ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA CÔNG NGHỆ PHẦN MỀM

KHOÁ LUẬN TỐT NGHIỆP

XÂY DỰNG VÀ LÀM GIÀU ONTOLOGY TIẾNG VIỆT CHUYÊN NGÀNH CÔNG NGHỆ THÔNG TIN

Giảng viên hướng dẫn:

Th.S HUỲNH NGỌC TÍN

Sinh viên thực hiện:

1. **TRẦN CÔNG DANH** **06520068**
2. **NGUYỄN NGỌC KHÁNH LINH** **06520252**

Lớp : CNPM01

Khoá : 1

TP. Hồ Chí Minh, tháng 3 năm 2011

LỜI MỞ ĐẦU

Ngày nay cùng với sự phát triển của internet thì dữ liệu của ngành công nghệ thông tin ngày càng gia tăng. Nhu cầu quản lý, chia sẻ, tìm kiếm thông tin trong ngành này cũng được đặt ra và đáp ứng một phần nhờ các công cụ tìm kiếm. Một số công cụ tìm kiếm nổi tiếng hiện nay như Google hay Yahoo đều có thể cho phép người dùng tìm kiếm dữ liệu có liên quan bằng cách nhập từ khóa và tìm những tài liệu có chứa từ khóa đó. Với phương pháp tìm như vậy thì kết quả tìm kiếm đôi khi chẳng liên quan gì đến cái mà người dùng muốn tìm, vì các công cụ tìm kiếm này không hiểu được ý nghĩa cần tìm. Việc tìm kiếm thông tin về từ khóa đã vậy thì việc trả lời những câu hỏi càng không thể đối với những công cụ tìm kiếm này.

Muốn cho máy tính và con người có thể hiểu được ngữ nghĩa của từ hay câu thì chúng ta cần có một ontology hỗ trợ bên dưới cho các công cụ này. Ontology giống như một cơ sở dữ liệu về một lĩnh vực cụ thể, nó mô tả mọi thứ trong lĩnh vực đó bao gồm cả định nghĩa những thuật ngữ, những tính chất của những đối tượng và quan hệ giữa chúng. Nó sẽ giúp cho máy tính có thể "hiểu" được ngữ nghĩa giống như con người, chia sẻ thông tin qua các hệ thống khác nhau.

Với nguồn dữ liệu rất lớn trong ngành công nghệ thông tin hiện nay và sự phát triển của các trang web ngữ nghĩa (semantic web) thì việc xây dựng một ontology cho lĩnh vực công nghệ thông tin là một nhu cầu cần thiết. Đặc biệt là đối với ngôn ngữ tiếng Việt, vì vậy chúng em chọn đề tài "Xây dựng và làm giàu ontology tiếng Việt chuyên ngành công nghệ thông tin", báo cáo này được chia thành 5 phần chính gồm:

Chương 1: Tổng quan: Chương này sẽ cho chúng ta thấy tổng quan về đề tài, trong đó có giới thiệu đề tài, giới hạn mục tiêu và phạm vi của đề tài, cho chúng ta biết được cái nhìn tổng quan về phương pháp thực hiện đề tài và kết quả dự kiến thu được.

Chương 2: Cơ sở lý thuyết: Phần này sẽ giải thích rõ về ontology và cho chúng ta thấy tình hình nghiên cứu về ontology hiện nay qua phần khảo sát các nghiên cứu có liên quan.

Chương 3: Xây dựng và làm giàu ontology tiếng Việt chuyên ngành công nghệ thông tin (ITVO): Phần này sẽ nêu chi tiết quá trình xây dựng ontology và đề xuất phương pháp làm giàu.

Chương 4: Hiện thực hệ thống và đánh giá: Phần này sẽ nêu chi tiết quá trình xây dựng công cụ làm giàu ontology, thực nghiệm và đánh giá công cụ.

Chương 5: Kết luận và hướng phát triển: Chương này sẽ tổng kết lại những kết quả đạt được và những hạn chế của đề tài, nêu ra hướng phát triển trong tương lai.

Ngoài ra, phần cuối của báo cáo sẽ nêu các tài liệu tham khảo và phụ lục.

LỜI CẢM ƠN

Đầu tiên, chúng em xin gởi lời cảm ơn đến Thầy, Cô khoa Công nghệ phần mềm trường Đại học Công nghệ thông tin đã tận tình dạy dỗ, dìu dắt chúng em suốt bốn năm đại học.

Chúng em cảm ơn Thầy Huỳnh Ngọc Tín, người đã đưa ra gợi ý về đề tài và tận tình hướng dẫn, giúp đỡ, động viên chúng em hoàn thành luận văn này.

Chúng tôi cảm ơn các bạn Nguyễn Thanh Hoàng và Huỳnh Minh Đức đã giúp đỡ, đóng góp ý kiến cho chúng tôi trong quá trình cài đặt, thử nghiệm chương trình.

Cuối cùng, chúng con cảm ơn Ba, Mẹ và những người thân đã khích lệ, động viên chúng con trong thời gian học tập, nghiên cứu để có được thành quả như ngày nay.

Mặc dù đã cố gắng rất nhiều nhưng chắc chắn chúng em không thể tránh khỏi những sai sót, kính mong nhận được sự đóng góp của quý thầy cô và các bạn.

Tháng 3 năm 2011

Sinh viên

Trần Công Danh - Nguyễn Ngọc Khánh Linh

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Ngày…… tháng……năm 2011

Ký tên

NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Ngày…… tháng……năm 2011

Ký tên

MỤC LỤC

DANH MỤC HÌNH

CHƯƠNG 1: TỔNG QUAN

1.1. Mở đầu

Chương này sẽ cho chúng ta thấy tổng quan về đề tài để trả lời cho vấn đề vì sao cần xây dựng đề tài này, mục tiêu của đề tài là để phục vụ và giải quyết vấn đề gì.

Từ đó chúng em giới hạn lại phạm vi và những yêu cầu cho đề tài. Cuối cùng là phần dự kiến kết quả đạt được sau khi thực hiện đề tài.

1.2. Đặt vấn đề

Ngày nay internet đã và đang là nguồn kiến thức vô tận mang lại nhiều lợi ích cho con người. Sự phát triển mạnh mẽ của nó kéo theo việc những kiến thức trong ngành công nghệ thông tin tăng lên nhanh chóng làm cho việc tra cứu kiến thức cần thiết trở nên khó khăn hơn. Với các công cụ tìm kiếm hiện nay như Google, Yahoo… chỉ giúp người dùng tìm được những tài liệu có chứa từ khóa. Từ đây người dùng phải tốn thời gian và công sức vào từng tài liệu để tìm được đúng thông tin mình cần mà có khi không tìm thấy hoặc tìm thấy thông tin sai lệch. Vấn đề đặt ra là làm sao để có được một công cụ tìm kiếm theo ngữ nghĩa, hiểu được và trả lời câu hỏi của người dùng bằng ngôn ngữ tự nhiên một cách thân thiện. Đặc biệt có thể tìm kiếm bằng tiếng Việt, nhu cầu mà hầu như rất ít công cụ hỗ trợ và kết quả còn hạn chế [1].

Dùng Ontology là một giải pháp biểu diễn tri thức và chia sẻ thông tin mà cả hệ thống và con người có thể hiểu được. Ontology chứa những đặc tả rõ ràng của các khái niệm về một lĩnh vực và quan hệ giữa các khái niệm đó [2]. Nó được dụng trong trí tuệ nhân tạo, công nghệ Web ngữ nghĩa (Semantic Web), các hệ thống kỹ thuật, kỹ thuật phần mềm, sinh tin học và kiến trúc thông tin như là một hình thức biểu diễn tri thức về thế giới hoặc một số lĩnh vực cụ thể [3, 4, 5].

Cùng với những nhu cầu đã nêu ở trên, giáo viên hướng dẫn đã gợi ý và đưa ra đề tài: **“Xây dựng và làm giàu ontology tiếng Việt chuyên ngành Công nghệ thông**

tin". Chúng em nhận thấy đây là một đề tài thú vị và thiết thực nên quyết định chọn nó là đề tài cho khóa luận tốt nghiệp của mình.

Đề tài này nhằm xây dựng một ontology là nền tảng cho những ứng dụng sau này như tìm kiếm thông tin tiếng Việt, hệ thống hỏi đáp tiếng Việt cho ngành công nghệ thông tin, hỗ trợ cho web ngữ nghĩa, giúp xác định thực thể có tên trong tài liệu công nghệ thông tin tiếng Việt. Ontology này có khả năng mở rộng cấu trúc và dữ liệu để phục vụ mục đích hỏi đáp của người dùng. Ngoài ra chúng em cũng sẽ xây dựng công cụ cho phép làm giàu ontology từ internet.

1.3. Mục tiêu và phạm vi đề tài

- **Mục tiêu:** Xây dựng ontology chuyên ngành công nghệ thông tin tiếng Việt phục vụ cho việc nhận diện thực thể có tên, không tên và xác định quan hệ giữa chúng trong tài liệu công nghệ thông tin tiếng Việt, hỗ trợ cho các ứng dụng, nghiên cứu khác về xử lý ngữ nghĩa văn bản tiếng Việt chuyên ngành công nghệ thông tin.
- **Phạm vi đề tài:** Xây dựng ontology tiếng Việt giới hạn trong lĩnh vực Công nghệ thông tin – Information Technology Vietnamese Ontology (ITVO) nhằm lưu trữ:
  - Các khái niệm trong lĩnh vực Công nghệ thông tin và quan hệ giữa chúng.
  - Thông tin các công ty, trường học, tổ chức, hiệp hội, chuyên gia, các sự kiện trong ngành và quan hệ ngữ nghĩa giữa chúng.
  - Các chương trình đào tạo Công nghệ thông tin.

Nguồn dữ liệu: từ ComputingOntology của nhóm nghiên cứu thuộc ACM, trang Wikipedia tiếng Việt, website Bộ thông tin và truyền thông, một số website báo điện tử, các bài báo lĩnh vực công nghệ thông tin tiếng Việt, website các trường có đào tạo ngành công nghệ thông tin trong nước, tài liệu từ internet tìm được từ công cụ tìm kiếm như Google, Yahoo.

1.4. Phương pháp và công cụ

- Xây dựng và nhập dữ liệu bằng tay cho ontology dùng công cụ Protégé.
- Tìm kiếm dữ liệu để làm giàu ontology từ internet sử dụng API của Google và Yahoo
- Dùng thuật toán SVM để phân loại tài liệu công nghệ thông tin tiếng Việt
- Dùng công cụ tách từ tiếng Việt vnTokenizer.
- Rút trích các cá thể từ tài liệu đã phân loại.
- Người dùng kiểm tra, chỉnh sửa và lưu vào ontology dùng API của Protégé.

1.5. Kết quả dự kiến

Kiến thức: Nắm được khái niệm, cấu trúc, mục đích, ứng dụng, cách xây dựng một ontology. Các công cụ hỗ trợ xây dựng ontology hiện nay và sử dụng ngôn ngữ Java để xây dựng công cụ làm giàu ontology (ITVO) bán tự động.

Dữ liệu: Dự kiến nhập bằng tay được khoảng 1000 lớp, 100 quan hệ và 100 cá thể, làm giàu cá thể bán tự động được 1000 cá thể.

1.6. Tổng kết chương

Trong chương này chúng em đã trình bày mục tiêu của việc nghiên cứu và xây dựng ontology hiện nay. Các ứng dụng của nó ngày càng được quan tâm và nó đã trở thành phần "lõi" cho các nghiên cứu ứng dụng liên quan đến ngữ nghĩa, tri thức hơn là những dữ liệu thông thường được lưu trữ trong các hệ quản trị cơ sở dữ liệu. Từ đó nêu ra nguyên nhân chúng em chọn thực hiện đề tài **"Xây dựng và làm giàu ontology tiếng Việt chuyên ngành Công nghệ thông tin"** cho khóa luận tốt nghiệp của mình. Đề tài được giới hạn trong phạm vi và mục tiêu đã nêu trong chương này.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Mở đầu

Ở chương này chúng em sẽ trình bày chi tiết phần lý thuyết về ontology. Cụ thể là gồm các phần như định nghĩa, sự cần thiết của ontology, thành phần, cách xây dựng một ontology và ngôn ngữ để xây dựng nó.

Ngoài ra, chúng em cũng trình bày về một số nghiên cứu có lên quan đến việc xây dựng và làm giàu ontology mà chúng em đã khảo sát. Đây sẽ là tài liệu tham khảo cho việc đề xuất ra phương pháp làm giàu ontology ở chương sau.

2.2. Tổng quan về ontology

2.2.1. Định nghĩa

Trong triết học, từ "ontology" tạm dịch là "bản thể học" được xuất phát từ tiếng Hy Lạp có nghĩa là bộ môn nghiên cứu về sự tồn tại (theo wikipedia). Hiện nay ontology được dùng trong nhiều lĩnh vực như khoa học máy tính, hệ thống kỹ thuật, kỹ thuật phần mềm, tin sinh học, khoa học thư viện, kiến trúc thông tin và các website ngữ nghĩa (Semantic web). Một số định nghĩa về ontology được sử dụng nhiều hiện nay gồm:

- Theo quan điểm triết học, "bản thể học" là ngành khoa học nghiên cứu về bản chất của sự vật, sự tồn tại hoặc những sự vật thực tế, cũng như các loại sự vật cơ bản và các mối quan hệ của chúng (wikipedia).
- Theo Gruber trong tài liệu [20], một ontology là một đặc tả rõ ràng của một sự trừu tượng hóa (An ontology is an explicit specification of a conceptualization).
- Theo John F.Sowa [46], một ontology có thể được đặc tả bởi một danh mục các loại được xác định hoặc không được xác định chỉ bằng những câu phát biểu bằng ngôn ngữ tự nhiên. Một ontology chính thức được xác định bởi một tập

hợp các tên khái niệm và loại quan hệ được tổ chức phân nhóm theo thứ tự cục bộ.

- Một ontology định nghĩa một tập từ vựng cho những nhà nghiên cứu sử dụng khi cần chia sẻ thông tin trong một lĩnh vực. Nó bao gồm những định nghĩa của các khái niệm cơ bản trong một lĩnh vực và mối quan hệ giữa chúng mà máy có thể hiểu được [2].
- Trong khoa học máy tính, một ontology là một mô hình dữ liệu biểu diễn một lĩnh vực và được sử dụng để suy luận về các đối tượng trong lĩnh vực đó và mối quan hệ giữa chúng [36].

Tóm lại, trong khoa học máy tính có thể hiểu ontology gồm những tri thức khái niệm về một lĩnh vực cụ thể và các mối quan hệ giữa chúng. Một ontology về một lĩnh vực sẽ mô tả rõ ràng những thực thể, khái niệm, ràng buộc, quan hệ ngữ nghĩa thuộc lĩnh vực giúp con người và máy có thể hiểu và suy luận được theo ngữ nghĩa trong phạm vi lĩnh vực đó.

2.2.2. Vì sao phải xây dựng ontology?

Ở phần trên, chúng em đã đề cập đến việc ontology đã và đang được sử dụng trong nhiều lĩnh vực, vậy ontology được sử dụng nhiều là vì:

- Để chia sẻ kiến thức chung giữa con người hoặc những tác tử phần mềm với nhau [20]. Nếu các hệ thống cùng chia sẻ chung một ontology bên dưới thì dữ liệu do con người nhập vào tại hệ thống này sau khi được xử lý thông qua ontology có thể được tổng hợp, phân tích tại một hệ thống khác và cung cấp thông tin cho người sử dụng khác.
- Cho phép tái sử dụng kiến thức về một lĩnh vực. Sau khi xây dựng một ontology cho một lĩnh vực, những người khác có thể tái sử dụng và mở rộng, làm giàu

thêm cho nó. Hoặc cũng có thể tích hợp những ontology có sẵn để mô tả nhiều khái niệm thuộc một lĩnh vực nhỏ trong một ontology về một lĩnh vực lớn.

- Làm rõ ràng những giả định thuộc chuyên ngành. Việc sử dụng một ontology ở bên dưới thay vì dùng ngôn ngữ lập trình sẽ giúp dễ dàng thay đổi những giả định thuộc chuyên ngành khi kiến thức về lĩnh vực này của chúng ta thay đổi. Nếu những giả định này được viết bằng ngôn ngữ lập trình thì sẽ gây khó hiểu và khó thay đổi, sửa chữa nhất là đối với những người không phải là chuyên gia lập trình.
- Có thể phân tích và suy luận kiến thức chuyên ngành vì những thuật ngữ, khái niệm cũng như các mối quan hệ giữa chúng đều được khai báo, đặc tả trong ontology với cấu trúc có thể suy luận được theo ngữ nghĩa. Cụ thể là do các khái niệm được lưu dưới cấu trúc cây phân cấp, tên của khái niệm và quan hệ là những từ và cụm từ có nghĩa biểu diễn cho những phát biểu có nghĩa.

2.2.3. Thành phần của ontology [37]

- **Các lớp (Classes) - Khái niệm**

Lớp là nhóm, tập hợp các đối tượng trừu tượng có thể chứa các cá thể, lớp khác hoặc cả hai. Các ontology biến đổi tuỳ thuộc vào cấu trúc và nội dung của nó: Một lớp có thể chứa các lớp con, có thể là một lớp tổng quan (chứa tất cả mọi thứ), có thể là lớp chỉ chứa những cá thể riêng lẻ. Các lớp được sắp xếp theo cấu trúc có thứ bậc, thông thường một ontology có một lớp thông dụng nhất kiểu Thing ở trên đỉnh và các lớp con rất cụ thể ở phía dưới cùng (theo Protégé 4 Tutorial).

Lớp có thể có các ràng buộc (restrictions) cho các quan hệ của cá thể thuộc lớp đó, ví dụ như một Tác giả phải viết một hoặc nhiều tác phẩm thì một

cá thể của tác giả phải có quan hệ "là tác giả của" với một hoặc nhiều cá thể của tác phẩm.

- **Các cá thể (Individuals)**

Là những đối tượng đại diện thuộc một lớp cụ thể trong một lĩnh vực (domain). Mỗi cá thể có thể có các thuộc tính của lớp mà nó thể hiện và quan hệ với các cá thể khác theo ràng buộc của lớp. Những cá thể còn có thể được coi như là những trường hợp của lớp. Trên thực tế một cá thể có thể có nhiều tên vì vậy có thể có trường hợp nhiều cá thể có tên khác nhau nhưng thực chất đều tham chiếu đến một cá thể thực sự. Ví dụ như lớp Quốc gia có 2 cá thể là Hoa Kì và Mỹ nhưng thực tế đây là cùng chỉ một quốc gia nên chúng sẽ cùng tham chiếu đến một cá thể. Nói cách khác, 2 tên đó là chỉ cùng một cá thể và chỉ có 1 cá thể được tạo ra để biểu diễn cho quốc gia đó.

- **Các thuộc tính (Properties)**

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

Giá trị của một thuộc tính có các kiểu thông thường như String, int, float, date… và cũng có thể có các kiểu dữ liệu phức tạp như một cá thể khác chẳng hạn.

- **Các mối quan hệ (Relations)**

Là thuộc tính để mô tả mối liên hệ giữa các đối tượng trong ontology. Một mối quan hệ là một thuộc tính có giá trị là một đối tượng nào đó trong ontology. Một đối tượng có thể có một hoặc nhiều quan hệ trong ontology bất

kể lớp của nó có quan hệ đó hay không, quan hệ của đối tượng phải tuân theo ràng buộc của lớp chứa đối tượng đó nếu có.

Ví dụ như một lớp Tác giả có quan hệ "nơi công tác hiện tại" với lớp Tổ chức. Quan hệ này có ràng buộc là một tác giả chỉ có một nơi công tác hiện tại, tức là một cá thể Tác giả chỉ có quan hệ với một cá thể của Tổ chức.

2.2.4. Làm thế nào để xây dựng một ontology?

a. Phương pháp xây dựng một ontology

Hiện nay không có phương pháp chuẩn nào cho việc xây dựng một ontology [2]. Khi xây dựng ontology chúng ta nên dựa vào nhu cầu của ứng dụng sẽ sử dụng nó để thiết kế cho phù hợp.

Quá trình xây dựng một ontology là một quá trình lặp, thường bắt đầu bằng một phiên bản thô rồi sao đó xem xét, chỉnh sửa, lọc lại ontology phiên bản trước và thêm vào các chi tiết.

Những khái niệm trong ontology là những đối tượng thực tế hoặc logic phản ánh thế giới thực và những quan hệ trong ontology thường là những động từ trong câu mô tả khái niệm trong lĩnh vực.

Theo tài liệu [2] thì phương pháp xây dựng ontology gồm các bước:

Bước 1: ***Xác định miền và phạm vi của ontology.*** Đây là bước chúng ta nên làm trước khi muốn xây dựng một ontology. Trong một hệ thống có sử dụng ontology thì các yêu cầu đối với nó thường là mô tả một lĩnh vực nào đó nhằm cung cấp cơ sở tri thức trong việc giải quyết những mục đích chuyên biệt. Để nhận diện chính xác những yêu cầu chúng ta cần phải trả lời một số câu hỏi như:

- Ontology cần mô tả lĩnh vực nào?
- Ontology phục vụ cho mục đích chuyên biệt gì?

- Cơ sở tri thức trong ontology sẽ trả lời những câu hỏi gì?
- Ontology nhằm vục vụ đối tượng nào?
- Ai là người sẽ xây dựng, quản trị ontology?

Các câu trả lời có thể thay đổi ở mỗi bước lặp trong quá trình xây dựng ontology tùy mục đích của ứng dụng hoặc có những tính năng cần bổ sung lúc đó. Trả lời các câu hỏi trên sẽ giúp giới hạn phạm vi thực sự của ontology cần mô tả và dự trù các kỹ thuật sẽ sử dụng trong quá trình phát triển. Ví dụ như ontology cần xây dựng có chức năng xử lý ngôn ngữ tự nhiên, ứng dụng dịch tài liệu tự động thì cần phải có kỹ thuật xác định từ đồng nghĩa.

Sau khi đã phát thảo phạm vi ontology dựa trên việc trả lời những câu hỏi trên, chúng ta tiếp tục tinh chỉnh lại bằng cách trả lời các câu hỏi kiểm chứng khả năng (competency question):

- Ontology đã có đủ thông tin để trả lời cho các câu hỏi được quan tâm trên cơ sở tri thức hay không?
- Câu trả lời của hệ thống dựa trên cơ sở tri thức đã đáp ứng được mức độ, yêu cầu nào của người sử dụng?
- Các ràng buộc và quan hệ phức tạp trong miền quan tâm đã được biểu diễn hợp lý chưa?

Bước 2: ***Xem xét việc kế thừa các ontology có sẵn:*** đây là một công đoạn thường hay sử dụng để giảm thiểu công sức xây dựng một ontology. Bằng cách kế thừa các ontology tương tự có sẵn, người xây dựng có thể thêm hoặc bớt các lớp, quan hệ giữa các lớp, thực thể… để tinh chỉnh tùy theo mục đích của mình. Ngoài ra, việc sử dụng lại các ontology có sẵn cũng rất quan trọng khi cần sự tương tác giữa các ứng dụng khác nhau vì các ứng dụng sẽ cần phải hiểu các lớp, thực thể, quan hệ… của nhau để thuận tiện trong việc trao đổi hoặc thống nhất thông tin.

Bước 3: ***Liệt kê các thuật ngữ quan trọng trong ontology:*** Liệt kê tất cả các thuật ngữ xuất hiện trong miền quan tâm (có thể đồng nghĩa hoặc chồng nhau) như tên khái niệm, quan hệ, thuộc tính… Thông thường, các thuật ngữ là danh từ sẽ trở thành các lớp, tính từ sẽ trở thành thuộc tính, còn động từ sẽ là quan hệ giữa các lớp.

Bước 4: ***Xây dựng các lớp và cấu trúc lớp phân cấp:*** Định nghĩa các lớp từ một số thuật ngữ đã liệt kê trong bước 3, sau đó xây dựng cấu trúc lớp phân cấp theo quan hệ *lớp cha-lớp con*. Lớp ở vị trí càng cao trong cấu trúc này sẽ có mức độ tổng quát càng cao. Vị trí đầu tiên thuộc về ***lớp gốc***, tiếp theo là các ***lớp trung gian***, và cuối cùng là ***lớp lá***. Lớp lá là lớp không thể triển khai được nữa và chỉ được biểu hiện bằng các thực thể.

Hình 1: Cấu trúc lớp phân cấp

Thực thể của lớp con *"là-một"* thực thể của lớp cha nó.

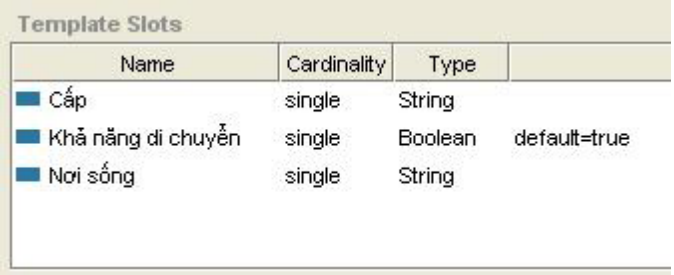
Có nhiều hướng tiếp cận khác nhau cho vấn đề xây dựng cấu trúc lớp phân cấp như:

- ✓ Hướng xây dựng ***từ trên xuống*** *(top-down)*: bắt đầu bằng các lớp có mức độ tổng quát cao nhất, sau đó triển khai dần đến lớp lá.

- ✓ Hướng xây dựng ***từ dưới lên*** *(bottom-up)*: Ngược với hướng xây dựng cấu trúc lớp phân cấp từ trên xuống, hướng này bắt đầu bằng việc xác định các lớp được cho là cụ thể nhất, sau đó tổng quát hóa đến khi được lớp gốc.
- ✓ Cách ***kết hợp*** *(combination)*: cách này kết hợp cả hai hướng xây dựng trên. Đầu tiên chọn các lớp nổi bật nhất trong miền quan tâm, sau đó tổng quát hóa và cụ thể hóa cho đến khi được cấu trúc mong muốn.

Bước 5: ***Định nghĩa các thuộc tính và quan hệ cho lớp:*** các lớp tạo ra ở bước 4 chỉ mới là những tên gọi, tiếp theo chúng ta cần định nghĩa thuộc tính của lớp là các thông tin bên trong của lớp, mô tả một khía cạnh nào đó của lớp và được dùng để phân biệt với các lớp khác. Có hai loại: ***thuộc tính đơn*** *(simple property)* và ***thuộc tính phức*** *(complex property)*. Thuộc tính đơn là các giá trị đơn ví dụ: chuỗi, số,… còn thuộc tính phức có thể chứa hoặc tham khảo đến một đối tượng khác. Một lớp sẽ kế thừa toàn bộ các thuộc tính của tất cả các lớp cha của nó.

Bước 6: ***Định nghĩa các ràng buộc về thuộc tính và quan hệ của lớp:*** Các ràng buộc (restrictions) giới hạn giá trị mà một thuộc tính có thể nhận. Hai ràng buộc quan trọng nhất đối với một thuộc tính là ***lượng số*** *(cardinality)* và ***kiểu*** *(type)*. Ràng buộc lượng số quy định số giá trị mà một thuộc tính có thể nhận. Hai giá trị thường thấy của ràng buộc này là đơn trị *(single)* và đa trị *(multiple)*. Ràng buộc thứ hai là về kiểu, các kiểu mà một thuộc tính có thể nhận là: chuỗi, số, luận lý (Boolean), liệt kê và kiểu thực thể. Riêng kiểu thực thể có liên quan đến hai khái niệm gọi là: ***miền*** *(domain)* và ***khoảng*** *(range)*. Khái niệm miền được dùng để chỉ lớp (hay các lớp) mà một thuộc tính thuộc về. Ví dụ như thuộc tính Tên là thuộc tính của lớp Tác giả, Trường, Tổ chức nên miền của nó là 3 lớp này. Trong khi đó, khoảng chính là lớp (hay các lớp) làm kiểu cho giá trị thuộc tính kiểu thực thể. Ví dụ thuộc tính Nơi sinh của lớp Tác giả có thể có giá trị là một cá thể (kiểu thực thể) của một lớp Quốc gia như Mỹ.



Hình 2: Ràng buộc về thuộc tính.

Bước 7: Đây là bước cuối cùng khép lại một vòng lặp xây dựng ontology. Việc chúng ta cần làm ở bước này là tạo thực thể cho mỗi lớp và gán giá trị cho các thuộc tính.

b. Ngôn ngữ để xây dựng ontology:

RDF: là mô hình dữ liệu mô tả các đối tượng và các mối quan hệ giữa chúng. Mô hình dữ liệu này dùng cú pháp của XML chỉ giúp cho thông tin được thể hiện ở dạng bộ ba theo đúng mô hình RDF chứ thông tin vẫn chưa thể hiện gì về mặt ngữ nghĩa.

Ví dụ sau minh họa cho việc dùng RDF chỉ để biểu diễn dữ liệu [40]:

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

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:si="http://www.w3schools.com/rdf/">
<rdf:Description rdf:about="http://www.w3schools.com">

      <si:title>W3Schools.com</si:title>

      <si:author>Jan Egil Refsnes</si:author>

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

RDF Schema: là một ngôn ngữ ontology cơ bản mô tả các thuộc tính (property) và các lớp (class) của đối tượng RDF. Nó được phát triển ở tầng trên của RDF cho nên

bản thân RDF-Schema cũng chính là RDF. Nó được mở rộng từ RDF và bổ sung thêm các tập từ vựng để hỗ trợ cho việc xây dựng các ontology được dễ dàng để hình thành nên ngữ nghĩa cho thông tin, là cơ sở để xây dựng các công cụ tìm kiếm ngữ nghĩa.

Ví dụ sau cho thấy RDF Schema có thể biểu diễn được các lớp, thuộc tính của đối tượng RDF [41]:

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

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xml:base="http://www.animals.fake/animals#">
<rdfs:Class rdf:ID="animal" />

<rdfs:Class rdf:ID="horse">

      <rdfs:subClassOf rdf:resource="#animal"/>

</rdfs:Class>
</rdf:RDF>
```

OWL: OWL là ngôn ngữ ontology khá mạnh, nó ra đời sau RDFS nên biết kế thừa những lợi thế của ngôn ngữ này đồng thời bổ sung thêm nhiều yếu tố giúp khắc phục được những hạn chế của RDFS.

Sau đây là một ví dụ dùng OWL để biểu diễn ontology:

```
<owl:Class rdf:ID="WineDescriptor" />

<owl:Class rdf:ID="WineColor">
  <rdfs:subClassOf rdf:resource="#WineDescriptor" />
  ...
</owl:Class>
```

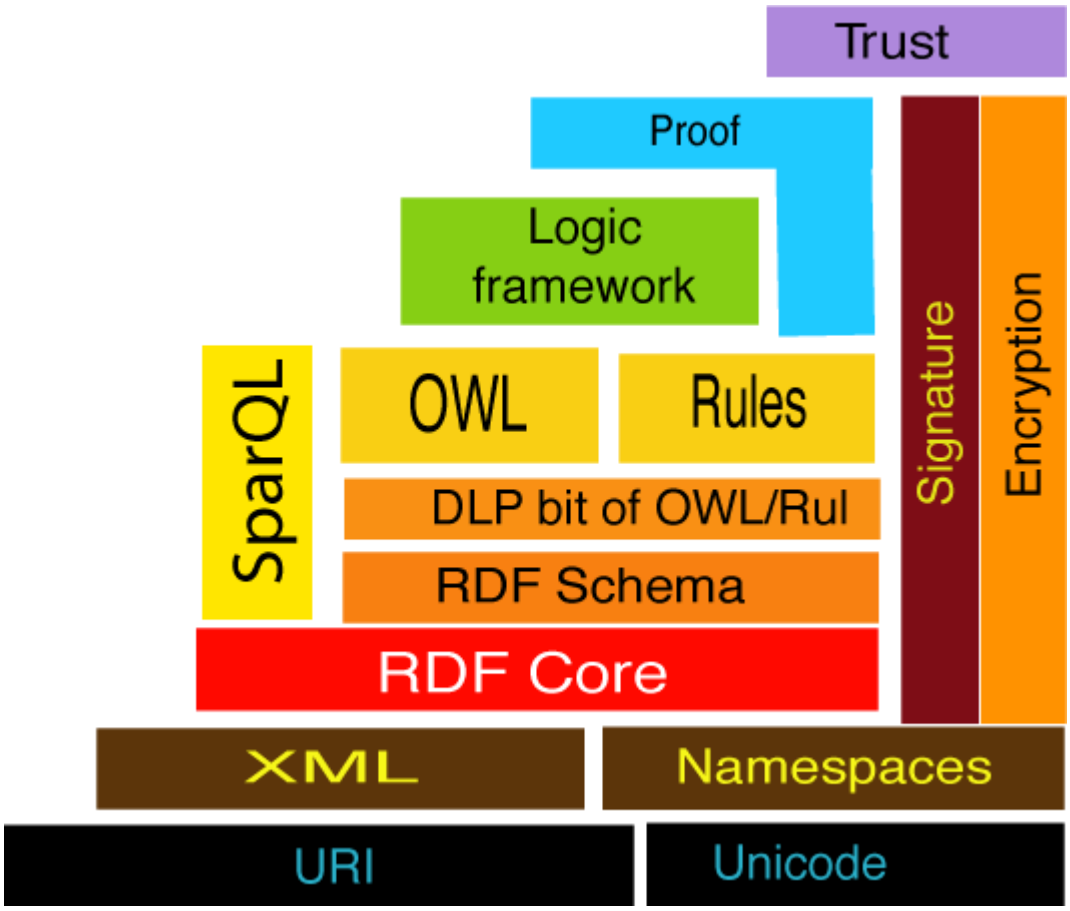
Đoạn phía trên là ví dụ mô tả lớp và các lớp con của nó trong ontology.

```
<owl:ObjectProperty rdf:ID="hasWineDescriptor">
  <rdfs:domain rdf:resource="#Wine" />
  <rdfs:range  rdf:resource="#WineDescriptor" />
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="hasColor">
  <rdfs:subPropertyOf rdf:resource="#hasWineDescriptor" />
  <rdfs:range rdf:resource="#WineColor" />
```

```
  ...
</owl:ObjectProperty>
```

Đoạn này để mô tả các quan hệ trong ontology.



Hình 3: Hình minh họa các tầng ngôn ngữ dùng trong ontology

Nguồn: http://groupme.org/GroupMe/resource/123

Nhìn vào hình trên chúng ta có thể thấy được 3 ngôn ngữ ontology trên đều sử dụng thư viện, cú pháp xuất phát từ XML và RDF là ngôn ngữ ở mức thấp nhất để mô tả một ontology. Trên nó là RDF Schema, là ngôn ngữ đã được bổ sung thêm một số thư viện để phù hợp với việc mô tả ontology. Và cuối cùng là OWL, ngôn ngữ mới nhất, và đầy đủ nhất để mô tả một ontology. DLP là viết tắt của Description Logic Programs là ngôn ngữ cục bộ để tích hợp những cơ sở tri thức được mô tả bằng

Description Logic (DL) và Logic Programs (LP), nó được định nghĩa là một tập giao giữa việc biểu diễn OWL bằng DL và LP [38].

❖ **OWL (Ontology Web Language)**

OWL là ngôn ngữ được phát triển mới nhất trong các ngôn ngữ ontology chuẩn được công nhận bởi World Wide Web Consortium (W3C) để thúc đẩy sự phát triển của các web ngữ nghĩa (Semantec Web).

OWL kế thừa từ DAML+OIL được phát triển bởi tổ chức W3C. Tên DAML+OIL là sự kết hợp giữa tên DAML-ONT (http://www.daml.org/2000/10/daml-ont.html) do Mỹ đề xuất và ngôn ngữ OIL (http://www.ontoknowledge.org/oil/) do Châu Âu đề xuất.

OWL giúp tăng thêm yếu tố logic cho thông tin và khả năng phân loại, ràng buộc kiểu cũng như lượng số tương đối mạnh. Là ngôn ngữ mô tả từ vựng phong phú để mô tả các thuộc tính và các lớp, các mối quan hệ giữa các lớp (như disjointness), số của giá trị (cardinality), tính tương đương (equality), định kiểu thuộc tính, đặc tính của thuộc tính (đối xứng). Một ví dụ về ràng buộc kiểu và số lượng dùng OWL như sau:

```
<owl:Class rdf:ID="Vintage">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasVintageYear"/>
      <owl:cardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Ở ví dụ trên ta thấy lớp Vintage có quan hệ hasVintageYear và quan hệ này bị ràng buộc không được là số nguyên âm và chỉ cho phép có 1 giá trị.

Một số cú pháp để khai báo các thành phần chính trong ontology dùng ngôn ngữ OWL như sau, các ví dụ tham khảo từ nguồn [39]:

– Đầu tiên, chúng ta cần phải khai báo các namespace để có thể sử dụng các thư viện cần thiết:

```
<rdf:RDF
    xmlns:owl ="http://www.w3.org/2002/07/owl#"
    xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:xsd ="http://www.w3.org/2001/XMLSchema#">
```

– Để khai báo một lớp dùng thẻ *<owl:Class>*, khai báo lớp hiện tại là lớp con dùng thẻ *<rdfs:subClassOf>*, khai báo một cấu trúc cây phân cấp (taxonomic tree) như sau:

```
<owl:Class rdf:ID="PotableLiquid">
<rdfs:subClassOf rdf:resource="#ConsumableThing" />
 ...
</owl:Class>
```

– Ví dụ khai báo một cá thể:

```
<Region rdf:ID="CentralCoastRegion" />
```

Khai báo cá thể tên CentralCoastRegion là một cá thể của lớp Region

– Khai báo thuộc tính gồm những thẻ chính sau: <owl:ObjectProperty> dùng để khai báo các quan hệ (là thuộc tính có kiểu giá trị là một lớp), <owl:DatatypeProperty> để khai báo thuộc tính có kiểu giá trị thông thường, <rdfs:subPropertyOf> dùng để khai báo một thuộc tính là thuộc tính con,<rdfs:domain> và <rdfs:range> dùng để khai báo domain và range cho thuộc tính.

Ví dụ cú pháp của DatatypeProperty trong đó thuộc tính tên là yearValue là thuộc tính của lớp VintageYear và có giá trị là số nguyên dương:

```
<owl:Class rdf:ID="VintageYear" />

<owl:DatatypeProperty rdf:ID="yearValue">
  <rdfs:domain rdf:resource="#VintageYear" />
  <rdfs:range  rdf:resource="&xsd;positiveInteger"/>
</owl:DatatypeProperty>
```

Ví dụ cú pháp của ObjectProperty trong đó lớp Wine có quan hệ hasWineDescriptor với lớp WineDescriptor và quan hệ hasWineDescriptor có quan hệ con là hasColor với WineColor.

```
<owl:Class rdf:ID="WineDescriptor" />

<owl:Class rdf:ID="WineColor">
  <rdfs:subClassOf rdf:resource="#WineDescriptor" />
  ...
</owl:Class>

<owl:ObjectProperty rdf:ID="hasWineDescriptor">
  <rdfs:domain rdf:resource="#Wine" />
  <rdfs:range  rdf:resource="#WineDescriptor" />
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="hasColor">
  <rdfs:subPropertyOf rdf:resource="#hasWineDescriptor" />
  <rdfs:range rdf:resource="#WineColor" />
  ...
</owl:ObjectProperty>
```

Một số các thẻ khác dùng để mô tả và khai báo các ràng buộc có thể tham khảo thêm từ website [39].

Hiện nay có ba loại OWL : OWL Lite, OWL DL (description logic), và OWL Full.

- OWL Lite: hỗ trợ cho những người dùng chủ yếu cần sự phân lớp theo thứ bậc và các ràng buộc đơn giản. Ví dụ: Trong khi nó hỗ trợ các ràng buộc về tập hợp, nó chỉ cho phép tập hợp giá trị của 0 hay 1. Điều này giúp OWL Lite dễ sử dụng và thực thi và việc cung cấp các công cụ hỗ trợ OWL Lite dễ dàng hơn so với các bản khác nhưng lại hạn chế trong việc diễn đạt.
- OWL DL (OWL Description Logic): hỗ trợ cho những người dùng cần cung cấp sự diễn đạt tối ưu và đảm bảo tất cả các kết luận là có thể dự tính được và sẽ hoàn thành trong một thời gian nhất định. OWL DL bao gồm tất cả các cấu trúc của ngôn ngữ OWL, nhưng chúng chỉ có thể được sử dụng với những hạn chế nào đó (Ví dụ: Trong khi một lớp có thể là một lớp con của rất nhiều lớp, một lớp không thể là một thể hiện của một lớp khác).

OWL mất toàn bộ tính tương thích với RDF. Thông thường, một tài liệu RDF phải được mở rộng theo một số cách và bị giới hạn theo các cách khác trước khi nó là một tài liệu OWL DL hợp lệ. Mọi tài liệu OWL DL hợp lệ là tài liệu RDF hợp lệ.

- OWL Full: sử dụng tất cả các từ vựng nền tảng (primitive) của ngôn ngữ OWL. Nó cho phép kết hợp tùy ý các từ vựng nền tảng với RDF và RDF Schema vì vậy nó tạo ra sự diễn đạt tối đa và tự do. Ví dụ, trong OWL Full, một lớp có thể được xem xét đồng thời như là một tập của các cá thể và như là một cá thể trong chính bản thân nó. OWL Full cho phép một ontology gia cố thêm ý nghĩa của các từ vựng được định nghĩa trước (RDF hoặc OWL) và hoàn toàn tương thích với RDF. Ngôn ngữ này trở nên quá mạnh mẽ đến mức là không thể quyết định được (undecidable), ảnh hưởng đến hỗ trợ lập luận đầy đủ hoặc hỗ trợ lập luận hiệu quả.

Các phiên bản này tách biệt về các tiện ích khác nhau, OWL Lite là phiên bản dễ hiểu nhất và phức tạp nhất là OWL Full. Việc lựa chọn ngôn ngữ con nào phù hợp nhất là phụ thuộc vào nhu cầu của mỗi người.

Mối liên hệ giữa các ngôn ngữ con của OWL:

- Mọi ontology hợp lệ dựa trên OWL Lite đều là ontology hợp lệ trên OWL DL.
- Mọi ontology hợp lệ dựa trên OWL DL đều là ontology hợp lệ trên OWL Full.
- Mọi kết luận hợp lệ dựa trên OWL Lite đều là kết luận hợp lệ trên OWL DL.
- Mọi kết luận hợp lệ dựa trên OWL DL đều là kết luận hợp lệ trên OWL Full

2.3. Khảo sát các nghiên cứu có liên quan

2.3.1. Các nghiên cứu trên thế giới

a. TheComputingOntology [11]

Được công bố năm 2005, ban đầu có tên là Ontology Project được xây dựng bởi một nhóm thuộc tổ chức ACM [10] nhằm biểu diễn kiến thức về máy tính và thông tin

có quan hệ chặt chẽ theo quy tắc phục vụ cho việc nghiên cứu hay giảng dạy trong lĩnh vực liên quan tới tính toán, quản lý và xử lý thông tin. Công việc được hỗ trợ bởi tổ chức khoa học quốc gia Mỹ (*National Science Foundation*), ACM (*Association for Computing Machinery*), IEEE và Đại học Mở của Hà Lan (*Open University of the Netherlands*).

Nguồn dữ liệu gồm: Tài liệu chương trình đào tạo của các trường đại học như: Lewis University, Villanova University… và những thuật ngữ quan trọng trong các môn học liên quan đến máy tính từ hệ thống phân lớp trên ACM (ACM Computing Classification System [13]). Được xây dựng dùng công cụ soạn thảo ontology là Protégé, đến nay đã có 6 phiên bản của ComputingOntology trên trang web chính thức của nó.

Nhận xét: Xây dựng được một ontology về tính toán và thông tin hỗ trợ phát triển hệ thống tư vấn về chương trình đào tạo, phát triển những chương trình học mới, kiểm tra những chương trình học đã có, làm rõ ràng các mối quan hệ giữa những môn học với nhau, phát triển những chương trình gồm nhiều ngành học, đóng góp cho việc phân lớp trong nghiên cứu. Tuy nhiên, dữ liệu của ontology là tiếng Anh không thể phục vụ cho các nghiên cứu chuyên ngành công nghệ thông tin tiếng Việt.

b. SwetoDBLP [12]

Được xây dựng bởi nhóm tác giả từ khoa Khoa học máy tính của trường Đại học Georgia, Mỹ. SwetoDblp [34] là một ontology có kích thước lớn tập trung vào dữ liệu thông tin của các bài báo về khoa học máy tính như: Tên, tác giả, nhà xuất bản… Dữ liệu chính của nó lấy từ cơ sở dữ liệu DBLP [16] (Digital Bibliography & Library Project) là cơ sở dữ liệu chỉ mục các bài báo khoa học trong lĩnh vực khoa học máy tính. Tính đến tháng 1/2011 DBLP chứa thông tin của 1, 5 triệu bài báo được đánh dấu chỉ mục thông qua việc phân tích danh sách các file đề mục (tables of contents– TOCs) của các hội nghị cũng như các tạp chí… Ngoài ra, còn có 3 nguồn dữ liệu khác được dùng để tạo

SwetoDblp là danh sách các trường đại học lấy từ Google có đường dẫn nguồn là www.google.com/intl/en/universities.html được chỉnh sửa bằng tay lại cho phù hợp, danh sách các website của nhà xuất bản và danh sách các hội thảo được tạo bằng tay theo dữ liệu trong DBLP.

Dữ liệu của SwetoDblp được lưu trữ dùng định dạng RDF, sử dụng bộ từ vựng lược đồ (schema-vocabulary) có sẵn như FOAF [17] và Dublin Core [18]. Việc tạo ra và cập nhật ontology được thực hiện dùng công cụ SAX-parser để chuyển dữ liệu dạng XML của DBLP sang RDF. Dữ liệu sẽ được cập nhật hàng tháng theo dữ liệu XML mới nhất từ DBLP và danh sách các trường đại học, nhà xuất bản và hội thảo.

SwetoDblp hiện đang được sử dụng để kiểm tra cho OptARQ, một cơ chế cho phép tối ưu hóa câu truy vấn vào ontology dùng SPARQL [19]. Ngoài ra, ontology này còn được dùng để tìm kiếm các bài báo và chuyên gia, phân biệt, tránh sự nhập nhằng giữa tên các nhà nghiên cứu trong danh sách mail của DBWorld [33].

Nhận xét: Ontology này như một thư viện điện tử với lượng thông tin lớn về các bài báo, không phục vụ cho việc tìm kiếm các khái niệm và thông tin trong ngành công nghệ thông tin.

2.3.2. Các nghiên cứu trong nước

a. Ontology for Vietnamese Language (OVL) – Open version [7]

Là một ontology tổng quát (Universal Ontology) được thực hiện bởi Nguyễn Tuấn Đăng, Võ Hoài An, Nguyễn Trí Phúc trường Đại học Công nghệ Thông tin. Xây dựng trên phiên bản Protégé 3.4.3. Mục tiêu tác giả xây dựng ontology này là để đóng góp cho những nghiên cứu về xử lý ngôn ngữ tiếng Việt, xây dựng tri thức phổ quát trong nhiều lĩnh vực bằng tiếng Việt.

Dữ liệu của ontology là dữ liệu tổng quát về các lĩnh vực gồm 10 lĩnh vực chính theo các mục được lấy theo VNExpress như: Khoa học, Pháp luật, Chính trị, Kinh

doanh, Thể thao, Văn hóa du lịch, Xã hội, Vi tính, Viễn thông, Ô tô xe máy. Ngoài ra còn lấy dữ liệu từ các nguồn như Wikipedia tiếng Việt, Yellow Page và nhiều website khác nhau liên quan đến các lĩnh vực trên [6].

Nhận xét: Kết quả tạo ra được ontology gồm số lượng lớp là 2.543, số lượng cá thể là 10.024, với 312 ràng buộc và 87 thuộc tính thuộc nhiều lĩnh vực. Tuy nhiên, dữ liệu của ontology mang tính phổ quát, không tập trung vào một lĩnh vực (domain) cụ thể. Ví dụ như trong ngành Công nghệ thông tin không có chứa thông tin về những khái niệm, chuyên gia hay chương trình đào tạo của ngành.

b. Ontology khoa học công nghệ [4]

Được thực hiện bởi Bộ môn Hệ thống thông tin của trường Đại học Bách khoa Hà Nội. Hệ thống hỗ trợ tìm kiếm dựa trên từ khóa, cấu trúc dữ liệu lưu trữ, tìm kiếm mở rộng dựa trên ngữ nghĩa và tri thức phục vụ cho việc quản lý tài liệu và thông tin trong lĩnh vực khoa học công nghệ. Nhằm giải quyết cho những yêu cầu đó tác giả đã đề xuất phương pháp xây dựng một ontology chuyên ngành khoa học công nghệ để khai thác các suy diễn ngữ nghĩa.

Những khái niệm được xây dựng dựa trên việc khảo sát nhu cầu quản lý thông tin tại phòng KHCN thuộc Đại học Bách Khoa Hà Nội, phòng KHCN thuộc sở Khoa học Công nghệ Thành Phố Hà Nội, sở Bưu chính Viễn thông. Người bảo trì có thể là tác giả hoặc những người có quan tâm và có kiến thức về ontology sẽ nâng cấp cập nhật thông tin khi có thay đổi.

Với việc sử dụng ontology này hệ thống ngoài việc dùng để tra cứu các đề tài, sản phẩm công nghệ, chuyên gia, tài liệu, giải pháp, công nghệ… thì còn có thể trả lời được những câu hỏi tổng hợp phân tích như: Có những đề tài nào thuộc lĩnh vực mà người dùng quan tâm? Đề tài nào dành được sự quan tâm nhiều nhất cũng như nhận định về giá trị, khả năng ứng dụng vào thực tiễn? Tài liệu đang được xem xét có những phiên bản nào, sự đánh giá của các độc giả đối với các phiên bản của tài liệu này như

thế nào? Tìm những chuyên gia đa lĩnh vực như chuyên gia vừa trong lĩnh vực CNTT vừa trong lĩnh vực Hoá sinh.

Ontology này được xây dựng dùng phần mềm soạn thảo cơ sở tri thức được viết dựa trên các API của Protégé. Cơ sở dữ liệu này chứa dữ liệu về khoảng 3000 chuyên gia, 1500 đề tài cùng với hơn 150 lĩnh vực KHCN.

Nhận xét: Không rút trích được khái niệm hay cá thể từ nội dung tài liệu hay bài báo khoa học. Dữ liệu của ontology KHCN không bao trùm hết lĩnh vực công nghệ thông tin.

2.4. Tổng kết chương

Trong chương này chúng em đã trình bày một số khảo sát các nghiên cứu về ontology trong và ngoài nước. Từ đó rút ra các nhận xét về tình hình nghiên cứu và ứng dụng ontology hiện nay.

Theo xu hướng đó, việc xây dựng các ontology tiếng Việt đã bắt đầu nhận được sự quan tâm của các nhà nghiên cứu khoa học ở Việt Nam như phần khảo sát đã đề cập. Đặc biệt đối với ngành công nghệ thông tin thì lượng dữ liệu ngày càng lớn và cần được quản lý một cách có hệ thống và có ngữ nghĩa.

Ngoài ra, trong chương này đã nêu tổng quan về ontology gồm có định nghĩa, các thành phần trong ontology, ngôn ngữ xây dựng ontology và phương pháp để xây dựng một ontology.

CHƯƠNG 3: XÂY DỰNG VÀ LÀM GIÀU ONTOLOGY TIẾNG VIỆT CHUYÊN NGÀNH CÔNG NGHỆ THÔNG TIN (ITVO)

Trong chương này chúng em xin trình bày về công cụ và quá trình xây dựng ontology tiếng Việt chuyên ngành công nghệ thông tin. Ngoài ra, chương này chúng em cũng sẽ trình bày đề xuất phương pháp để xây dựng công cụ làm giàu ontology này.

3.1. Xây dựng ontology tiếng việt chuyên ngành công nghệ thông tin (ITVO)

3.1.1. Công cụ sử dụng

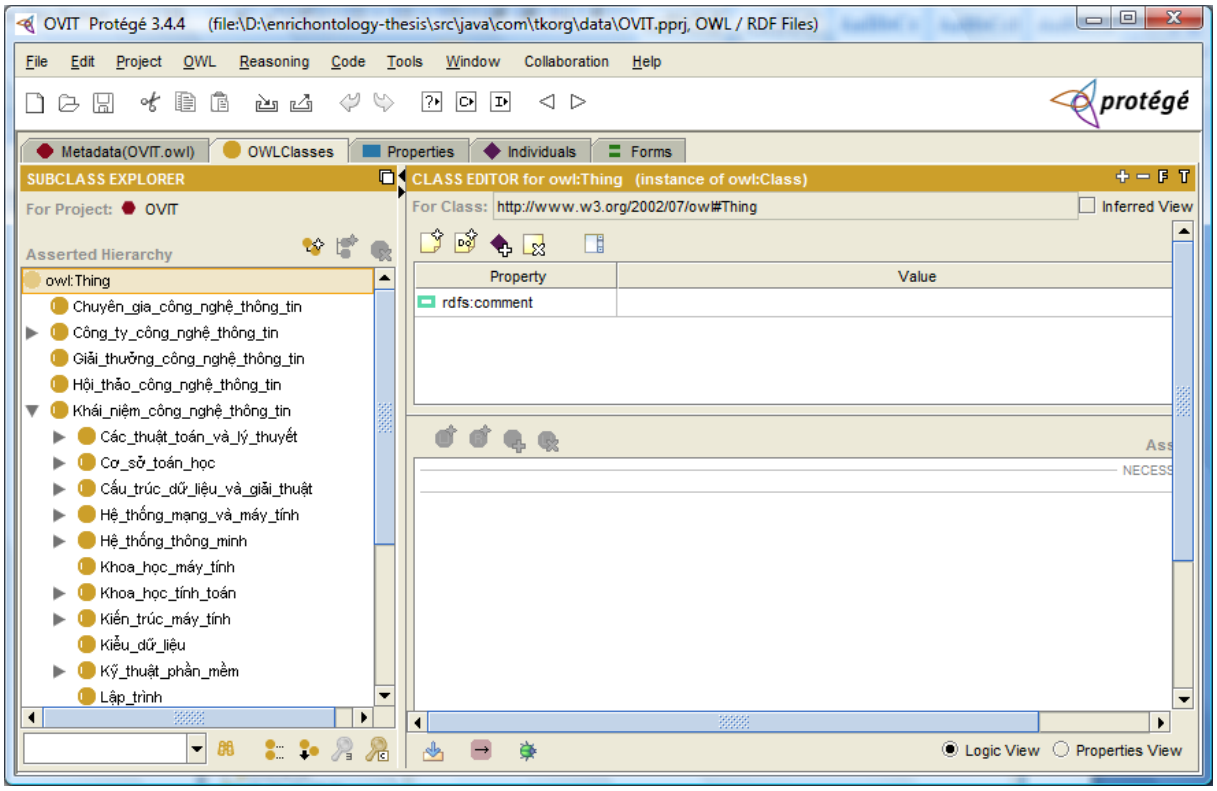
Ontology tiếng Việt chuyên ngành công nghệ thông tin (ITVO) được xây dựng dùng công cụ soạn thảo Protégé phiên bản 3.4.4 [21]. Đây là bộ phần mềm mã nguồn mở Java được nghiên cứu và phát triển từ năm 1998 bởi nhóm nghiên cứu của Mark Musen thuộc đại học Stanford, California nhằm quản lý các thông tin trong lĩnh vực sinh y học. Đây là dự án được nhận được sự quan tâm và tài trợ từ rất nhiều tổ chức, trong đó có Bộ Quốc Phòng Mỹ. Hiện nay, nó có một cộng đồng hàng nghìn người sử dụng và đã có rất nhiều miền ứng dụng khác nhau sử dụng sự hỗ trợ của công cụ này. Mã nguồn Protégé có thể được tìm thấy tại website: http://smi-protege.stanford.edu/repos/protege/owl/trunk.

Hiện tại, Protégé đã có phiên bản 4.1 hỗ trợ OWL 2. Tuy nhiên, do phiên bản này chưa có API hỗ trợ nên chúng em quyết định dùng phiên bản Protégé 3.4.4 có API hỗ trợ cho việc xây dựng công cụ làm giàu sau này. Công cụ Protégé có thể chia làm 2 loại là: Protégé-Frame và Protégé-OWL:

- Protégé-Frame cung cấp cho người dùng một giao diện chứa đầy đủ công cụ và kiến thức để hỗ trợ người dùng xây dựng và lưu trữ các ontology chuyên ngành dựa trên khung, tùy biến các hình thức nhập dữ liệu, và nhập dữ liệu tức thời.
- Protégé-OWL là một công cụ mở rộng của Protégé hỗ trợ các Web Ontology Language (OWL). Công cụ soạn thảo Protégé-OWL cho phép người dùng lưu và

xem các ontology OWL và RDF, xem và chỉnh sửa các lớp, cá thể, thuộc tính, quan hệ và các ràng buộc, kiểm tra tính đúng đắn của ontology.

Theo chúng em nhận xét thì công cụ Protégé-Frame sẽ phù hợp hơn cho nhu cầu xem chỉnh sửa và nhập dữ liệu cho ontology, trong khi nếu muốn xây dựng ontology mới và có giao diện phù hợp với việc xây dựng cấu trúc cho ontology thì dùng Protégé-OWL sẽ dễ dàng hơn. Ngoài ra, Protégé-OWL cũng hỗ trợ cho ngôn ngữ OWL tốt hơn là Protégé-Frame. Vì vậy, chúng em sẽ sử dụng công cụ Protégé-OWL để xây dựng ontology. Hướng dẫn sử dụng công cụ Protégé-OWL được nêu trong phần phụ lục A.



Hình 4: Giao diện protégé 3.4.4

❖ Các ưu điểm của Protégé là:

– Hỗ trợ đầy đủ ba phiên bản của ngôn ngữ OWL là OWL-Full, OWL-Lite và OWL-DL.

- Nhờ sử dụng mô hình hướng đối tượng của ngôn ngữ Java, Protégé rất hiệu quả trong việc mô hình hóa các lớp, thực thể, quan hệ…
- Giao diện thiết kế trực quan có tính tương tác cao. Người sử dụng có thể định nghĩa các thành phần của ontology trực tiếp từ các form. Nó hỗ trợ xây dựng các thành phần của một ontology rất nhanh và hiệu quả.
- Cho phép biểu diễn trực quan ontology dưới dạng các sơ đồ.
- Cho phép xây dựng ontology từ nhiều nguồn khác nhau.
- Protégé tự động lưu một bản tạm của ontology. Nếu có lỗi phát sinh trong quá trình thao tác thì ontology cũ sẽ tự động được phục hồi.
- Cung cấp chức năng tìm kiếm lỗi, kiểm tra tính nhất quán và đầy đủ của ontology.
- Cho phép các lớp và thuộc tính của ontology này có thể được sử dụng trong một Namespace khác mà chỉ cần sử dụng các URL để tham khảo.
- Hỗ trợ suy luận trực tiếp trên ontology dựa trên Interface chuẩn DL Implementation Group (DIG).
- Hỗ trợ sinh mã tự động. Protégé cho phép chuyển ontology thành mã nguồn RDF/XML, OWL, DIG, Java, EMF Java Interfaces, Java Schema Classes.. Các mã này có thể được nhúng trực tiếp vào ứng dụng và là đầu vào cho các thao tác trên ontology khi cần.

3.1.2. Quá trình xây dựng ontology

a. Xác định yêu cầu của ontology: ontology lưu trữ những thông tin về công nghệ thông tin bằng tiếng Việt đáp ứng được các nhu cầu của hệ thống như:

- Tìm kiếm thực thể có tên, không tên, xác định quan hệ giữa các thực thể

- Hỗ trợ trả lời cho hệ thống hỏi đáp về:
  - Các khái niệm trong lĩnh vực công nghệ thông tin.
  - Thông tin các chuyên gia trong lĩnh vực, các công ty hoạt động trong lĩnh vực công nghệ thông tin, giải thưởng, hội thảo, sự kiện, tổ chức, hiệp hội công nghệ thông tin và các trường có đào tạo công nghệ thông tin.

    Ví dụ:

    Java là gì?

    Trường nào đã đoạt giải vô địch cuộc thi Robocon năm 2009?
- Hỗ trợ phân tích bài báo công nghệ thông tin tiếng Việt.
- Sử dụng cho hệ thống tư vấn về chương trình đào tạo công nghệ thông tin.

b. Xem xét các ontology có sẵn: Theo khảo sát của chúng em thì có 3 ontologies về công nghệ thông tin có thể xem xét.

Thứ nhất là ontology về khoa học công nghệ của trường Đại học Bách Khoa Hà Nội đã khảo sát ở trên. Tuy nhiên ontology này không thể tìm được nguồn và cũng khi gửi mail liên lạc với tác giả thì không nhận được phản hồi, do đó không thể kế thừa được từ ontology này.

Thứ hai là ontology tổng quát OVL có chứa dữ liệu về công nghệ thông tin nhưng cấu trúc không phù hợp và chứa nhiều dữ liệu tổng quát thuộc nhiều lĩnh vực nên chúng em quyết định chỉ xem xét nhập một số dữ liệu chọn lọc từ ontology này chứ không sử dụng nó.

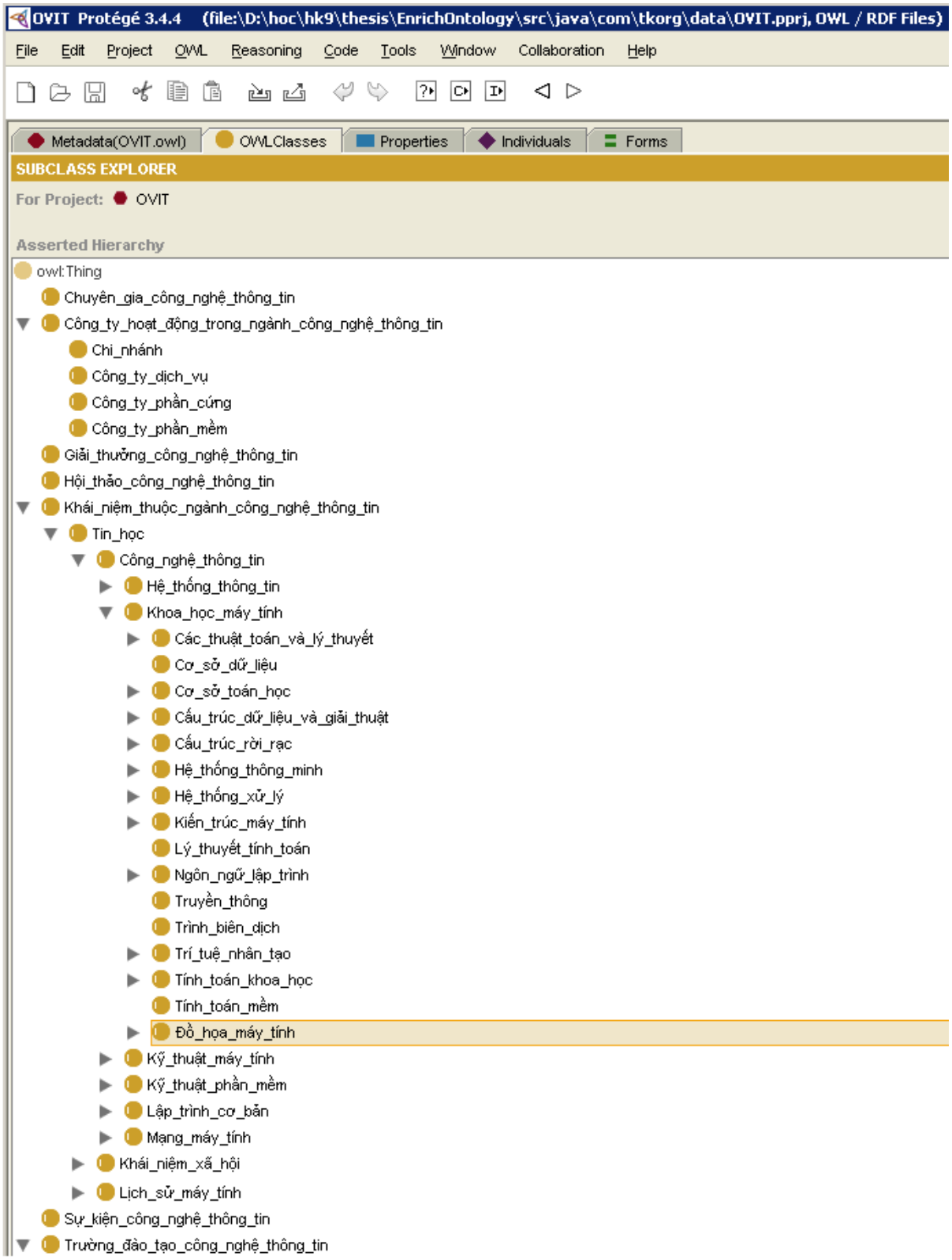
Cuối cùng là ComputingOntology là một ontology khá đầy đủ về các khái niệm và môn học trong lĩnh vực công nghệ thông tin tiếng Anh với cấu trúc của các khái niệm lấy từ ACM. Vì vậy, chúng em sẽ xây dựng cấu trúc lớp khái niệm trong ontology của mình theo cấu trúc lớp của ontology này bằng cách dịch và nhập bằng

tay để có thể chỉnh sửa một số chi tiết cho phù hợp với yêu cầu đối với ontology của mình.

Hiện nay, cũng có một số nghiên cứu đã xây dựng ontology theo cách sử dụng một số công cụ dịch kết hợp với các chuyên gia chỉnh sửa lại như ontology được xây dựng trong đề tài [22]. Tuy nhiên việc dịch như vậy thì cấu trúc của ontology được xây dựng có thể không đáp ứng được yêu cầu của ứng dụng.

c. Một số thuật ngữ quan trọng trong ontology: Dựa vào yêu cầu đã xác định ở trên chúng ta sẽ có một số khái niệm chính trong ontology như: Khái niệm trong lĩnh vực công nghệ thông tin, nguồn, định nghĩa, sự kiện công nghệ thông tin, công ty phần mềm, công ty phần cứng, chuyên gia công nghệ thông tin, trường đào tạo ngành công nghệ thông tin, tổ chức, hiệp hội, giải thưởng công nghệ thông tin, hợp tác đào tạo, sản xuất, trao giải thưởng, được trao giải thưởng.

d. Xây dựng cấu trúc lớp cho ontology: Dựa vào những thuật ngữ chính đã xác định ở trên và nguồn dữ liệu lấy từ website Wikipedia tiếng Việt chúng em đã xây dựng cấu trúc của ontology gồm các lớp chính như sau:


OVIT Protégé 3.4.4 (file:\D:\hoc\hk9\thesis\EnrichOntology\src\java\com\tkorg\data\OVIT.pprj, OWL / RDF Files)
File Edit Project OWL Reasoning Code Tools Window Collaboration Help
Metadata(OVIT.owl) OWLClasses Properties Individuals Forms
SUBCLASS EXPLORER
For Project: OVIT
Asserted Hierarchy
owl:Thing
Chuyên_gia_công_nghệ_thông_tin
Công_ty_hoạt_động_trong_ngành_công_nghệ_thông_tin
Chi_nhánh
Công_ty_dịch_vụ
Công_ty_phần_cứng
Công_ty_phần_mềm
Giải_thưởng_công_nghệ_thông_tin
Hội_thảo_công_nghệ_thông_tin
Khái_niệm_thuộc_ngành_công_nghệ_thông_tin
Tin_học
Công_nghệ_thông_tin
Hệ_thống_thông_tin
Khoa_học_máy_tính
Các_thuật_toán_và_lý_thuyết
Cơ_sở_dữ_liệu
Cơ_sở_toán_học
Cấu_trúc_dữ_liệu_và_giải_thuật
Cấu_trúc_rời_rạc
Hệ_thống_thông_minh
Hệ_thống_xử_lý
Kiến_trúc_máy_tính
Lý_thuyết_tính_toán
Ngôn_ngữ_lập_trình
Truyền_thông
Trình_biên_dịch
Trí_tuệ_nhân_tạo
Tính_toán_khoa_học
Tính_toán_mềm
Đồ_họa_máy_tính
Kỹ_thuật_máy_tính
Kỹ_thuật_phần_mềm
Lập_trình_cơ_bản
Mạng_máy_tính
Khái_niệm_xã_hội
Lịch_sử_máy_tính
Sự_kiện_công_nghệ_thông_tin
Trường_đào_tạo_công_nghệ_thông_tin

Hình 5: Các lớp chính trong ontology ITVO

Các lớp chính trong ontology được xây dựng dựa vào cấu trúc trong Wikipedia và ComputingOntology:

Khái niệm thuộc ngành công nghệ thông tin: tất cả các khái niệm đều được chuyển thành lớp con của lớp này, khi thêm vào những khái niệm mới sẽ là lớp con của các lớp bên dưới.

Tin học

Công nghệ thông tin

Lập trình cơ bản

Hệ thống thông tin

Khoa học máy tính

Mạng máy tính

Kỹ thuật phần mềm

Kỹ thuật máy tính

Khái niệm trong xã hội

Bảo mật

Hệ thống pháp lý

Hợp đồng

Kiểm soát

Sở hữu trí tuệ

Trách nhiệm nghề nghiệp

Đạo đức nghề nghiệp

Tác động của thay đổi công nghệ

Lịch sử máy tính

Thiết bị máy móc ban đầu

Hệ thống phần mềm phần cứng

Phần cứng không thuộc hệ thống

Phần mềm không thuộc hệ thống

Sự kiện công nghệ thông tin: gồm nhiều lớp con là những năm có xảy ra sự kiện, mỗi sự kiện là một cá thể của lớp năm.

Công ty hoạt động trong ngành công nghệ thông tin

Công ty phần mềm

Công ty phần cứng

Công ty dịch vụ

Chi nhánh

Trường đào tạo công nghệ thông tin

Trung tâm dạy nghề CNTT

Trung cấp

Cao đẳng

Đại học

Tổ chức/hiệp hội công nghệ thông tin

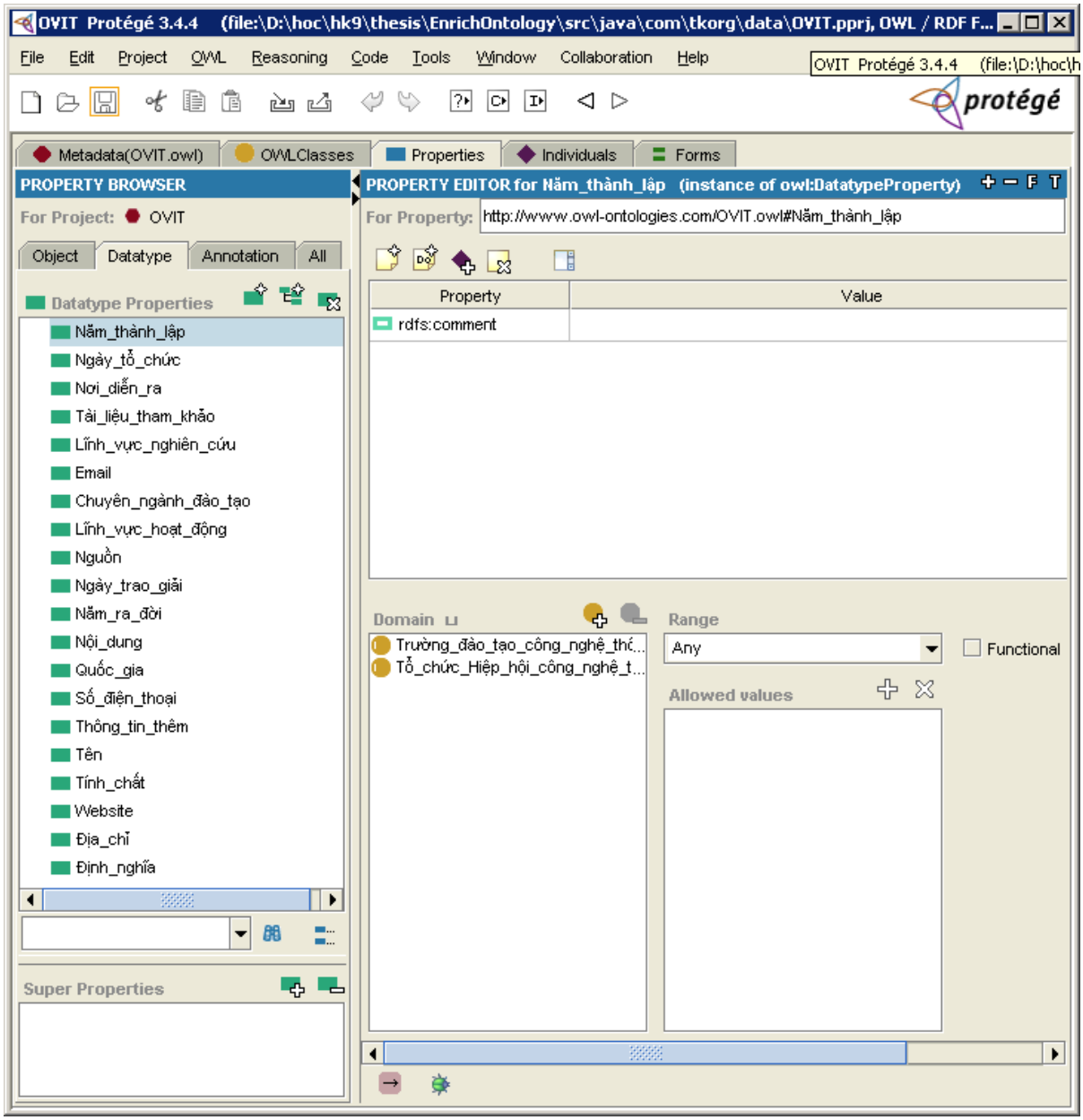
Giải thưởng công nghệ thông tin: Tên của mỗi giải thưởng là một lớp, mỗi lần giải thưởng được trao tạo ra một cá thể của giải thưởng đó.

Chuyên gia công nghệ thông tin: là những người có học vị tiến sĩ trở lên và có các bài báo khoa học chuyên ngành công nghệ thông tin.

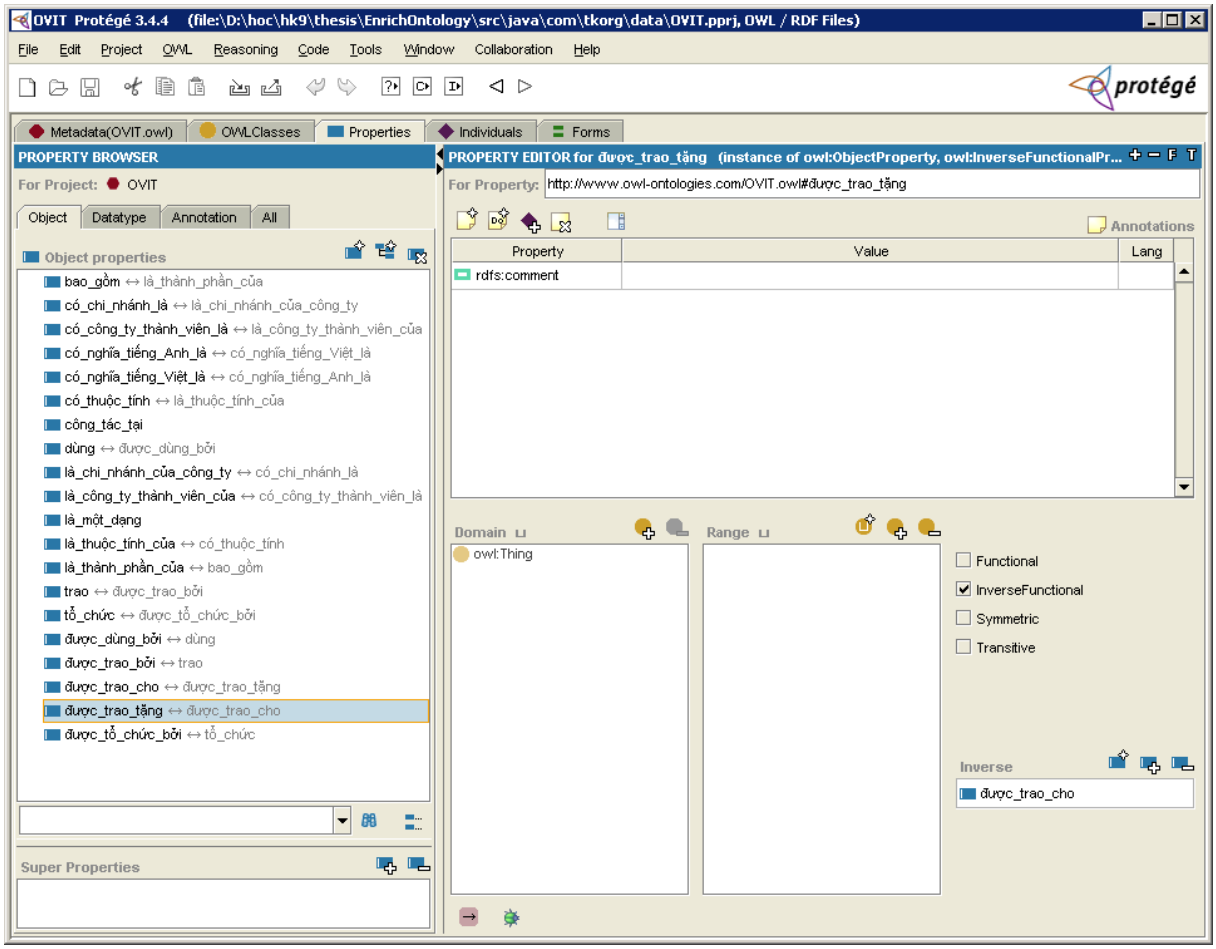
Hội thảo công nghệ thông tin: Giống như lớp *Giải thưởng công nghệ thông tin* ở trên, tên của mỗi hội thảo là một lớp, mỗi lần hội thảo được tổ chức tạo ra một cá thể của hội thảo đó.

Ngoài ra, chúng em đã nhập dữ liệu bổ sung thêm các khái niệm cho ontology bằng cách dịch và nhập bằng tay các lớp từ ComputingOntology vào như là các lớp con của lớp *Khái niệm thuộc ngành công nghệ thông tin*. Như vậy cấu trúc của các lớp khái niệm sẽ theo cấu trúc của ComputingOntology.

e. Định nghĩa các thuộc tính và quan hệ cho lớp:



Hình 6: Các thuộc tính trong ontology ITVO



Hình 7: Các quan hệ trong ontology ITVO

Khái niệm thuộc ngành công nghệ thông tin:

Thuộc tính:

Định nghĩa: là nội dung định nghĩa làm rõ cho khái niệm.

Nguồn: là đường dẫn tới trang web chứa định nghĩa.

Sự kiện công nghệ thông tin:

Thuộc tính:

Nguồn: Là đường dẫn đến bài báo chi tiết về sự kiện

Nội dung: nội dung sự kiện tóm tắt

Quan hệ: mỗi cá thể này có quan hệ tùy theo nội dung của sự kiện. Ví dụ như sự kiện do công ty tổ chức thì cá thể sự kiện này sẽ có quan hệ "được tổ chức bởi" với cá thể của công ty tương ứng, và ngược lại, cá thể của công ty sẽ có quan hệ "tổ chức" với cá thể sự kiện. Hai quan hệ "được tổ chức bởi" và "tổ chức" là hai quan hệ nghịch đảo với nhau.

Công ty hoạt động trong ngành công nghệ thông tin:

Thuộc tính:

Tên: Tên công ty

Địa chỉ

Số điện thoại

Lĩnh vực hoạt động

Quốc gia: Hoạt động ở nước nào

Website: trang web chính thức của công ty

Thông tin thêm: Một số thông tin của công ty ngoài những thông tin trên.

Quan hệ:

- Công ty phần cứng, Công ty phần mềm, Công ty dịch vụ: có quan hệ "có công ty thành viên là" đối với các tập đoàn có nhiều công ty con, và quan hệ nghịch đảo của nó là "là công ty thành viên của"
- Công ty phần cứng, Công ty phần mềm, Công ty dịch vụ: có quan hệ "có chi nhánh là" với một hoặc nhiều cá thể của lớp Chi nhánh
- Chi nhánh: có quan hệ "là chi nhánh của công ty" với cá thể công ty, là quan hệ nghịch đảo của quan hệ "có chi nhánh là".

- có quan hệ "tổ chức" với cá thể của Sự kiện công nghệ thông tin, Hội thảo công nghệ thông tin, và là quan hệ nghịch đảo của quan hệ "được tổ chức bởi".
- có quan hệ "được trao tặng" với một hay nhiều cá thể của lớp Giải thưởng công nghệ thông tin, và là quan hệ nghịch đảo với quan hệ "được trao cho".

Trường đào tạo công nghệ thông tin:

Thuộc tính:

Tên

Địa chỉ

Năm thành lập

Số điện thoại

Website

Chuyên ngành đào tạo

Thông tin thêm

Quan hệ:

- có quan hệ "tổ chức" với cá thể của Sự kiện công nghệ thông tin, Hội thảo công nghệ thông tin, và là quan hệ nghịch đảo của quan hệ "được tổ chức bởi".
- có quan hệ "được trao tặng" với một hay nhiều cá thể của lớp Giải thưởng công nghệ thông tin, và là quan hệ nghịch đảo với quan hệ "được trao cho".

Hướng phát triển: Sẽ thêm vào những thông tin về Chương trình đào tạo, Môn học để bổ sung dữ liệu đáp ứng được yêu cầu của ứng dụng trả lời cho những câu hỏi tư vấn về các chương trình đào tạo công nghệ thông tin.

Tổ chức/hiệp hội công nghệ thông tin:

Thuộc tính:

Tên

Địa chỉ

Quốc gia

Năm thành lập

Website

Thông tin thêm

Quan hệ:

- có quan hệ "trao" với 1 hoặc nhiều cá thể của Giải thưởng công nghệ thông tin, và là quan hệ nghịch đảo của quan hệ "được trao bởi".
- có quan hệ "tổ chức" với cá thể của Sự kiện công nghệ thông tin, Hội thảo công nghệ thông tin, và là quan hệ nghịch đảo của quan hệ "được tổ chức bởi".

Giải thưởng công nghệ thông tin:

Thuộc tính:

Năm ra đời

Ngày trao giải

Thông tin thêm

Quan hệ:

- có quan hệ "được trao bởi" với cá thể của lớp Tổ chức/hiệp hội công nghệ thông tin, trao tặng là quan hệ nghịch đảo với quan hệ "trao".
- có quan hệ "được trao cho" với cá thể của các lớp Chuyên gia, Công ty hoặc Trường đào tạo công nghệ thông tin, và là quan hệ nghịch đảo của quan hệ "được trao tặng".

Chuyên gia công nghệ thông tin:

Thuộc tính:

Tên

Email

Quốc gia

Lĩnh vực nghiên cứu

Tài liệu tham khảo: những bài báo, sách, công trình nghiên cứu có sự tham gia của chuyên gia.

Quan hệ:

- có quan hệ "công tác tại" với cá thể của các lớp Trường đào tạo công nghệ thông tin, Công ty hoạt động trong ngành công nghệ thông tin hay Tổ chức/hiệp hội công nghệ thông tin.
- có quan hệ "được trao tặng" với cá thể của lớp Giải thưởng công nghệ thông tin, và là quan hệ nghịch đảo với quan hệ "được trao cho".

Hội thảo công nghệ thông tin:

Thuộc tính:

Nơi diễn ra

Ngày tổ chức

Quan hệ:

– có quan hệ "được tổ chức bởi" với cá thể của lớp Tổ chức/hiệp hội công nghệ thông tin, Trường đào tạo công nghệ thông tin, Công ty hoạt động trong ngành công nghệ thông tin, và là quan hệ nghịch đảo của quan hệ "tổ chức".

f. Định nghĩa về thuộc tính và quan hệ của lớp:

Các ràng buộc về thuộc tính:

Thuộc tính	**Lượng số**	**Kiểu**	**Miền**
Định nghĩa	Đa trị	String	Khái niệm trong ngành công nghệ thông tin
Nguồn	Đa trị	String	Khái niệm thuộc ngành công nghệ thông tin Sự kiện công nghệ thông tin
Nội dung	1	String	Sự kiện công nghệ thông tin
Tên	Đa trị	String	Công ty hoạt động trong ngành công nghệ thông tin Trường đào tạo công nghệ thông tin Tổ chức/hiệp hội công nghệ thông tin Chuyên gia công nghệ thông tin
Địa chỉ	Đa trị	String	Công ty hoạt động trong ngành công nghệ thông tin Trường đào tạo công nghệ thông tin

			Tổ chức/hiệp hội công nghệ thông tin
Số điện thoại	Đa trị	String	Công ty hoạt động trong ngành công nghệ thông tin Trường đào tạo công nghệ thông tin
Lĩnh vực hoạt động	Đa trị	String	Công ty hoạt động trong ngành công nghệ thông tin
Quốc gia	Đa trị	String	Công ty hoạt động trong ngành công nghệ thông tin Tổ chức/hiệp hội công nghệ thông tin
Website	Đa trị	String	Công ty hoạt động trong ngành công nghệ thông tin Trường đào tạo công nghệ thông tin Tổ chức/hiệp hội công nghệ thông tin
Thông tin thêm	1	String	Công ty hoạt động trong ngành công nghệ thông tin Trường đào tạo công nghệ thông tin Tổ chức/hiệp hội công nghệ thông tin Giải thưởng công nghệ thông tin
Năm thành lập	1	Date	Trường đào tạo công nghệ thông tin Tổ chức/hiệp hội công nghệ thông tin
Chuyên ngành đào tạo	Đa trị	String	Trường đào tạo công nghệ thông tin
Năm ra đời	1	Date	Giải thưởng công nghệ thông tin

Ngày trao giải	1	Date	Giải thưởng công nghệ thông tin
Email	Đa trị	String	Chuyên gia công nghệ thông tin
Lĩnh vực nghiên cứu	Đa trị	String	Chuyên gia công nghệ thông tin
Tài liệu tham khảo	Đa trị	String	Chuyên gia công nghệ thông tin
Nơi diễn ra	Đa trị	String	Hội thảo công nghệ thông tin
Ngày tổ chức	Đa trị	Date	Hội thảo gia công nghệ thông tin

Các ràng buộc về quan hệ:

Tên quan hệ	**Lớp có quan hệ**	**Lớp được quan hệ**	**Số lượng**
được tổ chức bởi	Sự kiện công nghệ thông tin	Công ty, Tổ chức/ Hiệp hội, Trường đào tạo công nghệ thông tin	0, 1 hoặc nhiều
tổ chức	Công ty, Tổ chức/ Hiệp hội, Trường đào tạo công nghệ thông tin	Sự kiện công nghệ thông tin	0, 1 hoặc nhiều
có công ty thành viên là	Công ty hoạt động trong ngành công nghệ thông tin	Công ty hoạt động trong ngành công nghệ thông tin	0, 1 hoặc nhiều
là công ty thành viên của	Công ty hoạt động trong ngành công	Công ty hoạt động trong ngành công nghệ thông	0 hoặc1

	nghệ thông tin	tin	
có chi nhánh là	Công ty phần cứng, Công ty phần mềm, Công ty dịch vụ	Chi nhánh	0, 1 hoặc nhiều
là chi nhánh của công ty	Chi nhánh	Công ty phần cứng, Công ty phần mềm, Công ty dịch vụ	0 hoặc 1
được trao tặng	Công ty hoạt động trong ngành công nghệ thông tin, Trường đào tạo công nghệ thông tin, Chuyên gia công nghệ thông tin	Giải thưởng công nghệ thông tin	0, 1 hoặc nhiều
được trao cho	Giải thưởng công nghệ thông tin	Công ty hoạt động trong ngành công nghệ thông tin. Trường đào tạo công nghệ thông tin, Chuyên gia công nghệ thông tin	1 hoặc nhiều
trao	Tổ chức/hiệp hội công nghệ thông tin	Giải thưởng công nghệ thông tin	0, 1 hoặc nhiều
được trao bởi	Giải thưởng công	Tổ chức/hiệp hội công	1

	nghệ thông tin	nghệ thông tin	
công tác tại	Chuyên gia công nghệ thông tin	Trường đào tạo công nghệ thông tin, Công ty hoạt động trong ngành công nghệ thông tin, Tổ chức/hiệp hội công nghệ thông tin	0, 1 hoặc nhiều

g. Nhập các cá thể vào ontology: Nhóm tập hợp nguồn dữ liệu từ Wikipedia, tin tức trên các trang báo điện tử như: tuoitre.vn, vnexpress.net, vietnamnet… và thông tin từ trang web của Bộ thông tin truyền thông [30].

Kết quả xây dựng và nhập dữ liệu cho ontology:

Xây dựng cấu trúc ontology – ITVO với 980 lớp, trong đó có gần 950 lớp là các khái niệm công nghệ thông tin lấy từ nhiều nguồn trên internet chủ yếu là Wikipedia và ComputingOntology. Nhập được 50 cá thể, 19 thuộc tính của các lớp và 20 quan hệ.

3.2. Phương pháp làm giàu ontology tiếng Việt chuyên ngành công nghệ thông tin

3.2.1. Giới thiệu

Trước tiên chúng ta cần phân tích một chút về việc làm giàu ontology. Chúng ta có thể hiểu một cách đơn giản là việc làm giàu ontology là bổ sung dữ liệu và mở rộng cấu trúc của ontology, làm cho nó chứa nhiều thông tin hơn. Từ đó các ứng dụng sử dụng nó sẽ "thông minh" hơn trong việc "hiểu" và trả lời những kiến thức liên quan. Hiện nay, có các phương thức làm giàu ontology như làm giàu thủ công, tự động và bán tự động.

- Làm giàu ontology bằng phương pháp thủ công: Các chuyên gia dùng các công cụ soạn thảo ontology có sẵn như Protégé chẳng hạn để nhập dữ liệu làm giàu ontology từ nguồn dữ liệu do con người chọn lọc.
- Làm giàu ontology bằng phương pháp bán tự động (Semi-automatically): xây dựng công cụ làm giàu ontology từ nguồn xác định trước. Công cụ sẽ tự động chọn lọc và rút trích dữ liệu tương ứng với các thành phần có trong ontology, có sự tham gia chọn lọc lại của chuyên gia rồi mới cập nhật vào làm giàu ontology.
- Làm giàu ontology bằng phương pháp tự động (Automatic): xây dựng công cụ làm giàu ontology từ nguồn xác định trước. Công cụ này có thể tự động tìm kiếm, chọn lọc tài liệu từ nguồn và rút trích các thông tin cần thiết để làm giàu ontology. Công cụ này sẽ tự động cập nhật dữ liệu được rút trích vào ontology mà không cần hỏi ý kiến chuyên gia.

Vì sao cần phải xây dựng công cụ làm giàu ontology tự động và bán tự động? Đó là vì kiến thức của một lĩnh vực rất lớn, lượng thông tin cần lưu trữ của một ontology chuyên ngành để có thể sử dụng được cho các ứng dụng cũng phải thật phong phú và từ nhiều nguồn khác nhau. Với phương pháp nhập thủ công bởi con người thì sẽ tốn rất nhiều thời gian và chi phí. Vì vậy chúng ta cần có một công cụ để thu thập dữ liệu và nhập thông tin vào ontology một cách tự động. Tuy nhiên, nếu làm tự động từ bước từ thập dữ liệu đến việc rút trích và lưu trữ thông tin vào ontology thì độ chính xác không cao vì mỗi giai đoạn đều có xác xuất sai của nó, do đó chúng em sẽ xây dựng công cụ làm giàu ontology bán tự động. Có nghĩa là từ việc thu thập dữ liệu đến rút trích ra thông tin để lưu trữ sẽ thực hiện tự động, sau đó sẽ cho người dùng kiểm tra và xem xét lại kết quả trước khi lưu trữ vào ontology. Như vậy sẽ giúp loại bỏ bớt một số kết quả sai, giảm bớt lượng thông tin rác trong ontology.

Quá trình làm giàu ontology tự động hay bán tự động trong giai đoạn phát triển ontology gọi là quá trình học của ontology (ontology learning) [42]. Ontology có thể

học từ text, từ từ điển, từ cơ sở tri thức (knowledge base), từ những lược đồ bán cấu trúc hoặc từ lược đồ quan hệ [43].

Theo tài liệu [44] thì việc học của ontology được chia thành 2 loại là học từ lúc xây dựng ontology và học để mở rộng ontology hiện có. Do đó việc học của ontology sẽ gồm có những giai đoạn sau:

- Trong trường hợp chưa có ontology thì sẽ tiến hành tập hợp lại những thuật ngữ thành dạng cây khái niệm để xây dựng ontology ở dạng thô. Giai đoạn này có thể biểu diễn bằng vec tơ không gian, mạng liên kết, lý thuyết tập hợp.
- Với trường hợp đã có sẵn ontology thì việc học của ontology với dữ liệu đã có sẵn sẽ bắt đầu với việc phân lớp tài liệu. Giai đoạn này sẽ dùng những đặc trưng của tập dữ liệu có sẵn đã được phân lớp để huấn luyện cho một phương pháp máy học (Machine Learning) để tạo ra bộ phân lớp (classifier). Sau đó dùng bộ phân lớp này để phân lớp các tài liệu dùng để làm giàu ontology. Cuối cùng là rút trích thông tin cho các thành phần của ontology từ tài liệu có liên quan đã được phân lớp trước đó. Các hệ thống rút trích từ trước đến nay được thiết kế để rút trích các thực thể có tên nên chủ yếu được dùng để tìm các thực thể của các khái niệm.

3.2.2. Khảo sát phương pháp làm giàu ontology

a. Phương pháp làm giàu Wordnet [45]

Trong nghiên cứu này, tác giả đã xây dựng công cụ tự động làm giàu Wordnet từ nguồn internet. Theo [46] thì Wordnet là một từ điển trực tuyến tiếng Anh dựa trên lý thuyết về ngôn ngữ tâm lý. Theo wikipedia nó được dùng như là một ontology từ vựng trong lĩnh vực khoa học máy tính. Nó bao gồm các danh từ, động từ, tính từ, trạng từ được sắp xếp theo nghĩa của chúng và có các quan hệ từ vựng-ngữ nghĩa. Trong nghiên cứu này, tác giả xây dựng công cụ tự động làm giàu Wordnet phiên bản 1.6 [50].

Phương pháp làm giàu trong nghiên cứu này là tạo ra câu truy vấn bằng cách dùng thông tin trong ontology cụ thể là các khái niệm của nó. Sau đó dùng câu truy vấn để tìm tất cả tài liệu có liên quan đến các khái niệm từ internet dùng công cụ tìm kiếm Alta Vista [47]. Các tài liệu tìm được sẽ được phân lớp theo các khái niệm trong ontology. Từ đó rút ra danh sách các từ có quan hệ ngữ nghĩa bằng cách đo số lần xuất hiện của từ trong mỗi tập đã phân lớp và thực hiện một số công thức tính toán khác [45]. Trong tài liệu tác giả không đề cập đến việc đã sử dụng phương pháp phân lớp nào.

Những khái niệm được rút ra sẽ được giải quyết vấn đề nhập nhằng ngữ nghĩa dùng tập SemCor [48] chứa các câu giải thích nghĩa của từ tương ứng với các khái niệm trong Wordnet (tập hợp những câu này trong Wordnet gọi là word sense). Cuối cùng sẽ gom lại danh sách các từ rút được theo nghĩa của từ trong Wordnet.

Nhận xét: Trong nghiên cứu này đã đề xuất phương pháp làm giàu Wordnet, một dạng ontology. Với phương pháp này thì chỉ có thể sử dụng cho Wordnet để làm giàu các khái niệm theo nghĩa của từ (word sense) vì nó phụ thuộc vào cấu trúc của Wordnet và tập SemCor. Vì vậy không thể áp dụng phương pháp này để làm giàu ontology tiếng Việt chuyên ngành Công nghệ thông tin.

b. Phương pháp làm giàu ontology về lĩnh vực sinh học

Để làm giàu cho ontology này, trong tài liệu [43] tác giả đề xuất phương pháp làm giàu bán tự động. Để thực hiện việc này, tác giả đã xây dựng framework cho việc học ontology (ontology learning), hệ thống này hỗ trợ tự động lấy tài liệu từ web dùng phương pháp crawl tập trung (focused crawling), đây là một cơ chế tìm kiếm tài liệu dựa trên kỹ thuật thông minh [43]. Sau đó phân lớp tài liệu dùng bộ phân lớp SVM (Support Vector Machine) để xác định những tài liệu liên quan đến lĩnh vực cần tìm. Cuối cùng hệ thống sẽ tự động rút trích những thông tin cần thiết để

làm giàu cá thể và thuộc tính cho ontology dùng phương pháp khai mỏ văn bản (text mining). Kết quả sau khi rút trích được sẽ được các chuyên gia chọn lọc lại rồi mới cập nhật vào ontology.

Nguồn dữ liệu để làm giàu ontology là từ internet, tác giả dùng crawler để tìm kiếm tài liệu kết hợp với công cụ tìm kiếm tổng quát như Google, Yahoo và công cụ tìm kiếm khoa học như Google Scholar và thư viện từ điển trực tuyến như amphibanat.org.

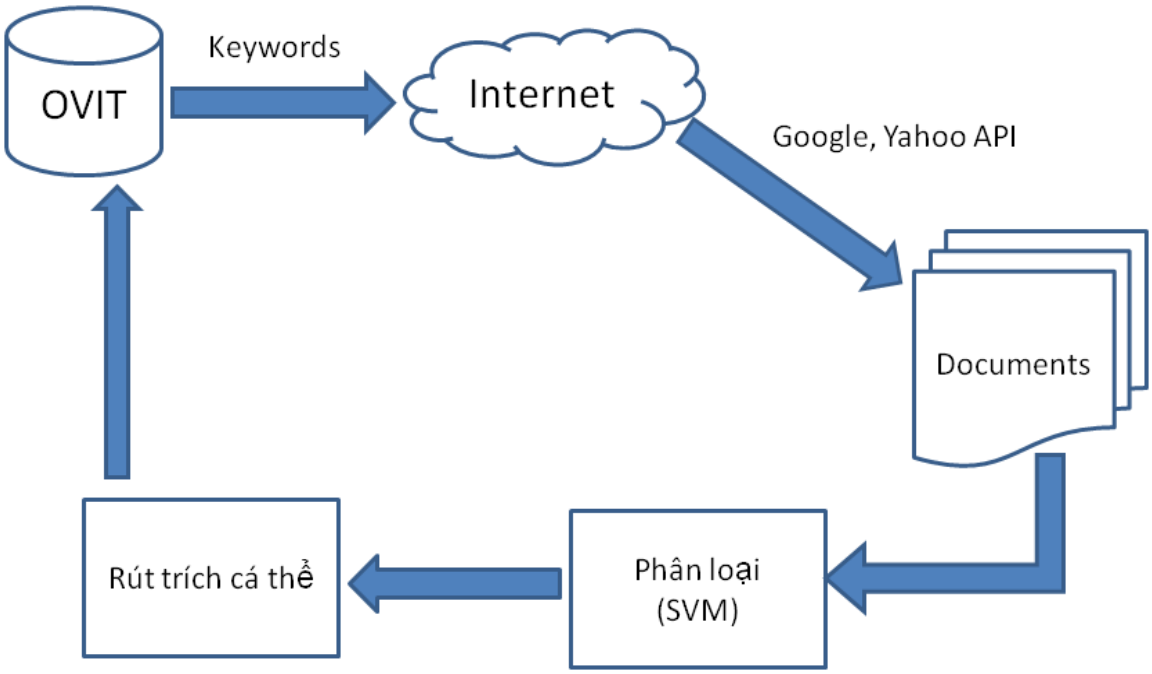
Để phân lớp tài liệu tác giả dùng công cụ phân lớp LibSVM [26] để phân tài liệu ra thành 2 lớp chính là có liên quan và không có liên quan đến lĩnh vực lưỡng cư và hình thái học.

Ngoài ra trong tài liệu [49] tác giả đã đề xuất dùng các từ biểu diễn nghĩa của khái niệm trong Wordnet (word sense) để làm giàu những mô tả khái niệm trong ontology về động vật lưỡng cư (một phần trong ontology về sinh học).

Nhận xét: Nguồn dữ liệu lấy từ internet dùng các công cụ tìm kiếm phổ biến hiện nay rất phong phú và đa dạng. Dùng SVM để phân lớp tài liệu với độ chính xác 77,5% (theo bài báo [51]) kết quả này theo chúng em là chấp nhận được. Và phương pháp làm giàu ontology bán tự động này theo chúng em sẽ cho kết quả tốt hơn vì có sự chọn lọc lại kết quả sau khi rút trích. Vì vậy chúng em sẽ thực hiện xây dựng công cụ làm giàu ontology tiếng Việt chuyên ngành công nghệ thông tin bằng phương pháp bán tự động.

3.2.3. Phương pháp thực hiện

Vì cấu trúc của mỗi ontology khác nhau nên công cụ làm giàu này chỉ phục vụ cho việc làm giàu ontology về công nghệ thông tin bằng tiếng Việt (ITVO). Tuy nhiên, việc xây dựng công cụ làm giàu cho các ontology khác có thể thực hiện tương tự.



Hình 8: Mô hình phương pháp làm giàu ontology

Việc làm giàu ontology có thể chia làm các phần nhỏ như: làm giàu các lớp, các cá thể, các quan hệ trong ontology. Ở mức giới hạn của đề tài tốt nghiệp, chúng em chỉ dừng ở việc xây dựng công cụ làm giàu cá thể. Hướng phát triển sau này của chúng em là cải tiến giai đoạn rút trích để làm giàu các lớp và các quan hệ trong ontology.

a. Nguồn dữ liệu

Chúng em sẽ làm giàu ontology ITVO từ nguồn dữ liệu là internet. Cụ thể sẽ đưa các từ khóa, chính là các khái niệm trong ontology lên internet và dùng Google, Yahoo API để tìm kiếm những tài liệu có liên quan làm nguồn dữ liệu đầu vào cho công cụ làm giàu ontology.

b. Phân loại

Sau khi dùng Google và Yahoo API để tìm kiếm các tài liệu có liên quan đến từ khóa trong ontology ta tiến hành phân loại các tài liệu đó. Phân loại văn bản là một tiến

trình đưa các văn bản chưa biết chủ đề vào các lớp văn bản đã biết (tương ứng với các chủ đề hay lĩnh vực khác nhau). Mỗi lĩnh vực được xác định bởi một số tài liệu mẫu của lĩnh vực đó. Để thực hiện quá trình phân loại, các phương pháp huấn luyện được sử dụng để xây dựng bộ phân loại từ các tài liệu mẫu, sau đó dùng bộ phân loại này để dự đoán lớp của những tài liệu mới (chưa biết chủ đề). Trong quá trình phân loại, các văn bản được biểu diễn dưới dạng vector với các thành phần (chiều) của vector này là các đặc trưng của lớp.

Cụ thể ở bài toán phân loại này các tài liệu được phân thành 2 lớp là lớp tài liệu về công nghệ thông tin và lớp tài liệu không thuộc công nghệ thông tin. Ở đây chúng em sử dụng thuật toán SVM để phân loại văn bản.

SVM (Support Vector Machine) là một phương pháp học có giám sát bằng cách phân tích dữ liệu và nhận ra các mẫu được sử dụng để phân lớp và phân tích hồi quy (wikipedia). Phương pháp này do Vapnik cùng nhóm nghiên cứu của ông đề nghị ở phòng thí nghiệm AT&T Bell vào năm 1992 [25].

❖ **Các phương pháp và công cụ hỗ trợ phân loại tài liệu**

- **Phương pháp tính trọng số TF*IDF:**

Đây là một phương pháp để đánh giá các thuật ngữ trong một tài liệu, là một cách định trọng số từ thông dụng. Ở đây ta sẽ dùng trọng số này để chọn các đặc trưng cho thuật toán phân loại SVM.

Tần suất từ (term frequency - TF): Trọng số từ là tần suất xuất hiện của từ đó trong tài liệu. Cách định trọng số này nói rằng một từ là quan trọng đối với một tài liệu nếu nó xuất hiện nhiều lần trong tài liệu đó.

TF IDF: Trọng số từ là tích của tần suất từ TF và tần suất tài liệu nghịch đảo của từ đó và được xác định bằng công thức

$$IDF = \log(N / DF) + 1$$

trong đó:

N là kích thước của tập tài liệu huấn luyện;

DF là tần suất tài liệu: là số tài liệu mà một từ xuất hiện trong đó.

Trọng số TFIDF kết hợp thêm giá trị tần suất tài liệu DF vào trọng số TF. Khi một từ xuất hiện trong càng ít tài liệu (tương ứng với giá trị DF nhỏ) thì khả năng phân biệt các tài liệu dựa trên từ đó càng cao. Các từ được dùng để biểu diễn các tài liệu cũng thường được gọi là các đặc trưng. Để nâng cao tốc độ và độ chính xác phân loại, tại bước tiền xử lý văn bản, ta loại bỏ các từ không có ý nghĩa cho phân loại văn bản.

Tại bước này, chúng ta gặp phải một bài toán nữa đó là tách từ tiếng Việt. Như chúng ta đã biết, nếu như đối với tài liệu tiếng Anh mỗi từ sẽ mang một nghĩa của riêng nó do vậy việc tách từ không mấy khó khăn và hiện nay cũng có nhiều công cụ hỗ trợ tốt cho việc này ví dụ như Gate. Đối với tiếng Việt thì mỗi từ (2 từ cách nhau bằng 1 khoảng trắng) nếu đi với những từ khác nhau sẽ có nghĩa khác nhau, có khi 2 hoặc 3 từ mới tạo thành nghĩa. Sau một thời gian khảo sát và tìm kiếm, chúng em quyết định sử dụng công cụ hỗ trợ tách từ tiếng Việt *vnTokenizer* do nhóm của tác giả Lê Hồng Phương xây dựng được nêu trong bài báo [23].

- **Công cụ tách từ vnTokenizer:**

Đây là công cụ tách từ tự đông cho tiếng Việt được viết bằng ngôn ngữ Java và độc lập nền. Phiên bản cũ nhất hiện giờ còn được công bố trên website chính thức của tác giả [24] là phiên bản vnTokenizer 2.0 được xây dựng vào năm 2005 khi đó nó mới là một ứng dụng đơn với giao diện đơn giản. Và phiên bản chúng em sử dụng là phiên bản mới nhất hiện giờ được công bố chính thức trên website vào ngày 4/8/2010, phiên bản vnTokenizer 4.1.1c.

Công cụ này được xây dựng sử dụng kết hợp từ điển (từ điển tiếng Việt được lấy từ đề tài VLSP [28]) và giải thuật ngram, trong đó mô hình ngram được huấn luyện

sử dụng treebank tiếng Việt (70,000 câu đã được tách từ). Treebank là kho ngữ liệu câu được chú giải ngữ pháp.

Với độ chính xác xấp xỉ 97% (theo thống kê của tác giả trên website[24]) công cụ có thể thực hiện tốt việc tách từ tiếng Việt nên chúng em quyết định sử dụng nó cho công đoạn tiền xử lý tài liệu để rút ra các đặc trưng.

Sau khi tách từ, chúng ta tiến hành loại bỏ những hư từ trong tiếng Việt vì không những các từ này không có ý nghĩa gì đối với việc phân lớp mà nó còn có thể gây nhiễu cho việc tìm các đặc trưng. Danh sách hư từ tham khảo từ website [29] và từ đề tài "*Nghiên Cứu Phân Loại Văn Bản Tiếng Việt*" của Trịnh Quốc Sơn. Danh sách tổng hợp các hư từ được liệt kê trong bảng phụ lục B.

- **Công cụ LibSVM:**

Đây là một thư viện đơn giản, dễ sử dụng và hiệu quả đối với việc phân lớp bằng SVM. Thư viện này được tạo ra bởi hai tác giả Chih-Chung Chan và Chih-Jen Lin. Mục tiêu của nó là để giúp người dùng từ các lĩnh vực khác nhau dễ dàng sử dụng như một công cụ. LIBSVM cung cấp một giao diện đơn giản mà người sử dụng có thể dễ dàng liên kết nó với các chương trình riêng của họ. Phiên bản hiện tại của của Libsvm là 3.0 được công bố vào tháng 9 năm 2010 [26].

Để có thể tiến hành sử dụng công cụ trên, ta phải xây dựng một tập tin huấn luyện và một tập tin để test. Hai tập tin này đều có định dạng giống nhau và được trình bày như bên dưới:

```
<label 1> <index1>:<value1> <index2>:<value2> ...
<label 2> <index1>:<value3> <index3>:<value4> ...
```

Trong đó:

- <label> là giá trị đích của tập huấn luyện. Đối với việc phân lớp, nó là một số nguyên xác định một lớp.
- <index> là một số nguyên bắt đầu từ 1. Cụ thể trong bài toán phân loại, nó đại diện cho các đặt trưng.
- <value> là một số thực. Giá trị này thể hiện mức độ liên quan của đặc trưng đối với một phân loại nằm trong khoảng [-1,1]. Nếu là đặc trưng nhị phân thì lúc huấn luyện giá trị này sẽ là 1.

Sau khi có được tập tin huấn luyện đúng định dạng, nhiệm vụ của libsvm là sẽ huấn luyện dựa trên tập tin định dạng và cho kết quả trả về là một tập tin *train_model* có đuôi là *.model*. Tập tin này là mô hình xây dựng dựa trên việc huấn luyện. Từ đó, ta chỉ việc sử dụng lại mô hình này để dự đoán các dữ liệu kiểm thử. Quá trình đưa dữ liệu kiểm thử cũng giống như huấn luyện, vẫn phải xây dựng tập tin kiểm thử theo định dạng như trên.

c. Rút trích cá thể

Sau khi đã phân loại các tài liệu tìm được, chúng ta tiến hành rút trích các cá thể trong các tài liệu thuộc lĩnh vực công nghệ thông tin. Trong phần này chúng em xin trình bày cách rút trích các khái niệm thuộc lĩnh vực công nghệ thông tin.

Như cấu trúc của ontology ITVO đã trình bày ở phần trên thì một cá thể của lớp *Khái niệm công nghệ thông tin* sẽ gồm có 2 thuộc tính là *Định nghĩa* và *Nguồn* của nó. Ở đây chúng ta sẽ rút ra định nghĩa của các khái niệm cũng chính là từ khóa được gửi lên để tìm kiếm tài liệu.

Sau khi đọc và tìm hiểu một số bài viết về cú pháp và ngôn ngữ tiếng Việt [14, 15, 22] và quan sát một tập các câu dùng để định nghĩa cho khái niệm hay thuật ngữ (khoảng 300 định nghĩa), chúng em đề xuất một số mẫu cho các câu định nghĩa của

một khái niệm. Các bước để xác định một câu là định nghĩa của một khái niệm như sau:

- Định nghĩa cho một khái niệm sẽ là 1 câu được rút ra từ văn bản dùng các dấu câu để tách câu (tham khảo từ [35]) như:
  + *dấu chấm .*
  + *dấu chấm hỏi ?*
  + *dấu cảm !*
  + *dấu lửng ...*
  + *dấu chấm phẩy ;*
  + *dấu ngoặc kép " "*
- Loại bỏ một số từ đứng đầu câu nhưng không có nghĩa như: Trong đó, Vì vậy, Theo đó, Do đó.
- Một câu thỏa một trong các mẫu sau sẽ được chọn là một khái niệm:
  + Có từ khóa (khái niệm) đứng đầu câu và theo sau là một trong các từ: là, có nghĩa là, được định nghĩa là, được hiểu như là, được hiểu là, có thể là, được biết như là, được biết là, được dùng trong, được dùng để, gồm, dấu ":", dấu "-".
  + Hoặc có từ khóa đứng đầu câu theo sau là một mệnh đề nằm trong dấu ngoặc đơn và tiếp theo là các từ như trên.
  + Từ khóa nằm cuối câu và trước nó là một trong các từ: gọi là, được gọi là.

Một cá thể được tạo ra với tên có dạng *tenlop_sothutu* với các thuộc tính *Định nghĩa* chính là câu định nghĩa rút được và *Nguồn* là từ nguồn của tài liệu chứa nó.

d. Lưu trữ cá thể

Trước khi lưu trữ vào ITVO, thì các cá thể sau khi rút trích sẽ được hiển thị lên cho người dùng xem và chỉnh sửa, có thể loại bỏ bớt những kết quả rút trích sai.

Sau khi người dùng đã đồng ý với kết quả thì sẽ lưu trữ vào ontology (ITVO) dùng API của Protégé.

3.3. Tổng kết chương

Trong chương này chúng em đã trình bày phương pháp để xây dựng và làm giàu ontology tiếng Việt chuyên ngành công nghệ thông tin (ITVO). Cụ thể phần này đã: giới thiệu công cụ để xây dựng ontology phổ biến hiện nay đó là Protégé, trình bày các bước để xây dựng nên ITVO dùng Protégé và thống kê kết quả công việc nhập dữ liệu cho ontology.

Ngoài ra, chúng em còn trình bày các phương pháp sử dụng ở từng giai đoạn để xây dựng công cụ làm giàu ontology, giới thiệu các công cụ, thuật toán và cách sử dụng chúng.

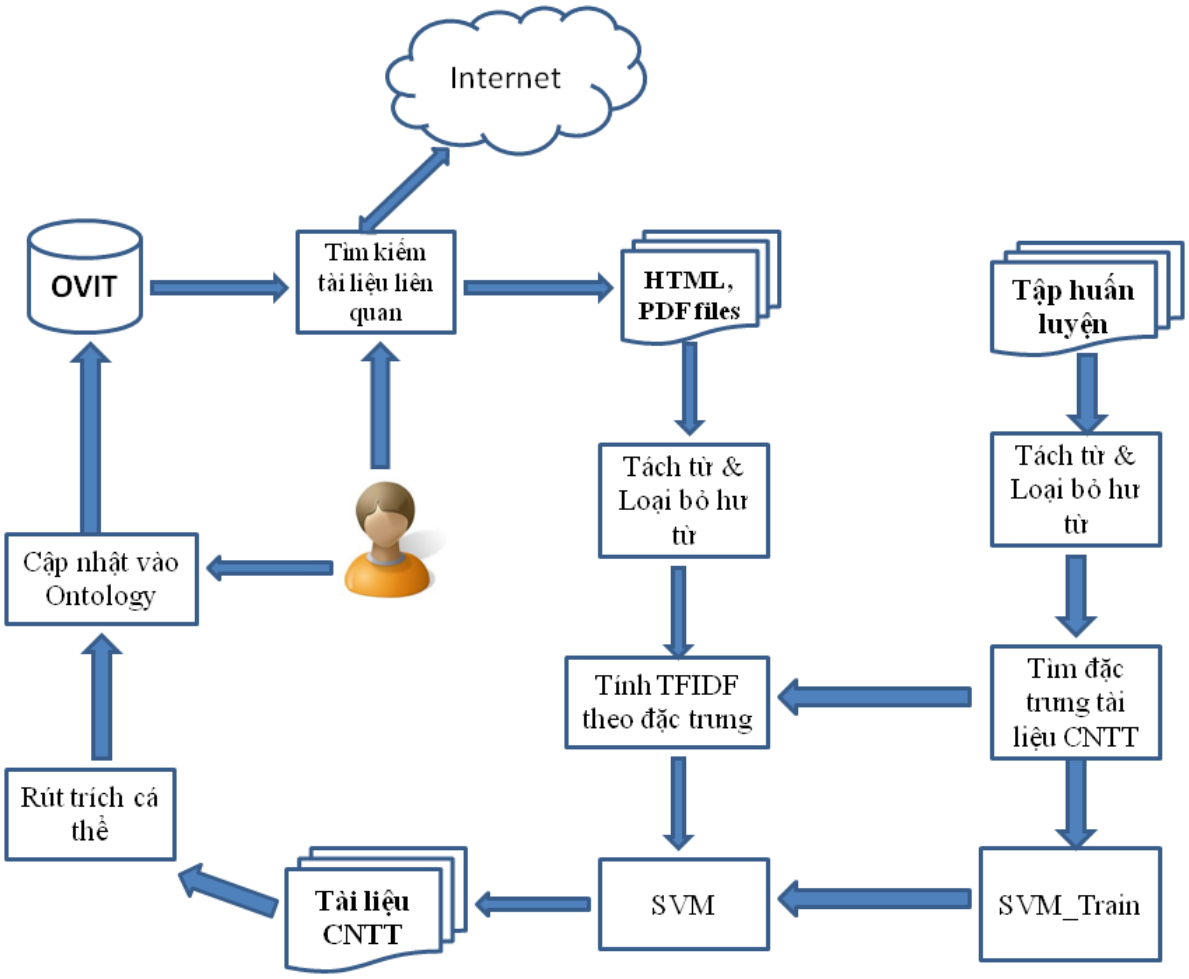
CHƯƠNG 4: HIỆN THỰC HỆ THỐNG VÀ ĐÁNH GIÁ

4.1. Mở đầu

Trong chương này chúng em sẽ hiện thực hệ thống làm giàu ontology sử dụng phương pháp và công cụ đã đề cập ở chương trước. Phần này sẽ trình bày chi tiết về kiến trúc của chương trình làm giàu ontology, cách cài đặt cũng như các bước chạy thử chương trình. Ngoài ra, chúng em sẽ nêu phần tự đánh giá về công cụ sau khi được hiện thực.

Công cụ dự kiến sẽ thực hiện việc tìm kiếm, phân loại tài liệu có liên quan và rút trích các cá thể của lớp Khái niệm thuộc ngành công nghệ thông tin một cách tự động. Sau đó sẽ cho phép người dùng chọn lọc lại rồi cập nhật vào ontology.

4.2. Kiến trúc chương trình làm giàu ontology



Hình 9: Kiến trúc chương trình làm giàu ontology ITVO

Công cụ được xây dựng trên nền web để có thể dễ dàng biểu diễn dữ liệu và chia sẻ hệ thống cho nhiều người có quan tâm đến ITVO và việc cập nhật nó.

Các package trong chương trình gồm:

- com.tkorg.search: dùng trong giai đoạn tìm kiếm link của các tài liệu liên quan.
  - ClassAction.java: dùng để load Ontology lên trang web.
  - GoogleSearchEngine.java: dùng để tìm kiếm trang web bởi google.
  - YahooSearchEngine.java: dùng để tìm kiếm trang web bởi yahoo.

  - SearchEngineAction.java: dùng để phân loại link nào sẽ được tìm kiếm theo google hay yahoo.
- Com.tkorg.token: dùng trong giai đoạn down, tách từ và loại bỏ stopwords của nội dung các link của các tài liệu liên quan.
  - SeperateWords.java: down nội dung theo các link của các tài liệu liên quan. Sau đó tách từ chúng. Nó cũng được dùng trong giai đoạn tách từ của quá trình Train.
  - Stopwords.java: load nội dung đã tách từ xong, và loại bỏ stopwords.
- com.tkorg.features: tính TFIDF, tìm đặc trưng.
  - DF.java: lớp thể hiện DF.
  - IDF.java: lớp thể hiện IDF.
  - MyFile.java: lớp thể hiện của một file (dùng trong tính TFIDF).
  - TF.java: lớp thể hiện TF.
  - TFIDF.java: lớp thể hiện TFIDF.
  - Features_Main.java: toàn bộ quá trình tính TFIDF hay tìm đặc trưng.
- com.tkorg.svm.train: thể hiện toàn bộ quá trình Train.
  - SVMTrain.java: sử dụng svm để train.
  - Train_Main.java: thực hiện quá trình Train từ tách từ, loại bỏ stopwords, tìm kiếm đặc trưng và sử dụng svm để train.
- com.tkorg.svm.classify: thể hiện toàn bộ quá trình từ lúc lấy link đến lúc phân loại bằng svm.
  - SVMTest.java: sử dụng svm để phân loại.
  - Classify_Main.java: thể hiện toàn bộ quá trình từ lấy link, download, tách từ, loại bỏ stopwords, tính TFIDF và sử dụng svm để phân loại.
- com.tkorg.extraction: thể hiện giai đoạn rút trích.
  - MyFile.java: lớp thể hiện của một file (dùng trong giai đoạn rút trích).
  - MyKeyword.java: lớp thể hiện của một từ khóa.

  - Extraction_Main.java: thể hiện giai đoạn rút trích.
- com.tkorg.util: chứa file Constant.java thể hiện các biến hằng.
- com.tkorg.data: chứa dữ liệu.
- com.tkorg.actions: xử lý ở tầng Action.
  - WelcomeAction.java: xử lý trong trang web *Welcome*.
  - SearchOntologyAction.java: xử lý trong trang web *SearchOntology*.
  - DisplayLinksAction.java: xử lý trong trang web *DisplayLinks*.
- com.tkorg.bussinesslogic: xử lý ở tầng Bussiness.
  - SearchOntologyBL.java: xử lý quá trình tìm kiếm link theo từ khóa.
  - DisplayLinksBL.java: xử lý quá trình download, tách từ, loại bỏ stopwords, tính TFIDF, phân loại và rút trích nội dung.
- Nhóm package vn.hus... : đây là api của tokenizer dùng để tách từ.

Chương trình thực hiện những chức năng chính: tìm kiếm tài liệu, phân loại tài liệu, rút trích cá thể từ tài liệu, và cập nhật cá thể vào ontology. Cụ thể như sau:

- Tìm kiếm tài liệu:
  - Package *ontology* chứa thư viện: icu4j_3_4, iri, jena, log4j-1.2.14, orphanNodesAlg, owlsyntax, protege, protege-owl, xercesImpl.
  - Package *search* chứa thư viện: htmlparser, json, yahoo_search-2.0.1.
  - Dùng thư viện pdfbox-app-1.4.0.

    Mục đích: cho phép người dùng chọn các khái niệm trong ontology để tìm kiếm tài liệu liên quan trên internet. Các tài liệu tìm thấy được tải xuống máy tính dưới dạng file html và pdf.
- Phân loại tài liệu:
  - Dùng thư viện libsvm.
  - Các package có dạng đuôi: vn.hus.

Mục đích là để chọn ra những tài liệu liên quan đến công nghệ thông tin và loại bỏ bớt những tài liệu không liên quan.Bước này gồm 2 quá trình *huấn luyện cho thuật toán SVM (tạo ra train_model)* và quá trình *phân loại tài liệu dùng SVM*. Trong hai quá trình, quá trình thứ nhất chỉ cần làm một lần sẽ tạo ra *model* có thể sử dụng nhiều lần cho quá trình thứ hai. Bây giờ ta sẽ xét từng quá trình:

Quá trình *huấn luyện cho thuật toán SVM*:

- Để huấn luyện cho thuật toán SVM trước tiên ta cần có *tập dữ liệu huấn luyện*: Ở đây, chúng em tập hợp được *tập huấn luyện* từ internet gồm 200 file text được chia làm hai phần: 100 file là *phần thuộc công nghệ thông tin* và 100 file *phần không thuộc công nghệ thông tin*.
- Tiếp theo, chúng ta cần tìm ra được đặc trưng của các tài liệu thuộc lĩnh vực công nghệ thông tin dùng phương pháp tính TFIDF. Tuy nhiên, để tăng độ chính xác cho các đặc trưng tìm được thì trước tiên các tài liệu trong *tập huấn luyện* cần được xử lý qua giai đoạn *tách từ và loại bỏ hư từ*: Quá trình tách từ sử dụng tokenizer để tách từ tiếng Việt. Sau đó, ta sẽ loại bỏ các hư từ trong các nội dung đó (đã được tách từ). Cuối cùng, các nội dung này (đã được tách từ và loại bỏ hư từ) sẽ được lưu xuống tương ứng với tên các file.
- Giai đoạn *tìm đặc trưng tài liệu công nghệ thông tin*: giai đoạn này ta có thể chia làm hai phân đoạn nhỏ.
  + Phân đoạn 1: ta sẽ tính TFIDF trong phần thuộc công nghệ thông tin của *tập huấn luyện* rồi chọn ra những từ có giá trị TFIDF lớn nhất làm đặc trưng (ở đây ta chọn 20 từ).
  + Phân đoạn 2: ta tính tần số xuất hiện của từng đặc trưng đó theo từng file thuộc công nghệ thông tin. Tiếp theo, ta lưu thành từng dòng với ký hiệu mỗi file là "1" và lưu lại thành chuỗi String gọi là chuỗi *cntt*. Tiếp đó, ta sẽ tính tần số xuất hiện của từng đặc trưng đó theo từng file không thuộc công nghệ thông tin của

tập huấn luyện. Sau đó, ta cũng lưu thành dòng với ký hiệu mỗi file là "-1" và lưu lại thành chuỗi String gọi là chuỗi *không cnnt*. Sau cùng, ta ghép chuỗi *cnnt* và chuỗi *không cntt* lại rồi lưu xuống file gọi là file *tfidf_features.txt*. Đây là file chứa các đặc trưng được định dạng theo chuẩn của LibSVM.

- Giai đoạn *SVM_Train*: đầu vào của quá trình này là file *tfidf_features.txt* và đầu ra là file *train_model.txt*. File này sẽ được sử dụng để phân loại tài liệu sau này.

Quá trình *phân loại tài liệu dùng SVM:*

- Các tài liệu ở dạng file HTML và PDF sau khi lưu xuống máy tính sẽ được chuyển sang dạng file. Giai đoạn này cần sử dụng gói htmlparser. Tất cả các file text có được sẽ được chứa trong folder *download_files*.
- Giai đoạn *tách từ và loại bỏ hư từ*: ta sẽ sử dụng API Tokenizer trong giai đoạn này. Đầu tiên, ta load nội dung các file trong folder *download_files* lên máy. Tiếp theo, ta sử dụng tokenizer để tách từ theo ngôn ngữ việt. Sau đó, ta sẽ loại bỏ các hư từ trong các nội dung đó (đã được tách từ). Cuối cùng, các nội dung này (đã được tách từ và loại bỏ hư từ) sẽ được lưu xuống tương ứng với tên các file.
- Giai đoạn *tính TFIDF theo đặc trưng*: ở giai đoạn này, ta sẽ sử dụng đặc trưng có được để tính TFIDF của từng file (là những file đã tách từ và loại bỏ hư từ). Tất cả các kết quả của chúng sẽ được lưu xuống một file text tên *tfidf_download_files.txt*.
- Giai đoạn *phân loại bằng SVM*: trong giai đoạn này, ta sẽ dùng SVM để phân loại văn bản, kết quả của chúng là ta có thể lấy ra được nhưng file có liên quan đến các từ khóa. Đầu vào của giai đoạn *phân loại bằng SVM* là file *tfidf_download_files.txt* và *SVM_Train*. Còn đầu ra của chúng là một file text *result.txt* cho chúng ta biết file nào được chọn, để từ đó chúng ta sẽ lưu lại các file được chọn này (*tài liệu công nghệ thông tin*).

- Rút trích cá thể: đầu tiên ta sẽ sử dụng nhiều bài báo để có thể rút ra được các luật, ví dụ như "thì, mà, là". Sau đó, ta sẽ dùng các luật này vào trong *tài liệu công nghệ*

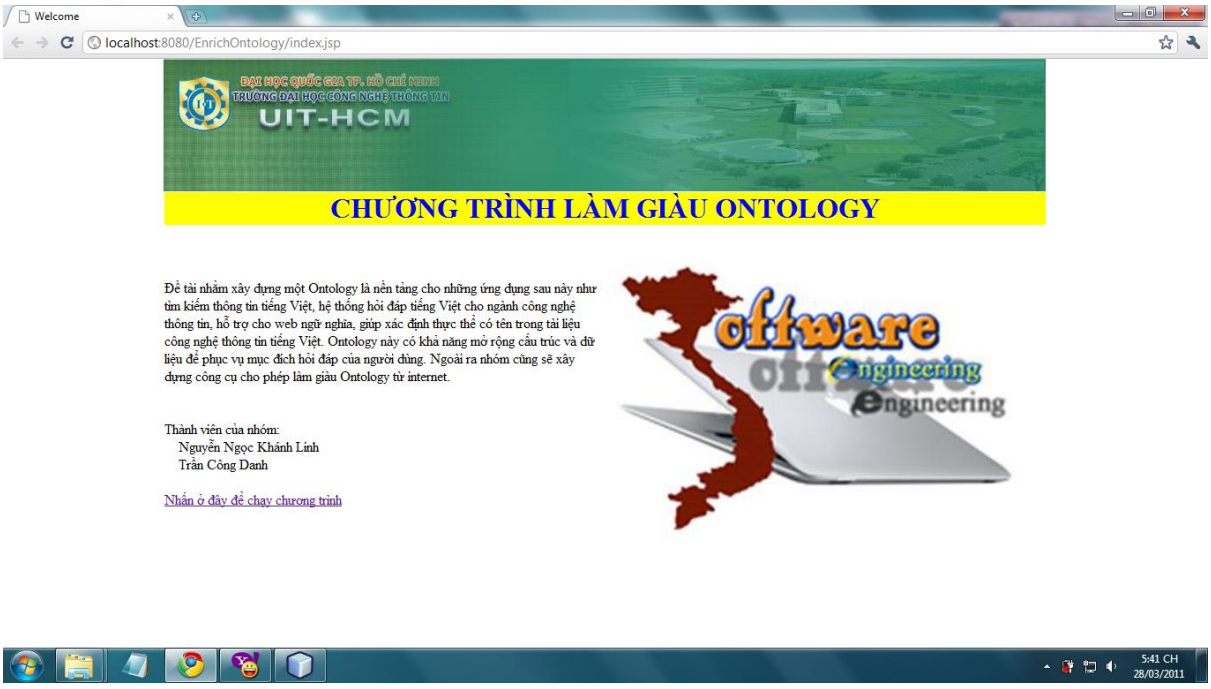
thông tin để rút ra được những định nghĩa và những thuộc tính liên quan đến các từ khóa. Trong quá trình tìm hiểu, ta đã rút ra hai dạng câu định nghĩa. Dạng thứ nhất là: từ trạng thái (nếu có) + ‘, ‘(nếu có) + từ khóa + followwords (thì, mà, là, ...) + định nghĩa. Dạng thứ hai là: định nghĩa + forwardwords (thì, mà, là, ...) + từ khóa. Ta sử dụng hai dạng câu này để tạo ra hai luật rút trích.

- Cập nhật ontology: sau khi rút ra được những định nghĩa và những thuộc tính trên, ta mới cập nhật vào ontology theo từ khóa. Sau đó, ta sẽ hiện kết quả ra màn hình.

4.3. Các bước chạy chương trình

Giao diện chương trình dùng ngôn ngữ là tiếng việt.

Khi chạy chương trình sẽ truy xuất file *index.jsp* trước tiên và hiển thị màn hình giao diện giới thiệu (hình 1). Giao diện này sẽ giới thiệu mục đích làm đề tài và nêu tên các thành viên trong nhóm. Nó có đường link để thực thi quá trình làm giàu ontology.



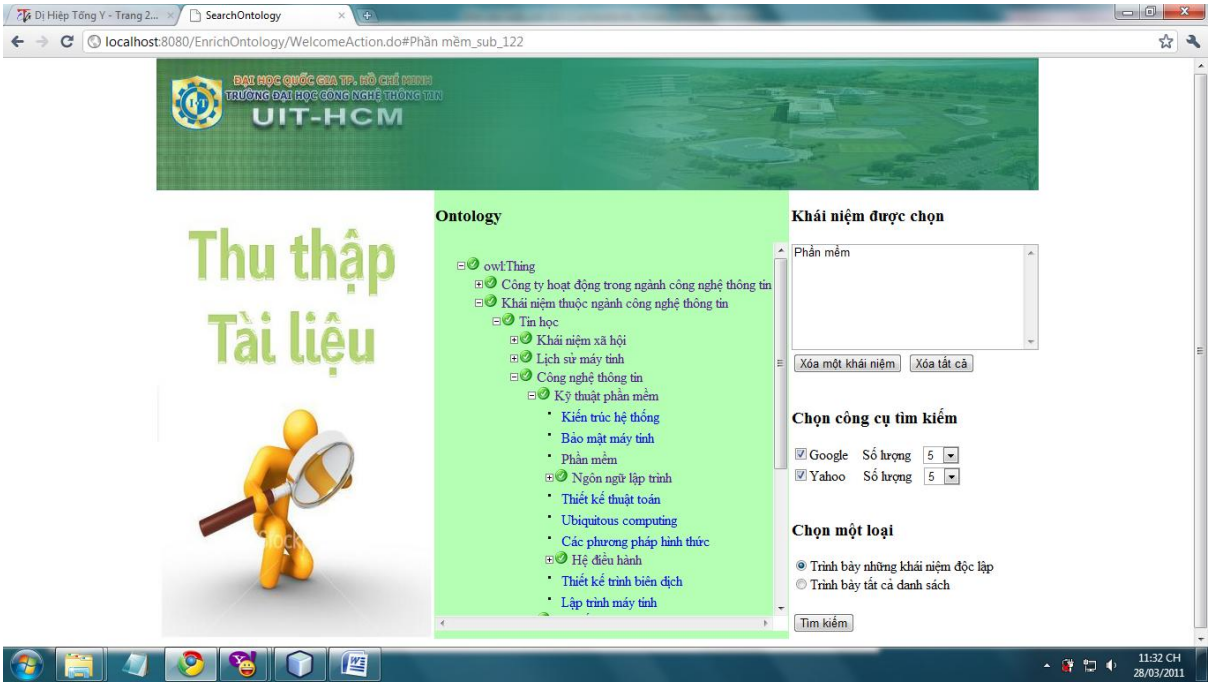
Hình 10: Màn hình giới thiệu

Khi ta nhấp vào link Nhấn vào đây để chạy chương trình, nó sẽ sang giao diện thu thập dữ liệu (hình 2). Giao diện này được chia làm 2 phần chính: phần hiển thị ontology và phần chọn lựa.

a. Trước tiên, ta cần mở ontology ra và nhấp vào những khái niệm mà ta muốn làm giàu. Tên những khái niệm này sẽ hiện ra bên phần *Những khái niệm được chọn*. Ta có thể làm giàu nhiều lớp một lúc, nhưng tốt đa chỉ được bốn lớp. Ở đây, ta chọn khái niệm *Phần mềm*.

b. Bên phần *Chọn công cụ tìm kiếm*, ta cần check vào checkbox *Google* hay *Yahoo* và chọn số lượng link cần tìm ở bên cạnh.

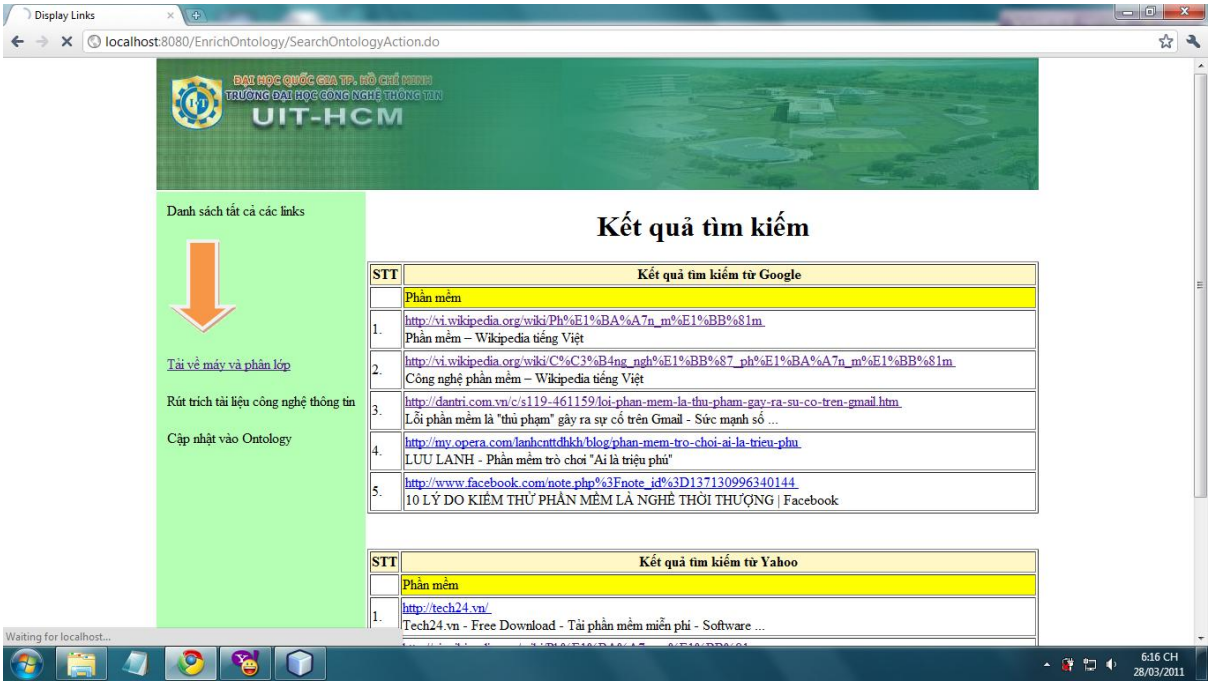
c. Sau khi đã chọn lựa xong, ta nhấn nút *Tìm kiếm*.



Hình 11: Màn hình thu thập tài liệu

Chương trình bây giờ sẽ ở giao diện màn hình kết quả thu thập (hình 3). Giao diện này chia làm hai phần. Giao diện bên phải sẽ hiện ra tất các link (và tiêu đề tương

ứng) mà công cụ tìm kiếm tra được. Phần bên trái sẽ thể hiện các giai đoạn chạy. Có tất cả bốn giai đoạn: Danh sách các link, Tải về máy và phân lớp, Rút trích tài liệu công nghệ thông tin, Cập nhật vào Ontology. Các giai đoạn này tuần tự từ trên xuống dưới và không thể quay ngược lại. Lúc này ta đang ở giai đoạn Thu thập tài liệu, giai đoạn tiếp theo là *Tải về máy và phân lớp*. Muốn đến giai đoạn tiếp theo, ta chỉ việc nhấn vào link *Tải về máy và phân lớp*.



Hình 12: Màn hình kết quả thu thập

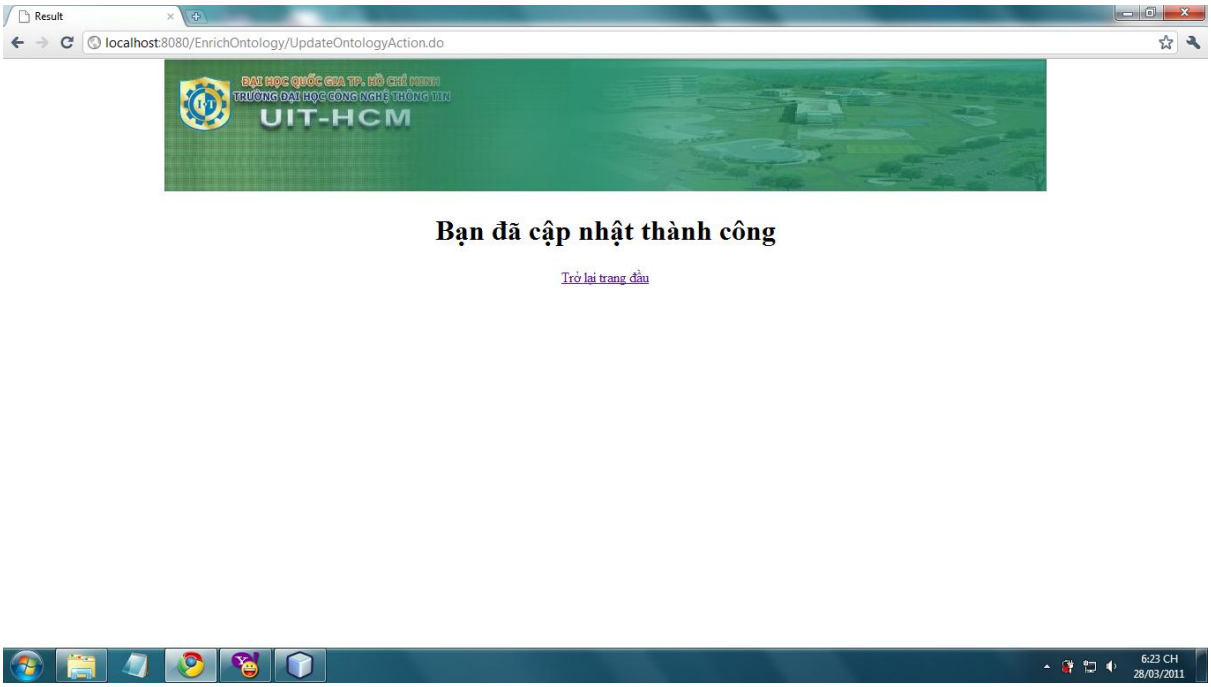
Sau khi nhấp vào link *Tải về máy và phân lớp* chương trình sẽ thực hiện các công đoạn: tải dữ liệu về máy theo các đường link đã có, tách từ, loại bỏ hư từ, phân lớp theo SVM. Khi giai đoạn kết thúc thì chương trình sẽ hiện ra màn hình kết quả phân lớp (hình 4) chứa danh sách các link (và tiêu đề tương ứng) được chọn ở phần bên phải. Còn phần bên trái sẽ hiện thị cho biết ta đang ở giai đoạn *Tải về máy và phân lớp* và giai đoạn tiếp theo là *Rút trích tài liệu công nghệ thông tin*.

Hình 13: Màn hình kết quả phân lớp

Sau khi nhấp vào link *Rút trích tài liệu công nghệ thông tin* sẽ thực hiện công đoạn rút trích ra định nghĩa từ những tài liệu được chọn. Khi xử lý xong thì chương trình sẽ hiện ra danh sách các định nghĩa được rút ra và nguồn của chúng ở phần bên phải. Người dùng có thể check vào các ô checkox tương ứng để loại bỏ những định nghĩa không chính xác cho khái niệm cần làm giàu, các ô không chọn sẽ được dùng để cập nhật ontology. Còn phần bên trái sẽ hiện thị cho biết ta đang ở giai đoạn *Rút trích tài liệu công nghệ thông tin* và giai đoạn tiếp theo là *Cập nhật Ontology*.

Hình 14: Màn hình kết quả rút trích

Sau khi nhấp vào link *Cập nhật Ontology* chương trình sẽ thực hiện công đoạn: cập nhật các định nghĩa và nguồn tương ứng với nó vào ontology. Sau đó, chương trình sẽ hiện lên thông báo cập nhật ontology có thành công hay không (hình 6). Đến đây, ta đã kết thúc toàn bộ quá trình làm giàu ontology. Nếu ta muốn tiếp tục làm giàu một khái niệm nào đó thì ta nhấn link *Trở lại trang đầu*.



Hình 15: Màn hình cập nhật thành công

4.4. Thực nghiệm và đánh giá

Thực nghiệm chương trình:

STT	Tên từ khóa	Số từ khóa	Số link chọn	Thời gian chạy chương trình	Số định nghĩa thu được	Số định nghĩa đúng (không tính trùng nhau)
1	Phần mềm	1	10	1 phút 35 giây	2	1
2	Tin học	1	10	1 phút 17 giây	2	1
3	Tin học	1	25	2 phút 50 giây	6	1
4	Phần mềm	1	30	1 phút 55 giây	12	1
4	Phần mềm – Tin học	2	10	1 phút 48 giây	4	2

5	Phần mềm – Tin học	2	20	5 phút 14 giây	13	2
6	Công nghệ thông tin	1	10	30 giây	4	2
7	Hệ thống thông tin	1	10	55 giây	0	0
8	Hợp đồng	1	15	1phút 52 giây	2	1
9	Bộ nhớ ảo – Hệ điều hành	2	60	10 phủ 16 giây	0	0
10	Phần mềm – Tin học – Hệ điều hành	3	30	3 phút 39 giây	11	4
11	Phần mềm – Tin học – Hệ điều hành – Ngôn ngữ lập trình	4	20	6 phút 38 giây	11	4
12	Lập trình	1	50	12 phút 5 giây	1	0

Đánh giá chương trình:

Chương trình có thể làm giàu những khái niệm trong Ontology chuyên ngành công nghệ thông tin tiếng việt.

Chưa thực hiện phần làm giàu cho các cá thể khác trừ các khái niệm trong ngành công nghệ thông tin, chưa làm giàu quan hệ trong Ontology.

Kết quả tìm kiếm từ các công cụ tìm kiếm, phân lớp dùng LibSVM khá chính xác. Kết quả rút trích từ tài liệu đã phân loại theo các mẫu định sẵn cho kết quả chấp nhận được với sai số 25%.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

Sau khi thực hiện đề tài này, chúng em đã thu được những kết quả sau:

- Về kiến thức: Chúng em đã nắm được khái niệm, công cụ, phương pháp xây dựng ontology, các ngôn ngữ biểu diễn ontology. Chúng em đã có kinh nghiệm sử dụng Google và Yahoo API để tìm kiếm tài liệu, Protégé API để lập trình thao tác và truy xuất ontology. Ngoài ra chúng em còn có thêm kiến thức về phương pháp phân lớp SVM.
- Về công cụ: Chúng em có kinh nghiệm sử dụng công cụ Protégé để xây dựng ontology bằng tay, sử dụng công cụ vnTokenizer để tách từ tiếng Việt và công cụ LibSVM để phân lớp tài liệu. Thực hiện đề tài này tạo cơ hội cho chúng em thực hành lập trình trên Netbean sử dụng Strut Framework.
- Về phương pháp: Chúng em biết được các phương pháp để xây dựng và làm giàu ontology hiện nay và cài đặt theo một số phương pháp.

Đề tài là bước khởi đầu cho một ontology hoàn thiện tiếng Việt về lĩnh vực công nghệ thông tin. Lượng dữ liệu hiện nay có thể dùng để xác định những thực thể có tên, tuy nhiên cần nhập thêm nhiều dữ liệu để có thể sử dụng cho các yêu cầu khác đã đặt ra. Công cụ làm giàu ontology đã có thể sử dụng để làm giàu các cá thể khái niệm trong lĩnh vực công nghệ thông tin, tuy nhiên vẫn cần được cải thiện để có thể làm giàu thêm lớp và quan hệ.

5.2. Hướng phát triển

Chương trình còn một số điểm cần khắc phục và phát triển như sau:

- Xử lý trùng lắp khi làm giàu ontology
- Làm giàu lớp và quan hệ trong ontology

- Cải thiện quá trình rút trích các thuộc tính của cá thể
- Bổ sung dữ liệu cho ontology.
- Mở rộng ontology, thêm các lớp thuộc Chương trình đào tạo để phục phụ cho ứng dụng tư vấn chương trình đào tạo của các trường.

Tài liệu tham khảo

❖ **Tiếng Việt:**

[1] Lương Quý Tịnh Hà, *Xây dựng công cụ tìm kiếm tài liệu học tập bằng các truy vấn ngôn ngữ tự nhiên trên kho học liệu mở tiếng Việt, Luận văn thạc sĩ, khoa Khoa học máy tính, trường Đại học Công nghệ Thông tin, Tp. HCM, 2009.*

[3] Lê Thành Nhân, Võ Trung Hùng, Cao Xuân Tuấn, Hoàng Thị Mỹ Lệ, *MATHIS – Hệ thống hỗ trợ tạo chú thích và tìm kiếm tài liệu khoa học, Tạp chí khoa học và công nghệ, Đại học Đà Nẵng - Số 4(39).2010*

[4] Trần Đình Khang, Vũ Tuyết Trinh, Đỗ Đức Thành, Đỗ Thị Ngọc Quỳnh, *Một phương pháp tìm kiếm dựa trên Ontology phục vụ quản lý thông tin khoa học công nghệ*, Bộ môn Hệ thống Thông tin, Trường Đại Học Bách Khoa Hà Nội, 2007.

[5] Phạm Thị Mỹ Phượng, Từ Thị Ngọc Thanh, *Tìm kiếm ngữ nghĩa ứng dụng trên lĩnh vực eDoc*, Đại học Khoa học tự nhiên, 2005.

[6] *Tài liệu hướng dẫn phiên bản mã nguồn mở OVL – Open 1.0.*

[22] Nhóm nghiên cứu của thầy Đỗ Phúc, *Phát triển một Hệ thống S.E Hỗ trợ Tìm kiếm Thông tin, thuộc lãnh vực CNTT trên Internet qua từ khóa bằng tiếng Việt,* Đại học Công nghệ thông tin, khoa Công nghệ thông tin, Đại học Khoa học tự nhiên Tp. HCM, 2010.

[27] Nguyễn Linh Giang, Nguyễn Mạnh Hiển, *Phân loại văn bản tiếng Việt với bộ phân loại vectơ hỗ trợ SVM,* khoa Công nghệ thông tin, Đại học Bách Khoa Hà Nội, khoa Công nghệ thông tin, Đại học Thủy lợi, 2009.

❖ **Tiếng Anh:**

[2] Natalya F. Noy and Deborah L. McGuinness, *Ontology Development 101: A Guide to Creating Your First Ontology, Stanford University, Stanford, CA, 9430, 2001.*

[20] Thomas R.Gruber, *Toward Principles for the Design of Ontologies Used for Knowledge Sharing,* Stanford Knowledge Systems Laboratory, 701 Welch Road, Building C Palo Alto, CA 94304, 1993.

[42] Paul Buitelaar, Philipp Cimiano and Bernardo Magnini, *Ontology Learning from Text: An Overview*, DFKI, Language Technology lab, AIFB, University of Karlsruhe, ITC-irst, Centro per la Ricerca Scientifica e Tecnologica, 2005.

[37] Matthew Horridge, *A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools Edition 1.2*, The University Of Manchester, 2009.

[9] Bijan Parsia and Evren Sirin, *Pellet: An OWL DL Reasoner,* MINDSWAP Research Group, University of Maryland, College Park, MD, 2004.

[43] Hiep Phuc Luong, Susan Gauch, Qiang Wang, *Ontology-based Focused Crawling,* CSCE Department, University of Arkansas, 2009.

[34] Boanerges Aleman-Meza, Farshad Hakimpour, I. Budak Arpinar. *SwetoDblp Ontology of Computer Science Publications,* LSDIS Lab, Computer Science Department, University of Georgia, Athens, GA, 2007.

[36] Grigoris Antoniou and Frank van Harmelen, *A Semantic web Primer*, The MIT Press Cambridge, Massachusett, London, England, p.31-33, 2004.

[45] Eneko Agirre, Olatz Ansa, Eduard Hovy and David Martinez, *Enriching very large ontologies using WWW,* IxA NLP group, University of the Basque Country, 649 pk, 20.080 Donostia, Spain, USC Informat on Sciences Institute 4676 Admiralty Way, Marina del Rey, CA 90292-6695, USA, 2000.

[49] Hiep Phuc Luong, Susan Gauch, Mirco Speretta, *Enriching concept descriptions in an Amphibian Ontology with vocabulary extracted from Wordnet*, Deparment of Computer Science & Computer Engineering, University of Arkansas, 2009.

[51] Hiep Phuc Luong, Susan Gauch, Qiang Wang, *Ontology learning through focused crawling and information extraction*, CSCE Department, University of Arkansas, 2009.

[25] B. E. Boser, I. M. Guyon, and V. N. Vapnik, *A training algorithm for optimal margin classifiers*, In D. Haussler, editor, 5th Annual ACM Workshop on COLT, pages 144-152. Pittsburgh, PA, 1992.

[23] L. H. Phuong, N. T.M. Huyen, R. Azim, H. T. Vinh, *A hybrid approach to word segmentation of Vietnamese texts*, Proceedings of the 2nd International Conference on Language and Automata Theory and Applications, LATA 2008, Springer LNCS 5196, Tarragona, Spain, 2008.

[46] Fellbaum, C, *Wordnet: An Electronic Lexical Database*. Cambridge: MIT Press, 1998.

[7] Nguyen Phi Minh Tri, Nguyen Tuan Dang, *Building a Universal Ontology for Vietnamese Language*, Faculty of Computer Science, University of Information Technology, 2010.

[48] Miller, G., C. Leacock, R. Tengi, and T. Bunker, *A Semantic Concordance*. Proc. Of ARPA Workshop on Human Language Technology, 1993.

Website tham khảo:

[10] http://www.acm.org/

[11] http://what.csc.villanova.edu/twiki/bin/view/Main/TheComputingOntology

[12] http://knoesis.wright.edu/library/ontologies/swetodblp/

[13]http://www.acm.org/education/curricula-recommendations [2001 -- 2005 curriculum recommendations]

[14] http://ngonngu.net/

[15] http://ngonnguhoc.org

[16] http://dblp.uni-trier.de/

[17] http://xmlns.com/foaf/spec/

[18] http://dublincore.org/

[19] http://www.w3.org/TR/rdf-sparql-query/

[21] http://protege.stanford.edu/

[24] http://www.loria.fr/~lehong/tools/vnTokenizer.php

[26] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[28] http://vlsp.vietlp.org:8080/demo/?page=resources

[29] http://www.xulyngonngu.com/sharing/?p=145

[30] http://mic.gov.vn/Trang/default.aspx

[33] http://www.cs.wisc.edu/dbworld/

[46] http://www.jfsowa.com/ontology/index.htm

[35] http://diendankienthuc.net/diendan/ngon-ngu-tieng-viet/7105-cac-dau-cau-trong-tieng-viet.html

[40] http://www.w3schools.com/RDF/rdf_example.asp

[41] http://www.w3schools.com/RDF/rdf_schema.asp

[38] http://www.ontotext.com/inference/rdfs_rules_owl.html

[39] http://www.w3.org/TR/owl-guide/

[47] www.altavista.com

[50] http://wordnet.princeton.edu/

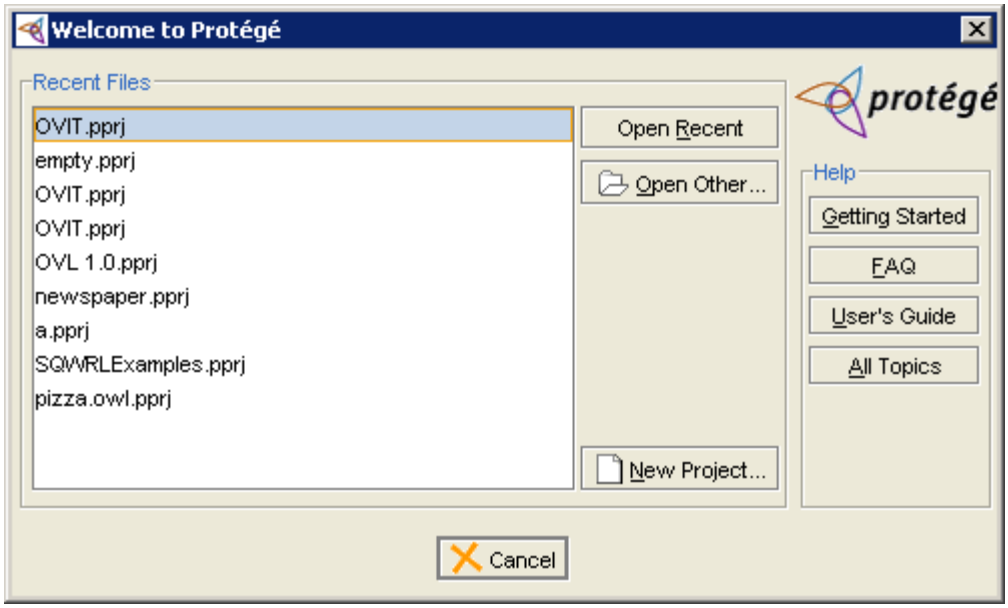
Phụ lục A: Hướng dẫn sử dụng Protégé

Trong phần hướng dẫn sử dụng này sử dụng chương trình Protégé 3.4.4 với giao diện Protégé-OWL. Chúng ta sẽ cùng tìm hiểu cách để:

- Tạo và mở một ontology
- Lưu một ontology
- Tạo lớp và ràng buộc
- Tạo các thuộc tính và quan hệ
- Tạo cá thể

1. Tạo và mở một ontology:

a. Hiểu khái niệm Project trong Protégé: Khi vừa khởi động Protégé lên chúng ta sẽ thấy cửa sổ "Welcome to Protégé" hiện ra đầu tiên như hình sau:



Hình: Giao diện “Welcome to Protégé”

Khi dùng Protégé để tạo và chỉnh sửa ontology chúng ta sẽ tạo ra ít nhất là 2 file:

- File project (có đuôi là .pprj): Lưu trữ thông tin liên quan đến việc tùy biến giao diện hoặc tùy chọn của trình soạn thảo mà bạn cài đặt. Nếu không có file này chúng ta vẫn có thể tạo một file project khác cho một ontology từ file nguồn.
- File nguồn (có đuôi là .owl, .rdf hoặc .rdfs): đây là file chứa dữ liệu thật sự của ontology, nó chứa các lớp, cá thể và thuộc tính được định nghĩa.

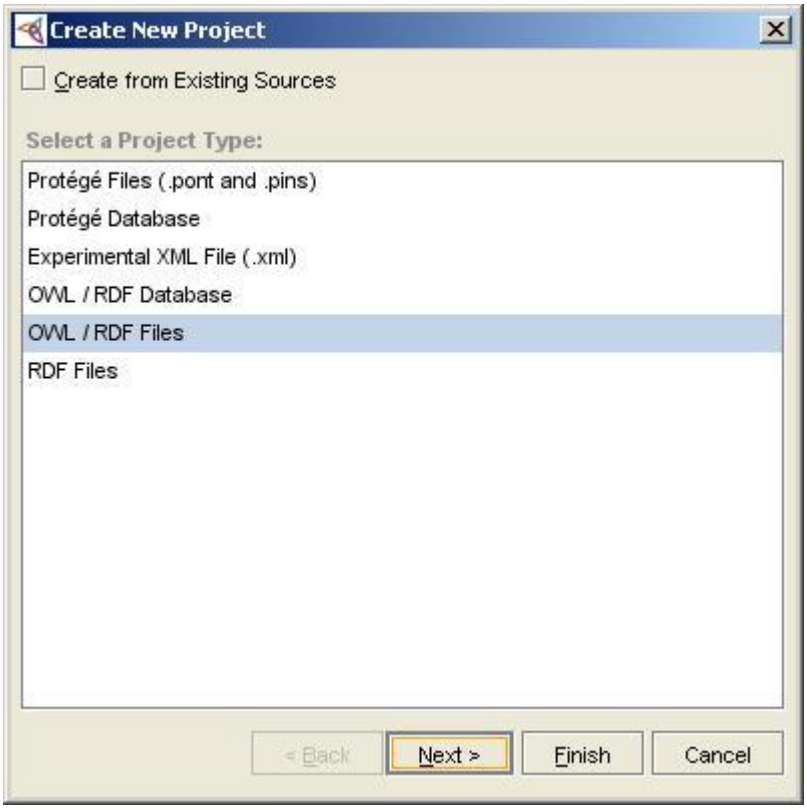
b. Mở một ontology có sẵn: có thể mở từ file project hoặc file nguồn.

Nếu có sẵn trên màn hình welcome thì nhấp đúp lên file ontology, hoặc chọn ontology muốn mở rồi bấm nút **Open Recent**.

Nếu không có sẵn trên màn hình welcome thì chọn nút **Open Other** và chọn ontology muốn mở.

c. Tạo một ontology mới:

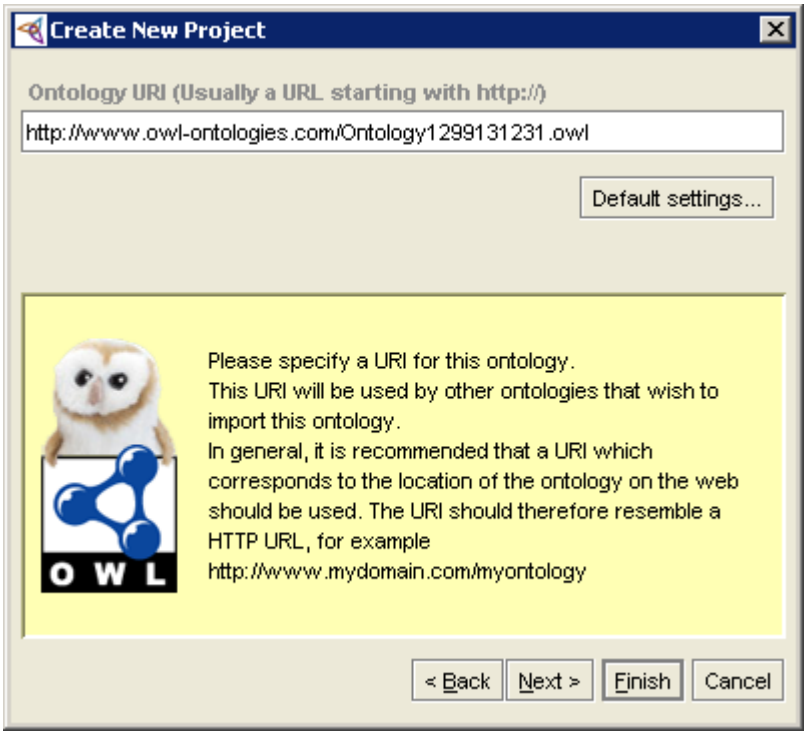
Từ giao diện welcome, chọn **New Project** hiển thị hộp thoại tạo project mới như hình sau:



Hình: Hộp thoại tạo project mới

Chọn **OWL/RDF Files** và bấm **Next >**

Màn hình xuất hiện cho bạn xác nhận một URI cho ontology của mình. Thông thường, URI sẽ biểu diễn nơi mà chúng ta công bố ontology, tuy nhiên cũng không bắt buộc phải tuân theo. Việc thiết lập một URI duy nhất cho ontology sẽ giúp phòng những vấn đề sau này nếu ta nhập thêm những ontology khác.



Hình : Hộp thoại đặt URI cho ontology mới

Bấm **Next** >. Một hộp thoại xuất hiện cho phép chọn ngôn ngữ muốn dùng

Hình: Hộp thoại chọn ngôn ngữ xây dựng ontology

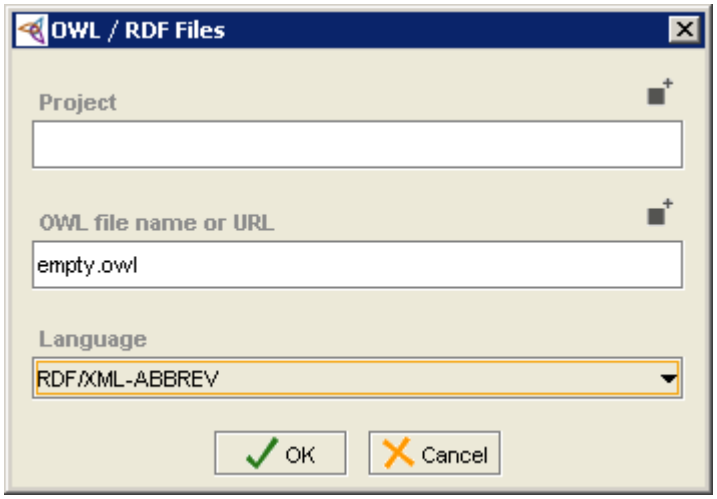
Bấm Finish để tạo ontology mới. Một hộp thoại xuất hiện để chúng ta có thể chọn cách hiển thị. Logic View là giao diện phù hợp hơn cho người dùng đã quen thuộc vì nó không được trực quan lắm, còn với người mới bắt đầu thì nên chọn Properties View vì nó có giao diện đơn giản hơn.

Hình: hộp thoại chọn cách hiển thị

Bấn Finish và một project mới được tạo sẵn sang để bạn có thể nhập và chỉnh sửa một ontology.

Nếu lúc mở Protégé lên mà hộp thoại welcome không xuất hiện thì ta chọn File | New Project

2. Lưu một ontology: bấm bút **Save** trên trình soạn thảo hoặc vào **File** chọn **Save Project.** Một hộp thoại xuất hiện để ta nhập tên ontology và tên project vào nếu như ta một project mới.

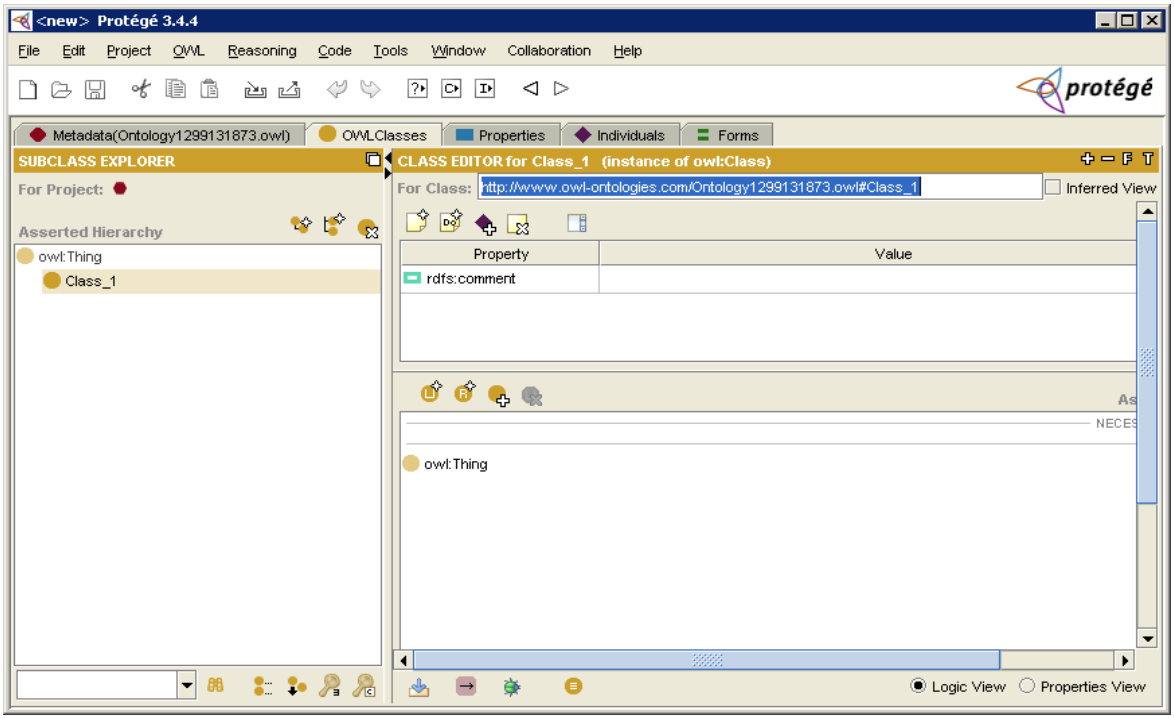


Hình : hộp thoại lưu ontology.

Thông thường tên project giống với tên của file OWL.

3. Tạo lớp:

Tại giao diện của Protégé ta chọn tab **OWL Classes,** mặc định mỗi ontology mới tạo có lớp cha là Thing. Để tạo một lớp mới ta chọn nút **Create subclass** sẽ tạo ra lớp con tên Class_1 như hình, ta có thể đổi tên ở textbox bên phải



Hình: tạo lớp mới trong ontology

Để tạo lớp con của lớp Class_1 ta chọn nó rồi làm tương tự, hoặc nhấp chuột phải lên nó chọn **Create subclass.**

Để tạo lớp ngang hàng với một lớp ta chọn nó rồi chọn nút **Create Sibling Class** , hoặc nhấp chuột phải chọn **Create Sibling Class.**

Để tạo ra một cấu trúc cây của gồn nhiều lớp có lớp cha là Class_1 ta nhấp phải vào nó chọn **Create subclasses**, sau đó một hộp thoại sẽ xuất hiện để ta nhập vào cấu trúc lớp với mỗi lớp là 1 dòng và lớp con thụt vào so với lớp cha. Ví dụ ta nhập:

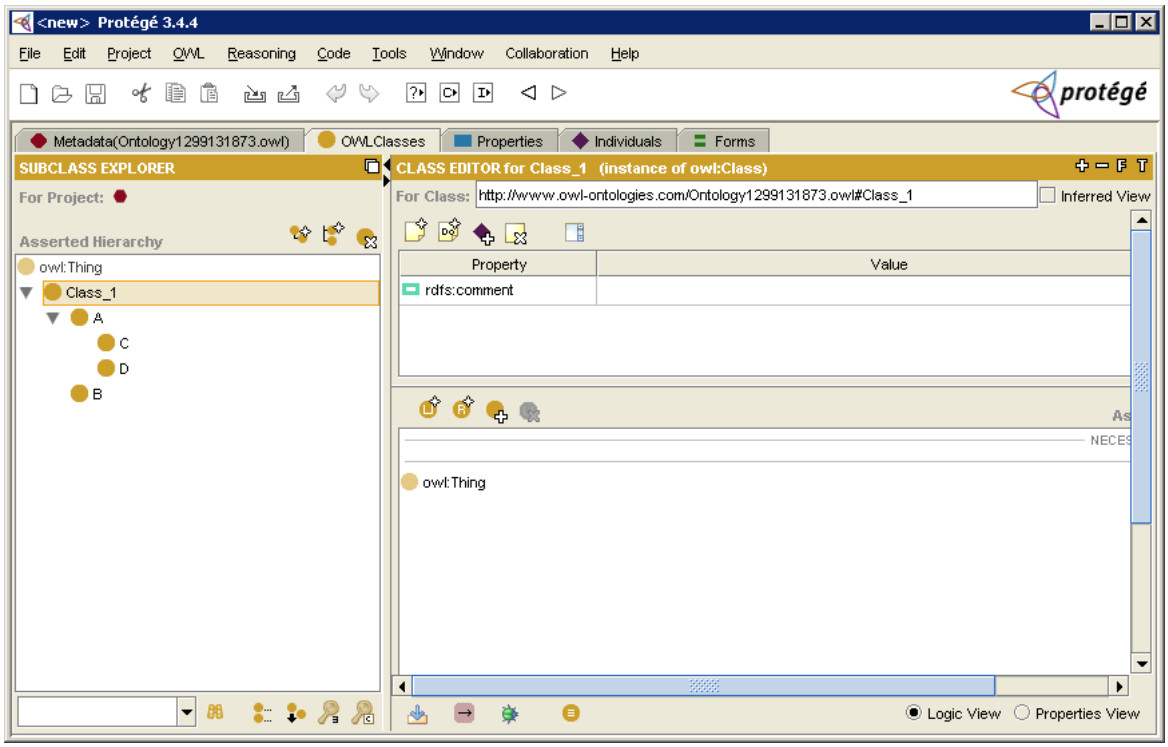
A

C

D

B

Ta sẽ tạo ra cấu trúc cây trong Class_1 như sau:



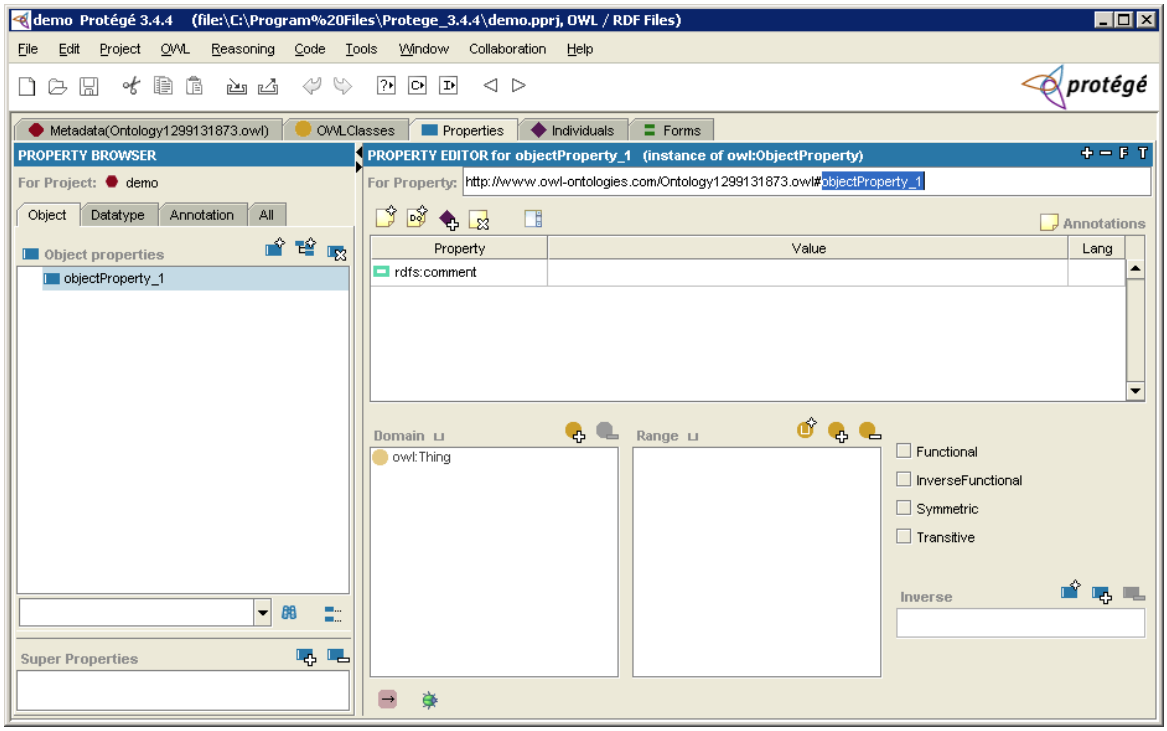
Hình: Tạo nhiều lớp trong ontology

Xóa một lớp ta chọn **Delete Class** . Nếu xóa lớp cha thì tất cả lớp con của nó đều bị xóa.

Ta có thể tạo các ràng buộc cho lớp dùng các nút có sẵn như **Create new expression** , **Create restriction** và **Add Named Class** để thêm lớp cha cho lớp đang chọn. Hoặc xóa đi các ràng buộc đã thêm dùng **Delete selected row** .

4. Tạo các thuộc tính và quan hệ:

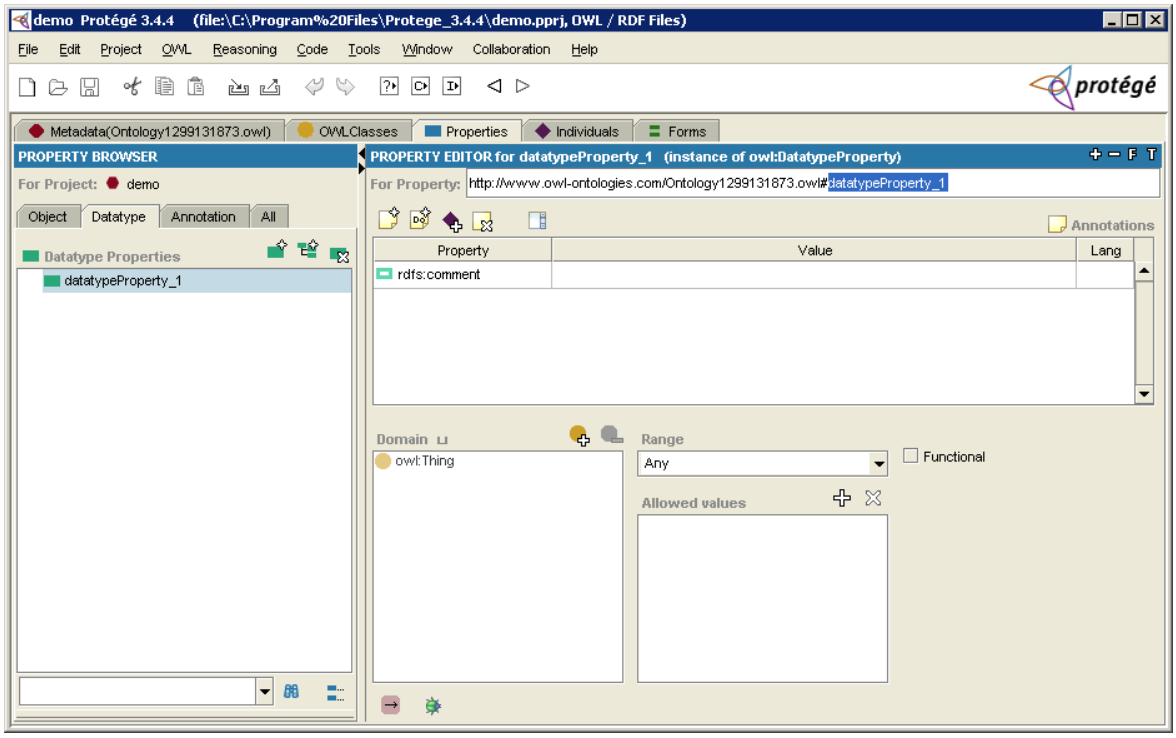
Tại giao diện Protégé ta chọn tab **Properties**. Trong đó ta chọn tab **Object** để thêm và chỉnh sửa các quan hệ trong ontology, chọn tab **Datatype** để thêm và chỉnh sửa các thuộc tính trong ontology.



Hình : Tạo quan hệ mới

Để tạo quan hệ mới ta cũng chọn nút **Create object property**  , và đổi tên quan hệ ở textbox bên tay phải như tạo lớp. Đối với quan hệ ta chú ý đến **Domain** và **Range** có thể được chỉnh sửa và thêm ở bên phải. Và một số tính chất của quan hệ như: Functional, InverseFunctional, Symmetric, Transitive, ngoài ra ta có thể thêm một vào quan hệ nghịch đảo của một quan hệ bằng cách thêm tại textbox Inverse.

Để tạo thuộc tính mới ta chọn tab Datatype và cũng chọn nút **Create Datatype property**  theo hình ở dưới

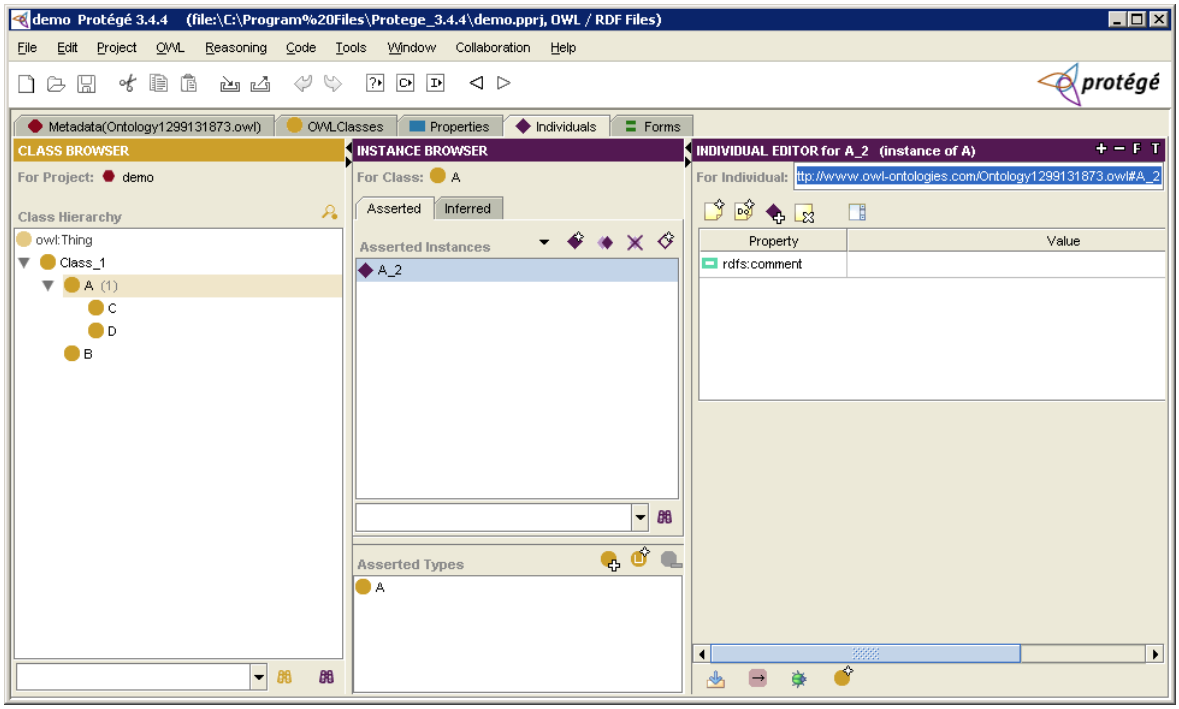


Hình : Tạo thuộc tính mới.

Đối với thuộc tính ta cũng có thể thay đổi **Domain** và **Range** cho nó bên phải màn hình. **Domain** sẽ xác định nó thuộc lớp nào, và **Range** sẽ xác định loại dữ liệu của nó. Ngoài ra, ta cũng có thể thêm vào một vài ràng buộc đơn giản cho nó như giới hạn một số giá trị được cho phép cho thuộc tính này.

5. Tạo cá thể

Để tạo cá thể ta chọn tab **Individuals,** chọn lớp mình muốn tạo cá thể rồi chọn nút **Create instance** ta sẽ tạo được cá thể như hình dưới. Các chức năng khác tương tự như trên.



Hình: Tạo cá thể mới

Phụ lục B: Danh sách các hư từ

bao giờ	chúng mình	nhất định	thật ra
á	chúng nó	nhất loạt	thật vậy
à	chung qui	nhất luật	thẩy
ạ	chung quy	nhất mực	thế
á à	chung quy lại	nhất nhất	thế à
a ha	chúng ta	nhất quyết	thế đấy
a lô	chúng tao	nhất sinh	thế là
à ơi	chúng tôi	nhất tâm	thế mà
ạ ơi	có	nhất tề	thế nào
ai	cô	nhất thiết	thế nên
ái	cơ	nhau	thế nhưng
ai ai	cô ấy	nhé	thế ra
ái chà	có chăng là	nhỉ	thế thì
ái dà	cơ chừng	nhiên hậu	thếch
ai nấy	cỡ chừng	nhiệt liệt	theo
alô	có dễ	nhiều	thì
amen	cơ hồ	nhỡ ra	thì ra
áng	cổ lai	nhón nhén	thi thoảng
anh	cơ mà	như	thình lình
anh ta	cô mình	như chơi	thỉnh thoảng

ào	có vẻ	như không	thoắt
ắt	cóc khô	như quả	thoạt
ắt hẳn	coi bộ	như thể	thoạt nhiên
ắt là	coi mòi	như tuồng	thốc
âu là	con	như vậy	thộc
ầu ơ	còn	nhưng	thốc tháo
ấy	cơn	những	thôi
ba	công nhiên	những	thôi thì
bài	cứ	những ai	thỏm
bản	cu cậu	nhưng mà	thốt
bạn	cụ thể	nhung nhăng	thọt
bằng	cứ việc	những như	thốt nhiên
bằng ấy	của	nhược bằng	thửa
bằng không	của bạn	nó	thuần
bằng nấy	của mày	nọ	thục mạng
bao giờ	của tôi	nớ	thực ra
bao gồm	cực kì	nóc	thực vậy
bao lâu	cực kỳ	nổi	thúng thắng
bao nả	cực lực	nữa	thuộc
bao nhiêu	cùng	nức nở	thương ôi
bập bà bập bõm	cũng	ồ	tiện thể

bập bõm	cùng cực	ơ	tiếp đến
bất	cùng nhau	ớ	tiếp đó
bất chấp	cũng như	ờ	tiếp theo
bất chợt	cũng vậy	ở	tiếp tục
bất cứ	cũng vậy thôi	ô hay	tít mù
bắt đầu từ	cùng với	ơ hay	tớ
bất đồ	cuộc	ô hô	tỏ ra
bất giác	cuối	ô kê	tò te
bất kể	cuối cùng	ô kìa	tỏ vẻ
bất kì	cuốn	ơ kìa	toà
bất kỳ	dạ	oái	tốc tả
bất luận	đã	oai oái	toé khói
bất nhược	đại để	ôi	toẹt
bất quá	đại loại	ối	tôi
bất thình lình	đại nhân	ơi	tối ư
bất tử	đại phàm	ôi chao	tông tốc
bấy	dần dà	ối dào	tọt
bảy	dần dần	ối giời	tột
bây bẩy	đang	ối giời ơi	trái
bay biến	đáng lẽ	ôi thôi	tràn cung mây
bấy chầy	đáng lí	ối trời	trên

bây chừ	đáng lý	ôkê	trển
bấy chừ	đằng sau	ổng	trệt
bây giờ	đằng trước	phải	trếu tráo
bấy giờ	đành đạch	phải chăng	trệu trạo
bấy lâu	đánh đùng	phải chi	trời đất ơi
bấy lâu nay	dào	phăn phắt	trời ơi
bấy nay	đáo để	phắt	trong
bây nhiêu	dẫu	phè	trỏng
bấy nhiêu	đâu	phỉ phui	trong khi
bèn	dầu sao	pho	trong lúc
bên	dẫu sao	phóc	trừ khi
bển	đây	phốc	trừ phi
bên dưới	để	phỏng	trước
bên phải	dễ sợ	phỏng như	trước đây
bên trái	dễ thường	phót	trước đó
bên trên	đều	phương chi	trước khi
béng	do	phụt	trước kia
bệt	đó	phứt	trước lúc
bị	dở chừng	quá	trước nay
biết bao	do đó	quả	trước tiên
biết bao nhiêu	do vậy	quá chừng	từ

biết chừng nào	do vì	quá độ	tự
biết đâu	đồng thời	quá đỗi	tù tì
biết đâu chừng	dù	quả đúng	tự vì
biết đâu đấy	dữ	quả là	tuần tự
biết mấy	đủ	quá lắm	tức
bớ	dù cho	qua quít	tức khắc
bộ	dù là	qua quýt	tức là
bỏ mẹ	dù rằng	quá sá	tức thì
bởi	dù thế	quả tang	tức tốc
bởi nhưng	được	quả thật	từng
bội phần	dưới	quá thể	tuốt luốt
bởi thế	duy	quả tình	tuốt tuồn tuột
bởi vậy	gì	quá trời	tuốt tuột
bởi vì	giữa	quá ư	tựu trung
bốn	gồm	quả vậy	tuy
bông	hai	quá xá	tuy là
bỗng	hầu hết	quý hồ	tuy nhiên
bỗng chốc	Hay	quyển	tuy rằng
bỗng đâu	hãy	quyết	tuy thế
bỗng dưng	hiện nay	quyết nhiên	tuy vậy
bỗng không	họ	ra	tuyệt nhiên

bỗng nhiên	hoặc	ra phết	ư
bức	hoàn toàn	ra trò	ừ
cả	hỡi	răng	ử
cả thảy	hồi nãy	rằng	ứ hự
cả thể	hơn	rằng là	ứ ừ
các	ít	ráo	ủa
cái	kế tiếp	ráo trọi	úi
cái gì	khi	rất	úi chà
căn	khoảng	rất chi là	úi dào
cần	khoảng chừng	rất đỗi	và
căn cắt	không	rất mực	vả chăng
càng	là	rày	vả lại
cật lực	lại	rén	vài
cật sức	làm	ren rén	vẫn
cây	lẫn	rích	vẫn là
cha	lên	riêng	vạn nhất
cha chả	liên tiếp	riệt	vân vân
chắc	liên tục	riu ríu	vâng
chậc	lúc	rồi	văng tê
chắc hẳn	lúc ấy	rón rén	vào lúc
chầm chập	lúc trước	rốt cục	vậy

chăn chắn	luôn	rốt cuộc	vậy là
chăng	luôn luôn	rứa	vậy mà
chẳng	mà	rút cục	vậy nên
chẳng lẽ	mặc dù	sa sả	vậy thế
chẳng những	mặc kệ	sạch	vậy thì
chẳng nữa	mãi	sẵn sàng	vậy thôi
chẳng phải	mãi mãi	sao	về
chành chạnh	mặt khác	sắp	về mặt
chao ôi	mày	sất	về phía
chết nỗi	mầy	sau	veo
chết thật	mi	sáu	vèo
chết tiệt	mỗi	sau chót	veo veo
chỉ	mọi	sau cùng	vì
chị	một	sau cuối	ví bằng
chí chết	mười	sau đó	vì chưng
chỉ do	năm	sẽ	ví dù
chỉ là	này	sì	ví dụ
chỉ tại	nãy	số	ví phỏng
chỉ vì	nấy	sở dĩ	vị tất
chiếc	nè	số là	vì thế
chín	nên	song le	ví thử

chỉn	nền	sốt sột	vì vậy
chính	nên chi	sự	vở
chính anh	nếu	suýt	vô hình trung
chính chị	nếu như	tà tà	vô kể
chính là	ngăn ngắt	tại	vô luận
chính thị	ngay	tại vì	vô vàn
chính tôi	ngay cả	tám	với
cho	ngày càng	tấm	với lại
chớ	ngay khi	tăm tắp	vốn dĩ
chớ chi	ngay lập tức	tấn	vừa
cho đến	ngay lúc	tanh	vừa mới
cho đến khi	ngày ngày	tao	vung tán tàn
cho là	ngay từ	tắp	vung tàn tán
cho nên	ngay tức khắc	tắp lự	vung thiên địa
cho rằng	ngày xưa	tất cả	vụt
cho tới	ngày xửa	tất tần tật	xa xả
cho tới khi	nghe chừng	tất tật	xăm xăm
choa	nghe đâu	tất thảy	xăm xắm
chốc chốc	nghen	tênh	xăm xúi
chợt	nghĩa là	thà	xềnh xệch
chú	nghiễm nhiên	tha hồ	xệp

chứ	nghỉm	thà là	xiết bao
chu cha	ngõ hầu	thà rằng	xoẳn
chứ lị	ngộ nhỡ	thái quá	xoành xoạch
chú mày	ngoài	thậm	xoét
chú mình	ngoải	thậm chí	xoẹt
chưa	ngôi	than ôi	xon xón
chui cha	ngọn	thanh	xuất kì bất ý
chủn	ngọt	thành ra	xuất kỳ bất ý
chùn chùn	ngươi	thành thử	xuể
chùn chũn	nhân dịp	thảo hèn	xuống
chúng	nhân tiện	thảo nào	ý
chung cục	nhất	thật là	ý chừng
chúng mày	nhất đán	thật lực	ý da