TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

Đây là đồ án tốt nghiệp về một hướng ứng dụng của Ontology trong thị trường chứng khoán Việt Nam. Đó là thu thập, tổng hợp thông tin chứng khoán tự động dựa trên Ontology và thể hiện đến cho người sử dụng.

Đồ án đã nêu lên các nghiên cứu về Ontology: khái niệm, cách xây dựng, ứng dụng và lợi ích. Một phương pháp sẽ được sử dụng trong xây dựng Ontology là MethOntology. Giới thiệu sơ lược về các công cụ cài đặt, xử lý Ontology: Protégé, Jena, SPARQL. Một số tìm hiểu về thị trường chứng khoán Việt Nam và cách ứng dụng ontology trong tổng hợp thông tin chứng khoán. Trên cơ sở đó đã xây dựng được một Ontology về giao dịch cổ phiếu và thực hiện suy diễn xếp hạng cổ phiếu trên toàn thị trường, theo ngành. Một công thức tính điểm số xếp hạng cổ phiếu đã được đề nghị. Một module cài đặt suy diễn này cũng đã được phát triển thành công. Thực hiện đánh giá kết quả xếp hạng và nêu hướng phát triển trong tương lai của module.

Phân loại sự kiện chứng khoán tự động là một ứng dụng của bài toán phân loại văn bản dựa trên Ontology được áp dụng vào bài toán trong đồ án này. Trên cơ sở phân tích, tìm hiểu các tin tức, sự kiện liên quan đến chứng khoán, Ontology về sự kiện chứng khoán đã được thiết kế và cài đặt. Ontology này không chỉ có chức năng cung cấp các khái niệm để phân loại sự kiện chứng khoán mà có thể sử dụng để trích chọn nội dung ngữ nghĩa. Một module Phân loại sự kiện chứng khoán tự động đã được cài đặt. Module này có chức năng phân loại sự kiện chứng khoán đầu vào thành các loại, trích chọn thông tin từ nội dung của sự kiện đã phân loại và phát sinh cảnh báo, thay đổi điểm xếp hạng nếu giá trị tìm được thỏa mãn điều kiện do người dùng cho trước. Đánh giá kết quả và hướng phát triển của module cũng được nêu ra.

ABSTRACT OF THESIS

This is the thesis about one direction of applying Ontology in the Vietnamese stock market. That is Ontology-based automatic acquisition and synthetizing securities information and presenting to the users.

The thesis has raised researches about Ontology: definition, how to build, applications and advantages. A methodology will be used in building Ontology is MethOntology. Brief introducing tools used for implementing and processing Ontology: Protégé, Jena, SPARQL. Some basic learning about the Vietnamese stock market and how to apply Ontology in synthetizing securities information. From these basis, an Ontology about stock trading has been built and perform inferences ranking stocks on whole market, on each sector. A formula for calculating ranking stocks point has proposed. A module implements these inferences has developed successfully. Performing result evaluation and suggesting development in the future for this ranking module.

Automatic classification securities events is an application of text classfication problem is applied in the problem of this thesis. On the basis of analysing, studying news, events relating to securities, Ontology about securities events has been designed and implemented. This Ontology not only supplies concepts to classify securities events, but also can be used to extract semantic content. A module for automatically classify securities events has been implemented. This module has functions of classifying inputting securities events into classifications, extracting information from content of classified events and firing warnings, changing ranking point if found values satisfy the conditions which users inputted. Result evaluation and the future development also presented.

Mục lục

Danh sách hình vẽ

Danh sách bảng

Danh mục thuật ngữ và từ viết tắt

Lời nói đầu

Chương 0. Giới thiệu chung

0.1. Tầm quan trọng của bài toán đặt ra trong đề tài

Thị trường chứng khoán Việt Nam ra đời và phát triển từ năm 2000. Hiện nay thị trường chứng khoán đã trở thành một môi trường kinh doanh lớn, với tổng giá trị thị trường chiếm hơn 50% GDP và sự tham gia giao dịch chứng khoán của hàng nghìn người. Trong kinh doanh giao dịch chứng khoán thì một yếu tố cực kì quan trọng là thông tin. Đó là rất nhiều thông tin về : tình hình giao dịch cổ phiếu trong quá khứ, tình hình kinh doanh của công ty phát hành cổ phiếu, các tin tức sự kiện liên quan đến chứng khoán. Các thông tin này rất phong phú và đến từ nhiều nguồn khác nhau. Việc tổng hợp thông tin tự động và trong thời gian ngắn nhất là rất quan trọng với các nhà kinh doanh chứng khoán, giúp họ đưa ra các thông tin mua, bán kịp thời.

Ở Việt Nam hiện nay việc tổng hợp thông tin chứng khoán phần lớn có sự tham gia nhiều của con người và thường được lưu trong cơ sở dữ liệu quan hệ. Với bài toán : Xây dựng Ontology phục vụ xếp hạng giao dịch cổ phiếu và phân loại tin chứng khoán , chúng tôi kì vọng sẽ tạo ra một hệ thống tự động ở mức cao trong việc thu thập thông tin chứng khoán, phân tích tổng hợp và cung cấp cho người dùng trong thời gian ngắn nhất. Đây thực sự là hướng đi mới , có tính ứng dụng thiết thực đối với thị trường chứng khoán Việt Nam.

Việc ứng dụng thành công Ontology vào thị trường chứng khoán còn đem lại khả năng áp dụng Ontology cho các nguồn thông tin khác. Trước hết đó sẽ là các nguồn thông tin về giao dịch tương tự như thị trường chứng khoán: giao dịch vàng, tiền tệ, giao dịch nhà đất, ... Mở rộng hơn nữa, có thể ứng dụng Ontology để quản lý nguồn thông tin tri thức phong phú từ nhiều lĩnh vực của đời sống (kinh tế, giáo dục, y học,...) mà một hướng đi ban đầu có thể từ bài toán được đặt ra trong đồ án tốt nghiệp này.

0.2. Mục đích, yêu cầu

- Xây dựng Ontology cho lĩnh vực kinh doanh chứng khoán.
- Nghiên cứu cài đặt các kĩ thuật trích chọn và tổng hợp thông tin với sự trợ giúp của Ontology.
- Ứng dụng trong xây dựng hệ thống tổng hợp thông tin chứng khoán tự động.
- Công nghệ : Jena, Protege, ngôn ngữ OWL, RDF/RDFS, Java.

0.3. Nội dung đồ án tốt nghiệp

Trước hết tôi đã nghiên cứu về Ontology, một khái niệm còn khá mới mẻ hiện nay. Quá trình nghiên cứu Ontology bao gồm tìm hiểu chi tiết về khái niệm, nguồn gốc Ontology, về các phương pháp xây dựng Ontology, chu trình sống của Ontology. Nghiên cứu về các ngôn ngữ hiện thực Ontology như: RDF/RDFS, OWL và tìm hiểu ứng dụng ngày càng rộng rãi của Ontology trong khoa học và thực tiễn.

Tiếp đó là những tìm hiểu sơ lược về thị trường chứng khoán Việt Nam. Các khái niệm chính, các vấn đề quan trọng đối với nhà đầu tư chứng khoán.

Trên cơ sở đó một Ontology về giao dịch chứng khoán đã được thiết kế và xây dựng. Các dữ liệu giao dịch cổ phiếu sẽ được tự động nhập vào Ontology thông qua nguồn là hai trung tâm giao dịch HASTC và HOSE, kết quả xếp hạng cổ phiếu trên từng thị trường và theo ngành sẽ được tự động suy diễn và cung cấp dịch vụ cho hệ thống hiển thị. Ontology về giao dịch chứng khoán bao quát toàn bộ thị

trường chứng khoán Việt Nam, có khả năng tùy biến cao, khả năng kết hợp với những Ontology khác (như Ontology về báo cáo tài chính) để cung cấp một hệ thống báo cáo tư vấn chứng khoán tự động.

Phân loại tài liệu là một lĩnh vực quan trọng trong khai phá dữ liệu (Data Mining). Nó có ứng dụng to lớn trong thời đại bùng nổ thông tin và thông tin Internet toàn cầu hiện nay. Việc phân loại bằng tay, sử dụng con người là chủ yếu ngày càng trở lên khó khăn và không thể đáp ứng nổi. Có nhiều nghiên cứu về tự động Data Mining và phân loại tài liệu đã được đưa vào trong đồ án, trong đó tập trung vào hướng sử dụng Ontology.

Đồ án nêu ra một nghiên cứu, thiết kế và cài đặt Ontology sử dụng trong phân loại tin chứng khoán của Việt Nam. Các tin tức, sự kiện chứng khoán từ nhiều nguồn (hai trung tâm giao dịch chứng khoán Hà Nội và thành phố Hồ Chí Minh, các báo điện tử) được lấy về và tự động phân loại theo nhiều khái niệm (sự kiện cổ phiếu, sự kiện ngành,...).

Quá trình phân loại còn được phát triển thêm một bước là trích chọn ra các thông tin quan trọng trong nội dung tin chứng khoán. Quá trình trích chọn cũng dựa trên sự hướng dẫn từ Ontology mà người dùng có thể tùy biến. Thông tin được trích chọn sẽ được phục vụ cho nhiều mục đích, một cài đặt trong đồ án là để thay đổi điểm số xếp hạng của cổ phiếu trên toàn thị trường.

0.4. Giới thiệu về báo cáo tốt nghiệp

Báo cáo tốt nghiệp này bao gồm các nội dung:

Chương 1: Nghiên cứu, tìm hiểu về Ontology. Bao gồm: nguồn gốc, khái niệm chi tiết về Ontology; các phương pháp xây dựng Ontology hiện nay; ngôn ngữ Ontology trong đó giới thiệu chi tiết về OWL; và cuối cùng là nêu một số cách ứng dụng phổ biến của Ontology hiện nay.

Chương 2: Nêu các công cụ để xây dựng, xử lý Ontology. Ở đây giới thiệu Protégé - công cụ soạn thảo Ontology, JENA – một framework để lưu trữ, xử lý Ontology và SPAQRL - ngôn ngữ để truy vấn thông tin trên Ontology.

Chương 3: Nêu các nội dung đã tìm hiểu được về thị trường chứng khoán Việt Nam, bao gồm các khái niệm và các chỉ số quan trọng.

Chương 4: Bắt đầu đi vào ứng dụng sử dụng Ontology trong tổng hợp thông tin chứng khoán. Chương này nêu lên cách sử dụng Ontology và giới thiệu kiến trúc tổng quan của hệ thống BKS: hệ thống phân phối thông tin chứng khoán tự động dựa trên Ontology.

Chương 5: Nêu rõ cách thiết kế và cài đặt module Xếp hạng giao dịch cổ phiếu, một module minh họa cách ứng dụng Ontology để tổng hợp thông tin giao dịch chứng khoán.

Chương 6: Giới thiệu bài toán tự động Phân loại và trích thông tin sự kiện chứng khoán. Chương này thể hiện những nghiên cứu về Data mining và cách ứng dụng Ontology trong một lĩnh vực được nhiều người quan tâm là phân loại văn bản. Chương này cũng nêu lên cơ sở toán học được áp dụng trong bài toán Phân loại tin chứng khoán được đặt ra.

Chương 7: Thiết kế và cài đặt module Phân loại và trích thông tin sự kiện chứng khoán. Có đánh giá kết quả thí nghiệm và hướng phát triển trong tương lai.

Chương 1. Ontology

1.1. Khái niệm Ontology

1.1.1 Công nghệ Web ngữ nghĩa (Semantic web technology)

World Wide Web (gọi tắt là Web) ra đời năm 1989 bởi ông Tim Berners-Lee đã trở thành một phương tiện trao đổi thông tin cực kì rộng lớn giữa mọi người trên thế giới. Lượng thông tin số được truyền tải trên Web có số lượng cực lớn và vô cùng phong phú, bao gồm rất nhiều định dạng, thể loại: từ kinh tế, văn hóa, xã hội, chính trị đến giải trí, tin tức, trao đổi thông tin,...

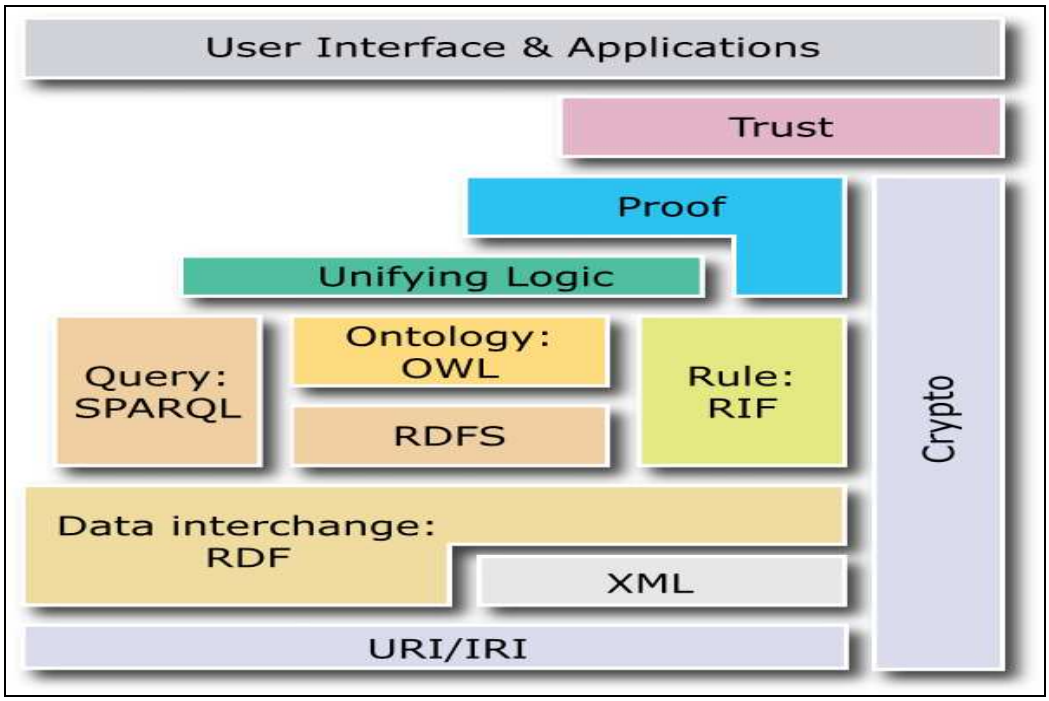
Hầu hết nội dung Web ngày nay chỉ thích hợp cho con người sử dụng. Tức là chỉ con người mới hiểu được ý nghĩa của nội dung đó và lọc ra theo ý muốn của mình để sử dụng. Gắn liền với Web là HTML (Hyper Text Markup Language), một ngôn ngữ đánh dấu được sử dụng để hiển thị nội dung trên trang web. Sử dụng HTML, chúng ta có thể dễ dàng yêu cầu máy tính thể hiện một đoạn text theo kiểu đậm hay đổi màu chữ nhưng không thể cho máy tính biết được đó là đoạn text về một người hay một địa danh. Công nghệ Web ngữ nghĩa ra đời (cũng từ ý tưởng của ông Tim Berners-Lee) cho phép biểu diễn và trao đổi thông tin với đầy đủ ý nghĩa của chúng, tạo ra khả năng xử lý tự động dữ liệu trên Web.

Sử dụng ngữ nghĩa chúng ta có thể cải thiện cách thông tin được biểu diễn. Ở dạng đơn giản, thay vì cung cấp một dãy kết quả tìm kiếm "vô hồn" thì kết quả có thể được phân chia theo ngữ nghĩa. Chúng ta có thể đi xa hơn bằng cách dùng ngữ nghĩa để trộn thông tin từ các nguồn liên quan, loại bỏ sự dư thừa, tóm tắt thông tin khi cần thiết. Quan hệ giữa các tài liệu cũng được thể hiện bằng ý nghĩa.

Việc sử dụng siêu dữ liệu (metadata) ngữ nghĩa còn có thể tích hợp thông tin từ nhiều nguồn khác xa nhau, trong một tổ chức hay giữa các tổ chức. Bằng cách sử dụng sự ánh xạ ngữ nghĩa thì ta có thể đạt được sự liên giao tiếp (interoperation) giữa các ứng dụng sử dụng các thông tin này.

Việc mô ta ngữ nghĩa cho đặc điểm và chức năng của các dịch vụ sẽ đem lại khả năng tích hợp dịch vụ từ dịch vụ đã có, mở rộng công nghệ hướng dịch vụ (SOA).

Web ngữ nghĩa thường được mô tả dưới dạng các "lớp bánh" (layer cake) như hình dưới đây:



Hình 1: Semantic Web Layer Cake [nguồn : http://www.w3.org/2001/sw/]

Kể từ dưới trở lên: Ở lớp dưới cùng là URI (Uniform Resource Identifier) hay bộ định danh tài nguyên thống nhất để biểu diễn một tài liệu duy nhất trên Web. Lớp trên là XML (eXtensible Markup Language), một ngôn ngữ đánh dấu để viết tài liệu web có cấu trúc với các thẻ do người dùng tự định nghĩa. RDF (Resource Description Framework) là một mô hình dữ liệu cơ bản giống như mô hình thực thể quan hệ dùng để viết mô tả đơn giản về tài nguyên Web. RDFS (RDF Schema) cung cấp mô hình nguyên thủy để tổ chức tài nguyên : khái niệm lớp, kế thừa, thuộc tính, miền (domain) và giới hạn phạm vi (range restrictions). Ontology tạo nên xương sống cho Web ngữ nghĩa. Nó cho phép máy tính hiểu được thông tin qua sự liên kết giữa thông tin và các khái niệm trong Ontology, ngoài ra còn là sự liên giao tiếp giữa các thông tin qua sự liên kết trong Ontology hay giữa các Ontology. Ontology sẽ được giải thích chi tiết qua các phần dưới. Lớp trên cùng là các ứng dụng chạy trên nền Web ngữ nghĩa và cung cấp giao diện với người sử dụng, đưa cho họ thông tin cần thiết theo yêu cầu.

1.1.2 Khái niệm Ontology

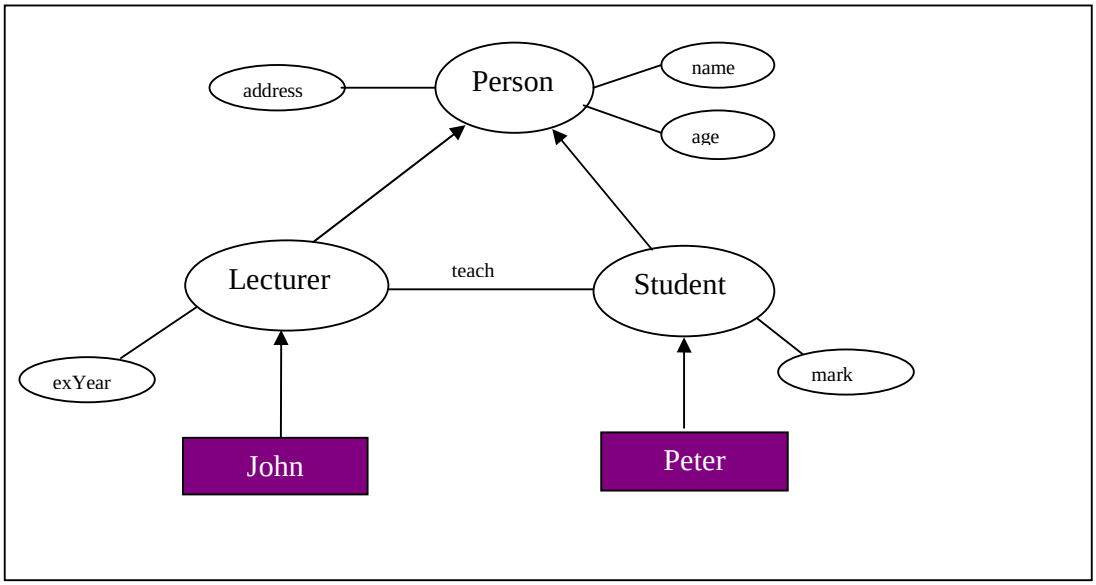
Tại trái tim của tất cả ứng dụng Web ngữ nghĩa là sự sử dụng Ontology. Thuật ngữ “ontology” có gốc từ triết học (nghĩa là “bản thể học”) nhưng đã được chuyển thành thuật ngữ khoa học máy tính từ nhiều năm nay. Một định nghĩa được chấp nhận rộng rãi về Ontology là : “Một Ontology là một đặc tả chính xác và hình thức (an explicit and formal specification) về một khái niệm (a conceptualisation) của một miền thông tin được quan tâm (a domain of interest)” (c.f. Gruber, 1993). Định nghĩa này nhấn mạnh hai điểm chính: đó là các khái niệm được hình thức hóa và bởi vậy cho phép suy diễn (reasoning) bởi máy tính; thứ hai nó nhấn mạnh mỗi Ontology được xây dựng cho một vài miền thông tin cần quan tâm, có như thế nó mới thể hiện được vai trò và tác dụng của nó. Ontology thể hiện sự hiểu biết chung về một miền, các ứng dụng có thể dùng sự hiểu biết chung này để giao tiếp với nhau.

Ontology bao gồm các khái niệm (*concepts*), các quan hệ (*relations*), các thể hiện (*instances*) và *axioms*. Bởi vậy một Ontology thường được biểu diễn dưới dạng bộ 4 {C, R, I, A} trong đó C là tập các khái niệm, R là tập các quan hệ, I là tập các thể hiện và A là tập các axiom [Staab and Studer, 2004]. Trong Ontology, các khái niệm được biểu diễn dưới dạng các lớp (Class), ví dụ : lớp *Person* biểu diễn cho khái niệm *Người*, lớp *Student* biểu diễn cho khái niệm *Sinh viên*. Các khái niệm chính là các đối tượng của miền thông tin (domain) được quan tâm mà chúng ta cần thể hiện.

Quan hệ (Relations) bao gồm trước hết là quan hệ phân cấp (quan hệ cha-con) giữa các lớp, gọi là hierachical relation hay *taxonomy*. Ontology cung cấp hai loại thuộc tính (hayy quan hệ) là thuộc tính đối tượng (object properties) và thuộc tính dữ liệu (datatype properties). Thuộc tính đối tượng cung cấp mối liên hệ giữa các lớp (như trong X *dạy* Y), còn thuộc tính dữ liệu kết nối thuộc tính của lớp với kiểu dữ liệu của nó (như trong X có *tên* kiểu xâu).

Cuối cùng Axiom được sử dụng để cung cấp thông tin suy diễn về các lớp và thuộc tính, ví dụ để nói rằng hai lớp là tương đương hoặc về phạm vi giá trị của một thuộc tính (cardinality).

Ví dụ về một Ontology đơn giản được sử dụng để biểu diễn quan hệ giữa Giáo viên (Lecturer) và Sinh viên (Student) trong một trường đại học. Cả Giáo viên và Sinh viên đều là các khái niệm con của khái niệm Người (Person). Giáo viên thì dạy Sinh viên. Người nào cũng có tên (name), tuổi (age), địa chỉ (address). Giáo viên có thêm thuộc tính năm giảng dạy (experimentYear), Sinh viên có thêm thuộc tính điểm số (mark). Một thể hiện của lớp Giáo viên là John, một thể hiện của lớp Sinh viên là Peter.



Hình 2 : Ví dụ về một Ontology đơn giản

1.2. Xây dựng Ontology

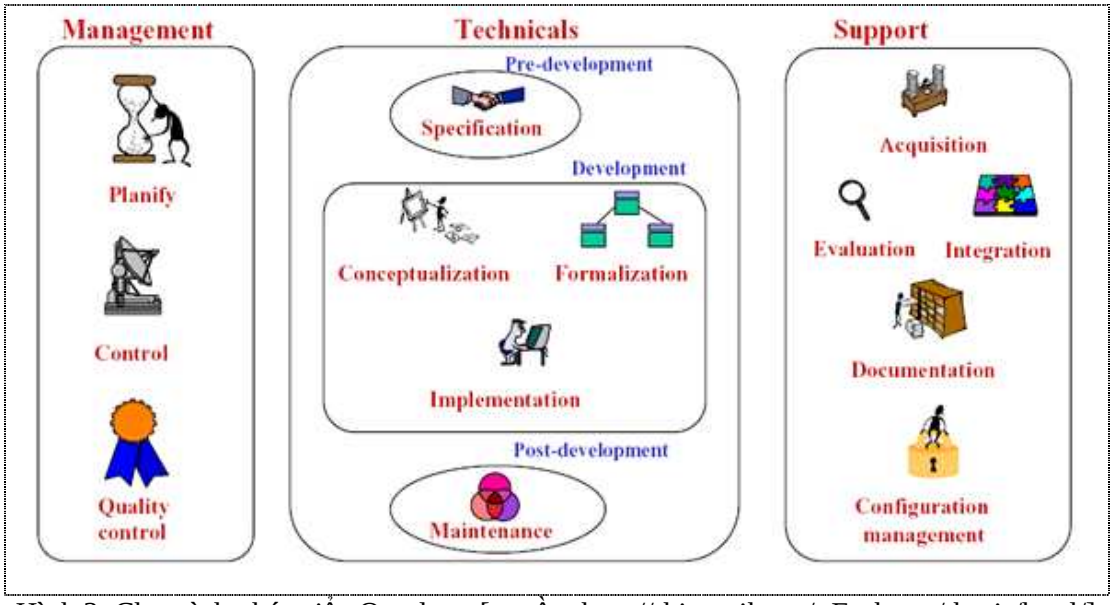
Một vấn đề quan trọng trong sử dụng Ontology là xây dựng Ontology từ nguồn thông tin. Việc xây dựng có thể do con người hoàn toàn thực hiện, bán tự động hoặc tự động. Có nhiều phương pháp đã được đề nghị trong xây dựng Ontology .

MethOntology là một phương pháp xây dựng Ontology mà được tiếp thu nhiều ý kiến từ công nghệ phần mềm. Chu trình phát triển Ontology được dựa trên các hoạt động được định nghĩa trong chuẩn IEEE về phát triển phần mềm. Các hoạt động này tạo thành vòng đời của Ontology qua các giai đoạn (stages) mà Ontology sẽ di chuyển trong suốt vòng đời của nó cũng như các hoạt động sẽ được thực hiện trong mỗi giai đoạn.

Chu trình phát triển Ontology :

Chu trình này đề cập đến các hoạt động nào sẽ được thực hiện khi xây dựng Ontology. Nó định nghĩa ba (3) nhóm hoạt động, được thể hiện trong hình và mô tả chi tiết dưới đây:

- **Các hoạt động quản lý Ontology**: Chúng bao gồm lập lịch (schedule), điều khiển (control) và đảm bảo chất lượng (quality assurance). Hoạt động *lập lịch* xác định các nhiệm vụ sẽ được thực hiện, sự sắp xếp của chúng, thời gian và tài nguyên cần thiết. Hoạt động *điều khiển* đảm bảo nhiệm vụ được hoàn thành theo yêu cầu. Hoạt động *đảm bảo chất lượng* kiểm tra chất lượng của các đầu ra của phương pháp MethOntology (ontology, phần mềm, tài liệu).
- **Các hoạt động hướng phát triển Ontology**: Được nhóm thành các hoạt động tiền phát triển (pre-development), phát triển (development) và sau phát triển (post-development). Trong suốt hoạt động *tiền phát triển*, môi trường nơi Ontology được sử dụng sẽ được nghiên cứu và đây là nghiên cứu về khả năng có thể thực hiện được. Trong giai đoạn *phát triển*, hoạt động *đặc tả* (specification) sẽ xác định tại sao cần xây dựng Ontology, mục đích sử dụng và người dùng. Hoạt động *khái niệm hóa* (conceptualization) cấu trúc hóa tri thức về miền thông tin thành một mô hình có nghĩa ở mức tri thức. Hoạt động *hình thức hóa* (formalization) sẽ chuyển mô hình mức khái niệm thành một mô hình hình thức hoặc mô hình tính toán được. Cuối cùng mô hình tính toán được sẽ được *cài đặt* sử dụng công cụ soạn thảo Ontology. Trong giai đoạn *hậu phát triển*, hoạt động *bảo trì* (maintainance) sẽ cập nhật và chữa lỗi cho Ontology nếu cần thiết và nó có thể dùng lại bởi Ontology hoặc ứng dụng khác.
- **Các hoạt động trợ giúp Ontology**: chúng được thực hiện vào cùng thời điểm với các hoạt động hướng phát triển Ontology. Trong suốt quá trình trợ giúp thì những hoạt động sau xảy ra. Hoạt động *thu thập tri thức* (knowledge acquistion) mà mục đích của nó là thâu nhận tri thức từ các chuyên gia hoặc bằng cách học ontology (bán) tự động. Hoạt động *đánh giá* (evaluation) sẽ phán xét ontology, phần mềm và tài liệu đã phát triển với một khung tham chiếu. Hoạt động *tích hợp* (integration) nếu dùng lại Ontology khác, đi kèm với nó có thể là các hoạt động *trộn* (merging) và *gán* (alignment) Ontology nếu có nhiều ontology được dùng lại và cần kết hợp. Hoạt động *lập tài liệu* (documentation) chi tiết lại mỗi giai đoạn và sản phẩm hoàn thành và hoạt động *quản lý cấu hình* (configuration management) lưu lại theo phiên bản Ontology, phần mềm, tài liệu để điều khiển sự thay đổi.



Hình 3: Chu trình phát triển Ontology [nguồn: http://rhizomik.net/~Eroberto/thesis/html/]

Chu trình phát triển Ontology ở trên xác định các hoạt động được thực hiện. Nó không nói gì về việc chúng được lập lịch như thế nào. Điều này được quyết định bởi phần khác trong phương pháp MethOntology, vòng đời Ontology, đưa ra các giai đoạn mà Ontology sẽ di chuyển qua trong suốt cuộc đời của nó và các hoạt động sẽ được thực hiện.

Chu trình sống của Ontology:

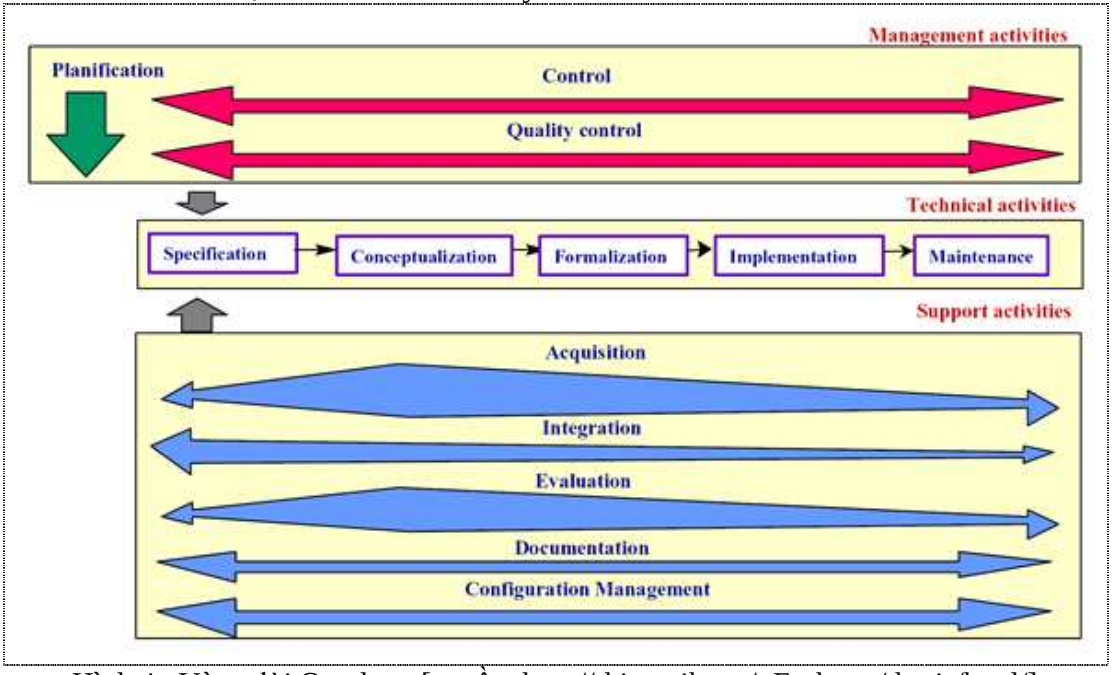
Chu trình sống của Ontology lập lịch cho các hoạt động phát triển Ontology được nêu chi tiết ở trên. Chu trình sống của ontology phát triển theo chu trình và được dựa trên mô hình nguyên mẫu (prototype) tiến hóa. Nó cho phép phát triển tăng trưởng Ontology để đảm bảo việc thẩm định sớm. Mỗi chu trình phát triển bắt đầu bằng việc lập lịch các nhiệm vụ cần thực hiện, xác định yêu cầu và tài nguyên cần thiết. Sau đó hoạt động phát triển được diễn ra, bắt đầu với việc đặc tả. Một cách đồng thời, các hoạt động quản lý, bao gồm điều khiển và đảm bảo chất lượng, và các hoạt động trợ giúp, thu thập tri thức, tích hợp, đánh giá, lập tài liệu và quản lý cấu hình được diễn ra. Chúng xảy ra song song với các hoạt động phát triển.

Trong mỗi chu trình, nguyên mẫu (prototype) Ontology di chuyển qua các hoạt động phát triển, từ đặc tả, qua khái niệm hóa, hình thức hóa và cài đặt tới tận bảo trì mặc dù không cần thiết phải đi qua tất cả. Cuối cùng, nguyên mẫu có thể đủ thành thục để đánh giá và một chu trình mới sẽ được đưa ra dựa trên kết quả của việc đánh giá này. Nếu một chu trình được hoàn thành thì những bước sau sẽ được thực hiện :

1. Đặc tả nguyên mẫu
2. Xây dựng một mô hình khái niệm từ các mẩu thông tin từ quá trình thu thập tri thức.
3. Hình thức hóa mô hình khái niệm
4. Cài đặt Ontology trên một ngôn ngữ biểu diễn Ontology.
5. Bảo trì ontology kết quả mà có thể dẫn tới một chu trình mới nếu ontology là không thích hợp hoặc yêu cầu mới được phát hiện.

Hình dưới đây sẽ thể hiện chu trình sống của Ontology, các hoạt động quản lý và trợ giúp sẽ diễn ra đồng thời với quá trình phát triển. Công sức bỏ ra cho các hoạt

động trợ giúp là không giống nhau trong toàn bộ chu trình, thu thập tri thức, tích hợp và đánh giá là lớn hơn trong suốt quá trình khái niệm hóa Ontology. Lý do có sự khác biệt này là hầu hết tri thức được yêu cầu trong lúc bắt đầu phát triển, các Ontology được tích hợp ở mức khái niệm trước khi cài đặt và cần đánh giá kết quả sớm ở mức khái niệm để tránh lỗi di truyền.



Hình 4 : Vòng đời Ontology [nguồn: http://rhizomik.net/~Eroberto/thesis/html/]

Quá trình phát triển Ontology (Development Process)

Quá trình phát triển bao gồm tất cả các hoạt động tạo ra một nguyên mẫu Ontology thích hợp.

Quá trình đặc tả (Specification): quá trình này xác định mục đích và phạm vi của Ontology. Tại sao Ontology được xây dựng, mục đích là gì và người dùng là ai. Đặc tả có thể là phi hình thức, trong ngôn ngữ tự nhiên hoặc hình thức.

Quá trình khái niệm hóa (Conceptualisation):

Mục đích của hoạt động này là tổ chức và xây dựng cấu trúc tri thức được yêu cầu trong quá trình thu thập tri thức sử dụng một hình thức biểu diễn ngoài độc lập với biểu diễn tri thức và hệ sẽ cài đặt Ontology. Một điểm nhìn phi hình thức của miền thông tin sẽ được chuyển thành mô hình bán hình thức sử dụng biểu diễn trung gian dựa trên hệ thống bảng và đồ thị. Những biểu diễn trung gian này (khái niêm, thuộc tính, quan hệ, axiom và luật) là có giá trị bởi chúng có thể hiểu được bởi các chuyên gia về miền thông tin và nhà phát triển Ontology. Bởi vậy chúng là cầu nối giữa tri giác về miền của con người với ngôn ngữ cài đặt ontology.

Để xây dựng một mô hình khái niệm đầy đủ và thống nhất, hoạt động khái niệm hóa đã định nghĩa một tập các công việc cần thực hiện kế tiếp nhau. Các công việc này làm tăng lên sự phức tạp trong việc biểu diễn mô hình khái niệm. Theo cách này, nó dễ dàng hơn để đảm bảo một mô hình khái niệm đầy đủ và thống nhất:

1. Trước hết cần xây dựng một **tập các thuật ngữ** (terms) sẽ được bao gồm trong ontology, giải thích bằng ngôn ngữ tự nhiên của chúng và tập các từ đồng nghĩa, từ rút gọn. Các thuật ngữ được xác định theo chiến thuật từ giữa ra (middle-out). Gốc của các thuật ngữ cơ bản sẽ được xác định trước, sau

đó chúng sẽ được đặc biệt hóa hay tổng quát hóa theo yêu cầu. Chiến lược này đưa ra một tập cân bằng các thuật ngữ bởi sự chi tiết chỉ đưa ra khi cần thiết và phân loại mức cao hơn được xây dựng tự nhiên.

2. Tiếp đó, các thuật ngữ được phân loại vào một hay nhiều **tập phân cấp các khái niệm** (taxonomies of concept), nơi một khái niệm là trừu tượng của một hay nhiều thuật ngữ. Khái niệm lớp con của quan hệ phân cấp được sử dụng, trong đó: C là lớp con của D khi và chỉ khi mọi thể hiện của C là của D.
3. **Quan hệ nhị phân** (Binary relation) được sử dụng để định nghĩa quan hệ giữa các khái niệm trong một Ontology và với khái niệm của Ontology khác. Quan hệ được quyết định bởi tên và khái niệm nguồn, đích.
4. **Từ điển khái niệm (concept dictionary)** được xây dựng. Nó mô tả mỗi khái niệm bằng cách xác định quan hệ lấy khái niệm này lam miền (domain) và các thể hiện khái niệm cũng như thuộc tính lớp. Các thuộc tính lớp là các thuộc tính của khái niệm.
5. Từ điển khái niệm được chi tiết hóa. Với mỗi quan hệ, nó xác định giới hạn (cardinality), quan hệ ngược (inverse ralation) và các thuộc tính toán học (đối xứng, bắc cầu, duy nhất,...). Các thuộc tính lớp cũng được mô tả về miền khái niệm của chúng, kiểu giá trị, đơn vị đo, phạm vi, giới hạn, giá trị, axiom liên quan và luật suy diễn giá trị của thuộc tính này hoặc dùng nó để suy diễn thuộc tính khác. Ngoài ra có một bảng **hằng số** để định nghĩa những phần không thể thay đổi của miền tri thức.
6. Một khi các khái niệm, quan hệ phân cấp, quan hệ và thuộc tính đã được xác định thì các axiom hình thức và luật được sử dụng để kiểm tra ràng buộc hoặc suy diễn giá trị cho thuộc tính. **Axioms** là các biểu thức logic luôn đúng thông thường được sử dụng để xác định ràng buộc. Chúng được định nghĩa phi hình thức trong dạng văn bản và một cách hình thức dạng logic bậc 1 (ví dụ: >,=,<,∃, ∀, ...). **Luật** (Rules) nói chung được sử dụng để suy diễn tri thức trong Ontology như xác định giá trị thuộc tính, thể hiện quan hệ,...

Quá trình hình thức hóa (Formalisation):

Mục đích của hoạt động này là hình thức hóa mô hình khái niệm. Có những công cụ phát triển Ontology mà tự động chuyển mô hình khái niệm thành các ngôn ngữ Ontology sử dụng bộ dịch. Bởi vậy đây không phải là hoạt động cần nhiều công sức.

Quá trình cài đặt (Implementation):

Hoạt động này xây dựng mô hình máy tính xử lý được sử dụng ngôn ngữ cài đặt Ontology. Có nhiều ngôn ngữ Ontology và chúng không có cùng một khả năng biểu diễn cũng như khả năng suy diễn.

Quá trình đánh giá (Evaluation):

Quá trình này sẽ cập nhật và chữa lỗi cho Ontology nếu Ontology xây dựng không hoạt động như mong muốn hoặc thay đổi yêu cầu trong chu trình phát triển hiện tại hoặc chu trình khác dùng lại Ontology này.

Như vậy phương pháp xây dựng ontology **MethOntology** là rất đầy đủ, mang tính lý thuyết cao. Phương pháp này đã lấy nhiều ý tưởng từ phương pháp phát triển

phần mềm và tạo ra một chu trình sống của Ontology với mô hình phát triển tiến hóa, tăng dần.

Tuy nhiên trong thực tế các bài toán xây dựng Ontology thì không cần thiết phải áp dụng tất cả các giai đoạn trên mà có thể tập trung vào những giai đoạn chủ yếu nhất trong việc tạo ra Ontology. Trong đồ án này, một vài phương pháp mang tính thực hành cao hơn sẽ được giới thiệu:

[1] **Noy và McGuiness (2003)** đã đề nghị một tập các bước để xây dựng Ontology như sau:

- **Bước 1 - Quyết định miền và phạm vi của Ontology**: Để trả lời câu hỏi này, nhà phát triển ontology phải xác định mục đích của ontology, người sử dụng, và thông tin sẽ được lưu trong ontology.
- **Bước 2 – Xem xét sự dùng lại của các Ontology đã có**: Nhà phát triển phải xem xét các ontology đã có trong cùng miền quan tâm. Sự dùng lại sẽ tối thiểu thời gian và công sức của quá trình xây dựng ontology, ngoài ra nó còn đem đến chất lượng cao hơn vì các ontology đã được kiểm thử kĩ càng.
- **Bước 3 – Xác định các thuật ngữ quan trọng của Ontology**: Và xác định các khái niệm mà những thuật ngữ này biểu diễn.
- **Bước 4 – Định nghĩa các lớp và phân cấp các lớp**: Bước này có thể được thực hiện theo một trong các cách tiếp cận (*Uschold và Gruninger, 1996*):
    - Top-down: Bắt đầu với việc định nghĩa nhiều khái niệm chung và tiếp theo chia các khái niệm thành các loại chi tiết hơn.
    - Bottom-up: Bắt đầu với việc định nghĩa các lớp chi tiết hơn và sau đó nhóm các lớp thành các khái niệm tổng quát hơn.
    - Lai (hybrid): giống như trong cách tiếp cận của phương pháp MethOntology.
- **Bước 5 – Định nghĩa các thuộc tính của các lớp**: Các lớp một mình là ít ý nghĩa, cần thiết phải định nghĩa các thuộc tính cho chúng. Đó là thuộc tính của riêng lớp hoặc thuộc tính quan hệ giữa các lớp.
- **Bước 6 – Định nghĩa đặc điểm của thuộc tính**: đó là các giới hạn, miền, phạm vi cho thuộc tính.
- **Bước 7 – Tạo ra các thể hiện**: Bước cuối cùng này là phải đưa vào Ontology các thể hiện cho mỗi lớp, bao gồm cả thuộc tính.

[1] Một phương pháp xây dựng Ontology được đề nghị bởi (**Uschold và KING**, 1995) liên quan đến những giai đoạn dưới đây: *xác định mục đích của Ontology* (tại sao lại xây dựng nó, nó được sử dụng như thế nào, phạm vi người dùng), *xây dựng Ontology*, *đánh giá* và *lập tài liệu*. Trong đó xây dựng Ontology được chia tiếp thành ba (3) bước. Bước đầu tiên là định hình Ontology, các khái niệm chính và quan hệ được xác định, định nghĩa của chúng được viết ra, các thuật ngữ được sử dụng liên quan đến khái niệm và quan hệ cũng được xác định, có sự chấp nhận của chuyên gia về các khái niệm và định nghĩa đó. Bước thứ hai liên quan đến mã hóa Ontology trong một vài ngôn ngữ Ontology. Bước thứ ba liên quan đến việc tích hợp có thể với Ontology đã có.

Khó khăn trong việc xây dựng Ontology:

Việc xây dựng Ontology cần tiêu tốn nhiều thời gian và công sức. Một thách thức chính cho các nhà nghiên cứu là phát triển công cụ để giúp đỡ quá trình này,

đặc biệt giảm thiểu công sức của con người. Đó có thể là các công cụ giúp xây dựng Ontology bán tự động (semi-automatic) hoặc tự động. Chức năng chính của chúng là phân tích văn bản để khám phá các khái niệm và quan hệ cho Ontology. Quá trình này sẽ là tự động nếu không cần sự tham gia của con người và là bán tự động nếu con người chỉ tham gia trong việc xác định các khái niệm và quan hệ chính xác nhất cho miền thông tin mà công cụ đã khám phá ra.

1.3. Ngôn ngữ Ontology

Việc giao tiếp trên Web ngữ nghĩa hay các ứng dụng sử dụng Ontology đòi hỏi phải có sự đồng ý trên một ngôn ngữ Ontology chung. Những ngôn ngữ như thế phải cung cấp sự biểu diễn ngữ nghĩa trên một tập các khái niệm và quan hệ. Một vài ngôn ngữ cũng cho phép biểu diễn chính xác các axioms hoặc quan hệ logic giữa các thuật ngữ cho cùng một mục đích. Sự lựa chọn ngôn ngữ Ontology sẽ quyết định loại cấu trúc tri thức nào mà chúng ta có thể biểu diễn.

Một điều quan trọng là không lẫn lộn giữa những ngôn ngữ này với hệ thống biểu diễn chúng. Hầu hết ngôn ngữ đều được biểu diễn dạng văn bản có cấu trúc kiểu XML (eXtensible Markup Language) hay dạng đồ họa kiểu đồ thị (graph) hoặc lược đồ (diagram). Dạng đồ họa vẫn đóng vai trò quan trọng cho con người trong việc soạn thảo, sử dụng hoặc khái niệm hóa Ontology.

Trong đồ án này sẽ nghiên cứu một vài ngôn ngữ biểu diễn Ontology, đi từ mức độ biểu diễn đơn giản đến phức tạp. Sau đó sẽ nêu lựa chọn ngôn ngữ được sử dụng để xây dựng các Ontology trong đồ án.

1.3.1 RDF/RDFS

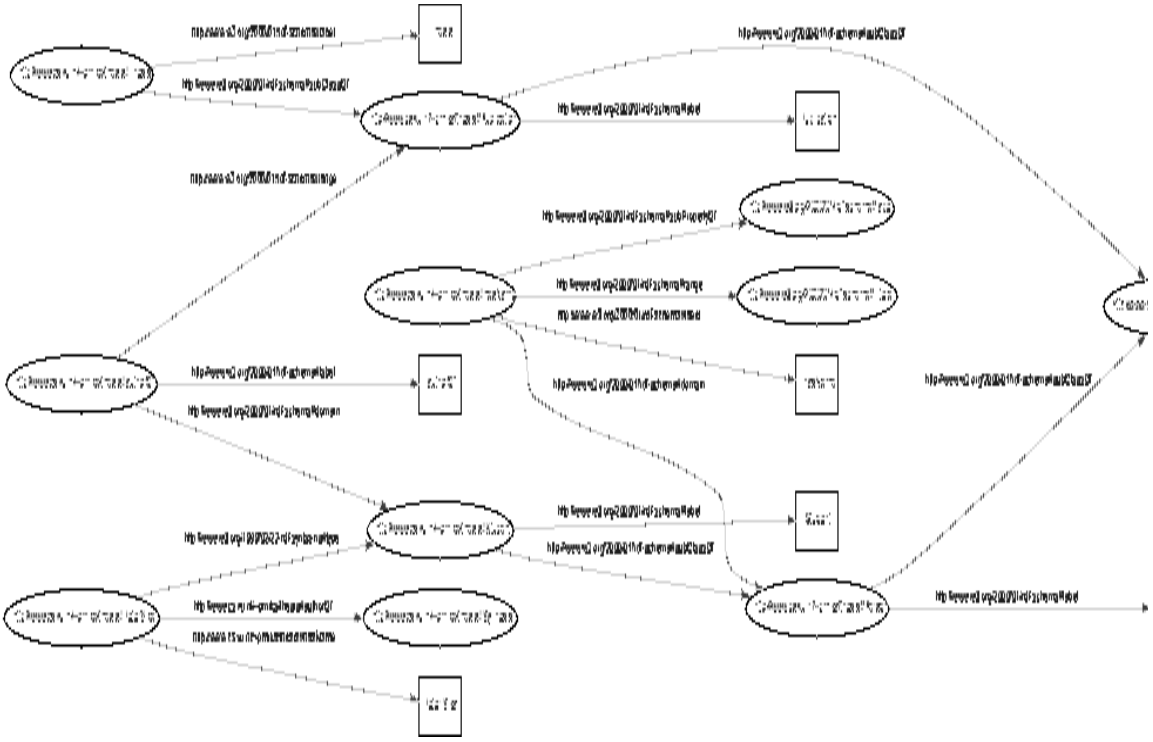
RDF/RDFS là ngôn ngữ biểu diễn ontology được thảo luận đầu tiên vì cấu trúc đơn giản và sự chấp nhận rộng rãi trong cộng đồng Web ngữ nghĩa bởi vì sự phù hợp cũng như khả năng mở rộng của nó.

RDF (Resource Description Framework) là một khuôn dạng biểu diễn tri thức có nguồn gốc là để mô tả siêu thông tin (metainformation) về các tài nguyên trên mạng toàn cầu (WWW) nên có tên là khung mô tả tài nguyên. RDF phiên bản 1.0 là một khuyến cáo chính thức của W3C.

Đơn vị xây dựng cơ bản của RDF là **resources** và **literals**. Một resource có thể là bất cứ thực thể gì cần biểu diễn trong khi một literal là một thực thể tự mô tả, ví dụ một giá trị nào đó. Literal không có định danh trong khi resource được định danh duy nhất và biểu diễn bởi URI (Uniform Resource Identifier). Cấu trúc tri thức được biểu diễn như một tập các xác thực (assertions) hay **statements** trong RDF. Statement được biểu diễn bởi bộ ba {subject, predicate, object}, trong đó resource có thể nằm trong cả ba vị trí nhưng literal chỉ có thể ở object. Một ví dụ của statement mà chúng ta sẽ nhìn thấy sau là {thesis:Thesis, rdfs:subClassOf, thesis:Publication}. RDF thường được biểu diễn dưới dạng đồ thị bằng cách ánh xạ resources, literals thành các nút (nodes) và statements là cung định hướng giữa các nút (Hình 5). Ngoài ra còn có dạng biểu diễn văn bản kiểu XML như trong RDF/XML hay kiểu các bộ ba trong N-triples. Tuy nhiên dạng đồ thị thì không thể được biểu diễn bởi các hệ thống này. Ví dụ dưới đây là một Ontology đơn giản mô tả về một luận văn thạc sĩ. [6] Tập các statements được thể hiện bởi các triples trong Bảng 1 và dạng đồ thị RDF trong Hình 5.

thesis:Thesis	rdfs:subClassOf	thesis:Publication
thesis:Thesis	rdfs:label	"Thesis"
thesis:PeterMika	thesis:authorOf	thesis:MyThesis
thesis:PeterMika	rdf:type	thesis:Student
thesis:PeterMika	thesis:hasName	"Peter Mika"
thesis:Student	rdfs:subClassOf	thesis:Person
thesis:Student	rdfs:label	"Student"
thesis:Publication	rdfs:subClassOf	thesis:Top
thesis:Publication	rdfs:label	"Publication"
thesis:authorOf	rdfs:domain	thesis:Student
thesis:authorOf	rdfs:range	thesis:Publication
thesis:authorOf	rdfs:label	"authorOf"
thesis:Person	rdfs:subClassOf	thesis:Top
thesis:Person	rdfs:label	"Person"
thesis:hasName	rdfs:domain	thesis:Student
thesis:hasName	rdfs:range	rdfs:Literal
thesis:hasName	rdfs:subPropertyOf	rdfs:label
thesis:hasName	rdfs:label	"hasName"

Bảng 1. Các triples từ một Ontology đơn giản



Hình 5: Sơ đồ "đỉnh và cung" của một Ontology đơn giản

RDF-S (RDF Schema) là một mở rộng của RDF để mô tả từ vựng RDF. Dưới đây chúng ta sẽ đi qua bộ từ vựng của RDF-S và tóm tắt ngữ nghĩa của nhiều loại cấu trúc.

rdfs:Resource: Mọi URI hoặc nút trống xuất hiện như là subject, predicate hay object trong một statement đều là thành viên của rdfs:Resource, lớp của tất cả resources.

rdfs:Class: Đây là tập các lớp (Class) và tất cả thành viên của nó đều là con của rdfs:Resource. Các thành viên của rdfs:Class bao gồm rdfs:Resource, rdfs:Literal, rdfs:Property và chính rdfs:Class.
rdfs:Literal: là lớp của tất cả literals.
rdfs:domain và *rdfs:range*: Trong RDF(S), thuộc tính (properties) và lớp là những thực thể tách biệt. rdfs:domain và rdfs:range phục vụ để định nghĩa liên kết giữa một thuộc tính và lớp mà nó áp dụng (domain) với lớp để nó lấy giá trị (range).
rdfs:subClassOf: Cung cấp khả năng biểu diễn quan hệ phân cấp các lớp trong RDF(S).
rdfs:Property: là lớp biểu diễn các property trong RDF(S). RDF(S) cũng cung cấp cách biểu diễn quan hệ cha-con là rdfs:subPropertyOf.

1.3.2 OWL

RDF/RDFS chưa có đủ các cấu trúc để mô tả Ontology với những cấu trúc phức tạp. Chính vì thế đã có nhiều ngôn ngữ Ontology được xây dựng và phát triển. Trước đây có ngôn ngữ DAML+OIL được tài trợ phát triển bởi các tổ chức của cả Mỹ và châu Âu. Ngôn ngữ được sử dụng rộng rãi hiện nay và được tổ chức W3C bảo trợ là **OWL** (Ontology Web Language). OWL là ngôn ngữ biểu diễn tri thức được sử dụng để xây dựng Ontology. OWL dựa trên RDF/RDFS, được viết ra dưới dạng cú pháp RDF/XML nhưng nó cung cấp khả năng biểu diễn đầy đủ và giầu có hơn RDF/RDFS. OWL là được coi là một trong các công nghệ nền tảng đằng sau Web ngữ nghĩa và được sử dụng trong cả lĩnh vực thương mại và học thuật.

Trên thực tế, OWL bao gồm ba loại: *OWL Lite*, *OWL DL* và *OWL Full*.
- *OWL Lite* cung cấp tập các đặc tính giới hạn, có thể chưa đầy đủ cho mọi ứng dụng nhưng được thiết kế để đảm bảo tốc độ xử lý tương đối hiệu quả. OWL Lite cung cấp quan hệ phân cấp lớp (classification hierarchy) và các ràng buộc tập hợp (cardinality) đơn giản (như ràng buộc 0,1). Chính vì việc hạn chế trong biểu diễn mà OWL Lite không được sử dụng phổ biến.
- *OWL DL*, tập cha của OWL Lite, được dựa trên một dạng của logic bậc một (first order logic) là Description Logic. OWL DL được thiết kế để cung cấp khả năng diễn đạt cao nhất mà vẫn giữ được sự đầy đủ về mặt tính toán, sự đơn định và khả năng áp dụng thuật toán suy diễn thực tế. Chính vì các khả năng này mà OWL DL được sử dụng phổ biến nhất.
- *OWL Full* là dạng ngôn ngữ đầy đủ nhất của OWL. OWL Full được thiết kế để đảm bảo sự tương thích với RDFS và không hạn chế gì trong việc biểu diễn tri thức. Ví dụ OWL Full cho phép một lớp vừa là một tập các thể hiện lại vừa có thể là một thể hiện; điều này không được cho phép trong OWL DL. Chính vì khả năng biểu diễn mạnh mẽ như vậy nên OWL Full có nhược điểm là có thể không đơn định và không tìm thuật toán suy diễn đầy đủ.

Có sự tương thích xuôi giữa các dạng của OWL với nhau:
- Mọi Ontology OWL Lite hợp lệ đều là Ontology OWL DL hợp lệ.
- Mọi Ontology OWL DL hợp lệ đều là Ontology OWL Full hợp lệ.

1.3.3 Cú pháp Ontology [10]

OWL xây dựng trên RDF/RDFS và sử dụng cú pháp dựa trên XML của RDF. Cấu trúc cú pháp của một tài liệu OWL bao gồm các phần :
- **Namespaces** : để xác định tập các bộ từ vựng đặc biệt nào sẽ được sử dụng trong Ontology.

Ví dụ :

```
<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#">
```

- **Ontology Headers**: Đây là tập hợp các xác thực (assertions) về ontology được nhóm trong thẻ <owl:Ontology>. Những thông tin trong ontology header bao gồm: chú thích <rdfs:comment>, quản lý phiên bản <owl:priorVersion> và khai báo Ontology được import <owl:import>.

Ví dụ :

```
<owl:Ontology rdf:about="">
    <rdfs:comment>An example OWL ontology</rdfs:comment>
    <owl:priorVersion rdf:resource="http://www.mydomain.org/ontology"/>
    <owl:imports rdf:resource="http://www.mydomain.org/persons"/>
    <rdfs:label>University Ontology</rdfs:label>
</owl:Ontology>
```

<owl:import> sẽ đưa ra danh sách các Ontology mà nội dung của chúng được sử dụng trong Ontology này. Thuộc tính này có tính bắc cầu.

- **Class** : Class được định nghĩa trong OWL sử dụng <owl:Class>.

Ví dụ định nghĩa class Student là lớp con của class Person như dưới đây:

```
<owl:Class rdf:ID=”student”>
    <rdfs:subClassOf rdf:resource=”#person”/>
</owl:Class>
```

- **Individuals**: Các thể hiện được gọi trong OWL là Individual. Mọi thể hiện trong OWL cần có một định danh duy nhất, được chỉ ra trong rdf:ID

Ví dụ cách khai báo một thể hiện:

```
<Stock rdf:ID=”Stock_ABT”>
```

trong đó lớp Stock đã được khai báo đâu đó trong Ontology.

Cách khai báo trên sẽ hoàn toàn tương đương với:

```
<rdf:Description rdf:about="#Stock_ABT">
    <rdf:type rdf:resource=”Stock”/>
</rdf:Description>
```

rdf:type là một thuộc tính RDF để gắn một thể hiện với lớp mà nó thuộc về.

- **Property** : Một property là một quan hệ nhị phân. Hai loại thuộc tính đã được định nghĩa trong OWL.
  - ✓ datatype properties: quan hệ giữa các thể hiện của lớp và RDF literal và loại dữ liệu XML Schema.
  - ✓ object properties: quan hệ giữa các thể hiện của hai lớp.

Ví dụ định nghĩa thuộc tính loại Datatype property:

```
<owl:DatatypeProperty rdf:ID="age"
    rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#nonNegativeInteger"/
>
</owl:DatatypeProperty>
```

OWL không có bất kì loại dữ liệu được định nghĩa trước nào, thay vào đó nó cho phép sử dụng loại dữ liệu XML Schema mà người dùng có thể tự định nghĩa. Ví dụ các loại dữ liệu XML Schema xây dựng sẵn được sử dụng thường xuyên là :

xsd:string xsd:int xsd:float xsd:dateTime xsd:boolean

Ví dụ khai báo object property thể hiện quan hệ *teach* giữa *Lecturer* và *Student* :

```
<owl:ObjectProperty rdf:ID=”teach”>
    <rdfs:domain rdf:resource=”#Lecturer”/>
    <rdfs:range rdf:resource=”#Student”/>
</owl:ObjectProperty>
```

- **Các đặc tính thuộc tính** : Cho phép xác định những đặc điểm của thuộc tính, cung cấp một cơ chế mạnh để cải thiện khả năng suy diễn trên thuộc tính.
  - ✓ owl:TransitiveProperty định nghĩa thuộc tính bắc cầu.

Cho thuộc tính P, được xác định là có tính bắc cầu cho x,y,z bất kì khi:

P(x,y) and P(y,z) sẽ suy diễn ra p(x,z)

Ví dụ:

Thuộc tính `locatedIn` có tính bắc cầu:

```
<owl:ObjectProperty rdf:ID="locatedIn">
  <rdf:type rdf:resource="&owl;TransitiveProperty" />
  <rdfs:domain rdf:resource="&owl;Thing" />
  <rdfs:range rdf:resource="#Region" />
</owl:ObjectProperty>

<Region rdf:ID="SantaCruzMountainsRegion">
  <locatedIn rdf:resource="#CaliforniaRegion" />
</Region>

<Region rdf:ID="CaliforniaRegion">
  <locatedIn rdf:resource="#USRegion" />
</Region>
```

Vì `SantaCruzMountainsRegion` được `locatedIn` trong `CaliforniaRegion`, nên nó phải `locatedIn` trong `USRegion`, vì `locatedIn` là thuộc tính bắc cầu.

- ✓ owl:SymmetricProperty định nghĩa thuộc tính đối xứng.

Cho thuộc tính P, được xác định là đối xứng cho x,y bất kì khi:

P(x,y) nếu và chỉ nếu P(y,x)

Ví dụ:

Thuộc tính `adjacentRegion` là đối xứng, trong khi `locatedIn` thì không.

```
<owl:ObjectProperty rdf:ID="adjacentRegion">
  <rdf:type rdf:resource="&owl;SymmetricProperty" />
  <rdfs:domain rdf:resource="#Region" />
  <rdfs:range rdf:resource="#Region" />
</owl:ObjectProperty>

<Region rdf:ID="MendocinoRegion">
  <locatedIn rdf:resource="#CaliforniaRegion" />
  <adjacentRegion rdf:resource="#SonomaRegion" />
</Region>
```

Vùng `MendocinoRegion` là kề cận với vùng `SonomaRegion` và ngược lại. Vùng `MendocinoRegion` nằm trong vùng `CaliforniaRegion` nhưng không có điều ngược lại.

- ✓ owl:FunctionalProperty định nghĩa thuộc tính chỉ có 1 giá trị.

Cho thuộc tính P, được xác định là functional với x,y bất kì khi:

P(x,y) và P(x,z) thì y=z.

Ví dụ: Trong Ontology giao dịch cổ phiếu, thuộc tính `marketName` là functional vì mỗi thị trường giao dịch chỉ có một tên duy nhất.

```
<owl:FunctionalProperty rdf:ID="marketName">
  <rdfs:label xml:lang="vi">Tên thị trường GD</rdfs:label>
  <rdfs:comment rdf:datatype="&xsd#string" >Ten cua thi truong giao dich
  </rdfs:comment>
  <rdfs:domain rdf:resource="#StockMarket"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
```

- **Giới hạn thuộc tính** : Thêm vào đặc tính thuộc tính thì chúng ta có thể xác định các giới hạn trong range của thuộc tính. Điều này được thực hiện trong ngữ cảnh của một thẻ <owl:Restriction>
  - ✓ *allValuesFrom*, *someValuesFrom* : giới hạn *allValuesFrom* yêu cầu rằng cho mỗi thể hiện của lớp có các thể hiện của thuộc tính xác định thì nó lấy tất cả giá trị của lớp được chỉ ra trong mệnh đề allValuesFrom.

Ví dụ dưới đây thể hiện giới hạn các khóa học năm đầu *chỉ được dạy* bởi các giáo sư:

```
<owl:Class rdf:about="#firstYearCourse">
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#isTaughtBy"/>
            <owl:allValuesFrom rdf:resource="#Professor"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
```

someValuesFrom có ý nghĩa tương tự.

- ✓ Cardinality: đây là giới hạn số lượng chính xác cho thuộc tính.

Ví dụ: yêu cầu tất cả khóa học được dạy bởi *ít nhất 1* người

```
<owl:Class rdf:about="#course">
<rdfs:subClassOf>
    <owl:Restriction>
        <owl:onProperty rdf:resource="#isTaughtBy"/>
        <owl:minCardinality
        rdf:datatype="&xsd;nonNegativeInteger">1</owl:minCardinality>
    </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
```

- ✓ hasValue (Chỉ có từ OWL DL trở lên): giới hạn này cho phép chúng ta xác định lớp được dựa trên sự tồn tại của giá trị thuộc tính đặc biệt. Bởi vậy, một thực thể sẽ là thành viên của một lớp bất cứ khi nào ít nhất một trong các giá trị thuộc tính của nó bằng với giá trị trong hasValue.

Ví dụ Khóa học Toán phải được dạy bởi ít nhất giáo sư David:

```
<owl:Class rdf:about="#mathCourse">
<rdfs:subClassOf>
    <owl:Restriction>
```

```
                <owl:onProperty rdf:resource="#isTaughtBy"/>
                <owl:hasValue rdf:resource="#David"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        </owl:Class>
```

1.3.4 Lựa chọn ngôn ngữ Ontology

Như đã nói ở trên thì việc lựa chọn ngôn ngữ Ontology sẽ ảnh hưởng đến các cấu trúc Ontology sẽ có thể biểu diễn. RDF-S có thể được sử dụng để tạo ra Ontology nhưng mục đích của nó là nhẹ (lightweight) với ít sức mạnh biểu diễn hơn OWL. Giống như OWL, RDF-S bao gồm các lớp và thuộc tính, cũng như ràng buộc miền (domain) và phạm vi (range) trên các thuộc tính. Nó cung cấp quan hệ thừa kế cho cả lớp và thuộc tính nhưng thiếu nhiều đặc tính như: kiểu dữ liệu, enumerations và khả năng định nghĩa thuộc tính phức tạp hơn. OWL ra đời đã cung cấp những khả năng biểu diễn phức tạp và đầy đủ hơn. Tuy nhiên cũng cần có sự cân nhắc giữa ba loại OWL-Lite, OWL-DL và OWL-Full. Lựa chọn giữa OWL-Lite và OWL-DL phụ thuộc vào mở rộng của người dùng đến các cấu trúc giới hạn mạnh hơn được cung cấp trong OWL-DL. Các bộ suy diễn trong OWL-Lite sẽ có độ phức tạp thấp hơn nếu suy diễn trong OWL-DL. Sự lựa chọn giữa OWL-DL và OWL-Full phụ thuộc chính vào việc mở rộng phía người dùng yêu cầu các đặc tính siêu mô hình (meta-modelling) của RDF Schema (ví dụ, định nghĩa lớp của lớp). Tuy nhiên suy diễn với OWL-Full có thể không đơn định. Trong đồ án này, các Ontology được xây dựng cần có khả năng biểu diễn và suy diễn mạnh, do đó lựa chọn OWL-DL là phù hợp và cho khả năng mở rộng trong tương lai.

1.3. Ứng dụng Ontology

1.3.1 Quản lý tri thức

Quản lý tri thức tập trung vào việc thu thập, truy nhập và bảo trì nguồn tri thức bên trong một tổ chức. Ngoài ra đó có thể là quản lý tri thức của một lĩnh vực nào đó mà nguồn thông tin của nó được lấy từ mạng lưới các trang web vô cùng rộng lớn. Hầu hết thông tin hiện tại có giá trị trong các dạng cấu trúc yếu như văn bản, âm thanh và hình ảnh. Từ khía cạnh quản lý tri thức, công nghệ hiện tại mới chỉ giới hạn trong các khu vực dưới đây :

- **Tìm kiếm thông tin**. Việc tìm kiếm thông tin hiện nay mới chỉ giới hạn ở tìm theo từ khóa (key-word) mà nổi bật là các máy tìm kiếm : Google, Yahoo, MSN. Việc tìm kiếm này sẽ gây nhiều khó khăn cho người dùng trong việc tinh lọc kết quả từ rất nhiều kết quả trả về. Nghịch lý là một phần rất lớn của kết quả trả về chẳng hề liên quan đến vấn đề mà người dùng đang tìm kiếm. Với công nghệ ngữ nghĩa áp dụng Ontology, chúng ta có thể lọc ngay ra các kết quả phù hợp vấn đề người dùng quan tâm thậm chí không đúng theo từ khóa họ đánh vào. Nó có thể đưa ra thêm rất nhiều thông tin khác liên quan hoặc gợi ý từ khóa cho người dùng.
- **Trích thông tin** : Với một khối lượng tài liệu thì việc trích ra các thông tin theo domain quan tâm là rất tốn thời gian và công sức của con người. Để cải tiến chúng ta có thể áp dụng xây dựng Ontology cho domain quan tâm và tự động trích ra thông tin, chú thích nó rồi chuyển đến cho người dùng.

- **Bảo trì thông tin** : Công nghệ Ontology sẽ được sử dụng trong việc tự động loại bỏ thông tin không toàn vẹn và thông tin quá hạn.

Tóm lại khi ứng dụng Ontology, tri thức sẽ được tổ chức trong không gian khái niệm theo ý nghĩa của nó mà máy tính cũng có thể xử lý được.

1.3.2 Thương mại điện tử nhà sản xuất đến khách hàng (Business-to-Customer B2C)

Thương mại điện tử B2C sẽ là loại hình thương mại chính của người dùng Web. Thông thường khách hàng sẽ đến thăm một vài cửa hàng trực tuyến, tìm mặt hàng họ muốn và đặt hàng. Một cách lý tưởng, khách hàng sẽ thu thập thông tin về giá cả, chất lượng của một mặt hàng từ nhiều cửa hàng và chọn cái tốt nhất. Tuy nhiên việc này rất mất thời gian và khách hàng thường chỉ viếng thăm một hoặc rất ít cửa hàng.

Để giải quyết tình huống này các công cụ cho phép thăm các cửa hàng trực tuyến, dưới dạng các shopbot hay software agent được phát triển. Và để các shopbot trích ra giá cả và thông tin về sản phẩm được chính xác thì công nghệ ngữ nghĩa và Ontology cần được sử dụng. Khi sử dụng Ontology các agent có thể trích ra chính xác thông tin về sản phẩm, có thể lấy dữ liệu từ không chỉ cửa hàng mà còn từ các nguồn khác, tăng khả năng tự động và tùy biến theo yêu cầu khách hàng.

1.3.3 Thương mại điện tử nhà sản xuất đến nhà tiêu thụ (Business-to-Business B2B)

Với công nghệ Ontology các nhà kinh doanh có thể ủy thác việc trao đổi thương lượng mua bán cho các agent tự động bởi chúng sẽ cùng nói chuyện trên một domain chung, là domain mà Ontology tạo ra. Sự khác nhau về thuật ngữ sẽ được giải quyết, dữ liệu kinh doanh sẽ được trao đổi sử dụng các dịch vụ giao dịch. Ngoài ra, tùy thuộc và Ontology được nạp cho nó mà mỗi agent sẽ biết thực hiện các giao dịch sao cho phù hợp nhất với mong muốn của người dùng hoặc phát tín hiệu thông báo nếu nó không tự quyết định được.

1.3.4 Data mining

Khai phá dữ liệu là quá trình trích ra các thông tin thích hợp, có ích từ một số lượng lớn dữ liệu. Ngày nay, một trong những vấn đề thách thức và quan trọng nhất trong khai phá dữ liệu là định nghĩa trước các tri thức. Các thông tin ngữ cảnh này có thể giúp lựa chọn thông tin thích hợp, các đặc tính hoặc công nghệ, làm giảm không gian dự đoán, biểu diễn kết quả đầu ra theo một cách tổng hợp nhất và cải thiện toàn bộ tiến trình khai phá dữ liệu. Bởi vậy chúng ta cần một mô hình khái niệm để biểu diễn tri thức này. Với Ontology chúng ta có thể biểu diễn tri thức tiến trình khai phá dữ liệu và tri thức miền. Kết quả là Ontology trở thành cơ sở cho việc khai phá tự động một cách hiệu quả nguồn thông tin tri thức. Khi chúng ta biểu diễn và bao gồm tri thức trong quá trình khai phá qua Ontology thì chúng ta có thể chuyển khai phá dữ liệu vào khai phá tri thức. Hiện nay, trên cơ sở Ontology đã có nhiều ứng dụng khai phá tri thức trong các lĩnh vực như: phân loại tài liệu, khai phá tri thức trong các ngành y học, thuốc, thông tin địa lý, ...

1.4. Các lợi ích của Ontology :

Ontology là hệ biểu diễn ngữ nghĩa nên cung cấp khả năng hiểu tri thức cho máy tính, tạo điều kiện xây dựng các hệ thống xử lý tự động.

Ontology có khả năng tích hợp. Điều này cho phép những Ontology về các miền thông tin khác nhau có thể tích hợp, trộn lẫn, ánh xạ với nhau, tạo sự liên giao tiếp giữa các ứng dụng khác nhau.

Ontology có thể dùng lại. Điều này giúp cho việc xây dựng Ontology mới được giảm nhẹ công sức, đồng thời tạo ra khả năng mở rộng dịch vụ của các ứng dụng một cách dễ dàng nhất là dịch vụ Web (Web services)

Ontology có thể được chia sẻ bởi nhiều ứng dụng, không phụ thuộc nền hệ điều hành. Một ứng dụng trên hệ Unix có thể hoàn toàn sử dụng dữ liệu trong Ontology được xây dựng bởi ứng dụng trên hệ Windows.

Ontology có thể giải quyết vấn đề đa ngôn ngữ. Các ứng dụng có thể xử lý dữ liệu được viết bởi các ngôn ngữ khác nhau (ví dụ tiếng Anh và tiếng Pháp) một cách giống nhau với một Ontology ánh xạ các thuộc tính của các ngôn ngữ này đến cùng một ngữ nghĩa. Ví dụ của ứng dụng này là trong tìm kiếm đồng thời trên nhiều ngôn ngữ.

Ontology biểu diễn theo ngữ nghĩa con người nên người dùng hoàn toàn hiểu được và có thể soạn thảo, tinh lọc, tùy biến Ontology theo ý muốn. Điều này giúp tạo ra các ứng dụng thông minh, mang tính tùy biến cao, là hướng phát triển của tương lai.

Chương 2. Các công cụ để xây dựng, xử lý Ontology

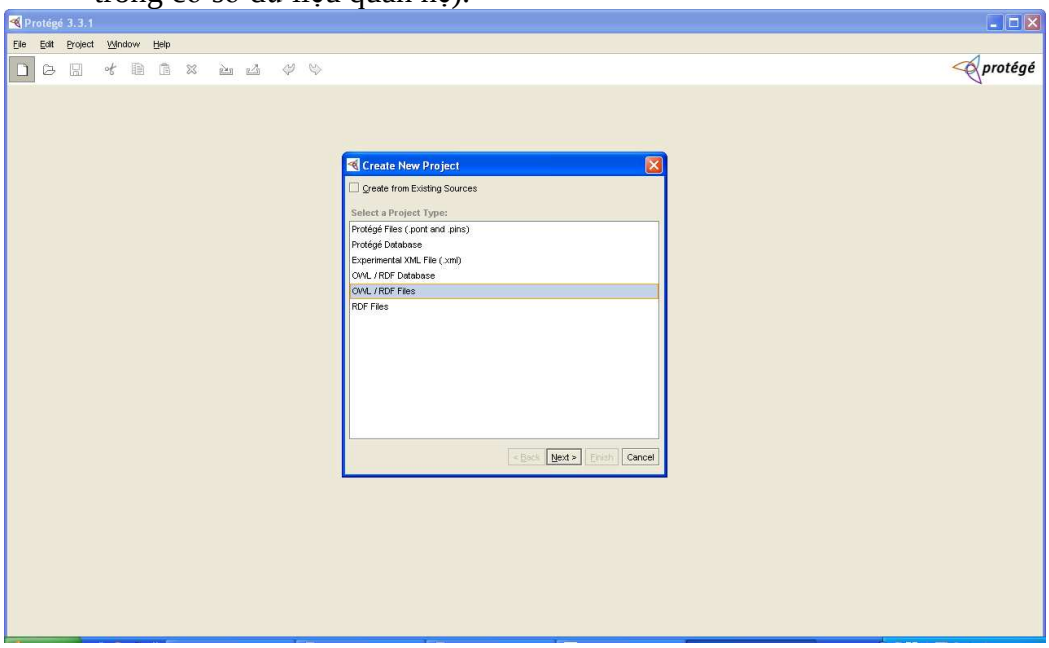
2.1. Protégé

Protégé là một trình soạn thảo Ontology miễn phí, mã nguồn mở và là framework dựa trên tri thức. Nền Protégé cung cấp 2 cách để mô hình ontology :

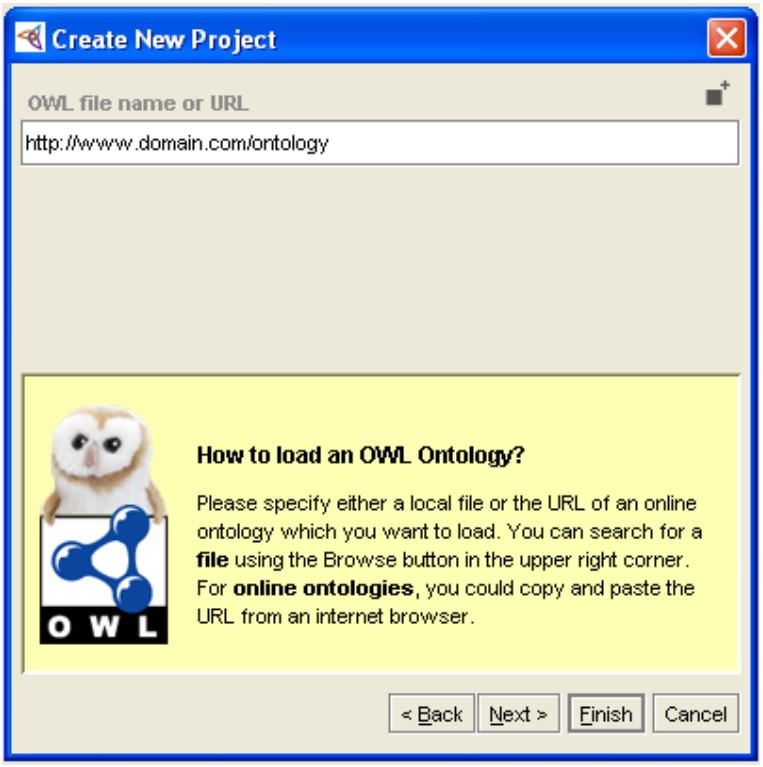
- ✓ Trình soạn thảo Protégé-Frames cho phép người dùng xây dựng và phân phối Ontology dựa trên frame, theo giao thức Open Knowledge Base Connectivity (OKBC).
- ✓ Trình soạn thảo Protégé-OWL là một mở rộng của Protégé để trợ giúp cho việc xây dựng OWL. Protégé -OWL cho phép người sử dụng : nạp và lưu các ontology dạng OWL và RDF; soạn thảo và quan sát các lớp, thuộc tính; thực hiện suy diễn, . . . Protégé-OWL được tích hợp chặt chẽ với Jena và cung cấp API để nhà phát triển tùy biến các thành phần giao diện và dịch vụ Web ngữ nghĩa bất kì.

Trong đồ án này, Protégé-OWL được sử dụng để soạn thảo Ontology. Những hướng dẫn trong việc tạo Ontology dùng Protégé-OWL như sau :

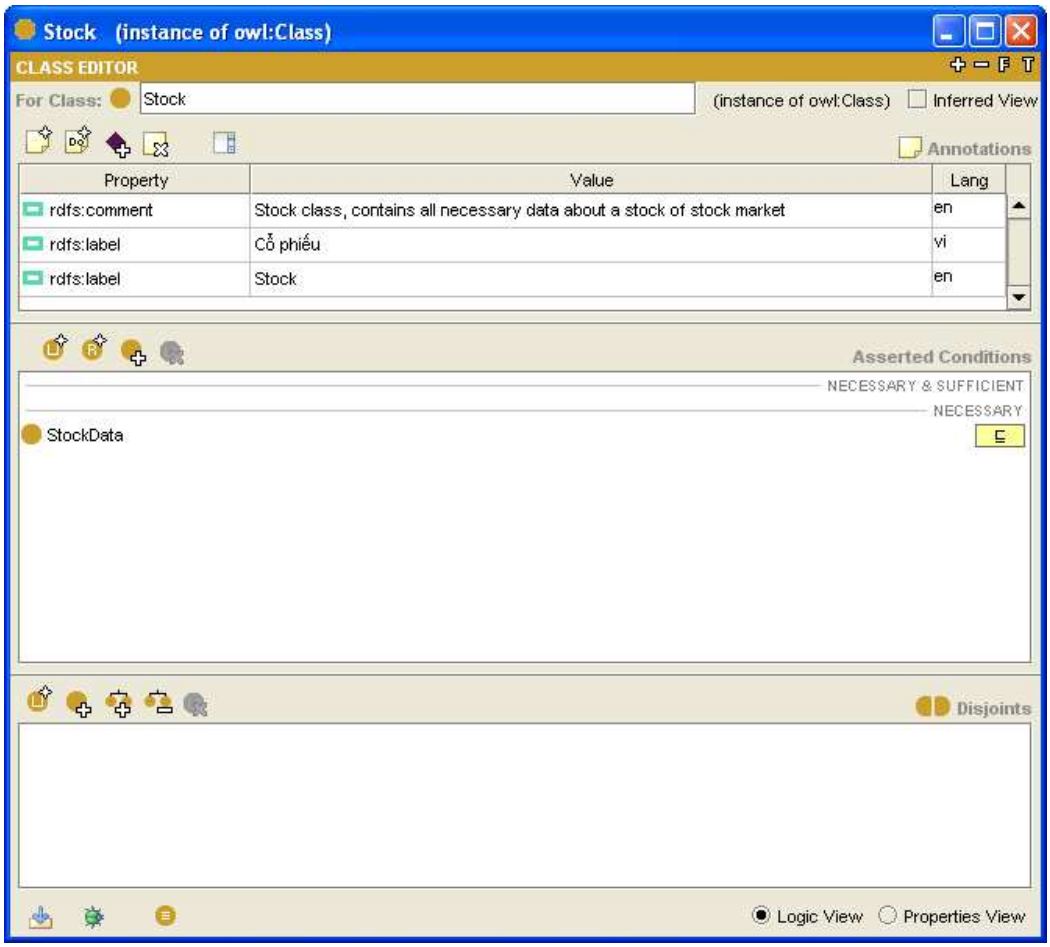
- Trước hết bộ soạn thảo Protégé-OWL được gắn với quá trình cài đặt "đầy đủ" của Protégé. Protégé-OWL sẽ được cài đặt như một "plug-in" cho Protégé.
- Tạo mới một project trong Protégé-OWL ta sẽ đến một hộp thoại yêu cầu chọn kiểu lưu trữ cho ontology : "OWL/RDF Files" hay "OWL/RDF Database". Chọn kiểu lưu trữ cho phù hợp với ứng dụng (lưu trong files hay trong cơ sở dữ liệu quan hệ).



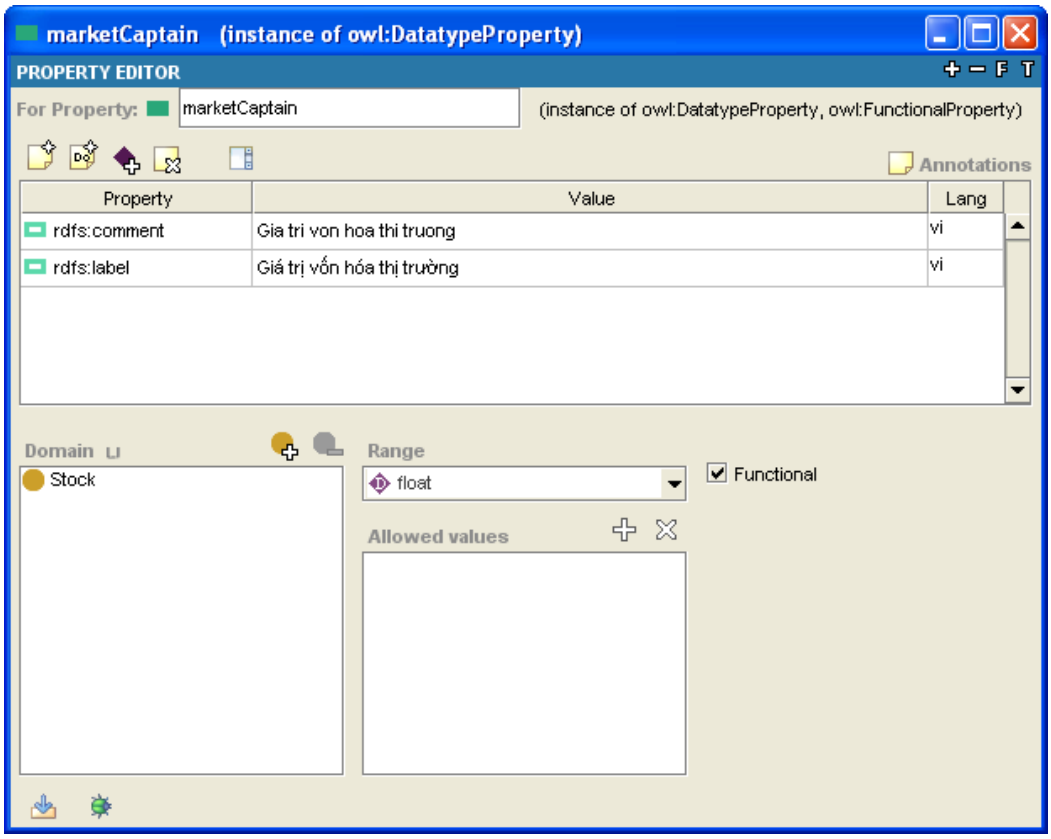
- Bên cạnh đó Protege-OWL còn cho phép chúng ta nạp một Ontology vào bằng cách chỉ đến file OWL hoặc RDF. File này có thể cục bộ hoặc nằm trên Web và được nạp vào thông qua URL.



- Khi đã tạo một project, chúng ta có thể thực hiện xây dựng các lớp, thuộc tính và thể hiện cho Ontology. Protege-OWL mở sẵn các tab : Metadata, OWL Classes, Properties, Individuals. Tab Metadata để xây dựng phần Header cho Ontology như : khai báo namespace, khai báo ontology được import, các chú thích cho Ontology (rdfs:label, owl:versionInfo, rdfs:comment, owl:backwardCompatibleWith, ...). Tab OWL Classes cho phép xây dựng cây phân cấp các lớp như : tạo lớp cùng với các chú thích cho nó, tạo lớp con (subClass), tạo lớp anh em (siblingClass).



- ✓ Tab Properties cho phép xây dựng các thuộc tính kiểu đối tượng (ObjectProperty) và kiểu dữ liệu (DatatypeProperty). Mỗi thuộc tính được xác định : tên, chú thích, Domain, Range, tính chất (Transitive, Functional, Symmetric, InverseFunctional).



✓ Tab Individual cho phép tạo thể hiện cho mỗi lớp.

Sau khi tạo xong Ontology thì Protege-OWL cho phép ghi lại ontology đó dưới dạng RDF/XML. Ngoài ra còn có thể ghi ra các dạng khác như : N3, N-Triple, Turtle.

- Các khả năng đáng chú ý khác mà Protege-OWL hỗ trợ : Truy vấn Ontology bằng SPARQL, thể hiện hình ảnh sơ đồ Ontology. Việc truy vấn được thực hiện trực tiếp trên Protege-OWL với câu lệnh truy vấn do người dùng đánh vào. Hình ảnh sơ đồ Ontology được thể hiện nhờ plugin Jambalaya.

2.2. Jena và ngôn ngữ truy vấn SPARQL

2.2.1 Jena

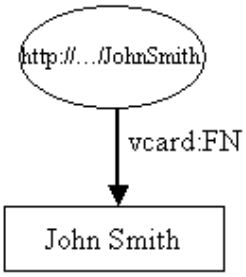
Jena là một Java framework để xây dựng các ứng dụng web ngữ nghĩa. Nó cung cấp môi trường lập trình cho RDF, RDF-S, OWL, SPARQL và bao gồm một máy suy diễn dựa trên luật (rule-based inference engine). Jena là mã nguồn mở và được thực hiện dưới chương trình HP Labs Semantic Web Programme.

Jena framework bao gồm :

✓ Một RDF API
✓ Đọc và viết RDF trong các dạng RDF/XML, N3 và N-Triples
✓ Một OWL API
✓ Lưu trữ trong bộ nhớ hoặc trong cơ sở dữ liệu
✓ Máy truy vấn SPARQL

a. RDF API : Jena cung cấp một Java API cho RDF. Resource Description Framework (RDF) là một chuẩn (theo khuyến nghị của W3C) để mô tả tài nguyên.

RDF được biểu diễn dưới dạng đồ thị các nút và cung. Ví dụ một mô tả về con người đơn giản (dạng VCARD) được biểu diễn trong RDF :



Jena cung cấp Java API có thể được sử dụng để tạo ra và xử lý đồ thị RDF như thế này. Jena có các lớp để biểu diễn đồ thị, tài nguyên, thuộc tính và literals. Các giao tiếp biểu diễn tài nguyên, thuộc tính và literals được gọi lần lượt là Resource, Property và Literal. Trong Jena, một đồ thị được gọi là một mô hình và biểu diễn bởi giao tiếp Model.

Mã tạo ra đồ thị hay mô hình trên như sau :

```
// some definitions
static String personURI    = "http://somewhere/JohnSmith";
static String fullName     = "John Smith";

// create an empty Model
Model model = ModelFactory.createDefaultModel();

// create the resource
Resource johnSmith = model.createResource(personURI);

// add the property
 johnSmith.addProperty(VCARD.FN, fullName);
```

Bắt đầu bằng việc tạo ra một Model rỗng, sử dụng phương thức createDefaultModel của lớp ModelFactory để tạo ra một mô hình nằm trong bộ nhớ. Jena cung cấp các cài đặt khác của giao tiếp Model, ví dụ tạo ra mô hình trong cơ sở dữ liệu, cũng dựa trên lớp ModelFactory.

- **Statement:** Mỗi cung trong đồ thị RDF được gọi là một *statement*. Mỗi statement mang một mô tả về tài nguyên. Một statement có 3 phần :
  subject là tài nguyên mà từ đó cung hướng ra ngoài
  predicate là thuộc tính trên nhãn của cung
  object là tài nguyên hay literal mà cung trỏ đến

Jena cung cấp giao tiếp Statement để truy nhập đến subject, predicate, object của một statement. Cách lấy ra dữ liệu như sau :

```
// list statements in model
StmtIterator si = model.listStatements();

// print out the predicate, subject and object of each statement
while (si.hasNext()) {
    Statement stmt      = si.nextStatement();  // get next statement
    Resource  subject   = stmt.getSubject();     // get the subject
    Property  predicate = stmt.getPredicate();   // get the predicate
    RDFNode   object    = stmt.getObject();      // get the object

    System.out.print(subject.toString());
    System.out.print(" " + predicate.toString() + " ");
}
```

- **Đọc và viết RDF** : Jena có các phương thức để đọc và viết RDF dạng XML. Cách này được sử dụng để lưu RDF dạng file và sau đó đọc trở lại.

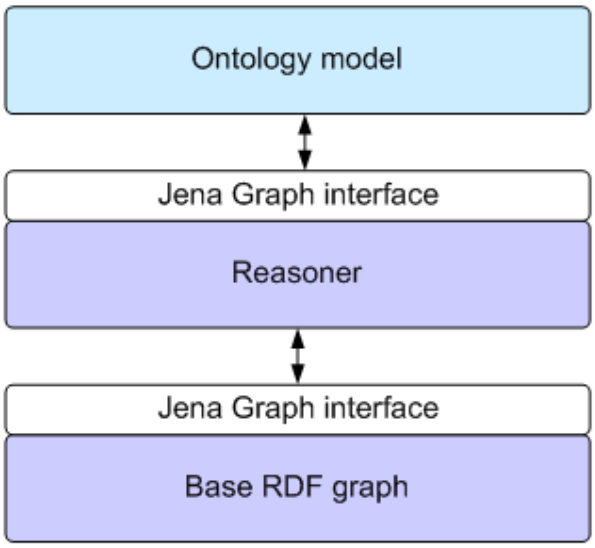
Viết RDF dạng XML : `model.write(System.out);`
Nếu muốn viết dạng N3 hay N-Triples thì cung cấp tham số cho phương thức write.
Đọc và RDF :

```
// create an empty model
 Model model = ModelFactory.createDefaultModel();

 // use the FileManager to find the input file
 InputStream in = FileManager.get().open( inputFileName );
if (in == null) {
    throw new IllegalArgumentException(
                                "File: " + inputFileName + " not found");
}

// read the RDF/XML file
model.read(in, "");
```

b. Jena 2 Ontology API :

Jena cung cấp API để sử dụng các ngôn ngữ tạo Ontology như RDF(S), DAML+OIL và OWL. Jena Ontology API là độc lập với ngôn ngữ, tên lớp Java không đề cập đến ngôn ngữ sử dụng (ví dụ OntClass hay ObjectProperty). Để thể hiện sự khác biệt giữa các khả năng biểu diễn khác nhau, mỗi ngôn ngữ có một *Profile*, mà nó sẽ đưa ra danh sách các cấu trúc được phép và URI của các lớp và thuộc tính. Profile được gắn với một mô hình ontology và được thể hiện trong lớp *OntModel*, lớp mở rộng của *Model*.

Statement mà đối tượng Java về ontology nhìn thấy phụ thuộc vào statement trong đồ thị RDF bên dưới và statement có thể được suy diễn.



Hình 6 : Cấu trúc Ontology Model trong Jena

- **Tạo ra ontology model** : Ontology model được tạo ra thông qua lớp ModelFactory của Jena. Cách đơn giản nhất là :

```
OntModel m = ModelFactory.createOntologyModel();
```

Cách này sẽ tạo ra một ontology model với các thiết đặt mặc định. Đó là: ngôn ngữ OWL, lưu trong bộ nhớ và suy diễn RDFS. Có thể có nhiều lựa chọn trong việc tạo model. Ví dụ tạo model trong bộ nhớ với ngôn ngữ DAML như sau:

```
OntModel m = ModelFactory.createOntologyModel( OntModelSpec.DAML_MEM );
```

- **Xử lý tài liệu ontology và import** :

Thuật ngữ tài liệu được sử dụng để chỉ một ontology được viết ra dưới dạng RDF/XML hay N3. Chúng ta nạp một tài liệu Ontology vào trong ontology model theo cách tương tự như mô hình Jena thông thường, sử dụng phương thức *read*. Có thể đọc từ địa chỉ URL, một stream hay một reader (file hay cơ sở dữ liệu).

Để xử lý việc import một tài liệu Ontology khác thì lớp DocumentManager cần được sử dụng. Mỗi tài liệu Ontology được import sẽ được giữ trong một cấu trúc đồ thị tách biệt. Mỗi Ontology model có một bộ quản lý tài liệu (document manager) để thực hiện và xử lý tài liệu ontology.
Ví dụ đọc vào ontology về camera được lưu cục bộ:

```
OntModel m = ModelFactory.createOntologyModel();
OntDocumentManager dm = m.getDocumentManager();
dm.addAltEntry( "http://www.xfront.com/owl/ontologies/camera/",
                "file:" + JENA + "src-examples/data/camera.owl"    );
m.read( "http://www.xfront.com/owl/ontologies/camera/" );
```

- **Class** : class là khối xây dựng cơ bản của một ontology. Một class được biểu diễn trong Jena bởi đối tượng OntClass. Lấy ra một class bằng cách gọi *getOntClass()* trên ontology model:

```
OntClass camera = m.getOntClass( camNS + "Camera" );
```

Ngoài ra ta có thể tạo ra một đối tượng OntClass trực tiếp:

```
OntClass pinCamera = m.createClass( camNS + "PinholeCamera" );
```

- **Property** : Một thuộc tính trong một ontology model là mở rộng của lớp Java RDF Property, lớp biểu diễn nó là OntProperty.

Ví dụ truy nhập thuộc tính của một class:

```
OntModel newM = ModelFactory.createOntologyModel();
OntClass Camera = newM.createClass( camNS + "Camera" );
OntClass Body = newM.createClass( camNS + "Body" );

ObjectProperty part = newM.getObjectProperty( camNS + "part" );
DatatypeProperty body = newM.getDatatypeProperty( camNS + "body" );
```

Trong Jena, ObjectProperty và DatatypeProperty là các loại con của OntProperty.

- **Instances (individuals)** :

Một lớp được coi là tập các thể hiện (trong Jena gọi là Individual). Thế nên cách lấy ra các thể hiện của một lớp là:

```
OntClass c = m.getOntClass(NS + “aClass”);
ExtendedIterator ei = c.listIndividuals();
while(ei.hasNext){
```

```
        Individual i = ei.getIndividual();
}
```

c. Lưu trữ trong Cơ sở dữ liệu :

Jena cho phép lưu dữ liệu RDF trong cơ sở dữ liệu. Nó hỗ trợ nhiều cơ sở dữ liệu phổ biến như: MySQL, HSQLDB, PostgreSQL, Oracle và Microsoft SQL Server.

Ví dụ tạo ra mô hình lưu giữ trong cơ sở dữ liệu như sau :

```
// database URL
String M_DB_URL         = "jdbc:mysql://localhost/test";
// User name
String M_DB_USER        = "test";
// Password
String M_DB_PASSWD      = "";
// Database engine name
String M_DB = "MySQL";
// JDBC driver
String M_DBDRIVER_CLASS = "com.mysql.jdbc.Driver";
// load the the driver class
Class.forName(M_DBDRIVER_CLASS);

// create a database connection
IDBConnection conn = new DBConnection(M_DB_URL, M_DB_USER, M_DB_PASSWD,
M_DB);

// create a model maker with the given connection parameters
ModelMaker maker = ModelFactory.createModelRDBMaker(conn);
// create a default model
Model defModel = maker.createDefaultModel();
...
// Open existing default model
Model defModel = maker.openModel();

// or create a named model
Model nmModel = maker.createModel("MyNamedModel");
...
// or open a previously created named model
Model prvModel = maker.openModel("AnExistingModel");
```

2.2.2 Ngôn ngữ truy vấn SPARQL

SPARQL là một ngôn ngữ truy vấn và giao thức để truy cập tài liệu RDF. Là "ngôn ngữ truy vấn", SPARQL hướng dữ liệu nghĩa là nó chỉ truy vấn thông tin được giữ trong mô hình mà không cung cấp sự suy diễn nào. Tất nhiên khi được sử dụng trong Jena thì chính mô hình Jena đã cung cấp khả năng suy diễn trong đó.

Hầu hết các dạng của câu truy vấn SPARQL chứa một tập hợp các mẫu bộ ba (triple patterns) được gọi là mẫu đồ thị cơ bản (basic graph pattern). Các mẫu bộ ba này giống như bộ ba của một RDF statement ngoại trừ rằng subject, predicate hay object có thể là các biến. Một mẫu đồ thị cơ bản đối sánh được với một đồ thị con của dữ liệu RDF khi mà các RDF term từ đồ thị con đó có thể thay thế cho biến và kết quả trả về là đồ thị RDF tương đương với đồ thị con.

Ví dụ dưới đây chỉ ra một câu truy vấn SPARQL tìm tiêu đề (title) của một quyển sách từ đồ thị dữ liệu đã cho. Câu truy vấn gồm 2 phần: mệnh đề SELECT xác định

biến xuất hiện trong kết quả truy vấn, và mệnh đề WHERE cung cấp mẫu đồ thị cơ bản để đối sánh đồ thị dữ liệu.
Dữ liệu:

```
<http://example.org/book/book1> <http://purl.org/dc/elements/1.1/title>
"SPARQL Tutorial" .
```

Truy vấn :

```
SELECT ?title
WHERE
{
  <http://example.org/book/book1>
<http://purl.org/dc/elements/1.1/title> ?title .
}
```

Kết quả :

title
"SPARQL Tutorial"

- **Đối sánh RDF Literals** :

- Với mã ngôn ngữ : `SELECT ?v WHERE { ?v ?p "cat"@en }`
- Với kiểu dữ liệu số : `SELECT ?v WHERE { ?v ?p 42 }`
-Với kiểu dữ liệu bất kì : `SELECT ?v WHERE { ?v ?p "abc"^^<http://example.org/datatype#specialDatatype> }`

- **Xây dựng đồ thị RDF**:

SPARQL có một vài dạng truy vấn. Dạng SELECT trả lại giá trị. Dạng CONSTRUCT trả lại một đồ thị RDF. Đồ thị được xây dựng dựa trên một template để tạo ra các bộ ba RDF theo mẫu đồ thị cơ bản :

```
PREFIX foaf:   <http://xmlns.com/foaf/0.1/>
PREFIX org:    <http://example.com/ns#>

CONSTRUCT { ?x foaf:name ?name }
WHERE  { ?x org:employeeName ?name }
```

- **Ràng buộc giá trị**:

Câu lệnh SPARQL FILTER sẽ cung cấp khả năng lọc giá trị cho các biến trong câu truy vấn.

- Lọc theo giá trị xâu:

```
PREFIX  dc:  <http://purl.org/dc/elements/1.1/>
SELECT  ?title
WHERE   { ?x dc:title ?title
  FILTER regex(?title, "^SPARQL")
 }
```

- Lọc theo giá trị số :

```
PREFIX  dc:  <http://purl.org/dc/elements/1.1/>
PREFIX  ns:  <http://example.org/ns#>
SELECT  ?title ?price
WHERE   { ?x ns:price ?price .
  FILTER (?price < 30.5)
  ?x dc:title ?title . }
```

- **Sử dụng SPARQL trong Jena**:

API để sử dụng SPARQL trong Jena là gói `com.hp.hpl.jena.query`. Các gói khác chứa các phần khác của hệ thống truy vấn như: máy thực hiện (engine), bộ

phân tích (parser), testing,... Hầu hết ứng dụng sẽ chỉ cần gói chính, chỉ có những yêu cầu như xây dựng câu truy vấn theo chương trình hay thay đổi hành vi của máy thực hiện truy vấn thì mới cần các gói khác.

Các lớp sử dụng chính trong gói `com.hp.hpl.jena.query` là :

- ✓ *Query* : một lớp biểu diễn cho chính câu truy vấn. Đối tượng của lớp *Query* thông thường được tạo ra bằng cách gọi một trong các phương thức của *QueryFactory* mà chúng cung cấp sự truy nhập đến các bộ phân tích câu truy vấn.
- ✓ *QueryExecution* : biểu diễn cho một sự thực hiện câu truy vấn
- ✓ *QueryExecutionFactory* : một nơi để lấy về thể hiện của *QueryExecution*.
- ✓ *DatasetFactory* : một nơi để tạo ra *Dataset*, bao gồm tạo ra *DataSource* (một *Dataset* có thể cập nhật)

Cho câu truy vấn SELECT :

- ✓ *QuerySolution* : một kết quả đơn của truy vấn
- ✓ *ResultSet* : tất cả *QuerySolution*, là một bộ lặp
- ✓ *ResultSetFormatter* : chuyển một *ResultSet* về nhiều dạng; text, RDF Graph hay XML.

Ví dụ xây dựng và thực hiện câu lệnh truy vấn trên một model:

```
import com.hp.hpl.jena.query.* ;
Model model = ... ;
String queryString = " .... " ;
Query query = QueryFactory.create(queryString) ;
QueryExecution qexec = QueryExecutionFactory.create(query, model) ;
try {
 ResultSet results = qexec.execSelect() ;
 for ( ; results.hasNext() ; )
 {
  QuerySolution solv = results.nextSolution() ;
  RDFNode x = solv.get("?varName");  // Get a result variable by name.
  Resource r = solv.getResource("?varR"); // Get a result variable - must
be a resource
  Literal l = soln.getLiteral("?varL") ; // Get a result variable - must
be a literal
 }
} finally { qexec.close() ; }
```

Ví dụ xây dựng và thực hiện câu truy vấn CONSTRUCT:

```
Query query = QueryFactory.create(queryString) ;
QueryExecution qexec = QueryExecutionFactory.create(query, model) ;
Model resultModel = qexec.execConstruct() ;
qexec.close() ;
```

Chương 3. Nghiên cứu về thị trường chứng khoán Việt Nam

3.1. Các khái niệm cơ bản của thị trường chứng khoán

3.1.1 Các khái niệm

Thị trường chứng khoán trong điều kiện của nền kinh tế hiện đại, được quan niệm là nơi diễn ra các hoạt động giao dịch mua bán chứng khoán trung và dài hạn. Việc mua bán này được tiến hành ở thị trường sơ cấp khi người mua mua được chứng khoán lần đầu từ những người phát hành, và ở những thị trường thứ cấp khi có sự mua đi bán lại các chứng khoán đã được phát hành ở thị trường sơ cấp. Như vậy, xét về mặt hình thức, thị trường chứng khoán chỉ là nơi diễn ra các hoạt động trao đổi, mua bán, chuyển nhượng các loại chứng khoán, qua đó thay đổi chủ thể nắm giữ chứng khoán.

1. Chức năng cơ bản của thị trường chứng khoán

- Huy động vốn đầu tư cho nền kinh tế
- Cung cấp môi trường đầu tư cho công chúng
- Tạo tính thanh khoản cho các chứng khoán
- Đánh giá hoạt động của doanh nghiệp
- Tạo môi trường giúp Chính phủ thực hiện các chính sách vĩ mô

2. Các chủ thể tham gia thị trường chứng khoán

Các tổ chức và cá nhân tham gia thị trường chứng khoán có thể được chia thành các nhóm sau: nhà phát hành, nhà đầu tư và các tổ chức có liên quan đến chứng khoán.

a) Nhà phát hành

Nhà phát hành là các tổ chức thực hiện huy động vốn thông qua thị trường chứng khoán. Nhà phát hành là người cung cấp các chứng khoán - hàng hoá của thị trường chứng khoán.

- Chính phủ và chính quyền địa phương là nhà phát hành các trái phiếu Chính phủ và trái phiếu địa phương.
- Công ty là nhà phát hành các cổ phiếu và trái phiếu công ty.
- Các tổ chức tài chính là nhà phát hành các công cụ tài chính như các trái phiếu, chứng chỉ thụ hưởng... phục vụ cho hoạt động của họ.

b) Nhà đầu tư

Nhà đầu tư là những người thực sự mua và bán chứng khoán trên thị trường chứng khoán. Nhà đầu tư có thể được chia thành 2 loại: nhà đầu tư cá nhân và nhà đầu tư có tổ chức.

- Các nhà đầu tư cá nhân
- Các nhà đầu tư có tổ chức

c) Các tổ chức kinh doanh trên thị trường chứng khoán

- Công ty chứng khoán
- Quỹ đầu tư chứng khoán
- Các trung gian tài chính

d) Các tổ chức có liên quan đến thị trường chứng khoán

- Cơ quan quản lý Nhà nước
- Sở giao dịch chứng khoán
- Hiệp hội các nhà kinh doanh chứng khoán
- Tổ chức lưu ký và thanh toán bù trừ chứng khoán
- Công ty dịch vụ máy tính chứng khoán

- Các tổ chức tài trợ chứng khoán
- Công ty đánh giá hệ số tín nhiệm...

3. Các nguyên tắc hoạt động cơ bản của thị trường chứng khoán

Thị trường chứng khoán hoạt động theo các nguyên tắc cơ bản sau:
- Nguyên tắc công khai
- Nguyên tắc trung gian
- Nguyên tắc đấu giá

4. Cấu trúc và phân loại cơ bản của thị trường chứng khoán

Thị trường chứng khoán là nơi diễn ra các giao dịch, mua bán những sản phẩm tài chính (cổ phiếu, trái phiếu, các khoản vay ngân hàng... có kỳ hạn trên 1 năm). Sau đây là một số cách phân loại TTCK cơ bản:

a) Căn cứ vào sự luân chuyển các nguồn vốn

Thị trường chứng khoán được chia thành thị trường sơ cấp và thị trường thứ cấp.

Thị trường sơ cấp: là thị trường mua bán các chứng khoán mới phát hành. Trên thị trường này, vốn từ nhà đầu tư sẽ được chuyển sang nhà phát hành thông qua việc nhà đầu tư mua các chứng khoán mới phát hành.

Thị trường thứ cấp: là nơi giao dịch các chứng khoán đã được phát hành trên thị trường sơ cấp, đảm bảo tính thanh khoản cho các chứng khoán đã phát hành.

b) Căn cứ vào phương thức hoạt động của thị trường

Thị trường chứng khoán được phân thành thị trường tập trung (Sở giao dịch chứng khoán) và phi tập trung (thị trường OTC).

c) Căn cứ vào hàng hoá trên thị trường

Thị trường chứng khoán cũng có thể được phân thành các thị trường: thị trường cổ phiếu, thị trường trái phiếu, thị trường các công cụ chứng khoán phái sinh.

- Thị trường cổ phiếu: thị trường cổ phiếu là thị trường giao dịch và mua bán các loại cổ phiếu, bao gồm cổ phiếu thường, cổ phiếu ưu đãi.
- Thị trường trái phiếu: thị trường trái phiếu là thị trường giao dịch và mua bán các trái phiếu đã được phát hành, các trái phiếu này bao gồm các trái phiếu công ty, trái phiếu đô thị và trái phiếu chính phủ.
- Thị trường các công cụ chứng khoán phái sinh

Thị trường các chứng khoán phái sinh là thị trường phát hành và mua đi bán lại các chứng từ tài chính khác như: quyền mua cổ phiếu, chứng quyền, hợp đồng quyền chọn...

5. Bản cáo bạch

Khi phát hành chứng khoán ra công chúng, công ty phát hành phải công bố cho người mua chứng khoán những thông tin về bản thân công ty, nêu rõ những cam kết của công ty và những quyền lợi cơ bản của người mua chứng khoán...để trên cơ sở đó người đầu tư có thể ra quyết định đầu tư hay không. Tài liệu phục vụ cho mục đích đó gọi là Bản cáo bạch hay Bản công bố thông tin.

Bản cáo bạch là một tài liệu rất quan trọng.Với tư cách là một nhà đầu tư, Bản cáo bạch là phương tiện giúp bạn đánh giá mức độ sinh lời và triển vọng của công ty trước khi bạn quyết định có đầu tư vào công ty hay không. Một quyết định thiếu thông tin có thể làm bạn phải trả giá đắt.

Bản cáo bạch thường gồm 8 mục chính sau:

- ✓ Trang bìa;

- ✓ Tóm tắt Bản cáo bạch;
- ✓ Các nhân tố rủi ro;
- ✓ Các khái niệm;
- ✓ Chứng khoán phát hành;
- ✓ Các đối tác liên quan tới đợt phát hành;
- ✓ Tình hình và đặc điểm của tổ chức phát hành;
- ✓ Phụ lục.

Bản cáo bạch nêu đầy đủ các thông tin về tình hình tài chính kinh doanh của công ty trong quá khứ, hiện tại cũng như dự định tương lai. Nó nêu các điểm mạnh, lĩnh vực mà công ty đang và sẽ theo đuổi, dự báo các khó khăn và cách vượt qua.

3.1.2 Cổ phiếu

Khi một công ty gọi vốn, số vốn cần gọi đó được chia thành nhiều phần nhỏ bằng nhau gọi là cổ phần. Người mua cổ phần gọi là cổ đông. Cổ đông được cấp một giấy chứng nhận sở hữu cổ phần gọi là cổ phiếu và chỉ có công ty cổ phần mới phát hành cổ phiếu. Như vậy, cổ phiếu chính là một chứng thư chứng minh quyền sở hữu của một cổ đông đối với một công ty cổ phần và cổ đông là người có cổ phần thể hiện bằng cổ phiếu.

Thông thường hiện nay các công ty cổ phần thường phát hành 02 dạng cổ phiếu: Cổ phiếu thường và cổ phiếu ưu đãi.

Một *cổ phiếu phổ thông* đại diện cho quyền sở hữu một phần công ty (represent a proportional ownership interest in a corporation). Nếu một công ty có 100 cổ phiếu đang lưu hành (outstanding stock) và bạn sở hữu một trong số đó thì có nghĩa là bạn sở hữu 1/100 công ty. Nếu công ty có 1.000.000 cổ phiếu đang lưu hành và bạn nắm giữ 1.000 cổ phiếu thì bạn sở hữu 1.000/1.000.000 hay 1/1.000 công ty.

Thông thường, một công ty có thể thay đổi số lượng cổ phiếu đang lưu hành bằng cách bán các *cổ phiếu bổ sung* (selling additional shares) hay mua lại và huỷ bỏ một phần các cổ phiếu đã phát hành trước đó (buying back and cancelling some of the shares previously issued). Trong cả hai trường hợp trên, tỷ lệ sở hữu của các cổ đông cũ trong công ty đều thay đổi.

Nghiệp vụ *tách và gộp cổ phiếu* là việc làm tăng hoặc giảm số cổ phiếu đang lưu hành của một công ty cổ phần mà không làm thay đổi vốn điều lệ, vốn cổ phần hay toàn bộ giá trị thị trường tại thời điểm tách hoặc gộp cổ phiếu.

Tách cổ phiếu sẽ làm tăng số lượng cổ phiếu đang lưu hành và làm giảm mệnh giá cổ phiếu tương ứng với tỷ lệ tách, do đó giá cổ phiếu trên thị trường cũng sẽ giảm tương ứng và giao dịch được thực hiện dễ dàng hơn. Việc tách cổ phiếu thường được thực hiện khi giá cổ phiếu trên thị trường tăng quá cao làm cho các giao dịch sẽ khó thực hiện và điều này sẽ làm giảm tính thanh khoản của cổ phiếu.

Ngược lại, trường hợp gộp cổ phiếu sẽ làm giảm số lượng cổ phiếu đang lưu hành, mệnh giá cổ phiếu tăng lên và giá thị trường của cổ phiếu cũng tăng lên tương ứng với tỷ lệ gộp cổ phiếu.

3.1.3 Giao dịch chứng khoán

1. Thời gian giao dịch: Từ 8h30-11h00 vào tất cả các ngày làm việc trong tuần (trừ các ngày nghỉ theo qui định tại Bộ Luật Lao động).

2. Giá tham chiếu:
a. Giá tham chiếu của cổ phiếu là bình quân gia quyền các giá thực hiện qua phương thức giao dịch báo giá của ngày có giao dịch gần nhất.
b. Đối với các cổ phiếu mới niêm yết hoặc cổ phiếu bị tạm ngừng giao dịch trong ngày đầu tiên giao dịch hoặc ngày giao dịch trở lại sẽ giao dịch không biên độ. Trong ngày giao dịch tiếp theo, giá tham chiếu của cổ phiếu này sẽ được tính như mục (a) ở trên.
3. Biên độ dao động giá:
- Biên độ dao động giá trong ngày giao dịch đối với cổ phiếu là ±10% đối với sàn Hà Nội (HASTC) và ±5% đối với sàn thành phố Hồ Chí Minh (HOSE).
- Tại thời điểm viết báo cáo này thì biên độ giá đã được điều chỉnh, tại sàn Hà Nội là 3%, còn tại sàn HOSE là 2%.
- Không áp dụng biên độ dao động giá đối với các giao dịch trái phiếu.
4. Hiệu lực của lệnh:
Trong phiên giao dịch, lệnh giới hạn được nhập vào hệ thống giao dịch có hiệu lực cho đến hết phiên hoặc cho đến khi lệnh bị huỷ trên hệ thống.
5. Nguyên tắc giao dịch: Các giao dịch phải được thực hiện thông qua công ty chứng khoán thành viên của Trung tâm GDCK Hà Nội hoặc Thành phố Hồ Chí Minh.
- Trước tiên, để thực hiện giao dịch nhà đầu tư phải có tài khoản giao dịch chứng khoán tại một công ty chứng khoán là thành viên của HASTC hoặc HOSE.
- Khi đặt lệnh mua bán chứng khoán, nhà đầu tư phải đảm bảo đủ tỉ lệ ký quỹ trên tài khoản. Cụ thể là, khi đặt lệnh bán thì nhà đầu tư phải có đủ số chứng khoán trong tài khoản, còn khi đặt lệnh mua thì nhà đầu tư phải có đủ số tiền kí quỹ theo thoả thuận với công ty chứng khoán.
6. Phương thức giao dịch: Giao dịch báo giá và giao dịch thỏa thuận
[Theo Trung tâm giao dịch chứng khoán Hà Nội : http://www.hastc.org.vn]

3.2 Các chỉ số chứng khoán

Chỉ số giá cổ phiếu là thông tin thể hiện giá chứng khoán bình quân hiện tại so với giá bình quân thời kỳ gốc đã chọn. Giá bình quân thời kỳ gốc thường được lấy là 100 hoặc 1.000. Ví dụ chỉ số HASTC-Index lấy gốc là 100 điểm tại thời điểm ngày 14/7/2005. Công thức tính :

$$\text{HASTC-Index} = \frac{\text{Tổng giá trị thị trường hiện tại (GTn)}}{\text{Tổng giá trị thị trường gốc (GTo)}} \times 100$$

Các chỉ số quan trọng trong phân tích chứng khoán :
- Nhóm hệ số giá trị : EPS, P/E, P/B, D/E.
- Nhóm các hệ số tài chính : ROE, ROA,...

1. Chỉ số EPS – Earning Per Share
EPS = thu nhập trên một cổ phiếu

Tổng thu nhập sau thuế - tổng số cổ tức của CP ưu đãi

EPS = ----------

Tổng số CP đang lưu hành

Nhận xét :

- ✓ EPS cho phép khả năng sinh lợi của công ty trên mỗi cổ phần mà cổ đông đóng góp là bao nhiêu.
- ✓ Chỉ số EPS càng cao thì khả năng sinh lời càng lớn và ngược lại
- ✓ So sánh chỉ số EPS qua các thời kì sẽ cho chúng ta biết được tốc độ tăng trưởng của doanh nghiệp đang phân tích

2. Chỉ số P/E – Thị giá/Thu nhập cổ phiếu (CP)

Thị giá hiện tại của CP

P/E = ----------

Thu nhập của CP (EPS)

Nhận xét :

- ✓ P/E cho biết nhà đầu tư sẵn sàng trả giá cao hơn cho CP bao nhiêu lần.P/E càng cao thì chứng tỏ CP được thị trường đánh giá cao và ngược lại. So sánh chỉ số P/E giữa các công ty cùng ngành để đánh giá giá trị của CP mình đang quan tâm.
- ✓ Theo quan điểm "bảo thủ" , P/E dưới 10 thì nên mua, nếu đang nắm CP có P/E từ 10-12 thì nên mua tiếp, P/E từ 12-18 cũng nên mua nếu thì trường đang tăng trưởng tốt, P/E trên 18 thì nên bán. Tuy nhiên P/E có thể được chấp nhận cao nếu lợi nhuận công ty tốt.

3. Chỉ số P/B – Price to Book

P/B = thị giá CP / giá trị sổ sách

Giá trị hiện tại của CP (stock price)

P/B = ----------

Tổng giá trị tài sản – giá trị tài sản vô hình và nợ

Nhận xét :

- ✓ P/B là công cụ để tìm kiếm được các CP có giá thấp mà phần lớn thị trường bỏ qua. Nếu một công ty đang bán CP với giá thấp hơn giá trị ghi sổ của nó (tức là P/B < 1) thì có 2 trường hợp xảy ra : hoặc là thị trường đang nghĩ giá trị tài sản công ty bị thổi phồng quá mức, hoặc thu nhập trên tài sản của công ty là quá thấp.
- ✓ Chỉ số P/B có ích khi xem xét các công ty có vốn lớn hoặc công ty tài chính còn giá trị ghi sổ thì không có ý nghĩa nhiều với công ty dịch vụ vì tài sản hữu hình của họ không cao.

4. Chỉ số D/E – Debt to Equity

D/E = Nợ/vốn chủ sở hữu

Nợ phải trả

D/E = ----------

Vốn chủ sở hữu

Nhận xét :

- ✓ D/E cho biết tài sản công ty chủ yếu từ nguồn nào, nợ hay vốn chủ sở hữu

- ✓ D/E dùng để phân tích tỉ lệ nợ của công ty, khả năng thanh toán nợ. Khi D/E nhỏ tức là nợ ít nhưng cũng sẽ giảm cơ hội chớp thời cơ, D/E lớn thì nợ nhiều, có khả năng không trả nổi hoặc lợi nhuận phải giành để trả nợ.

5. Chỉ số ROE – Return on Equity
ROE = Lợi nhuận sau thuế trên vốn chủ sở hữu
Nhận xét :

- ✓ ROE được dùng để đo lường xem công ty sử dụng vốn nhà đầu tư tốt đến mức nào. Thông thường công ty tốt thì phải có ROE cao hơn các công ty cùng ngành, thể hiện khả năng kiếm nhiều tiền hơn trên đồng vốn đầu tư
- ✓ ROE trung bình toàn thị trường là 20.55%. Nói chung, nên tránh công ty có ROE nhỏ hơn 15%, các công ty hàng đầu thường có ROE từ 20-30%.

6. Chỉ số ROA – Lợi nhuận trên tài sản.
Nhận xét : ROA thể hiện khả năng quản lý tốt của công ty. Nên tìm công ty có ROA cao hơn cùng ngành, trung bình toàn thị trường là 11.69%.

7. Ngoài ra còn có nhiều chỉ số khác liên quan đến hoạt động sản xuất kinh doanh của công ty.
Trong các chỉ số trên thì các chỉ số liên quan trực tiếp đến **giá cả cổ phiếu** là : **EPS**, **P/E** và **P/B**.

Chương 4. Giới thiệu về hệ thống phân phối thông tin chứng khoán tự động trên nền tảng Ontology (hệ thống BKS)

4.1. Cách ứng dụng Ontology trong lĩnh vực chứng khoán

- Trước hết Ontology sẽ được sử dụng để tổng hợp thông tin về giao dịch chứng khoán. Kết quả giao dịch (giá cả, khối lượng, tăng giảm) sẽ lưu trong Ontology và sau đó truy vấn để lấy ra các báo cáo mong muốn như : tổng hợp tình hình giao dịch trong ngày, xếp hạng các cổ phiếu, so sánh nhiều cổ phiếu, phân tích kĩ thuật.
- Phần báo cáo tài chính của các công ty cũng sẽ được lưu trong Ontology để kết hợp với Ontology về giao dịch tạo thành nguồn dữ liệu đầy đủ cho việc phân tích cổ phiếu.
- Ontology cũng sẽ được ứng dụng trong việc xây dựng nguồn từ vựng báo cáo, tạo khả năng xây dựng báo cáo tự động.
- Với các tin tức chứng khoán thì Ontology sẽ là công cụ đắc lực để tự động thu thập, trích chọn thông tin liên quan và lưu trữ để truy vấn tạo ra các phân tích đầy đủ cho nhà đầu tư.

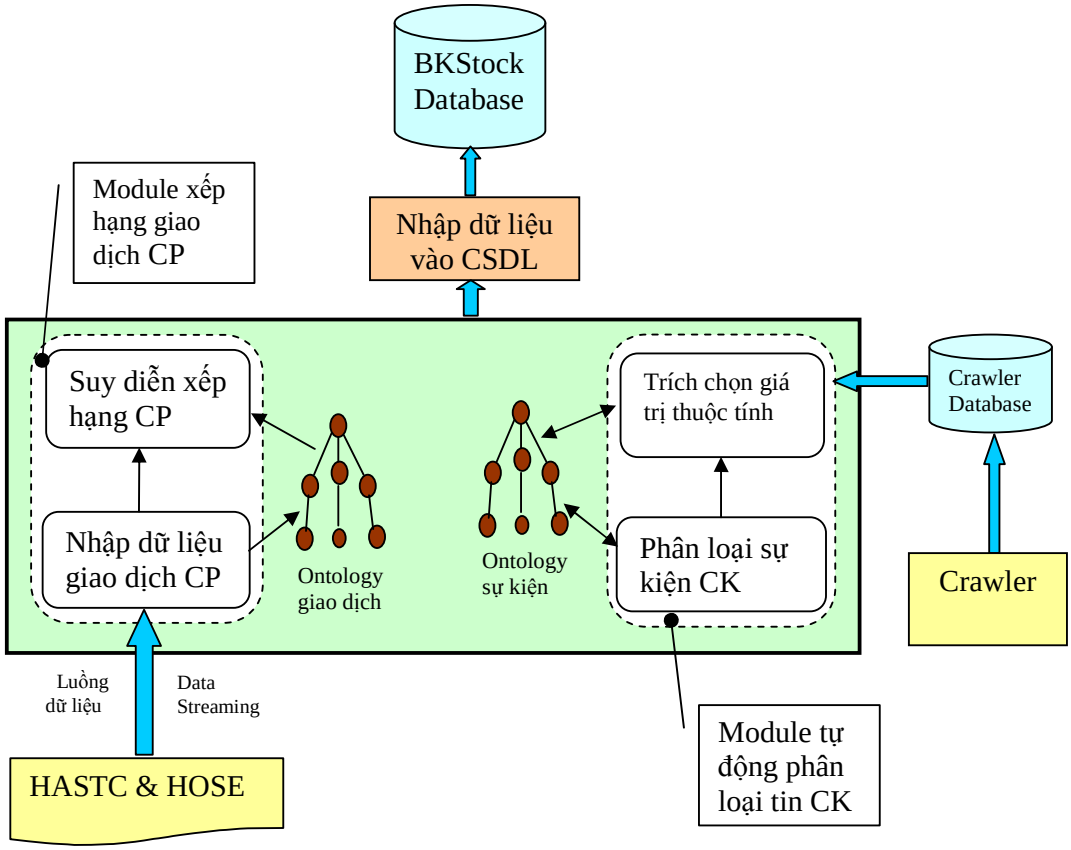
4.2. Kiến trúc của hệ thống BKS

BKS là một hệ thống phân phối thông tin chứng khoán tự động trên nền tảng Ontology. Mục đích của hệ thống này là thu thập tự động các thông tin chứng khoán bao gồm: thông tin giao dịch cổ phiếu trên hai (2) sàn giao dịch HASTC và HOSE, thông tin báo cáo tài chính của các công ty niêm yết, tin tức chứng khoán lấy từ hai sàn giao dịch và từ các báo điện tử. Các thông tin thu thập sẽ được chuyển vào các Ontology: Ontology giao dịch, Ontology báo cáo tài chính, Ontology sự kiện để xử lý. Kết quả xử lý sẽ được lưu vào cơ sở dữ liệu BKStock. Ngoài ra, từ các Ontology đã xây dựng và một Ontology Mẫu báo cáo tư vấn thì hệ thống sẽ tự động sinh ra các báo cáo tư vấn chứng khoán cho người sử dụng, các mẫu báo cáo này có thể tùy biến.

Hệ thống BKS được xây dựng gồm hai phần. Phần Front-End là một Portal người dùng, chịu trách nhiệm cung cấp nội dung cho người sử dụng. Phần Back-End chịu trách nhiệm xử lý Ontology và cung cấp dữ liệu cho phần Front-End. Portal người dùng được viết trên nền DotNetNuke Framework, sử dụng ngôn ngữ lập trình C#.Net. Phần Back-End sử dụng JENA framework và ngôn ngữ lập trình Java. Hai phần này của hệ thống trước mắt giao tiếp thông qua cơ sở dữ liệu BKStock và file XML báo cáo tư vấn tự động. Mô hình tổng quan của hệ thống được thể hiện trên Hình 7.

4.3. Kiến trúc module xếp hạng cổ phiếu và phân loại sự kiện CK

Kiến trúc tổng quát của module xếp hạng cổ phiếu và tự động phân loại sự kiện được thể hiện trong Hình 8 dưới đây. Trong hình này thì hai module được đặt trong một hình chữ nhật đậm để phân biệt với các thành phần khác trong hệ thống tổng thể.



Hình 8: Kiến trúc tổng quan của hai module xếp hạng CP và phân loại tin CK

Module xếp hạng giao dịch cổ phiếu bao gồm hai thành phần chính là: thành phần nhập dữ liệu giao dịch cổ phiếu và thành phần thực hiện suy diễn xếp hạng cổ phiếu trên toàn thị trường cũng như theo ngành. Dữ liệu đầu vào của module này là luồng dữ liệu (Data Streaming) giao dịch được cung cấp từ hai trung tâm giao dịch chứng khoán HASTC và HOSE. Đây sẽ là luồng dữ liệu cập nhật trực tuyến kết quả giao dịch trên sàn. Module con Nhập dữ liệu giao dịch sẽ tự động nhập dữ liệu giao dịch cổ phiếu vào Ontology giao dịch cổ phiếu. Sau đó module con Suy diễn xếp hạng cổ phiếu sẽ sử dụng Ontology giao dịch cổ phiếu để suy diễn ra kết quả xếp hạng. Dữ liệu giao dịch và kết quả xếp hạng sẽ được nhập vào cơ sở dữ liệu BKStock thông qua một module nhập dữ liệu.

Module Tự động phân loại tin (sự kiện) chứng khoán cũng bao gồm hai module con là: Phân loại sự kiện chứng khoán và Trích chọn giá trị thuộc tính. Dữ liệu đầu vào của module này là các bản tin đã được module Crawler thu thập từ các nguồn trên mạng và nhập vào một Crawler Database. Quá trình phân loại tin sẽ gồm hai bước là phân loại tin theo các nhóm và trích giá trị thuộc tính đối với tin đã phân loại. Hỗ trợ cho quá trình này là Ontology sự kiện (gồm Ontology sự kiện –

eventStock.owl và Ontology cơ sở tri thức sự kiện FullEventKB.owl). Kết quả phân loại tin chứng khoán cũng sẽ được nhập vào cơ sở dữ liệu BKStock.

Chương 5. Thiết kế và cài đặt module xếp hạng cổ phiếu

5.1. Thiết kế và xây dựng Ontology giao dịch cổ phiếu

Các Ontology trong đồ án được xây dựng dựa trên phương pháp MethOntology tuy nhiên chỉ tập trung vào các bước chính để tạo ra Ontology.

1. Các hoạt động đặc tả Ontology:

Bao gồm xác định mục đích Ontology, phạm vi của Ontology và người sử dụng.

a. Mục đích Ontology giao dịch cổ phiếu: Mục đích chính của Ontology này là để lưu trữ được các thông tin về thị trường chứng khoán Việt Nam. Từ đó cho phép suy diễn để cung cấp nhiều loại thông tin tổng hợp như: kết quả giao dịch, xếp hạng cổ phiếu, báo cáo tư vấn tự động.

b. Phạm vi của Ontology:

- Ontology giao dịch cổ phiếu không chỉ giới hạn trong các thông tin về kết quả giao dịch trên các sàn mà bao gồm những thông tin đầy đủ về một thị trường chứng khoán. Nó sẽ phải chứa được các thông tin như: thị trường, ngành, công ty niêm yết, cổ phiếu, chỉ số thị trường, giao dịch cổ phiếu (cả nhà đầu tư trong nước và nước ngoài), các thông tin thêm về phát hành thêm, tách/gộp cổ phiếu.
- Tuy nhiên Ontology này không bao gồm các thông tin về báo cáo tài chính của công ty niêm yết cổ phiếu, và chỉ giới hạn ở hai sàn giao dịch của thị trường chứng khoán Việt Nam. Tuy nhiên phải hoàn toàn có khả năng mở rộng được.

c. Phạm vi người sử dụng:

Người sử dụng Ontology sẽ là các nhà đầu tư chứng khoán, mong muốn tìm kiếm một hệ thống tổng hợp thông tin đầy đủ và nhanh chóng.

2. Xác định nguồn dữ liệu:

Nguồn dữ liệu cho Ontology giao dịch cổ phiếu là nguồn có được các thông tin về: thị trường giao dịch, ngành, công ty, cổ phiếu, kết quả giao dịch,... Các nguồn đã được xem xét trong quá trình xây dựng Ontology là các trang web:

- ✓ HASTC: http://www.hastc.org.vn/
- ✓ HOSE: http://www.vse.org.vn/
- ✓ FPTS: http://www.fpts.com.vn
- ✓ VNDS: http://www.vnds.com.vn

Ngoài ra có tham khảo một số trang web tài chính, chứng khoán nước ngoài như:

- ✓ Bloomberg: http://www.bloomberg.com/
- ✓ Google Finance: http://finance.google.com/finance
- ✓ Yahoo Finance: http://finance.yahoo.com

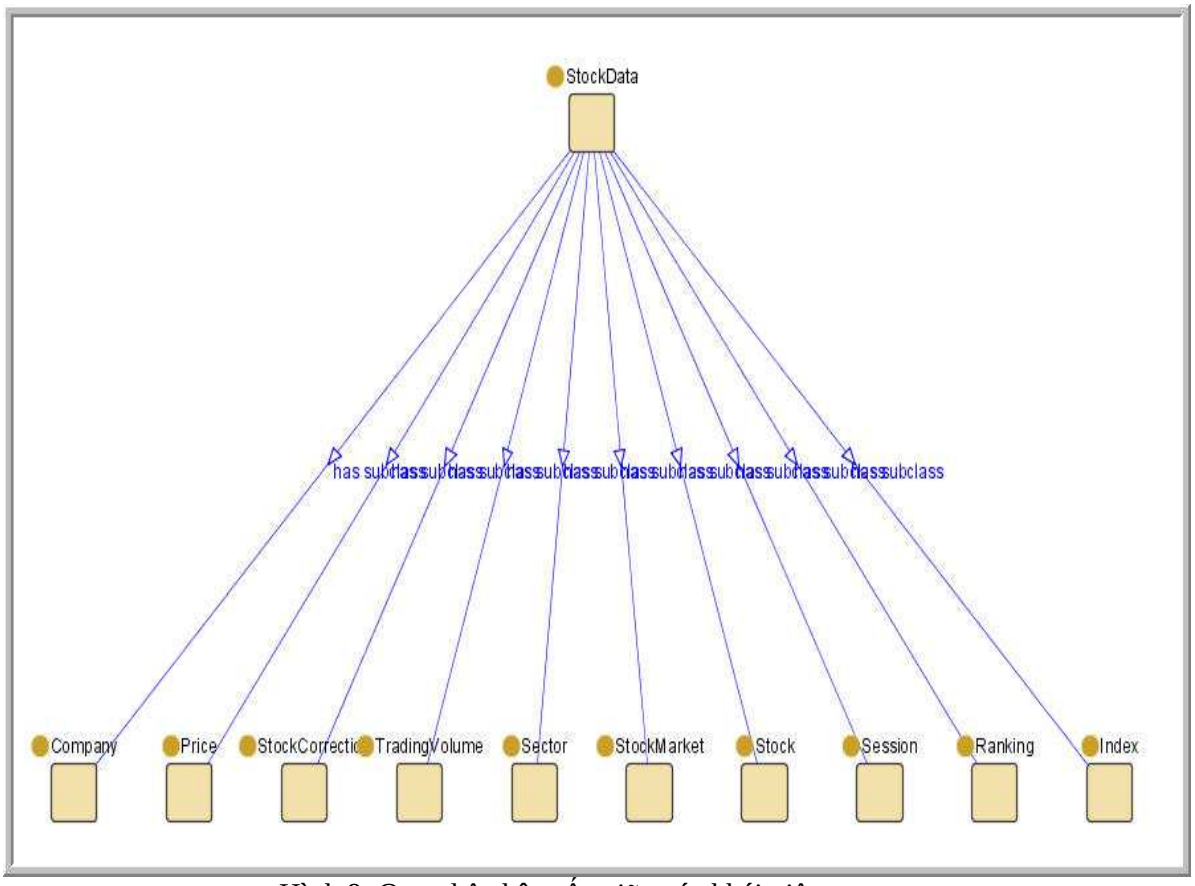
3. Xác định các thuật ngữ cho Ontology:

Căn cứ vào các nguồn thông tin trên, một danh sách các thuật ngữ cùng với giải thích của chúng đã được đưa ra:

Thuật ngữ: Stock, Share : biểu diễn về thông tin cổ phiếu

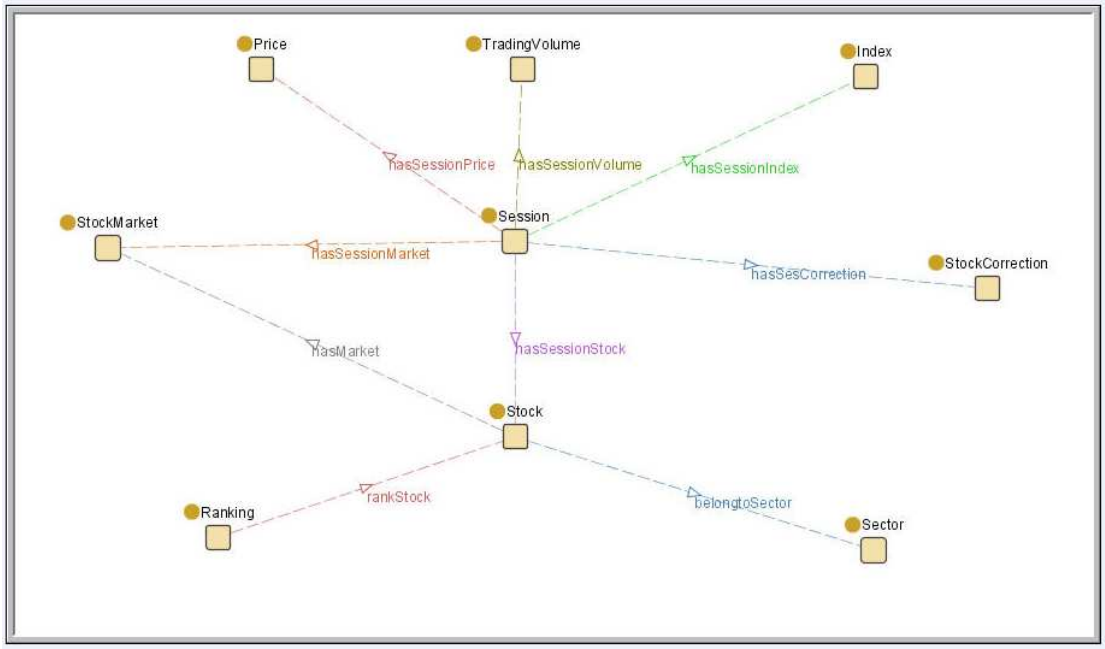
4. Khái niệm hóa Ontology:

Cấu trúc phân cấp các khái niệm trong Ontology được thể hiện ở Hình 9 bên dưới đây. Khái niệm Stock_Data là khái niệm tổng quát nhất, trở thành cha của các khái niệm khác. Các khái niệm bên dưới đều là con của Stock_Data và không có quan hệ phân cấp với nhau. Lý do là để giảm thiểu sự phức tạp của Ontology và trong thực tế chúng cũng không có quan hệ cha-con với nhau.



Hình 9: Quan hệ phân cấp giữa các khái niệm

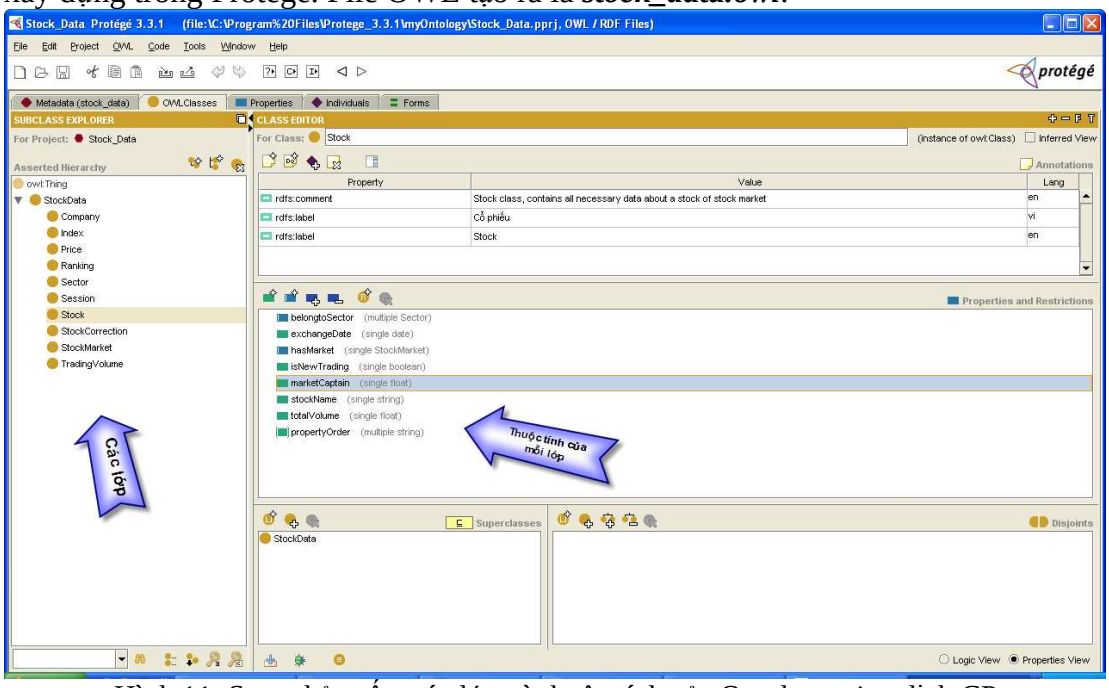
Quan hệ giữa các khái niệm cũng được thể hiện trong Hình bên dưới. Những mối quan hệ này sẽ cho phép thực hiện suy diễn trên Ontology.



Hình 10: Quan hệ giữa các khái niệm trong Ontology giao dịch cổ phiếu

5. Hình thức hóa Ontology:

Việc hình thức hóa Ontology dưới dạng một ngôn ngữ Ontology được thực hiện nhờ Protege. Như đã nói ở chương 1 thì ngôn ngữ Ontology được sử dụng là OWL-DL vì các ưu điểm của nó về sự phong phú trong mô tả Ontology và khả năng suy diễn. Hình dưới đây thể hiện các lớp cùng thuộc tính của chúng đã được xây dựng trong Protege. File OWL tạo ra là ***stock_data.owl***.



Hình 11: Soạn thảo cấu trúc lớp và thuộc tính của Ontology giao dịch CP

6. Tạo các thể hiện Ontology giao dịch cổ phiếu:

Các thể hiện của Ontology giao dịch cổ phiếu gồm thông tin tĩnh, ít thay đổi như: tên thị trường, tên cổ phiếu, tên công ty, . . . Chúng sẽ được tạo ra nhờ một module nhập liệu (vì số lượng cổ phiếu lên đến hơn 200 nên sử dụng chương trình sẽ nhanh và chính xác hơn). Còn các thể hiện là kết quả giao dịch cổ phiếu sẽ có một thành phần riêng và thực hiện tự động khi chạy hệ thống.

7. Đánh giá Ontology giao dịch cổ phiếu:

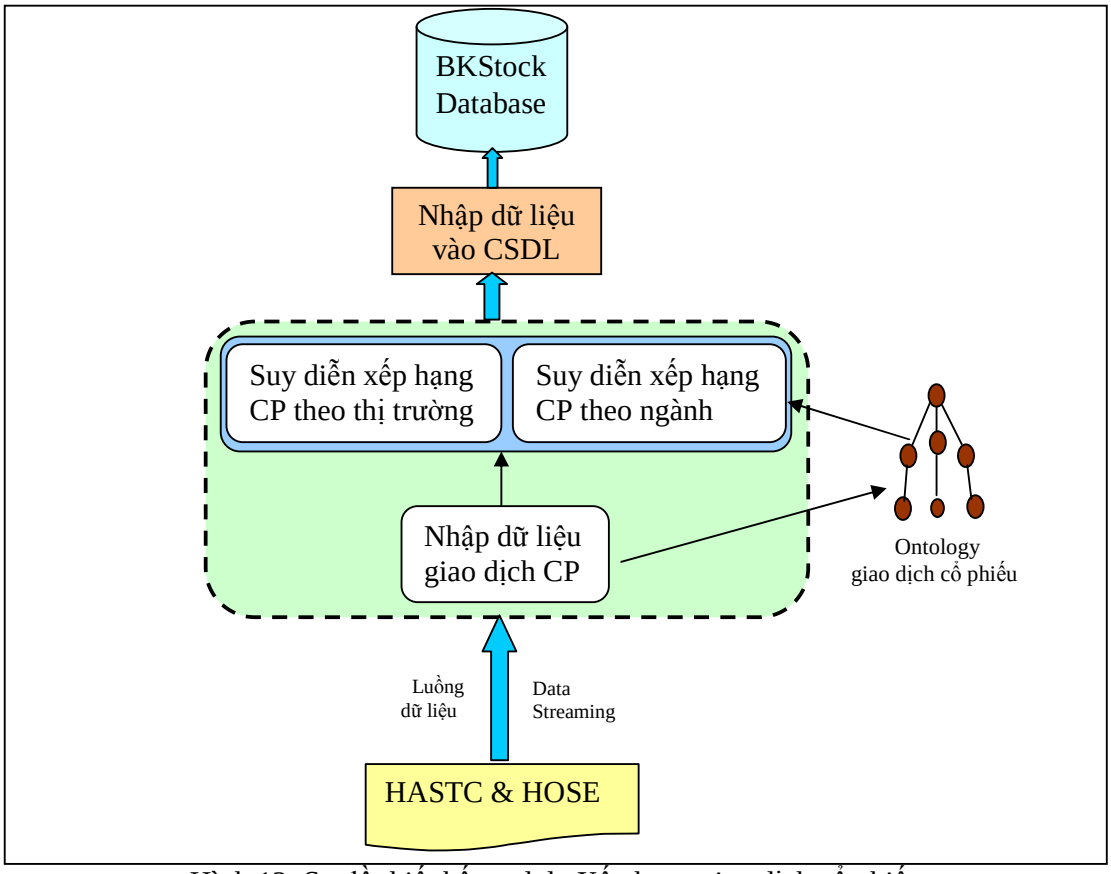
Ontology giao dịch cổ phiếu đã thỏa mãn đặc tả Ontology đề ra ban đầu. Nó đã được cài đặt và thực hiện chức năng suy diễn thành công trong hệ thống BKS. Tuy nhiên, do vấn đề tốc độ thực hiện xếp hạng cổ phiếu nên chỉ những dữ liệu giao dịch cổ phiếu mới nhất mới được đưa vào Ontology.

Thiết kế chi tiết của Ontology giao dịch cổ phiếu sẽ được thể hiện trong mục 1 của phụ lục.

5.2. Thiết kế chương trình

5.2.1 Sơ đồ thiết kế Module

Sơ đồ thiết kế của module Xếp hạng giao dịch cổ phiếu được thể hiện trong Hình 12 bên dưới. Đầu vào là luồng dữ liệu từ hai trung tâm giao dịch chứng khoán HASTC và HOSE. Đầu ra là dữ liệu về kết quả giao dịch cổ phiếu và xếp hạng cổ phiếu theo thị trường hoặc theo ngành. Các dữ liệu này sẽ được nhập vào cơ sở dữ liệu BKStock.



Hình 12: Sơ đồ thiết kế module Xếp hạng giao dịch cổ phiếu

Module Xếp hạng giao dịch cổ phiếu gồm các thành phần: Nhập dữ liệu giao dịch cổ phiếu, Suy diễn xếp hạng cổ phiếu theo thị trường, Suy diễn xếp hạng cổ phiếu theo ngành. Ontology sử dụng là Ontology giao dịch cổ phiếu.

Module con Nhập dữ liệu giao dịch cổ phiếu sẽ có nhiệm vụ nhận luồng dữ liệu, phân tích thành các thông tin giao dịch và nhập vào Ontology. Các thông tin giao dịch bao gồm: chỉ số thị trường (Index), giá cổ phiếu (Price), khối lượng giao dịch (TradingVolume) – gồm cả giao dịch nhà đầu tư trong nước và giao dịch nhà đầu tư nước ngoài. Các thông tin giao dịch sau khi đã được phân tích sẽ được phân chia về các lớp tương ứng của Ontology. Các thể hiện của mỗi lớp sẽ được tạo ra và nhập vào Ontology.

Việc thiết kế hai module con phục vụ suy diễn xếp hạng sẽ được nói chi tiết trong các phần bên dưới.

5.2.2 Công thức tính điểm xếp hạng

Để có thể xếp hạng được các cổ phiếu thì ở đây chương trình đã xác định một công thức cho phép tính ra điểm số cho mỗi cổ phiếu. Việc phân loại trước hết sẽ dựa theo kết quả giao dịch cổ phiếu nên các số liệu cần quan tâm là : giá cả cổ phiếu trong ngày và khối lượng giao dịch. Những số liệu này sẽ tạo ra kì vọng của nhà đầu tư về cổ phiếu. Số liệu đưa vào công thức : phần trăm thay đổi giá, giá trị giao dịch, khối lượng giao dịch.

Công thức tính điểm số:

```
Point = priceChgPe*50 + volValue/1E8 + %vol*100
```

trong đó:

priceChgPe là phần trăm thay đổi giá, được tính theo :

(giá đóng cửa – giá tham chiếu) / giá tham chiếu, lấy %.

volValue là giá trị giao dịch, đơn vị là tỉ đồng.

%vol là tỉ lệ khối lượng giao dịch trong ngày so với khối lượng niêm yết của cổ phiếu đó.

Các hệ số đưa vào cho mỗi thành phần của công thức là để đảm bảo giá trị mỗi thành phần tương đương với nhau (giá trị mỗi thành phần trong khoảng cỡ hàng trăm)

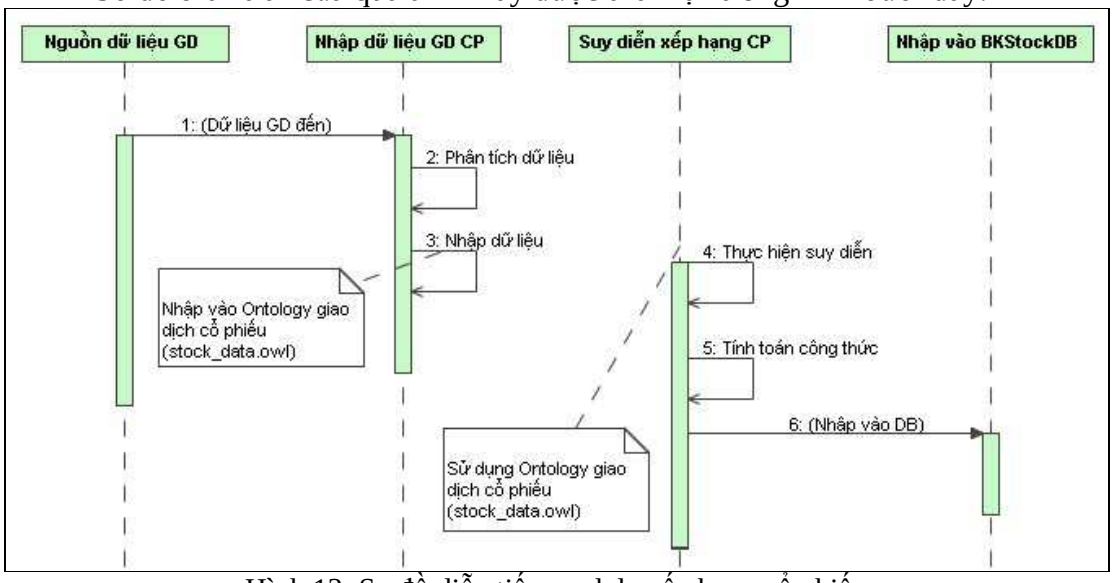
Nhận xét: Với công thức này thì chúng ta có thể xác định trong ngày giao dịch, cổ phiếu được nhà đầu tư kì vọng đến mức nào. Điểm số cao thể hiện việc tăng giá mạnh, lượng giao dịch lớn ; ngược lại điểm số thấp là cổ phiếu giảm giá, lượng giao dịch ít. Khi một cổ phiếu trong hàng *top* liên tục thì chứng tỏ nó đang được nhiều nhà đầu tư quan tâm nên giá liên tục tăng, giao dịch cũng lớn. Áp dụng công thức này thì các cổ phiếu trung bình cũng lọt vào *top* nếu sức mua nó lớn, tuy nhiên các cổ phiếu *bluechip* cũng không tụt xa do có *volValue* và *%vol* kéo lại.

Công thức đang được dùng chung cho xếp hạng cổ phiếu toàn thị trường và theo ngành.

5.2.3 Thiết kế quá trình suy diễn xếp hạng

Quá trình suy diễn xếp hạng trước hết phải xác định được tên thị trường và ngày xếp hạng. Căn cứ vào công thức tính điểm số trên thì những giá trị cần phải lấy ra cho mỗi cổ phiếu là: *priceChgPe* (phần trăm thay đổi giá), *volValue* (giá trị giao dịch), *volQty* (khối lượng giao dịch) và *totalVolume* (tổng lượng cổ phiếu phát hành). Sau khi thực hiện tính toán ra giá trị điểm xếp hạng cho mỗi cổ phiếu thì sắp xếp chúng theo thứ tự giảm dần và nhập vào Ontology giao dịch cổ phiếu theo các thể hiện của lớp *Ranking*. Kết quả cuối cùng sẽ nhập vào cơ sở dữ liệu BKStock.

Sơ đồ diễn tiến của quá trình này được thể hiện trong hình dưới đây.



Hình 13: Sơ đồ diễn tiến module xếp hạng cổ phiếu

5.3. Cài đặt chương trình và kết quả

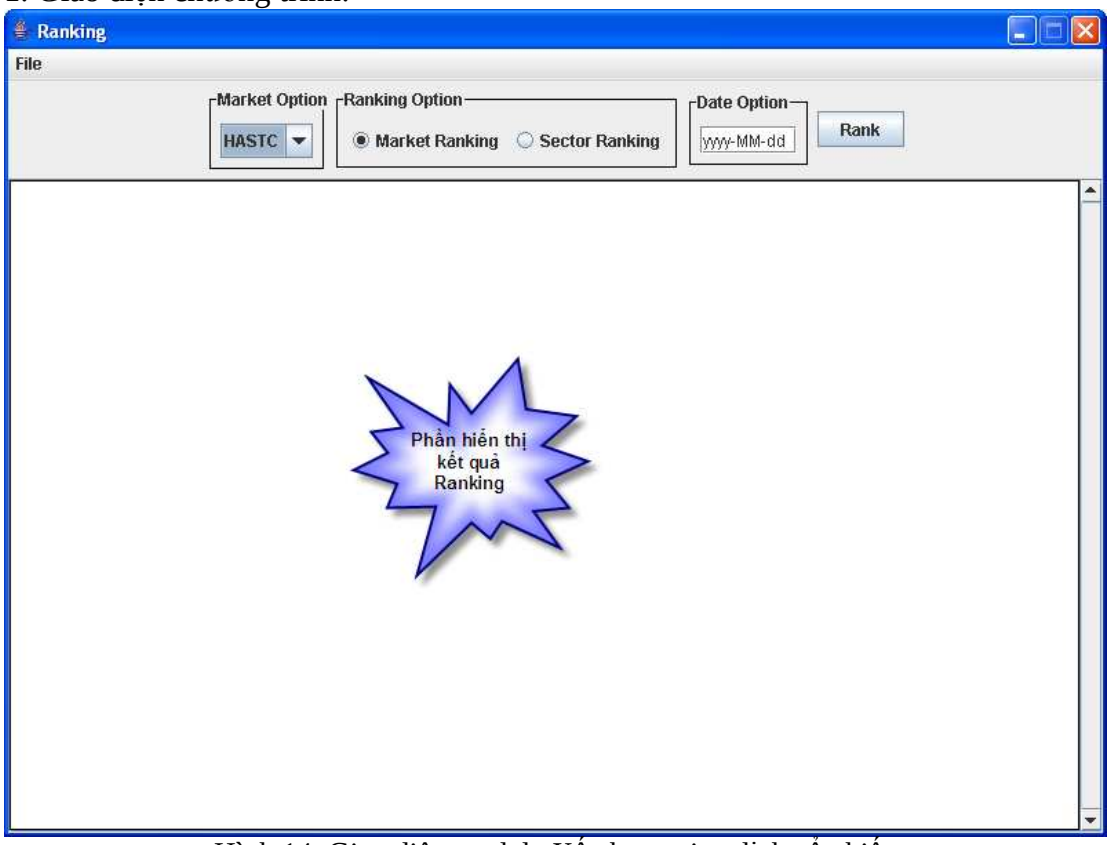
Module này thực chất sẽ là một module chạy nền, cung cấp dịch vụ cho hệ thống bên trên. Tuy nhiên để phục vụ việc quan sát cũng như nắm rõ hoạt động của

chương trình thì trong đồ án này đã tiến hành cài đặt giao diện GUI. Việc cài đặt này còn khá đơn giản nhưng vẫn đảm bảo tính dễ dùng, thân thiện của giao diện.

Hiện nay luồng dữ liệu từ hai trung tâm giao dịch là chưa lấy được nên việc nhập dữ liệu vẫn chỉ thực hiện trên file text có cấu trúc, theo dữ liệu của trang FPTS. Do đó khi có luồng dữ liệu thật sự thì module con Nhập dữ liệu giao dịch cổ phiếu sẽ phải thay đổi trong phần xác định thông tin giao dịch cổ phiếu.

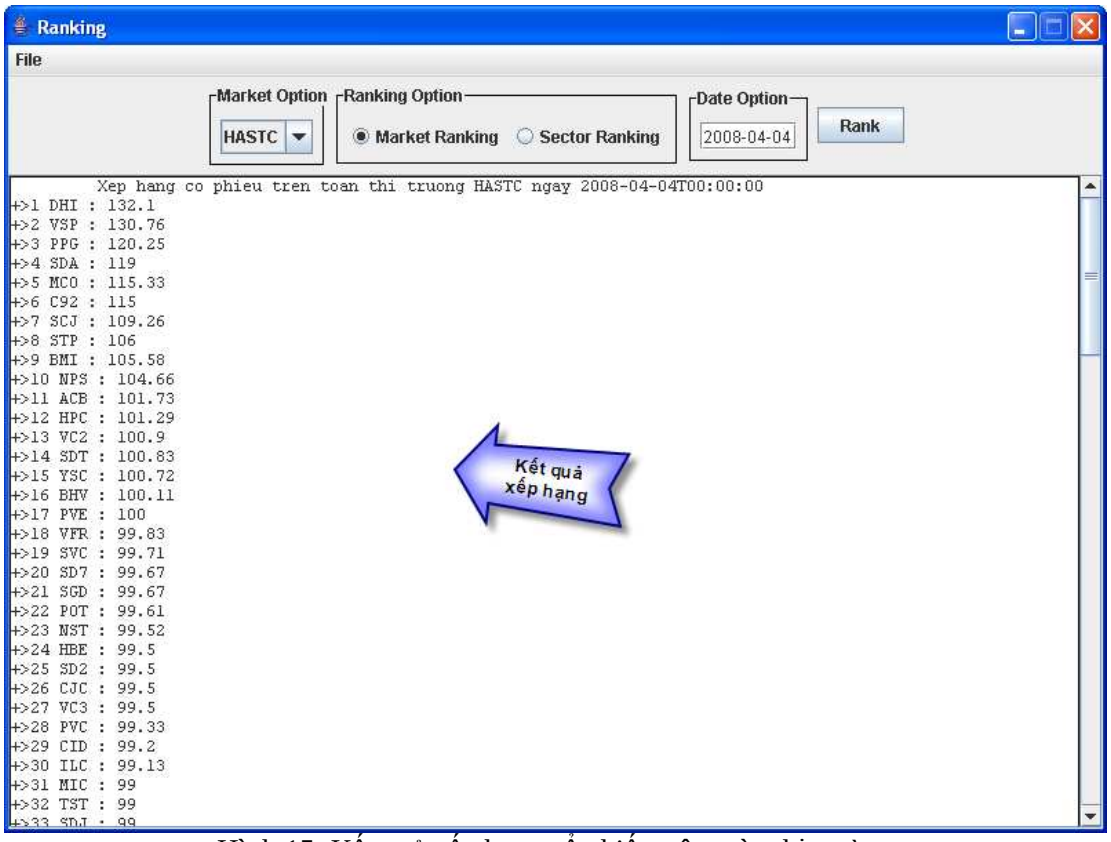
Kết quả cài đặt:

1. Giao diện chương trình:

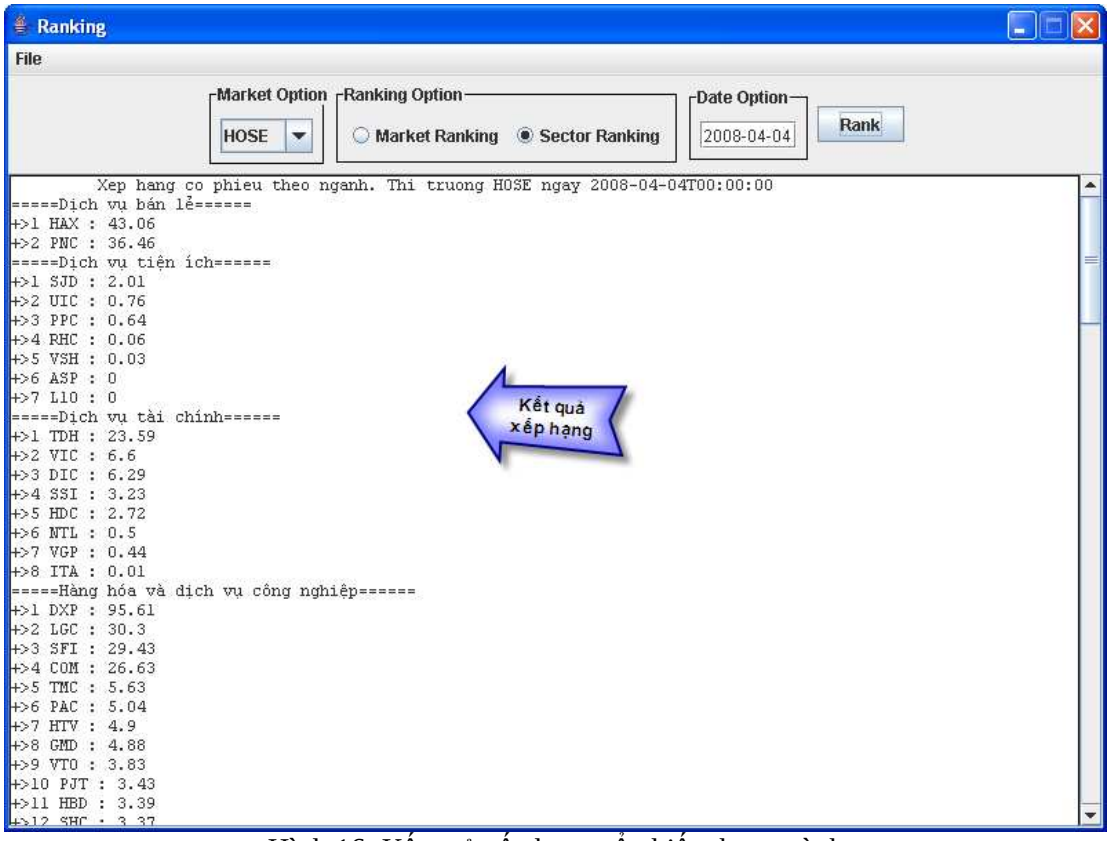


Hình 14: Giao diện module Xếp hạng giao dịch cổ phiếu

2. Kết quả xếp hạng cổ phiếu theo ngày giao dịch:



Hình 15: Kết quả xếp hạng cổ phiếu trên toàn thị trường



Hình 16: Kết quả xếp hạng cổ phiếu theo ngành

5.4. Nhận xét và hướng phát triển, mở rộng

Kết quả xếp hạng cổ phiếu trong ngày giao dịch 4/4/2008 được cho trong bảng như sau:

Thị trường HASTC:

Xếp hạng	*Tên cổ phiếu*	*Tên công ty*	*Điểm số*
1	DHI	Công ty In Diên Hồng	132.1
2	VSP	Công ty Vinashin	130.76
...			
11	ACB	Ngân hàng Á Châu	101.73
35	S99	Công ty Sông Đà 909	98.67

Thị trường HOSE:

Xếp hạng	*Tên cổ phiếu*	*Tên công ty*	*Điểm số*
1	DXP	Công ty Cảng Đoạn Xá	95.61
2	HAX	Công ty Ô tô Hàng Xanh	43.06
...			
21	VIC	Công ty VINCOM	6.6
44	FPT	Công ty FPT	2.6

Nhận xét:

- Điểm số xếp hạng trên hai thị trường HASTC và HOSE khác nhau khá nhiều do biên độ giao dịch giá của hai thị trường là khác nhau, của HASTC lớn hơn nên ảnh hưởng của sự thay đổi giá cao hơn.
- Thứ tự xếp hạng chưa phản ánh đúng kì vọng về cổ phiếu của nhà đầu tư. Cụ thể theo xếp hạng trên thị trường HASTC thì cổ phiếu DHI rất tốt nhưng trên thực tế thì đây không phải là cổ phiếu tốt theo đánh giá của nhiều người. Cũng như vậy, các cổ phiếu Bluechip chưa có được thứ hạng cao. Lý do là điểm số xếp hạng chỉ phụ thuộc vào giao dịch cổ phiếu mà chưa tính đến kết quả công bố kinh doanh của doanh nghiệp. Hướng phát triển là sử dụng Ontology báo cáo tài chính để đưa thêm ảnh hưởng của doanh số kinh doanh.

Kết quả xếp hạng cổ phiếu trong 3 ngày giao dịch 4/4/2008, 7/4/2008, 8/4/2008 :

Thị trường HASTC:

Ngày	*Xếp hạng*	*Tên CP*	*Tên công ty*	*Điểm số*
4/4/2008	1	DHI	Công ty In Diên Hồng	132.1
	2	VSP	Công ty Vinashin	130.76
	...			
	11	ACB	Ngân hàng Á Châu	101.73
	35	S99	Công ty Sông Đà 909	98.67
7/4/2008	HASTC-Index tăng 2.61%			
	1	NBC	Công ty Than Núi Béo	627.22
	...			
	4	DHI	Công ty In Diên Hồng	298.06
	21	S99	Công ty Sông Đà 909	164.84
	34	ACB	Ngân hàng Á Châu	154.07

8/4/2008	HASTC-Index giảm 0.78%			
	1	TLT	Gạch men Viglacera Thăng Long	1788.31
	...			
	5	S99	Công ty sông đà 909	853.11
	11	DHI	Công ty In Diên Hồng	568.07
	28	NBC	Công ty than Núi Béo	307.46
	109	ACB	Ngân hàng Á Châu	-0.01

Thị trường HOSE:

Ngày	*Xếp hạng*	*Tên CP*	*Tên công ty*	*Điểm số*
4/4/2008	1	DXP	Công ty Cảng Đoạn Xá	95.61
	2	HAX	Công ty Ô tô Hàng Xanh	43.06
	...			
	21	VIC	Công ty VINCOM	6.6
	44	FPT	Công ty FPT	2.6
7/4/2008	VN-Index tăng 1.75%			
	1	TMS	Công ty Kho vận, giao nhận TPHCM	127.11
	...			
	34	DXP	Công ty Cảng Đoạn Xá	8.6
	47	FPT	Công ty FPT	5.35
	105	VIC	Công ty VINCOM	1.34
8/4/2008	VN-Index tăng 1.15%			
	1	UNI	Công ty Viễn Liên	850.07
	...			
	9	HAX	Công ty Ô tô Hàng Xanh	215.16
	14	DXP	Công ty Cảng Đoạn Xá	185.59
	53	FPT	Công ty FPT	71.41
	95	VIC	Công ty VINCOM	24.86
	126	TMS	Công ty Kho vận, giao nhận TPHCM	2.04

Nhận xét:

- Các công ty Bluechip ít được vào top xếp hạng cổ phiếu. Lí do ít quan tâm đến giá trị nội tại của mỗi công ty.
- Mức độ nhảy bậc của nhiều cổ phiếu là cao (ví dụ TMS từ 1->126), chứng tỏ công thức tính điểm xếp hạng chưa ổn định.

Hướng phát triển trong tương lai của Module:

- Module bao gồm hai thành phần Nhập dữ liệu giao dịch cổ phiếu và suy diễn xếp hạng với độ liên kết là thấp. Hai thành phần chỉ liên kết thông qua Ontology giao dịch cổ phiếu. Do đó module này hoàn toàn có thể được tích hợp các thành phần suy diễn khác trên Ontology để mở rộng chức năng mà nó có thể cung cấp (hiện tại mới chỉ giới hạn ở chức năng xếp hạng cổ phiếu theo thị trường và theo ngành). Trong tương lai có thể phát triển thêm các chức năng thông tin chi tiết hơn, ví dụ

xây dựng hệ thống quản lý danh mục đầu tư gồm: thông tin giao dịch cổ phiếu trong danh mục, cảnh báo giá cổ phiếu,...
- Thành phần Suy diễn xếp hạng cổ phiếu được thực hiện theo thời gian nên có thể phát triển yếu tố thời gian trong Ontology.
- Module này có thể xây dựng thành một Web service cung cấp dịch vụ cho các ứng dụng khác trong một hệ thống thông tin chứng khoán lớn.

Chương 6. Phân loại và trích thông tin sự kiện chứng khoán

6.1. Bài toán phân loại tin sự kiện CK

6.1.1 Sự kiện chứng khoán

Sự kiện chứng khoán là các tin tức liên quan đến: cổ phiếu, công ty niêm yết chứng khoán, ngành nghề, lĩnh vực mà các công ty niêm yết hoạt động, tin tức về thị trường chứng khoán nói chung. Ngoài ra do thị trường chứng khoán rất nhạy cảm nên nó có thể bao gồm các tin chung về tình hình kinh tế, chính sách vĩ mô của nhà nước. Tuy nhiên để đảm bảo tính phù hợp với hoạt động đầu tư chứng khoán là chỉ đầu tư vào một số lượng nhỏ các cổ phiếu thì các sự kiện chứng khoán mà đồ án này xét tới phải là các tin tức liên quan đến một hay một vài cổ phiếu nào đó, không phải là tin tức chung chung hay không liên quan gì. Sự kiện chứng khoán hay tin chứng khoán là hai thuật ngữ sẽ được dùng thay thế nhau trong đồ án này.

Sự kiện chứng khoán sẽ bao gồm các thông tin như: thông tin giao dịch cổ phiếu với số lượng lớn, thông tin phát hành thêm, tách/gộp cổ phiếu; thông tin thay đổi ban giám đốc công ty, thông báo kết quả kinh doanh của công ty niêm yết; tin tức về tình hình các ngành có công ty niêm yết; tin tức thiên tai thảm họa liên quan đến công ty hay nhóm ngành nào đó; các tin chung, tin chính sách nhà nước nhưng có liên quan đến công ty niêm yết. Sự kiện chứng khoán sẽ không bao gồm các thông tin như: tin về công ty mới niêm yết, về các công ty chưa niêm yết, các tin chung ảnh hưởng đến toàn thị trường, tin chứng khoán, tài chính quốc tế (trừ các tin ảnh hưởng trực tiếp đến chứng khoán niêm yết tại Việt Nam ví dụ công ty nước ngoài đầu tư vào công ty niêm yết).

Nguồn sự kiện chứng khoán sẽ từ hai trung tâm giao dịch chứng khoán HASTC (http://www.hastc.org.vn/), HOSE (http://www.vse.org.vn/), từ các báo điện tử (ví dụ báo Economy.vn: http://economy.vn/).

6.1.2 Lợi ích của việc phân loại sự kiện chứng khoán

Theo trên đã nói thì các sự kiện chứng khoán gồm nhiều thông tin phong phú và được thu thập từ nhiều nguồn khác nhau. Nếu không phân loại thì chúng sẽ trở thành một "đống hỗn độn" các thông tin, mức độ có ích của thông tin sẽ rất thấp.

Lợi ích rõ thấy nhất của việc phân loại tin chứng khoán là biết được tin tức này sẽ ảnh hưởng đến cổ phiếu hoặc những cổ phiếu nào. Căn cứ vào ảnh hưởng tốt hay xấu mà nhà đầu tư có thể dự đoán được xu hướng giá lên xuống của cổ phiếu và có quyết định mua,bán kịp thời. Đây là vấn đề rất quan trọng, được các nhà đầu tư chứng khoán quan tâm hàng đầu hiện nay.

6.1.3 Nhiệm vụ cần đạt được cho bài toán phân loại sự kiện chứng khoán

Việc phân loại tin chứng khoán đã được nhiều trang web chứng khoán thực hiện nhưng hầu hết trong số đó được thực hiện bằng tay, với sự tham gia rất lớn của con người. Một trong những mục đích chính cần đạt được của bài toán phân loại sự kiện chứng khoán đặt ra trong đồ án này là thực hiện tự động việc phân loại. Đầu vào của quá trình phân loại sẽ là từ một số khá lớn các trang báo điện tử, đảm bảo khả năng phân loại lượng thông tin lớn trên Internet.

Phân loại tin chứng khoán không đơn thuần là chia nó thành các loại là xong. Các loại đó thường có độ tổng quát cao và vẫn cần công sức con người để phát hiện ra ngữ nghĩa trong đó. Một mục đích tiếp theo của bài toán phân loại sự kiện chứng khoán là phải có khả năng tự động trích chọn ra các thông tin ngữ nghĩa từ trong tin

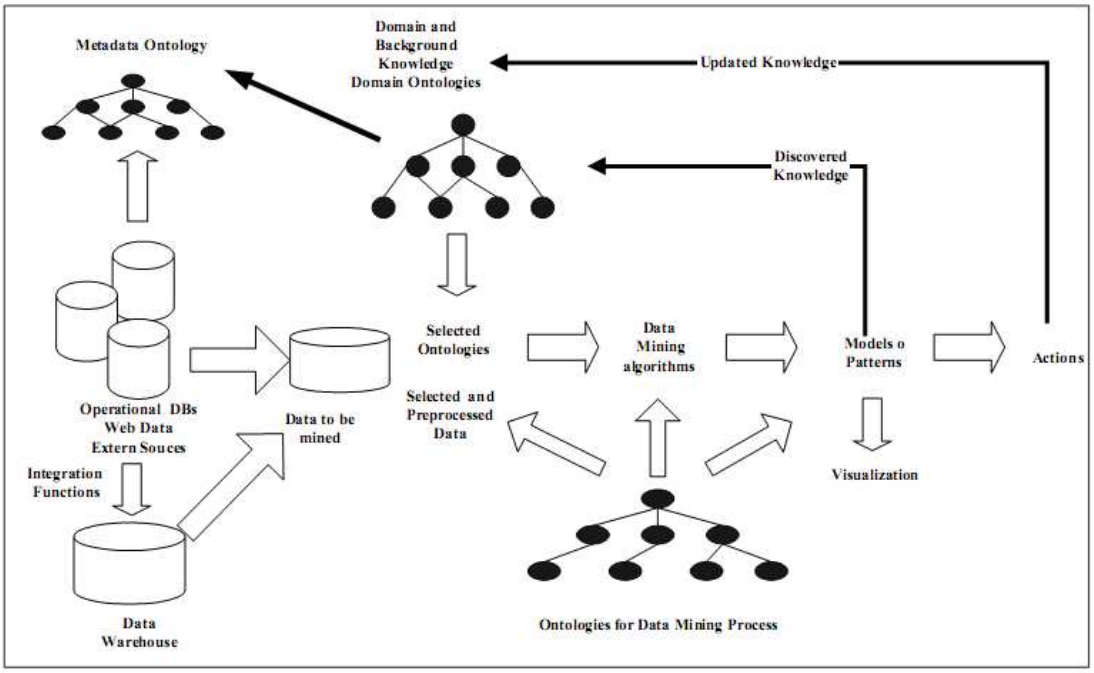
tức đã phân loại. Một ứng dụng quan trọng của các thông tin được trích chọn ra là để đưa ra cảnh báo cho nhà đầu tư nếu sự kiện này ảnh hưởng đến cổ phiếu quá một ngưỡng nào đó (ví dụ cổ phiếu được mua/bán với số lượng lớn). Việc đưa ra điều kiện cảnh báo có thể tùy biến được.

6.2. Ứng dụng Ontology trong phân loại thông tin

6.2.1 Một số nghiên cứu về Data mining với Ontology

[1] Data mining (cũng được biết đến như là Khai phá tri thức trong cơ sở dữ liệu – Knowledge Discovery in Databases, KDD) được định nghĩa là " việc trích ra các tri thức ẩn giấu, trước đây chưa được biết và các thông tin có thể có ích từ dữ liệu ". Nó có thể sử dụng công nghệ học máy (machine learning), thống kê, hình ảnh hóa để khám phá và thể hiện tri thức theo dạng mà dễ dàng tổng hợp với con người. Với Data mining hiện nay, đang có sự tăng lên đáng kể những quan tâm về nghiên cứu và khám phá các miền thông tin không cấu trúc hoặc bán cấu trúc như: văn bản (Text mining), mạng (Web mining), dữ liệu quan hệ (Relation Data mining), luồng dữ liệu (Stream mining). Với những ứng dụng đó, như đã nói trong phần 1.3 thì vai trò của Ontology ngày càng quan trọng.

Theo [3] thì quá trình Data mining với Ontology có thể như sau:



Hình 14: Framework chung cho Data mining với Ontologies

- ✓ Metadata Ontologies: những ontology này chứa metadata của quá trình khai phá dữ liệu. Đây là những thông tin chung về tri thức đã được khai phá.
- ✓ Domain Ontologies: các Ontology về miền ứng dụng đang được khai phá.
- ✓ Ontologies for Data mining Process: các Ontology về quá trình khai phá dữ liệu, chứa tri thức tiến trình, ví dụ: lựa chọn đặc tính gì, lựa chọn thuật toán tốt nhất, thiết lập quá trình thẩm định.

6.2.2 Ứng dụng Ontology trong phân loại tài liệu

[1,11] Việc phân loại tài liệu thường được dựa trên mô hình không gian vector (Vector Space Model – VSM). Trong mô hình này, mỗi tài liệu được biểu diễn bởi một vector các thuật ngữ (bởi vậy VSM còn được gọi là mô hình vector thuật ngữ - term vector model). Mỗi chiều của vector tương ứng với một thuật ngữ tách biệt. Mỗi thuật ngữ sẽ được gán một trọng số tỉ lệ với tần suất xuất hiện của nó trong tài liệu. Có nhiều cách đã được phát triển để tính giá trị trọng số thuật ngữ; một cách phổ biến là trọng số TF-IDF (Term Frequency – Inverse Document Frequency) trong đó có tính đến cả tần suất của thuật ngữ trong tài liệu (TF) và số tài liệu chứa thuật ngữ này trong tổng số tài liệu xem xét (IDF). Công thức toán học:

$W(t_i, d) = TF(t_i, d) * IDF(t_i)$ trong đó $IDF(t_i) = \log(\frac{D}{DF(t_i)})$

với $W(t_i, d)$ là trọng số của thuật ngữ t_i trong tài liệu d

$TF(t_i,d)$ là tần suất của thuật ngữ t_i trong tài liệu d, được tính bằng số lần xuất hiện t_i trong d

D là tổng số tài liệu đang xét và $DF(t_i)$ là số tài liệu mà thuật ngữ t_i xuất hiện ít nhất một lần.

Việc chọn thuật ngữ trong tài liệu phụ thuộc vào ứng dụng. Thông thường thuật ngữ là các từ đơn, từ khóa hay cụm từ dài hơn.

Ứng dụng Ontology

Quá trình phân loại tài liệu thông thường dựa trên việc xác định thuật ngữ và tính trọng số của chúng trong tài liệu. Việc này sẽ tạo ra một vector thuật ngữ với số chiều rất lớn, gây khó khăn đáng kể cho quá trình phân loại. Với việc ứng dụng Ontology chúng ta sẽ tạo ra một cấu trúc phân cấp các khái niệm và chuyển các thuật ngữ vào khái niệm, làm giảm đáng kể kích thước của vector thuật ngữ. Trong Ontology, mỗi khái niệm sẽ tương ứng với một loại mà tài liệu sẽ được phân vào.

Khi sử dụng phương pháp Ontology thì chúng ta có hai loại vector thuật ngữ. Một vector thuật ngữ cho mỗi khái niệm gọi là vector khái niệm (*concept vector*). Các concept vector có các thuật ngữ được xác định trọng số từ trước dựa trên một cách nào đó (có thể là dùng tri thức chuyên gia hay sử dụng một phương pháp tự động). Với mỗi tài liệu thì tương ứng với một khái niệm nó sẽ có một vector các thuật ngữ gọi là vector đặc tính (*feature vector*). Vector đặc tính có số thuật ngữ giống hệt như trong vector khái niệm nhưng trọng số của chúng sẽ được xác định trực tiếp qua tài liệu. Trọng số này thường tỉ lệ với tần suất xuất hiện của thuật ngữ trong tài liệu. Cuối cùng, để đánh giá xem tài liệu có thuộc về khái niệm (hay loại) đã cho hay không thì đánh giá tích vô hướng giữa vector khái niệm và vector đặc tính.

$$sim(d, C) = \overrightarrow{CV} * \overrightarrow{FV} = \sum_i (Wc_i * Wf_i)$$

trong đó: $sim(d,C)$: là độ tương tự của một tài liệu d với khái niệm C

$\overrightarrow{CV}$: là vector khái niệm, có các trọng số Wc_i

$\overrightarrow{FV}$: là vector đặc tính, có các trọng số Wf_i

Độ tương tự này thường được so sánh với một ngưỡng để đảm bảo:

- Không phân loại tài liệu không thuộc loại nào đang xét nhưng vẫn có $sim(d,C) \neq 0$
- Có thể phân một tài liệu vào nhiều hơn một loại.

6.3. Đặc điểm của tin tức, sự kiện CK

6.3.1 Đặc điểm của tin từ hai trung tâm giao dịch chứng khoán

Trong các tin của hai trung tâm giao dịch chứng khoán thì những tin được coi là sự kiện chứng khoán là tin từ các tổ chức niêm yết, tin từ trung tâm giao dịch chứng khoán. Nó bao gồm các dạng tin như sau:

- ✓ Tin giao dịch cổ phiếu với số lượng lớn, giao dịch cổ đông chính, cổ đông lớn.
- ✓ Tin phát hành thêm cổ phiếu, tách/gộp cổ phiếu.
- ✓ Tin thay đổi nhân sự lãnh đạo cấp cao.
- ✓ Tin báo cáo tài chính, lợi nhuận.
- ✓ Tin đầu tư vào công ty niêm yết hoặc công ty đầu tư kinh doanh.

Các tin của một trung tâm giao dịch thường có dạng tương đối giống nhau.

Ví dụ tin từ trung tâm giao dịch HASTC:

SVC: Kết quả giao dịch cổ phiếu của cổ đông nội bộ
(Cập nhật: 13/05/2008)

Căn cứ theo báo cáo giao dịch cổ phiếu của Tổng Công ty Bến Thành, TTGDCK Hà nội trân trọng thông báo như sau:

1. Tên tổ chức thực hiện giao dịch: Tổng Công ty Bến Thành
2. Đại diện: Nguyễn Quang Tiên
3. Chức vụ tại Tổ chức niêm yết: Chủ tịch HĐQT
4. Mã chứng khoán giao dịch: SVC;
5. Số lượng cổ phiếu nắm giữ trước khi thực hiện giao dịch: 5.056.420 cổ phiếu;
6. Số lượng cổ phiếu đã mua: 19.000 cổ phiếu;
7. Số lượng cổ phiếu nắm giữ sau khi thực hiện giao dịch: 5.075.420 cổ phiếu;
8. Mục đích: tăng tỷ lệ sở hữu
9. Thời gian giao dịch: 2/4/2008 đến ngày 2/5/2008

VC7: Giao dịch cổ phiếu của cổ đông nội bộ
(Cập nhật: 13/05/2008)

Ngày 13/5/2008 TTGDCK Hà Nội đã nhận được báo cáo giao dịch cổ phiếu của ông Nguyễn Văn Khắc cổ đông nội bộ CTCP Xây dựng số 7, TTGDCK Hà Nội trân trọng thông báo như sau:

1. Tên người thực hiện giao dịch: Ông Nguyễn Văn Khắc - Ủy viên Hội đồng quản trị - Phó Giám đốc Công ty
2. Mã chứng khoán giao dịch: VC7
3. Số lượng cổ phiếu sở hữu trước khi thực hiện giao dịch: 15.342 CP
4. Số lượng cổ phiếu đăng ký mua: 5.000 CP
5. Số lượng cổ phiếu nắm giữ sau khi thực hiện giao dịch: 20.342 CP
6. Mục đích thực hiện giao dịch: mua đầu tư
7. Thời gian giao dịch dự kiến: từ ngày 13/5/2008 đến ngày 13/6/2008.

SD5: Bổ sung nhân sự
(Cập nhật: 14/05/2008)

Ngày 9/5/2008, Trung tâm GDCK Hà Nội nhận được công văn số 163/SDD5-TCKT ngày 6/5/2008 của CTCP Sông Đà 5 về việc bổ nhiệm cán bộ của Công ty, TTGDCK Hà Nội xin thông báo như sau:

Kể từ ngày 20/4/2008, ông Nguyễn Bá Viễn được bầu là Phó Tổng Giám đốc phụ trách kỹ thuật của Công ty.

DTC: BCTC quý I/2008
(Cập nhật: 14/05/2008)

*** File đính kèm 2:** Dong Trieu Viglacera_BCTC quy 1.08.xls ( )

SJC được chấp thuận nguyên tắc niêm yết bổ sung cổ phiếu
(Cập nhật: 09/05/2008)

Ngày 9/5/2008, TTGDCK Hà Nội đã có quyết định chấp thuận về nguyên tắc cho CTCP Sông Đà 1.01 (MCK: SJC) được niêm yết bổ sung cổ phiếu tại TTGDCK Hà Nội với những nội dung sau:

- Mã chứng khoán: SJC;
- Số lượng chứng khoán được niêm yết bổ sung: 890.000 CP
- Tổng giá trị niêm yết bổ sung (theo mệnh giá): 8,9 tỷ đồng đồng

Ví dụ tin từ trung tâm giao dịch HOSE:

BT6: Kết quả giao dịch cổ phiếu cổ đông lớn

Kết quả giao dịch cổ phiếu cổ đông lớn của Công ty Cổ Phần Bê Tông 620 Châu Thới như sau:

Tên nhà đầu tư thực hiện giao dịch : Công ty TNHH Đầu tư Xây dựng và Phát triển Tân Việt
Mã chứng khoán thực hiện bán : BT6
Số lượng cổ phiếu nắm giữ trước khi thực hiện giao dịch: 1.982.076 cổ phiếu chiếm tỷ lệ :18,02 %
Số lượng cổ phiếu thực hiện giao dịch bán : 860.465 cổ phiếu
Số lượng cổ phiếu nắm giữ sau khi thực hiện giao dịch: 1.121.611 cổ phiếu chiếm tỷ lệ :10,2 %
Thời gian thực hiện giao dịch ngày : 06/05/2008 .

STB: Kết quả giao dịch cổ phiếu của cổ đông nội bộ

Kết quả giao dịch mua cổ phiếu STB của cổ đông nội bộ Ngân hàng TMCP Sài Gòn Thương Tín (Sacombank) như sau:

1. Người thực hiện giao dịch: Nguyễn Tấn Thành
Chức vụ hiện nay tại tổ chức niêm yết: Trưởng Ban kiểm soát
Mã chứng khoán: STB
Số lượng cổ phiếu sở hữu trước khi giao dịch: 321.866 cổ phiếu
Số lượng cổ phiếu đăng ký mua: 50.000 cổ phiếu
Số lượng cổ phiếu đã mua: 50.000 cổ phiếu
Số lượng cổ phiếu nắm giữ sau khi thực hiện giao dịch: 371.866 cổ phiếu
Thời hạn thực hiện giao dịch: từ ngày 02/04 đến 22/04/2008

2. Người thực hiện giao dịch: Nguyễn Minh Tâm
Chức vụ hiện nay tại tổ chức niêm yết: Phó tổng giám đốc
Mã chứng khoán: STB
Số lượng cổ phiếu sở hữu trước khi giao dịch: 63.168 cổ phiếu
Số lượng cổ phiếu đăng ký mua: 15.000 cổ phiếu
Số lượng cổ phiếu đã mua: 15.000 cổ phiếu
Số lượng cổ phiếu nắm giữ sau khi thực hiện giao dịch: 78.168 cổ phiếu
Thời hạn thực hiện giao dịch: từ ngày 02/04 đến 08/04/2008

PPC: Kết quả giao dịch cổ phiếu của tổ chức có liên quan đến cổ đông nội bộ

Kết quả giao dịch cổ phiếu của tổ chức có liên quan đến cổ đông nội bộ CTCP Nhiệt Điện Phả Lại như sau:
Tên tổ chức thực hiện giao dịch: Công ty tài chính dầu khí – PVFC
Người đại diện : Ông Đàm Minh Đức
Chức vụ hiện nay tại tổ chức niêm yết : Ủy viên hội đồng quản trị
Mã chứng khoán giao dịch: **PPC**
Số lượng, tỷ lệ cổ phiếu nắm giữ trước khi mua**: 7.491.625 cp** (chiếm 1.96%)
Số lượng cổ phiếu đăng kí mua : **1.000.000 cp**
Số lượng cổ phiếu đã mua: **36.000 cp**
Số lượng, tỷ lệ sở hữu sau khi mua: **7.527.625 cp** (chiếm 2%)
Thời gian thực hiện giao dịch : **Từ 28/03/2008 đến 30/04/2008**

PJT: Niêm yết bổ sung cổ phiếu

Ngày 08 tháng 05 năm 2008, Phó Tổng Giám đốc Lê Nhị Năng đã ký Quyết định chấp thuận cho Công ty Cổ phần Vận tải Xăng dầu đường thủy Petrolimex được niêm yết bổ sung cổ phiếu phát hành thêm với nội dung như sau:
Loại chứng khoán: cổ phiếu phổ thông
Mã chứng khoán: PJT
Mệnh giá: 10.000 đồng/ cổ phiếu
Số lượng chứng khoán niêm yết bổ sung: 3.500.000 cổ phiếu
(Ba triệu năm trăm ngàn cổ phiếu)
Tổng giá trị niêm yết bổ sung (theo mệnh giá): 35.000.000.000 đồng
(Ba mươi lăm tỷ đồng chẵn)

SSC: Báo cáo thường niên 2007

Báo cáo thường niên 2007 của CTCP Giống cây trồng Miền Nam:

-

File đính kèm:
20080512_SSC_BCTN 2007.doc

Nhận xét:
- Mặc dù cấu trúc các tin khá giống nhau nhưng việc nhập tin không theo một form chung nên nhiều tin về cùng một chủ đề có thứ tự thông tin là khác nhau thậm chí có tin có nội dung này, có tin không.
- Các tin về báo cáo tài chính phần lớn là sử dụng file đính kèm mà không đưa tin trực tiếp lên trang web, nếu có thì thường để ở cấu trúc dạng bảng.

6.3.2 Đặc điểm của tin từ các báo điện tử

Tin tức chứng khoán từ các báo điện tử thường nằm trong các dạng:

- ✓ Tin về ngành, lĩnh vực sản xuất, kinh doanh.
- ✓ Tin về hoạt động kinh doanh của một công ty có cổ phiếu niêm yết.
- ✓ Tin về thiên tai, thảm họa có liên quan đến một vài ngành, một vài công ty.
- ✓ Tin chung, tin chính sách cũng liên quan đến một vài ngành hay công ty.

Ví dụ một vài tin từ báo điện tử Economy.vn:

Cá tra, basa gặp khó vì Quyết định 346

Nguyễn Huyền

Các doanh nghiệp thủy sản ĐBSCL vừa kiến nghị Chính phủ một số chủ trương và giải pháp cụ thể để "gỡ khó" cho doanh nghiệp.

Vốn "đóng băng"!

Theo ông Phan Văn Danh, Chủ tịch Hiệp hội Nghề nuôi và chế biến thủy sản tỉnh An Giang (AFA) , trong suốt 40 ngày qua con cá tra, basa ở ĐBSCL đang chịu sự tác động bất lợi rất lớn và đang trong tình hình hết sức khó khăn, nhất là từ khi có Quyết định 346/QĐ-NHNN ngày 13/2/2008 của Thống đốc Ngân hàng Nhà nước về chống lạm phát và kiểm soát tăng giá.
. . .

Khởi công Khu liên hợp Lọc hóa dầu Nghi Sơn

Từ Nguyên

Sáng 10/5, Tập đoàn Dầu khí và các bên liên doanh đã chính thức khởi công Khu liên hợp Lọc hóa dầu Nghi Sơn.

Khu liên hợp Lọc hóa dầu Nghi Sơn được đặt tại Khu kinh tế Nghi Sơn (Thanh Hóa) có tổng vốn đầu tư là 6,2 tỷ USD do Tập đoàn Dầu khí (Petro Vietnam) và các bên liên doanh là: Công ty Dầu khí quốc tế Kuwait, Công ty Idemitsu Kosan (Nhật Bản) và Công ty Hóa chất Mitsui (Nhật Bản) góp vốn thành lập.

Các tin tức này sẽ có cấu trúc hoàn toàn khác nhau, theo văn phong ngôn ngữ báo.

Chương 7. Thiết kế và cài đặt Module phân loại và trích thông tin sự kiện CK

7.1. Thiết kế và xây dựng Ontology sự kiện

Quá trình xây dựng Ontology sự kiện stockEvent.owl được thực hiện bằng tay theo phương pháp MethOntology.

1. Đặc tả Ontology sự kiện:

- Mục đích Ontology: xây dựng một Ontology có khả năng phân loại được các tin tức chứng khoán và trích chọn giá trị thuộc tính.
- Phạm vi Ontology: Bao gồm các loại sẽ được phân chia cho tin tức chứng khoán và chỉ thuộc riêng về lĩnh vực chứng khoán

2. Nguồn dữ liệu cho Ontology sự kiện:

Như đã phân tích ở trên thì nguồn dữ liệu sẽ là hai trung tâm giao dịch chứng khoán và các báo điện tử.

3. Khái niệm hóa Ontology:

Căn cứ vào nguồn dữ liệu trên thì các khái niệm (các loại) sẽ được xác định cho Ontology sự kiện. Chúng bao gồm các khái niệm với mô tả như sau:

- ✓ Event : sự kiện chứng khoán nói chung, bao gồm các thuộc tính
  - eventDate: ngày thông báo sự kiện
  - eventSource: nguồn sự kiện (URL)
  - eventStatus: trạng thái sự kiện (PENDING hay COMPLETE)
- ✓ StockEvent : lớp con của Event : sự kiện về cổ phiếu, gồm thuộc tính
  - eventOfStock: sự kiện của cổ phiếu nào
  - hasWarning: sự kiện này có cảnh báo gì
  - ttThayDoiDiemSo: thuộc tính mà căn cứ vào đó để thay đổi điểm xếp hạng
- ✓ StockExchange : lớp con của StockEvent : sự kiện về giao dịch CP
  - tenNDT: tên nhà đầu tư
  - chucvuNDT: chức vụ nhà đầu tư
  - tenNLQ: tên người liên quan
  - chucvuNLQ: chức vụ người liên quan
  - mua_ban: giao dịch mua hay bán
  - cpTruocGD: số lượng cổ phiếu trước giao dịch
  - cpDangKiGD: số lượng cổ phiếu đăng kí giao dịch
  - cpDaGD: số lượng CP đã giao dịch
  - cpSauGD: số lượng cổ phiếu sau giao dịch
  - tuNgay: ngày bắt đầu giao dịch
  - denNgay: ngày kết thúc giao dịch
  - loaiGiaoDich: loại giao dịch: cổ đông nội bộ, cổ đông lớn,...
  - mucdichGD: mục đích tiến hành giao dịch
  - phuongthucGD: phương thức giao dịch
- ✓ StockMoreIssue : lớp con của StockEvent : sự kiện về phát hành thêm CP
  - luongCPThem: số lượng cổ phiếu thêm
  - giaTriCPThem: giá trị cổ phiếu thêm
  - menhGiaCPThem: mệnh giá cổ phiếu thêm
  - loaiCK: loại chứng khoán
  - ngayPhatHanh: ngày phát hành cổ phiếu thêm

- ngayGDChinhThuc: ngày cổ phiếu chính thức giao dịch thêm

✓ StockAdd_Split : lớp con của StockEvent : sự kiện về tách/gộp CP
- tach_gop: tách hay gộp cổ phiếu
- tiLe: tỉ lệ tách/gộp
- ngayTachGop: ngày tách/gộp

✓ CompanyEvent : lớp con của Event : sự kiện về công ty niêm yết
- eventOfCompany: sự kiện này của cổ phiếu nào (tương ứng với công ty niêm yết)

✓ ChangeManagement : lớp con của CompanyEvent : sự kiện về thay đổi ban lãnh đạo
- nguoiBoNhiem: tên người được bổ nhiệm
- chucvuBoNhiem: chức vụ được bổ nhiệm
- nguoiMienNhiem: tên người miễn nhiệm
- chucvuMienNhiem: chức vụ miễn nhiệm
- ngayThayDoi: ngày thay đổi

✓ CompanyInvest : lớp con của CompanyEvent : sự kiện về việc công ty đầu tư kinh doanh
- linhvucDT: lĩnh vực đầu tư vào
- luongvonDT: lượng vốn đầu tư
- thoigianDT: thời gian đầu tư

✓ InvestReport : lớp con của CompanyEvent : sự kiện về việc công ty được đầu tư.
- tenToChucDT: tên tổ chức thực hiện đầu tư
- trongNuoc_NN: trong nước hay nước ngoài
- vonDT: lượng vốn đầu tư
- ngayDT: thời gian đầu tư

✓ ProfitReport : lớp con của CompanyEvent : sự kiện về báo cáo kinh doanh công ty
- LN_Ki: kì kế toán nào
- luongTangGiam: lượng tăng/giảm

✓ SectorEvent : lớp con của Event : sự kiện về ngành
- eventOfSector: sự kiện về ngành nào

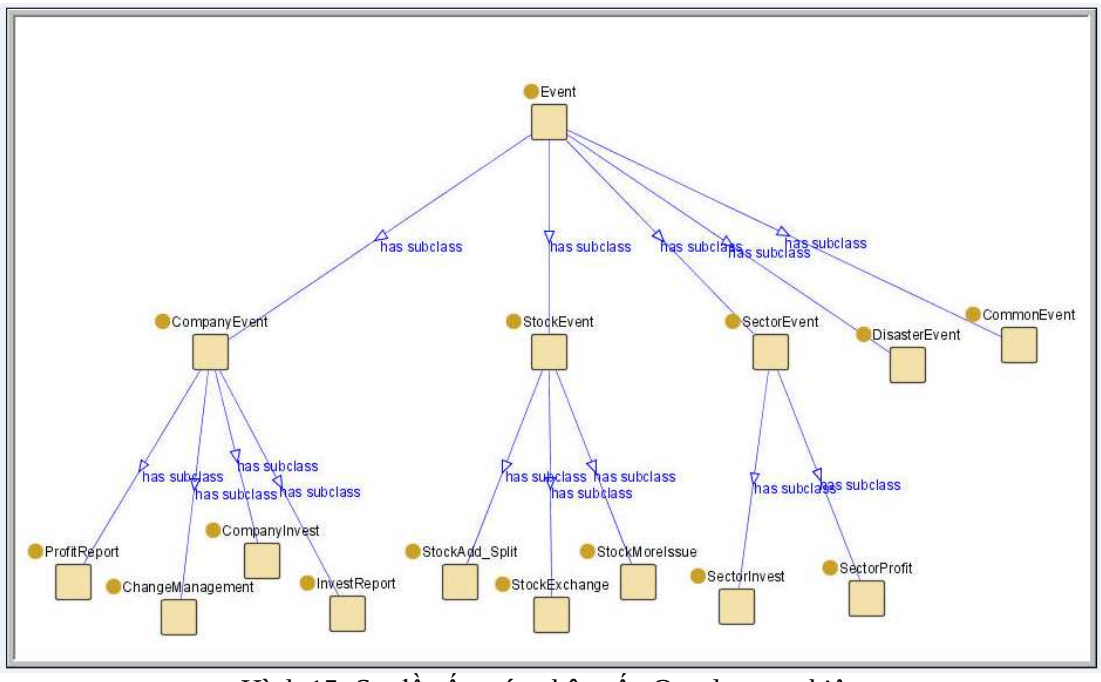
✓ SectorInvest : lớp con của SectorEvent : sự kiện về việc đầu tư vào ngành

✓ SectorProfit : lớp con của SectorEvent : sự kiện về tình hình kinh doanh trong ngành (thua lỗ, lợi nhuận)

✓ DisasterEvent : lớp con của Event : sự kiện về thiên tai, thảm họa
- nganhLQTH: ngành bị liên quan trong thảm họa này
- congtyLQTH: công ty có liên quan

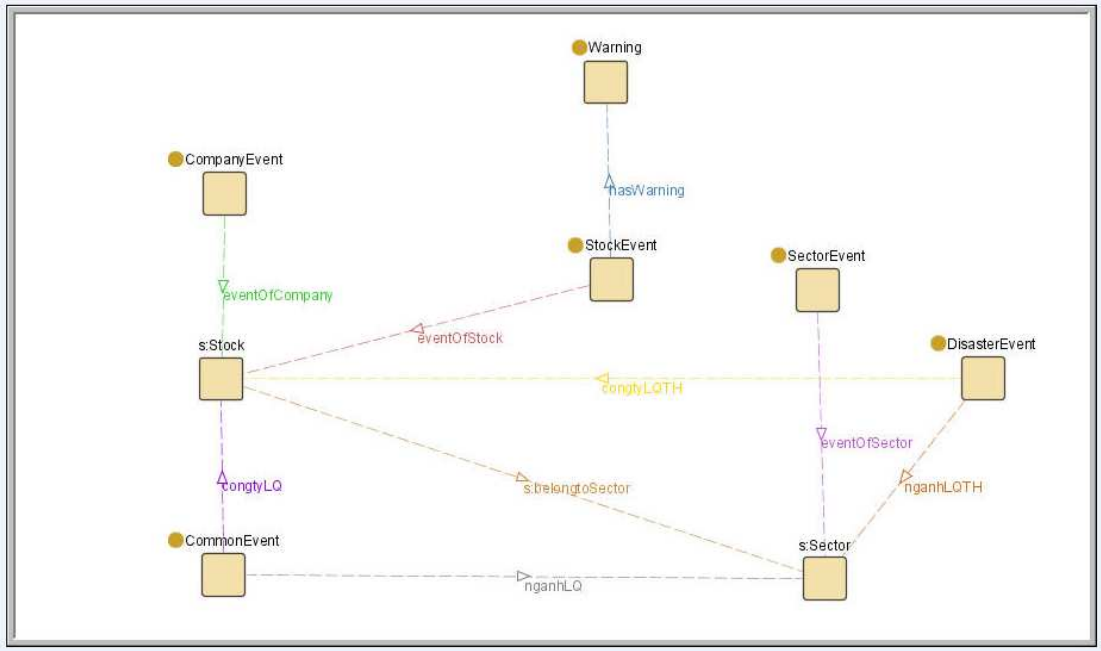
✓ CommonEvent : lớp con của Event : sự kiện chung, chính sách nhà nước.
- nganhLQ: ngành có liên quan
- congtyLQ: công ty có liên quan

Sơ đồ cấu trúc phân cấp trong Ontology sự kiện được xây dựng bằng Protege được thể hiện ở Hình bên dưới.



Hình 15: Sơ đồ cấu trúc phân cấp Ontology sự kiện

Sơ đồ quan hệ các lớp của Ontology sự kiện được thể hiện trong Hình bên dưới.



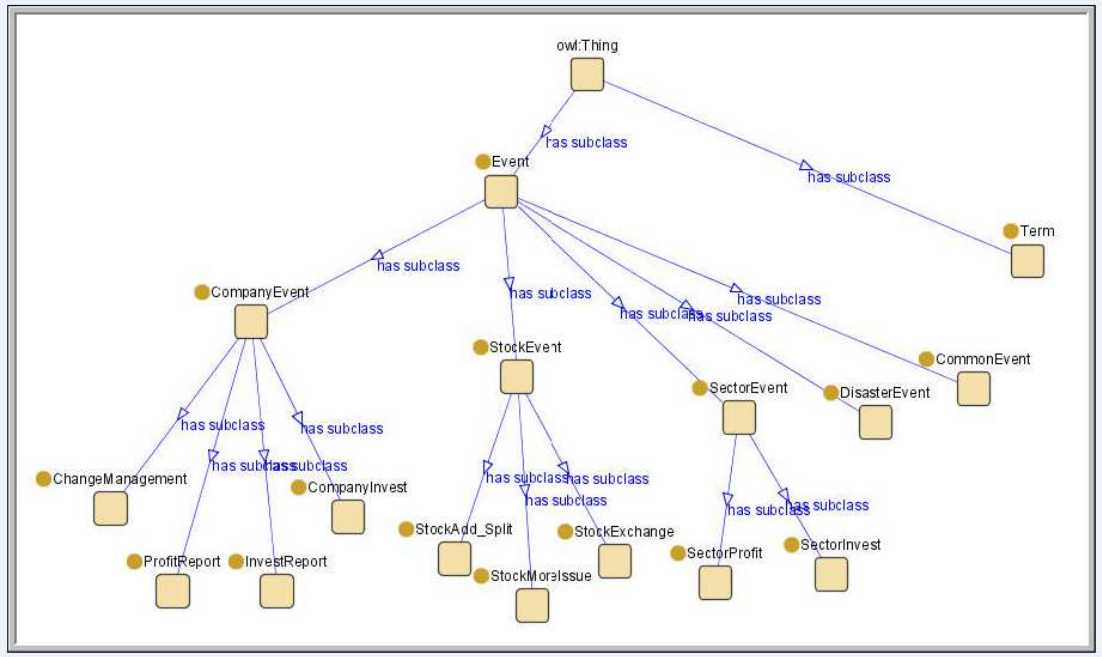
Hình 16: Sơ đồ quan hệ các lớp trong Ontology sự kiện

4. Hình thức hóa và cài đặt Ontology sự kiện:

Ontology sự kiện được hình thức hóa thành ngôn ngữ OWL-DL sử dụng trình soạn thảo Protégé.

Nếu Ontology sự kiện chỉ dùng riêng cho mục đích phân loại tin chứng khoán thì nó sẽ chỉ chứa các thuật ngữ của mỗi khái niệm nhưng do cần xác định cả giá trị các thuộc tính nên phải xây dựng một Ontology tiếp theo gọi là Ontology cơ sở tri thức sự kiện (KB Ontology).

Lý do cần xây dựng KB Ontology là để ngoài phân loại sự kiện chứng khoán thì còn xác định giá trị cho các thuộc tính. Như vậy về kiến trúc phân cấp các khái niệm phân loại thì KB Ontology sẽ hoàn toàn giống như của Ontology sự kiện (Xem Hình)



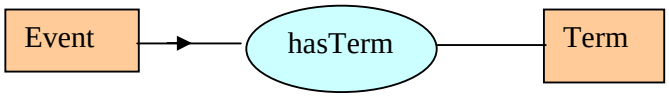
Hình 17: Sơ đồ cấu trúc phân cấp các lớp trong KB Ontology

Tuy nhiên trong KB Ontology thì có thêm một khái niệm (lớp) là *Term*. Đây là lớp sẽ chứa vector thuật ngữ cho mỗi khái niệm phân loại và cả thuật ngữ để xác định giá trị thuộc tính.

Lớp *Term* gồm các thuộc tính:

- ✓ termValue : giá trị của thuật ngữ
- ✓ termWeight : trọng số của thuật ngữ
- ✓ eventProperty : tên khái niệm hoặc thuộc tính)
- ✓ propertyType : kiểu (ví dụ khái niệm sẽ có kiểu Class, thuộc tính kiểu string, int, float, . ..)
- ✓ synonym : một tập các thuật ngữ đồng nghĩa

Quan hệ giữa các khái niệm phân loại với Term được thể hiện thông qua quan hệ hasTerm của lớp Event. Vì mọi khái niệm phân loại đều là con của Event nên theo luật suy diễn chúng cũng sẽ có quan hệ hasTerm với lớp Term.



Cài đặt KB Ontology:

File Ontology là FullEventKB.owl

Các thuật ngữ cho các khái niệm phân loại sự kiện có sự khác nhau. Cụ thể với sự kiện cổ phiếu StockEvent và sự kiện công ty CompanyEvent thì vector khái niệm chính là danh sách tất cả các cổ phiếu trên thị trường chứng khoán Việt Nam.

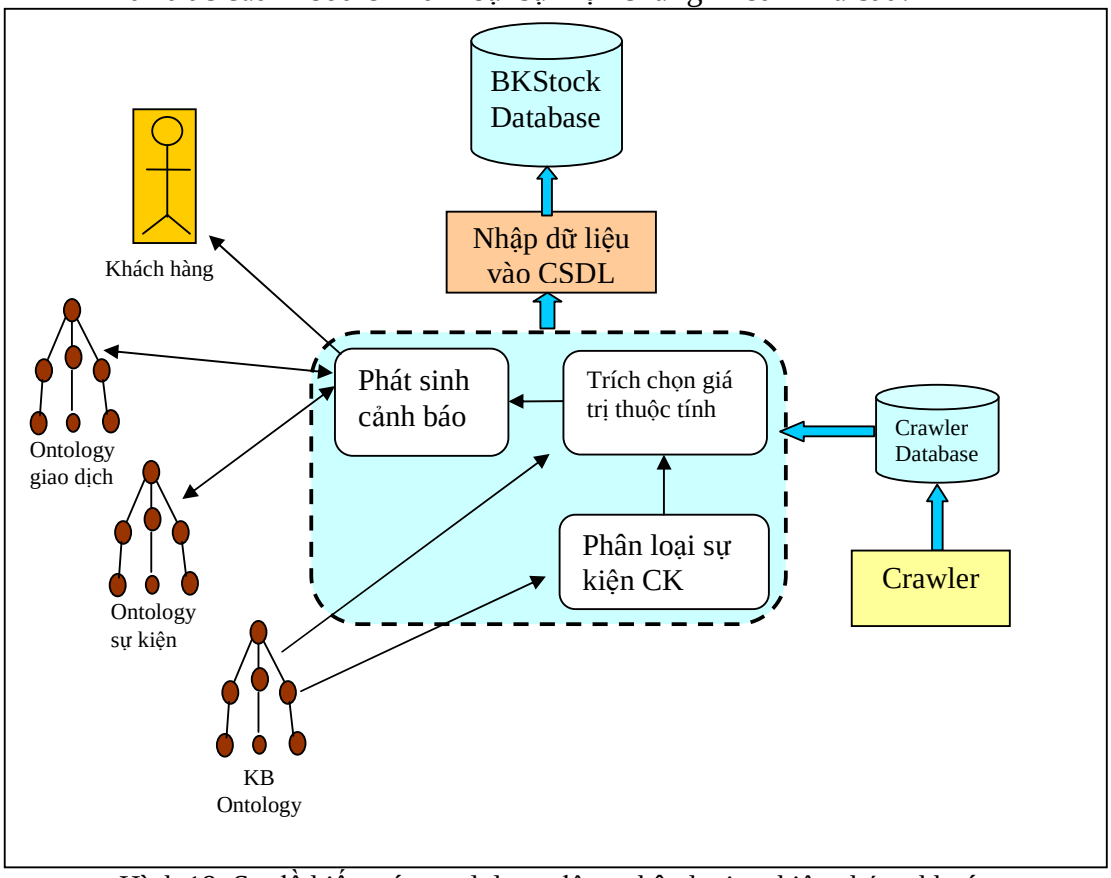
Lý do bởi vì các StockEvent và CompanyEvent sẽ chỉ lấy từ hai trung tâm giao dịch chứng khoán mà qua nghiên cứu thì các sự kiện thuộc hai loại này đều có dấu hiệu phân biệt rõ nhất là tên (mã) cổ phiếu. Trong phạm vi đồ án này, một sự kiện chỉ được coi là sự kiện cổ phiếu (StockEvent) nếu nó thỏa mãn một trong ba sự kiện con của StockEvent là StockExchange, StockMoreIssue, StockAdd_Split. Mỗi lớp này sẽ có một thể hiện chứa vector thuật ngữ để phục vụ việc phân loại. Tương tự như thế là với lớp CompanyEvent, một sự kiện chỉ được phân loại vào đây nếu nó thuộc một trong bốn lớp con của nó.

Thuật ngữ cho sự kiện ngành (SectorEvent), sự kiện thảm họa (DisasterEvent), sự kiện chung (CommonEvent) bao gồm đầy đủ giá trị thuật ngữ và trọng số. Chúng được lấy thông qua tri thức chuyên gia hoặc quá trình trích thuật ngữ tự động trong tài liệu. Điều này sẽ được giải thích rõ hơn trong phần 7.3.

Các thuật ngữ để trích chọn thông tin từ sự kiện chứng khoán có dạng tương tự như thuật ngữ để phân loại sự kiện. Một thể hiện của lớp Term chứa thuật ngữ trích chọn thông tin sẽ có: *eventProperty* là tên thuộc tính cần lấy giá trị, *propertyType* sẽ là một trong các kiểu string, select, int, float, date, dateTime.

7.2. Thiết kế chương trình

Kiến trúc của Module Phân loại sự kiện chứng khoán như sau:



Hình 18: Sơ đồ kiến trúc module tự động phân loại sự kiện chứng khoán

Module gồm ba thành phần chính là: Phân loại sự kiện chứng khoán, Trích chọn giá trị thuộc tính và Phát sinh cảnh báo. Dữ liệu đầu vào là Crawler Database là cơ sở dữ liệu về các bản tin đã được module Crawler tiến hành crawl trên mạng và thêm vào. Sử dụng ba Ontology là Ontology sự kiện, KB Ontology và Ontology

giao dịch cổ phiếu. Quá trình phân loại sự kiện và trích chọn giá trị thuộc tính đều sử dụng dữ liệu từ Crawler Database và KB Ontology; trong khi quá trình phát sinh cảnh báo sẽ sử dụng giá trị thuộc tính đã trích chọn và có sự hỗ trợ của Ontology sự kiện với Ontology giao dịch để đưa ra thông báo cảnh báo cho khách hàng và thay đổi điểm xếp hạng cổ phiếu. Kết quả phân loại tin chứng khoán cũng được nhập vào BKStock Database.

Thiết kế Module Crawler

Module Crawler là một module bổ trợ bên ngoài có nhiệm vụ crawl các trang web và trích ra thông tin. Nguyên tắc khi tiến hành Crawl là phát hiện được thẻ xác định nội dung thông tin cần trích ra. Ở đây, thực hiện Crawl trên ba website là:

- ✓ HASTC: http://hastc.org.vn
- ✓ HOSE: http://vse.org.vn
- ✓ Economy.vn: http://economy.vn

Đầu vào là địa chỉ URL của trang cần trích. Thông tin được trích ra sẽ bao gồm:

- ✓ title: Tiêu đề của bản tin
- ✓ date: ngày phát hành bản tin
- ✓ content: nội dung bản tin

Kết quả được lưu trong bảng NguonSuKien có cấu trúc như sau:

Tên trường	*Kiểu*	*Null*	*Giải thích*
MaNguonSuKien	int	no	
TieuDe	nvarchar	yes	
NoiDung	nvarchar	yes	
Ngay	datetime	yes	
DiaChiURL	nvarchar	yes	
NguonGocSuKien	int	yes	1: HASTC; 2: HOSE; 3 : Economy.vn

Module tìm thuật ngữ cho phân loại sự kiện ngành

Ý tưởng của module này sẽ là: lấy một tập đào tạo (training set) là các tài liệu thuộc một loại đã biết trước, tách các từ trong một tài liệu, xác định trọng số các từ, chọn lấy những từ có trọng số cao nhất làm thuật ngữ phân loại sự kiện.

Các từ phải được tách theo cấu trúc ngữ pháp tiếng Việt, không phải tách theo dấu hiệu phân tách thông thường. Trọng số các từ sẽ bao gồm tần suất từ trong tài liệu và ảnh hưởng của số lượng tài liệu chứa từ này.

Công thức đề nghị như sau:

$$W(w, d) = TF(w) * DF(w)/D$$

trong đó W(w,d) là trọng số của từ w trong tài liệu d.

TF(w) là tần suất của từ w trong tài liệu d, được tính bằng số lần xuất hiện w trong d trên tổng số từ của tài liệu d.

DF(w) là số tài liệu chứa từ w và D là số tài liệu xét.

Ở đây nhấn mạnh đến số tài liệu chứa từ w để tăng độ đại diện của từ w với tài liệu d.

Chú ý: phải loại bỏ stopwords (từ thường xuất hiện và không mang ý nghĩa đại diện cho tài liệu) trước khi thực hiện tìm từ.

Thiết kế chức năng phân loại sự kiện chứng khoán

Việc phân loại sự kiện sẽ được tiến hành theo thứ tự như sau:
StockEvent → CompanyEvent
→ SectorEvent → DisasterEvent → CommonEvent

Cách thực hiện sẽ là đọc vector khái niệm của mỗi lớp, bao gồm cả giá trị và trọng số thuật ngữ, kết hợp với xử lý thuật ngữ đồng nghĩa để lấy về một vector khái niệm đầy đủ. Xét một bản tin, tạo vector đặc tính của nó tương ứng với vector khái niệm. Trọng số của mỗi thuật ngữ là số lần xuất hiện của nó trong bản tin. Tính tích vô hướng hai vector, khái niệm nào cho giá trị lớn nhất thì phân loại sự kiện về khái niệm đó.

Sau khi đã phân loại sự kiện vào một trong các loại tổng quát trên thì tiếp tục phân loại nó thành những loại nhỏ hơn. Cách thực hiện tương tự trên.

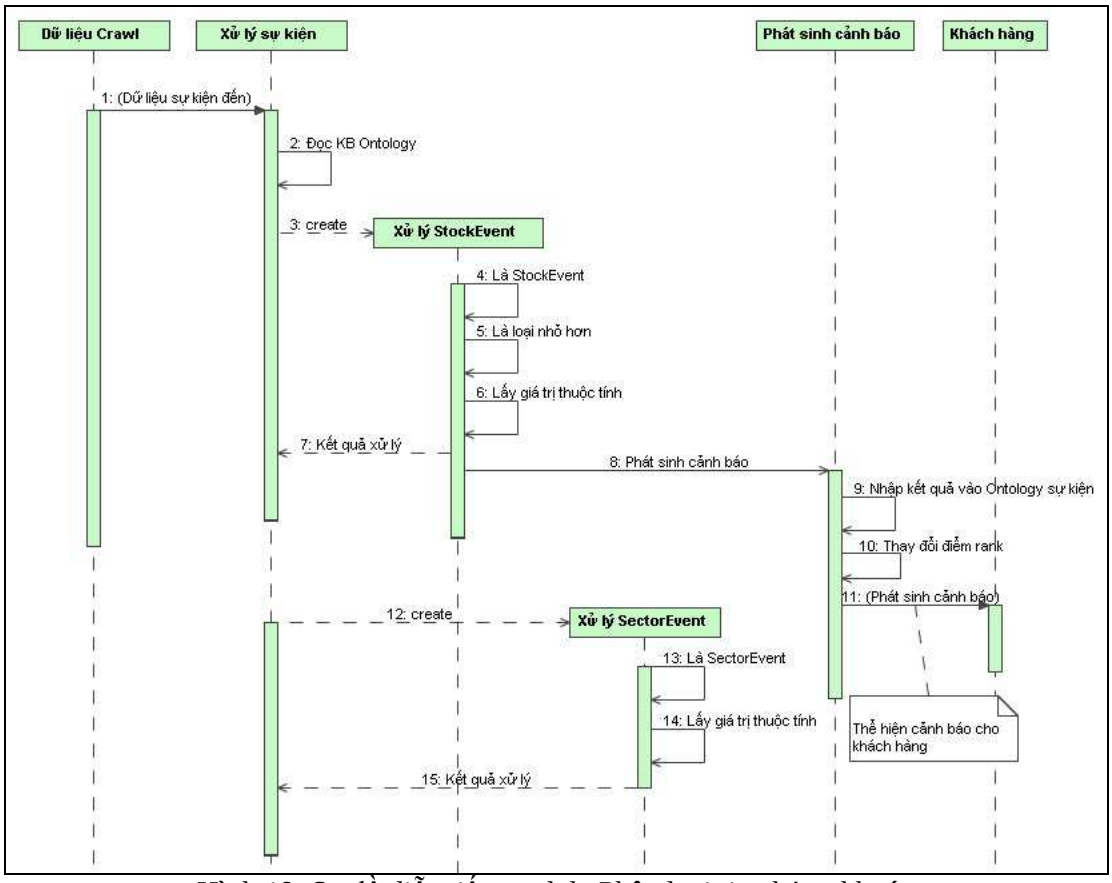
Thiết kế chức năng xác định giá trị thuộc tính

Giá trị thuộc tính được xác định theo mỗi khái niệm phân loại. Nếu trong KB Ontology có xây dựng các tập thuật ngữ cho thuộc tính nào thì có thể xác định được giá trị thuộc tính đó trong sự kiện. Nguyên tắc là truy vấn ra giá trị thuật ngữ, tên thuộc tính và kiểu thuộc tính. Đối với StockEvent hoặc CompanyEvent thì kiểm tra từng dòng một, nếu thỏa mãn tập thuật ngữ của thuộc tính nào thì tiến hành tìm giá trị. Căn cứ vào kiểu thuộc tính là string, select, int, float hay date, dateTime mà có phương pháp tìm phù hợp. Với SectorEvent thì có thể kiểm tra theo đoạn hoặc toàn bộ sự kiện. Tập các giá trị thuộc tính sau khi tìm được sẽ nhập vào Ontology sự kiện để xác định khả năng đưa ra cảnh báo (nếu có)

Thiết kế chức năng đưa ra cảnh báo

Các thuộc tính và giá trị của chúng mà có thể gây ra cảnh báo sẽ được xác định trong Ontology sự kiện. Căn cứ vào giá trị các thuộc tính tìm được mà thành phần Phát sinh cảnh báo sẽ đưa ra cảnh báo hay không. Nếu giá trị lớn hơn một ngưỡng cho trước thì điểm xếp hạng của cổ phiếu tương ứng sẽ được thay đổi, giá trị thay đổi cũng được cung cấp trong Ontology sự kiện. Giá trị thay đổi có thể là tăng hoặc giảm, tùy thuộc vào giá trị thuộc tính tìm được so với giá trị đã cho.

Sơ đồ diễn tiến của Module phân loại sự kiện chứng khoán



Hình 19: Sơ đồ diễn tiến module Phân loại tin chứng khoán

7.3. Cài đặt chương trình và kết quả

Module Crawler:

Cài đặt thuật toán trích nội dung thông tin từ nội dung HTML của trang Web và hai dấu hiệu phân tách đầu và cuối

```
/**
	* Ham thuc su trich noi dung theo dau hieu phan tach dau va cuoi
	* @param htmlContent
	* @param headDelimiter
	* @param tailDelimiter
	* @param keepHead - true then keep headDelimiter
	* @return
*/
public String extractContent(String htmlContent, String headDelimiter,
String tailDelimiter,
                boolean keepHead) throws ExtractContentException {
	if(htmlContent == null) throw new ExtractContentException("No HTML
Content");
	String content = null;
	int headIdx = htmlContent.indexOf(headDelimiter);
	int tailIdx = htmlContent.indexOf(tailDelimiter, headIdx+1);
	if(headIdx == -1 || tailIdx == -1) return content;

	if(keepHead){
		content = htmlContent.substring(headIdx, tailIdx);
	}
```

```
	else {
		content =
htmlContent.substring(headIdx+headDelimiter.length(), tailIdx);
	}

	return content;
}
```

Cài đặt Module Phân loại sự kiện chứng khoán

Toàn bộ module Phân loại sự kiện chứng khoán được chia thành các lớp:

1. EventCommonUtils : Chứa các hàm dùng chung trong module
2. EventStock : Phân loại sự kiện cổ phiếu và trích giá trị thuộc tính
3. EventSector : Phân loại sự kiện ngành
4. EventDataInput : Nhập giá trị thuộc tính vào Ontology sự kiện và phát sinh cảnh báo nếu có
5. EventExample : Demo quá trình xử lý phân loại sự kiện
6. EventExampleGUI : giao diện người dùng

Các truy vấn quan trọng:

1. Truy vấn lấy ra các giá trị thuật ngữ của một lớp có tên ‘eventType’:

```
PREFIX kb: <" + kbNS + ">"
"PREFIX     rdfs: <http://www.w3.org/2000/01/rdf-schema#>"
"SELECT ?term ?value "
"WHERE { ?x rdfs:label \"" + eventType + "\" ."
"     ?x kb:hasTerm ?term ."
"     ?term kb:termValue ?value ; "
"     kb:eventProperty \""+eventType+"\" ;"
"     kb:propertyType \"Class\" }";
```

2. Truy vấn lấy ra giá trị thuật ngữ của các thuộc tính của lớp có tên ‘eventType’:

```
"PREFIX kb: <" + kbNS +"> "
"PREFIX     rdfs: <http://www.w3.org/2000/01/rdf-schema#>"
"SELECT ?term ?value ?event ?type "
"WHERE { ?x rdfs:label \"" + eventType + "\" . "
"     ?x kb:hasTerm ?term . "
"     ?term kb:termValue ?value ; kb:eventProperty ?event ;"
"     kb:propertyType ?type ."
"     FILTER( ?type != \"Class\" ) }";
```

Câu truy vấn này sẽ lấy ra:
?value = giá trị thuật ngữ, ?event = tên thuộc tính, ?type = kiểu thuộc tính

3. Truy vấn lấy ra danh sách thuật ngữ để phân loại một ngành:

```
"PREFIX kb: <" + kbNS + ">"
"PREFIX     rdfs: <http://www.w3.org/2000/01/rdf-schema#> "
"SELECT ?term ?value ?sectorName ?x "
"WHERE { ?x rdfs:label \"SectorEvent\" ; "
"     kb:sectorName ?sectorName ;"
"     kb:hasTerm ?term ."
"     ?term kb:termValue ?value ; kb:propertyType \"Class\" }";
```

Câu truy vấn này sẽ lấy ra: ?sectorName = tên ngành cần phân loại

Cài đặt một số thuật toán quan trọng:

1. Thuật toán xác định một bản tin (strEvent) có là loại có tên ‘eventType’ hay không ? (Thuộc lớp EventCommonUtils.java)

```
/**
* Xac dinh <i>event</i> co phai la su kien thuoc loai <i>eventType</i>
hay ko?
* <br>Chu y la <i>event</i> da duoc phan loai thanh 1 trong cac loai :
* <ul><li>StockEvent</li><li>CompanyEvent</li><li>SectorEvent</li>
* <li>DisasterEvent</li><li>CommonEvent</li></ul>
*
* @param eventType loai su kien
* @return true, false
*/
@SuppressWarnings("unchecked")
public boolean isEventOfType(String strEvent, String eventType) {
	// Create termValuesMap
	Map<String, List<String>> termValuesMap =
createTermValuesMap(eventType);

	// Xac dinh xem su kien nay co la Event of eventType hay ko?
	strEvent = strEvent.toUpperCase();
	List<String> termList = (ArrayList<String>)
termValuesMap.get(eventType);
	for (String term : termList) {
		// Loc ra cac term phai cu`ng xuat hien trong title nay
		String[] requiredTerms = term.split("\\+");
		boolean match = true;
		for (String requiredTerm : requiredTerms) {
			int reqIdx = 0;
			if ((reqIdx = strEvent.indexOf(requiredTerm, reqIdx))
== -1) {
				match = false; // Neu 1 term ko thoa man => false
				break;
			}
		}
		if (match) {
			return true; // Neu da thoa man thi tra lai true luon,
ngc lai xu ly voi term khac
		}
	}

	// Cuoi cung ko co match = true nao thi tra lai false
	return false;
}
```

2. Thuật toán tạo ra vector thuật ngữ cho khái niệm , *termStr* là danh sách thuật ngữ, *weightStr* là danh sách trọng số cho thuật ngữ (trong file EventCommonUtils)

```
/**
* Tao ra cac sectorTerm bao gom ca termValue va termWeight
* @param rTerm
* @param termStr
* @param weightStr
* @return List
*/
public static List<EventTerm> createSectorTermList(Resource rTerm, String
termStr, String weightStr){
	List<EventTerm> sectorTermList = new ArrayList<EventTerm>();

	if(weightStr == null){
```

```
            List<String> termValueList = createTermList(rTerm, termStr);
            for(String termValue : termValueList){
                  double weight = 0.005;  // default weight
                  EventTerm term = new EventTerm(termValue, weight);
                  sectorTermList.add(term);
            }
            return sectorTermList;
      }
      // Tach lay cac term va weight tuong ung va dua vao List
      String[] terms = termStr.split(",");
      String[] weights = weightStr.split(",");
      List<Double> weightList = new ArrayList<Double>();
      for(String sWeight : weights){
            weightList.add(Double.parseDouble(sWeight));
      }
      // Lay ra cac tu dong nghia co dang "x=tu,dong,nghia"
      Map<String, String> synonymMap = null;
      if (rTerm.hasProperty(kbModel.getDatatypeProperty(kbNS +
"synonym"))) {
            synonymMap = new HashMap<String, String>();
            StmtIterator si =
Term.listProperties(kbModel.getDatatypeProperty(kbNS + "synonym"));
            for (; si.hasNext();) {
                  String s = si.nextStatement().getString();
                  String[] syns = s.split("=");
                  synonymMap.put(syns[0], syns[1]);
            }
      }
      EventTerm sectorTerm = null;
      for (int i=0; i<terms.length; i++) {
            String term = terms[i];
            double weight = weightList.get(i);
            // Xu ly them vao cac tu dong nghia, thay the cho {x} , neu
co
            if (term.indexOf("{") != -1 && term.indexOf("}") != -1) {
                  replaceBySynonymWithWeight(sectorTermList, weight,
synonymMap, term);
                  // remove the last appended object
                  sectorTermList.remove(sectorTermList.size()-1);
            }
            else {
                  sectorTerm = new EventTerm(term.toUpperCase(), weight);
                  sectorTermList.add(sectorTerm);
            }
      }
      return sectorTermList;
}
```

3. Thuật toán thay thế các thuật ngữ đồng nghĩa (có trọng số) (Trong lớp EventCommonUtils)

```
/**
* Thay the synonym va gan 1 trong so chung cho cac synonym
* @param sectorTermList
* @param weight - Trong so de gan
* @param synonymMap
* @param term
* @return
*/
public static String replaceBySynonymWithWeight(List<EventTerm>
sectorTermList, double weight,

      Map<String, String> synonymMap, String term) {
```

```
	String oldTerm = term;
	// Lay ra x trong {x} va thay the lan luot boi synonym
	while (term.indexOf("{") != -1 && term.indexOf("}") != -1) {
		String x = term.substring(term.indexOf("{") + 1,
term.indexOf("}"));
		String s = synonymMap.get(x);
		String syns[] = s.split(",");
		for (String syn : syns) {
			term = oldTerm;
			term = replaceBySynonymWithWeight(sectorTermList,
weight, synonymMap, term.replace("{" + x + "}", syn));
		}
		break;
	}
	// Neu chua co term nay thi add vao
	EventTerm sTerm = new EventTerm(term.toUpperCase(), weight);
	if (!sectorTermList.contains(sTerm)) {
		sectorTermList.add(sTerm);
	}
	return term;
}

/**
* Goi de quy de thay the lan luot tung tu dong nghia vao trong term
* @param synonymMap - chua cac tu dong nghia
* @param term - chua {x} can thay the
* @return term da thay the voi cac tu dong nghia
*/
public static String replaceBySynonym(List<String> termList, Map<String,
String> synonymMap, String term) {
	String oldTerm = term;
	// Lay ra x trong {x} va thay the lan luot boi synonym
	while (term.indexOf("{") != -1 && term.indexOf("}") != -1) {
		String x = term.substring(term.indexOf("{") + 1,
term.indexOf("}"));
		String s = synonymMap.get(x);
		String syns[] = s.split(",");
		for (String syn : syns) {
			term = oldTerm;
			term = replaceBySynonym(termList, synonymMap,
term.replace("{" + x + "}", syn));
		}
		break;
	}
	// Neu chua co term nay thi add vao
	if (!termList.contains(term.toUpperCase())) {
		termList.add(term.toUpperCase());
	}
	return term;
}
```

Kết quả cài đặt:

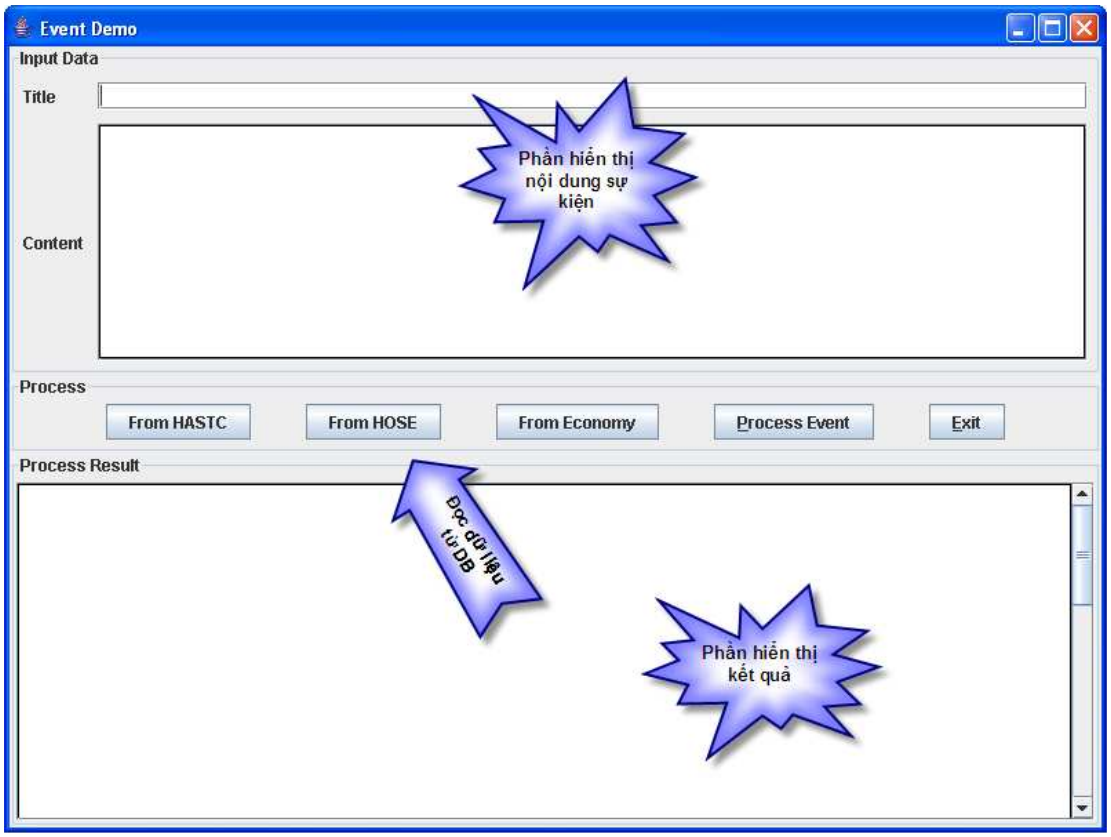
Module được cài đặt và chạy thử nghiệm với bộ dữ liệu crawl từ HASTC, HOSE và trang web Economy.vn. Những sự kiện đã được xây dựng thuật ngữ là: StockEvent (bao gồm cả ba sự kiện mức chi tiết hơn), SectorEvent (trong đó các ngành có thuật ngữ là: Nông sản và thủy hải sản, Thực phẩm, Ngân hàng). Về thuật ngữ cho các thuộc tính thì đã xây dựng cho các thuộc tính của lớp StockExchange (sự kiện giao dịch cổ phiếu), StockMoreIssue (sự kiện phát hành thêm cổ phiếu), StockAdd_Split (sự kiện tách/gộp cổ phiếu). Riêng lớp SectorEvent thì xây dựng thuật ngữ cho thuộc tính *mucdoAnhHuong*.

Các điều kiện phát sinh cảnh báo được cài đặt trong Ontology sự kiện như sau:

- Với sự kiện là StockExchange, nếu là giao dịch MUA thì điểm xếp hạng sẽ tăng lên, nếu là giao dịch BÁN (hoặc TẶNG, CHO) thì điểm xếp hạng sẽ giảm xuống.
- Sự kiện StockExchange, thuộc tính chucvuNDT (chức vụ nhà đầu tư) nằm trong Hội đồng quản trị, điểm xếp hạng thay đổi 5%.
- Sự kiện StockExchange, thuộc tính cpDangKiGD (cổ phiếu đăng kí giao dịch) hoặc cpDaGD (cổ phiếu đã giao dịch) có giá trị lớn hơn 0.5% của tổng cổ phiếu phát hành, điểm xếp hạng thay đổi 10%.

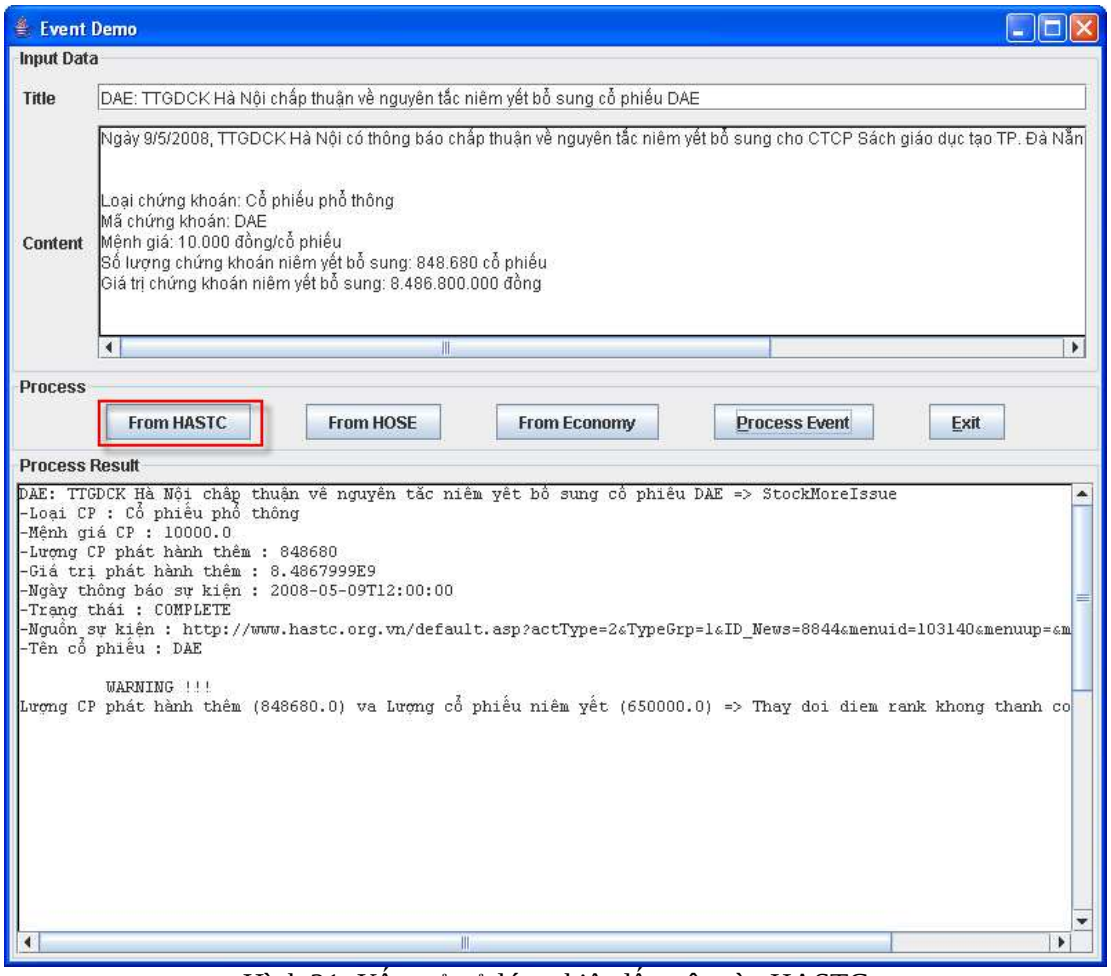
Sự kiện StockMoreIssue, thuộc tính luongCPThem (lượng cổ phiếu phát hành thêm) có giá trị lớn hơn 10% của tổng cổ phiếu niêm yết, điểm xếp hạng thay đổi 5%.

Giao diện của module đã được xây dựng để cung cấp khả năng quan sát chương trình hoạt động. Giao diện khi bắt đầu thực hiện module được thể hiện ở Hình bên dưới.

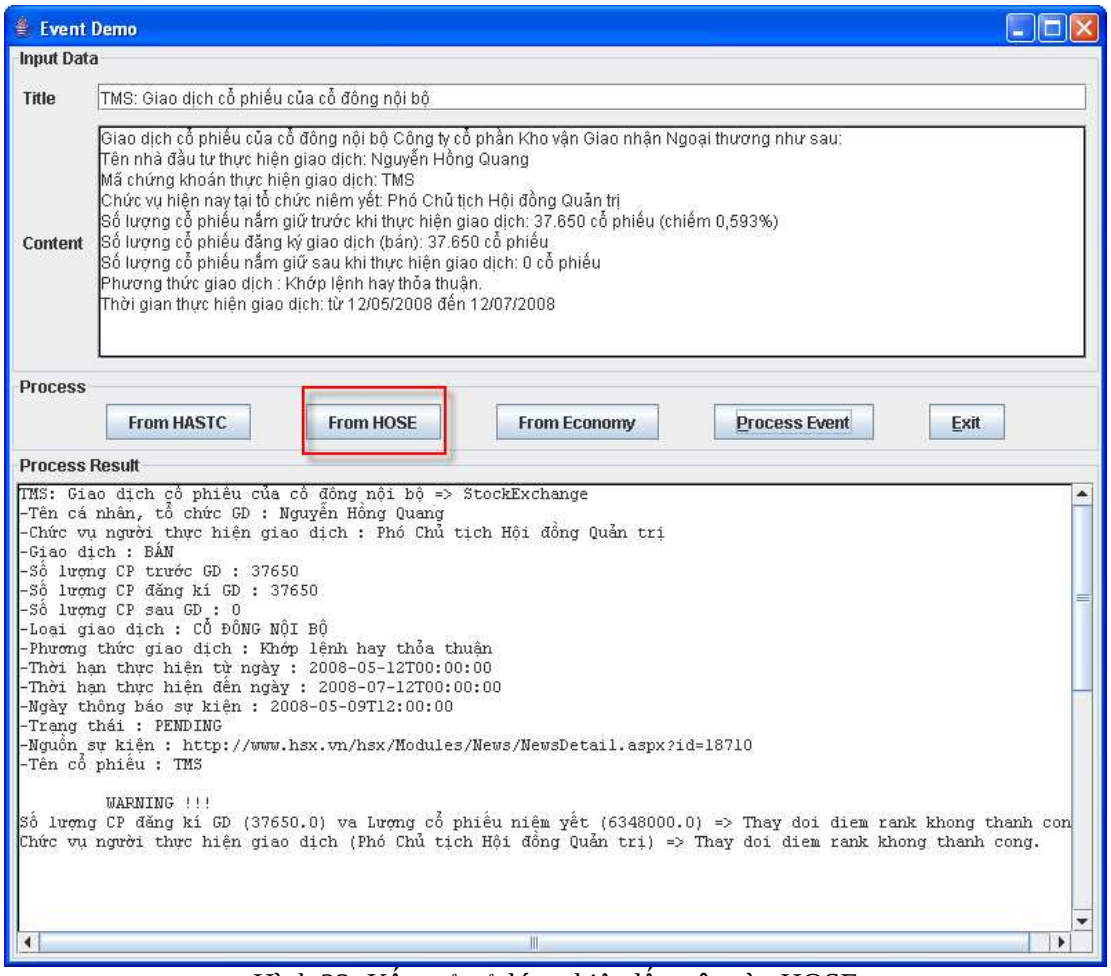


Hình 20: Giao diện module Phân loại sự kiện chứng khoán

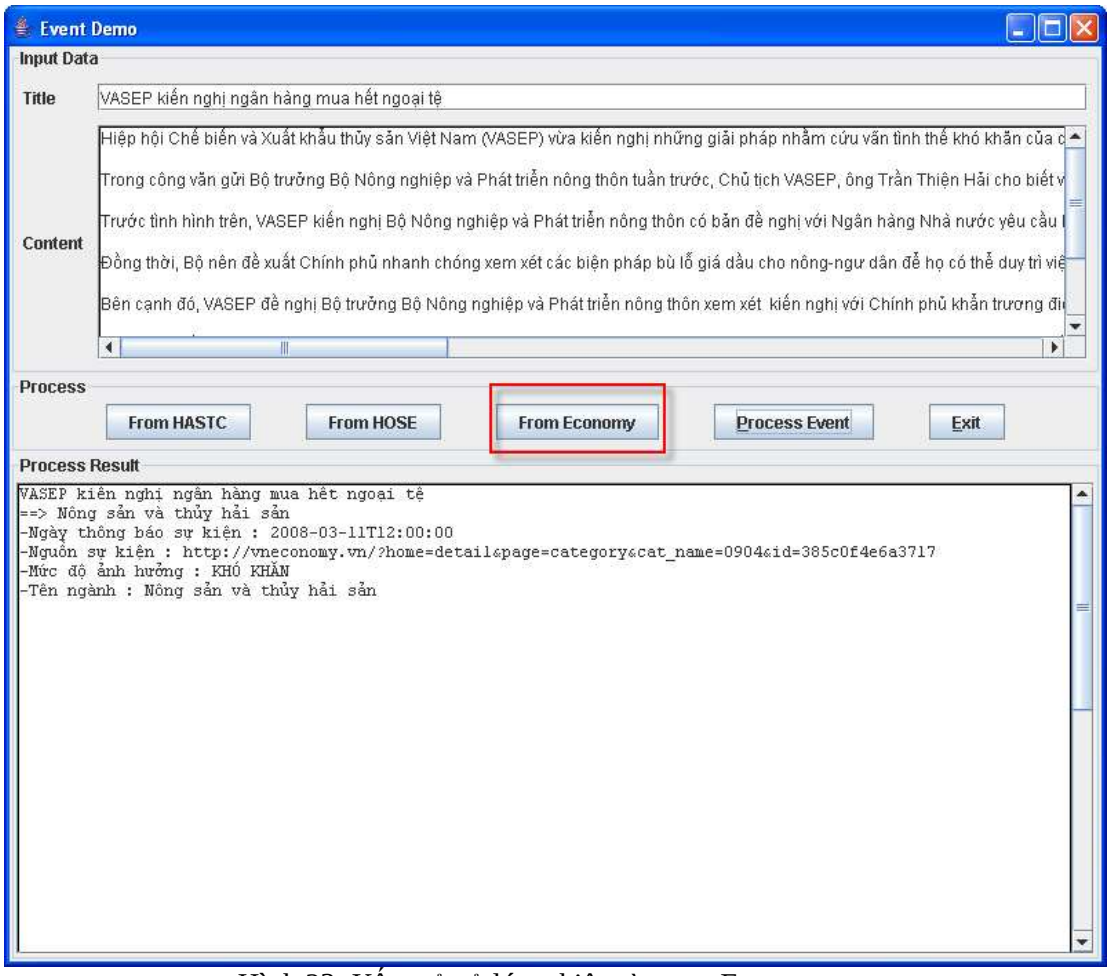
Lấy dữ liệu sự kiện từ Crawler Database và xử lý. Kết quả sẽ cho biết sự kiện này thuộc về loại nào và đưa ra các giá trị thuộc tính tìm được. Nếu không phân loại được bản tin này thuộc về loại nào thì sẽ hiển thị NO EVENT.



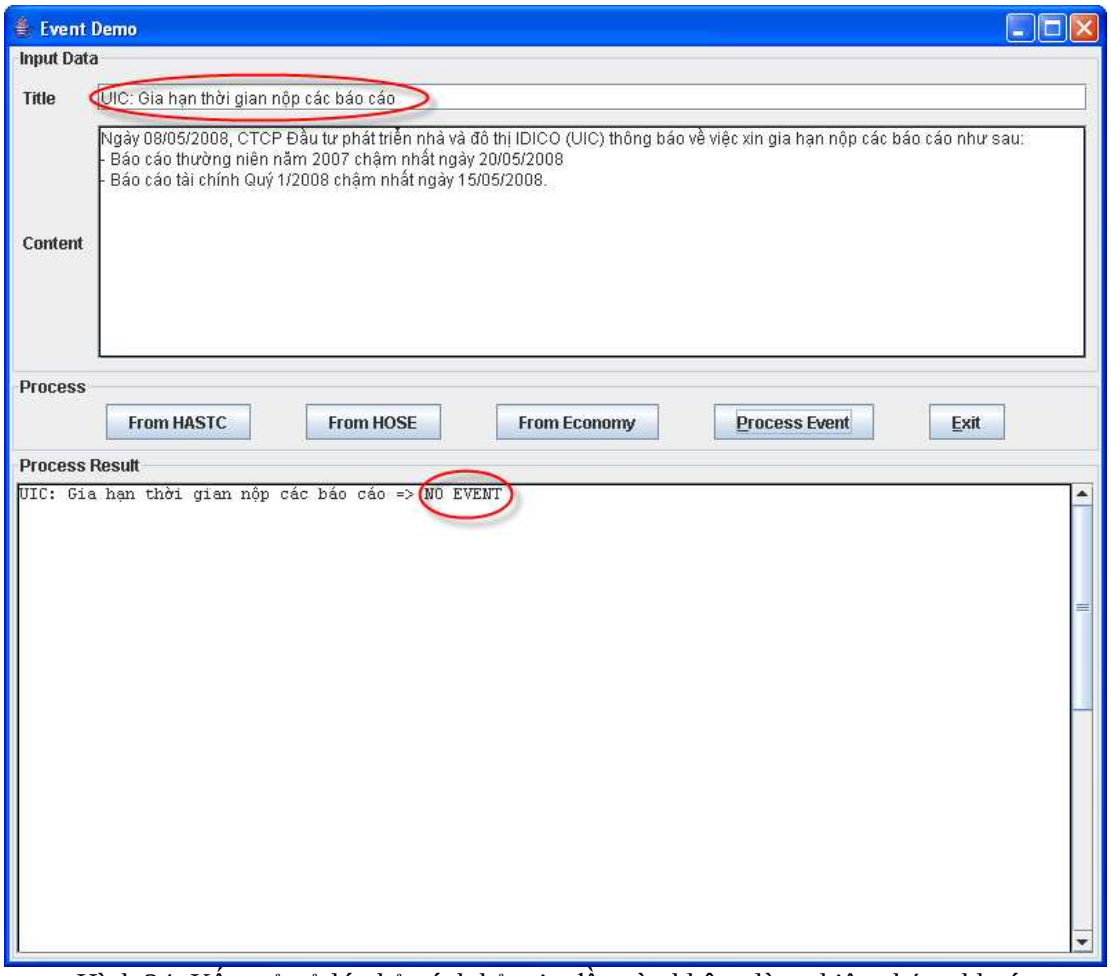
Hình 21: Kết quả xử lý sự kiện lấy trên sàn HASTC



Hình 22: Kết quả xử lý sự kiện lấy trên sàn HOSE



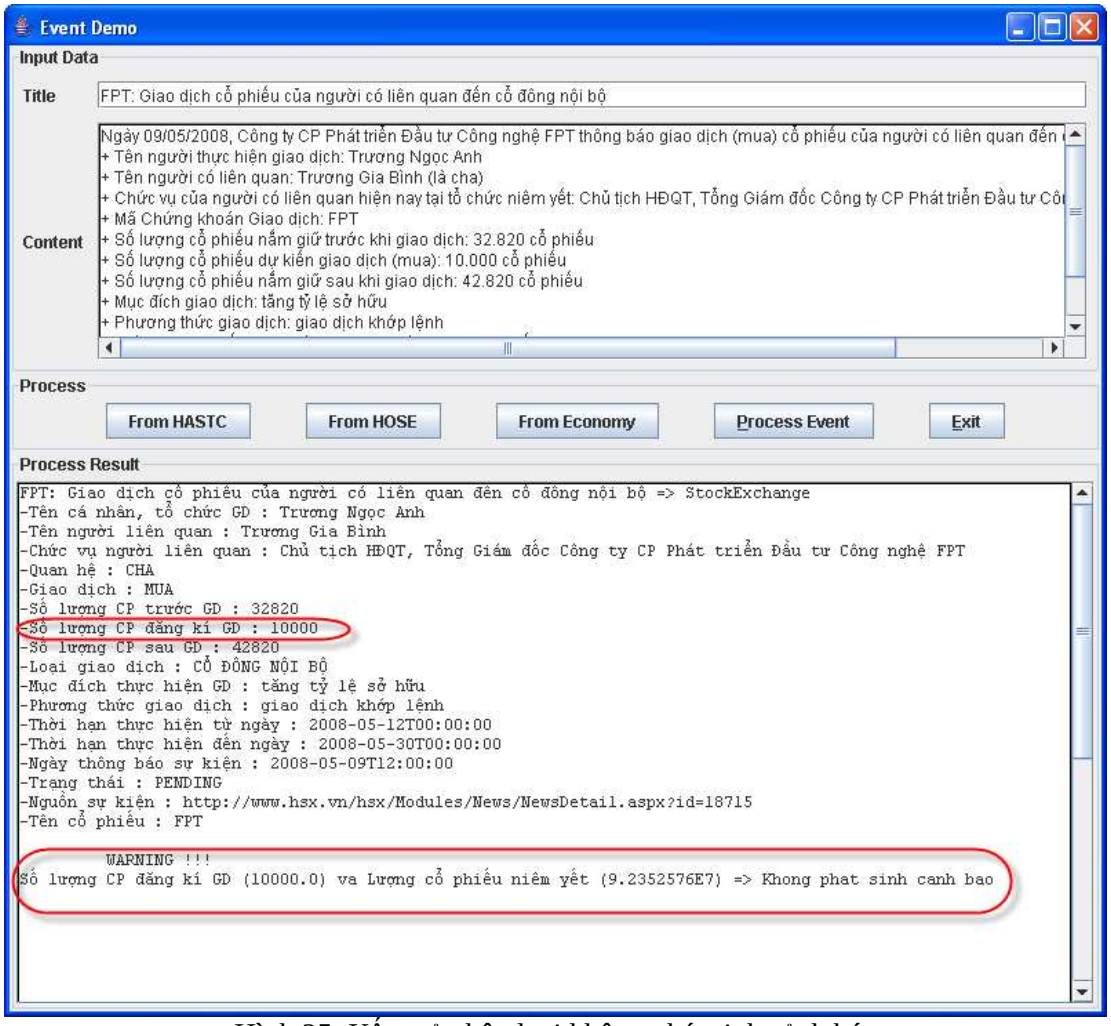
Hình 23: Kết quả xử lý sự kiện từ trang Economy.vn



Hình 24: Kết quả xử lý phản ánh bản tin đầu vào không là sự kiện chứng khoán

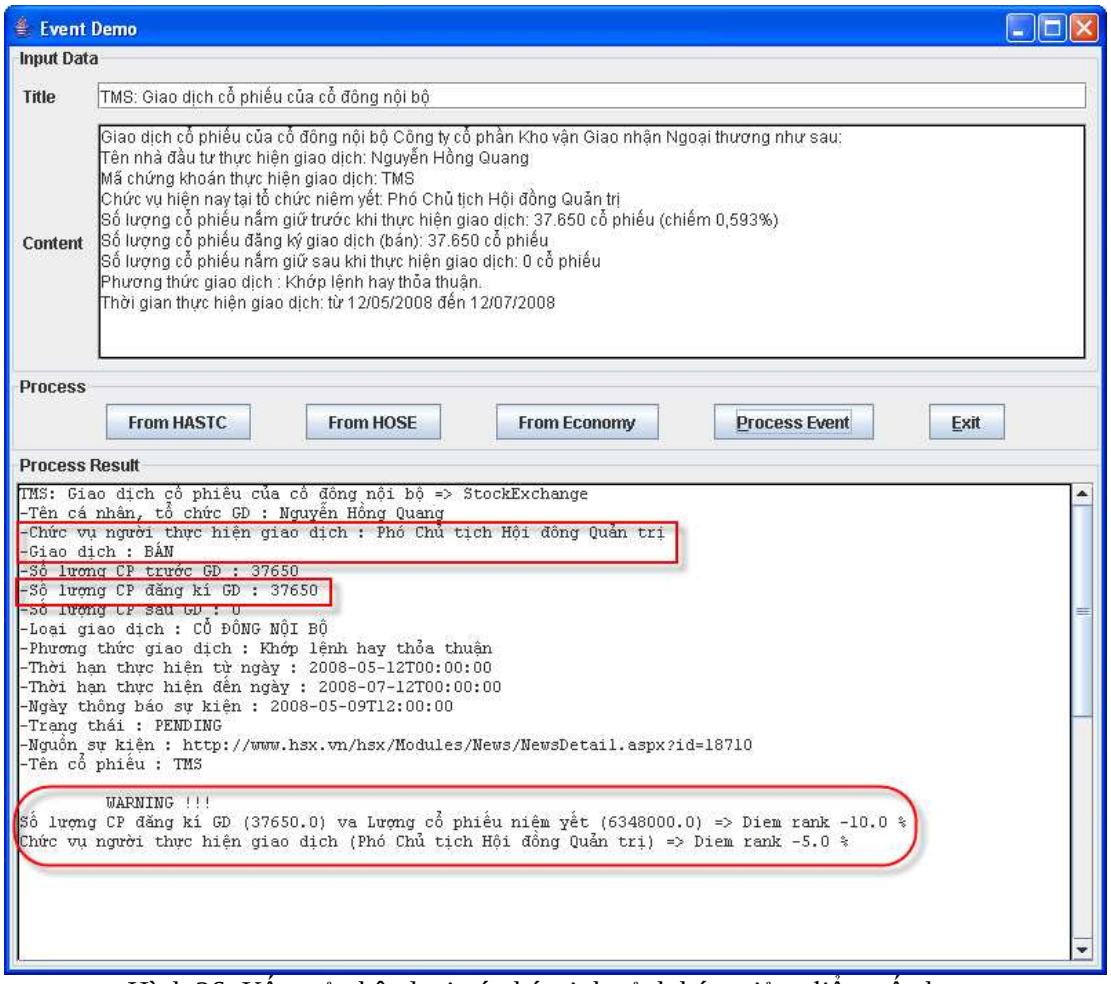
Hình ảnh kết quả phát sinh cảnh báo:

- Trường hợp giá trị thuộc tính có thể phát sinh cảnh báo tồn tại nhưng không thỏa mãn điều kiện thì thông báo: Không phát sinh cảnh báo

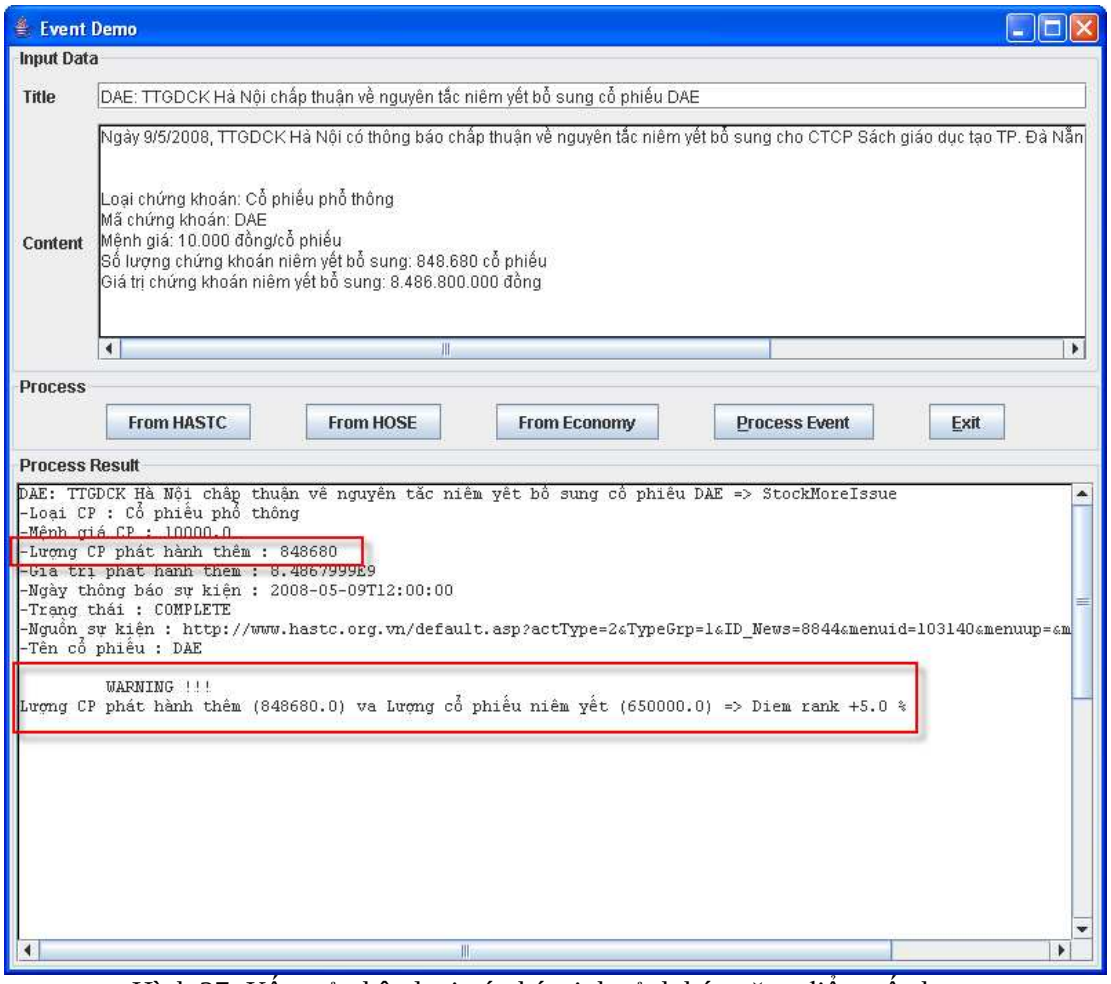


Hình 25: Kết quả phân loại không phát sinh cảnh báo

- Trường hợp giá trị thuộc tính cảnh báo thỏa mãn điều kiện phát sinh cảnh báo và thay đổi điểm xếp hạng => đưa ra cảnh báo và điểm xếp hạng thay đổi



Hình 26: Kết quả phân loại có phát sinh cảnh báo, giảm điểm xếp hạng



Hình 27: Kết quả phân loại có phát sinh cảnh báo, tăng điểm xếp hạng

7.4. Đánh giá kết quả và khả năng phát triển

7.4.1 Kết quả thực hiện được đánh giá như sau

- Việc phân loại các bản tin thuộc về một trong hai sự kiện StockExchange và StockMoreIssue khá chính xác.
- Khi đã phân loại thành một trong hai sự kiện trên thì việc xác định giá trị thuộc tính đạt độ chính xác khá cao (>80%), đa phần các thuộc tính quan trọng đều tìm được đúng giá trị.
- Khi đã tìm được chính xác giá trị các thuộc tính thì việc phát sinh cảnh báo và thay đổi điểm xếp hạng đã được thực hiện tốt, chỉ bị sai sót khi có giá trị của cả cpDaGD và cpDangKiGD.
- Phân loại thành các ngành mới chỉ dừng lại ở ý tưởng, việc xác định các thuật ngữ dựa trên xem xét một số bản tin nên chưa chính xác, trọng số cũng chưa chuẩn. Tuy nhiên với dữ liệu hiện có thì cũng đã thực hiện phân loại tương đối chính xác. Ngành hay bị sai là 'Ngân hàng' do công thức so sánh hai vector đặc tính và vector khái niệm đang sử dụng tìm tích vô hướng lớn nhất (tức ngưỡng giới hạn là 0).

7.4.2 Hướng cải thiện độ chính xác của module

- Xác định các thuật ngữ và trọng số hợp lý hơn, có thể sử dụng tri thức chuyên gia.

- So sánh giữa vector khái niệm và vector đặc tính nên sử dụng ngưỡng.
- Một dấu hiệu để phân loại bản tin thuộc về ngành là bản tin nói về một công ty kinh doanh trong ngành này hay nói về một sản phẩm đặc trưng của ngành.

7.4.3 Khả năng phát triển trong tương lai

- Xây dựng module trở thành một dịch vụ phân tích ngữ nghĩa của các bản tin và cung cấp thông tin cảnh báo cho người sử dụng.
- Mở rộng khả năng xử lý được một số lượng lớn tin tức về chứng khoán trên mạng, trở thành dịch vụ phân loại tin tức tự động và tìm kiếm tin chứng khoán thông minh.
- Tích hợp vào một hệ thống phân tích tin tức kinh tế, tài chính chung.

Phụ lục

[Cấu trúc chi tiết Ontology biểu đồ giá và Ontology sự kiện]

Tài liệu tham khảo