BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC ĐÀ NẴNG

NGUYỄN THỊ THU HẰNG

NGHIÊN CỨU WEB NGỮ NGHĨA VÀ ỨNG DỤNG VÀO XỬ LÝ THÔNG TIN DU LỊCH

Chuyên ngành: Khoa học máy tính
Mã số: 60.48.01

TÓM TẮT LUẬN VĂN THẠC SĨ KỸ THUẬT

ĐÀ NẴNG - 2011

Công trình được hoàn thành tại
ĐẠI HỌC ĐÀ NẴNG

Người hướng dẫn khoa học: PGS.TS. Lê Văn Sơn

Phản biện 1: PGS. TS. Lê Mạnh Thạnh

Phản biện 2: TS. Nguyễn Tấn Khôi

Luận văn được bảo vệ tại Hội đồng chấm Luận văn tốt nghiệp thạc sỹ kỹ thuật tại Đại học Đà Nẵng vào ngày 18 tháng 06 năm 2011.

* *Có thể tìm hiểu luận văn tại:*
- Trung tâm Thông tin - Học liệu, Đại học Đà Nẵng
- Trung tâm Học liệu, Đại học Đà Nẵng.

MỞ ĐẦU

1. Lý do chọn đề tài

Trong những năm qua, sự phát triển của World Wide Web cả về nội dung, người sử dụng và những cải tiến trong công nghệ công cụ tìm kiếm đã hoàn toàn thay đổi cách thức tri thức và thông tin được thu thập và chia sẻ. Thu thập thông tin chưa bao giờ trở nên dễ dàng và rộng mở cho người sử dụng như hiện nay, tuy nhiên vẫn còn một số trường hợp đáng kể các kết quả thu được thông qua một công cụ tìm kiếm có chứa lượng lớn các kết quả không liên quan. Nguyên nhân xuất phát từ chính sự đơn giản của web hiện tại, đã cản trở sự phát triển thông tin của nó. Trong mô hình này, các máy tính chỉ làm nhiệm vụ gửi nhận dữ liệu và thể hiện thông tin dưới dạng thô mà chỉ con người mới đọc hiểu và xử lý được.

XML ra đời và trở thành một công cụ trao đổi dữ liệu 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. Vì thế, một thách thức mới được đặt ra là làm thế nào để khai thác được thông tin trên Web một cách hiệu quả, mà cụ thể là làm thế nào để máy tính có thể trợ giúp xử lý tự động được chúng.

Những nỗ lực phát triển gần đây của công nghệ thông tin và truyền thông nhằm giải quyết những vấn đề này, không chỉ đối với thông tin học thuật mà còn đối với tất cả các dạng dữ liệu có thể chuyển giao trên Web. Trong những nỗ lực phát triển đó, thì mối quan tâm phát triển Web ngữ nghĩa là trọng tâm của Tim Berners-Lee và tổ chức W3C. Theo Lee, “web ngữ nghĩa là sự mở rộng của web thông thường mà trong đó thông tin được định nghĩa rõ ràng sao cho con người và

máy tính có thể cùng làm việc với nhau một cách hiệu quả hơn". Web ngữ nghĩa ra đời là một bước tiến vượt bậc dựa vào khả năng làm việc với thông tin "thông minh" hơn thay vì đơn thuần lưu trữ thông tin. Với sự lớn mạnh và khả năng lưu giữ thông tin ngữ nghĩa, web ngữ nghĩa sẽ trở thành một thế hệ web cho tương lai.

Hiện nay, ở nước ta, lĩnh vực du lịch là một lĩnh vực rất nhiều tiềm năng phát triển, việc ứng dụng xây dựng web ngữ nghĩa về du lịch sẽ rất giúp ích cho việc trao đổi và chia sẻ thông tin du lịch trên Internet. Và đó cũng là lý do tôi chọn đề tài: "*Nghiên cứu Web ngữ nghĩa và ứng dụng vào xử lý thông tin du lịch*".

2. Mục tiêu và nhiệm vụ nghiên cứu

Luận văn nghiên cứu sử dụng công nghệ Web ngữ nghĩa làm ngôn ngữ mô hình hóa cho lĩnh vực du lịch; nghiên cứu về RDF và RDF Schema; nghiên cứu ontology và những lý thuyết có liên quan để có thể hỗ trợ trong việc xây dựng ontology về du lịch.

3. Đối tượng và phạm vi nghiên cứu

a) Đối tượng nghiên cứu

Đối tượng được nghiên cứu cụ thể là: nghiên cứu lý thuyết về Web ngữ nghĩa, RDF và ontology; tìm hiểu các thông tin liên quan về du lịch của Việt Nam; tham khảo các ứng dụng và dự án hiện có trên các tạp chí khoa học và mạng Internet.

b) Phạm vi nghiên cứu

Trong phạm vi giới hạn của đề tài, luận văn nghiên cứu xây dựng tập từ vựng cơ bản về các địa điểm du lịch Việt Nam, tổ chức lưu trữ dữ liệu của ứng dụng với Protégé, và khai thác các tính năng về truy xuất dữ liệu trong ontology.

4. Phương pháp nghiên cứu

Với các mục tiêu trên, luận văn sử dụng phương pháp nghiên cứu

lý thuyết và phương pháp nghiên cứu thực nghiệm để thực hiện đề tài.

5. Ý nghĩa khoa học và thực tiễn của đề tài

a) Ý nghĩa khoa học

Hiểu và vận dụng được các kiến thức về RDF, Ontology, RDF/XML, RDF Schema, OWL ...

Cung cấp giải pháp xây dựng ontology

Xây dựng tập từ vựng cơ bản về các địa điểm du lịch Việt Nam

Khai thác các tính năng đọc/xuất, truy vấn trên một tài liệu có mô tả ngữ nghĩa.

b) Ý nghĩa thực tiễn

Việc nghiên cứu web ngữ nghĩa và ứng dụng vào xử lý thông tin du lịch mà cụ thể là thông tin về các địa điểm du lịch của Việt Nam làm cơ sở nhằm phục vụ việc tra cứu, chia sẻ thông tin về du lịch của Việt Nam. Đồng thời, kết quả nghiên cứu của đề tài làm nền tảng để giải quyết các bài toán xử lý ngữ nghĩa khác trong thực tiễn,

6. Bố cục luận văn

Luận văn được chia làm 3 chương:

Chương 1 trình bày nội dung nghiên cứu tổng quan về web ngữ nghĩa, kiến trúc web ngữ nghĩa và các khái niệm về URI, RDF, RDF Scheme, Ontology. Giới thiệu một số ngôn ngữ xây dựng web ngữ nghĩa và một số ứng dụng của web ngữ nghĩa.

Chương 2 đi sâu vào nghiên cứu RDF, Ontology và những đối tượng cần thiết để xây dựng RDF và Ontology. Đồng thời, trong quá trình nghiên cứu và phân tích về RDF và Ontology sẽ đưa ra giải pháp về ngôn ngữ và công cụ để xây dựng ứng dụng web ngữ nghĩa.

Chương 3 giới thiệu về ứng dụng, phân tích ứng dụng và đề ra giải pháp xây dựng ứng dụng. Tiến hành xây dựng ontology, xử lý dữ liệu, cài đặt ứng dụng và đưa ra một số kết quả thực hiện của ứng dụng.

Chương 1: TỔNG QUAN VỀ WEB NGỮ NGHĨA

1.1. WEB THÔNG THƯỜNG

1.1.1 Các đặc tính của web

Web thông thường chủ yếu là một tập các trang HTML siêu liên kết có thể được xem bởi các trình duyệt trên các thiết bị khác nhau và chủ yếu dành cho con người truy cập và đọc.

1.1.2 Các tiện ích của web

Web làm giảm thời gian truy cập thông tin và số lần giao tác, làm các giao dịch rẻ hơn và gần như không tốn phí cho các truyền thông siêu phương tiện trên thế giới. Nhiều cộng đồng được hưởng lợi từ web.

1.1.3 Các ứng dụng của web

1.1.4 Các hạn chế của web thông thường

Mặc dù web thông thường cung cấp nhiều tiện ích đáng kinh ngạc nhưng nó không cung cấp đủ cấu trúc để hỗ trợ cải tiến máy tính xử lý nội dung. Trong khi người dùng cần thông tin tốt hơn trình bày trên web để cho phép cải tiến được nhiều ứng dụng.

Như vậy, web thông thường chủ yếu là một dịch vụ truyền tải tài liệu tập trung vào nhu cầu người đọc sử dụng các trình duyệt. Tuy nhiên, các công nghệ web thông thường không đủ hỗ trợ nhiều yêu cầu tính toán phức tạp. Công nghệ web mới là cần thiết để cấu trúc thông tin, cải thiện tìm kiếm và đặt ngữ nghĩa vào thông tin.

1.2. GIỚI THIỆU WEB NGỮ NGHĨA

1.2.1. Khái niệm

Web ngữ nghĩa: là một web có dữ liệu thông minh mà máy có thể xử lý được. Ngoài ra còn có thể định nghĩa dữ liệu thông minh là dữ liệu mà độc lập với ứng dụng, được biên soạn, được phân loại và là thành phần của một hệ thống thông tin rộng lớn (ontology).

1.2.2. Lợi ích của việc sử dụng web ngữ nghĩa

Web ngữ nghĩa với thông tin đưa ra đã được xác định rõ ý nghĩa cho phép máy tính và con người hợp tác làm việc tốt hơn.

Web ngữ nghĩa sẽ mang lại cấu trúc cho các nội dung của các trang web có ý nghĩa, tạo ra một môi trường mà các máy tính có thể dễ dàng thực hiện nhiệm vụ phức tạp cho người dùng.

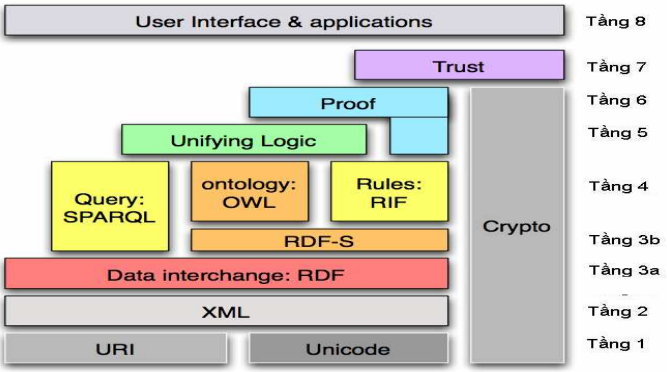
Web ngữ nghĩa đặt tên mỗi khái niệm đơn giản chỉ bằng một URI, nên cho phép bất cứ ai thể hiện các khái niệm mới mà họ nghĩ ra chỉ với nỗ lực tối thiểu.

1.2.3 Tính cần thiết của việc sử dụng web ngữ nghĩa

Web ngữ nghĩa sẽ giải quyết một số vấn đề cơ bản mà các kiến trúc công nghệ thông tin hiện nay phải đối mặt:

- *Thông tin quá tải*
- *Phá vỡ hệ thống Stovepipe*
- *Tích hợp nội dung nghèo nàn*

1.3. KIẾN TRÚC WEB NGỮ NGHĨA



Hình 1. 5 Kiến trúc Web ngữ nghĩa (phiên bản 4)

Đây là phiên bản được hoàn thiện năm 2006 và được giả định là tám tầng thay vì bảy tầng như các phiên bản trước.

1.3.1. Tầng 1 - Unicode và URI

1.3.2. Tầng 2 - XML

1.3.3. Tầng 3 – RDF và RDF Schema

1.3.4. Tầng 4 – Ontology và Rules

1.3.5. Tầng 5 - Logic

1.3.6. Tầng 6 - Proof

1.3.7. Tầng 7 - Trust

1.3.8. Tầng 8 – User Interface & Applications

Đây là một tầng riêng biệt nằm trên cùng trong kiến trúc của hệ thống.

1.4. NGÔN NGỮ CHO WEB NGỮ NGHĨA

Như được mô tả trong mục 1.3, các tầng của kiến trúc web ngữ nghĩa đã được quy chuẩn với các chuẩn đã được W3C đề xuất cũng như cộng đồng nghiên cứu web ngữ nghĩa thống nhất sử dụng trên thực tế. Theo đó, dữ liệu trong web ngữ nghĩa dựa cơ sở trên XML và được mô hình hóa bằng RDF. RDF cũng được chọn là chuẩn trao đổi dữ liệu trong web ngữ nghĩa. Ngôn ngữ ontology được chuẩn hóa là OWL dựa trên cơ sở của RDF(S); ontology cung cấp từ vựng cho việc trao đổi thông tin giữa các ứng dụng và dịch vụ Web. Bên cạnh đó, ngôn ngữ truy vấn SPARQL đã được sử dụng rộng rãi và là khuyến nghị của W3C, tuy nó chưa trở thành chuẩn thật sự.

1.5. CÁC ỨNG DỤNG SỬ DỤNG WEB NGỮ NGHĨA

Web có ngữ nghĩa cho phép tăng cường chức năng, mức độ thông minh và tính tự động hoá của nhiều ứng dụng hiện có. Những lĩnh vực ứng dụng đặc biệt hứa hẹn cho web có ngữ nghĩa là các dịch vụ Web, quản lý tri thức và thương mại điện tử.

Chương 2: CÁC CÔNG NGHỆ XÂY DỰNG WEB NGỮ NGHĨA

2.1. TRUY VẤN DỮ LIỆU RDF

2.1.1. Giới thiệu RDF

RDF cung cấp một framework cho việc mô tả và trao đổi siêu dữ liệu về các tài nguyên trên web. Trong RDF, các tài nguyên trên web được định danh bởi các URI và sử dụng URI để tạo ra các phát biểu về tài nguyên. Để làm cho máy dễ xử lý, RDF kế thừa cú pháp dựa trên XML. Do đó, RDF cung cấp khả năng tương tác giữa các ứng dụng và trao đổi thông tin trên web mà máy có thể hiểu được.

2.1.1.1 Mô hình RDF cơ bản

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

- *Tài nguyên (Resource):* chỉ mọi đối tượng có thể thấy trên web. Các tài nguyên thì luôn được định danh bởi URI.

- *Thuộc tính (Property):* Một thuộc tính là một khía cạnh, đặc điểm, đặc tính hay mối quan hệ cụ thể được dùng để mô tả một tài nguyên.

- *Phát biểu (Statement):* Phát biểu RDF cung cấp một tài nguyên chủ thể, một thuộc tính và một đối tượng. Mỗi phát biểu được biểu diễn theo cấu trúc bộ ba được gọi là “triple” bởi vì nó có ba thành phần cơ bản: Chủ-thể, Thuộc-tính, Đối-tượng (Subject, Predicate, Object).

2.1.1.2 Các cách hiển thị RDF

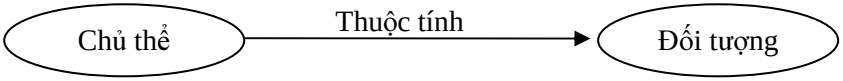
a) N3

N3 hay Notation3 phá vỡ một đồ thị RDF thành các triple riêng của nó, mỗi triple có chứa một chủ thể, một thuộc tính và một đối tượng được tách ra bởi các khoảng trống.

b) Đồ thị RDF

Một tập hợp các RDF triple tạo thành đồ thị RDF. Tập hợp các nút trong đồ thị RDF là tập các chủ thể và các đối tượng trong triple, các

cung trong đồ thị là các thuộc tính. RDF triple được khái niệm hóa bằng biểu đồ như sau:



c) RDF/XML

Phương thức thứ ba sử dụng XML để biểu diễn dữ liệu RDF. Vì có cú pháp cấu trúc mạnh nên XML là một nền tảng tốt cho xử lý tự động trong các tài liệu RDF.

2.1.1.3 Cú pháp RDF/XML

RDF sử dụng XML mã hóa như là cú pháp trao đổi của nó, còn gọi là cú pháp dựa trên XML. RDF/XML là sự kết hợp giữa cú pháp của ngôn ngữ XML và khả năng mô tả tài nguyên thông qua các URI.

2.1.2. RDF Schema

RDF Schema (RDFS) là sự mở rộng của RDF để cho phép mô tả sự phân loại của các lớp và các thuộc tính. RDFS định nghĩa các lớp và các thuộc tính để mô tả các lớp, các thuộc tính và các tài nguyên khác.

2.1.2.1 Lớp trong RDFS

Trong RDFS, các lớp là một nhóm các tài nguyên trên web có liên quan với nhau. Chúng được xác định bằng cách sử dụng tập từ vựng RDF như *rdfs:Class, rdfs:Resource* định nghĩa các nút (node) và *rdf:type, rdfs:subClassOf* định nghĩa các thuộc tính.

2.1.2.2. Thuộc tính trong RDFS

Thuộc tính trong RDFS chính là quan hệ giữa các chủ thể và đối tượng trong RDF. Thuộc tính hay được sử dụng là *rdfs:range*, *rdfs:domain, rdfs:subClassOf* và *rdfs:subPropertyOf*. Tương tự như các mô tả lớp, các thuộc tính được định nghĩa bởi thẻ *rdf:Property*.

2.1.3. Lưu trữ dữ liệu RDF

Một vài hệ thống được phát triển cho việc lưu trữ và truy vấn dữ

liệu RDF, ví dụ như Sesame và RDFSuite.

2.1.4. Truy vấn dữ liệu RDF

2.1.4.1. Truy vấn dữ liệu RDF

Các tài liệu RDF và RDF Schema có thể được xem xét truy vấn ở ba cấp độ trừu tượng khác nhau:

1. *Ở cấp độ cú pháp*, chúng là các tài liệu XML.

2. *Ở cấp độ cấu trúc*, chúng bao gồm một tập các triple.

3. *Ở cấp độ ngữ nghĩa*, chúng thiết lập một hoặc nhiều đồ thị với các thành phần ngữ nghĩa được định nghĩa trước.

2.1.4.2. SPARQL

SPARQL là một ngôn ngữ cho phép truy vấn các triple từ một cơ sở dữ liệu RDF (hoặc từ một kho lưu trữ triple). SPARQL truy vấn bằng cách định nghĩa một khuôn mẫu cho các triple tương ứng.

2.2. BIỂU DIỄN ONTOLOGY

2.2.1. Tổng quan về Ontology

Ontology là "một biểu diễn sự khái niệm hóa chung được chia sẻ" của một miền hay lĩnh vực nhất định. Nó cung cấp một bộ từ vựng chung bao gồm các khái niệm, các thuộc tính quan trọng và các định nghĩa về các khái niệm và các thuộc tính này.

Có bốn thành phần ontology cơ bản:

- Các thực thể (individuals)
- Các lớp (classes)
- Các thuộc tính (attributes)
- Các quan hệ (relations)

** Phân loại Ontology:*

Sự phân loại ontology có thể được tạo ra theo chủ đề của sự khái niệm hóa và được tổng kết thành bốn loại như sau:

- Top-level Ontology: diễn tả các khái niệm tổng quan trừu tượng

- Domain Ontology: lấy tri thức từ trong những lĩnh vực xác định
- Task Ontology: lấy tri thức về một tác vụ riêng biệt
- Application Ontology

2.2.2. Phương thức xây dựng Ontology

Một số nguyên tắc cơ bản của việc xây dựng ontology thông qua các công đoạn sau đây:

- ✓ *Xác định miền và phạm vi của ontology*
- ✓ *Xem xét việc kế thừa các ontology có sẵn*
- ✓ *Liệt kê các thuật ngữ quan trọng trong ontology*
- ✓ *Xây dựng các lớp và cấu trúc lớp phân cấp*
- ✓ *Xác định các thuộc tính và quan hệ cho lớp*
- ✓ *Xác định giới hạn của các thuộc tính và quan hệ của lớp*
- ✓ *Tạo các thực thể cho lớp*

2.2.3. Ngôn ngữ biểu diễn Ontology

Ngôn ngữ ontology chuẩn được phát triển gần đây nhất là OWL từ tổ chức W3C. Xây dựng dựa trên RDF và RDFS, OWL định nghĩa các kiểu quan hệ có thể được thể hiện trong RDF bằng cách sử dụng một tập từ vựng XML để chỉ ra các phân cấp và mối quan hệ giữa các tài nguyên khác nhau. OWL được phát triển bởi nó có nhiều tiện lợi để biểu diễn ý nghĩa và ngữ nghĩa hơn so với XML, RDF và RDFS, và vì OWL ra đời sau các ngôn ngữ này nên nó có khả năng biểu diễn các nội dung trên web mà máy có thể hiểu được.

Các phiên bản của OWL

OWL được chia làm ba phiên bản ngôn ngữ con: OWL Lite, OWL DL và OWL Full.

Lựa chọn ngôn ngữ con để sử dụng

Một cách cơ bản, ngôn ngữ con được sử dụng nên có nhiều biểu cảm cần thiết và càng ít phức tạp càng tốt.

Các thành phần của các ontology OWL

- *Lớp (Classes):* được sử dụng để tạo ra các nhóm có cùng đặc điểm.
- *Thực thể (Individuals):* là các thành viên của lớp, làm rõ hơn về lớp đó.
- *Thuộc tính (Properties):* liên kết hai đối tượng với nhau. Mỗi thuộc tính có thể có một thuộc tính nghịch đảo tương ứng.

Các kiểu thuộc tính OWL

Có hai kiểu thuộc tính chính là thuộc tính Object và Datatype. Thuộc tính Object là các mối quan hệ giữa hai thực thể, liên kết từ thực thể này đến thực thể kia. Thuộc tính Datatype mô tả mối quan hệ giữa các thực thể và giá trị dữ liệu. OWL cũng có kiểu thuộc tính thứ ba là thuộc tính Annotation được sử dụng để thêm thông tin vào lớp, vào thực thể hoặc thực thể khác thuộc hai kiểu trên object/datatype.

Domain và Range của thuộc tính

Thuộc tính có thể có một domain và một range cụ thể. Các thuộc tính liên kết các thực thể thuộc domain đến các thực thể thuộc range. Nhìn chung, domain của một thuộc tính là range của thuộc tính nghịch đảo của nó và ngược lại.

2.3. CÔNG CỤ VÀ PHẦN MỀM XÂY DỰNG WEB NGỮ NGHĨA

2.3.1. Protégé

Protégé là một bộ phần mềm mã nguồn mở miễn phí được viết bằng Java và thao tác các ontology dưới nhiều định dạng khác nhau bao gồm cả RDF(S), OWL và XML Schema.

2.3.2. OwlDotNetApi

OwlDotNetApi được viết bằng C# và cung cấp một API cho các ứng dụng .NET dựa trên phân tích cú pháp RDF Drive. API này sử dụng các mô hình dữ liệu cơ bản từ RDF Drive để xây dựng một đồ thị được liên kết trực tiếp từ các ontology OWL.

Chương 3: XÂY DỰNG ỨNG DỤNG MINH HỌA XỬ LÝ THÔNG TIN DU LỊCH

3.1. GIỚI THIỆU ỨNG DỤNG

Với những kiến thức tìm hiểu được về web ngữ nghĩa trên, luận văn nghiên cứu sử dụng công nghệ web ngữ nghĩa vào xây dựng một ứng dụng minh họa nhằm phục vụ cho việc lưu trữ, tìm kiếm, truy vấn thông tin về các địa điểm du lịch Việt Nam và đồng thời để hiểu rõ hơn về lĩnh vực web ngữ nghĩa này.

Mục tiêu chính của ứng dụng là để cải thiện kết quả tìm kiếm và truy xuất nhiều thông tin liên quan hơn cho mục đích sử dụng của người dùng. Vì vậy, chương trình ứng dụng sẽ cung cấp và chia sẻ các thông tin về các địa điểm du lịch theo hai phương thức:

- *Tìm kiếm thông tin*

Hệ thống sẽ hỗ trợ tìm kiếm chính xác và tìm gần đúng. Với tìm kiếm chính xác, người dùng nhập đầy đủ và chính xác từ khóa cần tìm, lúc này có hai trường hợp xảy ra:

+ Nếu từ khóa cần tìm là tên của một điểm đến du lịch thì kết quả trả về là tất cả các thực thể có quan hệ với địa điểm này

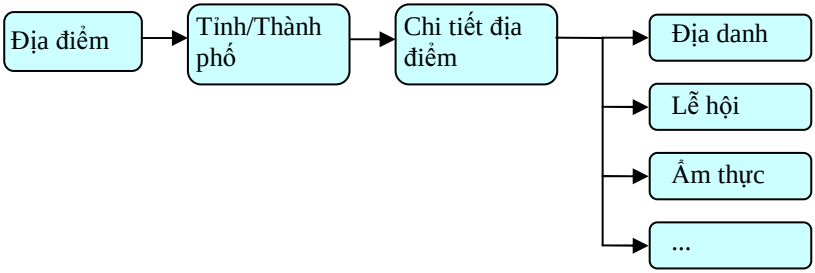
+ Ngược lại, nếu từ khóa cần tìm không phải là tên của một điểm đến du lịch thì kết quả trả về là thông tin về các thực thể liên quan nếu có tồn tại trong ứng dụng.

Với tìm kiếm gần đúng, người dùng chỉ cần nhập một phần trong chuỗi từ khóa cần tìm, kết quả trả về là tập các tên thực thể mà trong đó có chứa một phần giống với chuỗi gõ vào. Sau đó người dùng có thể chọn chức năng hiển thị thông tin chi tiết để xem.

- *Duyệt cây phân cấp dựa trên ontology*

Giao diện cây phân cấp được tổ chức theo thứ tự như hình 3.1 sau:

Địa điểm - Tỉnh/Thành phố - Chi tiết địa điểm



Hình 3. 1 Minh họa cây phân cấp duyệt thông tin

Trước tiên, người sử dụng chọn mục điểm đến, hệ thống sẽ hiển thị các tỉnh/thành phố. Tiếp theo có thể chọn một tỉnh/thành phố để xem các thông tin về địa điểm đã chọn này. Hệ thống sẽ hiển thị ra thông tin chi tiết về địa điểm đã chọn và các thông tin khác liên quan đến địa điểm. Kèm theo đó, hệ thống sẽ cung cấp một danh sách các thông tin đặc trưng về địa điểm đã chọn.

3.2. HƯỚNG TIẾP CẬN VÀ GIẢI PHÁP

**** Giải pháp kỹ thuật***

Chương trình ứng dụng sẽ thực hiện xử lý thông tin thông qua các công nghệ web ngữ nghĩa hiện có và các giải pháp đã được đề cập:

- Môi trường phát triển: Visual Studio
- Thi hành trên IE/Firefox
- Ngôn ngữ sử dụng: C# trên nền ASP.net
- Chương trình sử dụng công cụ Protégé, OwlDotNetApi.

Việc lựa chọn nền tảng .NET như là công cụ phát triển chính vì nó cho phép khả năng tương tác giữa nhiều ngôn ngữ.

**** Thiết kế ontology***

Trước hết không có giải pháp nào là bắt buộc về mô hình ontology. Các hướng tiếp cận khác nhau đều khả thi và giải pháp tốt nhất là phụ thuộc vào ứng dụng.

Một trong những ngôn ngữ ontology phổ biến hiện nay là Web Ontology Language (OWL). Có những hình thái OWL khác nhau (OWL Full, OWL Lite và OWL DL) mà khác nhau chủ yếu ở tính phức tạp và tính biểu cảm. Cách tiếp cận trình bày trong báo cáo này dựa trên OWL DL (Logic mô tả).

Đối với thiết kế ontology, Protégé (Protégé-OWL) được lựa chọn để sử dụng bởi nó là công cụ mạnh mẽ và phổ biến nhất hiện thời cho mục đích này.

Trong ứng dụng này, ontology sẽ được tích hợp vào một thư mục ở máy cục bộ để tiện cho việc minh họa ứng dụng. Tuy nhiên cũng có thể lấy các ontology này trực tiếp từ Internet (nếu có). Các ontology thường lưu dưới dạng tập tin có đuôi: .rdf, .rdfs, .owl, .daml, .xml, ... Ontology của ứng dụng được lưu với tập tin .owl.

**** Quản lý các tài nguyên của ứng dụng***

Quản lý các tài nguyên của ứng dụng bao gồm việc thêm, sửa và xóa thông tin các tài nguyên. Chức năng này chưa được triển khai trong ứng dụng nên sẽ được thực hiện bằng cách thao tác trực tiếp với ontology của ứng dụng thông qua công cụ Protégé.

**** Sử dụng thư viện mã nguồn mở OWLDotNetAPI***

Để sử dụng OwlDotNetApi, trước hết phải import thư viện DLL được biên dịch sẵn vào Visual Studio project. Bước tiếp theo là tạo một đồ thị OWL từ tập tin ontology:

IOwlParser parser = new OwlXmlParser();

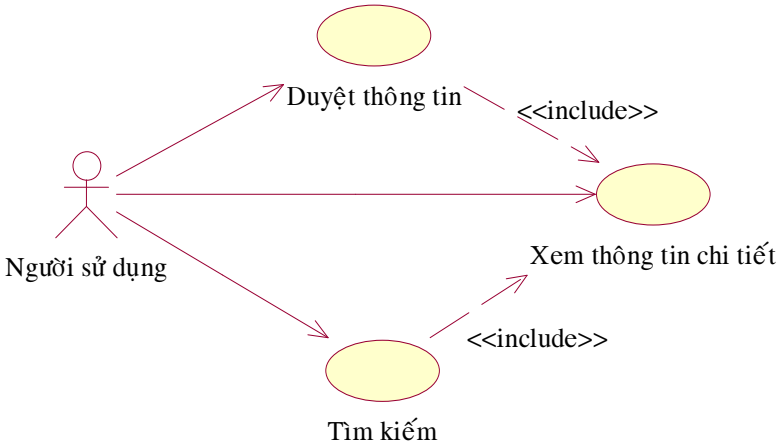
IOwlGraph graph = parser.ParseOwl(địa_chỉ_chứa_tập_tin_owl);

Một khi một đồ thị OWL được tạo ra, các phương thức sẽ được gọi ra để thực thi ứng dụng. Để truy xuất thông tin chi tiết về đồ thị thì có thể sử dụng các thuộc tính ChildEdges và ParentEdges.

3.3. PHÂN TÍCH ỨNG DỤNG

3.3.1. Mô hình use-case

Dựa trên mô tả về ứng dụng, 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 địa điểm cụ thể (duyệt thông tin) và xem các thông tin chi tiết của địa điểm. Mô hình use-case được thể hiện như hình 3.2 sau:



Hình 3. 2 Minh họa mô hình usecase của ứng dụng

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 vào ontology tìm tất cả các URIs liên quan đến từ khóa muốn tìm và trả về kết quả.

- Chức năng duyệt thông tin:

Chức năng này cho phép người sử dụng xem thông tin phân cấp trong hệ thống bằng cách chọn mục thông tin cần xem. Hệ thống vào ontology tìm các thực thể hoặc các phân cấp con liên quan đến mục thông tin được chọn và trả về các kết quả cho người sử dụng.

- Chức năng xem thông tin chi tiết:

Chức năng này cho phép người sử dụng xem thông tin chi tiết của một tài nguyên trong hệ thống bằng cách nhấn chọn tên tài nguyên muốn xem thông tin chi tiết. Hệ thống vào ontology tìm URIs cho tài nguyên cần tìm. Dựa vào URIs đã tìm được, hệ thống sẽ theo địa chỉ này để tìm thông tin chi tiết của tài nguyên và trả về kết quả tìm được.

3.4. XÂY DỰNG ONTOLOGY

Mục đích sử dụng ontology trong ứng dụng là giúp mô hình hóa dễ dàng các tri thức chung trong lĩnh vực du lịch. Tất cả các lớp được đặt theo tên của danh từ và tất cả các thuộc tính có tên với tiền tố là hình thức động từ *has* (như *hasName*) hoặc *is* (như *isRegionOf*).

Mỗi bước xây dựng ontology được trình bày chi tiết như sau:

3.4.1. Xác định miền và phạm vi của ontology

Lĩnh vực cần xây dựng ontology ở đây là thông tin liên quan đến các địa điểm du lịch, mà cụ thể là xem xét các thông tin chi tiết về địa điểm đã chọn như địa danh để thăm quan, nơi ăn uống, nơi ở, ... Miền mà ontology sẽ bao trùm là khái niệm và thông tin mô tả của các điểm đến du lịch, các địa danh có trong điểm đến, nơi ăn uống - nơi ở - nơi vui chơi giải trí – nơi tham quan tìm hiểu thuộc điểm đến. Tri thức về lĩnh vực du lịch trong ontology được chia sẻ theo cách cho phép người dùng tìm địa điểm du lịch cần đến trên đất nước Việt Nam.

3.4.2. Sử dụng lại các ontology đã có

Với ứng dụng này, không có ontology có sẵn nào được sử dụng lại.

3.4.3. Xác định các thuật ngữ quan trọng

Bước tiếp theo là viết ra thông tin về những thuật ngữ hoặc những khái niệm quan trọng. Danh sách ứng cử các khái niệm về địa điểm du lịch như sau:

- Địa điểm - Địa danh

- Khách sạn
- Nhà hàng
- Nơi ăn uống
- Quán bar
- Nơi mua sắm
- Quán cà phê
- Lễ hội
- Chợ
- Đặc sản
- Siêu thị

Tiếp đó, dựa trên các khái niệm để định nghĩa các quan hệ giữa chúng. Một số quan hệ cơ bản bao gồm: là địa danh thuộc điểm đến, là khách sạn thuộc điểm đến, có điểm đến, có thông tin chi tiết, …

3.4.4. Xác định lớp và xây dựng cây phân cấp lớp

Danh sách những thuật ngữ hoặc khái niệm đã mô tả ở trên sẽ được tổ chức thành các lớp và sau đó tổ chức chúng thành cấu trúc lớp phân cấp.

Các lớp được xây dựng như sau:

- Lớp chính được mô tả đầu tiên bao gồm: Địa điểm và Chi tiết địa điểm.
- Chi tiết địa điểm bao gồm các thông tin chi tiết về địa điểm nên sẽ có các lớp con: Địa danh, Khách sạn, Nơi ăn uống, Nơi mua sắm, Lễ hội, Đặc sản.
  - Lớp Nơi ăn uống mô tả những nơi phục vụ ăn uống thuộc địa điểm đã chọn gồm có các lớp con: Nhà hàng, Quán cà phê, Quán Bar.
  - Lớp Nơi mua sắm mô tả những nơi phục vụ nhu cầu mua sắm tại địa điểm đã chọn gồm có các lớp con: Chợ, Siêu thị.
- Địa điểm sẽ thuộc một vùng miền nào đó nên sẽ có thêm lớp Vùng miền.

Chương trình soạn thảo Protége được sử dụng để mã hoá ontology và áp dụng hướng tiếp cận từ trên xuống dưới như hình 3.8 sau:

Hình 3. 8 Minh họa các lớp của ontology bằng Protégé

3.4.5. Xác định các thuộc tính của các lớp

Các thuộc tính của mỗi lớp mô tả tính chất của lớp và các mối quan hệ của nó với các lớp khác. Việc xác định các thuộc tính khá phức tạp. Các thuộc tính Datatype thì dễ dàng nhận biết hơn các thuộc tính Object – là thuộc tính biểu diễn quan hệ giữa các thực thể.

Từ cây phân cấp lớp đã mô tả trên, các thuộc tính đối tượng của các lớp được xây dựng như sau:

- Lớp **Destination** (Địa điểm): có thuộc tính hasName, hasOtherName, hasDescription, hasLocation, hasPlaceName, hasRestaurant, hasCafes, hasBar, hasHotel, hasLocalfood, hasFestival, hasMarket, hasSuperMarket.
- Lớp **Region** (Vùng miền): có thuộc tính là hasDestination.
- Lớp **Characteristic** (Chi tiết địa điểm): có các thuộc tính: hasName, hasOtherName, hasDescription.
- Lớp **Festival** (Lễ hội): là lớp con của lớp Characteristic, ngoài các thuộc tính kế thừa từ Characteristic còn có thuộc tính hasTime.
- Lớp **EatingAndDrinking** (Ăn uống): là lớp con của lớp Characteristic, ngoài các thuộc tính kế thừa từ Characteristic còn có các thuộc tính: hasPhone, hasFax, hasAdress.

- Lớp **Bar, Cafe's, Restaurant** (Nhà hàng): là các lớp con của lớp EatingAndDrinking và có các thuộc tính kế thừa từ EatingAndDrinking.
- Lớp **Hotel** (Khách sạn): là lớp con của lớp Characteristic. Lớp Hotel ngoài các thuộc tính kế thừa từ Characteristic còn có các thuộc tính: hasPhone, hasFax, hasAdress, hasStandard.
- Lớp **LocalFood** (Đặc sản): lớp con của lớp Characteristic và có các thuộc tính kế thừa từ Characteristic.
- Lớp **PlaceName** (Địa danh): là lớp con của lớp Characteristic. Lớp PlaceName ngoài các thuộc tính kế thừa từ Characteristic còn có thuộc tính hasLocation.
- Lớp **Shopping** (Mua sắm): là lớp con của lớp Characteristic. Lớp Shopping ngoài các thuộc tính kế thừa từ Characteristic còn có thuộc tính hasAdress.
- Lớp **Market** (Chợ): lớp con của lớp Shopping và có các thuộc tính kế thừa từ Shopping.
- Lớp **SuperMarket** (Siêu thị): lớp con của lớp Shopping. Lớp SuperMarket ngoài các thuộc tính kế thừa từ Shopping còn có các thuộc tính: hasPhone, hasFax.

3.4.6. Mô tả các thuộc tính và xác định quan hệ cho các lớp

Bước này xử lý các thuộc tính bao gồm các giới hạn kiểu dữ liệu, domain và range. Thông thường, các thuộc tính Object có range là các lớp trong khi đó với các thuộc tính Datatype có range là các giá trị string, integer ... Range và domain của các thuộc tính được xác định như sau:

- Thuộc tính **hasName**: tên địa điểm hoặc tên các thành phần thuộc chi tiết địa điểm

Thuộc tính **hasOtherName**: một tên gọi khác của địa điểm hoặc tên các thành phần thuộc chi tiết địa điểm

Thuộc tính **hasDescription**: mô tả thông tin về địa điểm hoặc các thành phần thuộc chi tiết địa điểm

OWL Type: DatatypeProperty
Domain: Characteristic ∪ Destination
Range: xsd:String

- Thuộc tính **hasLocation**: mô tả vị trí của địa điểm hoặc địa danh thuộc địa điểm

OWL Type: DatatypeProperty
Domain: PlaceName ∪ Destination
Range: xsd:String

- Thuộc tính **hasPhone**: số điện thoại của siêu thị, khách sạn hoặc nơi ăn uống

Thuộc tính **hasFax**: số fax của siêu thị, khách sạn hoặc nơi ăn uống

OWL Type: DatatypeProperty
Domain: SuperMarket ∪ Hotel ∪ EatingAndDrinking
Range: xsd:String

- Thuộc tính **hasAdress**: địa chỉ của khách sạn, nơi mua sắm hoặc nơi ăn uống

OWL Type: DatatypeProperty
Domain: Shopping ∪ Hotel ∪ EatingAndDrinking
Range: xsd:String

- Thuộc tính **hasTime**: thời gian diễn ra lễ hội

OWL Type: DatatypeProperty
Domain: Festival
Range: xsd:String

- Thuộc tính **hasStandard**: xếp hạng sao của khách sạn

OWL Type: DatatypeProperty
Domain: Hotel
Range: xsd:String

– Thuộc tính **hasPlaceName**: mô tả quan hệ giữa địa điểm và địa danh

Thuộc tính **hasRestaurant**: mô tả quan hệ giữa địa điểm và nhà hàng

Thuộc tính **hasCafes**: mô tả quan hệ giữa địa điểm và quán cà phê

Thuộc tính **hasBar**: mô tả quan hệ giữa địa điểm và quán bar

Thuộc tính **hasHotel**: mô tả quan hệ giữa địa điểm và khách sạn

Thuộc tính **hasLocalFood**: mô tả quan hệ giữa địa điểm và đặc sản

Thuộc tính **hasFestival**: mô tả quan hệ giữa địa điểm và lễ hội

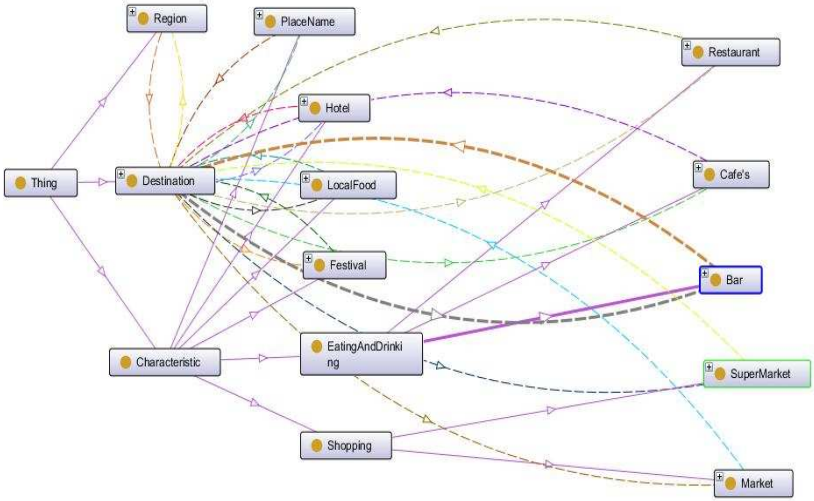
Thuộc tính **hasMarket**: mô tả quan hệ giữa địa điểm và chợ

Thuộc tính **hasSuperMarket**: mô tả quan hệ giữa địa điểm và siêu thị

Thuộc tính **hasDestination**: mô tả quan hệ giữa vùng và địa điểm

OWL Type: ObjectProperty

Domain và range của các thuộc tính object này được xác định theo lược đồ quan hệ giữa các lớp đã mô tả trên như hình 3.10 sau:



Hình 3.10 Lược đồ quan hệ của các lớp trong ontology

3.4.7. Xây dựng các thực thể

Bước cuối cùng là hoàn chỉnh ontology với các thực thể thì đòi hỏi rất nhiều thời gian. Cuối cùng, ontology của ứng dụng cơ bản định nghĩa 14 lớp và 29 thuộc tính. Sau khi mã hóa bằng Protégé, nội dung của ontology được lưu trữ trong tập tin semTravel.owl.

*** Sử dụng ontology**

Ontology được khai báo thông qua một URI:

http://www.semanticweb.org/ontologies/2011/2/Ontology1300196299906.owl

3.5. XÂY DỰNG ỨNG DỤNG

Phần này sẽ mô tả chi tiết các module được sử dụng và các phương thức được thực thi để phát triển ứng dụng. Trước hết, để thực hiện mã hóa các chức năng đã được đặc tả trên, các lược đồ tuần tự và các lược đồ cộng tác lần lượt được xây dựng cho các chức năng nhằm chỉ ra trình tự thực hiện các công việc và sự tương tác qua lại giữa các đối tượng, quan hệ giữa các đối tượng và giúp hình dung được khía cạnh động của ứng dụng.

Tiếp theo, tiến hành xây dựng các lớp và thành phần chính trong phát triển ứng dụng:

Bước đầu là tích hợp C# với cơ sở tri thức OWL thông qua OwlDotNetApi trong Visual Studio:

using OwlDotNetApi;

Sau đó tiến hành xây dựng các lớp và thành phần chính trong ứng dụng bao gồm:

- Lớp FindDB: chứa các thuộc tính và phương thức liên quan tìm kiếm tài nguyên trong ontology. Nó xử lý việc tìm kiếm thông tin trong ứng dụng.
- semTravel.owl: tập tin owl chứa ontology về các địa điểm du lịch

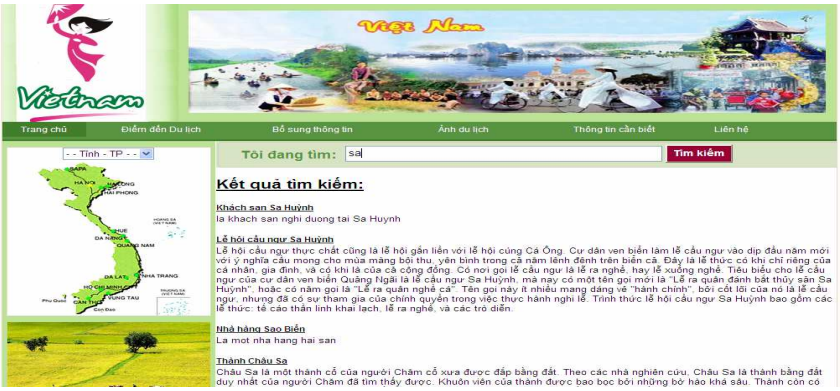
- Lớp DBDisplay: Truy xuất ontology để lấy các thông tin, dữ liệu
- Lớp InfoDisplay: Sử dụng lớp *OwlDotNetApi* để giao tiếp với thành phần semTravel.owl.
- OwlDotNetApi: có nhiệm vụ kết hợp C# và Owl. Trong số các lớp của OwlDotNetApi thì có một số lớp chính được sử dụng trong ứng dụng: *OwlXmlParser, OwlGraph, OwlEdgeCollection, OwlEdge, OwlNode, OwlClass, OwlProperty, OwlIndividual.*

3.6. CÀI ĐẶT ỨNG DỤNG

- Kết quả thực hiện tìm kiếm một điểm đến du lịch:

Hình 3. 24 Minh họa trang kết quả tìm kiếm 1

- Kết quả thực hiện tìm kiếm với từ khóa bất kỳ:



Hình 3. 25 Minh họa trang kết quả tìm kiếm 2

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận

Luận văn đã nghiên cứu và trình bày những kiến thức căn bản về web ngữ nghĩa như kiến thức về RDF, RDF Schema, ontology và các nguyên tắc được dùng để thiết kế một ontology được tốt. 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 ontology, từ đó hiểu được công nghệ web ngữ nghĩa để có thể dựa vào đó triển khai các ứng dụng khác, đồng thời cung cấp giải pháp để xây dựng một ontology.

Về ứng dụng minh hoạ, với mục tiêu làm rõ thêm lý thuyết, luận văn ứng dụng xây dựng web ngữ nghĩa với các công cụ hỗ trợ như Protégé và OwlDotNetApi. Cụ thể là xây dựng được tập từ vựng cơ bản về các địa điểm du lịch Việt Nam, khai thác các tính năng truy xuất trên một tài liệu có mô tả ngữ nghĩa nhằm chia sẻ tài nguyên thông tin về du lịch 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.

2. Hướng phát triển

Sẽ hoàn thiện ontology hơn 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 một reasoner ontology như HemiT hay Pellet để xây dựng module suy diễn dựa trên cơ sở tri thức của ứng dụng nhằm truy vấn dữ liệu hiệu quả hơn.