

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



NGUYỄN VĂN NHÂN

**MÔ HÌNH HOÁ TÀI NGUYÊN THÔNG TIN TRƯỜNG ĐẠI HỌC VÀ
HỖ TRỢ TRUY XUẤT THÔNG TIN THEO TIẾP CẬN LINKED
DATA**

LUẬN VĂN THẠC SĨ KỸ THUẬT
(Theo định hướng ứng dụng)

HÀ NỘI - NĂM 2020

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



NGUYỄN VĂN NHÂN

**MÔ HÌNH HOÁ TÀI NGUYÊN THÔNG TIN TRƯỜNG ĐẠI HỌC VÀ
HỖ TRỢ TRUY XUẤT THÔNG TIN THEO TIẾP CẬN LINKED DATA**

Chuyên ngành: Hệ thống thông tin

Mã số: 8.48.01.04

LUẬN VĂN THẠC SĨ KỸ THUẬT
(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC : PGS.TS. HOÀNG HUỖ HẠNH

HÀ NỘI - 2020

CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu khoa học độc lập của riêng tôi.

Các số liệu sử dụng phân tích trong luận văn có nguồn gốc rõ ràng, đã công bố theo đúng quy định. Các kết quả nghiên cứu trong luận văn do tôi tự tìm hiểu, phân tích một cách trung thực, khách quan và phù hợp với thực tiễn của Việt Nam. Kết quả này chưa từng được công bố trong bất kỳ nghiên cứu nào khác.

Tác giả luận văn ký và ghi rõ họ tên

Nguyễn Văn Nhân

LỜI CẢM ƠN

Để hoàn thành được luận văn, ngoài sự nghiên cứu và cố gắng của bản thân, tôi xin cảm ơn thầy giáo PGS.TS Hoàng Hữu Hạnh - người thầy trực tiếp hướng dẫn, tận tình chỉ bảo và định hướng cho tôi trong suốt quá trình thực hiện luận văn. Một lời cảm ơn chắc chắn không thể diễn tả hết lòng biết ơn sâu sắc của tôi tới thầy – một người thầy của tôi trên mọi phương diện!

Tôi xin gửi lời cảm ơn chân thành cảm ơn tất cả các thầy cô giáo của Học viện Công nghệ Bru chính Viễn thông đã giảng dạy, quan tâm nhiệt tình và dìu dắt tôi trong suốt quá trình học tập tại trường.

Cuối cùng, tôi xin gửi lời cảm ơn tới gia đình, bạn bè và những người đã luôn ở bên tôi cổ vũ, động viên, tạo điều kiện thuận lợi cho tôi học tập, tạo động lực tinh thần vô giá để tôi hoàn thiện luận văn này và ngày một hoàn thiện chính bản thân mình.

Trong quá trình nghiên cứu và thực hiện luận văn, mặc dù được sự hướng dẫn nhiệt tình của thầy giáo PGS.TS Hoàng Hữu Hạnh và những nỗ lực của bản thân nhưng cũng không thể tránh khỏi những thiếu sót hạn chế. Tôi rất mong nhận được ý kiến đóng góp, sửa chữa từ quý Thầy, Cô và các bạn bè đồng nghiệp để luận văn được hoàn thiện hơn.

Trân trọng cảm ơn!

Tác giả
Nguyễn Văn Nhân

MỤC LỤC

MỞ ĐẦU.....	1
Chương 1 - TỔNG QUAN VỀ WEB NGỮ NGHĨA VÀ LINKED DATA.....	5
1 Tổng quan về Web ngữ nghĩa	5
1.1 Web của ngày hôm nay.....	5
1.2 Web ngữ nghĩa	7
1.2.1 Khái niệm.....	7
1.2.2 Siêu dữ liệu	8
1.2.3 Kiến trúc Web ngữ nghĩa.....	8
1.2.4 Các khái niệm cơ bản của Web ngữ nghĩa	10
1.3 Ngôn ngữ Cơ cấu mô tả tài nguyên.....	13
1.3.1 Ngôn ngữ mô tả tài nguyên RDF	13
1.3.2 Lược đồ RDF và truy vấn RDF	20
1.4 Linked Data	26
1.4.1 Khái niệm về Linked Data	26
1.4.2 Quy tắc Linked Data	27
1.5 Tiểu kết chương 1.....	28
Chương 2 - ỨNG DỤNG QUẢN LÝ VÀ TRUY XUẤT TÀI NGUYÊN THÔNG TIN TRONG TRƯỜNG ĐẠI HỌC– LINKED OPEN PTIT DATA (LOPD).....	29
2.1 Tiến trình xây dựng ứng dụng LOPD.....	29
2.2 Jena	31
2.2.1 Giới thiệu	31
2.2.2 Jena API.....	31
2.2.3 Kiến trúc Jena 3	32
2.3. Mô hình hóa thông tin với Jena.....	33
2.3.1 Chương trình Hello World! trong Jena.....	33
2.3.2 Tạo mô hình RDF	33
2.3.3 Đọc mô hình RDF.....	39
2.4. Chuyển đổi dữ liệu web ngữ nghĩa.....	41

2.4.1 Dữ liệu từ Excel	41
2.4.2 Dữ liệu từ DBF	42
2.5 Chuẩn Dublin Core Metadata	44
2.6 Tiểu kết Chương 2	44
Chương 3 - PHÁT TRIỂN ỨNG DỤNG LOPD	47
3.1 Giới thiệu bài toán	47
3.1.1 Yêu cầu bài toán	47
3.1.2 Phân tích vấn đề	47
3.1.3 Chuẩn bị dữ liệu	48
3.2 Giải pháp hệ thống	49
3.2.1 Kiến trúc hệ thống	49
3.2.2 Thiết kế cơ sở dữ liệu	51
3.3 Xây dựng ứng dụng	57
3.3.1 Mô tả User case	57
3.3.2 Đặc tả chức năng	58
3.3.3 Thiết kế giao diện	59
KẾT LUẬN VÀ KIẾN NGHỊ	61
DANH MỤC TÀI LIỆ THAM KHẢO	62
PHỤ LỤC	63

DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng Việt
API	Application Programming Interface	Giao diện lập trình ứng dụng
CDF	Computable Document Format	Một định dạng của file
CSDL	Database	Cơ sở dữ liệu
HTML	Hyper Text Markup Language	Ngôn ngữ đánh dấu siêu văn bản
HTTP	Hyper Text Transfer Protocol	giao thức chuẩn của mạng Internet
LOPD	Linked Open PTIT Data	
OWL	Web Ontology Language	Ngôn ngữ bản thể web
URI	Uniform Resource Identifier	Một chuỗi các ký tự dùng để xác định một tài nguyên.
URL	Uniform Resource Locator	Địa chỉ dùng để tham chiếu đến các tài nguyên trên mạng Internet
URN	Uniform Resource Name	Định danh Tài nguyên thống nhất
WWW	World Wide Web	Hệ thống Web
XML	eXtensible Markup Language	Ngôn ngữ đánh dấu mở rộng

DANH MỤC CÁC BẢNG

Bảng 2.5 Các yếu tố cơ bản của chuẩn Dublin Core Metadata	45
Bảng 3.2.2.2 Thông tin cấu trúc bảng dblp_author_ref_new	55
Bảng 3.2.2.3 Thông tin cấu trúc bảng dblp_ref	55
Bảng 3.2.2.4 Thông tin cấu trúc bảng dbsa_sbj	56
Bảng 3.2.2.5 Thông tin cấu trúc bảng dbsa_pub_in_dblp	56
Bảng 3.2.2.6 Thông tin cấu trúc bảng dbsa_pub.....	57

DANH MỤC CÁC HÌNH

Hình 1. Mô hình liên kết dữ liệu trong Web ngữ nghĩa	2
Hình 1.3.1.3. Đồ thị RDF	15
Hình 1.3.2.1 Quan hệ kế thừa	20
Hình 1.3.2.3. Không gian miền và giới hạn của thuộc tính RDFS	23
Hình 1.3.1. Mô hình liên kết dữ liệu trong Web ngữ nghĩa	27
Hình 2.1. Tiến trình xuất bản Linked Data lên Web.....	29
Hình 2.2.2. Các thành phần của Jena API.....	32
Hình 2.2.3. Kiến trúc Jena 3.....	32
Hình 2.3.2.1. Biểu diễn phát biểu dưới dạng đồ thị	34
Hình 3.1.3.2. Cơ sở dữ liệu DBLP	48
Hình 3.2.1. Quá trình thực hiện LOSM	50
Hình 3.2.2.1 Các bảng trong cơ sở dữ liệu DBLP	52
Hình 3.2.2.2. Các bảng trong cơ sở dữ liệu hệ thống.....	57
Hình 3.3.1. Use case hệ thống tìm kiếm thông tin tác giả	58
Hình 3.3.3.1. Ứng dụng tìm kiếm thông tin tác giả	59
Hình 3.3.3.2.. Kết quả tìm kiếm tác giả PGS.TS Hoàng Hữu Hạnh và các gợi ý các tác giả có các bài báo hay công trình cùng chủ đề	59

MỞ ĐẦU

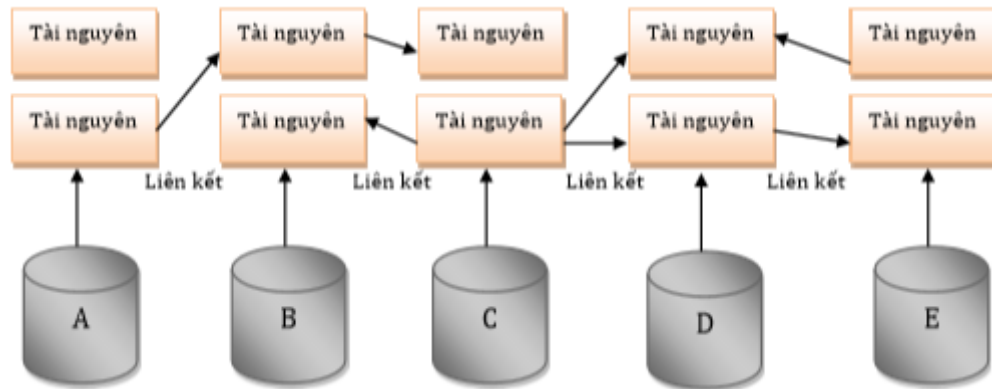
1. Lý do chọn đề tài

Thuật ngữ “Linked Data” được Tim Berner-Lee đưa ra trong các ghi chép về kiến trúc “Linked Data Web” của mình. Thuật ngữ này chỉ cách thức để xuất bản và liên kết các dữ liệu có cấu trúc trên Web. Giả thuyết cơ bản của Linked Data là lợi ích và giá trị của dữ liệu tăng lên khi nó được liên kết với các dữ liệu khác. Nói cách khác, Linked Data đơn giản là sử dụng Web để tạo ra các liên kết định kiểu (typed link) giữa các dữ liệu từ nhiều nguồn tài nguyên khác nhau. Điều này giống như hai CSDL của hai tổ chức ở các vùng địa lý khác nhau hay đơn giản là một hệ thống hỗn tạp của cùng một tổ chức không dễ dàng để trao đổi, liên thông ở mức dữ liệu. Do vậy, Linked Data nhằm vào dữ liệu được đưa lên Web theo cách máy tính có thể đọc được, có ngữ nghĩa rõ ràng và nó được liên kết đến tập dữ liệu bên ngoài và ngược lại các dữ liệu đó cũng liên kết đến nó[1].

Trong khi đơn vị chính của Web hiện nay (hay còn gọi là Web siêu liên kết) là tài liệu HTML được kết nối bởi các siêu liên kết không định kiểu (không xác định kiểu dữ liệu hay kiểu quan hệ), Linked Data dựa vào tài liệu chứa dữ liệu dạng RDF. Tuy nhiên, tốt hơn các tài liệu kết nối đơn giản, Linked Data sử dụng RDF để làm các câu định kiểu liên kết các đối tượng trong thế giới thực. Kết quả, những gì chúng ta tham chiếu trong Web của dữ liệu, có thể thực sự được mô tả như Web của những đối tượng trong thế giới thực, được mô tả bởi dữ liệu trên Web.

Linked Data sử dụng công nghệ Web hiện tại để kết nối các tài nguyên (đối tượng dữ liệu) liên quan đến nhau mà không cần liên kết trước, nghĩa là bỏ đi rào chắn liên kết dữ liệu mà hiện tại đang liên kết bởi nhiều phương thức khác nhau. Linked Data là kết nối dữ liệu phân tán trên Web.

Linked Data là thuật ngữ dùng để mô tả cách thức tốt nhất được đề nghị để duyệt, chia sẻ và kết nối các dữ liệu, thông tin, tri thức của Web ngữ nghĩa bằng cách sử dụng các URI và RDF.



Hình 1. Mô hình liên kết dữ liệu trong Web ngữ nghĩa

Dựa vào các đặc điểm và tính năng của Linked Data thì hàng loạt cá nhân và các nhóm đã đóng góp để xây dựng Web Dữ liệu, bỏ đi các hạn chế trong việc sử dụng lại, tích hợp và ứng dụng dữ liệu từ nhiều nguồn hỗn tạp, phân tán rất phù hợp cho việc quản lý tài nguyên dành cho giảng viên đại học và sinh viên trong quá trình giảng dạy và học tập. Qua thời gian, Linked Data là nền tảng, với các khuyến nghị của W3C, cùng với tầm nhìn và phát triển của Web ngữ nghĩa nên tôi chọn đề tài: *“Mô hình hoá tài nguyên thông tin trường đại học và hỗ trợ truy xuất thông tin theo tiếp cận Linked Data”*.

2. Tổng quan về vấn đề nghiên cứu

Đề tài là sự ứng dụng những đặc điểm và tính chất của mô hình và kỹ thuật Linked Data để triển khai và hoàn thiện một ứng dụng có tính áp dụng thực tiễn cao khi giải quyết được vấn đề còn tồn tại và hoa hụi do các phương thức truyền thống hay các ứng dụng hiện tại chưa đáp ứng được. Đề tài theo hướng ứng dụng và áp dụng những kỹ thuật và các công cụ hỗ trợ để đề tài hoàn thiện tốt hơn.

3. Mục đích nghiên cứu

Tiếp cận và tìm hiểu về Web ngữ nghĩa (Semantic Web), các công nghệ Web ngữ nghĩa (RDF/RDFS, ontology, OWL, SPARQL), Linked Data trong các ứng dụng thông minh trong hiện nay.

Xây dựng Ứng dụng Quản lý tài nguyên thông minh theo Linked Data và hỗ trợ giảng viên và sinh viên trong tìm kiếm các tài nguyên, và dữ liệu liên quan; tiến

hành phát triển demo được sản phẩm và định hướng tính ứng dụng của sản phẩm vào thực tiễn.

4. Đối tượng và phạm vi nghiên cứu

1. Đối tượng nghiên cứu:

- + Web ngữ nghĩa, RDF/RDFS, Ngôn ngữ truy vấn SPARQL
- + Mô hình dữ liệu Linked Data.
- + OWL - Web Ontology Language

2. Phạm vi nghiên cứu:

+ Đề tài thực hiện là một đề tài hướng ứng dụng và được thử nghiệm sử dụng trong môi trường nghiên cứu hoặc các cơ sở giáo dục.

5. Phương pháp nghiên cứu

- Phương pháp phân loại và hệ thống hoá lý thuyết
- Phương pháp phân tích và tổng hợp lý thuyết
- Phương pháp chuyên gia
- Phương pháp thực nghiệm xây dựng ứng dụng theo quy trình.

6. Nội dung nghiên cứu

Nội dung nghiên cứu của luận văn được bố cục thành các Chương như sau:

Chương 1: Tổng quan về Web ngữ nghĩa và Linked Data

Ở chương này luận văn sẽ tìm hiểu tổng quan về cơ sở lý thuyết và kiến trúc của Web ngữ nghĩa; đồng thời tìm hiểu về phương pháp mô hình hoá dữ liệu trong Web ngữ nghĩa là RDF cùng với các tiếp cận xây dựng cơ sở tri thức miền. Nội dung cũng được tìm hiểu sâu về công nghệ Web ngữ nghĩa là mô hình Linked Data (Dữ liệu Liên kết) và những vấn đề công nghệ trong thiết kế các ứng dụng sử dụng Linked Data.

Chương 2: Ứng dụng quản lý và truy xuất tài nguyên thông tin trong trường đại học – Linked Open PTIT Data (LOPD)

Chương này đi tìm hiểu và phân loại các nguồn tài nguyên thông tin trong trường đại học, và tập trung vào Học viện Công nghệ Bưu chính Viễn thông. Các nguồn tài nguyên thông tin sẽ tập trung tìm hiểu và phân tích gồm:

Tài nguyên về KHCN: đề tài KHCN, bài báo, công trình xuất bản trong và ngoài nước, hồ sơ KHCN cá nhân;

Các tài nguyên thông tin này sẽ được phân tích để xây dựng các lược đồ dữ liệu và chuyển đổi từ mô hình dữ liệu gốc sang mô hình dữ liệu Linked Data với các kết nối đến các datasets của hệ thống dữ liệu Linked Data mở toàn cầu (Linking Open Data).

Chương 3: Phát triển ứng dụng LOPD

Chương này tập trung vào các khía cạnh phát triển ứng dụng LOPD:

- + Các công cụ và mô hình phát triển ứng dụng theo Linked Data
- + Phân tích và thiết kế hệ thống LOPD;
- + Triển khai và phát triển
- + Kết quả cài đặt đạt được

Chương 1 - TỔNG QUAN VỀ WEB NGỮ NGHĨA VÀ LINKED DATA

1 Tổng quan về Web ngữ nghĩa

1.1 Web của ngày hôm nay

World Wide Web đã thay đổi xã hội con người vô cùng to lớn. Như đã thay đổi cách thức con người trao đổi với nhau cũng như cách điều hành công việc và kinh doanh và sự thay đổi là ở trung tâm của một cuộc cách mạng: chuyển đổi thế giới phát triển sang một nền kinh tế tri thức, hay nói rộng hơn là một xã hội tri thức. Sự phát triển này cũng thay đổi cách chúng ta nghĩ về các chiếc máy tính. Bây giờ máy tính không những được dùng để thực hiện các phép tính toán số học mà chúng hầu như được sử dụng cho việc xử lý thông tin, các ứng dụng đặc thù là các CSDL, xử lý văn bản, bảng tính và trò chơi điện tử.

Những thông tin hiện nay trên World Wide Web chủ yếu được biểu diễn ở dạng HTML, một ngôn ngữ phổ dụng để trình diễn thông tin. XML ra đời và trở thành một công cụ trao đổi dữ liệu không có cấu trúc, bán cấu trúc và có cấu trúc giữa các hệ thống, nâng cao sự tích hợp của các ứng dụng. Tuy nhiên, các giải pháp dựa trên XML cho quá trình tích hợp của các ứng dụng và các hệ thống chưa đủ, do dữ liệu được chuyển đổi thiếu mô tả tường minh về ngữ nghĩa của nó. Sự tích hợp của các ứng dụng cũng phải bao gồm sự tích hợp cả về ngữ nghĩa.

HTTP và HTML đã cung cấp các cách 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 thông tin trên Web, điều này làm nảy sinh vấn đề là làm thế nào để tìm kiếm chính xác tài nguyên mình mong muốn. Dữ liệu trong các file HTML có thể hữu ích ở ngữ cảnh này nhưng vô nghĩa đối với ngữ cảnh khác. Nhân loại đang dần dần tiến đến cuộc cách mạng công nghệ 4.0 và ngày các công nghệ thông tin và truyền thông đã có khả năng để thu thập được một số lượng lớn dữ liệu mà chúng có liên quan đến nhau về mặt khái niệm, tuy nhiên đa số những mối quan hệ này chỉ được con người “nhớ” chứ không được lưu trữ theo một cách mà giúp các máy tính có thể hiểu để xử lý. Thách thức này đã chỉ ra một hướng nghiên cứu đó là tạo ra khả năng

cho phép con người tạo, lưu giữ, sắp xếp, ghi phụ chú và truy xuất kho dữ liệu cá nhân rất lớn của mỗi người trong quá khứ theo hình thức như một nhật ký cuộc sống được cá thể hoá và sẽ trở thành một sự bổ sung và trợ giúp cho bộ nhớ con người.

Những hoạt động này đều đặc biệt không được hỗ trợ tốt của các công cụ phần mềm. Ngoài sự tồn tại của các liên kết để thiết lập các liên kết giữa các tài liệu, thì các công cụ có giá trị nhất trên Web hiện nay là các bộ tìm kiếm (search engines). Các công cụ tìm kiếm theo từ khoá như Yahoo! và Google là các công cụ chính trong việc sử dụng Web hiện nay. Rõ ràng rằng Web sẽ không thành công lớn như hiện nay nếu không có các công cụ tìm kiếm. Tuy nhiên, vẫn tồn tại các vấn đề liên quan đến các ứng dụng của chúng:

- Truy hồi cao, độ chính xác thấp: Ngay cả khi các trang liên quan chính được truy xuất, thì chúng vẫn không hữu ích khi rất rất nhiều các tài liệu ít liên quan hoặc không liên quan cũng được lấy về. Quá nhiều cũng dẫn đến không tốt cũng như quá ít.
- Truy hồi thấp hoặc không có. Trường hợp này xảy ra chúng ta không có được câu trả lời từ yêu cầu của chúng ta, hoặc các tài liệu liên quan và quan trọng không được lấy về. Cho dù việc truy hồi thấp khá hiếm khi có đối với các công cụ tìm kiếm, nhưng nó vẫn xảy ra.
- Các kết quả rất nhạy cảm với từ vựng. Thông thường các từ khoá tìm kiếm ban đầu không cho ta kết quả như mong muốn, lý do là các tài liệu liên quan sử dụng các thuật ngữ khác với truy vấn của chúng ta. Điều này rõ ràng là không thoả mãn bởi vì các truy vấn cùng ngữ nghĩa nên cho cùng một kết quả.
- Kết quả chỉ là những trang Web đơn giản. Nếu chúng ta cần những thông tin dàn trải trong các tài liệu khác nhau, chúng ta phải thực hiện nhiều truy vấn khác nhau để tập hợp các tài liệu liên quan; sau đó chúng ta sẽ xử lý bằng tay để trích rút các thông tin từng phần rồi kết hợp chúng lại với nhau.

1.2 Web ngữ nghĩa

1.2.1 Khái niệm

Web ngữ nghĩa không là Web riêng biệt mà 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. Web ngữ nghĩa được hình thành từ ý tưởng của Tim Berners-Lee, người phát minh ra WWW, URI, HTTP, và HTML. Web ngữ nghĩa 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. Web ngữ nghĩa 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. Web ngữ nghĩa còn cung cấp một môi trường chia sẻ và xử lý dữ liệu tự động bằng máy tính.[1]

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. 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 Web ngữ nghĩa ta có thể giải quyết vấn đề này bằng 2 cách:

Thứ nhất: Nó sẽ mô tả chi tiết dữ liệu. Do đó một chương trình xử lý không cần quan tâm đến các định dạng (format), hình ảnh, quảng cáo trên một trang Web để tìm ra sự liên quan của thông tin.

Thứ hai: Web ngữ nghĩa 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 cột mã quốc gia ‘*zip-code*’ trong cơ sở dữ liệu (database) với trường ‘*zip*’ ở trên giao diện (form) nhập liệu nếu chúng có chung ý nghĩa. Điều này cho phép máy tính theo các đường kết nối 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 giữa các

nguồn dữ liệu, tạo ra các mối liên hệ ngữ cảnh (contextual relationship), điều mà Web hiện tại chưa làm được.

1.2.2 Siêu dữ liệu

Metadata (siêu dữ liệu) dùng để mô tả tài nguyên thông tin. Thuật ngữ “meta” xuất xứ là một từ Hy Lạp dùng để chỉ một cái gì đó có bản chất cơ bản hơn hoặc cao hơn. Một định nghĩa chung nhất và được dùng phổ biến trong cộng đồng những người làm Công nghệ Thông tin: “Metadata là dữ liệu về dữ liệu khác” (Metadata is data about other data) hay có thể nói ngắn gọn là dữ liệu về dữ liệu.

Trong các phạm vi cụ thể, những chuyên gia đưa ra các quan điểm khác nhau về metadata:

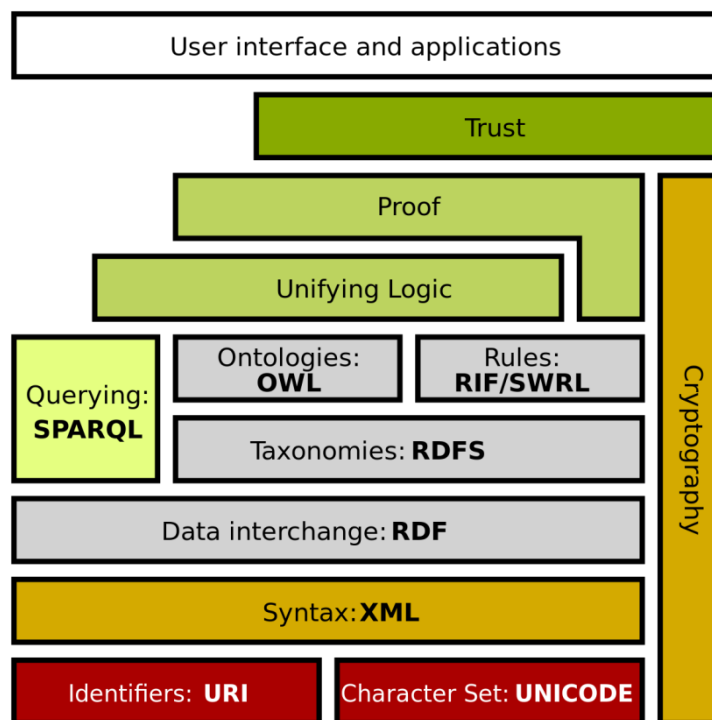
Theo Chris.Taylor giám đốc dịch vụ truy cập thông tin thư viện thuộc trường đại học Queensland¹ thì Metadata là dữ liệu có cấu trúc được dùng để mô tả những đặc điểm của tài nguyên. Một mẫu tin metadata bao gồm một số lượng những phần tử được định nghĩa trước gọi là elements dùng mô tả đặc tính, thông tin tài nguyên. Mỗi elements có thể có 1 hay nhiều giá trị.

Theo tiến sĩ Warwick Cathro thuộc thư viện quốc gia Australia thì một phần tử metadata hay còn gọi là metadata elements mô tả tài nguyên thông tin, hay hỗ trợ truy cập đến một tài nguyên thông tin.

Tóm lại, ta có thể hiểu metadata là thông tin dùng để mô tả tài nguyên thông tin.

1.2.3 Kiến trúc Web ngữ nghĩa

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



Hình 1.2.3. Kiến trúc Web ngữ nghĩa[3]

- *Lớp Unicode & URI*: Bảo đảm việc sử dụng tập kí tự quốc tế và cung cấp phương tiện nhằm định danh các đối tượng trong Web ngữ nghĩa. 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 (ví dụ: <http://www.cadkas.com>). 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.
- *Lớp XML* cùng với các định nghĩa về namespace (vùng tên gọi) và schema (lược đồ) bảo đảm rằng chúng ta có thể tích hợp các định nghĩa Web ngữ nghĩa với các chuẩn dựa trên XML khác.
- *Lớp RDF [RDF] và RDFSchema [RDFS]*: ta có thể tạo các câu lệnh (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 Web ngữ nghĩa.

- *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. Một Ontology (bản thể luận trong logic) 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 hay nhiều lĩnh vực.

- *Lớp Digital Signature*: được dùng để xác định chủ thể của tài liệu (ví dụ: tác giả hay nhan đề của một loại tài liệu).

- *Các lớp Logic, Proof, Trust*: Lớp logic cho phép viết ra các luật (rule) trong khi lớp proof (thử nghiệm) thì hành các luật và cùng với lớp trust (chấp nhận) đánh giá nhằm quyết định nên hay không nên chấp nhận những vấn đề đã thử nghiệm.

1.2.4 Các khái niệm cơ bản của Web ngữ nghĩa

1.2.4.1 Thực thể có tên

Thực thể có tên là con người, tổ chức, nơi chốn và những đối tượng khác được tham khảo đến bằng tên. Thực thể có tên khác về mặt bản chất lẫn ngữ nghĩa với các từ ở chỗ nó được dùng để chỉ các cá thể riêng biệt còn các từ được dùng để chỉ các khái niệm, quan hệ, thuộc tính nói chung. Lấy ví dụ trong câu:

“Hà Nội là thủ đô của nước Việt Nam” thì Hà Nội là thực thể có tên trong khi “thủ đô”, “nước” là các từ vựng.

Trong một tài liệu, các thực thể có tên tạo nên một phần quan trọng trong ý nghĩa của tài liệu đó. Do đó, nhúng ngữ nghĩa vào Web ngoài việc phân tích cú pháp các câu, ngữ nghĩa cho từ vựng đòi hỏi phải có bước xác định ngữ nghĩa cho các thực thể có tên. Từ “ngữ nghĩa” ở đây cũng có ý nghĩa hạn chế, ám chỉ việc một thực thể thuộc lớp hay thuộc tính cụ thể nào đã được định nghĩa từ trước. Đây là một việc phức tạp bởi ngữ nghĩa cho các thực thể có tên đòi hỏi tri thức trong thế giới thực. Nếu một thực thể không có trong cơ sở tri thức thì không thể có kết luận nào khác về thực thể và xem như nó không có ngữ nghĩa.

Bên cạnh vấn đề có liên quan đến cơ sở tri thức đã nói ở trên, trong thực tế một thực thể có thể có nhiều tên khác nhau trong khi các thực thể khác nhau lại có thể có cùng tên. Điều này gây nhập nhằng trong việc suy luận một thực thể chính xác thuộc lớp hay thuộc tính nào. Và do đó, làm ảnh hưởng lớn đến kết quả suy

luận. Trong tình huống này, Web ngữ nghĩa phải có khả năng phân tích ngữ cảnh chứa thực thể một cách tự động, cần thiết có thêm một vài chỉ dẫn trực tiếp từ con người để thu giảm không gian tìm kiếm và tăng mức độ tin cậy đối với kết quả suy luận. Trở lại ví dụ trên, cùng một tên “Hà Nội” nhưng có thể dùng để chỉ thủ đô của Việt Nam cũng có thể được dùng để chỉ một người có cùng tên nào khác.

1.2.4.2 Tài nguyên

Thuật ngữ “tài nguyên” hay “resource” trên Web là một phạm trù rộng lớn dùng để chỉ mọi đối tượng có thể tìm thấy trên Web như khái niệm, từ vựng, thực thể, tính chất và quan hệ giữa các đối tượng. Tài nguyên trên Web cũng chính là dữ liệu của trang Web đó, và là mục tiêu nghiên cứu của Web ngữ nghĩa.[1]

Tài nguyên trên Web là khái niệm rộng hơn thực thể có tên. Và cũng như thực thể có tên, cùng một tài nguyên nhưng có thể được đặt tên khác nhau trong khi có nhiều tài nguyên bản chất khác nhau nhưng lại có cùng tên. Điều đó nảy sinh yêu cầu định danh mỗi tài nguyên bằng một định danh duy nhất. Các tài nguyên khác nhau sẽ có định danh khác nhau. Định danh này được gọi là một URI.

Ví dụ: Cho URI: **www.somesite.com/rdf-syntax-ns#Statement**. Trong đó, **Statement** là một tài nguyên thuộc lớp **rdf-syntax-ns** được tìm thấy trên trang Web **www.somesite.com**. Ký tự „#“ được gọi là một identifier fragment (mẫu định danh).

- Tài nguyên thông tin:

Tài nguyên thông tin chỉ các tài nguyên được biểu diễn trên Web bằng các tài liệu điện tử như tệp văn bản, tệp âm thanh, hình ảnh,... Tài nguyên thông tin thường được biểu diễn bởi chính nó ở dạng các tài liệu điện tử. Ví dụ: các tệp văn bản ở dạng word, các trang web HTML.

- Tài nguyên phi thông tin

Tài nguyên phi thông tin còn gọi là tài nguyên thế giới thực chỉ các thực thể tồn tại bên ngoài thế giới thực như con người, nơi chốn, màu sắc, các khái niệm,... Tài nguyên phi thông tin không thể biểu diễn trực tiếp trên Web mà thường phải

biểu diễn qua một tài nguyên thông tin mô tả nó. Ví dụ: Ông Nguyễn Văn A, Thành phố Hà Nội, khái niệm thủ đô.

1.2.4.3 Định danh tài nguyên

Tài nguyên được định danh bằng URI (Uniform Resource Identifier - định danh tài nguyên thống nhất). URI là một chuỗi các ký tự dùng để định danh tài nguyên trên Internet. Những định danh này có khả năng tương tác với sự biểu diễn của tài nguyên trên mạng sử dụng giao thức cụ thể và phổ biến nhất là HTTP, do đó thường gọi là lược đồ HTTP URI. Có nhiều lược đồ khác ngoài HTTP URI như: ftp, tel, urn, mailto.

Một URI gồm có nhiều thành phần được liệt kê dưới đây:

- Lược đồ URI thường là tên giao thức (chẳng hạn http, ftp, news, mailto). Ở đây thường là lược đồ HTTP URI.
- Tên miền (ví dụ: http://www.portal.ptit.edu.vn).
- Chỉ định thêm cổng (có thể không cần nếu là cổng mặc định của dịch vụ).
- Đường dẫn tuyệt đối trên máy phục vụ của tài nguyên (ví dụ: thumuc/trang).
- Các truy vấn (tuỳ chọn).
- Chỉ định mục con (tuỳ chọn).

Cụ thể hơn:

<http://www.portal.ptit.edu.vn:80/folder/page?timkiem=khóadh#tintuc>

\ /	\ _____ /	\ / \ _____ /	\ _____ /	\ _____ /
Lược đồ	máy chủ	cổng	đường dẫn	truy vấn mục con

Trong ngữ cảnh Linked Data, chúng ta chỉ sử dụng HTTP URI vì đây là cách định danh tốt với hai lý do:

- HTTP URI cung cấp một cách đơn giản để tạo ra những tên duy nhất trên toàn cầu mà không cần quản lý tập trung.

- HTTP URI không chỉ làm việc như một cái tên mà còn có nghĩa là thông tin truy cập về tài nguyên trên Web.

1.3 Ngôn ngữ Cơ cấu mô tả tài nguyên

1.3.1 Ngôn ngữ mô tả tài nguyên RDF

Ngôn ngữ biểu diễn dữ liệu và tri thức là một khía cạnh quan trọng của Semantic Web. Như đã đề cập ở bên trên, XML là nền tảng cơ bản làm nên web ngữ nghĩa, tuy nhiên XML không đủ khả năng để tạo ra ngữ nghĩa trong web. Mặc dù XML cho phép người dùng thêm dữ liệu tùy ý vào cấu trúc tài liệu nhưng nó không đề cập gì đến ngữ nghĩa của tài liệu hàm chứa.

Ngôn ngữ Cơ cấu mô tả tài nguyên - RDF được đề xuất nhằm khắc phục những nhược điểm của XML không thể giải quyết được. Định nghĩa cơ bản của ngôn ngữ RDF là dùng để mã hóa các siêu dữ liệu của các tài nguyên vào một bộ ba (RDF Triple): [chủ ngữ], [vị từ] và [đối tượng]. Ta biết rằng mỗi một thực thể hay khái niệm đều có các thuộc tính, mỗi thuộc tính đều có các giá trị, vì vậy mọi tài nguyên cũng đều có thể được biểu diễn qua ngôn ngữ RDF.

XML cung cấp cú pháp để mã hóa dữ liệu, RDF là một cơ cấu chỉ ra điều gì đó về dữ liệu. RDF cung cấp một mô hình dữ liệu, và một cú pháp đơn giản sao cho các hệ thống độc lập có thể trao đổi và sử dụng nó. RDF được thiết kế sao cho hệ thống máy tính có thể hiểu được và có thể đọc được thông tin, chứ không phải để trình bày dữ liệu cho người dùng. Là một thành phần của Web ngữ nghĩa, được đặt trên XML, RDF sử dụng cú pháp của XML để biểu diễn thông tin, điều này có nghĩa là các tài liệu RDF được viết bằng XML. Ngôn ngữ XML dùng để biểu diễn thông tin trong RDF được gọi là RDF/XML. Thông qua định dạng này, các thông tin trong RDF có thể được trao đổi dễ dàng giữa các hệ thống máy tính cũng như các hệ điều hành hay các ngôn ngữ lập trình ứng dụng khác nhau.

RDF mô tả các nguồn tài nguyên bởi bộ ba [chủ ngữ], [vị từ], [đối tượng]. Một [vị từ] là một khía cạnh, tính chất, thuộc tính, hay mối liên hệ mô tả cho một tài nguyên. Một phát biểu bao gồm một tài nguyên riêng biệt, một thuộc tính được đặt tên, và giá trị thuộc tính cho tài nguyên đó ([đối tượng]). Giá trị này cơ bản có thể là một tài nguyên khác hay một giá trị mang tính nghĩa đen hay dạng chuỗi văn bản tùy ý. [Chủ ngữ] và đối tượng được xác định qua Định danh tài nguyên thống nhất –

URI, chẳng hạn chúng có thể là một liên kết của một trang web. Các [vị từ] cũng được xác định qua URI, do đó bất kì ai cũng có thể định nghĩa ra một khái niệm mới, một thuộc tính mới, bằng cách chỉ cần định nghĩa URI cho chúng. Bởi vì RDF sử dụng URI để biểu diễn các thông tin trong một tài liệu, các URI đảm bảo rằng các khái niệm không chỉ chứa văn bản thuần túy mà nó còn là định danh tài nguyên duy nhất mà tất cả người dùng có thể tìm kiếm được trên mạng. Trong RDF, các URI đóng một vai trò rất quan trọng: Chúng ta có thể tạo ra các (siêu) dữ liệu dựa trên bất kỳ một nguồn tài nguyên nào trên Web, ngữ nghĩa được đưa vào các nguồn tài nguyên Web thông qua các URI, và URI cho phép liên kết giữa các phần tử dữ liệu thông qua các thuộc tính.[1]

1.3.1.1 Mô hình dữ liệu RDF

Sự kết hợp của một nguồn tài nguyên ([chủ ngữ]), một thuộc tính ([vị từ]) và một giá trị của thuộc tính được ([đối tượng]) được đặt trong một Phát_biểu. Ví dụ một câu nói như sau: “Giám đốc của công ty Garden Network là Nguyễn Văn A”. Như vậy, ta có thể biểu diễn bởi một Phát_biểu với các “thông số” sau:

- Chủ ngữ của Phát_biểu RDF là: “Công ty Garden Network”
- Tính chất là: “Giám đốc”
- Đối tượng là: “Nguyễn Văn A”

Mô hình cơ bản của RDF gồm ba bộ phận sau:

- Tài nguyên: là tất cả những gì được mô tả bằng biểu thức RDF
- Thuộc tính: là đặc tính hay quan hệ mô tả tính chất tài nguyên
- Phát_biểu: mỗi phát biểu gồm ba thành phần sau
 - [Chủ ngữ]: địa chỉ hay vị trí tài nguyên muốn mô tả.
 - [Vị từ]: xác định tính chất của tài nguyên.
 - [Đối tượng]: nội dung gán cho thuộc tính.

1.3.1.2 RDF và Cơ sở dữ liệu quan hệ

Trong các Cơ sở dữ liệu quan hệ truyền thống, dữ liệu được lưu dưới dạng các bảng. Trong mỗi bảng, mỗi hàng là một bản ghi không có giới hạn về số lượng các trường. Ví dụ ta có bảng sau:

MSTL	Tên	Tác giả	NXB	Số trang	Số lượng
TL010100	CTDL & GT	Nguyễn A	ĐHQG	200	80
TL010100	Lập trình OOP	Trần B	ĐHQG	220	100

Nếu lưu các dữ liệu trên dưới dạng RDF, đòi hỏi các dữ liệu phải được chia nhỏ để lưu dưới dạng các bộ ba:

MSTL	Tên
TL010100	CTDL & GT
TL010100	Lập trình OOP

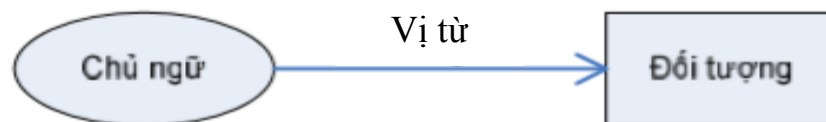
Phát_biểu: { TL010100, Tên, “CTDL & GT”}

So với CSDL quan hệ, cách lưu trữ dưới dạng RDF có những ưu điểm sau:

- Tổ chức dữ liệu đơn giản, đồng nhất nên thông tin dễ dàng chỉnh sửa
- Cấu trúc bộ ba giúp ta dễ truy xuất các thông tin bởi các hệ thống suy luận, tìm kiếm ngữ nghĩa. Cũng nhờ vậy mà những bộ xử lý RDF có thể suy ra những tri thức mới
- Chia sẻ dữ liệu trên mạng nhờ sự đồng nhất

1.3.1.3 Đồ thị RDF

Tập hợp các bộ ba tạo thành đồ thị RDF (đồ thị có hướng). Các nút trong đồ thị là [chủ ngữ] và [đối tượng], các cung trong đồ thị là [Vị từ] và luôn có hướng từ [chủ ngữ] tới [đối tượng]. Dùng đồ thị làm cho thông tin thể hiện rõ ràng và dễ hình dung hơn.



Hình 1.3.1.3. Đồ thị RDF

1.3.1.4 Cú pháp của RDF

Mô hình RDF thể hiện một mô hình ở mức trừu tượng để định nghĩa siêu dữ liệu. Cú pháp RDF được dùng để tạo ra và trao đổi siêu dữ liệu, vì thế 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=' URI-reference '' [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:

Xét phát biểu sau {ketquabongda.html, create-date, “15-05-2020”}

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:ext="http://www.24h.vn/">
4: <rdf:Description rdf:about="http://www.24h.vn/ketquabongda.html">
5: <ext:creation-date>15-05-2020</ext:creation-date>
6: </rdf:Description>
7: </rdf:RDF>

```

Trong đó:

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 mô tả RDF. Từ khóa này xác định tài liệu này được biểu diễn dưới dạng RDF. Tiếp theo là phần khai báo không gian tên XML được sử dụng trong tài liệu, tùy vào nhu cầu và mục đích sử dụng mà ta có thể dùng các không gian tên khác nhau cho từng tài liệu.

Dòng 4, 5, 6: mô tả những mệnh đề RDF. Để mô tả bất kỳ phát biểu nào dạng RDF/XML có thể dùng `rdf:Description`, và `rdf:about`, đây chính là [chủ ngữ] của phát biểu. Thẻ bắt đầu `rdf:Description` trong dòng 4 cho biết bắt đầu mô tả về một tài nguyên, và tiếp tục định danh tài nguyên này dùng thuộc tính `rdf:about` để chỉ ra URI của tài nguyên..

Dòng 5 cung cấp 1 phần tử thuộc tính, với Qname là `exterm:creation-date` như là thẻ của nó. Nội dung của phần tử thuộc tính này là [đối tượng] của Phát_biểu, có giá trị là kiểu chuỗi kí tự “15 tháng 05 năm 2020 “.

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

- Bộ chứa RDF

Để mô tả tập hợp của nhiều đối tượng như một bài báo khoa học được viết bởi nhiều tác giả, danh sách các sinh viên trong một khóa học, v.v... RDF cung cấp nhiều kiểu và nhiều thuộc tính tích hợp sẵn giúp mô tả được những tập như vậy, trong đó có kiểu khai báo “bộ chứa” (container), dùng để lưu danh sách các tài nguyên hoặc các kiểu giá trị (Một bộ chứa là một nguồn tài nguyên chứa những cái gì đó, những cái gì đó được đặt trong bộ chứa được gọi là các thành viên). Các phần tử của một bộ chứa có thể là các tài nguyên URI (có thể là rỗng) hay là các giá trị kiểu chuỗi kí tự. RDF định nghĩa 3 loại đối tượng “bộ chứa”: Bag, Sequence, và Alternative.

<rdf:Bag> là danh sách không có thứ tự của các tài nguyên hoặc các giá trị.

Bag cho phép những giá trị có thể trùng lặp nhau.

Ví dụ:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:group="http://www.abc.vn/group#">
<rdf:Description rdf:about="http://www.caohoc.com.vn/database">
  <group:member>
    <rdf:Bag>
      <rdf:li>Nguyễn Văn A</rdf:li>
      <rdf:li>Lê Thị C</rdf:li>
      <rdf:li>Trần B</rdf:li>
    </rdf:Bag>
  </group:member>
</rdf:Description>
</rdf:RDF>
```

<rdf:Seq> là danh sách có thứ tự của các tài nguyên hoặc các giá trị. Chẳng hạn 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.

Ví dụ:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:group="http://www.abc.vn/group#">
<rdf:Description rdf:about="http://www.semlib.com/database/dbs">
  < paper:format>
    <rdf:Alt>
      <rdf:li>pdf</rdf:li>
      <rdf:li>doc</rdf:li>
      <rdf:li>latex</rdf:li>
    </rdf:Alt>
  </paper:format>
</rdf:Description>
</rdf:RDF>
```

```

<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:group="http://www.abc.vn/group#">
<rdf:Description rdf:about="http://www.caohoc.com.vn/D18TS">
  <class:member>
    <rdf:Seq>
      <rdf:li>Nguyễn Văn A</rdf:li>
      <rdf:li>Lê Thị C</rdf:li>
      <rdf:li>Trần B</rdf:li>
      <rdf:li>Vũ D</rdf:li>
    </rdf:Seq>
  </class:member>
</rdf:Description>
</rdf:RDF>

```

< rdf:Alt> 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 (người dùng chỉ có thể được lựa chọn một trong các giá trị đó).

Ví dụ:

- Tập hợp RDF

Tập hợp RDF (Collection RDF) được sử dụng để mô tả các nhóm chỉ chứa những thành viên đã được đặc tả. Như ta đã biết, trong RDF, một bộ chứa RDF cho thấy các thành viên của nó là các nguồn tài nguyên, nó không cho biết rằng những thành viên nào không được phép. [4]

Một tập hợp được mô tả bởi thuộc tính `rdf:parseType="Collection"`.

Ví dụ:

```

<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns: paper ="http://www.abc.vn/paper#">
<rdf:Description rdf:about=" http://www.semlib.com/database">
<paper:sub rdf:parseType="Collection">
  <rdf:Description
rdf:about="http://www.semlib.com/database/relativedb"/>
<rdf:Description
rdf:about="http://www.semlib.com/database/activedb"/>
<rdf:Description rdf:about="
http://www.semlib.com/database/objectdb"/> <rdf:Description
rdf:about="http://www.semlib.com/database/xmlldb"/> </ paper:sub>
</rdf:Description>
</rdf:RDF>

```

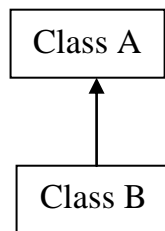
1.3.2 Lược đồ RDF và truy vấn RDF

1.3.2.1 Lược đồ RDF

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, RDF chỉ cho phép định nghĩa các quan hệ, chứ không nói rõ chỉ có thể có các loại quan hệ nào, hay các kiểu đối tượng có thể có trong miền hiện tại. Để làm được điều này, chúng ta sẽ phải sử dụng một phiên bản mở rộng của RDF, gọi là lược đồ RDF (RDF Scheme - RDFS). Lược đồ RDF là một ngôn ngữ bản thể luận dạng đơn nhất, nó cung cấp một khung để mô tả các lớp, thuộc tính của ứng dụng cụ thể. Các lớp trong RDFS giống như các lớp trong lập trình hướng đối tượng, cho phép các tài nguyên được định nghĩa như là một thực thể của lớp, hay lớp con của lớp. [2][3]

Để thực hiện phân chia các lớp và các lớp con, RDFS sử dụng các phần tử như: `rdfs:Class` và `rdfs:subClassOf`.

Ví dụ: Xét lược đồ sau: lớp B là con của lớp A



Hình 1.3.2.1 Quan hệ kế thừa

Sử dụng RDFS để định nghĩa, chú thích các mối quan hệ trên, ta có:

```

<?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#"
  xml:base="http://www.javasample.net/oop#">
  <rdf:Description rdf:ID="classA">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-
      schema#Class"/>
  </rdf:Description>
  <rdf:Description rdf:ID="classB"> <rdf:type
    rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#classA"/>
  </rdf:Description>
</rdf:RDF>
  
```

Khi lớp RDFS là tài nguyên RDF, ta có thể dùng `rdfs:Class` thay vì dùng `rdf:Description`, ví dụ trên được viết lại:

```
<?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#"
  xml:base="http://www.javasample.net/oop#">
  <rdfs:Class rdf:ID="classA"/>
  <rdfs:Class rdf:ID="classB">
    <rdfs:subClassOf rdf:resource="#classA"/>
  </rdfs:Class>
</rdf:RDF>
```

1.3.2.2 Định nghĩa lớp

Các tài nguyên trên Web có thể chia thành các nhóm gọi là lớp. Các thành viên của nhóm được xem như là thể hiện của lớp đó. Thông qua các định danh URI, các tài nguyên được truy xuất và có thể được mô tả bằng các thuộc tính RDF. Thuộc tính `rdf:type` được sử dụng để chỉ ra một tài nguyên là một thể hiện của một lớp.

Phần tử	Mô tả	Lớp con của (<code>rdfs:subClassOf</code>)	Kiểu (<code>rdf:type</code>)
<code>rdfs:Resource</code>	Tất cả các tài nguyên	<code>rdfs:Resource</code>	<code>rdfs:Class</code>
<code>rdfs:Class</code>	Tất cả các lớp	<code>rdfs:Resource</code>	<code>rdfs:Class</code>
<code>rdfs:Literal</code>	Giá trị đơn giản	<code>rdfs:Resource</code>	<code>rdfs:Class</code>
<code>rdfs:Datatype</code>	Kiểu dữ liệu	<code>rdfs:Class</code>	<code>rdfs:Class</code>
<code>rdf:XMLLiteral</code>	Giá trị đơn giản XML	<code>rdfs:Literal</code>	<code>rdfs:Datatype</code>
<code>rdf:Property</code>	Các thuộc tính	<code>rdfs:Resource</code>	<code>rdfs:Class</code>
<code>rdf:Statement</code>	Các khai báo	<code>rdfs:Resource</code>	<code>rdfs:Class</code>
<code>rdf:List</code>	Các danh sách	<code>rdfs:Resource</code>	<code>rdfs:Class</code>
<code>rdfs:Container</code>	Container	<code>rdfs:Resource</code>	<code>rdfs:Class</code>
<code>rdf:Bag</code>	Container không có thứ tự	<code>rdfs:Container</code>	<code>rdfs:Class</code>
<code>rdf:Seq</code>	Container có thứ tự	<code>rdfs:Container</code>	<code>rdfs:Class</code>
<code>rdf:Alt</code>	Container thay thế	<code>rdfs:Container</code>	<code>rdfs:Class</code>
<code>rdfs:Container</code> <code>MembershipProperty</code>	Thuộc tính quan hệ thành viên	<code>rdf:Property</code>	<code>rdfs:Class</code>

Bảng 1.3.2.2. Các lớp trong RDFS

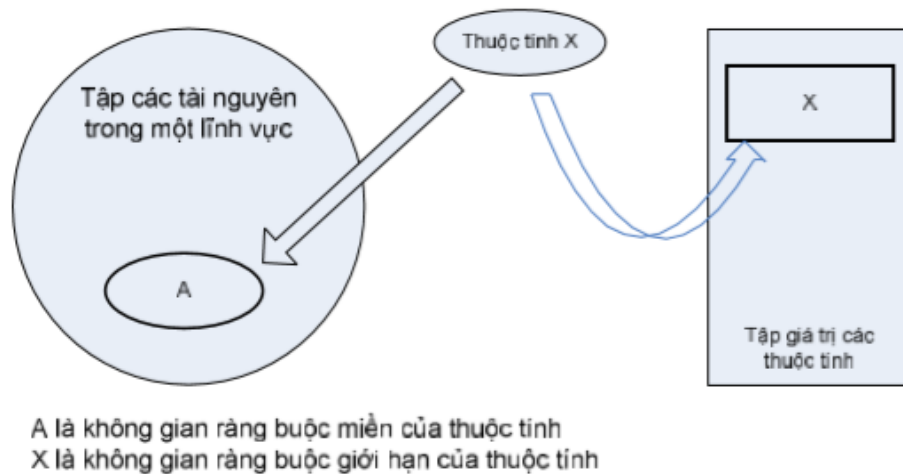
1.3.2.3 Định nghĩa thuộc tính

Mô tả các tính chất của khái niệm. Lược đồ RDF cung cấp một bộ từ vựng để mô tả làm thế nào mà các thuộc tính và lớp có thể được sử dụng cùng nhau trong RDF

Phần tử	Mô tả	rdfs:domain	rdfs:range
rdfs:range	Hạn chế các đối tượng	rdf:Property	rdfs:Class
rdfs:domain	Hạn chế các đối tượng	rdf:Property	rdfs:Class
rdf:type	Thể hiện của	rdfs:Resource	rdfs:Class
rdfs:subClassOf	Lớp con của	rdfs:Class	rdfs:Class
rdfs:subPropertyOf	Thuộc tính con của	rdf:Property	rdf:Property
rdfs:label	Nhãn có nghĩa (dành cho người dùng dễ hiểu)	rdfs:Resource	rdfs:Literal
rdfs:comment	Lời chú giải (dành cho người dùng dễ hiểu)	rdfs:Resource	rdfs:Literal
rdfs:member	Thành viên của container	rdfs:Resource	rdfs:Resource
rdf:first	Phần tử đầu tiên	rdf:List	rdfs:Resource
rdf:rest	Phần còn lại của danh sách	rdf:List	rdf:List
rdf:_1, rdf:_2	Thành viên của Container	rdfs:Container	rdfs:Resource
rdfs:seeAlso	Thông tin xem thêm	rdfs:Resource	rdfs:Resource
rdfs:isDefinedBy	Định nghĩa bởi	rdfs:Resource	rdfs:Resource
rdf:value	Các giá trị có cấu trúc	rdfs:Resource	rdfs:Resource
rdf:object	Đối tượng của khai báo	rdf:Statement	rdfs:Resource
rdf:predicate	Predicate khai báo	rdf:Statement	rdfs:Resource
rdf:subject	Chủ thể của khai báo	rdf:Statement	rdfs:Resource

Bảng 1.3.2.3. Các thuộc tính trong RDFS

Thuộc tính quan trọng nhất được sử dụng trong trường hợp này là rdfs:range và rdfs:domain



Hình 1.3.2.3. Không gian miền và giới hạn của thuộc tính RDFS

`<rdfs:range>` dùng để chỉ giá trị của một thuộc tính là thể hiện của một lớp. Ví dụ ta có phát biểu 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 [đối tượng] là thực thể của lớp `ex:Person`. Tuy nhiên một thuộc tính có thể có nhiều `rdfs:range`, chẳng hạn:

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

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

Thuộc tính `rdfs:range` có thể được sử dụng để chỉ ra giá trị của kiểu chuỗi kí tự

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

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

`<rdfs:domain>` được sử dụng để chỉ rằng một thuộc tính là thuộc tính của một lớp nào đó. Ví dụ như ta 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` khác:

`ex:weight rdfs:domain ex:PC.`

exterms:weight rdfs:domain ex:MotorVehicle.

Các luật suy diễn trong lược đồ RDF

Có 6 trường hợp suy diễn theo các cấp loại lớp, thuộc tính, lớp con, thuộc tính con, miền, giới hạn.

Trường hợp 1: Suy diễn về loại của lớp

IF (c2, subClassOf, c1)

AND (x, type, c2)

THEN (x, type, c1)

Ví dụ:

IF (Man, subClassOf, Person)

AND (Nam, type, Man)

THEN (Nam, type, Person)

Trường hợp 2: Suy diễn về thuộc tính của lớp

IF (p2, subPropertyOf, p1)

AND (x, p2, y)

THEN (x, p1, y)

Ví dụ:

IF (author, subPropertyOf, creator)

AND (Nam, author, KhoaLuan)

THEN (Nam, creator, KhoaLuan)

Trường hợp 3: Suy diễn về lớp con của lớp

IF (c2, subClassOf, c1)

AND (c3, subClassOf, c2)

THEN (c3, subClassOf, c1)

Ví dụ:

IF (ClassB, subClassOf, ClassA)

AND (ClassC, subClassOf, ClassB)

THEN (ClassC, subClassOf, ClassA)

Trường hợp 4: Suy diễn về thuộc tính con của thuộc tính

IF (p2, subPropertyOf, p1)
 AND (p3, subPropertyOf, p2)
 THEN (p3, subPropertyOf, p1)

Ví dụ:

IF (parent, subPropertyOf, ancestor)
 AND (father, subPropertyOf, parent)
 THEN (father, subPropertyOf, ancestor)

Trường hợp 5: Suy diễn trong miền (domain)

IF (p1, domain, c1)
 AND (x, p1, y)
 THEN (x, type, c1)

Ví dụ:

IF (author, domain, Human)
 AND (Nam, author, KhoaLuan)
 THEN (Nam, type, Human)

Trường hợp 6: Suy diễn trong dãy giới hạn (range)

IF (p1, range, c1)
 AND (x, p1, y)
 THEN (y, type, c1)

Ví dụ:

IF (author, range, Document)
 AND (Nam, author, KhoaLuan)
 THEN (KhoaLuan, type, Document)

1.3.2.4 Ngôn ngữ truy vấn RDF

RDF là một cách để mô tả thông tin về các tài nguyên Web một cách linh động. Với lượng thông khổng lồ trên Web cần phải có ngôn ngữ truy vấn các tài liệu RDF một cách nhanh chóng và chính xác. Tổ chức W3C đã phát triển ngôn ngữ truy vấn trong các tài liệu RDF dựa trên cú pháp của ngôn ngữ truy vấn SQL trong CSDL quan hệ. Trong phần này sẽ giới thiệu sơ lược ngôn ngữ truy vấn RDF thông

dụng là SPARQL. Chi tiết về ngôn ngữ SPARQL xem tại <http://www.w3.org/TR/rdf-sparql-query/>[6]

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

- Trích thông tin từ các dạng của URI
- Trích thông tin từ các lược con
- Xây dựng đồ thị RDF mới dựa trên thông tin trong đồ thị truy vấn

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 mẫu điều kiện cho các bộ ba.

Ví dụ:

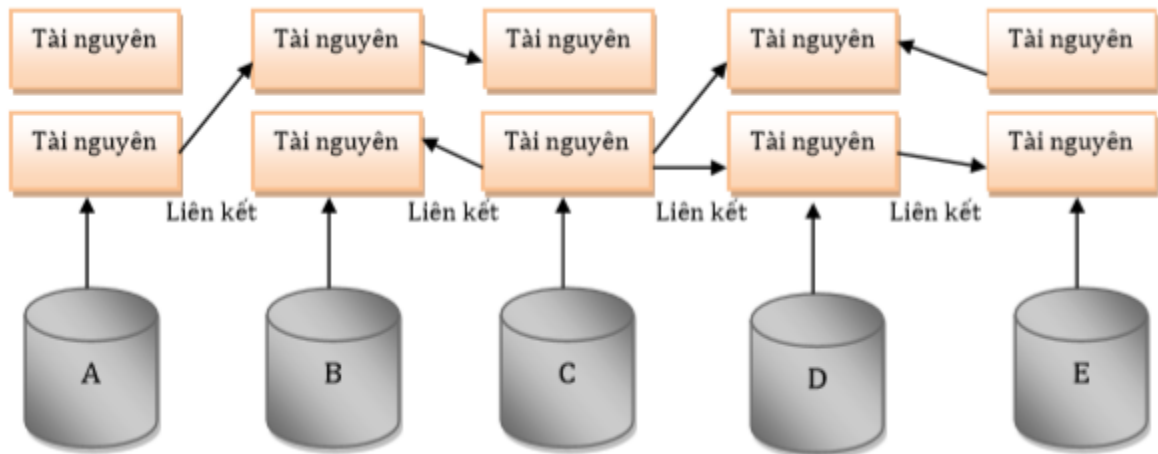
```
PREFIX rdf syntax : <http: //www.w3. org
/1999/02/22-rdf-syntax-ns#>
SELECT ?x
WHERE { ?x rdf syntax : type <ht tp : //example.com/someType> }
```

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 chuỗi kí tự. Biến trong ngôn ngữ truy vấn có giá trị toàn cục. Biến thường bắt đầu với kí tự ‘?’ hay kí tự ‘\$’ để chỉ một biến. SPARQL cung cấp một cơ chế viết tắt. Tiếp đầu ngữ (prefix) dùng để làm gọn các URI trong truy vấn.

1.4 Linked Data

1.4.1 Khái niệm về Linked Data

Trong hoạt động tính toán máy tính, Linked data mô tả một phương thức tạo ra dữ liệu có cấu trúc để có thể liên kết được với nhau và trở nên có ích. Linked data được xây dựng dựa trên các tiêu chuẩn công nghệ Web như HTTP và URI để mở rộng khả năng chia sẻ thông tin theo cách có thể được đọc tự động từ các máy tính hơn là việc chia sẻ nội dung trên các trang web để phục vụ cho người dùng. Điều này cho phép các nguồn tài nguyên khác nhau được kết nối và truy vấn. [1][2]



Hình 1.3.1. Mô hình liên kết dữ liệu trong Web ngữ nghĩa

1.4.2 Quy tắc Linked Data

Tim Berners-Lee liệt kê bốn quy tắc triển khai Linked data trong các thảo luận của mình được tóm gọn trong các dòng sau:

1. Dùng URI để định danh mọi tài nguyên.
2. Sử dụng HTTP URI để các tài nguyên này được tham chiếu và tìm kiếm ("tham chiếu lại") bởi mọi người hay các ứng dụng.
3. Cung cấp thông tin hữu ích về các tài nguyên khi các URI của nó được tham chiếu lại, sử dụng các định dạng tiêu chuẩn như RDF/XML.
4. Bao hàm các liên kết đến liên kết khác, các URL có quan hệ bên trong dữ liệu tìm thấy để phát triển khả năng tìm kiếm các thông tin liên quan khác trên Web.

Lưu ý rằng mặc dù luật quy tắc thứ ba đề cập đến "các định dạng tiêu chuẩn", nó không cần bất cứ tiêu chuẩn đặc tả nào cả, chẳng hạn như RDF/XML.

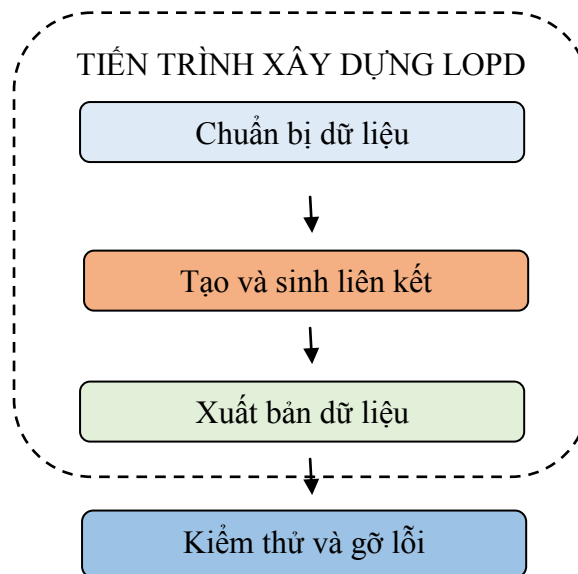
1.5 Tiểu kết chương 1

Chương này trình bày những khái niệm căn bản nhưng lại vô cùng quan trọng, là nền tảng lý thuyết và cốt lõi cho các hướng phát triển cao hơn sau này. Phần đầu chương đã giới thiệu Web ngữ nghĩa là gì, tại sao cần nghiên cứu nó, cách tiếp cận của nó để giải quyết vấn đề Web hiện tại gặp phải. Kiến trúc và các chuẩn được khuyến nghị sử dụng trong Web ngữ nghĩa, các thách thức hiện hữu. Phần sau nói về Linked Data - ý tưởng

đang dần được hoàn thiện trong thực tế: một đám mây dữ liệu toàn cầu, duy nhất, đầy đủ, được công bố rộng rãi và rất quan trọng, mọi thứ được liên kết với nhau.

Chương 2 - ỨNG DỤNG QUẢN LÝ VÀ TRUY XUẤT TÀI NGUYÊN THÔNG TIN TRONG TRƯỜNG ĐẠI HỌC– LINKED OPEN PTIT DATA (LOPD)

2.1 Tiến trình xây dựng ứng dụng LOPD



Hình 2.1. Tiến trình xuất bản Linked Data lên Web

Qua các khái niệm và cũng như các định nghĩa thì tôi đưa ra tiến trình để xây dựng ứng dụng LOPD gồm các tiến trình ở trên: “chuẩn bị dữ liệu, tạo và sinh liên kết, xuất bản dữ liệu, kiểm thử và gỡ lỗi”. Các bước thực hiện trong quy trình ở hình 2.1 yêu cầu phải mật thiết và có sự tương tác mạnh mẽ với nhau để chính xác theo nguyên lý Linked Data.

Trong đó:

- Bước 1:

Chuẩn bị dữ liệu: Bước đầu tiên để xây dựng ứng dụng là chuẩn bị dữ liệu cho ứng dụng. Dữ liệu của ứng dụng chính là các tài nguyên trên Web, do đó chuẩn bị dữ liệu trước hết chính là việc thiết kế không gian tên URI cho tập dữ liệu, có thể hiểu đó là cách tạo ra URI cho dữ liệu của ứng dụng. Và sau đó sẽ tạo mới hoặc chọn để sử dụng lại các bộ từ vựng có sẵn.

- Bước 2:

Tạo liên kết: Liên kết dữ liệu (typed link) chính là điều cốt lõi của Web dữ liệu. Nếu không có liên kết hay chỉ có những liên kết ở mức nội bộ, dữ liệu bị giới hạn trong những phạm vi nhất định. Liên kết dữ liệu cho phép sinh viên, giảng viên hay các ứng dụng duyệt qua những nguồn dữ liệu khác nhau và khám phá thêm dữ liệu, thông tin hữu ích. Trong các ứng dụng Linked Data, đó chính là các liên kết RDF dựa trên các lượt đồ RDF (RDFS)

- Bước 3:

Xuất bản dữ liệu: Việc xuất bản lên Web như thế nào hiệu quả phụ thuộc nhiều yếu tố. Đầu tiên phải kể đến là dữ liệu của chúng ta lớn bao nhiêu? Nếu chỉ xuất bản vài trăm bộ ba RDF, chúng ta có thể cung cấp chúng trong 1 tệp RDF tĩnh và tải lên Web. Nếu dữ liệu nhiều hơn, chúng ta có thể đưa vào trong kho lưu trữ RDF và dùng các công cụ giao tiếp như Pubby để xuất bản chúng. Tiếp đến, chúng ta cần xem xét hiện nay dữ liệu đang được lưu trữ như thế nào? Nếu thông tin hiện đang lưu trữ trong CSDL quan hệ, chúng ta có thể dùng D2R Server để chuyển đổi và xuất bản kiểu khung nhìn RDF. Nếu thông tin là có sẵn thông qua API, bạn có thể cài đặt một bao bọc quanh các API. Nếu thông tin của bạn ở dạng khác như MS Excel, CSV hay BibTeX, bạn cần chuyển qua RDF trước. Và một yếu tố nữa cần quan tâm là sự thay đổi hay cập nhật của dữ liệu. Nếu dữ liệu phải thay đổi thường xuyên, chúng ta có thể thích cách tiếp cận mà sinh ra khung nhìn RDF trên dữ liệu như D2R Server.

- Bước 4:

Kiểm thử và gỡ lỗi: Sau khi xuất bản thông tin dạng Linked Data trên Web, chúng ta cần kiểm tra thông tin có được truy cập đúng chưa. Chúng ta có thể kiểm tra các URI với dịch vụ kiểm tra Linked Data sử dụng Vapour Linked validation service tại địa chỉ <http://vapour.sourceforge.net/>. Dịch vụ này sinh ra các báo cáo chi tiết cách URI hoạt động với các yêu cầu HTTP như thế nào. Thêm vào đó, để xem dữ liệu Linked Data của chúng ta đã hiển thị đúng trong các trình duyệt và các liên kết có hoạt động đúng như mong muốn không, chúng ta có thể nhập các URI

vào trong các trình duyệt Linked Data như Tabulator, Marbles, OpenLink RDF Browser, Disco.

2.2 Jena

2.2.1 Giới thiệu

Jena là một Java framework dùng cho việc xây dựng các ứng dụng web ngữ nghĩa. Cung cấp một môi trường lập trình cho RDF, RDFS and OWL, SPARQL. Bao gồm các công cụ suy diễn từ các luật cơ sở (rule-based inference engine). Open source Phiên bản đầu tiên ra đời 2000 do HP Lab xây dựng Jena 2 ra đời vào 2003 – phiên bản hiện tại là Jena 3.14.0

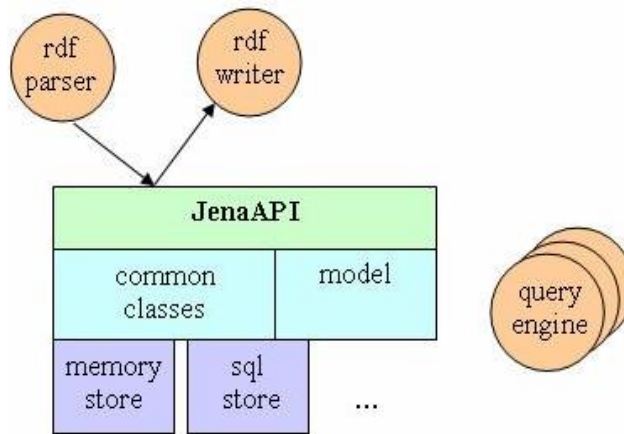
Jena Framework bao gồm:

- RDF API: Giao diện lập trình cho RDF
- OWL API: Giao diện lập trình cho OWL
- Xuất và đọc các file RDF theo dạng RDF/XML, N3 và N-Triples
- Cho phép lưu trữ trong bộ nhớ, lưu trữ cố định trong các tập tin đơn hay trong các hệ quản trị cơ sở dữ liệu (MySQL, HSQLDB, PostgreSQL, Oracle, Microsoft SQL Server)
- Hệ truy vấn SPARQL

2.2.2 Jena API

Kiến trúc của Jena đã được thiết kế để cho phép tích hợp một cách dễ dàng các thành phần xử lý có thể thay thế như bộ phân tích dữ liệu (parser), xuất bản (writer), lưu trữ và truy vấn.

Jena API bao gồm một tập các giao diện Java mô tả các nguồn tài nguyên (resources), các thuộc tính (properties), các phát biểu (statements) và các mô hình (model) dưới dạng khung mô tả tài nguyên RDF (Resource Description Framework)



Hình 2.2.2. Các thành phần của Jena API

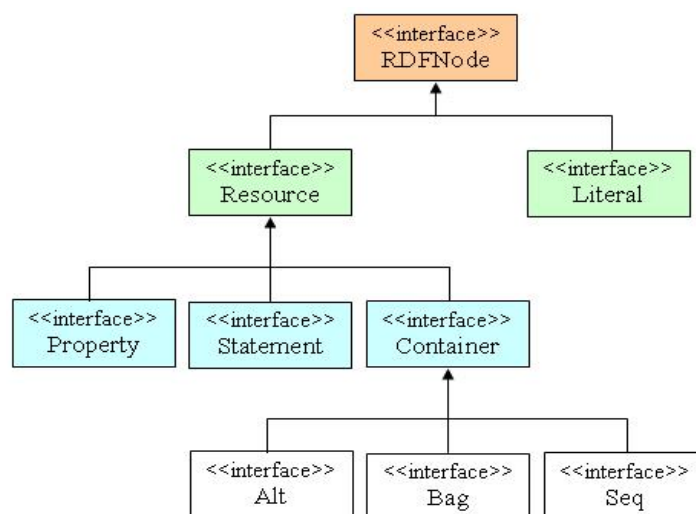
2.2.3 Kiến trúc Jena 3

RDFNode interface: Cung cấp các thành phần của các bộ ba RDF {predicate, subject, object}.

Literal interface: Dùng để mô tả các chuỗi và chuyển đổi một số kiểu của Java như String, Int, và Double. Các đối tượng hiện thực giao diện Property có thể là một predicate trong bộ ba {predicate, subject, object}.

Statement interface: mô tả bộ ba {predicate, subject, object}. Đồng thời nó cũng có thể được dùng như một đối tượng.

Các đối tượng hiện thực các giao diện Container, Alt, Bag, hoặc Seq có thể là đối tượng.



Hình 2.2.3. Kiến trúc Jena 3

2.3 Mô hình hóa thông tin với Jena

2.3.1 Chương trình Hello World! trong Jena

Chương trình sau mô tả cách tạo một mô hình RDF trong Jena, thêm tài nguyên, thuộc tính và xuất ra dưới định dạng Turtle:

```
public class Main {
//Tạo namespace
static private String nameSpace = "http://example.org/test/";
    public static void main(String[] args)
    {
        //Tạo mô hình RDF mới
        Model model = ModelFactory.createDefaultModel();

        //Thêm tài nguyên
        Resource subject = model.createResource(nameSpace + "message");

        //Thêm thuộc tính cho tài nguyên
        Property property = model.createProperty(nameSpace + "says");
        subject.addProperty(property, "Hello World!", XSDDatatype.XSDstring);
        model.write(System.out, "Turtle");
    }
}
```

Kết quả xuất ra dưới định dạng Turtle:

```
<http://example.org/test/message>
    <http://example.org/test/says> "Hello World!" .
```

2.3.2 Tạo mô hình RDF

2.3.2.1 Tạo tài nguyên và thêm thuộc tính

Trong phần này, tôi tạo ra một mô hình RDF trống từ đầu và sau đó thêm thuộc tính cho nó.

```

public class Main {
    // URI tài nguyên
    static private String personURI = "http://somewhere/JohnSmith";
    static private String fullName = "John Smith";

    public static void main(String[] args) {
        // Tạo mô hình RDF
        Model model = ModelFactory.createDefaultModel();

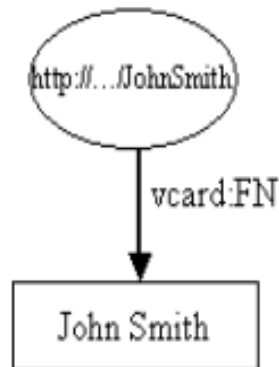
        // Tạo tài nguyên
        Resource johnSmith = model.createResource(personURI);

        // Thêm thuộc tính
        johnSmith.addProperty(VCARD.FN, fullName);

        // Xuất dưới dạng XML
        model.write(System.out);
    }
}

```

Biểu diễn dưới dạng đồ thị:



Hình 2.3.2.1. Biểu diễn phát biểu dưới dạng đồ thị

Kết quả chạy chương trình:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:vcard="http://www.w3.org/2001/vcard-rdf/3.0#">
  <rdf:Description rdf:about="http://somewhere/JohnSmith">
    <vcard:FN>John Smith</vcard:FN>
  </rdf:Description>
</rdf:RDF>
```

2.3.2.2 Thêm mối quan hệ, xuất ra tập tin RDF

Trong ví dụ dưới đây tôi xây dựng một mô hình RDF về gia 1 đình và mối quan hệ giữa các thành viên trong gia đình. Sau đó, xuất thông tin lưu trữ file định dạng RDF.

File *Main.java*

```
public class Main {

    public static void main(String[] args) throws FileNotFoundException {
        FamilyModel theFamily = new FamilyModel();
        File file=new File("E:\\family.rdf");
        FileOutputStream f1=new FileOutputStream(file);
        RDFWriter d = theFamily.model.getWriter("RDF/XML-ABBREV");
        d.write(theFamily.model,f1,null);
    }
}
```

File *FamilyModel.java*

```
public class FamilyModel {
    public Model model;
    static final String familyUri = "http://family/";
    static final String relationshipUri = "http://purl.org/vocab/relationship/";
}
```

```

FamilyModel() {
//Tạo mô hình RDF mới
    model = ModelFactory.createDefaultModel();

//Thêm các thuộc tính xác định mối quan hệ của các thành viên
    Property concua = model.createProperty(relationshipUri,"concua");
    Property chamecua = model.createProperty(relationshipUri,"chamecua");
    Property anhemcua = model.createProperty(relationshipUri,"anhemcua");
    Property vochongcua = model.createProperty(relationshipUri,"vochongcua");
//Thêm các tài nguyên thành viên trong gia đình
    Resource hai = model.createResource(familyUri+"hai");
    Resource mai = model.createResource(familyUri+"mai");
    Resource trung = model.createResource(familyUri+"trung");
    Resource hoa = model.createResource(familyUri+"hoa");
    Resource Nam = model.createResource(familyUri+"Nam");
    Resource Tung = model.createResource(familyUri+"Tung");
    Resource Son = model.createResource(familyUri+"Son");
    Resource Tuyet = model.createResource(familyUri+"Tuyet");
//Thêm các mối quan hệ dưới dạng thuộc tính của đối tượng
    hai.addProperty(anhemcua,mai);
    hai.addProperty(vochongcua,hoa);
    hai.addProperty(chamecua,Nam);
    hai.addProperty(chamecua,Tung);
    mai.addProperty(anhemcua,hai);
    mai.addProperty(vochongcua,trung);
    trung.addProperty(vochongcua,mai);
    hoa.addProperty(vochongcua,hai);
    hoa.addProperty(chamecua,Nam);
    hoa.addProperty(chamecua,Tung);

```

```

//Thêm các Phát biểu về mối quan hệ
Statement statement1 = model.createStatement(Nam,concua,hai);
Statement statement2 = model.createStatement(Nam,concua,hoa);
Statement statement3 = model.createStatement(Nam,anhemcua,Tung);
model.add(statement1);
model.add(statement2);
model.add(statement3);

//Thêm các Phát biểu về mối quan hệ theo dạng mảng Array
Statement statements[] = new Statement[5];
statements[0] = model.createStatement(Tung,concua,hai);
statements[1] = model.createStatement(Tung,concua,hoa);
statements[2] = model.createStatement(Tung,anhemcua,Nam);
statements[3] = model.createStatement(Tung,vochongcua,Son);
statements[4] = model.createStatement(Tung,chamecua,Tuyet);
model.add(statements);

//Thêm các Phát biểu về mối quan hệ theo dạng danh sách List
List<Statement> list = new ArrayList<Statement>();
list.add(model.createStatement(Son,vochongcua,Tung));
list.add(model.createStatement(Son,chamecua,Tuyet));
list.add(model.createStatement(Tuyet,concua,Tung));
list.add(model.createStatement(Tuyet,concua,Son));
model.add(list);
}
}

```

Kết quả xuất ra tập tin *family.rdf* như sau:

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:j.0="http://purl.org/vocab/relationship/">
  <rdf:Description rdf:about="http://family/mai">

```

```

<j.0:vochongcua>
  <rdf:Description rdf:about="http://family/trung">
    <j.0:vochongcua rdf:resource="http://family/mai"/>
  </rdf:Description>
</j.0:vochongcua>
<j.0:anhemcua>
  <rdf:Description rdf:about="http://family/hai">
    <j.0:chamecua>
      <rdf:Description rdf:about="http://family/Tung">
        <j.0:chamecua>
          <rdf:Description rdf:about="http://family/Tuyet">
            <j.0:concua>
              <rdf:Description rdf:about="http://family/Son">
                <j.0:chamecua rdf:resource="http://family/Tuyet"/>
                <j.0:vochongcua rdf:resource="http://family/Tung"/>
              </rdf:Description>
            </j.0:concua>
            <j.0:concua rdf:resource="http://family/Tung"/>
          </rdf:Description>
        </j.0:chamecua>
        <j.0:vochongcua rdf:resource="http://family/Son"/>
      </j.0:anhemcua>
      <rdf:Description rdf:about="http://family/Nam">
        <j.0:anhemcua rdf:resource="http://family/Tung"/>
        <j.0:concua>
          <rdf:Description rdf:about="http://family/hoa">
            <j.0:chamecua rdf:resource="http://family/Tung"/>
            <j.0:chamecua rdf:resource="http://family/Nam"/>
            <j.0:vochongcua rdf:resource="http://family/hai"/>
          </rdf:Description>
        </j.0:concua>
      </rdf:Description>
    </j.0:anhemcua>
  </rdf:Description>
</j.0:anhemcua>

```

```

        </rdf:Description>
        </j.0:concua>
        <j.0:concua rdf:resource="http://family/hai"/>
    </rdf:Description>
    </j.0:anhemcua>
    <j.0:concua rdf:resource="http://family/hoa"/>
    <j.0:concua rdf:resource="http://family/hai"/>
</rdf:Description>
</j.0:chamecua>
<j.0:chamecua rdf:resource="http://family/Nam"/>
<j.0:vochongcua rdf:resource="http://family/hoa"/>
<j.0:anhemcua rdf:resource="http://family/mai"/>
</rdf:Description>
</j.0:anhemcua>
</rdf:Description>
</rdf:RDF>

```

2.3.3 Đọc mô hình RDF

Phần này tôi sử dụng truy vấn SPARQL để xuất ra *?s ?p ?o* của tất cả các tài nguyên trong tập tin *family.rdf*.

```

public class Main {

    public static void main(String[] args) throws FileNotFoundException {
        //Đường dẫn đến tập tin RDF
        final String inputFileName = "E:\\LOPD\\LOPDDemo\\src\\family.rdf";
        Model model = ModelFactory.createDefaultModel();
        InputStream in = FileManager.get().open(inputFileName);
        model.read(in, "");
    }
}

```



```

//Chuỗi truy vấn SPARQL
String queryString =
    "PREFIX quanhe:<http://purl.org/vocab/quanhe/>" + "SELECT * " +
    " { " +
    " ?s ?p ?o. }";

Query query = QueryFactory.create(queryString);
QueryExecution qe = QueryExecutionFactory.create(query, model);
ResultSet results = qe.execSelect();

//Xuất ra bằng ResultSetFormatter
ResultSetFormatter.out(System.out, results, query);
qe.close();
}
}

```

Kết quả xuất ra dưới định dạng Turtle:

```

-----
| s | p | o |
-----
| <http://family/Son> | <http://purl.org/vocab/relationship/vochongcua> | <http://family/Tung> |
| <http://family/Son> | <http://purl.org/vocab/relationship/chamecua> | <http://family/Tuyet> |
| <http://family/Tung> | <http://purl.org/vocab/relationship/concua> | <http://family/hai> |
| <http://family/Tung> | <http://purl.org/vocab/relationship/concua> | <http://family/hoa> |
| <http://family/Tung> | <http://purl.org/vocab/relationship/anhemcua> | <http://family/Nam> |
| <http://family/Tung> | <http://purl.org/vocab/relationship/vochongcua> | <http://family/Son> |
| <http://family/Tung> | <http://purl.org/vocab/relationship/chamecua> | <http://family/Tuyet> |
| <http://family/hoa> | <http://purl.org/vocab/relationship/vochongcua> | <http://family/hai> |
| <http://family/hoa> | <http://purl.org/vocab/relationship/chamecua> | <http://family/Nam> |
| <http://family/hoa> | <http://purl.org/vocab/relationship/chamecua> | <http://family/Tung> |
| <http://family/Tuyet> | <http://purl.org/vocab/relationship/concua> | <http://family/Tung> |
| <http://family/Tuyet> | <http://purl.org/vocab/relationship/concua> | <http://family/Son> |
| <http://family/trung> | <http://purl.org/vocab/relationship/vochongcua> | <http://family/mai> |
| <http://family/mai> | <http://purl.org/vocab/relationship/anhemcua> | <http://family/hai> |
| <http://family/mai> | <http://purl.org/vocab/relationship/vochongcua> | <http://family/trung> |
| <http://family/hai> | <http://purl.org/vocab/relationship/anhemcua> | <http://family/mai> |
| <http://family/hai> | <http://purl.org/vocab/relationship/vochongcua> | <http://family/hoa> |
| <http://family/hai> | <http://purl.org/vocab/relationship/chamecua> | <http://family/Nam> |
| <http://family/hai> | <http://purl.org/vocab/relationship/chamecua> | <http://family/Tung> |
| <http://family/Nam> | <http://purl.org/vocab/relationship/concua> | <http://family/hai> |
| <http://family/Nam> | <http://purl.org/vocab/relationship/concua> | <http://family/hoa> |
| <http://family/Nam> | <http://purl.org/vocab/relationship/anhemcua> | <http://family/Tung> |
-----

```

2.4 Chuyển đổi dữ liệu web ngữ nghĩa

2.4.1 Dữ liệu từ Excel

Trong phần này tôi sử dụng thư viện *jxl.jar* để thao tác với tập tin Excel. Sau đó sử dụng Jena để tạo mô hình dữ liệu RDF. Thêm các tài nguyên và thuộc tính với dữ liệu đọc từ file Excel. Cuối cùng xuất dữ liệu ra tập tin RDF với định dạng XML.

Đoạn code sau mô tả quá trình trên:

```
import jxl Sheet;
import jxl Workbook;

.....

//Tạo mô hình RDF mới
model = ModelFactory.createDefaultModel();

//File Excel chứa dữ liệu các tác giả
File file = new File("C:\\TACGIA.xls");
Workbook wb = Workbook.getWorkbook(file2);
Sheet sheet = wb.getSheet(0);
int rows = sheet.getRows();
int cols = sheet.getColumns();

//Duyệt từng hàng để thêm các tài nguyên
for(int row1 = 0; row1 < rows; row1++)
{
    Resource tentacgia [] = new Resource[100];
    tentacgia [row1]=model.createResource(tacgiaUri
        +sheet.getCell(0, row1).getContents());
}

...

//Khởi tạo một đối tượng tác giả mới
TACGIA_RDF thetacgia = new TACGIA_RDF();
```

```
File file2=new File("C:\\TACGIA.rdf");
FileOutputStream f2=new FileOutputStream(file2);
//Dùng đối tượng RDFWriter của Jena để ghi dữ liệu dưới dạng XML
RDFWriter d2 = thetacgia.model.getWriter("RDF/XML-ABBREV");
d2.write(thetacgia.model2,f2,null);
```

2.4.2 Dữ liệu từ DBF

Với tập tin DBF tôi sử dụng thư viện net.iryndin.jdbf để thực hiện thao tác đọc dữ liệu. Và tương tự như dữ liệu từ Excel, tôi cũng dùng Jena để tạo mô hình, lưu trữ dữ liệu và xuất ra tập tin RDF theo định dạng XML.

Đoạn code sau mô tả quá trình trên:

```
import net.iryndin.jdbf.core.DbfRecord;
import net.iryndin.jdbf.reader.DbfReader;
...
//File DBF chứa thông tin các node từ DBLP
File file = new File("C:\\BRVT.dbf");
Charset stringCharset = Charset.forName("Utf-8");

//Dùng đối tượng DbfReader net.iryndin.jdbf để đọc tập tin DBF
DbfReader reader = new DbfReader(file);
DbfRecord rec = null;

//Thông tin dữ liệu DBF được lưu trữ bởi đối tượng List
List<Map<String,Object>> maps = new ArrayList<Map<String,Object>>();

//Duyệt từng dòng trong file DBF
while ((rec = reader.read()) != null) {
    rec.setStringCharset(stringCharset);

//Lưu thông tin một dòng vào đối tượng Map
```

```

Map<String, Object> map = rec.toMap();
maps.add(map);

//Khai báo các đối tượng để lưu thông tin các Field
Object s1=null,s2=null,s3=null,s4=null;

//Duyệt thông tin trong Map để lấy giá trị cụ thể
for (Map.Entry<String, Object> entry : map.entrySet())
{
    if(entry.getKey().equals("osm_id"))
        s1=entry.getValue();
    else
        if(entry.getKey().equals("name"))
            s2=entry.getValue();
        else
            if(entry.getKey().equals("tourism"))
                s3=entry.getValue();
            else
                if(entry.getKey().equals("tags"))
                    s4=entry.getValue();
}

//Tạo mảng các tài nguyên
Resource TENTACGIA[] = new Resource [10000];

//Thêm tài nguyên
TENTACGIA[i]=model.createResource(tacgiaUri+s1.toString());

//Thêm thuộc tính cho tài nguyên

```

```

TENTACGIA[i].addProperty(FOAF.accountName,String.valueOf(s1));
TENTACGIA[i].addProperty(FOAF.firstName,String.valueOf(s2));
TENTACGIA[i].addProperty(FOAF.nick,String.valueOf(s3));
TENTACGIA[i].addProperty(FOAF.theme,String.valueOf(s4));
}
reader.close();

```

2.5 Chuẩn Dublin Core Metadata

Dublin Core Metadata3 là một chuẩn metadata được nhiều người biết đến và được dùng rộng rãi trong cộng đồng các nhà nghiên cứu, chuyên gia về thư viện số. Dublin Core Metadata lần đầu tiên được xuất năm 1995 bởi Dublin Core Metadata Element Initiative. Dublin là tên một địa danh Dublin, Ohio ở Mỹ nơi đã tổ chức hội thảo OCLC/NCSA Metadata Workshop năm 1995. Core có nghĩa là một danh sách các thành phần cốt lõi dùng mô tả tài nguyên (Element metadata), những thành phần này có thể mở rộng thêm.

Theo [12], tháng 9/2001 bộ yếu tố siêu dữ liệu Dublin Core Metadata được ban hành thành tiêu chuẩn Mỹ, gọi là tiêu chuẩn “The Dublin Core Metadata Element Set” ANSI/NISO Z39.85-2001.

Dublin Core Metadata bao gồm 15 yếu tố cơ bản được mô tả chi tiết trong bảng bên dưới

STT	Yếu tố	Mô tả
1	Title	Nhan đề hay tiêu đề của tài liệu
2	Creator	Tác giả của tài liệu, bao gồm cả tác giả cá nhân và tác giả tập thể
3	Subject	Chủ đề tài liệu đề cập dùng để phân loại tài liệu. Có thể thể hiện bằng từ, cụm từ/(Khung chủ đề), hoặc chỉ số phân loại/(Khung phân loại).
4	Description	Tóm tắt, mô tả nội dung tài liệu. Có thể bao gồm tóm tắt,

		chú thích, mục lục, đoạn văn bản để làm rõ nội dung
5	Publisher	Nhà xuất bản, nơi ban hành tài liệu có thể là tên cá nhân, tên cơ quan, tổ chức, dịch vụ...
6	Contributor	Tên những người cùng tham gia cộng tác đóng góp vào nội dung tài liệu, có thể là cá nhân, tổ chức..
7	Date	Ngày, tháng ban hành tài liệu.
8	Type	Mô tả bản chất của tài liệu. Dùng các thuật ngữ mô tả phạm trù kiểu: trang chủ, bài báo, báo cáo, từ điển...
9	Format	Mô tả sự trình bày vật lý của tài liệu, có thể bao gồm; vật mang tin, kích cỡ độ dài, kiểu dữ liệu (.doc, .html, .jpg, xls, phần mềm....)
10	Identifier	Các thông tin về định danh tài liệu, các nguồn tham chiếu đến, hoặc chuỗi ký tự để định vị tài nguyên: URL (Uniform Resource Locators) (bắt đầu bằng http://), URN (Uniform Resource Name), ISBN (International Standard Book Number), ISSN (International Standard Serial Number), SICI (Serial Item & Contribution Identifier), ...
11	Source	Các thông tin về xuất xứ của tài liệu, tham chiếu đến nguồn mà tài liệu hiện mô tả được trích ra/tạo ra, nguồn cũng có thể là: đường dẫn (URL), URN, ISBN, ISSN...
12	Language	Các thông tin về ngôn ngữ, mô tả ngôn ngữ chính của tài liệu
13	Relation	Mô tả các thông tin liên quan đến tài liệu khác. có thể dùng đường dẫn (URL), URN, ISBN, ISSN...
14	Coverage	Các thông tin liên quan đến phạm vi, quy mô hoặc mức độ bao quát của tài liệu. Phạm vi đó có thể là địa điểm, không gian hoặc thời gian, tọa độ...
15	Rights	Các thông tin liên quan đến bản quyền của tài liệu

Bảng 2.5. Các yếu tố cơ bản của chuẩn Dublin Core Metadata

Trong hệ thống của chúng tôi, những thông tin metadata sau được rút ra từ tài liệu:

- Creator (Author): Thông tin tên của các tác giả tài liệu.
- Title: tựa đề tài liệu.
- Description (Abstract): Tóm tắt nội dung của tài liệu.
- Publisher: Nơi công bố, xuất bản tài liệu.
- Source (DOI): Nơi download tài liệu hoặc địa chỉ chứa thông tin bài báo.
- Date (Year): Năm công bố, xuất bản tài liệu.

2.6 Tiểu kết Chương 2

Chương 2 mô tả các thành phần, công cụ để tạo nên ứng dụng web ngữ nghĩa. Jena API – một thư viện Java dùng để chuyển đổi, lưu trữ dữ liệu web ngữ nghĩa. Cách chuyển đổi dữ liệu từ tập tin Excel, DBF sang dữ liệu RDF. Chương này cũng trình bày về JSP để lập trình web ngữ nghĩa, tạo các giao diện hiển thị kết quả các truy vấn SPARQL trên tập tin RDF.

Chương 3 – PHÁT TRIỂN ỨNG DỤNG LOPD

3.1 Giới thiệu bài toán

3.1.1 Yêu cầu bài toán

Mục tiêu chính của ứng dụng là cải thiện kết quả tìm kiếm và truy xuất nhiều thông tin hơn liên quan cho mục đích sử dụng của người dùng. Đồng thời có thể trả lời được các câu hỏi như “Các bài báo trong năm 2018 của PGS TS Hoàng Hữu Hạnh?”. Vì vậy, chương trình ứng dụng sẽ cung cấp và chia sẻ thông tin theo phương thức:

- Tìm kiếm thông tin: Hệ thống sẽ hỗ trợ tìm kiếm chính xác và gần đúng.

Với tìm kiếm chính xác, người dùng nhập đầy đủ và chính xác từ khóa cần tìm, lúc này hệ thống sẽ hiển thị tất cả lên các thông tin liên quan đến các giảng viên (Các điểm tương đồng, các bài báo cùng đề tài...).

Với tìm kiếm gần đúng, người dùng chỉ cần nhập một cụm từ liên quan đến các thực thể tồn tại trong hệ thống. Kết quả sẽ trả về tên các thực thể có liên quan và người dùng có thể chọn xem chi tiết để biết thông tin.

3.1.2 Phân tích vấn đề

Từ các hạn chế của các thư viện số hay các kho lưu trữ thông tin về các công trình, bài báo khoa học của các giảng viên cũng như các chủ đề tương đồng khi tìm kiếm gây cho việc tìm kiếm khó khăn và không tiếp cận được tối đa về thông tin tìm kiếm liên quan.

Hoặc như muốn sử dụng dữ liệu từ ác nguồn có sẵn để trả lời các câu hỏi như chủ đề này có những tác giả và bài báo nào tương đồng hay không. Hay các tác giả nào có các bài báo về chủ đề nào đó thì web bình thường khó có thể trả lời được câu hỏi này.

Từ những khó khăn trên tôi đưa ra giải pháp đó là sử dụng công nghệ web ngữ nghĩa để giải quyết bài toán. Công nghệ web ngữ nghĩa với đặc điểm lưu trữ dữ liệu dưới định dạng XML và mô hình dữ liệu thông minh nên việc lưu trữ dữ liệu có tính tùy biến cao và hỗ trợ tìm kiếm nhanh, thông tin chất lượng hơn.

3.1.3 Chuẩn bị dữ liệu

3.1.3.1 Dữ liệu từ trường Đại học

Dữ liệu về tài nguyên thông tin trường Đại học bao gồm thông tin khoa học của các giảng viên, tác giả các bài báo, các công trình nghiên cứu khoa học các cấp. Dữ liệu từ file Excel gồm các thông tin các đề tài, giải thưởng, công trình nghiên cứu...

3.1.3.2 Dữ liệu từ DBLP Computer Science Bibliography

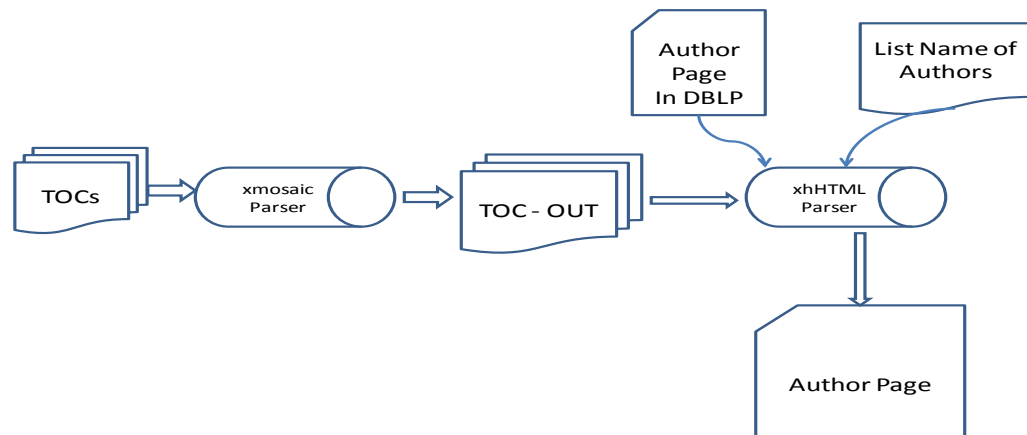
DBLP cung cấp thông tin về chỉ mục các bài báo trong lĩnh vực khoa học máy tính, hệ thống được phát triển bởi trường đại học Universität Trier của Đức. Tính đến tháng 1/2011 DBLP chứa thông tin chỉ mục của 1,5 triệu bài báo trong lĩnh vực khoa học máy tính được thu thập từ các thư viện số, các hội nghị và các tạp chí. Dữ liệu của DBLP được xuất ra các dạng CDF, XML và SQL, người phát triển có thể download các file này từ trên web của hệ thống. [5]

The screenshot shows the DBLP website interface. At the top, there's a header with the DBLP logo, a search bar, and navigation links (home, browse, search, about). Below the header, a banner reads 'Welcome to dblp'. The main content area is divided into several sections: a sidebar on the left with navigation links for authors, journals, conferences, series, and monographs; a central section for the 'dblp blog' with recent news items; and a right sidebar with 'About dblp', 'dblp statistics', and 'dblp tweets'. The 'dblp statistics' section shows the following data:

Category	Count
# of publications	5,065,046
# of authors	2,502,335
# of conferences	5,187
# of journals	1,691

Hình 3.1.3.2. Cơ sở dữ liệu DBLP[5]

DBLP được xây dựng lên dựa vào việc phân tích và rút trích thông tin từ các file đề mục – mục lục. Các file TOCs được tác giả sưu tầm từ các hội nghị, tạp chí.



Hình 3.1.3.3. Hệ thống xây dựng cơ sở dữ liệu DBLP

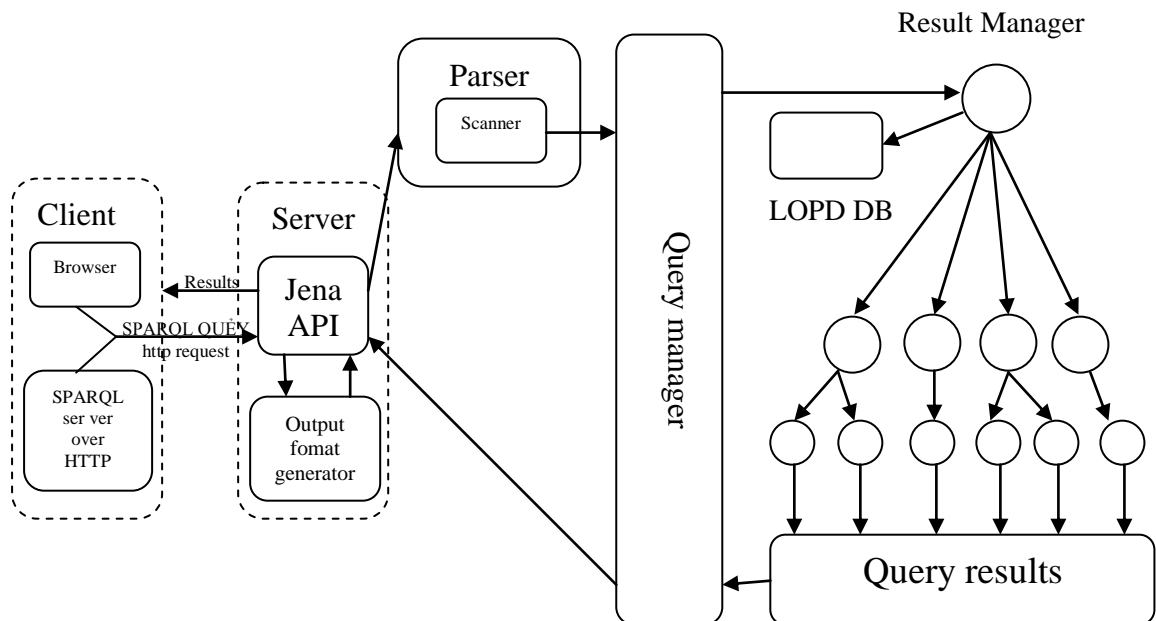
Từ đầu vào là file TOCs được nhập vào bởi tác giả hệ thống sẽ sử dụng các đoạn script và các parser để phân tích và rút trích thông tin các bài báo. Đồng thời với việc sử dụng các dữ liệu có sẵn như danh sách tên của tác giả, thông tin các bài báo đã có, hệ thống sẽ xây dựng lên trang thông tin của tác giả (Author Page). Author Page chứa thông tin về tác giả cũng như thông tin về các bài báo mà tác giả viết hoặc đồng tác giả.

3.2 Giải pháp hệ thống

3.2.1 Kiến trúc hệ thống

LOPD (Linked Open PTIT DATA): là một dịch vụ dùng SPARQL để truy vấn dữ liệu DBLP.

Hình 3.2.1 cho thấy tổng quan về kiến trúc của dịch vụ. Trong một vài trường hợp, Các hệ thống có thể dịch một truy vấn SPARQL tới một dãy cầu nối (lấp) các API, thu thập dữ liệu và trả lại cho người dùng. Dịch vụ hiện hỗ trợ SPARQL truy vấn thông qua HTTP GET. Bộ phân tích cú pháp sử dụng một máy quét để nhận dạng lexemes trong một truy vấn SPARQL và tạo ra các cấu trúc dữ liệu cần thiết bởi Query Manager. Mô-đun này có trách nhiệm tách các truy vấn vào các truy vấn con theo các điều khiển từ xa bởi các API sẵn có. ResultManager xử lý các truy vấn phụ và kết quả của nó để tạo ra bản đồ kết quả cuối cùng.



Hình 3.2.1. Quá trình thực hiện LODP

Như được mô tả ở hình 3.2.1, LODP được thiết kế với kiểu kiến trúc lớp tách biệt, đảm nhận các nhiệm vụ riêng, dựa trên tư tưởng của mô hình truy vấn liên hoàn. Theo đó, kiến trúc của ứng dụng được phân làm bốn lớp:

Lớp giao diện người sử dụng: hiển thị khung nhìn tích hợp cho người dùng cuối.

Lớp mashup: kết hợp nguồn dữ liệu truy vấn với LOPD DB thông qua Query manager.

Lớp tích hợp, tương tác và lưu trữ: đóng vai trò là cầu nối giữa chương trình và dữ liệu liên kết đã được xuất bản (LOPD và DBLP DB).

Lớp dữ liệu xuất bản: là cơ sở dữ liệu liên kết của ứng dụng.

3.2.2 Thiết kế cơ sở dữ liệu

3.2.2.1 Mô tả cấu trúc dữ liệu DBLP

Sau đây là cấu trúc bảng SQL của DBLP được Tiến sĩ Jörg Diederich xây dựng lên từ file XML của DBLP. Dữ liệu này được sử dụng trong hệ thống tìm kiếm Faceted DBLP và được cập nhật mỗi tuần một lần bằng cách sử dụng một đoạn script đọc dữ liệu trực tiếp từ file XML.

Dữ liệu được lưu trữ trong 3 bảng:

- dblp_pub_new: lưu thông tin bài báo.
















Thông tin trong bảng bao gồm: tựa đề bài báo, năm xuất bản, số trang, tên nhà xuất bản, và một số thông tin định danh bài báo trong file XML của DBLP được giải thích chi tiết trong phần mô tả phía dưới. Trong cơ sở dữ liệu này mỗi bài báo có một id riêng được dùng chung cho các bảng có liên quan đến bài báo.

- dblp_author_ref_new: lưu thông tin về tác giả bài báo.

Thông tin trong bảng bao gồm: tên tác giả, tác giả có phải là một người biên tập (editor) hay không. Trong bảng này, những tác giả viết cùng một bài báo thì có id giống nhau và giống id tương ứng của bài báo có trong bảng dblp_pub_new.

- dblp_ref_new: lưu thông tin về các tham chiếu (reference) giữa các bài báo.

Bài báo có id tương ứng trong bảng dblp_pub_new được tham chiếu bởi những bài báo nào được xác định bằng khóa dblp key.

<div>dblp_author_ref_new</div> <table><tr><td></td><td>id</td><td>int(8)</td></tr><tr><td></td><td>author</td><td>varchar(70)</td></tr><tr><td></td><td>editor</td><td>int(1)</td></tr><tr><td></td><td>author_num</td><td>int(3)</td></tr></table>		id	int(8)		author	varchar(70)		editor	int(1)		author_num	int(3)	<div>dblp_pub_new</div> <table><tr><td></td><td>id</td><td>int(8)</td></tr><tr><td></td><td>dblp_key</td><td>varchar(150)</td></tr><tr><td></td><td>title</td><td>longtext</td></tr><tr><td></td><td>source</td><td>varchar(150)</td></tr><tr><td></td><td>source_id</td><td>varchar(50)</td></tr><tr><td></td><td>series</td><td>varchar(100)</td></tr><tr><td></td><td>year</td><td>int(4) unsigned</td></tr><tr><td></td><td>type</td><td>varchar(20)</td></tr><tr><td></td><td>volume</td><td>varchar(50)</td></tr><tr><td></td><td>number</td><td>varchar(20)</td></tr><tr><td></td><td>month</td><td>varchar(30)</td></tr><tr><td></td><td>pages</td><td>varchar(100)</td></tr><tr><td></td><td>ee</td><td>varchar(200)</td></tr><tr><td></td><td>ee_PDF</td><td>varchar(200)</td></tr><tr><td></td><td>url</td><td>varchar(150)</td></tr><tr><td></td><td>publisher</td><td>varchar(250)</td></tr><tr><td></td><td>isbn</td><td>varchar(25)</td></tr><tr><td></td><td>crossref</td><td>varchar(50)</td></tr><tr><td></td><td>titleSignature</td><td>varchar(255)</td></tr><tr><td></td><td>doi</td><td>varchar(255)</td></tr><tr><td></td><td>mdate</td><td>date</td></tr></table>		id	int(8)		dblp_key	varchar(150)		title	longtext		source	varchar(150)		source_id	varchar(50)		series	varchar(100)		year	int(4) unsigned		type	varchar(20)		volume	varchar(50)		number	varchar(20)		month	varchar(30)		pages	varchar(100)		ee	varchar(200)		ee_PDF	varchar(200)		url	varchar(150)		publisher	varchar(250)		isbn	varchar(25)		crossref	varchar(50)		titleSignature	varchar(255)		doi	varchar(255)		mdate	date	<div>dblp_ref_new</div> <table><tr><td></td><td>id</td><td>int(8)</td></tr><tr><td></td><td>ref_id</td><td>varchar(150)</td></tr></table>		id	int(8)		ref_id	varchar(150)
	id	int(8)																																																																																	
	author	varchar(70)																																																																																	
	editor	int(1)																																																																																	
	author_num	int(3)																																																																																	
	id	int(8)																																																																																	
	dblp_key	varchar(150)																																																																																	
	title	longtext																																																																																	
	source	varchar(150)																																																																																	
	source_id	varchar(50)																																																																																	
	series	varchar(100)																																																																																	
	year	int(4) unsigned																																																																																	
	type	varchar(20)																																																																																	
	volume	varchar(50)																																																																																	
	number	varchar(20)																																																																																	
	month	varchar(30)																																																																																	
	pages	varchar(100)																																																																																	
	ee	varchar(200)																																																																																	
	ee_PDF	varchar(200)																																																																																	
	url	varchar(150)																																																																																	
	publisher	varchar(250)																																																																																	
	isbn	varchar(25)																																																																																	
	crossref	varchar(50)																																																																																	
	titleSignature	varchar(255)																																																																																	
	doi	varchar(255)																																																																																	
	mdate	date																																																																																	
	id	int(8)																																																																																	
	ref_id	varchar(150)																																																																																	

Hình 3.2.2.1 Các bảng trong cơ sở dữ liệu DBLP

Chi tiết về các trường trong các bảng được giải thích như sau:

Khóa chính	Trường (Field)	Kiểu (Type)	Chú thích (Comment)
PRIMARY	id	int(8)	Khóa chính.
	dblp_key	varchar(150)	Khóa trong file XML dữ liệu DBLP – Cấu trúc: tên loại tài liệu (book, conf)/ nhà xuất bản/tên tác giả đầu tiên trong tài liệu và năm công bố.
	title	longtext	Tựa đề của tài liệu.
	source	varchar(150)	Tên của hội nghị nơi bài viết được công bố: tên hội nghị, tạp chí, ...
	source_id	varchar(50)	Tham chiếu đến các nguồn xuất bản (phần đầu tiên của dblp_key) - cấu trúc: tên tạp chí/tênviết tắt hội nghị + Năm diễn ra hội nghị.
	series	varchar(100)	Series của tài liệu (chỉ áp dụng với sách và proceeding).
	year	int(4) unsigned	Năm xuất bản của tài liệu.
	type	varchar(20)	Thể loại của tài liệu ví dụ article, proceedings, ...
	volume	varchar(50)	Tập của nơi phát hành tài liệu. (tài liệu thuộc tập mấy trong cuốn phát hành)
	number	varchar(20)	Số tập của nơi phát hành tài liệu. (nơi phát hành có bao nhiêu tập)
	month	varchar(30)	Tháng tài liệu được xuất bản.
	pages	varchar(100)	Tài liệu thuộc trang bao nhiêu trong

			cuốn xuất bản.
	ee	varchar(200)	Địa chỉ URL tới bản điện tử của tài liệu.
	ee_PDF	varchar(200)	Địa chỉ URL tới bản PDF của tài liệu.
	url	varchar(150)	Địa chỉ của tài liệu trong dữ liệu của DBLP (bắt đầu bằng db/ ...).
	publisher	varchar(250)	Tên của nhà xuất bản ; tên trường đối với tài liệu là luận văn; hoặc trang chủ nơi xuất bản.
	Isbn	varchar(25)	<i>International Standard Book Number</i> - mã số tiêu chuẩn quốc tế có tính chất thương mại duy nhất để xác định một quyển sách.
	crossref	varchar(50)	Tham chiếu chéo đến một tài liệu khác. Các tài liệu trong cùng một hội nghị, cùng một năm thì có crossref giống nhau.
	titleSignature	varchar(255)	Tựa đề của tài liệu không bao gồm khoảng trắng và các ký tự đặc biệt.
	doi	varchar(255)	digital object identifier – cung cấp thông tin giúp người dùng có thể tìm được tài liệu trên Internet.
	mdate	Date	Lần cuối cùng chỉnh sửa thông tin tài liệu.

Bảng 3.2.2.1. Thông tin cấu trúc bảng dblp_pub_new

Khóa chính	Trường (Field)	Kiểu (Type)	Chú thích (Comment)
PRIMARY	id	int(8)	Khóa tương ứng với id trong bảng dblp_pub_new.
PRIMARY	author	varchar(70)	Tên của tác giả.
	editor	int(1)	Giá trị trả về giá trị là đúng khi tác giả cũng là một người biên tập (editor).
	author_num	int(3)	Số thứ tự của tác giả (tương ứng trong file gốc XML) . Một bài báo có 5 tác giả thì số tương ứng bắt đầu từ 0, tác giả có số tương ứng như thế nào thì có author_num tương tự vậy.

Bảng 3.2.2.2. Thông tin cấu trúc bảng dblp_author_ref_new

Khóa Chính	Trường (Field)	Kiểu (Type)	Chú thích (Comment)
PRIMARY	id	int(8)	Khóa tương ứng với id trong bảng dblp_pub_new
	ref_id	varchar(150)	Khóa dblp_key của những bài báo được trích dẫn.

Bảng 3.2.2.3. Thông tin cấu trúc bảng dblp_ref

3.2.2.2 Cơ sở dữ liệu hệ thống

Như vậy trong cấu trúc bảng của dblp được trình bày ở trên, hệ thống không chứa dữ liệu về tóm tắt của bài báo (abstract).

Từ cấu trúc này nhóm bổ sung thêm vào cấu trúc những bảng sau để đảm bảo việc có thể cập nhật được dữ liệu mới của DBLP và có thể lưu được các thông tin về chủ đề, cũng như tóm tắt của bài báo

- dbsa_sbj: lưu thông tin về chủ đề của lĩnh vực khoa học máy tính.
- dbsa_pub: lưu thông tin bài báo được thu thập về từ các thư viện số.
- dbsa_pub_in_dblp: bảng lưu thông tin bổ sung của các bài báo trong dữ liệu dblp bao gồm chủ đề, những đường dẫn mở rộng (nơi mà bài báo có thể được tìm thấy – trang cá nhân của tác giả ...).

Khóa Chính	Trường (Field)	Kiểu (Type)	Chú thích (Comment)
PRIMARY	Id	int(8)	Khóa chính của chủ đề
	sbj_name	varchar(150)	Tên của chủ đề.

Bảng 3.2.2.4. Thông tin cấu trúc bảng dbsa_sbj

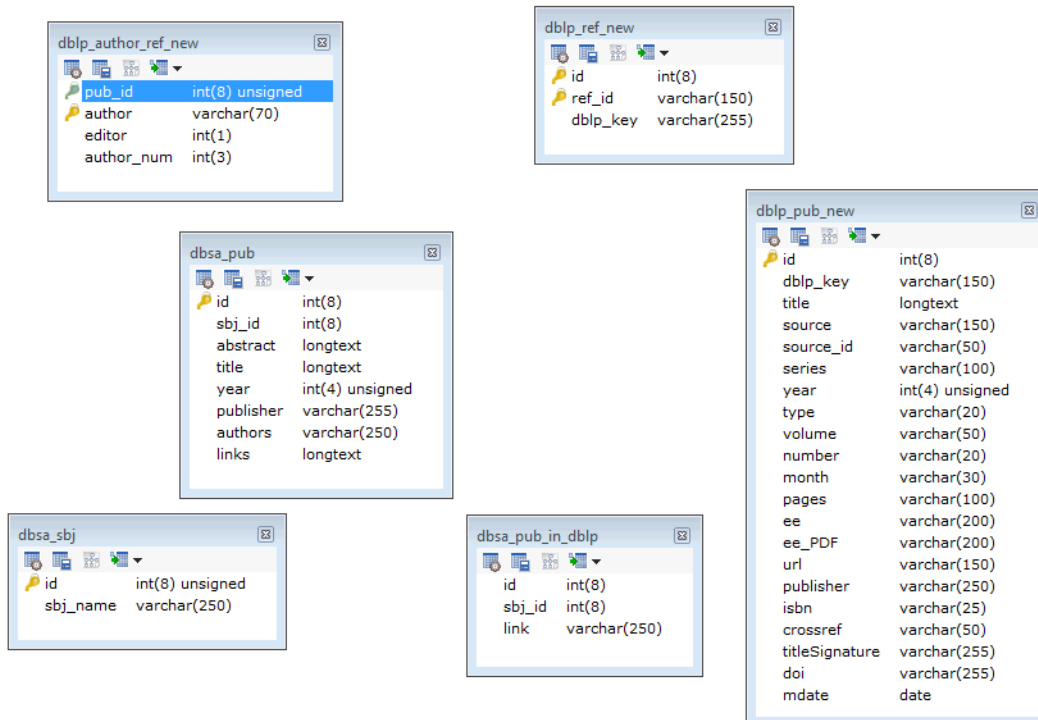
Khóa Chính	Trường (Field)	Kiểu (Type)	Chú thích (Comment)
PRIMARY	Id	int(8)	Khóa tương ứng với id trong bảng dblp_pub_new
	sbj_id	int(8)	Khóa ngoại của dbsa_sbj
	Links	longtext	Những đường dẫn mà người dùng có thể tìm được bài báo.

Bảng 3.2.2.5. Thông tin cấu trúc bảng dbsa_pub_in_dblp

Khóa Chính	Trường (Field)	Kiểu (Type)	Chú thích (Comment)
PRIMARY	Id	int(8)	Khóa chính của bảng
	sbj_id	int(8)	Khóa ngoại của dbsa_sbj
	Abstract	longtext	Tóm tắt của bài báo

	Title	longtext	Tựa đề bài báo
	year	int(4)	Năm xuất bản của tài liệu.
	publisher	varchar(250)	Tên nhà xuất bản
	Authors	Varchar(250)	Tên các tác giả của bài báo, mỗi tên được cách nhau bằng dấu “,”
	Links	longtext	Những đường dẫn mà người dùng có thể tìm được bài báo. Mỗi link khác nhau được cách nhau bằng dấu “,”.

Bảng 3.2.2.6. Thông tin cấu trúc bảng dbsa_pub



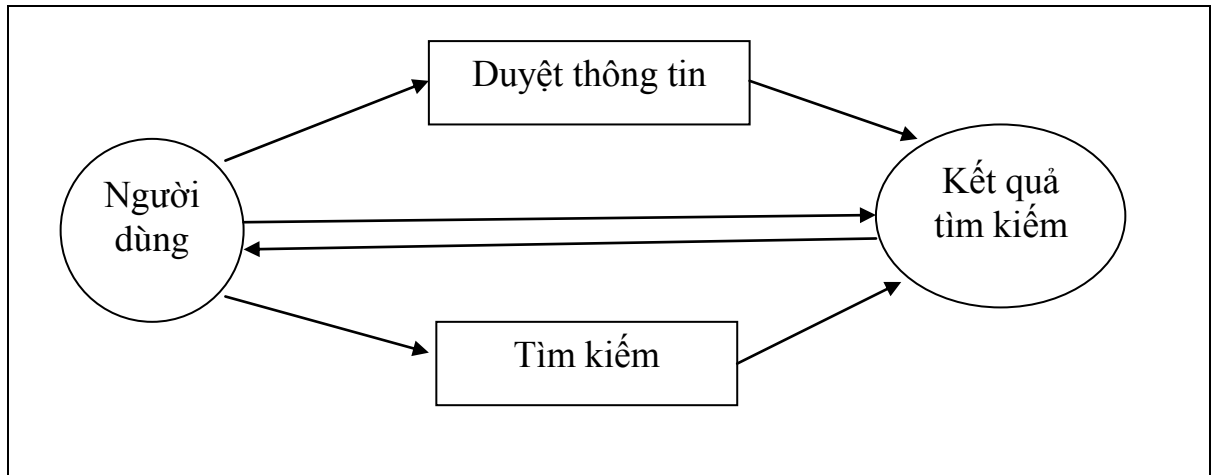
Hình 3.2.2.2. Các bảng trong cơ sở dữ liệu hệ thống

3.3 Xây dựng ứng dụng

3.3.1 Mô tả User case

Phần này sẽ xây dựng mô hình use-case nhằm cung cấp một cách chi tiết về các chức năng cơ bản như tìm kiếm thông tin, xem thông tin về một nhà khảo học cụ

thể (duyệt thông tin) và xem các thông tin chi tiết của nhà khoa học. Mô hình use-case được thể hiện như hình sau:



Hình 3.3.1. Use case hệ thống tìm kiếm thông tin tác giả

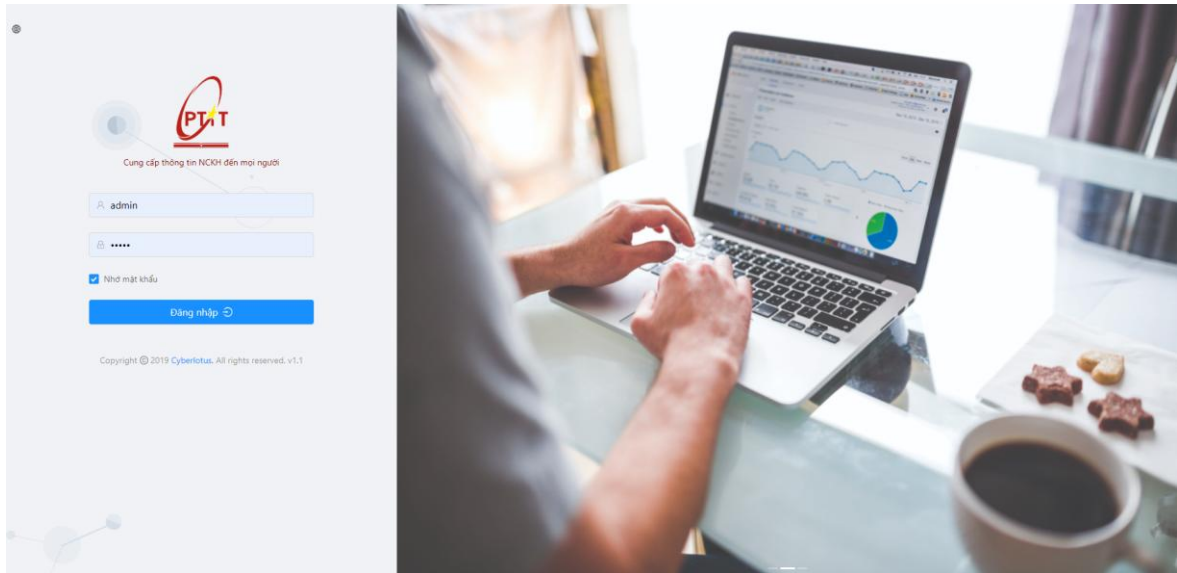
3.3.2 Đặc tả chức năng

- Chức năng tìm kiếm: Chức năng này cho phép người sử dụng tìm thông tin trong hệ thống bằng cách nhập từ khóa thông tin muốn tìm. Hệ thống duyệt file TACGIA.rdf tìm tất cả các tài nguyên liên quan đến từ khóa muốn tìm và trả về kết quả. Khi lấy được thông tin phù hợp sẽ hiển thị

- Chức năng duyệt thông tin: Chức năng này cho phép người sử dụng xem thông tin trong danh sách các tài nguyên liên quan đến từ khóa bằng cách chọn mục thông tin cần xem. Hệ thống tìm các thực thể liên quan đến mục thông tin được chọn và trả về các kết quả cho người sử dụng.

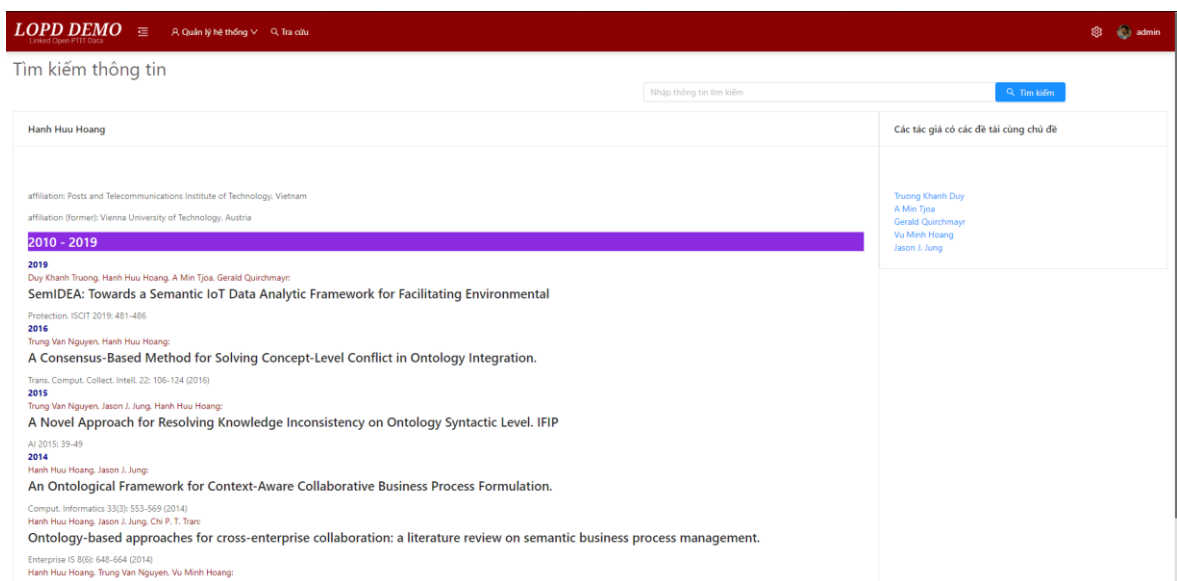
3.3.3 Thiết kế giao diện

3.3.3.1 Giao diện người dùng mặc định



Hình 3.3.3.1. Ứng dụng tìm kiếm thông tin tác giả

3.3.3.2 Kết quả tìm kiếm với tác giả



Hình 3.3.3.2. Kết quả tìm kiếm tác giả PGS.TS Hoàng Hữu Hạnh và các gợi ý các tác giả có các bài báo hay công trình cùng chủ đề

3.4 Tiểu kết Chương 3

Chương 3 đã tổng hợp lại các kiến thức về web ngữ nghĩa để xây dựng ứng dụng minh họa. Trước hết là phần mô tả về bài toán tìm kiếm thông tin tác giả, nhà khoa học. Tiếp theo là giải pháp cho hệ thống tìm kiếm thông tin tác giả: trình bày mô hình của hệ thống, chuyển đổi (RDF) và lưu trữ dữ liệu (Jena). Chương này cũng trình bày cách thức tải dữ liệu, chỉnh sửa dữ liệu. Cuối cùng là ứng dụng minh họa cụ thể về hệ thống tìm kiếm thông tin các tài nguyên thông tin trong trường đại học.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Luận văn đã nghiên cứu và trình bày những kiến thức căn bản về web ngữ nghĩa như kiến thức về RDF, RDF Schema, ngôn ngữ truy vấn dữ liệu web ngữ nghĩa SPARQL, môi trường lập trình web ngữ nghĩa và trực quan hóa dữ liệu.

Qua đó luận văn đạt được một số kết quả như sau:

Về lý thuyết, luận văn đã đi sâu nghiên cứu được nhiều kiến thức về RDF và RDFS, từ đó hiểu được công nghệ web ngữ nghĩa để có thể dựa vào đó triển khai các ứng dụng khác. Trình bày cụ thể phương pháp lập trình với web ngữ nghĩa, cách chuyển đổi dữ liệu cho web ngữ nghĩa. Luận văn cũng đã trình bày cách trích xuất và xử lý dữ liệu từ DBLP Computer Science Bibliography

Về ứng dụng minh họa, với mục tiêu làm rõ thêm lý thuyết, luận văn ứng dụng xây dựng web ngữ nghĩa với các công cụ hỗ trợ. Cụ thể là xây dựng được dữ liệu RDF cơ bản về các tác giả khoa học dựa trên dữ liệu thu thập từ DBLP trích xuất dữ liệu và khai thác các tính năng truy xuất trên một tài liệu có mô tả ngữ nghĩa nhằm chia sẻ tài nguyên thông tin về các bài báo, công trình khoa học và thực hiện tìm kiếm với những kết quả chính xác hơn, đồng thời cũng tận dụng được hết được các nguồn tài nguyên trong hệ thống.

Hướng phát triển:

Xây dựng ontology để hỗ trợ quá trình truy xuất và tìm kiếm thông tin hiệu quả hơn.

Phát triển ứng dụng với chức năng bổ sung và cập nhật thông tin trong ontology.

Sử dụng các công cụ lập trình di động và dữ liệu để tạo ứng dụng truy cập trên thiết bị di động.

DANH MỤC TÀI LIỆ THAM KHẢO

- [1]. Hoàng Hữu Hạnh - Lê Mạnh Thanh, *Giáo trình Web ngữ nghĩa*, NXB giáo dục.
- [2]. David Wood - Marsha Zaidman and Luke Ruth with Michael Hausenblas,
Linked data Structured Data on the Web.
- [3]. Glen Hart – Catherine Dolbear, *Linked Data A Geographic Perspective*.
- [4]. Liyang Yu, *A Developer's Guide to the Web ngữ nghĩa*.

Danh mục website tham khảo

- [5]. <http://www.dblp.uni-trier.de/>
- [6]. <https://www.w3.org/2004/02/skos/>

PHỤC LỤC

DataSample.rdf

```
<?xml version="1.0" encoding="US-ASCII"?>
<rdf:RDF xmlns:dblp="https://dblp.uni-trier.de/rdf/schema-2017-04-18#"
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:dcterms="http://purl.org/dc/terms/"
xmlns:foaf="http://xmlns.com/foaf/0.1" xmlns:bibtex="http://data.bibbase.org/ontology/#">
  <dblp:Person rdf:about="https://dblp.uni-trier.de/pers/h/Hoang:Hanh_Huu">
    <owl:sameAs rdf:resource="https://dblp.org/pers/h/Hoang:Hanh_Huu" />
  </dblp:Person>
  <dblp:Person rdf:about="https://dblp.org/pers/h/Hoang:Hanh_Huu">
    <dblp:primaryFullPersonName>Hanh Huu Hoang</dblp:primaryFullPersonName>
    <dblp:primaryAffiliation>Hue University, Vietnam</dblp:primaryAffiliation>
    <dblp:otherAffiliation>Vienna University of Technology, Austria</dblp:otherAffiliation>
    <dblp:primaryHomepage rdf:resource="http://www.ifs.tuwien.ac.at/~hhhanh/" />
    <dblp:authorOf rdf:resource="https://dblp.org/rec/conf/iscit/TruongHTQ19" />
    <dblp:authorOf rdf:resource="https://dblp.org/rec/journals/tcci/NguyenH16" />
    <dblp:authorOf rdf:resource="https://dblp.org/rec/conf/ifip12/NguyenJH15" />
    <dblp:authorOf rdf:resource="https://dblp.org/rec/journals/cai/HoangJ14" />
    <dblp:authorOf rdf:resource="https://dblp.org/rec/journals/eis/HoangJT14" />
    <dblp:authorOf rdf:resource="https://dblp.org/rec/journals/tcci/HoangNH14" />
    <dblp:authorOf rdf:resource="https://dblp.org/rec/conf/aciids/TranTHL14" />
    <dblp:authorOf rdf:resource="https://dblp.org/rec/conf/iccci/NguyenH14" />
    <dblp:authorOf rdf:resource="https://dblp.org/rec/conf/aciids/Hoang13" />
    <dblp:authorOf rdf:resource="https://dblp.org/rec/conf/aina/HoangNH13" />
    <dblp:authorOf rdf:resource="https://dblp.org/rec/conf/kesamsta/TranH13" />
    <dblp:authorOf rdf:resource="https://dblp.org/rec/conf/kesamsta/NguyenHH13" />
    <dblp:authorOf rdf:resource="https://dblp.org/rec/conf/aina/HoangH12" />
    <dblp:authorOf rdf:resource="https://dblp.org/rec/conf/aciids/HoangTL10" />
    <dblp:authorOf rdf:resource="https://dblp.org/rec/conf/iccci/HoangL09" />
    <dblp:authorOf rdf:resource="https://dblp.org/rec/jourdblpnals/jwe/HoangT07" />
    <dblp:authorOf rdf:resource="https://dblp.org/rec/conf/caise/HoangT06" />
    <dblp:authorOf rdf:resource="https://dblp.org/rec/conf/iiwas/HoangNAT06" />
    <dblp:authorOf rdf:resource="https://dblp.org/rec/conf/otm/HoangAT06" />
    <dblp:authorOf rdf:resource="https://dblp.org/rec/conf/webi/HoangAT06" />
    <dblp:authorOf rdf:resource="https://dblp.org/rec/conf/iiwas/AhmedHKKLLMMNRSNT04" />
    <dblp:coCreatorWith rdf:resource="https://dblp.org/pers/a/Ahmed:Mansoor" />
    <dblp:coCreatorWith rdf:resource="https://dblp.org/pers/a/Andjomshoaa:Amin" />
    <dblp:coCreatorWith rdf:resource="https://dblp.org/pers/h/Hoang:Minh_Tuan_Nguyen" />
    <dblp:coCreatorWith rdf:resource="https://dblp.org/pers/h/Hoang:Vu_Minh" />
    <dblp:coCreatorWith rdf:resource="https://dblp.org/pers/j/Jung:Jason_J" />
    <dblp:coCreatorWith rdf:resource="https://dblp.org/pers/k/Karim:Mohammad_Shuaib" />
    <dblp:coCreatorWith rdf:resource="https://dblp.org/pers/k/Khusro:Shah" />
    <dblp:coCreatorWith rdf:resource="https://dblp.org/pers/l/Lanzenberger:Monika" />
    <dblp:coCreatorWith rdf:resource="https://dblp.org/pers/l/Latif_0001:Khalid" />
    <dblp:coCreatorWith rdf:resource="https://dblp.org/pers/l/Le:Manh_Thanh" />
    <dblp:coCreatorWith rdf:resource="https://dblp.org/pers/m/Michlmayr:Elke" />
    <dblp:coCreatorWith rdf:resource="https://dblp.org/pers/m/Mustofa:Khabib" />
    <dblp:coCreatorWith rdf:resource="https://dblp.org/pers/n/Nguyen:H_T" />
    <dblp:coCreatorWith rdf:resource="https://dblp.org/pers/n/Nguyen:Tho_Man" />
    <dblp:coCreatorWith rdf:resource="https://dblp.org/pers/n/Nguyen:Trung_Van" />
    <dblp:coCreatorWith rdf:resource="https://dblp.org/pers/q/Quirchmayr:Gerald" />
  </dblp:Person>
</rdf:RDF>
```



```

<dblp:coCreatorWith rdf:resource="https://dblp.org/pers/r/Rauber:Andreas" />
<dblp:coCreatorWith rdf:resource="https://dblp.org/pers/s/Schatten:Alexander" />
<dblp:coCreatorWith rdf:resource="https://dblp.org/pers/t/Tjoa:A_Min" />
<dblp:coCreatorWith rdf:resource="https://dblp.org/pers/t/Tran:Chi_P=_T=" />
<dblp:coCreatorWith rdf:resource="https://dblp.org/pers/t/Tran:Phuong=Chi_Thi" />
<dblp:coCreatorWith rdf:resource="https://dblp.org/pers/t/Tran:Tuan_Nhat" />
<dblp:coCreatorWith rdf:resource="https://dblp.org/pers/t/Truong:Duy_Khanh" />
</dblp:Person>
<rdf:Description rdf:about="https://dblp.org/pers/h/Hoang:Hanh_Huu.rdf">
<rdfs:label>provenance information for RDF data of dblp person
&apos;h/Hoang:Hanh_Huu&apos;</rdfs:label>
<dcterms:creator rdf:resource="https://dblp.org" />
<dcterms:isPartOf rdf:resource="https://dblp.org/rdf/dblp.rdf" />
<dcterms:license rdf:resource="https://creativecommons.org/publicdomain/zero/1.0/" />
<dcterms:modified>2020-03-13T22:10:19+0100</dcterms:modified>
</rdf:Description>
<dblp:Person rdf:about="https://dblp.dagstuhl.de/pers/p/Phuong:Nguyen_Duy">
<owl:sameAs rdf:resource="https://dblp.org/pers/p/Phuong:Nguyen_Duy" />
</dblp:Person>
<dblp:Person rdf:about="https://dblp.org/pers/p/Phuong:Nguyen_Duy">
<dblp:primaryFullPersonName>Nguyen Duy Phuong</dblp:primaryFullPersonName>
<dblp:authorOf rdf:resource="https://dblp.org/rec/journals/eswa/PhuongLP19" />
<dblp:authorOf rdf:resource="https://dblp.org/rec/conf/econvn/ThanhPN19" />
<dblp:authorOf rdf:resource="https://dblp.org/rec/conf/kse/LienAP15" />
<dblp:authorOf rdf:resource="https://dblp.org/rec/conf/kse/QuangLP14" />
<dblp:authorOf rdf:resource="https://dblp.org/rec/conf/pricai/PhuongTP08" />
<dblp:authorOf rdf:resource="https://dblp.org/rec/conf/rivf/PhuongP08" />
<dblp:coCreatorWith rdf:resource="https://dblp.org/pers/a/Anh:Nguyen_Xuan" />
<dblp:coCreatorWith rdf:resource="https://dblp.org/pers/l/Lien:Do_Thi" />
<dblp:coCreatorWith rdf:resource="https://dblp.org/pers/n/Ngoc:Bui_Hoang" />
<dblp:coCreatorWith rdf:resource="https://dblp.org/pers/p/Phuong:Tu_Minh" />
<dblp:coCreatorWith rdf:resource="https://dblp.org/pers/q/Quang:Tran_Nhat" />
<dblp:coCreatorWith rdf:resource="https://dblp.org/pers/t/Thang:Le_Quang" />
<dblp:coCreatorWith rdf:resource="https://dblp.org/pers/t/Thanh:Pham_Ngoc" />
</dblp:Person>
<rdf:Description rdf:about="https://dblp.org/pers/p/Phuong:Nguyen_Duy.rdf">
<rdfs:label>provenance information for RDF data of dblp person
&apos;p/Phuong:Nguyen_Duy&apos;</rdfs:label>
<dcterms:creator rdf:resource="https://dblp.org" />
<dcterms:isPartOf rdf:resource="https://dblp.org/rdf/dblp.rdf" />
<dcterms:license rdf:resource="https://creativecommons.org/publicdomain/zero/1.0/" />
<dcterms:modified>2020-01-23T22:56:39+0100</dcterms:modified>
</rdf:Description>
</rdf:RDF>

```