

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
KHOA KHOA HỌC MÁY TÍNH**

**LÊ THỊ TUYẾT MAI  
NGUYỄN MINH PHÚC**

**KHÓA LUẬN TỐT NGHIỆP  
THIẾT KẾ, XÂY DỰNG  
HỆ THỐNG HỖ TRỢ KHUYẾN NÔNG TRÊN CÂY LÚA**

**CỦ NHÂN NGÀNH KHOA HỌC MÁY TÍNH**

**TP. HỒ CHÍ MINH, 2017**

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
KHOA KHOA HỌC MÁY TÍNH**

**LÊ THỊ TUYẾT MAI – 13520489  
NGUYỄN MINH PHÚC – 13520638**

**KHÓA LUẬN TỐT NGHIỆP  
THIẾT KẾ, XÂY DỰNG  
HỆ THỐNG HỖ TRỢ KHUYẾN NÔNG TRÊN CÂY LÚA**

**CỬ NHÂN NGÀNH KHOA HỌC MÁY TÍNH**

**GIẢNG VIÊN HƯỚNG DẪN  
HUỲNH THỊ THANH THƯƠNG**

**TP. HỒ CHÍ MINH, 2017**

## **DANH SÁCH HỘI ĐỒNG BẢO VỆ KHÓA LUẬN**

Hội đồng chấm khóa luận tốt nghiệp, thành lập theo Quyết định số .....  
ngày ..... của Hiệu trưởng Trường Đại học Công nghệ Thông tin.

1. PGS. TS. Đỗ Văn Nhơn – Chủ tịch.
2. ThS. Phạm Nguyễn Trường An – Thư ký.
3. ThS. Hồ Long Vân – Ủy viên.

TRƯỜNG ĐẠI HỌC  
CÔNG NGHỆ THÔNG TIN

Độc Lập - Tự Do - Hạnh Phúc

TP. HCM, ngày 26 tháng 7 năm 2017

**NHẬN XÉT KHÓA LUẬN TỐT NGHIỆP  
(CỦA CÁN BỘ HƯỚNG DẪN)**

**Tên khóa luận:**

Thiết kế, xây dựng hệ thống hỗ trợ khuyến nông trên cây lúa

**Nhóm SV thực hiện:**

Lê Thị Tuyết Mai: 13520489

Nguyễn Minh Phúc: 13520638

**Cán bộ hướng dẫn:**

Ths. Huỳnh Thị Thanh Thương

**Đánh giá Khóa luận**

1. Về cuốn báo cáo:

Số trang	136	Số chương	8
Số bảng số liệu	3	Số hình vẽ	41
Số tài liệu tham khảo	24	Sản phẩm	1

Một số nhận xét về hình thức cuốn báo cáo:

- Còn sai nhiều lỗi chính tả.
- Một số câu văn diễn đạt chưa mạch lạc, chưa rõ ý.

2. Về nội dung nghiên cứu:

Đề tài thuộc lĩnh vực nghiên cứu về “*Hệ chuyên gia và hệ tìm kiếm thông tin/tìm kiếm tài liệu*”. Qua quá trình thực hiện đề tài một cách liên tục và nghiêm túc, sinh viên đã đạt được thành quả đáng khích lệ. Bên cạnh việc nghiên cứu xây dựng ứng dụng, đề tài còn có những đóng góp về mặt khoa học. Đề có thể thiết kế và xây dựng một hệ thống cho kết quả chấp nhận

được thì sinh viên phải giải quyết rất nhiều vấn đề kỹ thuật, lập trình nhiều module xử lý khác nhau, cụ thể là:

- Đối với vấn đề quản lý tri thức của lĩnh vực, suy diễn giải quyết những vấn đề dựa trên tri thức: Thực hiện mô hình hóa tri thức của lĩnh vực; Nghiên cứu cách thức tổ chức CSTT trên máy tính và thiết kế các xử lý cơ bản trên CSTT; Nghiên cứu các chiến lược suy diễn tự động bao gồm suy diễn tiến, lùi, kết hợp thêm việc sử dụng các heuristic để có thể nhanh chóng tìm ra lời giải, suy luận với thông tin không chắc chắn dựa trên kỹ thuật logic mờ và đo lường độ tin cậy của phép suy luận thông qua hệ số chấn chắc CF của sự kiện, luật, ...
- Nghiên cứu các chiến lược tìm kiếm trên CSTT, tìm kiếm trên kho tin bài. Giới hạn bài toán: Tìm kiếm theo từ khóa trên kho tin bài, có xử lý ngữ nghĩa ở mức độ đơn giản bằng cách khai thác cấu trúc thông tin của thực thể và quan hệ giữa các thuật ngữ, tìm kiếm theo thực thể có tên.
- Nghiên cứu xây dựng các công cụ hỗ trợ: làm giàu ontology, thu thập và lưu trữ tin bài từ trang báo điện tử, ...
- Khảo sát đánh giá một số công cụ, công nghệ có liên quan, chọn lựa công cụ, công nghệ có thể hỗ trợ hoặc áp dụng được.
- Chuẩn bị dữ liệu thử nghiệm cho hệ thống: thu thập tin bài từ các trang báo và tổ chức lưu trữ; xây dựng kho dữ liệu ontology trên miền tri thức trồng lúa ; xây dựng các bộ dữ liệu thử nghiệm cho 2 phân hệ.

### 3. Về chương trình ứng dụng:

Hệ thống có 2 nhóm chức năng chính sau:

Phân hệ 1: Phân hệ tư vấn, chẩn đoán

- Hệ thống có một CSTT lưu trữ tri thức của lĩnh vực phục vụ cho việc giải quyết các yêu cầu hay vấn đề ứng dụng được đặt ra, hỗ trợ quản lý và xử lý cơ bản trên CSTT.
- Hệ thống sử dụng kinh nghiệm và tri thức của chuyên gia trong một lĩnh vực hẹp để cho ra kết quả dưới dạng chẩn đoán, tư vấn; Có khả năng suy luận, suy diễn linh hoạt nhờ sử dụng tri thức và các phương pháp suy diễn dựa trên tri thức.

- Hệ thống có tính tương tác thể hiện ở việc giao tiếp tương tác với người dùng để hướng dẫn và lấy thêm thông tin cần thiết cho quá trình suy luận để đi đến kết quả hay lời giải mong muốn.

#### Phân hệ 2: Tra cứu tìm kiếm

- Hỗ trợ thu thập và quản lý tin bài: cho phép thu thập tin bài từ các trang báo và quản lý lưu trữ kho tin bài; cung cấp một trang tin tổng hợp cho phép người sử dụng có thể xem danh sách các tin bài thu thập và đọc trực tiếp các tin bài.
- Hỗ trợ tìm kiếm: hệ thống cho phép tra cứu, tìm kiếm theo nhiều cách khác nhau: tìm kiếm trên kho tin bài và trên cơ sở tri thức.

#### 4. Về thái độ làm việc của sinh viên:

Kết quả đạt được chứng tỏ sinh viên nắm vững kiến thức, nắm vững vấn đề, giải pháp, có khả năng tham gia nghiên cứu triển khai các đề tài, dự án. Ngoài ra trong quá trình thực hiện đề tài sinh viên cũng làm việc rất nghiêm túc, thường xuyên liên lạc và trao đổi với giảng viên hướng dẫn, có tinh thần học hỏi cao và chủ động trong việc thực hiện đề tài.

#### **Đánh giá chung:**

Với các kết quả đạt được, luận văn đáp ứng được rất tốt các yêu cầu đối với một khóa luận ngành Khoa học Máy tính.

#### **Điểm đề nghị:** .

**Người nhận xét**

**(Ký tên và ghi rõ họ tên)**

**TRƯỜNG ĐẠI HỌC  
CÔNG NGHỆ THÔNG TIN**

**Độc Lập - Tự Do - Hạnh Phúc**

TP. HCM, ngày 26 tháng 7 năm 2017

**NHẬN XÉT KHÓA LUẬN TỐT NGHIỆP  
(CỦA CÁN BỘ PHẢN BIỆN)**

**Tên khóa luận:**

Thiết kế, xây dựng hệ thống hỗ trợ khuyến nông trên cây lúa

**Nhóm SV thực hiện:**

Lê Thị Tuyết Mai: 13520489

Nguyễn Minh Phúc: 13520638

**Cán bộ phản biện:**

Ths. Hồ Long Vân

**Đánh giá Khóa luận**

5. Về cuốn báo cáo:

Số trang	136	Số chương	8
Số bảng số liệu	3	Số hình vẽ	41
Số tài liệu tham khảo	24	Sản phẩm	1

Một số nhận xét về hình thức cuốn báo cáo:

- Báo cáo đúng bô cục của 1 luận văn đại học.
- Báo cáo trình bày chi tiết các phần nội dung luận văn.

6. Về nội dung nghiên cứu:

- Đề tài có ý nghĩa thực tiễn cao.
- Sinh viên dành nhiều thời gian nghiên cứu về lĩnh vực biểu diễn tri thức, tìm hiểu chi tiết tri thức về cây lúa nước cũng như xây dựng Ontology cho tri thức của lĩnh vực này, bên cạnh đó cũng tìm hiểu các công cụ (như Protégé) để xây dựng Ontology.

- Có nghiên cứu về lĩnh vực truy xuất thông tin và sử dụng mô hình Vector Space Model trong việc tìm kiếm thông tin.
- Hiểu được các thành phần cần có trong hệ thống và đã khái quát hóa thành các mô hình.

7. Về chương trình ứng dụng:

Xây dựng được ứng dụng web hỗ trợ khuyến nông trên cây lúa với các chức năng chính như: chẩn đoán bệnh, tra cứu thông tin...

8. Về thái độ làm việc của sinh viên:

Sinh viên tích cực, ham học hỏi khi thực hiện đề tài.

**Danh giá chung:** Tốt

**Điểm từng sinh viên:**

Lê Thị Tuyết Mai: **9/10**

Nguyễn Minh Phúc: **9/10**

**Người nhận xét**

**(Ký tên và ghi rõ họ tên)**

## LỜI CẢM ƠN

Chúng em xin được gửi đến Ths Huỳnh Thị Thanh Thương lời cảm ơn chân thành nhất, nhờ sự dùu dắt, hướng dẫn tận tình và định hướng nghiên cứu của cô trong suốt thời gian qua cũng như tài liệu khoa học quý báu mà cô cung cấp đã giúp chúng em hoàn thành khóa luận này.

Chúng em cũng xin được bày tỏ lòng biết ơn đến các thầy cô trường Đại Học Công nghệ Thông tin nói chung và các thầy cô, anh chị khoa Khoa học Máy tính nói riêng đã truyền đạt cho chúng em kiến thức và tài liệu nghiên cứu vô cùng quý báu để hình thành nên cơ sở vững chắc giúp chúng em thực hiện khóa luận này.

Chúng em cũng xin cảm ơn gia đình bạn bè và tập thể lớp KHMT2013 những người đã khích lệ và động viên chúng em nhiều.

Cuối cùng, mặc dù đã cố gắng hoàn thiện chuyên đề với tất cả sự nỗ lực của bản thân nhưng chắc chắn sẽ không tránh khỏi những thiếu sót. Chúng em kính mong nhận được sự thông cảm và chỉ bảo của quý thầy cô và các bạn.

Tp. Hồ Chí Minh, tháng 7 năm 2017

Nhóm thực hiện:

Lê Thị Tuyết Mai 13520489

Nguyễn Minh Phúc 13520638

TP. HCM, ngày 26 tháng 7 năm 2017

## ĐỀ CƯƠNG CHI TIẾT

**TÊN ĐỀ TÀI:** Thiết kế, xây dựng hệ hỗ trợ khuyến nông trên cây lúa

**Cán bộ hướng dẫn:** Ths. Huỳnh Thị Thanh Thương

**Thời gian thực hiện:** Từ ngày 1/12/2016 đến ngày 25/7/2016

**Lê Thị Tuyết Mai 13520489**

**Nguyễn Minh Phúc 13520638**

**Nội dung đề tài:** Đề tài sẽ tập trung nghiên cứu thiết kế và xây dựng một hệ thống ứng dụng trên nền web, tên là “*Hệ hỗ trợ khuyến nông trên cây lúa*”. Hệ thống là nhịp cầu giữa nông dân và các chuyên gia nông nghiệp nhằm hỗ trợ công tác lúa nước. Đây là một dạng hệ chuyên gia có khả năng đưa ra những kết quả ở dạng phán đoán, tư vấn, góp ý về sâu bệnh hại của lúa nguyên nhân, cách phòng tránh sâu bệnh hại, dựa trên kiến thức và kinh nghiệm của chuyên gia được đưa vào hệ thống. Bên cạnh đó cho phép người dùng thực hiện việc tra cứu thông tin trong miền thi thức lúa nước và tìm kiếm những bài viết có liên quan đến nhu cầu thông tin của người dùng trong miền tri thức đó.

### Kế hoạch thực hiện:

- Nghiên cứu khảo sát hiện trạng ứng dụng công nghệ thông tin trong nông nghiệp lúa nước, phân tích đánh giá thực trạng và nhu cầu xây dựng ứng dụng.
- Thu thập yêu cầu của người dùng và xác định các vấn đề bài toán cần giải quyết.
- Nghiên cứu phương pháp thiết kế hệ cơ sở tri thức đặc biệt là hệ chuyên gia.

- Thu thập tri thức của miền tri thức lúa nước từ các nguồn: (1) Từ kiến thức và kinh nghiệm của chuyên gia (nông dân, kỹ sư nông nghiệp), (2) Từ sách vở, các văn bản tài liệu chính quy, (3) Từ các báo điện tử, các cổng thông tin và diễn đàn về nông nghiệp tin cậy,... và phân loại sơ bộ tri thức thu thập được.
- Tìm hiểu các phương pháp biểu diễn tri thức và kỹ thuật suy diễn đã có, đặc biệt là tiếp cận Ontology. Trên cơ sở đó tìm cách vận dụng, phối hợp sao cho phù hợp với yêu cầu thực tế của ứng dụng mà đề tài hướng tới.
- Nghiên cứu tri thức thực tế thu thập được, từ đó tìm cách mô hình hóa chúng để có thể đặc tả được trên máy tính sao cho máy tính sử dụng được chúng để giải quyết một số vấn đề đặt ra.
- Tìm hiểu một số ngôn ngữ Ontology cùng với các công cụ hỗ trợ và phát triển Ontology, lựa chọn sử dụng ngôn ngữ OWL và công cụ Protégé cho đề tài.
- Sử dụng công cụ Protégé để làm dữ liệu Ontology dựa trên mô hình tri thức và ngôn ngữ đã đề xuất.
- Nghiên cứu thiết kế các kỹ thuật: tra cứu tìm kiếm trên cơ sở tri thức, suy diễn dựa trên tri thức.
- Nghiên cứu các trang web nông nghiệp về miền tri thức nông nghiệp, phân tích đánh giá cấu trúc của một webpage chứa nội dung nông nghiệp lúa nước.
- Thiết kế xây dựng bộ thu thập dữ liệu từ các trang web về lĩnh vực lúa.
- Thiết kế xây dựng database cho kho dữ liệu lưu trữ các bài viết crawler về.
- Sử dụng các kỹ thuật tìm kiếm để xây dựng bộ tìm kiếm (searching) trên kho tài liệu được lưu trữ.
- Thiết kế hệ thống ứng dụng gồm các mô-đun: (1) Quản lý cơ sở tri thức của ứng dụng; (2) Bộ tra cứu tìm kiếm; (3) Bộ suy diễn.
- Lập trình cài đặt dùng ngôn ngữ C#, Java sử dụng các công cụ hỗ trợ lập trình như: Visual Studio, IntelliJ IDEA,...
- Thực hiện việc thử nghiệm trên một số mẫu dữ liệu cụ thể.

<b>Xác nhận của CBHD</b>  (Ký tên và ghi rõ họ tên)	TP. HCM, ngày....tháng .....năm.....  <b>Sinh viên</b>  (Ký tên và ghi rõ họ tên)
---	---

## MỤC LỤC

Chương 1. TỔNG QUAN VỀ ĐỀ TÀI.....	1
1.1. ĐẶT VẤN ĐỀ .....	1
1.2. MỤC TIÊU NGHIÊN CỨU.....	2
1.3. NỘI DUNG VÀ PHƯƠNG PHÁP THỰC HIỆN.....	4
1.4. Ý NGHĨA CỦA ĐỀ TÀI.....	5
1.5. BỐ CỤC KHÓA LUẬN.....	6
Chương 2. CƠ SỞ LÝ THUYẾT.....	8
2.1. TỔNG QUAN VỀ HỆ CƠ SỞ TRI THỨC VÀ HỆ CHUYÊN GIA .....	8
2.1.1. Giới thiệu về Hệ cơ sở tri thức và Hệ chuyên gia .....	8
2.1.2. Kiến trúc chung của một Hệ cơ sở tri thức và Hệ chuyên gia .....	10
2.1.3. Một số ứng dụng Hệ cơ sở tri thức và Hệ chuyên gia .....	11
2.2. MỘT SỐ PHƯƠNG PHÁP BIỂU DIỄN TRI THỨC .....	12
2.2.1. Một số phương pháp biểu diễn tri thức cơ bản .....	12
2.2.1.1. Phương pháp biểu diễn bằng logic.....	12
2.2.1.2. Phương pháp dạng đồ thị – Mạng ngữ nghĩa và đồ thị khái niệm ..	13
2.2.1.3. Hệ luật dẫn.....	13
2.2.1.4. Frame và Script.....	14
2.2.2. Ontology .....	14
2.2.2.1. Định nghĩa Ontology .....	14
2.2.2.2. Các thành phần của Ontology.....	15
2.2.2.3. Phân loại Ontology.....	16
2.2.2.4. Vai trò và lợi ích sử dụng Ontology .....	17
2.2.2.5. Các ngôn ngữ đặc tả Ontology .....	17

2.3.	MỘT SỐ PHƯƠNG PHÁP SUY DIỄN TRÊN MÁY TÍNH .....	18
2.3.1.	Phương pháp suy diễn tiến – suy diễn lùi .....	18
2.3.1.1.	Phương pháp suy diễn tiến.....	18
2.3.1.2.	Phương pháp suy diễn lùi.....	19
2.3.2.	Phương pháp suy diễn sử dụng mẫu bài toán và bài toán mẫu.....	19
2.3.3.	Lập luận dựa trên tình huống.....	20
2.4.	TRUY HỒI THÔNG TIN.....	20
2.5.	MỘT SỐ KĨ THUẬT KHÁC ĐƯỢC ÁP DỤNG.....	22
2.5.1.	Ngôn ngữ đặc tả EBNF .....	22
2.5.2.	Regular expression.....	23
2.5.3.	BBCode.....	23
2.5.4.	Linked Data .....	24
2.5.5.	Thuật toán Quine-McCluskey và Petrick's method .....	26
Chương 3.	THIẾT KẾ CƠ SỞ TRI THỨC .....	28
3.1.	THU THẬP VÀ PHÂN LOẠI TRI THỨC .....	28
3.1.1.	Miền tri thức và nguồn tri thức thu thập.....	28
3.1.2.	Cách thức thu thập .....	29
3.1.3.	Kết quả thu thập .....	30
3.2.	MÔ HÌNH TRI THỨC CỦA ỦNG DỤNG .....	30
3.2.1.	Tập các khái niệm C .....	31
3.2.2.	Tập quan hệ giữa các lớp Re .....	32
3.2.3.	Tập các quan hệ phân cấp giữa các khái niệm H .....	33
3.2.4.	Tập các thuộc tính của các lớp Attrs.....	33
3.2.5.	Tập các luật của tri thức Rules.....	34

3.2.6.	Tập biến ngôn ngữ môi trường trong miền tri thức lúa懦c Fu .....	35
3.3.	TỔ CHỨC LUU TRỮ TRI THỨC TRÊN MÁY TÍNH.....	36
3.4.	XỬ LÝ TRUY XUẤT THÔNG TIN TRÊN FILE LUU TRỮ .....	45
3.4.1.	Xử lý truy xuất dữ liệu trên file .....	45
3.4.2.	Áp dụng phương pháp Quine – McCluskey và Petrick's Method để tối ưu luật	45
3.5.	QUY TRÌNH VÀ KẾT QUẢ XÂY DỰNG ONTOLOGY CỦA ỦNG DUNG.....	47
3.5.1.	Quy trình xây dựng Ontology .....	47
3.5.2.	Kết quả tổng hợp được sau khi xây dựng Ontology.....	52
Chương 4.	THIẾT KẾ BỘ SUY DIỄN .....	56
4.1.	PHÁT BIỂU VÀ MÔ HÌNH HOÁ BÀI TOÁN CHUẨN ĐOÁN .....	56
4.2.	Ý TƯỞNG GIẢI QUYẾT BÀI TOÁN.....	57
4.3.	THUẬT GIẢI GIẢI QUYẾT VẤN ĐỀ.....	58
4.3.1.1.	Xếp hạng luật.....	58
4.3.1.2.	Xử lý mờ hóa cho các sự kiện là biến ngôn ngữ.....	59
4.3.1.3.	Thuật giải suy diễn tìm kết quả.....	60
4.3.1.4.	Một số thuật giải suy diễn phụ cho bài toán suy diễn .....	62
4.3.1.5.	Ví dụ minh họa thuật giải.....	66
Chương 5.	THIẾT KẾ BỘ TRA CỨU - TÌM KIẾM .....	69
5.1.	TRA CỨU TRÊN CƠ SỞ TRI THỨC – DATA RETRIEVAL.....	69
5.1.1.	Mô hình hóa bài toán.....	69
5.1.1.1.	Bài toán 1: Tìm kiếm định hướng dạng cây .....	70
5.1.1.2.	Bài toán 2: Bài toán tìm kiếm với cụm từ .....	70

5.1.2.	Xây dựng giải pháp cho bài toán tra cứu.....	70
5.2.	TÌM KIẾM TRÊN KHO TIN BÀI – DOCUMENT RETRIEVAL .....	71
5.2.1.	Mô hình hóa bài toán tìm kiếm trên kho tin bài.....	72
5.2.2.	Thu thập lưu trữ kho dữ liệu tin bài .....	74
5.2.2.1.	Khảo sát và thu thập các tin bài .....	74
5.2.2.2.	Lưu trữ thông tin về cấu trúc trang báo điện tử và nội dung tin bài	
	79	
5.2.3.	Xây dựng từ điển .....	81
5.2.4.	Bài toán lập chỉ mục cho các tài liệu dựa trên Ontology ARK .....	83
5.2.5.	Biểu diễn vector thành túi từ .....	85
5.2.6.	Xử lý mở rộng câu truy vấn thành túi từ .....	86
5.2.7.	So khớp túi từ query với túi từ document và sắp xếp trả về kết quả	88
5.3.	VĂN ĐỀ HIỂN THỊ THÔNG TIN TRẢ VỀ CHO NGƯỜI DÙNG .....	89
5.3.1.	Phát biểu bài toán .....	89
5.3.2.	Xây dựng giải pháp.....	89
Chương 6.	ỨNG DỤNG VÀ THỬ NGHIỆM .....	93
6.1.	THIẾT KẾ HỆ THỐNG .....	94
6.1.1.	Kiến trúc của ứng dụng .....	94
6.1.2.	Mục tiêu ứng dụng .....	95
6.1.3.	Yêu cầu và chức năng của hệ thống.....	95
6.1.4.	Chức năng của hệ thống.....	96
6.1.5.	Tổ chức giao diện hệ thống.....	97
6.2.	CÀI ĐẶT VÀ THỬ NGHIỆM.....	106
6.2.1.	Nền tảng công nghệ .....	106

6.2.2. Cài đặt thử nghiệm - Chuẩn đoán sâu bệnh hại .....	106
6.2.2.1. Cách thức thử nghiệm .....	106
6.2.2.2. Kết quả thử nghiệm chuẩn đoán bệnh .....	106
6.2.3. Cài đặt thử nghiệm cho tra cứu trên kho tài liệu .....	111
<b>Chương 7. KẾT LUẬN .....</b>	<b>114</b>
7.1. Kết quả của đề tài.....	114
7.2. Hạn chế của đề tài.....	115
7.3. Hướng phát triển của đề tài .....	115
<b>Chương 8. PHỤ LỤC.....</b>	<b>118</b>
8.1. Phụ lục 1: Mô hình OWL 2 trong Protégé .....	118
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>137</b>

## DANH MỤC HÌNH VẼ

Hình 1: Kiến trúc Hệ CSTT & Hệ chuyên gia.....	10
Hình 2: Kiến trúc tổng quan của hệ tìm kiếm thông tin.....	21
Hình 3: Cấu trúc căn bản của RDF triples .....	24
Hình 4: Ví dụ về Linked Data .....	26
Hình 5: Quá trình thu thập tri thức.....	29
Hình 6: Quy trình xây dựng ontology của ứng dụng .....	48
Hình 7: Phân cấp lớp trong protégé .....	50
Hình 8: Xác định kiểu dữ liệu của thuộc tính .....	51
Hình 9: Tạo Relation trên protégé.....	51
Hình 10: Thể hiện Individual trên Protégé.....	52
Hình 11: Kết quả xây dựng Ontology trên protégé.....	53
Hình 12: Phân cấp lớp trong Ontology ứng dụng .....	53
Hình 13: Ví dụ quan hệ được xây dựng giữa lớp Thuốc với các lớp khác .....	54
Hình 14: Thể hiện của lớp Thuốc Bệnh Hại .....	54
Hình 15: Xây dựng thuộc tính của các lớp.....	55
Hình 16: Bài toán tra cứu trên Ontpology.....	69
Hình 17: Quy trình tổng quát tìm kiếm thông tin trên kho dữ liệu .....	71
Hình 18: Quá trình tìm kiếm trong kho tin bài .....	74
Hình 19: Nội dung tin bài thu thập ở một webpage nông nghiệp .....	77
Hình 20: Cấu trúc source HTML của một bài viết .....	77
Hình 21: Thiết kế cơ sở dữ liệu của kho tin bài .....	79
Hình 22: lưu trữ data trong bảng Articles .....	80
Hình 23: Quy trình xây dựng kho từ điển .....	81
Hình 24: Ma trận tương quan Global Analysis .....	82
Hình 25: Mô tả quy trình bài toán tách chỉ mục .....	84
Hình 26: Biểu diễn vector thành túi từ.....	86

Hình 27: Sơ đồ cấu trúc hệ thống.....	94
Hình 28: Tổ chức giao diện của hệ thống .....	97
Hình 29: Trang chủ của hệ thống .....	98
Hình 30: Tra cứu định hướng theo cây tìm kiếm.....	98
Hình 31: Tra cứu định hướng theo phân cấp khái niệm.....	99
Hình 32: Giao diện chuẩn đoán sâu bệnh hại cho cây lúa .....	99
Hình 33: Kết quả trả về của hệ thống chuẩn đoán .....	101
Hình 34: Giao diện tìm kiếm thông tin trên Ontology và kho tin bài .....	101
Hình 35: Giao diện trả về kết quả tìm kiếm dữ liệu trên kho tin bài .....	102
Hình 36: Giao diện trả về kết quả tìm kiếm trên ontology .....	102
Hình 37: Trả về thông tin tóm tắt của bài viết .....	103
Hình 38: Giao diện thêm trang tin.....	104
Hình 39: Giao diện quản lý trang tin.....	104
Hình 40: Giao diện xóa, sửa trang tin .....	105
Hình 41: Liên hệ với chuyên gia.....	105

## **DANH MỤC TỪ VIẾT TẮT**

BDTT	Biểu diễn tri thức
CSDL	Cơ sở dữ liệu
CSTT	Cơ sở tri thức
DBMS	Hệ quản trị cơ sở dữ liệu – Database management system
DSS	Hệ hỗ trợ ra quyết định – Decision support system
IMS	Hệ quản lý thông tin – Information management system
MIS	Hệ thống thông tin quản lý – Management information system
QAS	Hệ trả lời câu hỏi – Question answering system

## CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

### 1.1. ĐẶT VẤN ĐỀ

Nhờ vào sự phát triển vượt bậc của khoa học công nghệ, ngành nông nghiệp lúa nước tại Việt Nam nói chung và tỉnh Bình Định nói riêng, đã có tiến bộ vượt bậc trong canh tác. Từ sản xuất 1 vụ mùa/năm, nay đã tăng lên 3 vụ/năm, đáp ứng được nhu cầu lương thực trong nước, đưa Việt Nam trở thành nước xuất khẩu gạo đứng thứ 2 trên thế giới. Tuy nhiên, vào những năm gần đây sâu bệnh xuất hiện hoành hành cả 3 vụ Đông Xuân, Hè Thu, Thu Đông, làm cho lúa bị sâu bệnh gây hại với mức độ càng lúc càng nghiêm trọng. Thời tiết biến đổi thất thường cũng làm cho lúa mắc thêm nhiều bệnh khác ảnh hưởng đến hàng trăm ha lúa, giảm năng suất, nhiều nơi phải hủy bỏ. Để tránh sự gây hại của sâu bệnh, ngoài các biện pháp hướng dẫn canh tác của chuyên gia thì việc chủ động tìm kiếm giải pháp hữu hiệu và lâu dài cho người nông dân cần được chú trọng hơn. Thực tiễn sản xuất đang đòi hỏi phải cấp bách nghiên cứu tìm ra phương pháp, trợ giúp người nông dân có công cụ tốt phục vụ sản xuất. Có như vậy mới tạo cho sản xuất lúa bền vững lâu dài, giữ vững an toàn lương thực, đảm bảo xuất khẩu, từng bước nâng cao đời sống người nông dân cả nước nói chung và nông dân Bình Định nói riêng. Trên cơ sở kết quả phân tích thực trạng và nhu cầu phòng chống dịch bệnh trên cây lúa đề tài “*Thiết kế, xây dựng hệ thống hỗ trợ khuyến nông trên cây lúa*” nhằm nghiên cứu tạo ra một kho tri thức về nông nghiệp lúa nước. Với mục đích hỗ trợ bà con nông dân về cách nhìn tổng quan nhất về miềnn tri thức lúa nước, nhằm phòng trị bệnh trên cây lúa, nắm bắt được lịch thời vụ, thời gian xuất hiện bệnh, thời điểm phun thuốc... để giúp quá trình sản xuất hiệu, quả giảm chi phí. Hệ thống được tạo ra giống như một chuyên gia thực thụ giúp người nông dân chuẩn đoán, tư vấn phòng trừ sâu bệnh hại. Tại Việt Nam, đến thời điểm hiện tại nhóm tìm hiểu thì nhóm chỉ tìm được các trang web đáng tin cậy của nông nghiệp như sau:

1. Hiệp hội Lương thực Việt Nam: <http://www.vietfood.org.vn/>
2. Công ty Syngenta: <https://www.syngenta.com.vn/>
3. Sở Nông nghiệp và Phát triển Nông thôn Vĩnh Phúc: <http://nnptnt.vinhphuc.gov.vn/>
4. Bộ Nông nghiệp và Phát triển Nông thôn Việt Nam: <http://www.mard.gov.vn/>

Đây là những trang web tin cậy cung cấp những kiến thức, về tình hình xuất khẩu nhập khẩu lúa gạo ở Việt Nam, hiếm hoi những bài viết nói về sự phát triển cây lúa, sâu bệnh hại và dịch bệnh của cây lúa ở Việt Nam, thông tin về cây lúa được phân bố rời rạc dẫn đến việc tìm kiếm khó khăn. Tìm kiếm một hệ thống chuyên về lĩnh vực lúa, hay ứng dụng đưa ra những chuẩn đoán lời khuyên trên cây lúa thì hiếm thấy. Vì những lý do trên cùng với sự mong muốn đem lại một nơi chia sẻ kinh nghiệm của chuyên gia và các kỹ sư, là cầu nối giữa chuyên gia và nông dân, tạo ra một nơi để có thể trao đổi, tra cứu tìm kiếm thông tin. Mong muốn đề tài còn là nơi để những người nông dân có kinh nghiệm chia sẻ kiến thức, kinh nghiệm trồng lúa và những người nông dân không có kinh nghiệm tra cứu, chuẩn đoán sâu bệnh hại về lúa. Một vấn đề cộng đồng mà đề tài muốn hướng tới là mang “*tri thức nông nghiệp*” đặc biệt là cây lúa, những kinh nghiệm mà người nông dân lâu năm đúc kết để lan rộng đến tất cả nông dân nhằm nâng cao trình độ, giảm bớt sự mất mát cho người nông dân khi canh tác.

## 1.2. MỤC TIÊU NGHIÊN CỨU

### Mục tiêu chung

Đề tài sẽ tập trung nghiên cứu thiết kế và xây dựng một hệ thống ứng dụng trên nền web, tên là “*Hệ hỗ trợ khuyến nông trên cây lúa*”. Hệ thống là nhịp cầu giữa nhà nông dân và các chuyên gia nông nghiệp nhằm hỗ trợ công tác lúa nước. Đây là một dạng hệ chuyên gia có khả năng đưa ra những kết quả ở dạng phán đoán, tư vấn, gợi ý về sâu bệnh hại của lúa nguyên nhân, cách phòng tránh sâu bệnh hại, dựa trên kiến thức và kinh nghiệm của chuyên gia được đưa vào hệ thống. Bên cạnh đó cho phép người dùng thực hiện việc tra cứu thông tin trong miền thi thức lúa nước và tìm kiếm những bài viết có liên quan đến nhu cầu thông tin của người dùng trong miền tri thức đó.

### Mục tiêu cụ thể

Đề tài sẽ tập trung giải quyết các vấn đề cụ thể được đặt ra như sau:

- Về thiết kế cơ sở tri thức

- Xây dựng mô hình biểu diễn tri thức cho miền tri thức lúa nước theo tiếp cận Ontology. Xây dựng mô hình cơ sở tri thức là mã hóa tri thức của một lĩnh vực thành một kho tri thức dùng chung mà máy tính có thể hiểu được bằng cách phân tách khói tri thức lớn này thành các đối tượng tri thức nhỏ hơn và tìm ra mối quan hệ giữa chúng.
- Trên cơ sở mô hình biểu diễn tri thức đã đề xuất, xác lập cách thức lưu trữ cơ sở tri thức cụ thể trên máy tính, đặc biệt dựa trên ngôn ngữ đặc tả Ontology OWL và công cụ hỗ trợ xây dựng phát triển Ontology là Protégé.
- Thiết kế một số xử lý cơ bản trên cơ sở tri thức làm căn cứ để thiết kế bộ suy diễn và bộ tra cứu tìm kiếm trên Ontology và trên kho tin bài, chẳng hạn như hiệu chỉnh cập nhật ontology, truy vấn dữ liệu, định ra và kiểm soát một số ràng buộc dữ liệu và chuẩn hóa luật.

- Về việc thiết kế bộ suy diễn

Xây dựng một số thuật giải suy diễn nhằm giải quyết các bài toán liên quan đến tri thức bao gồm: bài toán chuẩn đoán bệnh trên cây lúa, nguyên nhân gây bệnh, tư vấn cho người dùng phương pháp phòng trừ sâu bệnh hại trên cây lúa....Áp dụng giải pháp được nghiên cứu để xây dựng thử nghiệm một hệ thống ứng dụng hỗ trợ khuyến nông trên cây lúa. Hệ thống này có những chức năng chính như sau: (1) Tổ chức quản lý lưu trữ tri thức của miền tri thức lúa nước; (2) Tra cứu một số thông tin có trong miền tri thức; (3) Thực hiện suy diễn để giải quyết vấn đề chuẩn đoán sâu bệnh hại và nguyên nhân gây bệnh cho cây lúa, tư vấn cách điều trị, phòng ngừa bệnh.

- Về việc tra cứu tìm kiếm

- Thực hiện thu thập quản lý các tin bài liên quan đến miền tri thức lúa nước.
- Xây dựng một kho tin bài để lưu trữ các bài viết liên quan đến miền tri thức lúa nước từ các trang web về lúa.
- Dựa vào câu truy vấn người dùng nhập vào, thực hiện xây dựng một số thuật giải tìm kiếm trên kho tài liệu hoặc trên kho cơ sở tri thức trả về danh sách các tin bài liên quan đến nội dung của câu truy vấn.

Một số nhiệm vụ khác có thể kể đến như:

- Nghiên cứu các phương pháp biểu diễn tri thức lĩnh vực đặc biệt là cách tiếp cận Ontology.
- Nghiên cứu một số kĩ thuật suy diễn tự động trên máy tính như suy diễn tiến lùi, suy diễn với thông tin mờ.
- Nghiên cứu các ngôn ngữ Ontology, các công nghệ, công cụ hỗ trợ xây dựng quản lý phát triển Ontology và lựa chọn áp dụng.
- Nghiên cứu cách thu thập và lưu trữ bài viết một cách hiệu quả.
- Nghiên cứu các phương pháp tìm kiếm thông tin (information retrieval) hiệu quả nhất.

### **1.3. NỘI DUNG VÀ PHƯƠNG PHÁP THỰC HIỆN**

- Nghiên cứu khảo sát hiện trạng ứng dụng công nghệ thông tin trong nông nghiệp lúa nước, phân tích đánh giá thực trạng và nhu cầu xây dựng ứng dụng.
- Thu thập yêu cầu của người dùng và xác định các vấn đề bài toán cần giải quyết.
- Nghiên cứu phương pháp thiết kế hệ cơ sở tri thức đặc biệt là hệ chuyên gia.
- Thu thập tri thức của miền tri thức lúa nước từ các nguồn: (1) Từ kiến thức và kinh nghiệm của chuyên gia (nông dân, kĩ sư nông nghiệp), (2) Từ sách vở, các văn bản tài liệu chính quy, (3) Từ các báo điện tử, các cổng thông tin và diễn đàn về nông nghiệp tin cậy,... và phân loại sơ bộ tri thức thu thập được.
- Tìm hiểu các phương pháp biểu diễn tri thức và kĩ thuật suy diễn đã có, đặc biệt là tiếp cận Ontology. Trên cơ sở đó tìm cách vận dụng, phối hợp sao cho phù hợp với yêu cầu thực tế của ứng dụng mà đề tài hướng tới.
- Nghiên cứu tri thức thực tế thu thập được, từ đó tìm cách mô hình hóa chúng để có thể đặc tả được trên máy tính sao cho máy tính sử dụng được chúng để giải quyết một số vấn đề đặt ra.
- Tìm hiểu một số ngôn ngữ Ontology cùng với các công cụ hỗ trợ và phát triển Ontology, lựa chọn sử dụng ngôn ngữ OWL và công cụ Protégé cho đề tài.

- Sử dụng công cụ Protégé để làm dữ liệu Ontology dựa trên mô hình tri thức và ngôn ngữ đã đề xuất.
- Nghiên cứu thiết kế các kỹ thuật: tra cứu tìm kiếm trên cơ sở tri thức, suy diễn dựa trên tri thức.
- Nghiên cứu các trang web nông nghiệp về miền tri thức nông nghiệp, phân tích đánh giá cấu trúc của một webpage chứa nội dung nông nghiệp lúa nước.
- Thiết kế xây dựng bộ thu thập dữ liệu từ các trang web về lĩnh vực lúa.
- Thiết kế xây dựng database cho kho dữ liệu lưu trữ các bài viết crawler về.
- Sử dụng các kỹ thuật tìm kiếm để xây dựng bộ tìm kiếm (searching) trên kho tài liệu được lưu trữ.
- Thiết kế hệ thống ứng dụng gồm các mô-đun: (1) Quản lý cơ sở tri thức của ứng dụng; (2) Bộ tra cứu tìm kiếm; (3) Bộ suy diễn.
- Lập trình cài đặt dùng ngôn ngữ C#, Java sử dụng các công cụ hỗ trợ lập trình như: Visual Studio, IntelliJ IDEA,...
- Thực hiện việc thử nghiệm trên một số mẫu dữ liệu cụ thể.

#### **1.4. Ý NGHĨA CỦA ĐỀ TÀI**

Việc thay đổi tập quán canh tác của nông dân là công việc hết sức khó khăn. Trong điều kiện sản xuất ngày càng bất lợi như hiện nay, người nông dân cần phải thay đổi, phải tiếp thu nhanh các kiến thức mới và vận dụng vào trong thực tế sản xuất của mình. Ngày nay có nhiều nhà nông lập nghiệp làm giàu từ chính vùng đất mà họ đang sống, nhưng rất ít, và đa số đó là những người có kiến thức nông nghiệp. Vì vậy, tại sao những người nông dân cả đời quanh quẩn với cây lúa nhưng không giàu lên được, có lẽ vì không được cập nhật thông tin đúng thời điểm đúng lúc. Chính vì vậy người nông dân phải trở thành chuyên gia, để tự ứng phó với các điều kiện sản xuất bất lợi, nâng cao năng suất, cải thiện thu nhập thông tin và có khả năng truyền đạt các kiến thức học được cho các nông dân khác cùng áp dụng. Vì vậy nhằm tháo gỡ khó khăn, cùng chia sẻ những kinh nghiệm của các nhà khoa học, các chuyên gia đầu đàn trong lĩnh vực nông nghiệp, nhóm nghiên cứu muốn xây dựng một hệ thống dựa trên những kiến thức được học, để có thể giúp người dân nào tập quen với việc tìm kiếm

thông tin. Xây dựng một hệ thống là nơi có thể giúp người dân tìm hiểu, cập nhật, chia sẻ thông tin,... để người nông dân phản ứng kịp thời với bất kỳ trường hợp nào xuất hiện trên cánh đồng, tránh những thiệt hại không đáng. Hệ thống giúp người sử dụng có cái nhìn tổng quan về miền tri thức lúa nước, cung cấp cho người dùng các tính năng tra cứu, chuẩn đoán, tư vấn, tìm kiếm giúp người dùng tra cứu thông tin, tìm kết quả sâu bệnh hại dựa trên những thông tin người dùng nhập vào, gợi ý người dùng cách phòng bệnh chữa bệnh... Hệ thống là sự kết nối giữa chuyên gia nông nghiệp với người dân, và giữa nông nghiệp với nông dân, cập nhật thông tin cho người dân, hạn chế những ảnh hưởng xấu nhất khi không cập nhật thông tin đúng lúc.

## 1.5. BỘ CỤC KHÓA LUẬN

Khóa luận gồm có 6 chương và có bộ cục như sau:

- Chương 1: Tổng quan về đề tài

Giới thiệu tổng quan về đề tài bao gồm việc đặt vấn đề cần thiết thực hiện đề tài. Nêu cầu cần thiết hiện nay của đề tài đối với người nông dân. Trình bày tóm tắt mục tiêu nghiên cứu, nội dung và phương pháp nghiên cứu, ý nghĩa của đề tài, bộ cục trình bày luận văn.

- Chương 2: Cơ sở lý thuyết

Trình bày về cơ sở lý thuyết của đề tài bao gồm tổng quan về hệ cơ sở tri thức và hệ chuyên gia, giới thiệu chi tiết về hệ cơ sở tri thức và hệ chuyên gia, kiến trúc chung của một hệ cơ sở tri thức và hệ chuyên gia, quy trình xây dựng một hệ cơ sở tri thức và hệ chuyên gia, một số ứng dụng của hệ chuyên gia. Tiếp theo trình bày các phương pháp biểu diễn tri thức như mạng ngữ nghĩa, logic vị từ, hệ luật dẫn, frame, script,... Trình bày những kỹ thuật phương pháp suy diễn trên hệ chuyên gia như suy diễn tiên, suy diễn lùi, suy diễn sử dụng mẫu bài toán và bài toán mẫu, lập luận dựa trên tình huống, lý thuyết về Ontology, thực trạng sử dụng, vai trò và lợi ích của việc sử dụng Ontology, các ứng dụng của Ontology. Cuối cùng là các kỹ thuật khác được sử dụng trong đề tài như thuật toán Quine-McCluskey và Petrick's method để tối ưu luật, ngôn ngữ BBCODE, ngôn ngữ đặc tả EBNF, regular expression và dữ liệu liên kết.

- Chương 3: Thiết kế cơ sở tri thức

Trình bày quy trình cách thức thu thập và phân loại tri thức, từ những tri thức thu thập và phân loại ta tiến hành mô hình hóa cơ sở tri thức. Sau khi thực hiện việc mô hình hóa cơ sở tri thức thì tiến hành lưu trữ tri thức thu thập được trên máy tính, giới thiệu về cách trích xuất thông tin trong kho dữ liệu, và cuối cùng là trình bày quy trình xây dựng Ontology.

- Chương 4: Thiết kế bộ suy diễn

Phát biểu các vấn đề suy diễn tìm ra sâu bệnh hại, cách phòng trừ..., trình bày ý tưởng giải quyết, thiết kế thuật giải suy diễn để giải quyết vấn đề đặt ra.

- Chương 5: Thiết kế các chiến lược tìm kiếm

Xây dựng các chiến lược suy diễn, tìm kiếm thông tin trên Ontology. Xây dựng bộ thu thập dữ liệu từ các trang web nông nghiệp, tiến hành trích xuất đánh chỉ mục cho từng tài liệu, thực hiện xử lý và mở rộng câu truy vấn, so khớp câu truy vấn với tài liệu và xếp hạng trả về danh sách các tài liệu liên quan.

- Chương 6: Cài đặt và thử nghiệm

Trình bày nội dung về các công cụ và ngôn ngữ sử dụng, giới thiệu giao diện ứng dụng, thử nghiệm và đánh giá.

- Chương 7: Kết luận

Rút ra nhận xét và hướng phát triển của đề tài.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

### 2.1. TỔNG QUAN VỀ HỆ CƠ SỞ TRI THỨC VÀ HỆ CHUYÊN GIA

#### 2.1.1. Giới thiệu về Hệ cơ sở tri thức và Hệ chuyên gia

Hệ cơ sở tri thức (hệ CSTT) là một chương trình máy tính được thiết kế để mô hình hóa khả năng giải quyết vấn đề của chuyên gia (con người). Hệ thống dựa trên tri thức cho phép mô hình hóa các tri thức của chuyên gia, dùng tri thức này để giải quyết vấn đề phức tạp thuộc cùng lĩnh vực. Hệ CSTT sẽ giúp ta quản lý tri thức của các miền tri thức khác nhau trên máy tính, giúp giải quyết những vấn đề liên quan đến tri thức, cao cấp hơn là hệ thống có khả năng suy luận hay suy diễn giải quyết vấn đề trừu tượng và phức tạp[1].

Có nhiều loại Hệ CSTT ứng dụng trong các lĩnh vực khác nhau của con người, một số Hệ CSTT phổ biến nhất: Hệ chuyên gia , Hệ hỗ trợ ra quyết định, Hệ điều khiển mờ , Hệ học, Tích hợp Hệ cơ sở tri thức và các Hệ quản trị cơ sở dữ liệu.

#### Hệ chuyên gia

Trong số các loại Hệ CSTT kể trên thì hệ chuyên gia được xem là phổ biến nhất. Theo E. Feigenbaum, hệ chuyên gia là một chương trình máy tính thông minh sử dụng tri thức và các thủ tục suy luận để giải những bài toán tương đối khó khăn đòi hỏi những chuyên gia mới giải được. Nói cách khác hệ chuyên gia có CSTT lưu tri thức chuyên gia bao gồm cả kinh nghiệm của chuyên gia, và có khả năng suy luận giải quyết những vấn đề chuyên môn trong lĩnh vực chuyên sâu của chuyên gia. Hệ chuyên gia thường được thiết kế cho một lĩnh vực ứng dụng cụ thể làm việc như một chuyên gia thực thụ, cung cấp các ý kiến, góp ý, tư vấn dựa trên kiến thức, kinh nghiệm của chuyên gia con người đưa vào hệ thống.

#### Hệ hỗ trợ ra quyết định

Khái niệm Hệ hỗ trợ ra quyết định (Decision Support System – DSS) được đề xuất bởi Michael Scott Morton vào những năm 1970. Hệ hỗ trợ ra quyết định là một hệ thống thuộc Hệ thống thông tin quản lý (Management information system), có nhiệm vụ cung cấp các thông tin hỗ trợ cho việc đề ra quyết định ở cấp chiến lược và chiến thuật trở nên dễ dàng, thuận tiện hơn. DSS dựa trên máy tính trợ giúp một (hoặc một nhóm) người giải quyết các vấn đề bán cấu trúc (semi-structured problem) hoặc không cấu trúc (un-structured problem) –

là những vấn đề không có phương pháp giải quyết chắc chắn, mà phần lớn phải dựa vào kinh nghiệm và phán đoán của chuyên gia. Ví dụ: Bố nhiệm nhân sự cho một chức vụ; chọn dự án để đầu tư; quyết định đầu tư cho một dự án; quyết định tiến hành một đợt khuyến mãi;... (Trích dẫn theo bài viết của Nguyễn Anh Hào, giảng viên Khoa Kinh tế – Quản lý, Đại học Bách khoa Hà Nội). Hệ hỗ trợ ra quyết định cũng được xem là một dạng hệ chuyên gia hướng tới khả năng tư vấn và hỗ trợ cho nhà quản lý đi tới những quyết định quan trọng.

### **Hệ điều khiển mờ**

Hệ điều khiển mờ được xây dựng dựa trên lý thuyết về logic mờ được đề xuất bởi giáo sư Lotfi A. Zadeh năm 1965. Lý thuyết này giải quyết các bài toán rất gần với cách tư duy của con người [20]. Tới nay, lý thuyết logic mờ đã phát triển rất mạnh mẽ và đạt được một số thành tựu như năm 1970 tại trường Mary Queen, Ebrahim Mamdani dùng fuzzy-logic điều khiển máy hơi nước; năm 1983 tại Nhật, Fuji Electronic ứng dụng điều khiển máy xử lý nước; năm 1987, Hitachi áp dụng vào điều khiển tàu điện ngầm. Một số ứng dụng sử dụng logic mờ trong đời sống như: các hệ thống con của các phương tiện giao thông ABS, quản lý hành trình (ví dụ Tokyo monorail), máy điều hòa nhiệt độ, camera, xử lý ảnh số (Digital image processing), phát hiện biên (edge detection),...

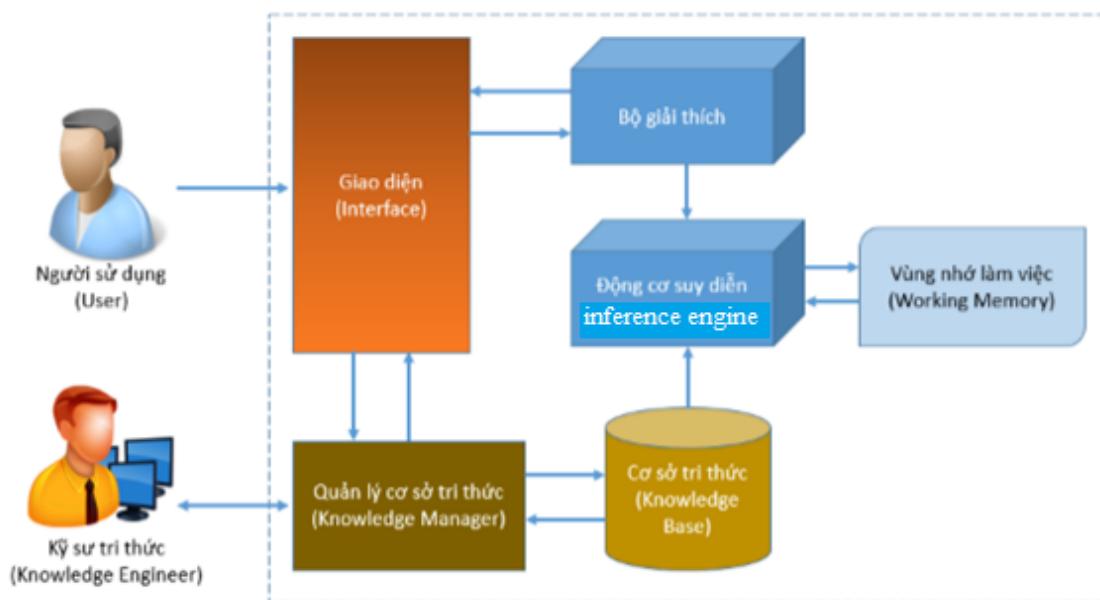
### **Hệ học**

Trong nhiều tình huống, sẽ không có sẵn tri thức mà kỹ sư tri thức cần thu nhận tri thức từ chuyên gia lĩnh vực, và cần biết các luật mô tả lĩnh vực cụ thể, bài toán không được biểu diễn tường minh theo luật, sự kiện hay các quan hệ, vì vậy bản thân hệ thống cần phải tiếp thu tri thức. Có hai tiếp cận cho hệ thống học: (1) Học từ ký hiệu: bao gồm việc hình thức hóa, sửa chữa các luật tường minh, sự kiện và các quan hệ; (2) Học từ dữ liệu số: được áp dụng cho những hệ thống được mô hình dưới dạng số liên quan đến các kỹ thuật nhằm tối ưu các tham số. Học theo dạng số bao gồm mạng nơron nhân tạo, thuật giải di truyền, bài toán tối ưu truyền thống. Các kỹ thuật học theo số không tạo ra CSTT tường minh. Có rất nhiều hình thức học như: học vẹt, học bằng cách chỉ dẫn, học bằng qui nạp, học bằng tương tự, học dựa trên giải thích, học dựa trên tình huống, khám phá hay học không dám sát.[21]

### **Tích hợp Hệ cơ sở tri thức và các Hệ quản trị cơ sở dữ liệu**

Có thể áp dụng cơ chế CSTT và cơ chế lập luận để nâng cao các khả năng cung cấp thông tin của các cơ sở dữ liệu (CSDL) hiện có. Một ví dụ tiêu biểu là trong CSDL về hành trình của các con tàu xuất phát từ cảng. Dựa trên các thông tin lưu trữ trong CSDL về giờ xuất phát và các quy luật hải hành có thể rút ra vị trí hiện tại của con tàu. Rõ ràng điều này không thể làm được với các câu lệnh truy vấn SQL truyền thống. Tuy vậy khi đưa các luật suy diễn vào CSDL, có thể dễ dàng tạo sinh thêm thông tin dựa trên các sự kiện cung cấp, các dữ liệu đang được lưu trữ trong CSDL và các luật, cơ chế suy diễn trong CSTT.[1]

### 2.1.2. Kiến trúc chung của một Hệ cơ sở tri thức và Hệ chuyên gia



Hình 1: Kiến trúc Hệ CSTT & Hệ chuyên gia

Về mặt kiến trúc thì Hệ CSTT và Hệ chuyên gia gồm các thành phần cơ bản như: cơ sở tri thức (CSTT), động cơ suy diễn, giao diện, bộ giải thích, vùng nhớ làm việc, bộ quản lý tri thức, mô-đun học (self-learning),... Trong các thành phần kể trên thì CSTT và động cơ suy diễn là 2 thành phần trung tâm của hệ thống. Việc thiết kế CSTT đòi hỏi nhà thiết kế phải nắm vững các phương pháp biểu diễn tri thức để vận dụng, cải biến cũng như nghiên cứu và phát triển các phương pháp đó. Sau đây là giới thiệu chức năng các thành phần kiến trúc Hệ CSTT và Hệ chuyên gia:

- Cơ sở tri thức: chứa tri thức chuyên sâu về lĩnh vực của chuyên gia, chứa kiến thức cần thiết dùng để giải quyết vấn đề. CSTT thường bao gồm: các khái niệm, các quan hệ, các sự kiện, các luật, ...
- Động cơ suy diễn: thực hiện các chiến lược suy diễn giải quyết vấn đề dựa trên việc sử dụng tri thức trong CSTT. Động cơ hoạt động trên thông tin về vấn đề đang xét, so sánh với tri thức lưu trong CSTT rồi rút ra kết luận.
- Bộ phận giải thích: cho phép chương trình giải thích quá trình suy luận giải quyết vấn đề được đặt ra của bài toán. Xuất ra các lời giải biện minh cho các kết luận của hệ thống (trả lời cho câu hỏi How); giải thích vì sao hệ cần dữ liệu đó (trả lời câu hỏi Why).
- Giao diện: thực hiện giao tiếp giữa Hệ chuyên gia và người dùng. Nhận các thông tin từ người dùng (các câu hỏi, các yêu cầu về lĩnh vực) và đưa ra các câu trả lời, các lời khuyên, các giải thích về lĩnh vực đó.
- Vùng nhớ làm việc: chứa các sự kiện của vấn đề đang xét.

### **2.1.3. Một số ứng dụng Hệ cơ sở tri thức và Hệ chuyên gia**

- Các hệ chuẩn đoán và dự báo như chuẩn đoán y khoa, chuẩn đoán hỏng hóc máy tính, dự báo thời tiết, dự báo thị trường chứng khoán... Hệ chuẩn đoán y khoa là dạng hệ chuyên gia có khả năng thực hiện việc chuẩn đoán bệnh như một người bác sĩ. Tiêu biểu nhất là Hệ chuyên gia MYCIN ra đời trong thời gian 1973-1978, và thế hệ tiếp theo là EMYCIN (Essential MYCIN).
- Các hệ thống quản lý tài liệu văn bản theo ngữ nghĩa: hệ thống này quản lý kho tài liệu văn bản điện tử trên máy tính và có khả năng hỗ trợ thực hiện các xử lý liên quan đến nội dung của tài liệu. Một số ví dụ về loại hệ thống này là hệ quản lý kho tài nguyên học tập về CNTT, Hệ hỗ trợ truy tìm các tin bài trên báo điện tử liên quan đến một chủ đề nào đó, hệ hỗ trợ kiểm tra việc ban hành văn bản quy phạm pháp luật... [14]
- Các hệ hỗ trợ tư vấn và trợ giúp quyết định, ví dụ hệ hỗ trợ tư vấn về pháp luật, tư vấn hướng nghiệp, tư vấn sức khỏe, tư vấn bữa ăn dinh dưỡng, tư vấn địa điểm du lịch, tư vấn mua máy tính...

- Hệ hỗ trợ kiểm tra đánh giá kiến thức.
- Hệ hỗ trợ giải các bài toán trên CSTT của các miền ứng dụng nhất định như hệ giải bài toán về hình học phẳng, hình học giải tích, đại số tuyến tính,...

## 2.2. MỘT SỐ PHƯƠNG PHÁP BIỂU DIỄN TRI THỨC

### 2.2.1. Một số phương pháp biểu diễn tri thức cơ bản

#### 2.2.1.1. Phương pháp biểu diễn bằng logic

##### **Biểu diễn tri thức bằng logic mệnh đề**

Mô hình tri thức logic mệnh đề:

(P, R)

Trong đó:

P: Tập các mệnh đề (mỗi mệnh đề biểu diễn bởi 1 ký hiệu);

R: Tập các luật, các sự kiện biểu diễn bởi các biểu thức logic và các quy tắc logic.

Trong mô hình này, các kiến thức được biểu diễn dưới các dạng mệnh đề. Giá trị của mệnh đề chỉ có thể hoặc là đúng hoặc là sai, đồng thời giá trị của mệnh đề không chỉ phụ thuộc vào bản thân mệnh đề đó, có những mệnh đề mà giá trị của nó luôn đúng hoặc sai bất chấp thời gian nhưng cũng có những mệnh đề mà giá trị của nó lại phụ thuộc vào thời gian, không gian và nhiều yếu tố khách quan khác. Bên cạnh đó, trong logic mệnh đề ta có thể sử dụng các thuật giải Vương Hạo và Robinson để kiểm tra giá trị của các mệnh đề tổng quát. Logic mệnh đề cho phép ta biểu diễn các sự kiện. Mỗi ký hiệu trong logic mệnh đề được minh họa như là một sự kiện trong thế giới hiện thực, sử dụng các kết nối logic ta có thể tạo ra các câu phức hợp biểu diễn các sự kiện mang ý nghĩa phức tạp hơn. Như vậy, khả năng biểu diễn của logic mệnh đề chỉ giới hạn trong phạm vi thế giới các sự kiện đơn giản với chân trị đúng hoặc sai. Khi biểu diễn tri thức bằng mệnh đề, chúng ta không thể can thiệp vào cấu trúc của một mệnh đề. Điều này làm hạn chế rất nhiều thao tác suy luận. [1]

##### **Biểu diễn tri thức bằng logic vị từ**

Biểu diễn tri thức bằng mệnh đề gấp phải một trớ ngai cơ bản là ta không thể can thiệp vào cấu trúc của một mệnh đề, hay nói một cách khác đây là mệnh đề không có cấu trúc. Điều

này làm hạn chế rất nhiều thao tác suy luận. Do đó, người ta đã đưa vào khái niệm vị từ và lượng từ ( $\forall$  - với mọi,  $\exists$  - tồn tại) để tăng cường tính cấu trúc của một mệnh đề. Trong logic vị từ, một mệnh đề được cấu tạo bởi hai thành phần là các đối tượng tri thức và mối liên hệ giữa chúng (gọi là vị từ). Các mệnh đề sẽ được biểu diễn dưới dạng:

Vị từ (<đối tượng 1>, <đối tượng 2>, ..., <đối tượng n>)

Ví dụ biểu diễn vị của các trái cây, các mệnh đề sẽ được viết lại thành:

Cam có vị ngọt  $\Rightarrow$  Vị (Cam, Ngọt)

Cam có màu Xanh  $\Rightarrow$  Màu (Cam, Xanh)

### 2.2.1.2. Phương pháp dạng đồ thị – Mạng ngữ nghĩa và đồ thị khái niệm

#### Mạng ngữ nghĩa

Mạng ngữ nghĩa là một phương pháp biểu diễn tri thức dưới dạng đồ thị, trong đó: Các đỉnh là các yếu tố hay thành phần cấu thành của tri thức. Các yếu tố tri thức này có thể là các đối tượng, các khái niệm, các quan hệ, các biểu thức, hay cũng có thể là các cấu trúc phức tạp hơn nữa; Các cung thể hiện các mối liên hệ giữa các yếu tố tri thức.[2]

#### Đồ thị khái niệm

Đồ thị khái niệm (Conceptual Graphs), được John F. Sowa trình bày lần đầu tiên vào năm 1976 [16]. Đồ thị khái niệm là một đồ thị hữu hạng, liên thông; các đỉnh được chia làm hai loại: Đỉnh khái niệm (concept nodes) biểu diễn cho các khái niệm cụ thể hoặc trừu tượng, đỉnh này được biểu diễn bằng hình chữ nhật; Đỉnh quan hệ (relation nodes) biểu diễn quan hệ giữa các khái niệm, đỉnh này được biểu diễn bởi hình oval có dán nhãn quan hệ.[18]

#### 2.2.1.3. Hệ luật dẫn

Phương pháp biểu diễn tri thức bằng luật sinh được phát minh bởi Newell và Simon trong lúc hai ông đang cố gắng xây dựng một hệ giải bài toán tổng quát. Đây là một kiểu biểu diễn tri thức có cấu trúc. Ý tưởng cơ bản là tri thức có thể được cấu trúc bằng một cặp điều kiện – hành động: “NẾU điều kiện xảy ra THÌ hành động sẽ được thi hành”. Ngày nay, hệ luật dẫn đã trở nên phổ biến và được áp dụng rộng rãi trong nhiều hệ thống trí tuệ nhân tạo khác nhau. Các luật được dùng như là những chỉ dẫn (tuy có thể không hoàn chỉnh) nhưng rất hữu ích để trợ giúp cho các quyết định trong quá trình tìm kiếm, từ đó làm giảm không gian tìm kiếm. Một ví dụ khác là luật sinh có thể được dùng để bắt chước hành vi của những chuyên gia.

Theo cách này, luật sinh không chỉ đơn thuần là một kiểu biểu diễn tri thức trong máy tính mà là một kiểu biểu diễn các hành vi của con người.

#### 2.2.1.4. Frame và Script

**Frame** là một cấu trúc dữ liệu chứa đựng tất cả những tri thức liên quan đến một đối tượng cụ thể nào đó, frame “đóng gói” toàn bộ một đối tượng, tình huống hoặc cả một vấn đề phức tạp thành một thực thể duy nhất có cấu trúc. Một frame mô tả một đối tượng (hay một object). Một frame bao gồm 2 thành phần cơ bản là slot và facet. Một slot là một thuộc tính đặc tả đối tượng được biểu diễn bởi frame. Ví dụ: trong frame mô tả xe hơi có 2 slot là trọng lượng và loại máy. Mỗi slot có thể chứa một hay nhiều facet. Các facet đặc tả một số thông tin hoặc thủ tục liên quan đến thuộc tính được mô tả bởi slot.

**Script** là một cách biểu diễn tri thức tương tự như frame nhưng thay vì đặc tả một đối tượng, nó mô tả một chuỗi các sự kiện. Để mô tả chuỗi sự kiện, script sử dụng một dãy các slot chứa thông tin về con người, đối tượng và hành động liên quan đến sự kiện đó. Tuy cấu trúc của các script là rất khác nhau tùy theo bài toán, nhưng nhìn chung một script thường bao gồm các thành phần sau: (1) điều kiện vào (entry condition), (2) diễn viên (role), (3) tác tố (prop), (4) tình huống (scene), (5) kết quả và (6) phiên bản (track).[1]

### 2.2.2. Ontology

Công nghệ ontology là một công nghệ được nghiên cứu và phát triển mạnh mẽ trong thời gian gần đây, có mặt trong nhiều lĩnh vực từ xử lý ngôn ngữ tự nhiên, công nghệ tri thức, các hệ thống trao đổi tích hợp thông tin, web ngữ nghĩa,... Phần tìm hiểu về ontology dưới đây được tham khảo và trích lượn từ tài liệu [15].

#### 2.2.2.1. Định nghĩa Ontology

Ontology là một thuật ngữ có nguồn gốc từ triết học diễn tả các thực thể tồn tại trong tự nhiên và các mối quan hệ giữa chúng. Theo cách nhìn của triết học, ontology là “một môn khoa học về nhận thức, cụ thể hơn là một nhánh của siêu hình học về tự nhiên và bản chất của thế giới, nhằm xem xét các vấn đề về sự tồn tại hay không tồn tại của các sự vật”.

Trong lĩnh vực trí tuệ nhân tạo có rất nhiều cách định nghĩa về Ontology, một số định nghĩa được xem là kinh điển và được thừa nhận rộng rãi:

- Theo Gruber (1993): “Ontology là một đặc tả rõ ràng cho việc khái niệm hóa trong một lĩnh vực”. Năm 1997, Borst sửa đổi một chút định nghĩa của Gruber: “Ontology được định nghĩa là sự đặc tả hình thức của sự khái niệm hóa được chia sẻ”. Studer (1998) giải thích hai định nghĩa của Gruber và Borst như sau: “Sự khái niệm hóa có nghĩa là mô hình trừu tượng của các sự vật, hiện tượng trên thế giới được xác định qua các khái niệm liên quan của sự vật, hiện tượng đó. Tường minh có nghĩa là các kiểu khái niệm và các ràng buộc giữa chúng là được xác định rõ ràng. Hình thức có nghĩa là Ontology phải được hiểu bởi máy tính, chia sẻ có nghĩa là tri thức trong Ontology được kết hợp xây dựng và được chấp nhận bởi một nhóm hoặc một cộng đồng chứ không theo tri thức chủ quan của cá nhân”.
- Motta (1999) định nghĩa “Ontology là đặc tả một phần của tập hợp các khái niệm được sử dụng hình thức hóa các tri thức của một lĩnh vực cần quan tâm. Vai trò cơ bản của một Ontology là nhằm chia sẻ và sử dụng lại tri thức”.
- Weiss (1999) định nghĩa “Ontology là một đặc tả của các khái niệm và quan hệ trong lĩnh vực quan tâm. Ontology không chỉ là phân cấp các lớp mà còn mô tả các quan hệ”.
- Theo định nghĩa của Hendler năm 2001, “Ontology là một tập hợp các thuật ngữ tri thức (knowledge term), bao gồm từ vựng, các quan hệ ngữ nghĩa, một số luật suy diễn và logic trong một lĩnh vực đặc thù”.

#### **2.2.2.2. Các thành phần của Ontology**

Các thành phần thường gặp của Ontology bao gồm:

- Lớp (class): là thành phần trung tâm của hầu hết các Ontology, mô tả các khái niệm trong miền lĩnh vực. Các lớp thường được tổ chức phân cấp, một lớp có thể có các lớp con biểu diễn khái niệm cụ thể hơn so với lớp cha.
- Thuộc tính (property hay rule, slot): mô tả các đặc tính, đặc trưng, tính chất khác nhau của lớp đối tượng và mỗi thuộc tính đều có giá trị.
- Quan hệ (relation): biểu diễn các kiểu quan hệ giữa các khái niệm.
- Thực thể hay thể hiện (instance): biểu diễn các phần tử riêng biệt của khái niệm, là các thể hiện của lớp.

- Hàm (function): là một loại thuộc tính hay quan hệ đặc biệt, trong đó phần tử thứ n là duy nhất đối với n-1 phần tử còn lại.
- Tiền đề (axiom): biểu diễn các phát biểu luôn đúng mà không cần phải chứng minh hay giải thích. Tiền đề được sử dụng để kiểm chứng sự nhất quán của Ontology hoặc CSTT.
- Luật (rule): mỗi luật cho ta một quy tắc suy diễn để từ các sự kiện giả thiết đang biết suy ra sự kiện mới thông qua định luật, định lý, quy tắc tính toán nào đó.
- Sự kiện (fact): mỗi sự kiện thể hiện một tính chất hay liên hệ nào đó trên các đối tượng hay trên thuộc tính của đối tượng.

#### **2.2.2.3. Phân loại Ontology**

Về cơ bản, Ontology có các loại sau:

**Ontology biểu diễn tri thức (Knowledge representation Ontology)** nắm giữ các biểu diễn nguyên thủy được dùng để chuẩn hóa tri thức trong một mô hình biểu diễn tri thức.

**Ontology tổng quát (Generic Ontology)** bao gồm từ vựng liên quan tới sự vật, hiện tượng, thời gian, không gian,... có ý nghĩa chung chung không chỉ dùng cho riêng một lĩnh vực nào.

**Ontology lĩnh vực (Domain Ontology)** là những Ontology có thể tái sử dụng trong một lĩnh vực nào đó, cung cấp từ vựng về các khái niệm và các mối quan hệ trong một lĩnh vực.

**Ontology tác vụ (Task Ontology)** cung cấp hệ thống các thuật ngữ dùng để giải quyết một vấn đề liên quan đến tác vụ có hoặc không có trong cùng một lĩnh vực nào đó.

**Ontology ứng dụng (Application Ontology):** bao gồm các tri thức cần thiết để mô hình hóa một lĩnh vực cụ thể đặc biệt.

Một số loại ontology khác có thể kể đến như: Metadata Ontology, Ontology lĩnh vực – tác vụ (Domain – Task Ontology); Ontology chỉ mục (Index Ontology); Ontology hỏi và trả lời (Tell and Ask Ontology),... Ngoài ra, cộng đồng nghiên cứu còn phân biệt các Ontology dựa trên độ phức tạp của mô hình biểu diễn: Lightweight Ontology và Heavyweight Ontology.

#### **2.2.2.4. Vai trò và lợi ích sử dụng Ontology**

Nhu cầu ban đầu cần có ontology là để cung cấp các nguồn thông tin giàu ngữ nghĩa của máy tính có thể xử lý và thao tác được, đồng thời vẫn có thể dùng ontology để chia sẻ tri thức giữa người với người và với các hệ thống khác. Một số lợi ích khác của việc sử dụng Ontology có thể kể đến là:

- Chia sẻ sự hiểu biết chung giữa các ứng dụng và con người.
- Cho phép tái sử dụng tri thức: ví dụ, nếu một nhóm nghiên cứu đã phát triển các ontology, nhóm nghiên cứu khác có thể sử dụng lại cho ứng dụng của họ
- Làm rõ lĩnh vực quan tâm, việc đặc tả rõ ràng về miền tri thức sẽ giúp cho người mới dễ tìm hiểu về lĩnh vực họ quan tâm.
- Phân tích tri thức: Phân tích hình thức của các khái niệm, cần thiết cho việc tái sử dụng và mở rộng ontology.

#### **2.2.2.5. Các ngôn ngữ đặc tả Ontology**

Ontology có thể được xây dựng bằng nhiều ngôn ngữ đặc tả khác nhau như: OIL, DAML+OIL, RDF, RDFS, OWL, SHOE... Trong đó, OWL (Web Ontology Language) là một trong những ngôn ngữ được sử dụng phổ biến nhất. OWL là kết quả của một nỗ lực nghiên cứu chung giữa hai ngôn ngữ biểu diễn tri thức chung là DARPA Agent Markup Language Ontology (DAML-OIL) và Ontology Interface Layer (OIL). Đây là kết quả của nhóm nghiên cứu phát triển một ngôn ngữ đánh dấu mới dành cho web ngữ nghĩa. OWL là một sự mở rộng từ RDF và RDFS. Mục đích chính của OWL là đưa ra khả năng suy luận vào web có ngữ nghĩa. OWL là một ngôn ngữ mô tả từ vựng phong phú, trong đó có sự mở rộng từ các lớp (class), lớp con (subclass), thuộc tính (property), thuộc tính con (subproperty) nguyên thủy của RDF và bổ sung thêm nhiều thành phần mới như mối quan hệ giữa các lớp, các giới hạn (restriction) trên tập giá trị (range), tập xác định (domain), điều kiện về số lượng phần tử trong tập hợp, hay các phép toán hội, giao, lấy phần bù trên tập hợp, tính chất đối xứng, bắc cầu của quan hệ, tính chất tương đương, nghịch đảo của hàm số. OWL có ba loại OWL Lite, OWL DL và OWL Full. Mỗi loại OWL sẽ có những đặc tính riêng và do đó sẽ phù hợp trong ngữ cảnh của một ứng dụng cụ thể nào đó.

## **2.3. MỘT SỐ PHƯƠNG PHÁP SUY DIỄN TRÊN MÁY TÍNH**

Khi xây dựng một hệ chuyên gia, bên cạnh việc nghiên cứu phương pháp biểu diễn tri thức của lĩnh vực và cách thức tổ chức lưu trữ cơ sở tri thức trên máy tính, cần phải thiết kế xây dựng một số thuật giải suy diễn tự động nhằm giải quyết các vấn đề – bài toán được đặt ra trong ứng dụng. Thuật giải suy diễn mô phỏng quá trình tư duy của con người trong việc tìm kiếm giải pháp giải quyết vấn đề. Đối với con người quá trình suy luận gồm các loại: suy diễn dạng diễn dịch, dạng quy nạp, suy diễn tương tự. Dựa trên các loại suy luận ấy, trong khoa học máy tính các nhà nghiên cứu đã đề xuất một số chiến lược suy diễn cơ bản trên cơ sở tri thức như sau:

### **2.3.1. Phương pháp suy diễn tiến – suy diễn lùi**

#### **2.3.1.1. Phương pháp suy diễn tiến**

Suy diễn tiến (lập luận tiến - forward chaining hoặc forward reasoning) mô phỏng tư duy suy diễn, diễn giải của con người. Trong [19] tác giả nêu lên định nghĩa cho phép suy diễn tiến như sau “chiến lược suy luận được bắt đầu bằng tập sự kiện đã biết, rút ra các sự kiện mới nhờ dùng các luật mà phần giả thiết khớp với sự kiện đã biết, và tiếp tục quá trình này cho đến khi tìm thấy trạng thái đích, hoặc cho đến khi không còn luật nào khớp được các sự kiện đã biết”.

##### **Ưu điểm suy diễn tiến**

- Làm việc tốt khi bài toán có bản chất là đi thu thập thông tin rồi thấy điều cần suy diễn.
- Cho ra khối lượng lớn các thông tin từ một số thông tin ban đầu, sinh ra nhiều thông tin mới.
- Suy diễn tiến là tiếp cận lý tưởng đối với các loại bài toán cần giải quyết các nhiệm vụ như lập kế hoạch, điều hành, điều khiển và diễn dịch.

##### **Nhược điểm suy diễn tiến**

- Không cảm nhận được rằng chỉ cần một vài thông tin quan trọng. Hệ thống hỏi các câu hỏi có thể hỏi mà không biết rằng chỉ một ít câu đã đi đến kết luận được.
- Hệ thống có thể hỏi cả câu hỏi không liên quan. Có thể các câu trả lời cũng quan trọng nhưng làm người dùng lúng túng khi phải trả lời các câu không dính đến chủ đề.

### **2.3.1.2. Phương pháp suy diễn lùi**

Suy diễn lùi là phép suy luận truy ngược đi từ mục tiêu trở về giả thiết. Xuất phát từ mục tiêu, ta xem xét hệ luật để tìm xem những sự kiện nào có thể dẫn ra được mục tiêu và sự kiện này sẽ được xem xét như là mục tiêu mới cho các bước truy ngược tiếp theo. Quá trình này kết thúc khi tất cả các mục tiêu phát sinh đầu đã được biết hay thuộc giả thiết.

#### **Ưu điểm suy diễn lùi**

- Phù hợp với bài toán đưa ra giả thuyết và kiểm tra xem liệu giả thuyết đó có đúng không.
- Tập trung vào đích đã cho. Nó tạo ra một loạt câu hỏi chỉ liên quan đến vấn đề đang xét, thuận tiện đối với người dùng.
- Khi suy diễn một điều gì từ thông tin đã biết, nó chỉ tìm trên một phần của cơ sở tri thức thích đáng đối với bài toán đang xét.
- Suy diễn lùi được đánh giá cao trong các bài toán như là chẩn đoán, dự đoán và tìm lỗi.

#### **Nhược điểm suy diễn lùi**

Nhược điểm cơ bản của loại suy diễn này là nó thường tiếp theo dòng suy diễn thay vì đúng ra phải dừng ở đó mà sang nhánh khác.

### **2.3.2. Phương pháp suy diễn sử dụng mẫu bài toán và bài toán mẫu**

Trong thực tế, khi giải quyết một bài toán, chúng ta thường không tìm ngay một lời giải mới mà theo George Pólya, trước tiên chúng ta cần phải trả lời các câu hỏi: “Ta đã gặp bài toán này lần nào chưa? Hay đã gặp bài toán này ở dạng khác chưa? Ta có biết bài toán nào có liên quan không? Xét kỹ cái chưa biết (ẩn), thử nhớ lại một bài toán quen thuộc có cùng ẩn hay ẩn tương tự. Đây là một bài toán có liên quan mà ta đã từng giải rồi. Có thể sử dụng nó không? Có thể sử dụng kết quả hay sử dụng phương pháp của nó không?”.

Việc đặt ra những câu hỏi như vậy, chính là quá trình con người đang tìm kiếm lại những bài toán, những dạng toán đã được gấp để từ đó đề xuất cách giải quyết cho phù hợp. Bên cạnh đó, khi tìm thấy bài toán liên quan, thì việc tìm lời giải sẽ cho kết quả nhanh hơn.

Bài toán mẫu thể hiện những dạng bài toán mà nếu sử dụng phương pháp suy diễn chung thì rất tốn thời gian, những dạng này sẽ được sử dụng ghi lại các bước giải để có thể giải quyết các bài toán. Khi gặp những bài toán tương tự, chúng ta sẽ sử dụng các bài toán mẫu này để giải quyết hoặc nếu gặp bài toán khác có “lệch” một chút, dựa trên các bước giải đã có, có thể nhanh chóng cập nhật một lời giải cho bài toán hiện tại.[1]

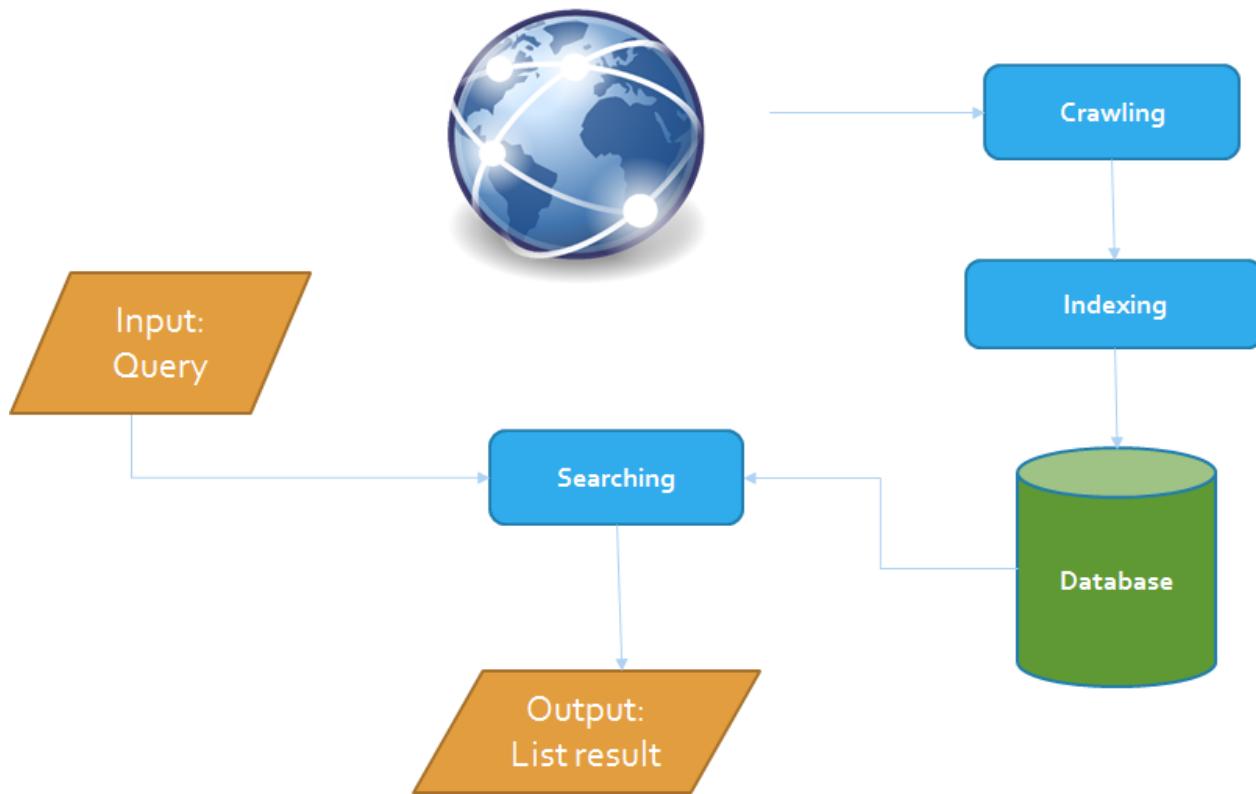
### 2.3.3. Lập luận dựa trên tình huống

Lập luận dựa trên tình huống (Case-based reasoning – CBR) là phương pháp tìm lời giải bài toán trên cơ sở hiệu chỉnh lời giải của các bài toán khác đã có. Trong một bài toán mới, hệ thống CBR sẽ tìm kiếm lại một hoặc một vài bài toán mẫu tương tự với nó từ tập các bài toán mẫu. Nếu lời giải của mẫu lập luận tìm được phù hợp với bài toán mới thì sẽ được sử dụng lại. Ngược lại, hệ thống sẽ tìm cách hiệu chỉnh lời giải cũ để đưa ra một lời giải mới. Việc hiệu chỉnh phụ thuộc nhiều vào miền ứng dụng và mỗi hệ thống CBR sẽ có một cơ chế hiệu chỉnh riêng. Đơn vị chính của một hệ thống CBR là các bài toán đã được giải trong quá khứ còn gọi là bài toán lập luận mẫu. Mỗi đơn vị tri thức này được mô tả gồm: mô tả bài toán (giả thiết, kết luận) và lời giải (tập các bước giải).[1]

## 2.4. TRUY HỎI THÔNG TIN

Vấn đề truy vấn thông tin (information retrieval – IR) đóng vai trò rất quan trọng trong thiết kế và xây dựng các hệ thống quản lý và tìm kiếm thông tin, tài liệu. Hiện nay có rất nhiều định nghĩa và cách giải thích khác nhau về khái niệm “information retrieval”. Một số định nghĩa được xem là kinh điển và được thừa nhận rộng rãi như: Theo (Salton, 1968) thì “Truy xuất thông tin là một nhánh nghiên cứu của Khoa học máy tính liên quan đến việc cấu trúc hóa, phân tích, tổ chức, lưu trữ, tìm kiếm và truy hỏi thông tin”. Định nghĩa tổng quát này có thể được áp dụng cho nhiều loại thông tin khác nhau (như văn bản, hình ảnh, âm thanh, video...) và cho nhiều loại ứng dụng tìm kiếm khác nhau. Các tác giả C.Bourne và B.Anderson cũng đã định nghĩa “Truy xuất thông tin là tìm kiếm thông tin (thường là các tài liệu) ở một dạng không có cấu trúc (thông thường là văn bản), nhằm thỏa mãn nhu cầu thông tin của người dùng từ những nguồn thông tin lớn (được lưu trữ trên các máy tính)”[14].

Một hệ thống IR thường có ba khối chức năng chính thực hiện việc thu thập, lập chỉ mục, và tìm kiếm. Kiến trúc tổng quát của một hệ thống tìm kiếm thông tin như hình bên dưới:



Hình 2: Kiến trúc tổng quát của hệ tìm kiếm thông tin

- Bộ thu thập (Crawling): Là một module có chức năng tự động thu thập các thông tin-tài liệu cho hệ thống.
- Bộ lập chỉ mục (Indexing): Rút trích những đơn vị thông tin trong tài liệu (từ, cụm từ, khái niệm) và biểu diễn tài liệu để máy tính có thể hiểu được.
- Searching: Tìm kiếm trong kho lưu trữ và trả về các kết quả phù hợp với nội dung câu truy vấn.

Nhìn chung có hai hướng tiếp cận chính cho việc nghiên cứu các hệ thống tìm kiếm thông tin: hướng thống kê và hướng ngữ nghĩa. Một số mô hình nổi tiếng được nghiên cứu theo hướng tiếp cận thống kê thuận túy có thể kể đến là mô hình Boolean (Boolean model), Không gian vector (Vector Space model), các mô hình xác suất (Probabilistic models), Language models, Lập chỉ mục ngữ nghĩa tiềm ẩn (Latent Semantic Indexing – LSI). Trong đó mô hình không gian vector được sử dụng phổ biến nhất. Mô hình không gian vector sẽ biểu diễn mỗi

tài liệu văn bản như một tập hợp các term xuất hiện trong toàn bộ tập văn bản và hình thành một không gian mà trong đó mỗi term riêng biệt đóng vai trò là một chiều trong không gian đó. Người ta gán thêm cho mỗi term một trọng số cụ bô, phản ánh mức độ hữu ích, tầm quan trọng của term đó trong việc mô tả nội dung hay chủ đề mà tài liệu đang đề cập tới. Có nhiều cách tính trọng số được sử dụng, trong đó, phương pháp tính  $\text{idf} \times \text{tf}$  được xem là phổ biến và sử dụng rộng rãi nhất. Sau khi đã biểu diễn tập tài liệu và câu truy vấn thành các vector trong không gian tài liệu, bước tiếp theo là tính toán độ tương quan (giống nhau) giữa chúng bằng cách sử dụng các độ đo như Inner-product, Cosin similarity, Hệ số Jaccard, Distance metrics. Ưu điểm của mô hình không gian vector là tính được mức độ tương đồng giữa một truy vấn và mỗi tài liệu, đại lượng này có thể được dùng để xếp hạng các tài liệu trả về. Nhược điểm là số chiều biểu diễn cho tập văn bản có thể rất lớn nên tốn không gian lưu trữ, các từ khóa biểu diễn được xem là độc lập với nhau, mối quan hệ về ngữ nghĩa giữa các từ không được xét đến[14].

## 2.5. MỘT SỐ KĨ THUẬT KHÁC ĐƯỢC ÁP DỤNG

### 2.5.1. Ngôn ngữ đặc tả EBNF

Nhu cầu về các hệ thống đặc tả hình thức đã được nói đến từ nhiều năm. Trong báo cáo ALGOL 60, John Backus đã trình bày một hệ thống ký hiệu hình thức để mô tả cú pháp ngôn ngữ lập trình BNF (Backus-Naur form). BNF là ngôn ngữ đặc tả mô tả cú pháp của ngôn ngữ chuyên biệt (domain-specific language – DSL). BNF được thiết kế một cách rõ ràng, không có sự nhập nhằng, để người ta có thể xây dựng một parser cho những ngôn ngữ chứa đầy những lý thuyết toán học, chỉ đơn giản, bằng BNF [22]. Cấu trúc của BNF bắt đầu bằng một ký hiệu, theo sau đó là những luật để thay thế ký hiệu đó.

symbol := alternative1 | alternative2 ...

EBNF (Extended Backus-Naur form) là ngôn ngữ đặc tả mở rộng của BNF. EBNF bổ sung thêm +, [ ] và { }[22]. Ví dụ mô tả một số thực bằng EBNF:

`<number> := '-'? <digit>+ ('.' <digit>+)?`

`<digit> := '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'`

### 2.5.2. Regular expression

Regular expressions (Biểu thức chính quy – viết tắt là Regex) là một chuỗi ký tự đặc biệt được dùng làm mẫu (pattern) để phân tích sự trùng khớp của một tập hợp các chuỗi nào đó. Regular expression được sử dụng trong nhiều ngôn ngữ lập trình như PHP, C#, Javascript... để kiểm tra tính hợp lệ các dữ liệu đầu vào, hầu hết những ngôn ngữ lập trình trên đều cung cấp các thư viện hoặc hàm xử lý để lập trình viên có thể làm việc với regular expression.

- Trong lập trình, Regex dùng để kiểm tra tính hợp lệ của dữ liệu đầu vào như email (ví dụ @mailserver.com), số điện thoại (10 số hoặc 11 số và phải là số), chứng minh nhân dân, ngày tháng năm sinh (dd/mm/yyyy) ...
- Trong một số phần mềm, Regex được áp dụng để xử lý chuỗi như tìm kiếm, thay thế chuỗi (giống như chức năng Find & Replace trong Microsoft Word) hoặc trích xuất chuỗi từ một chuỗi dài hơn.

### 2.5.3. BBCode

BBCode (Bulletin Board Code) là một ngôn ngữ định dạng bài viết cho các bản tin trên một số diễn đàn và trên wiki. Công dụng của BBCode là đưa ra các ký tự, hay câu, chúng sẽ hình thành một hệ thống bản tin trước khi được chuyển đổi thành ngôn ngữ HTML mà trình duyệt web có thể dịch được. BBCode được đưa ra để cung cấp thêm sự an toàn, tiện lợi, và nhiều phương thức hơn để người dùng có thể dễ dàng định dạng bài viết của họ trên các diễn đàn Internet. Bbcodes rất đơn giản để bổ sung các trích dẫn cần thiết. Trước đây, rất nhiều bản tin cho phép định dạng với các tag HTML, vùng có ảnh hưởng là một phần hiển thị cũng có thể phá vỡ hiệu ứng của cả trang web, hoặc HTML có thể sử dụng để chạy Javascript để chỉ đạo cho việc tấn công XSS. Một số bổ sung của BBCode bị chặn khi chuyển đổi mã từ BBCode sang HTML, điều này có vẻ ngược lại với tính bảo mật là mục đích mà BBCode muốn mang lại. Dưới đây là một số ví dụ bbcode:

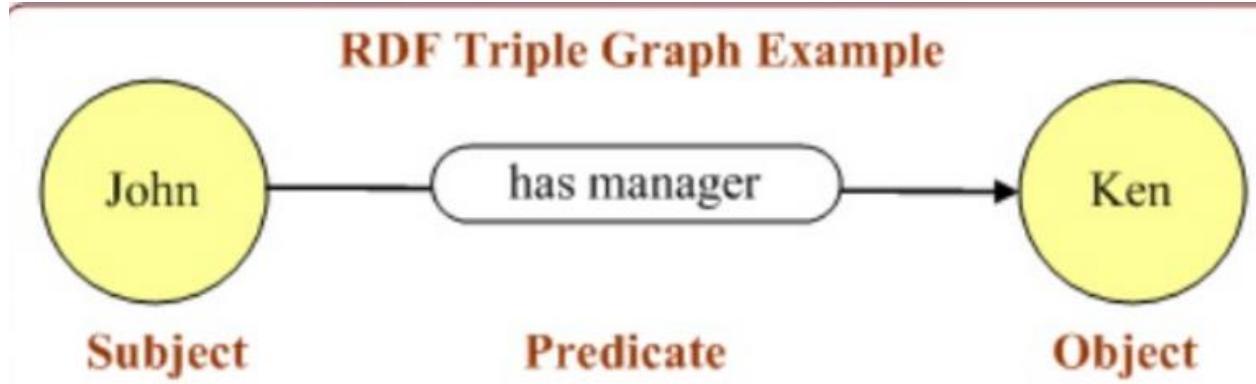
bbcode	html	Kết quả
[b]Chữ đậm[/b]	<strong>Chữ đậm</strong>	<b>Chữ đậm</b>

[img="uit.png"]UIT[/img]		 DAI HOC QUOC GIA THANG PHO HO CHI MINH TRUONG DAHOC CONG NGHE THONG TIN <b>UIT-HCM</b>
[url="uit.edu.vn"]UIT[/url]	<a href="uit.edu.vn">UIT</a>	<a href="uit.edu.vn">UIT</a>

#### 2.5.4. Linked Data

Linked Data (Dữ liệu liên kết) là thuật ngữ chỉ cách thức liên kết các nguồn dữ liệu trên Web hay đơn giản là sử dụng Web để kết nối các dữ liệu từ nhiều nguồn khác nhau, là một trong những cách tạo dựng nền tảng cho web ngữ nghĩa.

RDF (Resource Description Framework) là một tập hợp các nguyên tắc dành cho ngôn ngữ đánh dấu, cung cấp mô hình dữ liệu và cú pháp đơn giản sao cho các hệ thống độc lập có thể trao đổi và sử dụng. Đồng thời, nó được thiết kế sao cho hệ thống máy tính có thể hiểu được và có thể đọc được thông tin, chứ không chỉ đơn giản là để trình bày dữ liệu cho người dùng. RDF là mô hình các triples (các định dạng mà máy tính có khả năng đọc được như RDF/XML, json, N3, RDFA, Turle,...)



Hình 3: Cấu trúc căn bản của RDF triples

Trong đó:

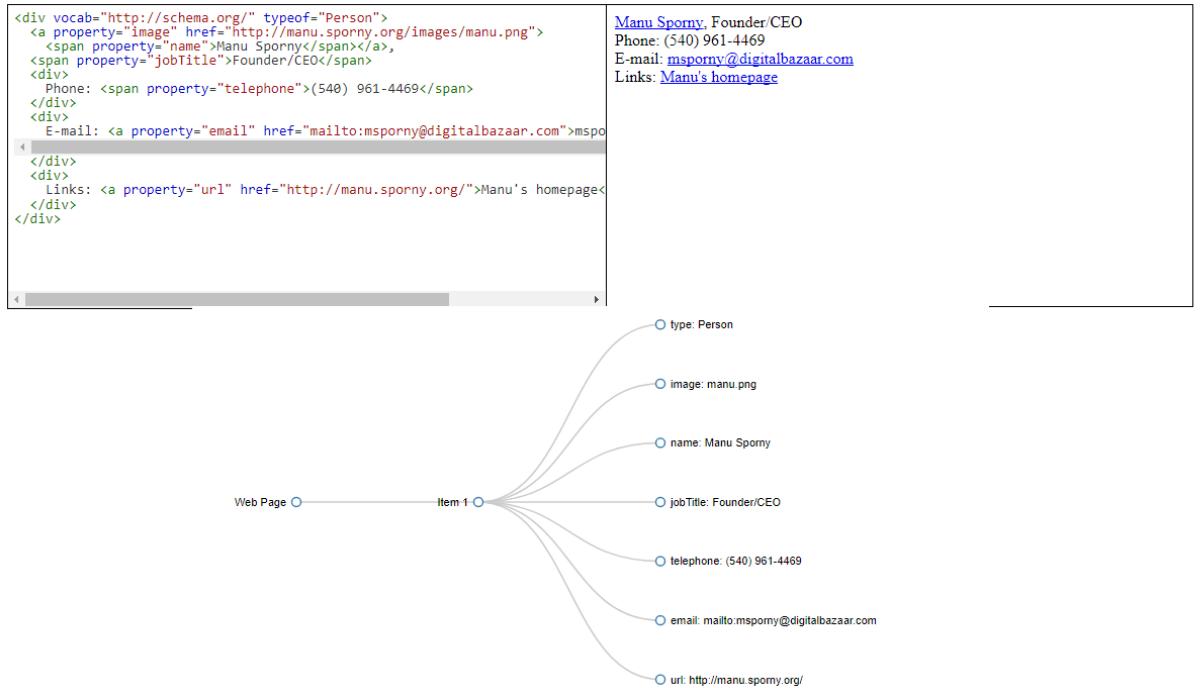
- S (subject): URL
- P (predicate): URL
- O (object): URL hoặc literal

Sau khi đã xây dựng được một Hệ thống chạy trên nền web, các web crawlers (GoogleBot[1], Bingbot[2], Slurp Bot[3], DuckDuckBot[4],...) sẽ dịch thông tin từ các tập tin

HTML. Để giúp những tác từ này hiểu được ngữ nghĩa của tập tin, ta cần sử dụng 1 chuẩn cấu trúc để bổ sung vào HTML. Một chuẩn phổ biến được W3C khuyến nghị là RDFa. RDFa không chỉ hoạt động trong HTML5, mà còn là một công nghệ chung cho đánh dấu Dữ liệu được Liên kết trong nhiều ngôn ngữ giống HTML. Điều này có nghĩa là nó hoạt động tốt trong các tài liệu XML, tài liệu SVG, cũng như tài liệu HTML4, XHTML1 và XHTML5. Trên thực tế, nếu ngôn ngữ đánh dấu của bạn dựa trên XML, thì bạn đã có thể sử dụng RDFa trong tài liệu của mình ngay hôm nay. Có một phiên bản của RDFa dành cho người mới bắt đầu được gọi là RDFa Lite. Phiên bản đầy đủ của RDFa mang tính biểu cảm hơn và cho phép các tác giả Web giải quyết bất kỳ vấn đề đánh dấu dữ liệu có cấu trúc nào mà họ có thể ném vào đó. Bất kể định dạng tài liệu Web nào bạn sử dụng, rất có thể RDFa đã làm việc trong đó.

Dựa trên chuẩn RDFa Lite. Sử dụng các thuộc tính sau để lưu thông tin metadata:

- about
- rel và rev
- src, href và resource
- property
- content
- datatype
- typeof



Hình 4: Ví dụ về Linked Data

### 2.5.5. Thuật toán Quine-McCluskey và Petrick's method

Mục đích của việc áp dụng phương pháp Quine-McCluskey và Petrick's method để tối ưu hóa việc biểu diễn luật, trình bày chi tiết ở chương 3. Phần lý thuyết về các thuật toán này có thể tham khảo từ tài liệu [23].

#### Petrick's method

Trong đại số boolean, Petrick's method là một kĩ thuật để xác định tổng các prime implicant tối thiểu từ bằng kết quả được tạo. Phương pháp Petrick's method rất tẻ nhạt đối với các biểu đồ lớn áp dụng như K-map nhưng đối với thuật toán Quine-McCluskey có thể áp dụng kĩ thuật này.[24]

#### Các bước thực hiện Petrick's method

- Bước 1: Loại bỏ essential prime implicant và minterm nhờ vào Quine-McCluskey như biểu đồ.
- Bước 2: Kí hiệu gán nhãn các row các kí hiệu như A, B, K, L,...
- Bước 3: Hình thành 1 hàm P đúng với tất cả các column.

- Bước 4: Thay thế  $p$  bằng 1 sản phẩm tối thiểu bằng cách áp dụng biểu thức  $X + XY = X$ .
- Bước 5: Mỗi term đại diện cho một giải pháp, chọn term có số biến ít nhất.
- Bước 6: Hình thành biểu thức  $f$  từ các term được chọn ở bước 5.

## CHƯƠNG 3. THIẾT KẾ CƠ SỞ TRI THỨC

Mục tiêu nghiên cứu của đề tài là thiết kế và xây dựng một hệ thống ứng dụng “*hỗ trợ khuyến nông trên cây lúa*”. Đây là một dạng hệ chuyên gia có khả năng quản lý tri thức của miền tri thức lúa nước trên máy tính, đồng thời giải quyết những vấn đề liên quan đến tri thức như đưa ra những kết quả ở dạng chuẩn đoán, tư vấn, gợi ý dựa trên kiến thức, kinh nghiệm của người nông dân và kỹ sư nông nghiệp được đưa vào hệ thống. Bên cạnh đó hệ thống còn cho phép người dùng tra cứu thông tin tri thức và tìm kiếm những bài viết có liên quan. Về mặt kiến trúc, hệ thống cần có ba thành phần cơ bản là cơ sở tri thức, bộ suy diễn và bộ tìm kiếm. Cơ sở tri thức chứa kiến thức của miền tri thức ứng dụng, làm cơ sở cho việc suy diễn tính toán, giải quyết các vấn đề và yêu cầu đặt ra đối với hệ thống. Tri thức cần phải được lưu trữ và biểu diễn dưới dạng thích hợp sao cho máy tính có thể hiểu và xử lý được. Chất lượng của việc thiết kế cơ sở tri thức sẽ ảnh hưởng đến chất lượng của bộ suy diễn và tìm kiếm được xây dựng ở các bước sau.

Dựa trên quy trình xây dựng một hệ chuyên gia nói chung, giai đoạn đầu tiên là thu thập tri thức và xây dựng mô hình biểu diễn tri thức cho ứng dụng. Trên cơ sở mô hình biểu diễn đã có, xác lập tổ chức cơ sở tri thức cụ thể trên máy và thiết kế các xử lý cơ bản trên cơ sở tri thức đó.

### 3.1. THU THẬP VÀ PHÂN LOẠI TRI THỨC

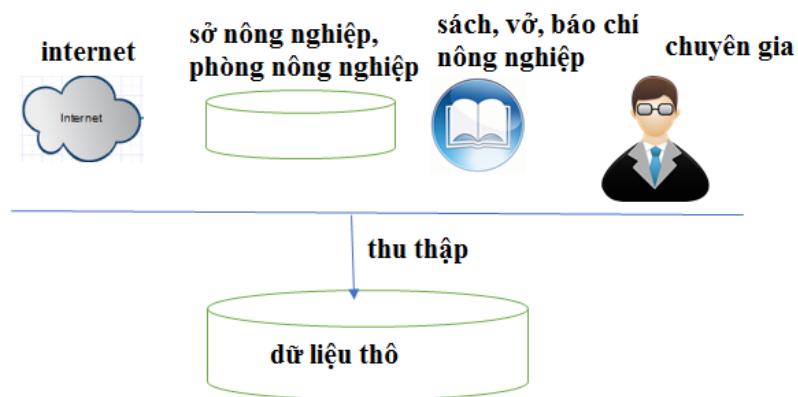
#### 3.1.1. Miền tri thức và nguồn tri thức thu thập

Trong việc thiết kế cơ sở tri thức, quá trình thu thập tri thức là công việc rất quan trọng, giúp ta xác định rõ tri thức của hệ thống, cung cấp lượng kiến thức cần thiết đủ để bộ suy diễn và bộ tìm kiếm giải quyết những vấn đề đặt ra trong ứng dụng. Để chất lượng tri thức thu thập được tốt, nhóm nghiên cứu đã chọn nguồn thu thập có mức độ tin cậy khá cao từ các tài liệu sách vở về lĩnh vực nông nghiệp, khảo sát lấy ý kiến từ các chuyên gia là những người nông dân và kỹ sư nông nghiệp, từ các trang web của Bộ Nông Nghiệp và Phát Triển Nông Thôn Việt Nam, Phòng Nông Nghiệp Và Phát Triển Nông Thôn Huyện Tuy Phước, Phòng Nông Nghiệp & Phát Triển Nông Thôn Hoài Nhơn... Để thực hiện đề tài, nhóm đã chọn Thị Xã An Nhơn-Tỉnh Bình Định là nơi nghiên cứu. Bình Định là nơi mà nông dân gắn liền với cây lúa,

có truyền thống nông nghiệp lúa nước, cũng là nơi đã hung đúc ý tưởng cho nhóm thực hiện đề tài này.

### 3.1.2. Cách thức thu thập

Đầu tiên là tìm hiểu sơ bộ về lĩnh vực nông nghiệp bằng cách dùng các công cụ tìm kiếm trên internet để tìm các bài viết có liên quan đến nông nghiệp lúa nước, sử dụng một danh sách các câu truy vấn như sau: (1) nông nghiệp lúa nước hiện nay, (2) các bệnh hại của cây lúa, (3) thực trạng lúa nước tại tỉnh Bình Định, (4) cách phòng trừ sâu bệnh vào mùa xuân... Sau khi tìm hiểu sơ bộ về miền tri thức này, để tìm hiểu kĩ hơn nhóm tìm đến người nông dân lâu năm tại các vùng của xã Nhơn Hạnh, xã Nhơn Phong, xã Nhơn An,... để tìm hiểu kinh nghiệm, kỹ năng trồng lúa, cách người dân phát hiện sâu bệnh hại, cách xử lý sâu bệnh hại.... Tiếp đến nhóm tìm đến UBND xã Nhơn Hạnh, Phòng Nông nghiệp Tỉnh Bình Định, các hợp tác xã tại khu vực, nhóm xin tài liệu bao gồm: các mẫu nghiên cứu, các kết quả thử nghiệm, thông báo, bài báo, bài nghiên cứu, các bản tin và tất cả tài liệu khác có liên quan. Tại phòng Nông nghiệp huyện Tuy Phước Tỉnh Bình Định nhóm có trao đổi với kĩ sư nông nghiệp về những tri thức thu thập được tại các vùng xã trước đó. Quá trình thu thập tri thức còn soát những thông tin cần thì nhóm tìm hiểu bằng các công cụ tìm kiếm trên các trang web, lọc những thông tin nhóm cần vào hệ thống.



Hình 5: Quá trình thu thập tri thức

### **3.1.3. Kết quả thu thập**

Kết quả thu thập được bao gồm các thông tin sau:

- Tìm hiểu sâu bệnh hại trên cây lúa, tìm hiểu về quy trình trồng
- Thu thập tài liệu về phân bón, thuốc trừ sâu, ốc, cỏ, vụ mùa, đất, môi trường, nhiệt độ, khí hậu, nước,...
- Thu thập các tài liệu, thí nghiệm về các giống lúa
- Tìm hiểu các giai đoạn, quy trình tưới nước, bón phân, đón đồng, các biện pháp chăm sóc và phát triển của cây lúa.
- Nhờ những kinh nghiệm của người nông dân cùng với các kỹ sư nông nghiệp nhóm đã xây dựng cho mình một bộ luật là những kinh nghiệm được đúc kết để chuẩn đoán sâu bệnh hại cho cây lúa.

## **3.2. MÔ HÌNH TRI THỨC CỦA ỨNG DỤNG**

Nhóm nghiên cứu chọn ontology làm mô hình biểu diễn tri thức vì: ontology cung cấp một bộ từ vựng chung chứa những thông tin dữ liệu có cấu trúc về một miền tri thức nào đó, có khả năng biểu diễn ngữ nghĩa có khả năng hỗ trợ lập luận. Thông tin dữ liệu của ontology máy có thể hiểu và xử lý được, nghĩa là ontology cung cấp một kiến thức quan trọng trong việc chuyển đổi thông tin và dữ liệu dạng tri thức. Ontology có thể chia sẻ, hiểu cấu trúc của thông tin giữa người dùng với nhau hoặc giữa các ứng dụng; Có thể tái sử dụng và tạo ra miền dữ liệu rõ ràng, đơn giản, dễ nhớ và dễ sử dụng. Vì những lý do trên nên nhóm chọn Ontology biểu diễn tri thức.

Mô hình ARK biểu diễn tri thức của “*hệ thống hỗ trợ khuyến nông trên cây lúa*” là một bộ gồm 6 thành phần:

$$ARK = (C, Re, H, Attrs, Rules, Fu)$$

Trong đó :

1. C: Tập hợp các lớp tương ứng với các khái niệm trong miền tri thức trồng lúa
2. Re: Tập các quan hệ giữ các lớp (khái niệm).
3. H: Tập quan hệ phân cấp giữa các lớp (khái niệm).

4. Attrs: Tập các thuộc tính trong miền tri thức tròng lúa.
5. Rules: Tập các luật suy diễn trong miền tri thức tròng lúa.
6. Fu: Tập biến ngôn ngữ môi trường trong miền tri thức lúa nước.

### 3.2.1. Tập các khái niệm C

Lớp, còn được gọi là khái niệm (concept) là thành phần quan trọng của mô hình ontology. Một lớp đại diện cho một nhóm các thể hiện khác nhau cùng chia sẻ những thuộc tính chung. Tập các khái niệm trong miền tri thức lúa nước bao gồm

$$C = \{ \text{Benh}, \text{Thuoc}, \text{Co}, \text{TacNhanGayBenh}, \text{Phan}, \text{PhanHuuCo}, \text{PhanVoCo}, \text{GiaiDoan},$$

$$\text{QuyTrinh} \dots \} \quad C = |47|$$

Với mỗi khái niệm  $c \in C$  có cấu trúc thông tin bên trong là

$$c = (\text{Name}, \text{Attributes}, \text{Individuals})$$

Trong đó:

- Name: Tên của lớp
- Attributes: là tập các thuộc tính của c, mỗi  $attr \in Attributes$  thể hiện đặc điểm, tính chất của lớp.  $Attributes \subset Attrs$  (tập các thuộc tính được định nghĩa bên ngoài).
- Individuals: là tập các cá thể của lớp, biểu diễn các phần tử riêng biệt của lớp, là các thể hiện của lớp, thể hiện sẽ có các giá trị thuộc tính tương ứng với mỗi thuộc tính của lớp.

Ví dụ 1: Lớp Benh

- Name = “Bệnh”
- Attributes = { Tên, Thời gian cách ly,  
Biện pháp phòng trừ,  
Cách trị bệnh }
- Individuals: { Đạo ôn, Sâu xanh, Cỏ lá, Óc bưu vàng }

Các giá trị thuộc tính Đạo ôn được thể hiện như sau: { Tên = “Đạo ôn”; Thời gian cách ly = 10 ngày; Biện pháp phòng trừ = “không nên lấy quá nhiều nước, bón nhiều đạm”; Cách trị bệnh = “bón nhiều phân lân” }

Ví dụ 2: Lớp SauBenh

- Name = “Sâu bệnh”
- Attributes = { Tên, Thuốc đặc trị, Nguồn gốc, Biện pháp phòng trừ, Đặc điểm, ... }
- Individuals = { Sâu năn, Sâu cuốn lá nhỏ, Sâu cuốn lá lớn, Sâu đục thân, Châu chấu, ... }

Các giá trị thuộc tính của cá thể “Sâu năn” được thể hiện như sau: { Tên = “Sâu năn”; Nguồn gốc = “Sâu năn hay còn gọi là muỗi hành,...”; Đặc điểm = “là loài muỗi nhỏ, dài khoảng 3-5mm, bụng màu hồng nhạt, chúng đẻ trứng rải rác từng quả, rất nhỏ, màu trắng”; Thuốc đặc trị = “AnphaA” }.

Các giá trị thuộc tính cá thể “Sâu cuốn lá nhỏ” được thể hiện như sau: { Tên = “Sâu cuốn lá nhỏ”; Đặc điểm = “là một loài dịch hại thường xuyên có mặt trên đồng ruộng. Chúng gây hại bằng cách, sau khi nở sâu non nhả tơ cuốn dọc hai mép...”; Biện pháp phòng trừ = “”; Đặc điểm = “thân màu trắng,...”; Thuốc đặc trị = “Comda gold 5WG” }.

### 3.2.2. Tập quan hệ giữa các lớp Re

Re là tập quan hệ hai ngôi trên 2 tập thể hiện của 2 lớp  $A * B \in C$  (tập các khái niệm).

Re = {TriBenh, CoThuoc, CoQuyTrinh, CoGiaiDoan, CoSau, CoThuoc, ...} mỗi  $r_i$  thuộc Re là 1 quan hệ 2 ngôi trên 2 tập thể hiện của 2 class  $A * B \in C$ ,  $r_i \subset A * B$  Trong đó  $r_i = \{ (a,b) | a \in A, b \in B \}$  trong đó a là individual thuộc lớp A, b là individual thuộc lớp B.

Ví dụ 1: Quan hệ TriBenh (trị bệnh) là tập quan hệ 2 ngôi trên 2 lớp Thuoc và Lớp Benh

TriBenh = {(DaoOn, Anpha), (VangLa, AnphaB), ...}

Trong đó: DaoOn, VangLa là individual của lớp Benh

Anpha, AnphaB là individual của lớp Thuoc

Ví dụ 2: Quan hệ XuatHienOGiaiDoan (xuất hiện ở giai đoạn) là tập quan hệ 2 ngôi trên 2 lớp Benh và lớp Giai Doan

$$\text{XuatHienOGiaiDoan} = \{( \text{DaoOn}, \text{GiaDoanTro} \}, \{ \text{VangLa}, \text{GiaiDoanDong} \}, \dots \}$$

Trong đó: DaoOn, VangLa là cá thể của lớp Benh

GiaiDoanTro, GiaiDoanDong là cá thể của lớp GiaiDoan

### 3.2.3. Tập các quan hệ phân cấp giữa các khái niệm H

H là quan hệ 2 ngôi phân cấp (cha con) của các khái niệm thuộc C. Cấu trúc của H như sau:

$$H = \{ \text{PhanCap} \}$$

Quan hệ PhanCap thuộc H là 1 quan hệ 2 ngôi (cha-con) của 2 class ( $C_i, C_j \in C$ ), mỗi  $H_i \in H$  sẽ có quan hệ như sau:  $H_i \subset C * C$  sẽ có cấu trúc  $H_i = \{ (C_i, C_j) \mid C_i, C_j \in C \}$ , trong đó  $C_i, C_j$  là class của C.

$$\begin{aligned} \text{PhanCap} = & \{ (\text{TacNhanGayBenh}, \text{Co}), (\text{TacNhanGayBenh}, \text{SauBenh}), \\ & (\text{Thuoc}, \text{ThuocKichThich}), \dots \} \end{aligned}$$

### 3.2.4. Tập các thuộc tính của các lớp Attrs

Attrs là tập các thuộc tính mô tả các đặc tính, đặc trưng, tính chất khác nhau của khái niệm của các class được định nghĩa ở C

$$\text{Attrs} = \{ \text{CachCheBien}, \text{DenNgay}, \text{NgayCachLy}, \text{Name}, \dots \}, |\text{Attrs}| = 40$$

Mỗi thuộc tính attribute  $\in \text{Attrs}$  là một bộ gồm ba thành phần:

$$\text{attribute} = (\text{attributeName}, \text{valueType}, \text{ranges})$$

- attributeName: tên thuộc tính.
- valueType: biểu diễn kiểu giá trị cho phép cho thuộc tính: kiểu *String* mô tả giá trị chuỗi; kiểu *Number* mô tả giá trị số; kiểu *Boolean* mô tả giá trị cờ có-không; kiểu *Enumerated* mô tả giá trị trong một danh sách mà thuộc tính được phép sở hữu.

- ranges: tập ràng buộc của thuộc tính, giới hạn miền giá trị có thể có cho thuộc tính.
- Ví dụ về thuộc tính “LuongThuoc”
  - LuongThuoc.attributeName = "Cách chế biến"
  - LuongThuoc.valueType = String
  - LuongThuoc.ranges = Thuoc
- Ví dụ về thuộc tính “ThoiGianThuHoach”
  - ThoiGianThuHoach.attributeName = “thời gian thu hoạch”
  - ThoiGianThuHoach.valueType = String
  - ThoiGianThuHoach.ranges = { VuMua, ThuHoachBaoQuan }

### 3.2.5. Tập các luật của tri thức Rules

Rules là tập các luật suy diễn trên các sự kiện liên quan đến các thuộc tính cũng như liên quan đến khái niệm. Mỗi luật rules cho ta một quy tắc suy diễn từ những sự kiện đã biết để suy ra sự kiện mới. Trong quá trình thu thập và phân loại tri thức lúa nước, luật có thể được phân tách thành hai loại cơ bản như sau:

- Luật quan hệ (Relation\_Rule)
- Luật tường minh (Rule\_Exp)

#### **Luật quan hệ (Relation\_Rule)**

Tương ứng với những quan hệ được xây dựng ở tập Re trong mô hình cơ sở tri thức, ta sẽ xây dựng được danh sách luật quan hệ tương ứng. Mỗi luật rule trong luật quan hệ gồm có 3 thành phần:

$$\text{Relation\_Rule} = (\text{NameRelation}, \text{Domain}, \text{Range})$$

- NameRelation: Tên quan hệ
- Domain: tên class 1 của quan hệ mà có tập các individual thuộc về
- Range: tên class 2 của quan hệ mà có tập các individual thuộc về

Ví dụ 1: Ta có luật Relation\_Rule\_1 được xây dựng dựa trên quan hệ TriBenh giữa 2 class Thuốc và Bệnh được phát biểu như sau:

Relation\_Rule\_1 = (TriBenh, Thuoc, Benh)

Ví dụ 2: Ta có luật Relation\_Rule\_2 được xây dựng dựa trên quan hệ XuatHienOGiaiDoan giữa 2 class Thuốc và Bệnh được phát biểu như sau:

Relation\_Rule\_2 = (XuatHienOGiaiDoan, Benh, GiaiDoan)

### Luật tường minh (Rule\_Exp)

Luật tường minh là các luật dạng luật dẫn, mỗi luật có dạng “if <fact> then <fact>”[1].

Mỗi luật r thuộc tập luật tường minh đều có cú pháp chung đó là:

$$\{sk_1, sk_2, \dots\} \rightarrow \{sk_i, sk_j, \dots\} \{cf\}$$

cf ∈ [0,1]: thể hiện sự tin tưởng các chuyên gia vào tin cậy của luật. cf càng dần về 0 độ tin cậy càng thấp, cf càng dần về 1 độ tin cậy càng cao.

Ví dụ 1: Người nông dân phát biểu như sau: Nếu như thân của cây lúa lùn và màu lá lúa màu vàng, lá úa thì kết luận lúa đó xuất hiện hiện tượng thiếu đạm cần bổ sung đạm độ tin cậy của phát biểu là 0.5. thì luật được biểu diễn như sau:

$$\begin{aligned} &\{HinhDangThan=ThanLun, MauLa=LaVang, HinhdangLa=LaUa\} \rightarrow \\ &\quad \{HienTuong=HienTuongThieuDam\}, cf = 0.5; \end{aligned}$$

Ví dụ 2: Người nông dân phát biểu như sau: Nếu như thân của cây lúa lùn và màu lá lúa màu vàng, Đốm vàng xuất hiện, đang ở giai đoạn làm đồng kết luận là bị bệnh đốm vẫn độ tin cậy của phát biểu là 0.9. thì luật được biểu diễn như sau:

$$\begin{aligned} &\{HinhDangThan=ThanLun, MauLa=LaVang, HinhdangLa=LaUa, \\ &\quad GiaiDoan=GiaiDoanLamDong, DomVang="XuatHien"\} \rightarrow \\ &\quad \{Benh=BenhDomVan\}, cf = 0.9; \end{aligned}$$

### 3.2.6. Tập biến ngôn ngữ mô trường trong miền tri thức lúa nước Fu

Trong lĩnh vực trồng lúa, có thể có những kiến thức “không chắc chắn” và sự “không hoàn chỉnh”. Ví dụ như Nhiệt độ cao, Độ ẩm thấp, Lượng mưa nhiều... là những kiến thức không có giá trị chắc chắn. Do đó hình thành nên thành phần Fu để mô tả cho các biến ngôn ngữ không chắc chắn đó:

$$Fu = \{X, T, F, Ranges\}$$

Ta có:

- X là tập tên các biến ngôn ngữ môi trường.
- T là miền các giá trị ngôn ngữ môi trường trong miền tri thức lúa nước.
- F là tập hàm liên thuộc của biến. Mỗi hàm liên thuộc U là một bộ gồm hai thành phần:

$$F = (\text{functionType}, \text{functionAttributes}, \text{name})$$

- functionType là kiểu hàm thuộc của tập mờ, có thể là tập mờ tam giác *Triangle*, hình thang *Trapezoid*, hình *L* hay gamma tuyến tính *R*.
- functionAttributes là tập các thuộc tính của hàm, số lượng thuộc tính phụ thuộc vào kiểu hàm functionType.
- name là giá trị cho biến mờ của tập T.
- Ranges là tập các giá trị ràng buộc giới hạn trên và giới hạn dưới của biến nếu có.

Ví dụ các biến mờ là LoaiRuong

$$X = \text{"Loại Ruộng"}$$

$$T = \{\text{"RuongCan"}, \text{"RuongSau"}\}$$

$$U = \{( \text{RuongCan}, \text{Triangle}, \left\{ \begin{array}{l} a = 0, b = 1.75, m = 0.75, h = 1, \mu(x) = \\ \frac{x}{0.75}, 0 < x < 0.75 \\ 1.75 - x, 0.75 < x < 1.75 \\ 1, x = 0.75 \end{array} \right\}), (\text{RuongSau}, \text{Triangle}, \{a = 1.25, b = 3, m = 3, h = 1, \mu(x) = \\ \left\{ \begin{array}{l} 0, x \leq 1.25 \text{ or } x \geq 3 \\ \frac{x-1.25}{0.75}, 1.25 < x < 2 \\ 3 - x, 2 < x < 3 \\ 1, x = 2 \end{array} \right\})\};$$

### 3.3. TỔ CHỨC LUU TRỮ TRI THỨC TRÊN MÁY TÍNH

Mô hình ontology ARK có thể được tổ chức lưu trữ ở mức vật lý bởi một hệ thống các tập tin có cấu trúc, thể hiện các thành phần trong mô hình như sau:

(1). Tập tin “rice.owl” lưu trữ các thành phần C, Attrs, H, Re, định nghĩa các biến ngôn ngữ Fu.

(2). Tập tin “inference-rules.txt” lưu lại tập luật Rules.

(3) Tập tin “fuzzy-model.txt” lưu các thành phần biến ngôn ngữ Fu trong Ontology ARK.

## Cấu trúc lưu trữ trên file rice.owl

### 1. Định nghĩa thành phần của lớp

Để khai báo lớp Ôc, Vụ mùa thì file owl sẽ có định nghĩa như sau:

```
<Declaration>
    <Class IRI="#Oc"/>
</Declaration>
<Declaration>
    <Class IRI="#VuMua"/>
</Declaration>
```

Định nghĩa individual Bệnh búu rẽ

```
<Declaration>
    <NamedIndividual IRI="#BenhBuuRe"/>
</Declaration>
```

Định nghĩa individual Bệnh lùn xoắn lá thuộc lớp Bệnh

```
<ClassAssertion>
    <Class IRI="#Benh"/>
    <NamedIndividual IRI="#BenhLunXoanLa"/>
</ClassAssertion>
```

Định nghĩa thành phần property “tên”

```
<AnnotationAssertion>
    <AnnotationProperty abbreviatedIRI="rdfs:label"/>
    <IRI>#ten</IRI>
    <Literal xml:lang="vi" datatypeIRI="http://www.w3.org/1999/02/22-rdf-syntax-
ns#PlainLiteral">ten</Literal>
```

```
</AnnotationAssertion>
```

Xây dựng giá trị thuộc tính Mô tả cho individual Phân xanh

```
<DataPropertyAssertion>
```

```
    <DataProperty IRI="#moTa"/>
```

```
    <NamedIndividual IRI="#PhanXanh"/>
```

```
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Sử dụng các loại  
cây lá tươi bón ngay vào đất không qua quá trình ủ do đó chỉ dùng đê bón lót. Cây phân xanh  
thường được dùng là cây họ đậu: đỉen thanh, muồng, keo dậu, cỏ Stylo, đỉen đỉen...</Literal>
```

```
</DataPropertyAssertion>
```

Xây dựng giá trị thuộc tính Tên cho individual Phân xanh

```
<DataPropertyAssertion>
```

```
    <DataProperty IRI="#ten"/>
```

```
    <NamedIndividual IRI="#PhanXanh"/>
```

```
    <Literal datatypeIRI="http://www.w3.org/2001/XMLSchema#string">Phân  
xanh</Literal>
```

```
</DataPropertyAssertion>
```

## 2. Lưu trữ quan hệ cha – con giữa 2 lớp

Trong mô hình tri thức ta có tồn tại quan hệ phân cấp H – quan hệ cha-con giữa 2 class  
được lưu trữ như sau:

Ví dụ: ta có Cỏ là con của lớp Tác nhân gây bệnh

```
<SubClassOf>
```

```
    <Class IRI="#Co"/>
```

```
    <Class IRI="#TacNhanGayBenh"/>
```

```
</SubClassOf>
```

## 3. Định nghĩa Tập các thuộc tính Attrs

Trong file owl khi khai báo tập các thuộc tính sẽ được định nghĩa như sau:

Ví dụ: thuộc tính ngày cách ly, mô tả,...

```
<Declaration>
```

```

<DataProperty IRI="#tuNgayCachLy"/>
</Declaration>
<Declaration>
    <DataProperty IRI="#denNgayCachLy"/>
</Declaration>
<Declaration>
    <DataProperty IRI="#moTa"/>
</Declaration>

```

Định nghĩa kiểu dữ liệu cho từng thuộc tính được lưu trữ như sau:

```

<DataPropertyRange>
    <DataProperty IRI="#moTa"/>
    <Datatype abbreviatedIRI="xsd:string"/>
</DataPropertyRange>

```

#### **4. Định nghĩa tập quan hệ Re**

Trong file owl khi khai báo quan hệ sẽ được định nghĩa như sau:

```

<Declaration>
    <ObjectProperty IRI="#coTacNhanGayBenh"/>
</Declaration>
<AnnotationAssertion>
    <AnnotationProperty abbreviatedIRI="rdfs:label"/>
    <IRI>#coTacNhanGayBenh</IRI>
    <Literal xml:lang="vi" datatypeIRI="http://www.w3.org/1999/02/22-rdf-syntax-
ns#PlainLiteral">có tác nhân gây bệnh</Literal>
</AnnotationAssertion>

```

#### **5. Định nghĩa các biến ngôn ngữ Fu**

Đối với thành phần T trong mô hình sẽ được lưu như sau:

Ví dụ khai báo lớp LinguisticVariable là lớp biến ngôn ngữ:

```
<Declaration>
```

```

<Class IRI="#LinguisticVariable"/>
</Declaration>
<!-- Khai báo các lớp con LinguisticVariable -->
<SubClassOf>
    <Class IRI="#DoAm"/>
    <Class IRI="#LinguisticVariable"/>
</SubClassOf>
<SubClassOf>
    <Class IRI="#DoSauRuong"/>
    <Class IRI="#LinguisticVariable"/>
</SubClassOf>

```

Định nghĩa các giá trị vật lý của các biến ngôn ngữ:

```

<ClassAssertion>
    <Class IRI="#DoAm"/>
    <NamedIndividual IRI="#DoAmCao"/>
</ClassAssertion>
<ClassAssertion>
    <Class IRI="#DoAm"/>
    <NamedIndividual IRI="#DoAmThap"/>
</ClassAssertion>
<ClassAssertion>
    <Class IRI="#DoAm"/>
    <NamedIndividual IRI="#DoAmTrungBinh"/>
</ClassAssertion>

```

### Lưu trữ file “fuzzy-model.txt”

Đối với hàm liên thuộc của biến ngôn ngữ sẽ không được lưu trong tập tin owl mà được lưu vào file fuzzy-model.txt. Dưới đây là cấu trúc đặc tả cho tập tin:

fuzzy-model = membership-functions linguistic-variables
---

```

membership-functions = membership-function { membership-function }
membership-function = function function-name ( parameters ) function-statement ;
parameter = parameter-name { , parameter-name }
function-statement = begin if-statement { ; if-statement } ; end
if-statement = if expression then output-statement
output-statement = OUTPUT := expression
expression = simple-expression | simple-expression relational-operator simple-expression
simple-expression = [ sign ] term { adding-operator term }
term = factor { multiplying-operator factor }
factor = INPUT | parameter-name | literal | ( expression ) | not factor
relational-operator = = | <> | < | <= | > | >=
sign = + | -
adding-operator = + | - | or
multiplying-operator = * | / | % | and
linguistic-variables = linguistic-variable { linguistic-variable }
linguistic-variable = var variable-name " unit-of-measure " variable-statement ;
variable-statement = begin term-statement { ; term-statement } ; end
term-statement = term term-name : function-name ( arguments )
arguments = literal { , literal }
function-name = IDENT
parameter-name = IDENT
variable-name = IDENT
term-name = IDENT
unit-of-measure = STRING
literal = NUMBER

```

Từ ngôn ngữ đặc tả trên xây dựng được file fuzzy-model.txt để lưu trữ các thành phần của biến ngôn ngữ như sau:

```
function trapmf(a, b, c, d)
```

```

begin
    if INPUT <= a then OUTPUT := 0;
    if (a <= INPUT) and (INPUT <= b) then OUTPUT := (INPUT - a) / (b - a);
    if (b <= INPUT) and (INPUT <= c) then OUTPUT := 1;
    if (c <= INPUT) and (INPUT <= d) then OUTPUT := (d - INPUT) / (d - c);
    if d <= INPUT then OUTPUT := 0;
end;

function trimf(a, b, c)
begin
    if INPUT <= a then OUTPUT := 0;
    if (a <= INPUT) and (INPUT <= b) then OUTPUT := (INPUT - a) / (b - a);
    if (b <= INPUT) and (INPUT <= c) then OUTPUT := (c - INPUT) / (c - b);
    if c <= INPUT then OUTPUT := 0;
end;

variable DoAm "phần trăm"
begin
    term DoAmThap : trimf(0, 10, 20);
    term DoAmTrungBinh : trimf(10, 20, 30);
    term DoAmCao : trimf(20, 30, 40);
end;

variable DoSauRuong "mét"
begin
    term RuongSau : trimf(0, 10, 20);
end;

variable LuongMua "milimét"

```

```

begin
    term MuaIt  : trimf(0, 200, 500);
    term MuaNhiu : trimf(500, 700, 1000);
end;

```

Ví dụ: Định nghĩa biến ngôn ngữ Độ âm có các giá trị là độ âm thấp và độ âm cao, có hàm liên thuộc là tam giác, độ âm thấp từ 0 → 20 độ C và độ âm cao từ 10 → 30 độ C.

```

variable NhietDo "độ"
begin
    term NhietDoThap : trimf(0, 10, 20);
    term NhietDoCao : trimf(10, 20, 30);
end;

```

- **Tập luật “ inference-rules.txt”**

Đối với dạng luật tường minh – dạng luật dựa trên kinh nghiệm của người nông dân, và luật quan hệ giữa các lớp trong ontology, sẽ được tổ chức trong file inference-rules.txt. Cấu trúc đặc tả để quản lý tập luật luật được thể hiện như sau:

```

inference-rules = inference-rule { inference-rule }

inference-rule = relation-rule | logic-rule

relation-rule = class-name -> class-name ;

logic-rule = expression -> conclusions [ { certainty-factor } ] ;

expression = term [ | term ]

term = factor [ & factor ]

factor = fact | ( expression )

conclusions = fact [ & fact ]

fact = class-name = individual-name

certainty-factor = NUMBER

class-name = IDENT

individual-name = IDENT

```

Từ ngôn ngữ đặc tả trên xây dựng được file luật inference-rules.txt. Biểu diễn một vài luật được xây dựng từ file luật inference-rules.txt:

**Luật quan hệ được tóm tắt như sau: Class A → Class B**

Benh -> Thuoc;

TacNhanGayHai -> Thuoc;

HienTuongBenh -> Thuoc;

Thuoc -> Benh;

**Luật tường minh được lưu như sau:**

MauLa=Test\_LaVang & HinhdangThan=Test\_ThanLun -> Benh=Test\_BenhVangLa {0.1e-10};

MauLa=LaVang & HinhdangThan=ThanUn -> Benh=BenhVangLa {1.0e+0};

HinhdangThan=ThanLun & (MauLa=LaVang | HinhdangLa=LaUa) ->

HienTuong=HienTuongThieuDam;

Trieuchung=DeNhanhKem & ChieuCao=ChieuCaoThap ->

HienTuong=HienTuongNgoDocHuuCo;

MauLa=LaVang & HinhdangLa=DauLaChay -> HienTuong=NgoDocPhen;

Dat=DatPhen -> HienTuong=NgoDocPhen {0.8};

Trieuchung=DeNhanhKem & ChieuCao=ChieuCaoThap & Dat=DatPhen &

Giaidoan=GiaidoanMa & (MauLa=MauVang | MauLa=MauNauNhat)->

HienTuong=HienTuongNgoDocHuuCo {0.95};

NhietDo=NhietDoCao & DoAm=DoAmThap -> Benh=BenhDaoOn {0.5};

MauLa=LaVang & HinhdangLa=DauLaChay & HinhdangRe=XuatHienPhen ->

HienTuong=NgoDocPhen {0.9};

Ray=RayXanhDuoiden -> Benh=BenhTungo {0.95};

HienTuong=NgoDocPhen -> Benh=BenhVangLa {0.95};

### **3.4. XỬ LÝ TRUY XUẤT THÔNG TIN TRÊN FILE LUU TRỮ**

#### **3.4.1. Xử lý truy xuất dữ liệu trên file**

Đối với việc trích xuất thông tin từ file owl nhóm sử dụng thư viện owl api trong java để đọc dữ liệu từ file owl theo cấu trúc của file owl. OWL API là một thư viện Java cho phép đọc và thay đổi thông tin của một ontology bất kỳ. Thư viện này cũng được công cụ protégé sử dụng. Hệ thống sẽ sử dụng thư viện suy luận HermiT.

##### **Thuật giải truy vấn thông tin từ file rice.owl**

Bước 1: Mở 1 tập tin ontology và kiểm tra tính đúng đắn của ontology

```
File file = /* địa chỉ vật lý tập tin ontology */;  
OWLOntology ontology = manager.loadOntologyFromOntologyDocument(file);  
String prefix = ontology.getOntologyID().getOntologyIRI().get() + "#";  
OWLReasonerFactory reasonerFactory = new ReasonerFactory(); // ReasonerFactory là  
factory cho bộ suy luận HermiT  
ConsoleProgressMonitor progressMonitor = new ConsoleProgressMonitor();
```

Bước 2: Tạo config cho phép lưu log ra console

```
OWLReasonerConfiguration config = new SimpleConfiguration(progressMonitor);  
OWLReasoner reasoner = reasonerFactory.createReasoner(ontology, config);
```

Bước 3: Tính toán suy luận trên ontology trước khi truy vấn

```
reasoner.precomputeInferences();
```

Bước 4: Kiểm tra tính đúng đắn của dữ liệu, nếu trả về true hệ thống mới cho phép truy vấn

```
boolean isConsistent = reasoner.isConsistentFile();
```

Bước 5: Lấy ontology lên theo format cấu trúc file owl

```
OWLOntologyFormat format = manager.getOntologyFormat(ontology);
```

#### **3.4.2. Áp dụng phương pháp Quine – McCluskey và Petrick's Method để tối ưu luật**

Áp dụng thuật giải cài đặt như thuật giải giới thiệu ở chương 2 Quine-McCluskey và Petrick's Method cài đặt vào vào chương trình.

Ví dụ mô phỏng thuật giải: Ta có luật

HinhDangThan=ThanUn & (MauLa=LaVang | HinhDangLa=LaUa)  $\rightarrow$

HienTuongBenh=HienTuongThieuDam

Đặt HinhDangThan=ThanUn là a, MauLa=LaVang là b và MauLa=LaUa là c ta suy ra được hàm  $F = a \wedge (b \vee c)$

Từ đó ta xây dựng được bảng Truth Table như sau:

<b>abc</b>	<b>True/False</b>
000	F
001	F
010	F
011	F
100	T
101	T
110	T
111	T

Hàm F (a,b,c) =  $\Sigma m(4,5,6,7) = ab'c' + ab'c + abc' + abc$

Number	minterm	Binary
1	M4	100
2	M5	101
2	M6	110
3	M7	111

$m(4,5) : 10\_ ; m(4,6) : 1\_0 ; m(5,7) : 1\_1 ; m(6,7) : 11\_$

4	5	6	7	a	b	c
<b>M(4,5)</b>	X	X		1	0	-
<b>M(4,6)</b>	X		X	1	-	0

<b>M(5,7)</b>	X	X	1	-	1
<b>M(6,7)</b>	X	X	1	1	-

Áp dụng Pertrick's Method:

Đặt  $L = M(4,5)$ ,  $M = M((4,6))$ ,  $N = M(5,7)$ ,  $K = M(6,7)$

$$P = (L+M)(L+N)(M+K)(N+K)$$

$$P = (L+MN)(K+MN)$$

$$P = LK + MN$$

Chọn  $LK & MN$

$F(a,b,c) = a'b + ab = ac' + ac$  kết quả trả về ta có thể phân tích ra 2 luật:

$a \wedge b$ : HinhDangThan=ThanUn v MauLa=LaVang  $\rightarrow$

HienTuongBenh=HienTuongThieuDam

$a \wedge c$ : HinhDangThan=ThanUn v HinhDangLa=LaUa  $\rightarrow$

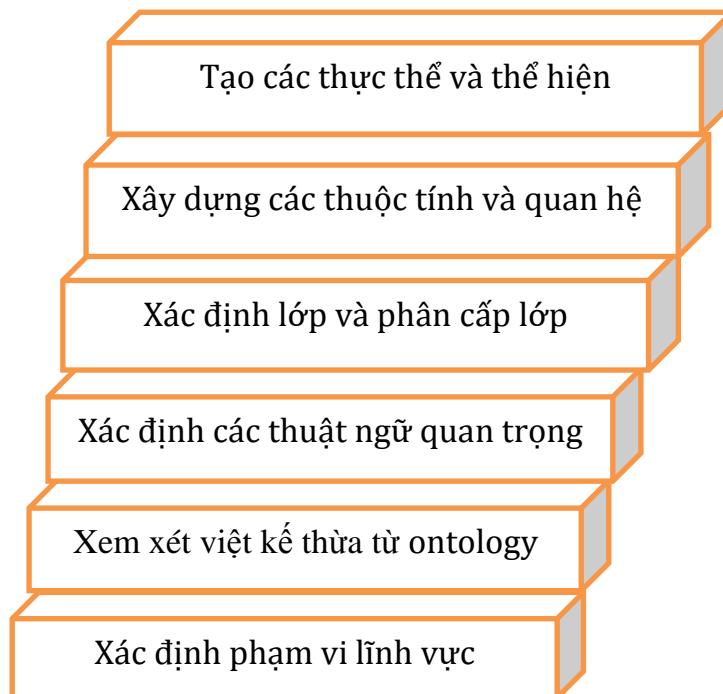
HienTuongBenh=HienTuongThieuDam

### 3.5. QUY TRÌNH VÀ KẾT QUẢ XÂY DỰNG ONTOLOGY CỦA ÚNG DỤNG

#### 3.5.1. Quy trình xây dựng Ontology

Có nhiều phương pháp khác nhau để xây dựng một ontology, nhưng nhìn chung các phương pháp đều thực hiện dựa trên hai bước cơ bản là: xây dựng cấu trúc lớp phân cấp và định nghĩa các thuộc tính cho lớp. Trong thực tế việc phát triển một ontology để mô tả miền cần quan tâm là một việc không hề đơn giản, phụ thuộc nhiều vào công cụ sử dụng, tính chất, quy mô,... Hiện nay có rất nhiều công cụ xây dựng & quản lý ontology trong đó có protégé, Swoop, và các trình soạn thảo khác. Protégé là một phần mềm soạn thảo ontology và quản lý tri thức được phát triển bởi trường đại học Stanford. Protégé là một công cụ có thể xây dựng các ontology trong nhiều lĩnh vực, protégé giúp định nghĩa các lớp, mối quan hệ giữa các thuộc tính của lớp trong ontology. Việc soạn thảo ontology trên protégé được sử dụng để soạn thảo và quản lý ontology trong các lĩnh vực khác nhau là 68.20%, tiếp theo đó là trình soạn thảo Swoop là 13,60%, các trình soạn thảo còn lại chiếm 13%. Điều này cho thấy protégé là

trình soạn thảo được nhiều người sử dụng nhất với các tính năng tiện ích, dễ sử dụng và quản lý hiệu quả, qua khảo sát đề tài quyết định lựa chọn trình soạn thảo protégé để cài đặt và quản lý ontology đã được xây dựng. Dưới đây là quy trình xây dựng Ontology trên protégé cho mô hình tri thức ứng dụng:



Hình 6: Quy trình xây dựng ontology của ứng dụng

### Bước 1: Xác định lĩnh vực và phạm vi mà ontology thể hiện

Một trong những cách để xác định phạm vi hoạt động là phát họa ra một danh sách các câu hỏi. Đối với miền tri thức việc xác định xây dựng phạm vi ontology nhóm xây dựng bằng cách đặt ra những câu hỏi về miền tri thức lúa nước:

- Có bao nhiêu giống lúa đặc điểm sinh trưởng của từng giống lúa?
- Có bao nhiêu loại bệnh, sâu bệnh hại? cách chữa trị và cách phòng trừ? Thời gian hoạt động? Thời gian cách ly? Thuốc chữa trị? Khả năng xấu nhất có thể xuất hiện?
- Quy trình trồng lúa như thế nào?
- Quy trình bón phân như thế nào?
- Các loại phân bón? bón như thế nào? thành phần?...

### Bước 2: Xem xét việc kê thừa những ontology đã được định nghĩa trước đó

Ontology có thể sử dụng lại tri thức về một miền lĩnh vực đã được xây dựng từ trước. Kiểm tra xem miền tri thức được xây dựng đã có ai xây dựng trước đó, xem xét lại có thể tái sử dụng bằng việc chỉnh sửa và mở rộng các nguồn hiện có cho miền tri thức hiện tại không? Tuy nhiên đối với miền tri thức lúa nước mà nhóm đang xây dựng, việc tìm kiếm và kiểm tra trên những miền ontology đã có sẵn thì hiện tại nhóm không tìm được miền ontology nào phù hợp cho miền tri thức của nhóm.

### **Bước 3: Xác định các thuật ngữ quan trọng trong ontology**

Việc xác định các thuật ngữ trong ontology là rất cần thiết, những thuật ngữ quan trọng miền tri thức lúa, những thuộc tính. Ví dụ xác định những thuật ngữ quan trọng trong miền tri thức lúa: lúa, thân, phân bón, thuốc trừ sâu, cỏ, bệnh hại, sâu, nước, nhà máy phân bón, thành phần, ngày cách ly, ngày bón phân, ngày lấy nước... điều quan trọng nhất là xác định một danh sách các thuật ngữ mà không cần phải lo lắng gì về việc chồng chéo nghĩa, quan hệ giữa các thuật ngữ. Hai bước tiếp theo xây dựng: (1) xác định thuật ngữ là thành phần lớp, (2) xác định các thuộc tính của các khái niệm được kết hợp với nhau. Thật khó để làm bước nào trước bước nào sau, thông thường sẽ tạo một vài định nghĩa của các khái niệm trong hệ thống phân cấp, sau đó sẽ mô tả thuộc tính cho các khái niệm này. Hai bước này cũng là bước quan trọng nhất trong ontology.

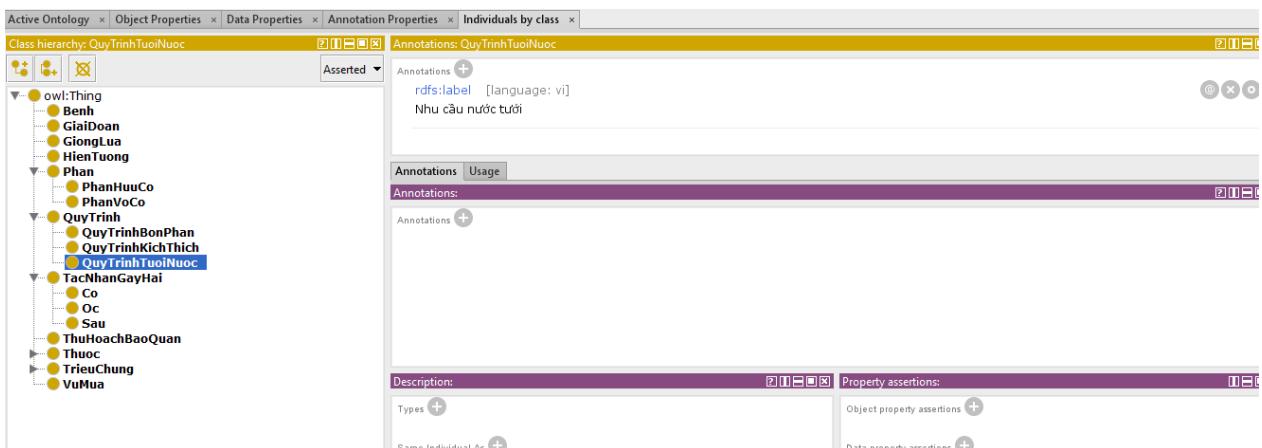
### **Bước 4: Xác định lớp và phân cấp lớp**

Có nhiều cách tiếp cận để phát triển hệ thống phân cấp lớp

- Phương pháp từ trên xuống (top-down): bắt đầu với định nghĩa của các lớp tổng quát nhất trong lĩnh vực và sau đó chuyên biệt hóa các khái niệm đó.
- Phương pháp từ dưới lên (bottom-up): bắt đầu với định nghĩa của các lớp cụ thể nhất, như các lá trong cây phân cấp. Sau đó gộp các lớp đó lại thành các khái tổng quát hơn.
- Phương pháp kết hợp: kết hợp giữa phương pháp từ trên xuống và từ dưới lên: bắt đầu từ định nghĩa các lớp dễ thấy trước và sau đó tổng quát hóa và chuyên biệt hóa các lớp đó một cách thích hợp.

Sau khi xác định được các lớp và hướng phân cấp lớp từ trên xuống, Ontology lúa nước đã xác định được với gần 50 lớp, nhóm đã xây dựng cài đặt trên hệ thống mềm protégé, và

ngôn ngữ sử dụng để quản lý ontology là OWL, Class Hierarchy chính là cấu trúc phân cấp của các lớp trong protégé:



Hình 7: Phân cấp lớp trong protégé

## Bước 5: Xây dựng các thuộc tính và ràng buộc của lớp

- Xây dựng các thuộc tính**

Để xác định thuộc tính cho các lớp, ta quay trở lại danh sách các thuật ngữ đã liệt kê được. Hầu hết các thuật ngữ còn lại (sau khi đã xác định lớp) là thuộc tính của các lớp đó. Với mỗi thuộc tính tìm được, ta phải xác định xem nó mô tả cho lớp nào. Các thuộc tính có thể có như kiểu giá trị, các giá trị cho phép, số các thuộc tính, và các đặc trưng khác mà giá trị của thuộc tính có thể nhận.

- String: là một loại kiểu dữ liệu đơn giản thường được sử dụng trong các thuộc tính như: tên, mô tả,...
- Boolean: là loại kiểu dữ liệu yes hoặc no. Ví dụ khi chúng ta chọn có xuất hiện triệu chứng này hay không xuất hiện triệu chứng này trên cây lúa.
- Number: là kiểu dữ liệu số, dùng để mô tả giá trị của các thuộc tính như giá lúa, giá phân bón...
- Enumerated: dùng để liệt kê danh sách những giá trị của thuộc tính.

### Xác định Domain và Range của thuộc tính

Domain là định nghĩa phạm vi mà thuộc tính đó thuộc về. Ví dụ thuộc tính Tên phạm vi mà nó thuộc về là các lớp: Benh, SauBenh, Co,...

Range: định nghĩa kiểu giá trị của thuộc tính. Ví dụ: Tên có kiểu string.

The screenshot shows the Protégé interface with the 'Data property hierarchy' tab selected. On the left, under 'owl:topDataProperty', there is a list of properties: 'cachCheBien', 'denNgay', 'denNgayCachLy', 'hasMembershipFunction', 'loai', 'mota', 'ten', 'thanhPhan', 'tuNgay', and 'tuNgayCachLy'. On the right, the 'Characteristics' and 'Description' panes are visible. In the 'Characteristics' pane, the 'Functional' checkbox is unchecked. In the 'Description' pane, the 'Ranges' section is active, showing 'xsd:string' as the range. There are also sections for 'Equivalent To', 'SubProperty Of', 'Domains (intersection)', and 'Disjoint With'.

Hình 8: Xác định kiểu dữ liệu của thuộc tính

### Bước 6: Định nghĩa các ràng buộc quan hệ

Dựa vào các khái niệm trong C, tìm quan hệ có thể được giữa các khái niệm. Các quan hệ có thể có của khái niệm trong C, lấy 1 quan hệ cụ thể  $r \in R$ , theo sự phân cấp khái niệm, chúng ta xác định miền (Domain) và phạm vi (Range) phù hợp của quan hệ.

Ví dụ: Đối với quan hệ miền tri thức lúa trong protégé được thể hiện bằng tab Object property hierarchy. CoThuoc là quan hệ phân cấp giữa các individual của 2 class Benh và Thuoc, trong đó Domain là Bệnh và Range là Thuốc.

The screenshot shows the Protégé interface with the 'Object Properties' tab selected. On the left, under 'owl:topObjectProperty', there is a list of properties: 'coGiaiDoan', 'coQuyTrinh', 'coTacNhanGayHai', 'coThuoc', and 'xuatHienOGiaiDoan'. On the right, the 'Annotations' and 'Characteristics' panes are visible. In the 'Annotations' pane, there are annotations for 'coThuoc': 'rdfs:label' with value '[language: vi] có thuốc' and 'rdfs:comment' with value '[language: vi] được chữa bởi thuốc'. In the 'Characteristics' pane, the 'Functional', 'Inverse functional', 'Transitive', 'Symmetric', 'Asymmetric', 'Reflexive', and 'Irreflexive' checkboxes are unchecked. The 'Domains (intersection)' section shows 'Benh or TacNhanGayHai' as the domain, and the 'Ranges (intersection)' section shows 'Thuoc' as the range. There is also a note 'Asserted in: http://csuit.edu.vn/ontologies/rice'.

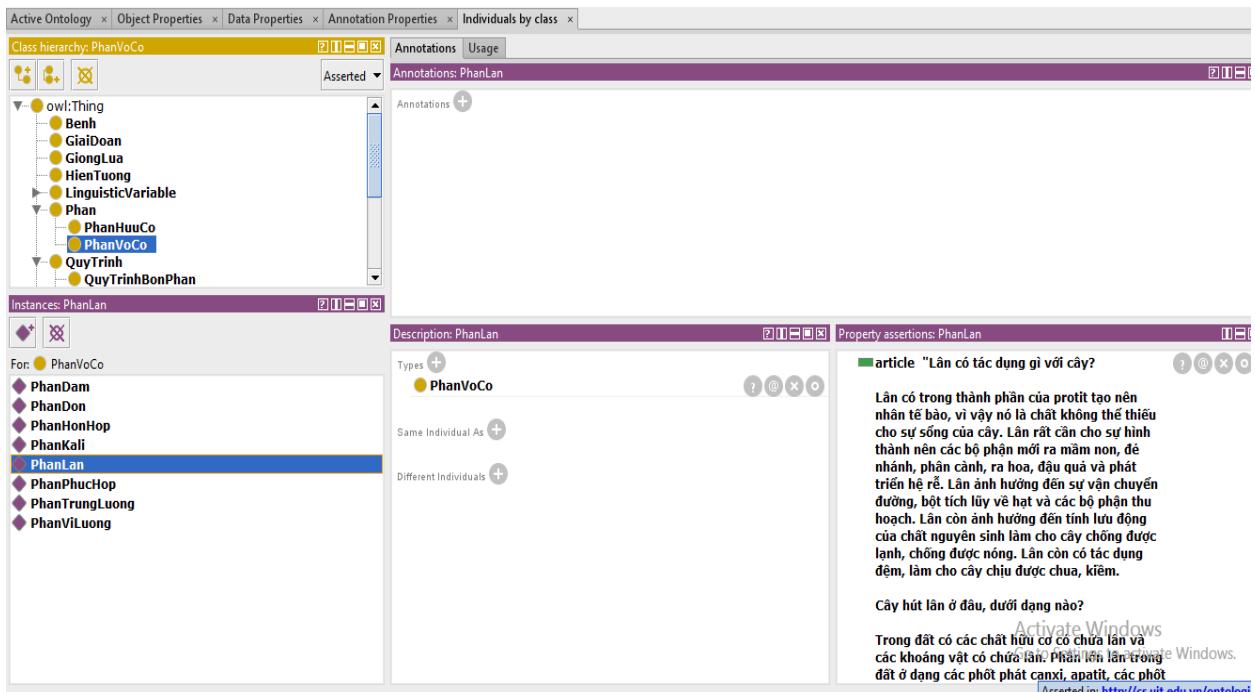
Hình 9: Tạo Relation trên protégé

### Bước 7: Tạo Individual

Đây là bước cuối cùng khép lại một vòng lặp xây dựng ontology. Việc tạo thể hiện cho một lớp là quá trình điền các thông tin vào các thuộc tính của lớp đó bằng các xác định:

- Chọn lớp mà các thể đó thuộc về
- Điền giá trị thuộc tính cho cá thể
- Điền vào quan hệ của thuộc tính

Dưới đây là những individual được xây dựng trong miền tri thức lúa, Instances là tab thể hiện các thực thể trong protégé.



Hình 10: Thể hiện Individual trên Protégé

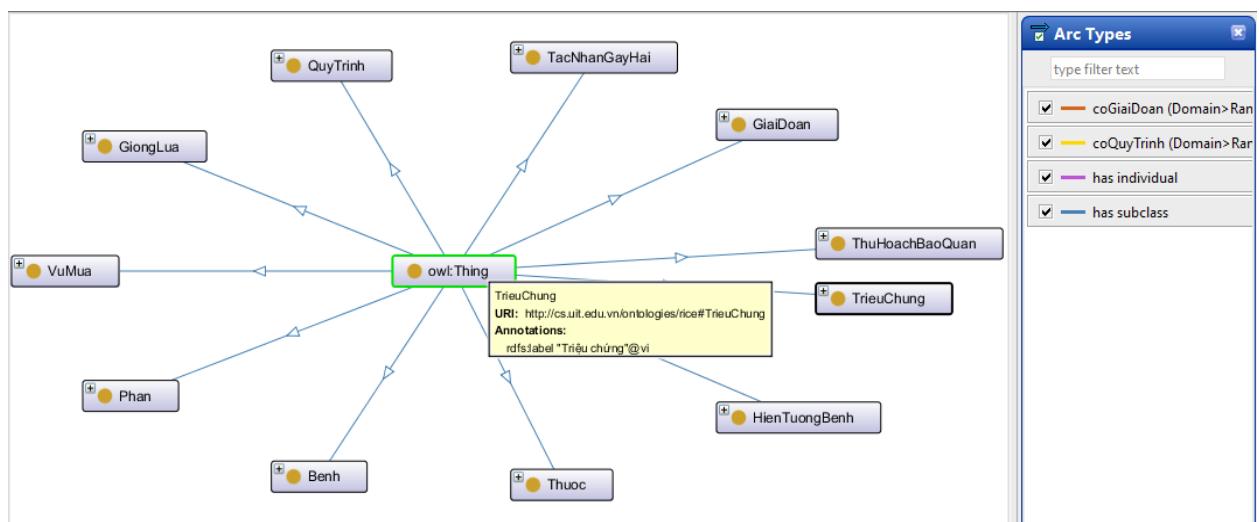
### 3.5.2. Kết quả tổng hợp được sau khi xây dựng Ontology

Ontology lúa nước mà nhóm xây dựng được cài đặt trên hệ thống phần mềm protégé được trình bày như trên, ngôn ngữ sử dụng để quản lý ontology là OWL. Kết quả sau khi xây dựng Ontology trên protégé như sau:

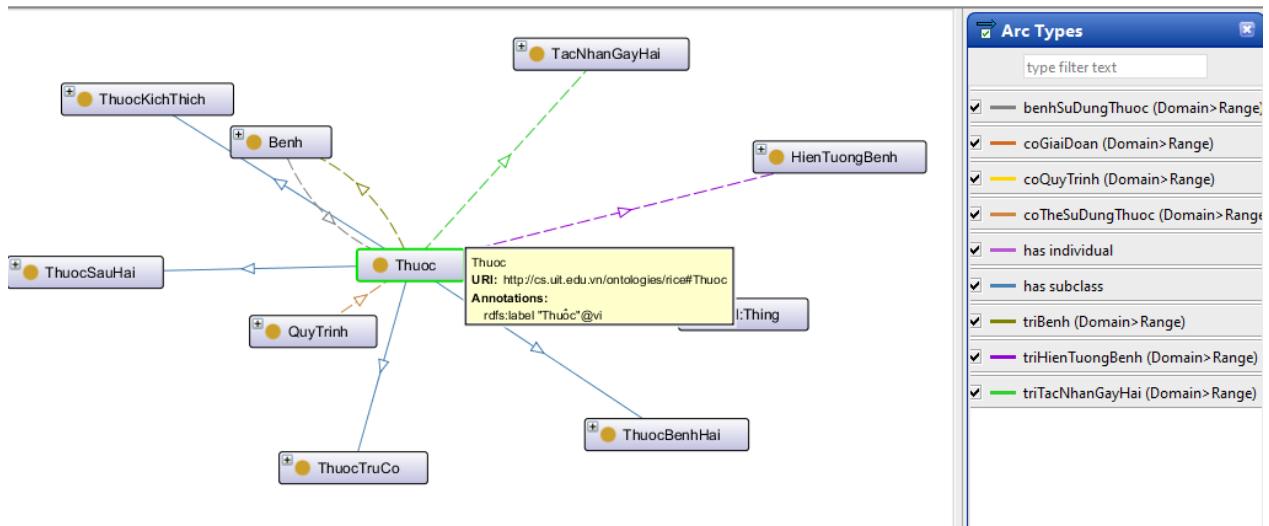
Ontology metrics:	
Metrics	
Axiom	1426
Logical axiom count	1052
Declaration axioms count	289
Class count	47
Object property count	6
Data property count	30
Individual count	203
Annotation Property count	2
DL expressivity	ALU(D)
Class axioms	
SubClassOf	35
EquivalentClasses	0
DisjointClasses	0
GCI count	0
Hidden GCI Count	0

Hình 11: Kết quả xây dựng Ontology trên protégé

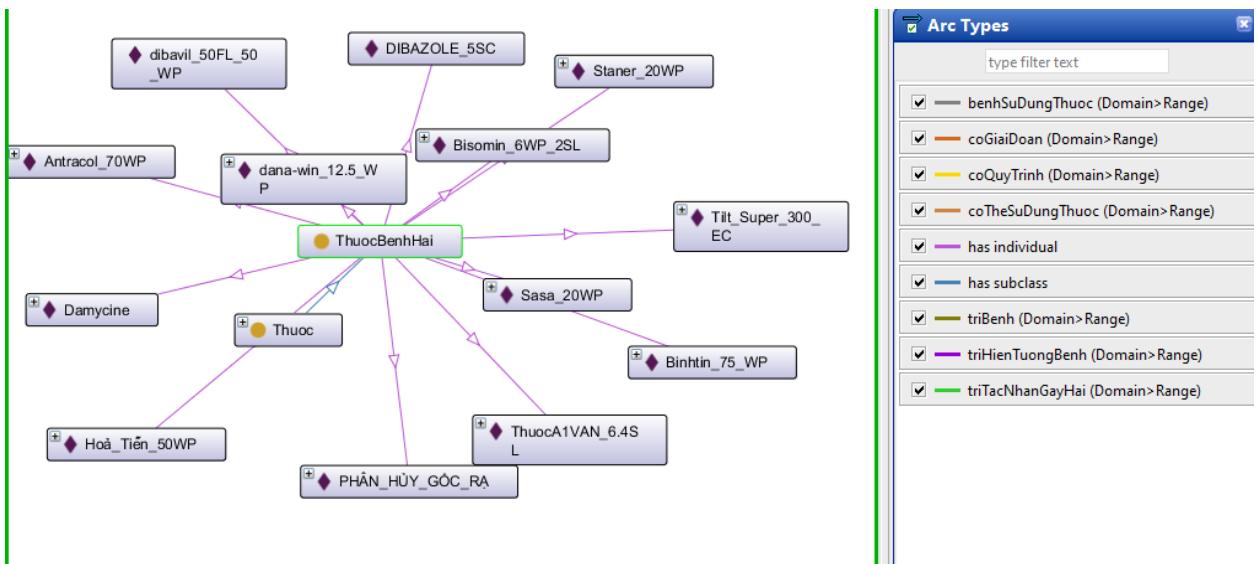
Khóa luận đã xây dựng được một Ontology bao gồm 47 lớp trong đó có 35 lớp con và 12 lớp cha, 10 mối quan hệ và trên 203 thể hiện của các lớp thực thể và 1426 tiền đề được xây dựng. Dưới đây là một số mô phỏng kết quả xây dựng Ontology:



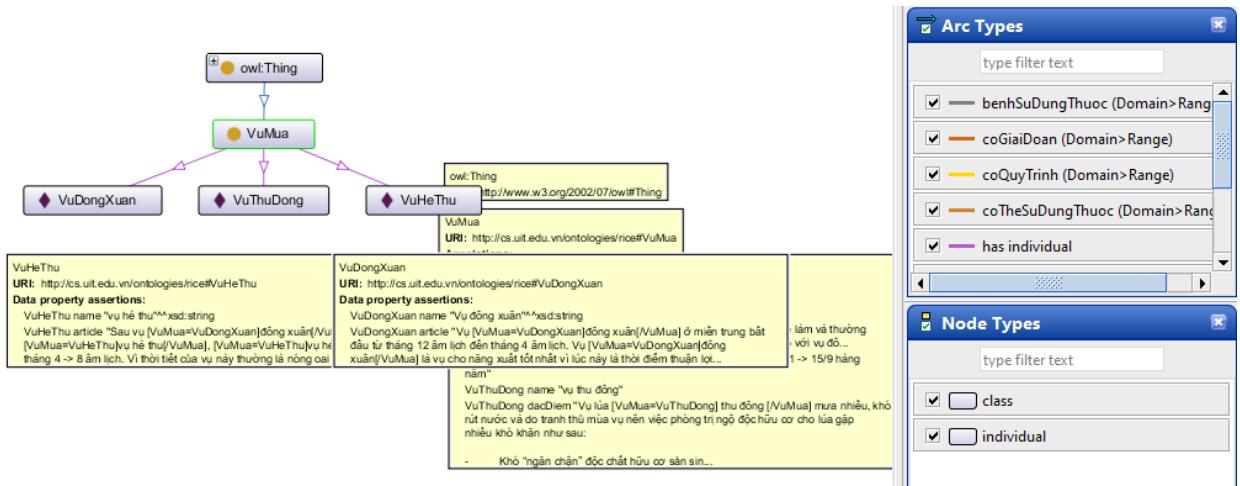
Hình 12: Phân cấp lớp trong Ontology ứng dụng



Hình 13: Ví dụ quan hệ được xây dựng giữa lớp Thuốc với các lớp khác



Hình 14: Thể hiện của lớp Thuốc Bệnh Hại



Hình 15: Xây dựng thuộc tính của các lớp

## CHƯƠNG 4. THIẾT KẾ BỘ SUY DIỄN

### 4.1. PHÁT BIỂU VÀ MÔ HÌNH HOÁ BÀI TOÁN CHUẨN ĐOÁN

Trong quá trình tìm hiểu yêu cầu của người nông dân nhóm nhận thấy được một số yêu cầu của người dùng như sau:

- Biết được những triệu chứng → bệnh, cỏ, ốc, sâu bệnh hại
- Bệnh hại → thuốc trị bệnh
- Sâu hại → thuốc diệt sâu hại
- Tên loại giống lúa → cách bảo quản
- Nhập vào giai đoạn → Hướng dẫn trồng lúa ở từng giai đoạn
- Lúa hiện đang ở giai đoạn → những bệnh có khả năng có
- Nhập vào sâu bệnh → thiên địch
- Nhập vào bệnh hại → thiên địch

Từ những bài thu thập ở trên nhận thấy các dạng bài toán như trên tổng quát có dạng như sau:

Nhập giả thiết → Kết quả

Chính vì vậy có thể phát biểu bài toán này dưới dạng một bài toán tổng quát như sau:

Cho trước một CSTT Ontology ARK, một cách tổng quát, mỗi bài toán có thể được mô hình hóa gồm các thành phần:

(O, Facts, Goals)

Trong đó:

- O: là một tập hợp các khái niệm được quan tâm trong bài toán, mỗi khái niệm được xác định bởi lớp mà đối tượng đó thuộc về.
- Facts: Tập hợp các sự kiện giả thiết thuộc loại sự kiện, được phân loại ở mô hình CSTT. Mỗi sự kiện thể hiện một tính chất của đối tượng.
- Goals: Tập sự kiện kết luận của bài toán.

Giả sử cho trước một bộ (O, Facts, Goals) yêu cầu tìm quá trình suy diễn và kết quả suy diễn để đạt được các sự kiện trong Goals từ tập sự kiện giả thiết Facts dựa trên ARK đã có sẵn.

Ví dụ bài toán suy diễn được phát biểu dưới dạng ngôn ngữ tự nhiên như sau: Người nông dân phát hiện trên cánh đồng xuất hiện trên cây lúa một số triệu chứng: lá vàng, thân ún, rễ thối, có óc, và người nông dân muốn tìm bệnh trên cây lúa. Áp bài toán vào mô hình hóa bài toán suy diễn tổng quát trên ta được các thành phần như sau: (1) O = {Bệnh, Óc}, (2) Facts = {MauLa = LaVang, HinhdangThan = ThanUn, HinhdangRe = ReThoi, Oc = XuatHien}, (3) Goals = {Bệnh = ? }.

## 4.2. Ý TƯỞNG GIẢI QUYẾT BÀI TOÁN

- Nhận yêu cầu Input từ người dùng
- Dựa vào yêu cầu Input của người dùng tiến hành xếp hạng luật, việc xếp hạng luật ở đây nhóm sử dụng một số thủ tục của heuristic. Sắp hạng những luật có khả năng cao nhất thì độ ưu tiên cao.
- Dựa vào Input người dùng nhập vào thực hiện việc kiểm tra có biến ngôn ngữ môi trường nào được định nghĩa trong tập mờ được sử dụng không, nếu tồn tại thực hiện việc mờ hóa.
- Sau khi xếp hạng luật thực hiện việc suy diễn tiến với những sự kiện Input nhập vào bằng cách do tìm các luật có thể áp dụng được. Quá trình suy diễn kết thúc khi dò tìm hết 1 lượt và vẫn không tìm thấy kết quả hoặc nếu tìm thấy kết quả và lượng giá kết quả trả về.
- Nếu quá trình suy diễn đã duyệt tất cả các luật và không cho ra kết quả thì tiến hành tương tác lại người dùng, bằng cách quay lui (backtracking) tìm những sự kiện có thể có khả năng cho kết quả nhanh nhất để hỏi người dùng. Ở bước này ta thực hiện xếp hạng lại bộ luật một lần nữa, vì sau quá trình suy diễn có nhiều sự kiện được thêm vào. Thực hiện suy diễn lùi để tìm những sự kiện để tương tác với người dùng.
- Sau khi tương tác với người dùng nhận thêm sự kiện từ người dùng, tiến hành quay lại suy diễn tiến để tìm kết quả một lần nữa.
- Nếu quá trình suy diễn lần 2 không cho kết quả thì tiến hành đánh giá, tính toán kết quả phù hợp nhất trả về cho người dùng.

### **4.3. THUẬT GIẢI GIẢI QUYẾT VẤN ĐỀ**

Ta có:

Cơ sở (Base) được áp dụng là: Ontology ARK đã được xây dựng.

Input:

- Tập sự kiện giả thiết người dùng nhập vào (bao gồm triệu chứng, khí hậu, môi trường, bệnh thuốc,...)
- Tập loại sự kiện kết luận người dùng yêu cầu (bệnh, thuốc, giai đoạn, sâu bệnh,...)

Output: tập sự kiện kết luận máy dự đoán phù hợp với yêu cầu người dùng.

#### **Thuật giải tổng quát cho bài toán tra cứu**

**Bước 1:** Nhận vào dạng bài toán người dùng yêu cầu.

**Bước 2:** Nhận Input người dùng nhập vào từ màn hình .

**Bước 3:** Thực hiện xếp hạng luật để đánh độ ưu tiên cho luật dựa vào Input nhập vào.

**Bước 4:** Kiểm tra xem có xuất hiện biến ngôn ngữ không và tiến hành xử lý mờ hóa.

**Bước 5:** Thực hiện thuật giải suy diễn tìm kết quả với những sự kiện người dùng nhập vào, nếu tìm được kết quả thì trả kết quả về cho người dùng tiến hành tính % độ tin cậy của kết quả trả về. Tính độ phân trăm kết quả trả về bằng cách quay lui tất cả các nhánh luật đã đi qua để đến kết quả cuối cùng.

**Bước 6:** Nếu ở bước 4 không có kết quả trả về thì tìm những sự kiện thiếu ở những luật có thể áp dụng và quay lại hỏi người dùng, nếu người dùng trả lời Có thì thêm sự kiện vào tập sự kiện đã biết. Ngược lại nếu sự kiện người dùng trả lời là Không thì tiến hành loại bỏ các luật mà chứa sự kiện đó, nếu người dùng trả lời Không biết thì vẫn giữ lại luật đó và tiến hành hỏi tiếp, sau khi hỏi ta thu được 1 tập sự kiện đã biết mới và tiến hành thực hiện suy diễn lại. Nếu xuất hiện kết quả trả về thì qua bước 7.

**Bước 7:** Nếu vẫn chưa có kết quả thì dự đoán những kết quả có thể trả về bằng các tính toán % có thể trả về cho người dùng.

##### **4.3.1.1. Xếp hạng luật**

Mục đích của việc xếp hạng luật, đánh những luật có khả năng để suy diễn ra kết quả một cách nhanh nhất, tiết kiệm chi phí.

Cho trước:

RelationRules là danh sách luật quan hệ

LogicRules là danh sách luật tường minh đã được tối ưu (được đề cập ở chương 3)

**Bước 1:** Khởi tạo

UnsortedRules := RelationRules  $\cup$  LogicRules;

Rules := [] // danh sách luật sau khi xếp hạng

**Bước 2:** Duyệt từng luật UnsortedRules và tiến hành xếp hạng

foreach (rule in UnsortedRules) {

**Bước 2.1** Nếu luật có cả GT và KL thuộc trong tập kiểu các đối tượng (Facts, Goals) của bài toán thì sẽ ưu tiên xếp hạng các luật này lên hàng đầu.

**Bước 2.2** Nếu luật chỉ có KL thuộc trong tập kiểu các đối tượng (Goals) của bài toán thì ưu tiên xếp hạng luật lên hàng giữa.

**Bước 2.3** Nếu luật không thuộc trong tập kiểu các đối tượng của bài toán thì có độ ưu tiên thấp nhất và xếp hạng cuối cùng.

}

#### 4.3.1.2. Xử lý mờ hóa cho các sự kiện là biến ngôn ngữ

Bước 1: Xét các sự kiện người dùng nhập vào kiểm tra xem đã xuất hiện những biến ngôn ngữ nào lưu vào các sự kiện là biến ngôn ngữ.

Bước 2: Lấy giá trị của biến ngôn ngữ được người dùng nhập vào

Bước 3: Đọc dữ liệu của biến ngôn ngữ từ file “**fuzzy-model.txt**”

Bước 4: So khớp giá trị của biến ngôn ngữ nhập vào với dữ liệu của biến ngôn ngữ và đưa ra giá trị vật lý của biến ngôn ngữ nhập vào.

Bước 5: Xuất kết quả trả về cho sự kiện sau khi mờ hóa

**Ví dụ xử lý mờ hóa:** Người dùng nhập vào nhiệt độ là 27 độ C

Bước 1: Biến ngôn ngữ là **NhiệtDo**

Bước 2 : giá trị của biến ngôn ngữ nhập vào là 27 độ C

Bước 3: Đọc dữ liệu nhiệt độ từ file “**fuzzy-model.txt**” như sau:

```

variable NheitDo "độ"
begin
    term NheitDoThap : trimf(0, 10, 25);
    term NheitDoCao : trimf(10, 25, 30);
end;

```

Bước 4:  $27 > 20 \&& 27 < 30 \rightarrow$  Nhiệt độ cao

Bước 5: NheitDo=NheitDoCao

#### 4.3.1.3. Thuật giải suy diễn tìm kết quả

**Bước 1:** Khởi tạo

Đặt trạng thái ban đầu cho một số biến điều khiển như sau:

canAskOneTime := true; // trong trường hợp không tìm ra KL, cho phép máy hỏi người dùng và chỉ cho hỏi tối đa một lần

unknownFacts := []; // tập lưu những sự kiện mà máy hỏi và người dùng trả lời Không biết

inferredRules := []; // tập luật tường minh đã được áp dụng

inferredRelations := []; // tập luật quan hệ đã được áp dụng

Goals := []; // tập sự kiện kết quả

Known := []; // tập những sự kiện đã biết

ReliableKnown := []; // tập những sự kiện đã biết, chỉ chứa những sự kiện người dùng nhập mà không chứa những sự kiện mới sinh ra, là tập con của Known

AddFactsToKnown(true, Input); // gán sự kiện người dùng nhập vào Known

**Bước 2:** Thực hiện một quá trình dò tìm cách suy diễn tính toán bằng cách sử dụng bộ luật ưu tiên đã được xếp hạng và các bước giải cơ bản để phát sinh hay suy ra sự kiện mới.

while (true) {

hasNewFacts = false; // biến kiểm tra xem có sự kiện mới được sinh ra hay không

foreach (rule in Rules) { // duyệt trong tập luật đã được xếp hạng

if (rule is RelationRule) { // nếu là luật quan hệ

foreach (fact in Known) {

```

        if (tập giả thiết của rule chứa fact) {
            hasNewFacts = InferRelationRule(fact, rule); // suy diễn
            luật quan hệ
            if (hasNewFacts) break; // nếu có sinh sự kiện mới thì thoát
            khỏi vòng foreach
        }
    }
else { // rule là tường minh
    hasNewFacts = InferLogicRule(rule); // suy diễn luật tường minh
    if (hasNewFacts) {
        Xóa rule khỏi tập Rules
        break; // Thoát khỏi vòng foreach
    }
}

```

```

If (!hasNewFacts) continue; // nếu chưa có sự kiện mới thì tiếp tục lặp
}
break; // thoát khỏi vòng while
}

```

```

if (tập Goals có chứa kết quả) {
    results := []; tập tuple chứa sự kiện kết quả và độ tin cậy đi kèm với kết quả
    foreach (fact in Goals) {
        Độ tin cậy := FindBestPath(fact, ReliableKnown, InferredRules); // Tìm đường
        đi suy diễn tốt nhất dẫn đến sự kiện fact
        Thêm tuple (fact, Độ tin cậy) vào results;
    }
    return results; // Trả về kết quả suy diễn Bước 5
}

```

**Bước 3:** Khi không tìm được kết quả, tìm sự kiện để hỏi người dùng bằng cách lọc chọn luật sinh ra được sự kiện kết quả.

```
if (canAskOneTime) { // Kiểm tra xem máy có thể hỏi thêm sự kiện mới hay không?  
    canAskOneTime := true;  
    maxRulesToAsk := 3; // chọn tối đa 3 luật để hỏi người dùng  
    guessableFacts := FindFactsToAsk(3); // những sự kiện được dùng để hỏi  
    foreach (fact in guessableFacts) { // Xét từng sự kiện, người dùng sẽ trả lời: Có, Không  
        và Không biết.  
        if (người dùng trả lời Có)  
            AddFactsToKnown(true, fact); // thêm sự kiện vào tập Known.  
        else if (người dùng trả lời Không)  
            Xoá những luật có sự kiện này và không xét đến nữa;  
        else người dùng trả lời không biết và thêm sự kiện vào tập unknownFacts  
    }  
    goto bước 2;  
}
```

**Bước 4:** Hệ thống vẫn chưa tìm ra kết quả, tiến hành suy diễn không đầy đủ

```
incompleteResults := InferIncompleteRules(); // tập sự kiện kết quả sau khi suy diễn  
trên bộ luật thiếu sự kiện giả thiết
```

**Bước 5:** Trả kết quả suy diễn về người dùng

#### 4.3.1.4. Một số thuật giải suy diễn phụ cho bài toán suy diễn

Thuật giải 1: Đếm số lượng sự kiện tồn tại trong Known

```
IsFactInKnown(Fact) {  
    Nếu Fact tồn tại trong Known, trả về true;  
    Không thì trả về false;  
}
```

```
CountFactsInKnown(List facts) {
```

```

count := 0;
foreach (fact in facts) {
    if (IsFactInKnown(fact)) count++;
}
return count;
}

```

Thuật giải 2: Thêm sự kiện vào tập Known và ReliableKnown

```
AddFactsToKnown(bool reliable, List facts) { // reliable cho biết đây là những sự kiện người
dùng nhập
```

```

count := 0; // số lượng sự kiện được thêm vào Known
tmpFacts := []; // danh sách sự kiện sẽ được thêm vào Known
foreach (fact in facts) {
    if (fact là sự kiện mờ) Mờ hoá sự kiện và thêm vào tập tmpFacts;
    else Thêm fact vào tập tmpFacts; // và fact là sự kiện
}
foreach (fact in tmpFacts) {
    if (fact chưa tồn tại trong Known) {

```

```

        Thêm fact vào Known;
        if (reliable) Thêm fact vào ReliableKnown;
        count++;
    }
}
```

Kiểm tra xem fact có thể là sự kiện kết quả hay không, nếu có thì thêm fact vào tập Goals.

```

    }
}
return count;
}
```

Thuật giải 3: Suy diễn luật tường minh và luật quan hệ

```
InferLogicRule(rule) {
```

```

        if (rule tồn tại trong inferredRules) trả về false;
        if (luật chưa đủ sự kiện giả thiết để áp dụng) trả về false;
        Áp dụng luật, Thêm luật vào inferredRules;
        count := AddFactsToKnown(false, tập sự kiện KL của rule); // Đếm số sự kiện mà máy
suy diễn từ luật
        return count > 0;
    }

InferRelationRule(fact, rule) {
    Kiểm tra trong inferredRelations, xem đã áp dụng tuple luật quan hệ (fact, rule) chưa,
nếu có trả về false;
    hasNewFacts := false;
    individuals := Dựa trên (fact, rule), duyệt ontology lấy về danh sách các thể hiện trên
quan hệ.
    foreach (individual in individuals) {
        logicRule := Tạo luật tường minh mới dựa trên fact và individual.
        if (InferLogicRule(logicRule)) hasNewFacts := true; // Suy diễn trên luật tường
minh này
    }
    Thêm tuple (fact, rule) vào inferredRelations;
    return hasNewFacts;
}

```

Thuật giải 4: Tìm đường đi suy diễn tốt nhất dẫn đến sự kiện mục tiêu

```

FindPaths(fact, knownFacts, rules) {
    paths := []; danh sách các đường đi
    inferrableRules := Tìm trong rules những luật có thể sinh ra fact;
    foreach (rule in inferrableRules) {
        previousPaths := []; // những đường đi đã biết
        newRules := rules \ rule; // tập luật mới đã bỏ đi rule;

```

```

foreach (h in tập sự kiện GT của rule) {
    if (knownFacts chúa h) continue;
    tmpPreviousPaths = FindPaths(h, knownFacts, newRules); // Tìm đường
đi sinh ra sự kiện giả thiết h;
    previousPaths := previousPaths ∪ tmpPreviousPaths
}
if (previousPaths khác rỗng) {
    foreach (previousPath in previousPath) {
        newPath := previousPath.append(rule) // nhánh đường đi mới
        Thêm newPath vào paths
    }
}
else { // Luật đã áp dụng được mà không cần quay lui trên sự kiện giả thiết
    Thêm rule vào paths;
}
}

FindBestPath(fact, knownFacts, rules) {
    paths := FindPaths(fact, knownFacts, rules); // Tìm những đường đi sinh ra fact
    Nếu paths không tồn tại trả về rỗng;
    Tìm đường đi có độ tin cậy cao nhất, trả về đường đi này bằng cách nhân mỗi độ tin
cậy của luật trong đường đi với nhau.
}

```

Thuật giải 5: Suy diễn luật không đầy đủ

```

InferIncompleteRules() {
    results := []; // Tập sự kiện kết quả
    rules := Danh sách luật có thể cho ra sự kiện kết quả;
    foreach (rule in rules) {

```

```

hypothesesInKnown := CountFactsInKnown(tập GT của rule); // đếm số sự kiện
GT tồn tại trong tập Known;
    if (hypothesesInKnown = 0) continue;
    newHypotheses := Tập sự kiện GT đã tồn tại trong Known;
    newCertaintyFactor := Độ tin cậy của luật * (hypothesesInKnown / số sự kiện
GT của luật); // Độ tin cậy mới của luật không đầy đủ
    incompleteRules := Tạo sự kiện tường minh mới dựa trên newHypotheses, tập
sự kiện kết luận của rule và newCertaintyFactor;
    newInferredRules := inferredRules ∪ incompleteRules; // Bộ luật tường minh
đã được áp dụng mới
    conclusions := Tập sự kiện kết luận của luật, mà có thể là sự kiện mục tiêu của
bài toán.

    foreach (c in conclusions) {
        var path := FindBestPath(c, ReliableKnown, newInferredRules); // Tìm
đường đi suy diễn tốt nhất dẫn đến sự kiện c
        if (c tồn tại trong results và độ tin cậy của path > độ tin cậy trong results)
            results[c] = path;
        else Thêm tuple (c, path) vào results;
    }
}
return results;
}

```

#### 4.3.1.5. Ví dụ minh họa thuật giải

Ví dụ bài toán suy diễn được phát biểu dưới dạng ngôn ngữ tự nhiên như sau: Người nông dân phát hiện trên cánh đồng xuất hiện trên cây lúa một số triệu chứng: lá vàng, thân ún, rễ thối, có óc, và người nông dân muốn tìm bệnh trên cây lúa. Áp bài toán vào mô hình hóa bài toán suy diễn tổng quát trên ta được các thành phần như sau: (1) O = {Bệnh, Óc}, (2) Facts =

{MauLa = LaVang, HinxDangThan = ThanUn, HinxDangRe = ReThoi, Oc = XuatHien}, (3)

Goals = {Bệnh = ? }.

Gia sử cho trước bộ luật

R1 : Benh -> Thuoc;

R2: Thuoc -> TacNhanGayHai;

R3: MauLa=LaVang & HinxDangThan=ThanUn → Benh=BenhThoiThan

R4: GiaiDoan=GiaiDoanLamDong & LuongMua=MuaNhieu→

HinxDangThan=ThanGay & HinxDangBong=KhongTroBong;

R5: GiaiDoan=GiaiDoanLamDong & MoiTruong=NangNhieu &

NuocRuong=KhongNuoc → Benh=BenhVangLa;

Bước 1 : Nhận yêu cầu từ người dùng

Bước 2: Input = { MauLa = LaVang, HinxDangThan = ThanUn, HinxDangRe = ReThoi, Oc = XuatHien, Benh=? }, Ouput = { Benh=? }

Bước 3: Bộ luật sau khi xếp hạng như sau

R3: MauLa=LaVang & HinxDangThan=ThanUn → Benh= BenhThoiThan;

R5: GiaiDoan=GiaiDoanLamDong & MoiTruong=NangNhieu &

NuocRuong=KhongNuoc → Benh=BenhVangLa;

R4: GiaiDoan=GiaiDoanLamDong & LuongMua=MuaNhieu→

HinxDangThan=ThanGay & HinxDangBong=KhongTroBong;

R1 : Benh -> Thuoc;

R2: Thuoc -> TacNhanGayHai;

Bước 4: Không xuất hiện biến ngôn ngữ mờ nên bỏ qua bước mờ hóa

Bước 5: Thực hiện suy diễn tìm kết quả

Known = { MauLa = LaVang, HinxDangThan = ThanUn, HinxDangRe = ReThoi, Oc = XuatHien }

Duyệt qua 5 luật đã xét hạng ở trên: R3 áp dụng được GT của R3 ∈ Known , thêm sự kiện Benh=BenhThoiThan vào Known. Đến luật R5 và R4 không phù hợp. Xét luật R1 ta thấy sự kiện GT R5 là Benh ∈ Known, nhưng vì R1 là một dạng luật quan hệ nên ta vào ontology lấy dữ liệu giá trị quan hệ về. Tìm tới quan hệ Bệnh được chữa bởi thuốc (Bệnh ->

Thuốc), tìm những thuốc mà trị bệnh BenhThoiThan đưa vào Known. Tương tự cho luật cuối cùng ta tìm được SauCuonLa. Tập Known cuối cùng là Known = { MauLa = LaVang, Than = ThanUn, Re = ReThoi, Oc = XuatHien, Thuoc = Anpha, Sau = SauCuonLaNho, Benh = BenhThoiThan }

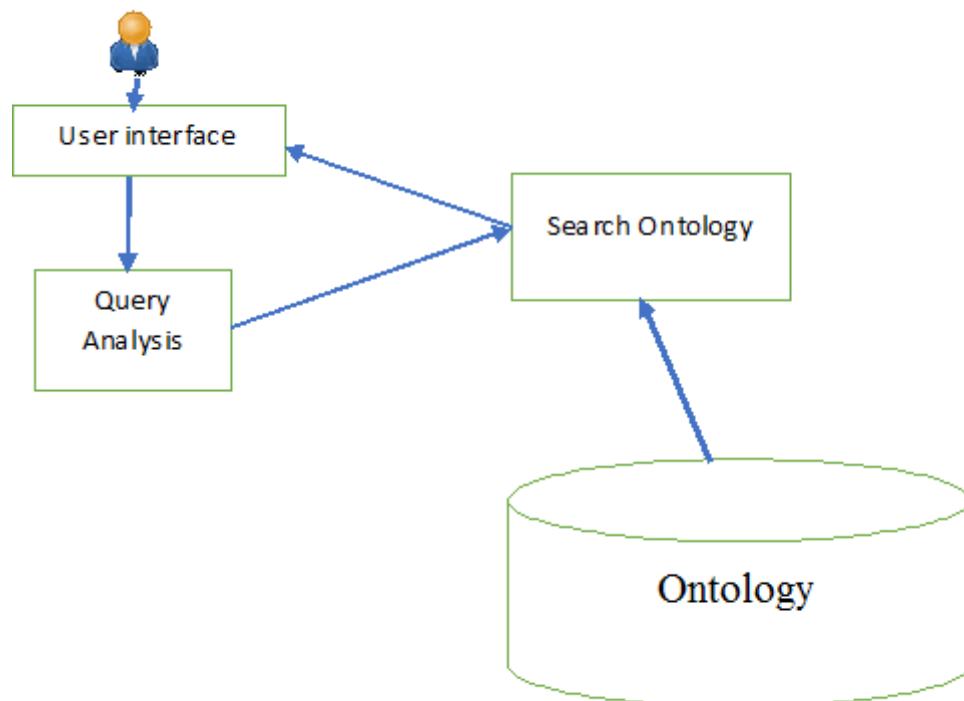
Bước 6: tìm được kết quả, xuất kết quả trả về cho người dùng là Benh=BenhThoiThan

## CHƯƠNG 5. THIẾT KẾ BỘ TRA CỨU - TÌM KIẾM

### 5.1. TRA CỨU TRÊN CƠ SỞ TRI THỨC – DATA RETRIEVAL

#### 5.1.1. Mô hình hóa bài toán

Tra cứu thông tin là một cách tìm thông tin dựa trên một truy vấn (người sử dụng yêu cầu) để đưa ra kết quả trả về tài liệu phù hợp với yêu cầu của người sử dụng. Có 2 dạng định hướng chính mà nhóm muốn hướng đến cho người dùng: (1) định hướng tìm kiếm theo cây tìm kiếm, (2) định hướng tìm kiếm theo cụm từ. Đối với định hướng (1) được sử dụng cho người dùng tìm hiểu tổng quát về miền tri thức lúa nước, đối với dạng tra cứu (2) đối với tìm kiếm dạng cụm từ bất kỳ, hệ thống tiến hành tìm kiếm trong ontology những cụm từ trùng khớp và trả về cho người dùng danh sách các class, individual liên quan đến câu truy vấn tìm kiếm, mỗi dòng trả về là link đến một bài viết được tổ chức trong ontology.



Hình 16: Bài toán tra cứu trên Ontpology

### **5.1.1.1. Bài toán 1: Tìm kiếm định hướng dạng cây**

Tra cứu định hướng theo hệ thống phân cấp khái niệm, đây là cách tra cứu đơn giản nhất dành cho người dùng. Hệ thống cung cấp cấp một sơ đồ tra cứu theo cấu trúc cây phân cấp khái niệm, cho phép người sử dụng duyệt trực tiếp trên cấu trúc này để tìm kiếm thông tin mình cần. Có thể duyệt tuần tự từ những khái niệm tổng quát cho đến những thực thể trong lĩnh vực. Nếu ở cấp độ khái niệm hệ thống trả về danh sách các individual của khái niệm đó. Nếu ở cấp độ thực thể thì bài viết trả về 1 trang thông tin có cấu trúc của thực thể được tìm kiếm.

### **5.1.1.2. Bài toán 2: Bài toán tìm kiếm với cụm từ**

Nội dung tìm kiếm được đặc tả dưới dạng cụm từ. Dựa vào câu truy vấn người dùng nhập vào hệ thống tiến hành so khớp với Ontology ARK và trả về danh sách các term có nghĩa từ câu truy vấn nhập vào. Tìm các bài viết liên quan đến các term và trả về bài viết có liên quan. Các dạng trả về khi tra cứu:

- Nếu term biểu thị cho một khái niệm thì hệ thống trả về danh sách các cá thể thuộc về khái niệm, kèm theo tương ứng mỗi cá thể là một liên kết (link) đến trang thông tin của cá thể đó.
- Nếu term tìm kiếm biểu thị cho một cá thể thì hệ thống trả về trang thông tin của cá thể đó.

## **5.1.2. Xây dựng giải pháp cho bài toán tra cứu**

### **Giải pháp tìm kiếm định hướng dạng cây**

Đối với bài toán định hướng tìm kiếm thuật giải quá trình này chủ yếu là cách truy xuất thông tin từ OWL. Trích xuất dữ liệu từ OWL bằng thư viện Java OWL API. Hệ thống xây dựng bằng C# trên nền ASP.NET nên việc đọc dữ liệu từ file owl khá khó khăn, vì vậy nhóm quyết định xây dựng chương trình đọc, trích xuất dữ liệuOWL bằng ngôn ngữ Java, sau đó tạo một kết nối socket truyền nhận giữ liệu với hệ thống.

Sau khi đã xây dựng được một Hệ thống chạy trên nền web, các web crawlers (GoogleBot, Bingbot, Slurp Bot, DuckDuckBot,...) sẽ dịch thông tin từ các tập tin HTML. Để giúp những tác tử này hiểu được ngữ nghĩa của của tập tin, ta cần sử dụng 1 chuẩn cấu trúc để bổ sung vào HTML. Một chuẩn phổ biến được W3C khuyến nghị là RDFa, dựa trên chuẩn RDFa Lite.

### Giải pháp tìm kiếm cụm từ

**Bước 1 :** Ghi nhận cụm từ người dùng nhập vào

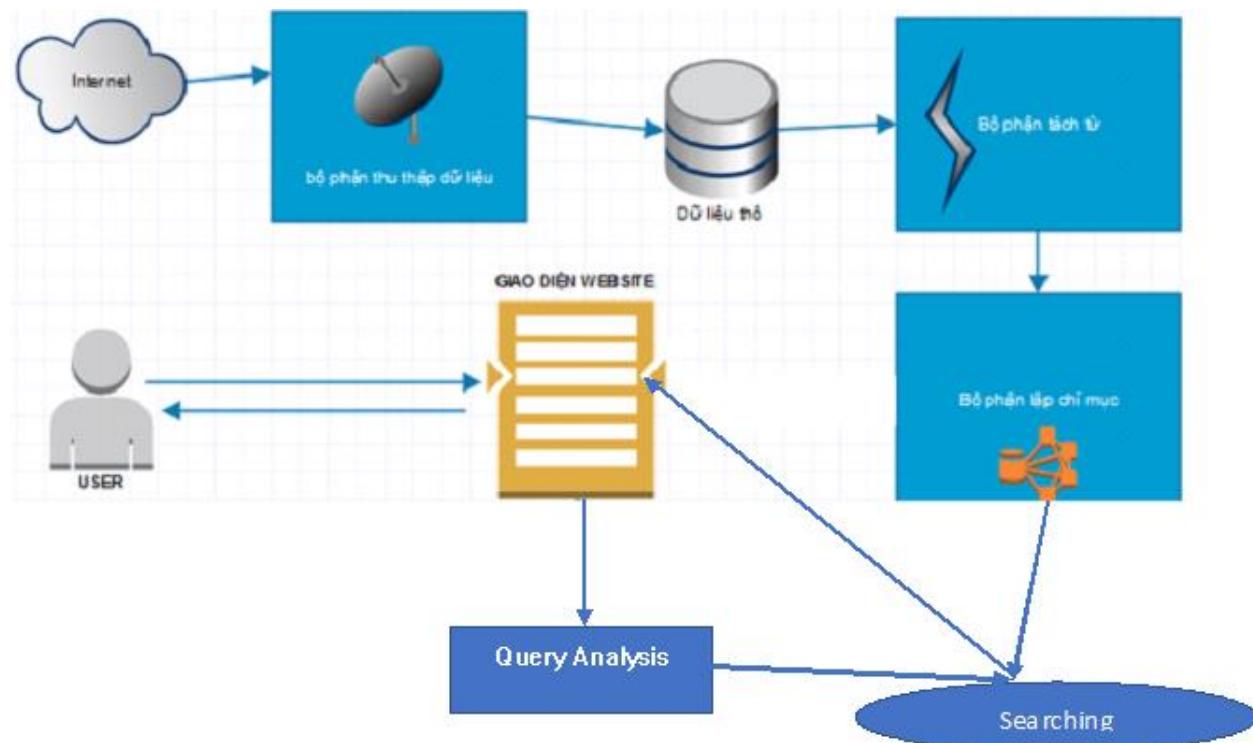
**Bước 2:** Kiểm tra cụm từ nhập vào thành phần nào trong Ontology ARK

**Bước 3:** + Nếu cụm từ nhập vào là một Individual thì trả về bài viết của individual đó.

+ Nếu cụm từ nhập vào là một class thì trả về danh sách các Individual của class.

**Bước 4:** trả về kết quả tìm kiếm cho người dùng bằng cách trích xuất dữ liệu trả về cho người dùng.

## 5.2. TÌM KIẾM TRÊN KHO TIN BÀI – DOCUMENT RETRIEVAL



Hình 17: Quy trình tổng quát tìm kiếm thông tin trên kho dữ liệu

- Bộ thu thập dữ liệu: Là một module trong hệ thống IR có chức năng tự động thu thập các tài liệu (văn bản, hình ảnh, video, âm thanh, ...) cho hệ thống.
- Bộ tách từ (Indexing): Rút trích những đơn vị thông tin trong tài liệu (từ, cụm từ, khái niệm) và biểu diễn tài liệu để máy tính có thể hiểu được.
- Bộ tìm kiếm (Searching): Truy xuất kho lưu trữ (cơ sở dữ liệu) để trả về danh sách các kết quả có liên quan (relevant) dựa vào câu truy vấn của người dùng.

### **5.2.1. Mô hình hóa bài toán tìm kiếm trên kho tin bài**

Bài toán tổng quát trong hệ tìm kiếm các tin bài được mô tả như sau: giả thiết là 1 câu truy vấn (query) Q và một tập các văn bản (Document) D chứa các nội dung đã thu thập, kết luận của bài toán là các văn bản có nội dung tương quan với câu truy vấn Q được trả về và có thể có sự sắp xếp nhất định về sự tương đồng giữa nội dung văn bản và câu truy vấn bằng cách so khớp các từ có trong câu truy vấn Q và các văn bản D. Hơn nữa, bài toán có thể trả về kết luận có tương đồng về ngữ nghĩa với câu truy vấn Q, không nhất thiết văn bản kết quả phải chứa từ ngữ giống hoàn toàn câu truy vấn Q.

**Input:**

- Tập D gồm n tài liệu và câu truy vấn (query) q của người dùng. Câu truy vấn được diễn đạt dưới dạng ngôn ngữ tự nhiên. Sau đây là một số ví dụ về câu truy vấn:
  - q = tìm bệnh đạo ôn của lúa
  - q = bệnh vàng lá
  - q = bệnh đạo ôn, vàng lá
  - q = quy trình tưới nước

**Output:**

Danh sách các tài liệu có liên quan đến câu truy vấn q, được xếp hạng theo thứ tự mức độ liên quan giảm dần.

Mức độ liên quan: Giá trị đánh giá sự tương đồng giữa tài liệu và truy vấn.

**Ý tưởng giải quyết**

Để giải quyết bài toán tra cứu trên kho tài liệu ta giải quyết vấn đề sau nhóm quyết định sử dụng mô hình vector space model kết hợp với túi từ. Các tài liệu và truy vấn được biểu diễn thành các vector tương ứng, mỗi chiều tương ứng với trọng số của 1 term (từ hoặc cụm từ). Tuy nhiên để giảm không gian lưu trữ, nhóm quyết định biểu diễn mỗi tài liệu thành 1 túi từ, trong túi từ chỉ chứa những từ đã được đánh trọng số.

Đầu tiên ta xây dựng cấu trúc dữ liệu để lưu trữ kho tài liệu D, và câu truy vấn q. Trong tìm kiếm dựa trên mô hình không gian vector, mỗi một tài liệu d thuộc D hay câu truy vấn q đều được biểu diễn dưới dạng một vector như sau:

$$\vec{d} = (w_{df_1}, w_{df_2}, w_{df_3}, w_{df_4}, \dots)$$

Trong đó:  $w_{df_1}$  : là trọng số của đặc trưng  $f_1$  trong d

$$\vec{q} = (w_{qf_1}, w_{qf_2}, w_{qf_3}, w_{qf_4}, \dots)$$

Trong đó:  $w_{qf_1}$  : là trọng số của đặc trưng  $f_1$  trong q

Vậy trọng số đó được ước lượng (tính) như thế nào? Ta sử dụng công thức

$$wd = tf(f, d) * idf(f)$$

Với các thông số sau:

$$tf = \frac{f}{max(n, f)}$$

Trong đó:

- $tf$  là tần số xuất hiện của term trong tài liệu, đánh giá độ quan trọng của term trong tài liệu
- $f$  là số lần xuất hiện của term trong tài liệu d
- $max(n, f)$  là số lần xuất hiện lớn nhất của các term trong một tài liệu.

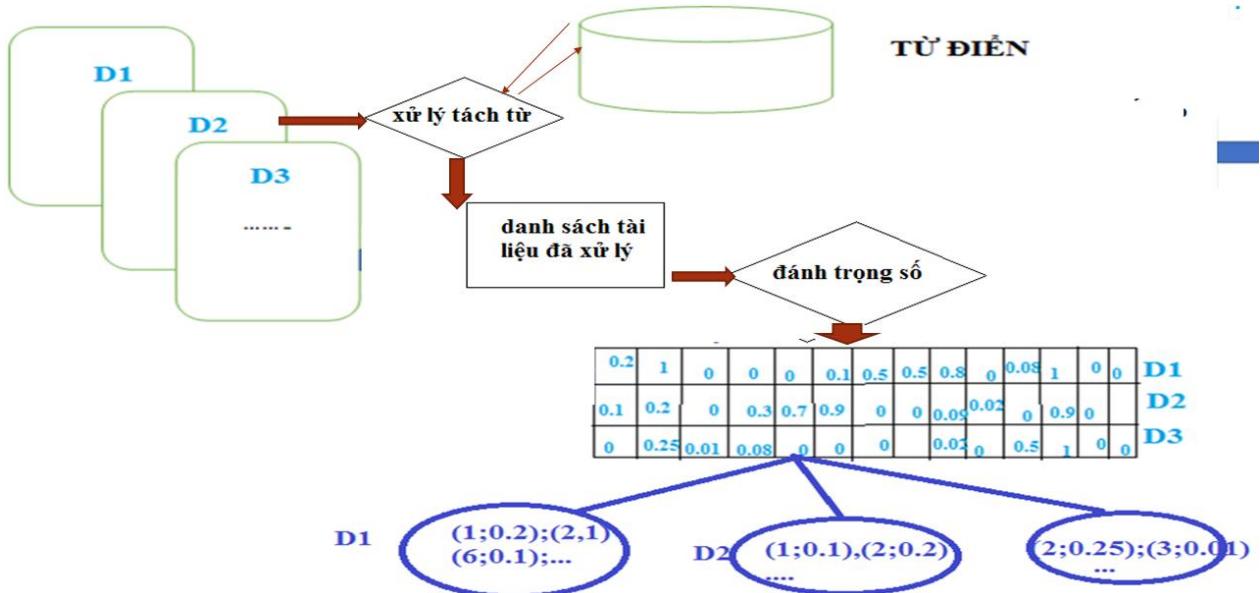
$$idf = \log\left(\frac{N}{N_f}\right)$$

Trong đó:

- N là số tài liệu có trong kho tài liệu D
- $N_f$  là số tài liệu có chứa term đó.

Sau khi tiến hành tách từ cho câu truy vấn tiến hành mở rộng câu truy vấn thành túi từ để kết quả tìm kiếm được đa dạng hơn. Sau đó tiến hành so khớp 2 túi từ với nhau, tìm phần

giao của 2 túi từ sử dụng công thức tính độ tương quan cos để tính được mức độ khớp của câu truy vấn đối với tài liệu.



Hình 18: Quá trình tìm kiếm trong kho tin bài

### 5.2.2. Thu thập lưu trữ kho dữ liệu tin bài

#### 5.2.2.1. Khảo sát và thu thập các tin bài

##### Khảo sát các trang thu thập

Thu thập dữ liệu là bài toán thu thập thông tin từ các trang web và lưu vào CSDL. Phân tích dữ liệu từ các webpage sau đó lấy những thông tin cần thiết. Thông thường, nếu không vì mục đích đặc biệt nào đó, thì các webpage có chức năng như nhau trong cùng một trang báo điện tử được thiết kế dựa trên một mẫu chung nào đó và do đó có giao diện và cấu trúc tương tự nhau. Cấu trúc thông tin của một trang báo điện tử ngoài chứa nội dung chính còn chứa nội dung không liên quan như: các quảng cáo, các thông báo, các liên kết,... Mục đích thu thập tin tức tự động yêu cầu hệ thống cần phân biệt đâu là nội dung quan tâm, cần thu thập đâu là nội dung không liên quan, cần bỏ qua. Tuy nhiên, việc hiểu và phân biệt điều này là phức tạp. Do đó để đơn giản, chính xác và hiệu quả trong việc thu thập tin

túc, hệ thống cần sự hỗ trợ của người sử dụng trong việc định nghĩa các thành phần của một trang báo điện tử hay nói cách khác là cấu trúc thông tin của trang báo điện tử.

**CỔNG GIAO TIẾP ĐIỆN TỬ**  
**NÔNG NGHIỆP & PTNT VĨNH PHÚC**

TRANG CHỦ GIỚI THIỆU TIN TỨC CSDL NÔNG NGHIỆP BÁO CÁO THỐNG KÊ VĂN BẢN THỦ TỤC HÀNH CHÍNH XÂY DỰNG NTM ĐẦU THẦU ĐẦU TƯ CÔNG

Thứ ba, ngày 18/07/2017 Nhập từ khóa tìm kiếm

**Tiêu đề** → **Giai đoạn làm hạt**

**Giai đoạn chín sửa**  
  
 Sau phơi màu 5 - 7 ngày, chất dự trữ trong hạt ở dạng lỏng, trắng như sữa. Hình dạng hạt đã hoàn thành, lúu hạt có màu xanh. Khối lượng hạt tăng nhanh ở thời kỳ này, có thể đạt 75 - 80 % khối lượng cuối cùng.

**Giai đoạn chín sáp**  
  
 Giai đoạn này chất dịch trong hạt dần dần đặc lại, hạt cứng. Màu xanh ở lưng hạt dần dần chuyển sang màu vàng. Khối lượng hạt tiếp tục tăng.

**Giai đoạn chín hoàn toàn**  
  
 Giai đoạn này hạt chắc cứng. Vỏ trấu màu vàng - vàng nhạt. Khối lượng hạt đạt tối đa.

Ngày đăng: 02/08/2016 Tác giả: ST - Số lần được xem: 148

In bài viết

**Tin bài cùng chuyên mục**

**Hình đại diện**  **Noi dung**

- Giai đoạn làm đong 02/08/2016
- Giai đoạn đẻ nhánh 02/08/2016
- Giai đoạn mạ 02/08/2016
- Giai đoạn nảy mầm 02/08/2016
- Thời gian sinh trưởng phát triển của cây lúa 02/08/2016
- Hoa lúa 02/08/2016
- Lá lúa 02/08/2016

**CSDL NÔNG NGHIỆP**

LÚA  
 NGÔ  
 LẠC, ĐẬU ĐỎ  
 RAU  
 HOA  
 CÂY ĂN QUẢ  
 TRÂU, BÒ  
 LỢN  
 GIA CĂM  
 THỦY SẢN  
 VẬT NUÔI KHÁC

**VIDEO** **HÌNH ẢNH**

**Thành tựu đổi mới**

- Đề người dân được sử dụng nông sản an toàn 2
- Triển khai sản xuất vụ Mùa - vụ Đông năm 2014
- Hướng tới hội diễn văn nghệ ngành Nông nghiệp & PTNT năm 2014
- Sở Nông nghiệp & PTNT giàn giới thiệu toàn đoàn tại Liên hoan văn nghệ công nhân, viên chức, lao động tỉnh Vĩnh Phúc năm 2014

Xem tất cả »

**ĐƠN VỊ THUỘC CƠ SỞ**

**LIÊN KẾT WEBSITE**

Chọn website liên kết

**TỔNG LƯỢT TRUY CẬP**

0|0|0|0|2|1|8|8|8

**CỔNG GIAO TIẾP ĐIỆN TỬ SỞ NÔNG NGHIỆP VÀ PHÁT TRIỂN NÔNG THÔN VĨNH PHÚC**

Chủ trách nhiệm: Ông Nguyễn Tiến Phong – Giám đốc Sở Nông nghiệp và PTNT Vĩnh Phúc  
 Trụ sở: Đường Nguyễn Việt Xuân, phường Đồng Đa, TP. Vĩnh Yên, tỉnh Vĩnh Phúc  
 ĐT: 0211.3862516; Fax: 0211.3860683; Email: sonnptnt@vinhphuc.gov.vn

## Hình 19: Nội dung tin bài thu thập ở một webpage nông nghiệp

Một website là tập hợp các webpage liên quan chứa nội dung ví dụ như text, hình ảnh, video, audio,... Một website được lưu trữ trên ít nhất 1 server, có thể truy cập qua mạng, ví dụ như mạng Internet hay mạng cục bộ riêng, bằng một địa chỉ URL (Uniform Resource Locator). Mỗi website có các thuộc tính sau:

- Name: tên website.
- URL: địa chỉ URL của website.
- Tập các webpage thuộc về website

Mỗi webpage là một tài liệu, thông thường được viết bằng văn bản thuận túy xen kẽ với cú pháp của HTML (Hypertext Markup Language). HTML là ngôn ngữ đánh dấu được thiết kế để tạo nên các webpage với các mẩu thông tin được trình bày trên World Wide Web. Ví dụ cấu trúc thông tin một webpage thu thập:

The screenshot shows the source code of a webpage from a browser. The code is primarily in HTML, with some JavaScript and CSS included. Key parts of the code include:

- A script tag at the top containing two scripts: one for "text/javascript" and another for "javascript" type="text/javascript".
- A main container div with id="detail-page" which contains:
  - A title h1: "Bệnh cháy bìa lá".
  - A Facebook sharing section with a div for "fb-root" and a "fb\_reset" button.
  - A script tag with a src attribute pointing to "https://apis.google.com/js/platform.js" and attributes lang="vi" and gapi\_processed="true".
  - A style block defining a social class with a float:right; height:20px; rule.
  - A date div containing the text "Ngày đăng: 26/07/2016".
  - A social div containing a list of social sharing links.
  - A grey horizontal line hr.
  - A dec div containing a list of links.
  - A list su kien div.
  - A tin cung su kien div.
  - An external script section with a div for "ExternalClass32FC26400C7C463BA9AEE6F3DD571E18" and a p align="center" tag.
  - A detailed description of rice blight in Vietnamese, mentioning the pathogen Xanthomonas oryzae, its symptoms (yellowish-green spots), and its impact on rice plants.

Hình 20: Cấu trúc source HTML của một bài viết

Tùy thuộc vào chức năng của mỗi webpage trong website mà chúng có cấu trúc thông tin gồm tập các HTML Element giống hoặc khác nhau. Bên cạnh đó, số loại HTML Element trong một webpage hoặc một website tùy thuộc vào chức năng của website, và mục tiêu thiết kế của hệ thống. Vì vậy khi định nghĩa những thông tin cần lấy trong webpage thì chỉ cần lấy những HTML Element quan tâm, cần thiết cho mục tiêu của hệ thống mình.

## Quá trình thu thập tin tức bao gồm 3 giai đoạn

Giai đoạn 1: Tìm kiếm, lựa chọn URL cho các trang tin bài

Với mỗi trang lĩnh vực và thông tin về cấu trúc trang lĩnh vực được định nghĩa, Module thu thập tin tức dễ dàng nhận biết, lựa chọn những địa chỉ URL tiềm năng chứa nội dung tin bài thuộc về trang lĩnh vực đó. Ở bước đầu cài đặt, quá trình chọn URL tin bài này được thực hiện với ý tưởng đơn giản là lấy tất cả các URL không trùng lặp xuất hiện trong nội dung HTML của trang lĩnh vực. Hệ thống tiếp tục tìm kiếm những URL của tin bài cũ hơn ở những phân trang tiếp theo của trang lĩnh vực. URL của các phân trang thuộc một trang lĩnh vực có thể được tìm thấy ở nội dung HTML.

#### Giai đoạn 2: Bóc tách nội dung tin tức

Sử dụng cấu trúc của trang tin bài đã định nghĩa cho nội dung HTML tương ứng với URL tiềm năng chọn được ở bước trước, nội dung tin bài được cố gắng bóc tách. Quá trình này được xem là thành công nếu thu được một tin bài mới với đầu đủ các phần: tiêu đề, nội dung, thời gian đăng tải, hình ảnh.

#### Giai đoạn 3: Index nội dung và lưu trữ

Tin bài sau khi bóc tách được tiếp tục thực hiện các bước hậu thu thập như: index nội dung, kiểm tra, đánh giá tin bài trùng lặp. Tất cả các thông tin về tin bài được ghi nhận và lưu lại trong CSDL hệ thống.

### **Thu thập một tin bài mới**

Quá trình khảo sát và thu thập một tin bài từ một trang web không chỉ đơn thuần là lấy dữ liệu HTML Document về. Để có được nội dung chính của tin bài và loại bỏ những thông tin nhiễu, ta cần phải phân tích cấu trúc của tin bài nhằm trích lọc thông tin dựa trên cấu trúc thông tin đã được lưu trữ ở database. Thông tin sau khi được thu thập gồm có: tên website, địa chỉ url, nội dung, tiêu đề,... sẽ lưu vào database như mô hình đã trình bày. Giải thuật thu thập tin bài được trình bày như sau:

**Input:** Url: địa chỉ webpage chưa nội dung chung tin bài cần thu thập.

**Output:** nội dung tin bài thu thập được vào database.

### **Thuật giải**

```
foreach (mỗi website trong các trang web nông nghiệp đã lưu ở CSDL) {
```

```
    foreach (mỗi trang url thể loại của website nông nghiệp này) {
```

```
        DS link bài viết = rút trích link trang article từ XPATH
```

```

foreach (mỗi trang url bài viết) {
    if (bài viết chưa được lưu vào CSDL) {
        Tên bài viết = Rút trích từ XPATH
        Nội dung = Rút trích từ XPATH
        Hình đại diện = Rút trích từ Nội dung
        if nếu Hình đại diện tồn tại then lưu hình về hệ thống
        Lưu bài viết về CSDL
    }
}
}

```

### 5.2.2.2. Lưu trữ thông tin về cấu trúc trang báo điện tử và nội dung tin bài

Phần này sẽ trình bày việc xây dựng và hiện thực hóa mô hình tổ chức lưu trữ cho dữ liệu của hệ thống, thiết kế mô hình cơ sở dữ liệu lưu trữ cấu trúc thông tin các trang báo điện tử và các tin bài. Bản thiết kế cơ sở dữ liệu phục vụ mục đích quản lý cấu trúc thông tin các trang báo điện tử, quản lý tin bài bao gồm 3 bản được mô tả như sau:

The image displays three database schema designs side-by-side:

- Articles Table:** Contains columns for Id (PK, INTEGER), Content (TEXT), Image (TEXT), RetrievedDate (TEXT), Title (TEXT), Url (TEXT), and WebsitelId (FOREIGN KEY, INTEGER).
- Categories Table:** Contains columns for Id (PK, INTEGER), ArticleXPath (TEXT), ContentXPath (TEXT), TitleXPath (TEXT), Url (TEXT), and WebsitelId (FOREIGN KEY, INTEGER).
- Websites Table:** Contains columns for Id (PK, INTEGER), Name (TEXT), and Url (TEXT).

Hình 21: Thiết kế cơ sở dữ liệu của kho tin bài

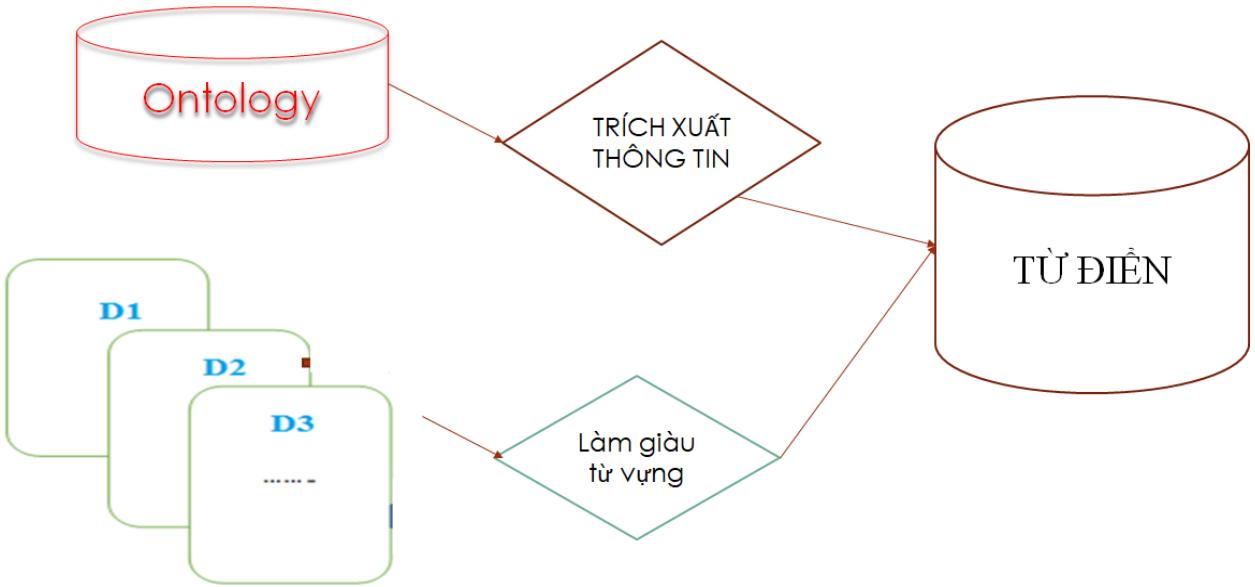
Dữ liệu sau khi thu thập được từ các trang web được xử lý và lưu trữ vào rice.db. Ví dụ lưu trữ các tin bài trong CSDL như sau:

19	152 1. Giới thiệu	2017-07-07 01:26:04.9173216	Bánh Ít (Plain Rice Flan)	<a href="http://nptnt.vinphuc.gov.vn/csdn...">http://nptnt.vinphuc.gov.vn/csdn...</a>	1
20	147 1. Giới thiệu	2017-07-07 01:26:04.8082578	Bánh Ít (Steamed Flat Cake)	<a href="http://nptnt.vinphuc.gov.vn/csdn...">http://nptnt.vinphuc.gov.vn/csdn...</a>	1
21	42 Bài 37: BÔNG LÚA: CÁC GIAI ĐOẠN...	2017-07-07 01:25:30.7379191	Bông lúa: các giai đoạn hình thàn...	<a href="http://nptnt.vinphuc.gov.vn/csdn...">http://nptnt.vinphuc.gov.vn/csdn...</a>	1
22	217 Thời gian hình thành bông kế từ kh...	2017-07-07 01:26:26.2666399	Bông và hạt lúa	<a href="http://nptnt.vinphuc.gov.vn/csdn...">http://nptnt.vinphuc.gov.vn/csdn...</a>	1
23	144 1. Giới thiệu	2017-07-07 01:26:01.714529	Bún (Rice Vermicelli)	<a href="http://nptnt.vinphuc.gov.vn/csdn...">http://nptnt.vinphuc.gov.vn/csdn...</a>	1
24	14 Bài 79: BỆNH LEM LÉP HẠT	2017-07-07 01:25:20.941494	Bệnh Lem lèp hạt	<a href="http://nptnt.vinphuc.gov.vn/csdn...">http://nptnt.vinphuc.gov.vn/csdn...</a>	1
25	16 Bài 74: BỆNH BẠC LÁ	2017-07-07 01:25:23.7687304	Bệnh bạc lá	<a href="http://nptnt.vinphuc.gov.vn/csdn...">http://nptnt.vinphuc.gov.vn/csdn...</a>	1
26	80 1. Triệu chứng	2017-07-07 01:25:43.5966459	Bệnh bạc lá ( <i>Xanthomonas oryzic...</i> )	<a href="http://nptnt.vinphuc.gov.vn/csdn...">http://nptnt.vinphuc.gov.vn/csdn...</a>	1
27	101 Stenchaetothrips biformis Bagnall	2017-07-07 01:25:49.6227119	Bệnh bọ trĩ	<a href="http://nptnt.vinphuc.gov.vn/csdn...">http://nptnt.vinphuc.gov.vn/csdn...</a>	1
28	1 Bài 78: BỆNH CHÁY BÌA LÁ	2017-07-07 01:25:16.5044693	Bệnh cháy bìa lá	<a href="http://nptnt.vinphuc.gov.vn/csdn...">http://nptnt.vinphuc.gov.vn/csdn...</a>	1
29	106 Microdochium oryzae	2017-07-07 01:25:52.8242994	Bệnh cháy lá	<a href="http://nptnt.vinphuc.gov.vn/csdn...">http://nptnt.vinphuc.gov.vn/csdn...</a>	1
30	13 Bài 71: BỆNH HOA CÚC	2017-07-07 01:25:20.9887643	Bệnh hoa cúc	<a href="http://nptnt.vinphuc.gov.vn/csdn...">http://nptnt.vinphuc.gov.vn/csdn...</a>	1
31	73 1. Triệu chứng	2017-07-07 01:25:40.377863	Bệnh hoa cúc ( <i>Ustilaginoidae vire...</i> )	<a href="http://nptnt.vinphuc.gov.vn/csdn...">http://nptnt.vinphuc.gov.vn/csdn...</a>	1
32	115 Bệnh phát sinh gây hại từ lúc lúa ph...	2017-07-07 01:25:52.6829435	Bệnh hoa cúc ( <i>Ustilaginoidae vire...</i> )	<a href="http://nptnt.vinphuc.gov.vn/csdn...">http://nptnt.vinphuc.gov.vn/csdn...</a>	1
33	3 Bài 70: BỆNH KHÔ VÂN	2017-07-07 01:25:15.9725191	Bệnh khô vân	<a href="http://nptnt.vinphuc.gov.vn/csdn...">http://nptnt.vinphuc.gov.vn/csdn...</a>	1
34	81 1. Triệu chứng	2017-07-07 01:25:43.5499833	Bệnh khô vân ( <i>Rhizoctonia solani</i> )	<a href="http://nptnt.vinphuc.gov.vn/csdn...">http://nptnt.vinphuc.gov.vn/csdn...</a>	1
35	97 Bệnh phát triển thuận lợi trong điều...	2017-07-07 01:25:49.7301236	Bệnh khô vân ( <i>Rhizoctonia solani</i> ...)	<a href="http://nptnt.vinphuc.gov.vn/csdn...">http://nptnt.vinphuc.gov.vn/csdn...</a>	1
36	98 Bệnh được phát hiện đầu tiên ở Việt...	2017-07-07 01:25:49.6831266	Bệnh lùn xoắn lá (Rice ragged stu...	<a href="http://nptnt.vinphuc.gov.vn/csdn...">http://nptnt.vinphuc.gov.vn/csdn...</a>	1
37	2 Bài 75: BỆNH LÚA VON	2017-07-07 01:25:16.1134602	Bệnh lúa von	<a href="http://nptnt.vinphuc.gov.vn/csdn...">http://nptnt.vinphuc.gov.vn/csdn...</a>	1
38	96 Triệu chứng điển hình và dễ nhận b...	2017-07-07 01:25:49.5324362	Bệnh lúa von ( <i>Fusarium monilifor...</i> )	<a href="http://nptnt.vinphuc.gov.vn/csdn...">http://nptnt.vinphuc.gov.vn/csdn...</a>	1
39	87 1. Triệu chứng	2017-07-07 01:25:46.6444839	Bệnh nghẹt rễ sinh lý lúa	<a href="http://nptnt.vinphuc.gov.vn/csdn...">http://nptnt.vinphuc.gov.vn/csdn...</a>	1
40	17 Bài 76: BỆNH THỐI BẸ	2017-07-07 01:25:24.1904183	Bệnh thối bẹ	<a href="http://nptnt.vinphuc.gov.vn/csdn...">http://nptnt.vinphuc.gov.vn/csdn...</a>	1
41	108 Bệnh thường xuất hiện và gây hại b...	2017-07-07 01:25:52.8861102	Bệnh thối bẹ ( <i>Sarocladium oryzae</i> )	<a href="http://nptnt.vinphuc.gov.vn/csdn...">http://nptnt.vinphuc.gov.vn/csdn...</a>	1
42	79 1. Triệu chứng	2017-07-07 01:25:43.5966459	Bệnh tiêm đòn sần (tuyền trùng D...	<a href="http://nptnt.vinphuc.gov.vn/csdn...">http://nptnt.vinphuc.gov.vn/csdn...</a>	1
43	00 Rãnh Ám mía nhất hiện đầu tiên Ám mía	2017-07-07 01:25:40.800274	Rãnh viền lúa rã /Diseases of rice	<a href="http://nptnt.vinphuc.gov.vn/csdn...">http://nptnt.vinphuc.gov.vn/csdn...</a>	1

Hình 22: lưu trữ data trong bảng Articles

Hiện tại có khoảng 221 bản tin được crawl về từ các trang web. Tuy nhiên số lượng này sẽ tăng lên và được cập nhật liên tục dựa vào bộ quản lý thu thập.

### 5.2.3. Xây dựng từ điển



Hình 23: Quy trình xây dựng kho từ điển

Quá trình xây dựng kho từ điển gồm có 2 quá trình:

- Trích xuất các khái niệm, cá thể trong ontology bỏ vào kho từ điển.
- Làm giàu ontology là mở rộng các từ đồng nghĩa gần nghĩa, từ có liên quan bỏ vào từ điển.

Từ các bài viết trong kho tài liệu, tiến hành tách từ trong từng document công cụ **UETsegmenter2**. Công cụ này có hỗ trợ bộ từ điển dành riêng cho tool, nhóm tiến hành bổ sung từ điển ontology xây dựng được vào tool để tiến hành tách từ hiệu quả hơn. Sau khi tách từ từ các tin bài, để tìm được giữa những term có mối quan hệ nào đó như đồng nghĩa, gần nghĩa hoặc có sự liên quan ta có thể dùng cách tra từ điển, tuy nhiên, từ tìm được trong từ điển chưa chắc sẽ tồn tại trong kho tin bài. Vì thế, để tăng độ chính xác và biết 1 term sẽ tương quan (tương đồng) nhất với term nào trong kho tài liệu, ta có thể áp dụng kỹ thuật phân tích thống kê Global Analysis. Global Analysis sẽ tính ma trận “tương quan” (association matrix) trên toàn bộ term trong corpus. Dùng term có độ tương đồng

lớn nhất với một term đang xét để xây dựng mạng lưới từ đồng nghĩa, gần nghĩa.  
Association Matrix có dạng:

Term	w <sub>1</sub>	w <sub>2</sub>	w <sub>3</sub>	.....	w <sub>n</sub>
w <sub>1</sub>	c <sub>11</sub>	c <sub>12</sub>	c <sub>13</sub>	.....	c <sub>1n</sub>
w <sub>2</sub>	c <sub>21</sub>				
w <sub>3</sub>	c <sub>31</sub>				
.	.				
.	.				
w <sub>n</sub>	c <sub>n1</sub>				

Với  $c_{ij}$  là độ tương quan giữa term  $w_i$  và  $w_j$ , được tính như sau:

$$c_{ij} = \sum_{d_k \in D} f_{ik} \times f_{jk}$$

$f_{ik}$ : Tần số xuất hiện của term  $w_i$  trong document  $k$

Hình 24: Ma trận tương quan Global Analysis

Tuy nhiên, độ tương quan dựa vào tần số thường chọn những term có tần số lớn, vì vậy cần chuẩn hóa lại. Ta sẽ thay thế  $c_{ij}$  bằng  $s_{ij}$  (normalize association score):

$$s_{ij} = \frac{c_{ij}}{c_{ii} + c_{jj} - c_{ij}}$$

$s_{ij}$  sẽ bằng 1 nếu 2 term  $w_i$  và  $w_j$  có cùng tần số trong tất cả các document. Bộ từ điển sau khi xây dựng có thể mô hình hóa như sau:

$$\text{Dictionary} = (T, Re)$$

Trong đó  $T$ : tập các term có trong từ điển

$Re$ : là tập quan hệ giữa các term trong từ điển

Tập  $Re$  được biểu diễn là  $Re = \{ \text{syn}, \text{near}, \text{related}, \text{is-a}, \text{has-a} \}$

Trong đó : syn là quan hệ đồng nghĩa; near là quan hệ gần nghĩa, is-a là quan hệ là một, has-a là quan hệ có một. Ví dụ, ta có term đạo ôn, related của đạo ôn là rồng vàng, near của đạo ôn là vàng lá, syn của đạo ôn là bệnh đạo ôn.

#### 5.2.4. Bài toán lập chỉ mục cho các tài liệu dựa trên Ontology ARK

Dựa vào từ điển đã được xây dựng trước đó, tiến hành đánh chỉ mục cho từng document. Từ các tài liệu được lấy về tiến hành bỏ các kí tự đặc biệt sau đó tiến hành so khớp chuỗi với hàm so khớp, ta được 1 list danh sách cách từ trong document. Các keyword sẽ được đánh trọng số, dựa theo mức độ quan trọng của nó trong bài viết và trong toàn hệ thống. Cơ sở lý thuyết cho việc tính trọng số là mô hình không gian vector. Với mỗi từ khoá trong bài viết, công thức tính trọng số của nó là  $\mathbf{W} = \mathbf{tf} * \mathbf{idf}$ .

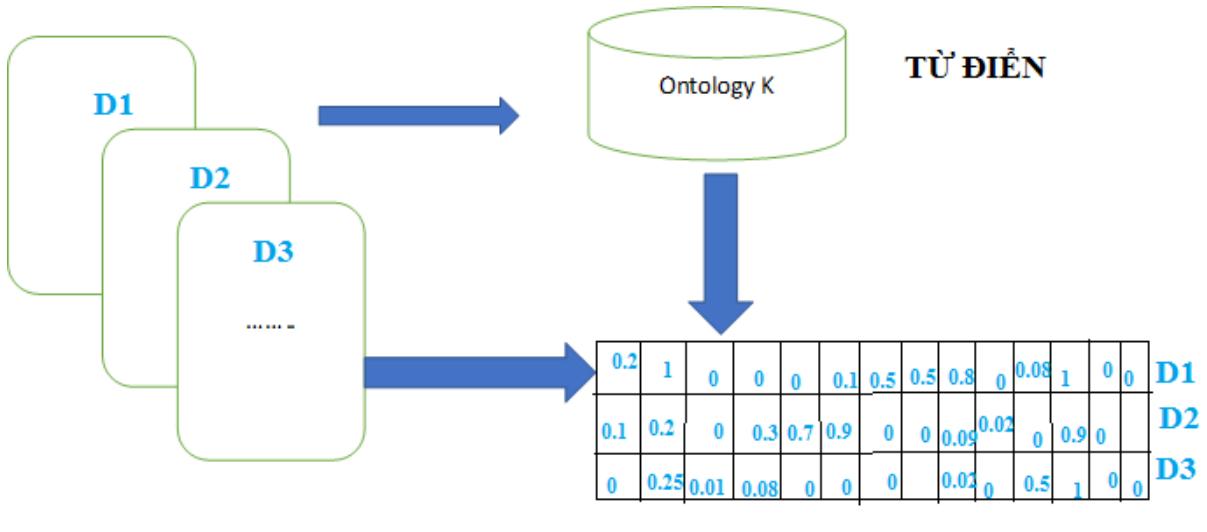
Trong đó:

- **tf:** mức độ quan trọng của keyword trong bài viết.
- **idf:** mức độ quan trọng của keyword trên toàn bộ tài liệu.

$$tf(t, d) = \frac{f(t, d)}{\max\{f(w, d): w \in d\}}$$

Với:

- $f(t, d)$  là số lần xuất hiện của thuật ngữ  $t$  trong document  $d$ .
- $\max\{f(w, d): w \in d\}$  là số lần xuất hiện nhiều nhất của 1 từ trong  $d$ .
- $idf(t, D) = \log_{10} \left( \frac{|D|}{|\{d \in D: t \in d\}|} \right)$
- $idf(t, D)$ : tần số nghịch của một thuật ngữ  $t$  trong toàn bộ Document
- $|D|$ : số lượng Document
- $|\{d \in D: t \in d\}|$ : số lượng Document có chứa thuật ngữ  $t$ , chính xác hơn nên có +1 để loại bỏ trường hợp chia cho 0.



Hình 25: Mô tả quy trình bài toán tách chỉ mục

Ví dụ bài toán đánh chỉ mục như sau:

Có tập tài liệu gồm 3 tài liệu: Doc1, Doc2, Doc3

Doc : {Doc1: “đạo ôn vàng lá lem lép hạt...”

Doc2: “sâu năng thân ún vàng lá...”,

Doc3: “anphaA, đốm vàng, đạo ôn...” }

Giả sử ta có đạo ôn xuất hiện Doc1 là 10 và Doc3 là 11. Vàng lá xuất hiện ở Doc1 là 12 và Doc2 là 13, lem lép hạt xuất hiện ở Doc1 1 lần, sâu năng xuất hiện ở Doc2 11 lần, Anpha A , đốm vàng xuất hiện Doc3 9 và 5 lần. Tính trọng số cho từng term trong mỗi bài viết bằng công thức idf và tf ta được như sau:

	đạo ôn	vàng lá	lem lép hạt	sâu năng	thân ún	anphaA	đốm vàng
Doc1	0.08	0.084	0.47	0	0	0	0
Doc2	0	0.091	0	0.477	0.47	0	0
Doc3	0.09	0	0	0	0	0.47	0.47

Mỗi tài liệu được biểu diễn bằng 1 vector tương ứng với số từ diễn xuất hiện trong ontology.

Ví dụ ontology có 300 term thì mỗi document sẽ tương ứng với vector 300 chiều.

**Input:** kho tài liệu, từ điển (Ontology ARK)

**Output:** danh sách chỉ mục của kho tài liệu.

```
Indexing(title, tags, danh sách từ khoá)
```

```
{
```

```
    Bước 1: Tách từ trong title và tags, đánh trọng số w = 1
```

```
    Bước 2: Gom nhóm các từ khoá trùng nhau và ghi lại số lượng.
```

```
    Khởi tạo danh sách từ được đánh trọng số = null.
```

```
    foreach(keyword trong distsinctWordsCounting)
```

```
{
```

```
        Bước 3: Tính tf = keyword.count / tổng số lượng từ khoá.
```

```
        Bước 4: Lưu vào danh sách từ được đánh trọng số.
```

```
}
```

```
    return danh sách từ được đánh trọng số.
```

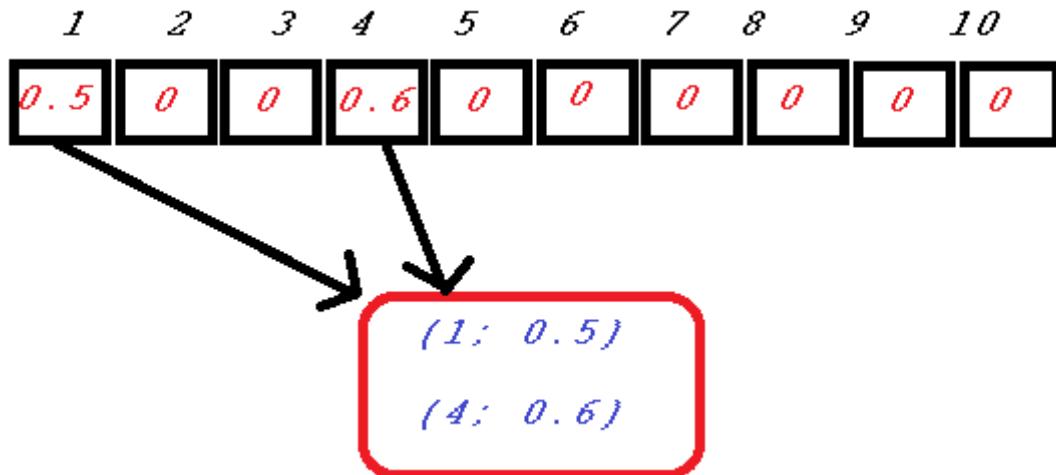
```
}
```

### 5.2.5. Biểu diễn vector thành túi từ

Mô hình không gian vector truyền thống cung cấp các phép toán để so khớp các vector tài liệu với vector câu truy vấn, tuy nhiên các vector tài liệu được biểu diễn bằng các vector rất thưa (nhiều thành phần có giá trị là 0). Do đó, một trong những cách giảm bớt sự lãng phí trong việc lưu trữ và xử lý các vector là đưa biểu diễn vector truyền thống về các túi từ (bag of words), trong đó các thành phần có giá trị bằng 0 của vector được bỏ qua mà không làm mất mát thông tin được biểu diễn trong vector ban đầu.

Thí dụ: Giả sử có 1 vector tài liệu trên không gian 10 chiều:

Các thành phần khác không (1 và 4) sẽ được đưa vào túi từ theo từng bộ



Hình 26: Biểu diễn vector thành túi từ

### Thuật giải biểu diễn vector thành túi từ

Input: Danh sách các vector tài liệu

Output: Các túi từ tương ứng với từng vector đầu vào

Mỗi túi từ là một tập hợp của các bộ giá trị T, trong đó mỗi bộ giá trị T gồm 2 thành phần là:

Chỉ số chiều: là vị trí (index) của thành phần trên vector.

Giá trị trọng số: là giá trị trọng số tương ứng của thành phần trên vector.

Với mỗi vector tài liệu V:

Duyệt từng thành phần  $V_i \in V$ :

Nếu ( $V_i \neq 0$ ) thì lập bộ  $T_i = (i, V_i)$ . Sau đó đưa  $T_i$  vào túi từ

#### 5.2.6. Xử lý mở rộng câu truy vấn thành túi từ

Một trong những nguyên nhân giảm thiểu độ chính xác của hệ thống tìm kiếm đó là, khi nâng từ bài toán cơ sở so khớp toàn bộ “string matching” lên bài toán “semantic search”, có rất nhiều yếu tố mới xuất hiện, tác động lên sự liên kết giữa truy vấn và tài liệu. Trong đó không thể kể đến các mối quan hệ tiềm ẩn của ngôn ngữ tự nhiên - vốn không tồn tại trong bài toán cơ sở “string matching”, chẳng hạn, trong không gian ngữ nghĩa, hai biểu diễn từ có thể rất khác nhau về cách viết dưới dạng xâu nhưng lại gần gũi trong miền ngữ nghĩa (từ đồng nghĩa), nếu cứ áp dụng “string matching” thì các liên hệ này sẽ không được khai thác tốt, làm cho độ chính xác của hệ thống truy vấn trên khía cạnh ngữ nghĩa bị giảm xuống. Do đó, để

hạn chế sự giảm thiểu độ chính xác của hệ thống do nguyên nhân trên, ta cần khai thác các mối quan hệ giữ các biểu diễn từ. Người ta đã đề xuất rất nhiều phương pháp để giải quyết vấn đề trên, trong hệ thống này, chúng ta đi theo hướng tiếp cận mở rộng câu truy vấn (query expansion). Như tên gọi của nó, phương pháp mở rộng câu truy vấn nỗ lực làm giàu ý nghĩa cho câu truy vấn của người dùng hơn là chỉnh sửa các cơ chế so khớp và xếp hạng. Có rất nhiều cách để mở rộng câu truy vấn, nhưng dựa trên miền tri thức mà nhóm xây dựng thì nhóm quyết định mở rộng câu truy vấn 2 cấp như sau:

**Cấp 1:** Sau khi dùng biểu thức chính quy (regex) để xử lý câu truy vấn ta sẽ được một danh sách các term. Tiến hành vào danh sách từ điển đã tìm xem term trong danh sách là term có quan hệ gì, nếu term đó là từ đồng nghĩa thì đánh trọng số 0.9, nếu term đó là từ gần nghĩa thì đánh trọng số 0.8, nếu là từ liên quan thì đánh trọng số 0.5. Sau đó tiến hành mở rộng danh sách các term sang các từ đồng nghĩa gần nghĩa có liên quan.

**Cấp 2 :** Mở rộng túi từ câu truy vấn

- Nếu túi từ câu truy vấn tồn tại class trong Ontology thì mở rộng câu truy vấn để tìm kiếm thêm là các individual của class đó hoặc lớp con của class. Ví dụ: {Bệnh } → { Bệnh, Đạo ôn, vàng lá, lem lép hạt , .. }
- Nếu trong túi từ của câu truy vấn là individual thì mở rộng câu truy vấn sang các quan hệ đồng nghĩa, bằng nghĩa. Ví dụ từ bằng nghĩa của đạo ôn là bệnh đạo ôn.

Tương ứng với mỗi term trong Query được gán cho một trọng số này, mà trọng số này ban đầu được đặt ra bất kì, sau quá trình thử nghiệm thì trọng số này ngày càng được hiệu chỉnh chính xác hơn.

### Thuật giải mở rộng câu truy vấn

Bước 1: Nhập danh sách term mà query người dùng nhập vào

Bước 2: khởi tạo query\_new = query

Bước 3: Duyệt từng từ term và mở rộng từng term

```
foreach (term t in query) {  
    if (term is Individual) then
```

```
        Tìm từ bằng nghĩa của individual add vào query_new
```

```

    Tìm từ đồng nghĩa của individual add vào query_new
    if (t is Class) then Tìm individual của class và add vào query_new
}

```

Bước 4: xuất tập query\_new mới

### 5.2.7. So khớp túi từ query với túi từ document và sắp xếp trả về kết quả

Sau khi mở rộng câu truy vấn và tính toán trọng số cho các keyword trong tài liệu, ta bắt đầu so khớp túi từ giữa câu query và túi từ document, sắp xếp kết quả trả về. Công thức so khớp túi từ và câu query như sau:

$$\text{Độ liên quan } (d_i, q) = \cos(d_i, q) = \frac{\vec{d_i} \cdot \vec{q}}{|d_i| \cdot |q|}$$

Trong đó:  $d_i$  là 1 document

$q$  là câu truy vấn người dùng nhập vào

$\cos(d_i, q)$  càng nhỏ thì độ tương quan giữa câu truy vấn và tài liệu  $q$  càng lớn.

**Input:** Các túi từ tài liệu D và túi từ truy vấn Q

**Output:** Danh sách các bài viết trả về

Có 2 giai đoạn chính

**Giai đoạn 1:** Tìm phần giao và tính % phần giao đối với từng tài liệu

foreach (document Di in D) { // Duyệt các túi tài liệu D

Bước 1: tìm các term giao giữa query và document

Bước 2: tính độ liên quan giữa tài liệu  $d_i$  và query bằng công thức độ liên quan

$$\cos(d_i, q) = \frac{\vec{d_i} \cdot \vec{q}}{|d_i| \cdot |q|}$$

}

**Giai đoạn 2:** Xếp hạng độ liên quan giữa tài liệu và câu query sau đó trả về kết quả

Mỗi tài liệu có một độ liên quan  $\cos(d_i, q)$ ,  $\cos(d_i, q)$  càng nhỏ thì độ liên quan giữa tài liệu đó với câu truy vấn càng gần, sắp hạng  $\cos(d_i, q)$  và trả về kết quả cho người dùng.

### 5.3. VĂN ĐỀ HIỂN THỊ THÔNG TIN TRẢ VỀ CHO NGƯỜI DÙNG

#### 5.3.1. Phát biểu bài toán

Tương tự như wiki và các diễn đàn hệ thống muốn trả về những bài viết đã định dạng có cấu trúc, định hướng tìm kiếm cho những thông tin mới nhất trên bài viết cho người dùng dễ dàng nhất, tiện lợi nhất. Ví dụ người dùng đang đọc bài viết về bệnh ôn nhưng bài viết có thể liên kết dữ liệu đến các bài khác như là rày nâu, rày xanh,...

#### 5.3.2. Xây dựng giải pháp

Nhóm quyết định xây dựng ngôn ngữ Semantic Code, là một biến thể của BBCode, để đặc tả định dạng thông tin. Sau khi định dạng ta sẽ xây dựng một hệ thống parser chuyển đổi thành ngôn ngữ HTML mà trình duyệt web có thể dịch được. Một vài thẻ Semantic Code được hệ thống sử dụng:

Nhóm	Semantic code	HTML code
Ký tự escaped	left [[square bracket]	left [square bracket]
Thẻ HTML thông dụng	[b]bold[/b] text	<strong>bold</strong> text
	[i]italic[/i] text	<em>italic</em> text
	[u]underline[/u] text	<u>underline</u> text
	[sub]subscript[/sub] text	<sub>subscript</sub> text
	[sup]suberscript[/sub] text	<sup>superscript</sup> text
	[quote]quote text[/quote]	<blockquote>quote text</blockquote>
	[code]code text[/code]	<code>code text</code>
	h1, h2, h3, h4, h5	
	[h1]heading 1[/h1]	<h1>heading 1</h1>
	[center]center text[/center]	<center>center text</center>

	[img="http://imgur.com/pic.png"]	
	Alt text[/img]	
	[simg="benh-vang-la.png"]	
	Bệnh vàng lá[/img]	
	[hr]	<hr />
	[br] line break	  line break
	[ol]	<ol>
	[*] One	<li>One</li>
	[*] Two	<li>Two</li>
	[/ol]	</ol>
	[ul=square]	<ul style="list-style-type:square">
	[*] One	<li>One</li>
	[*] Two	<li>Two</li>
	[/ul]	</ul>
Thẻ màu font	red, green, blue, navy, fuchsia, orange, yellow, gray, purple	
	[red]red[/red] text	<font color="red">red</font> text
Thẻ Hyperlink	[url="http://google.com"]google	<a href="http://google.com">google</a>
	[/url]	
	[Benh]các loại bệnh[/Benh]	<a href=" <b>đường dẫn url</b> ">các loại bệnh</a>

```
[Benh=BenhVangLa]bệnh vàng lá    <a href=""đường dẫn
[ /Benh]                                         url">bệnh vàng lá</a>
```

Ngôn ngữ Semantic Code được nhóm thiết kế đặc tả như sau:

```
nodes = { node }

node = singular | text-container | list-container | text

singular = [ singular-type ]

text-container = color-container | link-container | image-container | simple-text-container |
ontology-container

color-container = [ color-type ] nodes [ / color-type ]

link-container = [ url = " online-link " ] nodes [ / url ]

image-container = online-image-container | static-image-container

online-image-container = [ img = " online-link " ] nodes [ / img ]

static-image-container = [ simg = " file-name " ] nodes [ / simg ]

simple-text-container = [ text-type ] nodes [ / text-type ]

ontology-container = class-container | individual-container

class-container = [ class-name ] nodes [ / class-name ]

individual-container = [ class-name = individual-name ] nodes [ / class-name ]

list-container = ordered-list-container | unodered-list-container

ordered-list-container = [ ol [ = ordered-type ] ] { list-item-container } [ / ol ]

unordered-list-container = [ ul [ = unordered-type ] ] { list-item-container } [ / ul ]

color-type = red | green | blue | navy | fuchsia | orange | yellow | gray | purple

text-type = b | i | u | sub | sup | quote | code | center | h1 | h2 | h3 | h4 | h5

singular-type = hr | br

ordered-type = 1 | A | a | I | i

unordered-type = disc | circle | square | none

list-item-container = [ * ] nodes

text = STRING
```

online-link = <b>STRING</b>
file-name = <b>STRING</b>
class-name = <b>IDENT</b>
individual-name = <b>IDENT</b>

**Ví dụ:** Thuộc tính triệu chứng của bệnh cháy lá có đoạn text được định dạng dưới dạng Semantic Code như sau:

[Benh=BenhChayLa]Bệnh cháy lá[Benh] [b]Triệu chứng[/b] biểu hiện giống như là bệnh bong lá và biến đổi phụ thuộc vào tuổi, giai đoạn của cây trồng, phương pháp canh tác và mật độ cây trồng. Triệu chứng ban đầu phát triển trên đinh hoặc ở gờ của phiến lá, diện tích vết bệnh ước tính khoảng 1-2 cm<sup>2</sup>.

[br]

[simg="chayla.jpg"]cháy lá[/simg]

[br]

Sẽ được hiển thị trên trình duyệt:

**Bệnh cháy lá** **Triệu chứng** biểu hiện giống như là bệnh bong lá và biến đổi phụ thuộc vào tuổi, giai đoạn của cây trồng, phương pháp canh tác và mật độ cây trồng. Triệu chứng ban đầu phát triển trên đinh hoặc ở gờ của phiến lá, diện tích vết bệnh ước tính khoảng 1-2 cm<sup>2</sup>.

/\*hình ảnh cháy lá \*/

## CHƯƠNG 6. ÚNG DỤNG VÀ THỬ NGHIỆM

Mục tiêu của ứng dụng là xây dựng một phần mềm máy tính với giao diện web hỗ trợ tìm kiếm, tra cứu, tư vấn dành cho nông nghiệp lúa nước. Ứng dụng phục vụ cho nhu cầu thực tế của người nông dân hiện nay, hệ thống bao gồm các module chính sau:

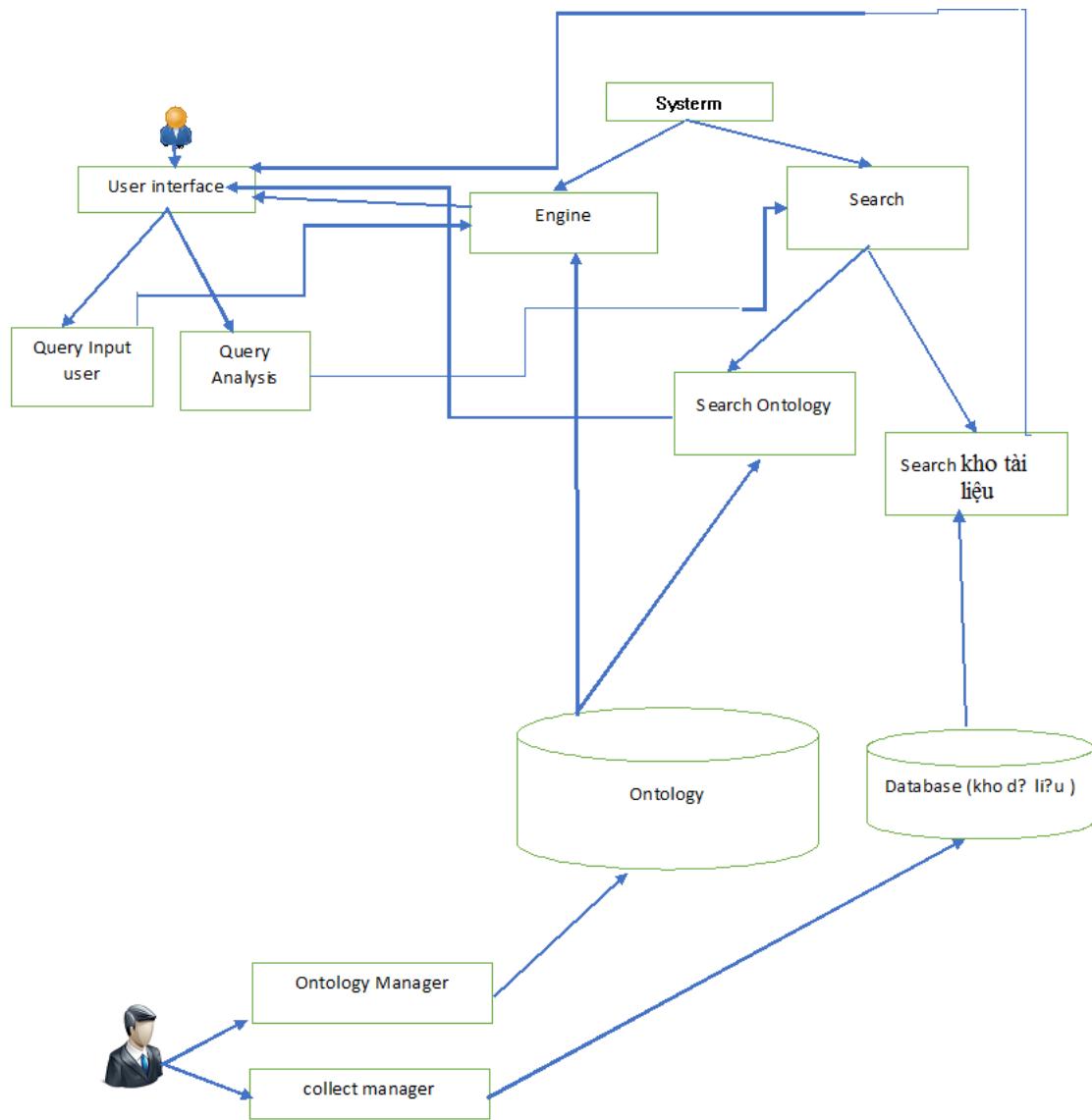
**Module chuẩn đoán:** thực hiện chuẩn đoán dựa trên những input người dùng nhập vào hệ thống tiến hành xử lý và đưa ra kết quả chuẩn đoán cho người dùng, gợi ý người dùng cách phòng bệnh, trị bệnh,...

**Module tìm kiếm tra cứu:** hệ thống cho phép tìm kiếm trên kho tin bài và tra cứu trên Ontology. Tìm kiếm các tin bài viết theo nhiều chức năng, không chỉ hỗ trợ tìm kiếm theo từ khóa mà còn hỗ trợ tìm kiếm bằng cách mở rộng câu truy vấn để tìm kiếm (một dạng tìm kiếm ngữ nghĩa cơ bản). Tra cứu bài viết trên Ontology có nhiều định hướng tra cứu như: định hướng tra cứu dạng cây khái niệm, tra cứu theo cụm từ, tra cứu theo dạng câu truy vấn tổng quát.

**Module thu thập-quản lý:** hệ thống thực hiện việc lập lịch và tự động thu thập các bài viết mới nhất từ các trang web đã được lưu trữ. Quản lý việc thêm, xóa các tin bài được crawl về.

## 6.1. THIẾT KẾ HỆ THỐNG

### 6.1.1. Kiến trúc của ứng dụng



Hình 27: Sơ đồ cấu trúc hệ thống

**User Interface:** Giao diện giao tiếp giữa người dùng và hệ thống. Giao diện hệ thống phải được thiết kế phù hợp với chức năng, thân thiện, tiện dụng đối với từng đối tượng sử dụng. Thực hiện vai trò tiếp nhận thông tin từ người dùng, sau đó chuyển tiếp yêu cầu này đến các module tiếp theo, hiển thị kết quả trả về của hệ thống cùng với những đề xuất nếu có.

**Query Analysis:** Bộ phận phân tích truy vấn người dùng sẽ tiếp nhận từ User Interface, chuẩn hóa thông tin bằng cách xóa đi những thông tin nhiễu và không cần thiết, mở rộng câu truy vấn thành một túi để mở rộng kết quả tìm kiếm. Câu truy vấn sẽ được biểu diễn theo cấu trúc đơn giản là một danh sách các từ khóa. Kết quả của giai đoạn này sẽ là một danh sách các term của người dùng được dùng làm đầu vào của bộ tìm kiếm của hệ thống.

**Ontology Searching:** Bộ tìm kiếm nhận query từ người dùng, thực hiện đi vào Ontology tìm kiếm và trả về cho User Interface.

**Article Searching:** Bộ tìm kiếm sẽ nhận cấu trúc đặc tả từ Query Analysis, thực hiện việc tính toán, tìm kiếm và trả về kết quả cho User Interface. Tùy thuộc vào chức năng tìm kiếm mà sẽ so khớp truy vấn và lập chỉ mục của các tin bài khác nhau.

**Advisory Query Analysis:** bộ phận phân tích các yêu cầu của người dùng cho hệ thống chuẩn đoán, tiếp nhận sự kiện Input mà người dùng nhập vào. Kết quả của giai đoạn này sẽ là danh sách các sự kiện input làm đầu vào cho bộ phận Engine để thực hiện việc chuẩn đoán.

**Engine (Bộ phận suy diễn):** thực hiện công việc suy diễn, đưa ra lời khuyên cho người dùng.

**Ontology manager (Bộ quản lý tri thức):** Bộ quản lý Ontology cho phép tổ chức lưu trữ, cập nhật và tìm kiếm trên Ontology.

**Collecting manager (Bộ quản lý thu thập tin bài) :** Bộ quản lý thu thập trang tin có nhiệm vụ thu thập các tin bài trên các trang web. Lập lịch thời gian tự động để hệ thống có thể crawl bài viết mới nhất từ các trang web cập nhật vào database.

### 6.1.2. Mục tiêu ứng dụng

Hệ thống là nhịp cầu giữa nông dân và các chuyên gia nông nghiệp nhằm hỗ trợ công tác lúa nước, có khả năng đưa ra những kết quả ở dạng phán đoán, tư vấn, góp ý dựa trên kiến thức và kinh nghiệm của chuyên gia được đưa vào hệ thống. Bên cạnh đó cho phép người dùng thực hiện việc tra cứu thông tin trong miền thi thức lúa nước và tìm kiếm những bài viết có liên quan đến nhu cầu thông tin của người dùng trong miền tri thức đó.

### 6.1.3. Yêu cầu và chức năng của hệ thống

#### Yêu cầu của hệ thống

- **Đối với kỹ sư tri thức**

Cho phép tổ chức quản lý kho dữ liệu, xử lý các thao tác với dữ liệu (vd: thêm, xoá tài liệu). Thực hiện bổ sung các tập luật suy diễn.

- **Đối với người dùng phổ thông**

Người dùng có thể tìm kiếm, tra cứu thông tin liên quan đến lĩnh vực trồng lúa, Ngoài ra, người dùng sử dụng chức năng chuẩn đoán sâu bệnh hại trên cây lúa và tham khảo lời khuyên và biện pháp phòng trừ.

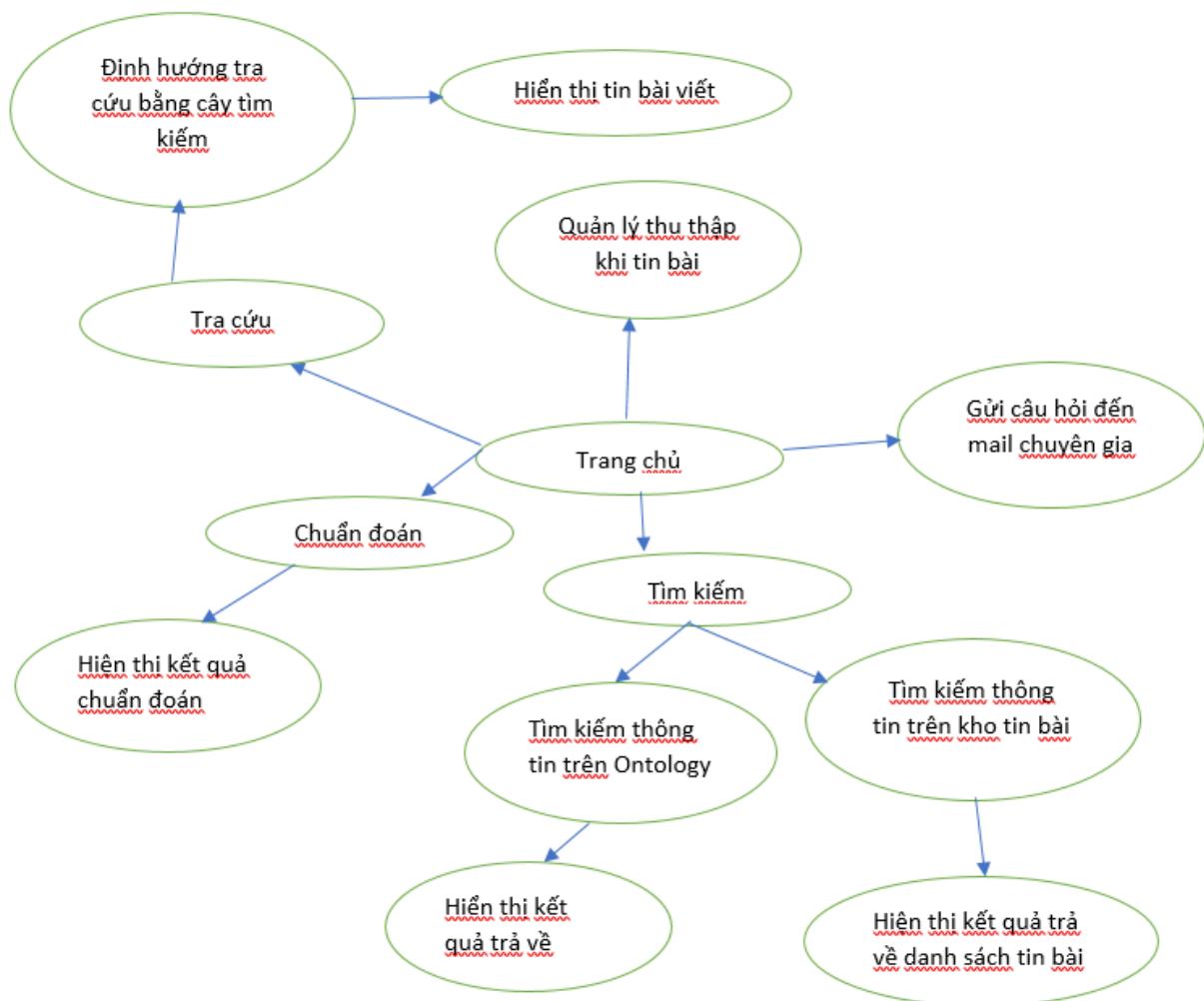
- **Đối với nhà quản trị**

Điều hành hệ thống, thực hiện phân quyền cho người sử dụng hệ thống.

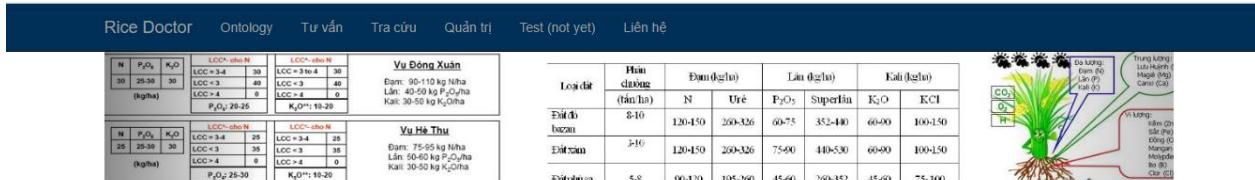
#### **6.1.4. Chức năng của hệ thống**

- Chức năng tra cứu trên hệ thống.
- Chức năng tìm kiếm thông tin trên kho tin bài.
- Chuẩn đoán sâu bệnh hại, tư vấn, đưa ra cách phòng trừ sâu bệnh hại trên cây lúa
- Chức năng quản lý thu thập kho tin bài.
- Chức năng gửi các câu hỏi người dùng đến địa chỉ mail của các chuyên gia.

### 6.1.5. Tổ chức giao diện hệ thống



Hình 28: Tổ chức giao diện của hệ thống



#### Tin tức mới nhất



##### Thời gian sinh trưởng phát triển của cây lúa

(TT-GTDT Vĩnh Phúc) - Thời gian sinh trưởng của cây lúa được tính từ khi hạt lúa nảy mầm đến khi chín hoàn toàn, thay đổi tuỳ theo giống và điều kiện ngoại cảnh. - Đối với lúa cấy: Bao gồm thời gian ở ruộng ma và thời gian ở ruộng lúa cây. - Đối với lúa gieo thẳng: Được tính từ thời gian gieo hạt đến thu hoạch. Ở ... [xem thêm]



##### Giai đoạn phát triển đốt thân

(TT-GTDT Vĩnh Phúc) - Trên đồng ruộng sau khi đạt số nhánh tối đa cây lúa chuyển sang thời kỳ làm đốt \* Thời gian làm đốt - Thời gian làm đốt dài hay ngắn có liên quan chất chẽ đến thời kỳ trổ bông, cũng như liên quan đến số lóng kéo dài trên thân nhiều hay ít. - Giống lúa ngắn ngày có thời gian làm đốt khoảng 25 -30 ... [xem thêm]



##### Giai đoạn đẻ nhánh

(TT-GTDT Vĩnh Phúc) - Điều kiện bình thường sau cấy 5 -7 ngày cây lúa có thể bén rễ hồi xanh, chuyển sang đẻ nhánh. Trời ẩm ướt, thiếu ánh sáng, nhiệt độ thấp, thời gian bén rễ hồi xanh kéo dài 15 -20 ngày, thậm chí 25 - 30 ngày ở vụ chiêm xuân phía Bắc. Thời kỳ đẻ nhánh, cây lúa sinh trưởng nhanh và mạnh và rễ và lá ... [xem thêm]

### Hình 29: Trang chủ của hệ thống

Trang chủ cung cấp những thông tin về lĩnh vực lúa nước và các hướng dẫn trồng lúa từ chuyên gia nổi tiếng cho người dùng. Thông tin được lấy từ các kênh tin tức về lúa và được cập nhật thường xuyên.



## Ontology

### Lớp và Thể hiện

#### Tri thức về lĩnh vực lúa nước miền Trung

### Quan hệ

Biểu tượng  $\Theta$  đại diện cho lớp.

### Thuộc tính

Biểu tượng  $\Theta$  đại diện cho quan hệ.

Biểu tượng  $A$  đại diện cho thuộc tính.

Biểu tượng  $\Theta$  đại diện cho thể hiện.

### Hình 30: Tra cứu định hướng theo cây tìm kiếm

Rice Doctor   **Ontology**   Tư vấn   Tra cứu   Quản trị   Liên hệ

— Thing

**Bệnh**

- ⊖ Giai đoạn
- ⊖ Giống lúa
- ⊖ Hiện tượng bệnh
- + ⊕ Phân
- + ⊕ Quy trình
- + ⊕ Tác nhân gây hại
- ⊖ Thu Hoạch bảo quản sau khi thu hoạch
- + ⊕ Thuốc
- + ⊕ Triệu chứng
- ⊖ Vụ mùa

**Lớp Bệnh**

19 Thể hiện của Bệnh

- ⓘ bệnh bạc lá
- ⓘ Bệnh bướu rễ do tuyến trùng hại lúa
- ⓘ Bệnh cháy bìa lá
- ⓘ bệnh cháy lá
- ⓘ bệnh đóm nâu
- ⓘ Bệnh đốm vàng
- ⓘ bệnh hoa cúc
- ⓘ bệnh khô vẫn
- ⓘ bệnh lem lép hạt
- ⓘ bệnh lúu von
- ⓘ bệnh lùn sọc đen
- ⓘ bệnh lùn xoắn lá

Hình 31: Tra cứu định hướng theo phân cấp khái niệm

Tra cứu dạng cây phân cấp khái niệm được chia làm hai hướng đối tượng: Thông tin cho người dùng và thông tin cho chuyên gia, kỹ sư tri thức. Ví dụ: ở lớp Bệnh, người dùng xem được các loại Bệnh trong Ontology, các lớp cha và con của lớp Bệnh, và những thuật ngữ, từ khoá chuyên môn liên quan. Đối với chuyên gia, họ còn biết thêm Bệnh có những loại thuộc tính và kiểu giá trị nào, các quan hệ có thể có, v.v...

Rice Doctor   **Ontology**   Tư vấn   Tra cứu   Quản trị   Liên hệ

Chọn loại câu hỏi cần tư vấn,  
sau đây là một số câu hỏi thường gặp

- Biết Bệnh hại, tìm Thuốc trị bệnh
- Biết Tác nhân gây hại (Sâu, Cỏ, Óc), tìm Thuốc diệt trừ
- Biết Triệu chứng và các giá trị Môi trường, tìm Bệnh
- Biết lúa hiện đang ở Giai đoạn, tìm Bệnh có thể có trong Giai đoạn này
- Nếu không tìm được câu hỏi phù hợp, chọn tư vấn dạng tổng quát

Nhập loại kết quả cần tìm

Nhập các sự kiện đã biết về cây lúa

Nhập các giá trị môi trường (nếu có)

Độ ẩm	(phần trăm)
Độ sâu ruộng	(mét)
Lượng mưa	(milimét)
Nhiệt độ	(độ)

Xử lý

Hình 32: Giao diện chuẩn đoán sâu bệnh hại cho cây lúa

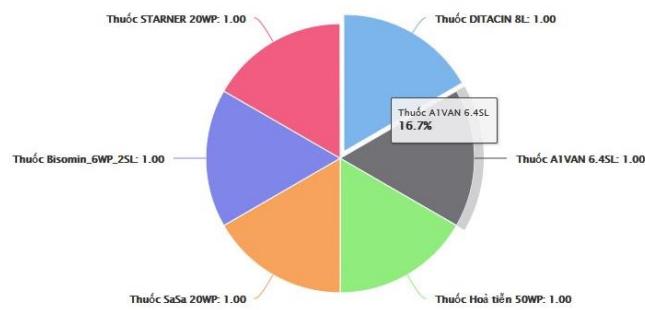
Để giúp người dùng chọn câu hỏi cần tư vấn, hệ thống chuẩn bị sẵn một danh sách các câu hỏi thường gặp để giúp người dùng chọn. Danh sách này được lưu trữ dưới dạng JSON như sau:

```
[  
  {  
    "Type": "Biết Tác nhân gây hại (Sâu, Cỏ, Ốc), tìm Thuốc diệt trừ",  
    "GoalTypes": ["ThuocSauHai", "ThuocTruCo"],  
    "SuggestTypes": ["TacNhanGayHai"],  
    "SuggestFuzzyTypes": false  
  },  
  ...  
]
```

Nếu người dùng không tìm được câu hỏi phù hợp, họ có thể chọn dạng câu hỏi tổng quát: người dùng nhập thủ công các dạng sự kiện mục tiêu cần tìm.

### Chuẩn đoán thành công

Kết quả chuẩn đoán



Highcharts.com

Hình 33: Kết quả trả về của hệ thống chuẩn đoán

Sau quá trình suy diễn và hỏi thêm sự kiện thông tin từ người dùng, hệ thống trả về kết quả chuẩn đoán dưới dạng biểu đồ hình tròn.

### Tra cứu

Câu truy vấn...

**Tra cứu**

Tìm kiếm trên Ontology

Tìm kiếm trên tài liệu web

Hình 34: Giao diện tìm kiếm thông tin trên Ontology và kho tin bài

## Tra cứu

sâu đục thân

Tìm kiếm trên Ontology     Tìm kiếm trên tài liệu web

Tim thấy 17 bài viết liên quan đến câu truy vấn

11-07-2017    **Sâu đục thân năm vạch đầu đen** - 0.707034174216651    Chilo polychrysus Meyrin - Thường xuyên điều tra để dự báo chính xác lứa sâu hại. - Sau khi thu hoạch, thu dọn rơm rạ đốt hoặc ngâm đầm để diệt nguồn sâu. - Chăm sóc hợp lý. - Sử dụng những l... [\[xem thêm\]](#)



11-07-2017    **Sâu đục thân 5 vạch đầu đen (Chilotraea auricillis)** - 0.706438416755817    1. Đặc điểm nhận biết - Ngài: cánh trước màu vàng nâu, mép ngoài cánh có 7 chấm đen. Cánh sau màu nâu vàng nhạt, lông viền cánh màu bạc trắng. - Trứng: hình bầu dục và đẹp. Trứng mới đẻ màu trắng.... [\[xem thêm\]](#)



Hình 35: Giao diện trả về kết quả tìm kiếm dữ liệu trên kho tin bài

## Tra cứu

sâu đục thân

Tìm kiếm trên Ontology     Tìm kiếm trên tài liệu web

Tim thấy 1 dữ liệu Ontology

- **Sâu đục thân**

Hình 36: Giao diện trả về kết quả tìm kiếm trên ontology

Thông tin dạng tóm tắt

Thông tin dạng bài viết

## ❶ Thể hiện đạo ôn

### Mục lục

1. Giới thiệu
2. A Biện pháp khắc phục
3. A Hình ảnh
4. A Nguồn gốc
5. A Triệu chứng
6. A Từ vựng liên quan

### Giới thiệu

đạo ôn là thể hiện của lớp **Bệnh**.

đạo ôn được chữa bởi thuốc A1VAN 6.4SL, Thuốc Binhtin\_75\_WP, Thuốc Bisomin\_6WP\_2SL và Thuốc TAIYOU 20SC.

### Biện pháp khắc phục

Cần áp dụng biện pháp phòng trừ dịch hại tổng hợp IPM:

- Nên chọn mua giống lúa xá nhân ở nhà cung cấp giống tin tưởng, phải có tính ""kháng bệnh" hoặc "kháng vừa" kết hợp với khả năng kháng được **❶ rầy nâu**. Các giống này phải phù hợp với điều kiện đất tại địa phương cũng như năng suất cao và chất lượng tốt. Tùy theo mục đích để đạt chất lượng gạo ngon, dễ tiêu dùng trong nước và xuất khẩu, bà con có thể chọn một số giống như sau: IR64, VNĐ 95-20, VNĐ 99-3, OMCS 2000, OM 1490, MTL 250, OM 3536, VĐ 20, Jasmine 85...

- Nên chọn hạt giống sạch bệnh, khử lẩn tạp hạt cỏ, xử lý một số loại bệnh trên vỏ hạt bằng cách pha 20 cc thuốc CRUISER Plus với 2 lít nước phun lên 100 kg hạt giống trong giai đoạn ủ từ 6-12 giờ trước khi đem đi gieo sa.

- Nên dùng biện pháp撒 hàng với lượng giống trung bình: 80-120 kg/ha

- Bón phân cần đối N-P-K. Không bón thừa phân đạm: 80-100kg N/ha là đủ. Nên bón phân đạm theo nhu cầu cây lúa, áp dụng bằng so màu lá lúa LCC.

- Sau mùa thu hoạch nên cày vùi rơm ra để trả lại nguồn hữu cơ cho đất đồng thời diệt được mầm bệnh; hạn chế đốt rơm vì biện pháp này chỉ trả lại một số chất khoáng có trong tro; ~~đất đắng, đất lèo, đất mặn, đất muối, đất sét~~

Hình 37: Trả về thông tin tóm tắt của bài viết

Thông tin thể hiện được biểu diễn dưới hai dạng: dạng tóm tắt và dạng bài viết. Ở dạng bài viết, thông tin có nội dung đầy đủ hơn, bô cục bài viết biểu diễn rõ ràng, mạch lạc nhờ vào ngôn ngữ Semantic Code được đề cập chương 5.

## Thêm trang tin

Tên trang tin	Ví dụ: TT-GTĐT Vĩnh Phúc
Url	http://nnptnt.vinphuc.gov.vn/Pages/home.aspx
Url trang thẻ loại	http://nnptnt.vinphuc.gov.vn/csdlnongnghiep/pages/luu.aspx?date=&date1=&chkNgayDang=False&chkNgayHa=False&EventID=0
XPath tài liệu ở trang thẻ loại	//a[@class='more' and starts-with(@href, 'luu.aspx')]/@href
XPath tiêu đề ở trang tài liệu	//h1[@class='title']
XPath nội dung ở trang tài liệu	//div[contains(@class, 'fck_detail')]
<b>Thêm</b>	

<< Trở về

Hình 38: Giao diện thêm trang tin

Quản lý trang tin      Quản lý tài liệu

Show  entries      Search:

ID	Tiêu đề tài liệu	Trang tin	Ngày lưu
1	Bọ xít xanh	TT-GTĐT Vĩnh Phúc	2017-07-1
2	Bệnh Đạo ôn	TT-GTĐT Vĩnh Phúc	2017-07-1
3	Bệnh khô vằn	TT-GTĐT Vĩnh Phúc	2017-07-1
4	Bệnh lúa von	TT-GTĐT Vĩnh Phúc	2017-07-1
5	Bệnh chàm bìa lá	TT-GTĐT Vĩnh Phúc	2017-07-1
6	Bệnh đóm nâu	TT-GTĐT Vĩnh Phúc	2017-07-1
7	Bệnh Lem lép hạt	TT-GTĐT Vĩnh Phúc	2017-07-1
8	Bệnh hoa cúc	TT-GTĐT Vĩnh Phúc	2017-07-1
9	Bệnh vàng lui	TT-GTĐT Vĩnh Phúc	2017-07-1
10	Bệnh vàng lùn (lúa cỏ) và bệnh lùn xoắn lá lúa	TT-GTĐT Vĩnh Phúc	2017-07-1

Showing 1 to 10 of 221 entries      Previous      ...

Hình 39: Giao diện quản lý trang tin

## Sửa trang tin - TT-GTĐT Vĩnh Phúc

<b>Id</b>	1
<b>Tên trang tin</b>	TT-GTĐT Vĩnh Phúc
<b>Url</b>	http://nnptnt.vinphuc.gov.vn/Pages/home.aspx
<b>Trang thể loại</b>	http://nnptnt.vinphuc.gov.vn/csdlnongnghiep/pages/luu.aspx?date=&date1=&chkNgayDang=False&chkNgayHa=False&EventID=0&Page=1 XPath tài liệu: //a[@class='more' and starts-with(@href, 'luu.aspx')]/@href XPath tiêu đề: //h1[@class='title'] XPath nội dung: //div[contains(@class, 'fck_detail')]
	<a href="#">Xóa</a>
	http://nnptnt.vinphuc.gov.vn/csdlnongnghiep/pages/luu.aspx?date=&date1=&chkNgayDang=False&chkNgayHa=False&EventID=0&Page=2 XPath tài liệu: //a[@class='more' and starts-with(@href, 'luu.aspx')]/@href XPath tiêu đề: //h1[@class='title'] XPath nội dung: //div[contains(@class, 'fck_detail')]
	<a href="#">Xóa</a>
	http://nnptnt.vinphuc.gov.vn/csdlnongnghiep/pages/luu.aspx?date=&date1=&chkNgayDang=False&chkNgayHa=False&EventID=0&Page=3 XPath tài liệu: //a[@class='more' and starts-with(@href, 'luu.aspx')]/@href
	<a href="#">Xóa</a>

Hình 40: Giao diện xóa, sửa trang tin

## Liên hệ

Bạn có vấn đề thắc mắc cần trao đổi với chuyên gia? Hãy điền vào các thông tin sau và chúng tôi sẽ liên hệ lại với bạn sớm nhất có thể.

<b>E-mail</b>	<input type="text" value="Địa chỉ e-mail"/>
<b>Tên</b>	<input type="text" value="Tên"/>
<b>Tiêu đề</b>	<input type="text" value="Tiêu đề"/>
<b>Nội dung</b>	<input type="text" value="Nội dung cần trao đổi"/>
<input type="button" value="Phản hồi"/>	

Hình 41: Liên hệ với chuyên gia

## **6.2. CÀI ĐẶT VÀ THỬ NGHIỆM**

### **6.2.1. Nền tảng công nghệ**

- Ngôn ngữ sử dụng để cài đặt ứng dụng là ngôn ngữ Java và c#.
- Công cụ Visual Studio 2017, IntelliJ IDEA 2017, .NET Core 1.0, JDK 1.8, Protégé 5.2
- Nền tảng ASP.NET MVC Core để xây dựng website.
- Entity Framework Core: thư viện .NET để truy cập database SQLite.
- Json.NET: thư viện .NET để serialize/deserialize json.
- Bootstrap: hỗ trợ xây dựng giao diện responsive design cho web.
- datatables.net: xây dựng bảng table cho trang quản trị website và tin bài Thư viện hỗ trợ tra cứu theo từ khóa, paging, sorting.
- highcharts: hỗ trợ vẽ piechart để xuất kết quả suy diễn.
- select2: hỗ trợ gợi ý nhập sự kiện inputs và output trong quá trình tư vấn.

### **6.2.2. Cài đặt thử nghiệm - Chuẩn đoán sâu bệnh hại**

#### **6.2.2.1. Cách thức thử nghiệm**

Sau khi đã cài đặt được hệ thống bước tiếp theo là tiến hành thử nghiệm trên hệ thống vừa xây dựng. Cách thử nghiệm là dựa trên những mẫu đã có sẵn, tức là những thí nghiệm, những kinh nghiệm của chuyên gia đã biết trước, tiến hành thử nghiệm lại hệ thống đã xây dựng trên những mẫu đã có sẵn này.

#### **6.2.2.2. Kết quả thử nghiệm chuẩn đoán bệnh**

Bảng so sánh chuyên gia - hệ trợ giúp ra quyết định

ST T	Các sự kiện GT nhập vào	Loại kết quả cần tìm	Các sự kiện GT bổ sung (Trường hợp hệ thống hỏi thêm thông tin)	Kết quả chuẩn đoán của chuyên gia	Kết quả chuẩn đoán của máy

1	MauLa=LaVang HinhDangThan=Th anLun HinhDangRe=ReBi nhThuong HinhDangBong=Bo ngNhieu	Bệnh	Hệ thống không hồi	Bệnh vàng lùn	Bệnh vàng lùn 100%
2	HinhDangThan=Th anLun MauLa=LaXanhDa m HinhDangLa=Mep LaRach HinhDangLa=Mep LaXoan HinhDangHat=Hat Lep	Bệnh	Hệ thống không hồi	Bệnh lùn xoăn lá	Bệnh lùn xoăn lá 100%
3	NuocRuong=Nuoc Cao HinhDangThan=Th anUn MauLa=LaVang	Bệnh	Hệ thống không hồi	Bệnh vàng lá	Bệnh vàng lá 100%
4	LuongMua=700 milimét MoiTruong=GioLo n GiaiDoan=GiaiDoa nLamDong	Bệnh	Hệ thống không hồi	Bệnh đốm văn	Bệnh đốm văn 100%

	MauLa=LaNau HinhDangLa=LaVe chDom				
5	MauLa=LaVang	Bệnh Hiện tượng bệnh	Sự kiện Thân ún Không biết Sự kiện Lá vêch đốm Không biết Sự kiện Thân lùn Không biết	Bệnh vàng lá	Bệnh vàng lá 50% Bệnh khô vắn 40% Hiện tượng ngộ độc phèn 30% Bệnh búu rẽ 20% Bệnh vàng lùn 23% Bệnh cháy bìa lá 30% Hiện tượng thiếu đậm 30%
6	GiaiDoan=GiaiDoa nLamDong NhietDo=22 độ	Bệnh Sâu	Độ ẩm thấp Có Lá nâu Không Nắng nhiều Có Không nước Không biết	Không có kết quả	Bệnh đạo ôn 50%
7	Benh=BenhTungo XuatHienSau=CoSa u	Bệnh Sâu		Sâu Rầy xanh Đuôi đen	Benh Tungo 100% Sâu Rầy xanh đuôi đen 60%
8	NhietDo=25 độ TrieuChung=DeNh anhKem	Bệnh Sâu	Độ ẩm thấm thấp Không biết Độ ẩm cao Không biết Giai đoạn mạ Không biết Lá vàng Không biết Thân ún Không biết	Không chuẩn đoán được kết quả	Bệnh đạo ôn 25% Sâu năn 20%

9	Benh=BenhChayLa	Thuốc trị bệnh	Hệ thống không hồi	Không chuẩn đoán được	Không chuẩn đoán được
10	NuocRuong=Nuoc Cao NhiетDo=30 độ TrieuChung=MuiH oi HinhDangRe=ReIt	Bệnh	Lá vàng Không Lá cháy Có Thân thối Không biết Mưa nhiều Có Gió lớn Không Giai đoạn mạ Không biết	Bệnh thối thân	Bệnh thối thân 45% Bệnh bạc lá 20%
ST T	Các GT nhập vào		Kết quả tìm kiếm	Kết quả chuẩn đoán của chuyên gia	Kết quả chuẩn đoán của máy
1	MauLa=LaVang HinhDangThan=ThanLun HinhDangRe=BinhThuong HinhDangBong=BongNieu		Bệnh	Bệnh vàng lùn	Bệnh Vàng Lùn
2	HinhDangThan=ThanLun MauLa=LaXanhDam HinhDangLa=MepLaRanh HinhDangLa=MepLaXoan HinhDangHat=HatLep		Bệnh	Bệnh Lùn Xoắn Lá	Bệnh Lùn Xoắn Lá
3	NuocRuong=NuocCao HinhDangThan=ThanUn MauLa=MauLaVang		Bệnh	Bệnh Vàng Lá	Bệnh Vàng Lá
4	LuongMua=MuaNhieu MoiTruong=GioLon		Bệnh	Bệnh Đốm Văn	Bệnh Đốm văn, Bệnh Khô Văn

	GiaiDoan=GiaiDoanLamDong MauLa=LaNau HinhDangLa=LaVechDom			
5	TrieuChung=DeNhanhKem ChieuCao=ChieuCaoThap Dat=DatPhen GiaiDoan=GiaiDoanMa MauLa=MauVang	Bệnh Sâu Hiện tượng	Bệnh Vàng lá Ngộ độc phèn	Hiện tượng ngộ độc hữu cơ NgoDocPhen Hiện tượng ngộ độc hữu cơ bệnh vàng lá bệnh vàng lá
6	GiaiDoan=GiaiDoanLamDong NhietDo=NhietDoCao	Bệnh Sâu	Không có kết quả	Không Có kết quả Dự đoán: Đạo ôn 50%
7	Benh=BenhTungo XuatHienSau=CoSau	Bệnh Sâu	Sâu= Rày Xanh Đuôi đen	Benh=BenhTungo Sâu= Rày Xanh Đuôi đen
8	NhietDo=NhietDoCao TrieuChung=DeNhanhKem	Bệnh Sâu	Không chuẩn đoán được kết quả	Bệnh Đạo ÔN 40 % Sâu năn 25 %
9	MauLa=LaVang	Benh	Khong Chuẩn đoán được	Không tìm thấy kết quả chuẩn đoán
10	NuocRuong=NuocNhieu NhietDo=NhietDoCao MauLa=LaVang TrieuChung=MuiHoi HinhDangRe=ReIt	Benh	Bệnh Thối Thân	Bệnh thối thân 30% Bệnh cháy bìa lá 30% đạo ôn 25%

### 6.2.3. Cài đặt thử nghiệm cho tra cứu trên kho tài liệu

Để đánh giá hiệu quả truy tìm của hệ thống đã xây dựng, nhóm nghiên cứu sử dụng hai độ đo cơ bản là độ chính xác (precision) và độ bao phủ (recall) để đo sự thỏa mãn của người dùng với các tin bài mà hệ thống tìm thấy, hai độ đo này được tính bởi công thức như sau:

$$\text{Precision} = S/T$$

$$\text{Recall} = S/U$$

Trong đó:

- S: số tin bài hệ thống trả về được người dùng đánh giá có liên quan
- T: số tin bài hệ thống trả về
- U: số tin bài liên quan có trong kho dữ liệu

Công tác thực nghiệm trong giai đoạn đầu gặp nhiều khó khăn vì đòi hỏi kiến thức của chuyên gia về lĩnh vực và phụ thuộc nhiều vào ngôn ngữ. Hơn nữa, việc đánh giá hiệu quả truy tìm của hệ thống cũng đòi hỏi nhiều công sức của con người trong việc xác định tập tin bài có liên quan đến từng mẫu truy vấn trên tổng số các tin bài có trong kho (cần phải hiểu rõ được nội dung chính của mỗi tin bài đó) để so sánh với kết quả trả về của hệ thống. Ứng với mỗi kho tin bài, thực hiện khảo sát trên các bộ gồm những câu truy vấn và tính toán các độ đo recall, precision tương ứng.

Thông kê kết quả tìm kiếm trên kho tin bài gồm 221 tin bài

STT	Câu truy vấn	Số tin bài hệ thống trả về (T)	Số tin bài trả về được người dùng đánh giá có liên quan (S)	Số tin bài liên quan có trong kho tài liệu (U)	Độ chính xác (S/T)	Độ bao phủ (S/U)
1	Bệnh lem lép hạt	1	1	1	1	1

2	Bệnh rầy nâu	0	0	0	0	0
3	Bệnh vàng lá	8	7	10	0.875	0.7
4	Đạo ôn	38	32	40	0.84	0.8
5	Giai đoạn đồng	0	0	0	0	0
6	Thé nào là lem lép hạt	1	1	1	1	1
7	Tìm bệnh	221	189	245	0.85	0.77
8	Tìm kiểm bệnh trên cây lúa	221				
9	Tìm thuốc	163	146	152	0.89	0.96
10	Tra cứu sâu của cây lúa	197	190	201	0.96	0.94
111	Vụ đông xuân	24	20	30	0.83	0.66
12	Giai đoạn trồ	1	1	1	1	1
13	Thuốc diệt sâu hại	9	5	10	0.55	0.5

14	phân vô cơ cho giai đoạn làm đồng	200	150	220	0.75	0.68
15	bệnh vàng lá, bệnh đạo ôn, bệnh vàng lùn	45	30	50	0.66	0.6

## CHƯƠNG 7. KẾT LUẬN

### 7.1. KẾT QUẢ CỦA ĐỀ TÀI

Mục tiêu ban đầu của đề tài “*Thiết kế, xây dựng hệ thống hỗ trợ khuyến nông trên cây lúa*” là xây dựng một hệ thống với mục đích cho người nông dân tra cứu thông tin về miềnn tri thức lúa nước, chuẩn đoán sâu bệnh hại, ốc, cỏ.. giúp cho người nông dân tìm kiếm thông tin một cách hiệu quả nhất, tập dần bỏ đi tập quán canh tác tự do của người nông dân, quen với tập quán canh tác tìm kiếm thông tin, cập nhật những tri thức mới nhất. Kết quả của nhóm xây dựng được như sau:

- Về mặt khoa học
  - Nghiên cứu phương pháp thiết kế Hệ cơ sở tri thức đặc biệt là hệ chuyên gia.
  - Thu thập tri thức của miềnn tri thức lúa nước.
  - Tìm hiểu được các phương pháp biểu diễn tri thức và kĩ thuật suy diễn đã có, đặc biệt là tiếp cận Ontology. Trên cơ sở đó tìm cách vận dụng, phối hợp sao cho phù hợp với yêu cầu thực tế của ứng dụng mà đề tài hướng tới.
  - Nghiên cứu tri thức thực tế thu thập được, từ đó tìm cách mô hình hóa chúng để có thể đặc tả được trên máy tính sao cho máy tính sử dụng được chúng để giải quyết một số vấn đề đặt ra.
  - Tìm hiểu một số ngôn ngữ Ontology.
  - Sử dụng công cụ Protégé để làm dữ liệu Ontology một cách hiệu quả
  - Xây dựng được mô-đun: Suy diễn, tra cứu, tìm kiếm .
  - Nghiên cứu các trang web nông nghiệp về miềnn tri thức nông nghiệp, phân tích đánh giá cấu trúc của một webpage chứa nội dung nông nghiệp lúa nước.
  - Xây dựng thành công bộ thu thập dữ liệu từ các trang web về lĩnh vực lúa.
  - Sử dụng các kĩ thuật tìm kiếm để xây dựng bộ tìm kiếm trên kho tài liệu được lưu trữ.
- Về mặt ứng dụng

- Nhóm nghiên cứu đã xây dựng hệ thống phần mềm web có khả năng tra cứu thông tin về tri thức lúa, đưa ra chuẩn đoán sâu bệnh hại, khả năng thu thập tin bài từ các trang web, quản lý kho tin bài thu thập được, hỗ trợ tìm kiếm các tin bài. Ứng dụng phục vụ nhu cầu thực tế của người nông dân hiện nay.

## 7.2. HẠN CHẾ CỦA ĐỀ TÀI

Dữ liệu của ứng dụng chỉ mới thu thập còn chưa đầy đủ, tri thức chuyên gia được thu thập chỉ mới ở một khu vực nhỏ của tỉnh Bình Định nên tri thức thu thập được còn thưa thớt, dẫn đến xây dựng bộ luật còn yếu ớt kéo theo việc chuẩn đoán trả về kết quả cho người dùng chưa chính xác như một chuyên gia thực thụ.

Việc xử lý tiếng việt, mở rộng túi từ, mở rộng phạm vi tìm kiếm làm cho kết quả trả về bị nhiễu, độ chính xác khi tìm kiếm sẽ bị giảm.

Sự chính xác của việc thu thập tin bài phụ thuộc nhiều vào cấu trúc của các trang web được định nghĩa. Theo thời gian, một khi cấu trúc thông tin của trang báo điện tử thay đổi thì công việc định nghĩa cấu trúc trang báo điện tử cần được xem xét và thực hiện lại. Công việc quản lý, bảo trì hệ thống còn phụ thuộc vào con người chứ chưa thực sự tự động hoàn toàn.

Ontology lĩnh vực lúa nước được xây dựng nhằm mục đích hỗ trợ cho việc tra cứu tìm kiếm các thông tin liên quan đến lúa, chuẩn đoán sâu bệnh hại, biện pháp phòng trừ,... Tuy nhiên, quy mô còn nhỏ, do đó chưa thực sự bao quát hết lĩnh vực, bên cạnh đó nhân lực còn hạn chế nên việc cải tiến Ontology lúa nước còn chưa được thực hiện.

## 7.3. HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI

Hướng phát triển của đề tài là nghiên cứu mở rộng, phát triển các vấn đề, phương pháp, kỹ thuật mới hiệu quả hơn, trên cơ sở đó, đáp ứng các yêu cầu thực tế tốt hơn bao gồm những yêu cầu mới mà đề tài chưa giải quyết.

Về mặt khoa học kĩ thuật:

- + Cải tiến ontology lúa nước bằng cách chuẩn hóa một số thành phần định nghĩa, bổ sung những thành tố mới vốn chưa đề cập tới trước đó.

+ Nghiên cứu phát triển thêm phương pháp, kỹ thuật suy diễn để cho kết quả tốt hơn, tìm hiểu các phương pháp tìm kiếm mới nhất, hướng tới mục đích tìm kiếm ngữ nghĩa cao cấp hơn,...

Nếu tiếp tục phát triển và nâng cấp khóa luận, nhóm sẽ mở rộng hệ thống trên nhiều lĩnh vực khác thay vì chỉ gói gọn trong miền tri thức lúa nước mà thay vào đó có thể mở rộng sang café, hoa quả, rau,... mở rộng thành một trang web chuyên về nông nghiệp. Mục đích ban đầu khi xây dựng hệ thống là xây dựng một trang web chuyên về nông nghiệp, phục vụ việc tìm kiếm thông tin cho người dùng, thực hiện các chuẩn đoán, đưa ra các lời khuyên, liên hệ với chuyên gia nhờ chuyên gia giải quyết, mang ứng dụng công nghệ thông tin vào sản xuất.

Mặc dù công trình tìm hiểu nghiên cứu còn khó khăn và giới hạn, nhưng những kết quả thành công ban đầu đạt được, cũng là nền tảng cho việc nghiên cứu sau này, hi vọng những thành công ban đầu có thể giúp xã hội có thể phát triển đê tài, vì đây là một vấn đề rất cần thiết đối với người nông dân được chú ý. Nhóm hi vọng đê tài có thể được nhiều người đón nhận với mong muốn: giúp người dân bỏ đi tập tính canh tác tự túc, tạo điều kiện cho các nhà khoa học, người dân chia sẻ và sử dụng tri thức về nông nghiệp, phục vụ cho quá trình sản xuất ngày một hiệu quả hơn.

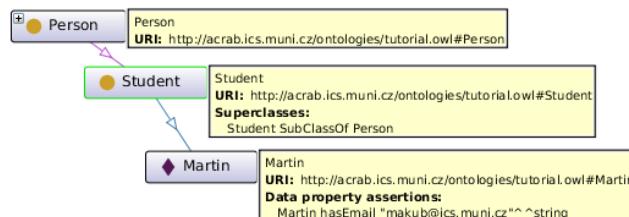
# PHỤ LỤC

## CHƯƠNG 8. PHỤ LỤC

### 8.1. PHỤ LỤC 1: MÔ HÌNH OWL 2 TRONG PROTÉGÉ

#### I. OWL Ontology

- OWL ontology là một tập **các tiên đề**, cung cấp các khảng định logic tường minh về ba loại – **lớp, thể hiện và thuộc tính**. Bằng cách sử dụng một phần mềm gọi là **bộ suy diễn** (reasoner) ta có thể suy ra các sự kiện ngầm chứa trong mô hình ontology. Ví dụ nếu một thể hiện Martin thuộc lớp Student, và lớp Student là lớp con của lớp Person, một bộ suy diễn sẽ suy ra rằng Martin là một Person.
- Có hai loại thuộc tính trong OWL ontology. **Thuộc tính dữ liệu** là các quan hệ nhị phân liên kết một thể hiện đến một giá trị có kiểu, như xsd:dateTime hoặc xsd:string literal. **Thuộc tính đối tượng** là các quan hệ nhị phân liên kết một đối tượng đến một đối tượng.
- Có nhiều cách làm một OWL ontology có thể *được viết*, tuy nhiên điều quan trọng là *ngữ nghĩa*, được thể hiện ở hình ảnh Ontograf dưới đây, được xây dựng bởi công cụ Protégé. Đây là một ontology nhỏ với hai lớp và một thể hiện, trong đó có một thuộc tính dữ liệu:



- Các tiên đề tương tự có thể được viết bằng nhiều cách. Phổ biến nhất là định dạng XML/RDF, đặc tả OWL 2 sử dụng định dạng *functional syntax*, các tác giả Protégé phát triển định dạng ngắn hơn Manchester, và định dạng ngắn gọn nhất là Turtle.

#### II. Các loại tiên đề

- Có rất nhiều loại tiên đề có thể được biểu diễn trong OWL.
  - a. Khai báo
- **Khai báo lớp** định nghĩa một lớp. Một lớp có thể chứa nhiều thể hiện.



Declaration( Class( :Person ) )

- **Khai báo thẻ hiện** định nghĩa một thẻ hiện có tên.



Declaration( NamedIndividual( :Martin ) )

- **Khẳng định lớp** phát biểu một thẻ hiện thuộc về một lớp.



ClassAssertion( :Student :Martin )

- **Khẳng định lớp con** phát biểu tất cả thẻ hiện thuộc một lớp cũng thuộc một lớp khác.



SubClassOf( :Student :Person )

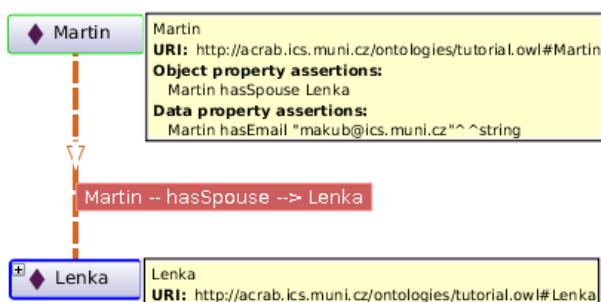
- **Khai báo thuộc tính** định nghĩa hoặc thuộc tính dữ liệu kiến kết một thẻ hiện đến dữ liệu, hoặc thuộc tính đối tượng liên kết đến một thẻ hiện.



Declaration( DataProperty( :hasEmail ) )

Declaration( ObjectProperty( :hasSpouse ) )

- **Khẳng định thuộc tính** phát biểu quan hệ giữa một thẻ hiện đến một dữ liệu hoặc thẻ hiện.



DataPropertyAssertion( :hasEmail :Martin "makub@ics.muni.cz"^^xsd:string )

ObjectPropertyAssertion( :hasSpouse :Martin :Lenka )

- **Khẳng định thuộc tính tiêu cực** phát biểu quan hệ giữa một thể hiện đến một dữ liệu hoặc thể hiện **không tồn tại**. OWL sử dụng Open World Assumption, nên nếu một thể hiện không được liên kết bởi những thuộc tính có những giá trị, nó có thể bị gây ra bởi hai lý do – hoặc nó thực sự không có thuộc tính có giá trị, hoặc nó **không xác định** bởi thông tin còn thiếu từ ontology. Một khẳng định thuộc tính tiêu cực phát biểu rằng thể hiện không thể có giá trị thuộc tính đó.

NegativeObjectPropertyAssertion( :hasSpouse :Martin :Peter)

NegativeDataPropertyAssertion( :hasEmail :Martin "president@whitehouse.gov")

- **Khẳng định chú thích** cho phép chú thích bất kỳ thứ gì với một số thông tin, ví dụ ta có thể dùng ICS như là tên của một thể hiện bởi vì nó là tên viết tắt, và sử dụng chú thích để gán nhãn tên viết tắt với ý nghĩa đầy đủ “Institute of Computer Science”.



Declaration( NamedIndividual( :MU ) )

AnnotationAssertion( rdfs:label :MU "Masaryk University" )

Declaration( NamedIndividual( :ICS ) )

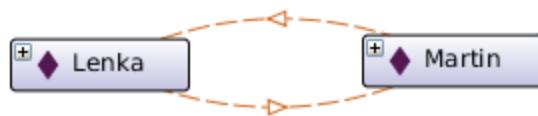
AnnotationAssertion( rdfs:label :ICS "Institute of Computer Science" )

Declaration( NamedIndividual(:SC) )

AnnotationAssertion( rdfs:label :SC "Supercomputing Center" )

b. Tiên đề thuộc tính

- Ta có thể định nghĩa nhiều thông tin về các thuộc tính, một thuộc tính là **transitive**, **symmetric**, **asymmetric**, **reflexive**, **irreflexive**, **functional** (có thể chỉ có một giá trị), **inverse-functional** (nghịch đảo của nó là functional), **inverse** đến các thuộc tính khác, **subproperty** của các thuộc tính khác, **equivalent** đến các thuộc tính khác, hoặc **disjoint** với các thuộc tính khác (hai thể hiện có thể không được liên kết bởi cả hai thuộc tính trong cùng một thời điểm).
- Trong một ví dụ đơn giản, ta có thể định nghĩa một thuộc tính *hasSpouse* là **symmetric**, **functional** và **irreflexive**:

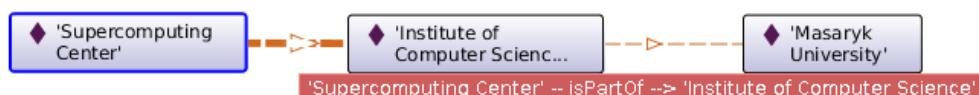


SymmetricObjectProperty( :hasSpouse )

FunctionalObjectProperty( :hasSpouse )

IrreflexiveObjectProperty( :hasSpouse )

- Ở một ví dụ phức tạp hơn, một thuộc tính **transitive** *isPartOf* được định nghĩa, kết nối những bộ phận tổ chức của Đại học Masaryk. Sau đó thuộc tính *hasPart* được định nghĩa như một thuộc tính **inverse** của *isPartOf*.



Declaration( ObjectProperty( :isPartOf ))

TransitiveObjectProperty( :isPartOf )

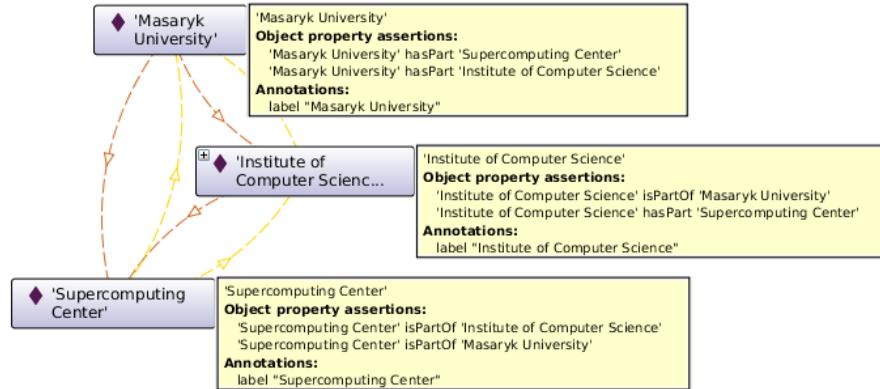
ObjectPropertyAssertion( :isPartOf :SC :ICS )

ObjectPropertyAssertion( :isPartOf :ICS :MU )

Declaration( ObjectProperty(:hasPart) )

InverseObjectProperties( :hasPart :isPartOf )

- Giờ ta có thể sử dụng một **bộ suy diễn** để suy ra các thuộc tính khác. Ví `isPartOf` là transitive, SCB `isPartOf` MU, và vì `hasPart` là nghịch đảo của `isPartOf`, ta có được các quan hệ được suy ra hoàn chỉnh.



`ObjectPropertyAssertion( :isPartOf :SC :MU )`

`ObjectPropertyAssertion( :hasPart :MU :SC )`

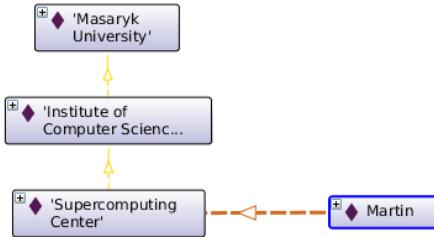
`ObjectPropertyAssertion( :hasPart :MU :ICS )`

`ObjectPropertyAssertion( :hasPart :ICS :SC )`

- Nó cho thấy sức mạnh của suy diễn – ta không cần phải khảo báo tất cả quan hệ, chỉ cần những quan hệ tối thiểu, và các quan hệ còn lại có thể được suy ra.

### c. Mắt xích thuộc tính

- **Mắt xích thuộc tính** cho phép định nghĩa những quan hệ giữa ba thẻ hiện, ví dụ nổi bật nhất là thuộc tính *uncle* được định nghĩa như một mắt xích của thuộc tính *parent* và *brother*.
- Một thuộc tính có thể mắt xích với chính nó. Ví dụ, ta định nghĩa thuộc tính *isEmployedAt*, là một mắt xích của chính nó và thuộc tính transitive *isPartOf*, nghĩa là nếu một người đi làm ở những đơn vị tổ chức thì người đó cũng làm ở các đơn vị tổ chức lớn hơn.

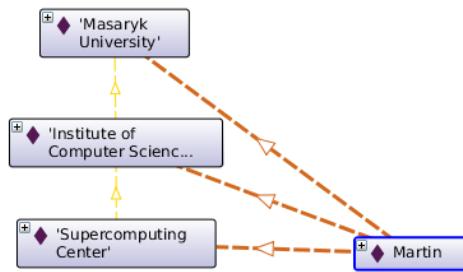


Declaration( ObjectProperty( :isEmployedAt ) )

ObjectPropertyAssertion( :isEmployedAt :Martin :SC )

SubObjectPropertyOf( ObjectPropertyChain( :isEmployedAt :isPartOf ) :isEmployedAt)

- Một lần nữa, ta sử dụng bộ suy diễn để suy ra các quan hệ hoàn chỉnh.

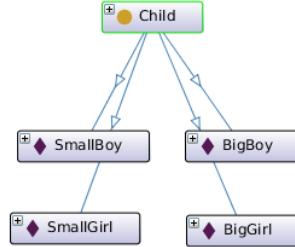


ObjectPropertyAssertion( :isEmployedAt :Martin :ICS )

ObjectPropertyAssertion( :isEmployedAt :Martin :MU )

#### d. Biểu thức lớp

- Lớp có thể được định nghĩa bằng **biểu thức lớp**. Một định nghĩa lớp thông thường là một khảng định một lớp có tên **tương đương** với những lớp (không tên/vô danh) được định nghĩa bằng một biểu thức. Lớp cũng có thể được định nghĩa là **disjoined** với lớp khác, nghĩa là chúng không chia sẻ thể hiện nào với nhau.
- **Các phép trong tập (set): liệt kê, hợp, giao, hội**
- Ví dụ, ta định nghĩa một lớp mới *Child* chứa bốn thể hiện mới có tên *SmallBoy*, *BigBoy*, *SmallGirl*, *BigGirl*. Những thể hiện này phải được khai báo khác nhau, nếu không một bộ suy diễn OWL có thể xem chúng là như nhau.



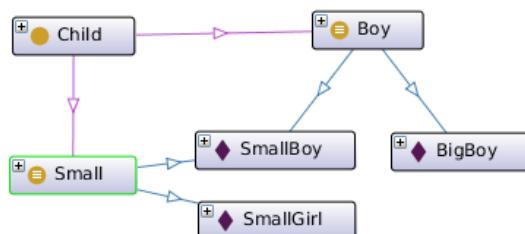
```

Declaration( Class( :Child ) )
Declaration( NamedIndividual( :BigBoy ) )
Declaration( NamedIndividual( :BigGirl ) )
Declaration( NamedIndividual( :SmallBoy ) )
Declaration( NamedIndividual( :SmallGirl ) )
ClassAssertion( :Child :BigBoy )
ClassAssertion( :Child :BigGirl )
ClassAssertion( :Child :SmallBoy )
ClassAssertion( :Child :SmallGirl )
DifferentIndividuals( :BigBoy :BigGirl :SmallBoy :SmallGirl )

```

i. Phép liệt kê

- Ta định nghĩa hai lớp cơ sở bằng cách liệt kê các thành viên, lớp *Boy* chứa những con trai và lớp *Small* chứa những đứa trẻ nhỏ con. Ví dụ, thẻ hiện *SmallBoy* vừa trong lớp *Boy* và lớp *Small*.



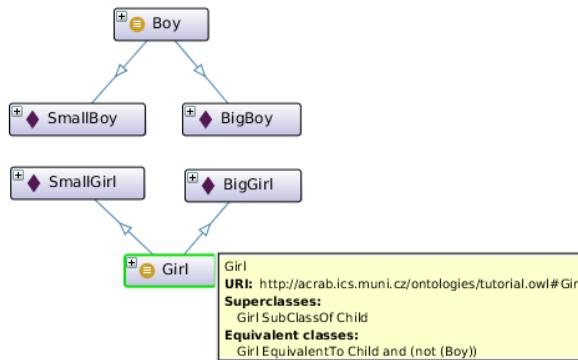
```

SubClassOf(:Boy :Child)
SubClassOf(:Small :Child)
EquivalentClasses(:Boy ObjectOneOf(:SmallBoy :BigBoy))
EquivalentClasses(:Small ObjectOneOf(:SmallBoy :SmallGirl))

```

ii. Phép hội và giao

- Giờ ta định nghĩa lớp mới *Girl* là lớp **giao** giữa lớp *Child* và lớp không tên là **hội** của lớp *Boy*. Lớp *Girl* tương đương với **Child and not Boy**, nghĩa là một con gái là một đứa trẻ không phải là một con trai.
- Bô suy diễn sẽ tìm ra rằng lớp *Girl* chứa những thẻ hiện *SmallGirl* và *BigGirl*.



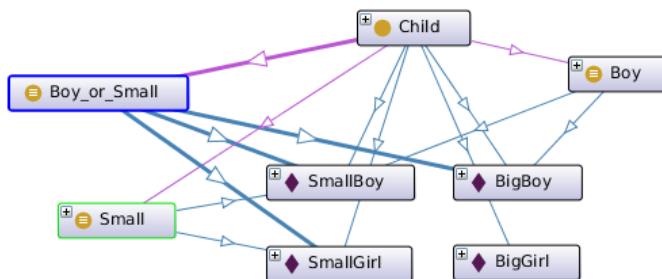
Declaration( Class( :Girl ) )

SubClassOf( :Girl :Child )

EquivalentClasses( :Girl ObjectIntersectionOf( ObjectComplementOf( :Boy ) :Child ) )

### iii. Phép hợp

- Ta có thể định nghĩa một lớp mới *Boy\_or\_small* là **hợp** giữa lớp *Small* và *Boy*, nghĩa là **Boy or Small**. Bô suy diễn sẽ tìm ra rằng lớp chứa những thẻ hiện *SmallBoy*, *SmallGirl*, *BigBoy*.



Declaration( Class( :Boy\_or\_Small ) )

EquivalentClasses( :Boy\_or\_Small ObjectUnionOf( :Small :Boy ) )

### e. Định lượng hiện sinh (Existential quantification)

- Một biểu thức lớp có thể nói rằng lớp chỉ chứa những thẻ hiện được kết nối bởi một thuộc tính cho trước với những thẻ hiện từ một lớp cho trước. Ta có thể định nghĩa một lớp mới *OrgUnit* chứa ba thẻ hiện MU, ICS, SC.

```

Declaration( Class( :OrgUnit ))
ClassAssertion( :OrgUnit :ICS )
ClassAssertion( :OrgUnit :MU )
ClassAssertion( :OrgUnit :SC )

```

- Sau đó ta có thể định nghĩa một lớp mới *Employee* là lớp con của *Person*, nghĩa là **tương đương** với một lớp vô danh của những thẻ hiện được liên kết bởi thuộc tính *isEmployedAt* đến những thẻ hiện từ lớp *OrgUnit*. Nghĩa là những nhân viên là những người được tuyển dụng ở đâu đó.

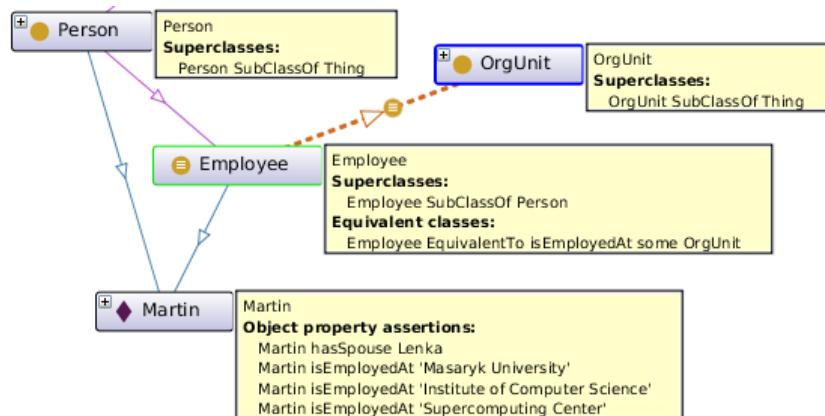


```

Declaration( Class( :Employee ))
SubClassOf( :Employee :Person )
EquivalentClasses( :Employee ObjectSomeValuesFrom( :isEmployedAt :OrgUnit ) )

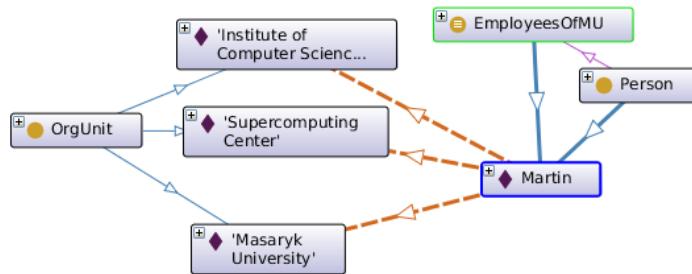
```

- Bằng sử dụng một bộ suy diễn ta có thể suy ra rằng lớp Employee chứa thẻ hiện Martin.



f. Ràng buộc giá trị thẻ hiện

- Một biểu thức lớp cũng có thể nói rằng lớp chứa **những thể hiện được kết nối bởi một thuộc tính cho trước với một thể hiện cho trước**.
- Ví dụ, ta định nghĩa một lớp mới *EmployeesOfMU* chứa những thể hiện được kết nối bởi thuộc tính *isEmployedAt* với thể hiện *MU*, nghĩa là những người làm việc tại MU. Bộ suy diễn sẽ tìm thấy lớp chứa thể hiện *Martin*.



Declaration( Class( :EmployeesOfMU ) )

SubClassOf( :EmployeesOfMU :Person )

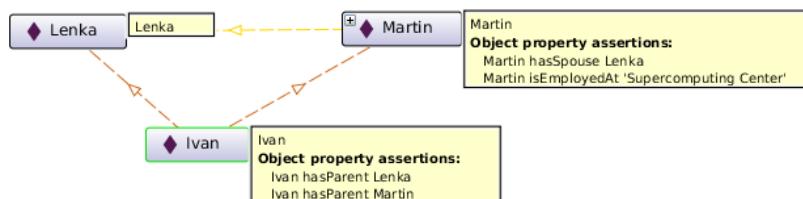
EquivalentClasses( :EmployeesOfMU ObjectHasValue( :isEmployedAt :MU ) )

#### g. Định lượng phổ quát (Universal quantification)

- Định lượng phổ quát (phép  $\forall$  trong toán học) trong OWL có một chút khác biệt bởi tính **open world assumption** mà OWL có.
- Một định lượng hiện sinh được mô tả ở trên (Phép *ObjectSomeValuesFrom()* của OWL) được định nghĩa trong OWL 2 Direct Semantics phát biểu “những thể hiện mà *có sự tồn tại* thể hiện từ một lớp cho trước được kết nối bởi một thuộc tính cho trước”, trong ký hiệu toán học *ObjectSomeValuesFrom( OPE CE )* : {  $x \mid \exists y : (x, y) \in (OPE)^{OP}$  and  $y \in (CE)^C$  } với OPE biểu thị một biểu thức thuộc tính đối tượng,  $.^{OP}$  là hàm thuộc tính đối tượng diễn dịch (interpretation), CE biểu thị một biểu thức lớp,  $.^C$  là hàm lớp diễn dịch.
- Định lượng phổ quát được định nghĩa là phép *ObjectAllValuesFrom( OPE CE )* : {  $x \mid \forall y : (x, y) \in (OPE)^{OP}$  với  $y \in (CE)^C$  } nghĩa là **những thể hiện mà luôn được kết nối bởi thuộc tính cho trước chỉ đến những thể hiện từ lớp cho trước**.
- Vấn đề với open world assumption là trong một thế giới mở một ontology có thể không chứa tất cả những thông tin. Ví dụ, có một ontology phát biểu rằng thể hiện Peter có

hai đứa con, con trai John và Paul. Ta có thể định nghĩa một lớp *ManWhoHaveOnlySons* là lớp của các thể hiện được kết nối bởi thuộc tính *hasChild* chỉ đến những thể hiện từ lớp *Boy*, nghĩa là *ObjectAllValuesFrom( :hasChild :Boy)*. Tuy nhiên thể hiện Peter không nhất thiết trong một lớp, vì Peter cũng có thể có một đứa con gái Jane, chỉ là ontology không chứa thông tin này.

- Định lượng phổ quát trong OWL chỉ hoạt động với **giới hạn số lượng** (cardinality restrictions), nên nếu ta biết rằng một thể hiện có thể được kết nối bởi một lớp cho trước đến một số **tối đa** các thể hiện khác nhau, và ta biết toàn bộ thể hiện, ta có thể rút ra được kết luận.
- Ví dụ, mỗi người có tối đa hai cha mẹ (parents). Nếu cả hai cha mẹ đều có cùng quốc tịch, thì người đó cũng có cùng quốc tịch.
- Ta tạo một thể hiện mới *Ivan*, được kết nối bởi một thuộc tính đối tượng mới *hasParent* đến những thể hiện *Martin* và *Lenka*. Nghĩa là Ivan là một đứa con của Martin và Lenka.



`Declaration( NamedIndividual( :Ivan ) )`

`ClassAssertion( :Person :Ivan )`

`Declaration( ObjectProperty( :hasParent ) )`

`ObjectPropertyAssertion( :hasParent :Ivan :Martin )`

`ObjectPropertyAssertion( :hasParent :Ivan :Lenka )`

- Ta có thể nói rằng một Person có tối đa hai cha mẹ, và định nghĩa một lớp mới *Czech* chứa *Martin* và *Lenka*. Điều quan trọng mới là ở hai tiên đề cuối cùng – định lượng phổ quát rằng các thể hiện có tất cả các cha mẹ Czech là (một lớp con của) người Czech, và mọi thể hiện có tên của ontology là khác nhau.

`SubClassOf( :Person ObjectMaxCardinality( 2 :hasParent ) )`

```

Declaration( Class( :Czech ) )
SubClassOf( :Czech :Person )
ClassAssertion( :Czech :Lenka )
ClassAssertion( :Czech :Martin )

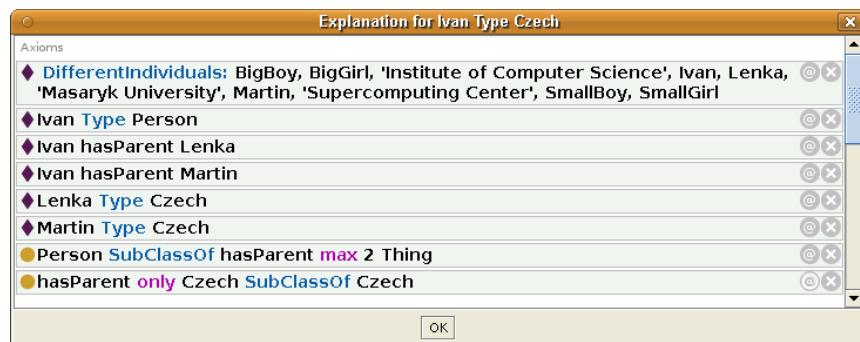
```

```

SubClassOf( ObjectAllValuesFrom( :hasParent :Czech ) :Czech)
DifferentIndividuals( :BigBoy :BigGirl :ICS :Ivan :Lenka :MU :Martin :SC :SmallBoy
:SmallGirl)

```

- Với những tiên đề này, ta có một kết luận rút ra được từ bộ suy diễn là – **Ivan thuộc lớp Czech**. Anh ấy là một Person, nên anh ấy có thể có tối đa hai cha mẹ, anh ấy có hai cha mẹ là hai thể hiện khác nhau, và đều là người Czech. Do đó không có thêm cha mẹ nào tồn tại cho anh, và định lượng phô quát được thỏa mãn – anh ấy chỉ có cha mẹ người Czech.



#### h. Ràng buộc kiểu dữ liệu

- Đối với các thuộc tính dữ liệu, ta có thể ràng buộc các giá trị của một thuộc tính dữ liệu. Ví dụ như tiên đề sau định nghĩa rằng những thể hiện của lớp Person có đúng một giá trị của thuộc tính hasAge là một số nguyên trong khoảng từ 0 đến 130.

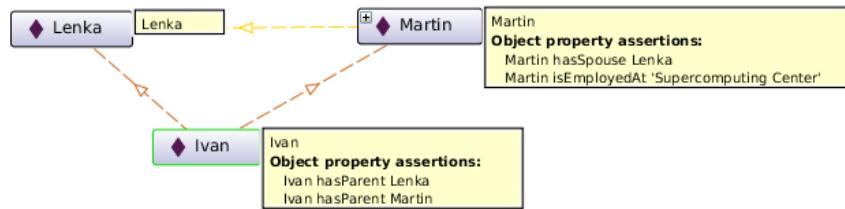
```

EquivalentClasses( :Person DataExactCardinality( 1 :hasAge DatatypeRestriction(
xsd:integer xsd:minInclusive "0"^^xsd:integer xsd:maxInclusive "130"^^xsd:integer)))

```

### III. Luật SWRL

- Ngôn ngữ OWL 2 không thể thể hiện tất cả các quan hệ. Ví dụ như không thể biểu diễn quan hệ *child of married parents*, vì không thể trong OWL 2 để diễn tả quan hệ giữa các thể hiện mà một thể hiện có những quan hệ tới.
- Cách biểu diễn của OWL có thể được mở rộng bằng luật SWRL (Semantic Web Rule Language) cho một ontology. Các luật SWRL tương tự như ngôn ngữ Prolog hay DATALOG. Thực tế, luật SWRL là luật DATALOG với unary predicates để mô tả lớp và kiểu dữ liệu, binary predicates cho thuộc tính, và những predicates built-in n-ary đặc biệt.
- Với ví dụ về Ivan, đứa con của Martin và Lenka, thuộc tính symmetric *hasSpouse* kết nối Martin và Lenka.



SymmetricObjectProperty(:hasSpouse)

ObjectPropertyAssertion(:hasSpouse :Martin :Lenka)

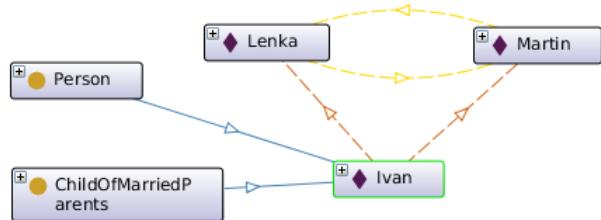
ObjectPropertyAssertion( :hasParent :Ivan :Martin )

ObjectPropertyAssertion( :hasParent :Ivan :Lenka )

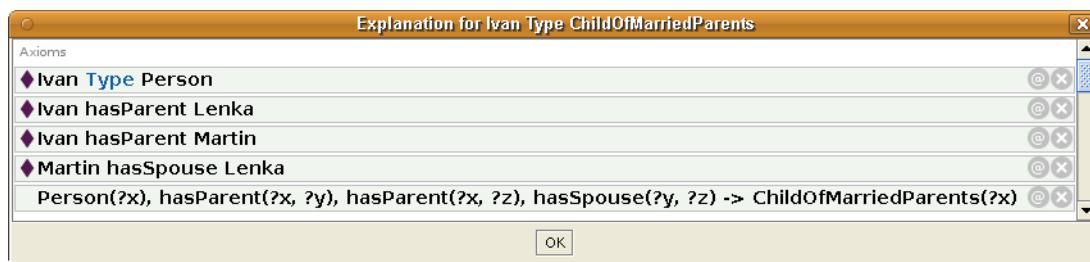
- Ta có thể thêm một luật SWRL nói rằng một thể hiện X trong lớp Person, mà có cha mẹ Y và Z với Y có vợ/chồng là Z, sẽ thuộc lớp mới *ChildOfMarriedParents*.

Person(?x),    hasParent(?x,    ?y),    hasParent(?x,    ?z),    hasSpouse(?y,    ?z)    ->  
ChildOfMarriedParents(?x)

- Khi ta sử dụng bộ suy diễn có hỗ trợ luật SWRL, nó suy ra được *Ivan* thuộc lớp *ChildOfMarriedParents*.



ClassAssertion( :ChildOfMarriedParents :Ivan )



#### IV. Thẻ hiện khác biệt và khóa

- **Open world assumption** của OWL nói rằng một ontology có thẻ không chứa tất cả mọi dữ liệu và do đó vài thông tin có thẻ là không xác định. Đây là lý do vì sao **những thẻ hiện có tên khác nhau có thể đại diện cho cùng thẻ hiện** trừ khi nó được tường minh phát biểu rằng chúng khác nhau. Cũng là lý do vì sau ta phải chỉ định ở những ví dụ bên trên (các phép trong tập và định lượng phô quát) rằng những thẻ hiện là khác nhau.
- Ta có thể khai báo các thẻ hiện có tên khác nhau bằng hai cách. Cách một là bằng tiền đề DifferentIndividuals()

DifferentIndividuals(:BigBoy :BigGirl :SmallBoy :SmallGirl :ICS :MU :SC :Ivan :Lenka :Martin )

- Tuy nhiên cách này được cho là gây vấn đề về hiệu năng cho bộ suy diễn khi số lượng thẻ hiện lớn. Cách thứ hai để chỉ những thẻ hiện khác nhau là bằng thuộc tính dữ liệu function với những giá trị duy nhất. Thuộc tính functional chỉ có thẻ có một giá trị, do đó làm cho các thẻ hiện **khác nhau** khi giá trị của chúng khác nhau. Nếu thuộc tính được khai báo là một **khóa**, nó làm cho các thẻ hiện có **cùng giá trị** là cùng một thẻ hiện.

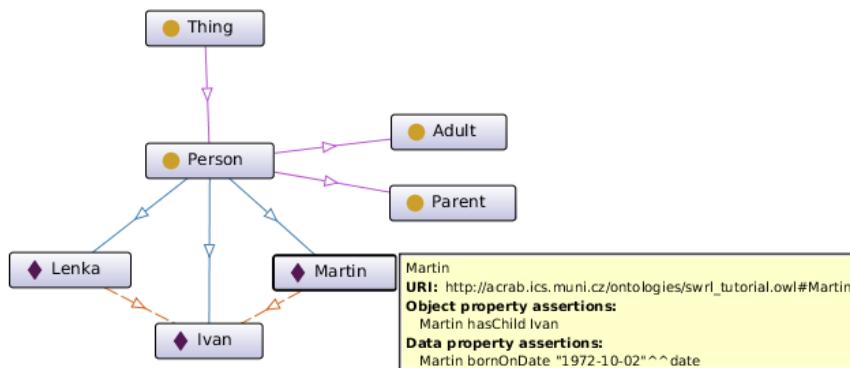
```

Declaration(DataProperty(:hasId))
FunctionalDataProperty(:hasId)
HasKey( :Person () (:hasId) )
DataPropertyAssertion(:hasId :Lenka "1234")
DataPropertyAssertion(:hasId :Martin "5648")

```

## V. SWRL predicates

- Luật SWRL có thể sử dụng những predicate khác hơn là chỉ với tên lớp và tên thuộc tính. Những predicate có thể là
  - o **Biểu thức lớp** – biểu thức lớp tùy ý, không chỉ với lớp có tên
  - o **Biểu thức thuộc tính** – hiện không có nhu cầu sử dụng biểu thức thuộc tính trong SWRL
  - o **Ràng buộc miền dữ liệu** – chỉ định kiểu cho giá trị dữ liệu, như số nguyên, ngày, hợp của các kiểu XML Schema, kiểu liệt kê
  - o **sameIndividual** và **differentIndividuals** – để xác định thẻ hiện giống và khác nhau
  - o **core SWRL built-ins** – những predicate đặc biệt được định nghĩa trong SWRL proposal để thao tác với các giá trị dữ liệu, ví dụ như cộng thêm một số
  - o **custom SWRL built-ins** – ta có thể tự định nghĩa các built-ins bằng code
- Ví dụ, một ontology đơn giản có luật, nó định nghĩa ba thẻ hiện với ngày sinh:



```

Declaration(Class(:Person))
Declaration(ObjectProperty(:hasChild))

```

Declaration(Class(:Parent))  
 Declaration(Class(:Adult))  
 SubClassOf(:Adult :Person)  
 SubClassOf(:Parent :Person)  
 Declaration(DataProperty(:bornOnDate))  
 Declaration(DataProperty(:bornInYear))  
 Declaration(DataProperty(:hasAge))  
 FunctionalDataProperty(:hasAge)  
 Declaration(DataProperty(:hasDriverAge))

Declaration(NamedIndividual(:Martin))  
 ClassAssertion(:Person :Martin)  
 DataPropertyAssertion(:bornOnDate :Martin "1972-10-02"^^xsd:date)

Declaration(NamedIndividual(:Lenka))  
 ClassAssertion(:Person :Lenka)  
 DataPropertyAssertion(:bornOnDate :Lenka "1975-11-10"^^xsd:date)

Declaration(NamedIndividual(:Ivan))  
 ClassAssertion(:Person :Ivan)  
 DataPropertyAssertion(:bornOnDate :Ivan "2006-04-14"^^xsd:date)

ObjectPropertyAssertion(:hasChild :Martin :Ivan)  
 ObjectPropertyAssertion(:hasChild :Lenka :Ivan)

- Ontology cũng định nghĩa một vài luật:

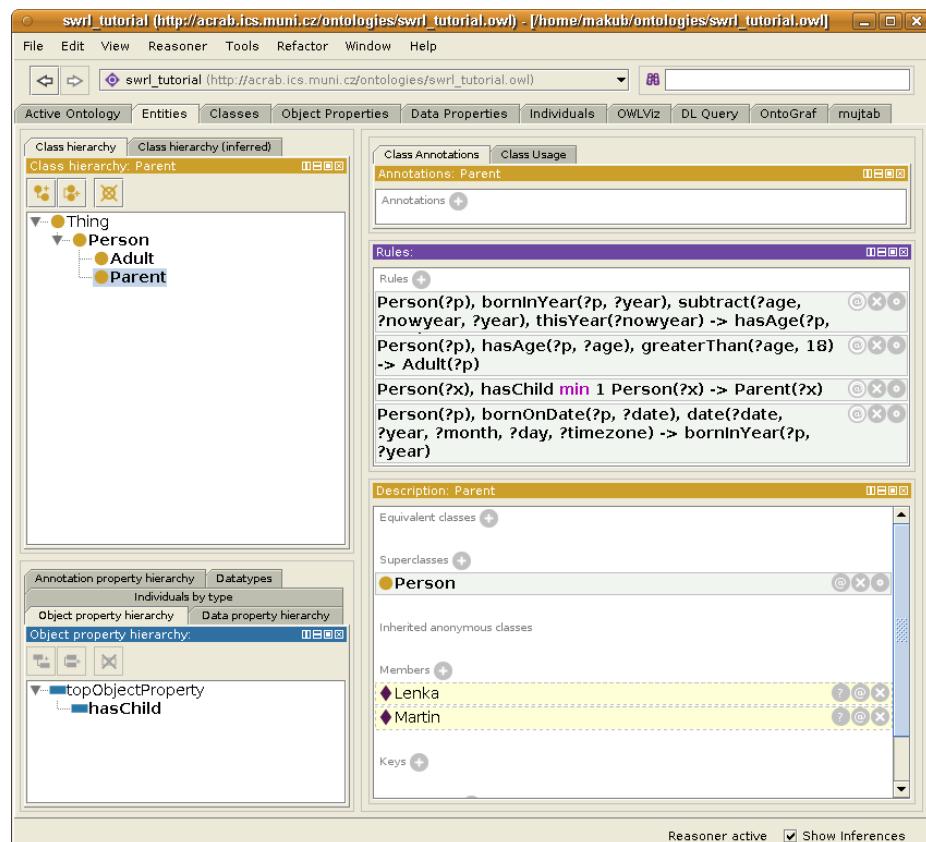
Rules	
Person(?p), bornInYear(?p, ?year), subtract(?age, ?nowyear, ?year), thisYear(?nowyear) -> hasAge(?p, ?age)	⊕ ⊗ ⊖ ⊕
Person(?p), hasAge(?p, ?age), greaterThan(?age, 18) -> Adult(?p)	⊕ ⊗ ⊖ ⊕
Person(?p), integer[>= 18 , <= 65](?age), hasAge(?p, ?age) -> hasDriverAge(?p, true)	⊕ ⊗ ⊖ ⊕
Person(?x), hasChild min 1 Person(?x) -> Parent(?x)	⊕ ⊗ ⊖ ⊕
Person(?p), date(?date), bornOnDate(?p, ?date), date(?date, ?year, ?month, ?day, ?timezone) -> bornInYear(?p, ?year)	⊕ ⊗ ⊖ ⊕

## VI. Luật SWRL với biểu thức lớp

- Luật ở danh sách trên được biểu diễn như

$\text{Person}(\text{x}), \text{hasChild} \text{ min } 1 \text{ Person}(\text{x}) \rightarrow \text{Parent}(\text{x})$

- Nghĩa là với những thể hiện, mà thuộc lớp Person, và có ít nhất một thuộc tính hasChild với một thể hiện thuộc lớp Person, phải thuộc lớp Parent. Hay nói cách khác, những người đó với có ít nhất một đứa con là cha mẹ.
- Khi sử dụng bộ suy diễn, nó sẽ suy ra rằng Martin và Lenka ở cùng lớp Parent:



- Cũng trong ví dụ này, ta có thể mô hình hóa lại đơn giản bằng thiết lập lớp Parent là lớp con của biểu thức lớp “(hasChild min 1 Person)” trong OWL. Hoặc là nó tương đương với biểu thức lớp, là một khẳng định cao hơn. Luật SWRL chỉ thực sự cần cho mô hình cấu trúc không phải dạng cây, không thể mô hình trong OWL.

## VII. Luật SWRL với ràng buộc miền dữ liệu

- Luật sau sử dụng ràng buộc miền dữ liệu:

`Person(?p), integer[>= 18 , <= 65](?age), hasAge(?p, ?age) -> hasDriverAge(?p, true)`

- Ràng buộc thỏa mãn khi biến ?age có giá trị số nguyên giữa 18 và 65.
- Ví dụ này biểu diễn một ràng buộc miền dữ liệu phức tạp với facets, một ví dụ đơn giản sẽ là `integer(?x)` hoặc `xsd:date(?y)` nhằm ràng buộc kiểu mà không xài facets.
- Một ví dụ miền dữ liệu sử dụng phép hội, hợp, giao và facets cùng một lúc:

`((integer[>=1,<=2] or integer[>5,<7]) and not ({0}))(?y)`

## VIII. Luật SWRL với core built-ins

- SWRL proposal cũng định nghĩa những predicate đặc biệt, dùng để thao tác với các giá trị dữ liệu. Những predicate này được gọi là SWRL core built-ins, và được xác định bởi các IRI trong namespace swrlb.
- Những luật sau từ danh sách trên sử dụng core built-ins:

`Person(?p), hasAge(?p, ?age), swrlb:greaterThan(?age, 18) -> Adult(?p)`

`Person(?p), bornOnDate(?p, ?date), xsd:date(?date), swrlb:date(?date, ?year, ?month, ?day, ?timezone) -> bornInYear(?p, ?year)`

- Luật thứ nhất nói rằng những người có tuổi lớn hơn 18 là người lớn. Luật thứ hai lấy giá trị dữ liệu từ thuộc tính bornOnDate, phải có kiểu xsd:date, phân nhỏ nó thành nhiều phần, và thêm phần năm thành giá trị của thuộc tính bornInYear cho người đó.



## IX. Luật SWRL với custom built-ins

- Ta có thể tự định nghĩa SWRL built-ins.
- Luật sau sử dụng my:this year custom-built-in để bind giá trị số nguyên của năm hiện tại cho biến ?nowyear, và dùng swrlb:subtract core built-in để tính tuổi theo năm, và gán kết quả thành giá trị của thuộc tính hasAge:

Person(?p), bornInYear(?p, ?year), my:thisYear(?nowyear), swrlb:subtract(?age, ?nowyear, ?year) -> hasAge(?p, ?age)

## X. Giới hạn của OWL 2 và SWRL

- Propositional logic
- SWRL chỉ là một mảnh của first order predicate logic, SWRL không thể biểu diễn những loại sau:
  - o Biểu thức mờ - “It often rains in autumn.”
  - o non-monotonicity - “Birds fly, penguin is a bird, but penguin does not fly.”
  - o propositional attitudes - “Eve thinks that 2 is not a prime number.” (It is true that she thinks it, but what she thinks is not true.)
  - o modal logic
    - possibility and necessity - “It is possible that it will rain today.”
    - epistemic modalities - “Eve knows that 2 is a prime number.”
    - temporal logic - “I am always hungry.”
    - deontic logic - “You must do this.”

## TÀI LIỆU THAM KHẢO

- [1]TS.Đỗ Văn Nhơn, “ Biểu diễn tri thức và suy luận” ,Trường Đại học Công Nghệ Thông Tin- ĐHQG tp HCM.
- [2]GS.TSKH Hoàng Kiếm, TS. Đỗ Phúc, TS. Đỗ Văn Nhơn, “Các hệ cơ sở tri thức”, Trường Đại học Công Nghệ Thông Tin- ĐHQG tp HCM.
- [3] Hoàng Hữu Hạnh (2008), “Web ngữ nghĩa: Những thách thức và hướng tiếp cận mới”, Tạp chí Khoa học, Đại học Huế, Số 48, tr. 31-40.
- [4]"Ontology Development 101: A Guide to Creating Your First Ontology",[http://protege.stanford.edu/publications/ontology\\_development/ontology\\_101.pdf](http://protege.stanford.edu/publications/ontology_development/ontology_101.pdf),- 9/7/3027
- [5]"Extended Backus–Naur form",[https://en.wikipedia.org/wiki/Extended\\_Backus–Naur\\_form](https://en.wikipedia.org/wiki/Extended_Backus–Naur_form),- 9/7/3027
- [6]"Learn OWL and RDFS",<http://www.cambridgesemantics.com/semantic-university/learn-owl-and-rdfs>,- 9/7/3027
- [7]"Learn RDF",<http://www.cambridgesemantics.com/semantic-university/learn-rdf>,- 9/7/3027
- [8]"Linked Data in HTML",<https://rdfa.info/>,- 9/7/3027
- [9]"OWL Web Ontology Language Reference",<http://www.w3.org/TR/owl-ref/>,- 9/7/3027
- [10]"BBCode",<https://www.bbcode.org>,- 9/7/3027
- [11]"Query Expansion techniques",[http://www.cfilt.iitb.ac.in/resources/surveys/survey\\_ashish\\_multiStruct\\_PRF.pdf](http://www.cfilt.iitb.ac.in/resources/surveys/survey_ashish_multiStruct_PRF.pdf),- 9/7/3027
- [12]R. Akerkar and P. Sajja, “*Knowledge-Based Systems*”. Jones & Bartlett Learning, 2010.
- [13]Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, “*Introduction to Information Retrieval*”, Cambridge University Press. 2008.

- [14] Huỳnh Thị Thanh Thương, Phạm Nguyễn Trường An, Nghiên cứu phát triển giải pháp quản trị kho tài nguyên học tập theo ngữ nghĩa, Đề tài NCKH cấp cơ sở, Trường Đại học Công nghệ thông tin - ĐHQG TP.HCM, 2016.
- [15] Lê Tấn Hùng, Từ Minh Phương, Huỳnh Quyết Thắng, Tác tử công nghệ phần mềm hướng tác tử. Nhà xuất bản khoa học về kỹ thuật, Hà Nội, 2006.
- [16]. Sowa J.F., Conceptual Graphs for a DataBase Interface, IBM Journal of Research and Development 20(4), 336–357, July, (1976).
- [17] Nguyễn Hoàng Tú Anh, Nguyễn Trần Kim Chi, Nguyễn Hồng Phi, Mô hình biểu diễn văn bản thành đồ thị. Trường Đại học Khoa học Tự nhiên, ĐHQG –HCM (Bài nhận ngày 09 tháng 04 năm 2008, hoàn chỉnh sửa chữa ngày 26 tháng 09 năm 2008)
- [18] GS.TSKH. Hoàng Kiếm, ThS. Đinh Nguyễn Anh Dũng. Giáo trình Nhập môn trí tuệ nhân tạo. Trường Đại học Công Nghệ Thông Tin- ĐHQG tp HCM.
- [19] Stuart Russell và Peter Norving (2010), Artificial Intelligence – A modern approach (third edition), prentice Hall
- [20] Passino K., Yurkovich S., Fuzzy control, Addison Wesley, 1998
- [21] Nguyễn Đinh Thuân, Các Hệ cơ sở tri thức KBS: Knowledge Based Systems, Khoa Công nghệ Thông tin Đại học Nha Trang, 2007
- [22] Lars Marius Garshol, BNF and EBNF: What are they and how do they work?, 2008
- [23]Quine–McCluskey algorithm,”[http://www.mathematik.uni-marburg.de/~thormae/lectures/ti1/code/qmc/”](http://www.mathematik.uni-marburg.de/~thormae/lectures/ti1/code/qmc/), 9/7/2017
- [24]pertrick Method,” [https://www.allaboutcircuits.com/technical-articles/prime-implicant-simplification-using-petricks-method/”](https://www.allaboutcircuits.com/technical-articles/prime-implicant-simplification-using-petricks-method/), 9/7/2017