

**BỘ GIÁO DỤC ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM**



ĐỒ ÁN TỐT NGHIỆP

XÂY DỰNG CƠ SỞ TRI THỨC VỀ DU LỊCH

SINH VIÊN THỰC HIỆN : Phạm Tùng Linh

MÃ SINH VIÊN : 1451020145

KHOA : CÔNG NGHỆ THÔNG TIN

HÀ NỘI - 2024

**BỘ GIÁO DỤC ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM**



PHẠM TÙNG LINH
XÂY DỰNG CƠ SỞ TRI THỨC VỀ DU LỊCH

CHUYÊN NGÀNH : CÔNG NGHỆ THÔNG TIN
MÃ SỐ : 74.80.201

NGƯỜI HƯỚNG DẪN: GV. TRẦN ĐỨC MINH

HÀ NỘI - 2024

LỜI CAM ĐOAN

Tôi, Phạm Tùng Linh, xin cam đoan rằng đồ án tốt nghiệp với chủ đề "Xây dựng cơ sở tri thức về du lịch" là do tôi tự thực hiện và chịu trách nhiệm về nội dung của nó. Tôi xin cam đoan rằng:

1. Đây là công trình nghiên cứu của riêng tôi, không sao chép hay mượn từ bất kỳ nguồn tài liệu nào khác mà không được trích dẫn và chứng minh rõ ràng.

2. Tôi đã cẩn thận sử dụng các tài liệu tham khảo và trích dẫn đúng theo quy định.

3. Nếu có bất kỳ thông tin, số liệu nào không chính xác, tôi xin hoàn toàn chịu trách nhiệm.

Nếu vi phạm lời cam đoan này, tôi xin chịu mọi hình thức kỷ luật theo quy định của trường.

Hà Nội, ngày 27/5/2024

Người cam đoan

Phạm Tùng Linh

DANH MỤC KÝ HIỆU HOẶC CHỮ VIẾT TẮT

STT	Ký hiệu chữ viết tắt	Chữ viết tắt đầy đủ
1	OWL	Web Ontology Language
2	RDF-S	RDF Schema
3	CNTT	Công nghệ thông tin
4	OWL DL	OWL Description Logic
5	W3C	World Wide Web Consortium

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI.....	8
1. Thực trạng và sự cấp thiết của đề tài.....	8
2. Mục đích của đề tài.....	8
3. Đối tượng của đề tài	8
4. Các phương pháp thực hiện đề tài	9
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	10
2.1. Khái niệm về Tri thức.....	10
2.2. Khái quát Hệ cơ sở tri thức.....	10
2.3. Khái quát Hệ chuyên gia	10
2.4. Giới thiệu về Ontology	12
2.4.1. Khái niệm Ontology :	12
2.5.2 Các thành phần của Ontology.....	13
2.6. Ngôn ngữ OWL	15
2.7. Các bước xây dựng Ontology.....	15
2.8. Công cụ xây dựng Ontology.....	25
2.9. Truy vấn Sparql	26
CHƯƠNG 3: XÂY DỰNG ONTOLOGY TRONG MIỀN DU LỊCH	28
3.1. Tìm hiểu về ngành du lịch tại Việt Nam	28
3.1.1. Hiện trạng về ngành du lịch.....	28
3.1.2 Đánh giá về các nguồn tìm kiếm thông tin du lịch hiện tại.....	29
3.1.3 Hướng triển khai khắc phục.....	30
3.2. Nghiên cứu một số kịch bản tìm kiếm thông tin du lịch.....	30
3.3 Thu thập dữ liệu trong miền du lịch	33
3.4 Xây dựng Ontology phục vụ tra cứu thông tin về du lịch tại Việt Nam...	33
CHƯƠNG 4: TRIỂN KHAI ỨNG DỤNG ONTOLOGY TRONG MIỀN DU LỊCH.....	47

4.1 Môi trường triển khai thực nghiệm Bảng.....	47
4.1.1: Cấu hình máy thử nghiệm	47
4.1.2 Môi trường và các công cụ sử dụng trong luận văn:	47
4.2 Mục tiêu triển khai thử nghiệm	47
4.3 Thiết kế ứng dụng.....	48
4.3.1 Thiết kế chức năng ứng dụng □	48
4.3.2 Thiết kế các kịch bản tìm kiếm.....	48
4.3.3 Thiết kế kiến trúc hệ thống □	50
4.3.4 Thiết kế luồng nghiệp vụ chung	53
4.4 Luồng nghiệp vụ cụ thể và kết quả hiển thị cho các kịch bản	54
4.5 Đánh giá ứng dụng thử nghiệm.....	58
KẾT LUẬN	59
TÀI LIỆU THAM KHẢO.....	60

LỜI CẢM ƠN

Em chân thành gửi lời cảm ơn sâu sắc đến thầy Trần Đức Minh đã tận tình giúp đỡ em trong đồ án Hệ cơ sở tri thức. Nhờ sự hướng dẫn nhiệt tình của thầy mà em hiểu biết nhiều hơn về khối kiến thức mới này. Trong thời gian học tập và làm việc cùng thầy, em không ngừng tiếp thu được những điều bổ ích khác từ chuyên ngành Công nghệ thông tin. Bởi thế trong quá trình thực hiện đồ án môn học Hệ cơ sở tri thức, em đã đạt được kết quả cần thiết. Tuy nhiên vẫn sẽ không tránh khỏi những sai sót trong việc nghiên cứu đề tài nhưng em mong thầy sẽ chỉ rõ và nhận xét một cách thẳng thắn nhất để tụi em có thể hoàn thiện cho những đồ án sau này. Em xin chân thành cảm ơn thầy và chúc thầy công tác thật tốt!

LỜI MỞ ĐẦU

Ở thời đại công nghệ 4.0 như hiện nay, việc số hoá dữ liệu trong ngành du lịch càng nhanh càng tốt là rất quan trọng. Quá trình số hoá dữ liệu sẽ mang lại hiệu quả tối ưu cho ngành du lịch và mang đến nhiều lợi ích cho khách du lịch. Để làm được điều này, ngành du lịch cần phải sử dụng công nghệ thông tin và các kỹ thuật tiên tiến nhất.

Ứng dụng công nghệ thông tin trong ngành du lịch giúp cải thiện trải nghiệm của khách du lịch. Việc sử dụng ứng dụng di động, trang web và phần mềm du lịch giúp khách du lịch dễ dàng tìm hiểu thông tin về điểm đến, đặt phòng khách sạn, mua vé máy bay và tìm kiếm các hoạt động du lịch. Ngoài ra, việc số hoá thông tin du lịch giúp quản lý dữ liệu khách hàng, quảng bá thông tin du lịch và tăng cường quan hệ khách hàng.

Trong bối cảnh đại dịch Covid-19 vừa qua, việc đẩy mạnh khôi phục du lịch trở thành một yêu cầu quan trọng trong ngành du lịch. Các công nghệ tiên tiến như trí tuệ nhân tạo, trí tuệ nhân tạo, trí tuệ nhân tạo và trí tuệ nhân tạo có thể được áp dụng để tiếp cận khách du lịch. Ví dụ, du khách có thể tham gia đặt các tour du lịch, khám phá địa điểm du lịch thông qua thực tế.

Ngoài ra, việc áp dụng tri thức và công nghệ trong ngành du lịch cũng có thể giúp quản lý dữ liệu khách hàng, dự đoán và phân tích xu hướng du lịch, cải thiện quy trình đặt phòng và giao tiếp với khách hàng. Các công nghệ như trí tuệ nhân tạo, big data và học máy có thể được sử dụng để phân tích dữ liệu và đưa ra các khuyến nghị cá nhân hóa cho khách du lịch, từ việc đề xuất điểm đến phù hợp cho đến gợi ý các hoạt động du lịch dựa trên sở thích và lịch sử du lịch của khách hàng.

Hiểu rõ vấn đề đó, em quyết định ứng dụng kiến thức môn học Hệ cơ sở tri thức để thực hiện đề án “Xây dựng hệ cơ sở tri thức về du lịch” nhằm kiểm tra, đánh giá nhu cầu của khách du lịch như thế nào và tiến hành đưa ra gợi ý về địa điểm nào phù hợp nhất cho khách nhằm đẩy mạnh phát triển ngành du lịch.

CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

1. Thực trạng và sự cấp thiết của đề tài

Du lịch là ngành trọng điểm của nền kinh tế Việt Nam, với tốc độ phát triển rất nhanh trong những năm gần đây. Số lượng khách du lịch nội địa và quốc tế liên tục tăng mạnh và nằm trong số các nước có tốc độ phát triển cao nhất thế giới. Ngành du lịch đang đóng góp lớn vào ngân sách nhà nước (chiếm 9.2% GDP của Việt Nam năm 2019). Để đẩy mạnh phát triển du lịch, ngoài việc nâng cao chất lượng dịch vụ và tăng cường quảng bá thương hiệu thì việc đưa công nghệ thông tin ứng dụng vào ngành du lịch là một xu hướng tất yếu.

Hiện nay, rất nhiều ứng dụng tìm kiếm thông tin du lịch tại Việt Nam được xây dựng để hỗ trợ cho khách du lịch chuẩn bị cho những chuyến đi của họ. Dữ liệu về du lịch rất đa dạng và phong phú tuy nhiên đang nằm phân tán và rải rác ở rất nhiều website khác nhau, đặc biệt về nội dung chưa có nhiều sự liên kết thông tin với nhau về ngữ nghĩa. Có những ứng dụng chuyên về đặt phòng khách sạn, có những ứng dụng chuyên về điểm đến hay về ăn uống, đi lại, mua sắm. Khách du lịch muốn chuẩn bị thông tin cho chuyến đi của mình sẽ cần nhiều thời gian tra cứu thông tin ở nhiều nguồn khác nhau và tổng hợp lại. Để giải quyết vấn đề trên, tôi thực hiện đề tài: “Xây dựng hệ cơ sở tri thức về du lịch”, tạo ra một ứng dụng tra cứu đầy đủ từ tìm kiếm các cơ sở lưu trú, ăn uống, điểm đến, mua sắm và vận chuyển. Khách du lịch chỉ cần tra cứu trên một ứng dụng duy nhất nhưng có đầy đủ thông tin cần thiết cho một chuyến du lịch.

2. Mục đích của đề tài

Mục tiêu của luận văn này sẽ trình bày một cách tiếp cận nhằm mục đích hỗ trợ khách du lịch tìm kiếm theo ngữ nghĩa, các khái niệm tương đồng ngữ nghĩa và kích hoạt các gợi ý thông minh trong các hệ thống hỗ trợ du lịch sử dụng Ontology. Một Ontology du lịch sẽ được phát triển và tích hợp vào một ứng dụng tìm kiếm thông tin du lịch cho phép người dùng truy vấn các thông tin về du lịch như khách sạn, nhà hàng, quán ăn, các điểm đến, nơi mua sắm, ... với nhiều khả năng gợi ý dựa trên kiến thức được tập hợp từ nhiều nguồn tri thức trên Internet và các kho tri thức đã có. Kết quả của đề tài sẽ có tính thực tiễn và ứng dụng cao, nâng cao hiệu quả tìm kiếm hỗ trợ các khách du lịch cho các chuyến đi của họ.

3. Đối tượng của đề tài

Đối tượng chính của đề tài này là những người hoặc tổ chức có quan tâm đến thông tin, kiến thức và dịch vụ liên quan đến du lịch

4. Các phương pháp thực hiện đề tài

- Nghiên cứu tài liệu.
- Khảo sát, thu thập dữ liệu thực tế.
- Phân tích, đánh giá thông tin.
- Xây dựng cơ sở tri thức.
- Kiểm tra, đánh giá.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Khái niệm về Tri thức

Tri Thức: hay còn gọi là kiến thức (tiếng anh: knowledge) bao gồm những kiến thức, thông tin, sự hiểu biết, hay kỹ năng có được nhờ trải nghiệm, thông qua giáo dục hay tự học hỏi. Trong tiếng Việt, cả "tri" lẫn "thức" đều có nghĩa là biết. Tri thức có thể chỉ sự hiểu biết về một đối tượng, về mặt lý thuyết hay thực hành.

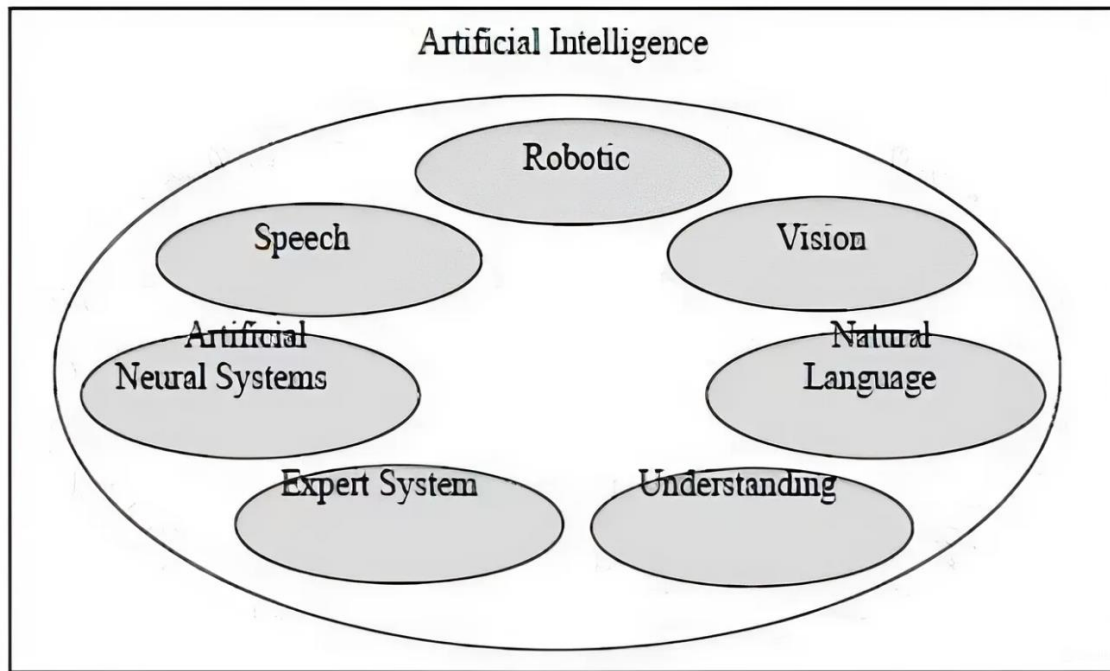
2.2. Khái quát Hệ cơ sở tri thức

Hệ cơ sở tri thức là chương trình máy tính được thiết kế để mô hình hoá khả năng giải quyết vấn đề của chuyên gia con người. Hệ cơ sở tri thức là hệ thống dựa trên tri thức, cho phép mô hình hoá các tri thức của chuyên gia, dùng tri thức này để giải quyết vấn đề phức tạp thuộc lĩnh vực cụ thể.

Hai yếu tố quan trọng trong Hệ cơ sở tri thức là: tri thức chuyên gia và lập luận, tương ứng với hai khối chính là: cơ sở tri thức và động cơ suy diễn.

2.3. Khái quát Hệ chuyên gia

Hệ chuyên gia là một hệ thống tin học có thể mô phỏng (emulates) năng lực quyết đoán (decision) và hành động (making ability) của một chuyên gia (con người). Hệ chuyên gia là một trong những lĩnh vực ứng dụng của trí tuệ nhân tạo (Artificial Intelligence) như hình dưới đây:



Hình 2.1: Một số lĩnh vực ứng dụng của trí tuệ nhân tạo

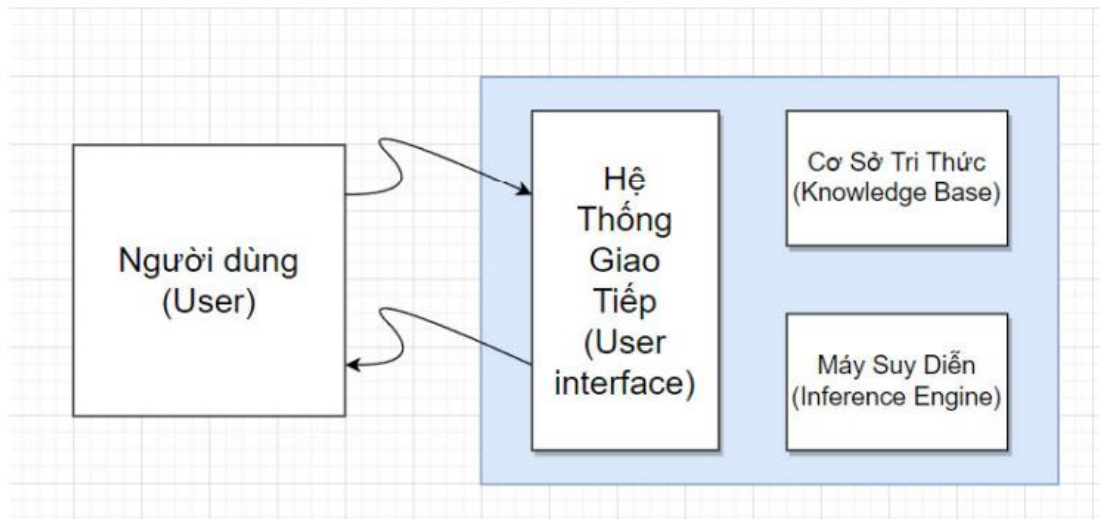
Hệ chuyên gia sử dụng các tri thức của những chuyên gia để giải quyết các vấn đề (bài toán) khác nhau thuộc mọi lĩnh vực.

Tri thức (knowledge) trong hệ chuyên gia phản ánh sự tinh thông được tích tụ từ sách vở, tạp chí, từ các chuyên gia hay các nhà bác học. Các thuật ngữ hệ chuyên gia, hệ thống dựa trên tri thức (knowledge - based system) hay hệ chuyên gia dựa trên tri thức (knowledge - based expert system) thường có cùng nghĩa.

Một hệ chuyên gia gồm ba thành phần chính là: cơ sở tri thức (knowledge base), máy suy diễn hay mô-đen suy diễn (inference engine), và hệ thống giao tiếp với người sử dụng (user interface). Cơ sở tri thức chứa các tri thức để từ đó, máy suy diễn tạo ra câu trả lời cho người sử dụng qua hệ thống giao tiếp.

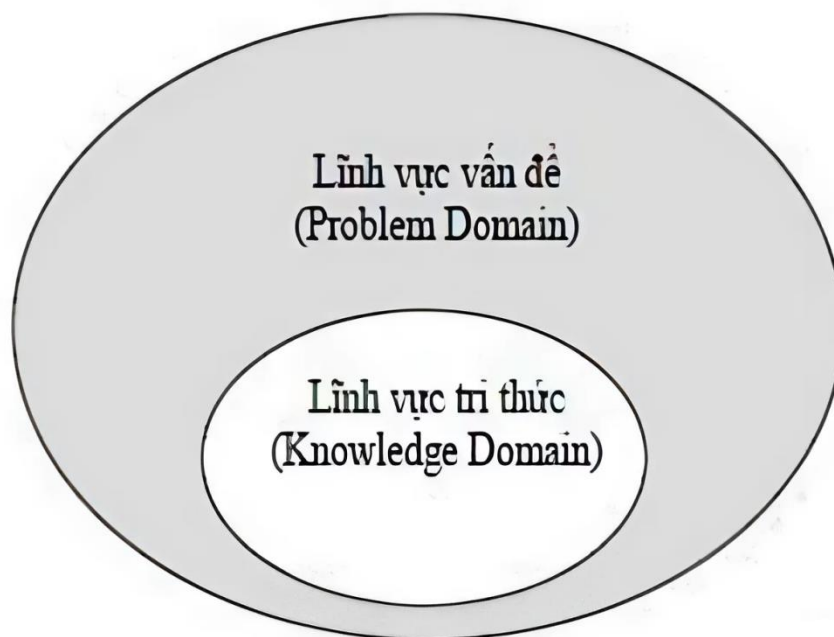
Người sử dụng (user) cung cấp sự kiện (facts) là những gì đã biết, đã có thật hay những thông tin có ích cho hệ chuyên gia, và nhận được những câu trả

lời là những lời khuyên hay những gợi ý đúng đắn (expertise). Hoạt động của một



hệ chuyên gia dựa trên tri thức được minh họa như sau:

Mỗi hệ chuyên gia chỉ đặc trưng cho một lĩnh vực vấn đề (problem domain) nào đó, như y học, tài chính, khoa học hay công nghệ, v.v..., mà không phải cho bất cứ một lĩnh vực vấn đề nào. Tri thức chuyên gia để giải quyết một vấn đề đặc trưng được gọi là lĩnh vực tri thức (knowledge domain).

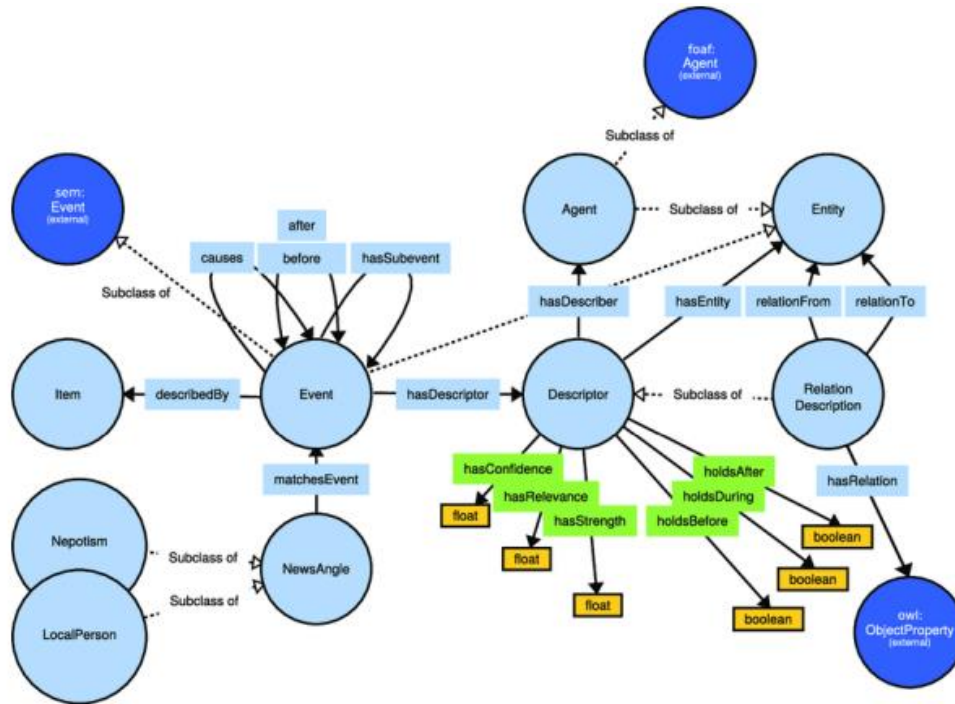


Hình 2.2: Mối quan hệ giữa lĩnh vực vấn đề và lĩnh vực tri thức

2.4. Giới thiệu về Ontology

2.4.1. Khái niệm Ontology :

Ontology là một hình thức mô tả chính quy và tường minh các khái niệm được chia sẻ. Một ontology được thể hiện bằng các khái niệm và mối quan hệ giữa các khái niệm. Ontology là một trong các hình thức biểu diễn tri thức rộng rãi và tiên tiến nhất hiện nay. Với hình thức biểu diễn tri thức này, mô hình của các khái niệm và mối quan hệ giữa các khái niệm trong miền tri thức cho phép tri thức có thể được chia sẻ giữa các ứng dụng



Hình 2.3: Ví dụ minh họa biểu diễn tri thức trong Ontology

Ontology được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau như thư viện điện tử, xử lý ngôn ngữ tự nhiên, kỹ thuật phần mềm, trí tuệ nhân tạo... Tuy nhiên, một trong những ứng dụng phổ biến nhất của ontology là trong lĩnh vực web ngữ nghĩa (Semantic Web). Đây chính là nền tảng cung cấp ngữ nghĩa cho dữ liệu, cho phép dữ liệu có thể được hiểu bởi máy tính.

2.5.2 Các thành phần của Ontology

Ontology bao gồm các thành phần: thực thể (Individuals), lớp (Classes), thuộc tính (Properties) và quan hệ (Relations).

➤ Thực thể

Thực thể là các thành phần cơ bản của ontology. Các thực thể là các đối tượng cụ thể, ví dụ về địa danh, điểm đến, khách sạn, nhà hàng, ... Ví dụ về các

thực thể: Hồ Hoàn Kiếm, Khách Sạn Thăng Lợi, Nhà Hàng Sen Hồ Tây, Trung tâm thương mại Tràng Tiền Plaza ...

➤ **Lớp**

Lớp là các nhóm hay tập hợp các đối tượng có chung một số thuộc tính. Lớp có thể bao gồm các lớp con, các thực thể hoặc bao gồm cả hai. Một lớp có thể gộp vào hoặc bị gộp vào các lớp khác. Một lớp gộp vào lớp khác được gọi là lớp con (subclass) của lớp gộp.

Ví dụ:

+Lớp Lưu trú có các lớp con là: Khách sạn, Nhà nghỉ, Homestay, ...

+Lớp Hồ có các thực thể là: Hồ Hoàn Kiếm, Hồ Tây, Hồ Trúc Bạch, ...

➤ **Thuộc tính**

Thuộc tính được sử dụng để chỉ ra mối quan hệ giữa các thực thể (thuộc tính đối tượng) hoặc giữa thực thể và kiểu dữ liệu được mô tả (thuộc tính kiểu dữ liệu). Ví dụ:

+ Thuộc tính đối tượng “ở gần” chỉ ra mối quan hệ về khoảng cách gần nhau giữa địa điểm Tràng Tiền Plaza và Hồ Hoàn Kiếm

+Thuộc tính kiểu dữ liệu với một địa điểm du lịch có thể có các thuộc tính: tên địa điểm, địa chỉ, số điện thoại, đánh giá... Giá trị của một thuộc tính dữ liệu có thể có các kiểu dữ liệu khác nhau như kiểu chuỗi, kiểu số...

➤ **Mối quan hệ**

Mối quan hệ được sử dụng để xác định sự liên quan của đối tượng này với một đối tượng khác bằng các thuộc tính tương ứng. Cùng với đó là tập hợp các quan hệ mô tả ngữ nghĩa của miền như quan hệ đồng nghĩa, trái nghĩa và các quan hệ khác. Điểm mạnh của Ontology là khả năng mô tả các mối quan hệ. Tập hợp các kiểu quan hệ được sử dụng và hệ thống phân cấp của chúng mô tả sức mạnh biểu đạt của ngôn ngữ mà Ontology được thể hiện.

Ví dụ:

+ Mối quan hệ “anh em ruột” giữa Nguyễn Văn An và Nguyễn Văn Bình .

+ Mỗi quan hệ về khoảng cách “ở gần” giữa hai địa điểm Trảng Tiền Plaza và Hồ Hoàn Kiếm

2.6. Ngôn ngữ OWL

OWL (Web Ontology Language) được thiết kế cho các ứng dụng cần xử lý nội dung thông tin thay vì chỉ trình bày thông tin cho con người. OWL được sử dụng để trình bày rõ ràng ý nghĩa của các thuật ngữ và mối quan hệ giữa các thuật ngữ đó. Sự thể hiện các thuật ngữ và mối quan hệ qua lại giữa chúng được gọi là Ontology. OWL làm tăng khả năng diễn đạt nội dung web so với XML, RDF và RDF Schema (RDF-S) bằng cách bổ sung thêm từ vựng cùng với ngữ nghĩa chính quy. OWL là bản sửa đổi của ngôn ngữ bản thể luận web DAML + OIL kết hợp các kinh nghiệm từ việc thiết kế và ứng dụng DAML + OIL.

OWL có ba ngôn ngữ con tăng dần theo mức độ biểu đạt nội dung: OWL Lite, OWL DL và OWL Full. Các phiên bản này được thiết kế cho các cộng đồng người triển khai và người dùng theo các nhu cầu cụ thể:

- **OWL Lite:** Hỗ trợ cho người sử dụng cần sự phân lớp và các ràng buộc đơn giản. OWL Lite có độ phức tạp hình thức thấp hơn OWL DL. Lợi ích của OWL Lite là nó là một ngôn ngữ ít đòi hỏi hiểu biết từ người dùng và nó cũng ít yêu cầu hơn đối với các nhà thiết kế để thực thi các thiết bị và ứng dụng cũng như tiếp cận sâu các dạng DL và Full.

- **OWL DL (OWL Description Logic):** Hỗ trợ cho những người dùng cần sự biểu đạt tối đa trong khi vẫn duy trì được tính hoàn chỉnh của tính toán và tất cả các tính toán sẽ hoàn thành trong khoảng thời gian hữu hạn. OWL DL cũng được đặt tên tương ứng với các logic mô tả, một lĩnh vực nghiên cứu trong logic đã hình thành nền tảng chính thức của OWL.

- **OWL Full:** Đề cập tới những người dùng cần sự biểu đạt tối đa và sự tự do về cú pháp của RDF mà không cần đảm bảo sự tính toán. Ví dụ, một lớp là một tập của các thực thể và cũng là thực thể của chính nó. OWL Full làm cho ontology tăng ý nghĩa của các từ vựng được định nghĩa trước (RDF hoặc OWL).

Mỗi ngôn ngữ con này là một phần mở rộng của ngôn ngữ trước đơn giản hơn của nó, cả về những gì có thể được diễn đạt một cách hợp lệ và những gì có

thể được kết luận một cách hợp lệ. Dựa trên nhu cầu thực tế, chúng ta sẽ xem xét và lựa chọn ngôn ngữ con phù hợp nhất.

➤ Mô tả ngôn ngữ OWL Lite

OWL Lite chỉ sử dụng một số tính năng của ngôn ngữ OWL, có nhiều hạn chế so với OWL DL hoặc OWL Full. Ví dụ, các lớp trong OWL Lite chỉ có thể được định nghĩa theo các lớp cha được đặt tên và chỉ một số loại hạn chế lớp nhất định mới có thể được sử dụng.

Các tính năng OWL Lite dưới đây liên quan đến Lược đồ RDF được bao gồm:

- **Class:** Một lớp xác định một nhóm các thực thể có cùng chung một số thuộc tính. Ví dụ, Hồ Hoàn Kiếm và Hồ Tây đều là thực thể của lớp “Hồ”. Các lớp có thể được tổ chức phân cấp bằng cách sử dụng `subClassOf`. Lớp tổng quát nhất được tích hợp sẵn có tên là `Thing`, là lớp của tất cả các thực thể và là lớp cha của tất cả các lớp OWL khác. Ngoài ra còn có một lớp được tích hợp sẵn có tên là `Nothing` là lớp không có thực thể và là lớp con của tất cả các lớp OWL.

- **`rdfs:subClassOf`:** Phân lớp được tạo ra để thể hiện một lớp là lớp con của một lớp khác. Ví dụ, lớp “Khách sạn” là một lớp con của lớp “Lưu trú”. Từ điều này có thể suy ra rằng nếu một thực thể là khách sạn, thì thực thể đó cũng là một cơ sở lưu trú.

- **`rdf:Property`:** Thuộc tính được sử dụng để chỉ ra mối quan hệ giữa các thực thể hoặc giữa các thực thể và giá trị dữ liệu. Ví dụ: thuộc tính “ở gần” được sử dụng để liên kết giữa một thực thể của lớp “Khách sạn” với một thực thể khác của lớp “Nhà hàng” (thuộc tính liên kết này gọi là `ObjectProperty`) và thuộc tính “đánh giá” được sử dụng để liên kết một thực thể của lớp “Khách sạn” với một thực thể kiểu dữ liệu `integer` (thuộc tính liên kết này gọi là `DatatypeProperty`). Cả `owl:ObjectProperty` và `owl:DatatypeProperty` là các lớp con của lớp `RDF rdf:Property`.

- **`rdfs:subPropertyOf`:** Phân cấp thuộc tính thể hiện một thuộc tính là thuộc tính con của một hoặc nhiều thuộc tính khác. Ví dụ: “anh em họ” là thuộc tính con của “họ hàng”. Từ điều này có thể suy ra rằng nếu một thực thể có liên

quan với thực thể khác bởi thuộc tính “anh em họ”, thì nó cũng liên quan đến thực thể kia bởi thuộc tính “họ hàng”.

- **rdfs: domain:** Miền của thuộc tính giới hạn các thực thể mà thuộc tính có thể được áp dụng. Nếu một thuộc tính liên kết một thực thể với một thực thể khác và thuộc tính bị giới hạn trong một lớp cụ thể, thì thực thể đó phải thuộc về lớp đó. Ví dụ: thuộc tính “hạng sao” giới hạn trong miền “Khách sạn”. Từ đó có thể suy ra rằng nếu Khách sạn Thắng Lợi có “hạng sao” là 4 sao, thì Khách sạn Thắng Lợi nằm trong miền “Khách sạn”. rdfs: domain được gọi là giới hạn toàn cục vì giới hạn được nêu trên thuộc tính chứ không chỉ trên thuộc tính khi nó được liên kết với một lớp cụ thể. □

- **rdfs: range:** Phạm vi thuộc tính giới hạn các thực thể mà thuộc tính có thể có như giá trị của nó. Nếu một thuộc tính liên kết hai thực thể và thuộc tính nằm trong phạm vi của một lớp, thì thực thể phải thuộc về phạm vi của lớp đó. Ví dụ, thuộc tính “ở gần” có phạm vi thuộc lớp “Địa điểm”. Từ đó suy ra rằng nếu Nhà Hát Lớn có quan hệ “ở gần” với Hồ Hoàn Kiếm thì Nhà Hát Lớn là thuộc lớp “Địa điểm”.

- **Individual:** Các thực thể là đối tượng của các lớp và các thuộc tính có thể được sử dụng để liên kết giữa thực thể này với thực thể khác. Ví dụ: một thực thể tên là “Nhà Hát Lớn” là đối tượng của lớp “Nhà hát” và thuộc tính “ở gần” có thể được sử dụng để liên hệ thực thể “Nhà Hát Lớn” với thực thể “Hồ Hoàn Kiếm”.

Các tính năng OWL Lite sau đây có liên quan đến tương đương và không tương đương.

- **equivalentClass:** Hai lớp có thể được cho là tương đương, hay còn gọi là hai lớp đồng nghĩa khi chúng có các đối tượng giống nhau. Ví dụ, lớp “Ô tô” là tương đương với lớp “Xe hơi”. Từ điều này có thể suy ra rằng bất kỳ thực thể nào là đối tượng của lớp “Ô tô” cũng là đối tượng của lớp “Xe hơi” và ngược lại.

- **equivalentProperty:** Hai thuộc tính có thể được cho là tương đương, hay còn gọi là các thuộc tính đồng nghĩa khi chúng liên kết một thực thể với cùng một tập hợp các thực thể khác. Ví dụ, “có lãnh đạo là” là equivalentProperty với “có sếp là”. Từ đó có thể suy ra rằng nếu Bình liên quan đến Cường bởi thuộc tính “có lãnh đạo là”, thì Bình cũng liên quan đến Cường bởi thuộc tính “có sếp là” và ngược lại.

- **sameAs**: Hai thực thể có thể được cho là giống nhau khi cùng đề cập đến cùng một đối tượng giống nhau. Ví dụ: thực thể “Hồ Hoàn Kiếm” và “Hồ Gươm” là hai thực thể giống nhau.

- **differentFrom**: Thể hiện một thực thể có thể khác với những thực thể khác. Ví dụ, thực thể “Hồ Hoàn Kiếm” khác với thực thể “Hồ Tây” và khác với thực thể “Hồ Trúc Bạch”.

- **AllDifferent**: Sử dụng để tuyên bố sự khác biệt lẫn nhau trong một nhóm các thực thể. Ví dụ: “Hồ Hoàn Kiếm”, “Hồ Tây” và “Hồ Trúc Bạch” được tuyên bố là khác biệt lẫn nhau bằng cách sử dụng AllDifferent, điều này có nghĩa là “Hồ Hoàn Kiếm” khác “Hồ Tây”, “Hồ Tây” khác “Hồ Trúc Bạch” và “Hồ Trúc Bạch” khác “Hồ Hoàn Kiếm”.

➤ Các đặc điểm thuộc tính OWL Lite

Có những nhận dạng đặc biệt trong OWL Lite được sử dụng để cung cấp thông tin liên quan đến các thuộc tính và giá trị của chúng. Sự khác biệt giữa ObjectProperty và DatatypeProperty đã được đề cập ở trên trong phần mô tả thuộc tính.

- **inverseOf**: Một thuộc tính là nghịch đảo của một thuộc tính khác. Ví dụ: “có cha” là nghịch đảo của “có con” và An “có cha” là Bình, thì có thể suy luận rằng Bình “có con” là An.

- **TransitiveProperty**: Là các thuộc tính có tính bắc cầu. Ví dụ: thuộc tính tổ tiên có tính bắc cầu, nếu An là tổ tiên của Bình và Bình là tổ tiên của Cường có thể suy luận rằng An là tổ tiên của Cường.

- **SymmetricProperty**: Là các thuộc tính có tính đối đối xứng. Ví dụ: “ở gần” là một thuộc tính đối xứng, Tràng Tiền Plaza “ở gần” Hồ Hoàn Kiếm thì có thể suy luận rằng Hồ Hoàn Kiếm cũng “ở gần” Tràng Tiền Plaza.

- **FunctionalProperty**: Là các thuộc tính có một giá trị duy nhất. Ví dụ: “địa chỉ” là một FunctionalProperty thì không có địa điểm nào có nhiều hơn một địa chỉ.

- **InverseFunctionalProperty:** Được sử dụng khi các thuộc tính có chức năng nghịch đảo lẫn nhau. Ví dụ: “có số CCCD là” và “là số CCCD của” là thuộc tính nghịch đảo của nhau, có nhiều nhất một giá trị đối với bất kỳ thực thể nào trong loại số căn cước công dân. Do đó, bất kỳ số căn cước công dân nào của một người là giá trị duy nhất cho thuộc tính “là số CCCD của” của họ. Từ điều này có thể suy ra rằng không có hai người khác nhau nào có cùng số căn cước công dân. Ngoài ra, cũng có thể suy luận rằng nếu hai đối tượng có cùng số căn cước công dân, thì hai trường hợp đó đề cập đến cùng một thực thể.

Các giới hạn thuộc tính của OWL Lite

OWL Lite cho phép đặt các giới hạn các thuộc tính có thể được sử dụng bởi các thực thể của một lớp. Những loại này được sử dụng trong ngữ cảnh của owl:Restriction. Phần tử owl:onProperty cho biết thuộc tính bị hạn chế. Hai hạn chế sau giới hạn giá trị nào có thể được sử dụng trong khi hạn chế của phần tiếp theo giới hạn số lượng giá trị có thể được sử dụng. □

- **allValuesFrom:** Giới hạn allValuesFrom được nêu trên một thuộc tính liên quan đến một lớp. Có nghĩa là thuộc tính trên lớp cụ thể này có giới hạn phạm vi cục bộ liên quan đến nó. Ví dụ, thuộc tính “có cha là” có giới hạn allValuesFrom trong lớp “có người”. □

- **someValuesFrom:** Hạn chế someValuesFrom được nêu trên một thuộc tính liên quan đến một lớp. Một lớp cụ thể có thể có một giới hạn đối với một thuộc tính mà ít nhất một giá trị cho thuộc tính đó thuộc một loại nhất định. Ví dụ, thuộc tính “có cha là” có giới hạn someValuesFrom trong lớp “Thầy giáo”.

- OWL Lite chứa một thuộc tính giao nhau nhưng giới hạn việc sử dụng nó.

□

- **intersectionOf:** OWL Lite cho phép định nghĩa phần giao của các lớp và các giới hạn của chúng. Ví dụ, lớp “Thầy giáo” có thể được mô tả như là phần giao của lớp “Đàn ông” và lớp “Giáo viên”.

➤ Mô tả ngôn ngữ OWL DL và OWL Full

Cả OWL DL và OWL Full đều sử dụng cùng một bộ từ vựng mặc dù OWL DL có một số hạn chế hơn OWL Full. Về cơ bản, OWL DL yêu cầu phân tách kiểu (một lớp không thể là một thực thể hoặc thuộc tính, một thuộc tính cũng

không thể là một thực thể hoặc lớp). Hơn nữa, OWL DL yêu cầu các thuộc tính đối tượng (ObjectProperties) hoặc thuộc tính kiểu dữ liệu (DatatypeProperties): DatatypeProperties là quan hệ giữa các thực thể của lớp và các ký tự RDF và các kiểu dữ liệu của lược đồ XML, trong khi ObjectProperties là quan hệ giữa các thực thể của hai lớp. Bộ từ vựng OWL DL và OWL Full được mở rộng so với OWL Lite như sau:

- **oneOf: (các lớp được liệt kê):** Các lớp có thể được mô tả bằng cách liệt kê các thực thể tạo nên lớp. Các thành viên của lớp chính xác là tập hợp các thực thể được liệt kê; không nhiều hơn, không ít hơn. Ví dụ, lớp “Các ngày trong tuần” có thể được mô tả bằng cách chỉ cần liệt kê các thực thể Chủ nhật, Thứ Hai, Thứ Ba, Thứ Tư, Thứ Năm, Thứ Sáu, Thứ Bảy.

- **hasValue: (giá trị thuộc tính):** Một thuộc tính liên kết một lớp giới hạn với một giá trị riêng lẻ hoặc giá trị dữ liệu. Ví dụ, các thực thể của lớp người Việt Nam có thể được mô tả là những người có giá trị quốc tịch là Việt Nam.

- - **disjointWith:** Các lớp có thể được thể hiện tách rời với nhau. Ví dụ, “đàn ông” và “phụ nữ” là các lớp riêng biệt, có thể suy luận rằng nếu An là một thực thể của lớp “đàn ông”, thì An không phải là một thực thể thuộc lớp “phụ nữ”. □ **unionOf, complementOf và intersectionOf:** OWL DL và OWL Full cho phép các kết hợp Boolean tùy ý của các lớp và các hạn chế: unionOf, complementOf và intersectionOf.

- **minCardinality, maxCardinality, cardinality (full cardinality):** Trong khi trong OWL Lite, các giá trị bị hạn chế ở ít nhất, nhiều nhất hoặc chính xác là 1 hoặc 0, OWL Full cho phép các phát biểu về giá trị cho các số nguyên không âm tùy ý.

- **complex classes:** Trong nhiều cấu trúc, OWL Lite hạn chế cú pháp đối với các tên lớp đơn (subClassOf, equivalentClass). OWL Full mở rộng hạn chế này để cho phép các mô tả lớp phức tạp tùy ý, bao gồm các lớp được liệt kê, các hạn chế thuộc tính và các kết hợp Boolean. Ngoài ra, OWL Full cho phép các lớp được sử dụng như các thực thể (còn OWL DL và OWL Lite thì không).

2.7. Các bước xây dựng Ontology

Để xây dựng một Ontology chúng ta sẽ tham khảo quy trình gồm 7 bước do nhóm nghiên cứu của đại học Stanford đưa ra.

✓ **Bước 1: Xác định lĩnh vực và phạm vi của Ontology**

Bước đầu tiên sẽ xác định lĩnh vực và phạm vi của Ontology:

- + Lĩnh vực mà Ontology sẽ đề cập đến là gì?
- + Mục đích của việc xây dựng và ứng dụng của Ontology?
- + Ontology cần trả lời cho những câu hỏi truy vấn thông tin gì?
- + Đối tượng sử dụng và khai thác Ontology?

✓ **Bước 2: Xem xét việc sử dụng lại các Ontology có sẵn**

Xem xét các Ontology hiện có và kiểm tra xem có thể kế thừa và mở rộng hay không. Nhiều Ontology đã có sẵn có thể được kế thừa và nhập vào môi trường phát triển Ontology mà chúng ta sử dụng.

Cộng đồng phát triển Ontology cho các lĩnh vực đang rất đa dạng, rất nhiều Ontology đã được tạo ra với các mục đích khác nhau. Do vậy trước khi bắt đầu xây dựng Ontology, cần tìm kiếm và xem xét kế thừa và phát triển các Ontology đã có nhằm mục đích tiết kiệm chi phí và công sức cho quá trình xây dựng Ontology. Nếu không có Ontology nào liên quan đã tồn tại thì chúng ta sẽ thực hiện phát triển mới từ đầu.

✓ **Bước 3: Liệt kê các thuật ngữ quan trọng trong Ontology**

Ở bước này chúng ta cần liệt kê danh sách các thuật ngữ muốn đề đưa ra và giải thích cho người dùng.

- + Các thuật ngữ chúng ta muốn đề cập đến là gì?
- + Các thuật ngữ đó có những thuộc tính gì?
- + Chúng ta muốn nói gì về những thuật ngữ đó.

Ví dụ: Các thuật ngữ quan trọng liên quan đến khách sạn sẽ bao gồm tên khách sạn, địa chỉ, số điện thoại, trang web, hạng sao, đánh giá, cơ sở vật chất, ... Ban đầu, điều quan trọng là có được một danh sách đầy đủ các thuật ngữ mà không phải lo về sự trùng lặp giữa các khái niệm mà chúng đại diện, mối quan hệ giữa

các thuật ngữ hoặc bất kỳ thuộc tính nào mà các khái niệm có thể có, hoặc liệu các khái niệm có phải là lớp hay thuộc tính.

✓ **Bước 4: Xác định các lớp và phân cấp của các lớp**

Một số cách tiếp cận để phát triển các phân cấp của các lớp:

+ Tiếp cận theo hướng từ trên xuống **top-down**: Quá trình phát triển từ trên xuống bắt đầu với việc xác định các khái niệm chung nhất trong lĩnh vực và sau đó sẽ tiếp tục cụ thể hóa các khái niệm. Ví dụ, chúng ta có thể bắt đầu với việc tạo các lớp cho các khái niệm chung về Du lịch. Sau đó, chúng ta cụ thể hóa lớp Du lịch bằng cách tạo ra một số phân lớp của nó: Cờ sở lưu trú, Điểm đến, Mua sắm, Vận chuyển, ... Tiếp theo chúng ta có thể phân loại lớp Cơ sở lưu trú thành các lớp con là Khách sạn, Nhà nghỉ, Homestay, Apartment, ...

+ Tiếp cận theo hướng từ dưới lên **bottom-up**: Quá trình phát triển bắt đầu từ các lớp cụ thể nhất, sau đó thực hiện nhóm các lớp này thành lớp tổng quát hơn là các lớp cha. Ví dụ, chúng ta bắt đầu bằng cách xác định các lớp Khách sạn, Nhà nghỉ, Homestay, Apartment, Trung tâm thương mại, Chợ, Cửa hàng, ... Sau đó, chúng ta tạo ra một lớp cha chung cho Khách sạn, Nhà nghỉ, Homestay, Apartment là lớp Lưu trú và lớp chung cho Trung tâm thương mại, Chợ, Cửa hàng là Mua sắm.

+ Tiếp cận theo hướng kết hợp: Quá trình phát triển kết hợp là sự kết hợp của các phương pháp **bottom-up và top-down**: Trước tiên, chúng ta xác định xem các khái niệm nổi bật, sau đó khái quát hóa và chi tiết hóa chúng một cách thích hợp. Chúng ta có thể bắt đầu với một vài khái niệm cấp cao nhất và một vài khái niệm cụ thể. Sau đó, chúng ta có thể liên hệ chúng với một khái niệm cấp trung.

Không có phương pháp nào trong số ba phương pháp trên được khẳng định tốt hơn các phương pháp còn lại. Nếu nhà phát triển Ontology có xu hướng nghĩ về phân loại chung nhất trước tiên, thì cách tiếp cận từ trên xuống có thể phù hợp hơn. Nếu nhà phát triển muốn bắt đầu bằng cách dựa trên các ví dụ cụ thể, thì cách tiếp cận từ dưới lên có thể phù hợp hơn.

✓ **Bước 5: Xác định các thuộc tính của các lớp**

Sau khi xác định các lớp, chúng ta phải mô tả cấu trúc các khái niệm. Chúng ta đã chọn các lớp từ danh sách các thuật ngữ đã tạo ở Bước 3. Hầu hết các thuật ngữ còn lại có thể là thuộc tính của các lớp này. Các thuật ngữ này bao gồm, ví dụ: tên gọi, địa chỉ, số điện thoại, trang web, đánh giá, số lượng đánh giá của một khách sạn. Đối với mỗi thuộc tính trong danh sách, chúng ta phải xác định nó mô tả lớp nào. Các thuộc tính này trở thành các thuộc tính gắn vào các lớp. Ví dụ thuộc tính “hạng sao” gắn vào lớp “Khách sạn” nhưng không gắn vào lớp “Nhà nghỉ”.

Vì vậy, ngoài các thuộc tính chúng ta đã xác định trước đó, chúng ta cần thêm các thuộc tính sau trong lớp “Khách sạn”: tên gọi, địa chỉ, số điện thoại, hạng sao, ... Tất cả các lớp con của một lớp đều kế thừa thuộc tính của lớp đó do vậy một thuộc tính cần được đính kèm ở lớp chung nhất có thể có thuộc tính đó.

✓ Bước 6: Xác định giới hạn của các thuộc tính

Thuộc tính có thể có các khía cạnh khác nhau để mô tả loại giá trị, giá trị cho phép, số lượng giá trị (số lượng) và các đặc điểm khác của giá trị mà thuộc tính có thể nhận. Ví dụ: thuộc tính “tên gọi” được giới hạn kiểu dữ liệu dạng chuỗi (string) còn thuộc tính “đánh giá” được giới hạn kiểu dữ liệu dạng số nguyên (integer).

Tập hợp thuộc tính

Tập hợp thuộc tính định nghĩa số lượng giá trị mà một thuộc tính có thể có. Một số hệ thống chỉ phân biệt giữa tập hợp đơn (cho phép nhiều nhất một giá trị) và tập hợp đa (cho phép bất kỳ số lượng giá trị nào).

Loại thuộc tính giá trị:

Một ràng buộc kiểu giá trị mô tả những loại giá trị nào có thể điền vào thuộc tính. Dưới đây là danh sách các loại giá trị phổ biến:

+ **String** là kiểu giá trị đơn giản nhất được sử dụng cho các thuộc tính, chẳng hạn như tên gọi một khách sạn.

+ **Number (Float, Integer, ...)** mô tả các thuộc tính có giá trị số. Ví dụ, giá phòng khách sạn có thể có kiểu giá trị integer.

+ **Boolean** là kiểu giá trị có - không. Ví dụ: nếu chọn “Bể bơi” là một trong những tiện ích để đánh giá khách sạn thì các khách sạn có thể “có” hoặc “không có” tiện tích này.

+ **Enumerated** chỉ rõ một danh sách các giá trị được phép cho thuộc tính. Ví dụ thuộc tính “đánh giá” có thể nhận một trong ba giá trị có thể có: tốt, trung bình, kém.

+ Các thuộc tính kiểu **Instance** cho phép xác định mối quan hệ giữa các thực thể. Các thuộc tính có kiểu giá trị Instance cũng phải xác định danh sách các lớp được phép mà các thực thể thuộc vào.

Domain (miền) và range (phạm vi) của một thuộc tính:

Các lớp được phép gán cho các thuộc tính thường được gọi là một phạm vi của một thuộc tính. Một số hệ thống cho phép giới hạn phạm vi của thuộc tính khi thuộc tính được gán cho một lớp cụ thể. Các lớp mà một thuộc tính được gán vào hoặc một lớp thuộc vùng mà thuộc tính mô tả, được gọi là miền của thuộc tính đó.

Các quy tắc cơ bản để xác định miền và phạm vi thuộc tính tương tự nhau: Khi xác định miền hoặc phạm vi cho thuộc tính, hãy tìm các lớp hoặc lớp chung nhất có thể tương ứng là miền hoặc phạm vi cho thuộc tính. Mặt khác, không xác định miền và phạm vi quá chung chung

- Nếu danh sách các lớp xác định một phạm vi hoặc một miền của một thuộc tính bao gồm một lớp và lớp con của nó, hãy xóa phạm vi và miền trên lớp con đó.

- Nếu danh sách các lớp xác định một phạm vi hoặc một miền của một thuộc tính chứa tất cả các lớp con của một lớp, nhưng không phải chính lớp đó, thì phạm vi chỉ nên chứa cả lớp cha chứ không cần chứa từng lớp con.

- Nếu danh sách các lớp xác định một phạm vi hoặc một miền của một thuộc tính chứa tất cả trừ một vài lớp con của một lớp cha, hãy xem xét liệu lớp cha có thể đưa ra một định nghĩa phạm vi phù hợp hơn hay không.

Trong các hệ thống mà việc gán một thuộc tính vào một lớp cũng giống như việc thêm lớp đó vào miền của thuộc tính: Một mặt, chúng ta nên cố gắng làm cho nó càng tổng quát càng tốt. Mặt khác, chúng ta phải đảm bảo rằng mỗi

lớp mà chúng ta gán thuộc tính thực sự có thể có thuộc tính mà thuộc tính đó đại diện.

✓ **Bước 7: Tạo các thể hiện / thực thể.**

Bước cuối cùng là tạo các thực thể của các lớp trong hệ thống phân cấp. Việc xác định một thực thể riêng lẻ của một lớp yêu cầu: (1) chọn một lớp, (2) tạo một thực thể riêng của lớp đó và (3) điền vào các giá trị thuộc tính. Ví dụ: tạo một thực thể “Khách sạn Thắng Lợi” thuộc lớp “Khách sạn”. Thực thể này có các giá trị thuộc tính sau được xác định:

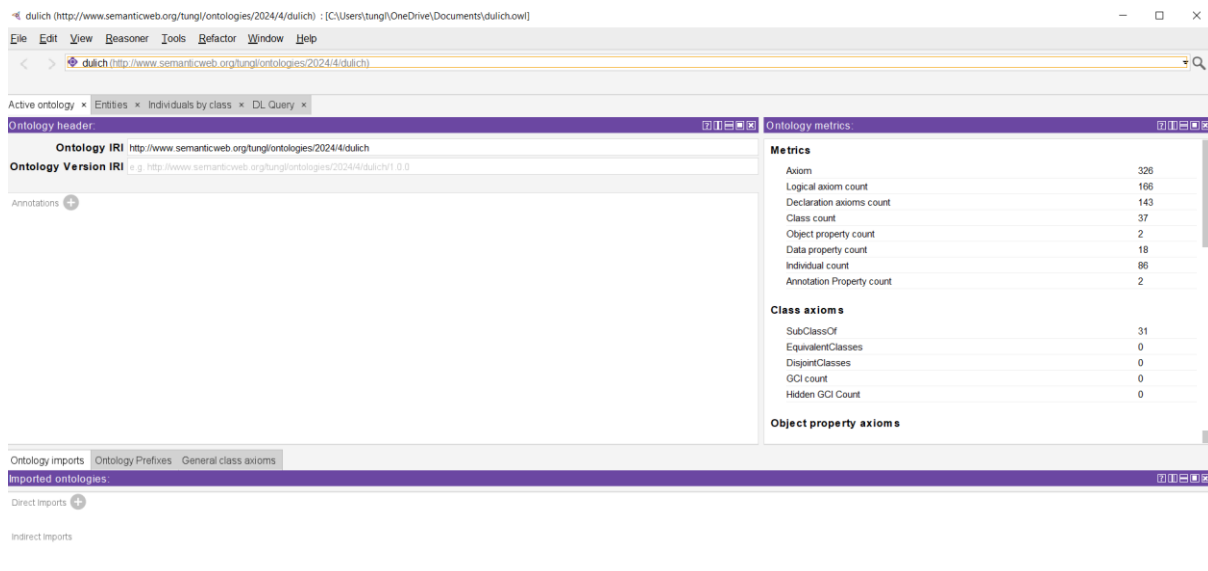
- Tên gọi: Khách sạn Thắng Lợi
- Địa chỉ: 200 P. Yên Phụ, Yên Phụ, Tây Hồ, Hà Nội
- Số điện thoại: 024 3829 4211
- Hạng sao: 4 sao
- Đánh giá: 3.9
- Số lượng đánh giá: 706

2.8. Công cụ xây dựng Ontology

Hiện nay có rất nhiều các công cụ hỗ trợ xây dựng Ontology, các công cụ này cung cấp các giao diện nhằm hỗ trợ người dùng phát triển một ontology một cách linh hoạt và thuận tiện. Protégé là một trong công cụ đang được sử dụng rộng rãi nhất để phát triển Ontology.

Protégé (<https://protege.stanford.edu/>) được phát triển bởi Trung tâm nghiên cứu Tin học Y sinh Stanford thuộc Trường đại học Y Stanford. Protégé là một nền tảng mã nguồn mở miễn phí cung cấp cho cộng đồng người dùng đang phát triển một bộ công cụ để xây dựng các mô hình miền và các ứng dụng dựa trên tri thức với các Ontology.

Protégé được hỗ trợ bởi một cộng đồng mạnh mẽ gồm người dùng học thuật, chính phủ và doanh nghiệp, những người sử dụng Protégé để xây dựng các giải pháp dựa trên tri thức trong các lĩnh vực đa dạng như y sinh, thương mại điện tử và mô hình tổ chức.



Hình 2.4: Giao diện phần mềm công cụ Protégé

Các ưu điểm của Protégé:

- Tuân thủ tiêu chuẩn W3C, hỗ trợ đầy đủ cho OWL
- Giao diện người dùng trực quan, đơn giản, có thể tùy chỉnh được
- Được tối ưu hóa cho cộng tác
- Hỗ trợ nhiều định dạng upload/download

2.9. Truy vấn Sparql

SPARQL là một ngôn ngữ truy vấn thông tin và thao tác nội dung với đồ thị RDF. RDF là một định dạng dữ liệu đồ thị có nhãn, có hướng để biểu diễn thông tin trong Web.

Hầu hết các dạng truy vấn SPARQL đều chứa một bộ ba mẫu (subject, predicate và object) được gọi là mẫu đồ thị cơ bản (basic graph pattern), trong đó mỗi thành phần trong bộ ba mẫu có thể là một biến. Một mẫu đồ thị cơ bản khớp với một đồ thị con của dữ liệu RDF khi các thuật ngữ RDF từ đồ thị con đó có thể thay thế cho các biến và kết quả là đồ thị RDF tương đương với đồ thị con.

Ví dụ dưới đây cho thấy một truy vấn SPARQL để tìm thực thể có tên gọi là “Hồ Hoàn Kiếm” từ biểu đồ dữ liệu đã cho. Truy vấn bao gồm hai phần: mệnh đề SELECT xác định các biến sẽ xuất hiện trong kết quả truy vấn và mệnh đề WHERE cung cấp mẫu đồ thị cơ bản để khớp với biểu đồ dữ liệu. Mẫu đồ thị

cơ bản trong ví dụ này bao gồm một mẫu ba đơn với một biến duy nhất (? subject) ở vị trí đối tượng.

```
PREFIX etourism:
```

```
<http://www.semanticweb.org/tungl/ontologies/2024/4/dulich#>
```

```
SELECT ?subject
```

```
WHERE { ?subject etourism:ten_goi "Hồ Hoàn Kiếm" }
```

SPARQL có bốn hình thức truy vấn. Các hình thức truy vấn này sử dụng các giải pháp từ so khớp mẫu đến tập kết quả mẫu hoặc biểu đồ RDF. Các hình thức truy vấn là:

- SELECT: Trả về tất cả hoặc một tập hợp con của các biến được ràng buộc trong một mẫu truy vấn phù hợp

- CONSTRUCT: Trả về đồ thị RDF được tạo bằng cách thay thế các biến trong một tập hợp ba mẫu.

- ASK: Trả về một boolean cho biết một mẫu truy vấn có khớp hay không.

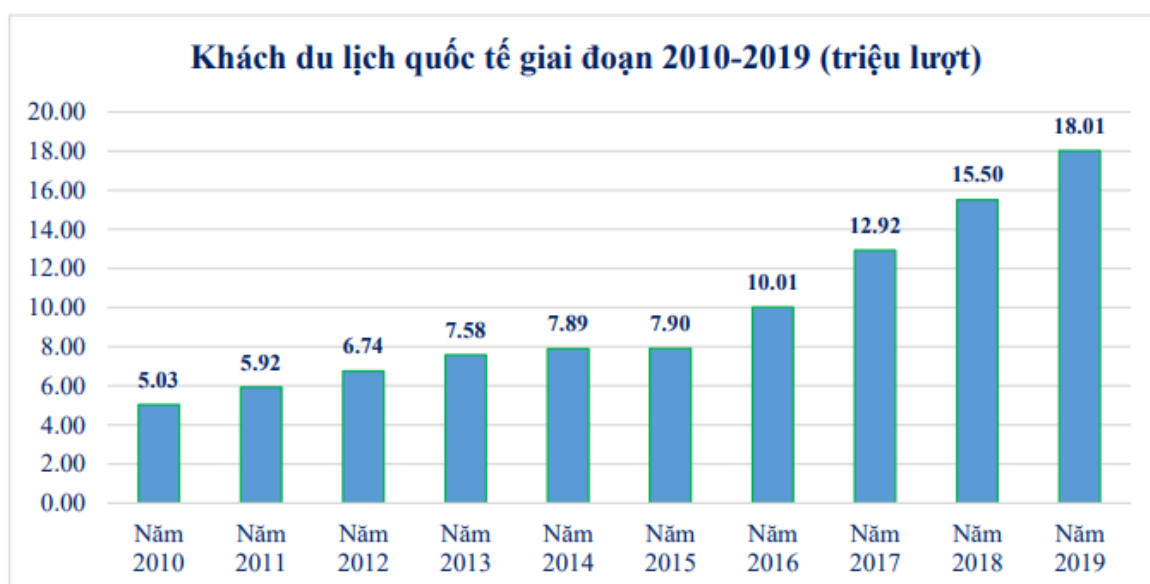
- DESCRIBE: Trả về biểu đồ RDF mô tả các tài nguyên được tìm thấy

CHƯƠNG 3: XÂY DỰNG ONTOLOGY TRONG MIỀN DU LỊCH

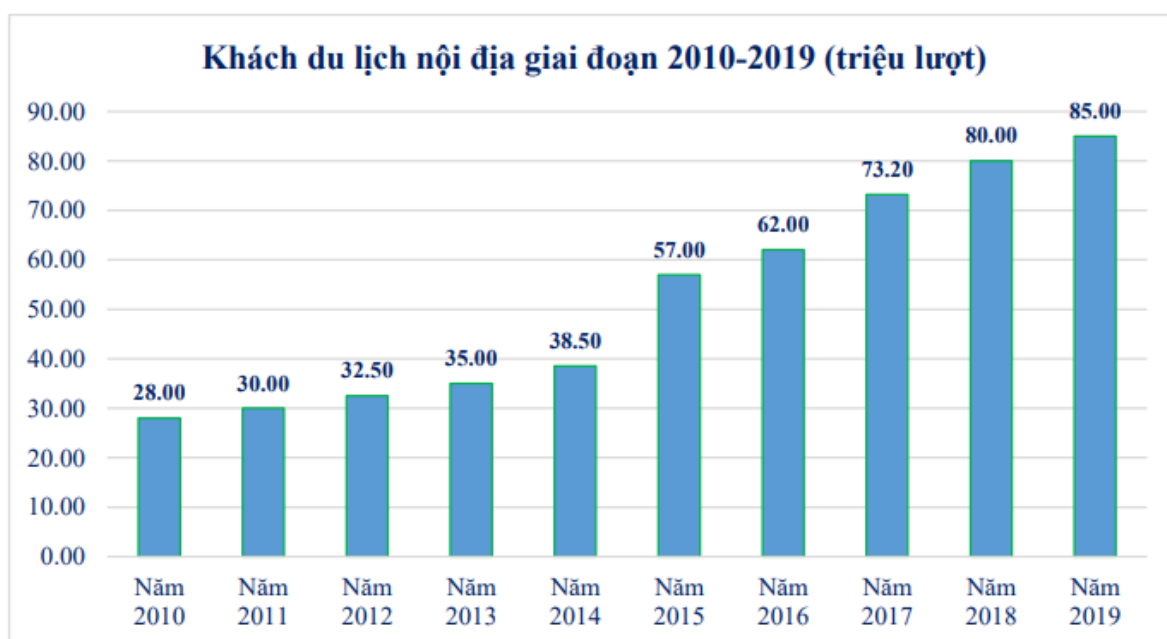
3.1. Tìm hiểu về ngành du lịch tại Việt Nam

3.1.1. Hiện trạng về ngành du lịch

Du lịch đang là ngành trọng điểm của nền kinh tế Việt Nam, có tốc độ phát triển rất nhanh trong những năm gần đây. Ngành du lịch đóng góp rất lớn vào GDP chung của Việt Nam, tăng từ 6.3% năm 2015 lên 9.2% năm 2019. Số lượng khách du lịch quốc tế và nội địa liên tục tăng mạnh, trong đó số lượng khách quốc tế đạt 18 triệu lượt và số lượng khách nội địa đạt 85 triệu lượt vào năm 2019. Tốc độ tăng trưởng khách du lịch của Việt Nam nằm trong top các nước cao nhất trên thế giới.



Hình 3.1: Khách du lịch đến Việt Nam, giai đoạn 2010-2019



Hình 3.2: Khách du lịch trong nước, giai đoạn 2010-2019

Để đẩy mạnh phát triển du lịch, ngoài việc nâng cao chất lượng dịch vụ và tăng cường quảng bá thương hiệu thì việc đưa công nghệ thông tin ứng dụng vào ngành du lịch là một xu hướng tất yếu. Khi việc ứng dụng CNTT được triển khai mạnh mẽ, khách du lịch có thể tra cứu và xác định một loạt các dịch vụ phong phú và đa dạng để đưa ra lựa chọn đi du lịch phù hợp với mục đích cá nhân. Điều này giúp cải thiện về dịch vụ, tăng trải nghiệm và sự tin cậy cho khách hàng.

Sản phẩm du lịch có bản chất vô hình ở chỗ khách hàng không thể cảm nhận về mặt vật chất các dịch vụ được cung cấp cho đến khi được trải nghiệm thực tế. Điều này làm cho việc tiếp cận với những thông tin đáng tin cậy và xác thực về các sản phẩm du lịch trước khi trải nghiệm thực tế là rất có giá trị. Ontology là một đặc tả hình thức, rõ ràng về các khái niệm của miền, cung cấp một nền tảng khả thi để phát triển các dịch vụ thông tin du lịch dựa trên tri thức đáng tin cậy.

3.1.2 Đánh giá về các nguồn tìm kiếm thông tin du lịch hiện tại

Hiện nay có rất nhiều nguồn tìm kiếm thông tin liên quan đến du lịch tuy nhiên còn có những hạn chế: □

+ Dữ liệu về du lịch rất đa dạng và phong phú tuy nhiên đang nằm phân tán và rải rác ở rất nhiều website khác nhau. Có những ứng dụng chuyên về dịch vụ

lưu trú, có những ứng dụng chuyên về điểm đến hay về ăn uống, đi lại, mua sắm nên chưa có sự tổng hợp thống nhất thành một kho dữ liệu chung dẫn đến khó khăn trong việc tìm kiếm. □

+ Chưa hỗ trợ tìm kiếm theo ngữ nghĩa và tương đồng ngữ nghĩa để có thể tìm kiếm được chính xác theo nội dung cần tìm kiếm. □

+ Các công cụ tìm kiếm chưa có những khuyến nghị đa dạng về các hoạt động du lịch quanh đối tượng cần tìm kiếm.

3.1.3 Hướng triển khai khắc phục

Để giải quyết vấn đề trên, em nghiên cứu xây dựng Ontology ứng dụng trong miền du lịch, tạo ra một ứng dụng tra cứu đầy đủ từ tìm kiếm các cơ sở lưu trú đến ăn uống, điểm đến, mua sắm và vận chuyển. Khách du lịch chỉ cần tra cứu trên một ứng dụng duy nhất nhưng có đầy đủ thông tin cần thiết cho một chuyến du lịch.

Du lịch được coi là một ngành chủ chốt trong nền kinh tế, việc tạo ra một ứng dụng hỗ trợ, tạo điều kiện thuận lợi cho du khách là thực sự cần thiết góp phần thúc đẩy ngành du lịch phát triển.

Qua những thông tin và kiến thức thu thập được ở trên, tôi đề xuất phát triển hệ thống tìm kiếm các thông tin về du lịch tại Việt Nam theo các hướng như sau:

- Xây dựng ontology chuyên sâu trong miền du lịch Việt Nam, trước tiên sẽ tập trung xây dựng cho một số tỉnh thành trọng điểm như Hà Nội, Hồ Chí Minh, Đà Nẵng, Hà Giang, Lào Cai, Quảng Ninh, Quảng Bình, Thừa Thiên Huế, Quảng Nam, Khánh Hòa, Kiên Giang sau đó sẽ mở rộng ra cho 63 tỉnh thành trên toàn quốc
- Xây dựng ứng dụng tìm kiếm du lịch, dựa trên công nghệ Sematic Web nhằm tạo điều kiện cho khách du lịch có thể tìm kiếm được các thông tin cần thiết trước mỗi chuyến đi.

3.2. Nghiên cứu một số kịch bản tìm kiếm thông tin du lịch

- **Kịch bản 1:** Khách du lịch tìm kiếm một địa điểm du lịch, lưu trú, ăn uống, nơi mua sắm, phương tiện vận chuyển cụ thể theo từ khóa. Khó khăn hiện tại: Dữ liệu trả về đa dạng, tuy nhiên còn chứa nhiều thông tin không liên quan,

chưa có sự liên kết và khuyến nghị đến các cơ sở lưu trú, điểm đến, nơi ăn uống, mua sắm xung quanh để khách du lịch có thêm các lựa chọn phong phú

Kịch bản mong muốn: ☐

+ Hệ thống trả về tất cả các kết quả tìm kiếm chứa từ khóa trên với các thông tin liên quan như tên gọi, địa chỉ. ☐

+ Khi khách hàng chọn kết quả mong muốn hệ thống trả về kết quả chi tiết, các khái niệm tương đồng về ngữ nghĩa, kèm theo các khuyến nghị các cơ sở lưu trú, điểm đến, nơi ăn uống, mua sắm xung quanh để khách du lịch có thêm các lựa chọn phong phú

- **Kịch bản 2:** Khách du lịch tìm kiếm một địa điểm lưu trú theo các điều kiện lọc như phân loại cơ sở lưu trú, giới hạn khu vực, hạng sao, đánh giá, ở gần khu vực cụ thể, ... Khó khăn hiện tại: Đã có nhiều ứng dụng tìm kiếm nhưng chưa có sự liên kết và khuyến nghị đến các cơ sở lưu trú, điểm đến, nơi ăn uống, mua sắm xung quanh để khách du lịch có thêm các lựa chọn phong phú

Kịch bản mong muốn: ☐

+ Hệ thống trả về danh sách thỏa mãn các điều kiện ☐

+ Khi khách hàng chọn kết quả mong muốn hệ thống trả về kết quả chi tiết, các khái niệm tương đồng về ngữ nghĩa, kèm theo các khuyến nghị các cơ sở lưu trú, điểm đến, nơi ăn uống, mua sắm xung quanh để khách du lịch có thêm các lựa chọn phong phú

- **Kịch bản 3:** Khách du lịch tìm kiếm một địa điểm du lịch theo các điều kiện lọc như phân loại điểm đến, giới hạn khu vực, ở gần khu vực cụ thể Khó khăn hiện tại: Số lượng ứng dụng hỗ trợ còn hạn chế, chưa có sự liên kết và khuyến nghị đến các cơ sở lưu trú, điểm đến, nơi ăn uống, mua sắm xung quanh để khách du lịch có thêm các lựa chọn phong phú

Kịch bản mong muốn: ☐

+ Hệ thống trả về danh sách thỏa mãn các điều kiện ☐

+ Khi khách hàng chọn kết quả mong muốn hệ thống trả về kết quả chi tiết, các khái niệm tương đồng về ngữ nghĩa, kèm theo các khuyến nghị các cơ sở lưu

trú, điểm đến, nơi ăn uống, mua sắm xung quanh để khách du lịch có thêm các lựa chọn phong phú

- **Kịch bản 4:** Khách du lịch tìm kiếm một địa điểm ăn uống theo các điều kiện lọc như phân loại địa điểm ăn uống, giới hạn khu vực, đánh giá, ở gần khu vực cụ thể 25 Khó khăn hiện tại: Đã có ứng dụng tìm kiếm nhưng chưa có chức năng khuyến nghị các cơ sở lưu trú, điểm đến, nơi ăn uống, mua sắm xung quanh để khách du lịch có thêm các lựa chọn phong phú

Kịch bản mong muốn: ☐

+ Hệ thống trả về danh sách thỏa mãn các điều kiện ☐

+ Khi khách hàng chọn kết quả mong muốn hệ thống trả về kết quả chi tiết, các khái niệm tương đồng về ngữ nghĩa, kèm theo các khuyến nghị các cơ sở lưu trú, điểm đến, nơi ăn uống, mua sắm xung quanh để khách du lịch có thêm các lựa chọn phong phú

- **Kịch bản 5:** Khách du lịch tìm kiếm một địa điểm mua sắm theo các điều kiện lọc như phân loại địa điểm mua sắm, giới hạn khu vực, đánh giá, ở gần khu vực cụ thể. Khó khăn hiện tại: Chưa có ứng dụng hỗ trợ tìm kiếm địa điểm mua sắm theo các điều kiện lọc cụ thể.

Kịch bản mong muốn: ☐

+ Hệ thống trả về danh sách thỏa mãn các điều kiện ☐

+ Khi khách hàng chọn kết quả mong muốn hệ thống trả về kết quả chi tiết, các khái niệm tương đồng về ngữ nghĩa, kèm theo các khuyến nghị các cơ sở lưu trú, điểm đến, nơi ăn uống, mua sắm xung quanh để khách du lịch có thêm các lựa chọn phong phú

- **Kịch bản 6:** Khách du lịch tìm kiếm một loại phương tiện vận chuyển như taxi, xe buýt, tàu điện, ... Khó khăn hiện tại: Phải tra cứu thông tin trên nhiều trang web khác nhau cho mỗi loại hình dịch vụ vận chuyển.

Kịch bản mong muốn: ☐

+ Hệ thống trả về danh sách thỏa mãn các điều kiện ☐

+ Khách hàng chọn phương tiện phù hợp

3.3 Thu thập dữ liệu trong miền du lịch

- Nguồn thu thập dữ liệu

STT	Nguồn	Trang web
1	Tổng cục du lịch Việt Nam	https://vietnamtourism.gov.vn/
2	Sở du lịch Hà Nội	https://sodulich.hanoi.gov.vn/
3	Sở du lịch Hồ Chí Minh	http://sodulich.hochiminhcity.gov.vn/
4	Cổng thông tin du lịch Quảng Ninh	http://halongtourism.com.vn
5	Thông tin địa điểm lưu trú, điểm đến, nhà hàng, nơi mua sắm	https://www.google.com/maps

3.4 Xây dựng Ontology phục vụ tra cứu thông tin về du lịch tại Việt Nam

✓ Bước 1: Xác định lĩnh vực và phạm vi của Ontology

- **Lĩnh vực:** Xây dựng Ontology trong miền du lịch, phục vụ tra cứu thông tin về du lịch tại Việt Nam.

- **Mục đích:** Hỗ trợ cho khách du lịch có thể tìm kiếm các cơ sở lưu trú, lựa chọn địa điểm tham quan, mua sắm, phương tiện vận chuyển đi lại theo sở thích.

- **Đối tượng phục vụ:** Khách du lịch trong và ngoài nước có nhu cầu tìm kiếm thông tin liên quan đến chuyến đi như khách sạn, nhà hàng, các điểm du lịch hấp dẫn, ...

- **Phạm vi:** Các tỉnh thành phố là Hà Nội, Hà Giang, Lào Cai, Hải Phòng, Quảng Ninh, Quảng Bình, Thừa Thiên Huế, Đà Nẵng, Quảng Nam, Khánh Hòa, Thành phố Hồ Chí Minh, Kiên Giang.

✓ Bước 2: Xem xét việc sử dụng lại các Ontology có sẵn

- Hiện chưa có Ontology trong miền du lịch tại Việt Nam chính thức để kế thừa và tiếp tục phát triển.

✓ Bước 3: Liệt kê các thuật ngữ quan trọng

- Các thuật ngữ liên quan đến nơi ở, lưu trú: Khách sạn, nhà nghỉ, homestay, apartment sẽ có các thuộc tính là: tên gọi, cơ sở vật chất, hạng sao, điểm đánh giá, số lượng đánh giá, giá phòng, trang web, địa chỉ, số điện thoại.

- Các thuật ngữ liên quan đến đại điểm ăn uống: Các nhà hàng, quán ăn, món ăn ngon, khu ẩm thực, quán bar, quán café, quán trà có các thuộc tính là: tên gọi, điểm đánh giá, số lượng đánh giá, địa chỉ, số điện thoại.

- Các thuật ngữ liên quan đến địa điểm mua sắm: chợ, cửa hàng, trung tâm thương mại có các thuộc tính là: tên gọi, địa chỉ, số điện thoại.

- Các thuật ngữ liên quan đến điểm đến thu hút: di sản di tích, bảo tàng, công viên, sông, núi, hồ, cầu, sinh thái nghỉ dưỡng, tham quan ngắm cảnh, vui chơi giải trí, làng nghề, bãi biển, bán đảo, vịnh, hang động ... có các thuộc tính là: tên gọi, địa chỉ, số điện thoại.

- Các thuật ngữ liên quan đến vận chuyển: hàng không, sân bay, đường sắt, taxi, xe buýt có các thuộc tính là: tên gọi, địa chỉ, số điện thoại.

- Các thuật ngữ liên quan đến địa chỉ: tỉnh, thành phố, quận, huyện, thị xã, phường, xã.

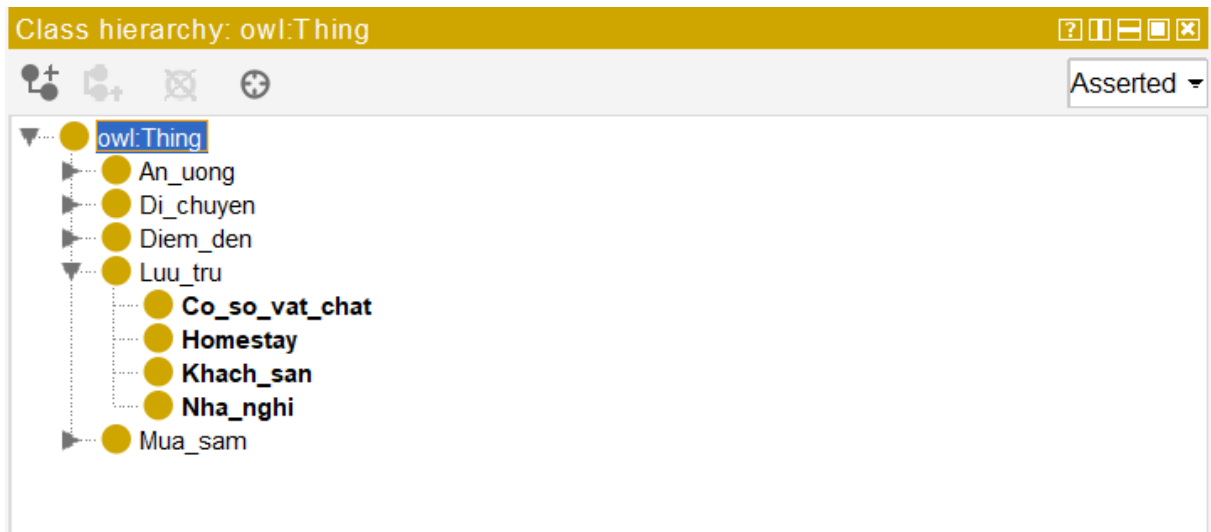
Cơ sở lưu trú	Điểm đến	Ăn uống	Mua sắm	Vận chuyển	Địa chỉ
<ul style="list-style-type: none">• Khách sạn,• Nhà nghỉ,• Homestay,• Apartment• Cơ sở vật chất• Hạng sao• ...	<ul style="list-style-type: none">• Di sản di tích• Sinh thái nghỉ dưỡng• Bảo tàng• Công viên• Bãi biển• ...	<ul style="list-style-type: none">• Nhà hàng• Quán ăn• Món ăn• Khu ẩm thực• Quán café• Quán bar• ...	<ul style="list-style-type: none">• Trung tâm thương mại• Chợ• Cửa hàng• ...	<ul style="list-style-type: none">• Sân bay• Đường sắt• Taxi• Xe buýt• Tàu điện• ...	<ul style="list-style-type: none">• Tỉnh/thành phố• Quận/huyện• Phường xã

Hình 3.3: Liệt kê các thuật ngữ quan trọng

✓ Bước 4: Xác định các lớp và phân cấp của các lớp

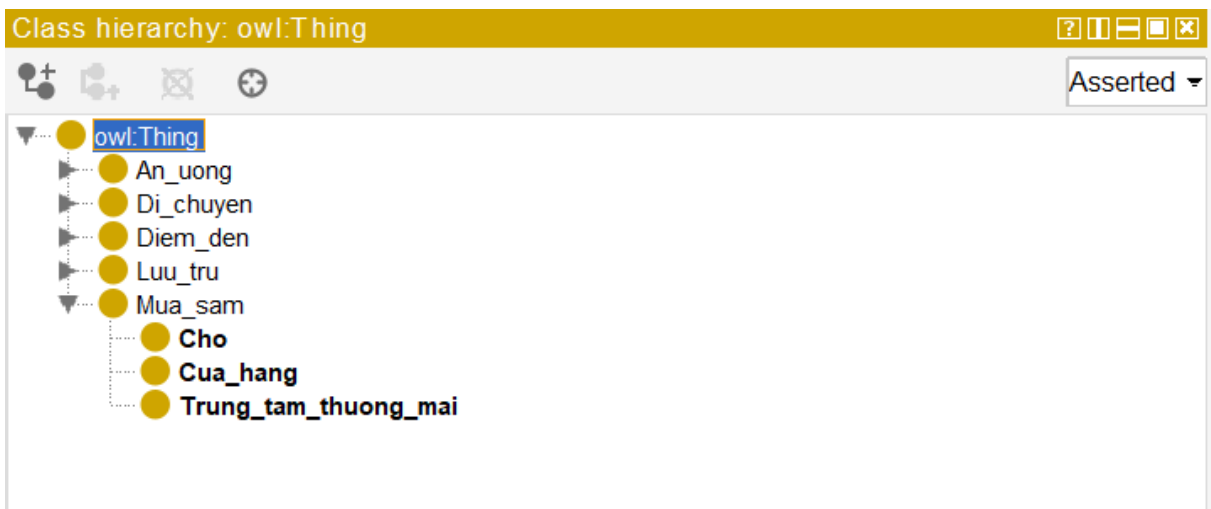
Dựa trên các thuật ngữ đã liệt kê ở Bước 3, chúng ta sẽ đưa các thuật ngữ vào các lớp như sau:

- **Lớp Lưu trú:** Bao gồm các phân lớp con là Khách_sạn, Homestay, Co_so_vat_chat, Nha_nghi



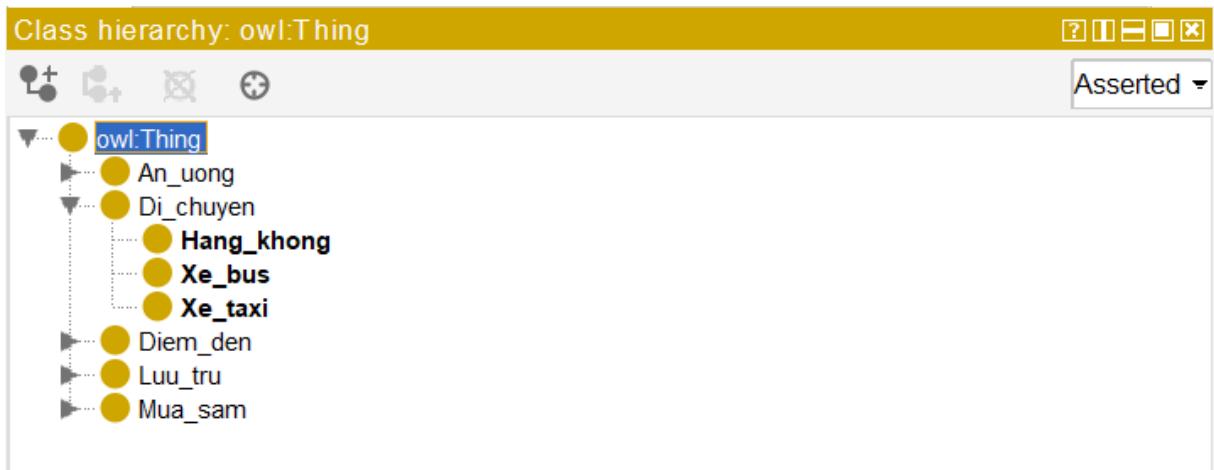
Hình 3.4: Lớp Lưu trú trong Ontology

- **Lớp Mua sắm:** Bao gồm các phân lớp con là Cho, Cua_hang, Trung_tam_thuong_mai,



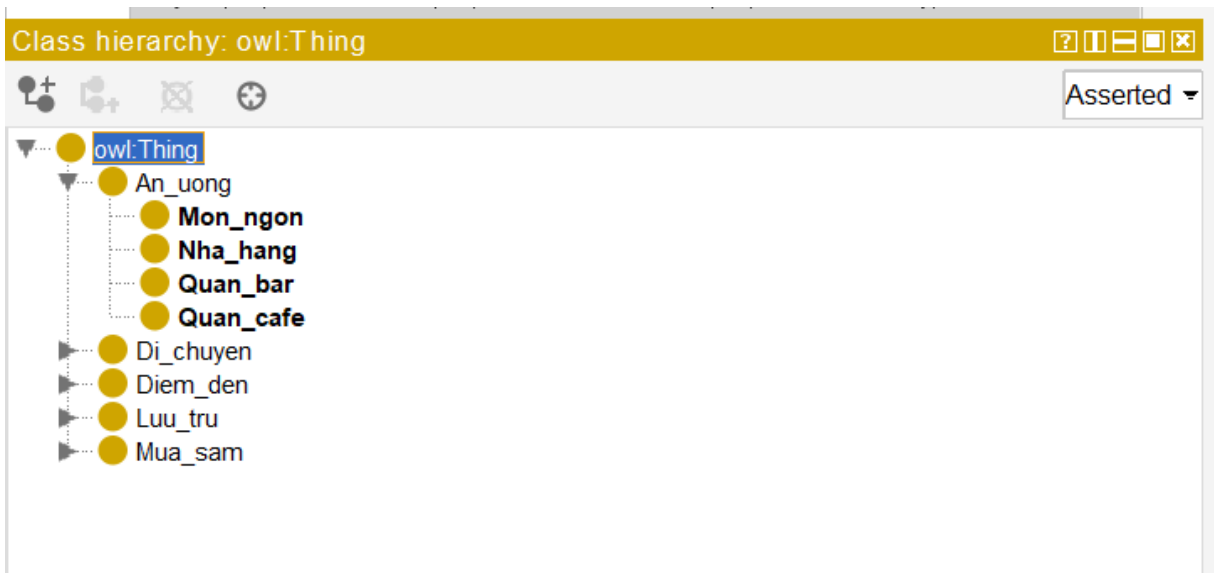
Hình 3.5: Lớp Mua sắm trong Ontology

- **Lớp Di chuyển:** Bao gồm các phân lớp con là Hang_khong, xe_bus, xe_taxi



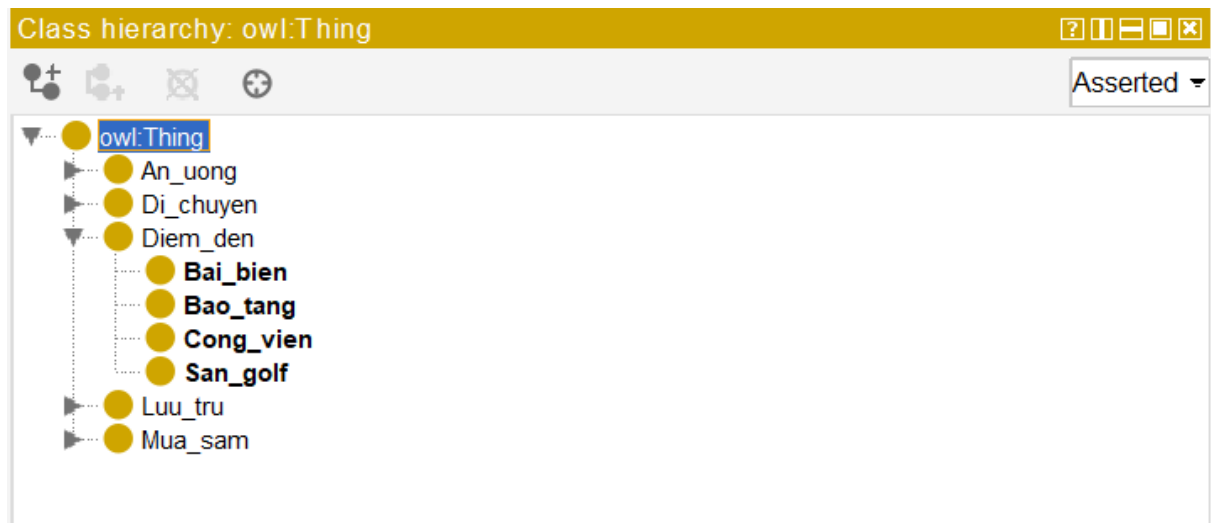
Hình 3.6: Lớp Di chuyển trong Ontology

- **Lớp Ăn uống:** Bao gồm các phân lớp con là Món_ngon, Nhà_hàng, Quán_bar, Quán_cafe



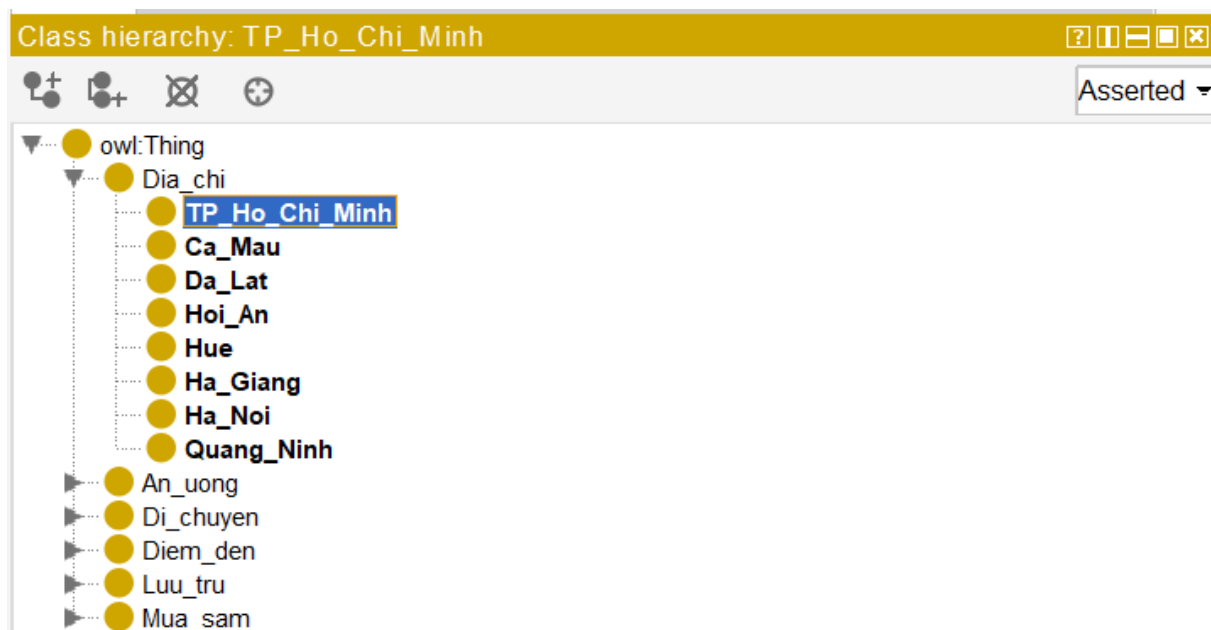
Hình 3.7: Lớp Ăn uống trong Ontology

- **Lớp Điểm đến:** Bao gồm các phân lớp con Bãi biển, Bảo tàng, Công viên, Sân golf, công viên

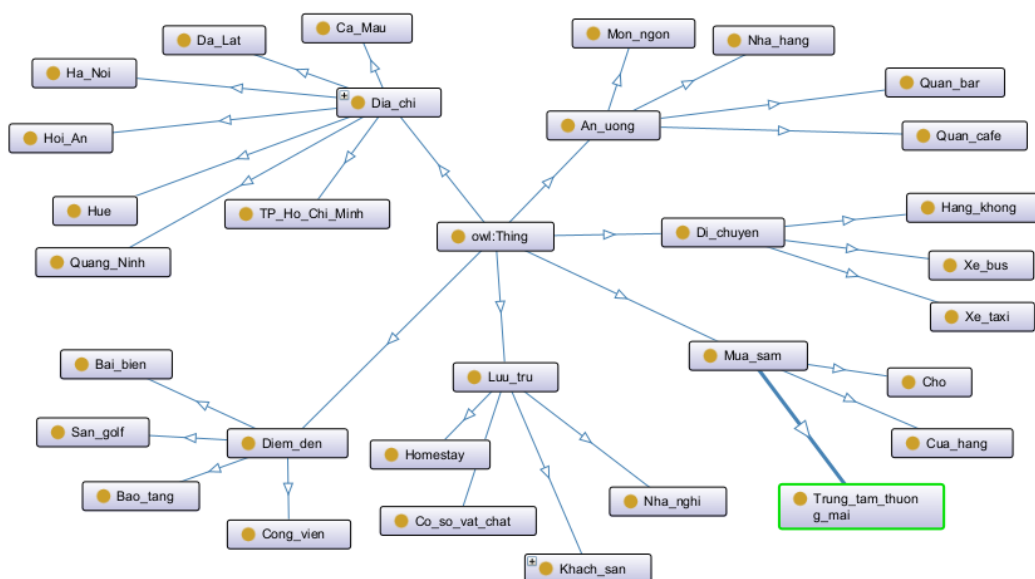


Hình 3.8: Lớp điểm đến trong Ontology

- **Lớp Địa chỉ:** Bao gồm các phân lớp con là các tỉnh, thành phố du lịch trọng điểm tại Việt Nam: TP_Ho_Chi_Minh, Ca_Mau, Da_Lat, Hoi_An, Hue, Ha_Giang, Ha_Noi, Quang_Ninh. Mỗi tỉnh, thành phố có các lớp con là các đơn vị hành chính quận, huyện, thị xã trực thuộc. Mỗi quận huyện, thị xã có các lớp con các đơn vị thành chính xã, phường trực thuộc.



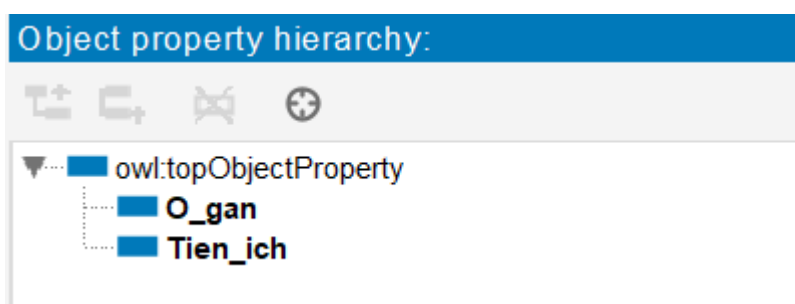
Hình 3.9: Lớp Địa chỉ trong Ontology



Hình 3.10: Ontology Graf

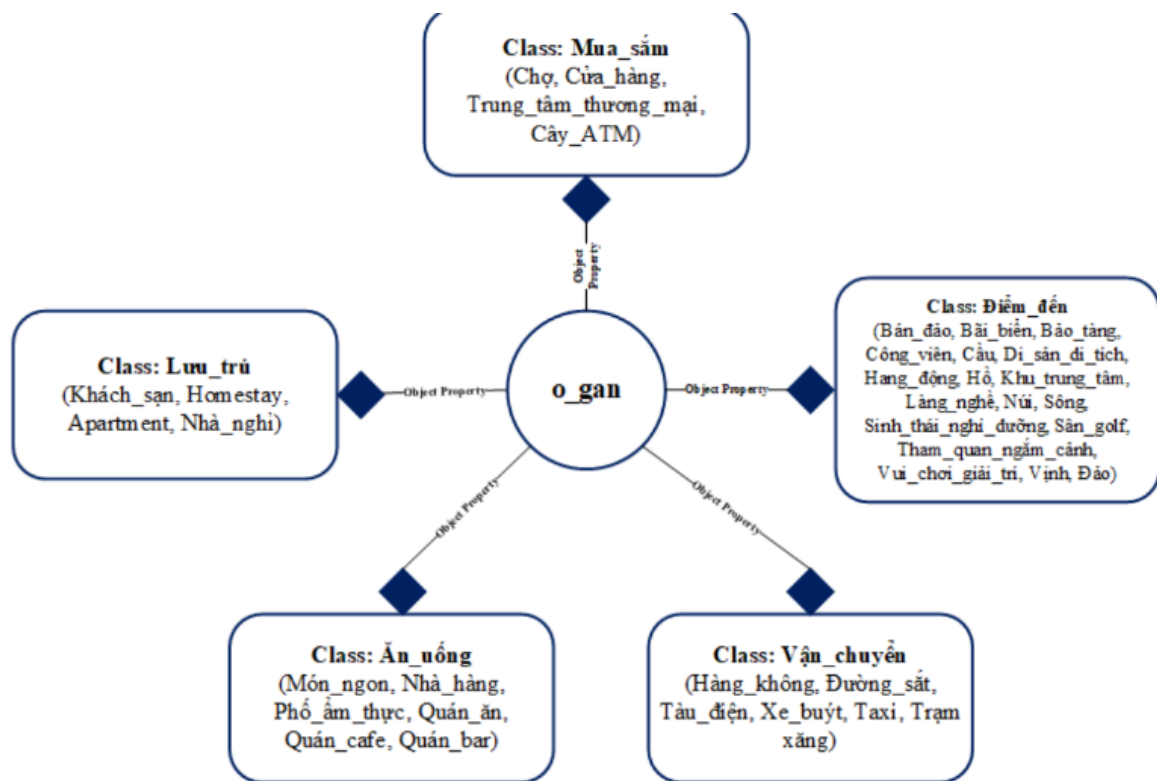
✓ Bước 5: Xác định các thuộc tính

- Xác định các thuộc tính đối tượng (ObjectProperty)

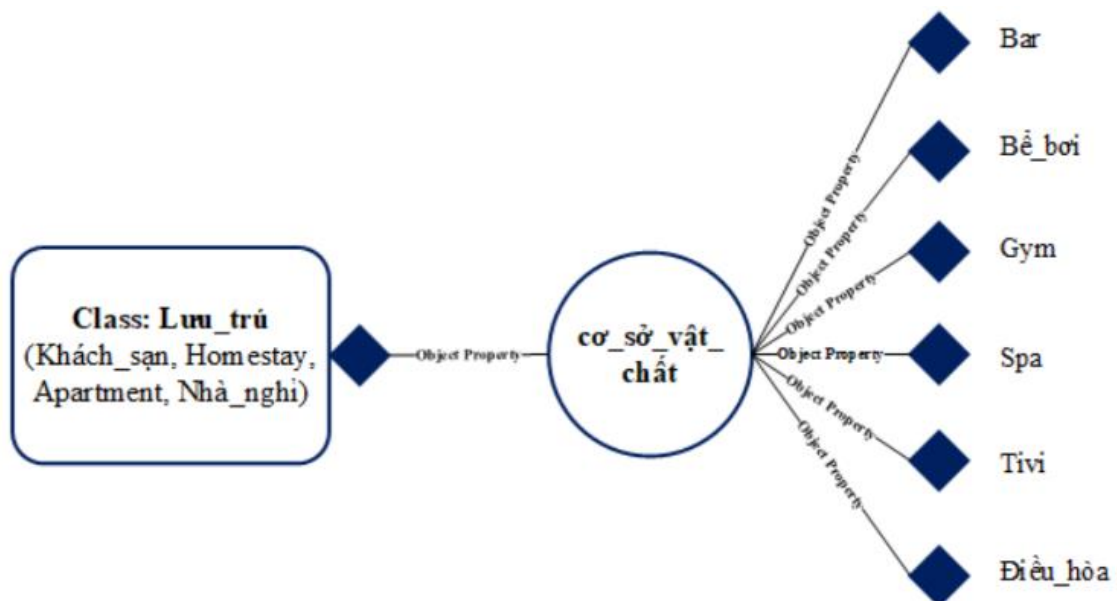


Hình 3.11: Các thuộc tính đối tượng trong Ontology

Thuộc tính “o_gan” là thuộc tính liên kết các thực thể ở gần nhau về mặt địa lý (ví dụ: Khách sạn Trầg Tiên ở gần Hồ Hoàn Kiếm) Thuộc tính “tien_ich” là thuộc tính liên kết các thực thể trong lớp lưu trú với các thực thể liên quan đến cơ sở vật chất như: Bar, Bể_bơi, Gym, Spa, Tivi, Điều_hòa

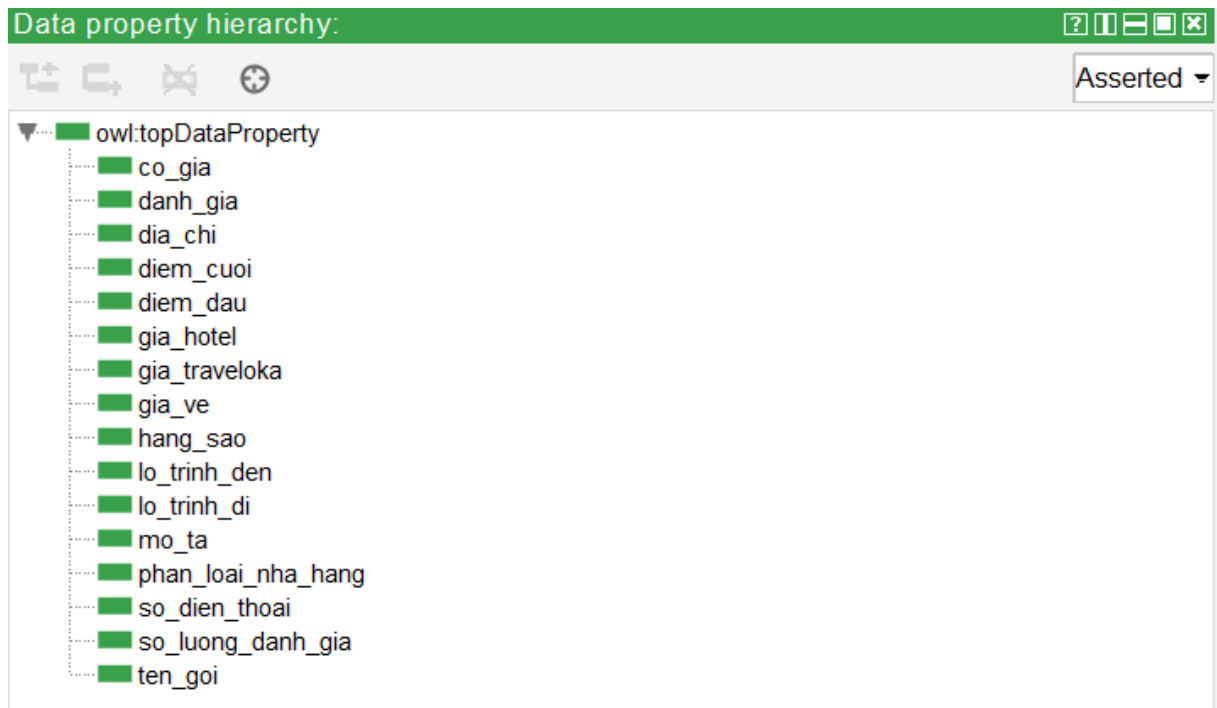


Hình 3.12: Biểu diễn thuộc tính ở gần trong Ontology

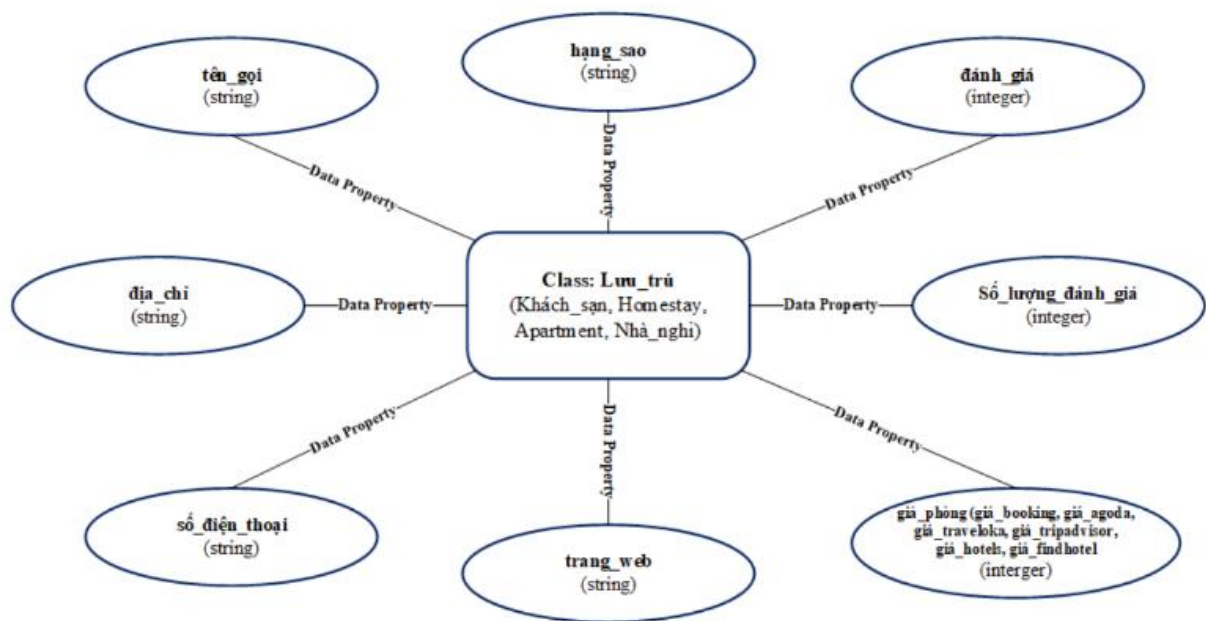


Hình 3.13: Biểu diễn thuộc tính cơ sở vật chất trong Ontology

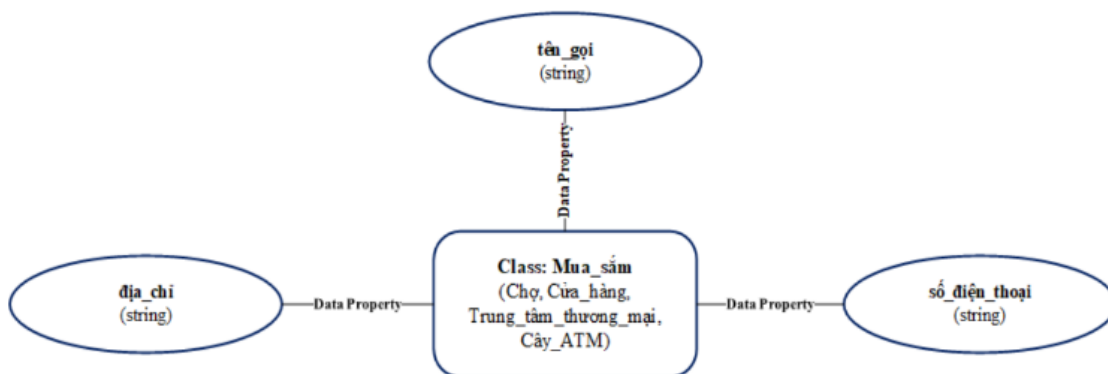
- Xác định các thuộc tính dữ liệu DataProperty



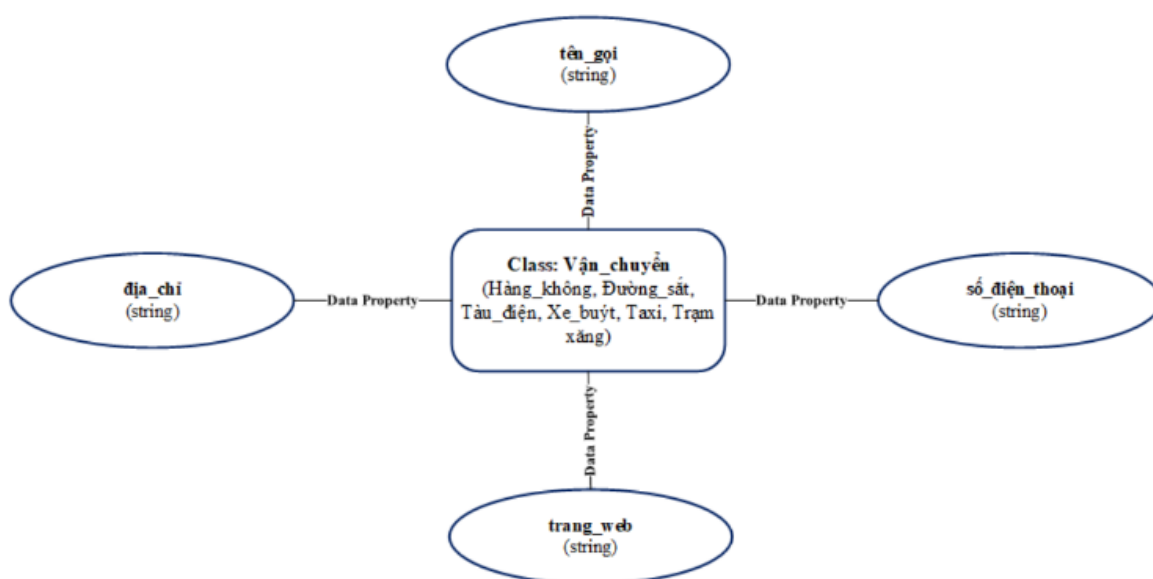
Hình 3.14: Danh sách các thuộc tính kiểu dữ liệu trong Ontology



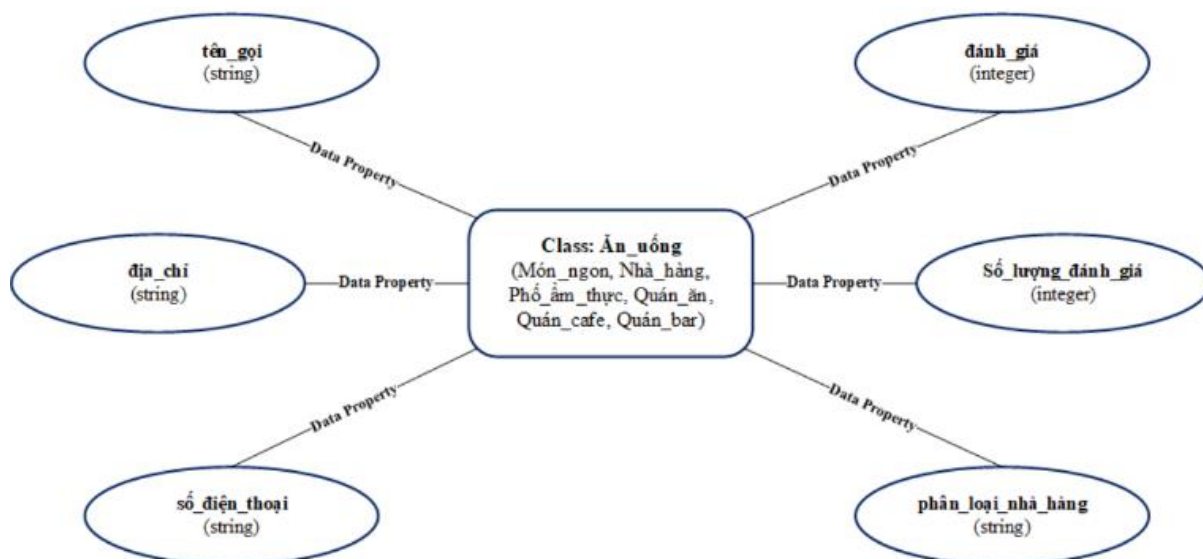
Hình 3.15: Biểu diễn các thuộc tính gắn với lớp Lưu trú



Hình 3.16: Biểu diễn các thuộc tính gắn với lớp Mua sắm



Hình 3.17: Biểu diễn các thuộc tính gắn với lớp Vận chuyển



Hình 3.18: Biểu diễn các thuộc tính gắn với lớp Ăn uống

✓ **Bước 6: Xác định giới hạn của các thuộc tính**

STT	Thuộc tính	Kiểu dữ liệu	Miền
1	Ten_goi	string	Thing
2	Hang_sao	string	Khach_san
3	Dia_chi	string	Thing
4	So_dien_thoai	string	Thing
5	Danh_gia	interger	Thing
6	So_luong_danh_gia	interger	Thing
7	Gia_hotel	interger	Luu_tru
8	Diem_cuoi	string	Di_chuyen
9	Diem_dau	string	Di_chuyen
10	Lo_trinh_den	string	Di_chuyen
11	Lo_trinh_di	string	Di_chuyen
12	Mo_ta	string	thing
13	Phan_loai_nha_hang	string	Nha_hang
14	Quan_huyen	string	Thing
15	Trang_web	string	Thing
16	Gia_ve	interger	Thing
17	Co_gia	interger	Thing
18	Gia_traveloka	interger	Luu_tru

Bảng 3.2: Giới hạn của các thuộc tính kiểu dữ liệu

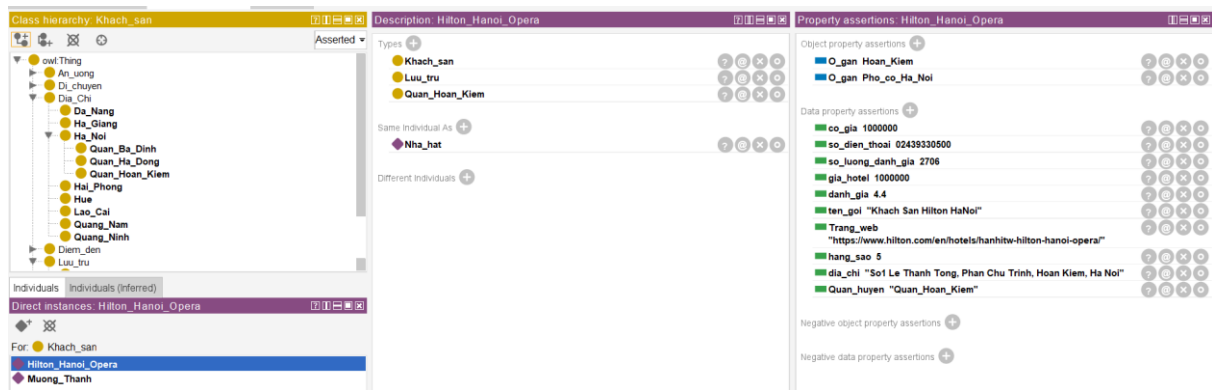
No	Object Properties	Characteristics	Remarks
1	O_gan	Symmetric	Thuộc tính đối xứng
2	Tien_ich		

Bảng 3.3: Giới hạn của các thuộc tính đối tượng

✓ Bước 7: Tạo các thực thể

- Tạo thực thể trong lớp lưu trữ:

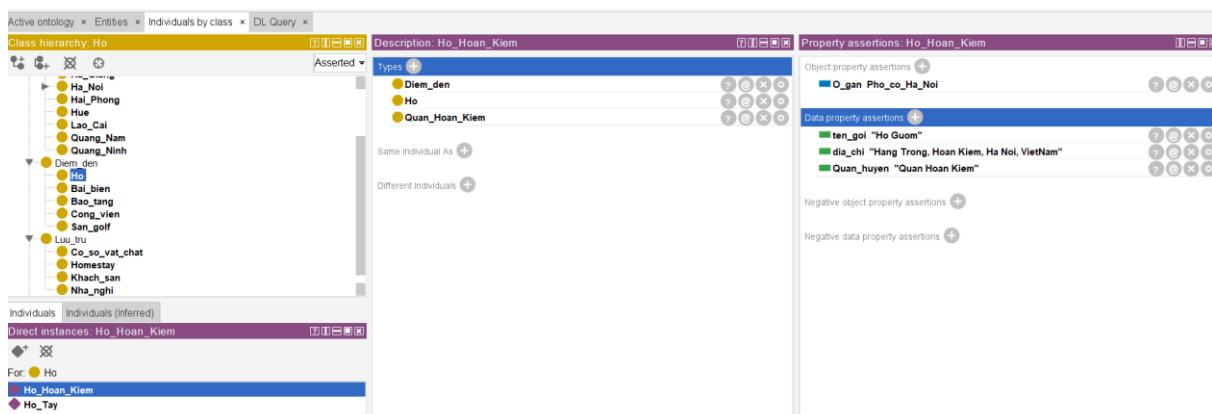
- ☐ Tên thực thể: Ví dụ Khách sạn Hilton Hanoi Opera
- ☐ Giới hạn domain về loại lưu trú: Khách sạn
- ☐ Giới hạn domain về khu vực: Quận Hoàn Kiếm
- ☐ Bổ sung các thuộc tính kiểu dữ liệu cho thực thể:
 - ☐+ Tên gọi: Khách sạn Hilton Hanoi Opera
 - ☐+ Địa chỉ: Số 1 Lê Thánh Tông, Phan Chu Trinh, Hoàn Kiếm, Hà Nội
 - ☐+ Số điện thoại: 02439330500
 - ☐+ Hạng sao: 5 sao
 - ☐+ Trang web: <https://www.hilton.com/en/hotels/hanhitw-hilton-hanoi-opera/>
 - ☐+ Đánh giá: 4.4
 - ☐+ Số lượng đánh giá: 2706
- ☐ Bổ sung các thuộc tính đối tượng ở gần, cơ sở vật chất
 - ☐+ Khách sạn Hilton Hanoi Opera ở gần Hồ Hoàn Kiếm
 - ☐+ Khách sạn Hilton Hanoi Opera ở gần Phố cổ Hà Nội
 - ☐+ Khách sạn Hilton Hanoi Opera có các cơ sở vật chất: Bể bơi, Gym, Bar, Điều hòa, Tivi, Tủ lạnh, Điện thoại



Hình 3.19: Tạo các thực thể trong lớp Khách sạn

- Tạo thực thể trong lớp Điểm đến:

- ☐ Tên thực thể: Ví dụ Hồ Gươm
- ☐ Giới hạn domain về loại điểm đến: Hồ
- ☐ Giới hạn domain về khu vực: Quận Hoàn Kiếm
- ☐ Các tên gọi khác của Hồ Gươm là Hồ Hoàn Kiếm và Hồ Lục Thủy
- ☐ Bổ sung các thuộc tính kiểu dữ liệu cho thực thể:
 - ☐ + Tên gọi: Hồ Gươm
 - ☐ + Địa chỉ: Hàng Trống, Hoàn Kiếm, Hà Nội, Việt Nam
- ☐ Bổ sung các thuộc tính đối tượng: Hồ Gươm ở gần Phố cổ Hà Nội

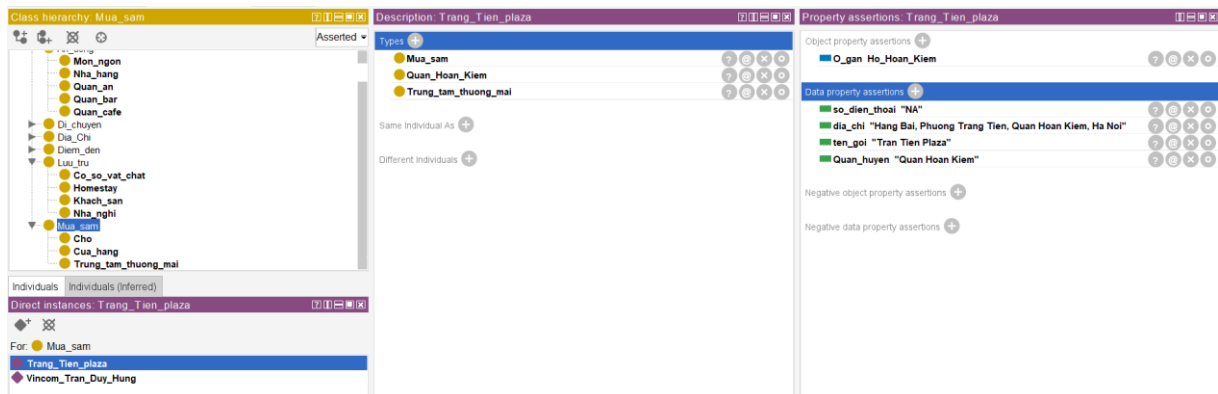


Hình 3.20: Tạo các thực thể trong lớp Điểm đến

- Tạo thực thể trong lớp Mua sắm:

- Tên thực thể: Ví dụ Tràng Tiền Plaza

- Giới hạn domain về loại địa điểm mua sắm: Trung tâm thương mại
- Giới hạn domain về khu vực: Quận Hoàn Kiếm
- Bổ sung các thuộc tính kiểu dữ liệu cho thực thể:
 - + Tên gọi: Tràng Tiền Plaza
 - + Địa chỉ: 24 P. Hai Bà Trưng, Tràng Tiền, Hoàn Kiếm, Hà Nội
 - + Số điện thoại: 19003149
- □ Bổ sung các thuộc tính đối tượng: Tràng Tiền Plaza ở gần Hồ Hoàn Kiếm



Hình 3.21 : Tạo các thực thể trong lớp Mua sắm

- Tạo thực thể trong lớp Ăn uống:

- Tên thực thể: Ví dụ Nhà hàng Thủy Tạ
- Giới hạn domain về loại địa điểm ăn uống: Nhà hàng
- Giới hạn domain về khu vực: Quận Hoàn Kiếm
- Bổ sung các thuộc tính kiểu dữ liệu cho thực thể:

+ Tên gọi: Nhà hàng Thủy Tạ

+ Địa chỉ: Số 1 Lê Thái Tổ, Hàng Trống, Hoàn Kiếm, Hà Nội

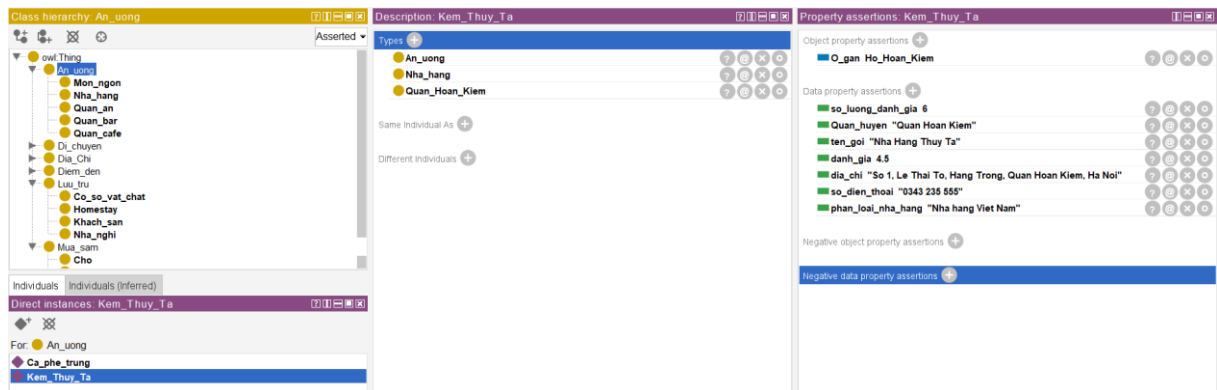
+ Số điện thoại: 0343 235 555

+ Loại nhà hàng: Nhà hàng Việt Nam

+ Đánh giá: 4.5

+ Số lượng đánh giá: 6

- Bổ sung các thuộc tính đối tượng: Nhà hàng Thủy Tạ ở gần Hồ Hoàn Kiếm



Hình 3.22: Tạo các thực thể trong lớp Nhà hàng

CHƯƠNG 4: TRIỂN KHAI ỨNG DỤNG ONTOLOGY TRONG MIỀN DU LỊCH

4.1 Môi trường triển khai thực nghiệm Bảng

4.1.1: Cấu hình máy thử nghiệm

- CPU Intel(R) Core(TM) i5-8400 CPU @ 2.80GHz (6 CPUs), ~2.8GHz
- RAM 1GB
- OS Windows 10 Home 64-bit
- Disk 128GB SSD + 1TB HDD

4.1.2 Môi trường và các công cụ sử dụng trong luận văn:

- Môi trường run-time: Java SE Development Kit
- Ngôn ngữ phát triển ứng dụng web: Java,
- Ngôn ngữ mô tả ngữ nghĩa: RDF, OWL
- Công cụ xây dựng Ontology: Protégé
- Công cụ truy vấn dữ liệu: SPARQL
- Thư viện hỗ trợ truy vấn Ontology: Jena
- Trình duyệt: Google chrome

4.2 Mục tiêu triển khai thử nghiệm

- Xây dựng ứng dụng web để tìm kiếm thông tin về du lịch dựa trên Ontology đã hoàn thiện ở phần trên bao gồm các thông tin về các cơ sở lưu trú, mua sắm, vận chuyển, ăn uống, điểm đến hấp dẫn. Khách du lịch chỉ cần tra cứu trên một ứng dụng duy nhất nhưng có đầy đủ thông tin cần thiết cho một chuyến du lịch.

- Đáp ứng các chức năng tìm kiếm một cách tối ưu, hỗ trợ tìm kiếm theo ngữ nghĩa, tương đồng ngữ nghĩa và đưa ra các khuyến nghị phù hợp, nêu bật được sự khác biệt của ứng dụng so với các trang web tìm kiếm thông tin hiện tại.

4.3 Thiết kế ứng dụng

4.3.1 Thiết kế chức năng ứng dụng □

- **Chức năng tìm kiếm theo từ khóa**

Người dùng gõ từ khóa cần tìm kiếm vào ô tìm kiếm (ví dụ: “Hồ Hoàn Kiếm”). Hệ thống sẽ chuyển từ khóa này thành câu lệnh sparql để truy vấn file ontology và hiển thị thông tin các kết quả có chứa từ khóa này. □

- **Chức năng tìm kiếm theo điều kiện lọc**

Người dùng tìm kiếm bằng cách vào các tỉnh thành phố cụ thể sau đó chọn mục cần tìm kiếm cơ sở lưu trú, ăn uống, điểm đến, mua sắm, vận chuyển sau đó người dùng chọn cụ thể theo các điều kiện lọc khác nhau.

Hệ thống sẽ phân tích các điều kiện lọc này và tìm kiếm tất cả các đối tượng thỏa mãn điều kiện. □

- **Chức năng tìm kiếm tương đồng ngữ nghĩa**

Hệ thống phân tích được các đối tượng tương đồng ngữ nghĩa để hiển thị cho người dùng. Ví dụ: Hồ Gươm, Hồ Hoàn Kiếm, Hồ Lục Thủy là cùng là một địa danh với các tên gọi khác nhau. □

- **Chức năng khuyến nghị**

Hệ thống phân tích được mối quan hệ giữa các đối tượng để đưa ra các khuyến nghị phù hợp: □

Khuyến nghị các địa điểm du lịch, lưu trú, ăn uống, mua sắm quanh một địa điểm đang tham quan. □

Khuyến nghị các đối tượng khác có cùng một, hay nhiều điểm chung nào đấy như cùng một chuỗi vận hành, cùng một chủ sở hữu.

4.3.2 Thiết kế các kịch bản tìm kiếm

- **Kịch bản 1:** Khách du lịch tìm kiếm một địa điểm du lịch, lưu trú, ăn uống, nơi mua sắm, phương tiện vận chuyển cụ thể theo từ khóa.

Kết quả: □

Hệ thống trả về tất cả các kết quả tìm kiếm chứa từ khóa trên với các thông tin liên quan như tên gọi, địa chỉ. ☐

Khi khách hàng chọn kết quả mong muốn hệ thống trả về kết quả chi tiết, các khái niệm tương đồng về ngữ nghĩa, kèm theo các khuyến nghị các cơ sở lưu trú, điểm đến, nơi ăn uống, mua sắm xung quanh để khách du lịch có thêm các lựa chọn phong phú

- **Kịch bản 2:** Khách du lịch tìm kiếm một địa điểm lưu trú theo các điều kiện lọc như phân loại cơ sở lưu trú, giới hạn khu vực, hạng sao, đánh giá, ở gần khu vực cụ thể

Kết quả: ☐

Hệ thống trả về danh sách thỏa mãn các điều kiện ☐

Khi khách hàng chọn kết quả mong muốn hệ thống trả về kết quả chi tiết, các khái niệm tương đồng về ngữ nghĩa, kèm theo các khuyến nghị các cơ sở lưu trú, điểm đến, nơi ăn uống, mua sắm xung quanh để khách du lịch có thêm các lựa chọn phong phú

- **Kịch bản 3:** Khách du lịch tìm kiếm một địa điểm du lịch theo các điều kiện lọc như phân loại điểm đến, giới hạn khu vực, ở gần khu vực cụ thể

Kết quả: ☐

Hệ thống trả về danh sách thỏa mãn các điều kiện ☐

Khi khách hàng chọn kết quả mong muốn hệ thống trả về kết quả chi tiết, các khái niệm tương đồng về ngữ nghĩa, kèm theo các khuyến nghị các cơ sở lưu trú, điểm đến, nơi ăn uống, mua sắm xung quanh để khách du lịch có thêm các lựa chọn phong phú

- **Kịch bản 4:** Khách du lịch tìm kiếm một địa điểm ăn uống theo các điều kiện lọc như phân loại địa điểm ăn uống, giới hạn khu vực, đánh giá, ở gần khu vực cụ thể.

Kết quả: ☐

Hệ thống trả về danh sách thỏa mãn các điều kiện ☐

Khi khách hàng chọn kết quả mong muốn hệ thống trả về kết quả chi tiết, các khái niệm tương đồng về ngữ nghĩa, kèm theo các khuyến nghị các cơ sở lưu trú, điểm đến, nơi ăn uống, mua sắm xung quanh để khách du lịch có thêm các lựa chọn phong phú

- **Kịch bản 5:** Khách du lịch tìm kiếm một địa điểm mua sắm theo các điều kiện lọc như phân loại địa điểm mua sắm, giới hạn khu vực, đánh giá, ở gần khu vực cụ thể.

Kết quả: ☐

Hệ thống trả về danh sách thỏa mãn các điều kiện ☐

Khi khách hàng chọn kết quả mong muốn hệ thống trả về kết quả chi tiết, các khái niệm tương đồng về ngữ nghĩa, kèm theo các khuyến nghị các cơ sở lưu trú, điểm đến, nơi ăn uống, mua sắm xung quanh để khách du lịch có thêm các lựa chọn phong phú

- **Kịch bản 6:** Khách du lịch tìm kiếm một loại phương tiện vận chuyển như taxi, xe buýt, tàu điện, ...

Kết quả: ☐

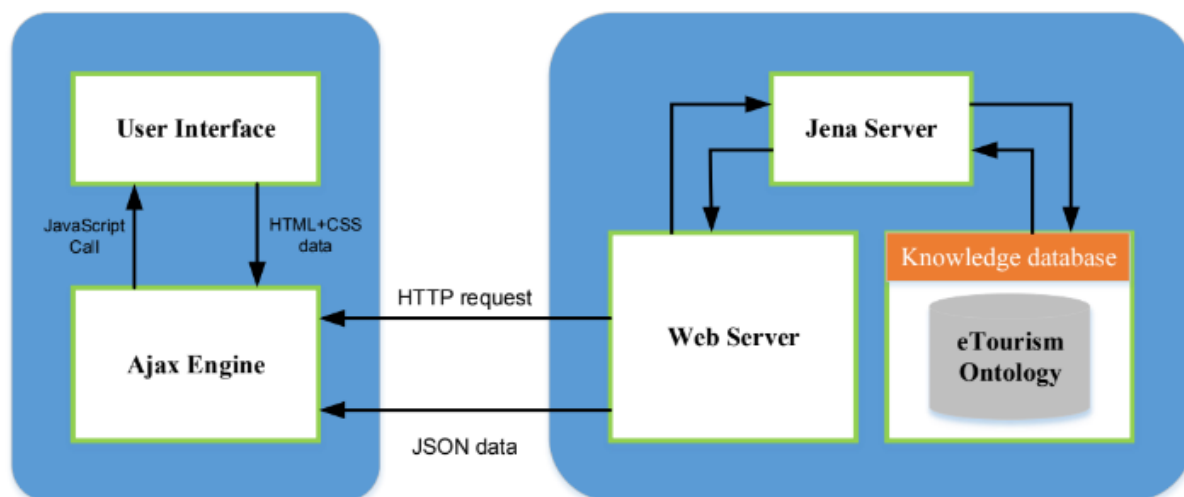
Hệ thống trả về danh sách thỏa mãn các điều kiện ☐

Khách hàng chọn phương tiện phù hợp

4.3.3 Thiết kế kiến trúc hệ thống ☐

● Kiến trúc hệ thống

Ứng dụng tra cứu thông tin du lịch là một ứng dụng thông minh được phát triển dựa trên Ontology và công nghệ web ngữ nghĩa.



Hình 4.1 mô tả kiến trúc của hệ thống

Hệ thống tra cứu thông tin du lịch được xem như là một kiến trúc hai cấp, bao gồm một trình duyệt và một hệ thống phía máy chủ. Hệ thống phía máy chủ chứa Ontology là cơ sở tri thức, Jena Server và Webserver. Ứng dụng khách trình duyệt bao gồm công cụ Ajax và giao diện người dùng. Các hoạt động phía máy khách hình thành các truy vấn người dùng được biên dịch bởi các giao diện người dùng; Các truy vấn này được gửi đến Ajax Engine. Ajax Engine gửi một yêu cầu HTTP đến máy chủ, yêu cầu này được nhận bởi Web Server, máy chủ này sẽ chuyển tiếp yêu cầu đến Jena Server, Jena Server lần lượt truy vấn Ontology, kết quả được gửi lại từ máy chủ đến máy khách và sau đó được hiển thị qua giao diện người dùng. Chi tiết về các công nghệ này được đưa ra sau đây.

- **Ajax (Asynchronous JavaScript and XML)**

Ajax là sự kết hợp của các công nghệ cho phép phát triển các ứng dụng web tương tác. Thiếu sót hiện tại của các ứng dụng web tiêu chuẩn là mỗi khi người dùng đặt một yêu cầu, một yêu cầu HTTP sẽ được gửi đến máy chủ và khi máy chủ phản hồi, toàn bộ trang sẽ được tải lại. Ajax khắc phục điều này bằng cách thêm một công cụ Ajax giữa người dùng và máy chủ. Công cụ Ajax gửi các phần nhỏ của dữ liệu đến máy chủ chứ không phải toàn bộ nội dung của trang. Điều này giúp loại bỏ nhu cầu tải lại toàn bộ trang mỗi khi máy chủ phản hồi yêu cầu.

- **Webserver**

Webserver được sử dụng để nhận, diễn giải và tạo các tài liệu JSON một cách linh động. Các tài liệu JSON được tạo sẽ được sử dụng cho mục đích trao đổi dữ liệu giữa các ứng dụng. Webserver nhận dữ liệu qua HTTP và tạo đầu ra ở định dạng JSON. Trong mô hình ứng dụng này, Webserver nhận yêu cầu HTTP dưới dạng truy vấn từ trình duyệt, truy vấn được xử lý và kết quả được chuyển tiếp trở lại trình duyệt dưới dạng tài liệu JSON.

● Jena Server

Jena Server là phần mềm kết nối trình lý luận, Ontology và giao diện người dùng. Jena là một khung công tác Java để xây dựng các ứng dụng web ngữ nghĩa; nó là mã nguồn mở và đã được phát triển bởi HP Labs. Jena bắt đầu như một API RDF, nó đã được mở rộng với API nhận biết OWL và kết nối cơ sở dữ liệu và truy vấn SPARQL. Jena có bốn hệ thống con: công cụ truy vấn, giao diện cơ sở dữ liệu, công cụ suy luận và quản lý Ontology. Kiến trúc của Jena cho phép các bộ lý luận bên ngoài được tích hợp vào các mô hình Jena. Các từ khóa tra cứu của người dùng được gửi qua giao diện người dùng tới Jena Server, mã Java chuyển các từ khóa này thành câu lệnh truy vấn SPARQL và truy vấn Ontology, sau đó kết quả truy vấn được gửi lại và hiển thị cho người dùng. Câu lệnh dưới đây là một ví dụ về mã SPARQL để thực hiện truy vấn trên Ontology du lịch thông qua Jena Server. Truy vấn này truy xuất các khách sạn 4 sao tại

Quận Hoàn Kiếm có đánh giá > 4.5 và ở gần Phố cổ Hà Nội.

```
PREFIX etourism: <http://www.semanticweb.org/tungl/ontologies/2024/4/dulich>
SELECT ?subject ?tengoi ?diachi ?dienthoai ?trangweb ?danhgia ?soluongdanhgia
WHERE { ?subject
        rdf:type etourism:Khách_sạn, etourism:Quận_Hoàn_Kiểm;
        etourism:hang_sao "4_sao";
        etourism:o_gan etourism:Pho_co_Ha_Noi;
        etourism:ten_goi ?tengoi;
        etourism:dia_chi ?diachi;
```

```

etourism:so_dien_thoai ?dienthoai;

etourism:trang_web ?trangweb;

etourism:danh_gia ?danhgia;

etourism:so_luong_danh_gia ?soluongdanhgia;

FILTER (?danhgia >4.5)

}

```

● Knowledge database

eTourism Ontology là kho tri thức trong lĩnh vực du lịch đã được tổng hợp và lưu trữ dưới dạng file owl và được tải vào cơ sở dữ liệu.

4.3.4 Thiết kế luồng nghiệp vụ chung

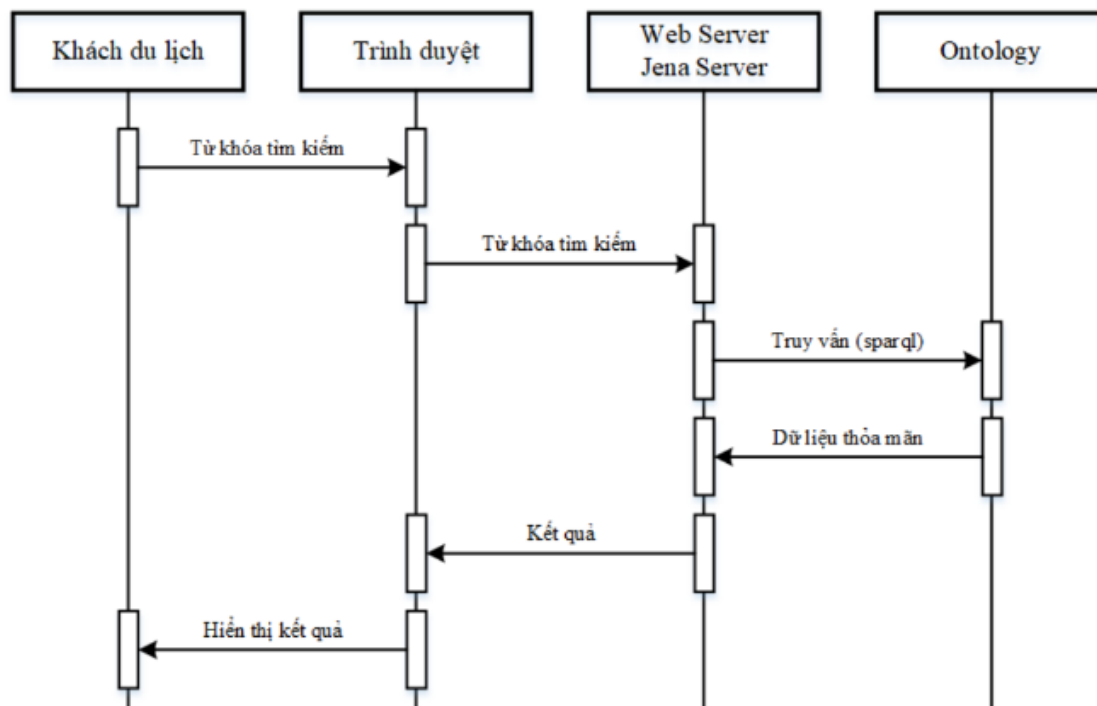
● □ Luồng đồ chung

Bước 1: Người dùng gõ từ khóa vào ô tìm kiếm hoặc lựa chọn tham số tìm kiếm theo các điều kiện lọc.

Bước 2: FrontEnd gửi API có keyword theo từ khóa và các tham số trên sang BackEnd

Bước 3: BackEnd tạo câu lệnh Sparql tương ứng để truy vấn Ontology

Bước 4: BackEnd nhận kết quả truy vấn gửi lại cho FrontEnd để hiển thị kết quả

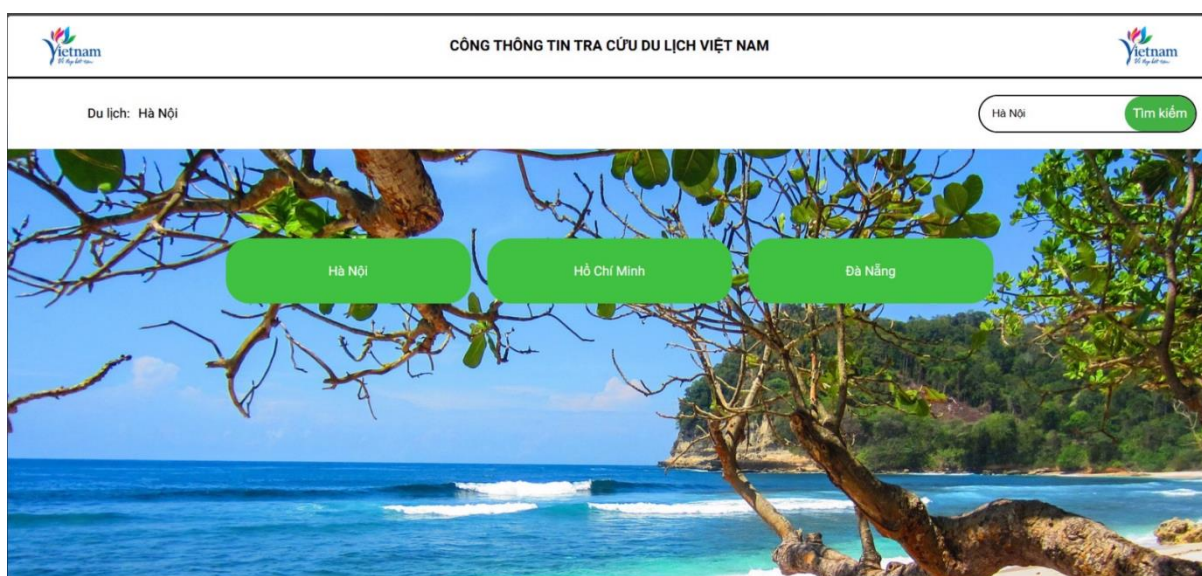


Hình 4.2: Luồng nghiệp vụ chung

4.4 Luồng nghiệp vụ cụ thể và kết quả hiển thị cho các kịch bản

- **Kịch bản 1:** Tìm kiếm một địa điểm du lịch, lưu trú, ăn uống, nơi mua sắm, phương tiện vận chuyển cụ thể theo từ khóa

☐ Tìm kiếm địa điểm du lịch theo từ khóa



Hình 4.3: Tìm kiếm địa điểm du lịch theo từ khóa từ màn hình chính

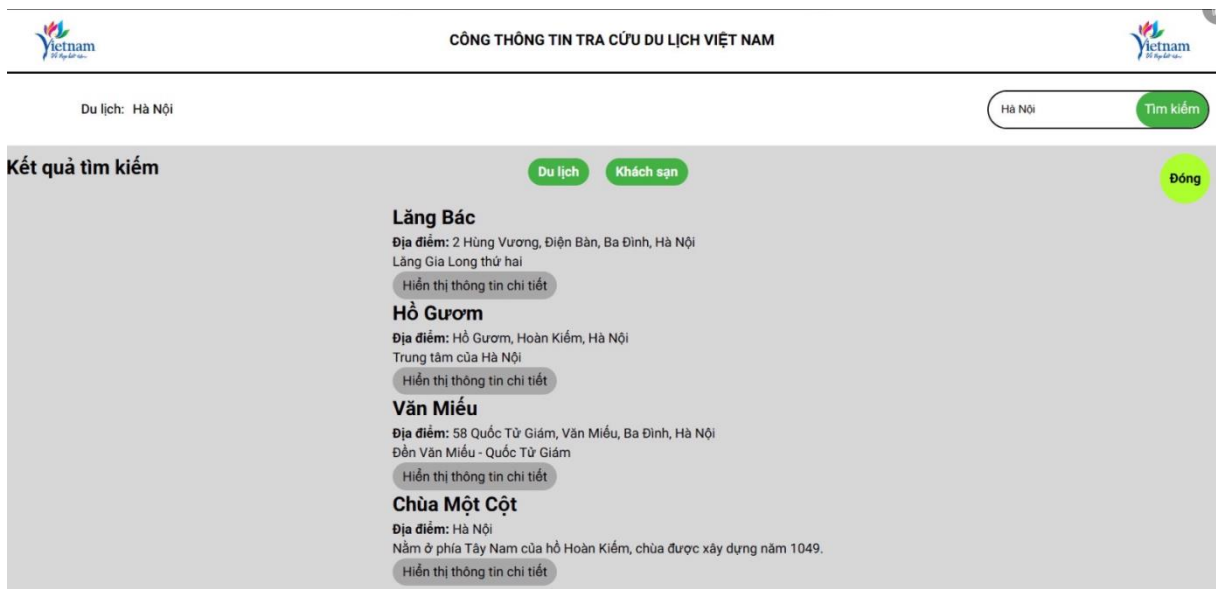
Mô tả kịch bản người dùng:

Ví dụ

1. Người dùng gõ từ khóa vào ô tìm kiếm, ví dụ “Hà Nội”
2. FrontEnd gửi API có keyword “Hà Nội” sang BackEnd
3. BackEnd tạo câu lệnh Sparql để truy vấn Ontology

CÂU LỆNH SPARQL
<pre>PREFIX etourism: <http://www.semanticweb.org/tungl/ontologies/2024/4/dulich> SELECT ?subject ?tengoi ?diachi WHERE { ?subject etourism:ten_goi ?tengoi; etourism:dia_chi ?diachi; FILTER regex(?tengoi, "Ha Noi") }</pre>

4. BackEnd nhận kết quả trả về FrontEnd để hiển thị như sau



5. Người dùng bấm vào 1 lựa chọn để xem chi tiết, ví dụ bấm vào lựa chọn “Lăng Bác”
6. Hệ thống tiếp tục xử lý với chuỗi câu lệnh sau

CÂU LỆNH SPARQL
<p>PREFIX etourism: <http://www.semanticweb.org/tungl/ontologies/2024/4/dulich></p> <p>SELECT ?subject ?diachi</p> <p>WHERE { ?subject</p> <p>etourism:ten_goi "Lăng Bác";</p> <p>etourism:dia_chi ?diachi;</p> <p>}</p>
<p>Tìm kiếm các điểm du lịch quanh khu vực này</p> <p>(Tham số truyền vào chính là trường subject ở câu lệnh trên)</p>
<p>PREFIX etourism: <http://www.semanticweb.org/tungl/ontologies/2024/4/dulich></p> <p>SELECT ?subject ?tengoi ?diachi</p> <p>WHERE { ?subject</p> <p>etourism:o_gan etourism:Lang_Bac;</p> <p>rdf:type etourism:Điểm_đến;</p> <p>etourism:ten_goi ?tengoi;</p> <p>etourism:dia_chi ?diachi;</p> <p>}</p>
<p>Tìm kiếm các cơ sở lưu trú quanh khu vực này</p>
<p>PREFIX etourism: <http://www.semanticweb.org/tungl/ontologies/2024/4/dulich></p> <p>SELECT ?subject ?tengoi ?diachi</p> <p>WHERE { ?subject</p> <p>etourism:o_gan etourism:Lang_Bac;</p>

```

rdf:type etourism:Luu_trú;

etourism:ten_goi ?tengoi;

etourism:dia_chi ?diachi;

}

```

Tìm kiếm các dịch vụ ăn uống quanh khu vực này

```

PREFIX etourism:
<http://www.semanticweb.org/tungl/ontologies/2024/4/dulich>

SELECT ?subject ?tengoi ?diachi

WHERE { ?subject

etourism:o_gan etourism:Lang_Bac;

rdf:type etourism:Ăn_uống;

etourism:ten_goi ?tengoi;

etourism:dia_chi ?diachi;

}

```

Tìm kiếm các điểm mua sắm quanh khu vực này

```

PREFIX etourism:
<http://www.semanticweb.org/tungl/ontologies/2024/4/dulich>

SELECT ?subject ?tengoi ?diachi

WHERE { ?subject

etourism:o_gan etourism:Lang_Bac;

rdf:type etourism:Mua_sắm;

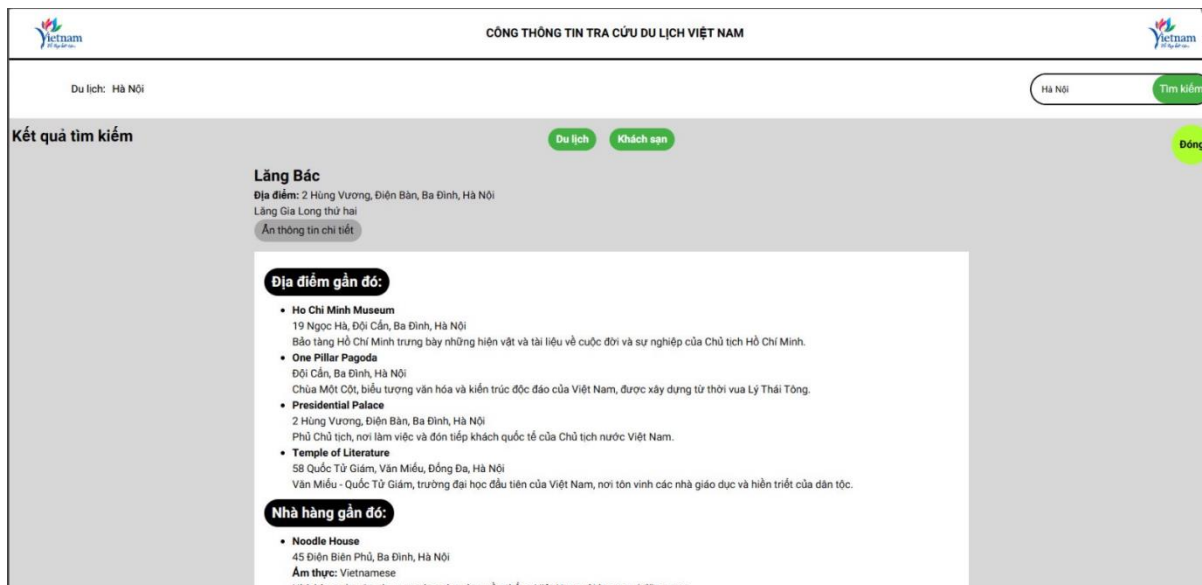
etourism:ten_goi ?tengoi;

etourism:dia_chi ?diachi;

}

```

KẾT QUẢ HIỂN THỊ



4.5 Đánh giá ứng dụng thử nghiệm

Ứng dụng cho phép người dùng tìm kiếm các thông tin du lịch một cách chính xác theo ngữ nghĩa và kèm theo các khuyến nghị rất hữu ích cho các chuyến đi.

- Ứng dụng được xây dựng với mục đích tìm kiếm thông tin tập trung, đã hỗ trợ

tìm kiếm thông tin khác nhau trong hệ sinh thái du lịch từ cơ sở lưu trú, các điểm đến, mua sắm, ăn uống đến phương tiện vận chuyển.

- Ứng dụng đã có khả năng tìm kiếm chính xác theo từ khóa cần tìm kiếm.

- Ứng dụng đưa ra được các khuyến nghị về các cơ sở lưu trú, điểm đến, nơi ăn uống, mua sắm xung quanh các địa điểm tìm kiếm.

- Ứng dụng có khả năng tìm kiếm thông tin du lịch theo các điều kiện lọc đa dạng khác nhau.

KẾT LUẬN

Kết quả đạt được của luận văn

Sau toàn bộ quá trình hoàn thành luận văn, em đã đạt được một số kết quả như sau:

- Nắm vững cơ sở lý thuyết về Ontology, các bước xây dựng một ontology miền.
- Thu thập được dữ liệu về du lịch từ nhiều nguồn khác nhau:
- Xây dựng được Ontology trong miền du lịch từ dữ liệu



Ontology metrics:	
Metrics	
Axiom	326
Logical axiom count	166
Declaration axioms count	143
Class count	37
Object property count	2
Data property count	18
Individual count	86
Annotation Property count	2

Hình 4.1: Kết quả xây dựng Ontology

- Xây dựng được ứng dụng thực nghiệm đáp ứng được các kịch bản và mục tiêu đã thiết kế, đánh giá so sánh với các ứng dụng đang hoạt động.

Các hạn chế

- Giao diện và các chức năng của ứng dụng còn ở mức cơ bản.
- Chưa cập nhật được Ontology trực tiếp từ ứng dụng.

Hướng phát triển

Để luận văn có thể ứng dụng đi vào vào thực tiễn, em đề xuất các hướng phát triển tiếp theo như sau:

- Xây dựng được phần mềm tạo file ontology tự động để thêm các thông tin
- Tích hợp các chức năng khác

TÀI LIỆU THAM KHẢO

- 1) Ontology editors Protege Installation - Dr. Biswanath Dutta
- 2) <https://viblo.asia/p/tim-hieu-ve-semantic-annotation-phan-3-ontology-va-rdf-GrLZDO1wKk0>
- 3) <https://viblo.asia/p/tim-hieu-ve-semantic-annotation-phan-1-semantic-web-web-ngu-nghia-oOVIYpv4Z8W>
- 4) <https://prodima.vn/semantic-web-la-gi>
- 5) Web Semantic and Ontology - Elodie Marie Gontier
- 6) Protege Stanford (2022), <https://protege.stanford.edu/>, Stanford University School of Medicine.
- 7) <https://vietnamtourism.gov.vn/post/37435>