tính: Họ_tên, ngày_sinh, quê_quán, số_cmnd...

Giá trị của một thuộc tính có thể có các kiểu dữ liệu phức tạp.

Các mối quan hệ (Relation)

Một trong những ứng dụng quan trọng của việc sử dụng các thuộc tính là để mô tả mối liên hệ giữa các đối tượng trong ontology. Một mối quan hệ là một thuộc tính có giá trị là một đối tượng nào đó trong ontology.

Một kiểu quan hệ quan trọng là kiểu quan hệ xếp gộp (subsumption). Kiểu quan hệ này mô tả các đối tượng nào là các thành viên của các lớp nào của các đối tượng.

Hiện tại, việc kết hợp các ontology là một tiến trình được làm phần lớn là thủ công, do vậy rất tốn thời gian và đắt đỏ. Việc sử dụng các ontology là cơ sở để

cung cấp một định nghĩa thông dụng của các thuật ngữ cốt lõi có thể làm cho tiến trình này trở nên dễ quản lý hơn. Hiện đang có các nghiên cứu dựa trên các kỹ thuật sản sinh để nối kết các ontology, tuy nhiên lĩnh vực này mới chỉ hiện hữu về mặt lý thuyết.

3. Ngôn ngữ OWL DL

Web Ontology Language (OWL) là ngôn ngữ đánh dấu được sử dụng để xuất bản và chia sẻ dữ liệu sử dụng các ontology trên Internet. OWL là một bộ từ vựng mở rộng của khung mô tả tài nguyên (RDF) và được kế thừa từ ngôn ngữ DAML+OIL Web ontology – một dự án được hỗ trợ bởi W3C. OWL biểu diễn ý nghĩa của các thuật ngữ trong các từ vựng và mối liên hệ giữa các thuật ngữ này để đảm bảo phù hợp với quá trình xử lý bởi các phần mềm.

OWL được xem như là một kỹ thuật trọng yếu để cài đặt cho Semantic Web trong tương lai. OWL được thiết kế đặc biệt để cung cấp một cách thức thông dụng trong việc xử lý nội dung thông tin của Web. Ngôn ngữ này được kỳ vọng rằng sẽ cho phép các hệ thống máy tính có thể đọc được thay thế cho con người. Vì OWL được viết bởi XML, các thông tin OWL có thể dễ dàng trao đổi giữa các kiểu hệ thống máy tính khác nhau, sử dụng các hệ điều hành và các ngôn ngữ ứng dụng khác nhau. Mục đích chính của OWL là sẽ cung cấp các chuẩn để tạo ra một nền tảng để quản lý tài sản, tích hợp mức doanh nghiệp và để chia sẻ cũng như tái sử dụng dữ liệu trên Web. OWL được phát triển bởi nó có nhiều tiện lợi để biểu diễn ý nghĩa và ngữ nghĩa hơn so với XML, RDF và RDFS, và vì OWL ra đời sau các ngôn ngữ này, nó có khả năng biểu diễn các nội dung mà máy có thể biểu diễn được trên Web.

| Cú pháp trừu tượng | Cú pháp DL | Ngữ nghĩa |
|--------------------------------|------------|---|
| ObjectProperty(R) | R | $R^I \subseteq \Delta^I \times \Delta^I$ |
| ObjectProperty(S inverseOf(R)) | R- | $(R-)^I \subseteq \Delta^I \times \Delta^I$ |

Bảng 2.1. Các mô tả thuộc tính đối tượng OWL

| Cú pháp trừu tượng | Cú pháp DL | Ngữ nghĩa |
|-------------------------------------|--------------------------|---|
| Class(A) | A | $A^I \subseteq \Delta^I$ |
| Class(owl:Thing) | T | $T^I = \Delta^I$ |
| Class(owl:Nothing) | \perp | $\perp^I = \emptyset$ |
| intersectionOf(C_1, C_2, \dots) | $C_1 \sqcap C_2$ | $(C_1 \sqcap C_2)^I = C_1^I \cap C_2^I$ |
| unionOf(C_1, C_2, \dots) | $C_1 \sqcup C_2$ | $(C_1 \sqcup C_2)^I = C_1^I \cup C_2^I$ |
| complementOf(C) | $\neg C$ | $(\neg C)^I = \Delta^I \setminus C^I$ |
| oneOf(o_1, o_2, \dots) | $\{o_1\} \sqcup \{o_2\}$ | $(\{o_1\} \sqcup \{o_2\})^I = \{o_1^I, o_2^I\}$ |
| restriction(R someValuesFrom(C)) | $\exists R.C$ | $\{x \in \Delta^I \mid \exists y. \langle x, y \rangle \in R^I \wedge y \in C^I\}$ |
| restriction(R allValuesFrom(C)) | $\forall R.C$ | $\{x \in \Delta^I \mid \forall y. \langle x, y \rangle \in R^I \rightarrow y \in C^I\}$ |
| restriction(R hasValue($\{o\}$)) | $\exists R. \{o\}$ | $(\exists R. \{o\})^I = \{x \in \Delta^I \mid \langle x, o^I \rangle \in R^I\}$ |
| restriction(R minCardinality(m)) | $\geq mR$ | $\{x \in \Delta^I \mid (\# y. \langle x, y \rangle \in R^I) \geq m\}$ |
| restriction(R maxCardinality(m)) | $\leq mR$ | $\{x \in \Delta^I \mid (\# y. \langle x, y \rangle \in R^I) \leq m\}$ |

Bảng 2.2. Các mô tả thuộc tính lớp OWL

| Cú pháp trừu tượng | Cú pháp DL | Ngữ nghĩa |
|---|---|--|
| Class(A partial $C_1 \dots C_n$) | $A \sqsupseteq C_1 \sqcap \dots \sqcap C_n$ | $A^I \subseteq C_1^I \cap \dots \cap C_n^I$ |
| Class(A complete $C_1 \dots C_n$) | $A \equiv C_1 \sqcap \dots \sqcap C_n$ | $A^I = C_1^I \cap \dots \cap C_n^I$ |
| EnumeratedClass(A $o_1 \dots o_n$) | $A \equiv \{o_1\} \sqcap \dots \sqcap \{o_n\}$ | $A^I = \{o_1^I, \dots, o_n^I\}$ |
| SubClassOf(C_1, C_2) | $C_1 \sqsupseteq C_2$ | $C_1^I \subseteq C_2^I$ |
| EquivalentClasses($C_1 \dots C_n$) | $C_1 \equiv \dots \equiv C_n$ | $C_1^I = \dots = C_n^I$ |
| DisjointClasses($C_1 \dots C_n$) | $C_i \sqsupseteq \neg C_j, (1 \leq i < j \leq n)$ | $C_i^I \cap C_j^I = \emptyset (1 \leq i < j \leq n)$ |
| SubPropertyOf(R_1, R_2) | $R_1 \sqsupseteq R_2$ | $R_1^I \subseteq R_2^I$ |
| EquivalentProperties($R_1 \dots R_n$) | $R_1 \equiv \dots \equiv R_n$ | $R_1^I = \dots = R_n^I$ |
| ObjectProperty(R) | $R \sqsupseteq R_i$ | $R^I \subseteq R_i^I$ |
| super(R_1)...super(R_n) | $\geq 1R \sqsupseteq C_i$ | $R^I \subseteq C_i^I \times \Delta^I$ |
| domain(C_1)... domain(C_k) | $T \sqsupseteq \forall R. C_i$ | $R^I \subseteq \Delta^I \times C_i^I$ |
| range(C_1) ... range(C_h) | $T \sqsupseteq \forall R. C_i$ | $R^I \subseteq \Delta^I \times C_i^I$ |
| [Symmetric] | $R \equiv R^-$ | $R^I = (R^-)^I$ |

| | | |
|---|--|---|
| [Functional] [InverseFunctional] [Transitive]) AnnotationProperty(R) | Func(R) Func(R ⁻) Trans(R) | $\{ \langle x, y \rangle \mid (\# y. \langle x, y \rangle \in R^I) \leq 1 \}$ $\{ \langle x, y \rangle \mid (\# y. \langle x, y \rangle \in (R^-)^I) \leq 1 \}$ $R^I = (R^-)^+$ |
| Individual(o type(C ₁) ... type(C _n) value(R ₁ , o ₁)... value(R _n , o _n)) | o: C _i , 1 ≤ i ≤ n $\langle o, o_i \rangle : R_i, 1 \leq i \leq n$ | $o^I \in C_i^I, 1 \leq i \leq n$ $\langle o^I, o_i^I \rangle \in R_i^I, 1 \leq i \leq n$ |
| SameIndividual(o ₁ ... o _n) | o ₁ = ... = o _n | $o_1^I = \dots = o_n^I$ |
| DifferentIndividuals(o ₁ ... o _n) | o _i ≠ o _j , 1 ≤ i < j ≤ n | $o_i^I \neq o_j^I, 1 \leq i < j \leq n$ |

Bảng 2.3. Các tiên đề và các sự kiện của OWL

Bảng 2.3. trình bày một phần của Ontology "ThuVien" trong cú pháp trừu tượng OWL.

```

Namespace(elp=<http://example.org/Library#>)
Ontology(elp:Ontology
  Class(elp:Sach)
  Class(elp:Nguoi)
  Class(elp:TacGia partial exp:Nguoi)
  Class(elp:NguyenDuy partial exp:TacGia )
  Class(elp:HoChiMinh partial exp:TacGia )
  Class(elp:PhanThiTuoi partial exp:TacGia )
  ObjectProperty(elp:viet)
  Class(elp:TacGia Complete elp: Nguoi
    restriction(elp:viet allValuesFrom(
      unionOf (elp:Sach))
    )
  )
  Class(elp: VN-KIMChoWebVietCoNguNghia partial exp:Sach)
  Individual(elp: VN-KIMChoWebVietCoNguNghia
    type(elp:CaoHoangTru)
    value(elp:viet)
  )
)

```

Bảng 2.4. Một Ontology ví dụ trong cú pháp trừu tượng của OWL DL

4. Tổng kết chương

Trong phần này giới thiệu ngôn ngữ OWL DL là ngôn ngữ biểu diễn Ontology. Trong chương này cũng trình bày về cú pháp và ngữ nghĩa của OWL DL. Ngôn ngữ OWL DL cho phép định nghĩa các Ontology, giúp xây dựng bộ từ vựng cho Web ngữ nghĩa, hỗ trợ chú thích ngữ nghĩa cho tài nguyên Web.

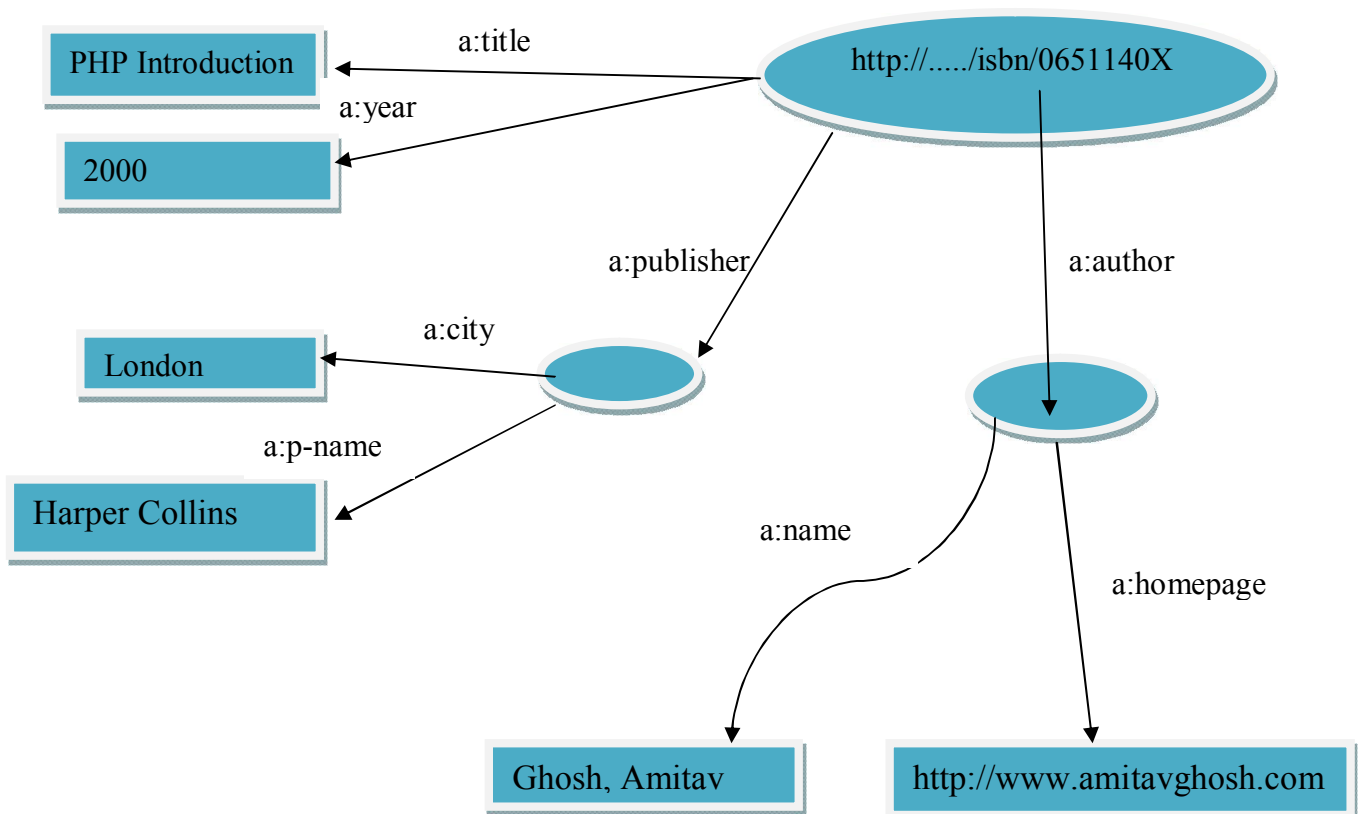
IV. Ứng dụng bookstore với semantic web

1. Dữ liệu bookstore của cơ sở dữ liệu A

| ISBN | Author | Title | Publisher | Year |
|----------|--------|------------------|-----------|------|
| 0651140X | Id_xyz | The Glass Palace | Id_qpr | 2000 |

| ID | Name | HomePage |
|--------|---------------|----------------------------|
| Id_xyz | Ghosh, Amitav | http://www.amitavghosh.com |

| ID | Publisher_name | City |
|--------|----------------|--------|
| Id_qpr | Harper Collins | London |



2. Dữ liệu bookstore của cơ sở dữ liệu F

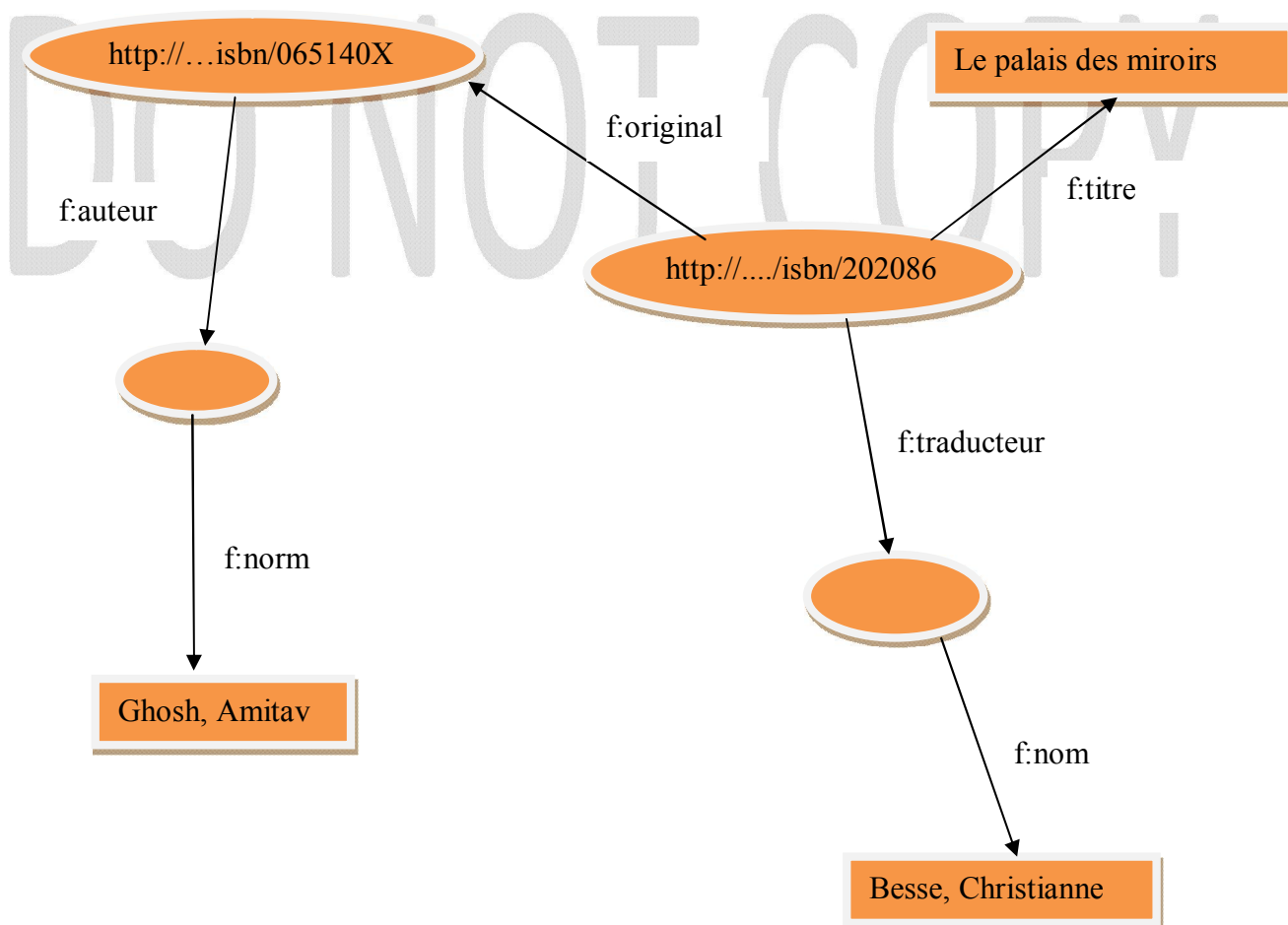
| ID | Titre(tiêu đề) | Traducteur (dịch) | Original |
|-------------|-----------------------|--------------------|--------------|
| ISBN 202086 | Le Palais des Miroirs | \$A12\$ | ISBN 065140X |

| ID | Auteur(tác giả) |
|---------------|-------------------|
| ISBN 0651140X | \$A11\$ |

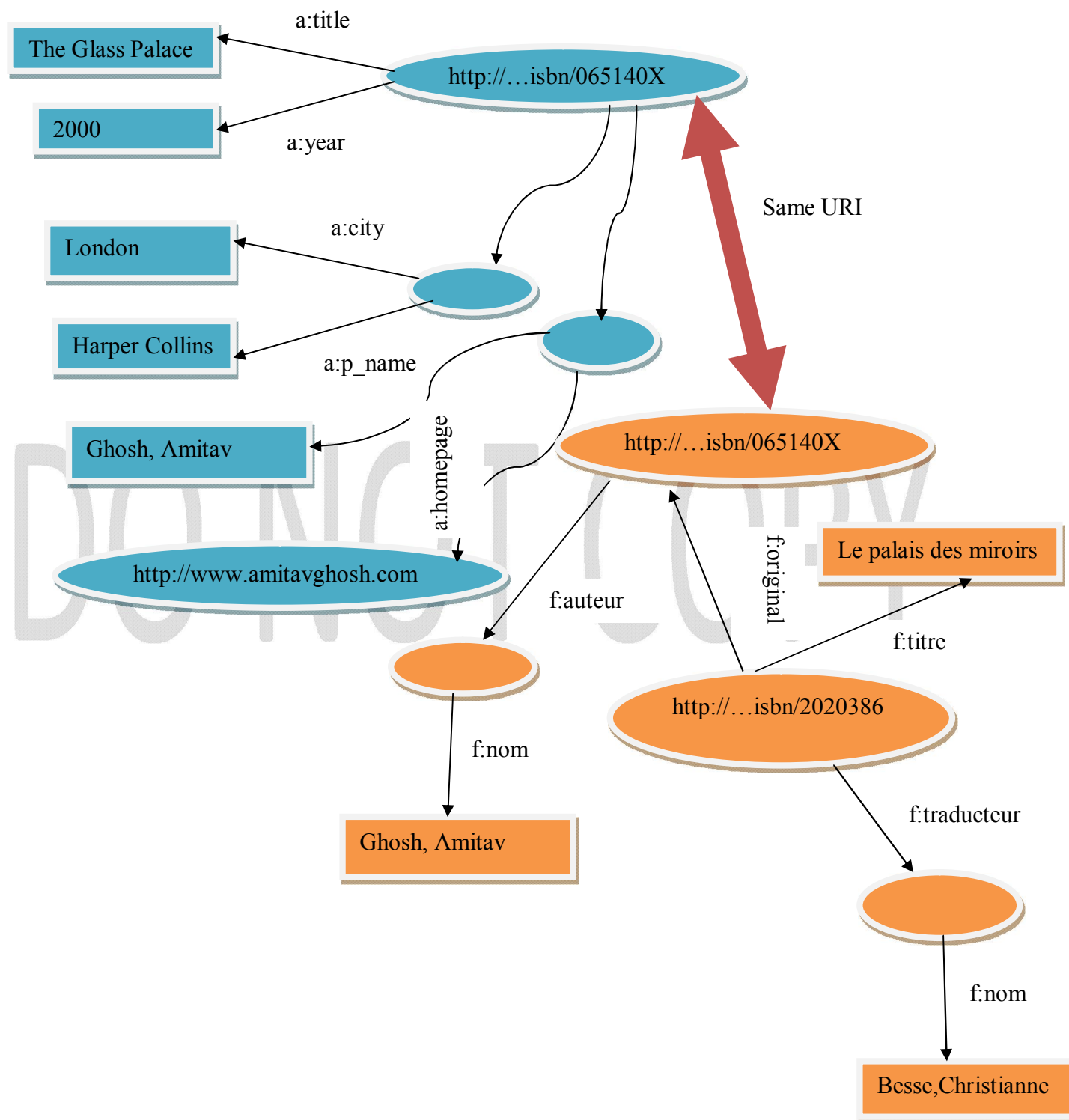
| Nom |
|--------------------|
| Ghosh, Amitav |
| Besse, Christianne |

Nom : tên

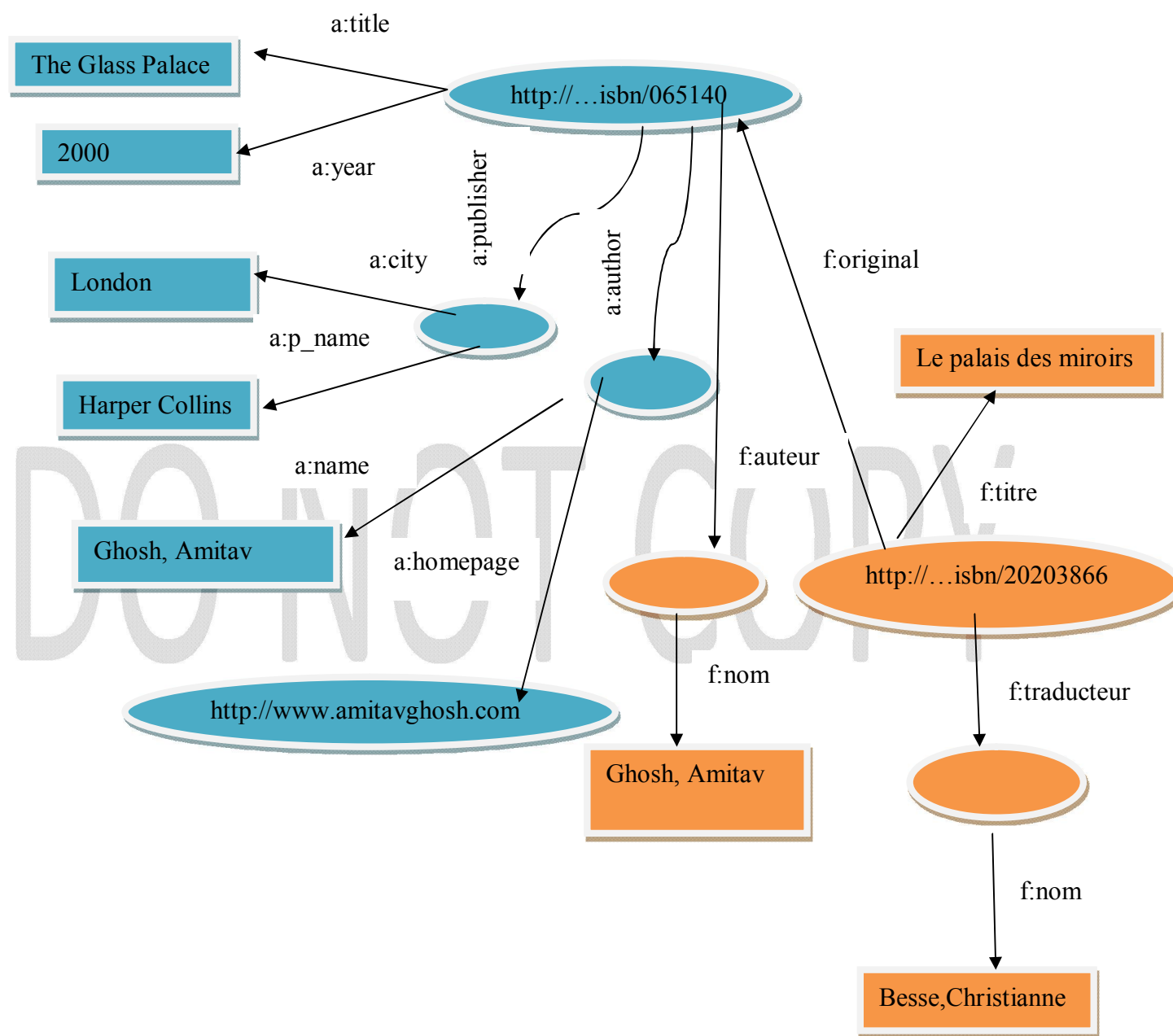
Kết hợp 2 tập dữ liệu



3. Bắt đầu việc trộn dữ liệu



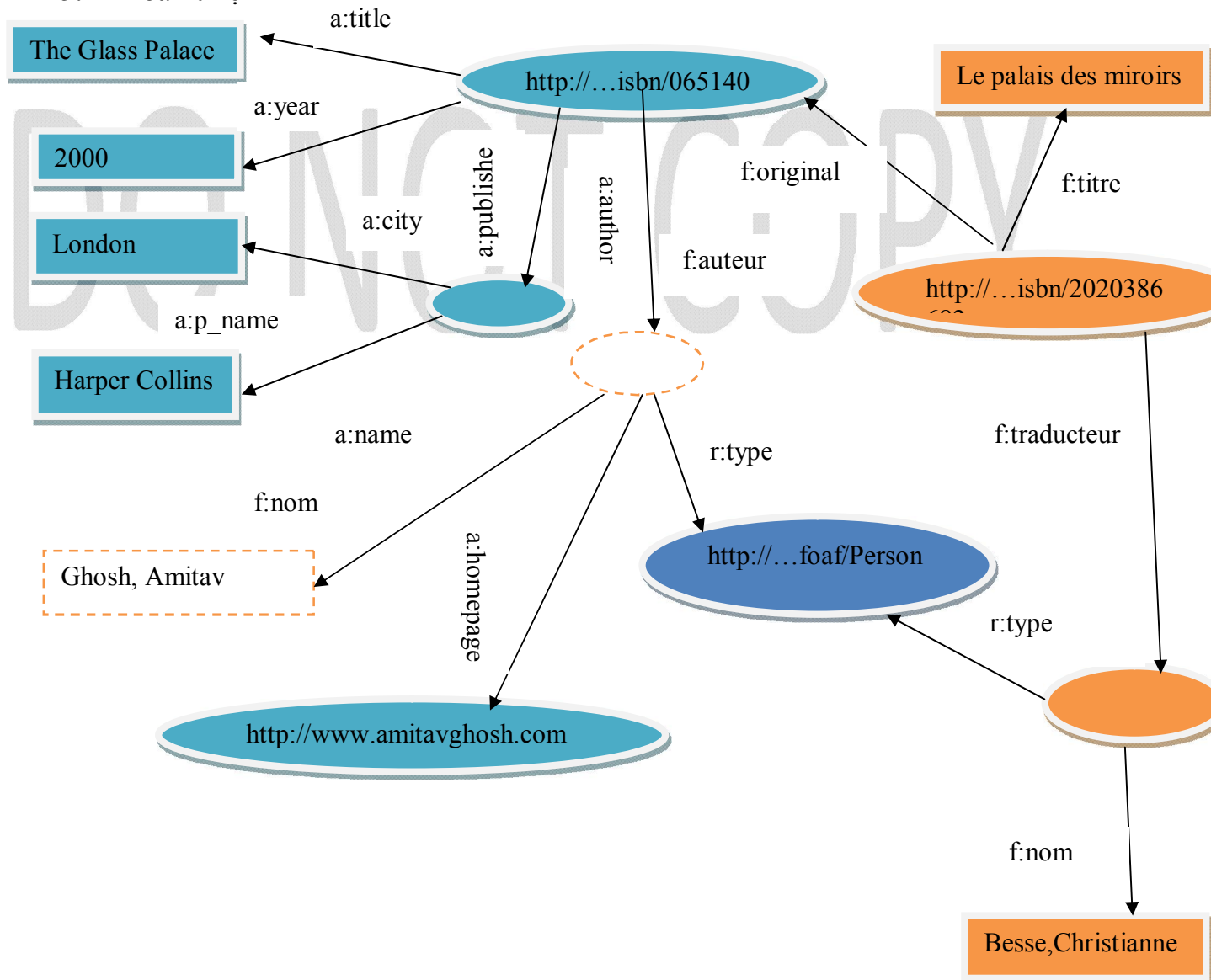
4. Trộn dữ liệu giữa cơ sở dữ liệu A và F



Một số câu truy vấn trong cơ sở dữ liệu F (bảng tiếng pháp)

- Lấy ra tiêu đề của quyển sách gốc -> thông tin này ko này trong cơ sở dữ liệu F nhưng có thể lấy bằng cách trộn với cơ sở dữ liệu A
- Chúng ta thấy a:author và f:auteur là giống nhau nhưng việc trộn tự động lại không biết được điều này .
- Chúng ta thêm 1 vài thông tin để dữ liệu có thể trộn tự động được
- a:author giống như f:auteur
- Thêm 1 định danh person , như bổ sung thêm 1 tài nguyên

5. Hoàn thiện



V. Tài liệu tham khảo

- [I] Luận văn tìm hiểu , nghiên cứu và xây dựng ứng dụng với semantic web- 2005-Đại học khoa học tự nhiên
- [II] Tutorial on Semantic web -2011-Ivan Herman
- [III] Realizing a Semantic Web Application - Emanuele Della Valle
- [IV] Semantic web - <http://www.phpvn.org/index.php/board,27.20.html>
- [V] Ứng dụng Semantic Web trong cuộc sống

DO NOT COPY