BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

LÂM THỊ PHƯƠNG THẢO

XÂY DỰNG ONTOLOGY TỪ KHO NGỮ LIỆU DẠNG VĂN BẢN

LUẬN VĂN THẠC SĨ

Chuyên ngành : Công nghệ Thông tin

Mã số ngành: 60480201

TP. HỒ CHÍ MINH, tháng 4 năm 2015

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

LÂM THỊ PHƯƠNG THẢO

XÂY DỰNG ONTOLOGY TỪ KHO NGỮ LIỆU DẠNG VĂN BẢN

LUẬN VĂN THẠC SĨ

Chuyên ngành : Công nghệ Thông tin

Mã số ngành: 60480201

CÁN BỘ HƯỚNG DẪN KHOA HỌC: **TS. NGUYỄN CHÁNH THÀNH**

TS. LÊ MẠNH HẢI

TP. HỒ CHÍ MINH, tháng 4 năm 2015

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

Cán bộ hướng dẫn khoa học : Tiến sĩ Nguyễn Chánh Thành - Tiến sĩ Lê Mạnh Hải

Luận văn Thạc sĩ được bảo vệ tại Trường Đại học Công nghệ TP. HCM ngày … tháng … năm …

Thành phần Hội đồng đánh giá Luận văn Thạc sĩ gồm:
(Ghi rõ họ, tên, học hàm, học vị của Hội đồng chấm bảo vệ Luận văn Thạc sĩ)

TT	**Họ và tên**	**Chức danh Hội đồng**
1		Chủ tịch
2		Phản biện 1
3		Phản biện 2
4		Ủy viên
5		Ủy viên, Thư ký

nận của Chủ tịch Hội đồng đánh giá Luận sau khi Luận văn đã được sửa chữa (nếu có).

Chủ tịch Hội đồng đánh giá LV

TRƯỜNG ĐH CÔNG NGHỆ TP. HCM
PHÒNG QLKH – ĐTSĐH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

TP. HCM, ngày..… tháng….. năm 20..…

NHIỆM VỤ LUẬN VĂN THẠC SĨ

Họ tên học viên: Lâm Thị Phương Thảo Giới tính: Nữ

Ngày, tháng, năm sinh: 19/07/1966 Nơi sinh: Qui Nhơn, Bình Định

Chuyên ngành: Công nghệ Thông tin MSHV: 1341860023

I- Tên đề tài:

Tìm hiểu phương pháp xây dựng ontology bán tự động từ kho ngữ liệu dạng văn bản

II- Nhiệm vụ và nội dung:

- Khảo sát các phương pháp xây dựng ontology từ kho ngữ liệu dạng văn bản.
- Đề xuất (hoặc cải tiến) một phương pháp xây dựng ontology từ kho ngữ liệu dạng văn bản trên cơ sở kết hợp xử lý ngôn ngữ tự nhiên.
- Tiến hành thực nghiệm, đánh giá và hiệu chỉnh phương pháp.

III- Ngày giao nhiệm vụ: 19/08/2014

IV- Ngày hoàn thành nhiệm vụ: 10/03/2015

V- Cán bộ hướng dẫn:

Tiến sĩ Nguyễn Chánh Thành, tiến sĩ Lê Mạnh Hải.

CÁN BỘ HƯỚNG DẪN
(Họ tên và chữ ký)

KHOA QUẢN LÝ CHUYÊN NGÀNH
(Họ tên và chữ ký)

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi. Các số liệu, kết quả nêu trong Luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Tôi xin cam đoan rằng mọi sự giúp đỡ cho việc thực hiện Luận văn này đã được cảm ơn và các thông tin trích dẫn trong Luận văn đã được chỉ rõ nguồn gốc.

Học viên thực hiện Luận văn

Lâm Thị Phương Thảo

LỜI CÁM ƠN

Trong quá trình hoàn thành luận văn này, tôi đã được các thầy cô Khoa Công nghệ Thông tin trường Đại học Công nghệ thành phố Hồ Chí Minh , và cơ quan nơi tôi đang công tác tạo mọi điều kiện thuận lợi cùng bạn bè đồng nghiệp và gia đình thường xuyên động viên khích lệ.

Tôi muốn được bày tỏ lòng biết ơn sâu sắc tới tiến sĩ Nguyễn Chánh Thành và tiến sĩ Lê Mạnh Hải, đây là hai giảng viên đã hết lòng hướng dẫn tôi trong quá trình thực hiện luận văn.

Tôi cũng muốn được bày tỏ lòng biết ơn các thầy cô Khoa Công nghệ Thông tin trường Đại học Công nghệ thành phố Hồ Chí Minh đã giúp đỡ và tạo điều kiện cho tôi rất nhiều trong quá trình học tập và nghiên cứu để hoàn thành luận văn.

Tôi xin được bày tỏ lòng biết ơn đến gia đình, người thân đã hỗ trợ, động viên tinh thần, và tạo điều kiện cho tôi trong suốt quá trình học tập và hoàn thành luận văn này.

Và cuối cùng, tôi xin được bày tỏ lòng biết ơn đến bạn bè đồng nghiệp và Ban giám hiệu nơi tôi công tác đã động viên, giúp đỡ, tạo điều kiện để tôi hoàn thành khoá học và luận văn này.

Tác giả Luận văn

Lâm Thị Phương Thảo

TÓM TẮT

Bản thể học ngày nay đã trở nên phổ biến và cần thiết trong nhiều lĩnh vực. Bản thể học trở thành một chủ đề nghiên cứu phổ biến trong một loạt các ngành, với mục đích làm tăng sự hiểu biết và xây dựng một sự đồng thuận trong một lĩnh vực nhất định của tri thức. Bản thể học cũng hướng đến việc chia sẻ kiến thức giữa các hệ thống và con người. Vì tầm quan trọng của bản thể học trong lĩnh vực Công nghệ Thông tin, trong việc chia sẻ kiến thức và phát triển khả năng tương tác giữa các bên sử dụng thông tin. Vì nhu cầu xây dựng bản thể học cho các lĩnh vực kiến thức cụ thể dựa trên kho ngữ liệu văn bản có sẵn, đề tài thực hiện " **XÂY DỰNG BẢN THỂ HỌC TỪ KHO NGỮ LIỆU DẠNG VĂN BẢN**".

Đã có nhiều công trình nghiên cứu về cách tạo bản thể học, và tất cả đều tập trung vào phương pháp bán tự động. Không có một phương pháp duy nhất hoàn hảo để tạo bản thể học, mỗi công trình có đều có những điểm mạnh và các điểm chưa tốt.

Đề tài thực hiện tìm hiểu cách tạo bản thể học bán tự động và trình bày các bước tạo bản thể học bán tự động từ nguồn ngữ liệu dạng văn bản, trong từng bước đề tài trình bày cách sử dụng công cụ hỗ trợ để thực hiện, hiện thực các thuật toán, cải tiến để có thể thực hiện tốt hơn. Kết quả quá trình thực nghiệm cho thấy tính khả thi của trình tự thực hiện mà luận văn đã trình bày.

ABSTRACT

Today, ontology has become popular and necessary in many areas. Ontology become a popular research topic in a variety of fields, for the purpose to increase understanding and build a consensus in domain-specific knowledge. The ontology also aims to share knowledge between agents and people. Because of the importance of ontology in information technology, in sharing of knowledge and the development of interoperability between the parties using information. Because of the need to build ontologies for specific knowledge based on the available text corpus, subject performed: DOMAIN ONTOLOGY CREATION FROM TEXT RESOURCES.

There have been many researches about how to create ontologies, and they all focused on semi-automatic method. There is not a single perfect method to create ontologies, each of which has strengths and weaknesses.

The research done to study how to create a semi-automatic ontology and presents the steps to create a semi-automatic ontology from text corpus, the research presented step by step how to use the tools supporting for presentation, the algorithm implementation, improving to be made better. Results of the experiments show the feasibility of the implementation process that thesis presented.

MỤC LỤC

DANH MỤC CÁC TỪ VIẾT TẮT

Số TT	Từ viết tắt	Từ đầy đủ
1	GATE	General Architecture for Text Engineering
2	ANNIE	A Nearly-New IE system
3	LHS	left hand side
4	RHS	right hand side
5	JAPE	a Java Annotation Patterns Engine

DANH MỤC CÁC BẢNG

DANH MỤC CÁC HÌNH ẢNH

Chương 1. MỞ ĐẦU

1.1 Lý do chọn đề tài

Trong những năm gần đây ontology (bản thể học) trở nên phổ biến vì những tiện ích mà nó mang lại cho người dùng. Có nhiều lý do để chúng ta xây dựng và phát triển một bản thể học:

- Bản thể học được sử dụng để hỗ trợ khả năng tương tác và sự hiểu biết chung giữa các bên khác nhau, là một thành phần quan trọng trong việc giải quyết vấn đề không đồng nhất ngữ nghĩa, vì thế cho phép khả năng tương tác ngữ nghĩa giữa các ứng dụng web và dịch vụ khác nhau. Gần đây, bản thể học đã trở thành một chủ đề nghiên cứu phổ biến trong nhiều cộng đồng, bao gồm cả kiến thức kỹ thuật, thương mại điện tử, quản lý và xử lý ngôn ngữ tự nhiên.
- Mục tiêu của bản thể học là để đạt được kiến thức chung và có thể chia sẻ giữa người với người và giữa các hệ thống ứng dụng. Vì thế, bản thể học đóng một vai trò quan trọng trong việc đạt được khả năng tương tác giữa các tổ chức. Bản thể học được sử dụng để cho phép thao tác giữa các ứng dụng web từ những lĩnh vực khác nhau hoặc từ các quan điểm khác nhau trên một lĩnh vực. Vì lý do đó, bản thể học cần thiết để thiết lập ánh xạ giữa các khái niệm khác nhau để nắm bắt sự tương ứng ngữ nghĩa giữa chúng. Tuy nhiên, việc thiết lập một sự tương ứng như vậy không phải là một công việc dễ dàng.

Bản thể học trở thành một chủ đề nghiên cứu phổ biến trong một loạt các ngành, với mục đích làm tăng sự hiểu biết và xây dựng một sự đồng thuận trong một lĩnh vực nhất định của tri thức. Bản thể học cũng hướng đến việc chia sẻ kiến thức giữa các hệ thống và con người.

Việc xây dựng bản thể học từ kho ngữ liệu có sẵn là rất cần thiết trong việc tăng sự hiểu biết và xây dựng một sự đồng thuận trong một lĩnh vực nhất định của tri thức, chia sẻ thông tin giữa các người dùng cùng quan tâm đến một lĩnh vực.

Vì tầm quan trọng của bản thể học trong lĩnh vực công nghệ thông tin, trong việc chia sẻ kiến thức và phát triển khả năng tương tác giữa các bên sử dụng thông tin. Vì nhu cầu xây dựng bản thể học cho các lĩnh vực kiến thức cụ thể dựa trên kho ngữ liệu văn bản có sẵn, đề tài " **XÂY DỰNG BẢN THỂ HỌC TỪ KHO NGỮ LIỆU DẠNG VĂN BẢN**" được chọn thực hiện.

1.2 Mục đích, đối tượng và phạm vi nghiên cứu

Từ động cơ nghiên cứu nêu trên, luận văn thực hiện tìm hiểu phương pháp xây dựng bản thể học bán tự động từ kho ngữ liệu dạng văn bản. Đây là vấn đề trọng tâm và là mục tiêu nghiên cứu của luận văn.

Hiện nay trên thế giới đã có nhiều phương pháp tạo bản thể học đã được công bố như là:

- Phương pháp học từ các bản thể học đã có (Learning Ontology) (theo [3]), đây là phương pháp tạo bản thể học mới dựa trên việc học từ các bản thể học đã có. Learning Ontology là việc thu thập 1 bản thể học trên 1 miền tri thức (local ontology) mới từ những bản thể học đang có.
- Phương pháp tiếp cận hỗn hợp (theo [4]), đây là phương pháp tạo bản thể học dựa trên phương pháp lập luận kết hợp với mô hình bản thể học đã có.

Có nhiều công cụ hỗ trợ tạo bản thể học khác nhau, theo [5], bao gồm: Protégé, OilEd, OntoLingua, Apollo, OntoEdit, RDFedt, WebODE, KAON, WebOnto, ICOM, DOE, Medius Visual Ontology Modeler, LinKFactory Workbench, K-Infinity[i].

Mỗi phương pháp đều có các ưu nhược điểm cùng với những công trình nghiên cứu và các thực nghiệm liên quan. Từ việc tìm hiểu các phương pháp đã có, để tìm hiểu phương pháp tạo bản thể học bán tự động từ kho ngữ liệu dạng văn bản đề tài tập trung thực hiện các nhiệm vụ sau:

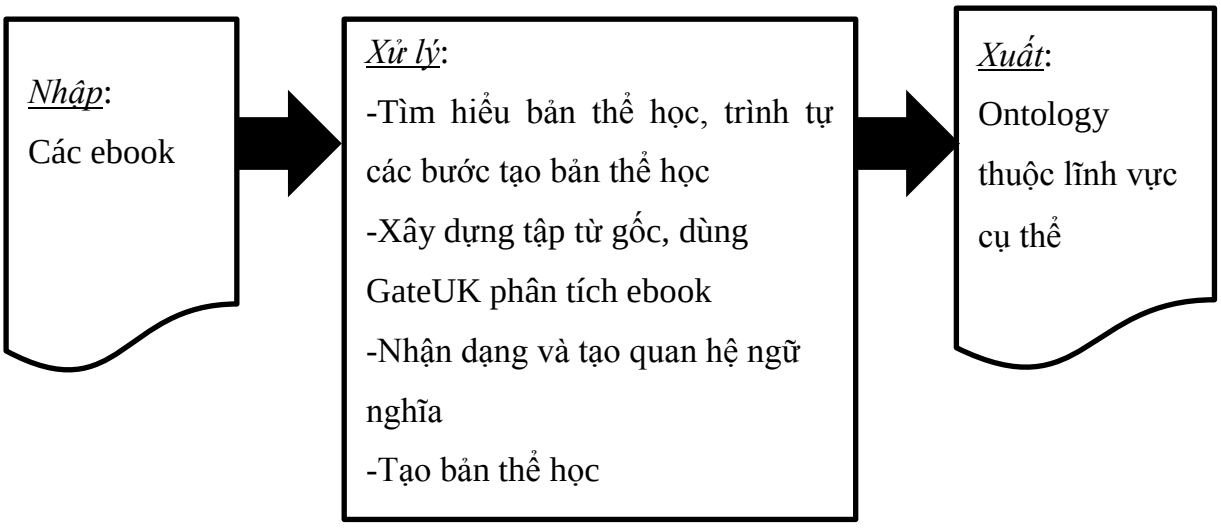
- Khảo sát các phương pháp xây dựng ontology từ kho ngữ liệu dạng văn bản.

[i] Website của những công cụ này được trình bày trong Phụ lục 1

- Đề xuất (hoặc cải tiến) một phương pháp xây dựng ontology từ kho ngữ liệu dạng văn bản trên cơ sở kết hợp xử lý ngôn ngữ tự nhiên.
- Tiến hành thực nghiệm, đánh giá và hiệu chỉnh phương pháp.

Để thực hiện các nhiệm vụ trên đề tài giải quyết các vấn đề chính sau:

- **Vấn đề thứ nhất**: tìm hiểu bản thể học, trình tự các bước tạo bản thể học.
- **Vấn đề thứ hai**: vận dụng xử lý ngôn ngữ tự nhiên để: xây dựng tập từ gốc, dùng GateUK phân tích các ebook xác định các từ, cụm từ liên quan đến bản thể học.
- **Vấn đề thứ ba**: nhận dạng và tạo các quan hệ ngữ nghĩa giữa các từ, cụm từ.
- **Vấn đề thứ tư**: tạo bản thể học, thử nghiệm và đánh giá



Hình 1-1. Nội dung thực hiện của luận văn

1.3 Ý nghĩa khoa học và thực tiễn của đề tài

Đề tài tập trung tìm hiểu và trình bày một phương pháp tạo bản thể học bán tự động từ nguồn ngữ liệu dạng văn bản, từ đó góp phần hỗ trợ người dùng tạo bản thể học cho một miền kiến thức cụ thể từ nguồn ngữ liệu văn bản.

Các đóng góp chính của luận văn:

- Trình bày được các bước tạo bản thể học bán tự động từ kho ngữ liệu văn bản.

- Trình bày được các bước sử dụng công cụ Gate UK để phân tích nội dung văn bản theo đặc trưng.
- Trình bày cách nhận dạng và tạo quan hệ ngữ nghĩa giữa các từ , cụm từ.
- Trình bày các bước tạo bản thể học dùng công cụ Protégé.

1.4 Cấu trúc của luận văn

Luận văn bao gồm năm chương:

Chương 1: Mở đầu, trình bày lý do chọn đề tài, mục tiêu, phạm vi và những đóng góp chính của luận văn, giới thiệu cấu trúc của luận văn.

Chương 2: Tổng quan, chương này giải quyết vấn đề thứ nhất giới thiệu các khái niệm về bản thể học, ngoài ra chương này còn phân tích, đánh giá các công trình nghiên cứu liên quan đến việc xây dựng bản thể học, chỉ ra những vấn đề mà đề tài cần tập trung nghiên cứu, giải quyết.

Chương 3: Các nghiên cứu liên quan, chương này trình bày các nghiên cứu liên quan để thực hiện đề tài: GateUK, Wordnet, các nghiên cứu liên quan đến việc tạo bản thể học, tập trung giải quyết vấn đề thứ hai và thứ ba : cách xây dựng tập từ gốc, thực hiện phân tích nội dung, cách nhận dạng và tạo quan hệ ngữ nghĩa, cách tạo bản thể học bán tự động

Chương 4: Thực nghiệm, chương này giải quyết vấn đề thứ tư: trình bày cách thức và kết quả quá trình thực nghiệm.

Chương 5: Kết luận và kiến nghị, là phần tổng kết, trong đó trình bày tóm lược kết quả luận văn và những đề nghị liên quan đến luận văn.

Danh mục tài liệu tham khảo.

Phụ lục

Chương 2. TỔNG QUAN

2.1 Giới thiệu bản thể học

Trình bày lý thuyết tổng quan về bản thể học, về cách tổ chức dữ liệu trong bản thể học, các công trình nghiên cứu liên quan đến tạo bản thể học.

2.1.1 Khái niệm Bản thể học

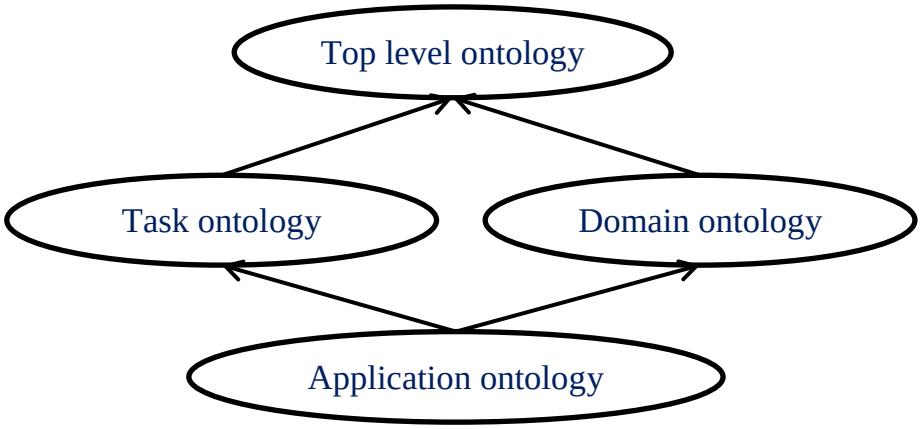
Theo Wikipedia[ii], "ontology" (bản thể học) là từ có nguồn gốc từ triết học và được dùng trong nhiều lĩnh vực như khoa học máy tính, hệ thống kỹ thuật, kỹ thuật phần mềm, sinh tin học, khoa học thư viện, kiến trúc thông tin và các website ngữ nghĩa (semantic web). Một số định nghĩa về bản thể học được sử dụng nhiều hiện nay :

- Theo quan điểm triết học, bản thể học là nghiên cứu về bản chất của sinh vật, sự tồn tại hoặc những sự vật thực tế, cũng như các loại sinh vật cơ bản và các mối quan hệ của chúng (theo Wikipedia).
- Trong khoa học máy tính, một bản thể học là một đặc tả rõ ràng của một sự trừu tượng hóa (theo [14]).
- Theo Mizoguchi, bản thể học là một hệ thống các khái niệm/ từ vựng được sử dụng như một khối trong hệ thống xử lý thông tin.
- Một bản thể học 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 (theo [9]).

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

Có nhiều cách phân loại bản thể học, theo [16], dựa trên đối tượng của khái niệm thì bản thể học được chia thành 4 loại như sau:

[ii] http://en.wikipedia.org/wiki/Ontology



Hình 2-1: Phân loại ontology theo đối tượng của khái niệm

- *Top level ontology*: mô tả khái niệm rất chung chung hoặc kiến thức thông thường như không gian, thời gian, sự kiện, hành động, ... Những khái niệm độc lập của một vấn đề hay một lĩnh vực cụ thể.
- *Domain ontology*: là một tập hợp các từ vựng và các khái niệm mô tả một miền ứng dụng hoặc các mục tiêu cụ thể. Hầu hết các bản thể học hiện nay là Domain ontology.
- *Task ontology*: được sử dụng để khái niệm hóa các nhiệm vụ cụ thể trong hệ thống. Nó điều chỉnh một tập hợp các từ vựng và các khái niệm mô tả một cấu trúc thực hiện các nhiệm vụ độc lập với miền.
- *Application ontology*: bản thể học này là cụ thể nhất. Các khái niệm trong bản thể học ứng dụng là ứng dụng trên lĩnh vực cụ thể và đặc biệt. Nói cách khác, các khái niệm thường tương ứng với vai trò của các lĩnh vực trong khi thực hiện một hoạt động nào đó.

Ngoài việc phân loại nêu trên, về mặt tổ chức thì bản thể học bao gồm bốn thành phần chính: khái niệm, các thể hiện, mối quan hệ và các tiên đề.

2.1.2 Ứng dụng của bản thể học hiện nay

Trong những năm gần đây, bản thể học đã trở thành một chủ đề nghiên cứu phổ biến trong một loạt các ngành, với mục đích tăng sự hiểu biết và xây dựng một sự thống nhất trong một lĩnh vực tri thức nhất định. Bản thể học cũng giải quyết việc chia sẻ kiến thức giữa các hệ thống và con người với nhau. Bản thể học xuất hiện

đầu tiên trong phòng thí nghiệm trí tuệ nhân tạo, trước khi được sử dụng trong các lĩnh vực khác, hiện nay bản thể học được sử dung rộng rãi trong nhiều lĩnh vực khác nhau: web ngữ nghĩa (Semantic Web), khám phá dịch vụ web ngữ nghĩa (Semantic Web Service Discovery), Trí tuệ nhân tạo (Artificial Intelligence), máy tìm kiếm (Search Engines), thương mại điện tử (E-Commerce).

2.1.3 Hiệu quả mang lại từ việc sử dụng bản thể học

Bản thể học trở thành một chủ đề nghiên cứu phổ biến trong một loạt các ngành, với mục đích tăng sự hiểu biết và xây dựng một sự đồng thuận trong một lĩnh vực nhất định của tri thức. Bản thể học cũng hỗ trợ việc chia sẻ kiến thức giữa các hệ thống và con người.

2.1.4 Cách tổ chức dữ liệu trong một bản thể học

Bản thể học bao gồm bốn thành phần chính:

- **Concept (**class hoặc term): khái niệm là một nhóm trừu tượng, tập hợp các đối tượng. Đây là yếu tố cơ bản của tên miền và thường đại diện cho một nhóm hoặc lớp mà các thành viên chia sẻ thuộc tính chung.
- **Instance:** một thể hiện, biểu diễn cho một lớp, một đối tượng cụ thể.
- **Relation:** mối quan hệ giữa các khái niệm.
- **Axiom:** được sử dụng để ràng buộc giá trị của các class hoặc các thể hiện, vì vậy tiên đề axiom sử dụng ngôn ngữ logic; chúng được sử dụng để xác minh tính hợp lệ của bản thể học.

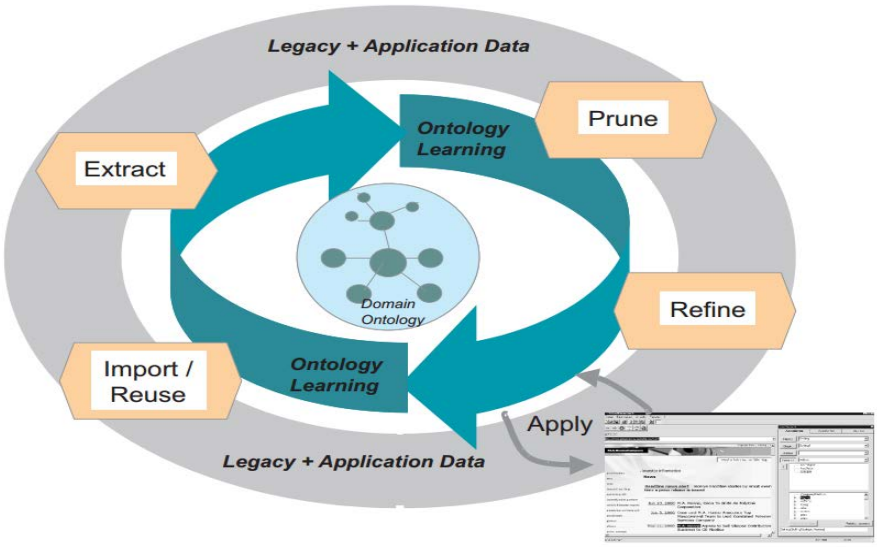
2.2 Các công trình nghiên cứu đã có liên quan mật thiết đến đề tài

Hiện nay trên thế giới và trong nước đã có nhiều công trình nghiên cứu về cách tạo bản thể học:

2.2.1 Học từ bản thể học

- Học từ bản thể học, theo [3], có thể được mô tả như là việc thu thập một bản thể học trên một miền tri thức (local ontology) mới từ những bản thể học đang có. Learning Ontology cần dữ liệu đầu vào để học những khái niệm liên quan đến miền đã biết trước, những định nghĩa của khái niệm cũng như các mối quan hệ tổ chức giữa những định nghĩa này.

- Học từ bản thể học bao gồm việc mượn kiểu dữ liệu không cấu trúc, bán cấu trúc, có cấu trúc để hỗ trợ, quá trình thiết kế bản thể học bán tự động. Học từ bản thể học tiến hành thông qua nhập (import), rút trích (extract), tỉa (prune), lọc (refine), và ước lượng tạo cho người thiết kế bản thể học có nhiều sự phối hợp các công cụ cho mẫu bản thể học.
- Quá trình Learning Ontology trải qua năm bước:



Hình 2-2: Các bước quá trình Học bản thể học

(nguồn: trang 3 tài liệu tham khảo [3])

- Bước 1: bản thể học hiện có được import và tái sử dụng từ sự kết hợp các cấu trúc hiện tại, hoặc định nghĩa sự sắp xếp các qui tắc giữa cấu trúc hiện tại và bản thể học đã được thành lập
- Bước 2: trong bản thể học đã có trích 1 phần từ bản thể học đích được mô hình với sự hỗ trợ từ web document.
- Bước 3: hình dáng thô của bản thể học đích cần cắt tỉa để điều chỉnh thành bản thể học tốt hơn bản ban đầu của nó.
- Bước 4: bản thể học chọn lọc những lợi thế từ domain ontology nhất định hoàn thành bản thể học đích tốt (trái ngược với khai thác).
- Bước 5: cung cấp một ứng dụng đích như là một giới hạn để kiểm tra bản thể học kết quả.

2.2.2 Tạo bản thể học bán tự động từ kho ngữ liệu văn bản

Phương pháp này, theo [2], thực hiện tự động trích xuất xâu ngữ nghĩa thông tin từ các nguồn văn bản và nhanh chóng tạo ra bản thể học miền ngữ nghĩa phong phú trong khi vẫn giữ can thiệp bằng tay đến mức tối thiểu.

Phương pháp này sử dụng công cụ Polaris để tự động trích xuất thông tin ngữ nghĩa từ văn bản. Sau đó sử dụng công cụ Jaguar xây dựng bán tự động mô hình bản thể học trên miền tri thức .

Polaris [2], là bộ phân tích cú pháp ngữ nghĩa, tự động trích xuất thông tin ngữ nghĩa từ văn bản. Polaris được xây dựng dựa trên một tập hợp 26 mối quan hệ ngữ nghĩa mà Lymba đã xác định. Polaris chạy phân loại trên từng phần của văn bản phù hợp với một mô hình cú pháp. Phân loại kiểm tra tính năng của các văn bản và xác định xem có quan hệ nào của 26 quan hệ ghép vào giữa các yếu tố của mô hình không. Hầu hết các phân loại dựa trên các thuật toán học máy khác nhau, và tán xạ ngữ nghĩa (một thuật toán học mới sử dụng các lớp WordNet để tìm mối quan hệ có thể xảy ra nhất giữa hai danh từ).

Jaguar xử lý các nguồn tài nguyên văn bản và nhanh chóng xây dựng bản thể học lĩnh vực chuyên biệt trong định dạng Lymba hoặc trong các định dạng tiêu chuẩn như RDF của W3C và OWL. Các văn bản đầu vào cho Jaguar có thể từ nhiều nguồn khác nhau bao gồm cả văn bản, MS Word, PDF và trang web HTML , ... Một Jaguar kiến thức cơ bản bao gồm các thành phần sau đây:

- Các khái niệm bản thể học: khối xây dựng cơ bản của một bản thể học
- Hệ thống phân cấp: cấu trúc để lấy được kiến thức phổ quát trên một số khái niệm bản thể học thông qua các mối quan hệ bắc cầu (ví dụ như IS-A, Whole-Part, LOCATIVE , v.v…)
- Kiến thức theo ngữ cảnh: cụm kiến thức lấy được thông qua tất cả các mối quan hệ ngữ nghĩa được phát hiện bởi bộ phân tích ngữ nghĩa.
- Tiên đề theo yêu cầu: lấy được khẳng định về khái niệm của điều quan tâm tạo ra từ những kiến thức có giá trị và có ích cho việc suy luận trên văn bản.

Phương pháp này, áp dụng một thủ tục tổng quát và cải tiến để tự động trích xuất thông tin ngữ nghĩa từ nguồn tài nguyên văn bản và tạo ra bản thể học lĩnh vực cụ thể phong phú trong khi vẫn giữ can thiệp bằng tay đến mức tối thiểu.

2.2.3 Kỹ thuật tạo bản thể học bán tự động sử dụng mô hình

Theo [4], do nhu cầu sử dụng bản thể học ngày càng tăng nên nhu cầu xây dựng bản thể học ngày càng nhiều vì vậy kỹ thuật tạo bản thể học cần được thực hiện bán tự động để giảm đáng kể công sức tạo bản thể học. Việc tái sử dụng kiến thức (đã được huấn luyện) trong xây dựng bản thể học sẽ làm giảm công sức xây dựng và tăng chất lượng của đầu ra của bản thể học. Nghiên cứu này đề xuất tập trung vào một cách tiếp cận kết hợp (lai) để xây dựng bản thể học.

Thiết kế bản thể học thủ công là một công việc tẻ nhạt và phức tạp, do đó các nghiên cứu tập trung vào phương pháp thực hiện bán tự động. Nghiên cứu giải quyết hai vấn đề:

- Tự động hóa trong suốt quá trình xây dựng.
- Tái sử dụng kiến thức, những hoạt động chung của công việc cần được khai thác, cũng như hoạt động tốt nhất trong thiết kế bản thể học.

Mục đích của nghiên cứu này là phát triển một phương pháp kết hợp (tên OntoCase) kết hợp quan điểm học từ tình huống dựa trên lý luận (CBR) và tái sử dụng kiến thức đồng thuận thông qua mô hình. Nghiên cứu này tập trung vào bản thể học ứng dụng trong các công việc, được sử dụng chủ yếu cho việc cấu trúc và tìm kiếm thông tin, tập trung vào thiết kế và kiến trúc mô hình bán tự động. Trong phương pháp này, mô hình kiến trúc chủ yếu là tập hợp các ràng buộc dẫn hướng và hạn chế các thành phần của bản thể học từ các mẫu thiết kế.

Các cơ sở của một phương pháp tiếp cận CBR trong OntoCase tương ứng với một mô hình danh mục (mô hình cơ sở), bao gồm cả thiết kế bản thể học và mô hình kiến trúc. Các mẫu thiết kế được biểu diễn như là bản thể học nhỏ.

CBR là phương pháp sử dụng các kiến thức đã có để tìm ra vấn đề mới. Chu kỳ CBR có thể được mô tả như một quá trình lặp đi lặp lại các giai đoạn:

- Giai đoạn thu hồi phân tích ngữ liệu văn bản đầu vào và phát sinh đại diện của mình, sau đó so khớp điều này với mô hình cơ sở và lựa chọn mô hình thích hợp.
- Giai đoạn tái sử dụng quan tâm đến sự thích ứng của các mô hình, kết hợp chúng thành một bản thể học.
- Giai đoạn điều chỉnh bao gồm việc mở rộng bản thể học, dựa trên kết quả đánh giá.
- Giai đoạn duy trì mô hình bao gồm việc phát hiện ra các mô hình mới và cải thiện mô hình hiện có

Trong đó: thu hồi và tái sử dụng là phần chính của nghiên cứu, điều chỉnh lại và duy trì vẫn chưa giải quyết.

Những đóng góp chính của phương pháp này bao gồm:

- Tự động hóa hơn nữa trong quá trình xây dựng bản thể học.
- Tăng chất lượng của các bản thể học, so với phương pháp tiếp cận OL hiện có. Chất lượng tăng này chủ yếu là do việc sử dụng các mô hình, đại diện cho cả kiến thức chuyên môn và kinh nghiệm trước đây, và sự ra đời của một số bước sửa đổi trong bốn giai đoạn của phương pháp.

2.2.4 Kỹ thuật bán tự động của bản thể học từ văn bản

Nghiên cứu [11] đưa ra một số đề xuất để tạo điều kiện thiết kế bản thể học thông qua việc tự động phát hiện từ miền dữ liệu, văn bản ngôn ngữ tự nhiên, lĩnh vực chuyên biệt.

Nghiên cứu này trình bày một khuôn khổ cho các kỹ thuật bán tự động của bản thể học, một cách tiếp cận mới để phát hiện các mối quan hệ khái niệm không phân loại từ văn bản và để tạo điều kiện cho các kỹ thuật của mối quan hệ không phân loại. Xây dựng trên phần phân loại của bản thể học, phương pháp tiếp cận của nghiên cứu phân tích các văn bản trên miền cụ thể sử dụng các phương pháp xử lý văn bản để xác định cặp từ có mối liên hệ ngôn ngữ với nhau.

2.2.5 Dafoe: Một Nền tảng cho việc xây dựng bản thể học từ văn bản

DAFOE [12] là một nền tảng xây dựng bản thể học sử dụng các loại văn bản đầu vào khác nhau (văn bản gốc, kết quả của các công cụ xử lý ngôn ngữ tự nhiên, thuật ngữ hoặc từ chuẩn). Dafoe hỗ trợ cấu trúc kiến thức và mô hình khái niệm từ những mục ngôn ngữ cũng như hình thức hóa bản thể học. Dafoe cung cấp mô hình với hai tính năng ban đầu: một bản thể học nối với một thành phần từ vựng và một nối với các văn bản, ngôn ngữ đầu vào với mục đích định nghĩa chúng. Các yêu cầu của nền tảng và phát triển của nó tập trung vào 3 vấn đề:

- Tích hợp các loại công cụ đang được sử dụng trong phạm vi một nền tảng mô hình duy nhất.
- Đảm bảo sự bền bỉ và truy xuất nguồn gốc của toàn bộ quá trình xây dựng bản thể học.
- Phát triển trên nền tảng trong môi trường mã nguồn mở và phần mở rộng có thể thêm vào.

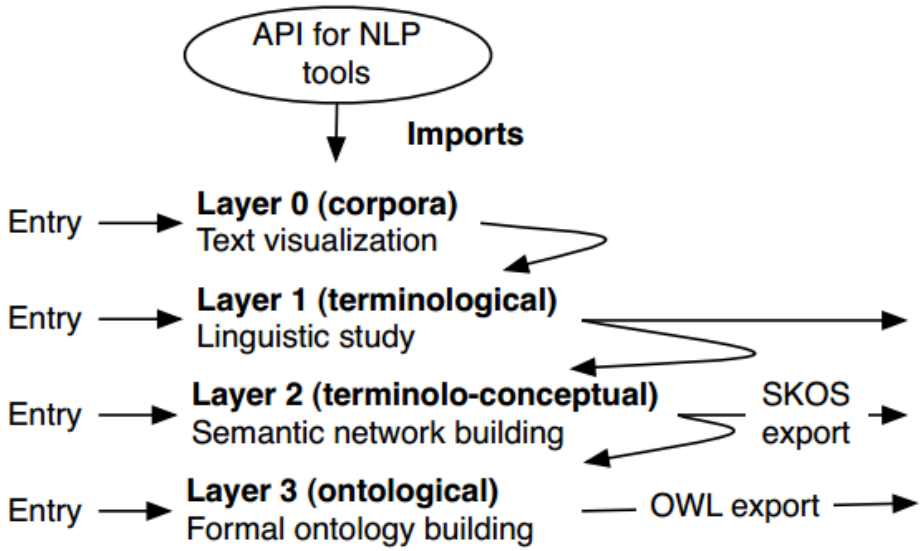
Dafoe đề xuất các công cụ tương tự như Text2Onto, nhưng sự giám sát của con người sẽ đóng một vai trò quan trọng cho việc lựa chọn công cụ, xác nhận kết quả và khái niệm. Kết quả của Dafoe thường sẽ là một nguồn thuật ngữ-bản thể học nơi bản thể học được kết nối với một thành phần từ vựng.

Dữ liệu đầu vào của Dafoe là bất cứ nguồn thông tin nào (văn bản, thuật ngữ, bộ từ chuyên môn) đều được sử dụng.

Phương pháp này có tính đến toàn bộ quá trình "chuyển hóa" dữ liệu văn bản vào bản thể học và phân chia thành các giai đoạn khác nhau, tương ứng với mức độ đầu vào khác nhau

- Phương pháp này dựa trên hai ý tưởng chính:
  - Dữ liệu, văn bản là một nguồn thông tin quan trọng để xây dựng bản thể học, đặc biệt là nếu bản thể học được sử dụng để ghi chú các tài liệu dạng văn bản
  - Dữ liệu, văn bản không thể được ánh xạ trực tiếp vào một bản thể học và việc chuyển đổi phải qua trung gian.

- Mô hình dữ liệu được cấu trúc thành bốn lớp:
  - Corpora Layer
  - Terminological Layer
  - Termino-Conceptual Layer
  - Ontology Layer



Hình 2-3: Mô hình dữ liệu

(Nguồn: trang 2 tài liệu tham khảo [12])

Dafoe được thiết kế để cung cấp một loạt các phương pháp kỹ thuật bản thể học. Sự đa dạng này không thể quản lý trong một mô hình duy nhất và tĩnh, nghiên cứu đã áp dụng kiến trúc OntoDB để hỗ trợ mô hình quản lý và bổ sung. Sức mạnh của phương pháp tiếp cận Dafoe là:

- Định nghĩa chính xác các bước khác nhau có thể thiết kế một hình thức bản thể học;
- Mô hình dữ liệu bảo đảm sự bền bỉ và truy xuất nguồn gốc của toàn bộ quá trình xây dựng bản thể học;
- Cung cấp các hướng dẫn về phương pháp linh hoạt;
- Kiến trúc dựa trên mô hình MOF và bổ sung khả năng thích ứng để đảm bảo khả năng mở rộng của mô hình và quy trình xung quanh một công cụ cốt lõi;

- Các đặc điểm kỹ thuật của các chiến lược khác nhau dựa trên mô hình đầu vào / đầu ra khác nhau của nền tảng này;
- Sản phẩm cuối cùng của bản thể học liên quan đến một thành phần thuật ngữ.

2.3 Những vấn đề mà đề tài tập trung nghiên cứu, giải quyết

Hiện nay, vì trên thế giới đã có nhiều công trình nghiên cứu về cách tạo bản thể học, và hầu như đa số đều tập trung vào cách tạo bản thể học bằng kỹ thuật bán tự động. Vì vậy, để tìm hiểu phương pháp xây dựng bản thể học bán tự động từ kho ngữ liệu dạng văn bản chúng tôi tìm hiểu và giải quyết các vấn đề sau:

2.3.1 Tìm hiểu bản thể học

Tìm hiểu khái niệm bản thể học, cấu trúc bản thể học, cách xây dựng bản thể học .

2.3.2 Xây dựng tập từ gốc, xác định các từ, cụm từ

Vận dụng kết hợp các kết quả nghiên cứu và công cụ trong lĩnh vực Xử lý Ngôn ngữ Tự nhiên để:

- Sưu tập nguồn văn bản: các ebook thuộc lĩnh vực cụ thể.
- Xây dựng tập từ gốc thủ công và kết hợp WordNet.
- Sử dụng Gate UK phân tích các ebook lấy ra các thông tin theo chủ đề đã xây dựng trong tập từ gốc.

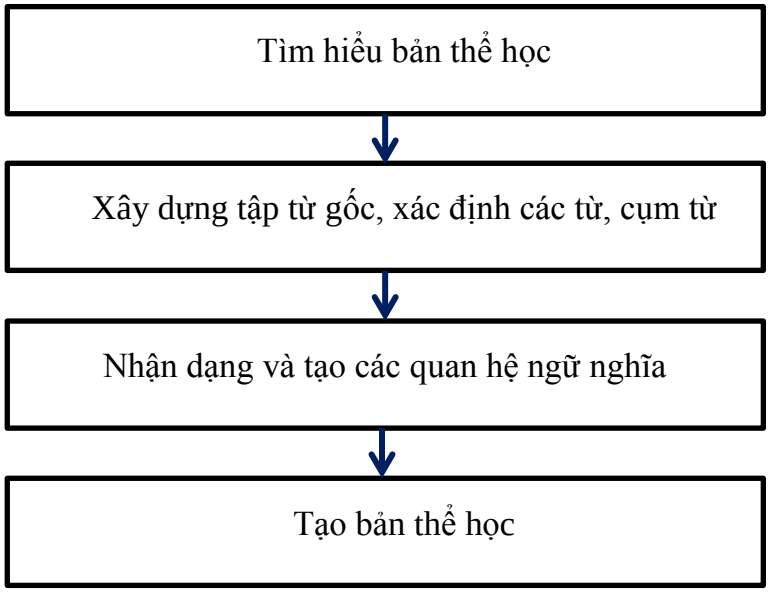
2.3.3 Nhận dạng và tạo các quan hệ ngữ nghĩa

Từ nguồn kết quả có được sau khi xác định các từ, cụm từ tiến hành nhận dạng và tạo các mối quan hệ ngữ nghĩa:

- Xây dựng cơ sở dữ liệu cho các từ, cụm từ.
- Phân tích mối quan hệ thủ công.
- Kết hợp với xây dựng các thủ tục phân loại khái niệm.

2.3.4 Tạo bản thể học

Sử dụng công cụ Protégé để tạo bản thể học và tiến hành thực nghiệm đánh giá.



Hình 2-4: Các bước thực hiện tạo ontology từ kho ngữ liệu văn bản.

Tóm lại: trong chương 2 luận văn đã trình bày một số nghiên cứu về cách tạo bản thể học đã có, từ đó trình bày hướng giải quyết vấn đề của đề tài, trình tự các bước thực hiện để tìm hiểu cách tạo một bản thể học bán tự động từ nguồn ngữ liệu văn bản.

Chương 3. CÁC NGHIÊN CỨU LIÊN QUAN

Chương này trình bày các nghiên cứu liên quan để thực hiện luận văn như: Gate-UK, Wordnet, các nghiên cứu liên quan đến việc tạo bản thể học, cách thực hiện phân tích nội dung, cách nhận dạng và tạo quan hệ ngữ nghĩa, cách tạo bản thể học bán tự động.

3.1 Gate UK

GATE (General Architecture for Text Engineering)[17], được phát triển bởi một nhóm nghiên cứu của Trường Đại học Sheffield, Anh Quốc từ năm 1995, là một hệ thống các phương pháp và công cụ xây dựng và phát triển các ứng dụng xử lý ngôn ngữ tự nhiên đặt biệt là rút trích thông tin. GATE nổi trội về xử lý văn bản. Cộng đồng người dùng của GATE rất lớn, đa dạng và trải rộng trên hầu hết các châu lục.

GATE là một phần mềm mã nguồn mở, người dùng có thể nhận được hỗ trợ miễn phí từ cộng đồng người dùng và các nhà phát triển thông qua GATE.ac.uk. Đây là dự án xử lý ngôn ngữ tự nhiên mã nguồn mở lớn .

GATE hỗ trợ các nhà nghiên cứu và phát triển phần mềm theo ba lĩnh vực:

- Kiến trúc phần mềm (Software Architecture)
- Khuôn mẫu (Framework)
- Môi trường phát triển (Development Environment)
- Gate UK có nhiều plugin, từng plugin có các công dụng khác nhau.

GATE chứa các thành phần phục vụ những tác vụ xử lý ngôn ngữ khác nhau như các bộ phân tích, dán nhãn, các công cụ tìm kiếm thông tin, các thành phần chiết xuất thông tin,...GATE cung cấp hệ thống chiết xuất thông tin ANNIE được sử dụng rộng rãi.

3.1.1 ANNIE

ANNIE (A Nearly-New IE system) chứa một tập các tài nguyên xử lý cốt lõi ngoài ra khi cần có thể gọi thêm các tài nguyên khác có sẵn vào ứng dụng. Mỗi tài nguyên trong ANNIE tạo ra một số chú thích mới hoặc sửa đổi những cái hiện có. Các tài nguyên thông dụng trong ANNIE:

- Document Reset PR: loại bỏ các chú thích

- ANNIE English Tokeniser: tạo chú thích Token
- ANNIE Gazetteer: tạo chú thích Lookup
- ANNIE Sentence Splitter: tạo chú thích Sentence, Split
- ANNIE POS tagger: thêm các nhãn từ loại vào chú thích Token

3.1.2 JAPE

JAPE (a Java Annotation Patterns Engine) là một ngôn ngữ để viết các biểu thức chính quy qua các chú thích, và sử dụng các mô hình phù hợp làm cơ sở để tạo thêm các chú thích. Các công cụ trích xuất thông tin tích hợp trong GATE dùng JAPE (kết hợp với những công cụ khác).

Văn phạm JAPE bao gồm tập các giai đoạn, mỗi giai đoạn bao gồm tập các qui tắc mẫu hoặc hành động. Các giai đoạn chạy tuần tự và tạo thành một chuỗi các bộ chuyển đổi trạng thái hữu hạn thông qua các chú thích.

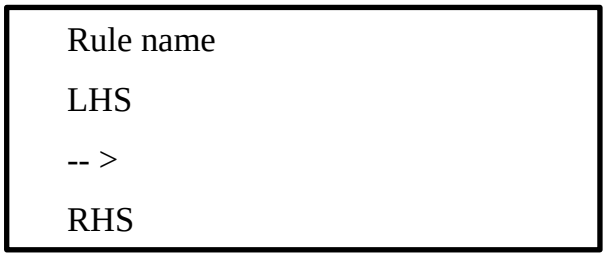
- Vế trái của qui tắc LHS (left hand side) bao gồm mẫu chú thích có thể chứa các biểu thức toán thông thường(*, ?, +) , mô tả khuôn mẫu để so khớp, những điều kiện và những đặt trưng của chú thích, có dạng:

  (

  .....

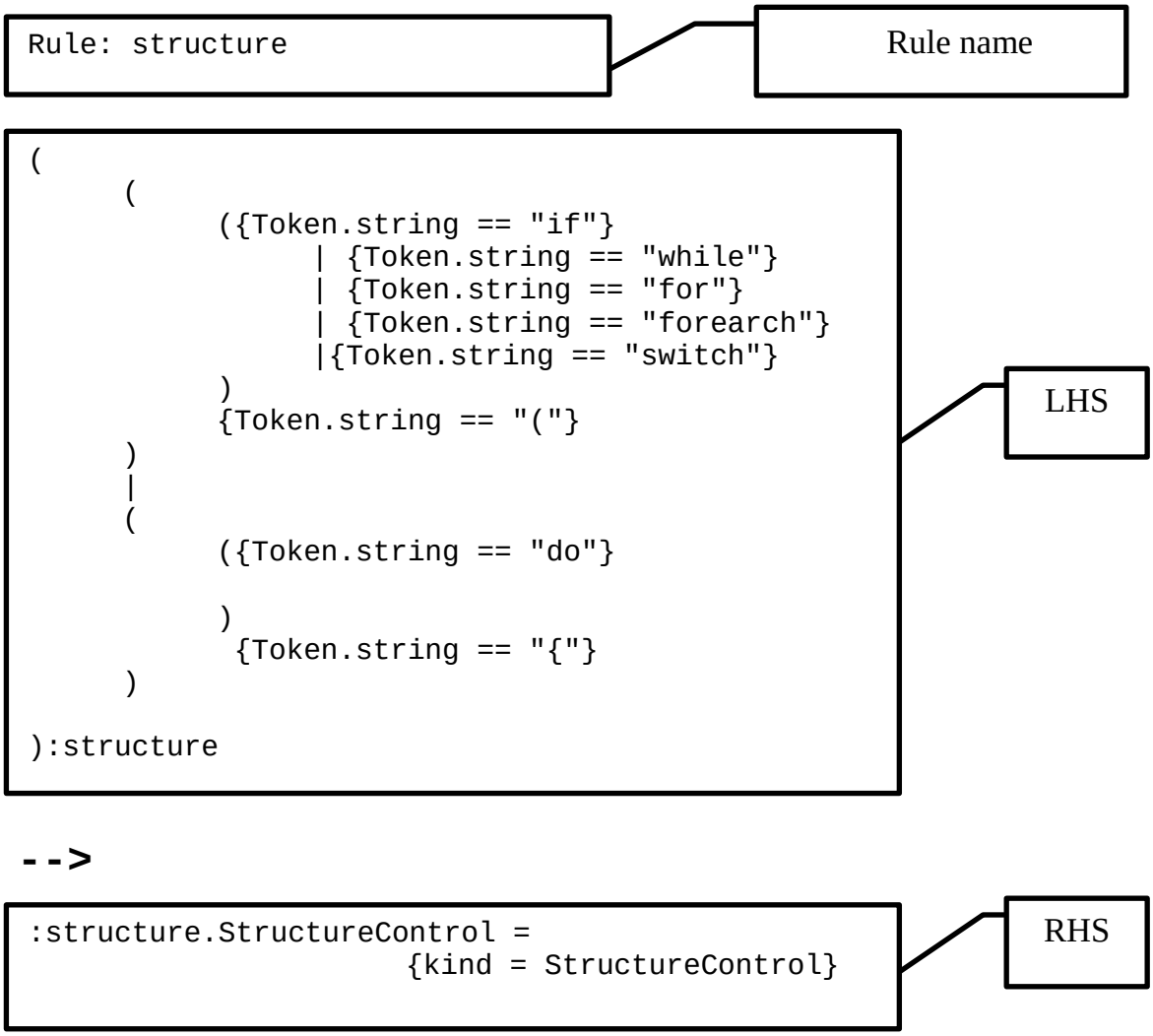
  ): label

- Vế phải của qui tắc RHS (right hand side) bao gồm các câu lệnh thao tác của chú thích, có cấu trúc **Label. Annotation type = features + value**
- Rule name có cấu trúc **Rule: ruler name**



Hình 3-1: Cấu trúc luật JAPE.

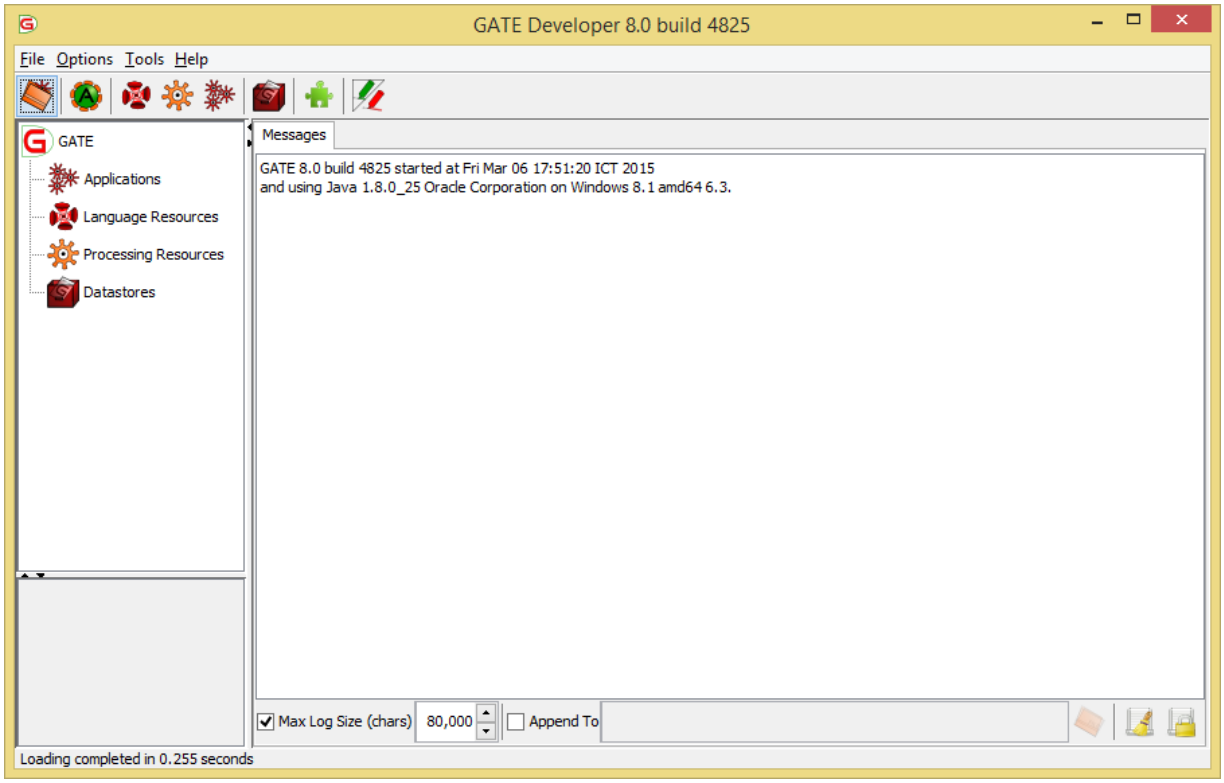
Ví dụ:



Hình 3-2:Ví dụ cấu trúc luật JAPE

3.1.3 Trình tự thực hiện tạo ứng dụng trong Gate UK

- Khởi động Gate UK.
- Tạo tài liệu (GATE Document), tập tài liệu (GATE Corpus) đã chọn.
- Tải ANNIE.
- Tạo ứng dụng mới (corpus pipeline).
- Chọn các plugin ANNIE và luật JAPE cần thực thi trong ứng dụng.
- Thực thi ứng dụng.



Hình 3-3: Giao diện Gate UK.

3.2 WordNet

WordNet [20] được Miller và cộng sự tại trường Đại học Princeton (Mỹ) xây dựng vào năm 1980. WordNet là một bản thể học tổng quát và là một hệ cơ sở tri thức từ vựng tiếng Anh. Các từ vựng trong WordNet được phân loại và tổ chức thành các tập đồng nghĩa gọi là synset. Mỗi tập synset biểu diễn một ý niệm cơ bản. Các synset được nối với nhau bởi nhiều loại quan hệ (relation) khác nhau.WordNet tổ chức thành 25 cấu trúc cây phân cấp riêng biệt tương ứng với các lĩnh vực có ngữ nghĩa khác nhau cho synset. Ngoài ra, WordNet còn bao gồm một số thành phần chủ yếu như word, sense, category ... và các quan hệ ngữ nghĩa liên kết .

Các quan hệ của WordNet bao gồm:

- Quan hệ đồng nghĩa Synonymy
- Quan hệ trái nghĩa Antonymy
- Quan hệ bao hàm Hypernymy: diễn tả ý nghĩa A là một loại (is a kind of).....

Ví dụ: ngôn ngữ lập trình là một loại ngôn ngữ

- Quan hệ thuộc cấp Hyponymy (Is-A): diễn tả ý nghĩa .... là một loại (is a kind of) của A

  Ví dụ: C là một loại ngôn ngữ lập trình

- Quan hệ bộ phận-toàn thể Meronymy: diễn tả ý nghĩa những thành phần của (parts of) A là ...

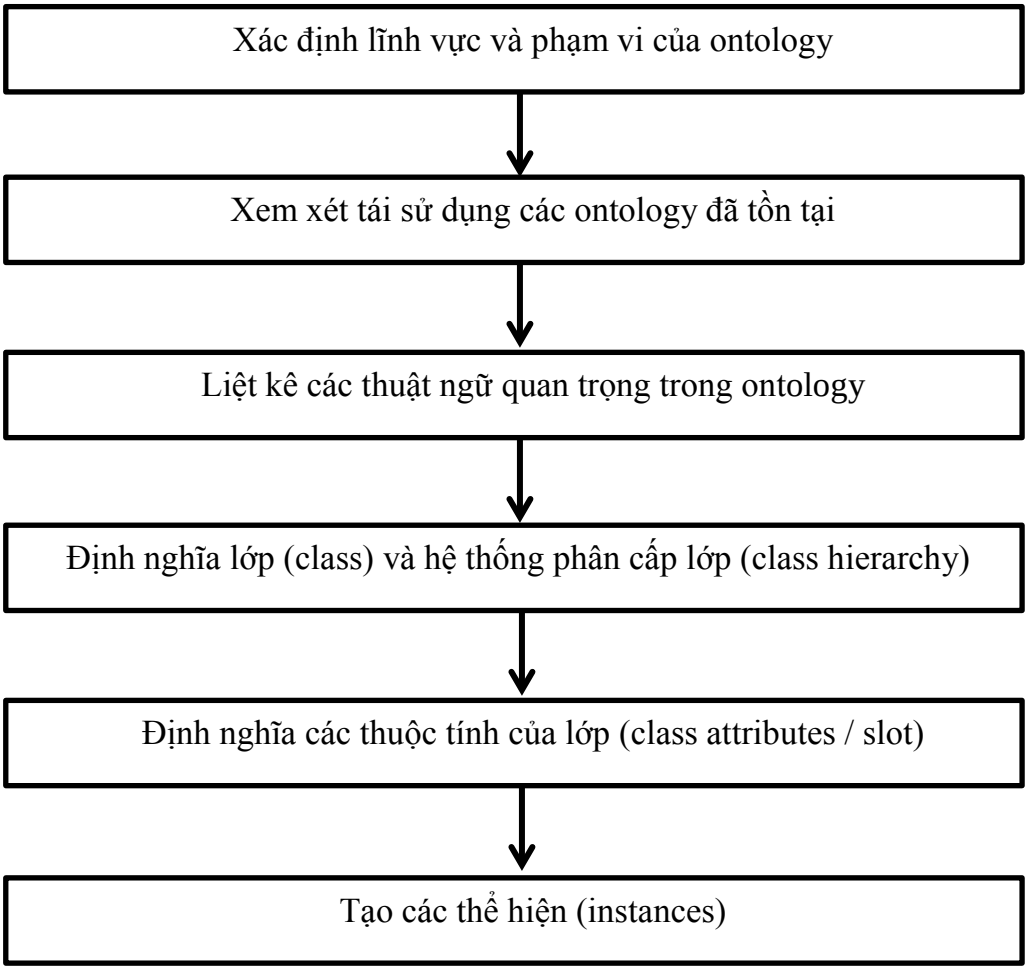
  Ví dụ: ngón tay "là thành phần của" tay

- Quan hệ toàn thể- bộ phận Holonymy: diễn tả ý nghĩa A là thành phần của (is a part of).....

  Ví dụ: tay "là thành phần của" cơ thể

3.3 Phương pháp thực hiện tạo bản thể học

Theo [9] không có một phương pháp duy nhất thật sự hoàn chỉnh để thiết kế một bản thể học. Trong đề tài này trình bày các bước thực hiện tạo ontology dựa theo [2], [7], [9]



Hình 3-4: Sơ đồ các bước thực hiện tạo bản thể học.

Bước 1: Xác định lĩnh vực và phạm vi của bản thể học.

Để thực hiện điều này cần trả lời các câu hỏi sau:

- Lĩnh vực bản thể học sẽ bao phủ là gì?
- Chúng ta sẽ sử dụng bản thể học để làm gì?
- Thông tin trong bản thể học sẽ cung cấp cho loại câu hỏi nào?
- Ai sẽ sử dụng và duy trì bản thể học?

Sau khi đã xác định lĩnh vực và phạm vi cụ thể tiến hành chọn các ebook là nguồn tư liệu chính để thực hiện bản thể học

- Xác định nguồn tài nguyên (tư liệu, chuyên gia, và các bản thể học đã có)
  - Nếu là chuyên gia trong lĩnh vực: chúng ta có thể tự tin thực hiện
  - Nếu chỉ có kiến thức trong lĩnh vực chúng ta cần bổ sung nhiều kiến thức hơn bằng cách:

- Nhờ chuyên gia: cần phải tham khảo ý kiến chuyên gia về tất cả kiến thức liên quan đến vấn đề mà bạn quan tâm.
- Tham khảo các tài liệu liên quan
- Tham khảo các bản thể học đã tồn tại.

Bước 2: Xem xét tái sử dụng các bản thể học đã tồn tại.

Đây là bước thường được sử dụng để giảm công sức thực hiện bản thể học. Bằng cách kế thừa các bản thể học có sẵn, người xây dựng có thể thêm, bớt các lớp, điều chỉnh quan hệ giữa các lớp, thực thể ... để điều chỉnh theo mục đích sử dụng của bản thể học. Ngoài ra việc sử dụng các bản thể học có sẵn cũng đóng vai trò quan trọng trong việc tương tác giữa các ứng dụng khác nhau, vì các ứng dụng cần phải hiểu các lớp, thực thể ... của nhau để thuận tiện trong việc thống nhất, trao đổi thông tin.

Hiện nay bản thể học về lĩnh vực ngôn ngữ lập trình (tiếng Anh) chưa nhiều, trong quá trình thực hiện đề tài chúng tôi tham khảo tài liệu [8], tuy nhiên vẫn không tìm được tài nguyên nên đề tài phải xây dựng một bản thể học mới.

Bước 3: Liệt kê các thuật ngữ quan trọng trong bản thể học.

- Bước 3.1: Sưu tập các từ gốc.
- Bước 3.2: Dùng Gate phân tích các ebook, phối hợp sử dụng plugin ANNIE và JAPE để phân tích, xác định các từ, cụm từ liên quan.
- Bước 3.3: Tạo danh sách và/hoặc sơ đồ của tất cả danh từ, động từ đã phân tích được, điều này rất quan trọng. Đối với từng thuật ngữ cố gắng viết ra: tên, từ đồng nghĩa, mô tả, kiểu, nguồn gốc .
- Bước 3.4: Xây dựng cơ sở dữ liệu cho tập các thuật ngữ đã phân tích được.

Bước 4: Định nghĩa lớp (class) và hệ thống phân cấp lớp (class hierarchy).

Thông thường, cần tạo ra một vài định nghĩa cho các khái niệm trong hệ thống phân cấp và sau đó tiếp tục bằng việc mô tả tính chất của các khái niệm này.

- Một số cách tiếp cận trong việc phát triển hệ thống phân cấp lớp:

- *Cách tiếp cận từ trên xuống (Top- Down):* bắt đầu với việc xác định những khái niệm chung nhất trong lĩnh vực và tiếp theo là xác định các khái niệm chuyên biệt hơn
- *Cách tiếp cận từ dưới lên (Bottom- Up)*: bắt đầu với xác định các lớp cụ thể nhất, các lá của các hệ thống phân cấp, tiếp theo xác định nhóm các lớp trong khái niện tổng quát hơn.
- *Cách kết hợp cả hai cách Top- Down và Bottom- Up trên (combination)*: bắt đầu với các khái niệm nổi bật hơn, sau đó là các khái niệm chung và chuyên biệt. Có thể bắt đầu với các khái niệm ở mức độ cao, và một vài khái niệm chuyên biệt sau đó liên kết chúng bằng các khái niệm trung gian

Có nhiều cách khác nhau để phân loại các mối quan hệ lớp cha (superclass) và lớp con (subclass)

- Lớp con (subclass): khái niệm C1 là lớp con của khái niệm C2, nếu và chỉ nếu mỗi thể hiện của C1 cũng là thể hiện của C2
- Sự phân hoạch rời của C (Disjoint decomposition of C): tập các lớp con của C không có chung thể hiện và không bao phủ C
- Sự phân hoạch toàn diện của C (Exhaustive decomposition of C): tập các lớp con của C có thể có cùng thể hiện và lớp con và tập này bao phủ C
- Sự phân chia của C (Partition of C): tập các lớp con không chia xẻ chung các thể hiện nhưng bao phủ C

Bước 5: Định nghĩa các thuộc tính của lớp (class attributes / slot)

Sau khi xác định một số lớp, chúng ta phải mô tả cấu trúc bên trong của các khái niệm. Hầu hết các thuật ngữ còn lại có thể sẽ là thuộc tính của các lớp này

Ở giai đoạn này một số danh từ trong danh sách có thể được xem là thuộc tính, các thuật ngữ được dùng để mô tả cho các thuật ngữ khác.

- Bản thể học phân biệt giữa thuộc tính của lớp và thuộc tính của thể hiện:
  - Thuộc tính của lớp (class attributes): thuật ngữ mô tả các khái niệm sẽ nhận được giá trị trong lớp định nghĩa chúng.

- Thuộc tính của thể hiện (instance attributes): là những thuật ngữ mô tả những khái niệm sẽ nhận giá trị của chúng trong thể hiện và có thể khác nhau giữa các thể hiện.

- Một số cách thực hiện:
  - Bước này và bước định nghĩa cách phân loại thường thực hiện đan xen nhau: một số lớp có thể sẽ chỉ là các thuộc tính để mô tả các lớp khác hoặc các thể hiện.
  - Cố gắng chọn những thuộc tính để lớp hoặc khái niệm chung nhất có thể có thuộc tính đó.
- Nếu khái niệm có thể có những kiểu định nghĩa chính xác (integer, string, float,...) đó là thuộc tính, không là class.
- Cố gắng xác định kiểu của các thuộc tính(integer, string, float ...).
- Cố gắng xác định phạm vi, giá trị, độ chính xác, các lớp liên quan.

Thu thập tất cả các thông tin về mỗi thuộc tính: tên, tên khái niệm liên quan đến thuộc tính, loại giá trị, phạm vi, giá trị.

Mô tả một vài vấn đề chung của thuộc tính lớp:

-Thuộc tính số lượng giá trị (Slot cardinality): định nghĩa số lượng giá trị thuộc tính có thể có.

-Thuộc tính giá trị kiểu (Slot-value type): mô tả kiểu giá trị có thể điền vào thuộc tính (String, Number, Boolean ...), trong đó thuộc tính Instance cho phép xác định quan hệ giữa các thực thể. Thuộc tính Instance cũng xác định danh sách các lớp cho phép lấy các thể hiện từ đó.

Bước 6: Tạo các thể hiện (instances)

Để tạo các thể hiện cần thực hiện:

- Chọn lớp.
- Tạo thuộc tính của lớp.
- Tạo thể hiện và điền giá trị các thuộc tính.

3.4 Xây dựng tập từ gốc, xác định các từ, cụm từ

Phần này trình bày cách thực hiện phân tích nội dung để xác định tập từ gốc, cách sử dụng Gate UK để phân tích các ebook xác định các từ, cụm từ liên quan đến vấn đề.

Bước 1: Sưu tập các từ gốc:

Lựa chọn bộ từ gốc ban đầu cho chủ đề được thực hiện bằng tay dựa trên các khái niệm được tìm thấy trong các mô tả chủ đề, tiêu đề, tiêu đề phụ, chỉ tập trung vào khái niệm danh từ và quan hệ ngữ nghĩa của chúng. Sau đó tập từ gốc được tăng lên với khái niệm có mối quan hệ thứ bậc với các từ gốc. Các quan hệ Is-A, Part- Whole, đồng nghĩa là các quan hệ thường được sử dụng để làm tăng tập từ gốc.

Bước 2: Dùng Gate phân tích các ebook, phối hợp sử dụng plugin ANNIE và JAPE để phân tích, xác định các từ, cụm từ liên quan

- Xác định các đặc trưng của vấn đề cần được xác định.
- Xây dựng các luật JAPE để phân tích, xác định các từ, cụm từ liên quan.
- Lựa chọn các thuật ngữ cần dùng.

Bước 3: Tạo danh sách và/hoặc sơ đồ của tất cả từ, cụm từ đã chọn_ điều này rất quan trọng. Đối với từng thuật ngữ cố gắng viết ra: tên, mô tả, kiểu, nguồn gốc.

3.5 Nhận dạng và tạo quan hệ ngữ nghĩa

Phần này trình bày cách thực hiện áp dụng trong thực nghiệm.

Để nhận dạng và tạo quan hệ ngữ nghĩa ngoài việc phân tích thủ công (tham khảo ý kiến chuyên gia), chúng ta có thể thực hiện các bước:

Từ cơ sở dữ liệu cho tập các thuật ngữ đã phân tích được, chọn ra các thuật ngữ mô tả các đối tượng có sự tồn tại độc lập. Những thuật ngữ này sẽ là lớp trong bản thể học và sẽ trở thành neo trong hệ thống phân cấp lớp.

Bước 1: Từ tập các thuật ngữ đã xác định trên, xem xét các quan hệ IS-A

- Quan hệ IS-A liên kết một khái niệm WordNet và 1 khái niệm c trích xuất từ văn bản, khái niệm c được liên kết vào WordNet và thêm vào hệ thống phân cấp.

- Quan hệ bao hàm (hypernymy) liên kết một khái niệm từ gốc và khái niệm không từ gốc tìm được trong văn bản. Vì vậy khái niệm không từ gốc được thêm vào danh sách nhưng chưa liên kết với thành phần khác

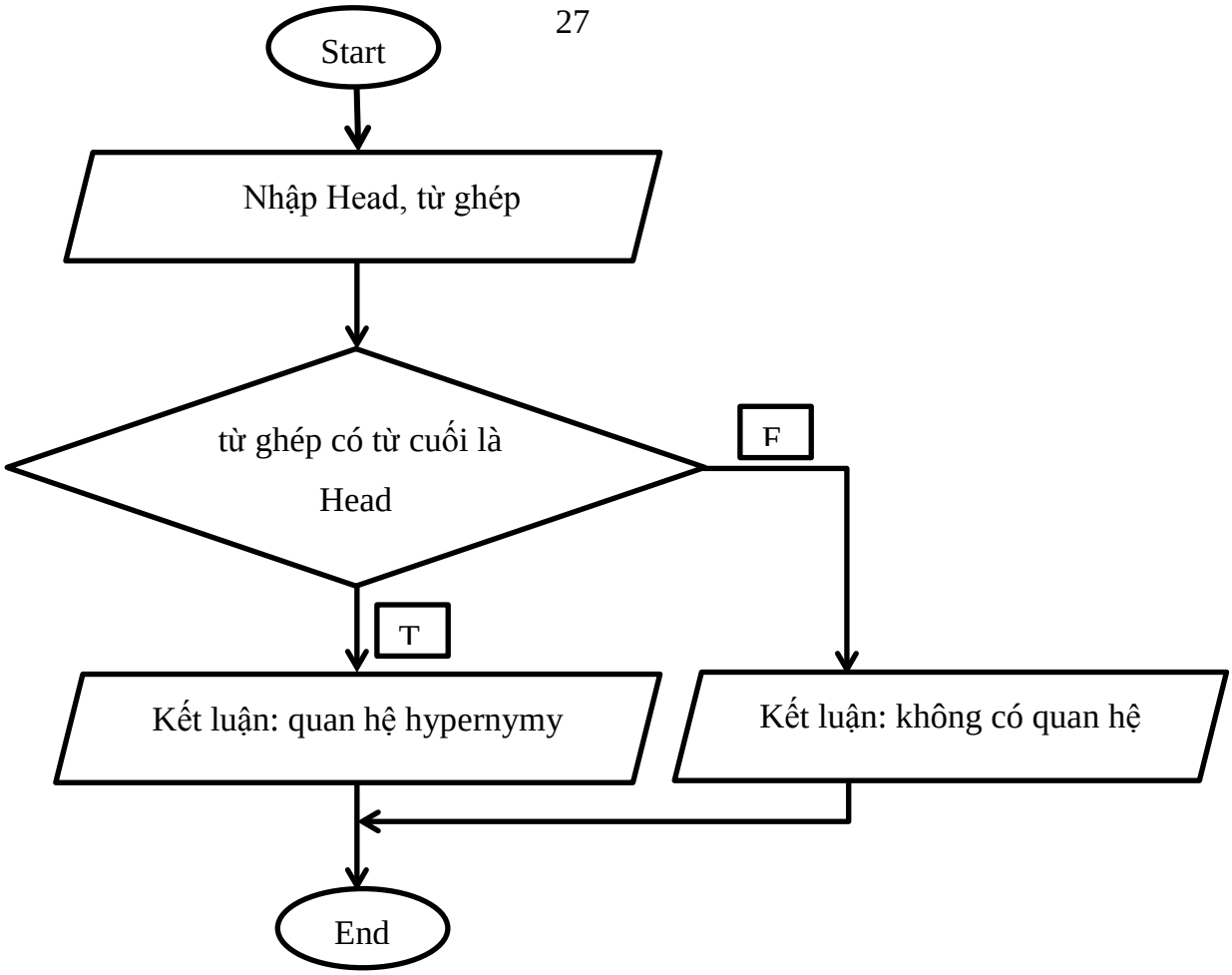
Bước 2: Sử dụng các hệ thống dữ liệu thu được ở bước một chạy các thủ tục sau đây trên các khái niệm mà không liên kết trực tiếp hoặc gián tiếp đến WordNet

Thủ tục 1: ***Phân loại một khái niệm về hình thức [word, head] liên quan đến khái niệm [head]*** .

Ở đây, chúng ta chỉ xem xét những head là danh từ / tính từ mà không có bất kỳ quan hệ thuộc cấp hyponyms.

Ý tưởng của thuật toán: dựa trên khái niệm từ ghép [word, head], từ ghép [word, head] được bao hàm bởi khái niệm [head].

Ví dụ, checking account **is a kind of** account, do đó liên kết bởi một mối quan hệ hypernymy (account, checking account).

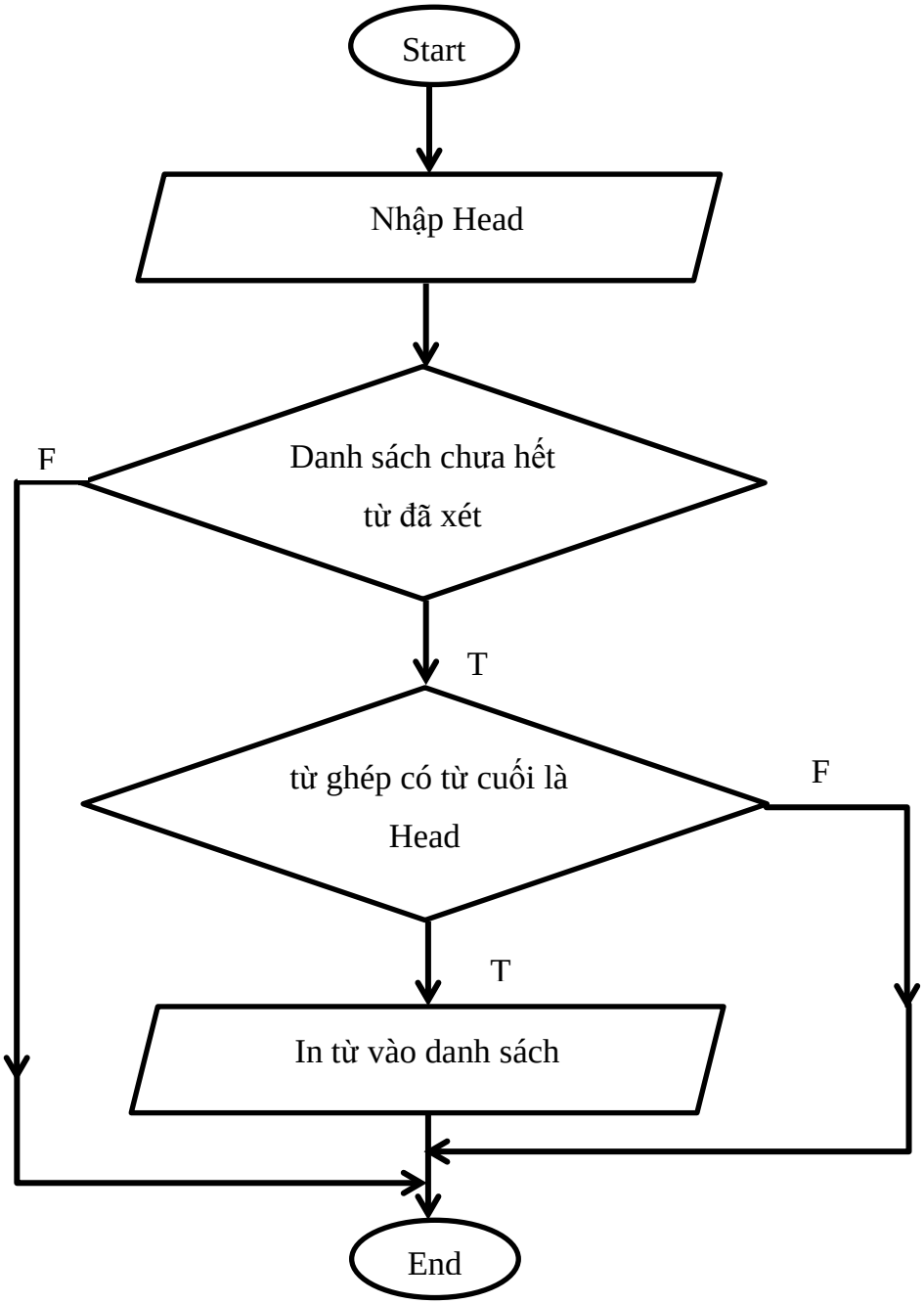


Hình 3-5: Lưu đồ thuật toán Thủ tục 1.

Với cách thực hiện này khi lượng dữ liệu lớn cần phải tìm từng từ trong danh sách và tiến hành kiểm tra nhiều lần, điều này dẫn đến tình trạng có thể bỏ sót thông tin. Để tránh tình trạng phải tìm từng cụm từ trong danh sách sau đó thực hiện kiểm tra luận văn đề xuất cải tiến thủ tục 1 như sau:

Cải tiến Thủ tục 1:

Ý tưởng: kiểm tra tất cả các từ trong danh sách, từ ghép nào thoả điều kiện có từ cuối bên phải là Head, xuất từ đó vào danh sách kết quả.

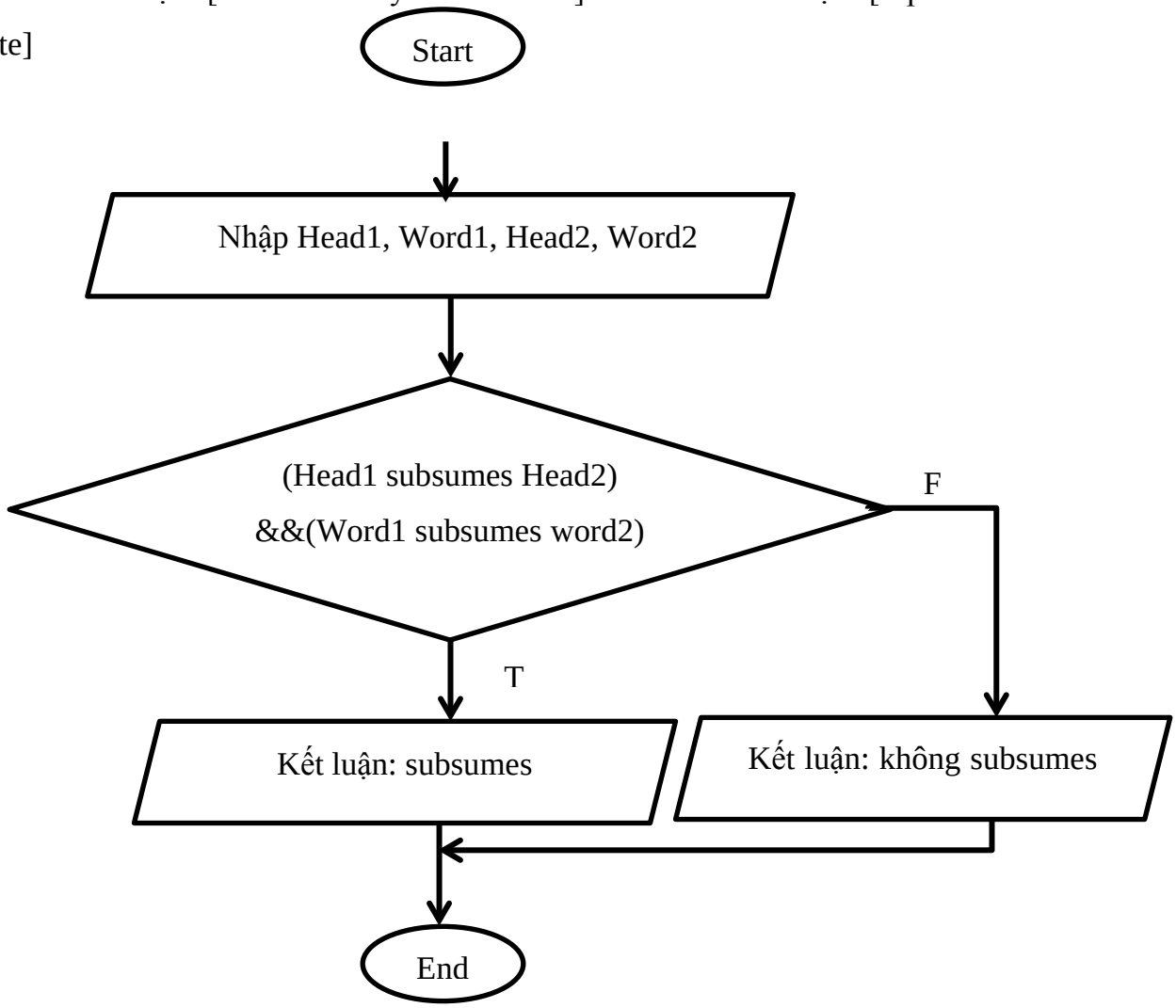


Hình 3-6: Lưu đồ thuật toán Thủ tục 1 Cải tiến

<u>Thủ tục 2</u>: ***Phân loại mối quan hệ khái niệm [word1, head 1] liên quan đến khái niệm[word2, head2]***

Nếu head1 bao hàm (subsumes) head2 và word1 bao hàm word2, khi đó ta có [word1, head1] bao hàm [word2, head2]

Ví dụ: [Asian country] bao hàm [Japan] và [interestrate] bao hàm [discount rate] vì thế khái niệm [Asian country interest rate] bao hàm khái niệm [Japan discount rate]

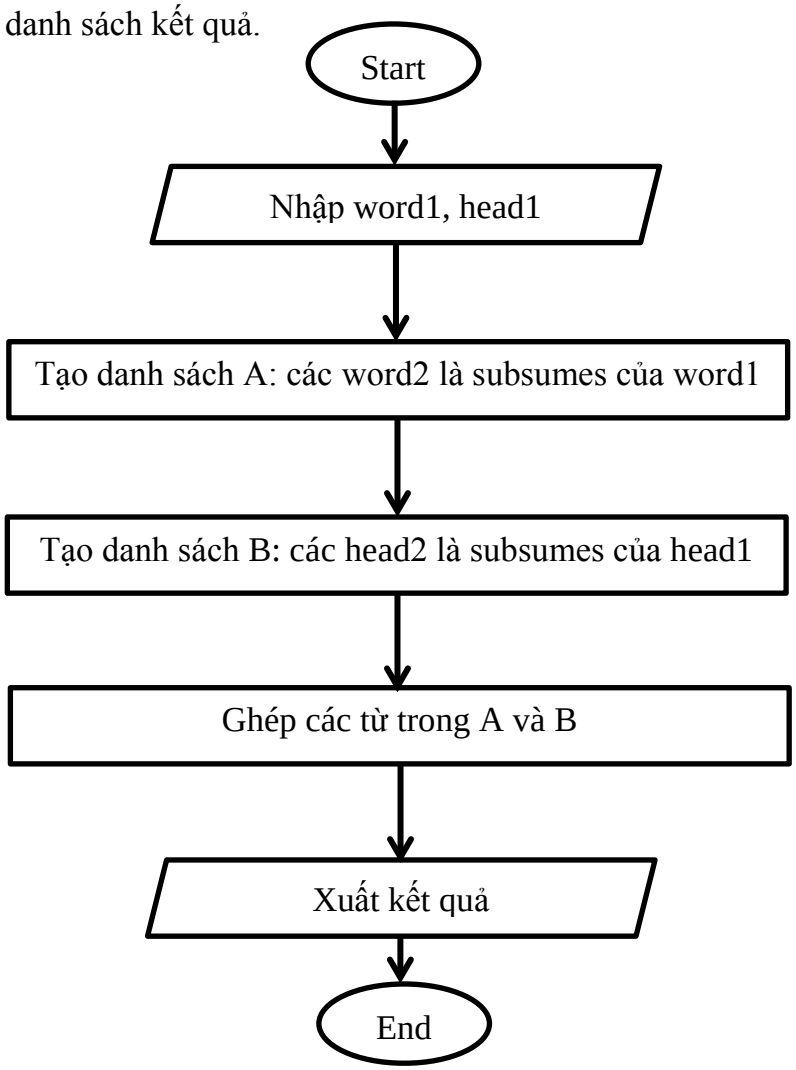


Hình 3-7: Lưu đồ thuật toán Thủ tục 2.

Với cách thực hiện này khi lượng dữ liệu lớn cần phải tìm từng từ trong danh sách và tiến hành kiểm tra nhiều lần, điều này dẫn đến tình trạng có thể bỏ sót thông tin và mất nhiều thời gian. Để tránh tình trạng phải tìm từng cụm từ trong danh sách sau đó thực hiện kiểm tra luận văn đề xuất cải tiến thủ tục 2 như sau:

Cải tiến thủ tục 2:

Ý tưởng: tạo danh sách A gồm các word2 là subsumes của word1, danh sách B gồm các head2 là subsumes của head1, ghép từng word2 và head2 của A và B, xuất kết quả vào danh sách kết quả.



Hình 3-8: Lưu đồ thuật toán Thủ tục 2 Cải tiến.

• **Bước 3:** Thêm các loại quan hệ khác IS-A vào cơ sở kiến thức mới. Quan hệ IS-A đã được sử dụng trong việc hình thành hệ thống phân cấp, nhưng các loại liên quan khác như Nguyên nhân (Cause), bộ phận_toàn thể (Part_Whole), ảnh hưởng (Influence) ... cũng cần được bổ sung vào cơ sở tri thức.

3.6 Cách tạo bản thể học bán tự động

Hiện nay để tạo ontology có nhiều công cụ tuy nhiên đề tài này chọn công cụ Protégé 4.3, vì hiện nay đây là công cụ phổ biến.

Sau bước phân tích các từ cụm từ tìm được. Sau khi thực hiện phân tích mối liên hệ giữa các từ, cụm từ. Trình tự thực hiện:

- Tạo các class theo hệ thống phân cấp đã phân tích trên.
- Tạo các thuộc tính của class.
- Xây dựng mối quan hệ giữa các class.

Chương 4. THỰC NGHIỆM

Chương này trình bày cách thực hiện và kết quả quá trình thực nghiệm.

- Toàn bộ quá trình thực nghiệm được thực hiện trên laptop cá nhân, cấu hình:
  - CPU Intel Core I5
  - RAM 4GB
  - HDD 500GB
  - Hệ điều hành WIN 8
- Các phần mềm, công cụ sử dụng trong quá trình thực nghiệm:
  - WordNet 2.1.
  - GATE Developer 8.0.
  - Protégé 4.3.0
  - Visual Studio 2013(lập trình bằng ngôn ngữ C#).
  - SQL Server.

4.1 Xác định lĩnh vực và phạm vi của bản thể học

Để minh hoạ các bước thực hiện tạo bản thể học, luận văn xác định :

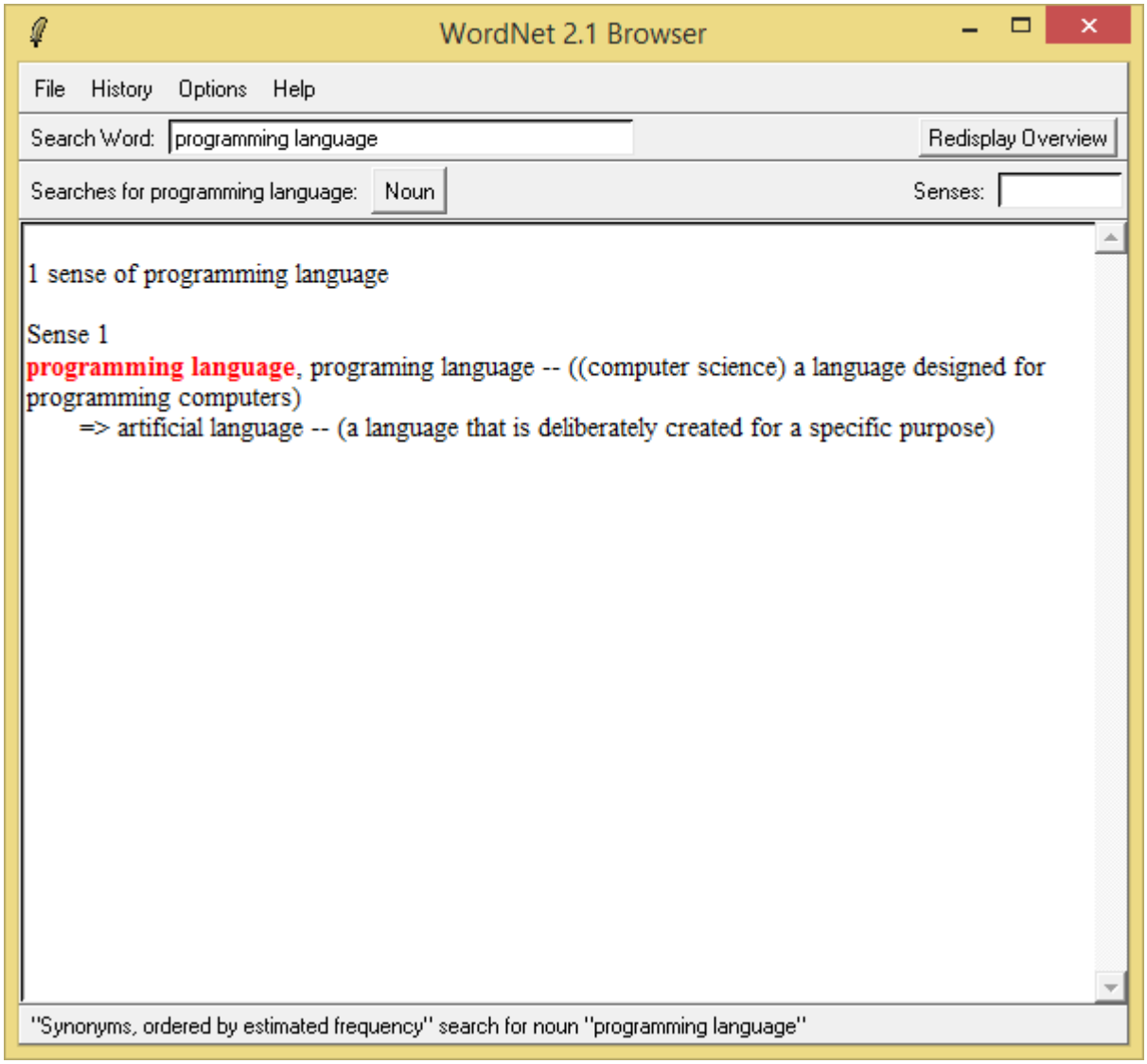
- Phạm vi cụ thể của bản thể học là Programing Language (ngôn ngữ lập trình) bao gồm các loại ngôn ngữ, các kiểu dữ liệu cơ bản của ngôn ngữ, các loại cấu trúc điều khiển của ngôn ngữ.
- Nguồn tài nguyên đề tài sử dụng là các ebook về ngôn ngữ lập trình (bằng tiếng Anh), và tham khảo ý kiến chuyên gia.

Hiện nay bản thể học về lĩnh vực ngôn ngữ lập trình chưa nhiều, trong quá trình thực hiện đề tài chúng tôi tham khảo tài liệu [8], tuy nhiên vẫn không tìm được nguồn tài nguyên nên đề tài phải xây dựng một bản thể học mới.

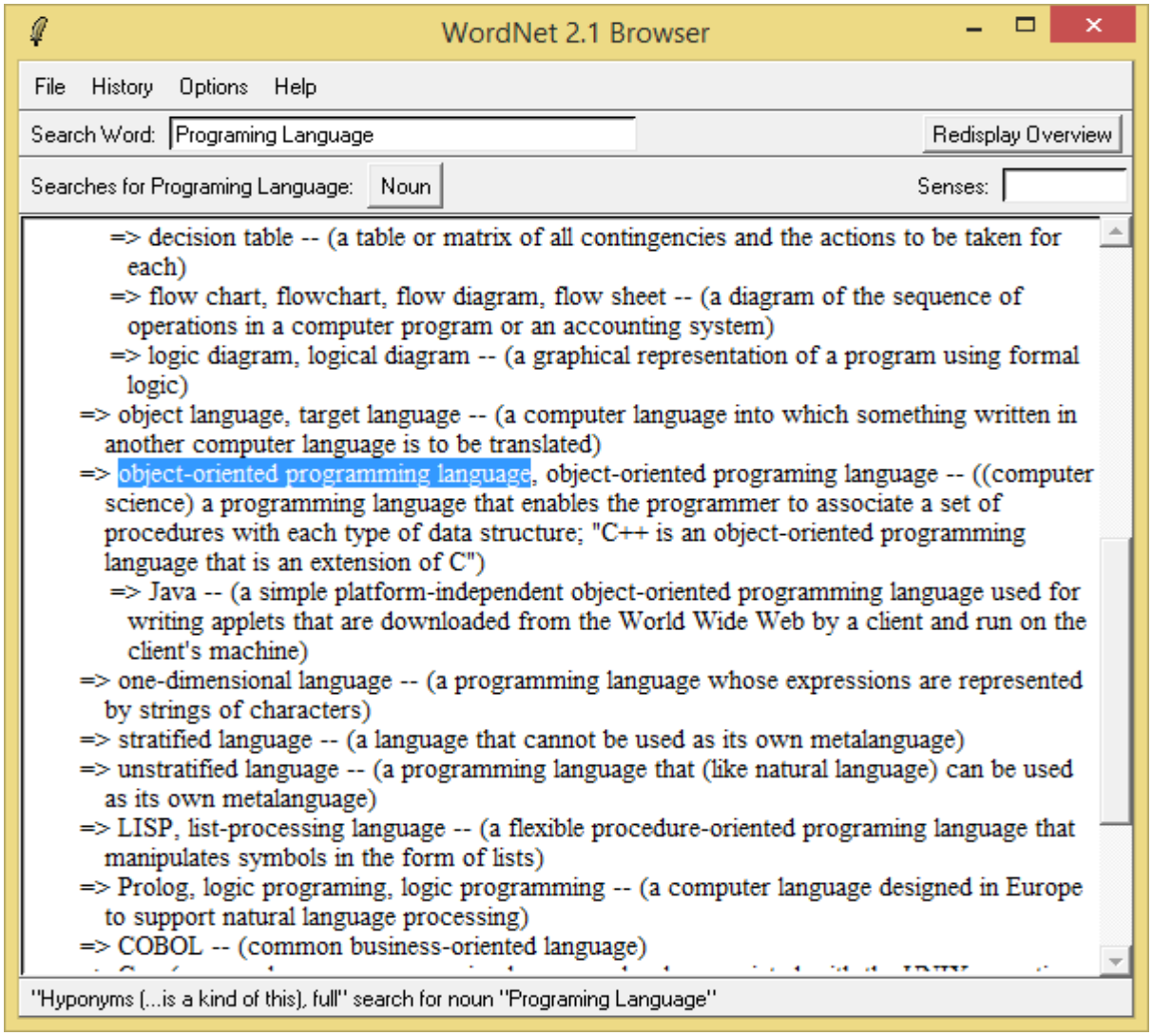
4.2 Xây dựng tập từ gốc, xác định các từ, cụm từ

4.2.1 Sưu tập các từ gốc

Dựa vào mô tả ý tưởng chính của chủ đề “Programming Language” từ gốc đầu tiên được chọn là “Programming Language”, kết hợp WordNet và phân tích thủ công, kết quả có được như sau:



Hình 4-1: Sử dụng WordNet xác định từ đồng nghĩa với từ gốc đầu tiên.



Hình 4-2: Sử dụng WordNet xác định từ có quan hệ Is_A với từ gốc đầu tiên

Bảng 4-1: Danh sách các từ tìm được.

Từ đồng nghĩa	**Quan hệ Is_A**	**Phân tích thủ công**
artificial language	algorithmic language	Procedure programming language
	assembly language	Structure programming language
	computer language	Data type
	multidimensional language	Control structure
	object-oriented programming language	programming language type
	one-dimensional language	Object-oriented programming language.
	logic programming	Logic programming language.
	C, Pascal, LISP, COBOL, BASIC	
	computer-oriented language	
	machine language	
	machine-oriented language	
	decision table	
	flowchart, flow diagram, flow sheet	
	one-dimensional language	
	stratified language	
	unstratified language	

Các từ gốc được chọn:

- Programming language.
- Programming language type.

- Procedure programming language.
- Object-oriented programming language.
- Logic programming language.
- Structure programming language.
- Data type.
- Control structure.

Đánh giá kết quả:

Số từ xác định được : 30

Số từ được chọn làm từ gốc: 8

Tỷ lệ 8/30 = 26.67%

4.2.2 Xác định các từ, cụm từ liên quan

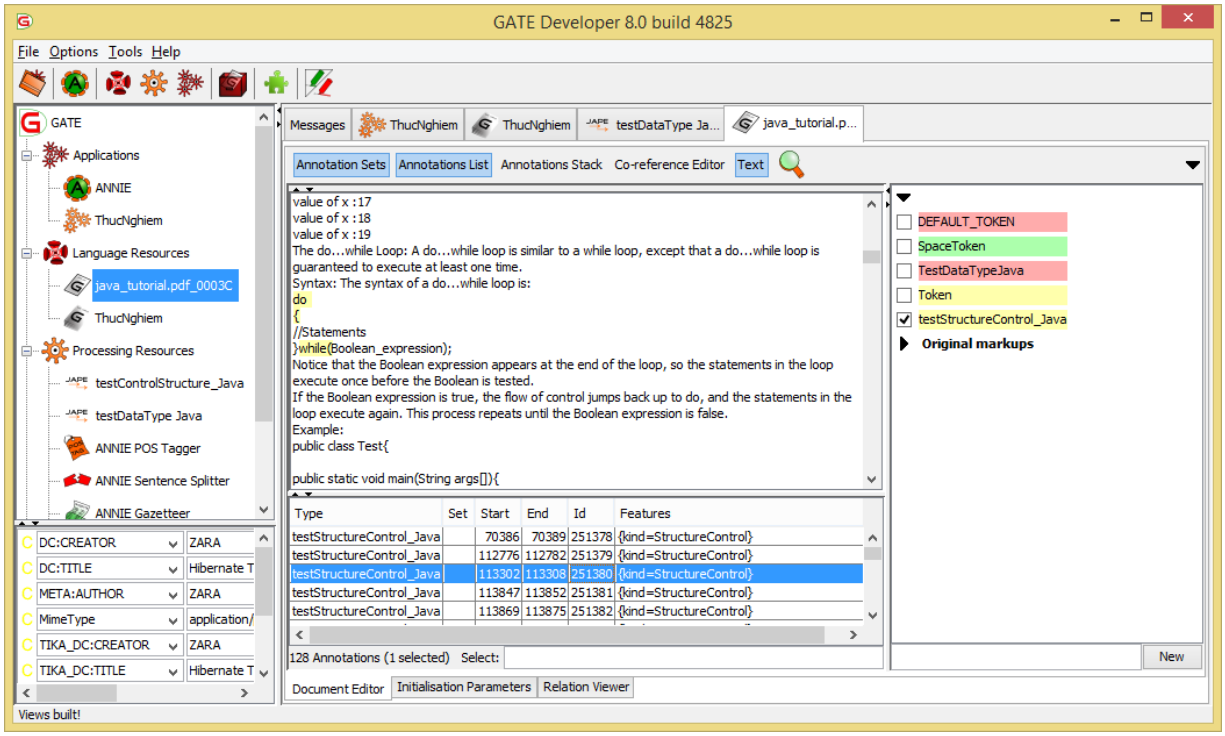
Phần này sử dụng công cụ Gate UK thực hiện việc phân tích các ebook được chọn để xác định các từ, cụm từ liên quan. Để thực hiện đề tài sử dụng các plugin trong ANNIE, kết hợp các luật JAPE để phân tích ebook, xác định các từ, cụm từ theo đặc trưng được xác định.

4.2.2.1 Xác định đặt trưng của các từ, cụm từ liên quan

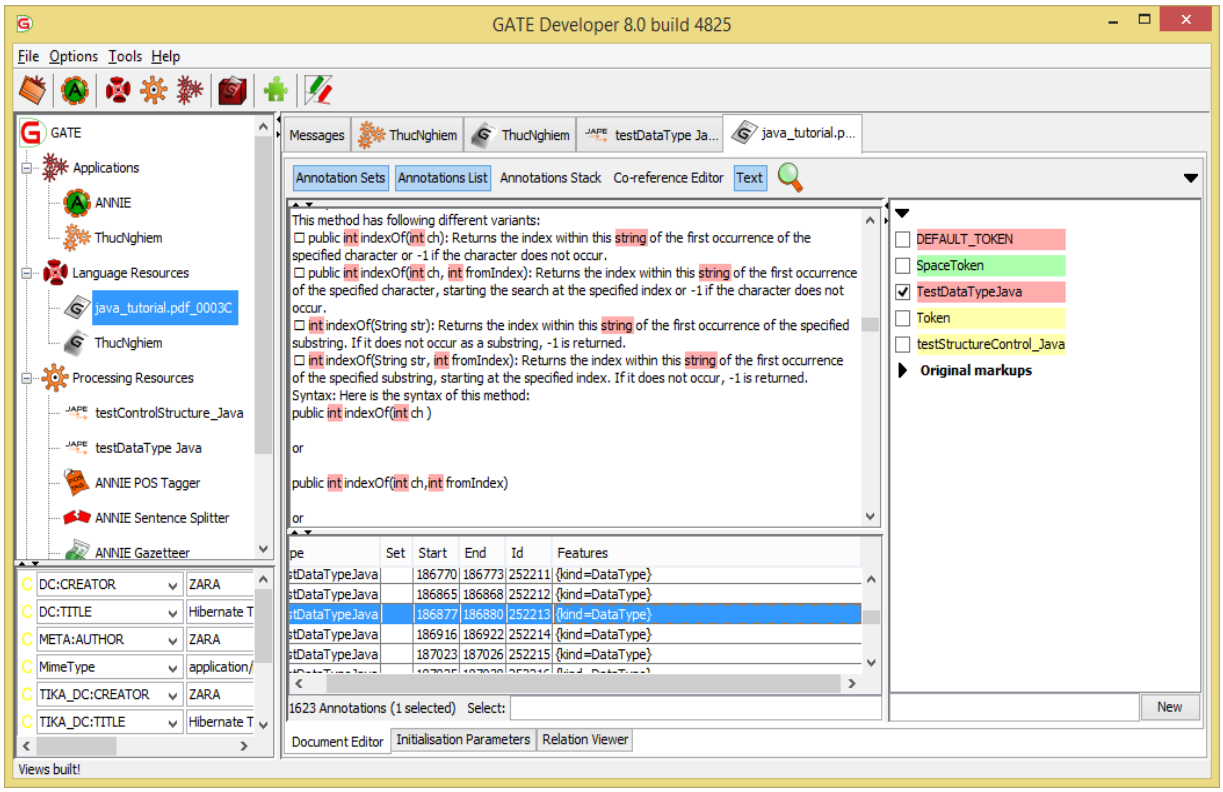
- Control structure trong ngôn ngữ C, C++, Java, C# là cụm từ bắt đầu bằng if, for, while, switch và “(“ hoặc bắt đầu bằng “do” và “{“.
- Control structure trong ngôn ngữ Pascal là cụm từ bắt đầu bằng IF, FOR, WHILE, CASE, REPEAT.
- Control structure trong ngôn ngữ VisualBasic, VB.net là cụm từ bắt đầu bằng If, For, While, Select, Do
- Data type gồm int, long, ...
- Dựa vào các đặt trưng trên xây dựng luật JAPE.

4.2.2.2 Xây dựng luật JAPE

- Dùng plugin ANNIE, JAPE trong Gate phân tích các ebook để xác định các từ, cụm từ liên quan đến khái niệm “ control structure” và “data type”.



Hình 4-3: Kết quả phân tích “structure control”.



Hình 4-4: Kết quả phân tích "data type".

- Tham khảo ý kiến chuyên gia, phân tích thủ công cho các trường hợp còn lại.

4.2.3 Kết quả đạt được

Bảng 4-2: Bảng các từ, cụm từ xác định được.

Các từ, cụm từ gốc	***Các từ, cụm từ xác định được***
programming language	programming language type
	Control structure
	data type
programming language type	procedure programming language
	object-oriented programming language
	logic programming language
	structure programming language
Các từ, cụm từ gốc	***Các từ, cụm từ xác định được***
procedure programming language	C
	Pascal
	VB6
object-oriented programming language	C++
	C#
	Java
	VB.net
logic programming language	Prolog
	LISP
structure programming language	Ada
	dBase
Control structure	loop statement
	decision statement

Decision statement	if
	if else
	if elseif else
	if then
	if then else
	if then elseif else
	If Then EndIf
	If Then Else EndIf
	If Then Elseif Else EndIf
	switch
	case
	case else
Các từ, cụm từ gốc	***Các từ, cụm từ xác định được***
Decision statement	Select Case
Loop statement	Do Loop Until
	Do Loop While
	Do Until Loop
	Do While Loop
	do while
	while do
	while
	While End While
	repeat until
	for
	foreach
	For Next
	for do
	Do Loop Until

Data type	bool
	boolean
	Byte
	byte in java C#
	char
	char in CSharp Java
	Currency
	Date
	decimal
	Decimal in VB.net
Các từ, cụm từ gốc	***Các từ, cụm từ xác định được***
Data type	Double
	Extended
	Comp
	double
	float
	int
	int in Csharp Java
	Integer
	Long
	long
	long in Csharp Java
	long double
	Object
	sbyte
	short
	Shortint
	signed char

	Signed in VB
	Single
	String
	uint
	Uinteger
	ulong
	unsigned char
	unsigned int
Các từ, cụm từ gốc	***Các từ, cụm từ xác định được***
Data type	unsigned long
	ushort
	void
	Variant
	Real
	Word

Bảng 4-3: Bảng thống kê số lượng từ xác định được.

ebook sử dụng	VB.net	VB	C	C++	C#	Java	Pascal	Tổng
Số cụm từ xác định được	301	232	617	830	776	1751	669	5176
Số cụm từ đúng	223	165	413	413	511	1017	480	3281
Số cụm từ được chọn	22	22	18	19	22	18	17	68/138

****Đánh giá kết quả***

Theo [13], để đánh giá kết quả cách khai thác thông tin đề tài sử dụng độ đo Precision, Recall và F-measure. Với cách mở rộng từ vựng sử dụng Gate, vì mỗi từ được lặp lại nhiều lần nên phải xác định các từ trùnh nhau chỉ chọn một từ.

Precision (P) xác định tỷ lệ phần trăm của từ có ý nghĩa đúng so với những từ tìm được

$$P = \frac{\text{Số từ có ý nghĩa đúng}}{\text{Số từ tìm được}} \qquad (4.1)$$

Recall (R) xác định tỷ lệ số từ có ý nghĩa đúng so với danh sách từ đúng

$$\boxed{\text{P} = \frac{\text{Số từ có ý nghĩa đúng}}{\text{Số từ trong danh sách đúng}}} \quad (4.2)$$

$$P = \frac{68}{138} = 49.27\%$$

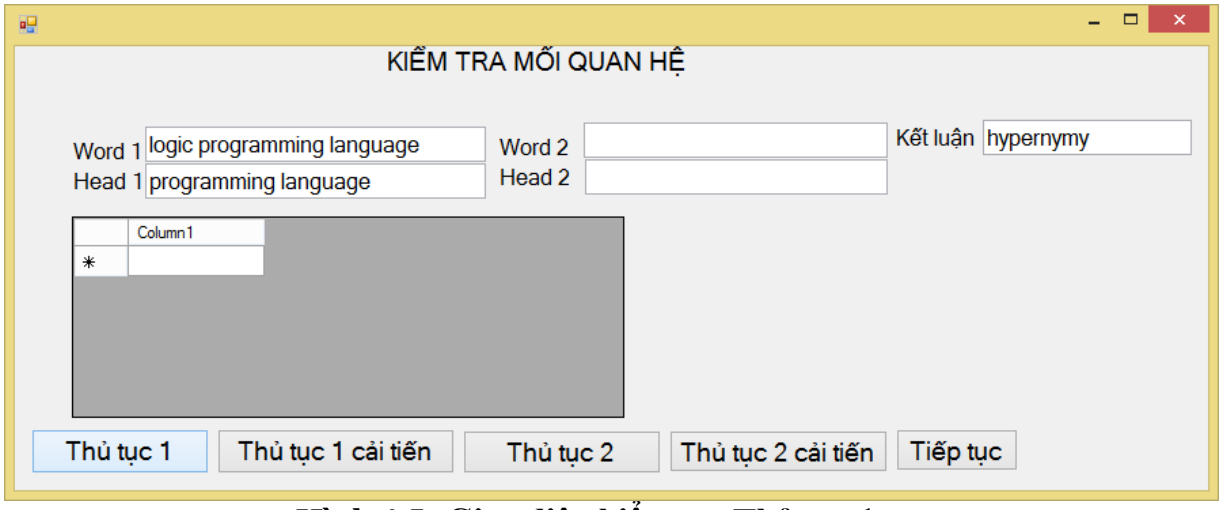
$$R = \frac{68}{88} = 77.27\%$$

Tỷ lệ từ có ý nghĩa đúng / số từ trong danh sách đúng rất thấp vì lý do trong từng tài liệu mỗi từ được lặp lại rất nhiều lần trong các ví dụ minh hoạ.

4.3 Nhận dạng quan hệ ngữ nghĩa

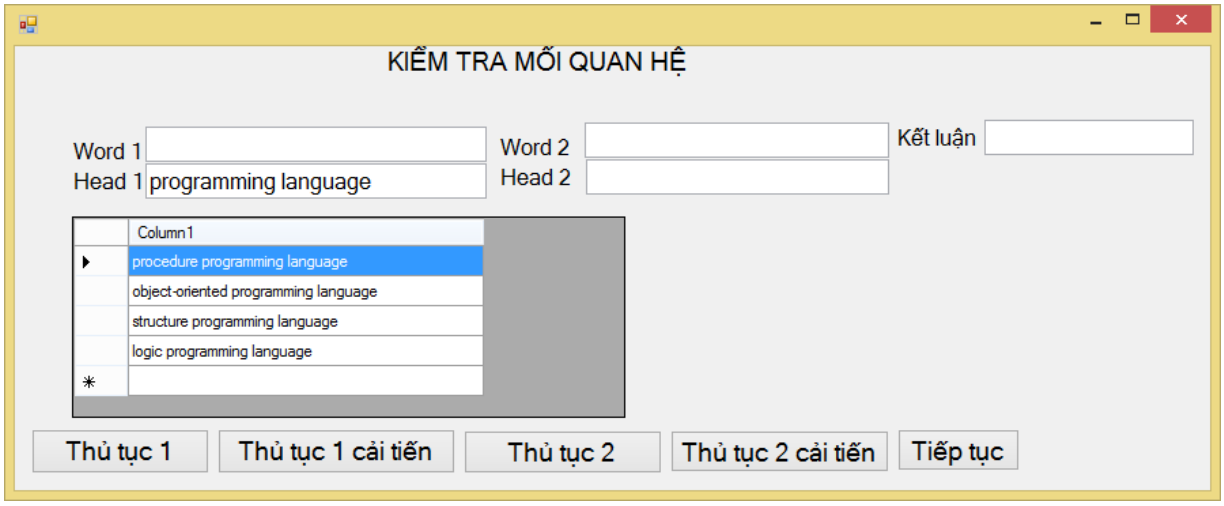
Để kiểm tra các thủ tục trong báo cáo, đề tài thực hiện xây dựng cơ sở dữ liệu các từ, cụm từ xác định được trong SQL server và xây dựng giao diện xử lý trong C#

- **Nút lệnh Thủ tục 1**: thực hiện kiểm tra 1 cụm từ được nhập trong textbox Word1 có quan hệ hypernymy với từ, cụm từ được nhập trong textbox Head1 hay không, và xuất kết quả ”hypernymy” hoặc ”không hypernymy” vào textbox Kết luận.
  - Cơ chế thực hiện nút lệnh Thủ tục 1**:**
    - Trích chuỗi bên phải của cụm từ trong textbox Word1 có số ký từ bằng số ký tự chuỗi nhập trong textbox Head1
    - So sánh với từ, cụm từ trong textbox Head1 nếu trùng khớp kết luận “hypernymy“.
  - Kết quả khi thực thi Thủ tục 1:



Hình 4-5: Giao diện kiểm tra Thủ tục 1.

- **Nút lệnh Thủ tục 1 cải tiến**: thực hiện xuất tất cả các cụm từ có quan hệ hypernymy với từ, cụm từ được nhập trong textbox Head1 vào bảng.
  - Cơ chế thực hiện Thủ tục 1 cải tiến:
    - Xuất tất cả cụm từ trong danh sách vào bảng tạm
    - Ứng với mỗi cụm từ, trích chuỗi bên phải với số ký tự bằng chiều dài chuỗi nhập trong textbox Head1
    - So sánh với từ nhập trong textbox Head1 nếu từ nào có kết quả trùng khớp xuất vào bảng kết quả.
  - Kết quả khi thực thi Thủ tục 1 Cải tiến:



Hình 4-6: Giao diện kiểm tra Thủ tục 1 Cải tiến.

Với Thủ tục 1: mỗi lần thực thi chỉ kiểm tra được một cụm từ có sẵn có thoả điều kiện có từ cuối bên phải là "Head" hay không. Sau đó việc xác định cụm từ có phù hợp thực tế hay không được xác định thủ công.

Với Thủ tục 1 Cải tiến mỗi lần thực thi tương ứng một giá trị head sẽ tìm được nhiều từ ghép thoả điều kiện có từ cuối trong cụm từ là head. Việc kiểm tra các cụm từ trong danh sách có phù hợp thực tế hay không sẽ được thực hiện thủ công. Tuy nhiên bằng cách này, mỗi bước thực hiện số lượng từ xác định được cao hơn so với Thủ tục 1 và tránh được khả năng bỏ sót từ.

Bảng 4-4: Bảng thống kết quả Thủ tục 1 Cải tiến.

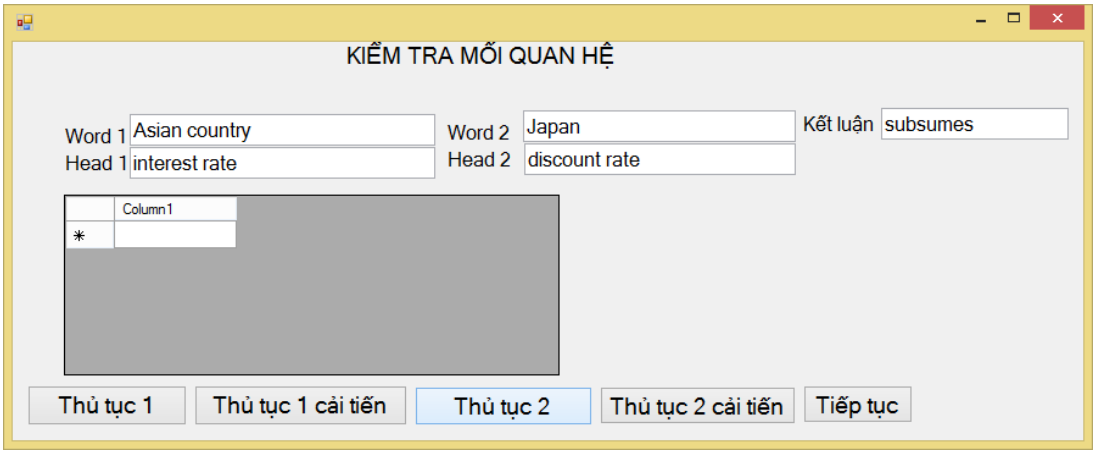
Head	Số từ xác định được	Số từ đúng	Số từ được chọn
programming language	4	4	4
Loop statement	13	13	13
Decision statement	14	14	13
Tổng	31	31	30

Áp dụng công thức tính precision (P) và recall(R) :

$$P = \frac{30}{31} = 96.77\%$$

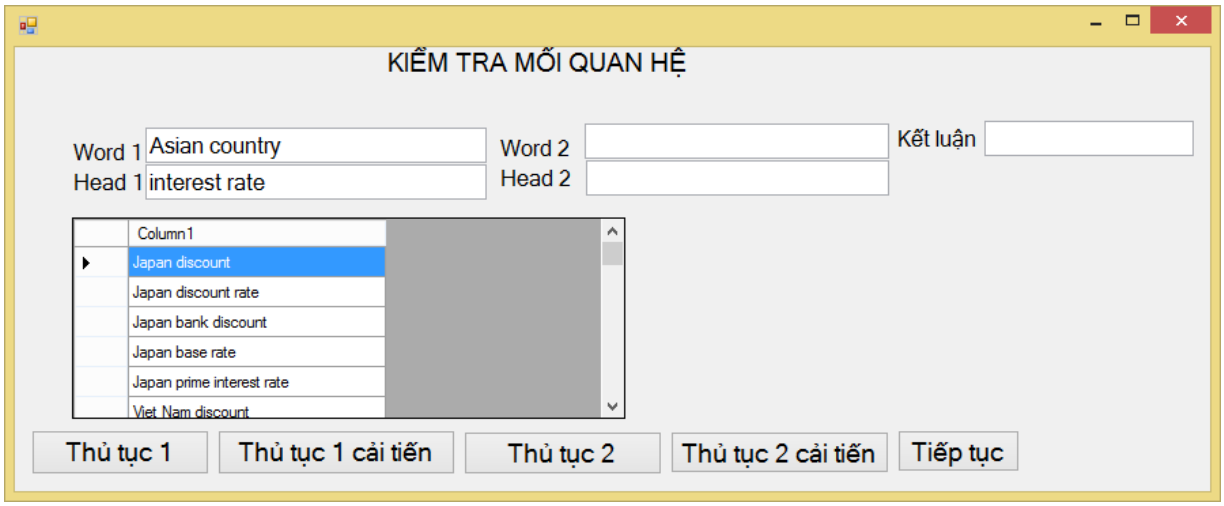
$$R = \frac{30}{30} = 100\%$$

- **Nút lệnh Thủ tục 2**: thực hiện kiểm tra 1 cụm từ được ghép bởi từ hoặccụm từ nhập trong textbox Word2_ Head2 có quan hệ subsumes với cụm từ được ghép bởi từ/ cụm từ nhập trong textbox Word1_ Head1 hay không , và xuất kết quả ” subsumes” hoặc ” không subsumes” vào textbox Kết luận.
  - Cơ chế thực hiện Thủ tục 2:
    - ➢ Xuất danh sách các từ, cụm từ là subsumes của từ, cụm từ nhập trong textbox Word1 vào bảng phụ 1.
    - ➢ Xuất danh sách các từ, cụm từ là subsumes của từ, cụm từ nhập trong textbox Head1 vào bảng phụ 2.
    - ➢ So sánh từ, cụm từ nhập trong textbox Word2 với các từ, cụm từ trong bảng phụ 1
    - ➢ So sánh từ, cụm từ nhập trong textbox Head2 với các từ, cụm từ trong bảng phụ 2
    - ➢ Nếu cả 2 phép so sánh trên cùng cho kết quả đúng kết luận “subsumes” trong textbox Kết luận, các trường hợp khác kết luận “không subsumes”
  - Kết quả khi thực thi Thủ tục 2:



Hình 4-7: Giao diện kiểm tra Thủ tục 2.

- **Nút lệnh Thủ tục 2 Cải tiến**: thực hiện xuất vào danh sách kết quả các cụm từ thoả điều kiện có quan hệ subsumes với cụm từ được ghép bởi từ, cụm từ nhập trong textbox Word1_ Head1.
  - Cơ chế hoạt động Thủ tục 2 cải tiến:
    - ➢ Xuất danh sách các từ, cụm từ là subsumes của từ, cụm từ nhập trong textbox Word1 vào bảng phụ 1.
    - ➢ Xuất danh sách các từ hoặc cụm từ là subsumes của từ hoặc cụm từ nhập trong textbox Head1 vào bảng phụ 2.
    - ➢ Ghép mỗi từ trong bảng phụ 1 với tất cả từ trong bảng phụ 2.
  - Kết quả khi thực thi Thủ tục 2 Cải tiến:



Hình 4-8: Giao diện kiểm tra Thủ tục 2 Cải tiến.

Với Thủ tục 2: mỗi lần thực thi chỉ kiểm tra được một cụm từ có sẵn (word2, haed2) có thoả điều kiện được xếp vào nhóm (word1, haed1) hay không. Sau đó việc xác định cụm từ có phù hợp thực tế hay không được xác định thủ công.

Với Thủ tục 2 Cải tiến, tương ứng 1 bộ giá trị (word1, head1) chúng ta sẽ tìm được danh sách nhiều (word2, head2). Từ danh sách (word2, head2) việc chọn ra các từ phù hợp thực tế sẽ được thực hiện thủ công. Bằng cách thực hiện này mỗi bước thực hiện khả năng xác định được nhiều từ hơn so với Thủ tục 2 nên tránh được trường hợp bỏ sót từ.

Bảng 4-5: Bảng các lớp

Lớp cha(super class)	Lớp con(sub class)
programming language	programming language type
	Control structure
	data type
Programing language type	procedure programming language
	object-oriented programming language
	logic programming language
	structure programming language
Control structure	Loop statement
	decision statement

Bảng 4-6: Bảng các thể hiện.

Lớp	Thể hiện
Decision statement	if
	if else
	if elseif else
	if then
	if then else
	if then elseif else
	If Then EndIf
	If Then Else EndIf
	If Then Elseif Else EndIf
	switch
	case
	case else
	Select Case
Loop statement	Do Loop Until

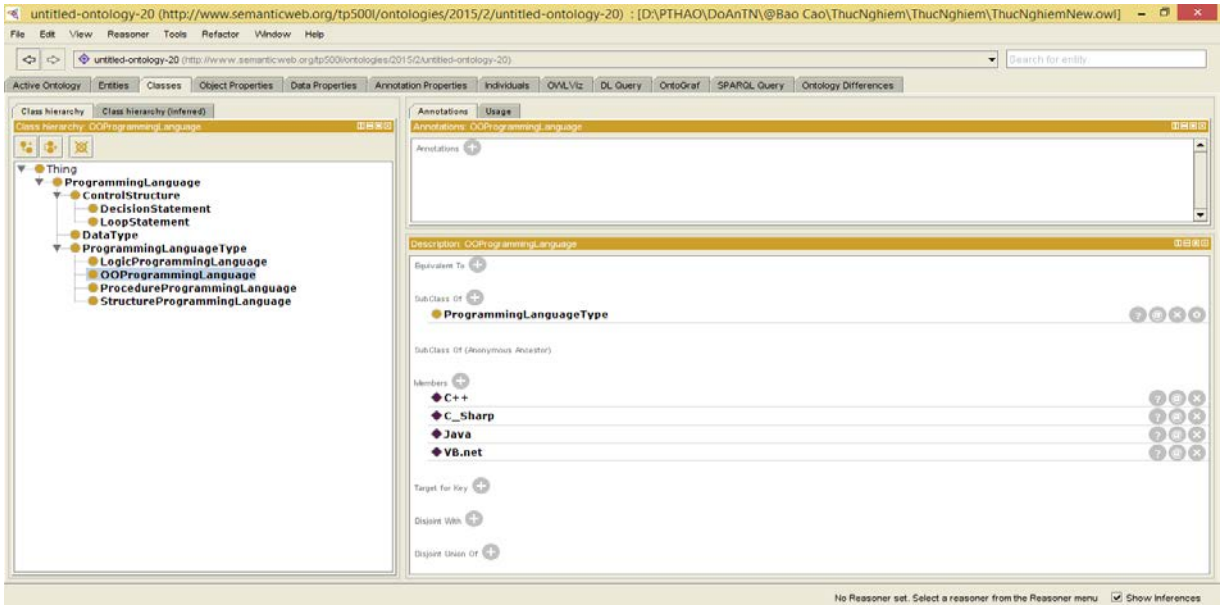
	Do Loop While
	Do Until Loop
	Do While Loop
	do while
	while do
	while
Lớp	**Thể hiện**
Loop statement	While End While
	repeat until
	for
	foreach
	For Next
	for do
Data type	bool
	boolean
	Byte
	byte in java C#
	char
	char in CSharp Java
	Currency
	Date
	decimal
	Decimal in VB.net
	Double
	Extended
	Comp
	double
	float

	int
	Int in Csharp Java
	Integer
	Long
Lớp	**Thể hiện**
Data type	long
	long in Csharp Java VB.net
	long double
	Object
	sbyte
	short
	Shortint
	signed char
	Signed in VB
	Single
	String
	uint
	Uinteger
	ulong
	unsigned char
	unsigned int
	unsigned long
	ushort
	void
	Variant
	Real
	Word
Logic programming language	Prolog

	LISP
Lớp	**Thể hiện**
Object Oriented programming language	C++
	C Sharp
	Java
	VB.net
Procedure programming language	C
	Pascal
	Visual Basic 6
Structure programming language	ADA
	dBase

4.4 Tạo bản thể học

4.4.1 Tạo các class theo hệ thống phân cấp đã phân tích trên.



Hình 4-9: Cây phân cấp lớp.

4.4.2 Tạo các thuộc tính của class, xây dựng mối quan hệ giữa các lớp

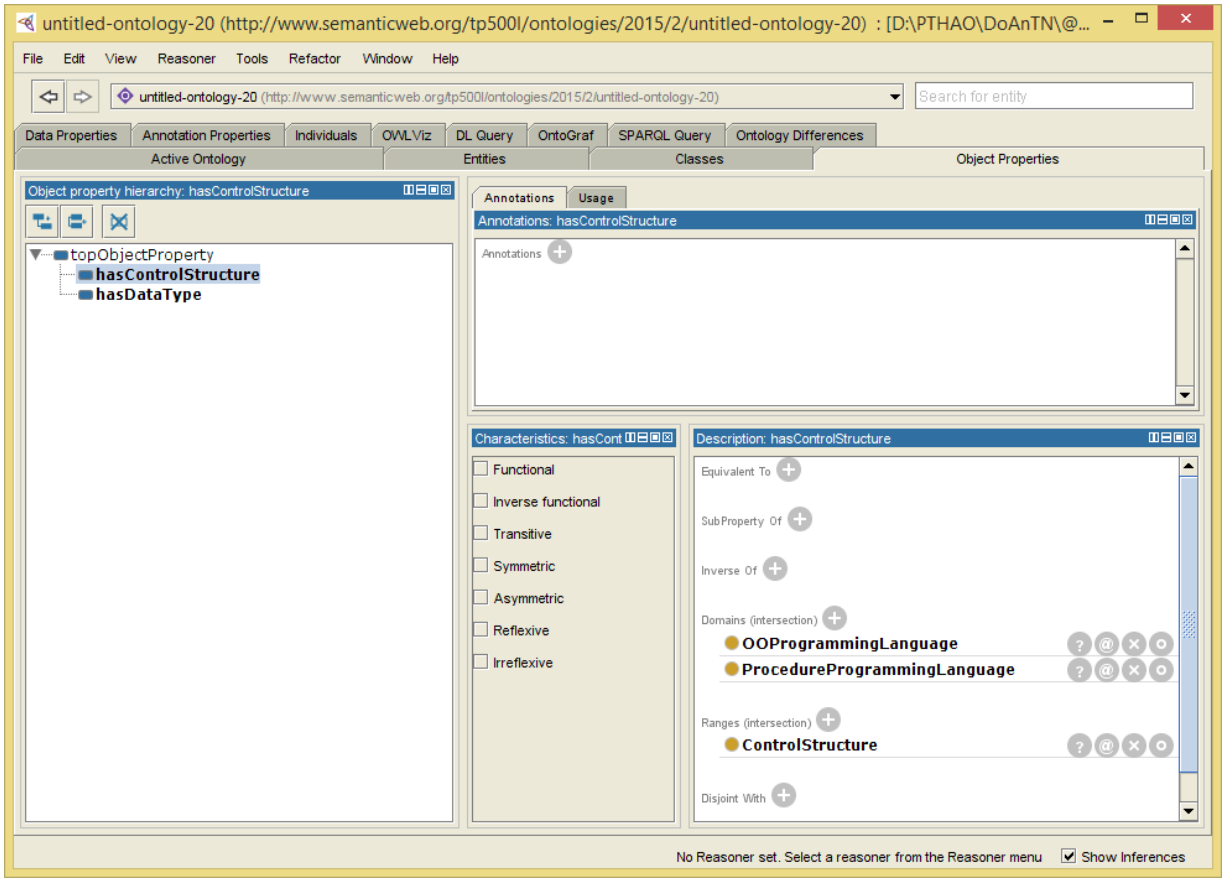
Phần này mô tả các thuộc tính của các class, đề tài tập trung xây dựng các class trong class OOProgrammingLanguage và ProcedureProgrammingLanguage.

Thuộc tính của lớp mô tả các tính chất của lớp, mối quan hệ giữa lớp này và lớp khác.

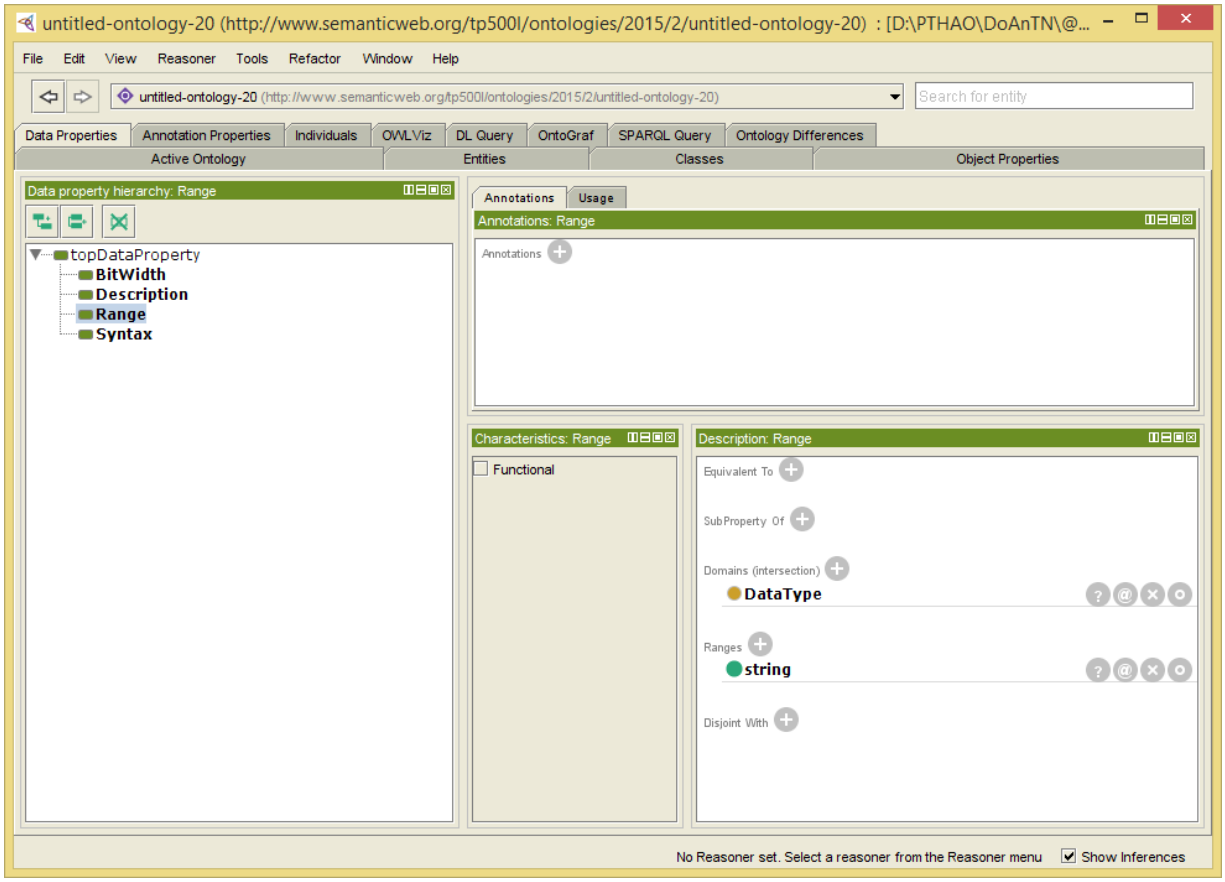
- Các lớp trong OOProgramming Language, Procedure Programming Language có thuộc tính hasControlStructure, hasDataType.
- Các thực thể trong ControlStructure có thuộc tính Syntax, Description.
- Các lớp trong DataType có thuộc tính BitWidth, Range.

Ý nghĩa các thuộc tính:

- Description: mô tả công dụng của từng ngôn ngữ lập trình hoặc công dụng của từng cấu trúc điều khiển.
- Syntax: mô tả cú pháp của cấu trúc điều khiển.
- BitWidth: mô tả độ rộng của kiểu dữ liệu.
- Range: mô tả phạm vi giá trị của kiểu dữ liệu
- hasControlStructure: mô tả quan hệ giữa từng ngôn ngữ lập trình và các cấu trúc điều khiển.
- hasDataType: mô tả quan hệ giữa từng ngôn ngữ lập trình và các kiểu dữ liệu cơ bản.

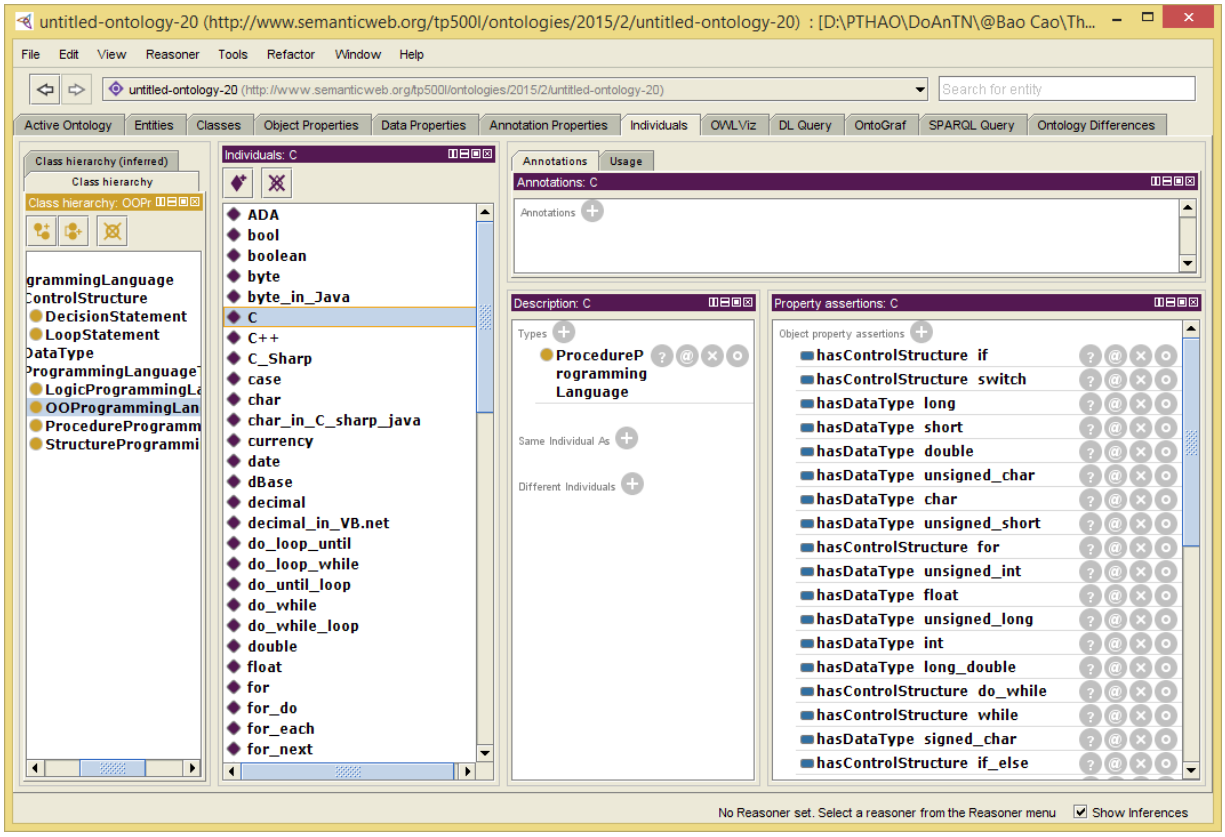


Hình 4-10: Tạo liên hệ giữa các lớp.



Hình 4-11: Tạo thuộc tính của lớp.

4.4.3 Tạo các thể hiện



Hình 4-12: Tạo các thể hiện.

Tóm lại, trong chương 4 chúng tôi đã trình bày các bước thực hiện để tạo bản thể học, các thủ tục để nhận dạng mối quan hệ giữa các từ, cụm từ, cách tạo bản thể học mà luận văn đã trình bày.

Chương 5. KẾT LUẬN VÀ KIẾN NGHỊ

Từ chương 1 đến chương 3, luận văn đã trình bày toàn bộ các tìm hiểu của chúng tôi về trình tự, cách thực hiện tạo một bản thể học về lĩnh vực cụ thể từ nguồn dữ liệu văn bản. Các chương đầu lần lược trình bày lý do tại sao đề tài được chọn để thực hiện, các nghiên cứu đã có liên quan đến việc tạo bản thể học, chương 3 trình bày cách thức thực hiện để tạo một bản thể học thuộc lĩnh vực cụ thể dựa trên nguồn dữ liệu văn bản, chương 4 trình bày phần thực hiện thực nghiệm của đề tài: xây dựng bản thể học Programing Language từ nguồn ngữ liệu là các ebook, với nguồn ngữ liệu là các dạng văn bản khác, cách thực hiện tương tự. Nội dung trình bày bám sát mục tiêu đề ra ban đầu của luận văn "X**ây dựng bản thể học từ kho ngữ liệu dạng văn bản**".

5.1 Kết quả đạt được

Từ ý tưởng tìm hiểu phương pháp xây dựng bản thể học bán tự động từ kho ngữ liệu dạng văn bản, chúng tôi đã tìm hiểu, trình bày và hiện thực các thuật toán từ các ý tưởng trình bày trong các tài liệu. Phần tìm hiểu và hiện thực của luận văn đã đáp ứng nhiệm vụ đề ra của đề tài:

- Khảo sát các phương pháp xây dựng ontology từ kho ngữ liệu dạng văn bản (chương 2)
- Đề xuất (hoặc cải tiến) một phương pháp xây dựng ontology từ kho ngữ liệu dạng văn bản trên cơ sở kết hợp xử lý ngôn ngữ tự nhiên (chương 3).
- Tiến hành thực nghiệm, đánh giá và hiệu chỉnh phương pháp (chương 3,4).

Kết quả tìm hiểu của luận văn có ý nghĩa với các đóng góp như sau:

- Luận văn đã tìm hiểu và trình bày được một trình tự các bước thực hiện để tạo bản thể học bán tự động từ kho ngữ liệu dạng văn bản trên cơ sở kết hợp xử lý ngôn ngữ tự nhiên.
- Luận văn đã trình bày cách sử dụng công cụ Gate UK để phân tích văn bản, trích ra các từ, cụm từ theo 1 đặc trưng nào đó.

Nhìn chung, kết quả bước đầu của những thực nghiệm ở chương 4 thể hiện tính khả thi của trình tự thực hiện mà tác giả đã tìm hiểu và trình bày trong luận văn

đồng thời phản ánh tính hiệu quả của các thuật toán tác giả đã hiện thực. Những thực nghiệm, nghiên cứu và việc cải tiến các thuật toán là cần thiết để nâng cao tính hiệu quả của trình tự thực hiện mà luận văn đã trình bày.

5.2 Hướng phát triển

Các thực nghiệm trong chương 4 đã thử nghiệm cho tất cả các bước thực hiện và giải thuật mà luận văn tìm hiểu. Tuy nhiên, một số vấn đề cần được nghiên cứu trong giai đoạn tiếp theo:

- Vấn đề 1: Cần tối ưu hơn phần cài đặt các giải thuật và tận dụng các công cụ có sẵn để nâng cao hiệu suất thực hiện, giảm chi phí đến mực thấp nhất có thể.
- Vấn đề 2: Xem xét tìm hiểu để có thể cải tiến thành việc xây dựng bản thể học tiếng Việt thay vì tiếng Anh như hiện nay.

5.3 Lời kết

Toàn bộ nội dung trình bày trong luận văn là kết quả quá trình nghiên cứu, tìm hiểu của chúng tôi. Các kết quả trình bày trong 5.1 là kết quả quá trình làm việc, tìm hiểu, học hỏi của chúng tôi. Phần hướng phát triển trình bày trong 5.2 là vấn đề chúng tôi ấp ủ và mong muốn thực hiện được trong thời gian tới nhằm giúp người thực hiện có cách nhìn cụ thể về việc tạo bản thể học đặc biệt là bản thể học tiếng Việt.

TÀI LIỆU THAM KHẢO

-Tiếng Việt:

[1]. Nguyễn Chánh Thành (2010). Xây dựng mô hình mở rộng truy vấn trong truy xuất thông tin văn bản. Luận văn Tiến sĩ Kỹ thuật. Chuyên ngành Khoa học máy tính, Đại học Bách khoa tp HCM.

-Tiếng Anh:

[2]. Mithun Balakrishna, Dan Moldovan, Marta Tatu, Marian Olteanu. "Semi-Automatic Domain Ontology Creation from Text Resources"(2010). Lymba Corporation Richardson TX75080 USA.3187-3194.

[3]. Alexander Maedche, Steffen Staab. "Learning Ontologies for the Semantic Web". IEEE Intelligent Systems archive Volume 16 Issue 2, March 2001 Page 72-79

[4]. Eva Blomqvist, " Semi-automatic Ontology Engineering using Patterns" (2007), 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007.

[5]. Howard Beck, Helena Sofia Pinto, "Overview of approach, Methodologies, Standards, and Tools for Ontologies",(2002), The Agricultural Ontology Service, UN FAO.

[6]. Youn Seongwook and McLeod Dennis, "Ontology Development Tools for Ontology- Based Knowledge Management ", 2006, Non – published Research Reports. Paper 100. From http://research.create.usc.edu/nonpublished_reports/100/

[7]. Julita Bermejo,"A Simplified Guide to Create an Ontology". ASLabR-2007-004 v 0.1 Draft May 22, 2007.

[8]. Mr. Izzeddin A.O. Abuhassan, Akram M.O. AlMashaykhi,"Domain Ontology for Programming Languages", (2012), Journal of

Computations & Modelling, vol.2, no.4, 2012, 75-91 ,ISSN: 1792-7625 (print), 1792-8850 (online) Scienpress Ltd, 2012

[9]. Natalya F. Noy and Deborah L. McGuinness**.** "*Ontology Development 101: A Guide to Creating Your First Ontology*", Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.

[10]. M. Poesio and A. Almuhareb. "Identifying concept attributes using a classifier". In Proceedings of the ACL Workshop on Deep Lexical Acquisition, pages 18–27, 2005.

[11]. A. Maedche and S. Staab. "Semi-automatic engineering of ontologies from text". In Proceedings of the 12th International Conference on Software Engineering and Knowledge Engineering, 2000.

[12]. Sylvie Szulman, Nathalie Aussenac-Gilles, Adeline Nazarenko, Henry Valéry Teguiak, Eric Sardet, Jean Charlet. "DAFOE: A Platform for Building Ontologies from Texts", 2008, Conference: KEOD 2009 - Proceedings of the International Conference on Knowledge Engineering and Ontology Development, Funchal - Madeira, Portugal, October 6-8, 2009

[13]. Hiep Luong, Susan Gauch and Qiang Wang (2012). "Ontology Learning Using Word Net Lexical Expansion and Text Mining", Theory and Applications for Advanced Text Mining, Prof. Shigeaki Sakurai (Ed.), ISBN: 978-953-51-0852-8, InTech, DOI: 10.5772/51141.

[14]. Thomas R.Gruber**. "**Toward Principles for the Design of Ontologies Used for Knowledge Sharin". Volume 43 Issue 5-6, Nov./Dec. 1995 Pages 907 - 928.

[15]. B. Chandrasekaran, John R. Josephson,V. Richard Benjamins. " What are ontologies, and Why do we need them?". IEEE Intelligent Systems archive January/February 1999 page 1094

[16]. Nicola Guarino.” Formal Ontology and Information Systems”, 1998, Proceedings of FOIS’98, Trento, Italy, 6-8 June 1998. Amsterdam, IOS Press, pp. 3-15.

Trang web:

[17]. Gate UK, http://gate.ac.uk

[18]. Gate UK, http://gate.ac.uk/releases/gate-8.0-build4825-ALL/doc/tao/splitch3.html

[19]. Protégé, http://protege.stanford.edu/

[20]. WordNet, http://wnsqlbuilder.sourceforge.net

[21]. http://www.princeton.edu/~achaney/tmve/wiki100k/docs/Ontology_(information_science).html

Ebook:

[22]. Juan Soulié, C++ Language Tutorial, 2007, http://www.cplusplus.com/doc/tutorial/

[23]. C Programming Tutorial, http://www.tutorialspoint.com/

[24]. Faraz Rasheed, C# School, www.programmersheaven.com

[25]. Java Tutorial, http://www.tutorialspoint.com/

[26]. Sam A.Abolrous, Learn Pascal in Three days, 2002, Wordware Publishing, Inc.

[27]. Liew Voon Kiong ,Visual Basic 2010 Made Easy, 2011

[28]. Liew Voon Kiong ,Visual Basic 6 Made Easy, 2006

PHỤ LỤC 1 – Ontology Engineering Tools (Ontology Editors) [2008/04/24]

Nguồn: http://www.hozo.jp/OntoTools/

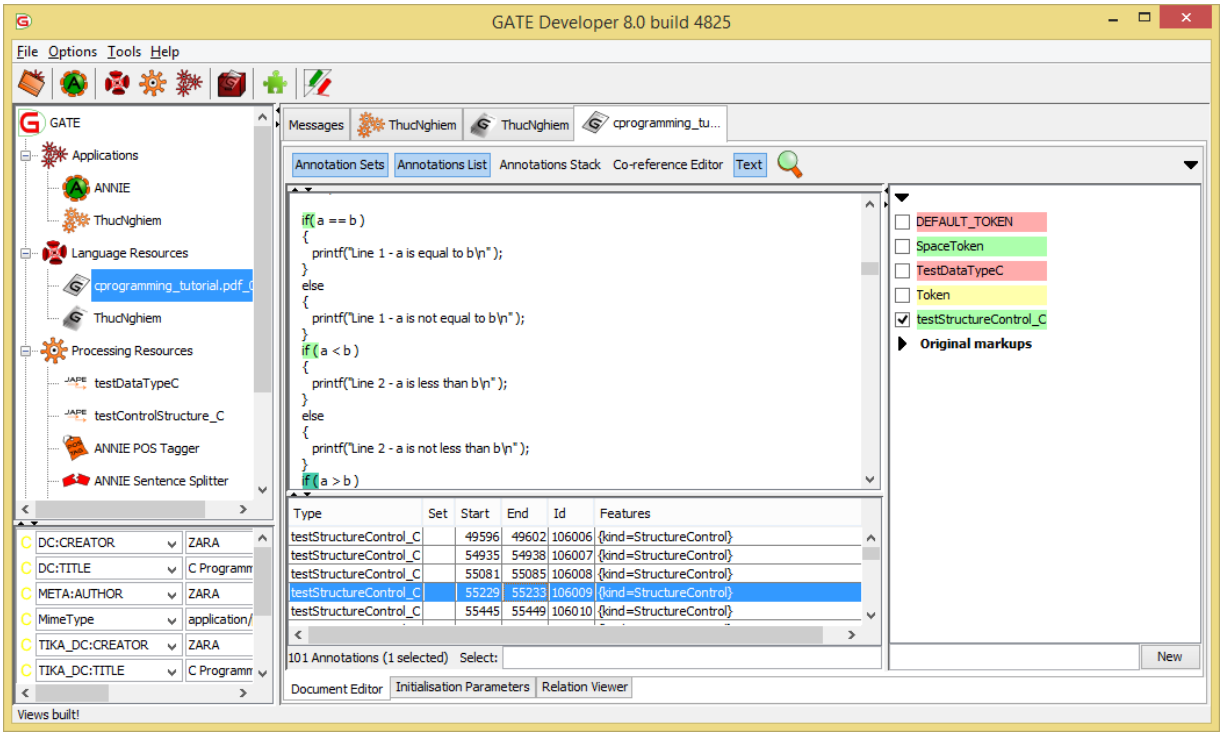
Name of Tool	Web site
Amilcare	http://www.aktors.org/technologies/amilcare/
Apollo	http://apollo.open.ac.uk/
ArgoUML	http://argouml.tigris.org/
BioPortal	http://www.bioontology.org/tools/portal/bioportal.html
Cerebra Server	http://www.webmethods.com/
Chimæra	http://www.ksl.stanford.edu/software/chimaera/
COBrA	http://www.xspan.org/cobra/index.html
COE(CampTools Ontology Editor)	http://cmap.ihmc.us/coe/
COE: A collaborative ontology editor based on a peer-to-peer framework	
CoGITaNT	http://cogitant.sourceforge.net/
CoGui	http://www.lirmm.fr/cogui/
ConcepTool	http://www.csd.abdn.ac.uk/research/IKM/projects/ConcepTool/
CONE	http://briefs.cs.hut.fi/phase4/Ontologies_and_KB/index.html
Construct	http://www.networkinference.com/
Contextia	http://www.modulant.co/
COPORUM OntoBuilder	http://www.ontoserver.cognit/
Corese	http://www-sop.inria.fr/acacia/soft/corese/
Cypher	http://www.monrai.com/products/cypher
DAG-Edit	http://amigo.geneontology.org/dev//java/dagedit/docs/index.html
DAML UML Enhanced Tool (DUET)	http://codip.grci.com/Tools/Tools.html
DAML+OIL Plug-in for Protege-2000	http://www.ai.sri.com/daml/DAML+OIL-plugin/
Disciple Learning Agent Shell	http://lalab.gmu.edu/
DL-workbench	http://projects.opencascade.org/dl-workbench/
DOE - The Differential Ontology Editor	http://homepages.cwi.nl/~troncy/DOE/
DogmaModeler	http://www.starlab.vub.ac.be/staff/mustafa/phd-thesis/
DOME	http://dome.sourceforge.net/
DUET	http://codip.grci.com/Tools/Tools.html
e-COSer - e-COGNOS Ontology Server	www.e-cognos.org
Enterprise Information Integration＜@Semantics'＞	http://www.asemantics.com/
Enterprise Information Integrator (EII)	
ExClaim & CommonKADS Workbench	
Experiment Design Automation (XDA)＜Teranode＞	http://www.teranode.com/products/index.php
EXPRESS Data Manager VisualExpress	http://www.epmtech.jotne.com/
ezOWL	http://iweb.etri.re.kr/ezowl/
Freedom(formerly Enterprise Semantic Platform)	http://www.computas.com/
GALEN Case Environment	http://www.kermanog.com/
GINO	http://iswc2006.semanticweb.org/items/Bernstein2006tg.pdf

Name of Tool	Web site
GKB Editor	http://www.ai.sri.com/~gkb/
Graphl	http://home.subnet.at/flo/mv/graphl/
GrOWL	http://ecoinformatics.uvm.edu/technologies/growl-knowledge-modeler.html
Haystack	http://haystack.lcs.mit.edu/
ICOM	http://www.inf.unibz.it/~franconi/icom/
InferEd	http://www.intellidimension.com/
Integrated Ontology Development Environment(IODE)	http://www.ontologyworks.com/
IODT(IBM Integrated Ontology Development Toolkit)	http://www.alphaworks.ibm.com/tech/semanticstk
IsaViz	http://www.w3.org/2001/11/IsaViz/
Jambalaya	http://www.thechiselgroup.org/jambalaya
JOE	http://www.topbraidcomposer.com/
JSemWed	http://proyecto-rg.tripod.com/
KAON	http://kaon.semanticweb.org/
KBST-EM	http://www.aiai.ed.ac.uk/~jessicac/project/2-workflow-tech-profile-sub/details.html
K-Infinity Knowledge Builder	http://www.i-views.de/
KMgen	http://www.algo.be/ref-projects.htm#KMgen
Knoodl	http://www.knoodl.com/ui/home.html
KSMSA Ontology Editor	http://virtual.cvut.cz/ksmsaWeb/notes/ontoEdit.zip/qBuT6NEcJMJ147Pc.html
LegendBurster Ontology Editor	http://www.georeferenceonline.com/LegendBurster/
LexGrid Editor	http://informatics.mayo.edu/LexGrid/
LexiLink™	http://www.arity.com/?Tab=products&Tab2=lexilink
LinkFactory®	http://www.landcglobal.com/pages/linkfactory.php
M3t4.Studio Semantic Toolkit	http://www.m3t4.com/index.jsp
McCullough Knowledge Explorer (MKE)	http://mkrmke.org/
Medius Visual Ontology Modeler	http://www.sandsoft.com/products.html
Metis Enterprise	http://www.computas.com/
MOMIS	http://www.dbgroup.unimo.it/Momis/
morla	http://www2.autistici.org/bakunin/morla/
MR3	http://www.yamaguchi.comp.ae.keio.ac.jp/mmm/mr3/index.html
myWeb	http://www.ontologyonline.org/main.html
NeoClassic	http://www-out.bell-labs.com/project/classic/
OBO Converter	http://www.bioontology.org/tools/oboinowl/obo_converter.html
OBO-Edit	http://geneontology.sourceforge.net/
OCW Ontology Craft Workbench (formerly Onto-Builder	http://ontology.univ-savoie.fr/condillac/
OilEd	http://oiled.man.ac.uk/
OLR3 Schema Editor	http://www.kbs.uni-hannover.de/%7Etkunze
OnoEdit	http://www.ontoknowledge.org/tools/ontoedit.shtml
OntoBilder (OntoX .etc)	http://iew3.technion.ac.il/OntoBuilder/
Onto-Builder	http://www.onto-med.de/en/applications/ontobuild/
OntoGen	http://ontogen.ijs.si/

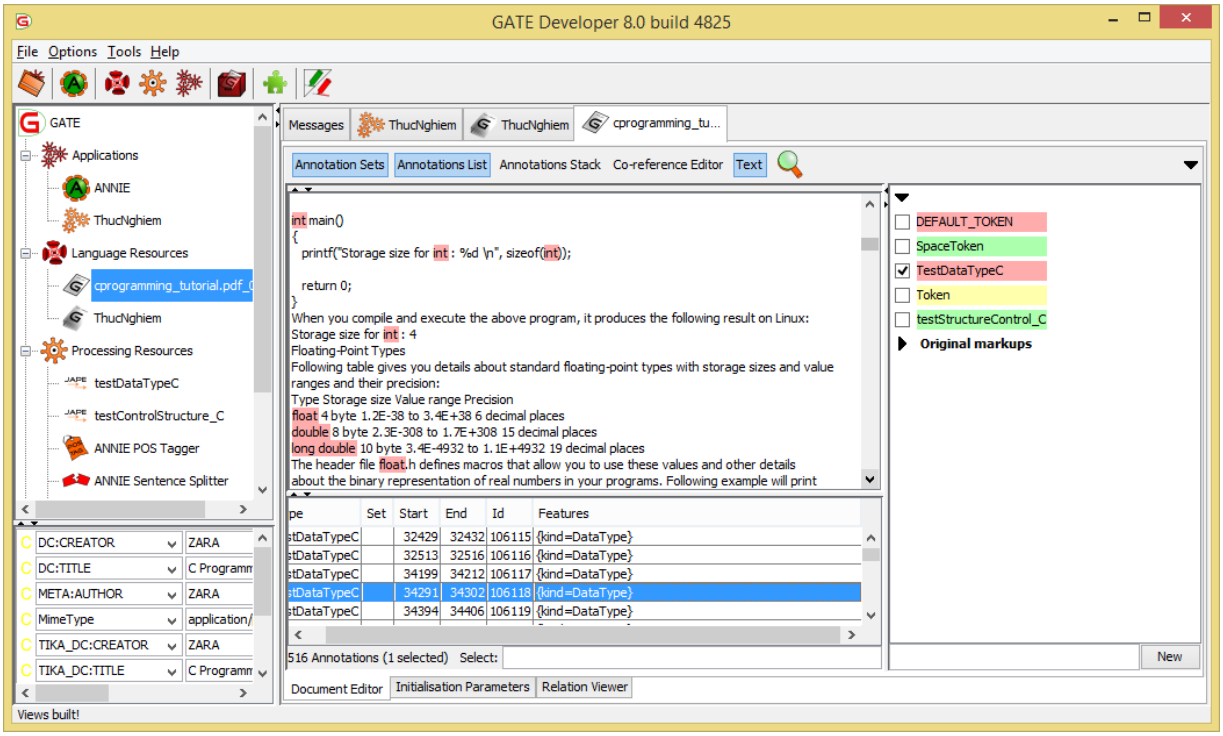
Name of Tool	Web site
Ontolingua	http://www.ksl.stanford.edu/software/ontolingua/
Ontology Editor for Eclipse	http://ebiquity.umbc.edu/project/html/id/26/
Ontology Generator	http://progos.hu/tools/og/
Ontology Graph(OGraph)	http://codip.grci.com/Tools/Components.html
Ontology Management System (SNOBASE)	http://www.alphaworks.ibm.com/tech/snobase
OntoMerge	http://www.cs.yale.edu/homes/dvm/daml/ontology-translation.html
Ontopia Knowledge Suite	http://www.ontopia.net/solutions/products.html
Ontosaurus	http://www.isi.edu/isd/ontosaurus.html
OntoStudio	http://www.ontoprise.de/
OntoTerm	http://www.ontoterm.com/
OntoTrack	http://www.informatik.uni-ulm.de/ki/ontotrack/
OntoXpl	http://www.cs.concordia.ca/ying_lu/
OPCAT -Object-Process CASE Tool	http://objectprocess.org/
Open Ontology Forge	http://research.nii.ac.jp/~collier/resources/OOF/index.html
OpenCyc Knowledge Server	http://www.opencyc.org/
OpenKnoMe	http://www.opengalen.org/sources/software.html
OpenLink Data Spaces (ODS)	http://virtuoso.openlinksw.com/wiki/main/Main/OdsIndex
Orient	http://apex.sjtu.edu.cn/projects/orient/index.htm
OWL Emacs Mode	http://projects.semwebcentral.org/projects/owl-emacs/
OWL Filetype Plugin for VIM	http://projects.semwebcentral.org/projects/owlvim/
OWL Plugin for Protege 2000	http://protege.stanford.edu/overview/protege-owl.html
OWL verbalizer	http://attempto.ifi.unizh.ch/site/docs/verbalizing_owl_in_controlled_english.html
OWL-S Editor	http://owlseditor.semwebcentral.org/
OWL-S IDE	http://projects.semwebcentral.org/projects/owl-s-ide/
Oyster	http://oyster.ontoware.org/
PCPACK	http://www.epistemics.co.uk/
Phenote	http://www.bioontology.org/tools/phenote/phenote.html
pOWL	http://sourceforge.net/projects/powl
Prompt	http://protege.cim3.net/cgi-bin/wiki.pl?Prompt
protégé	http://protege.stanford.edu/
RDF Gravity	http://semweb.salzburgresearch.at/apps/rdf-gravity/index.html
RDF InferEd	http://www.intellidimension.com/default.rsp?topic=/pages/site/products/infered/default.rsp
RDFAuthor	http://rdfweb.org/people/damian/RDFAuthor/
RDFe - A Schema-Aware RDF Editor	
RDFedt	http://www.jan-winkler.de/dev/e_rdfe.htm
RIC	http://www.mindswap.org/~mhgrove/RIC/RIC.shtml
Rice	http://www.ronaldcornet.nl/rice/
Scholarly Ontologies Project	http://kmi.open.ac.uk/projects/scholonto/
Seamark Navigator（Siderean）	http://www.siderean.com/index.aspx
Semantic Information Router （Profium）	http://www.profium.com/
Semantic Web Client Library	http://sites.wiwiss.fu-berlin.de/suhl/bizer/ng4j/semwebclient/
Semantica	http://www.semanticresearch.com/

Name of Tool	Web site
SemanticWorks	http://www.altova.com/products/semanticworks/semantic_web_rdf_owl_editor.html
SemTalk	http://www.semtalk.com/
SLRP(Semantic Layered Research Platform）	http://ibm-slrp.sourceforge.net/
SMORE	http://www.mindswap.org/2005/SMORE/
Snoggle	http://projects.semwebcentral.org/projects/snoggle/
Specware	http://www.specware.org/
SUO-KIF Browser	http://virtual.cvut.cz/ksmsa/resources/index.html
SWeDE	http://owl-eclipse.projects.semwebcentral.org/
SWOOP	http://www.mindswap.org/2004/SWOOP/
SymOntoX	http://www.symontox.org/
Taxonomy Builder	http://www.semansys.com/
Taxonomy Management System	http://www.wordmap.com/
Terminae	http://www-lipn.univ-paris13.fr/%7Eszulman/TERMINAE.html
The Discovery Machine	http://www.discoverymachine.com/
The Model Futures OWL Editor	http://www.modelfutures.com/OwlEditor.html
Thetus Publisher	http://www.thetus.com/
TMTab (A Topic Map Plug-in For Protégé)	http://www.techquila.com/
TopBraid	http://www.topbraidcomposer.com/
TOPKAT	http://www.aiai.ed.ac.uk/~jkk/topkat.html
Triple20	http://www.swi-prolog.org/packages/Triple20/
Unicorn System	
Visio for Enterprise Architects	
VisualKii	http://www.visualkii.com/
VisualText Conceptual Grammar KB Editor	http://www.textanalysis.com/
Web Ontology Manager （IBM）	http://www.alphaworks.ibm.com/tech/wom?open&S_TACT=105AGX59&S_CMP=GR&ca=dgr-lnxwd01awwom
WebKB	http://www.webkb.org/
WebODE	http://webode.dia.fi.upm.es/WebODEWeb/index.html
WebOnto	http://kmi.open.ac.uk/projects/webonto/
WSMO	http://www.wsmo.org/
XMP（Extensible Metadata Platform）	http://www.adobe.com/products/xmp/index.html
Xtractica with Coherent Description Framework (CDF)	http://www.xsb.com/technology.html
Zeus	http://labs.bt.com/projects/agents/zeus/

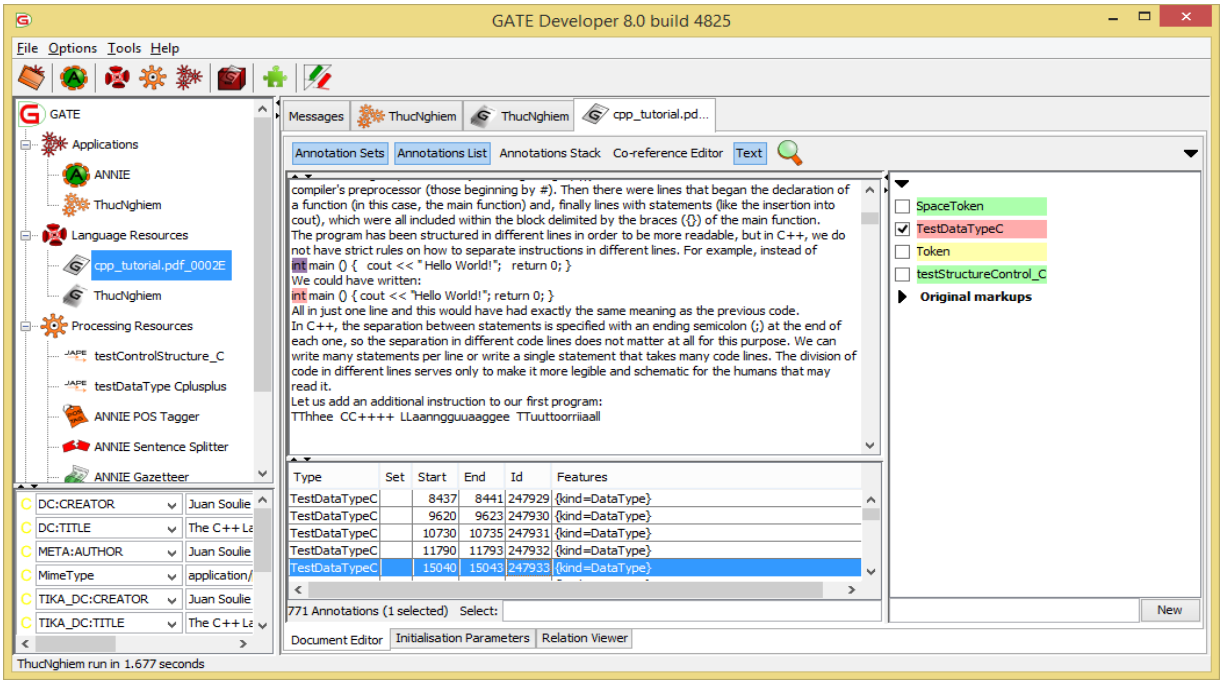
PHỤ LỤC 2 – Kết quả thực hiện phân tích trong GATE



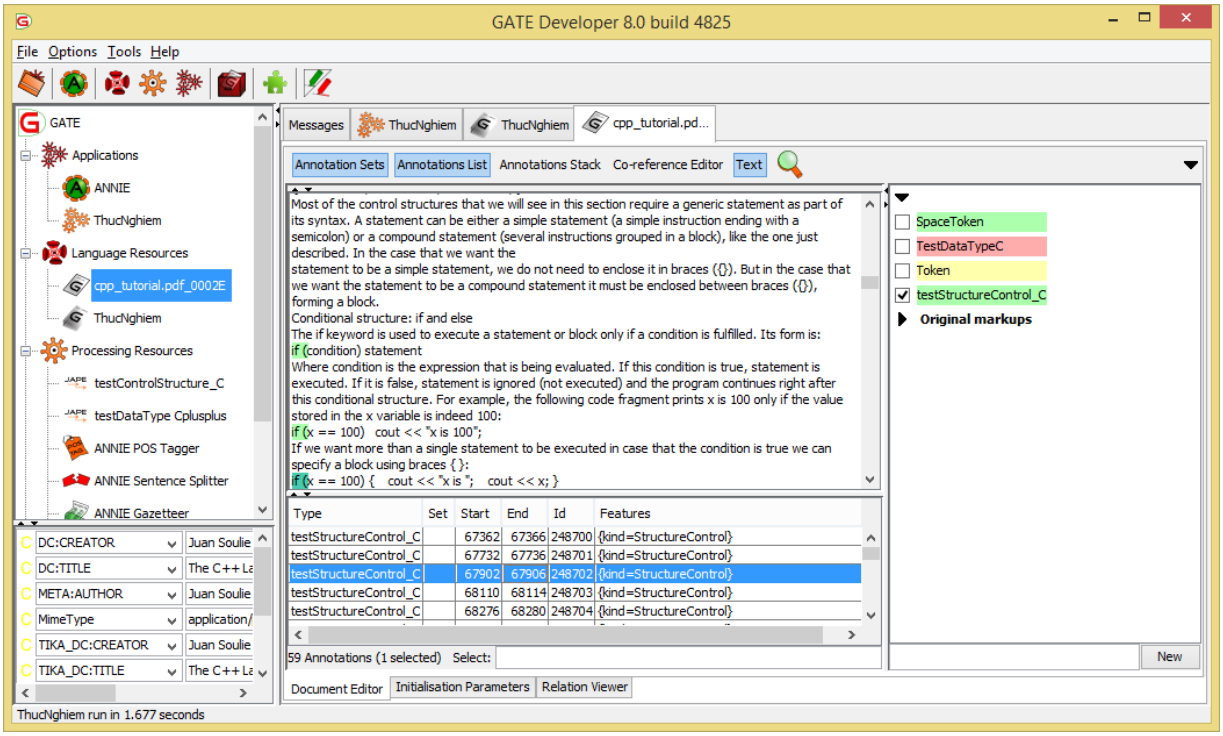
Hình 0-1: Kết quả thực hiện phân tích testControlStructure trong ebook C.



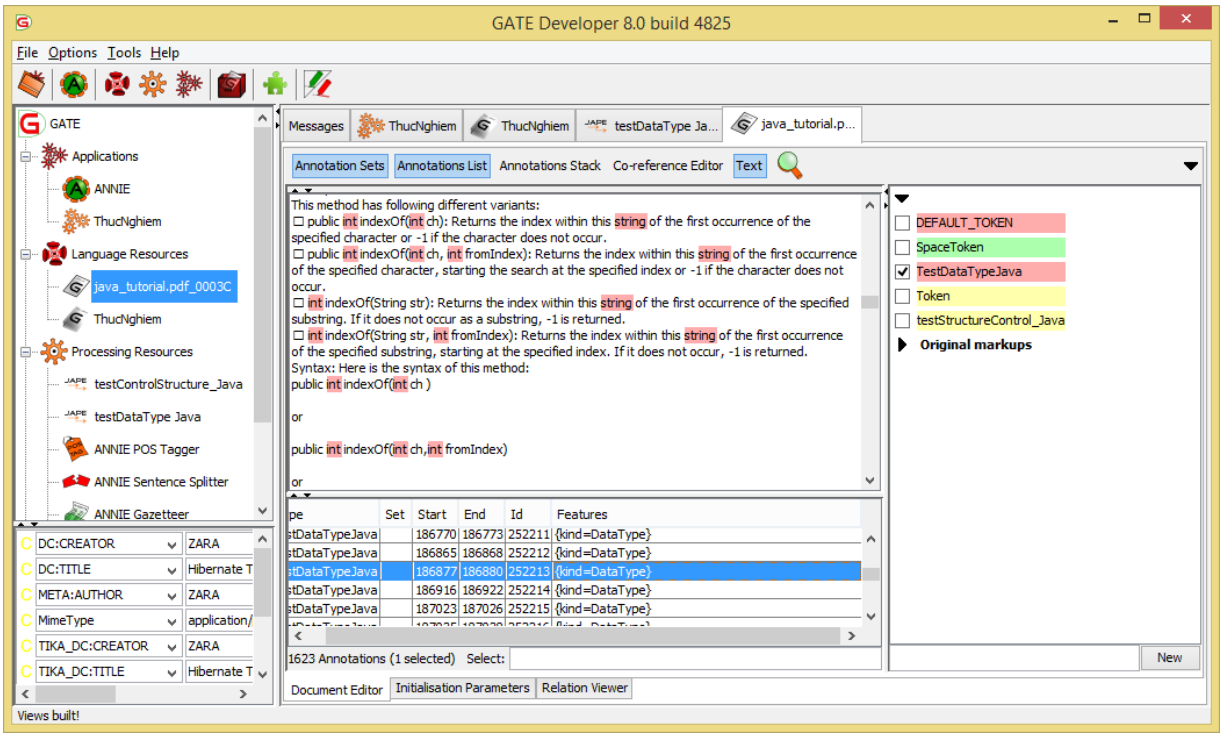
Hình 0-2: Kết quả thực hiện phân tích testDataType trong ebook C.



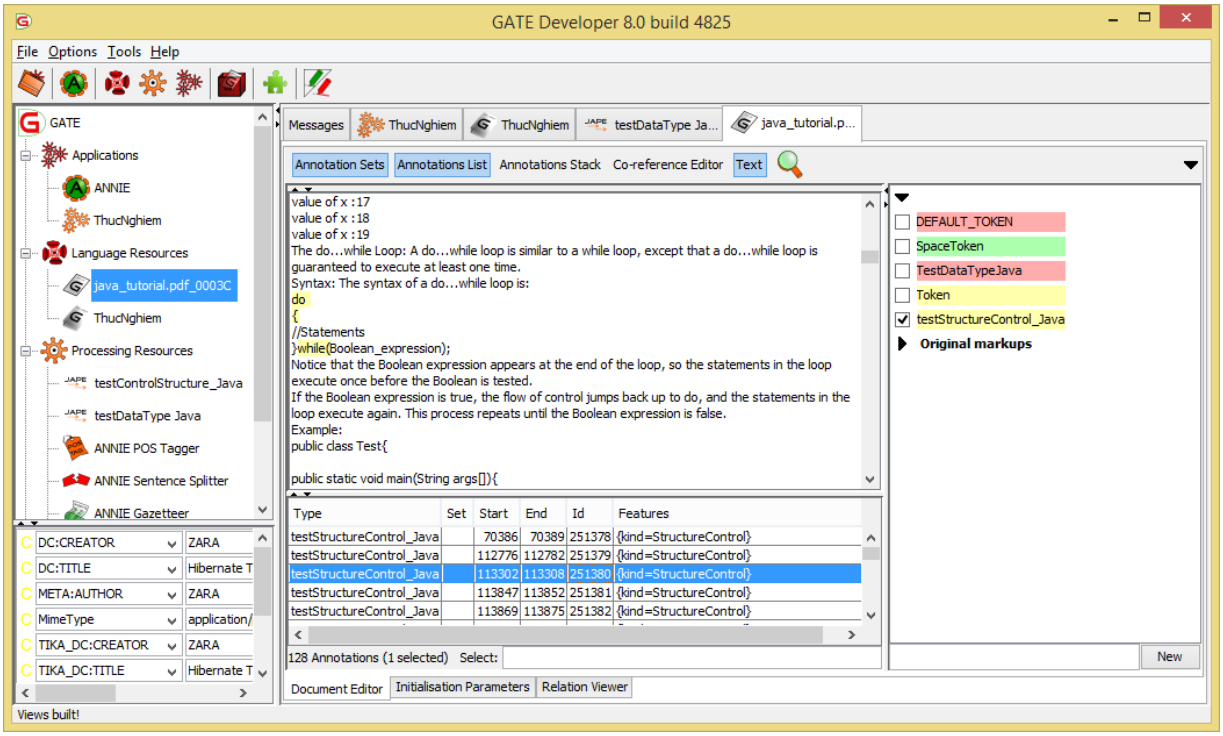
Hình 0-3:Kết quả thực hiện phân tích testDataType trong ebook C++.



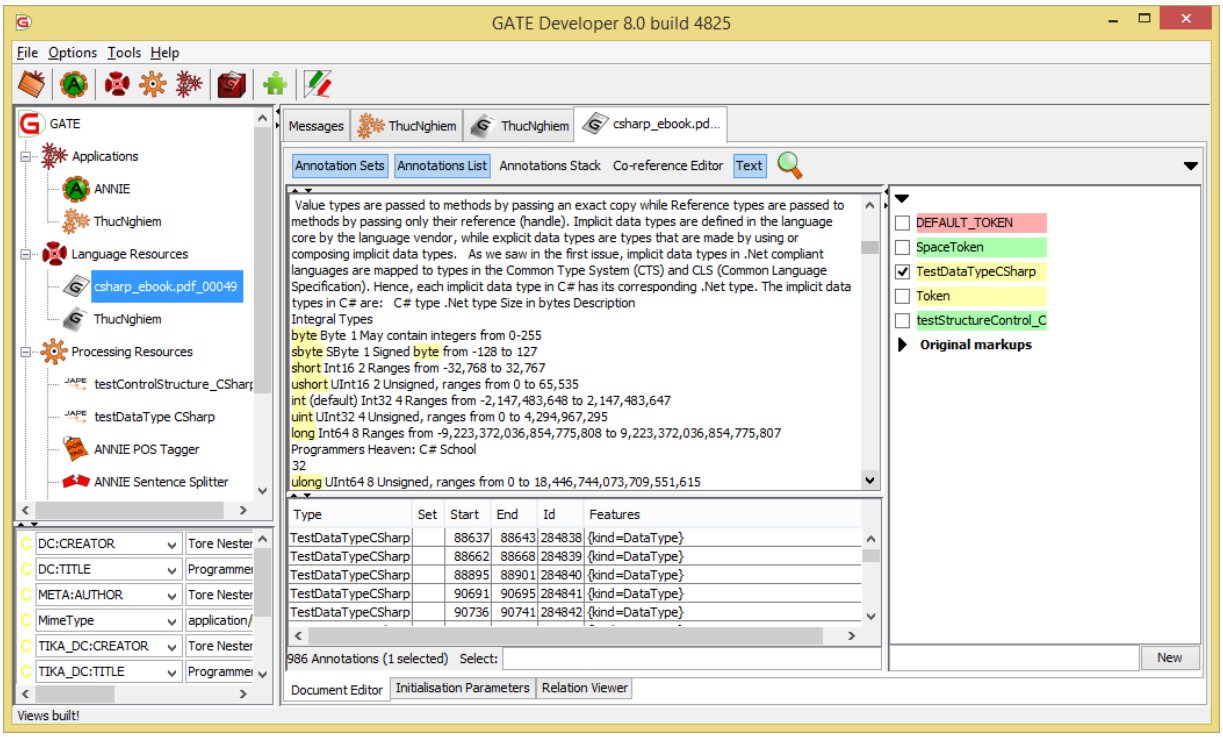
Hình 0-4: Kết quả thực hiện phân tích testControlStructure trong ebook C++.



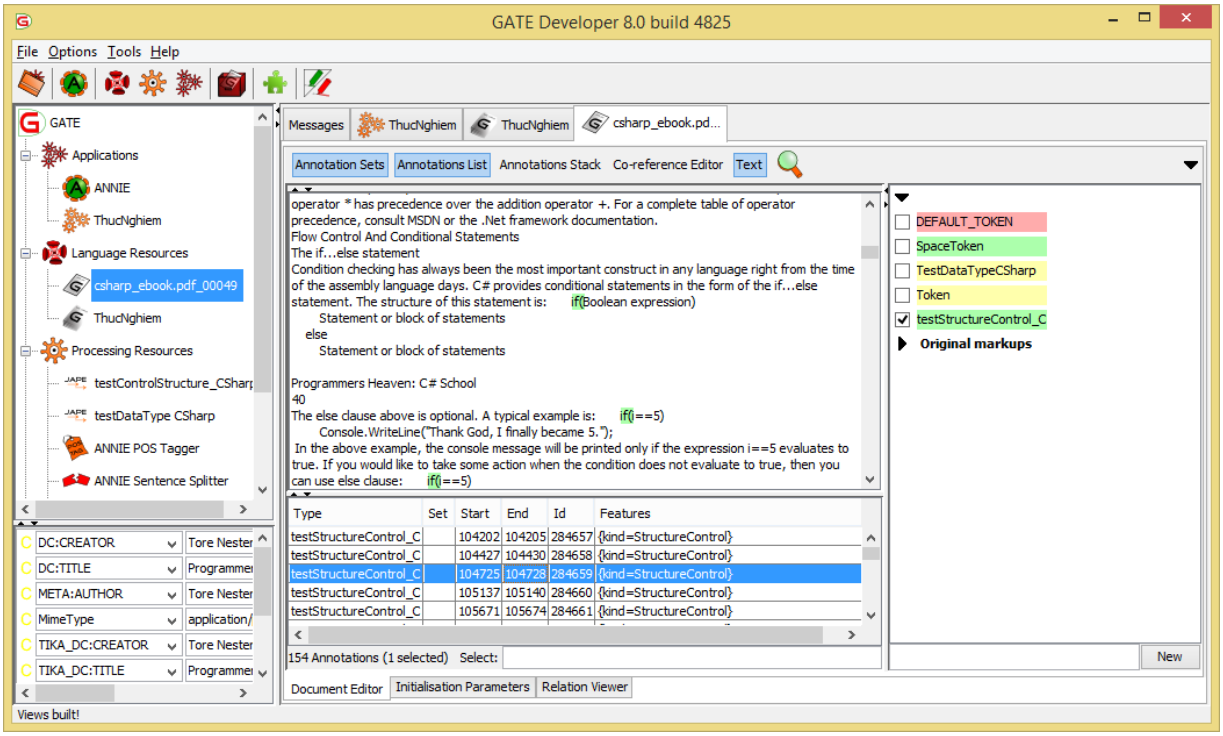
Hình 0-5: Kết quả thực hiện phân tích testDataType trong ebook Java.



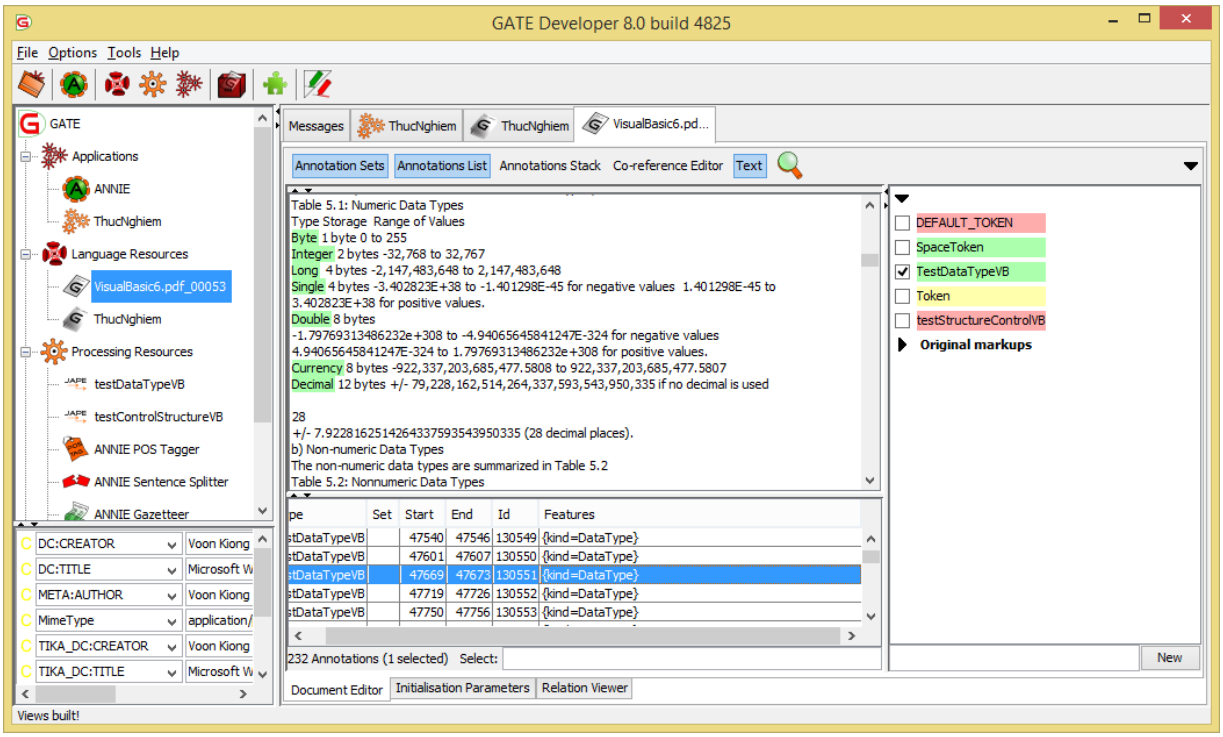
Hình 0-6: Kết quả thực hiện phân tích testControlStructure trong ebook Java.



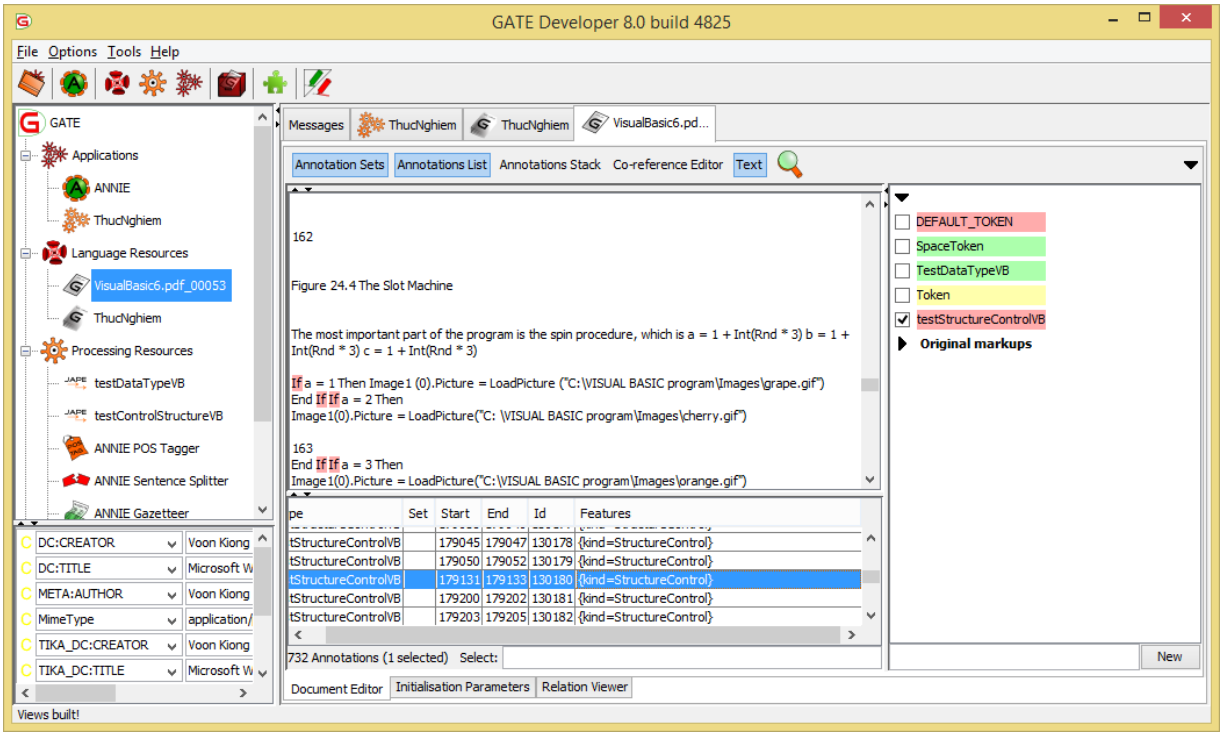
Hình 0-7:Kết quả thực hiện phân tích testDataType trong ebook C#



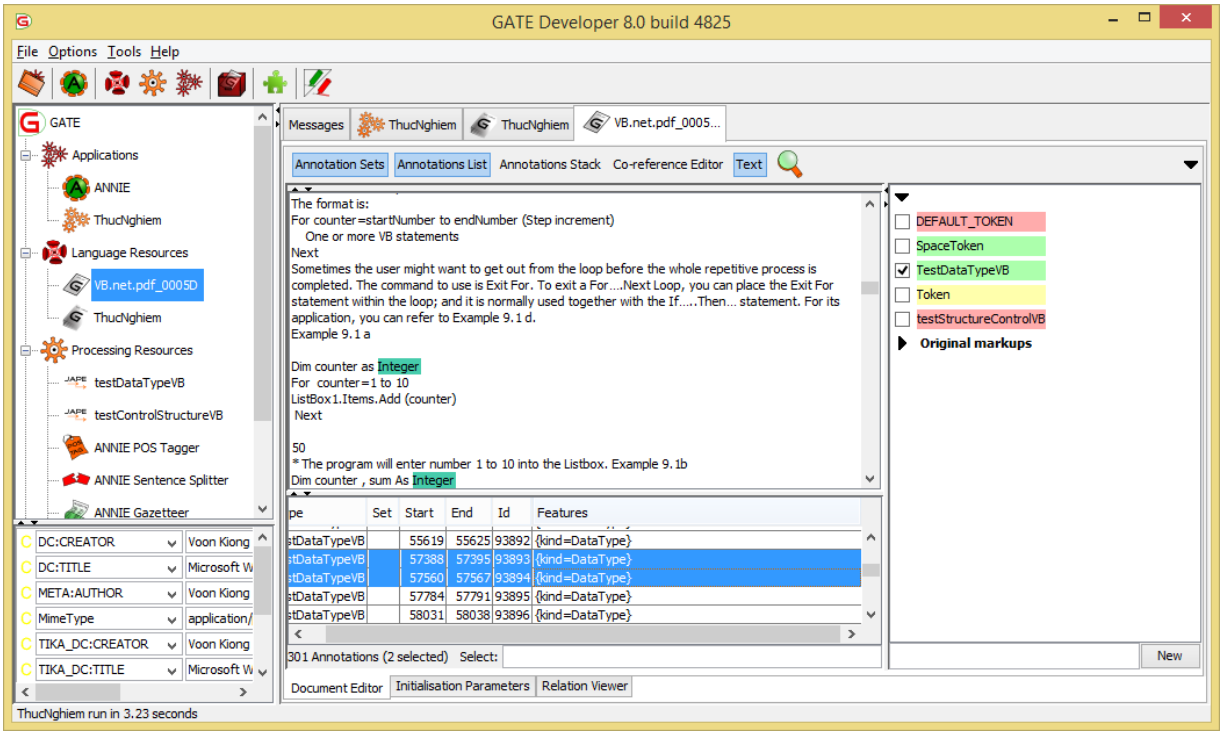
Hình 0-8: Kết quả thực hiện phân tích testControlStructure trong ebook C#.



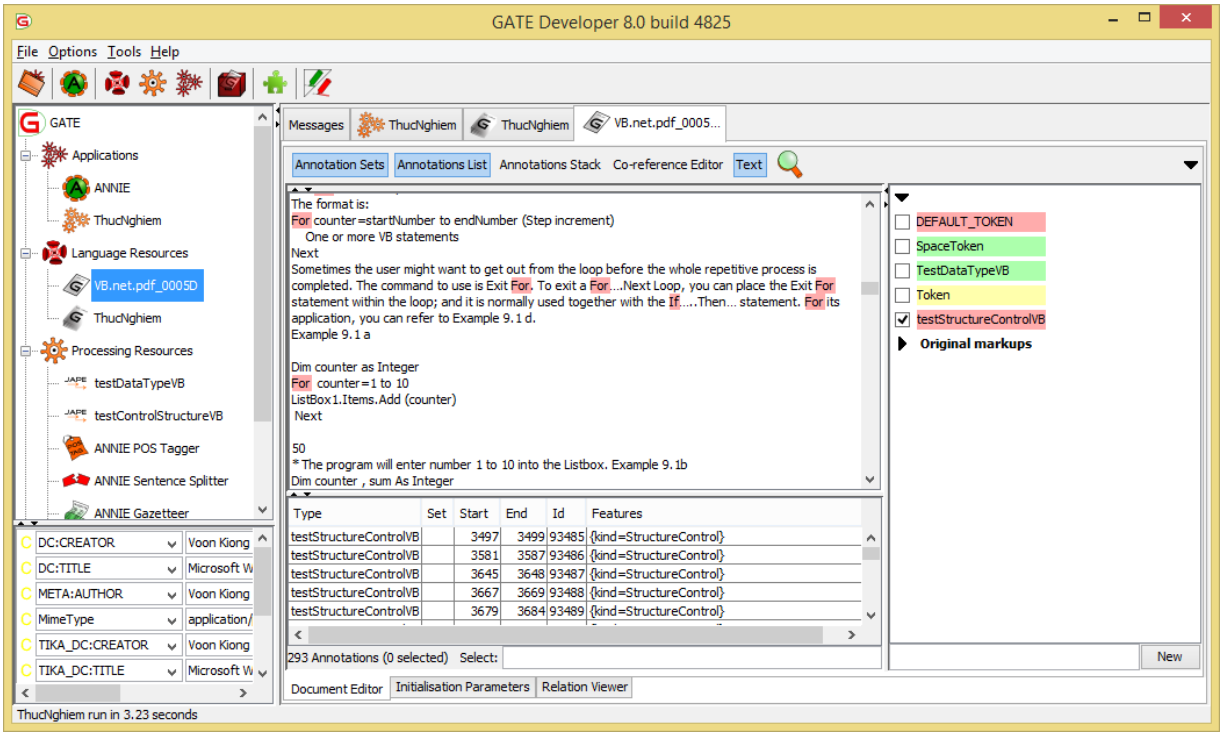
Hình 0-9: Kết quả thực hiện phân tích testDataType trong ebook VisualBasic.



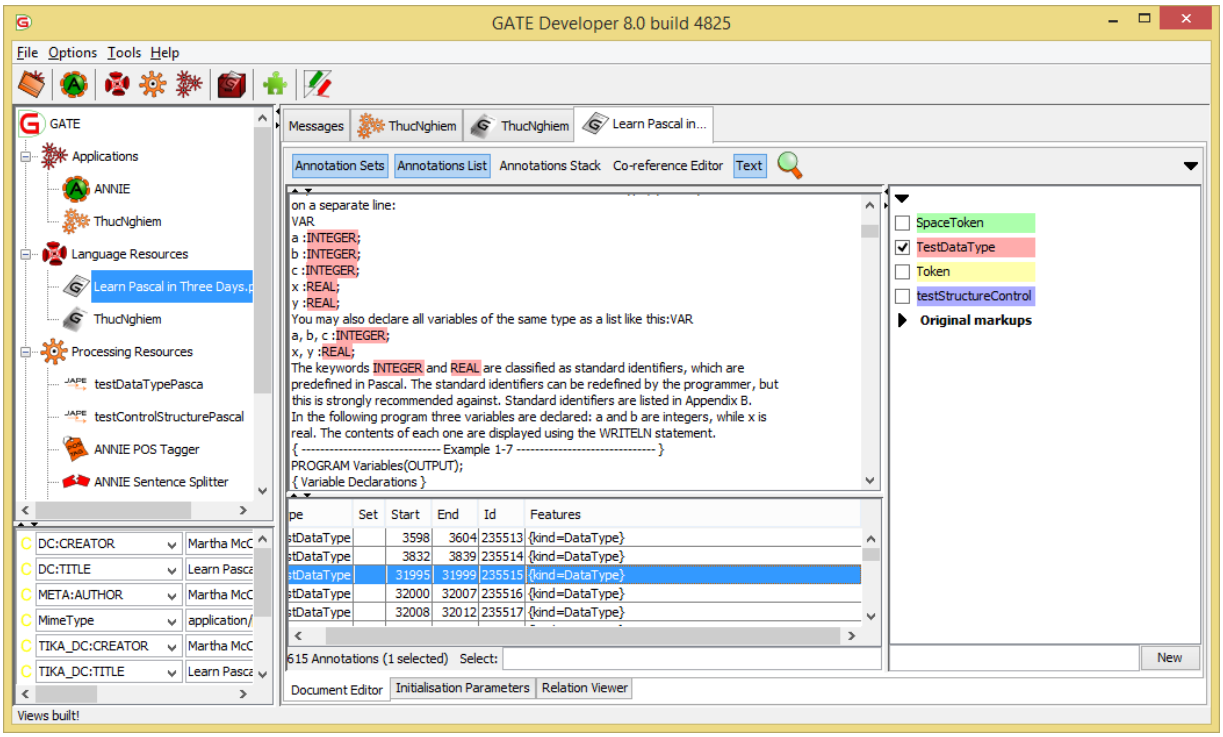
Hình 0-10: Kết quả thực hiện phân tích testControlStructure trong ebook VisualBasic.



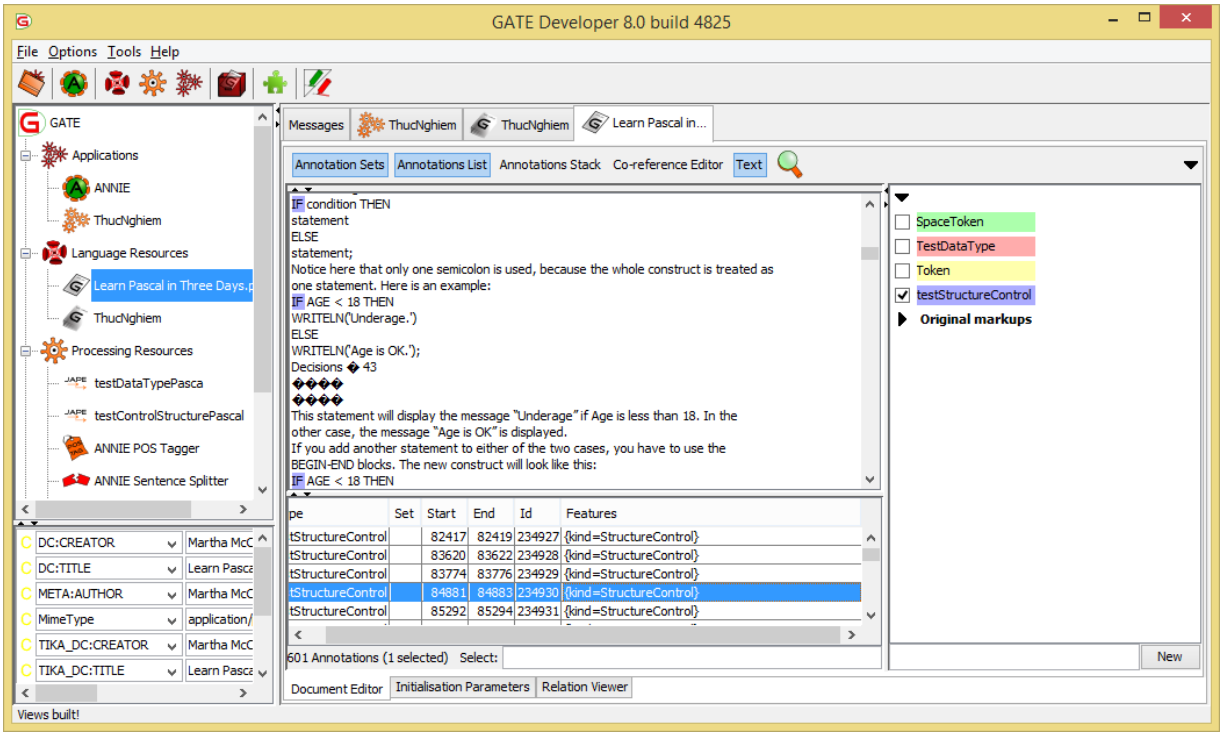
Hình 0-11:Kết quả thực hiện phân tích testDataType trong ebook VB.net



Hình 0-12: Kết quả thực hiện phân tích testControlStructure trong ebook VB.net



Hình 0-13: Kết quả thực hiện phân tích testDataType trong ebook Pascal.


GATE Developer 8.0 build 4825
File Options Tools Help
GATE
Applications
ANNIE
ThucNghiem
Language Resources
Learn Pascal in Three Days.p
ThucNghiem
Processing Resources
testDataTypePasca
testControlStructurePascal
ANNIE POS Tagger
ANNIE Sentence Splitter
DC:CREATOR
Martha McC
DC:TITLE
Learn Pasca
META:AUTHOR
Martha McC
MimeType
application/
TIKA_DC:CREATOR
Martha McC
TIKA_DC:TITLE
Learn Pasca
Views built!
Messages
ThucNghiem
ThucNghiem
Learn Pascal in...
Annotation Sets
Annotations List
Annotations Stack
Co-reference Editor
Text
IF condition THEN
statement
ELSE
statement;
Notice here that only one semicolon is used, because the whole construct is treated as
one statement. Here is an example:
IF AGE < 18 THEN
WRITELN('Underage.')
ELSE
WRITELN('Age is OK.');
Decisions
This statement will display the message "Underage" if Age is less than 18. In the
other case, the message "Age is OK" is displayed.
If you add another statement to either of the two cases, you have to use the
BEGIN-END blocks. The new construct will look like this:
IF AGE < 18 THEN
SpaceToken
TestDataType
Token
testStructureControl
Original markups
pe
Set
Start
End
Id
Features
tStructureControl
82417
82419
234927
{kind=StructureControl}
tStructureControl
83620
83622
234928
{kind=StructureControl}
tStructureControl
83774
83776
234929
{kind=StructureControl}
tStructureControl
84881
84883
234930
{kind=StructureControl}
tStructureControl
85292
85294
234931
{kind=StructureControl}
601 Annotations (1 selected)
Select:
New
Document Editor
Initialisation Parameters
Relation Viewer