TRƯỜNG ĐẠI HỌC ĐIỆN LỰC

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CHUYÊN ĐỀ HỌC PHẦN**

**HỆ CHUYÊN GIA**

**ĐỀ TÀI: XÂY DỰNG HỆ CHUYÊN GIA**

**Y TẾ.**

**Sinh viên thực hiện : CAO ANH ĐỨC**

**Giảng viên hướng dẫn : VŨ VĂN ĐỊNH**

**Ngành : CÔNG NGHỆ THÔNG TIN**

**Chuyên ngành : CÔNG NGHỆ PHẦN MỀM**

**Lớp : D13CNPM2**

**Khóa : 2018-2023**

***Hà Nội, 01 tháng 11 năm 2021***

**PHIẾU CHẤM ĐIỂM**

Sinh viên thực hiện:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ và tên** | **Nội dung thực hiện** | **Điểm** | **Chữ ký** |
| 1 | Cao Anh Đức |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |

Giảng viên chấm điểm:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **Chữ ký** | **Ghi chú** |
| **1** | Giảng viên 01:  ……………………... |  |  |
| **2** | Giảng viên 02:  ……………………... |  |  |
| **3** | Giảng viên 03:  ……………………... |  |  |

**MỤC LỤC**

[**LỜI CẢM ƠN** 4](#_Toc87593051)

[**MỞ ĐẦU** 5](#_Toc87593052)

[**CHƯƠNG 1. GIỚI THIỆU CHUNG VỀ HỆ CHUYÊN GIA VÀ EXPERTA** 6](#_Toc87593053)

[**1.1. Tổng quan về hệ chuyên gia** 6](#_Toc87593054)

[**1.1.1 Thiết kế một hệ chuyên gia** 6](#_Toc87593055)

[**1.1.2 Các vấn đề phù hợp để xây dựng một hệ chuyên gia** 7](#_Toc87593056)

[**1.1.3 Quy trình công nghệ tri thức** 8](#_Toc87593057)

[**1.1.4 Hệ chuyên gia dựa trên luật** 10](#_Toc87593058)

[**1.1.5 Hệ chuyên gia dựa trên mô hình** 11](#_Toc87593059)

[**1.2 Tổng quan về package Experta** 13](#_Toc87593060)

[**1.2.1. Giới thiệu** 13](#_Toc87593061)

[**1.2.2. Cài đặt** 13](#_Toc87593062)

[**1.2.3. Khái niệm cơ bản** 14](#_Toc87593063)

[**1.2.3.1. Facts** 14](#_Toc87593064)

[**1.2.3.2. Rules** 15](#_Toc87593065)

[**1.2.3.2. Facts vs Patterns** 16](#_Toc87593066)

[**1.2.3.3. DefFacts** 17](#_Toc87593067)

[**1.2.3.4. KnowledgeEngine.** 17](#_Toc87593068)

[**1.2.3.5. Handling facts.** 18](#_Toc87593069)

[**CHƯƠNG 2. HỆ CHUYÊN GIA CHẨN ĐOÁN BỆNH** 22](#_Toc87593070)

[**2.1. Tổng quan:** 22](#_Toc87593071)

[**2.2. Thu thập cơ sở tri thức:** 23](#_Toc87593072)

[**2.3. Biểu diễn tri thức** 23](#_Toc87593073)

[**2.4. Thiết kế cơ sở tri thức** 23](#_Toc87593074)

[**CHƯƠNG 3. THỰC NGHIỆM** 26](#_Toc87593075)

[**KẾT LUẬN** 27](#_Toc87593076)

[**TÀI LIỆU THAM KHẢO** 28](#_Toc87593077)

# **LỜI CẢM ƠN**

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, sự giúp đỡ dù ít hay nhiều, dù là trực tiếp hay gián tiếp của người khác. Trong suốt thời gian từ khi bắt đầu học tập, tôi đã nhận được rất nhiều sự quan tâm, giúp đỡ của thầy cô, gia đình và bạn bè.

Tôi xin chân thành cám ơn thầy cô đã tận tâm hướng dẫn chúng tôi qua từng buổi học trên lớp cũng như những buổi nói chuyện, thảo luận về môn học. Trong thời gian được học tập và thực hành dưới sự hướng dẫn của thầy cô, tôi không những thu được rất nhiều kiến thức bổ ích, mà còn được truyền sự say mê và thích thú đối với bộ môn Hệ chuyên gia. Nếu không có những lời hướng dẫn, dạy bảo của thầy cô thì tôi nghĩ tiểu luận này của chúng tôi rất khó có thể hoàn thành được.

Xin chân thành gửi lời cảm ơn sâu sắc tới các thầy, các cô , đặc biệt là thầy Vũ Văn Định đã tận tình giảng dạy và truyền đạt cho tôi những kiến thức, kinh nghiệm quý báu trong suốt thời gian học tập và rèn luyện.

Sau cùng, tôi xin kính chúc các thầy cô trong Khoa Công nghệ Thông tin dồi dào sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh của mình là truyền đạt kiến thức cho thế hệ mai sau.

Chúng tôi xin chân thành cảm ơn.

# **MỞ ĐẦU**

**Lý do chọn đề tài**

Ngày nay công nghệ thông tin (CNTT) trở thành một lĩnh vực mũi nhọn trong công cuộc phát triển kinh tế xã hội, CNTT vừa là công cụ, vừa là động lực thúc đẩy quá trình công nghiệp hóa hiện đại hóa đất nước. Việc ứng dụng CNTT vào thực tiễn đã có một bước phát triển rất mạnh mẽ. CNTT được ứng dụng rộng rãi, đóng vai trò to lớn trong quá trình xử lý của nhiều lĩnh vực và nhiều ngành: y tế, giáo dục, tài chính, ...

Với sự phát triển của khoa học máy tính ngày nay, ngoài việc lưu trữ các thông tin, người ta còn muốn có một hệ xử lý thông tin có khả năng suy luận để rút ra những kết luận từ các dữ liệu, các sự kiện có sẵn. Hệ thống này được gọi là “hệ chuyên gia”. Ở Việt Nam, hệ chuyên gia còn khá mới mẻ và được ứng dụng rất ít.

Nắm bắt được điều đó tôi đã tiến hành thực hiện tiểu luận “***Xây dựng hệ chuyên gia y tế*”**. Mục đích của đề tài được thực hiện với mục tiêu xây dựng một hệ chuyên gia hỗ trợ chẩn đoán bệnh và chứa đựng một số lượng tương đối các kiến thức cần thiết cho việc chẩn đoán bệnh tật.

Mặc dù đã rất cố gắng nhưng trong quá trình hoàn thành bài tập lớn chắc chắn sẽ còn những thiếu sót rất mong có được sự đóng góp của thầy cô và các bạn để giúp chúng tôi có thể hoàn thiện hơn trong tương lai.

**Cấu trúc của tiểu luận:**

*Chương 1. Giới thiệu chung về hệ chuyên gia và Experta (Python package)*

*Chương 2: Hệ chuyên gia chẩn đoán bệnh*

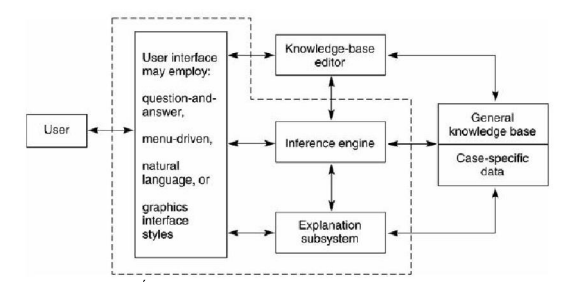
*Chương 3: Thực nghiệm*

# **CHƯƠNG 1. GIỚI THIỆU CHUNG VỀ HỆ CHUYÊN GIA VÀ EXPERTA**

# **1.1. Tổng quan về hệ chuyên gia**

# **1.1.1 Thiết kế một hệ chuyên gia**

Hình dưới đây cho thấy các khối quan trọng nhất tạo nên một hệ chuyên gia. Người dùng tương tác với hệ chuyên gia thông qua giao diện người sử dụng (user interface), giao diện này đơn giản hoá việc giao tiếp và che giấu phần lớn sự phức tạp của hệ thống. Các hệ chuyên gia sử dụng một số lượng phong phú các kiểu giao diện, bao gồm hỏi và trả lời, điều khiển bởi đơn trình, ngôn ngữ tự nhiên, hay đồ họa,...



*Hình 1.1: Kiến trúc một hệ chuyên gia tiêu biểu*

Trái tim của hệ chuyên gia là cơ sở tri thức tổng quát chứa tri thức giải quyết vấn đề của một ứng dụng cụ thể. Cơ sở tri thức bao gồm tri thức tổng quát (cũng như thông tin của một tình huống cụ thể).

Động cơ suy luận áp dụng tri thức cho việc giải quyết các bài toán thực tế, về căn bản nó là một trình thông dịch cho cơ sở tri thức.

Trong hệ sinh, động cơ suy diễn thực hiện chu trình điều khiển nhận dạng - hành động. Việc tách biệt cơ sở tri thức ra khỏi động cơ suy diễn là rất quan trọng vì rất nhiều lý do:

1. Sự tách biệt của tri thức dùng để giải quyết vấn đề và động cơ suy diễn sẽ tạo điều kiện cho việc biểu diễn tri thức theo một cách tự nhiên hơn.
2. Bởi vì cơ sở tri thức được cách ly khỏi các cấu trúc điều khiển cấp thấp của chương trình, các nhà xây dựng hệ chuyên gia có thể tập trung một cách trực tiếp vào việc nắm bắt và và tổ chức giải quyết vấn đề hơn là phải thực hiện trên các chi tiết của việc cài đặt vào máy tính.
3. Sự phân chia tri thức và điều khiển cho phép thay đổi một phần cơ sở tri thức mà không tạo ra các hiệu ứng lề trên các phần khác nhau của chương trình.
4. Sự tách biệt này cũng cho phép một phần mềm điều khiển và giao tiếp có thể sử dụng cho nhiều hệ thống khác nhau.
5. Sự module hóa này cho phép chúng ta thử nghiệm nhiều chế độ điều khiển khác nhau trên cùng một cơ sở luật.

Hệ con giải thích cho phép chương trình giải thích quá trình suy luận của nó cho người dùng .Các câu trả lời này bao gồm các 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).

Trình soạn thảo cơ sở tri thức: giúp các nhà lập trình xác định và hiệu chỉnh lỗi trong quá trình làm việc của hệ thống, thường là bằng cách truy xuất những thông tin cung cấp bởi hệ con giải thích.

# **1.1.2 Các vấn đề phù hợp để xây dựng một hệ chuyên gia**

Các hệ chuyên gia luôn đòi hỏi sự đầu tư rất lớn về tiền bạc và sức lực con người. Những cố gắng để giải quyết một bài toán quá phức tạp, quá ít hiểu biết, hoặc có những yếu tố không phù hợp với công nghệ hiện đại có thể dẫn đến thất bại, hao tốn tiền của. Các nhà nghiên cứu đã xây dựng một tập hợp các chỉ dẫn có tính không hình thức cho việc xác định khi nào một bài toán thích hợp giải quyết bằng hệ chuyên gia:

1. Cần thiết phải có một giải pháp biện minh cho chi phí và sức lực cho việc xây dựng hệ chuyên gia vì nếu không nó sẽ là một sự lãng phí:
2. Hiểu biết chuyên môn của con người không có sẵn ở mọi nơi cần đến nó (một hệ chuyên gia chữa bệnh sẽ giúp cho một bác sĩ bình thường có được một sự chẩn đoán và điều trị ở mức độ chuyên gia);
3. Vấn đề có thể được giải quyết bằng cách sử dụng các kỹ thuật suy luận ký hiệu:
4. Phạm vi vấn đề được cấu trúc tốt và không đòi hỏi sự suy luận theo lẽ thường tình:
5. Vấn đề có thể không giải quyết được bằng cách sử dụng các phương pháp tính toán truyền thống:
6. Có sự hợp tác và hiểu ý giữa các chuyên gia (kinh nghiệm của các chuyên gia được tích lũy trong quá trình làm việc cực nhọc nên họ có thể không hợp tác là điều có thể vì như vậy họ lo hệ chuyên gia sẽ thay thế họ -> phải có cách xử lý phù hợp):
7. Vấn đề cần giải quyết phải có kích thước và quy mô đứng mức, vấn đề không được vượt quá trình độ của công nghệ hiện đại:

# **1.1.3 Quy trình công nghệ tri thức**

Những người chủ yếu trong việc xây dựng hệ chuyên gia là kỹ sư tri thức và chuyên gia và những người sử dụng cuối.

Kỹ sư tri thức là chuyên gia về ngôn ngữ và biểu diễn trong công nghệ thông tin, với nhiệm vụ chính là chọn các công cụ phần mềm và phần cứng cho đề án, giúp đỡ các chuyên gia phát biểu các tri thức cần thiết và cài đặt tri thức đó vào một cơ sở tri thức đúng đắn và hiệu quả. Thường ban đầu kĩ sư thường không hiểu gì về lĩnh vực ứng dụng.

Chuyên gia cung cấp tri thức về lĩnh vực ứng dụng, đây là người từng công tác trong lĩnh vực ứng dụng và hiểu biết những vấn đề kĩ thuật của nó: chẳng hạn như cách làm tắt, cách sửa lỗi, các đánh giá giải pháp cục bộ và nhiều kĩ năng khác chứng tỏ anh ta là một chuyên gia.

Trong phần lớn các ứng dụng thì người dùng cuối quyết định những ràng buộc thiết kế chính. Những kỹ năng và nhu cầu của người cần dùng cần phải xem xét trong suốt quá trình thiết kế, chương trình có làm cho công việc của người dùng dễ dàng hơn không, nhanh hơn, thuận tiện hơn không?

Giống như hầu hết các bài toán lập trình trong công nghệ thông tin, việc xây dựng một hệ chuyên gia đòi hỏi một chu trình phát triển theo kiểu không truyền thống dựa trên một bản mẫu được tạo ra ban đầu và việc xem xét lại mã lệnh một cách tăng dần, phương pháp này được gọi là lập trình thăm dò.

Nói chung quá trình xây dựng hệ thống thường bắt đầu với việc kĩ sư tri thức cố gắng làm quen với phạm vi xác định vấn đề, điều này giúp ích cho việc giao tiếp với chuyên gia dễ dàng hơn. Nó thường được thực hiện bằng những bài phỏng vấn mở đầu với chuyên gia, bằng quan sát chuyên gia trong quá trình họ làm việc, hoặc thông qua việc đọc những tài liệu liên quan đến lĩnh vực đó. Tiếp theo kĩ sư và chuyên gia bắt đầu khai thác những tri thức giải quyết vấn đề của chuyên gia này bằng cách đưa ra các câu hỏi, các ví dụ, các trường hợp,...

Ngay sau khi kỹ sư có cái nhìn tổng qua về lĩnh vực vấn đề và đã cùng chuyên gia giải quyết một số bài toán, anh ta bắt đầu vào thiết kế hệ thống: chọn phương pháp biểu diễn tri thức, như luật hay frame, xác định các chiến lược tìm kiếm, …

Sau cùng kỹ sư thiết kế một phiên bản dùng thử và cùng chuyên gia kiểm tra hiệu quả, đồng thời với việc sửa chữa, cập nhật.

Đặc trưng thứ hai của hệ chuyên gia là cần xem chương trình như không bao giờ có kết thúc. Một cơ sở may rủi lớn sẽ luôn luôn có những hạn chế của nó, vì tri thức luôn đổi mới vì vậy luôn luôn phải cập nhật thông tin.

Có hai loại hệ chuyên gia được sử dụng:

1. Hệ chuyên gia dựa trên luật.
2. Hệ chuyên gia dựa trên mô hình.

# **1.1.4 Hệ chuyên gia dựa trên luật**

Các hệ chuyên gia dựa trên luật biểu diễn tri thức dưới dạng if... then. Cách tiếp cận này thích hợp với mô hình cơ bản và là một trong những kỹ thuật cổ điển và được sử dụng rộng rãi nhất dùng cho biểu diễn tri thức về một lĩnh vực trong hệ chuyên gia.

Với hệ chuyên gia dựa trên luật, dữ liệu cho trường hợp cụ thể được giữ trong bộ nhớ làm việc; động cơ suy diễn thực hiện chu trình nhận dạng - hành động của hệ sinh; cơ chế điều khiển này có thể hướng từ dữ liệu hay hướng từ mục tiêu.

Tuy nhiên với một hệ chuyên gia thì tiếp cận hướng từ mục tiêu sẽ tạo điều kiện cho quá trình giải thích hơn: vì từ một hệ hướng mục tiêu, việc suy luận theo đuổi một mục tiêu nào đó, mục tiêu đó bị chia thành nhiều mục tiêu con và cứ như vậy, kết quả là việc tìm kiếm luôn luôn được hướng dẫn thông qua sự phân cấp mục tiêu và mục tiêu con này. Để có ví dụ cụ thể về vấn đề giải quyết theo hướng mục tiêu, ta xét một hệ chuyên gia nhỏ dùng để chẩn đoán trục trặc của xe hơi:

Luật 1: IF (động cơ nhận được xăng AND động cơ khởi động được)

THEN (trục trặc là do bugi)

Luật 2: IF (động cơ không khởi động được AND đèn không sáng)

THEN (trục trặc do acquy hoặc dây cáp)

Luật 3: IF (động cơ không khởi động được AND đèn sang)

THEN (trục trặc là do mô tơ khởi động)

Luật 4: IF (còn xăng trong bình chứa nhiên liệu AND còn xăng trong bộ chế hoà khí)

THEN (động cơ nhận được xăng)

Kết luận về hệ chuyên gia dựa trên Luật:

Ưu điểm:

* + - Khả năng sử dụng trực tiếp các tri thức thực nghiệm của các chuyên gia.
    - Tính module của luật làm cho việc xây dựng và bảo trì luật dễ dàng.
    - Có thể thực hiện tốt trong các lĩnh vực hạn hẹp.
    - Có tiện ích giải thích tốt.
    - Các luật ánh xạ một cách tự nhiên vào không gian tìm kiếm trạng thái.
    - Dễ dàng theo dõi một chuỗi các luật và sửa lỗi.
    - Sự tách biệt giữa tri thức và điều khiển giúp đơn giản hoá quá trình phát triển của hệ chuyên gia.

Nhược điểm:

* + - Các luật đạt được từ chuyên gia mang tính heuristic rất cao. Ví dụ: trong y học luật “If sốt cao Then bị nhiễm trùng” mà không thể hiện lí thuyết sau hơn trong y học có quan sát (là cơ chế cơ thể phản ứng để chống lại vi khuẩn).
    - Các luật heuristic “dễ vỡ”, không xử lí được các trường hợp ngoài dự kiến; phải cần một chuyên gia có quan sát kỹ lưỡng mới phát hiện ra, nếu không đúng với dữ liệu thì hệ thống không giải quyết được.
    - Có khả năng giải thích chứ không chứng minh được.
    - Các tri thức thường rất phụ thuộc vào công việc, không thể sử dụng cho công việc khác.
    - Khó bảo trì các cơ sở luật lớn.

# **1.1.5 Hệ chuyên gia dựa trên mô hình**

Dựa vào lỗi của hệ chuyên gia dựa trên luật, thì hệ chuyên gia dựa trên mô hình được đưa ra.

Hệ chuyên gia dựa trên mô hình là một hệ thống mà sự phân tích căn cứ dựa vào mô tả chi tiết và chức năng của một hệ thống vật lí. Trong thiết kế và sử dụng, hệ chuyên gia dựa trên mô hình tạo ra một sự mô phỏng bằng phần mềm đối với chức năng của cái mà chung ta muốn tìm hiểu hay sửa chữa.

Một hệ thống chẩn đoán dựa trên mô hình đòi hỏi:

* Mô tả cho mỗi bộ phận trong thiết bị. Từ những mô tả này mà hệ chẩn đoán có thể mô phỏng hành vi của từng thiết bị.
* Một mô tả cấu trúc bên trong của thiết bị. Đây thường là một biểu diễn của các thành phần và mối quan hệ qua lại giữa chung. Những thông tin này sẽ giúp cho hệ thống mô phỏng sự tương tác giữa các thành phần của thiết bị.
* Việc chẩn đoán một lỗi cụ thể đòi hỏi sự quan sát việc thực hiện thật sự của thiết bị, thông thường là việc đo các thông số vào/ra của nó.

Vì vậy, nhiệm vụ của hệ sẽ xác định bộ phận nào có lỗi dựa trên các hành vi quan sát được. Điều này đòi hỏi phải có thêm các luật mô tả các chế độ có lỗi đã biết cho các bộ phận khác nhau và sự kết nối giữa chung. Hệ suy luận khi đó cần tìm ra những lỗi có khả năng nhất có thể giải thích hành vi quan sát được của hệ thống.

Kết luận về hệ suy luận dựa trên mô hình:

Một số ưu điểm của hệ này như sau:

* Tạo khả năng sử dụng tri thức về cấu trúc và chức năng của lĩnh vực trong giải quyết vấn đề.
* Vượt qua hạn chế của hệ chuyên gia dựa trên luật, hệ chuyên gia này có khuynh hướng mạnh, “khó vỡ”.
* Một số tri thức có thể chuyển tải cho công việc khác.
* Có khả năng cung cấp các lời giải thích rõ ràng cho các nguyên nhân.

Một số hạn chế của hệ:

* Mô hình chỉ mang tính trừu tượng, không thể chi tiết và khái quát hoá được chính xác.
* Hạn chế về thế giới đóng - tức là những gì không nằm trong mô hình coi như không tồn tại.
* Khi mô hình không chính xác hoặc không phù hợp thì coi như không có cách giải quyết hợp lí.
* Hệ thống tạo ra có thể lớn và chậm, độ phức tạp cao có nhiều tình huống ngoại lệ.

# **1.2 Tổng quan về package Experta**

# **1.2.1. Giới thiệu**

Experta là một giải pháp thay thế Python cho CLIP, càng tương thích càng tốt. Với mục tiêu giúp lập trình viên CLIPS dễ dàng chuyển tất cả kiến thức của mình sang nền tảng này.

Tính năng:

* Tương thích python 3.
* Triển khai Pure Python.
* Matcher dựa trên thuật toán RETE.

Sự khác biệt giữa CLIP và EXPERTA

1. CLIPS là một ngôn ngữ lập trình, Experta là một thư viện Python. Điều này áp đặt một số hạn chế đối với các công trình chúng ta có thể làm (đặc biệt là về LHS của một quy tắc).
2. CLIP được viết bằng chữ C, Experta trong Python. Một tác động đáng chú ý trong hiệu suất là được mong đợi.
3. Trong CLIP bạn thêm các sự kiện bằng cách sử dụng *assert,*trong Python *khẳng định* là một từ khóa, vì vậy chúng tôi sử dụng *khai báo* thay thế.

# **1.2.2. Cài đặt**

Từ PyPI:

Để cài đặt Experta, hãy chạy lệnh này trong terminal của bạn:

$ pip install experta

Lấy mã nguồn:

Experta được phát triển [trên Github.](https://github.com/nilp0inter/experta)

Bạn có thể nhân bản kho lưu trữ bằng lệnh git:

$ git clone https://github.com/nilp0inter/experta.git

Hoặc bạn có thể [tải xuống các bản phát hành](https://github.com/nilp0inter/experta/releases) ở định dạng .zip hoặc .tar.gz.

Khi bạn có một bản sao của nguồn, bạn có thể cài đặt nó chạy lệnh này:

$ python setup.py install

# **1.2.3. Khái niệm cơ bản**

# **1.2.3.1. Facts**

Là đơn vị cơ bản của thông tin Experta. Chúng được hệ thống sử dụng để lập luận về vấn đề.

Hãy cùng liệt kê một số thông tin về Facts, vì vậy… sự kiện meta

1. Lớp Fact là một lớp con của dict.

**** 2. Do đó, một Fact không duy trì thứ tự nội bộ của các mục.

**** 3. Ngược lại với dict, bạn có thể tạo Fact mà không có khóa (chỉ các giá trị) và Fact sẽ tạo một chỉ mục số cho các giá trị của bạn.

**** 4. Bạn có thể kết hợp các giá trị số tự động với khóa-giá trị, nhưng số tự động phải được khai báo trước:

****

5. Bạn có thể phân lớp Fact để thể hiện các loại dữ liệu khác nhau hoặc mở rộng nó với chức năng tùy chỉnh của bạn:

****

Ảnh có chứa văn bản

Mô tả được tạo tự động

Các trường Fact có thể được xác thực tự động cho bạn nếu bạn xác định chúng bằng cách sử dụng Trường. Trường sử dụng thư viện lược đồ nội bộ để xác thực dữ liệu. Ngoài ra, một trường có thể được khai báo là bắt buộc hoặc có giá trị mặc định.

# **1.2.3.2. Rules**

Trong Experta, Rules là một quy tắc có thể gọi được

Rules có hai thành phần, LHS (bên trái) và RHS (bên phải).

LHS mô tả (sử dụng các mẫu) các điều kiện mà quy tắc \* phải được thực thi (hoặc kích hoạt).

RHS là tập hợp các hành động để thực hiện khi quy tắc được kích hoạt.

Để Fact khớp với một Mẫu, tất cả các hạn chế của mẫu phải là True khi Fact được đánh giá dựa trên nó.

Ảnh có chứa văn bản

Mô tả được tạo tự động

Ảnh có chứa văn bản

Mô tả được tạo tự động

Để Rule trở nên hữu ích, nó phải là một phương thức của lớp con KnowledgeEngine.

Note: Để biết danh sách các toán tử phức tạp hơn, bạn có thể kiểm tra module.experta.operator

# **1.2.3.2. Facts vs Patterns**

Sự khác biệt giữa Facts và Patterns là nhỏ. Trên thực tế,

Patterns chỉ là Facts chứa "Phần tử có điều kiện của mẫu" thay vì dữ liệu thông thường. Chúng chỉ được sử dụng trong LHS của một quy tắc.

Nếu bạn không cung cấp nội dung của một mẫu dưới dạng PCE, thì Experta sẽ tự động đưa giá trị vào LiteralPCE cho bạn.

Ngoài ra, bạn không thể khai báo bất kỳ Dữ kiện nào có chứa PCE, nếu bạn khai báo, bạn sẽ nhận lại được một ngoại lệ tuyệt vời**.**

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

# **1.2.3.3. DefFacts**

Hầu hết các hệ thống chuyên gia về thời gian cần một tập hợp các dữ kiện có mặt để hệ thống hoạt động. Đây là mục đích của trình trang trí DefFacts.

****

Note: Phương pháp trang trí PHẢI là máy phát điện.

# **1.2.3.4. KnowledgeEngine.**

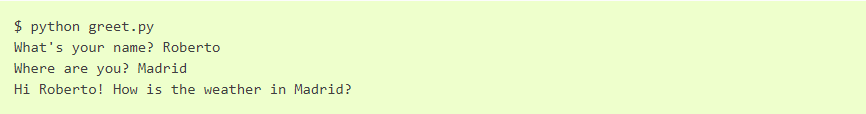
Đây là nơi mà tất cả những điều kỳ diệu xảy ra.

Bước đầu tiên là tạo một lớp con của nó và sử dụng Rule để trang trí các phương thức của nó.

Sau đó, bạn có thể khởi tạo nó, điền vào các dữ kiện và cuối cùng là chạy nó.

Ảnh có chứa văn bản

Mô tả được tạo tự động

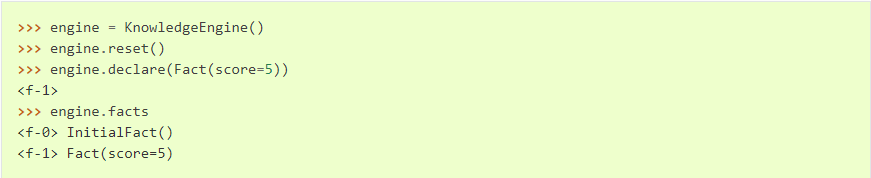


# **1.2.3.5. Handling facts.**

Các phương pháp sau đây được sử dụng để thao tác với tập hợp các dữ kiện mà động cơ biết.

#### Declare:

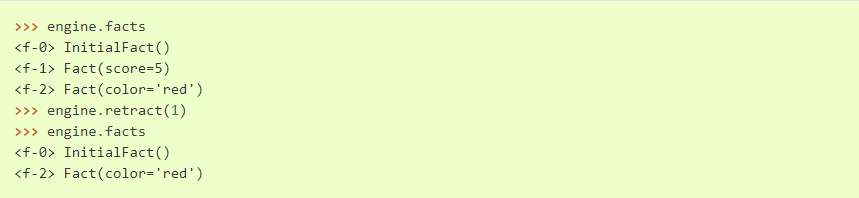
Thêm một sự kiện mới vào danh sách sự kiện (danh sách các sự kiện mà động cơ đã biết).

****

Note: Không thể khai báo cùng một dữ kiện hai lần trừ khi fact.duplication được đặt thành True.

Retract:

Loại bỏ một sự kiện hiện có khỏi danh sách sự kiện.



*Both, the index and the fact can be used with retract*

Modify: Rút một số dữ kiện khỏi danh sách dữ kiện và khai báo một dữ kiện mới với một số thay đổi. Các thay đổi được chuyển dưới dạng đối số.

Ảnh có chứa văn bản

Mô tả được tạo tự động

Duplicate: Thêm một dữ kiện mới vào danh sách dữ kiện bằng cách sử dụng một dữ kiện hiện có làm mẫu và thêm một số sửa đổi.

Ảnh có chứa văn bản

Mô tả được tạo tự động

Quy trình thực hiện động cơ

Đây là quy trình thông thường để thực thi một KnowledgeEngine.

Tất nhiên, lớp phải được khởi tạo.

Phương thức đặt lại phải được gọi là:

Điều này khai báo sự kiện đặc biệt InitialFact. Cần thiết để một số quy tắc hoạt động bình thường.

Khai báo tất cả các dữ kiện thu được bằng các phương thức được trang trí bằng @DefFacts.

Phương thức chạy phải được gọi. Điều này bắt đầu chu kỳ thực hiện.

Chu kỳ thực hiện:

Trong phong cách lập trình thông thường, điểm bắt đầu, điểm dừng và trình tự hoạt động được lập trình viên xác định rõ ràng. Với Experta, luồng chương trình không cần phải được định nghĩa một cách rõ ràng như vậy. Kiến thức (Quy tắc) và dữ liệu (Sự kiện) được tách biệt, và KnowledgeEngine được sử dụng để áp dụng kiến thức vào dữ liệu.

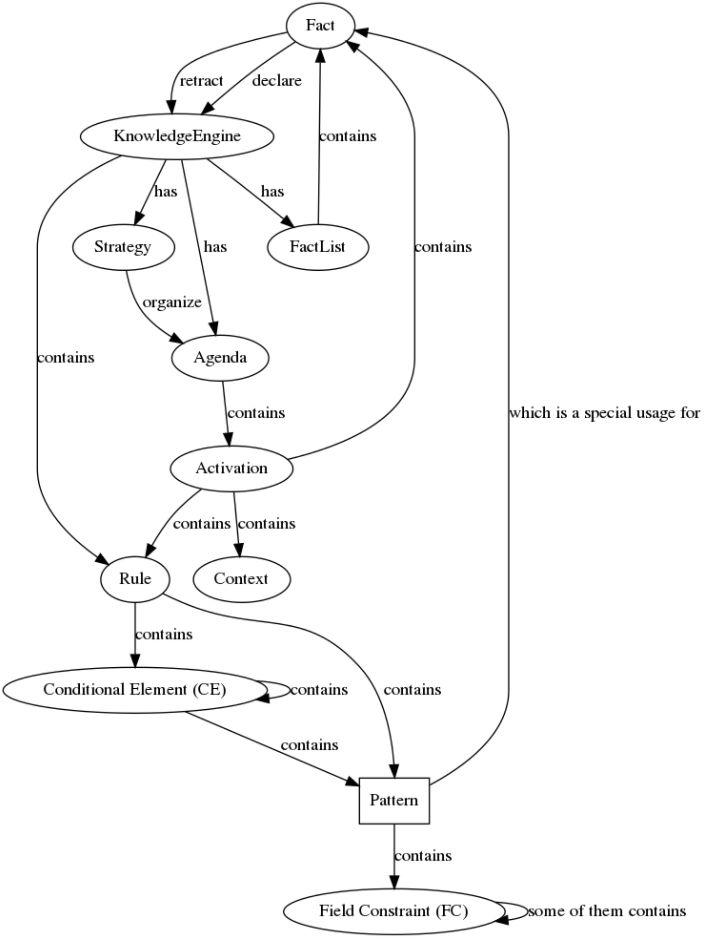
Difference between DefFacts and declare (Chu kỳ thực hiện cơ bản như sau:)

Nếu đã đạt đến giới hạn kích hoạt quy tắc, việc thực thi sẽ bị tạm dừng.

Quy tắc hàng đầu trong chương trình làm việc được chọn để thực hiện. Nếu không có quy tắc nào trong chương trình làm việc, việc thực thi sẽ bị tạm dừng.

Các hành động RHS của quy tắc đã chọn được thực thi (phương thức được gọi). Do đó, các quy tắc có thể được kích hoạt hoặc hủy kích hoạt. Các quy tắc đã kích hoạt (những quy tắc có các điều kiện hiện được đáp ứng) được đưa vào chương trình làm việc. Vị trí trong chương trình nghị sự được xác định bởi tính phù hợp của quy tắc và chiến lược giải quyết xung đột hiện tại. Các quy tắc đã hủy kích hoạt sẽ bị xóa khỏi chương trình làm việc.

Tham khảo:



# **CHƯƠNG 2. HỆ CHUYÊN GIA CHẨN ĐOÁN BỆNH**

# **2.1. Tổng quan:**

Các hệ thống chuyên gia điển hình sử dụng kiến thức để suy luận về dữ liệu đầu vào và tạo ra kết quả có ý nghĩa. Kiến thức này chủ yếu bao gồm các quy tắc đơn giản nếu sau đó, như nếu giá trị cảm biến nhiệt độ được > 100C thì tắt ấm đun nước điện.

Cơ sở kiến thức và đồ thị vẫn đang đóng một vai trò rất lớn trong nhiều hệ thống thông minh. Nhiều khu vực nông thôn ở Việt Nam có khả năng tiếp cận rất hạn chế với tư vấn y tế. Mọi người đi du lịch đường dài đến các phòng khám, hoặc các cơ sở y tế và thiếu các chuyên gia y tế ở hầu hết các cơ sở này. Điều này dẫn đến dịch vụ chậm, và bệnh nhân cuối cùng phải chờ đợi nhiều giờ mà không nhận được bất kỳ sự chú ý nào. Do đó, các hệ thống chuyên gia y tế có thể đóng một vai trò quan trọng trong những trường hợp như vậy mà các chuyên gia y tế không có sẵn. Một hệ thống chuyên gia chẩn đoán có thể giúp rất nhiều trong việc xác định các bệnh đó và mô tả các phương pháp điều trị được thực hiện.

Tính năng:

Experta(pyknow trước đó),một thư viện Python được sử dụng làm lõi.

Động cơ yêu cầu người dùng cho 10 triệu chứng khác nhau.

Người dùng gõ có hoặc không cho từng triệu chứng.

Động cơ cố gắng xác định bệnh dựa trên phản ứng của người dùng và các quy tắc được xác định trước.

Một khi bệnh được tìm thấy, một lời giải thích chi tiết về bệnh và phương pháp điều trị được hiển thị cho người dùng.

Nếu triệu chứng không phù hợp với một bệnh cụ thể, bệnh phù hợp với số lượng triệu chứng cao nhất được hiển thị cho người dùng.

Cơ sở kiến thức cho các bệnh có thể được chứng minh là một biểu đồ lưỡng đảng, với các bệnh ở một bên và các triệu chứng ở bên kia. Biểu đồ này có thể được hình dung bằng cách chạy tệp do\_thi\_luong\_phan.py.

Lưu ý: Dự án này chỉ dành cho mục đích trình diễn. Các triệu chứng và bệnh tật có thể không phù hợp. Đó là một nỗ lực để cho thấy các hệ thống cơ bản này có thể có được sức mạnh như thế nào. Vui lòng không sử dụng cho bất kỳ lời khuyên y tế.

# **2.2. Thu thập cơ sở tri thức:**

Sau quá trình tìm hiểu các nguồn tài liệu có liên quan đến y học,em đã thu thập được triệu chứng của một số bệnh thường gặp để triển khai chương trình

# **2.3. Biểu diễn tri thức**

Cơ sở tri thức là nơi lưu trữ, biểu diễn các tri thức mà hệ đảm nhận, làm cơ sở cho các hoạt động của hệ. Cơ sở tri thức bao gồm các sự kiện và các luật. Trong hệ chuyên gia chuẩn đoán bệnh thì các sự kiện chính là các triệu chứng của bệnh. Các luật được tạo ra dựa trên các sự kiện hay nói cách khác là khi thỏa mãn một số sự kiện nào đó thì sẽ tạo thành luật. Bài báo cáo lựa chọn cách biểu diễn luật ở dạng các mệnh đề IF … THEN.

# **2.4. Thiết kế cơ sở tri thức**

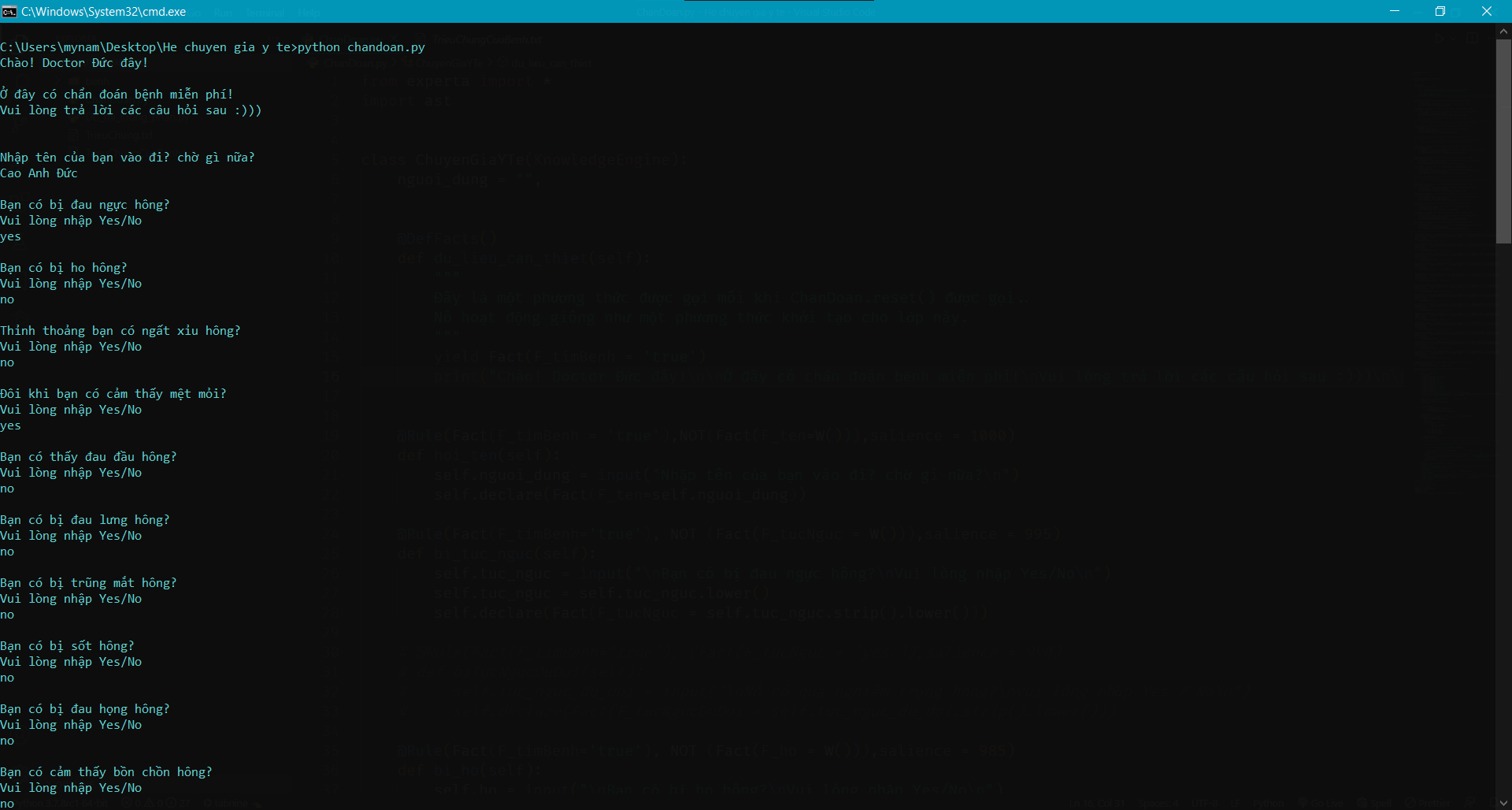
|  |  |  |
| --- | --- | --- |
| **Mã luật** | **Chẩn đoán bệnh** | |
| **Giả thiết** | **Kết luận** |
| R1 | IF Mệt mỏi  AND Tức ngực | THEN Alzheimers |
| R2 | IF Đau lưng  AND Bồn chồn | THEN Viêm khớp |
| R3 | IF Mệt mỏi  AND Trùng mắt | THEN Hen suyễn |
| R4 | IF Sốt  AND Ho | THEN Covid |
| R5 | IF Mệt mỏi  AND Bồn chồn | THEN Tiểu đường |
| R6 | IF Nhức đầu  AND Trùng mắt | THEN Động kinh |
| R7 | IF Sốt  AND Đau họng | THEN Tăng nhãn áp |
| R8 | IF Ngất xỉu | THEN Bệnh tim |
| R9 | IF Ngất xỉu  AND Sốt | THEN Say nắng |
| R10 | IF Trùng mắt  AND Bồn chồn | THEN Cường giáp |
| R11 | IF Tức ngực  AND Mệt mỏi  AND Đau họng | THEN Hạ thân nhiệt |
| R12 | IF Sốt  AND Ho  AND Nhức đầu | Then Vàng da |
| R13 | IF Trùng mắt  AND Sốt  AND Mệt mỏi  AND Bồn chồn | THEN Bệnh lao |

**Mô tả các sự kiện về triệu chứng các loại bệnh**

|  |  |
| --- | --- |
| **Mã luật** | **Sự kiện** |
| R1 | "Alzheimers": "met\_moi,tuc\_nguc" |
| R2 | "Viem\_khop": "dau\_lung,bon\_chon" |
| R3 | "Hen\_suyen":"met\_moi,trung\_mat" |
| R4 | "Covid":"sot,ho" |
| R5 | "Tieu\_duong":"met\_moi,bon\_chon" |
| R6 | "Dong\_kinh":"nhuc\_dau,trung\_mat" |
| R7 | "Tang\_nhan\_ap": "sot,dau\_hong" |
| R8 | "Benh\_tim": "ngat\_xiu" |
| R9 | "Say\_nang": "ngat\_xiu,sot" |
| R10 | "Cuong\_giap": "trung\_mat,bon\_chon" |
| R11 | "Ha\_than\_nhiet": "tuc\_nguc,met\_moi,dau\_hong" |
| R12 | "Vang\_da": "sot,ho,nhuc\_dau" |
| R13 | "Viem\_xoang": "nhuc\_dau,sot,dau\_hong" |
| R14 | "Benh\_lao": "trung\_mat,sot,met\_moi,bon\_chon" |

# **CHƯƠNG 3. THỰC NGHIỆM**

**Giao diện và kết quả**:  
sau đây là một số màn hình chính khi chạy chương trình Python chẩn đoán bệnh tật:



Ảnh có chứa văn bản

Mô tả được tạo tự động

# **KẾT LUẬN**

Trong đề tài này em đã giới thiệu một cách tương đối đầy đủ về cơ sở dữ liệu suy diễn, về khả năng biểu diễn cơ sở dữ liệu thông qua ngôn ngữ logic; đồng thời giới thiệu một ngôn ngữ lập trình tương thích để biểu diễn cơ sở tri thức bằng logic mô tả, đó là ngôn ngữ Python (package Experta). Đề tài đã trình bày khả năng ứng dụng logic trong biểu diễn mô hình cơ sở dữ liệu suy diễn. Đề tài cũng thảo luận về khả năng thực hiện truy vấn thông qua cách thức biểu diễn của ngôn ngữ Python, biến đổi các luật của Phthon thành các mô tả khái niệm trong logic mô tả.

Với khả năng biểu diễn, truy vấn bằng logic mô tả ta có thể suy rộng ra rằng, cơ sở dữ liệu có thể được biến đổi thành cơ sở tri thức, với ngữ nghĩa truy vấn phong phú hơn. Hệ chuyên gia ở trên đã có thể đặt những câu hỏi cho bệnh nhân và đưa ra lời khuyên hữu ích.

Tuy nhiên, ngoài những công việc đã làm được, trong đề tài này mới mang tính lý thuyết. Chúng em chỉ mới xây dựng được chương trình trình diễn đơn giản, nên chắc chắn chưa hoàn toàn thuyết phục. Vì vậy, kế hoạch công việc trong tương lai để hoàn thành hơn sẽ là đi sâu, ứng dụng một trong những ngôn ngữ suy diễn để xây dựng được một chương trình ứng dụng thực tế với quy mô lớn hơn, hoàn thiện hơn.

Ở nước ta cũng như trên thế giới việc nghiên cứu về cơ sở dữ liệu suy diễn rất ít, do vậy nguồn tài liệu nghiên cứu về đề tài cũng rất hạn chế. Do đó đề tài chắc chắn còn nhiều những hạn chế, chúng em rất mong các thầy, các cô và các bạn đồng nghiệp đóng góp ý kiến để đề tài được hoàn thiện hơn.

**TÀI LIỆU THAM KHẢO**

[1] PGS.TS. Phan Huy Khánh, “*Lập trình logic trong Prolog*”, NXB Đại học Quốc gia Hà Nội, 2004.

[2] PGS.TS. Phan Huy Khánh, “*Giáo trình hệ chuyện gia*” – Đại học Đà Nẵng, 2010.

[3] TS. Nguyễn Thiện Thành , “*Trí tuệ nhân tạo và hệ chuyên gia*” – Đại học Quốc Gia.

[4] Phạm Thọ Hoàn, Phạm Thị Anh Lê, “*Trí tuệ nhân tạo*” – Đại học Sư phạm Hà Nội.

[5] Đỗ Trung Tuấn, “*Hệ chuyên gia*”, NXB. Giáo dục, 199