การออกแบบและสร้างระบบผู้เชี่ยวชาญแบบเฟรมโดยใช้ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์

FRAME-BASED OBJECT-RELATIONAL DATABASE EXPERT SYSTEM DESIGN AND IMPLEMENTATION

ชุลีรัตน์ รัตนประทีป

CHULEERAT RATTANAPRATEEP

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2550

การออกแบบและสร้างระบบผู้เชี่ยวชาญแบบเฟรมโดยใช้ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์

FRAME-BASED OBJECT-RELATIONAL DATABASE EXPERT SYSTEM DESIGN AND IMPLEMENTATION

ชุลีรัตน์ รัตนประทีป

CHULEERAT RATTANAPRATEEP

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2550

FRAME-BASED OBJECT-RELATIONAL DATABASE EXPERT SYSTEM DESIGN AND IMPLEMENTATION

CHULEERAT RATTANAPRATEEP

A THESIS SUBMITTED IN PARTAIL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN COMPUTER ENGINEERING
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF THECHNOLOGY LADKRABANG
2007

COPYRIGHT 2007

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

หัวข้อวิทยานิพนธ์ การออกแบบและสร้างระบบผู้เชี่ยวชาญแบบเฟรมโดยใช้ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์

นักศึกษา นางสาว ชุลีรัตน์ รัตนประทีป

รหัสนักศึกษา 45061058

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา วิศวกรรมคอมพิวเตอร์

พ.ศ. 2550

อาจารย์ผู้ควบคุมวิทยานิพนธ์ รศ.ดร.ศุภมิตร จิตตะยโศธร

บทคัดย่อ

ในระบบผู้เชี่ยวชาญหรือเรียกอีกอย่างว่าระบบฐานความรู้นั้น ฐานความรู้จะเป็นส่วนจัดเก็บความรู้ซึ่งก็คือข้อเท็จจริงและกฎในรูปแบบที่สามารถนำข้อเท็จจริงและกฎเหล่านั้นมาใช้โดยกลไกการวินิจฉัยได้แต่ในระบบผู้เชี่ยวชาญขนาดใหญ่ฐานความรู้ส่วนมากจะใช้การแทนความรู้แบบเฟรมและถูกเรียกว่าระบบผู้เชี่ยวชาญแบบเฟรม หรือระบบฐานความรู้แบบเฟรม

ระบบผู้เชี่ยวชาญแบบเฟรมนั้นได้ถูกใช้อย่างกว้างขวางในการแทนความรู้สำหรับระบบผู้เชี่ยวชาญขนาดใหญ่ที่รองรับฐานความรู้จำนวนมาก ซึ่งระบบผู้เชี่ยวชาญขนาดใหญ่หลายๆระบบจะมีความสามารถที่จะเชื่อมต่อกับฐานข้อมูลภายนอกได้โดยที่ข้อเท็จจริงนั้นจะถูกเก็บอยู่ในฐานข้อมูลภายนอกนั้นและสามารถถูกดึงขึ้นมาเพื่อนำมาใช้ในฐานความรู้ของระบบผู้เชี่ยวชาญนั้นๆได้ การวินิจฉัยจะถูกกระทำโดยกลไกการวินิจฉัย (inference engine) ของระบบผู้เชี่ยวชาญซึ่งในหลายๆกรณีนั้น ข้อเท็จจริงที่อยู่ภายนอกนั้นจะถูกอ้างถึงหลายๆครั้ง ในการวินิจฉัยแต่ละครั้ง ดังนั้นจึงมีข้อมูลจำนวนมาก ถูกส่งไปมาในการติดต่อระหว่างระบบ

งานวิจัยนี้จึงได้นำเสนอระบบผู้เชี่ยวชาญแบบเฟรมโดยใช้ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ซึ่งมีการเชื่อมต่ออย่างแนบแน่นระหว่างระบบผู้เชี่ยวชาญและระบบฐานความรู้ภายนอกที่เก็บอยู่ในฐานข้อมูลซึ่งฐานความรู้ที่จัดเก็บในฐานข้อมูลนี้ใช้เฟรมในการแทนความรู้นอกจากนั้นระบบฐานข้อมูลนี้ยังมีกลไกการวินิจฉัยของมันเองด้วยดังนั้นการวินิจฉัยสามารถถูกกระทำบนฝั่งฐานความรู้ภายนอกที่จัดเก็บเฟรมในฐานข้อมูลได้เลย ผลลัพธ์ที่ได้จะไม่เป็นเพียงข้อเท็จจริงที่ถูกดึงขึ้นมาเพื่อให้ระบบผู้เชี่ยวชาญทำการวินิจฉัยต่อไปเท่านั้นแต่จะเป็นผลลัพธ์ที่สามารถเป็นคำตอบให้กับคำถามได้โดยใช้ข้อเท็จจริงที่อยู่บนฝั่งฐานข้อมูลได้เลย งานวิจัยนี้ได้พัฒนาระบบผู้เชี่ยวชาญสำหรับการวินิจฉัยโรคเป็นตัวอย่างในงานวิจัย โดยใช้ระบบการจัดการฐานข้อมูลออราเคิล 10g ซึ่งเป็นระบบการจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์

Thesis Title	Frame-based object-relational database expert system design and implementation
Student	Miss Chuleerat Rattanaprateep
Student ID.	45061058
Degree	Master of engineering
Program	Computer engineering
Year	2007
Thesis Advisor	Assoc. Prof. Dr. Suphamit Chittayasothorn

ABSTRACT

An expert database system is an expert system which is developed on databases using DBMS technology to manage facts and rules.

An expert system or known as knowledge-based system comprises of knowledge base and inference engine in which their knowledge base is used to represent expertise knowledge as data and rule. The knowledge can be called upon when needed to solve a problem by the inference engine. In large expert system the knowledge-based can be represented using frames called frame-based expert system or frame-based system.

Frame-based expert systems are widely used as the knowledge representation for such expert systems with large knowledge bases. Many systems have the ability to connect to external databases. Facts stored in databases can be loaded into expert system's knowledge base and inference is performed by the inference engine of the expert system. In many cases, such external facts are required several times for each inference. Thus, a lot of communication traffic takes place. This research work present a design and implementation of a frame-based object-relational database system which has a tight coupling between the expert system and the external knowledge base. The external knowledge base also use frame as its knowledge representation. Moreover, it has its own inference engine so that inference can be perform on the knowledge base side and the results, not only simple facts, are sent back to the expert system for further inference. A medical consultation system is used as an illustrated example an Oracle 10g object relational DBMS is used as the database platform.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดีด้วยความเมตตาในการให้คำปรึกษาและแนะนำตลอดทุกขั้นตอนการดำเนินการรวมทั้งความอนุเคราะห์ผลิตภัณฑ์ซอฟท์แวร์ที่ใช้ในการดำเนินการวิจัยจาก รศ.ดร.ศุภมิตร จิตตะยโสธร ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ ผู้วิจัยรู้สึกซาบซึ้งในความเมตตาของท่านมากและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์และคณาจารย์ทุกท่านในภาควิชาวิศวกรรมคอมพิวเตอร์คณะวิศวกรรมศาสตร์ที่ได้ให้ความกรุณาประสิทธิประสาทความรู้และตักเตือนแนะนำให้ผู้วิจัยสามารถดำเนินการวิจัยมาได้จนเสร็จสิ้นได้ด้วยดี

ขอขอบคุณแพทย์หญิงชวิดา รัตนประทีป แพทย์เฉพาะทางด้าน โสต ศอ นาสิก และลาริงซ์วิทยา และนายแพทย์เลิศรัช วงศ์ราช แพทย์เฉพาะทางด้านศัลยกรรมโรงพยาบาลศูนย์ขอนแก่นที่ให้ความอนุเคราะห์และอำนวยความสะดวกในการศึกษาหาข้อมูลด้านกระบวนการวินิจฉัยโรคและเอกสารต่างๆ

ขอขอบคุณ นางสาวปิยะอร คงศักดิ์ตระกูล และนางสาวเจนจิต จากบริษัทซอฟสแคว์ที่ให้ความช่วยเหลือด้านระบบการจัดการฐานข้อมูลและคำแนะนำในการพัฒนาระบบบนฐานข้อมูล

ขอขอบคุณ นายประกิต และนายศราวุธ คงยัง ผู้ช่วยเหลือและให้คำปรึกษาด้านการพัฒนาระบบ

ขอขอบคุณบัณฑิตวิทยาลัยและบัณฑิตศึกษาคณะวิศวกรรมศาสตร์ตลอดจนเจ้าหน้าที่ทุกคนที่ได้ช่วยอำนวยความสะดวกและประสานงานให้ผู้วิจัยด้วยดีตลอดมา

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจและให้การสนับสนุนในทุกเรื่องๆ ทำให้ข้าพเจ้าสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

คุณค่าและประโยชน์อันพึงมาจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

ชุลีรัตน์ รัตนประทีป

สารบัญ

หน้า

บทคัดย่อภาษาไทย..I

บทคัดย่อภาษาอังกฤษ..II

กิตติกรรมประกาศ..III

สารบัญ..IV

สารบัญตาราง..VI

สารบัญรูป..VII

บทที่ 1 บทนำ..1

1.1 ความเป็นมาและความสำคัญของปัญหา..1

1.2 วัตถุประสงค์..4

1.3 สมมุติฐานของการศึกษา..4

1.4 ขอบเขตการศึกษา..5

1.5 ขั้นตอนการศึกษา..5

1.6 โครงสร้างของวิทยานิพนธ์..5

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง..7

2.1 ระบบผู้เชี่ยวชาญ..7

2.2 การแทนความรู้..11

2.3 กลไกการวินิจฉัยบนระบบผู้เชี่ยวชาญ..16

2.4 หลักการการโปรแกรมเชิงวัตถุ..24

2.5 เปรียบเทียบหลักการเชิงวัตถุกับเฟรม..25

2.6 ระบบการจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์..27

2.7 ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ในเชิงพาณิชย์..32

2.8 การบูรณาการระบบฐานความรู้กับระบบฐานข้อมูล..33

2.9 การติดต่อสื่อสารระหว่างระบบฐานความรู้..36

บทที่ 3 สถาปัตยกรรมของระบบ..40

3.1 สถาปัตยกรรมของระบบ..40

3.2 การเชื่อมต่อระบบฐานความรู้ภายนอกกับระบบสารสนเทศอื่นๆ..43

สารบัญ (ต่อ)

หน้า

3.3 การติดต่อสื่อสารระหว่างไคลเอนท์กับเซิร์ฟเวอร์ด้วยภาษา KQML....................................45

บทที่ 4 รายละเอียดเกี่ยวกับการสร้างระบบ..47

4.1 เปรียบเทียบหลักการเชิงวัตถุสัมพันธ์กับเฟรม...47

4.2 การสร้างระบบฐานความรู้บนระบบฐานข้อมูลเชิงวัตถุสัมพันธ์.......................................49

4.3 กลไกการวินิจฉัยที่ฝั่งฐานความรู้ภายนอก..53

4.4 การดึงข้อมูลสารสนเทศจากฐานข้อมูลอื่นเข้ามายังระบบเฟรม.......................................58

4.5 โปรโตคอลที่ใช้ในการติดต่อสื่อสารระหว่างไคลเอนท์กับเซิร์ฟเวอร์................................60

บทที่ 5 การพัฒนาระบบผู้เชี่ยวชาญสำหรับการวินิจฉัยโรค...62

5.1 ขั้นตอนการทำงานของระบบ..62

5.2 การสร้างเฟรมใช้แทนความรู้..63

5.3 การวินิจฉัยหาคำตอบของระบบผู้เชี่ยวชาญที่สร้างขึ้น..69

5.4 การแปลงข้อมูลไปเป็น KQML message ในตัวอย่างการพัฒนาระบบผู้เชี่ยวชาญสำหรับงานการวินิจฉัยโรค..71

5.5 ผลการทดลอง...73

บทที่ 6 สรุปผลและข้อเสนอแนะ...77

6.1 สรุปผลการวิจัย..77

6.2 ข้อเสนอแนะ 78

เอกสารอ้างอิง..79

ภาคผนวก..81

ประวัติผู้เขียน..91

สารบัญตาราง

ตารางที่ หน้า

2.1 สรุปค่าพารามิเตอร์ที่เป็นคีย์เวิร์ดและความหมาย.. 38

5.1 แสดงเฟรมที่เกี่ยวกับโรคหูและโรคที่เกี่ยวข้องกับการปวดหู... 63

5.2 เฟรมคำถามเกี่ยวกับอาการที่เกี่ยวข้องกับการปวดหู.. 68

สารบัญรูป

รูปที่ หน้า

2.1 แนวคิดพื้นฐานของระบบผู้เชี่ยวชาญ........8
2.2 โครงสร้าง และความสัมพันธ์ระหว่างส่วนประกอบ ในระบบผู้เชี่ยวชาญ........9
2.3 ตัวอย่างโครงสร้างของเฟรม........13
2.4 ตัวอย่างโครงสร้าง และความสัมพันธ์ของเฟรมรถยนต์........15
2.5 สถานะของเนื้องาน สแต็กเป้าหมาย และสแต็กของกฎ I........17
2.6 สถานะของเนื้องาน สแต็กเป้าหมาย และสแต็กของกฎ II........17
2.7 สถานะของเนื้องาน สแต็กเป้าหมาย และสแต็กของกฎ III........18
2.8 สถานะของเนื้องาน สแต็กเป้าหมาย และสแต็กของกฎ IV........19
2.9 สถานะของเนื้องาน สแต็กเป้าหมาย และสแต็กของกฎ V........20
2.10 สถานะของเนื้องานก่อนที่จะเริ่มกระบวนการวินิจฉัย........21
2.11 สถานะของเฟรม A และ B ก่อนที่จะถูกกระตุ้นด้วยเดมอน........23
2.12 สถานะของเฟรม A และ B หลังจากถูกกระตุ้นด้วยเดมอน........23
2.13 เทคนิคแบบการสร้างการเชื่อมต่อ........33
2.14 เทคนิคแบบการสร้างส่วนเชื่อมประสานภายในฐานความรู้ไปยังฐานข้อมูล........34
2.15 เทคนิคแบบการสร้างฟังก์ชันงานฐานข้อมูลภายในฐานความรู้........34
2.16 เทคนิคแบบการสร้างสร้างส่วนเชื่อมประสานภายในฐานข้อมูลไปยังฐานความรู้........34
2.17 เทคนิคแบบการสร้างฟังก์ชันงานฐานความรู้ภายในฐานข้อมูล........35
2.18 KQML string syntax ในรูปแบบ BNF........37
3.1 แสดงสถาปัตยกรรมของระบบ........41
3.2 แสดงสถาปัตยกรรมของระบบฝั่งเซิร์ฟเวอร์........42
3.3 แสดงสถาปัตยกรรมของระบบฝั่งไคลเอนท์........43
3.4 แสดงการเชื่อมต่อระหว่างระบบกับฐานข้อมูลภายนอก........44
3.5 โปรโตคอลที่ใช้ในการติดต่อสื่อสารระหว่างไคลเอนท์กับเซิร์ฟเวอร์........45
4.1 แผนภาพโครงสร้างข้อมูลแบบวัตถุ........47
4.2 แผนภาพโครงสร้างข้อมูลแบบเฟรม........48
4.3 A Frame sub schema Using NIAM........50
4.4 A Frame main schema Using NIAM........51
4.5 Object Schema ของเฟรม........51

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.6 Object-Relational Schema ของเฟรม.....52

4.7 อัลกอริทึมในการวินิจฉัยของระบบ.....55

4.8 แสดงการทำงานของกระบวนการภายในเฟรม.....56

4.9 อัลกอริทึมในการหาค่าของสล๊อต.....57

4.10 อัลกอริทึมสำหรับการดึงค่าจากสล๊อต.....57

4.11 อัลกอริทึมสำหรับการหาเฟรมแม่.....57

4.12 แสดงการใช้มุมมองเชิงวัตถุกับฐานข้อมูลภายนอก.....59

4.13 แสดงการเข้าถึงข้อมูลของฐานข้อมูลอื่นจากเฟรม.....59

4.14 โปรโตคอลที่ใช้ในการติดต่อสื่อสารระหว่างไคลเอนท์กับเซิร์ฟเวอร์.....60

5.1 แสดงขั้นตอนการทำงานของระบบวินิจฉัยโรค.....63

5.2 เฟรมผู้ป่วยที่ฝั่งไคลเอนท์.....70

5.3 ตัวอย่างเฟรมผู้ป่วยที่ฝั่งไคลเอนท์.....74

บทที่ 1

บทนำ

1.1 ความเป็นมา และความสำคัญของปัญหา

ระบบผู้เชี่ยวชาญ เป็นสาขาหนึ่งในการศึกษาด้านปัญญาประดิษฐ์ และเป็นโปรแกรมคอมพิวเตอร์ที่มีพฤติกรรมเลียนแบบความฉลาดของมนุษย์ ซึ่งจะประกอบไปด้วยสองส่วน คือส่วนฐานความรู้ และส่วนกลไกการวินิจฉัยเรียกว่า เปลือกระบบผู้เชี่ยวชาญ (expert system shell)

อย่างไรก็ตามเปลือกระบบผู้เชี่ยวชาญส่วนใหญ่จะมีฐานความรู้อยู่ภายใน ซึ่งจะต้องมีการโหลดข้อมูลจากฐานความรู้เข้ามายังหน่วยความจำหลักระหว่างกระบวนการวินิจฉัยเพื่อทำการวินิจฉัยแก้ปัญหาอีกที และเพื่อรองรับขนาดของฐานความรู้ที่เพิ่มมากขึ้นเรื่อยๆ ในปัจจุบันการใช้รูปแบบการแทนความรู้แบบเฟรม (Frame-based Knowledge Representation) เป็นเทคนิคที่ได้รับความยอมรับว่าสามารถสร้างตัวแทนความรู้ที่มีความซับซ้อน และมีขนาดใหญ่ สำหรับระบบฐานความรู้ที่ใช้ในระบบผู้เชี่ยวชาญขนาดใหญ่ได้เป็นอย่างดี ดังจะเห็นได้จากความนิยมในการนำหลักการนี้มาพัฒนา และวิจัยระบบผู้เชี่ยวชาญต่างๆ [1][2][3] และนอกจากนี้ยังได้มีการนำระบบฐานข้อมูลเข้ามาจัดเก็บฐานความรู้ด้วย อย่างเช่น Perk, Sophia, PARKA-DB และ EcoCyc เป็นต้น [3][4][5]

ซึ่งในระบบผู้เชี่ยวชาญที่มีขนาดไม่ซับซ้อนมากนัก จะใช้การโหลดทุกๆเฟรมทั้งที่มีความเกี่ยวข้องและไม่เกี่ยวข้องเข้ามายังระบบผู้เชี่ยวชาญเพื่อทำการวินิจฉัย ส่วนในระบบที่มีซับซับซ้อนมากขึ้นจะโหลดเฉพาะเฟรมที่เกี่ยวข้องเท่านั้น ซึ่งในระบบที่กล่าวมาทั้งหมดนี้ จะไม่มีการวินิจฉัยที่บนฐานความรู้โดยตรงเลย เป็นแต่เพียงการจัดเก็บเฟรมในระบบฐานข้อมูลเท่านั้น และทำการดึงเฟรมจากฐานความรู้ที่จัดเก็บอยู่ในระบบฐานข้อมูลขึ้นมาทำการวินิจฉัย เท่านั้น

ซึ่งฐานความรู้นี้โดยทั่วๆไปจะไม่มีเครื่องอำนวยความสะดวกในการจัดการกับข้อมูลขั้นสูงเหมือนที่มีอยู่แล้วในระบบฐานข้อมูลขนาดใหญ่ตามท้องตลาด อย่างเช่น การสร้างและการใช้งานอินเดกซ์, การออปติไมซ์ คิวรี, การควบคุมแบบขนาน และ ระบบการกู้ข้อมูล ซึ่งมีอยู่แล้วในระบบการจัดการฐานข้อมูลสมัยใหม่ทั่วๆไป

ในระหว่างกระบวนการวินิจฉัยนั้น ข้อเท็จจริงจะได้มาจากสองส่วน คือส่วนแรกคือข้อเท็จจริงที่ได้มาจากผู้ใช้ป้อนเข้าสู่ระบบในลักษณะโต้ตอบไปมา กับส่วนที่สองคือข้อเท็จจริงที่ได้จากการวินิจฉัยแก้ปัญหา กฎในการวินิจฉัยในระบบผู้เชี่ยวชาญทั่วไปจะถูกกระทำโดยกลไกการวินิจฉัยของระบบผู้เชี่ยวชาญซึ่งใช้ข้อเท็จจริงที่ได้จะมาจากการสอบถามโต้ตอบกับผู้ใช้เสียเป็นส่วนใหญ่

และเนื่องจากว่าระบบผู้เชี่ยวชาญบางครั้งมีการถามผู้ใช้บ่อยเกินไป ทำให้เมื่อมีการใช้งานบ่อยครั้งเข้า จะก่อให้เกิดความไม่สะดวกในการใช้ได้ และทำให้การป้อนข้อมูลของผู้ใช้เพื่อนำไปประกอบการตัดสินใจเกิดความผิดพลาดขึ้นได้ อีกทั้งบางคำถามอาจจะเกี่ยวข้องกับข้อมูลในอดีตหรือข้อมูลที่ผู้ใช้ เองอาจจะไม่ทราบ และต้องไปสืบค้นดูอีกที อย่างเช่นระบบฐานความรู้ที่เกี่ยวข้องกับการวินิจฉัยโรคที่มีการถามถึงข้อมูลประวัติการรักษา และการแพ้ยา ซึ่งผู้ใช้จะต้องทำการค้นข้อมูลเพื่อตอบคำถามให้กับระบบผู้เชี่ยวชาญอีกที เป็นต้น และถ้าเราต้องการจะหาว่าการผ่าตัดที่จะเกิดขึ้นมีความเสี่ยงในการที่จะเกิดภาวะหัวใจล้มเหลวหรือไม่ หรือควรจะมีการทำการผ่าตัดในกรณีนั้นๆหรือไม่ ตัวระบบฐานความรู้ ควรจะต้องส่งค่าที่ระบบผู้เชี่ยวชาญ ต้องการจริงๆ กลับมา หรือส่งคำตอบที่ได้จากการถามนี้กลับมา ซึ่งอาจจะเป็นข้อมูลทางสถิติ หรือปัจจัยความเสี่ยงต่างๆ ซึ่งทำให้ระบบผู้เชี่ยวชาญสามารถทำการวินิจฉัยต่อได้ทันที ไม่ใช่เป็นการโหลดเฟรมทั้งหมด หรือบางส่วน มาไว้ที่เปลือกระบบผู้เชี่ยวชาญ แล้วค่อยทำการวินิจฉัยอีกที และในขณะเดียวกัน ถ้ามีการเรียกใช้ข้อมูลเพื่อทำการวินิจฉัยเพิ่มเติมจากฐานความรู้อื่นๆ อย่างเช่นข้อมูลเวชระเบียน ซึ่งจะมีการเก็บประวัติคนไข้และผลการรักษาที่ผ่านมาพร้อมทั้งประวัติการแพ้ยาไว้อยู่แล้วเป็นต้น ก็จะต้องมีการโหลดข้อมูลจากเวชระเบียนมาอีกที ซึ่งเป็นอีกฐานข้อมูลหนึ่ง ทำให้การทำงานของระบบผู้เชี่ยวชาญจะมีความซับซ้อนมากขึ้น และทำให้การพัฒนาระบบมีความยุ่งยากและซับซ้อนเป็นอย่างมาก

งานวิจัยนี้จึงได้นำเสนอสถาปัตยกรรมที่แตกต่างออกไป โดยได้สร้างกลไกการวินิจฉัยไปไว้บนฐานความรู้ที่จัดเก็บอยู่ในฐานข้อมูลด้วย ทำให้กระบวนการวินิจฉัยถูกกระทำที่บนฝั่งฐานความรู้ที่จัดเก็บอยู่ในระบบฐานข้อมูลด้วย แทนที่จะกระทำที่ฝั่งเปลือกระบบผู้เชี่ยวชาญอย่างเดียว และขณะเดียวกันที่ฝั่งเปลือกระบบผู้เชี่ยวชาญ ก็มีฐานความรู้อยู่ด้วยเหมือนกัน ซึ่งวิธีการนี้จะทำให้การวินิฉัยสามารถกระทำได้ทั้งสองฝั่งคือที่ฝั่งเปลือกระบบผู้เชี่ยวชาญ และฝั่งฐานความรู้ที่จัดเก็บอยู่ในฐานข้อมูล ซึ่งทำให้ระบบมีความคล่องตัวในการวินิจฉัย เนื่องถ้าระบบผู้เชี่ยวชาญสามารถตอบคำถามผู้ใช้ได้จากการโต้ตอบโดยไม่ต้องมีการถามถึงข้อมูลในอดีตเลย แต่มีการร้องขอข้อเท็จจริงมีไม่ซับซ้อนมากนัก ระบบก็จะสามารถให้คำตอบกับผู้ใช้ได้ทันที แต่ถ้ามีการถามผู้ใช้ถึงข้อมูลที่ต้องไปสืบค้นจากฐานข้อมูลอื่นด้วย การวินิจฉัยก็จะกระทำการที่ส่วนฐานความรู้เลย ดังนั้นจะมีเพียงผลลัพธ์สุดท้ายของการวินิจฉัยจากทางฐานความรู้เท่านั้นที่จะถูกส่งกลับมายังเปลือกระบบผู้เชี่ยวชาญเท่านั้น แทนที่จะเป็นการส่งข้อเท็จจริงจากฐานข้อมูลหลายๆครั้งแทน ในระหว่างกระบวนการวินิจฉัย

จากการศึกษาพบว่าการพัฒนาระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ เพื่อรองรับความต้องการในการใช้งานข้อมูลที่มีความซับซ้อนขึ้น เป็นแนวโน้มที่กำลังได้รับความนิยมอยู่ในขณะนี้ โดยจะเห็นได้จากการที่บริษัทผู้ผลิตซอฟท์แวร์ระบบการจัดการฐานข้อมูลขนาดใหญ่ต่างๆ ที่หันมาให้ความสำคัญ และพัฒนาผลิตภัณฑ์ของตนให้เป็นระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ ซึ่งถึงแม้

มาตรฐานเทคโนโลยีที่ใช้จะขึ้นอยู่กับแต่ละผลิตภัณฑ์เอง แต่หลักการส่วนใหญ่จะอ้างอิงมาจากทั้งมาตรฐานของ ANSI และ OMG ทำให้ช่วยเพิ่มความสะดวกให้แก่โปรแกรมเมอร์ในการพัฒนาซอฟท์แวร์ที่สร้างขึ้นภายใต้หลักการโปรแกรมเชิงวัตถุที่ต้องการใช้งานระบบฐานข้อมูล ซึ่งนอกจากการพัฒนาระบบฐานข้อมูลใหม่ขึ้นมานี้จะสามารถสนับสนุนการพัฒนาโปรแกรม หรือระบบสารสนเทศใหม่ให้แก่องค์กรต่างๆได้โดยที่ไม่จำเป็นต้องเปลี่ยนไปใช้ระบบฐานข้อมูลเชิงวัตถุ ขณะเดียวกันก็ยังสามารถสนับสนุนระบบงานเดิมที่มีอยู่ได้ด้วยเช่นกัน

งานวิจัยนี้จึงได้นำเสนอสถาปัตยกรรมในการออกแบบระบบฐานความรู้โดยใช้ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ในการแทนความรู้แบบเฟรมที่ฝังระบบฐานข้อมูล เพราะเราต้องการความสามารถในจัดการเฟรมในระดับตัวบ่งบอกคุณสมบัติของเฟรมได้ และขณะเดียวกันสามารถเชื่อมต่อกับระบบฐานข้อมูลอื่นที่มีอยู่แล้วซึ่งส่วนใหญ่จะเป็นระบบฐานข้อมูลเชิงสัมพันธ์ได้ นอกจากนี้ยังเป็นการง่ายต่อการดึงข้อมูลจากฐานข้อมูลตัวอื่นๆ ซึ่งมีข้อมูลบางส่วนที่ต้องใช้อยู่แล้ว เพราะเป็นการติดต่อกันระหว่างฐานข้อมูลสองตัว

และเนื่องจากงานวิจัยนี้ได้มีการวินิจฉัยแบบสองทางคือที่ฝั่งเปลือกระบบผู้เชี่ยวชาญและฐานความรู้ที่จัดเก็บอยู่ในระบบฐานข้อมูล งานวิจัยนี้จึงได้ใช้ประโยชน์จากภาษาที่เป็นมาตรฐานที่ใช้ในการติดต่อสื่อสารระหว่างระบบมัลติเอเจ้นท์มาใช้ในการติดต่อสื่อสารและแลกเปลี่ยนความรู้ระหว่างเปลือกระบบผู้เชี่ยวชาญและฐานความรู้ภายนอกด้วย ซึ่งภาษาที่ใช้ในงานวิจัยนี้คือภาษา KQML (Knowledge Query and Manipulation Language) [6] ซึ่งเป็นรูปแบบโปรโตคอลที่ถูกใช้อย่างกว้างขวางในการติดต่อสื่อสารระหว่างระบบฐานความรู้ด้วย

ระบบต้นแบบจะถูกพัฒนาขึ้นบนฐานข้อมูลเชิงวัตถุสัมพันธ์โดยใช้ระบบการจัดการฐานข้อมูล ออราเคิล 10g [7] เนื่องจากเป็นฐานข้อมูลขนาดใหญ่ที่ได้รับความนิยมและใช้อยู่อย่างแพร่หลายและเป็นระบบฐานข้อมูลสมัยใหม่ที่ถูกพัฒนามาล่าสุด รวมทั้งมีความคล่องตัวต่อการติดต่อกับฐานข้อมูลทั่วๆไป และงานวิจัยนี้ยังใช้การประมวลผลข้อมูลจากฐานความรู้ที่จัดเก็บไว้ที่ระบบฐานข้อมูลอีกด้วยทำให้เพิ่มความสามารถของระบบฐานข้อมูลที่เก็บความรู้แบบเฟรมนี้ให้มีประสิทธิภาพมากยิ่งขึ้น นอกจากนี้ยังเป็นการ ลดความยุ่งยาก ที่จะเกิดขึ้นในการ โหลดข้อมูลจากฐานข้อมูล เข้ามายัง ระบบผู้เชี่ยวชาญ เพื่อทำการวินิจฉัย และลดปัญหาความยุ่งยากและความผิดพลาดในการถามผู้ใช้ถึงข้อมูลที่มีอยู่แล้วทุกครั้งได้จากการดึงข้อมูล จากฐานข้อมูลที่มีอยู่แล้วอีกด้วย นอกจากนี้ยังเป็นการง่ายต่อการดึงข้อมูลจากฐานข้อมูล ตัวอื่นๆ ซึ่งมีข้อมูลบางส่วนที่ต้องใช้อยู่แล้ว โดยไม่จำเป็นต้องปรับเปลี่ยนการทำงานของระบบผู้เชี่ยวชาญมากนัก

1.2 วัตถุประสงค์

- เพื่อใช้เทคโนโลยีของระบบฐานข้อมูลเพื่อเพิ่มประสิทธิภาพการสืบค้นและปรับปรุงฐานความรู้ขนาดใหญ่ให้มีประสิทธิภาพยิ่งขึ้น ซึ่งในงานวิจัยนี้ได้ทำการแมบโครงสร้างของการแทนความรู้แบบเฟรม เข้ากับฐานข้อมูลเชิงวัตถุสัมพันธ์ และทำการออกแบบและพัฒนากลไกการวินิจฉัย (inference engine) บน ฐานความรู้ที่จัดเก็บอยู่ในระบบฐานข้อมูลด้วย นอกจากนี้ยังเป็นการ ลดค่าใช้ที่จะเกิดขึ้นในการ โหลดข้อมูลจากฐานข้อมูลอื่น เข้ามายังเปลือกระบบผู้เชี่ยวชาญ เพื่อทำการวินิจฉัย และลดปัญหาความยุ่งยากและความผิดพลาดในการถามผู้ใช้ ถึงข้อมูลที่มีอยู่แล้วทุกครั้งได้จากการดึงข้อมูล จากฐานข้อมูลที่มีอยู่แล้วอีกด้วย นอกจากนี้ยังเป็นการง่ายต่อการดึงข้อมูลจากฐานข้อมูล ตัวอื่นๆ ซึ่งมีข้อมูลบางส่วนที่ต้องใช้อยู่แล้ว เพราะเป็นการติดต่อกันระหว่างฐานข้อมูลสองตัว โดยไม่จำเป็นต้องปรับเปลี่ยนการทำงานของระบบผู้เชี่ยวชาญมากนัก
- พัฒนาให้ระบบมีความสามารถเพิ่มมากขึ้นในการวินิจฉัยได้ทั้งสองฝั่ง คือฝั่งระบบผู้เชี่ยวชาญที่ทำหน้าที่ติดต่อกับผู้ใช้ และฝั่งระบบฐานความรู้ส่วนที่จัดเก็บอยู่ในระบบฐานข้อมูล เพื่อเพิ่มประสิทธิภาพของระบบ และลดปัญหาการส่งข้อมูลจำนวนมาก และหลายๆ ครั้ง ระหว่างระบบผู้เชี่ยวชาญ กับฐานข้อมูล
- พัฒนาให้ระบบมี โปรโตคอล มาตรฐานในการสื่อสารระหว่างทั้งสองฝั่งโดยในงานวิจัยนี้ โปรโตคอล ที่ใช้ คือ KQML เนื่องจากเป็น โปรโตคอล ที่ใช้กันอย่างแพร่หลายในการติดต่อสื่อสารระหว่างระบบในงานทางด้านปัญญาประดิษฐ์
- ให้ระบบสามารถทำการติดต่อกับฐานข้อมูลภายนอกอื่นๆที่มีข้อเท็จจริงที่เกี่ยวข้อง เพื่อการใช้งานให้มีประสิทธิภาพยิ่งขึ้นได้

1.3 สมมติฐานของการศึกษา

ในงานวิจัยนี้ระบบจะต้องมีความสามารถในการทำการวินิจฉัยได้ทั้งสองทาง โดยที่ในส่วนฐานความรู้ที่จัดเก็บอยู่ในระบบฐานข้อมูลจะต้องส่งผลลัพธ์ที่ผ่านการวินิจฉัยเรียบร้อยแล้วกลับมายังเปลือกระบบผู้เชี่ยวชาญได้ และระบบจะต้องสามารถนำข้อมูลจากฐานข้อมูลอื่น ซึ่งจัดเก็บไว้อยู่แล้วมาใช้ร่วมในกระบวนการวินิจฉัยที่ฝั่งฐานข้อมูลได้

โดยในงานวิจัยได้นำเสนอกรณีศึกษาระบบผู้เชี่ยวชาญการวินิจฉัยโรค เพื่อทดสอบสมมติฐาน ข้างต้น

1.4 ขอบเขตการศึกษา

1.4.1 ออกแบบระบบผู้เชี่ยวชาญที่ใช้รูปแบบการแทนความรู้แบบเฟรม บนระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์

1.4.2 การวินิจฉัยแบบเฟรมบนระบบฐานข้อมูลเชิงวัตถุสัมพันธ์

1.4.3 การติดต่อสื่อสารด้วยภาษา KQML

1.4.4 พัฒนาระบบผู้เชี่ยวชาญเป็นกรณีศึกษา โดยใช้ระบบการวินิจฉัยโรคเป็นกรณีศึกษา ซึ่งระบบผู้เชี่ยวชาญที่ได้จะต้องมีความสามารถดังต่อไปนี้

1.4.4.1 สามารถทำการวินิจฉัยโดยผู้ใช้ป้อนข้อมูลอาการป่วยของผู้ป่วย เข้ามาเท่าที่จำเป็นต้องทราบได้

1.4.4.2 สามารถดึงข้อมูลจากฐานข้อมูลภายนอกเข้ามาช่วยในการวินิจฉัยได้

1.4.4.3 สามารถติดต่อสื่อสารกันด้วย KQML ได้

1.5 ขั้นตอนการศึกษา

เพื่อให้การดำเนินการศึกษา และพัฒนาเป็นไปด้วยความเรียบร้อยวิทยานิพนธ์ฉบับนี้ได้แบ่งขั้นตอนการดำเนินงานไว้ดังนี้

1.5.1 ศึกษา และเปรียบเทียบหลักการเชิงวัตถุสัมพันธ์ และเฟรม

1.5.2 พัฒนาโครงสร้างของเฟรมสำหรับระบบฐานความรู้ ด้วยโปรแกรมระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ ออราเคิล 10g

1.5.3 ศึกษากลไกการวินิจฉัยแบบเฟรม

1.5.3 ศึกษา KQML ซึ่งเป็นโปรโตคอลมาตรฐาน

1.5.4 ศึกษากระบวนการวินิจฉัยโรคเป็นกรณีศึกษา (Knowledge Acquisition)

1.5.5 พัฒนาระบบผู้เชี่ยวชาญสำหรับงานการวินิจฉัยโรคเป็นกรณีศึกษา

1.5.6 จัดทำเอกสาร

1.6 โครงสร้างของวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้แบ่งเนื้อหาออกเป็น 6 บท แต่ละบทประกอบด้วยเนื้อหาดังนี้

บทที่ 1 กล่าวถึงความเป็นมา และความสำคัญของปัญหา ความมุ่งหมาย และวัตถุประสงค์ของการศึกษา สมมติฐานของการศึกษา ขอบเขตของงานวิจัย และขั้นตอนของการศึกษา

บทที่ 2 กล่าวถึงทฤษฎีเกี่ยวกับเฟรมและระบบเฟรม กลไกการวินิจฉัยในระบบเฟรม รวมทั้งทฤษฎีของระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ วิธีการในการเชื่อมโยงระบบ

ผู้เชี่ยวชาญกับระบบฐานข้อมูลที่ได้มีอยู่ และระบบที่มีการเชื่อมโยงระหว่างระบบผู้เชี่ยวชาญกับระบบฐานข้อมูล ที่มีอยู่แล้ว

บทที่ 3 กล่าวถึงสถาปัตยกรรมของระบบในส่วนของการออกแบบระบบ ซึ่งจะกล่าวถึงการทำงานของระบบที่ฝั่งเซิร์ฟเวอร์ และฝั่งไคเอนท์ พร้อมทั้งโปรโตคอลในการเชื่อมต่อ

บทที่ 4 กล่าวถึงขั้นตอนในการสร้างระบบ พร้อมทั้งวิธีการในการสร้างระบบ อธิบายถึงการแปลงเฟรมให้อยู่ในรูปตาราง เทคนิคในการทำการวินิจฉัยบนระบบที่สร้างขึ้น การติดต่อกับระบบฐานข้อมูลภายนอกอื่นๆ

บทที่ 5 กล่าวถึงระบบต้นแบบ เพื่อทดลองระบบที่ได้ออกแบบไว้ข้างต้น พร้อมทั้งตัวอย่างระบบผู้เชี่ยวชาญการวินิจฉัยโรค และผลการทดลอง

บทที่ 6 สรุปผลการทดลอง ข้อเสนอแนะ และแนวทางการพัฒนาระบบ ต่อไปในอนาคต

บทที่ 2
ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

2.1 ระบบผู้เชี่ยวชาญ

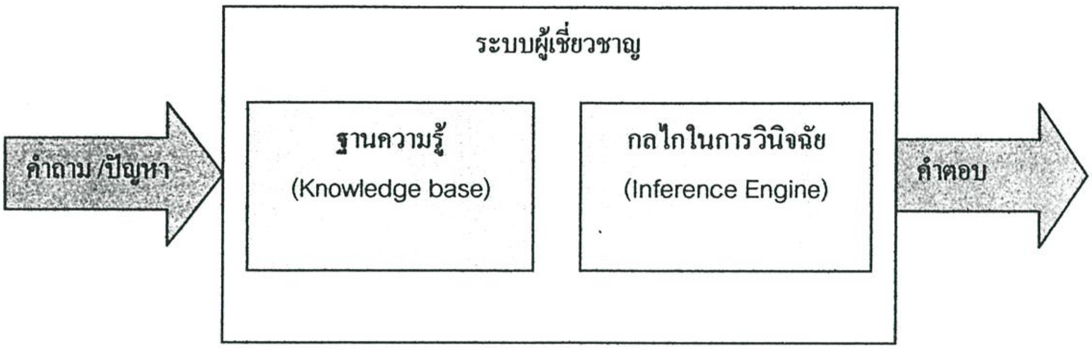
เป็นโปรแกรมคอมพิวเตอร์ที่ถูกออกแบบมาเพื่อทำหน้าที่เหมือนเป็นผู้เชี่ยวชาญในการแก้ปัญหาในขอบเขตนั้นๆ ซึ่งโปรแกรมจะใช้ความรู้ในขอบเขตที่ถูกโค็ดเข้าไปในนั้นและระบุแผนการควบคุมความรู้นั้นเพื่อให้ได้มาซึ่งคำตอบหรือวิธีการแก้ปัญหา โดยระบบผู้เชี่ยวชาญส่วนมากมักถูกเรียกว่าระบบฐานความรู้ด้วยเหมือนกัน ระบบผู้เชี่ยวชาญนั้นจะไม่เรียกว่าเป็นโปรแกรม แต่เป็นระบบเพราะมันรายล้อมไปด้วยส่วนประกอบหลายส่วน อย่างเช่น ฐานความรู้ กลไกการวินิจฉัย ส่วนอธิบาย เป็นต้น ซึ่งส่วนประกอบเหล่านี้จะมีผลต่อกันและกันในการเลียนแบบกระบวนการแก้ปัญหาโดยที่ได้รับการยอมรับจากผู้เชี่ยวชาญในขอบเขตนั้นๆ[8]

ระบบผู้เชี่ยวชาญจะจำลองกระบวนการตัดสินใจโดยใช้ข้อเท็จจริงและความรู้ ซึ่งต้องการมากกว่าความรู้เดียวในกระบวนการตัดสินใจ และส่วนประกอบหลักของระบบผู้เชี่ยวชาญประกอบไปด้วยสองส่วนคือ [9]

ฐานความรู้ - ซึ่งเป็นกลุ่มของความรู้ที่ต้องใช้ในการแก้ปัญหา และ

กลไกการ*วินิจฉัย* - ซึ่งใช้ตรวจสอบข้อเท็จจริงที่มีอยู่ เลือกแหล่งความรู้ที่เหมาะสามจากฐานความรู้ จับคู่ข้อเท็จจริงกับความรู้ และก่อให้เกิดข้อเท็จจริงเพิ่มเติม

แต่เนื่องจากปัญหามีเพิ่มขึ้นเรื่อยๆ และข้อเท็จจริงจำนวนมากได้มาจากข้อเท็จจริงที่มีอยู่แล้ว ดังนั้นในแอพพลิเคชั่นการตัดสินใจหรือข้อเท็จจริงจะเกิดขึ้นต่อๆกันไปเป็นลำดับโดยใช้ข้อเท็จจริงที่มีอยู่ ซึ่งกระบวนการนี้จะใช้ข้อเท็จจริงปัจุบันจะความรู้ที่อยู่ในฐานความรู้เพื่อก่อให้เกิดข้อเท็จจริงเพิ่มเติมต่อไปเรื่อยๆเป็นห่วงโซ่ จนกว่าข้อเท็จจริงที่เป็นเป้าหมายจะถูกค้นพบ กลไกนี้ถูกเรียกว่าการวินิจฉัย ซึ่งมีได้หลายแนวทางในการที่ความรู้ในรูปแบบของกฎจะถูกทำการวินิจฉัย ดังนั้นกลไกที่ใช้ควบคุมสามารถประกอบไปด้วยแนวทางการวินิจฉัยได้หลากหลายแนวทาง ดังนั้น ฐานความรู้ และ กลไกการวินิจฉัย จึงเป็นส่วนประกอบหลักของระบบฐานความรู้ [10] และส่วนประกอบหลักของระบบผู้เชี่ยวชาญดังแสดงในรูปที่ 2.1



รูปที่ 2.1 ส่วนประกอบหลักของระบบผู้เชี่ยวชาญ

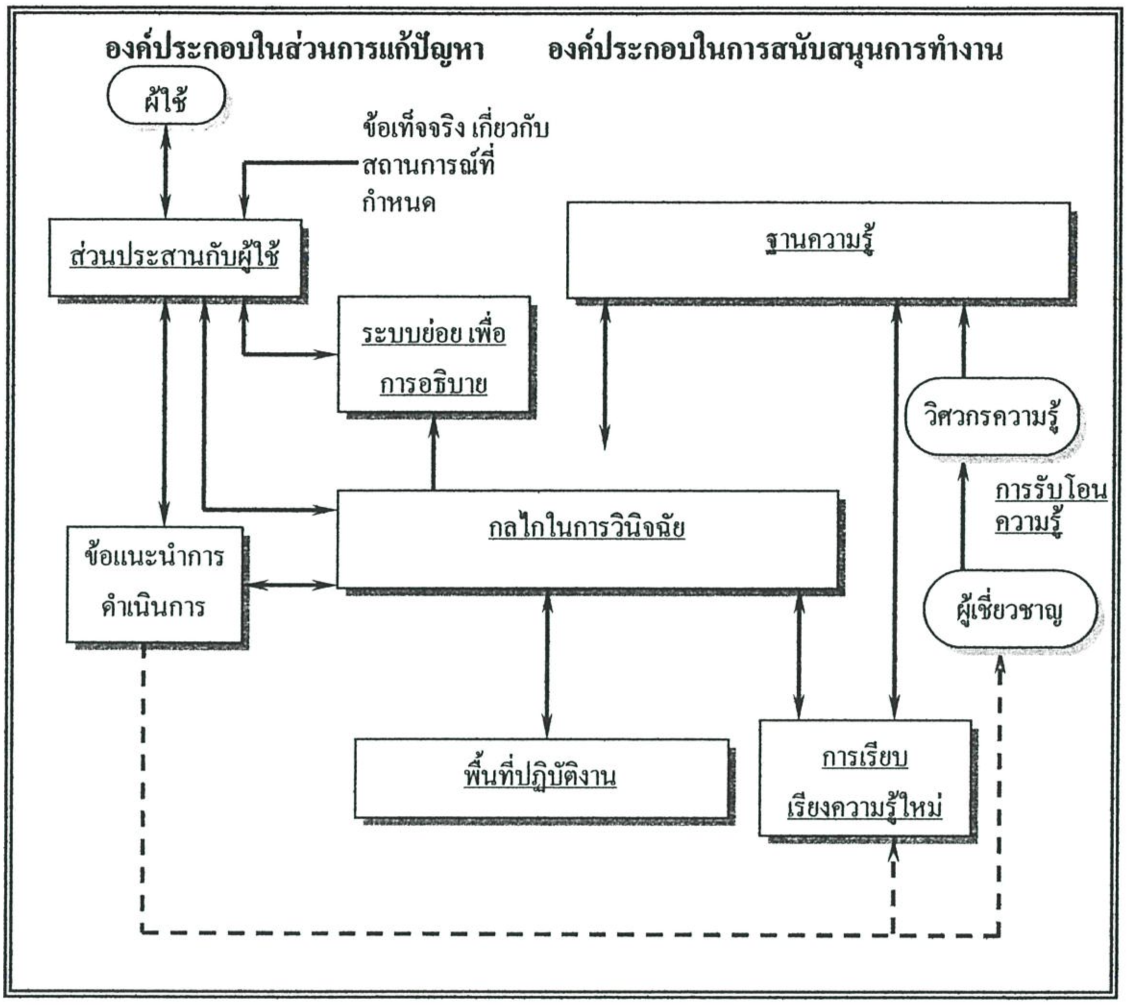
2.1.1 สถาปัตยกรรมของระบบผู้เชี่ยวชาญ

จากที่ได้อธิบายในห้อข้อข้างต้น ฐานความรู้และกลไกการวินิจฉัยจะประกอบไปด้วยกลไกหลายกลไก เมื่อระบบผู้เชี่ยวชาญเริ่มกระบวนการวินิจฉัย มันจะถูกร้องขอให้จัดเก็บข้อเท็จจริงเพื่อใช้งานในอนาคต ซึ่งเซตของข้อเท็จจริงที่ถูกตั้งขึ้นมาจะแทนเนื้องาน(context) เช่น สถานะปัจจุบันของปัญหาที่กำลังถูกแก้ไข ดังนั้นส่วนประกอบนี้ส่วนมากจะถูกเรียกว่า เนื้องาน (context) หรือหน่วยความจำปฏิบัติงาน (working memory)

เมื่อไหร่ก็ตามที่ผู้เชี่ยวชาญใช้การตัดสินใจ เราก็อยากจะรู้ว่าผู้เชี่ยวชาญตัดสินใจได้อย่างไร นอกจากนี้เมื่อผู้เชี่ยวชาญถามถึงข้อมูล เราก็อยากจะรู้ว่าทำไมข้อมูลนั้นจึงถูกร้องขอ ผู้เชี่ยวชาญจะใช้ความรู้ที่พวกเขามีและเนื้อหาของปัญหาในการตอบคำถามอย่างเช่นว่า ทำไมถึงตัดสินใจอย่างนั้น? หรือว่าทำไมต้องการข้อมูลนี้? ซึ่งส่วนนี้ก็เป็นส่วนประกอบภายในของระบบผู้เชี่ยวชาญเหมือนกัน

กระบวนการในการรวบรวม บริหารจัดการ และคอมไพล์ความรู้ พร้อมทั้งสร้างมันในรูปแบบของฐานความรู้นั้นเป็นงานที่ต้องใช้ความพยายาม เพราะมันไม่ได้จบลงในส่วนของการสร้างระบบ ฐานความรู้จะต้องถูกอัพเดต หรือไม่ก็ถูกเพิ่มเข้าไป ขึ้นอยู่กับการเพิ่มขึ้นของความรู้ในขอบเขตของความรู้นั้นๆ เครื่องอำนวยความสะดวกในส่วนการรับโอนความรู้ ซึ่งจะทำหน้าที่เหมือนเป็นส่วนประสานงานระหว่างผู้เชี่ยวชาญหรือวิศวกรความรู้กับฐานความรู้นั้นสามารถเป็นองค์ประกอบภายในของระบบผู้เชี่ยวชาญได้

ผู้ใช้ในระบบผู้เชี่ยวชาญต้องติดต่อกับระบบในการป้อนข้อมูล กำหนดข้อเท็จจริง และสังเกตสถานะของการแก้ปัญหา ในส่วนติดต่อกันผู้ใช้จะเป็นส่วนที่เป็นรูปแบบและข้อความที่แสดงให้ผู้ใช้ทำการติดต่อกับระบบได้ รูปที่ 2.2 แสดงสถาปัตยกรรมของระบบฐานความรู้และองค์ประกอบในฐานความรู้ รวมทั้งวิธีการที่แต่ละองค์ประกอบจะติดต่อซึ่งกันและกัน



รูปที่ 2.2 สถาปัตยกรรมของระบบฐานความรู้

ดังที่ได้กล่าวมาแล้ว ฐานความรู้และกลไกการวินิจฉัยจะเป็นส่วนที่สำคัญที่สุดของระบบฐานความรู้ ดังนั้นในหัวข้อถัดไปจะอธิบายรายละเอียดเกี่ยวกับเทคโนโลยีทางด้านฐานความรู้และกลไกการวินิจฉัย

2.1.2 ฐานความรู้ (Knowledge base)

ฐานความรู้จะมีความรู้ในขอบเขตของปัญหาที่เฉพาะเจาะจงเพื่อการแก้ปัญหานั้นๆ ฐานความรู้ถูกสร้างขึ้นโดยวิศวกรความรู้ที่ได้มาจากการสอบถามผู้เชี่ยวชาญและทำการเรียบเรียงความรู้ในรูปแบบที่ระบบสามารถนำไปใช้งานได้ วิศวกรความรู้ต้องมีเทคโนโลยีในการแทนความรู้และควรรู้ว่าจะพัฒนาระบบผู้เชี่ยวชาญอย่างไร ซึ่งก่อนที่จะตัดสินใจว่าโครงสร้างของฐานความรู้จะเป็นแบบไหน วิศวกรความรู้ควรจะเลือกได้ว่าน่าจะใช้การแทนความรู้แบบใดและความเหมาะสมในการเลือกใช้เทคนิคนั้นๆภายใต้สถานการณ์ต่างๆ

ความรู้ในการแก้ปัญหานั้นสามารถจำแนกได้เป็น 3 กลุ่มหลักๆ คือ ความรู้ที่ผ่านการคอมไพล์แล้ว (compiled knowledge) ความรู้เชิงคุณภาพ (qualitative knowledge) และความรู้เชิงปริมาณ (quantitative knowledge) ซึ่งความรู้แบบคอมไพล์จะเป็นได้ทั้งผลจากประสบการณ์ของผู้เชี่ยวชาญในขอบเขตนั้นๆ ความรู้ที่ได้จากคู่มือ ข้อมูลเก่า รายการมาตรฐาน และอื่นๆ ส่วนความรู้เชิงคุณภาพจะประกอบไปด้วยกฎตายตัว ทฤษฎีคร่าวๆ โมเดลของกระบวนการอย่างมีเหตุผล และสันชาตญาณ ส่วนความรู้เชิงปริมาณจะจัดการกับเทคนิคบนพื้นฐานของทฤษฎีทางคณิตศาสตร์หรือเทคนิคในทางจำนวนเป็นต้น ซึ่งความรู้เชิงคุณภาพและความรู้ที่ผ่านการคอมไพล์แล้วสามารถถูกจำแนกได้อีกเป็น 2 กลุ่ม คือ ความรู้เชิงพรรณา (declarative knowledge) และความรู้เชิงกระบวนการ(procedural knowledge) ความรู้เชิงพรรณนาจะเกี่ยวข้องกับความรู้บนคุณสมบัติทางกายภาพของขอบเขตของปัญหา ในขณะที่ความรู้เชิงกระบวนการจะเกี่ยวข้องกับเทคนิคการแก้ปัญหา

ในการพัฒนาระบบผู้เชี่ยวชาญจะเกี่ยวข้องกับความรู้ในรูปแบบเหล่านี้ เพราะมันเป็นขอบเขตของแอพพลิเคชันที่ตัดสินธรรมชาติของความรู้ที่อยู่ภายในนั้น ยกตัวอย่างเช่น พิจารณาเกี่ยวกับการวินิจฉัยสภาพอันตรายในการก่อสร้าง ซึ่งจะต้องหาว่าผนังมีความร้าวอย่างไร เพื่อหาสามารถที่ทำให้ผนังร้าว ระบบต้องมีความรู้เกี่ยวกับตำแหน่งของผนัง ชนิดของผนัง ความหนาของผนัง อัตราส่วนที่ใช้ในการผสมปูน ชนิดของสิ่งก่อสร้าง เป็นต้น ในส่วนของชนิดของผนังนั้นความรู้เกี่ยวกับฐานราก ชนิดของดิน รายละเอียดของแบบคานและเสา เป็นต้นจะถูกร้องขอ และเพิ่มเติมจากข้างต้น ความรู้เกี่ยวกับการใช้ข้อมูลที่มีอยู่ในการหาเหตุผลของการร้าวที่ผนังก็จะถูกร้องขอด้วย ซึ่งส่วนแรกของความรู้ (ซึ่งก็คือความรู้เชิงพรรณา) ที่ได้กล่าวถึงข้างต้นจะอธิบายเหตุการณ์ ในขณะที่ความรู้ส่วนที่สอง (ความรู้เชิงกระบวนการ) จะอธิบายถึงการใช้ความรู้ในเพื่อให้ได้มาซึ่งการตัดสินใจ ดังนั้นในการพัฒนาระบบฐานความรู้ จะต้องรู้ถึงความแตกต่างระหว่างการแทนความรู้แต่ละอย่าง และผลกระทบระหว่างมัน

ในการพัฒนาระบบผู้เชี่ยวชาญนั้นจะเกี่ยวข้องกับงานอย่างเช่น การรับโอนความรู้จากผู้เชี่ยวชาญในขอบเขตงานนั้นๆ การจัดการเอกสารและการรวบรวมความรู้ การผลิตเครือข่ายความรู้ (knowledge net) เพื่อตรวจสอบความสัมพันธ์ระหว่างแหล่งความรู้ที่แตกต่างกัน การตรวจสอบความถูกต้องในความรู้ และสุดท้ายคือการแปลงเครือข่ายความรู้ไปเป็นโปรแกรมโดยใช้เครื่องมือที่เหมาะสม ซึ่งโปรแกรมที่ได้มานั้นจะถูกเรียกว่าระบบผู้เชี่ยวชาญ ซึ่งเป็นระบบอย่างเป็นทางการสำหรับการจัดเก็บข้อเท็จจริงและความสัมพันธ์ระหว่างมันและวิธีการในการใช้มัน โดยทั่วๆไปประบบผู้เชี่ยวชาญจะมีความรู้เกี่ยวกับวัตถุ ความสัมพันธ์ระหว่างมัน เหตุการณ์ความสัมพันธ์ระหว่างเหตุการณ์ และความสัมพันธ์ระหว่างวัตถุกับเหตุการณ์ นอกจากนี้ชนิดของกลไกการค้นหาอาจต้องถูกร้องขอเพื่อใช้ในการขับเคลื่อนระบบ ซึ่งการการตัดสินใจเลือกการแทนความรู้นั้นจะขึ้นอยู่กับชนิดของแอพพลิเคชันที่จะสร้าง และวิศวกรความรู้ต้องตัดสินว่าส่วนไหน

ของความรู้ควรถูกแทนในรูปแบบใดได้นั้นขึ้นอยู่กับลักษณะของความรู้และประสิทธิภาพในการใช้มัน

ฐานความรู้และกลไกการวินิจฉัยจะเป็นส่วนที่สำคัญที่สุดของระบบฐานความรู้ ดังนั้นในหัวข้อถัดไปจะอธิบายรายละเอียดเกี่ยวกับเทคโนโลยีทางด้านฐานความรู้และกลไกการวินิจฉัย

2.2 การแทนความรู้ (Knowledge Representation)

ในเวลาที่ผ่านมา ได้มีการศึกษาวิจัย และกำหนดรูปแบบการสร้างตัวแทนความรู้ไว้หลายรูปแบบ แต่เราสามารถจัดเป็นกลุ่มใหญ่ ที่ได้รับความนิยมได้ 3 รูปแบบได้แก่ แบบใช้กฎ (Production Rule) แบบเครือข่ายความหมาย (Semantic Network) และแบบใช้เฟรม (Frame) ซึ่งในวิทยานิพนธ์เล่มนี้ จะขอกล่าวถึงเฉพาะแบบใช้กฎ และแบบใช้เฟรมเท่านั้น เนื่องจากในการสร้างตัวแทนความรู้แบบใช้กฎนั้นเป็นวิธีพื้นฐานที่นิยมใช้ในการพัฒนาระบบผู้เชี่ยวชาญมากที่สุด และการสร้างตัวแทนความรู้แบบใช้เฟรม เป็นรูปแบบที่วิทยานิพนธ์เล่มนี้จะนำมาเป็นแนวทางในการพัฒนาระบบผู้เชี่ยวชาญเชิงวัตถุสัมพันธ์

2.2.1 การแทนความรู้แบบกฎ (Production Rules) เป็นรูปแบบที่ไม่ซับซ้อนแต่ทรงพลังสำหรับการแทนความรู้ ซึ่งมันจะมีความยืดหยุ่นในการผสมผสานความรู้เชิงพรรณากับความรู้เชิงกระบวนการ ซึ่งในการแทนความรู้แบบกฎนั้นจะประกอบไปด้วยเซตของเหตุ (antecedents) และเซตของผล (consequents) ซึ่งเหตุจะระบุเงื่อนไขหรือเหตุการณ์ และผลเป็นเซตของการกระทำที่เกิดจากเหตุนั้น ตัวอย่างเช่น

IF span of beam is known
THEN depth of beam is (span of bema/12)

IF flow is open channel flow
AND Fround number > 1
THEN flow is supercritical

IF flow is open channel flow
AND Fround number = 1
THEN flow is critical

IF the object has two openings
AND opening at top is larger than that at the bottom
AND enough space is available on left side
AND space available on right side is too small
THEN take the pipe by the left side of the object

AND take pipe into the object from top opening

เหตุการที่ระบุในกฎที่หนึ่งและสองนั้นเรียบง่าย แต่กฎในข้อที่สามนั้นซับซ้อน ค่าที่ถูกใส่ให้กับตัวแปรสามารถถูกใช้แทนข้อเท็จจริงในกฎข้อที่หนึ่งและสองแต่ในกรณีกฎข้อที่สี่ วัตถุและคุณสมบัติของมันต้องถูกกำหนดในการแทนข้อเท็จจริง ในกรณีทั้งหมดข้างต้น เมื่อส่วนเหตุ (IF) ของกฎนั้นเป็นจริงโดยข้อเท็จจริงที่ถูกจัดเก็บในส่วนเนื้อหา หรือข้อเท็จจริงนั้นถูกป้อนเข้ามาโดยผู้ใช้ การกระทำที่ระบุไว้ในส่วนผล (THEN) จะถูกกระทำและกฎจะทำงาน โดยทั่วไปแล้วฐานความรู้จะประกอบไปด้วยกฎหลายๆกฎ ซึ่งตามเหตุผลแล้วกฎจะถูกรวบรวมอยู่ในฐานกฎ (rule bases)

2.2.2 การแทนความรู้แบบเฟรม (Frames)

การแทนความรู้แบบเฟรมได้ถูกเสนอเป็นครั้งแรกโดย Marvin Minsky ในปี 1974 [13] และได้มีการนำเสนอแนวคิดนี้ในงานวิจัยจนถึงปัจจุบัน(ปี 2004) [1] แนวคิดของเฟรมคือโครงสร้างข้อมูล (data structure) สำหรับรวบรวมข้อมูลชนิดต่างๆของแนวคิดหรือความรู้ (typical knowledge) และเรียกว่าโครงสร้างแบบนี้ว่าเฟรม [12][14] ดังนั้นเฟรมเป็นการแทนความรู้แบบหนึ่ง ซึ่งมีทั้งโครงสร้างข้อมูลในการแทนความรู้และความสามารถในการวินิจฉัยด้วย มันยังเหมาะสำหรับเป็นตัวแทนของแนวคิดการจำแนกประเภทด้วย (classification) และยังเหมาะที่จะเป็นตัวแทนของอนุกรมวิธานหรือการจำแนกหมวดหมู่แบบลำดับขั้นด้วย (taxonomy hierarchy) [12][15] ใน เอนทิตี้ (entity) หนึ่งของเฟรมจะรวบรวมความรู้ที่จำเป็นทั้งหมดเกี่ยวกับวัตถุหรือแนวคิดนั้น

2.2.2.1 โครงสร้างของเฟรม โครงสร้างของเฟรมประกอบไปด้วย ชื่อของเฟรมที่ใช้อ้างถึงเฟรม, สล็อตหรือ แอททริบิวต์ของเฟรม และ ฟาเซท[1][14] เฟรมมักจะมีหลากหลายความหมาย บางทีก็อ้างถึงวัตถุเฉพาะเจาะจง (particular object) และ บางทีก็อ้างถึง กลุ่มของวัตถุที่มีคุณสมบัติเหมือนกัน (group of similar object) เพื่อแยกแยะความแตกต่างนี้ จึงแยกชนิดของ เฟรมออกเป็น คลาสเฟรม และ อินสแตนท์เฟรม

Automobile Frame

Class of: *Wheeled Vehicle*

Manufacturer: *Audi*

Model: *5000 Turbo*

Number of doors: *4* (default)

Number of wheels: *4* (default)

Color: *Red*

Gas mileage: *22 mpg average* (procedural attachment)

รูปที่ 2.3 ตัวอย่างโครงสร้างของเฟรม

คลาสเฟรม : จะใช้เพื่ออธิบายกลุ่มของวัตถุ หรือ คลาสของวัตถุ และสามารถถูกรวบรวมแบบเป็นอนุกรมวิธาน (taxonomy) ได้ด้วย ซึ่ง คลาสเฟรม จะมี สล็อตที่เหมือนกันในทุกๆ คลาสเฟรม คือ สล๊อดแม่และ สล๊อตลูก ยกเว้น รูธคลาส ที่ไม่มี สล๊อดแม่โดยที่ สล็อตอื่นๆ จาก เฟรมแม่สามารถถูกสืบทอดไปยังเฟรมลูกได้ด้วยการ สืบทอด

อินสแตนท์เฟรม : จะใช้ในการอธิบายถึงวัตถุเฉพาะเจาะจง ซึ่งส่วนมากเป็น โหนดสุดท้าย ของ อนุกรมวิธาน และเป็นเฟรมที่ไม่มี สล๊อตลูก อีกแล้ว เป็น เฟรมที่ใช้ในการอ้างถึง วัตถุเฉพาะเจาะจง หรือ อินสแตนท์เฟรม จะมีเพียง สล็อตที่บอกความสัมพันธ์ เป็นสล๊อตแม่เท่านั้น และรับถ่ายทอดคุณสมบัติหรือ สล็อตต่างๆ มาจาก คลาสเฟรม โดยที่อาจจะมีการเปลี่ยนแปลงค่าในสล็อตที่ อินสแตนท์เฟรม ก็ได้ เพราะฉะนั้น ค่าของ สล็อตใน อินสแตนท์เฟรม จึงไม่จำเป็นต้องมีค่าเดียวกันกับ คลาสเฟรม เสมอไป ในอีกนัยหนึ่งอาจกล่าวได้ว่า คลาสเฟรม นั้นจะทำหน้าที่เสมือนเป็นต้นแบบของ อินสแตนท์เฟรม นั่นเอง [1][2][14][15][16]

สล็อต: ในแต่ละเฟรมจะประกอบไปด้วย สล็อตหรือ แอททริบิวต์เพื่อบอกคุณสมบัติของเฟรมนั้นๆ ในเฟรมหนึ่งๆอาจมีได้หลาย สล็อตและบางทีอาจกล่าวได้ว่า แนวคิดของเฟรมนั้นจะถูกกำหนดโดยการรวบรวม สล็อตเข้าด้วยกัน [12] ในแต่ละ สล็อตจะอธิบายถึงคุณสมบัติ หรือการดำเนินการของเฟรม และในแต่ละ สล็อตจะมีค่าของ สล็อตซึ่งเรียกว่า ค่าในสล็อต หรือ ฟิลเลอร์ [15] ซึ่งค่าของ สล็อตนั้นอาจเป็นได้ทั้งค่าโดยปริยาย (default value), เป็นพอยด์เตอร์ ชี้ไปยังเฟรมอื่น, หรือเซตของกระบวนการที่ค่านั้นได้มาก็ได้ [11][14]

ฟาเซท: ในแต่ละ สล็อตจะมี ฟาเซทซึ่งอาจมีมากกว่า หนึ่ง ฟาเซทในหนึ่ง สล็อตเพื่อใช้ควบคุมค่าของ สล็อตและตัวดำเนินการใน สล็อตนั้น ฟาเซทยังสามารถใช้เพื่อกำหนดค่าเริ่มต้นให้กับ ค่าของสล๊อต, กำหนดชนิดของข้อมูล, กำหนดขอบเขตของค่าเป็นไปได้, และกำหนดกิจกรรมที่จะกระทำการต่อไปได้ ชนิดของแต่ละ ฟาเซทมีดังนี้ [14]

Value Type : เป็น ฟาเซทประเภทพื้นฐานที่อธิบายชนิดของค่าใน สล็อตเป็นการกำหนด type restriction ของค่าใน ค่าของสล๊อต ซึ่ง ฟาเซทนี้จะสามารถมีได้แค่ ค่าเดียว โดยจะเป็นสัญลักษณ์เดี่ยวๆหรือ ลิสต์ ของสัญลักษณ์ เช่น integer, string หรือเป็นลิสต์ ของ ค่าที่สามารถเป็นไปได้ เป็นต้น

Default : เป็นค่าที่ถูกนำมาใช้ เมื่อ ค่าของสล๊อต นั้นว่าง หรือไม่มีการกำหนดค่าใดๆให้

Constraint : เป็น ฟาเซทที่กำหนดค่าที่อนุญาตให้มีได้

Minimum cardinality : เป็น ฟาเซทที่กำหนดจำนวนค่าของ ค่าของสล๊อต ว่ามีได้อย่างน้อยสุด N ค่า

Maximum cardinality : เป็น ฟาเซทที่กำหนดจำนวนค่าของ ค่าของสล๊อต ว่ามีได้อย่างมากที่สุด N ค่า

Range : เป็น ฟาเซทที่ทำหน้าที่ระบุขอบเขตของข้อมูลที่จะปรากฏใน ค่าของสล๊อต เช่น เลขจำนวนเต็ม, เลขทศนิยม หรือ 10 ถึง 100 เป็นต้น

If added facet : นี้เป็น ฟาเซทแบบกระบวนการที่ระบุการปฏิบัติที่จะเกิดขึ้น เมื่อมีการเพิ่มค่าเข้ามาในสล็อตบางครั้งเราอาจเรียกว่าเป็น ดีมอน (demons)

If needed : เป็น ฟาเซทประเภทเดียวกับ ถ้าถูกเพิ่มเติม แต่จะทำงานเมื่อไม่มีการระบุค่าให้ค่าของสล๊อต ที่มีความต้องการ เพื่อใช้ประมวลผลขณะนั้น

If removed : เป็นกระบวนการที่จะปฏิบัติ เมื่อค่าภายใน สล็อตถูกลบออก

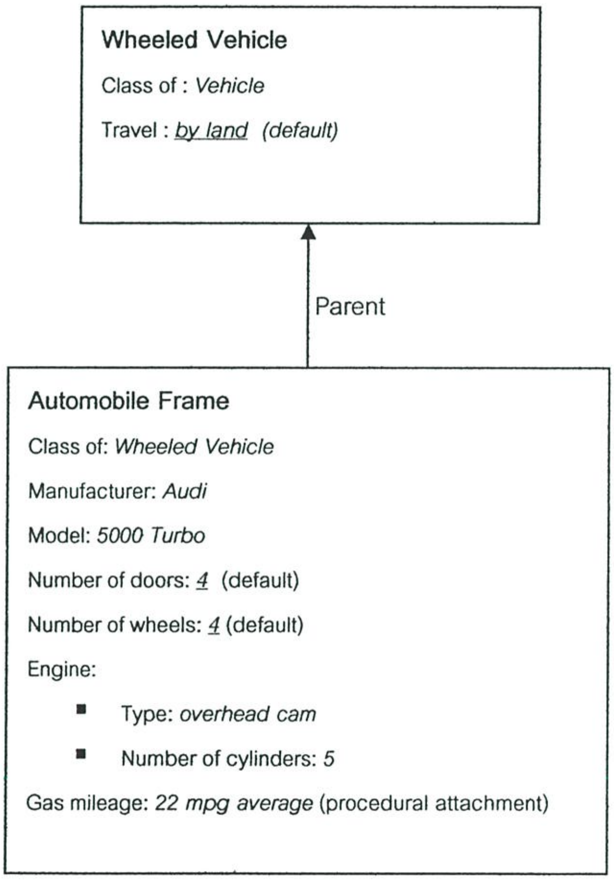
ฟาเซทสามารถมีโครงสร้างเป็นกฎได้ นั่นหมายความว่าเราสามารถกำหนดค่าของสล็อตหรือ ฟาเซทในลักษณะหลายค่า (Multi-Value) ได้ ดังนั้นอาจสรุปได้ว่าค่าของ ฟาเซทนั้น ไม่จำเป็นต้องมีคุณสมบัติอะตอมมิค (Atomic) นั่นเอง

ซึ่งจากที่ได้กล่าวมาข้างต้น สรุปได้ว่า ฟาเซทนั้นเป็น การตรวจสอบค่า อย่างหนึ่งของเฟรม ที่ทำการควบคุมและกำหนด ค่าของสล๊อต หรือ ฟิลเลอร์ ให้เป็นไปตามข้อกำหนด ซึ่งทำหน้าที่เสมือนเป็นกฎที่ใช้ในการจัดการเกี่ยวกับค่าที่เพิ่ม, ลบ, หรือ ปรับปรุง ค่าใน ฟิลเลอร์

นอกจากนี้ เนื่องจาก ฟาเซทเป็นกฎที่ใช้ในการเพิ่มค่าเข้าไปได้ด้วย ทำให้เมื่อมีค่าที่เพิ่มเข้ามาใหม่ใน ฟิลเลอร์ แล้ว ฟาเซทจะมีหน้าที่ไปตรวจสอบค่าใน สล็อตอื่นๆ หรือเมื่อมีการร้องขอดูค่าใน ฟิลเลอร์ ของ สล็อตนั้นๆ ฟาเซทก็สามารถร้องขอค่าของ ฟิลเลอร์ ที่ สล็อตของเฟรมแม่ได้ ดังนั้น ฟาเซทจึงมีความสามารถในการวินิจฉัยได้ด้วย ซึ่งการวินิจฉัยแบบนี้เป็นการวินิจฉัยแบบสืบทอด ดังนั้นจึงกล่าวได้ว่าระบบการแทนความรู้แบบเฟรม จะสามารถทำการวินิจฉัยโดยการสืบทอดโดยอัตโนมัตินั่นเอง

ในรูปที่ 2.4 เป็นตัวอย่างโครงสร้างและความสัมพันธ์ของเฟรม โดยที่ เฟรมรถยนต์ (automobile frame) จะมีความสัมพันธ์กับเฟรมเครื่องยนต์ที่มีล้อ (wheeled vehicle) โดยเฟรมรถยนต์จะเป็นเฟรมลูกของ ของเฟรมเครื่องยนต์ที่มีล้อ เป็นต้น

เฟรมนั้นมีความคล้ายคลึงกันกับวัตถุในภาษาทางด้านโปรแกรมเชิงวัตถุ และ ฐานข้อมูลเชิงวัตถุ แต่ข้อแตกต่างที่สำคัญระหว่าง เฟรม กับภาษาทางด้านวัตถุ คือ มันไม่มีคุณสมบัติการห่อหุ้มวัตถุ (encapsulation) ในภาษาทางด้านวัตถุนั้น วัตถุจะถูกห่อหุ้มรายละเอียดภายในไว้ แต่เฟรมนั้นไม่ เพราะฉะนั้นคุณสมบัติของคลาสของวัตถุจะไม่สามารถถูกมองเห็นได้จากผู้ใช้ จะเห็นเพียงแต่กระบวนการเท่านั้น รูปแบบทางด้านวัตถุที่ไม่มีคุณสมบัติของการห่อหุ้ม นั้นเป็นที่รู้จักกันดีในชื่อของ รูปแบบเชิงวัตถุสัมพันธ์ ซึ่งเข้ากันได้ดียิ่งกับรูปแบบของเฟรม งานวิจัยนี้จึงใช้ระบบการจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์



รูปที่ 2.4 ตัวอย่างโครงสร้างและความสัมพันธ์ของเฟรมรถยนต์

2.2.2.2 การใช้เทคนิคการแทนความรู้แบบเฟรมในงานวิจัยทั่วไป สำหรับเทคนิคการการแทนความรู้แบบเฟรมนี้ เป็นที่ได้รับความสนใจ และนิยมนำไปใช้พัฒนาในงานวิจัยหลายด้าน โดยเฉพาะงานวิจัยที่ต้องการสร้างตัวแทนความรู้สำหรับความรู้ที่มีโครงสร้างซับซ้อน เพราะเป็นที่ยอมรับกันโดยทั่วไปว่า หลักแนวคิดในแบบเฟรม ซึ่งมองรายละเอียดโครงสร้างความรู้ในรูปแบบใกล้เคียงกับวัตถุ จะสามารถรองรับการประมวลผลความรู้ที่ใกล้เคียงกับโลกแห่งความเป็นจริงได้มากขึ้น ถึงแม้ยังจะพบข้อจำกัดบางประการอยู่ [1] แต่ก็ยังมีนักวิจัยที่พยายามพัฒนารูปแบบ

การแทนความรู้แบบเฟรมขึ้นมาอยู่เสมอ ดังจะเห็นได้จากการพยายามนำเทคนิคการแทนความรู้แบบเฟรมนี้มาพัฒนางานทั้งด้านการแพทย์ [1][16], งานวิศวกรรม [14] , การออกแบบพัฒนาซอฟท์แวร์ [4] รวมถึงการพัฒนาระบบผู้เชี่ยวชาญด้านอื่นๆ โดยทั่วไป [2] โดยเครื่องมือการพัฒนาส่วนใหญ่ที่ใช้มักจะอยู่ในรูปของภาษา, เครื่องมือ หรือเปลือกระบบผู้เชี่ยวชาญทียึดมาตรฐานตามภาษาด้านปัญญาประดิษฐ์ในกลุ่มภาษา ลิสพ์ (LISP)หรือ โปรลอก (Prolog) เป็นต้น ซึ่งเครื่องมือที่ได้รับความนิยมส่วนใหญได้แก่ Smalltalk, KEE, Kappa-PC, KL-One, Protégé-2000 เป็นต้น

2.3 กลไกการวินิจฉัยบนระบบผู้เชี่ยวชาญ (Inference Mechanisms)

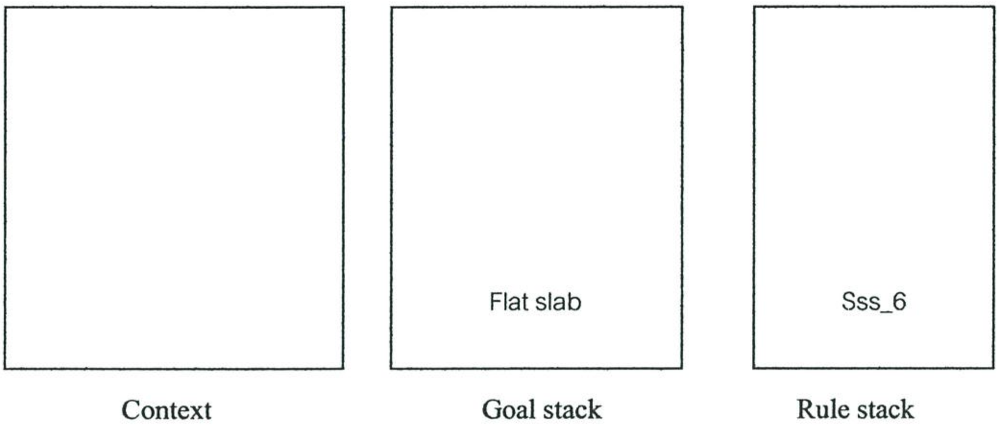
เป็นส่วนควบคุมหรือเทคนิคการค้นหาซึ่งจะค้นหาทั้งฐานความรู้เพื่อการตัดสินใจ ฐานความรู้จะเป็นส่วน state space และกลไกการวินิจฉัยจะเป็นกระบวนการค้นหา กระบวนการวินิจฉัยจะใช้สัญลักษณ์โดยการเลือกกฎที่สอดคล้องกับสัญลักษณ์ของข้อเท็จจริงและรันกฎเพื่อให้ได้ข้อเท็จจริงใหม่ ซึ่งกระบวนการนี้จะกระทำต่อเนื่องเป็นห่วงโซ่จรกว่าจะพบเป้าหมาย ในระบบผู้เชี่ยวชาญนั้นกระบวนการวินิจฉัยสามารถถูกกระทำได้หลายทาง ซึ่งกระบวนการวินิจฉัยที่นิยมใช้กันมากสองวิธีคือ backward chaining และ forward chaining โดยที่ backward chaining คือกระบวนการที่ขับเคลื่อนโดยเป้าหมาย (goal-driven process) ในขณะที่ forward chaining เป็นกระบวนการที่ขับเคลื่อนโดยข้อมูล (data-driven process)

2.3.1 Backward chaining

เป็นกระบวนการที่ขับเคลื่อนโดยเป้าหมาย โดยการตั้งเป้าหมายที่อยู่ในฐานความรู้ขึ้นมาเป็นตัวแปร และกระบวนการวินิจฉัยจะหยุดทำงานเมื่อได้ค่าของตัวแปร โครงสร้างข้อมูลที่ใช้ในระหว่างกระบวนการวินิจฉัยคือส่วนหน่วยความจำปฏิบัติงาน (working memory) หรือ เนื้องาน (context) ส่วน สแต็กของกฎ (rule stack) และส่วนสแต็กของเป้าหมาย (goal stack) ซึ่งเมื่อไหร่ก็ตามที่เกิดการกระทำอย่างหนึ่งในกระบวนการวินิจฉัย เช่น การเลือก(select) การจับคู่(match) และการรันกระบวนการ(execute) เกิดขึ้น ข้อมูลที่ใช้ในส่วนกระบวนการวินิจฉัยเหล่านั้นจะถูกเปลี่ยนแปลง การศึกษาถึงผลกระทบของการกระทำบนโครงสร้างข้อมูลระหว่างกระบวนการวินิจฉัยเป็นทางที่ดีกว่าในการศึกษากระบวนการของกลไกการวินิจฉัย ดังนั้นการทำงานของกระบวนการวินิจฉัยจะถูกอธิบายผ่านการเปลี่ยนแปลงที่เกิดขึ้นกับส่วนเนื้องาน ส่วนสแต็กของกฎและส่วนสแต็กของเป้าหมาย ซึ่งในส่วนสแต็กของกฎ และสแต็กเป้าหมายจะเป็นโครงสร้างข้อมูลชั่วคราวที่สร้างขึ้นเพื่อจัดเก็บ

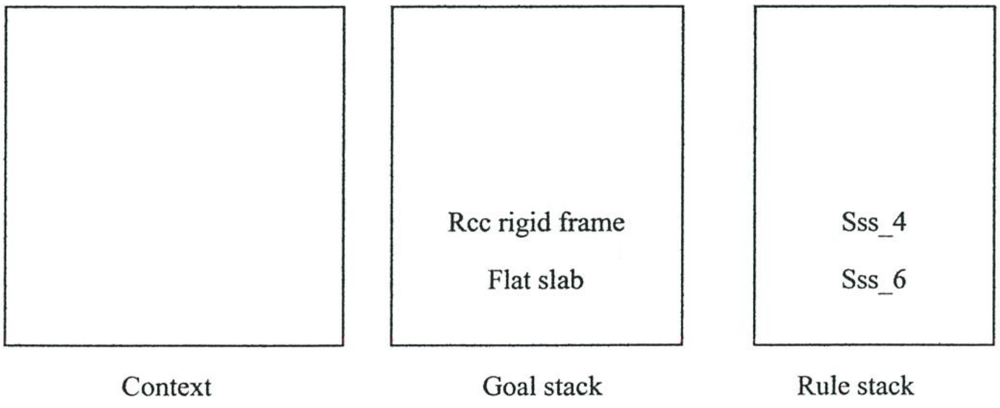
โดยเมื่อเริ่มกระบวนการในส่วนของเนื้องานจะว่างเปล่าและตัวแปรเป้าหมายจะถูกดันเข้าไปในสแต็กเป้าหมาย จากนั้นกระบวนการจะทำการเลือกกฎแรกในฐานกฎซึ่งมีตัวแปรเป้าหมายในส่วน THEN และดันเข้าไปในสแต็กของกฎ ซึ่งตัวอย่างอย่างกฎข้อที่ 6 จะเป็นกฎแรกที่มีตัวแปร

เป้าหมายในส่วนผล ดังนั้นกฎข้อที่ 6 จะถูกดันเข้าไปในสแต็กของกฎและตัวแปรเป้าหมายจะอยู่ในสแต็กเป้าหมาย สถานะของเนื้องาน สแต็กเป้าหมาย และสแต็กของกฎแสดงในรูปที่ 2.5 ดังนี้



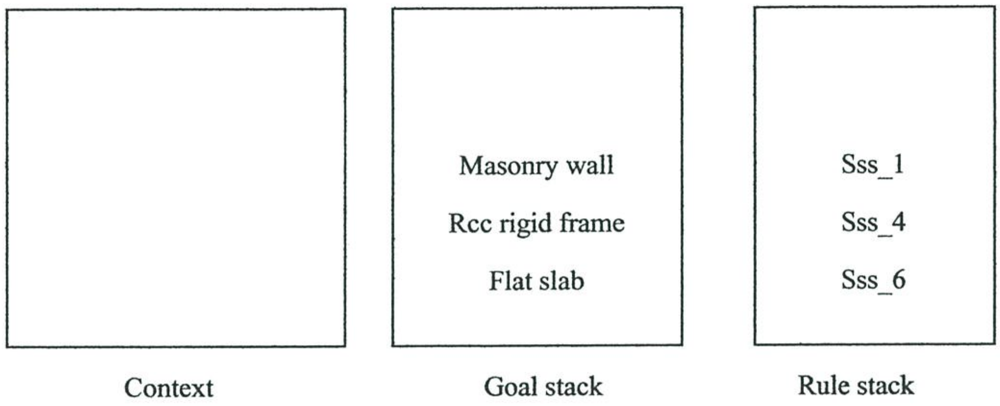
รูปที่ 2.5 สถานะของเนื้องาน สแต็กเป้าหมาย และสแต็กของกฎ I

จะเป็นหากกฎข้อที่ 6 เป็นกฎแรกที่มีตัวแปรเป้าหมายในส่วนของผล ซึ่งเมื่อกฎข้อที่ 6 ถูกเลือกขึ้นมา มันจะเริ่มทำการจับคู่กับเหตุในส่วนที่มันปรากฏขึ้นในกฎซึ่งเงื่อนไขแรกคือ "structural system IS rcc rigid frame" การจับคู่กฎสามารถกระทำต่อไปได้ก็ต่อเมื่อข้อเท็จจริงนี้มันได้ถูกตั้งไว้แล้ว ดังนั้นกระบวนการวินิจฉัยจะเซตตัวแปร "structural system" ให้เป็นเป้าหมายปัจจุบัน ซึ่งมันจะถูกเซตโดยการดันตัวแปรเข้าไปในสแต็กเป้าหมายและยังได้ดันกฎแรกในลิสต์ที่มีตัวแปรเป้าหมายในส่วน THEN เข้าไปในสแต็กของกฎด้วย ดังนั้นกฎข้อที่ 4 จะถูกเลือกสำหรับการหาค่าโดยสถานะของเนื้องานและสแต็กทั้งสองแสดงในรูปที่ 2.6



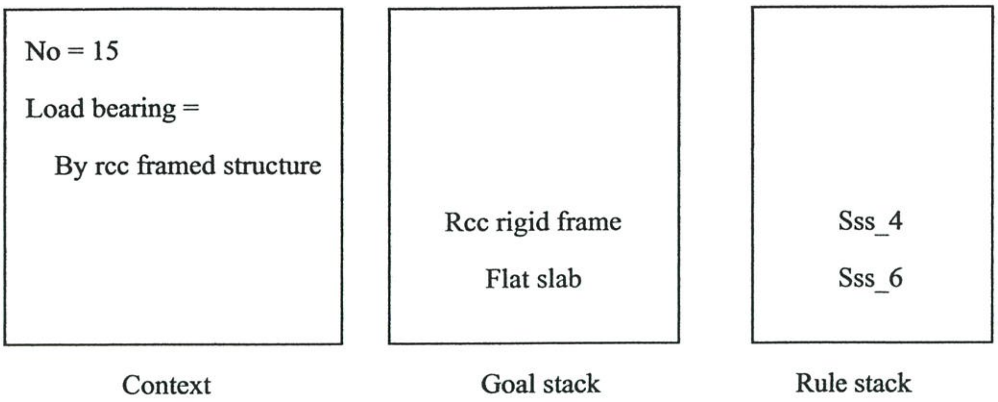
รูปที่ 2.6 สถานะของเนื้องาน สแต็กเป้าหมาย และสแต็กของกฎ II

ตอนนี้กระบวนการวินิจฉัยจะเริ่มตรวจสอบเงื่อนไขแรกของกฎข้อที่ 4 ซึ่งก็คือ "load bearing IS by rcc framed structure" ซึ่งจากผลที่ได้ตัวแปร "load bearing" จะกลายเป็นเป้าหมายปัจจุบันและถูกดันเข้าไปในสแต็กเป้าหมายและกฎข้อที่ 1 จะถูกดันเข้าไปในสแต็กของกฎดังแสดงในรูปที่ 2.7



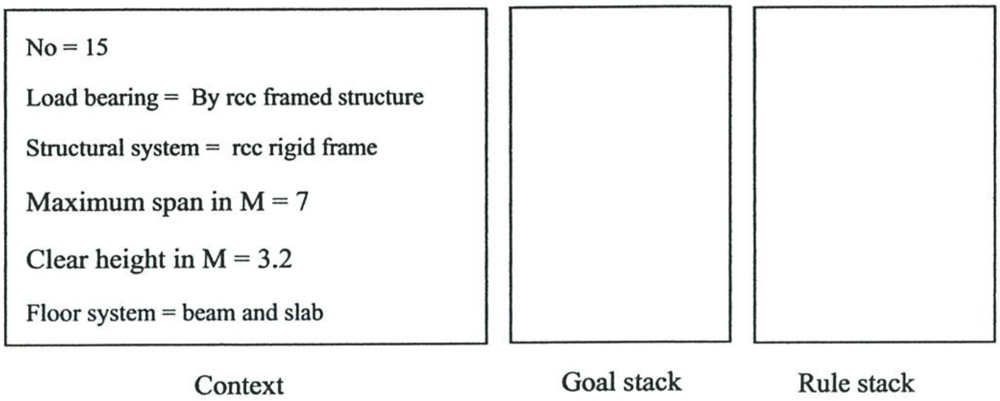
รูปที่ 2.7 สถานะของเนื้องาน สแต็กเป้าหมาย และสแต็กของกฎ III

เงื่อนไขแรกของกฎข้อที่ 1 นั้นคือ "no of stories" ซึ่งร้องขอค่าให้ผู้ใช้ป้อนค่าสำหรับตัวแปรนี้ ซึ่งสมมติให้ผู้ใช้ป้อนค่าเป็น 15 ซึ่งเป็นข้อเท็จจริงแรกและมันจะถูกจัดเก็บอยู่ในเนื้องาน ทำให้เงื่อนไขของกฎข้อที่ 1 คือ "no of stories <= 5" เป็นเท็จ และกฎข้อที่ 1 จะไม่ถูกใช้และถูกปฏิเสธไปโดยการดึงออกจากสแต็กของกฎ กระบวนการวินิจฉัยจะไปยังกฎถัดไปที่มีผลเหมือนกัน (ซึ่งก็คือกฎข้อที่ 2) และดันเข้าไปในสแต็กของกฎและเริ่มการหาค่าในส่วนเงื่อนไขแรกของเหตุ ซึ่งก็เป็นเท็จเหมือนกัน สแต็กของกฎก็จะถูกดึงออกและกฎถัดไปซึ่งมีตัวแปรเป้าหมาย ('load bearing') ในส่วนผลซึ่งก็คือกฎข้อที่ 3 จะถูกดันเข้าไปในสแต็กซึ่งในที่นี้เงื่อนไขในส่วนเหตุจะสอดคล้องกับส่วนเนื้องานและได้ค่าเป็นจริงดังนั้นกฎจะถูกใช้ ซึ่งจะทำให้ได้เป้าหมายปัจจุบันและข้อเท็จจริงที่ได้จะถูกเก็บไว้ในส่วนเนื้องาน ซึ่งเมื่อเป้าหมายปัจจุบันถูกตั้งขึ้นและกฎถูกใช้ไปแล้ว ทั้งสแต็กของกฎและสแต็กเป้าหมายจะถูกดึงออกดังแสดงในรูปที่ 2.8



รูปที่ 2.8 สถานะของเนื้องาน สแต็กเป้าหมาย และสแต็กของกฎ IV

และในขณะนี้เป้าหมายปัจจุบันคือส่วนที่อยู่บนสุดของสแต็กเป้าหมาย ซึ่งก็คือ 'structural system' และกฎปัจจุบันคือกฎข้อที่ 4 ซึ่งเงื่อนไขของกฎเป็นจริงโดยการเปรียบเทียบกับข้อเท็จจริงในส่วนเนื้องานซึ่งจะได้ข้อเท็จจริงใหม่คือ 'structural system IS rcc rigid frame' ได้ถูกเพิ่มเข้าไปในส่วนเนื้องาน และสแต็กทั้งสองจะถูกดึงออกและส่วนเป้าหมายปัจจุบันจะกลายเป็น 'floor system' โดยกฎปัจจุบันก็คือกฎข้อที่ 6 เงื่อนไขแรกของกฎปัจจุบันจะเป็นจริงและเงื่อนไขที่สองจะถูกหยิบขึ้นมา ซึ่งเป็นตัวแปรในสถานะเริ่มต้นของฐานความรู้โดยผู้ใช้ต้องการการป้อนค่าเข้าไป สมมติว่าผู้ใช้ป้อนค่า 7 เข้ามา ซึ่งทำการผลลัพธ์ที่ได้จากเงื่อนไขนี้มีค่าเป็นเท็จและกฎถูกปฏิเสธ และถูกดึงออกจากสแต็ก โดยที่กฎถัดไปซึ่งมีเป้าหมายปัจจุบันในส่วนของผลจะถูกเลือกขึ้นมาแทน ในที่นี้คือกฎข้อที่ 7 ซึ่งจะถูกปฏิเสธด้วยเหมือนกันเนื่องจากเงื่อนไขข้อที่สองเป็นเท็จ ดังนั้นกฎข้อที่ 8 จะถูกเลือกจะใส่เข้าไปในสแต็กกฎ ซึ่งเงื่อนไขสองอันแรกเป็นจริงจะผู้ใช้จะต้องใส่ค่าตัวแปรในเงื่อนไขที่สาม โดยให้ผู้ใช้ป้อนค่าเข้ามาเป็น 3.2 ซึ่งจะได้ค่าเป็นจริงและกฎจะถูกใช้ซึ่งจะได้ผลลัพธ์เป็นค่า 'beam and slab' สำหรับตัวแปรเป้าหมาย ซึ่งเมื่อกฎถูกใช้ทั้งสองสแต็กจะถูกดึงออกจะข้อเท็จจริงใหม่จะถูกเพิ่มเข้ามายังเนื้องาน ซึ่งตอนนี้ทั้งสองสแต็กมันว่างเปล่าและกระบวนการวินิจฉัยก็สิ้นสุดลงโดยสถานะของเนื้องานของตัวอย่างนี้แสดงในรูปที่ 2.9



รูปที่ 2.9 สถานะของเนื้องาน สแต็กเป้าหมาย และสแต็กของกฎ V

ในส่วนการอธิบายการดำเนินงานของระบบผู้เชี่ยวชาญจะมีกลไกสำหรับการสอบถามเนื้องานในการได้มาซึ่งค่าของตัวแปร (what?) และรู้ว่าข้อเท็จจริงเกิดขึ้นได้อย่างไร (how?) คำตอบสำหรับคำถามแรกจะได้มาจากส่วนเนื้องานโดยตรง ในขณะที่คำตอบที่ได้จากคำถามที่สองจะต้องการการอ้างอิงถึงฐานความรู้ โดยกระบวนการที่เกี่ยวข้องกับคำถามว่าเกิดขึ้นได้อย่างไรจะค้นหากฎซึ่งต้องการข้อเท็จจริงในส่วน THEN และแสดงกฎซึ่งแสดงให้เห็นว่าข้อเท็จจริงเกิดขึ้นได้อย่างไร การได้มาซึ่งคำตอบของคำถามแล้วถ้า (what if?) จะมีประโยชน์อย่างมากถ้าผู้ใช้ของระบบต้องการรู้กระบวนการตัดสินใจโดยการเปลี่ยนค่าที่ป้อนเข้าไป ซึ่งสามารถทำได้ในสองขั้นตอน ขั้นแรกข้อเท็จจริงเหล่านั้นที่ขึ้นอยู่กับการเปลี่ยนแปลงตัวแปรต้องถูกยกเลิกก่อน จากนั้นมันต้องถูกกำหนดใหม่โดยกระบวนการวินิจฉัยจากจุดนั้น ซึ่งจะค่อนข้างซับซ้อนและต้องมีการรักษาเส้นทางการวินิจฉัยที่เสร็จสิ้นแล้วด้วยขึ้นอยู่กับข้อเท็จจริงที่อยู่ในเนื้องาน กระบวนการวินิจฉัยจะสร้างเครือข่ายที่ขึ้นกันทั้งหมดจาก TMS (Truth Maintenace System) ซึ่งเครือข่ายนี้จะช่วยในการย้อนกลับ (backtrack) จากขั้นตอนใดๆไปยังขั้นตอนก่อนหน้าโดยยกเลิกการกระทำที่เกี่ยวข้อง ซึ่งจะได้ผลลัพธ์ในการลบข้อเท็จจริงต่างๆออกไปจากส่วนเนื้องาน ซึ่งการย้อนกลับนี้จะกระทำต่อไปเรื่อยๆจนกว่าจะถึงส่วนก่อตั้งตัวแปรข้อเท็จจริง โดยค่าของมันจะถูกเปลี่ยนแปลงด้วยคำถามแล้วถ้า (what if?) กระบวนการวินิจฉัยจะเริ่มใหม่จากจุดนั้นจะต่อเนื่องไปเรื่อยๆจนกว่าเป้าหมายสุดท้ายจะเกิดขึ้น และเพราะว่าความยากที่เกิดขึ้นในการบำรุงรักษาเครือข่ายที่ขึ้นต่อกันของการตัดสินใจดังนั้นเครื่องมือที่ใช้ในเชิงพาณิชย์จึงไม่มีส่วนนี้

2.3.2 Forward chaining

เป็นกระบวนการวินิจฉัยโดยขับเคลื่อนข้อมูล ผู้ใช้ของระบบต้องให้ข้อมูลทั้งหมดก่อนที่จะเริ่มกระบวนการวินิจฉัยซึ่งกลไกการวินิจฉัยจะพยายามที่จะตั้งข้อเท็จจริงที่อยู่ในฐานความรู้จนกว่าเป้าหมายจะถูกค้นพบ พิจารณาฐานกฎเดียวกัน ผู้ใช้จะให้ข้อมูลและสถานะของเนื้องานก่อนที่จะเริ่มกระบวนการวินิจฉัยดังแสดงในรูปที่ 2.10

No of storie = 15
Maximum span in M = 7
Clear height in M = 3.2

Context

รูปที่ 2.10 สถานะของเนื้องานก่อนที่จะเริ่มกระบวนการวินิจฉัย

กระบวนการวินิจฉัยจะเลือกกฎแรกในฐานกฎขึ้นมาและปฏิเสธมันถ้าเงื่อนไขมันเป็นเท็จ จากนั้นมันจะไปยังกฎที่สองซึ่งก็ถูกปฏิเสธเช่นกัน แต่ส่วนเงื่อนไขในกฎข้อที่สามนั้นเป็นจริงและกฎจะถูกใช้ได้ผลเป็นข้อเท็จจริงใหม่ที่ถูกเพิ่มเข้ามาในส่วนเนื้องาน กฎข้อที่ 4 ก็ถูกใช้แต่กฎข้อที่ 5, 6, 7 จะถูกปฏิเสธและสุดท้ายกฎข้อที่ 8 จะถูกใช้เพื่อให้ได้เป้าหมายและสิ้นสุดกระบวนการวินิจฉัย ในฐานกฎขนาดใหญ่จะมีการทำซ้ำก่อนที่จะเกิดเป้าหมาย ลำดับของกฎซึ่งอยู่ในฐานกฎจะทำหน้าที่สำคัญในแนวทางการวินิจฉัยในกระบวนการนี้ ในขณะที่ลำดับจะไม่มีบทบาทใน backward chaining แต่ลำดับที่เงื่อนไขถูกลิสต์อยู่ในกฎนั้นสำคัญใน backward chaining ลำดับคำถามจะถามผู้ใช้เพื่อรอบรับลำดับเหล่านี้ ดังนั้นก่อนที่จะสร้างฐานกฎ วิศวกรความรู้ควรตัดสินว่าจะใช้ backward chaining หรือ forward chaining ในการวินิจฉัย

วิธีการวินิจฉัยแบบที่สามคือกลไกการวินิจฉัยที่ผสมกันระหว่างกระบวนการ backward chaining และ forward chaining ซึ่งกระบวนการแบบ backward นั้นจะเหมาะสมถ้ามีเป้าหมายน้อยแต่มีสถานะเริ่มต้นเยอะ ลำดับของข้อมูลที่ถูกร้องขอขึ้นกับการไหลของกระบวนการวินิจฉัย ส่วนในกระบวนการแบบ forward นั้นทุกๆข้อมูลที่ผู้ใช้รู้ต้องถูกป้อนเข้ามาในระบบก่อนและระบบไม่ต้องรอให้ผู้ใช้ป้อนเข้ามาในระบบอีก ซึ่งถ้าข้อมูลมันเพียงพอก็จะบรรลุเป้าหมาย หรือไม่ก็จะหยุดโดยบอกว่าไม่สามารถบรรลุเป้าหมายได้ด้วยข้อมูลที่มีอยู่ ดังนั้นกระบวนการแบบ forward จะเหมาะสมเมื่อมีสถานะเริ่มต้นน้อยๆแต่มีได้หลากหลายเป้าหมายเท่านั้น ซึ่งระบบผู้เชี่ยวชาญขนาดใหญ่ จะมีได้หลากหลายสถานเริ่มต้น และหลายเป้าหมาย ดังนั้นผู้ใช้อาจไม่สามารถให้ข้อมูลที่จำเป็นทั้งหมดก่อนได้ และเขาอาจไม่สามารถมองออกว่าการไหลของกระบวนการวินิจฉัยเป็นอย่างไร ซึ่งลักษณะนี้กลไกการวินิจฉัยแบบผสมอาจเหมาะสม โดยมันจะเริ่มจากการทำงานแบบ forward ก่อนเพื่อให้ได้ข้อเท็จจริงบางส่วนจากนั้นจะใช้การทำงานแบบ backward เพื่อพิสูจน์มันซึ่งกระบวนการแบบผสมคือ forward ซึ่ง backward จะถูกเรียกขึ้นมาเมื่อข้อเท็จจริงถูกตั้งขึ้น ผู้ใช้จะไม่ต้องป้อนข้อมูลทั้งหมดตั้งแต่เริ่มต้น แต่จะป้อนข้อมูลส่วนหนึ่งเข้ามาก่อนแล้วป้อนอีกครั้งเมื่อมีการร้องขอ

ถ้าเฟรมถูกใช้กับกฎในการแทนความรู้ ข้อเท็จจริงที่เกี่ยวข้องกับตัวแปรที่ถูกในเฟรมจะถูกเก็บในเนื้องานแยกออกมาต่างหากเรียกว่าฐานเฟรม ฐานเฟรมสามารถมีได้หลายเฟรมอยู่ภายในซึ่งชนิดของสล๊อตมีได้ทั้ง concrete หรือ abstract และเฟรมยังสามารถถูกเชื่อมโยงโดยใช้ความสัมพันธ์อย่างมีความหมายซึ่งสามารถสร้างเครือข่ายความหมายสำหรับแทนความรู้แบบพรรณา แต่โดยทั่วๆ ไปนั้น chaining จะถูกกระทำบนตัวแปรที่อยู่ในฐานกฎและไม่อยู่ในตัวแปรเฟรม ดังนั้นทุกๆตัวแปรกที่กำหนดในกระบวนการวินิจฉัยจะถูกกำหนดเป็นตัวแปรฐานกฎทั้งหมด

2.3.3 การวินิจฉัยโดยใช้ความรู้ในการแทนความรู้แบบเฟรม

ในกรณีที่มีชนิดของการวินิจฉัยที่แตกต่างกันสามารถทำได้ด้วยการแทนความรู้แบบเฟรม การวินิจฉัยเท่านั้นจะไม่มีรูปแบบควบคุมวิธีการทั้งหมดในการแก้ปัญหาอย่างในกรณีการวินิจฉัยบนฐานกฎ แต่มันจะแสดงลักษณะที่ฉลาดหลายๆอย่างที่มนุษย์ใช้ในกระบวนการของการแก้ปัญหา ลักษณะอย่างหนึ่งคือสามัญสำนึก ดังตัวอย่างต่อไปนี้ซึ่งแสดงการจำลองสามัญสำนึกโดยใช้เฟรม

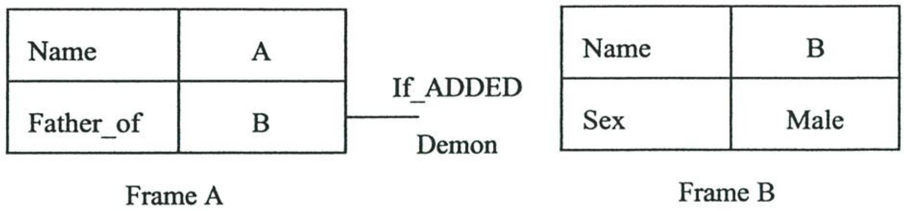
พิจารณาสถานการณ์ซึ่งมีบางคนบอกเราว่า A เป็นพ่อของ B และ B เป็นผู้ชาย จากนั้นอ้างถึงลูกชายของ A ซึ่งจะทำให้เราคิดได้ทันทีว่า B เป็นลูกชายของ A และไม่มีข้อกำหนดว่า B เป็นลูกชายของ A แต่ด้วยสามัญสำนึก (โดยมีพื้นฐานบนความรู้ที่เรามีเรื่องความสัมพันธ์ระหว่างพ่อแม่และลูก) เราจะสามารถสร้างข้อกำหนดได้ว่า B เป็นลูกชายของ A และในสถานการณ์ที่คล้ายคลึงกัน ซึ่งก็คือการวินิจฉัยซึ่งมีพื้นฐานมาจากสามัญสำนึกสามารถถูกจำลองสถานการณ์ในโปรแกรมคอมพิวเตอร์โดยใช้เทคนิคซึ่งเกี่ยวข้องกับเฟรมและการจัดการเฟรมและทำได้อย่างไรดังนี้

ลักษณะพื้นฐานสองประการของระบบการจัดการเฟรมคือเฟรมจะถูกสร้างระหว่างเวลารัน (run time) และสล๊อตกับค่าของมันจะถูกเพิ่มเข้าไประหว่างกระบวนการวินิจฉัย

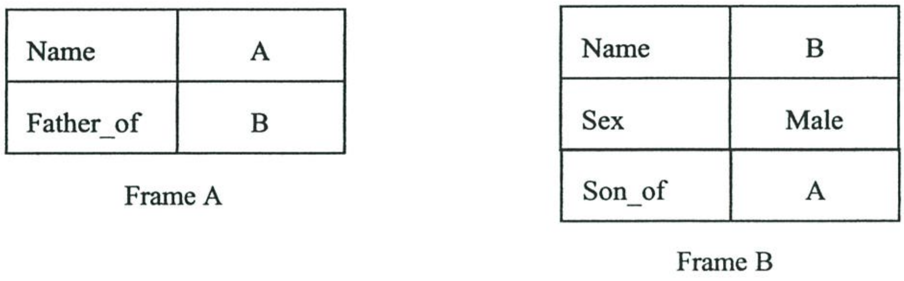
สมมติให้ A และ B เป็นเฟรมสองเฟรมซึ่งมีสองสล๊อตคือ name และ father_of ซึ่งสำหรับ A และมีสล๊อต name และ sex สำหรับเฟรม B โดยใช้สล๊อต name ถูกใส่ค่าเข้าไปเป็น A และ B ตามลำดับและสล๊อต sex ของเฟรม B ถูกใส่ค่าเข้าไปเป็น male โดยการระบุข้อกำหนด A เป็นพ่อของ B จะสำเร็จด้วยการกำหนดให้ใส่ค่า B ในสล๊อต father of ในเฟรม A ซึ่งเดมอน (กระบวนการทำงาน) ชนิด if_added จะถูกผูกติดอยู่กับสล๊อต father_of ของเฟรม A ซึ่งเดมอนนี้จะถูกกระตุ้นขึ้นมาเมื่อค่าถูกใส่เข้าไปในสล๊อต รูปที่ 2.11 จะแสดงสถานะของเฟรมก่อนถูกกระตุ้นด้วยเดมอน

เมื่อค่าถูกใส่เข้าไปในสล๊อต father_of ของเฟรม A เดมอนที่ผูกติดอยู่กับสล๊อตจะถูกปล่อยออกมาอย่างอัตโนมัติซึ่งจะเพิ่มสล๊อตที่ชื่อว่า son_of ให้กับเฟรม B และใส่ค่าของสล๊อตเป็น A ให้กับมัน ความรู้ในเรื่องความสัมพันธ์จะถูกห่อหุ้มไว้ในส่วนเดมอนที่ผูกติดอยู่กับสล๊อต status

ของเฟรมหลังจากการอ้างของเดมอนที่ผูกติดอยู่กับสล๊อต father_of ของเฟรม A ดังแสดงในรูปที่ 2.12



รูปที่ 2.11 สถานะของเฟรม A และ B ก่อนที่จะถูกกระตุ้นด้วยเดมอน



รูปที่ 2.12 สถานะของเฟรม A และ B หลังจากถูกกระตุ้นด้วยเดมอน

วิศวกรความรู้ผู้ออกแบบฐานความรู้จะผูกเดมอนเข้ากับสล๊อตซึ่งจะถูกกระตุ้นโดยอัตโนมัติเมื่อมีเกิดเหตุการณ์ในการใส่ค่าเข้าไปในสล๊อต ซึ่งแบบจำลองการวินิจฉัยนี้มาจากสามัญสำนึก และผลกระทบเดียวกันสามารถถูกกระทำได้โดยการผูกเดมอนที่ชื่อว่า if_needed กับสล๊อต son_of ของเฟรม B ซึ่งเดมอนนี้จะถูกปลุกขึ้นมาถ้ามีการอ้างถึงสล๊อต son_of ของเฟรม B ซึ่งข้อได้เปรียบของวิธีการนี้คือว่าค่าจะถูกวินิจฉัยและเพิ่มเข้าไปในสล๊อตก็ต่อเมื่อมีการอ้างถึงสล๊อตนั้น เดมอนอีกชนิดหนึ่งคือ if_deleted ซึ่งเดมอนชนิดนี้จะถูกปลุกขึ้นมาถ้าสล๊อตถูกลบออกไปจากเฟรม ซึ่งเดมอนนี้จะมีประโยชน์อย่างมากในปัญหาวิศวกรรมหลายๆปัญหา ซึ่งเนื้องานจะถูกอัพเดตระหว่างกระบวนการแก้ปัญหาที่มาจากการเปลี่ยนแปลงสมมติฐานหรือมาจากค่าที่ถูกคำนวนขึ้นมา ชนิดของการวินิจฉัยที่หลากหลายนี้สามารถถูกจำลองโดยแนวคิดของเดมอน

ระบบผู้เชี่ยวชาญที่ใช้วิธีการแทนความรู้แบบเฟรมส่วนใหญ่นั้น มักจะใช้เทคนิคการสืบทอด (inheritance) ในการวินิจฉัย และแก้ปัญหาบนพื้นฐานของความรู้ที่จัดเก็บไว้ในฐานความรู้ โดยลักษณะการทำงานของกลไกการวินิจฉัยนั้น มีทั้งรูปแบบในการใช้กระบวนการเป็นหลัก ซึ่งเป็นการจัดเก็บกระบวนการต่างๆ ไว้ที่ ค่าของฟาเซท ใน แต่ละสล็อต และจะมีตัวแปลภาษา (Interpreter) คอย กระทำ กระบวนการ เหล่านั้นอีกครั้ง หรืออาจใช้ กระบวนการ (method) ที่ถูก ผูกติด (attached) ไว้ใน ฟาเซทของ สล็อตเป็นตัวดำเนินการเองก็ได้

ในงานวิจัยนี้การวินิจฉัยระบบฐานความรู้นั้นจะมีการวินิจฉัย คือ การใช้ การสืบทอดกับอีกวิธีหนึ่งคือการ ใช้การติดต่อสื่อสารระหว่าง เฟรมผ่านทางการใช้ กระบวนการที่ผูกติดอยู่กับสล๊อต ซึ่งส่วนใหญ่ที่ใช้อยู่คือกระบวนการ IF-NEEDED และ IF-CHANGED หรือก็คือการใช้ การส่งผ่านข้อความ (message passing) ระหว่างอินสแตนท์เฟรม [2][14]

2.4 หลักการการโปรแกรมเชิงวัตถุ (Object-oriented Programming)

การเขียนโปรแกรมเชิงวัตถุ (Object-oriented programming, OOP) คือหนึ่งในรูปแบบการเขียนโปรแกรมคอมพิวเตอร์ ที่ให้ความสำคัญกับ วัตถุ ซึ่งสามารถนำมาประกอบกันและนำมาทำงานรวมกันได้โดยการแลกเปลี่ยนข่าวสารเพื่อนำมาประมวลผลและส่งข่าวสารที่ได้ไปให้ วัตถุอื่นๆที่เกี่ยวข้องเพื่อให้ทำงานต่อไป

แนวคิดการเขียนโปรแกรมแบบดั้งเดิมมักนิยมใช้การเขียนโปรแกรมเชิงกระบวนการ (Procedural Programming) ซึ่งให้ความสำคัญกับขั้นตอนกระบวนการที่ทำ โดยแบ่งโปรแกรมออกเป็นส่วนๆตามลำดับขั้นตอนการทำงาน แต่แนวคิดการเขียนโปรแกรมเชิงวัตถุนั้นให้ความสำคัญกับ ข้อมูล(data) และ พฤติกรรม(behavior) ของวัตถุ และความสัมพันธ์กันระหว่างวัตถุกันมากกว่า

2.4.1 แนวคิดที่สำคัญของการเขียนโปรแกรมเชิงวัตถุ

- *คลาส (Class)* - ประเภทของวัตถุ เป็นการกำหนดว่าวัตถุจะประกอบไปด้วยข้อมูล(data) หรือคุณสมบัติ(property) และพฤติกรรม(behavior)หรือการกระทำ(method) อะไรบ้าง ซึ่งคลาสเป็นโครงสร้างพื้นฐานของการเขียนโปรแกรมเชิงวัตถุ
- *วัตถุ (Object)* - โดยมากจะเรียกว่า อ็อบเจกต์ คือ ตัวตน(instance) ของ คลาส(เช่น นายทักษิณ, นายสนธิ) ซึ่งจะเกิดขึ้นระหว่าง run-time โดยแต่ละอ็อบเจกต์ จะมีข้อมูลเฉพาะของตัวเอง ทำให้อ็อบเจกต์แต่ละอ็อบเจกต์ของคลาสซึ่งใช้ source code เดียวกันมีคุณลักษณะและคุณสมบัติที่แตกต่างกัน
- *การห่อหุ้ม (Encapsulation)* - การปิดบังข้อมูล เป็นวิธีการกำหนดสิทธิในการเข้าถึงข้อมูล หรือการกระทำกับ อ็อบเจกต์ ของ คลาสนั้นๆ ทำให้แน่ใจได้ว่าข้อมูลของอ็อบเจกต์นั้นจะถูกเปลี่ยนแปลงแก้ไขผ่านทาง methods หรือ properties ที่อนุญาตเท่านั้น (เช่น การกำหนดตำแหน่งทางการเมือง เป็น public method ที่ผู้อื่นสามารถกระทำได้ ส่วนการลาออกจากตำแหน่ง เป็น private method ที่มีแต่ อ็อบเจกต์ ของ คลาส เท่านั้นที่จะสามารถทำได้ แต่การกดดันและการขับไล่สามารถสร้าง data ที่อาจจะส่งผลเกิดการลาออกได้เช่นกัน)

• *การสืบทอด (Inheritance)* - การสืบทอดคุณสมบัติ เป็นวิธีการสร้าง คลาสย่อย ที่เรียกว่า ซับคลาส (subclass) ซึ่งจะเป็นกำหนดประเภทของวัตถุให้จำเพาะเจาะจงขึ้น ซึ่ง ซับคลาส จะได้รับถ่ายทอดคุณสมบัติต่างๆมาจากคลาสหลักด้วย (เช่น คลาส มนุษย์ สืบทอดมากจาก คลาส สิ่งมีชีวิต)

• *นามธรรม (Abstraction)* - นามธรรม เป็นการแสดงถึงคุณลักษณะและพฤติกรรมของ object เท่าที่จำเป็นต้องรับรู้และใช้งาน โดยซ่อนส่วนที่เหลือเอาไว้เพื่อไม่ให้เกิดความสับสน เช่น ตามปกติแล้ว นายทักษิณ จัดเป็นตัวตนของ คลาส มนุษย์ ซึ่งจะมีพฤติกรรม การกระทำทุกอย่างที่ตามที่กำหนดไว้ตามโครงสร้างของ คลาส มนุษย์ แต่ในบางกรณีที่นำไปใช้งาน เราไม่ต้องการให้เกิดการสับสนต่อการใช้งานหรือการจัดประเภทมากเราสามารถจัดการหรือใช้งาน อ็อบเจกต์ นายทักษิณ ให้อยู่ในรูปของสิ่งมีชีวิต ก็ได้

• *ภาวะที่มีหลายรูปแบบ (Polymorphism)* - ภาวะที่มีหลายรูปแบบ เป็นวิธีการกำหนดรูปแบบการกระทำที่เหมือนกันแต่ได้ผลที่แตกต่างการ เช่น การเปล่งเสียง เป็น method หลักของ คลาส สิ่งมีชีวิต ซึ่งมีคลาส มนุษย์ และคลาสสุนัข เป็น ซับคลาส แต่ผลของการเปล่งเสียงของอ็อบเจกต์จากคลาสทั้งสองจะออกมาไม่เหมือนกัน

2.5 เปรียบเทียบหลักการเชิงวัตถุกับเฟรม

ในปัจจุบันนี้ภาษาทางด้านปัญญาประดิษฐ์ได้มีส่วนขยายทางด้านวัตถุซึ่งสามารถใช้เป็นเครื่องมือทางด้านปัญญาประดิษฐ์ได้ และระบบเฟรมก็มีแนวโน้มที่จะมีคุณลักษณะของการโปรแกรมเชิงวัตถุมากขึ้น ในหัวข้อนี้จะทำการเปรียบเทียบโครงสร้างของเฟรมกับหลักการเชิงวัตถุ ซึ่งข้อแตกต่างระหว่างเฟรมกับวัตถุก็คือ

รูปประโยค (Syntax):

เฟรมนั้นโดยทั่วไปแล้วจะเขียนง่ายกว่าวัตถุ เพราะว่ารูปประโยคของวัตถุจะขึ้นอยู่กับระบบที่ใช้ แต่รูปประโยคของเฟรมจะขึ้นอยู่กับภาษามากกว่าระบบ

การสืบทอด (inheritance):

ในเฟรมล้วนๆนั้นการสืบทอดจะไม่ตายตัว (dynamic) เช่น ค่าของสล๊อตในเฟรมแม่จะไม่ถูกแพร่ลงมา มันคือกลไกการสืบทอดซึ่งจะค้นหาผ่านเส้นทางการสืบทอดแต่ละครั้งที่ต้องการค่าในสล๊อต แต่วัตถุนั้นโดยทั่วไปแล้วอินสแตนท์จะถูกใส่ค่าให้เรียบร้อยแล้วเมื่อมันถูกสร้างขึ้นมาเพื่อเพิ่มประสิทธิภาพ อย่างไรก็ตามเทคนิคผสมได้ถูกใช้ในระบบเฟรมเพื่อหลีกเลี่ยงการเสียเวลาของการสืบทอดแบบไม่ตายตัวแต่ต้องไม่ลืมว่าในระบบเฟรมโดยปกตินั้นจะมีลำดับขั้นตามการเฉพาะเจาะจง เช่น เราจะมีการสืบทอดระหว่าง "วัตถุ" ซึ่งจะเฉพาะเจาะจงด้วยโครงสร้างของมัน แต่

ในระบบเชิงวัตถุนั้นส่วนมากจะอ้างถึงลำดับขั้นตามพฤติกรรมที่คล้ายคลึงกัน เช่น ถ้าเราต้องการที่จะเพิ่มคลาสใหม่เข้ามา เราจะเพิ่มมันเข้าไปกับวัตถุซึ่งมีพฤติกรรมที่คล้ายคลึงกัน (ที่มันกว้างกว่า) แต่ไม่จำเป็นต้องมีโครงสร้างคล้ายคลึงกัน

การห่อหุ้ม (encapsulation):

ในภาษาเชิงวัตถุส่วนใหญ่นั้น ข้อมูลและการควบคุมจะถูกห่อหุ่มไว้ในวัตถุเป็นเสมือนกล่องดำซึ่งมีการคอมไพล์ข้อมูล การเปลี่ยนแปลง และการหาเหตุผลรวมอยู่ในนั้น แต่ในระบบเฟรมจะไม่มีการห่อหุ้ม

แต่ว่ามันก็จริงเพียงส่วนหนึ่งเท่านั้น เพราะระบบเฟรมจะถูกใช้ด้วยการโปรแกรมเชิงการเข้าถึง เช่น เราจะเข้าถึงสล๊อตและกระบวนการอาจจะไม่ถูกใช้ก็ได้ (มันถูกซ่อนไว้) วัตถุจะถูกเข้าถึงด้วยกระบวนการและสล๊อตจะถูกซ่อนไว้ เพราะฉะนั้นเราจะเข้าถึงโครงสร้างของเฟรมและถามถึงพฤติกรรมของวัตถุ

จากที่ได้กล่าวมาแล้วถึงข้อแตกต่างระหว่างทั้งสองหลักการ ซึ่งข้อแตกต่างอยู่ที่จุดประสงค์ของมันซึ่งหลักการเชิงวัตถุจะเป็นเครื่องมือในส่วน softward engineering ในขณะที่เฟรมจะเป็นเครื่องมือทางด้านปัญญาประดิษฐ์สำหรับการแทนความรู้ การเปรียบเทียบเฟรมกับหลักการเชิงวัตถุ เฟรมนั้นจะมีกระบวนการดังนี้

- การเข้าถึงข้อมูลในระดับคลาส : ซึ่งชื่อของสล๊อตและขอบเขต (domain) เป็นหัวใจสำคัญ
- การป้อนค่าของสล๊อตแบบไดนามิก
- การโปรแกรมด้วยการขับเคลื่อนเหตุการณ์ (even-driven programming) โดยใช้เดมอน
- รองรับสล๊อตที่มีค่าเป็นเซต ซึ่งจะถูกใช้ในการแทนความสัมพันธ์ หรือคาดินาลลิตี้มากกว่า 1
- การสร้างคลาสแบบไดนามิก

ทั้งเฟรมและหลักการเชิงวัตถุจะเป็นโครงสร้างแบบลำดับขั้นซึ่งมีการสืบทอดและสล๊อตเดมอน ดังนั้นสล๊อตของเฟรมจะมีเดมอน (และไม่มีกระบวนการ) และหลักการเชิงวัตถุจะมีกระบวนการ (แต่ไม่มีสล๊อตและไม่มีเดมอน)

สรุปเฟรมเป็นแนวคิดทางด้านปรัชญาของการแทนความรู้และการจัดการโดยมนุษย์ ซึ่งเฟรมเป็นแนวคิดโครงร่างซึ่งต้องมีการเติมเต็มด้วยรายละเอียดในการอธิบายวัตถุเฉพาะในภายหลัง ในขณะที่ หลักการเชิงวัตถุเป็นแนวคิดทางการโปรแกรมซึ่งเป็นสัญลักษณ์ของการจัดการโดยโปรแกรม ซึ่งวัตถุจะอ้างถึงโปรแกรมของคลาสซึ่งถูกควบคุมโดยพฤติกรรมที่ได้อธิบายไว้ก่อนแล้วโดยคลาส

ในหลักการเชิงวัตถุนั้นส่วนมากจะสร้างรูปแบบที่เรียกว่า is-a hierarchy ซึ่งการสืบทอดจะเป็นการกระจายวิธีการ(method) และข้อมูลโครงสร้างของวัตถุซึ่งส่วนมากแล้วการกระจายการสืบทอดจะอยู่ในเวลาการกำหนด (defenetion time) และเมื่อโปรแกรมถูกสร้างด้วยหลักการเชิงวัตถุถูกรันขึ้นมาการสืบทอดจะไม่ตายตัว

ส่วนในระบบเฟรมนั้นการสืบทอดจะถูกใช้ในการขยายค่าโดยปริยายของสล๊อต (default slot value) โดยทั่วๆไปการสืบทอดจะอยู่ในเวลารัน (run time) ซึ่งจะอนุญาตให้ค่าโดยปริยายสามารถถูกเปลี่ยนแปลงได้ในระหว่างการปฏิบัติการ นอกจากนี้ระบบเฟรมอาจยังอนุญาตการใช้เส้นทางการสืบทอดมากกว่าจะเป็นการใช้ is-a hierarchy

ข้อแตกต่างในการสืบทอดจะถูกเชื่อมโยงกับความแตกต่างในการเข้าถึงของโปรแกรมในการสืบทอดชนิดด้วยซึ่งในหลักการเชิงวัตถุนั้นอินสแตนซ์เท่านั้นที่จะเป็นวัตถุของระบบแต่ในระบบเฟรมนั้นชนิดของเฟรมจะไม่แตกต่างกับอินสแตนซ์ของมันนักดังนั้นจึงสามารถที่จะเข้าถึงชนิดของเฟรมได้โดยตรง

หลักการเชิงวัตถุถูกออกแบบสำหรับการโปรแกรมและลักษณะเชิงวัตถุจะถูกใช้ในการโปรแกรมให้มีประสิทธิภาพ ซึ่งแตกต่างกับในระบบเฟรมที่ถูกออกแบบสำหรับแทนความรู้โดยในส่วนการโปรแกรมนั้นไม่ถูกเน้นให้มีความสำคัญมากนัก

2.6 ระบบการจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์

ก่อนที่จะได้พูดถึงพื้นฐานสถาปัตยกรรมระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ จำเป็นจะต้องทำการเปรียบเทียบระบบฐานข้อมูลเชิงสัมพันธ์ กับระบบฐานข้อมูลเชิงวัตถุ เพื่อให้เห็นความแตกต่างของเทคโนโลยีทั้ง 2 แบบ และเป็นพื้นฐานในการศึกษาระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ ที่เกิดจากการผสมผสานทั้ง 2 เทคโนโลยีเข้าด้วยกัน โดยการเปรียบเทียบจะแบ่งออกเป็น 3 ส่วนคือ โครงสร้างข้อมูล, ภาษาที่ใช้ในการจัดการฐานข้อมูล และกลไกที่ใช้ในการเข้าถึงข้อมูล เพื่อให้เข้าใจถึงหลักการของระบบฐานข้อมูลทั้ง 2 แบบ

2.6.1 ฐานข้อมูลเชิงสัมพันธ์ (Relational Database)

ในการศึกษาฐานข้อมูลเชิงสัมพันธ์นี้จะสามารถแบ่งการศึกษาออกเป็น 3 ส่วนคือ ส่วนโครงสร้างข้อมูล, ส่วนภาษาที่ใช้ในการจัดการฐานข้อมูล และกลไกที่ใช้ในการเข้าถึงข้อมูล

2.6.1.1 โครงสร้างข้อมูล ถ้ากล่าวโดยสรุปแล้ว โครงสร้างข้อมูลของระบบฐานข้อมูลเชิงสัมพันธ์จะจัดเก็บข้อมูลเป็นเซตของรีเลชั่น (Relation) ซึ่ง รีเลชั่น จะประกอบไปด้วยเซตของทูเพิล (Tuple) ซึ่งมีแอตทริบิวต์แบบเดียวกัน และในส่วนชนิดข้อมูลในแต่ละแอตทริบิวต์

จะต้องเหมือนกัน และแต่ละแอตทริบิวต์ จะมีได้เพียงหนึ่งชนิดเท่านั้น และข้อมูลที่ถูกจัดเก็บในแต่ละแอตทริบิวต์ จะมีค่าได้เพียงค่าเดียว (คุณสมบัติ Atomic) [18]

2.6.1.2 ภาษาที่ใช้ในการจัดการฐานข้อมูล ระบบฐานข้อมูลเชิงสัมพันธ์ใช้ภาษา SQL (Structured Query Language) สำหรับการสร้าง (create), เปลี่ยนแปลง (modify), ค้นคืน (retrieve) และ เข้าถึง (manipulate) ข้อมูลในระบบฐานข้อมูลเชิงสัมพันธ์ ภาษา SQL ได้ถูกกำหนดให้เป็นภาษามาตรฐานโดย ANSI (American National Standards Institute) และ ISO (International Organization for Standardization) [19]

2.6.1.3 กลไกที่ใช้ในการเข้าถึงข้อมูล การดำเนินการทั้งหมดในการเข้าถึงข้อมูลจะใช้ค่าของข้อมูลที่ถูกจัดเก็บไว้ใน แต่ละ ทูเพิล โดยที่ไม่สามารถสนับสนุนการอ้างอิงจาก ทูเพิล หนึ่งไปยังอีก ทูเพิล หนึ่งได้ และผลการ คิวรี จะอนุญาตให้เกิดการปฏิบัติการต่างๆ กับข้อมูลได้ครั้งละ 1 ทูเพิล เท่านั้น ซึ่งกลไกนี้จะใช้ทั้งในการทำ คิวรี หรือการ เปลี่ยนแปลงแก้ไขข้อมูล

2.6.2 ฐานข้อมูลเชิงวัตถุ (Object-Oriented Database)

ถึงแม้ว่าฐานข้อมูลเชิงวัตถุนี้จะยังไม่มีมาตรฐานอย่างเป็นทางการ แต่ในทางปฏิบัติแล้ว บริษัทผู้พัฒนาซอฟท์แวร์ระบบฐานข้อมูลเชิงวัตถุได้อ้างอิงเอามาตรฐานที่ Morgan Kaufmann ได้ตีพิมพ์ไว้ ซึ่งการจัดทำมาตรฐานนี้ได้รับการสนับสนุนจาก Object Database Management Group (ODMG) มาตรฐานดังกล่าวที่ใช้คือ ODMG-V2.0 [20] โดยที่เนื้อหาของมาตรฐานจะเน้นหนักไปในเรื่องของความสัมพันธ์ระหว่างวัตถุ ที่จะสามารถสนับสนุนการเขียนโปรแกรมด้วยภาษาเชิงวัตถุได้ รวมถึงการจัดเก็บวัตถุต่างๆ ในฐานข้อมูล

2.6.2.1 โครงสร้างข้อมูล การใช้งานข้อมูลในฐานข้อมูลเชิงวัตถุจะอยู่ในรูปแบบของ คลาส ซึ่งเป็นแนวความคิดพื้นฐานของทฤษฎีทางวัตถุโดยที่ในแต่ละ คลาส จะประกอบด้วย แอททริบิวต์ (attribute) และกระบวนการ (method) รวมไปถึงเงื่อนไขความถูกต้อง (Integrity Constrain) ของ คลาส นั้นๆ ซึ่งจะมีการกำหนด Object Identifier (OID) เอาไว้ เพื่อแบ่งแยกความแตกต่างระหว่าง คลาส นอกจากนี้ยังสนับสนุนการทำงานแบบ ห่อหุ้มข้อมูล (encapsulation) และ สืบทอด (inheritance) ด้วย สำหรับชนิดของข้อมูลนั้นสามารถรองรับชนิดข้อมูลแบบ Abstract Data Type ได้ จากมาตรฐานของ ODMG-V2.0 [20]

เนื่องจากฐานข้อมูลเชิงวัตถุนี้เกิดขึ้นจากการรวมเอาพื้นฐานแนวความคิดเชิงวัตถุ กับหลักการด้านภาษาสำหรับการโปรแกรมเชิงวัตถุ และความสามารถด้านฐานข้อมูลเข้าด้วยกัน ผลลัพธ์ที่ได้คือทำให้เกิดความเข้ากันได้เป็นอย่างดีระหว่างโครงสร้างข้อมูลภายในฐานข้อมูล กับโครงสร้างข้อมูลที่ใช้ในการพัฒนาโปรแกรม ซึ่งอำนวยความสะดวกในการใช้ภาษาสำหรับการ

โปรแกรมเชิงวัตถุต่างๆ เพื่อพัฒนาโปรแกรมที่ใช้งานร่วมกับฐานข้อมูลเชิงวัตถุ ทำให้สามารถลดจำนวนโค้ดที่ใช้ในการเขียนโปรแกรมต่างๆลงได้

2.6.2.2 ภาษาที่ใช้ในการจัดการฐานข้อมูล ระบบฐานข้อมูลเชิงวัตถุจะใช้ภาษาสำหรับการโปรแกรมเชิงวัตถุ ทั้งในการจัดการฐานข้อมูล และการพัฒนาโปรแกรม (เช่น C++, SmallTalk, Java) เนื่อง วัตถุในแอบพริเคชั่น (Application Object) กับ วัตถุที่ถูกจัดเก็บ (Stored Object) ในระบบฐานข้อมูลชนิดนี้มีความสัมพันธ์กันโดยตรง ดังนั้นการจัดการกับฐานข้อมูล และการคิวรี จึงสามารถทำได้ด้วยภาษาสำหรับการโปรแกรมเชิงวัตถุดังกล่าว นอกจากนั้นแล้ว ตามมาตรฐาน ODMG-93 ได้มีการกำหนดภาษา ขึ้นมาอีกภาษาหนึ่งคือ OQL (Object Query Language) อย่างไรก็ดีในมาตรฐานใหม่ของ SQL 3 จะได้มีบางส่วนที่จะได้เพิ่มความสามารถเหล่านี้ลงไป ซึ่งผลจากการพัฒนา OQL จะสามารถสร้างความหลากหลายของผลลัพธ์ที่เกิดขึ้นจากการทำคิวรีได้ อันได้แก่ ผลลัพธ์ที่อยู่ในรูปของ อะตอม, โครงสร้าง, วัตถุ หรือแม้แต่ เซตของวัตถุ [20]

2.6.2.3 กลไกที่ใช้ในการเข้าถึงข้อมูล การสร้าง และ แก้ไขข้อมูลในระบบฐานข้อมูลเชิงวัตถุจะใช้การเข้าถึงโดยตรงจากภาษาสำหรับการโปรแกรมเชิงวัตถุในลักษณะของภาษาหลัก (Native Language) สำหรับวัตถุ ที่ถูกสร้างขึ้นในระบบฐานข้อมูลเชิงวัตถุนี้จะถูกกำหนด OID ให้โดยอัตโนมัติ และ OID นี้จะมีความเฉพาะ (Unique) ไปตลอดอายุการใช้งานวัตถุ นั้นๆ นอกจากนี้ในตัว วัตถุแต่ละ วัตถุยังสามารถจัดเก็บ OID ของ วัตถุตัวอื่น เพื่อสร้างการอ้างอิงแบบลอจิก (Logical Reference) ซึ่งตัวอ้างอิงนี้เอง จะเป็นประโยชน์ในการสร้างความสัมพันธ์ระหว่างวัตถุที่เกิดขึ้นตามโลกแห่งความเป็นจริงที่สามารถแบ่งตามกฎเกณฑ์ได้

จะเห็นได้จากการเปรียบเทียบดังกล่าวว่า ระบบฐานข้อมูลทั้ง 2 แบบข้างต้น มีความแตกต่างกันอย่างมาก ทั้งทางโครงสร้างข้อมูล ที่เกิดจากแนวความคิดที่ต่างกัน รวมถึงภาษาที่ใช้ในการจัดการฐานข้อมูล และกลไกที่ใช้ในการเข้าถึงข้อมูล ที่ทำให้เกิดผลลัพธ์ในการใช้งานข้อมูลที่ต่างกัน

2.6.3 ฐานข้อมูลเชิงวัตถุสัมพันธ์ (Object-Relational Database)

เป็นที่ทราบกันดีอยู่แล้วว่าการพัฒนาโปรแกรมในยุคปัจจุบัน มีแนวโน้มไปทางการโปรแกรมเชิงวัตถุมากขึ้น ทำให้เกิดความต้องการในการใช้ข้อมูลที่มีความซับซ้อนมากขึ้นตามไปด้วย ซึ่งระบบฐานข้อมูลเชิงสัมพันธ์แบบเดิมไม่สามารถรองรับได้เต็มที่ จึงได้เกิดแนวคิดที่จะมีการเพิ่มความสามารถในการจัดการข้อมูลในรูปแบบวัตถุให้แก่ระบบฐานข้อมูลเชิงสัมพันธ์ ระบบฐานข้อมูลใหม่นี้เรียกว่า ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ [19][21] ซึ่งได้รับความสนใจเป็นอย่างมาก จนทำให้เกิดแนวคิดและวิธีการที่หลากหลายในการพัฒนาระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ขึ้น [21] โดยเฉพาะเมื่อ Michael Stonebraker ได้เขียนหนังสือชื่อ Object-Relational DBMSs, The Next

Great Wave. ที่ถูกตีพิมพ์โดยสำนักพิมพ์ Morgan Kaufman และได้ร่วมกับทีมวิจัยของตนพัฒนาระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ตัวอย่างขึ้นชื่อ Postgres (ซึ่งก็คือ Informix ในปัจจุบัน) [21] จนทำให้บริษัทผู้ผลิตซอฟท์แวร์ระบบการจัดการฐานข้อมูลอื่นๆ เช่น IBM, Unisys, Oracle และ UniSQL สนใจที่จะนำแนวความคิดดังกล่าวมาพัฒนาซอฟท์แวร์ระบบการจัดการฐานข้อมูลของตน โดยแนวทางในการพัฒนาระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ หรือการเพิ่มความสามารถในการจัดการข้อมูลแบบ วัตถุให้แก่ระบบฐานข้อมูลเชิงสัมพันธ์

ต่อมาในภายหลัง ANSI จึงได้เริ่มดำเนินการเพื่อเพิ่มมาตรฐานด้านเชิงวัตถุ ให้แก่มาตรฐาน SQL ที่มีอยู่ โดยการดำเนินการดังกล่าวอยู่ในความรับผิดชอบของทีมพัฒนาชุด X3H2 ของ ANSI ซึ่งได้ใช้วิธีการเพิ่มส่วนขยายเชิงวัตถุ (Object Extension) ให้แก่ระบบฐานข้อมูลเชิงสัมพันธ์โดยตรงเป็นแนวทางในการพัฒนามาตรฐานไปสู่ SQL 3 (SQL-1999) [19]

2.6.3.1 มาตรฐาน SQL 3 เพื่อให้สามารถเปรียบเทียบความแตกต่างของระบบฐานข้อมูล เชิงวัตถุสัมพันธ์ กับระบบฐานข้อมูลเชิงสัมพันธ์ และระบบฐานข้อมูลเชิงวัตถุ จึงได้สรุปคุณสมบัติของระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ (ตามมาตรฐาน SQL 3) ออกเป็น 3 ส่วน ดังนี้

2.6.3.1.1 โครงสร้างข้อมูล

ดังที่กล่าวมาแล้วว่าการพัฒนามาตรฐาน SQL 3 เพื่อรองรับการใช้งานข้อมูลประเภทวัตถุของระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ ใช้วิธีการเพิ่ม ส่วนขยายเชิงวัตถุ ให้แก่ระบบฐานข้อมูลเชิงสัมพันธ์ ซึ่งการดำเนินการในส่วนนี้ถือเป็นหัวใจของมาตรฐาน SQL 3 โดยการเพิ่ม ส่วนขยายเชิงวัตถุดังกล่าวคือ การเพิ่มชนิดข้อมูลแบบใหม่ให้แก่ระบบฐานข้อมูลเชิงสัมพันธ์ โดยที่โครงสร้างการจัดเก็บข้อมูลยังถูกแสดงอยู่ในรูปแบบของตารางเช่นเดิม ชนิดข้อมูลแบบใหม่นี้ถูกเรียกว่า Abstract Data Type (ADT) ซึ่งจะมีลักษณะคล้าย คลาส ในหลักการโปรแกรมเชิงวัตถุ กล่าวคือ ADT จะถูกใช้เพื่อรวมกลุ่มของข้อมูลที่มีความสัมพันธ์กันทั้งในส่วนของคุณสมบัติของข้อมูล และสถานการณ์ที่จะเกิดขึ้นกับข้อมูลเหล่านั้น โดยที่ชนิดข้อมูลแบบ ADT นี้ จะสนับสนุนคุณสมบัติ การห้อหุ้ม หรือการมีความสัมพันธ์กับ ADT กลุ่มอื่นในรูปแบบของ Suptype-Supertype ทั้งยังมีคุณสมบัติสืบทอดในการถ่ายทอดคุณสมบัติของ แอททริบิวต์มาจาก ADT ที่เป็น Supertype ของตนด้วย นอกจากนี้แล้วยังสามารถกำหนด การดำเนินการ (Operation) และ คำสั่ง (Function) ที่มีความสัมพันธ์กับข้อมูล เพื่อใช้ใน การสร้างและการใช้งานอินเดกซ์, จับเก็บและค้นคืนข้อมูล ตามคุณลักษณะพิเศษของข้อมูลแต่ละประเภทได้ เช่น ข้อมูลที่เป็นมัลติมีเดียต่างๆ ที่จะต้องมีวิธีการ ค้นคืน ที่แตกต่างออกไปจากปกติเป็นต้น ซึ่ง การดำเนินการ และ คำสั่ง ของ ADT เหล่านี้สามารถนำมาใช้ ร่วมกับการค้นหา (Search Predicate) ในการทำคิวรีข้อมูลได้

สำหรับชนิดข้อมูล ที่ ADT สนับสนุนมีทั้ง Row Type (หรือชนิดของแถวที่อยู่ในตาราง), Distinct Type (หรือการสร้าง User-Defined Type จากชนิดที่ต่างกัน ที่ระบบมีอยู่) และ การสร้างต้นแบบชนิดข้อมูล (Type Template) ขณะเดียวกันยังมีการเตรียมเครื่องมือภายในระบบเอาไว้สนับสนุนการสร้างชนิดข้อมูลในรูปแบบ Collection Type จำพวก LIST หรือ SET อีกด้วย ซึ่งจะเห็นได้ว่า ADT จะสามารถรองรับการใช้งานข้อมูลที่มีโครงสร้างซับซ้อนมากให้ดีกว่าเดิมได้ [19]

2.6.3.1.2 ภาษาที่ใช้ในการจัดการข้อมูล

เพื่อสนับสนุนการใช้งานข้อมูล ADT ในระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ จึงได้มีการเพิ่มเติมความสามารถในการเข้าถึงข้อมูลดังกล่าวให้แก่ภาษา SQL ประเด็นสำคัญของส่วนที่เพิ่มเติมใหม่นี้คือการสนับสนุนการใช้งานข้อมูลในรูปแบบ วัตถุที่รวมการจัดการข้อมูลแบบ Nested Object, set-valued Attribute, ความสามารถในการใช้ การดำเนินการ หรือคำสั่ง ของ วัตถุร่วมกับการค้นหาในการคิวรีข้อมูลแบบ ADT ได้ โดยยังยึดเอา SQL เป็นภาษาหลักในการทำจัดการฐานข้อมุลและการทำคิวรี สำหรับผลของการทำคิวรี นั้น ข้อมูลจะยังถูกแสดงในรูปแบบของตารางอยู่ อย่างไรก็ดีในอนาคตได้มีการวางแผนที่จะพัฒนามาตรฐานใหม่เป็น SQL 4 ที่สามารถแสดงผลการทำคิวรี ให้อยู่ในรูปของ วัตถุด้วยการผสมผสานเทคนิคของ OQL เข้ามา หรือการใช้เทคนิคที่ใกล้เคียง

นอกจากนี้มาตรฐาน SQL 3 ยังมีส่วนของการเพิ่ม Procedural Extension (SQL/PSM) ซึ่งการดำเนินการดังกล่าวได้ครอบคลุมไปถึงการพัฒนารูปแบบคำสั่ง SQL ที่ใช้ในการทำ Stored Procedure และ User-Defined Function เพื่อใช้งานร่วมกับฐานข้อมูลด้วย กล่าวคือ กระบวนการและ คำสั่ง เหล่านี้จะสามารถสร้างขึ้นได้จากภาษา SQL เอง ซึ่งมีการพัฒนารูปแบบคำสั่งใหม่ที่ใกล้เคียงกับภาษาสำหรับการโปรแกรมทั่วไปในยุคปัจจุบัน ได้แก่ความสามารถในการใช้คำสั่งเงื่อนไขประเภท IF/THEN/ELSE, การใช้ลูป แบบ WHILE และ DO/UNTIL หรือการใช้ CASE เป็นต้น ทั้งยังให้ โปรแกรมเมอร์ สามารถสร้างฟังก์ชั่นไลบราลี ด้วยภาษาอื่น เพื่อใช้งานร่วมกับภาษา SQL ที่มีอยู่ได้ ซึ่งจะเพิ่มขีดความสามารถในการทำคิวรีได้ดียิ่งขึ้น

อีกส่วนหนึ่งที่มีความสำคัญในมาตรฐาน SQL 3 ก็คือส่วนขยายเชิงสัมพันธ์ (relational extension) ซึ่งจะทำให้โปรแกรมเมอร์สามารถสร้าง Recursive Query, สนับสนุน Query Expression ที่สามารถใช้งานร่วมกันได้ รวมไปถึงการพัฒนามุมมอง (View) ให้ดีขึ้นจากมาตรฐานเดิม ซึ่งจะสามารถสนับสนุนระบบการทำงาน ในส่วนของ Trigger และ Integrity Constrain ได้

เนื่องจากภาษาหลักที่ใช้ในการจัดการฐานข้อมูลยังคงเป็น SQL และระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ได้ถูกพัฒนาขึ้นมาจากระบบฐานข้อมูลเชิงสัมพันธ์ ดังนั้นกลไกในการเข้าถึงข้อมูลส่วนใหญ่จะยังคงเป็นเช่นเดียวกับระบบฐานข้อมูลเชิงสัมพันธ์อยู่ ส่วนการเพิ่มความสามารถในการ

เข้าถึงข้อมูลประเภท ADT หรือการเข้าถึงข้อมูลในรูปแบบ วัตถุนั้นยังมีปัญหาเมื่อต้องการเข้าถึงโดยตรงจากภาษาการโปรแกรมเชิงวัตถุ ทำให้ยังจำเป็นที่จะต้องมีการแปลงรูปให้กลับมาอยู่ในลักษณะตารางเสียก่อน

2.7 ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ในเชิงพาณิชย์

ปัจจุบันมีบริษัทผู้ผลิตซอฟท์แวร์ระบบการจัดการฐานข้อมูลจำนวนมากที่หันมาให้ความสนใจกับระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ โดยเฉพาะเมื่อบริษัทที่เป็นผู้นำในกลุ่มได้เริ่มพัฒนาผลิตภัณฑ์ของตน เพื่อการรองรับการใช้งานข้อมูลที่เป็น วัตถุตามกระแสความนิยมของแนวคิดเชิงวัตถุมากขึ้น ในขณะที่มาตรฐาน SQL 3 ยังไม่เสร็จสมบูรณ์ ทำให้การดำเนินการพัฒนาระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ของแต่ละบริษัทจะมความแตกต่างกันออกไป ซึ่งบางทีอาจทำให้เกิดความสับสนในรูปแบบของเทคโนโลยีของแต่ละผลิตภัณฑ์ ดังนั้นงานวิจัยนี้จึงได้ยกตัวอย่างผลิตภัณฑ์ซอฟท์แวร์ระบบการจัดการฐานข้อมูล ในกลุ่มที่เป็นระบบฐานข้อมูลเชิงวัตถุสัมพันธ์เพื่อมาศึกษาเปรียบเทียบระหว่างผลิตภัณฑ์ต่างๆ เหล่านั้น กับมาตรฐาน SQL 3 ที่กำหนดขึ้น โดยผลิตภัณฑ์ที่เลือกมา คือออราเคิล ซึ่งถือว่าเป็นซอฟท์แวร์ระบบการจัดการฐานข้อมูล ที่ได้รับความนิยม และการยอมรับอย่างกว้างขวาง ในท้องตลาดผลิตภัณฑ์ซอฟท์แวร์

2.7.1 ผลิตภัณฑ์ของออราเคิล

ออราเคิลเป็นอีกผลิตภัณฑ์หนึ่งที่มีการพัฒนาระบบการจัดการฐานข้อมูลของตนเพื่อรองรับการขยายตัวในการใช้งานที่ซับซ้อนขึ้น [7][21] โดยเฉพาะข้อมูลแบบ วัตถุซึ่ง ออราเคิลสนับสนุนการทำงานดังกล่าว โดยการเพิ่มชนิดข้อมูลแบบ วัตถุเข้ามาในระบบฐานข้อมูล ซึ่ง ข้อมูลแบบวัตถุ (Object Type) นี้จะใช้ในการกำหนดชนิดของข้อมูลแบบใหม่ที่แตกต่างไปจากเดิมที่มีอยู่รวมทั้งสามารถสร้างกระบวนการ สำหรับการทำงานกับ วัตถุต่างๆ ที่สร้างขึ้นได้ โดย Collection ของ วัตถุจะสามารถแสดงอยู่ได้ทั้งในรูปของโครงสร้างแบบ Array หรือ Nested Table สำหรับข้อมูลแบบวัตถุ ของออราเคิล จะสามารถเทียบได้กับ Row Type ของมาตรฐาน SQL 3 กล่าวคือข้อมูลแบบวัตถุนั้นสามารถเป็นได้ทั้งระดับ Column Type และ Table Type โดยที่ออราเคิล ยังคงใช้ภาษา PL/SQL เป็นที่มีการพัฒนาขึ้นตามมาตรฐาน SQL 3 เป็นภาษาหลักในการจัดการ และ คิวรีข้อมูลอยู่เช่นเดิม

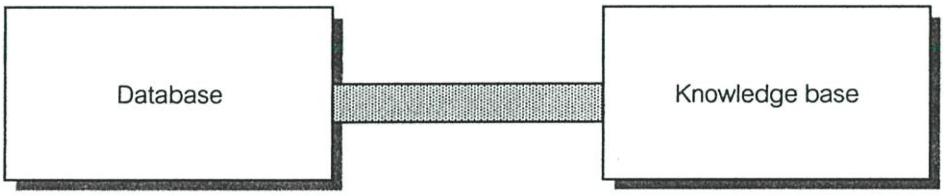
2.8 การบูรณาการระบบฐานความรู้กับระบบฐานข้อมูล

จากที่ได้กล่าวถึงโครงสร้าง และเทคนิคในการแทนความรู้แบบเฟรม ตลอดจนข้อขัดข้อง และข้อจำกัดต่างๆ ในการพัฒนาระบบผู้เชี่ยวชาญที่จำเป็นต้องใช้เครื่องมือเฉพาะ ด้วยสาเหตุนี้เองทำให้มีนักวิจัยจำนวนหนึ่ง ต้องการที่จะลดข้อจำกัด และจัดการกับปัญหาที่เกิดขึ้นเหล่านี้ โดยความพยายามนำระบบฐานข้อมูลเข้ามาช่วยเหลือ ซึ่งจะได้กล่าวถึงแนวการบูรณาการระบบฐานความรู้กับระบบฐานข้อมูล เพื่อช่วยเพิ่มความสามารถของระบบฐานความรู้สำหรับระบบผู้เชี่ยวชาญ และจะได้ยกตัวอย่างงานวิจัยที่เกี่ยวข้องบางส่วนมาอธิบายพอสังเขปดังนี้ [22]

2.8.1 สถาปัตยกรรมการบูรณาการระบบฐานความรู้กับระบบฐานข้อมูล

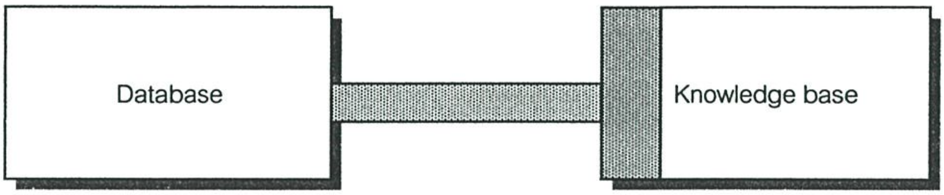
ในการประยุกต์ใช้ระบบฐานข้อมูลร่วมกับระบบฐานความรู้ สำหรับระบบผู้เชี่ยวชาญนั้น มีแนวทางการดำเนินการอยู่หลายลักษณะ บางระบบก็เป็นเพียงการใช้ฐานข้อมูลอย่างเรียบง่ายเพื่อเก็บความรู้แทนการใช้ไฟล์ข้อมูลปกติเท่านั้น ซึ่งจากแนวทางต่างๆ สามารถจำแนกประเภทสถาปัตยกรรมในการพัฒนาระบบฐานความรู้ด้วยการใช้ระบบฐานข้อมูลได้ 5 รูปแบบดังนี้

2.8.1.1 การสร้างการเชื่อมต่อระหว่างฐานข้อมูล และฐานความรู้ ในรูปแบบนี้นักพัฒนาระบบจะสร้างระบบฐานข้อมูลเพื่อเก็บข้อมูลเบื้องต้น ที่ใช้ในการดำเนินระบบงานทั่วไปไว้ต่างหาก และสร้างการเชื่อมต่อในการดึงข้อมูลจากระบบฐานข้อมูลไปสร้างระบบฐานความรู้อีกส่วนหนึ่ง เพื่อใช้สำหรับการวินิจฉัยระบบผู้เชี่ยวชาญโดยเฉพาะ สำหรับการเชื่อมต่อนั้นอาจเป็นการสร้างฟังก์ชันพิเศษขึ้นเอง หรือการใช้เทคโนโลยีจำพวก ODBC ก็ได้ ในบางครั้งเราอาจเรียกวิธีนี้ว่าการเชื่อมต่อแบบหลวมๆ (Loose Coupling) ดังจะเห็นตามลักษณะดังรูปที่ 2.13



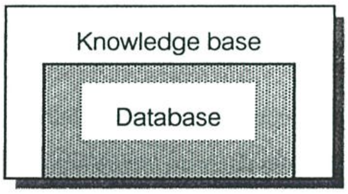
รูปที่ 2.13 เทคนิคแบบการสร้างการเชื่อมต่อ

2.8.1.2 สร้างส่วนเชื่อมประสานภายในฐานความรู้ไปยังฐานข้อมูล ในลักษณะการสร้างส่วนเชื่อมประสานภายในฐานความรู้ไปยังฐานข้อมูลนั้น จะเป็นการสร้างฟังก์ชันการเชื่อมประสานไว้ภายในระบบฐานความรู้ มีหน้าที่คอยเข้าถึงข้อมูลที่ต้องการภายในระบบฐานข้อมูลที่ใช้ ดังรูปที่ 2.14



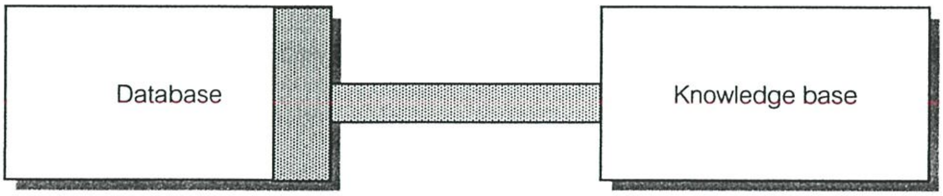
รูปที่ 2.14 เทคนิคแบบการสร้างส่วนเชื่อมประสานภายในฐานความรู้ไปยังฐานข้อมูล

2.8.1.3 สร้างฟังก์ชันงานฐานข้อมูลภายในฐานความรู้ วิธีแบบการสร้างฟังก์ชันงานฐานข้อมูลไว้ภายในฐานความรู้ดังที่เห็นตามแผนภาพในรูปที่ 2.15 เป็นวิธีที่มีความซับซ้อนในการสร้าง เนื่องจากฟังก์ชันงานในการบริหารจัดการข้อมูลในรูปแบบฐานข้อมูลนั้นมีความซับซ้อน ทำให้การใช้ระบบฐานความรู้มาสร้างฟังก์ชันดังกล่าวจะเกิดความซับซ้อนมากขึ้นไปอีก เพราะปกติระบบฐานความรู้ จะมิได้ให้ความสำคัญกับวิธีการจัดเก็บ และบริหารจัดการข้อมูลมากนัก



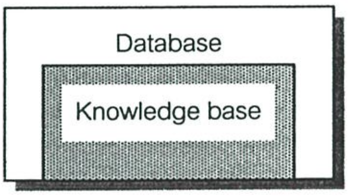
รูปที่ 2.15 เทคนิคแบบการสร้างฟังก์ชันงานฐานข้อมูลภายในฐานความรู้

2.8.1.4 สร้างส่วนเชื่อมประสานภายในฐานข้อมูลไปยังฐานความรู้ วิธีนี้เป็นวิธีที่คล้ายกับข้อ 2.2.1.2 แต่ต่างกันตรงที่ฟังก์ชันการเชื่อมประสานข้อมูลจะอยู่ฝั่งฐานข้อมูล ดังรูปที่ 2.16



รูปที่ 2.16 เทคนิคแบบการสร้างสร้างส่วนเชื่อมประสานภายในฐานข้อมูลไปยังฐานความรู้

2.8.1.5 สร้างฟังก์ชันงานฐานความรู้ภายในฐานข้อมูล เป็นเทคนิคที่ได้รับความนิยมอีกรูปแบบหนึ่ง ด้วยการใช้ระบบฐานข้อมูลที่จัดเก็บข้อมูลที่ต้องการอยู่แล้ว และสร้างฟังก์ชันงาน หรือทำการสร้างตัวแทนความรู้แบบที่ต้องการลงไป โดยอาศัยความสามารถในด้านการจัดการ และบริหารข้อมูลของระบบฐานข้อมูลเป็นหลัก วิธีนี้นอกจากจะมีความสะดวก เพียงการกำหนดการสร้างตัวแทนความรู้ที่เหมาะสมบนฐานข้อมูลแล้ว ยังสะดวกในเรื่องการจัดใช้งานเครื่องมือ เนื่องจากระบบฐานข้อมูลมีความสามารถในการดำเนินการ ทั้งการจัดเก็บ, การค้นคืน และการเข้าถึงข้อมูลอยู่แล้ว ดังแสดงให้เห็นในรูปที่ 2.17



รูปที่ 2.17 เทคนิคแบบการสร้างฟังก์ชันงานฐานความรู้ภายในฐานข้อมูล

2.8.2 งานวิจัยที่ใช้การบูรณาการระบบฐานความรู้กับระบบฐานข้อมูล

โดยส่วนมากแล้วในการพัฒนาฐานความรู้โดยการบูรณาการระบบฐานความรู้กับระบบฐานข้อมูลนั้นระบบงานที่เลือกใช้เทคนิคการแทนความรู้แบบเฟรมที่พบ กลับเลือกที่จะใช้ระบบฐานข้อมูลเชิงสัมพันธ์เพื่อทำการจัดเก็บความรู้ ทั้งในรูปแบบที่พยายามแมปโครงสร้างข้อมูลของเฟรม ให้มาอยู่ในรูปเชิงสัมพันธ์โดยตรง [5] หรือการแมปโครงสร้างของเฟรม ให้มาอยู่ในโครงสร้างเชิงวัตถุเสียก่อน แล้วจึงค่อยแมปจากโครงสร้างเชิงวัตถุให้มาอยู่ในรูปเชิงสัมพันธ์ [23] ด้วยจุดประสงค์ที่ต้องการรักษาความหมาย (Semantic) ของความรู้ไว้ให้มากที่สุด โดยจำเป็นจะต้องเลือกโมเดลเชิงวัตถุที่มีความสามารถในการอำนวยความสะดวกในการแมปโครงสร้างเชิงวัตถุให้มาอยู่ในรูปเชิงสัมพันธ์ เพื่อลดความยุ่งยากในการดำเนินงาน

การเชื่อมโยงกันระหว่างระบบฐานความรู้กับระบบฐานข้อมูลเป็นการขยายความสามารถของระบบฐานความรู้ให้สามารถรองรับฐานความรู้ขนาดใหญ่ได้ [4] ซึ่งในปัจจุบันได้มีเครื่องมือที่ใช้เชื่อมต่อทั้งสองระบบเข้าด้วยกัน อย่างเช่น Perk [3], EcoCyc [4], Sophia [5] และ PARKA-DB [24]

2.8.2.1 Perk

ระบบนี้จะทำงานในลักษณะเป็น backend ของระบบเฟรมที่ชื่อว่า Ocelot ภายใต้รูปแบบ OKBC โดยจะจัดเก็บข้อเท็จจริงไว้ที่ฐานข้อมูล แล้วใช้คำสั่งรูปแบบ OKBC เพื่อเข้าถึง

เฟรมที่ต้องการ โดยการสร้างมุมมองเชิงวัตถุ ขึ้นมา เพื่อเรียกเฟรมที่เกี่ยวข้อง ขึ้นมาในลักษณะของเฟรม แล้วนำเฟรมที่ได้มาทำการวินิจฉัย ที่ Ocelot ซึ่งอยู่บนฝั่งไคล์เอนท์ อีกที

2.8.2.2 EcoCyc

เป็นเครื่องมือที่ใช้ในการสร้างระบบฐานความรู้ และใช้ระบบฐานความรู้เชิงสัมพันธ์เป็นส่วนจัดเก็บข้อเท็จจริง ระบบนี้จะใช้กลไลการจัดเก็บแบบฝั่งแน่น (persistent storage mechanisim) ซึ่งเฟรมจะถูกจัดเก็บอยู่ในระบบฐานข้อมูลและถูกบีบอัดให้เป็นข้อความ ASCII ก่อนจากนั้นเมื่อมีการเข้าถึงเฟรมจะทำการแปลงกลับให้เป็นเฟรมและจะถูก paged in เข้าไปยังหน่วยความจำอีกที

2.8.2.3 Sophia

จะจัดเก็บการแทนความรู้แบบเฟรม บนฐานข้อมูลเชิงวัตถุสัมพันธ์ โดยใช้ MS-Acess97 ในการจัดเก็บความรู้แบบเฟรม และจะทำการค้นคืนค่าขึ้นมาโดยใช้ คำสั่ง SQL ในการโหลดเฟรมภายใต้รูปแบบ OKBC

2.8.2.4 PARKA-DB

จะเก็บการแทนความรู้แบบเฟรม โดยใช้ระบบการจัดการฐานข้อมูลเชิงสัมพันธ์ และจะโหลดเข้าไปในหน่วยความจำหลักอีกที

ซึ่งจากระบบที่ได้ทำการศึกษามาข้างต้น ระบบดังกล่าวไม่มีกลไกการวินิจฉัยที่ฝั่งฐานข้อมูลที่จัดเก็บฐานความรู้เลย แต่จะเป็นในลักษณะจัดเก็บข้อเท็จจริง หรือเฟรม ไว้ที่ฐานข้อมูลแล้วทำการโหลดข้อเท็จจริงขึ้นมาทำการวินิจฉัยอีกที

2.9 การติดต่อสื่อสารระหว่างระบบฐานความรู้

ในการติดต่อสื่อสารเพื่อแลกเปลี่ยนความรู้ระหว่างระบบฐานความรู้ได้นั้นจำเป็นต้องมีภาษาและโปรโตคอลที่ใช้ในการติดต่อสื่อสารระหว่างระบบเพื่อเป็นมาตรฐานในการแลกเปลี่ยนความรู้ระหว่างระบบฐานความรู้ ซึ่งภาษาและโปรโตคอลที่เป็นที่นิยมและถูกใช้เป็นภาษามาตรฐานในการติดต่อสื่อสารระหว่างระบบฐานความรู้คือภาษา KQML [6]

2.9.1 รูปแบบภาษา KQML (KQML String Syntax)

KQML หรือ Knowledge Query and Manipulation Language เป็นภาษาและโปรโตคอลสำหรับการติดต่อสื่อสารระหว่างซอฟต์แวร์เอเจ้นท์และระบบฐานความรู้ ซึ่งถูกพัฒนาขึ้นมาในปี 1994 โดย Tim Finin [6] โดยเป็นส่วนหนึ่งของ ARPA Knowledge Sharing Effort ซึ่งมีจุดมุ่งหมายในการพัฒนาเทคนิคสำหรับสร้างฐานความรู้ขนาดใหญ่ที่สามารถแชร์ความรู้และสามารถนำความรู้

กลับมาใช้ใหม่ได้ซึ่งแนวคิดแรกเริ่มคือเป็นส่วนติดต่อของระบบฐานความรู้และในภายหลังได้ถูกนำเสนอใหม่เป็นภาษาที่ใช้ในการติดต่อสื่อสารระหว่างเอเจ้นท์

รูปแบบข้อความ KQML (KQML message) และโปรโตคอลสามารถถูกใช้เพื่อติดต่อกับระบบอัจฉริยะ (intelligent system) ได้ ซึ่งอาจจะติดต่อกันโดยใช้โปรแกรมหรือโดยระบบอัจฉริยะตัวอื่นๆ ในข้อความ KQML นั้น "performatives" จะเป็นส่วนบ่งบอกการดำเนินการบนฐานความรู้อื่นๆ

BNF ในรูปที่ 2.18 [6] กำหนดนิยามสำหรับ <ascii>, <alphabetic>, <numberic>, <double-quote>, <backslash>, และ <whitespace> ในนิยามนี้ “*” หมายถึงจำนวนใดๆของเหตุการณ์ที่เกิดขึ้น และ “-“ หมายถึง set difference หมายเหตุ <performative> เป็น specialization ของ <expression>

```
<performative> ::= (<word> {<whitespace> :<word> <whitespace>
<expression>}*)

<expression> ::= <word> | <quotation> | <string> |
   (<word> {<whitespace> <expression>}*)

<word> ::= <character><character>*

<character> ::= <alphabetic> | <numeric> | <special>

<special> ::=    <     | > | = | + | - | * | / | & | ^ | ~ | _ |
                 @     | $ | % | : | . | ! | ?

<quotation> ::= '<expression> | '<common-expression>

<comma-expression> ::= <word> | <quotation> | <string> | ,<comma-expression>
                       (<word> {<whitespace> <comma-expression>}*)

<string> ::= "<stringchar>*" | #<digit><digit>*"<ascii>*

<stringchar> ::= \<ascii> | <ascii>-\-<double-qoute>
```

รูปที่ 2.18 KQML string syntax ในรูปแบบ BNF

2.9.2 Reserved Performative Parameters

performative จะรับ parameter ที่ถูกระบุถึงโดย keyword ดังตารางที่ 2.1 [6]

:sender <word>

:receiver <word>

Parameter พวกนี้แสดง ผู้ส่งและผู้รับจริงๆ ของ performative ซึ่งแตกต่างจาก ผู้รับและผู้ส่ง แบบ virtual ใน :from และ :to parameter ของ networking performative

ตารางที่ 2.1 สรุปค่าพารามิเตอร์ที่เป็นคีย์เวิร์ด และความหมายของมัน

Keyword	*Meaning*
:content	The information about which the performative expresses an attitude
:force	Whether the sender will ever deny the meaning of the performative
:in-reply-to	The expected label in a reply
:language	The name of representation language of the :content parameter
:ontology	The name of the ontology (e.g., set of term definitions) used in the **:content** parameter
:receiver	The actual receiver of the performative
:reply-with	Whether the sender expects a reply, and if so, a label for the reply
:sender	The actual sender of the performative

:reply-with <expression>

:in-reply-to <expression>

ถ้า <expression> เป็น nil หรือ parameter นี้มันขาดหายไปจาก performative แล้วผู้ส่งจะไม่คาดหวังให้มีการตอบกลับ ถ้า <expression> เป็น t แล้วผู้ส่งจะคาดหวังการตอบกลับ มิฉะนั้น ผู้ส่งคาดหวังการตอบกลับที่อยู่ใน :in-reply-to parameter ด้วยค่าที่เหมือนกับ <expression>

:content <expression>

:language <word>

:ontology <word>

:content parameter เป็นการแสดงถึง "direct object" ของ performative ตัวอย่างเช่น ถ้า performative name เป็น tell แล้ว :content จะเป็น sentence ที่ได้ถูกบอกกล่าว <expression>

ใน :content parameter ต้องเป็น valid expression ใน representation language ที่ถูกระบุโดย :language parameter หรือ KQML ถ้า :language parameter ไม่ปรากฏ นอกจากนี้ ค่าคงที่ที่ถูกใช้ใน expression ต้องเป็น subset ของที่ซึ่งถูกกำหนดโดย ontology named ของ :ontology parameter หรือ standard ontology สำหรับ representation language ถ้า :ontology parameter ไม่ปรากฏ

:force <word>

ถ้าค่าของ parameter นี้เป็น permanent แล้วผู้ส่งจะการันตีว่ามันจะไม่ปฏิเสธความหมายของ performative ค่าอื่นๆจะบ่งชี้ว่า ผู้ส่งอาจปฏิเสธความหมายในอนาคต (parameter นี้มีอยู่เพื่อช่วยให้เอเจ้นท์ หลีกเลี่ยง overhead ที่ไม่จำเป็น)

บทที่ 3
สถาปัตยกรรมของระบบ

ในบทนี้นำเสนอสถาปัตยกรรมแบบใหม่ในการเชื่อมต่อระบบฐานความรู้กับระบบฐานข้อมูลเข้าด้วยกัน โดยเพิ่มความสามารถของระบบฐานข้อมูลให้สามารถจัดเก็บโครงสร้างของเฟรม และสามารถทำการวินิจฉัยบนฝั่งระบบฐานข้อมูลได้ โดยไม่จำเป็นต้องโหลดข้อมูลจากฝั่งฐานข้อมูลทั้งหมดมายังระบบฐานความรู้เพื่อทำการวินิจฉัย แต่จะเป็นการถามหาคำตอบจากฝั่งฐานข้อมูลในส่วนที่ต้องทำการวินิจฉัยเพิ่มโดยที่ฝั่งฐานข้อมูลจะทำการวินิจฉัยโดยใช้ กระบวนการสำหรับ การติดต่อสื่อสารระหว่างเฟรม แล้วจึงส่งข้อมูลที่ได้รับการวินิจฉัยระดับหนึ่งแล้วให้กับระบบผู้เชี่ยวชาญอีกที โดยระบบฐานข้อมูลนี้เราจะใช้ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ ซึ่งในงานวิจัยนี้จะใช้ระบบการจัดการฐานข้อมูล ออราเคิล10g ในการ สร้างระบบ เพื่อให้สามารถใช้ลักษณะแบบวัตถุ เหมือนกันภาษาเชิงวัตถุได้โดยที่ยังคงมีความสามารถในการเข้าถึง แอตทริบิวต์ของทุกเฟรมได้แบบฐานข้อมูลเชิงสัมพันธ์ ซึ่งอาจจะมีการดึงข้อมูลจากระบบสารสนเทศอื่นๆ ซึ่งอาจเป็นฐานข้อมูลเชิงสัมพันธ์หรือฐานข้อมูลเชิงวัตถุ ก็ได้ โดยอ้างถึงค่าในสล๊อต จากเฟรมไปยังฐานข้อมูลซึ่งเก็บค่าไว้ได้

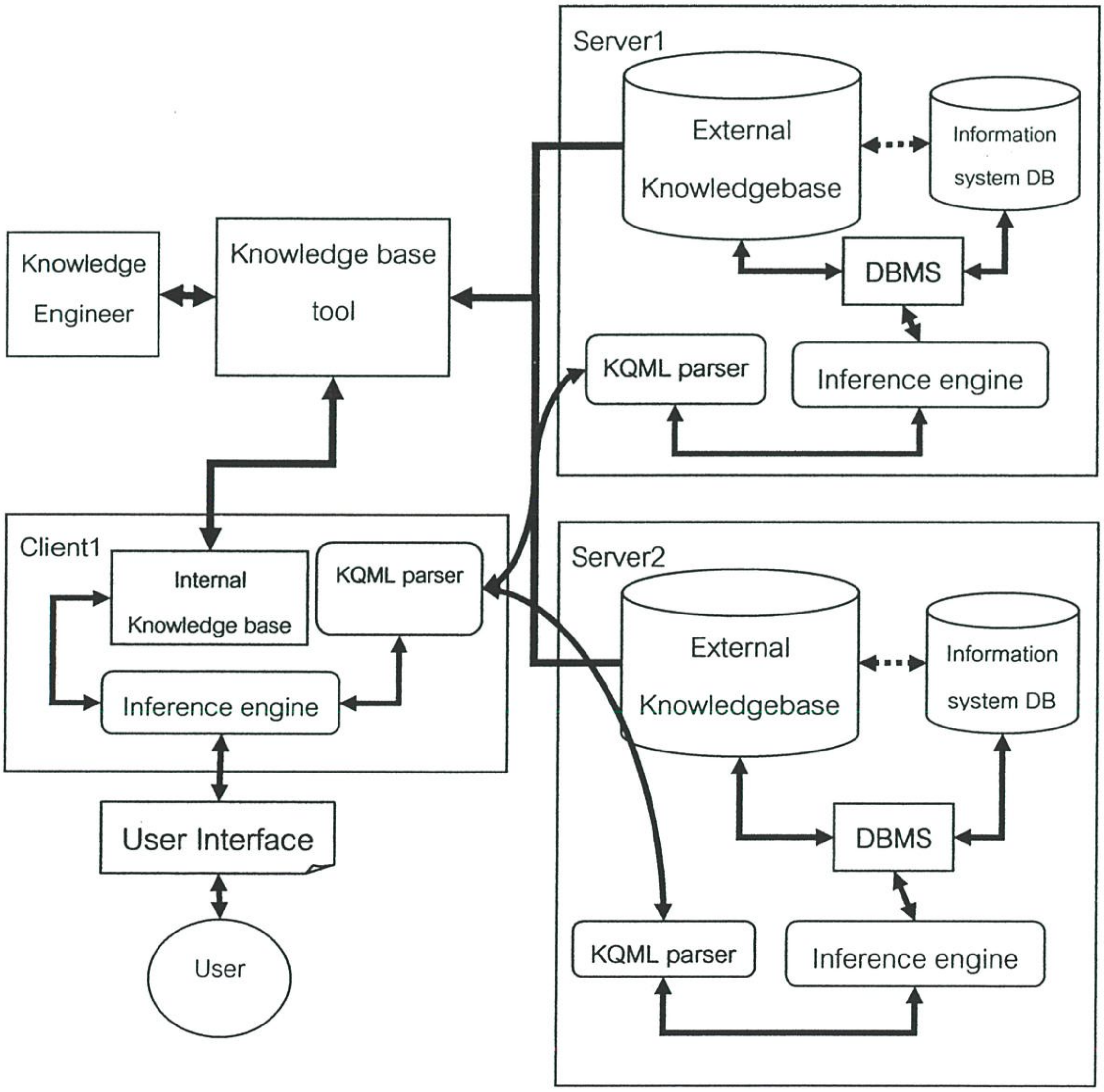
3.1 สถาปัตยกรรมของระบบ

จากรูปที่ 3.1 แสดงสถาปัตยกรรมของระบบ FORXDB (Frame-based Object Relational Expert Database System) ซึ่งคุณสมบัติใหม่ของสถาปัตยกรรมนี้คือระบบสามารถที่จะทำการวินิจฉัยบนฝั่งฐานความรู้ภายนอกที่จัดเก็บอยู่ในระบบฐานข้อมูลได้ และในส่วนระบบผู้เชี่ยวชาญเอง ก็มีกลไกการวินิจฉัยอยู่ภายในด้วยเช่นกัน เพราะฉะนั้นสถาปัตยกรรมนี้ จึงมีการแบ่งการวินิจฉัยออกเป็นสองส่วน ซึ่งการแทนความรู้ของทั้งสองส่วนจะมีโครงสร้างความรู้แบบเฟรม โดยงานวิจัยนี้จะเรียกส่วนระบบผู้เชี่ยวชาญว่าไคลเอนท์ และจะเรียกส่วนฐานความรู้ภายนอกว่าส่วนเซิร์ฟเวอร์ ในส่วนไคลเอนท์ นั้นจะเป็นส่วนที่สื่อสารโต้ตอบกันระหว่างระบบผู้เชี่ยวชาญกับผู้ใช้และรวมรวมข้อเท็จจริงที่ได้จากผู้ใช้ป้อนเข้ามายังระบบด้วย การวินิจฉัยในส่วนนี้สามารถจะอ้างไปถึงข้อเท็จจริงที่อยู่บนฐานความรู้ภายนอกได้ ซึ่งเป็นส่วนเซิร์ฟเวอร์ ทำให้สามารถทำการวินิจฉัยและส่งผลลัพธ์สุดท้ายกลับไปยังฝั่งไคลเอนท์ ได้

ในส่วนไคลเอนท์ นี้จะประกอบไปด้วย ส่วนติดต่อกับผู้ใช้ ฐานความรู้ ตัวสร้างฐานความรู้ และ กลไกการวินิจฉัยบนฝั่งไคลเอนท์ และในส่วนของเซิร์ฟเวอร์นั้นจะประกอบไปด้วย ฐานข้อมูลที่เก็บการแทนความรู้แบบเฟรม กลไกการวินิจฉัยบนฝั่งเซิร์ฟเวอร์ และฐานข้อมูลอื่นๆ (ถ้ามี)

ในระหว่างกระบวนการตอบคำถามนั้น ผู้ใช้จะถามไปยังไคลเอนท์ซึ่งเป็นระบบผู้เชี่ยวชาญแบบเฟรมเช่นกัน ซึ่งจุดประสงค์หลักของการวินิจฉัยที่ส่วนนี้คือการรวบรวมข้อมูลปัจจุบันในหัวข้อนั้นๆ ซึ่งกฎการวินิจฉัยในเฟรมแรกจะเลือกเฟรมที่เหมาะสมที่จะกระทำต่อไป อาจมีหลายเฟรมที่เกี่ยวข้องที่ส่วนไคลเอนท์นี้ก่อนที่ส่วนนี้จะทำการรวบรวมข้อมูลได้เพียงพอ เพื่อส่งไปยังฐานความรู้ภายนอกบนฝั่งเซิร์ฟเวอร์เพื่อทำการวินิจฉัยต่อไป

การวินิจฉัยบนฐานความรู้ภายนอกจะเน้นไปที่การใช้ข้อเท็จจริงที่มีอยู่แล้วซึ่งได้ถูกจัดเก็บเป็นเฟรมไว้ในฐานความรู้และระบบสารสนเทศอื่นๆด้วย คำตอบที่ได้จากการวินิจฉัยบนฝั่งนี้จะถูกส่งกลับไปยังระบบผู้เชี่ยวชาญบนฝั่งไคลเอนท์ และจากนั้นจึงส่งคำตอบไปยังผู้ใช้

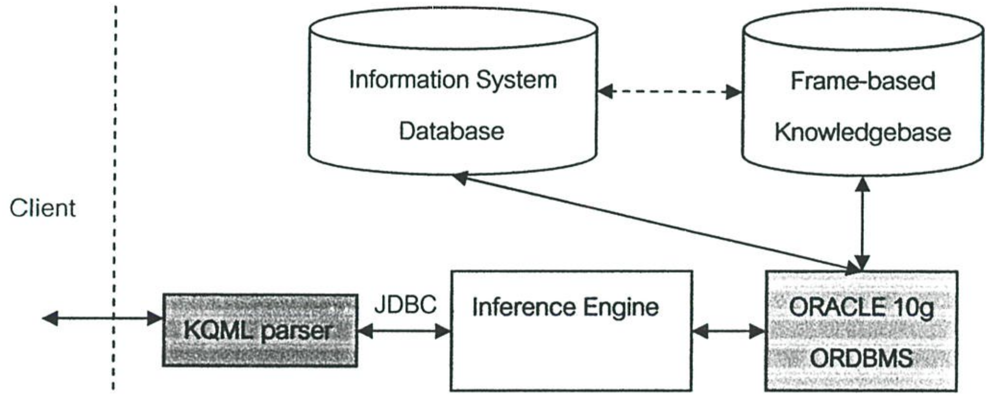


รูปที่ 3.1 แสดงสถาปัตยกรรมของระบบ

ในการติดต่อสื่อสารระหว่างไคลเอนท์ กับเซิร์ฟเวอร์ นั้น ในงานวิจัยนี้ได้เพิ่มความสามารถให้ไคลเอนท์ แต่ละตัวนั้นมีความสามารถที่จะติดต่อได้หลายเซิร์ฟเวอร์ ซึ่งทำให้ในงานวิจัยนี้ได้มีการประยุกต์ใช้ KQML เป็นภาษาและโปรโตคอลในการติดต่อสื่อสารระหว่างไคลเอนท์กับเซิร์ฟเวอร์ ซึ่ง KQML นี้เป็นภาษาที่ใช้ในการติดต่อสื่อสารระหว่างเอเจ้นท์ในระบบ multi-agent system

3.1.1 ฝั่งเซิร์ฟเวอร์

ในส่วนสถาปัตยกรรมของระบบที่ฝั่งเซิร์ฟเวอร์ ในรูปที่ 3.2 นั้น งานวิจัยนี้จะยกส่วนการวินิจฉัยไปอยู่ที่ฝั่ง เซิร์ฟเวอร์ฐานข้อมูล โดยที่ฝั่งเซิร์ฟเวอร์จะรับข้อมูลจากฝั่งไคลเอนท์ แล้วทำการแปลงข้อมูลที่ส่งมาด้วยภาษา KQML ให้เป็นคำสั่ง SQL จากนั้น ก็ส่งไปยังเซิร์ฟเวอร์ฐานข้อมูล และส่วนเซิร์ฟเวอร์ฐานข้อมูลนี้จะเป็นส่วนดำเนินการวินิจฉัย โดยใช้ความสามารถของ object extension ของระบบการจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์



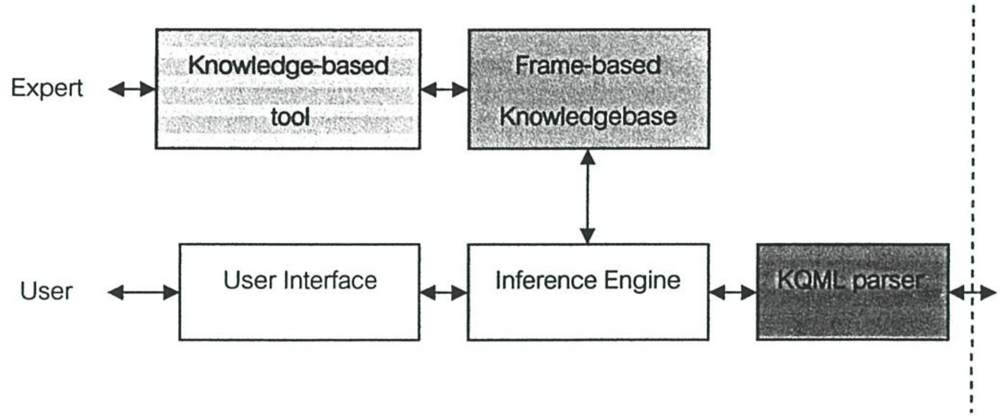
รูปที่ 3.2 แสดงสถาปัตยกรรมของระบบฝั่งเซิร์ฟเวอร์

ที่ฝั่งเซิร์ฟเวอร์นี้จะแบ่งการทำงานออกเป็น 2 ส่วน ได้แก่ ส่วนระบบฐานความรู้ภายนอกบนระบบฐานข้อมูล และการเชื่อมต่อระบบฐานความรู้ภายนอกกับระบบสารสนเทศอื่นๆ (ฝั่งเซิร์ฟเวอร์)

3.1.2 ฝั่งไคลเอนท์

ในฝั่งนี้จะทำการรับข้อเท็จจริงการผู้ใช้เป็นหลักและสามารถมีการวินิจฉัยที่ฝั่งไคลเอนท์ได้เลยถ้าข้อมูลที่ได้รับจากผู้ใช้มีเพียงพอ โดยที่การทำงานนั้นผู้ใช้จะทำการสอบถามจากระบบถึงคำตอบ ระบบจะทำการสอบถามข้อมูลจากผู้ใช้ซึ่งถ้าข้อมูลจากผู้ใช้มีไม่เพียงพอหรือข้อมูลที่ต้องการจะสอบถามอยู่ที่ฝั่งเซิร์ฟเวอร์ ก็จะทำการส่งคำถามที่ต้องการ ไปยังเซิร์ฟเวอร์ โดยส่งไปให้รูปแบบของ KQML message แล้วจากนั้นก็รอรับคำตอบที่ได้จากเซิร์ฟเวอร์

เมื่อทางฝั่งไคลเอนท์ได้รับคำตอบที่ต้องการแล้ว ก็จะทำการวินิจฉัยต่อไปหรือส่งผลลัพธ์ที่ได้กลับไปยังผู้ใช้ได้ดังรูปที่ 3.3



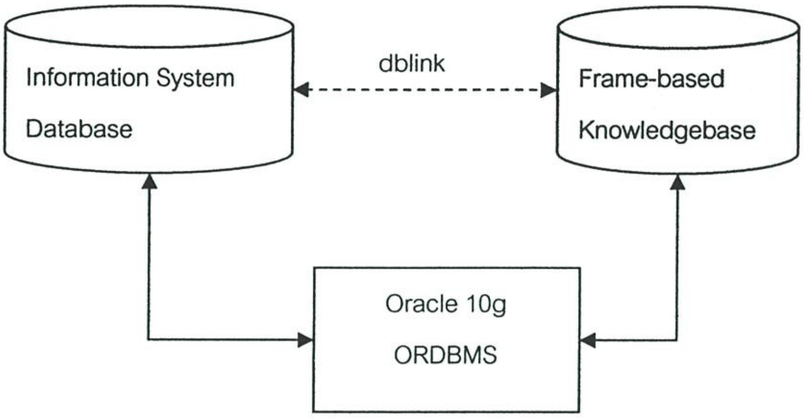
รูปที่ 3.3 แสดงสถาปัตยกรรมของระบบฝั่งไคลเอนท์

3.2 การเชื่อมต่อระบบฐานความรู้ภายนอกกับระบบสารสนเทศอื่นๆ

ในงานวิจัยนี้ ข้อมูลจากระบบสารสนเทศอื่นจะถูกใช้เป็นข้อมูลภายนอก ของระบบ FORXDB นี้ กลไกการวินิจฉัยจะถูกกระทำบนเฟรมที่จะอ้างไปยังเฟรมอื่นๆจนกว่าข้อเท็จจริงจะถูกค้นพบในฐานข้อมูล กระบวนการจะถูกกระทำเพื่อที่จะอ้างถึงข้อเท็จจริงที่อยู่ในฐานข้อมูลภายนอกโดยไม่จำเป็นต้องคัดลอดข้อมูลอย่างถาวรมาไว้ในเฟรมแต่จะเป็นการเรียกดูข้อมูลเมื่อจำเป็นต้องใช้เท่านั้น ดังนั้นจึงสามารถหลีกเลี่ยงปัญหา data inconsistency เมื่อมีการอัพเดตที่ data source ได้และจากการที่ระบบผู้เชี่ยวชาญของเราอยู่บนฐานข้อมูลทำให้เหมือนเป็นการติดต่อระหว่างฐานข้อมูลกับฐานข้อมูลเท่านั้นทำให้การติดต่อกับฐานข้อมูลของระบบสารสนเทศอื่นทำได้โดยง่ายเพราะเป็นเพียงการเชื่อมต่อระบบฐานข้อมูลเข้าด้วยกัน

งานวิจัยนี้ใช้ระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์ออราเคิล 10g ในการจัดการกับฐานความรู้ภายนอกและออราเคิล 10g ได้มีคุณสมบัติที่ชื่อว่า dblink หรือ database link ซึ่งอนุญาตให้ฐานข้อมูลสามารถอ้างถึงฐานข้อมูลออราเคิลอื่นๆได้โดยใช้ SQL statement โดยตรงได้เลยแต่ถ้าในกรณีที่ฐานข้อมูลที่อ้างถึงไม่ใช่ฐานข้อมูลออราเคิลจะมี utility ชื่อว่า OCA (Oracle Open Client Adapter) ซึ่งสามารถใช้เพื่อดึงข้อเท็จจริงจากฐานข้อมูลอื่นขึ้นมาได้

และเนื่องจากข้อเท็จจริงที่ได้จากฐานข้อมูลภายนอกจะถูกกำหนดไว้แล้วจากผู้เชี่ยวชาญตัวอย่างเช่นประวัติผู้ป่วยในเวชระเบียนเป็นต้นซึ่งถ้าไม่มีประวัติระบบผู้เชี่ยวชาญจะไปถามโต้ตอบกับผู้ใช้แทนเพราะฉะนั้นข้อมูลบางอย่างเช่นข้อมูลประวัติคนไข้หรือเวชระเบียนก็สามารถเชื่อมโยงกับระบบเพื่อช่วยในการวินิจฉัยโรคบางอย่างที่ต้องอาศัยข้อมูลประจำตัวในการตัดสินใจ เช่น ข้อมูลประวัติการแพ้ยาหรือประวัติการรับวัคซีนเป็นต้น



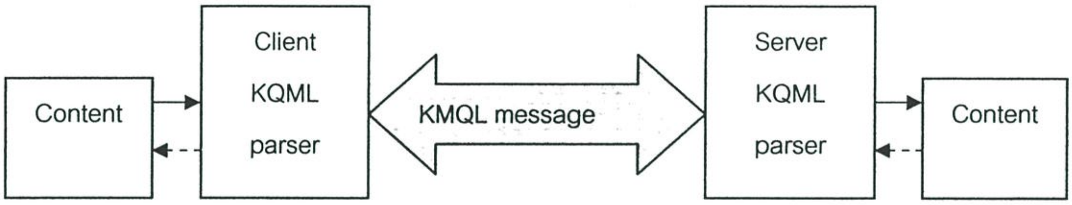
รูปที่ 3.4 แสดงการเชื่อมต่อระหว่างระบบกับฐานข้อมูลภายนอก

ฟังก์ชันที่จะใช้ในการวินิจฉัยจะเรียกเฟรมขึ้นมาทีละเฟรม และเชื่อมต่อเฟรมไปเรื่อยๆ เพื่อหาผลลัพธ์ซึ่งอาจอยู่บนฐานข้อมูลอื่นอีกที และถ้ามีข้อเท็จจริงอยู่บนระบบสารสนเทศอื่นก็จะใช้ วิธีดึงเฉพาะข้อมูลที่เฟรมนั้นสนใจออกมาแล้วทำการวินิจฉัยต่อไป และค่อยไล่ดูเฟรมที่เกี่ยวข้องต่อเนื่องไปเรื่อยๆ อีกที

ยกตัวอย่างเช่นตัวอย่างการวินิจฉัยโรค ถ้าเราต้องการทราบว่า เรามีความเสี่ยงในการที่จะเกิดโรคนั้นๆ เท่าใด ในระบบก็จะทำการสร้าง ตัวอย่างของโรคนั้นขึ้นมา (instance) แล้วมีข้อมูลให้ผู้ใช้ กรอกอาการของผู้ป่วย แล้วระบบก็จะคำนวณโดยดูจากความตรงกันของข้อมูล ซึ่งถ้ามีข้อมูลอาการเพียงครึ่งหนึ่ง ของอาการที่ต้องแสดง ระบบก็จะคำนวณความเสี่ยงต่อโรคนี้ว่ามี 50% อย่างเช่น ความเป็นไปได้ในการที่จะเป็นโรคหัวใจ ของบุคคลหนึ่ง ในระบบการจัดการฐานข้อมูลออราเคิล ก็จะมีคลาสเฟรม ที่บอกถึงอาการของโรคหัวใจไว้ และเราจะ สร้างอินสแตนซ์เฟรมขึ้นมาเพื่อใช้ประมวลผลความเสี่ยงต่อโรคหัวใจของบุคคลนั้นๆ และค่าของ slot ในเฟรม นั้นๆ จะอ้างไปยังเฟรม อื่นๆ หรือ ฐานข้อมูล อื่นๆ อีกที และจะทำการ insert ค่า slot value ของ instance frame นั้น เพิ่มเข้าไปยัง ฐานความรู้จนกว่าจะมีข้อมูลครบถ้วนในการวินิจฉัย แล้วจึงส่งค่าการวินิจฉัย ซึ่งในที่นี้เป็น เปอร์เซ็นของความเสี่ยง กลับไปยังระบบผู้เชี่ยวชาญในส่วนฝั่งไคลเอนท์ อีกที เพื่อใช้ในการวิเคราะห์ข้อมูลเพิ่มเติมต่อไป

3.3 การติดต่อสื่อสารระหว่างไคลเอนท์กับเซิร์ฟเวอร์ด้วยภาษา KQML

ในการติดต่อสื่อสารระหว่างไคลเอนท์กับระบบฐานความรู้ส่วนที่อยู่บนฝั่งเซิร์ฟเวอร์นั้นไคลเอนท์แต่ละตัวสามารถติดต่อได้หลายเซิร์ฟเวอร์ ซึ่งทำให้ในงานวิจัยนี้ได้มีการประยุกต์ใช้ KQML เป็นภาษาและโปรโตคอลสำหรับการติดต่อสื่อสารระหว่างไคลเอนท์กับเซิร์ฟเวอร์ ซึ่ง KQML นี้เป็นภาษาที่ใช้ในการติดต่อสื่อสารระหว่างเอเจ้นท์ในระบบมัลติเอเจ้นท์ (multi-agent system)



รูปที่ 3.5 โปรโตคอลที่ใช้ในการติดต่อสื่อสารระหว่างไคลเอนท์ กับเซิร์ฟเวอร์

ในการติดต่อสื่อสารระหว่างไคลเอนท์กับเซิร์ฟเวอร์นั้นจะต้องมีโปรโตคอลที่ใช้เชื่อมต่อระหว่างกัน ซึ่งในงานวิจัยนี้จะใช้ KQML เป็นโปรโตคอลและ KQML จะสามารถแยกย่อยออกเป็น 3 ชั้น คือ ส่วนชั้นของเนื้องาน, ชั้นของข้อความ, และ ชั้นในการส่ง [6]

ในงานวิจัยนี้การแปลงข้อมูลที่เราจะส่งไปให้อยู่ในรูปข้อความ KQML มีขั้นตอนดังนี้ จากข้อมูลหรือเนื้องานซึ่งในงานวิจัยนี้คือข้อมูลที่ไคลเอนท์จะส่งไปให้กับเซิร์ฟเวอร์ จะใช้ performative ชนิด ask เป็นการถามไปยังฝั่งเซิร์ฟเวอร์ และจะมีตัว parser ทางฝั่งไคลเอนท์ ทำการเข้ารหัสส่วนเนื้องานที่ฝั่งไคลเอนท์จะส่งไปให้อยู่ในรูปแบบข้อความ KQML จากนั้นถึงส่งข้อความไป

ในส่วนของผู้รับทางฝั่งเซิร์ฟเวอร์ ก็จะทำการแปลข้อความที่ได้รับโดยใช้ parser ทำการแปลข้อความ KQML ให้เป็นเนื้องานที่ทางไคลเอนท์ส่งมาแล้วนำไปประมวลผลที่ฝั่งเซิร์ฟเวอร์ต่อไป เมื่อทางฝั่งเซิร์ฟเวอร์มีการประมวลผลเรียบร้อยแล้ว ก็จะทำการส่งคำตอบที่ได้กลับมายังไคลเอนท์ โดยใช้ performative ชนิด tell แล้วทำการเข้ารหัสคำตอบที่ได้ให้อยู่ในรูปแบบ KQML โดยใช้ parser ส่งกลับมายังฝั่งไคลเอนท์ ต่อไป

ในงานวิจัยนี้การแปลงข้อมูลที่เราจะส่งไปให้อยู่ในรูปแบบข้อความ KQML มีขั้นตอนดังรูปที่ 3.5 ดังนี้ คือ จากข้อมูลหรือเนื้องานซึ่งในงานวิจัยนี้ก็คือข้อมูลที่ไคลเอนท์ จะส่งไปให้กับเซิร์ฟเวอร์จะใช้ performative ชนิด ask เป็นการถามไปยังฝั่งเซิร์ฟเวอร์ และจะมีตัว parser ทางฝั่งไคลเอนท์ทำการ encode ส่วน content ที่ฝั่งไคลเอนท์จะส่งไป ให้อยู่ในรูปแบบข้อความ KQML จากนั้น ถึงส่งข้อความไป

ในส่วนของผู้รับทางฝั่งเซิร์ฟเวอร์ ก็จะทำการแปลข้อความที่ได้รับโดยใช้ parser ทำการแปลข้อความ KQML ให้เป็นเนื้องานที่ทางไคลเอนท์ส่งมาแล้วนำไปประมวลผลที่ฝั่งเซิร์ฟเวอร์ต่อไป เมื่อทางฝั่งเซิร์ฟเวอร์มีการประมวลผลเรียบร้อยแล้วก็จะทำการส่งคำตอบที่ได้กลับมายังไคลเอนท์ โดยใช้ performative ชนิด tell แล้วทำการเข้ารหัสคำตอบที่ได้ให้อยู่ในรูปแบบ KQML โดยใช้ parser ส่งกลับมายังฝั่งไคลเอนท์ต่อไป

บทที่ 4
รายละเอียดเกี่ยวกับการสร้างระบบ

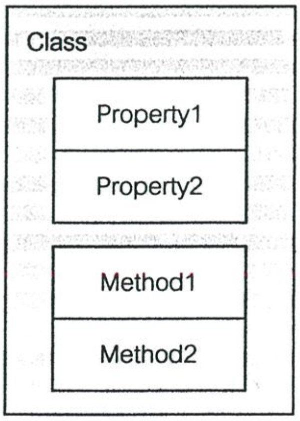
งานวิจัยนี้ได้แบ่งการทำงานออกเป็น 4 ส่วน ได้แก่ การสร้างระบบฐานความรู้ภายนอกบนระบบฐานข้อมูล กลไกการวินิจฉัยบนฝั่งฐานความรู้ภายนอก การเชื่อมต่อระบบฐานความรู้ภายนอกกับระบบสารสนเทศอื่นๆ การติดต่อสื่อสารระหว่างไคลเอนท์กับเซิร์ฟเวอร์ด้วยภาษา KQML

4.1 เปรียบเทียบหลักการเชิงวัตถุสัมพันธ์กับเฟรม

เนื่องจากฐานข้อมูลเชิงวัตถุสัมพันธ์เป็นการขยายความสามารถในด้านวัตถุให้กับฐานข้อมูลเชิงสัมพันธ์ ประกอบกับรูปแบบการแทนความรู้แบบเฟรม กับแนวคิดเชิงวัตถุมีความใกล้เคียงกันดังนั้นในงานวิจัยนี้จึงได้ทำการศึกษาโครงสร้างข้อมูลเชิงวัตถุเปรียบเทียบกับโครงสร้างข้อมูลแบบเฟรมเสียก่อน แล้วเปรียบเทียบความเป็นไปได้ที่จะนำหลักการแนวคิดเชิงวัตถุมาใช้กับเฟรม โดยใช้ฐานข้อมูลเชิงวัตถุสัมพันธ์

4.1.1 โครงสร้างข้อมูลเชิงวัตถุ

โครงสร้างข้อมูลเชิงวัตถุโดยทั่วไป เป็นโครงสร้างข้อมูลที่ยึดหลักการออกแบบโดยอาศัยแนวคิดทางวัตถุสิ่งของทั้งที่จับต้องได้และไม่ได้ โดยจะสร้างโครงสร้างข้อมูลที่รวบรวมเอาตัวบ่งบอกคุณสมบัติของวัตถุนั้นๆ เข้าไว้ด้วยกัน ในลักษณะการเอ็นแคปซูเลต (Encapsulate) โดยภายในตัววัตถุเองอาจประกอบด้วยเมธอด หรือโพรซิเยอร์ ที่เกี่ยวข้องกับวัตถุรวมกันไว้ ดังแสดงให้เห็นในรูปที่ 4.1

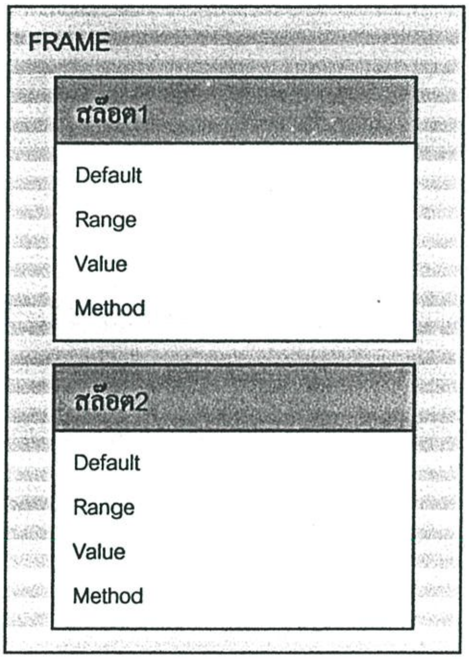


รูปที่ 4.1 แผนภาพโครงสร้างข้อมูลแบบวัตถุ

4.1.2 โครงสร้างข้อมูลแบบเฟรม

เป็นโครงสร้างที่เกิดมาจากแนวคิดในการออกแบบโดยยึดหลักของการออกแบบตามวัตถุ หรือคอนเซ็ปต์ เช่นเดียวกับแนวคิดเชิงวัตถุหากแต่เฟรมจะให้ความสำคัญกับองค์ประกอบในระดับตัวบ่งบอกคุณสมบัติของวัตถุมากกว่า กล่าวคือเฟรมจะมองสิ่งที่มีหน้าที่รวบรวมเอาตัวบ่งบอกคุณสมบัติของวัตถุเป็นเพียงเสมือนกรอบที่ถูกวาดไว้เพื่อรักษาการรวมกลุ่มกันของตัวบ่งบอกคุณสมบัติเท่านั้น นั่นจึงเป็นสาเหตุที่แนวคิดนี้ถูกเรียกว่าเฟรม

ซึ่งในขณะที่ตัวบ่งบอกคุณสมบัติของวัตถุทั้งหลายจะถูกให้ความสนใจเป็นเสมือนวัตถุหน่วยย่อยๆ ที่ภายในตัวเองจะมีคุณสมบัติสำหรับตัวเองรวมถึงเมธอด โพรซิเยอร์ และเดมอนที่เกี่ยวข้องกับตัวเองไว้ด้วยกัน หรือกล่าวอีกนัยหนึ่งคือโพรซิเยอร์หรือเดมอนต่างๆที่มีความเกี่ยวข้องกับตัวบ่งบอกคุณสมบัติใด จะถูกรวบรวมไว้กับตัวบ่งบอกคุณสมบัตินั้นเอง ทำให้หน่วยย่อยที่ทำหน้าที่เป็นตัวบ่งบอกคุณสมบัติของวัตถุนั้นถูกเรียกว่าสล็อต ดังแสดงให้เห็นในรูปที่ 4.2



รูปที่ 4.2 แผนภาพโครงสร้างข้อมูลแบบเฟรม

จากข้อมูลเบื้องต้นด้านโครงสร้างข้อมูลแบบวัตถุ และเฟรม ทำให้พบว่า มีความแตกต่างในด้านการให้ความสำคัญระดับกับข้อมูลดังที่ได้กล่าวมาแล้วข้างต้นทำให้โครงสร้างข้อมูลปกติของระบบฐานข้อมูลเชิงวัตถุไม่สามารถรองรับการจัดเก็บและการบริหารจัดการกับข้อมูลที่อยู่ในโครงสร้างแบบเฟรมได้โดยตรงเป็นสาเหตุให้งานวิจัยโดยส่วนใหญ่ที่ต้องการพัฒนาระบบ

ผู้เชี่ยวชาญต้องเลือกใช้วิธีการแมปหรือแปลงโครงสร้างข้อมูลแบบเฟรมให้มาอยู่ในรูปวัตถุก่อนจะจัดเก็บ

สำหรับงานวิจัยในวิทยานิพนธ์ฉบับนี้มีความต้องการที่จะประยุกต์ใช้โครงสร้างข้อมูลเชิงวัตถุสัมพันธ์ให้สามารถจัดเก็บข้อมูลที่ถูกสร้างด้วยการแทนความรู้แบบเฟรมได้โดยตรงเพราะโครงสร้างข้อมูลเชิงวัตถุนั้นจะมีการห่อหุ้มข้อมูล (encapsulation) แต่ในโครงสร้างข้อมูลแบบเฟรมนั้นไม่มีการห่อหุ้มข้อมูล และด้วยข้อกำจัดในความแตกต่างด้านการให้ความสำคัญระดับข้อมูลนี้ทำให้งานวิจัยนี้จึงเลือกใช้ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์

4.2 การสร้างระบบฐานความรู้บนระบบฐานข้อมูลเชิงวัตถุสัมพันธ์

จากผลการศึกษาความแตกต่างระหว่างโครงสร้างข้อมูลเชิงวัตถุ และเฟรมตามที่ได้อธิบายไว้ข้างต้นแล้ว จะพบว่าข้อแตกต่างหลักอยู่ที่การให้ความสำคัญในระดับของข้อมูลที่ต่างกัน กล่าวคือ หลักการเชิงวัตถุจะให้ความสำคัญเน้นที่ระดับวัตถุหรือคอนเซ็ปต์ แต่เฟรมจะให้ความสำคัญในระดับที่ต่ำกว่า คือระดับตัวบ่งบอกคุณสมบัติของวัตถุ หรือคอนเซ็ปต์ ทำให้ความรู้ที่ใช้เทคนิคการแทนความรู้แบบเฟรมจะไม่สามารถจัดเก็บลงในโครงสร้างข้อมูลเชิงวัตถุได้โดยตรง ดังนั้นจึงมีความจำเป็นจะต้องดัดแปลง หรือสร้างโครงสร้างใหม่ภายในระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ ที่มีรูปแบบเหมือน หรือใกล้เคียงกับ เฟรมมากที่สุด

จากสาเหตุข้างต้น ทำให้งานวิจัยนี้เลือกใช้ฐานความรู้เชิงวัตถุสัมพันธ์ในการจัดเก็บโครงสร้างแบบเฟรม ซึ่งมีความสามารถเข้าในระดับตัวบ่งบอกคุณสมบัติของเฟรมได้ และในขณะเดียวกัน ก็สามารถจะใช้มุมมองในรูปแบบเชิงวัตถุ ได้ด้วย

ในสถาปัตยกรรมของระบบที่สร้างขึ้นนั้น เพื่อที่จะเก็บฐานความรู้บนระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ นั้นเราต้องกำหนดการแปลงจากโมเดลความรู้ (knowledge model) ของระบบฐานความรู้แบบเฟรมไปเป็นโมเดลข้อมูล (data model) ของระบบฐานข้อมูลเชิงวัตถุสัมพันธ์เสียก่อน

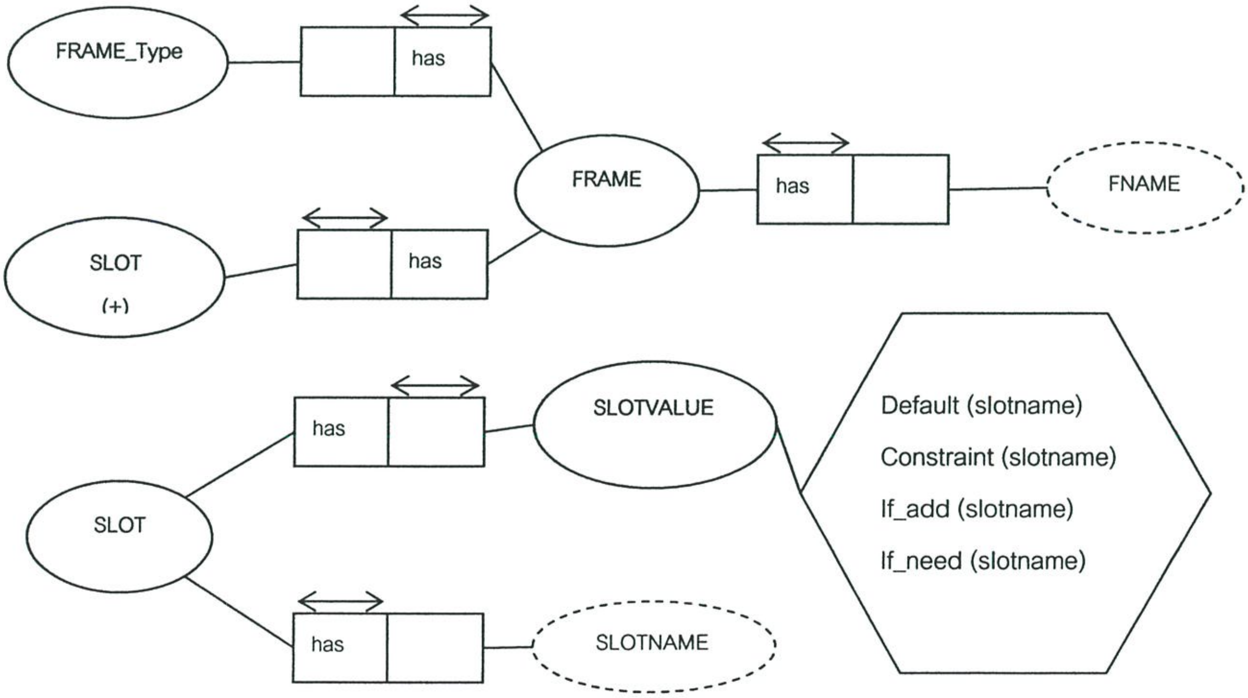
4.2.1 การจัดเก็บเฟรมในฐานข้อมูลเชิงวัตถุสัมพันธ์

เนื่องจากในการที่จะจัดเก็บโครงสร้างแบบเฟรม ในฐานข้อมูลเชิงวัตถุสัมพันธ์นั้น เราจะต้องทำการแปลงโครงสร้างแบบเฟรม ให้สามารถจัดเก็บอยู่ในฐานข้อมูลเชิงวัตถุสัมพันธ์ให้ได้เสียก่อน ซึ่งจากการศึกษาพบว่า ในการจัดเก็บฐานความรู้ในระบบฐานข้อมูลนั้น ไม่จำเป็นจะต้องเป็นฐานความรู้ชนิดเดียวเท่านั้น ที่สามารถถูกจัดเก็บได้ เพื่อที่ว่าเวลาเรียกข้อมูลจากฐานความรู้ขึ้นมา หรืออ้างอิงไปยังอีกฐานความรู้หนึ่ง จะทำให้เกิดความซับซ้อนในขั้นตอนการดำเนินงาน

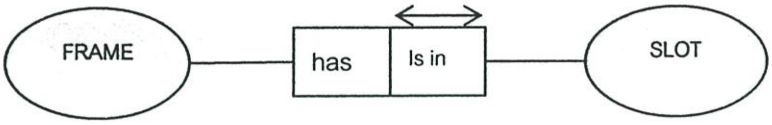
งานวิจัยนี้ยังได้ทำเสนอการออกแบบตารางที่ใช้เก็บฐานความรู้แบบเฟรม โดยใช้ NIAM [25] สำหรับออกแบบฐานข้อมูลในการจัดเก็บเฟรมต่างๆ ซึ่ง NIAM เป็น fact-based conceptual

schema model ซึ่งประกอบไปด้วย ชนิดของเอนทิตี้ (entity type), ชนิดของเลเบล (label type), ชนิดของข้อเท็จจริง (fact type) และ ชนิดของการอ้างอิง (reference type) นอกจากนี้กฎควบคุมความถูกต้องของข้อมูล (static integrity constraints) ก็สามารถถูกแสดงบน conceptual schema diagram ได้ด้วย และ มีส่วนขยายเพื่อที่จะรองรับรูปแบบของฐานข้อมูลเชิงวัตถุได้ด้วย ดังนี้ [26] [27]

1. ชนิดของเอนทิตี้สามารถเกิดขึ้นได้โดยไม่จำเป็นต้องมี ตัวบ่งบอกค่าเฉพาะ (unique identifier) ซึ่งเรียกว่า เอนทิตี้ชนิดซับซ้อน(complex entity type) ซึ่ง เอนทิตี้ชนิดซับซ้อนนี้จะแสดงถึง คลาสของวัตถุ ซึ่งตัวบ่งบอกอินสแตนท์ของวัตถุ (object instance identifier) จะถูกสร้างขึ้นจากระบบสร้าง ids ของวัตถุขึ้นมาเอง
2. เพิ่ม สัญลักษณ์ สำหรับกระบวนการ
3. เพื่อที่จะสนับสนุน วิธีการแบบบนลงล่าง (top-down) แนวคิดของ main schema และ sub schema ได้ถูกนำเสนอขึ้นด้วย โดยที่ sub schema จะอธิบายถึง ชนิดของข้อเท็จจริง และ ชนิดของการอ้างอิง ซึ่งเป็นไพเวท และถูกห่อหุ้ม ภายใน เอนทิตี้ชนิดซับซ้อนส่วน main schema จะอธิบายถึง ชนิดของข้อเท็จจริง ของ เอนทิตี้ชนิดซับซ้อน



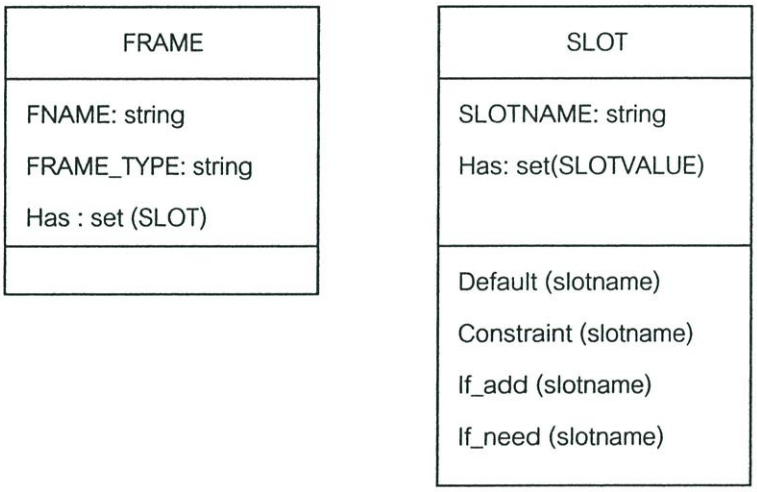
รูปที่ 4.3 A Frame sub schema Using NIAM



รูปที่ 4.4 A Frame main schema Using NIAM

จากรูปที่ 4.3 และรูปที่ 4.4 จะได้ว่า ระบบนี้ในฐานข้อมูลหนึ่งจะมีเฟรมได้หลายเฟรมซึ่งในแต่ละเฟรมจะมีชื่อของเฟรม และ ชนิดของเฟรม รวมทั้งมีเซต ของ สล๊อตหรือ แอตทริบิวต์ บรรจุอยู่ภายใน โดยที่ในแต่ละ สล๊อตจะประกอบไปด้วย ชื่อของสล๊อต และค่าของสล๊อตซึ่งค่าของสล๊อต จะถูกกำหนดโดย ฟาเซท ซึ่งเป็น ข้อกำหนด หรือ กระบวนการ สำหรับควบคุมค่าใน สล๊อต ดังรูป

จากรูปที่ 4.3 และรูปที่ 4.4 จะได้ schema ออกมา ดังนี้



รูปที่ 4.5 Object Schema ของเฟรม

จากรูปที่ 4.5 จะได้ว่าระบบนี้มีสองวัตถุ คือ เฟรม และ สล๊อตโดยใน วัตถุ เฟรม จะเก็บชื่อของเฟรม และชนิดของเฟรม พร้อมทั้ง collection ของ สล๊อตส่วน สล๊อตวัตถุ ก็จะมีชื่อ และ collection ของ ค่าสล๊อต โดยมี กระบวนการไว้ควบคุม ค่าของสล๊อต ซึ่งก็คือฟาเซต นั่นเอง

และจากตัวอย่างข้างต้น เราสามารถทำการแปลงให้อยู่ในรูปแบบ nested table ได้ในรูปที่ 4.6 ดังนี้

<table>
<tr><td colspan="4">FRAME</td></tr>
<tr><td rowspan="4">FRAMENAME</td><td rowspan="4">FRAMETYPE</td><td rowspan="2">SLOTNAME</td><td>SLOTVALUE</td></tr>
<tr><td>SLOTVALUE</td></tr>
<tr><td rowspan="2">SLOTNAME</td><td>SLOTVALUE</td></tr>
<tr><td>SLOTVALUE</td></tr>
</table>

รูปที่ 4.6 Object-Relational Schema ของเฟรม

ระบบฐานข้อมูลออราเคิล 10g นั้นเป็นระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ โดยระบบจะทำการเพิ่มความสามารถของระบบฐานข้อมูลเชิงสัมพันธ์ ด้วยการเพิ่มความสามารถเชิงวัตถุ บนระบบฐานข้อมูลเชิงสัมพันธ์อีกที ซึ่งเมื่อมีการติดต่อกับฐานข้อมูลมันจะทำการติดต่อกับข้อมูลเสมือนว่าข้อมูลนั้นๆถูกจัดเก็บแบบวัตถุและด้วยความสามารถแบบวัตถุที่เพิ่มเข้ามาทำให้สามารถจัดเก็บข้อมูลวัตถุในฐานข้อมูลได้ และกระบวนการสามารถถูกเรียกโดยตรงจากภาษา SQL ได้ ซึ่งอย่างไรก็ตามระบบก็ยังต้องแปลงข้อมูลเกี่ยวกับวัตถุไปเป็นข้อมูลในตารางอยู่ดีและจัดการกับข้อมูลเหมือนกับฐานข้อมูลเชิงสัมพันธ์ ในขณะเดียวกันเมื่อมีการค้นคืนข้อมูลออกมามันก็จะแปลงจากข้อมูลตารางมาเป็นวัตถุที่ซับซ้อน (complex object) ดังนั้นระบบฐานข้อมูลเชิงวัตถุสัมพันธ์จึงสามารถลดการเกิดความซ้ำซ้อนได้จากการสร้างตารางข้อมูลเชิงสัมพันธ์ที่ไม่มีความซ้ำซ้อน แล้วใช้มุมมองเชิงวัตถุในการเรียกดูเฟรมขึ้นมาแบบเชิงวัตถุซึ่งวิธีนี้ระบบสามารถเข้าถึงข้อมูลทั้งในแบบเชิงวัตถุและเชิงสัมพันธ์ได้ทั้งสองแบบตามการใช้งาน

ซึ่งในฐานข้อมูลออราเคิล 10g เราสามารถจัดเก็บเฟรมได้ดังนี้คือ

1. สร้างตารางที่จัดเก็บข้อมูลของฟาเซ็ทขึ้นมาโดยสร้างเป็น collection type ของวัตถุ
2. สร้างตารางที่จัดเก็บสล็อตโดยที่ภายในแต่ละแถวของสล็อตจะมีตารางของฟาเซ็ทอยู่
3. สร้างชนิดเชิงวัตถุที่เป็นเฟรมซึ่งประกอบไปด้วยชื่อของเฟรมซึ่งแต่ละเฟรมจะมีตารางของสล็อตซึ่งในแต่ละสล็อตจะมีตารางของฟาเซ็ท

```
CREATE OR REPLACE TYPE t_facets_row AS OBJECT
 ( id   NUMBER(10),
   slot_id NUMBER(10),
   name VARCHAR2(50),
   facet_value VARCHAR2(50)
```

```
 )
CREATE OR REPLACE TYPE t_facets_tab AS TABLE OF t_facets_row
CREATE OR REPLACE TYPE t_slots_row AS OBJECT (
 id        NUMBER(10),
 frame_id  NUMBER(10),
 name      VARCHAR2(50),
 facets    t_facets_tab
)
CREATE OR REPLACE TYPE t_slots_tab AS TABLE OF t_slots_row
CREATE OR REPLACE TYPE t_frame_row AS OBJECT
 ( id    NUMBER,
   name   VARCHAR2 (50),
   frametype VARCHAR2(50),
   slots    t_slots_tab
 )
```

4.3 กลไกการวินิจฉัยของระบบ

ในระบบผู้เชี่ยวชาญแบบเฟรมนั้นมีการวินิจฉัยได้หลายรูปแบบ เช่น การวินิจฉัยโดยการสืบทอด (inference by inheritance) เป็นการตอบคำถามด้วยการหาเฟรม ไม่ว่าจะเป็นคลาสเฟรม หรืออินสแตนซ์เฟรม ที่มีสล๊อตและค่าของสล๊อตที่ต้องการ ซึ่งบางครั้งการสืบทอดสามารถถูกนิยามเป็นกระบวนการ ได้ [4] ตัวอย่างการวินิจฉัยโดยการสืบทอด เช่น Does Shania Twain have a spleen? จากคำถามนี้ถ้าเราไล่ไปยัง คลาสเฟรม ของ อินสแตนซ์เฟรม Shania Twain จะพบว่าเป็นสมาชิกของคลาสเฟรม woman จะสรุปได้ว่า Shania Twain is woman และเมื่อไล่ขึ้นไปอีกจะพบว่า a woman is a kind of person ในคลาสเฟรม person ก็สรุปได้ว่า persons have spleens ดังนั้นเราสามารถสรุปได้ว่าโดยการวินิจฉัยโดยการสืบทอด Shania Twain has a spleen เป็นต้น

การจำแนกโดยการจับคู่ (Classification by matching) เป็นการจำแนก อินสแตนซ์เฟรม โดยการจับคู่ค่าใน อินสแตนซ์เฟรม กับ ค่าใน คลาสเฟรม ที่เกี่ยวเนื่องกัน ว่าอันไหนตรงที่สุด ตัวอย่างเช่น What kind of singer is Shania Twain? คำถามนี้ระบบจะทำการจับคู่ อินสแตนซ์เฟรม ของ Shania Twain เข้ากับ คลาสเฟรมต่างๆเหล่านี้ คือ opera-singer, country-singer, rock-singer, และ pop-singer ซึ่งถ้าคุณสมบัติของอินสแตนซ์เฟรม Shania Twain มันตรงกันกับ คลาสเฟรม pop-singer มากที่สุด ก็จะสามารถสรุปได้ว่า Shania Twain ถูกจำแนกให้เป็น pop singer

ระบบผู้เชี่ยวชาญที่ใช้วิธีการแทนความรู้แบบเฟรมส่วนใหญ่นั้น มักจะใช้เทคนิคการจำแนกประเภท (Classification) และ inheritance ในการวินิจฉัย และแก้ปัญหาบนพื้นฐานของความรู้ที่จัดเก็บไว้ในฐานความรู้ ดังนั้นในงานวิจัยนี้จึงใช้กลไกการวินิจฉัยสองแบบคือ pattern matching และการสืบทอด

4.3.1 ขั้นตอนการวินิจฉัยของระบบ

กระบวนการวินิจฉัยนี้จะมีอยู่ด้วยกันสองแบบ แบบแรกคือให้ผู้ใช้กรอกข้อมูลเข้ามาแล้วทำการจับคู่ข้อมูลที่ได้รับกับเฟรม เพื่อระบุว่าเป็นอินสแตนท์ของคลาสเฟรมใด ส่วนแบบที่สองจะเป็นการถามว่าเฟรมที่ป้อนเข้ามาเป็นอินสแตนท์ของเฟรมที่ระบุหรือไม่ ซึ่งกระบวนการจะมีขั้นตอนดังนี้

ฝั่งไคลเอนต์

1. ระบบจะทำการเปรียบเทียบค่าที่ได้ป้อนเข้ามา กับเฟรมอาการแรกเริ่มที่มีอยู่
2. จากนั้นจะเลือกเฟรม ขึ้นมาก่อนและตั้งเฟรมเป็นเป้าหมาย
3. เมื่อได้เป้าหมายแล้วก็จะใส่ค่าในสล๊อตของเฟรม
4. และในเฟรมเป้าหมายนี้ ถ้าไม่มีข้อเท็จจริงในเฟรมก็จะให้ผู้ใช้ตอบคำถาม
5. ทริกเกอร์ในสล๊อตนี้ไปเรียกเฟรมถัดมาเมื่อมีการใส่ค่าเข้าไป
6. จากนั้นเมื่อได้รับข้อมูลจากผู้ใช้ครบถ้วนแล้วระบบจะส่งคำถามไปยังฝั่งเซิร์ฟเวอร์

ฝั่งเซิร์ฟเวอร์

1. ฝั่งเซิร์ฟเวอร์เมื่อรับเฟรมเข้ามาก็จะทำการเลือกเฟรมเป้าหมาย
2. เมื่อได้ลิสต์ของเป้าหมายแล้วก็เก็บไว้ในลิสต์ goal_list
3. เลือกเฟรมใน goal_list มาหนึ่งเฟรม โดยลบเฟรมออกจากลิสต์ goal_list ด้วย
4. จับคู่อาการในเฟรมนั้นกับข้อมูลที่ได้มา
5. ถ้ามีการร้องขอข้อมูลเพิ่มเติมจากในฐานข้อมูลภายนอก ให้ไปดึงข้อมูลจากฐานข้อมูลภายนอกเข้ามา
6. ถ้ามีการร้องขอให้ผู้ใช้ป้อนข้อมูลเพิ่มเติม ส่งคำถามไปยังฝั่งไคลเอนต์เพื่อให้ผู้ใช้ตอบคำถาม แล้วส่งคำตอบมายังฝั่งเซิร์ฟเวอร์
7. ถ้าเฟรมเป้าหมายสอดคล้องกับข้อมูลทั้งหมด ก็ส่งเฟรมนั้นเป็นคำตอบกลับไปยังผู้ใช้
8. ถ้าข้อมูลไม่สอดคล้องกับเฟรมเป้าหมาย ให้เลือกเฟรมเป้าหมายใหม่จากลิสต์ goal_list แล้วทำการจับคู่เฟรมอีกครั้งจนกว่าจะได้คำตอบ หรือจนกว่าไม่มีเฟรมเหลืออยู่ใน goal_list อีกแล้ว

จากขั้นตอนการวินิจฉัยของระบบข้างต้นเราสามารถเขียนอัลกอริทึมเพื่อทำการวินิจฉัยโดยรวมของระบบได้ดังนี้

```
Function:    find frame
Input:       facts
Output:      answer

1. find frame that match facts

2. loop
3.     find children of matched frame
4.     push into goal stack list
5. until no children of frame found

6. pop frame from goal stack
7. bind slot variable with fact
8. match frame with current goal frame

9. return matched frame
```

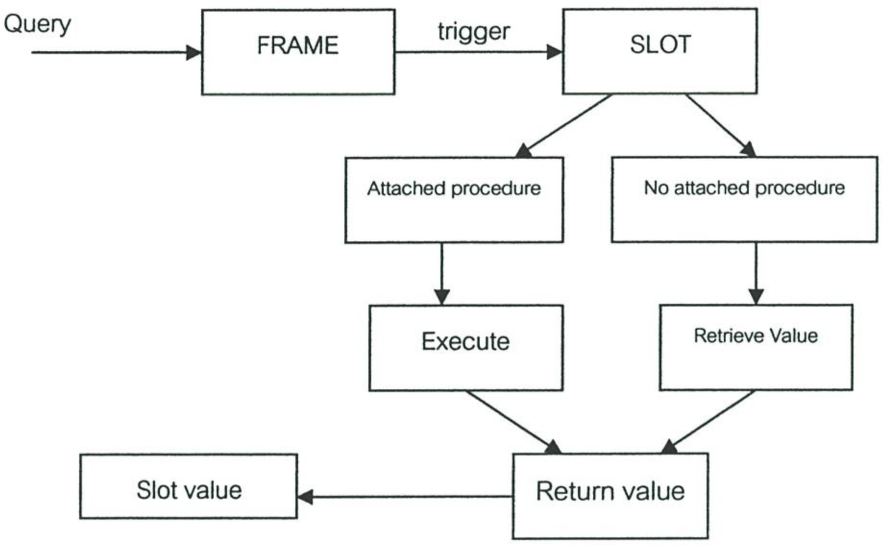
รูปที่ 4.7 อัลกอริทึมในการวินิจฉัยของระบบ

จากขั้นตอนการวินิจฉัยข้างต้นกลไกการวินิจฉัยนี้จะกระทำก็ต่อเมื่อได้มีการส่งคำถามมาจากฝั่งไคลเอนท์เพื่อให้เซิร์ฟเวอร์ทำการวินิจฉัยต่อโดยที่ข้อมูลทางฝั่งไคล์เอนต์จะทำการวินิจฉัยไประยะนึงก่อนเพื่อให้ผู้ใช้ได้ตอบคำถาม (forward reasoning) ซึ่งเมื่อได้ข้อมูลมาพอสมควร ระบบจะทำการตั้งเป้าหมายแล้วค้นหาว่าข้อมูลที่ได้มาตรงกับเป้าหมายใดมากที่สุด ซึ่งจะเข้าสู่กระบวนการวินิจฉัยเพื่อจับคู่เฟรมที่สอดคล้องที่สุด (backward reasoning) ดังนั้นในงานวิจัยนี้การวินิจฉัยจะเป็นแบบผสม คือมีการ forward reasoning ไประยะนึงก่อนจากนั้นถึงทำการวินิจฉัยแบบ backward reasoning ต่อไปจนพบเป้าหมายแล้วส่งผลลัพธ์หรือคำตอบกลับไปยังผู้ใช้ได้

4.3.2 กลไกการวินิจฉัยที่ฝั่งฐานความรู้ภายนอก

งานวิจัยนี้จะใช้โปรแกรมที่เขียนด้วยภาษาจาวา เป็นตัวติดต่อกับฐานข้อมูล ผ่านทาง JDBC driver ของระบบการจัดการฐานข้อมูล ออราเคิล 10g โดยเทคนิคในการวินิจฉัยนั้นจะมีอยู่ด้วยกันสองวิธี หนึ่งคือ รับเฟรมจากฝั่งไคลเอนท์เข้ามาเก็บไว้ในเฟรมบนฝั่งเซิร์ฟเวอร์ซึ่งกระบวนการจะถูกเรียกขึ้นมาจากการเปลี่ยนแปลงค่าในสล๊อต และสองคือ เก็บข้อมูลที่ได้จากเฟรมบนฝั่งไคลเอนท์มาเป็น พารามิเตอร์ของ store procedure ซึ่งจะทำการวินิจฉัยโดยใช้ store procedure ต่อไป ซึ่งวิธีที่สองนี้จะเหมาะสำหรับข้อมูลที่มาจากฝั่งไคลเอนท์มีจำนวนไม่มาก โดยที่ในส่วนฐานข้อมูลฝั่งเซิร์ฟเวอร์ซึ่งเราเก็บฐานความรู้นั้น จะมีการเก็บ store procedure ไว้ในระบบการจัดการฐานข้อมูล เพื่อกระทำการตามสถานการณ์ โดยตัวที่จะใช้เรียก ฟังก์ชัน จะอยู่ในส่วนของโปรแกรมซึ่งค่าที่ได้จากการประมวลผลจะถูก ส่งกลับไปยังไคลเอนท์ต่อไป

ซึ่งค่าของ สล๊อต จะถูกเรียกใช้เมื่อต้องการ ซึ่งในส่วนของ attached procedure นี้จะแตกต่างกันไปในแต่ละทูเพิลซึ่งบางที อาจต้องเรียกต่อเนื่องไปเรื่อยๆ จนกว่าจะได้คำตอบ แล้วจึงประมวลค่าอีกครั้งเพื่อส่งค่าที่ได้กลับไปยังฝั่งไคลเอนท์อีกที



รูปที่ 4.8 แสดงการทำงานของกระบวนการภายในเฟรม

ในส่วนของกระบวนการจะมีการทำงานดังนี้คือเมื่อรับคำสั่งจากผู้ใช้หรือระบบผู้เชี่ยวชาญฝั่งไคลเอนท์มันจะทำการค้นหาค่าในเฟรมและเมื่อได้เฟรมที่ต้องการก็จะไปตรวจสอบที่สล๊อตว่าได้มี stored procedure สำหรับเหตุการณ์นั้นๆหรือไม่ จากนั้นจึงรับค่าที่ได้กระทำการในส่วนนี้ ไปยังใส่ไว้ในฟิลเลอร์ซึ่งเป็นค่าของสล๊อตในเฟรมอินสแตนซ์อีกที โดยที่การใส่ค่าในสล๊อตก็อาจจะมีกระบวนการที่กระทำภายหลังจากที่ใส่ค่าแล้วต่อไปก็ได้ ในรูปที่ 4.7 เป็นการแสดงการทำงานของกระบวนการภายในเฟรมโดยอันดับแรกจะมีการป้อนค่าหรือเรียกดูค่าในส่วนของค่าของสล๊อตของเฟรมอินสแตนซ์โปรแกรมจะดูว่าได้มีกระบวนการที่ต้องกระทำหลังป้อนค่าหรือก่อนเรียกดูค่าหรือไม่ ถ้ามีก็จะไปเรียก store procedure โดยอ้างอิงจาก attached procedure ขึ้นมาซึ่งกระบวนการนี้อาจมีการร้องขอให้มีการป้อนข้อมูลที่สล๊อตอื่นได้ ซึ่งในรูปที่ 4.8 จะมีอัลกอลิทึมดังนี้

```
Function:        get_all_slot_value
Input:           frame_id, slot_name
Output:          answer
answer:=null;
parent:=null;
answer = call get_slot_value;
if answer = null
      parent = call find_parent;
      while parent is not null AND answer = null
              parent = call find_parent;
      end loop
return answer;
```

รูปที่ 4.9 อัลกอริทึมในการหาค่าของสล๊อต

กระบวนการในรูปที่ 4.9 จะเป็นการหาค่าสล๊อตจากเฟรมปัจจุบันไล่ขึ้นไปยังเฟรมพ่อแม่ตามลำดับขั้นและจะมีการเรียกฟังก์ชันในรูปที่ 4.10 และ 4.11 ตามลำดับดังนี้

```
Function :  get_slot_value
Input:      frame, slot_name
Output:     answer
answer:= null;
begin
        SELECT id INTO v_slot_id FROM   slots_tab
        if v_slot_id is not null then
        begin
              SELECT  'value' INTO ret
              when no_data_found
                     SELECT 'default' INTO ret
                     when no_data_found
                            SELECT if_needed INTO ret
                            when no_data_found
                                   answer = NULL;
  end if;
  answer ret;
end;
```

รูปที่ 4.10 อัลกอริทึมสำหรับการดึงค่าจากสล๊อต

```
Function:    find_parent
Input:       frame, slot_name
Output:      parent
begin
        parent := null;
        SELECT parent INTO parent FROM  parent_tab
        if  no_data_found then
             parent := null;
        return parent
end;
```

รูปที่ 4.11 อัลกอริทึมสำหรับการหาเฟรมแม่

4.4 การดึงข้อมูลสารสนเทศจากฐานข้อมูลอื่นเข้ามายังระบบเฟรม

ในงานวิจัยนี้ เราจะทำการอ้างถึงข้อมูลจากฐานข้อมูลที่มีอยู่เข้ามายังฝั่งฐานข้อมูล ที่เป็นระบบผู้เชี่ยวชาญโดยการเข้าถึงเฟรมที่มีอยู่ แล้วเรียกต่อๆ กันไปจนไปถึงข้อมูลจริงๆ ซึ่งอยู่บนฐานข้อมูลอื่น โดยมีกระบวนการที่อ้างไปถึงข้อเท็จจริง ที่อยู่ในฐานข้อมูล อื่น ในเฟรม นั้นๆ ซึ่งไม่จำเป็นต้องทำการ โหลดข้อมูลจากฐานข้อมูลที่มีอยู่แล้วเข้ามายังเฟรม ก่อน แต่จะเป็นการอ้างถึงข้อมูลที่เกี่ยวข้อง ซึ่งก็คือข้อเท็จจริง จากฐานข้อมูลอื่นๆ มายังระบบเฟรม ที่อยู่ใน ฐานข้อมูลของออราเคิล อีกที โดยที่ข้อมูลจากฐานข้อมูลตัวอื่น ที่เราเชื่อมต่อกับระบบผู้เชี่ยวชาญแบบเฟรม นี้ ไม่จำเป็นต้องมีการเปลี่ยนแปลงรูปแบบใดๆ และผู้ใช้ ของระบบนั้นสามารถเข้าถึงข้อมูลได้ตามปกติ ซึ่งระบบที่เราวิจัยนี้ระบบฐานข้อมูลที่เก็บฐานความรู้บนฝั่งเซิร์ฟเวอร์จะทำหน้าที่เสมือนกับเป็นตัวระบบผู้เชี่ยวชาญ ที่เชื่อมต่อกับฐานข้อมูลอีกที โดยระบบของเราจะเป็นระบบผู้เชี่ยวชาญ ที่เป็นระบบจัดการฐานข้อมูลด้วย ทำให้การเชื่อมต่อกับระบบฐานข้อมูลอื่นถูกกระทำได้โดยง่ายและเป็นการใช้ฐานข้อมูลที่มีอยู่แล้วให้มีประสิทธิภาพด้วย

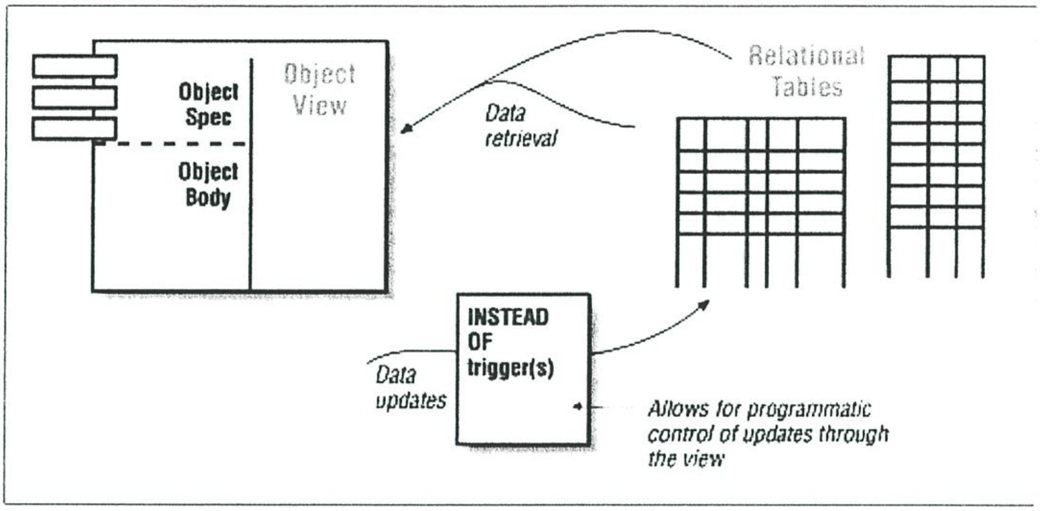
และด้วยเหตุที่เราต้องเชื่อมต่อระบบผู้เชี่ยวชาญนี้ กับระบบฐานข้อมูลอื่นๆ การเลือกใช้ฐานข้อมูลเชิงวัตถุสัมพันธ์ ก็เป็นประโยชน์ซึ่งทำให้เราสามารถเชื่อมโยงกับระบบฐานข้อมูลอื่นที่มีอยู่แล้วที่ส่วนใหญ่จะเป็นระบบฐานข้อมูลเชิงสัมพันธ์ หรือบางทีอาจจะเป็นฐานข้อมูลเชิงวัตถุสัมพันธ์ก็ได้ เพราะระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ นี้สามารถรองรับการเชื่อมต่อกับระบบฐานข้อมูลทั้งสองแบบ และกระทำได้โดยง่าย

ซึ่งในงานวิจัยนี้เราได้ใช้คุณสมบัติที่มีอยู่แล้วของออราเคิล 10g นั่นคือ dblink ทำให้สามารถเชื่อมต่อกับฐานข้อมูลอื่นที่เป็นฐานข้อมูลของออราเคิลเหมือนกันได้และสามารถทำการจัดการ (DML) กับฐานข้อมูลนั้นได้เลย แต่ในกรณีที่เป็นการเชื่อมต่อกับฐานข้อมูลอื่นที่ไม่ใช่ออราเคิลเหมือนกันอย่างเช่น MySQL เป็นต้นในระบบของเราจะเป็นเพียงการดึงข้อมูลขึ้นมาเพื่อให้ได้ข้อเท็จจริงที่ต้องการเท่านั้นโดยอาจจะใช้ OCA (Oracle Open Client Adapter) สำหรับเชื่อมต่อกับฐานข้อมูลอื่นและในขั้นตอนนี้ส่วนใหญ่เราต้องการเพียงข้อเท็จจริงเพิ่มเติมเท่านั้นจึงไม่มีความจำเป็นที่จะต้องเปลี่ยนแปลงข้อมูลของฐานข้อมูลที่เชื่อมต่ออยู่ด้วยเพราะฉะนั้นจึงเป็นเพียงการสร้างฟังก์ชัน SQL เพื่อทำการคิวรีข้อมูลจากฐานข้อมูลอื่นขึ้นมาเท่านั้น

ในงานวิจัยนี้การติดต่อกับฐานข้อมูลภายนอกอื่นๆจะจำลองรูปแบบการเรียกใช้งานของระบบผู้เชี่ยวชาญที่จัดเก็บเฟรมอยู่ในฐานข้อมูลแต่ว่าระบบที่สร้างขึ้นนี้จะต้องไม่ไปยุ่งเกี่ยวกับโครงสร้างที่มีอยู่แล้วในฐานข้อมูลภายนอกอื่นๆดังนั้นในงานวิจัยนี้จึงได้มีการทำตารางเสมือนขึ้นมาเพื่อสร้างมุมมองเชิงวัตถุ (object view) เพื่อมองข้อมูลภายนอกเป็นลักษณะเฟรม และทำให้ฐานความรู้ที่จัดเก็บอยู่ในระบบฐานข้อมูลสามารถเรียกข้อมูลขึ้นมาใช้งานได้

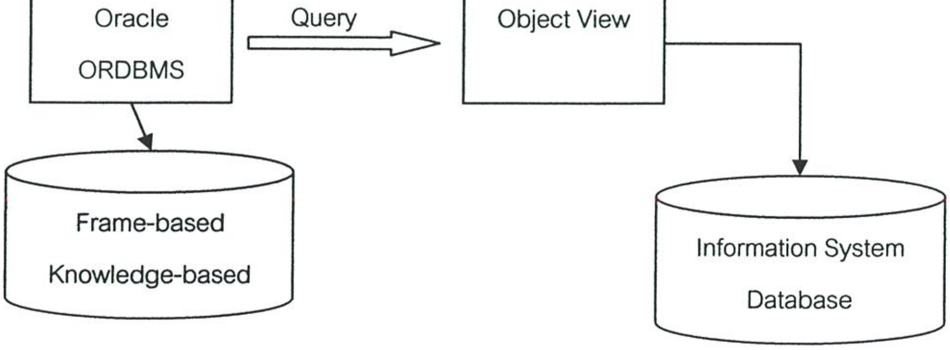
ในรูปที่ 4.12 [7] จะแสดงการเรียกดูข้อมูลจากฐานข้อมูลโดยผ่านมุมมองเชิงวัตถุในฐานข้อมูลออราเคิลตัวอื่น โดยที่ ข้อมูลที่ต้องถามไปที่ฐานข้อมูลอื่นจะมีการไปคิวรี จากฐานข้อมูล

ก่อนซึ่งถ้ายังไม่เคยมีค่าในระบบ ก็จะทำการสอบถาม ผู้ใช้ต่อไป เช่นข้อมูลทางการแพทย์ ที่จะถามถึงประวัติคนไข้ในฐานข้อมูลที่เป็นเวชระเบียน ซึ่งถ้าคนไข้ไม่เคยมีประวัติมาก่อน เมื่อทำการ คิวรีแล้วไม่ได้คำตอบ ก็จะสอบถามผู้ใช้ถึงประวัติคนไข้ เพื่อเป็นข้อมูลไว้อ้างถึงข้อเท็จจริงในระบบผู้เชี่ยวชาญในภายหลังได้ เพราะฉะนั้นตัวฐานข้อมูลที่เป็นระบบผู้เชี่ยวชาญ ของงานวิจัยนี้จะเก็บ SQL คิวรี หรือกระบวนการที่เกี่ยวข้องผูกไว้กับเฟรมเพื่ออ้างถึงข้อเท็จจริงนั้นเพื่อเชื่อมไปยังข้อเท็จจริงที่อยู่ในฐานข้อมูลอื่นต่อไป



รูปที่ 4.12 แสดงการใช้มุมมองเชิงวัตถุกับฐานข้อมูลภายนอก

เฟรมที่จะถามจากฐานข้อมูลนั้นจะเป็นอินสแตนซ์เฟรม ส่วนคลาสเฟรมจะมี attached procedure ซึ่งจะสืบทอด attached procedure นี้มายังอินสแตนซ์เฟรม ซึ่งเมื่อต้องการจะค้นคืน ค่าขึ้นมา ก็จะไปค้นหาข้อเท็จจริงจากฐานข้อมูลก่อน



รูปที่ 4.13 แสดงการเข้าถึงข้อมูลของฐานข้อมูลอื่นจากเฟรม

จากรูปที่ 4.13 เราอธิบายได้ว่า ผู้ใช้ได้กรอกข้อมูลลงไป สมมติว่ามีข้อมูลอยู่แล้วเป็นคนไข้เก่า ก็จะกรอกเพียงแค่เลขที่บัตรประชาชนเท่านั้น เมื่อส่งข้อมูลไปทำการวินิจฉัยบนฝั่งเซิร์ฟเวอร์จะส่งค่าเลขที่บัตรประชาชนไปด้วย เพื่อว่าถ้ามีการร้องขอข้อมูลส่วนตัวจากฐานข้อมูลอื่นในภายหลัง ก็จะไปดึงข้อมูลขึ้นมาได้เลยโดยไม่จำเป็นต้องถามผู้ใช้อีกครั้ง แต่ในขณะเดียวกันถ้าข้อมูลเป็นประวัติคนไข้ใหม่ผู้ใช้ก็ต้องขึ้นประวัติคนไข้ใหม่ โดยอาจให้เจ้าหน้าที่ทางส่วนเวชระเบียนป้อนข้อมูลต่างๆ ลงบนฐานข้อมูลให้เรียบร้อยก่อน แล้วจึงมาทำการวินิจฉัยเหมือนกับเป็นคนไข้เก่าปกติ

4.5 โปรโตคอลที่ใช้ในการติดต่อสื่อสารระหว่างไคลเอนท์กับเซิร์ฟเวอร์

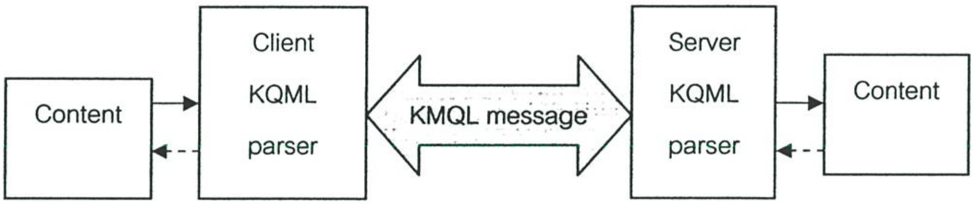
ในการติดต่อสื่อสารระหว่างผู้เชี่ยวชาญอย่างมีประสิทธิภาพ จะต้องประกอบไปด้วย

a) ภาษากลางในการติดต่อสื่อสาร (common language)

b) ความเข้าใจเกี่ยวกับความรู้ที่จะแลกเปลี่ยน (exchange) แบบเดียวกัน

c) ความสามารถในการแลกเปลี่ยนอะไรก็ตามที่อยู่ใน (a) กับ (b)

ในงานวิจัยนี้ เราเลือกใช้ KQML เป็นโปรโตคอลที่ใช้ในการติดต่อสื่อสารระหว่างไคลเอนท์กับเซิร์ฟเวอร์ เนื่องจากเป็นที่นิยมใช้อย่างแพร่หลาย ในการติดต่อสื่อสารระหว่างเอเจ้นท์ ซึ่งในการติดต่อสื่อสาร ระหว่าง เอเจ้นท์นั้น ข้อมูลที่จะถูกส่งไป จะต้องถูกแปลงให้อยู่ในรูปแบบภาษา KQML



รูปที่ 4.14 โปรโตคอลที่ใช้ในการติดต่อสื่อสารระหว่างไคลเอนท์กับ เซิร์ฟเวอร์

จากรูปที่ 4.14 จะแสดงถึงการแปลงข้อมูลที่ต้องการจะส่งให้อยู่ในรูปแบบ KQML เสียก่อนด้วย KQML parser ให้อยู่ในรูปแบบข้อความ KQML และส่งข้อความนั้น ไปยังผู้รับ ซึ่งก็คือเซิร์ฟเวอร์โดยเมื่อได้รับข้อความแล้วจะต้องทำการแปลงข้อความ KQML ให้ออกมาเป็นข้อมูลเสียก่อน แล้วนำข้อมูลนั้นไปทำการวินิจฉัยต่อไป เมื่อทำการวินิจฉัยเสร็จสิ้นแล้ว ก็ส่งคำตอบที่ได้กลับไปยังฝั่งไคลเอนท์ด้วยวิธีการเดิม

เฟรมที่ได้จากฝั่งไคลเอนท์นี้จะอยู่ในรูปแบบของ script ในภาษาจาวา และเมื่อเราจะทำการส่งไปยังเซิร์ฟเวอร์ที่เกี่ยวข้อง เราต้องทำการเข้ารหัสเฟรมนี้ให้อยู่ในรูปแบบ ข้อความ KQML ดังนี้

```
(ask-one
        :sender client1
        :content "get-value(dog,blood-type)"
        :receiver server1
        :reply-with knowledge-base1
        :language LISP
        :ontology animal-kingdom)
```

ในข้อความนี้เนื้องาน ก็คือส่วนของเฟรมทางฝั่งไคลเอนท์ที่เราทำการส่งไป ontology คือ ขอบเขตของความรู้ ส่วนที่เราต้องการจะถามเกี่ยวกับเรื่องนั้นๆ ซึ่งในที่นี้ เกี่ยวข้องกับฐานความรู้เกี่ยวกับอาณาจักรสัตว์ ผู้รับ คือ server1 และกระบวนการคิวรี ถูกเขียนขึ้นโดยใช้ภาษาจาวา ค่าของ parameter ต่อไปนี้คือ sender, receiver และ reply-with จะเป็นส่วน communication layer ในส่วน performative name (ในที่นี้คือ ask-one), language และ ontology จะเป็นส่วน message layer และส่วน เนื้องาน จะเป็นส่วน content layer ใน KQML

ทางฝั่งเซิร์ฟเวอร์จะเก็บข้อมูลไว้ใน store procedure ต่างหาก และจะไล่ตามเฟรม dog ก่อนแล้ว ค้นหาไปยังเฟรมต่างๆ ที่มีสล๊อตเกี่ยวข้อง ซึ่ง เมื่อได้ผลลัพธ์แล้ว ทางฝั่งเซิร์ฟเวอร์ก็จะส่งกลับไปให้ไคลเอนท์ที่ทำการร้องขอ และจะทำการ encode ให้อยู่ในรูปข้อความ KQML ดังนี้

```
(tell
        :sender server1
        :content "warm"
        :receiver client1
        :in-reply-to knowledge-base1
        :language sql
        :ontology animal-kingdom)
```

บทที่ 5
การพัฒนาระบบผู้เชี่ยวชาญ
สำหรับการวินิจฉัยโรค

การวินิจฉัยโรคเป็นกระบวนการที่ต้องอาศัยผู้เชี่ยวชาญคือแพทย์ ในการวินิจฉัยอาการที่ตรวจพบในเบื้องต้นของผู้ป่วย เพื่อทำไปสรุปหรือวิเคราะห์ความน่าจะเป็นว่าผู้ป่วย ป่วยเป็นโรคอะไร และต้องทำการรักษาแบบไหน โดยในบทนี้จะกล่าวถึงวิธีการวินิจฉัยโดยสังเขป รวมถึงการสร้างระบบ

ในงานวิจัยนี้เราได้กำหนดผู้ใช้ เป็นบุคคลที่มีความรู้ในด้านสาธารณสุข และมีความสามารถในการตรวจโรคเบื้องต้น ซึ่งอาจเป็น พยาบาล บุรุษพยาบาล อาสาสมัคร หรือ หมอประจำตำบล ซึ่งส่วนใหญ่จะเป็นที่ๆ ซึ่งขาดแคลนหมอ และเครื่องมือแพทย์ที่ทันสมัย ดังนั้นระบบผู้เชี่ยวชาญนี้ จึงมีประโยชน์อย่างมากในการรักษาและวินิจฉัยโรคเบื้องต้นให้กับคนไข้ในชนบทหรือที่ๆ อยู่ห่างไกลความเจริญ

ในการวินิจฉัยโรคสิ่งที่ต้องการทราบคือ ประวัติข้อมูลเกี่ยวกับผู้ป่วย อาการที่ตรวจพบและการซักประวัติความเจ็บป่วย เพื่อให้เห็นภาพมากขึ้น จึงขอยกตัวอย่างการวินิจฉัยโรค ซึ่งคนไข้มาด้วยอาการปวดหูเป็นหลัก

5.1. ขั้นตอนการทำงานของระบบ

ระบบผู้เชี่ยวชาญนี้จะทำการวินิจฉัยโรคจากการวิเคราะห์อาการ โดยรับข้อมูลที่จำเป็นทั้งหมด แล้วส่งข้อมูลไปวินิจฉัยบนฝั่งเซิร์ฟเวอร์จากนั้นจึงส่งผลการวินิจฉัยกลับมาว่าเป็นโรคใด ในขณะที่ทางฝั่งไคลเอนท์จะทำการวินิจฉัย เพื่อถามอาการในเรื่องที่เกี่ยวข้อง เช่น ถ้าคนไข้มาด้วยอาการปวดหู จะต้องถามอะไรบ้าง หรือถ้าคนไข้มาด้วยอาการเจ็บคอ ควรถามอะไรบ้าง เป็นต้น ดังนั้นบนฝั่งเซิร์ฟเวอร์จะเป็นเฟรมที่เกี่ยวกับโรคต่างๆ พร้อมทั้งอาการของโรคนั้นๆ ส่วนฝั่งไคลเอนท์จะเป็นเฟรมที่เกี่ยวข้องกับการถามคำถาม และการเลือกคำถามออกมาถามผู้ป่วย โดยจะสรุปเป็นกระบวนการได้จากรูปที่ 5.1 จะแสดงขั้นตอนการทำงานของระบบดังที่กล่าวไว้ข้างต้นดังนี้

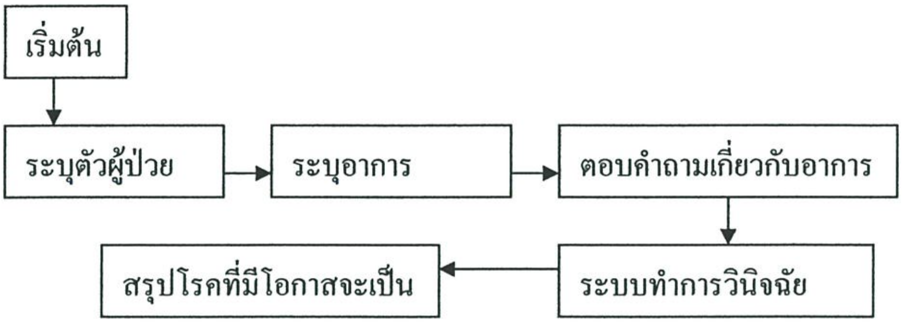
5.1.1. ระบุตัวผู้ป่วยที่ต้องการวินิจฉัยโรค

5.1.2. ระบุอาการเบื้องต้น เช่น ปวดหู เจ็บคอ ไอ มีไข้

5.1.3. ระบบจะถามอาการเบื้องต้นของผู้ป่วย หรืออาจให้ user ตรวจอาการแล้วป้อนข้อมูลเข้าไป

5.1.4. ระบบส่งข้อมูลที่ได้ไปยังเซิร์ฟเวอร์เพื่อทำการวินิจฉัย

5.1.5. สรุปว่ามีโอกาสที่จะเป็นโรคอะไรได้บ้าง และแสดงรายละเอียดของโรค



รูปที่ 5.1 แสดงขั้นตอนการทำงานของระบบวินิจฉัยโรค

5.2. การสร้างเฟรมใช้ แทนความรู้

ในการสร้างเฟรมนั้น เราจะแบ่งออกเป็น 2 ฝั่ง คือ ฝั่งเซิร์ฟเวอร์และฝั่งไคลเอนท์โดยในฝั่งเซิร์ฟเวอร์จะเป็นเฟรมเกี่ยวกับโรคทั้งหมด โดยในที่นี้คือเฟรมเกี่ยวกับโรคหูทั้งหมด ส่วนในฝั่งไคลเอนท์จะเป็นเฟรมที่เกี่ยวข้องกับการถามอาการของโรคโดยจะเรียงลำดับตามความเร่งด่วน เช่น ถ้าคนผู้ป่วยมาด้วยอาการปวดหู สิ่งที่ต้องถามเป็นอันดับแรกคือ การบาดเจ็บที่ศีรษะอย่างรุนแรง เพราะเป็นภาวะเร่งด่วน ควรรับการรักษาทันที เป็นต้น ตารางที่ 5.1 จะแสดงเฟรมเกี่ยวกับโรคหูและโรคที่เกี่ยวข้องกับการปวดหู

ตารางที่ 5.1 แสดงเฟรมที่เกี่ยวกับโรคหู และโรคที่เกี่ยวข้องกับการปวดหู

No.	frame	slot	slot value	facet	facet value
N1	basilar skull fracture	Parent	-		
		Earache	yes	default	yes
		Head injury	yes	default	yes
		Cerebrospinal fluid otorrhea	yes	default	yes
N2	Secondary otalgia (referred pain)	Parent	-		
		Earache	yes	default	yes
		Pain on pulling the auricles	no	default	no

ตารางที่ 5.1 (ต่อ)

E1	Primary otalgia (Disease of the Ear)	Parent	-		
		Children	External Ear, Middle Ear, Inner Ear		
		Earache	yes	default	yes
		Pain on pulling the auricles	yes	if changed	if slot value = no then go to "Secondary otalgia"
E2	External Ear	Parent	Primary otalgia		
		Children	otomycosis, acute cellulitis, furunculosis, herpes, relapsing polychondritis, Ear Trauma, Foreign body, Impacted cerumen		
		Pain and swelling of external ear	yes	default	yes
				if changed	if slot value = no then go to "Middle Ear"
E3	Middle Ear	Parent	Primary otalgia		
		Children	AOM, OME, Trauma to TM		
		Otitis media	yes	default	yes
				if changed	if slot value = no then go to "Inner Ear"

ตารางที่ 5.1 (ต่อ)

E4	Inner Ear	Parent	Primary otalgia		
		Children	Bell palsy, Temporal bone tumor		
E5	Otomycosis	Parent	External Ear		
		Characteristic of ear wax	{white plaque, black spore}*	constraint	{white plaque, black spore}
E6	Acute cellulitis	Parent	External Ear		
		Infection and inflammation	yes	default	yes
E7	Furunculosis	Parent	External Ear		
		Small firm tender red nodule in skin	yes	default	yes
E8	Herpes	Parent	External Ear		
		Itch	yes	default	yes
		Burning pain	yes	default	yes
		Small red bumps or blisters appear around auricles and mouth	yes	default	yes

ตารางที่ 5.1 (ต่อ)

E9	Relapsing polychondritis	Parent	External Ear		
		inflammation and deterioration of cartilaginous tissue	yes	default	yes
		inflammation of other connective tissue	{hearing loss, eyes inflammation, joint inflammation}*	constraint	{hearing loss, eyes inflammation, joint inflammation}
		Serology test	negative	default	negative
E10	Ear Trauma	Parent	External Ear		
		Trauma to auricles	yes	default	yes
E11	Foreign body	Parent	External Ear		
		Foreign body	yes	default	yes
E12	Impacted cerumen	Parent	External Ear		
		partial loss of hearing	yes	default	yes
		a sensation of fullness in the ear	yes	default	yes
E13	Acute otitis media (AOM)	Parent	Middle Ear		
		Pus in the middle ear	yes	default	yes
		Infection	yes	default	yes
				If changed	If slot value = no then go to "OME"
		redness of the eardrum	yes	default	yes

ตารางที่ 5.1 (ต่อ)

E14	Otitis media with effusion (OME)	Parent	Middle Ear		
		Glue ear	yes	default	yes
		Presence of fluid in the middle ear	yes	default	yes
		Infection	no	default	no
				If changed	If slot value = yes then go to "AOM"
E15	Trauma to TM	Parent	Middle Ear		
		Trauma to eardrum	{Barotrauma, Blunt trauma, Laceration}*	constraint	{Barotrauma, Blunt trauma, Laceration}
E16	Bell palsy	Parent	Inner Ear		
		Unilateral or bilateral acute facial nerve palsy	yes	default	yes
E17	Temporal bone tumor	Parent	Inner Ear		
		-	-		

* หมายถึงเลือกอย่างใดอย่างหนึ่ง

จากเฟรมใน ตาราง 5.1 เราจะทำการจัดเก็บไว้ที่ฝั่งเซิร์ฟเวอร์โดยที่จะรับค่าอาการต่างๆ มาจากฝั่งไคลเอนท์และทำการวินิจฉัยพร้อมสรุปอาการส่งให้โดยในฝั่งไคลเอนท์ก็จะมีเฟรมที่ใช้ถามอาการต่างๆ ของโรค ดังนี้

ตารางที่ 5.2 เฟรมคำถามเกี่ยวกับอาการที่เกี่ยวข้องกับการปวดหู

No	frame	slot	slot value	facet	facet value
B1	Basic Question	Earache		if changed	if slot value = yes then go to "Symptoms to earache"
		Cough		if changed	if slot value = yes then go to "Symptoms to cough"
		Sore throat		if changed	if slot value = yes then go to "Symptoms to sore throat"
		Fever		if changed	if slot value = yes then go to "Symptoms fever"
B2	Symptoms to earache	Earache	yes	default	yes
		Severe head injury		if changed	if slot value = yes then go to "Emergency"
		Pain on pulling the auricles		if changed	if slot value = yes then go to "Primary otalgia"
B3	Emergency	Cerebrospinal fluid otorrhea		if changed	if slot value = no then go to "Symptoms to earache" and set "Severe head injury" = no else admit

ตารางที่ 5.2 (ต่อ)

B4	Primary otalgia	Characteristic of external ear		default	normal
				constraint	{Normal, Inflammation}
		Characteristic of ear wax		default	normal
				constraint	{normal, white, black}*
		Itche		default	no
				constraint	yes/no
		Inflammation		default	no
				constraint	yes/no
		Characteristic of eardrum		default	normal
				constraint	{normal, redness of the eardrum, glue ear, perforation of eardrum}*
		symptoms about facial nerve		default	normal
				constraint	{normal, Unilateral or bilateral acute facial nerve palsy}

จากตารางที่ 5.2 ผู้ใช้ สามารถเลือกกรอกเฉพาะอาการที่แสดงได้ และเมื่อได้ข้อมูลอาการของโรค แล้ว ระบบจะสร้างเฟรมขึ้นมาเพื่อบอกอาการทั้งหมด พร้อมทั้งหมายเลขที่อ้างอิงถึงคนไข้ เมื่อกรอกข้อมูลแล้ว จึงส่งข้อมูลไปยังฝั่งเซิร์ฟเวอร์เพื่อประมวลผลต่อไป

5.3 การวินิจฉัยหาคำตอบของระบบผู้เชี่ยวชาญที่สร้างขึ้น

5.3.1. เมื่อผู้ป่วยบอกอาการเบื้องต้นแล้ว ระบบจะดูว่าเป็นอาการเบื้องต้นแบบใด แล้วเรียกคำถามที่ต้องถามเกี่ยวกับโรคที่มีอาการเบื้องต้นนั้น

5.3.2. เมื่อผู้ป่วยกรอกข้อมูลที่เกี่ยวข้องแล้ว จะทำการส่งค่าทั้งหมดไปยัง ฝั่งเซิร์ฟเวอร์เพื่อทำการวินิจฉัยแยกเฉพาะโรค

5.3.3. เมื่อฝั่งเซิร์ฟเวอร์รับเฟรมที่บอกอาการต่างๆของโรค จะทำการเปรียบเทียบเพื่อหาอาการของโรค และแบ่งประเภทโรค ออกจากอาการนั้นๆ ได้ โดยไล่หาที่เฟรมของโรคต่างๆที่เกี่ยวกับหู แล้วสร้าง อินสแตนซ์ ขึ้นมาเพื่อเปรียบเทียบลักษณะอาการ จนได้โรคนั้นๆ ออกมา ซึ่งอาการโดยรวมต่างๆ จะ สืบทอด มาจากคลาสเฟรม

5.3.4. ในกรณี ที่ผู้ป่วย อาจกรอกข้อมูลไม่ครบถ้วน เนื่องจากต้องอ้างอิงถึงประวัติเก่า เช่น ประวัติแพ้ยา หรือประวัติการให้วัคซีน ระบบจะทำการดึงข้อมูลเพิ่มเติม จากฐานข้อมูลอื่นๆ อย่างเช่น เวชระเบียน เพื่อทำการวินิจฉัยต่อไป

5.3.5. เมื่อฝั่งเซิร์ฟเวอร์ทำการวินิจฉัย และสรุปผลการตรวจโรคแล้ว จะส่งผลลัพธ์ ของโรคที่น่าจะเป็นส่งไปให้ผู้ป่วย ที่อยู่บนฝั่งไคลเอนท์ได้

ในกรณีตัวอย่างนี้สมมุติผู้ป่วยมาด้วยอาการปวดหู เพียงอย่างเดียว โดยทดสอบดึงที่ใบหูแล้วพบว่ามีอาการเจ็บปวด เมื่อถูกดึง จากนั้น ผู้ป่วย ไม่มีอาการผิดปกติ อย่างอื่นอีกเลย ทางฝั่งไคลเอนท์จะได้ข้อสรุปคร่าวๆ ว่าผู้ป่วยมีอาการของโรคหู แล้วจะทำการสร้างเฟรม คนผู้ป่วย ซึ่งจะมีสล๊อต ดังนี้ หมายเลขอ้างอิง, ชื่อผู้ป่วย, กลุ่มอาการ, อาการที่แสดง ดังรูปที่ 5.2 ดังนี้

frame	slot	slot value
Patient	reference number	12345
	Patient Name	นายปลอดโรค แข็งแรงดี
	Symptoms	Otalgia
	Earache	yes
	Pain on pulling the auricles	yes

รูปที่ 5.2 เฟรมผู้ป่วยที่ฝั่งไคลเอนท์

ขั้นตอนการไป query ค่าเพื่อเปรียบเทียบนั้น จะเริ่ม เปรียบเทียบ จาก class แม่ ก่อน และไล่ไปตาม taxonomy ไปจนถึง โรคที่ระบุ โดยเราจะใช้ method ที่เก็บไว้ใน ตาราง facet เป็นตัวชี้นำในการค้นหาซึ่งเราจะรู้ คลาสแม่ที่จะเริ่มค้นหาได้จาก slot basic ที่อยู่ในเฟรม patient information1 ที่ได้มาจากทางฝั่งไคลเอนท์นั่นเอง จากนั้น ในส่วนของ slot symptom จะเป็นอาการของโรค ซึ่งจะเอาไปเปรียบเทียบกับ slot ของเฟรม อื่นๆ ที่เป็น เฟรมลูกต่อไป

เนื่องจากในการสร้าง trigger ขึ้นมานั้น มันจะทำงานกับทุกๆ row ของ ตาราง แต่ไม่สามารถกำหนดให้ 1 row มี trigger ต่างกันได้ ดังนี้ ในงานวิจัยนี้ เราจึงสร้าง trigger ให้ทำงานตาม condition ที่ให้มา ซึ่งใน condition จะประกอบไปด้วย if_needed, if_changed, constraint และ

default ซึ่งเป็นชนิดของ facet นั่นเอง จากนั้น ในส่วน attached procedure นั้นจะเก็บไว้ใน database ส่วนที่เป็น store procedure แล้ว rule ต่างๆ ที่จะกระทำเมื่อมีเหตุการณ์ต่างๆ จะเก็บเป็น ชื่อฟังก์ชั่นไว้ใน facet value แล้วตัว trigger จะมี condition เป็นตัวกำหนดว่าจะทำ execute หรือ retrieve ข้อมูลขึ้นมา ตามชื่อฟังก์ชั่น ซึ่งวิธีนี้จะทำให้เราสามารถ attached procedure ให้กับแต่ละทูเพิลได้

จากฝั่งไคลเอนท์จะส่งเฟรมข้างต้น ไปยังเซิร์ฟเวอร์เพื่อวินิจฉัยหาโรคที่เกิด ซึ่งทางฝั่งเซิร์ฟเวอร์จะไล่ตามอาการของโรคหู แล้วค้นหา ไปยังเฟรมต่างๆ ที่มีอาการที่เกี่ยวข้อง ซึ่ง จะตัดเฟรมที่มีอาการนอกเหนือจากที่ได้มาทิ้งไป นอกเสียจากว่า กลุ่มอาการโรคบางอย่าง ต้องอาศัยข้อมูลประวัติการแพ้ยา หรือประวัติการได้รับวัคซีนเป็นตัวประกอบการตัดสินใจด้วย ส่วนในตัวอย่างนี้ ตัววินิจฉัยจะไล่ไปยังเฟรมลูกของโรคหู อย่างเช่น โรคหูชั้นนอก ซึ่งพบว่ามีอาการเพิ่มเติม คือหูชั้นนอกบวมแดง ก็ตัดเฟรมนั้นทิ้ง ไปหาเฟรมใหม่ซึ่งเป็นเฟรมลูกของเฟรมโรคหูอีกที ซึ่งก็คือ เฟรมโรคหูชั้นกลาง แล้วพบว่า มีอาการเพิ่มเติมคือ แก้วหูอักเสบ ก็ตัดเฟรมนั้นทิ้งไปอีก จนมาถึงเฟรมสุดท้าย คือโรคหูชั้นใน แล้วจากเฟรมโรคหูชั้นใน ก็จะทำการวินิจฉัยต่อไปยังเฟรมลูกของโรคหูชั้นใน เหมือนกับที่วินิจฉัยก่อนหน้านี้ จนพบอาการ ซึ่งก็คือ โรค temporal bone tumor ซึ่งในกรณีนี้ โรคที่พบ มีเพียงโรคเดียว

5.4 การ map ข้อมูลไปเป็น kqml message ในตัวอย่างการพัฒนาระบบผู้เชี่ยวชาญสำหรับงานการวินิจฉัยโรค

ในกรณีตัวอย่างนี้สมมติผู้ป่วยมาด้วยอาการปวดหูเพียงอย่างเดียว โดยทดสอบดึงที่ใบหูแล้วพบว่ามีอาการเจ็บปวด ผู้ป่วยไม่มีอาการผิดปกติอย่างอื่นอีกเลย ทางฝั่งไคลเอนท์จะได้ข้อสรุปโดยสังเขปว่าผู้ป่วยมีอาการของโรคหูชั้นนอก แล้วจะทำการสร้างเฟรมคนผู้ป่วย ซึ่งจะมี slot ดังนี้ หมายเลขอ้างอิง ชื่อผู้ป่วย กลุ่มอาการ อาการที่แสดง

เฟรมที่ได้จากฝั่งไคลเอนท์นี้จะอยู่ในรูปแบบของ script ในภาษาจาวา และเมื่อเราจะทำการส่งไปยังเซิร์ฟเวอร์ที่เกี่ยวข้องกับอาการปวดหู เราต้องทำการเข้ารหัสเฟรมนี้ให้อยู่ในรูปแบบข้อความ KQML ดังนี้

(ask-one

:sender client1

:content “INSERT INTO patient VALUES ('12345', 'นายปลอดโรค แข็งแรงดี', 'Otalgia', 'earache', 'pain on pulling the auricle')”

:receiver server1

:reply-with earache_symptom

:language sql

:ontology earache)

ในข้อความนี้เนื้องานเป็นส่วนของเฟรมทางฝั่งไคลเอนท์ที่เราทำการส่งไป ontology คือขอบเขตของความรู้ ส่วนที่เราต้องการจะถามเกี่ยวกับเรื่องนั้นๆ ซึ่งในที่นี้เกี่ยวข้องกับอาการปวดหู ผู้ส่งคือ server1 และกระบวนการคิวรีถูกเขียนขึ้นโดยใช้ภาษาจาวา ค่าของ parameter ต่อไปนี้คือ sender, receiver และ reply-with จะเป็นส่วน communication layer ในส่วน performative name (ในที่นี้คือ ask-one), language และ ontology จะเป็นส่วน message layer และส่วนเนื้องานจะเป็นส่วน content layer ใน KQML

ทางฝั่งเซิร์ฟเวอร์จะเก็บข้อมูลของ patient ไว้ในเฟรม patient ต่างหาก และจะไล่ตามอาการของโรคหูแล้วค้นหาไปยังเฟรมต่างๆ ที่มีอาการที่เกี่ยวข้อง ซึ่งจะตัดเฟรมที่มีอาการนอกเหนือจากที่ได้มาทิ้งไป นอกเสียจากว่ากลุ่มอาการโรคบางอย่าง ต้องอาศัยข้อมูลประวัติการแพ้ยา หรือประวัติการได้รับวัคซีนเป็นตัวประกอบการตัดสินใจด้วย ส่วนในตัวอย่างนี้ตัววินิจฉัยจะไล่ไปยังเฟรมลูกของโรคหู เช่น โรคหูชั้นนอก ซึ่งพบว่ามีอาการเพิ่มเติม คือหูชั้นนอกบวมแดง ก็ตัดเฟรมนั้นทิ้ง ไปหาเฟรมใหม่ซึ่งเป็นเฟรมลูกของเฟรมโรคหูอีกที ซึ่งเป็นเฟรมโรคหูชั้นกลางแล้วพบว่ามีอาการเพิ่มเติมคือแก้วหูอักเสบ ก็ตัดเฟรมนั้นทิ้งไปอีก จนมาถึงเฟรมสุดท้ายเป็นเฟรมโรคหูชั้นใน แล้วจากเฟรมโรคหูชั้นในก็จะทำการวินิจฉัยต่อไปยังเฟรมลูกของโรคหูชั้นใน เหมือนกับที่วินิจฉัยก่อนหน้านี้ จนพบอาการซึ่งคือ โรค temporal bone tumor ซึ่งในกรณีนี้โรคที่พบ มีเพียงโรคเดียว เมื่อได้ผลลัพธ์แล้วทางฝั่งเซิร์ฟเวอร์ก็จะส่งกลับไปให้ไคลเอนท์ที่ทำการร้องขอ และจะทำการ encode ให้อยู่ในรูปข้อความ KQML ดังนี้

```
(tell
  :sender server1
    :content "temporal bone tumor"
    :receiver client1
    :in-reply-to earache_symptom
    :language sql
:ontology earache)
```

5.5 ผลการทดลอง

- เมื่อผู้ป่วยบอกอาการเบื้องต้นแล้ว ระบบจะดูว่าเป็นอาการเบื้องต้นแบบใด แล้วเรียกคำถามที่ต้องถามเกี่ยวกับโรคที่มีอาการเบื้องต้นนั้น
- เมื่อผู้ป่วยกรอกข้อมูลที่เกี่ยวข้องแล้ว จะทำการส่งค่าทั้งหมดไปยัง ฝั่งเซิร์ฟเวอร์เพื่อทำการวินิจฉัยแยกเฉพาะโรค
- เมื่อฝั่งเซิร์ฟเวอร์รับเฟรมที่บอกอาการต่างๆของโรค จะทำการเปรียบเทียบเพื่อหาอาการของโรค และแบ่งประเภทโรค ออกจากอาการนั้นๆ ได้ โดยไล่หาที่เฟรมของโรคต่างๆที่เกี่ยวกับหู แล้วสร้าง อินสแตนซ์ ขึ้นมาเพื่อเปรียบเทียบลักษณะอาการ จนได้โรคนั้นๆออกมา ซึ่งอาการโดยรวมต่างๆ จะ สืบทอด มาจากคลาสเฟรม
- ในกรณี ที่ผู้ป่วย อาจกรอกข้อมูลไม่ครบถ้วน เนื่องจากต้องอ้างอิงถึงประวัติเก่า เช่น ประวัติแพ้ยา หรือประวัติการให้วัคซีน ระบบจะทำการดึงข้อมูลเพิ่มเติม จากฐานข้อมูลอื่นๆ อย่างเช่น เวชระเบียน เพื่อทำการวินิจฉัยต่อไป
- เมื่อฝั่งเซิร์ฟเวอร์ทำการวินิจฉัย และสรุปผลการตรวจโรคแล้ว จะส่งผลลัพธ์ ของโรคที่น่าจะเป็นส่งไปให้ผู้ป่วย ที่อยู่บนฝั่งไคลเอนท์ได้

ในการทดลองระบบต้นแบบ จะทดลองการวินิจฉัยสองแบบคือแบบแรก ให้ผู้ใช้กรอกอาการของผู้ป่วย เท่าที่ทราบ แล้วให้ระบบทำการวินิจฉัยโดยอาจถามอาการของผู้ป่วยเพิ่มเติม หรือว่าไปเลือกข้อมูลขึ้นมาจากเวชระเบียน เพื่อทำการวินิจฉัย คำตอบที่ได้จะเป็น โรคที่มีโอกาสจะเกิด ส่วนอีกแบบคือ ผู้ใช้ป้อนชื่อโรคที่สงสัยว่าจะเป็นเข้ามา ระบบจะให้ผู้ใช้ตอบคำถามหรือไปถามจากระบบอื่นและส่งคำตอบกลับมาว่าผู้ใช้มีโอกาสเป็นโรคที่สงสัยหรือไม่

จากผลการทดลองแบบแรกโดยสมมติว่าผู้ป่วยมาด้วยโรคหูชั้นกลางอักเสบเฉียบพลัน ซึ่งอาการแรกเริ่มคือ ปวดหู โดยพบร่วมกับ เป็นไข้หวัด เจ็บคอ มีไข้สูง หูอื้อ หรือ เยื่อแก้วหูบวมแดง ซึ่งตอนแรกสุดผู้ใช้จะสามารถตอบคำถามได้แค่ว่า มีอาการหลักคือปวดหู และเป็นไข้ เจ็บคอ และมีขั้นตอนการวินิจฉัยดังนี้

ฝั่งไคลเอนต์

1. ระบบจะทำการเปรียบเทียบค่าที่ได้ป้อนเข้ามา กับเฟรมอาการแรกเริ่มที่มีอยู่จะได้มา สามเฟรมอยู่ในลิสต์ goal_list (คือเฟรมโรคหู เฟรมโรคไข้ และเฟรมเจ็บคอ)
2. จากนั้นจะเลือกเฟรมโรคหู (B2 จากตารางที่ 5.2) ขึ้นมาก่อนและตั้งเฟรมโรคหูให้เป็นเป้าหมาย
3. เมื่อได้เป้าหมายแล้วก็มาดูอาการของโรคหูหลักๆก่อน
4. จากเฟรมเป้าหมาย B2 ระบบจะให้ผู้ใช้ตอบคำถามว่า ผู้ป่วยมีอาการบาดเจ็บทางศีรษะหรือไม่
5. ซึ่งในตัวอย่างนี้จะตอบว่าไม่

6. และในเฟรมเป้าหมายนี้ ก็จะให้ผู้ใช้ตอบคำถามอีกหนึ่งคำถามว่าผู้ป่วยมีอาการปวดเวลาดึงใบหูหรือไม่
7. ซึ่งในตัวอย่างนี้จะตอบว่าไม่
8. และทริกเกอร์ในสล๊อตนี้ไปเรียกเฟรมถัดมา นั่นก็คือเฟรมอาการปวดหูที่มีสาเหตุจากโรคหูเป็นหลักขึ้นมา (B4 จากตารางที่ 5.2)
9. จากนั้นก็ให้ผู้ใช้กรอกอาการเพิ่มเติม ซึ่งก็คือ เยื่อแก้วหูอักเสบ
10. จากนั้นเมื่อระบบรู้แล้วว่าอาการที่เป็นมีสาเหตุมาจากโรคหูก็ส่งข้อมูลผู้ป่วย พร้อมทั้งอาการที่แสดง ที่ผู้ใช้ได้ป้อนเข้ามาไปยังฝั่งเซิร์ฟเวอร์

frame	slot	slot value
Patient	reference number	12345
	Patient Name	นายปลอดโรค แข็งแรงดี
	Symptoms	Primary Otalgia
	Earache	yes
	Pain on pulling the auricles	yes
	Redness of the eardrum	yes

รูปที่ 5.3 ตัวอย่างเฟรมผู้ป่วยที่ฝั่งไคลเอนท์

ฝั่งเซิร์ฟเวอร์

1. ได้รับข้อมูลว่าเป็นอาการปวดหูที่เกี่ยวข้องกับโรคหู และมีอาการปวดเวลาดึงใบหู พร้อมทั้งมีอาการเยื่อแก้วหูอักเสบ
2. ระบบที่ฝั่งเซิร์ฟเวอร์ จะเปรียบเทียบอาการ โดยเลือกเฟรมเป้าหมายมาไว้ในลิสต์ goal_list ก่อน
3. ซึ่งในตัวอย่างนี้จากเฟรมโรคหู จะมีเฟรมลูกอยู่สามเฟรม ซึ่งก็คือ เฟรมโรคหูชั้นนอก เฟรมโรคหูชั้นกลาง และเฟรมโรคหูชั้นใน (E2, E3, E4 ในตารางที่ 5.1 ตามลำดับ)
4. ในตอนแรกระบบจะทำการเลือกเฟรมโรคหูชั้นนอกคือ E2 ออกมาจาก goal_list ก่อน
5. โดยเฟรม E2 จะมีสล๊อตเพิ่มมาอีกหนึ่งสล๊อตคือ หูชั้นนอกอักเสบบวมแดง
6. ระบบจะทำการเปรียบเทียบค่าของสล๊อตกับอาการที่ถูกส่งเข้ามาที่ฝั่งเซิร์ฟเวอร์
7. ในตัวอย่างนี้จะไม่พบอาการดังกล่าว ก็จะปฏิเสธสล๊อตนี้แล้วไปเลือกเฟรมเป้าหมายใหม่ ซึ่งก็คือเฟรม E3 มาแทน

8. เมื่อได้เฟรมโรคหูชั้นกลางแล้ว (E3) ระบบก็จะเปรียบเทียบสล๊อตในเฟรม E3 กับค่าที่ถูกส่งเข้ามาซึ่งในตัวอย่างนี้จะมีค่าเป็น yes
9. ระบบจะทำการตรวจสอบว่าเฟรมที่ได้มานี้มีเฟรมลูกอีกหรือไม่
10. ซึ่งในตัวอย่างนี้เฟรม E3 จะมีเฟรมลูกคือ (E13, E14, E15 จากตารางที่ 5.1 ตามลำดับ)
11. ระบบจะทำการเก็บเฟรมนั้นเพิ่มเข้าไปในลิสต์เป้าหมาย goal_list ในลักษณะของสแต็ก (คือเข้าไปที่หลังออกมาก่อน)
12. ระบบจะเลือกเฟรม E13 ขึ้นมาก่อนแล้วทำการจับคู่สล๊อตของเฟรม กับค่าที่มี
13. จากตัวอย่างนี้ เฟรม E13 จะมีสองสล๊อต ซึ่งมีค่าทั้งสองสล๊อต
14. และเมื่อระบบทำการตรวจสอบ ก็พบว่าเฟรม E13 ไม่มีเฟรมลูกอีกแล้ว
15. ระบบก็จะส่งเฟรมที่ได้ กลับไปยังฝั่งไคล์เอนท์ เป็นคำตอบโรคที่มีโอกาสจะเกิดกลับไป ซึ่งก็คือเฟรม E13 หรือโรคหูชั้นกลางอักเสบเฉียบพลัน

จากผลการทดลองส่วนแรกนี้ ทำให้พบว่าระบบมีความสามารถในการวินิจฉัยหาโรคที่เกี่ยวข้องกับอาการได้ และสามารถส่งผลลัพธ์ที่ถูกต้องกลับไปยังฝั่งไคลเอนท์ได้ถูกต้อง แต่ผู้ใช้จะต้องป้อนข้อมูลเข้ามาในระบบพอสมควรก่อน จึงจะสามารถทำการวินิจฉัยต่อไปได้ ถ้าผู้ใช้ไม่ป้อนข้อมูลเข้ามาในระบบเลย แต่ถามถึงโรคที่มีโอกาสจะเกิด ระบบจะไม่สามารถตอบคำถามของผู้ใช้ได้ เพราะระบบไม่มีข้อมูลที่จะใช้เพื่อตั้งเป็นเป้าหมายได้

ดังนั้นจากผลการทดลองจะได้ว่า ระบบจะทำการวินิจฉัยได้ ก็ต่อเมื่อมีข้อเท็จจริงเริ่มต้นบางส่วนถูกป้อนเข้าไปในระบบก่อน จากนั้นระบบถึงจะสามารถทำการวินิจฉัยแล้วทำการหาคำตอบต่อไปได้ หรือสามารถถามคำถามเพิ่มเติมกับผู้ใช้ได้ หรือไปหาข้อเท็จจริงที่ผู้ในฐานข้อมูลภายนอกอื่นได้

ในการทดลองการวินิจฉัยแบบที่สองในกรณีที่ผู้ใช้ไม่พอใจผลการวินิจฉัย หรือต้องการจะทราบว่าผู้ใช้มีโอกาสจะเป็นโรคที่ระบุหรือไม่ สมมติว่าผู้ใช้ต้องการจะทราบว่ามีโอกาสจะเกิดโรคเนื้องอกในกระดูกหูชั้นในหรือไม่ก็จะมีขั้นตอนดังนี้

ฝั่งไคลเอนต์

1. ผู้ใช้จะทำการป้อนอาการหลักของโรคก่อน ซึ่งก็คือมาด้วยอาการปวดหู
2. ซึ่งฝั่งไคลเอนต์จะส่งอาการปวดหูไปที่ฝั่งเซิร์ฟเวอร์ ให้ส่งค่าเฟรมทั้งหมดของโรคที่เกี่ยวข้องกับอาการปวดหูกลับมา ซึ่งก็คือเฟรม E1-E17 มีด้วยกัน 17 เฟรม
3. ผู้ใช้จะเลือกเฟรมขึ้นมาจากในลิสต์ ซึ่งในตัวอย่างนี้ผู้ใช้จะเลือกเฟรม E4 โรคหูชั้นในขึ้นมา
4. ฝั่งไคลเอนต์ก็จะส่งค่าโรคหูชั้นในไปยังฝั่งเซิร์ฟเวอร์ เพื่อให้ฝั่งเซิร์ฟเวอร์ส่งค่าเฟรมทั้งหมดของโรคที่เกี่ยวข้องกับเฟรม E4 กลับมา
5. ผู้ใช้จะทำการเลือกอีกครั้งซึ่งในตัวอย่างนี้ผู้ใช้จะทำการเลือกเฟรมโรคเนื้องอกในกระดูกหูชั้นใน (E17) ขึ้นมา

6. เฟรม E17 จะนำผู้ใช้ให้กรอกข้อมูลในแต่ละสล๊อต
7. ซึ่งในตัวอย่างนี้ เฟรม E17 จะสืบทอดสล๊อตทั้งหมดจากคลาสแม่ มายังเฟรมนี้ด้วย
8. ซึ่งจากเฟรมตัวอย่างนี้จะมีอยู่สองสล๊อต คือปวดหู กับปวดเวลาดึงใบหู
9. และผู้ใช้ได้กรอกข้อมูลทั้งหมดครบถ้วนซึ่งสอดคล้องกับค่าในสล๊อต
10. ระบบจะทำการส่งค่ากลับไปยังผู้ใช้ว่า มีโอกาสเป็นโรคที่สงสัยข้างต้น คือโรคเนื้องอกที่กระดูกหู แน่นอน

จากผลการวินิจฉัยในแบบที่สองนี้ ผลลัพธ์ที่ได้นั้นแม้ว่าการวินิจฉัยตามอาการนั้นถูกต้องก็ตามแต่ก็ถือว่าวินิจฉัยไม่ตรงตามที่ผู้เชี่ยวชาญจะทำการวินิจฉัย เพราะโอกาสที่ผู้ป่วยจะเป็นโรคชนิดนี้นั้นน้อยมาก ผลการวิจัยนี้จึงทำให้ทราบว่า ระบบควรทำการปรับปรุง และพัฒนาให้สามารถทำการวินิจฉัยโดยขึ้นอยู่กับความน่าเป็นที่จะเกิดคำตอบนั้นขึ้นด้วย และจากผลการทดลองในการวินิจฉัยแบบที่สองนี้จะได้ว่า ผู้ใช้จะต้องทำการป้อนข้อมูลอาการหลักของโรคที่สงสัยเสียก่อน เพื่อลดขอบเขตของเฟรมโรคที่จะต้องค้นหา ซึ่งถ้าผู้ใช้ไม่ทำการกรอกข้อมูลอาการหลักของโรคเลย ระบบจะไม่สามารถไปดึงเฟรมโรคของแต่ละฐานความรู้ที่อยู่กระจัดกระจายขึ้นมาได้ เพราะระบบต้นแบบจะจัดเก็บความรู้เกี่ยวกับโรคหู โรคไข้ โรคเจ็บคอ และอื่นๆ ไว้ในแต่ละเซิร์ฟเวอร์แยกกัน ดังนั้นผู้ใช้จำเป็นต้องป้อนข้อมูลอาการหลักของโรคก่อน เพื่อระบุขอบเขตของฐานความรู้และสามารถเลือกโรคที่อยู่ในขอบเขตนั้นๆ เพื่อทำการวินิจฉัยได้

บทที่ 6
สรุปผล และข้อเสนอแนะ

จากวัตถุประสงค์หลักของงานวิจัยนี้ คือต้องการพัฒนาระบบผู้เชี่ยวชาญแบบเฟรมด้วยการใช้ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์สร้างฐานความรู้ โดยผู้วิจัยได้นำเอาระบบการวินิจฉัยโรคมาสร้างเป็นระบบผู้เชี่ยวชาญ เพื่อเป็นกรณีศึกษา ให้กับแนวคิดที่ผู้วิจัยได้นำเสนอไว้ ซึ่งจากการดำเนินการวิจัยทำให้ผู้วิจัยสามารถสรุปผล และข้อเสนอแนะเพื่อเป็นแนวทางในการนำแนวคิดจากงานวิจัยนี้ไปพัฒนาต่อในอนาคต โดยมีรายละเอียดดังต่อไปนี้

6.1 สรุปผลการวิจัย

งานวิจัยนี้นำเสนอสถาปัตยกรรมในการออกแบบและสร้างระบบผู้เชี่ยวชาญแบบเฟรมบนระบบฐานข้อมูลเชิงวัตถุสัมพันธ์และต้องการขยายความสามารถของระบบผู้เชี่ยวชาญให้มีความสามารถในการวินิจฉัยได้สองทางซึ่งจากสมติฐานพบว่าสามารถพัฒนาระบบผู้เชี่ยวชาญโดยใช้การแทนความรู้แบบเฟรมบนระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ได้

จากการเปรียบเทียบหลักการเชิงวัตถุกับเฟรมนั้นพบว่า หลักการเชิงวัตถุกับเฟรมนั้นพบว่า หลักการเชิงวัตถุกับเฟรมมีความคล้ายคลึงกันมาก และสามารถทำมาใช้ในหลักการของเฟรมได้ โดยหลักการของเฟรมนั้นจะแตกต่างในการมองปัญหา และการจัดลำดับโครงสร้างแต่หลักการของเฟรมจะไม่มีคุณสมบัติการห่อหุ้ม เหมือนในหลักการเชิงวัตถุ ดังนั้นงานวิจัยนี้จึงสามารถใช้หลักการเชิงวัตถุสัมพันธ์ ในการแทนความรู้แบบเฟรมได้

ผลการทดลองในระบบต้นแบบที่พัฒนาสามารถทำการวินิจฉัยสองทางได้และสามารถใช้งานได้ตรงตามสมมติฐานโดยระบบผู้เชี่ยวชาญสำหรับงานการวินิจฉัยโรคพบว่าระบบสามารถใช้งานได้ตามสมมติฐานที่ตั้งไว้โดยระบบต้นแบบนี้ได้ศึกษาจากผู้เชี่ยวชาญตำแหน่งนายแพทย์แล้วระบบต้นสามารถตอบคำถามในลักษณะของผู้เชี่ยวชาญได้ แต่ยังไม่สามารถทำการวินิจฉัยได้ถูกต้องร้อยเปอร์เซ็นต์ตามแบบผู้เชี่ยวชาญ

งานวิจัยนี้พัฒนาระบบผู้เชี่ยวชาญบนระบบการจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์ออราเคิล 10g จากการทดลองในระบบต้นแบบระบบการจัดการฐานความรู้ออราเคิล 10g นั้นมีความสามารถในการบริหารจัดการข้อมูลได้ในระบบต้นแบบและมีแนวโน้มที่จะสามารถจัดการกับฐานข้อมูลที่มีขนาดใหญ่ได้ในอนาคตและระบบต้นแบบสามารถที่จะทำงานได้จึงมีความเป็นไปได้ในการที่จะพัฒนาระบบผู้เชี่ยวชาญแบบเฟรมบนฐานข้อมูลเชิงวัตถุสัมพันธ์นี้ให้จัดเก็บข้อมูลที่มากขึ้นได้

จากการทดลองสมมติฐานดังกล่าวสามารถสรุปได้ว่าแนวคิดในการสร้างระบบผู้เชี่ยวชาญแบบเฟรมบนระบบการจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์ใช้งานได้และสามารถพัฒนาระบบต้นแบบให้มีขนาดเพิ่มมากขึ้นได้โดยระบบต้นแบบจะสามารถลดภาระการสร้างส่วนบริหารจัดการข้อมูลซึ่งเป็นการใช้จุดเด่นของระบบฐานข้อมูลที่มีอยู่แล้วอีกด้วยและยังสามารถเชื่อมโยงเข้ากับระบบฐานข้อมูลอื่นที่มีใช้อยู่ทั่วไปได้ง่ายเพราะเป็นการเชื่อมต่อระหว่างฐานข้อมูลและทำให้สามารถพัฒนาระบบให้ใช้ฐานข้อมูลที่มีอยู่แล้วให้มีประสิทธิภาพซึ่งจะลดภาระการเปลี่ยนแปลงใหม่ทั้งระบบอีกทั้งลดปัญหาที่จะตามมาในการแก้ไขระบบฐานข้อมูลเดิมที่มีอยู่แล้วด้วย นอกจากนี้ยังได้มีการนำโปรโตคอลที่ใช้ในการแลกเปลี่ยนข้อมูลระหว่างระบบฐานความรู้มาประยุกต์ใช้ในการติดต่อสื่อสารระหว่างไคลเอนท์กับเซิร์ฟเวอร์ด้วยทำให้รองรับการใช้งานในอนาคตได้เป็นอย่างดี

6.2 ข้อเสนอแนะ

งานวิจัยนี้ได้มีการจัดเก็บเฟรมไว้ที่ฝั่งไคลเอนต์และฝั่งเซิร์ฟเวอร์โดยที่ได้ทำการจัดเก็บเฟรมที่ฝั่งไคลเอนต์ไว้ในรูปแบบเท็กซ์ไฟล์ธรรมดาซึ่งสามารถประยุกต์ใช้รูปแบบ XML ในการจัดเก็บเฟรมบนฝั่งไคลเอนต์ได้ในอนาคต

นอกจากนี้เนื่องจากในงานวิจัยนี้มีการวินิจฉัยได้สองทางซึ่งฝั่งไคลเอนต์จะเป็นส่วนติดต่อกับผู้ใช้และส่งข้อมูลไปยังเซิร์ฟเวอร์เพื่อทำการวินิจฉัยต่อในลักษณะที่ไคลเอนต์จะรู้อยู่แล้วว่าควรจะติดต่อไปยังเซิร์ฟเวอร์ไหนซึ่งสามารถพัฒนาต่อยอดให้ไคลเอนต์สามารถค้นหาได้โดยอัตโนมัติว่ามีเซิร์ฟเวอร์ไหนบ้างที่สามารถทำการวินิจฉัยต่อได้

ในงานวิจัยนี้ระบบต้นแบบนั้นถูกออกแบบเพื่อทดสอบสมมติฐานเท่านั้น ซึ่งในการพัฒนาระบบการวินิจฉัยโรคให้มีประสิทธิภาพมากขึ้นสามารถออกแบบให้มีประสิทธิภาพมากขึ้นได้ ขึ้นอยู่กับวิศวกรความรู้

เอกสารอ้างอิง

[1] Natalya F. Noy, Mark A. Musen, Jose L.V. Mejino, Cornelius Rosses. "Pushing the Envelope : Challenges in a Frame-Based Representation of Human Anatomy" **Data & Knowledge Engineering**, Volume 48 Issue 3 (ACM), March 2004.

[2] Benjamin Kuipers. "Algernon for expert system" **Draft document in Computer Science Department University of Texas at Austin**, 18 January 1994.

[3] Gang Luo and Vinay K. Chaudhri. "Implementing OKBC Knowledge Model Using Object Relational Capabilities of Oracle 8", **Technical Report.**

[4] P.D. Karp et al. **The EcoCyc Database.** Nucleic Acids Research. Vol. 30. No. 1 56-58. : Oxford University Press. 2002.

[5] Neil F. Abernethy, Russ B. Altman. "Sophia : Providing Basic Knowledge Service with A Common DBMS" **Proceedings of the 5th KRDB Workshop**, 1998.

[6] Tim Finin, Richard Fritzson, Don McKay, and Robin McEntire. "KQML as an Agent Communication Language" **In Proceeding of the 3rd International Conference on Information and Knowledge Management**, 1994. Pp. 456 - 463.

[7] **Oracle Documentation**, http://otn.oracle.com/documentation/content.html

[8] Peter Lucas and Linda van der gang. **Principle of expert systems.** : Addison-Wesley. 1991.

[9] C.S. Krishnamoorthy; S. Rajeev. **Artificial Intelligence and Expert Systems for Engineers.** : CRC Press. 1996.

[10] Stuart E. Savory. **Expert systems for professionals.** : ELLIS HORWOOD Limited. 1990.

[11] Peter D. Karp. **The Design Space of Frame Knowledge Representation System**. SRI AI Center Technical Note #520. 1993.

[12] Negnevitsky M. **Artificial Intelligence: A Guide to Intelligent Systems**. Harlow, England : Addison Wesley. 2002.

[13] Marvin Minsky. **A Framework for Representing Knowledge**. Reprinted in The Psychology of Computer Vision, P. Winston (Ed.) : McGraw-Hill. 1975.

[14] Durkin J. **Expert Systems : Design and Development**. Macmillan Inc. 1994.

[15] Richard Fikes and Tom KehLer. "The Role of frame-based representation in reasoning" **Communications of the ACM**, 28(9), 1985.

[16] Kamran Parsaye. **Expert systems for experts**. John Wiley & Sons Inc. 1988.

[17] A.K. Sharma et al. "A FUZZY FRAME BASED EXPERT SHELL" **Proceedings: National Workshop on IT Services and Applications (WITSA2003),** Feb 27-28. 2003.

[18] E. F. Codd. "A Relational Model of Data for Large Shared Data Banks" **Reprinted from Communications of the ACM,** Vol. 13, No. 6, June 1970, Pp. 377-387.

[19] Jim Melton, Nelson Mendoca Mattos. "An Overview of the Emerging Third-Generation SQL Standard" **Proceedings of the 1995 ACM SIGMOD international conference on Management of data** , 1995. Pp. 468.

[20] Steve McClure. "**Object Database vs. Object-Relational Database**". IDC Bulletin #14821E – August 1997.

[21] Martin Rennhackkamp. **Extending Relational DBMSs.** Miller. : Freeman,Inc. 1999.

[22] Chao Chen, Philip D. Udo-Inyang, Frederick C. Schmitt. **Integration of a Database Management System and a Knowledge-Based Expert System in Construction: A Review,** 1994.

[23] M. C. Norrie, U. Reimer, P. Lippuner, M. Rys, H. J. Schek. "Frames, Objects and Relations : Three Semantic Levels for Knowledge Base Systems" **Swiss Priority Programme in Computer Science,** No. 5003-34347, 1995.

[24] M.P. Evett, J.A. Hendler, and L. Spector. "Parallel Knowledge Representation on the Connection Machine" **Journal of Parallel and Distributed Computing,** Vol.22, 1994. Pp.168-184.

[25] Nijssen, G.M. and Halpin, T.A. **Conceptual Schema and Relational Database Design A Fact Oriented Approach.** : Prentice Hall. 1989.

[26] Pasaya, B. Chittayasothorn, S. "A temporal object oriented conceptual schema model" **Conference on Communication, Computers and signal Processing, PACRIM.** 2001.

[27] Puntheeranurak, S. Chittayasothorn, S. "An extended NIAM conceptual schema model for object databases" **Proceeding of the 24th International Conference on Information Technology Interfaces, ITI.** 2002.

[28] Patrick Henry Winston. **Artificial Intelligence.** : Addision Wesley. 1992.

[29] Dimitris Metaxas and Timos Sellis. "A database implementation for large frame-based systems" **Second International Conference on Data and Knowledge Systems for Manufacturing and Engineering,** 1989.

[30] Neil F. Abernethy, Julie J. Wu, Micheal Hewett, and Russ B Altman. "Sophia: A Flexible, Web-Based Knowledge Server" **Stanford University Medical Center,** 1999.

ภาคผนวก

ผลงานวิจัยในระหว่างการศึกษาที่ได้รับการตีพิมพ์เผยแพร่

[1]. Chuleerat Rattanaprateep, Suphamit Chittayasothorn, "Expert Database System Architecture and Implementation on Object Relational Databases" **Journal of WSEAS TRANSACTIONS on COMPUTERS**, Issue 3, Volume 5, March 2006.

WSEAS TRANSACTIONS on COMPUTERS

Issue 3, Volume 5, March 2006
ISSN 1109-2750 http://www.wseas.org

A Novel Fast Feature Based Stereo Matching Algorithm with Low Invalid Matching 469
Payman Moallem, Mohsen Ashourian, Behzad Mirzaeian, Mohamad Ataei

A Novel Cluster Validity Index: Variance of the Nearest Neighbor Distance 477
Ferenc Kovacs, Renata Ivancsy

Meta-scheduler in Grid environment with multiple objectives by using genetic algorithm 484
Siriluck Lorpunmanee, Mohd Noor Md Sap, Abdul Hanan Abdullah, Surat Srinoy

Research of supply chain plant location considered the bill of materials and time restriction 492
Shihua Ma, Xiaoqun Liu

Some Succinctness Properties of Ω-DTAFA 500
A. Feliah and S. Noureddine

Approaching Software Process Improvement to Organizations 507
Antonio Amescua, Javier Garcia, Maria-Isabel Sánchez-Segura, Fuensanta Medina-Dominguez

An hybrid model of MCDA for the GIS: Application to the localization of a site for the implantation of a dam 515
Abdessadek Tikniouine, Abdelaziz Elfazziki, Tarik Agouti

The Rigid-Flexible Coupling Dynamics Characteristic Between Mobile Robot Along Overhang Flexible Cable and its Moving Path 521
Xiao Xiaohui, Gongping Wu, Li Sanping

New Polynomial Families for Generating More Suitable Elliptic Curve for Pairing-Based Cryptosystems 528
Pu Duan, Shi Cui, Choong Wah Chan

Comparative Study of B-Spline and Cardinal Spline with Genetic Algorithm for Invariant Shape Object Recognition 536
Pisit Phokharatkul, Skul Kamnuanchai, Chom Kimpan, Supachai Phaiboon

Improving Wormhole Adaptive Routing in Networks on Chip 544
Giuseppe Ascia, Vincenzo Catania, Maurizio Palesi, Davide Patti

Arabic Proper Names: Extraction and Classification 552
Saleem Abuleil

Expert Database System Architecture and Implementation on Object Relational Databases 560
Chuleerat Rattanaprateep, Suphamit Chittayasothorn

Prospects for Fixed-mobile Convergence in Asia Pacific Region 568
Chin Chin Wong, Andy L. Y. Low, Pang Leang Hiew

560 WSEAS TRANSACTIONS ON COMPUTERS Issue 3, Vol. 5, March 2006 ISSN: 1109-2750

Expert Database System Architecture and Implementation on Object Relational Databases

CHULEERAT RATTANAPRATEEP*, SUPHAMIT CHITTAYASOTHORN**
Department of Computer Engineering Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520
THAILAND
chuleerat_r@yahoo.com*, suphamit@kmitl.ac.th**

Abstract: An expert database system is an expert system which is developed on databases using DBMS technology to manage facts and rules. Frame-based expert systems are widely used as the knowledge representation for such expert systems with large knowledge bases. Many systems have the ability to connect to external databases. Facts stored in databases can be loaded into expert system's knowledge base and inference is performed by the inference engine of the expert system. In many cases, such external facts are required several times for each inference. Thus, a lot of communication traffic takes place. This research work present the design and implementation of a frame-based object-relational database system which has a tight coupling between the expert system and the external knowledge base. The external knowledge base also use frame as its knowledge representation. More over, it has its own inference engine so that inference can be perform on the knowledge base side and the results, not only simple facts, are sent back to the expert system for further inference. A medical consultation system is used as an illustrated example an Oracle 10g object relational DBMS is used as the database platform.

Key-Words: - Frame, Expert System, Object-Relational Database

1 Introduction

Frames are widely used as the knowledge representation of large, complex expert systems [1]. However, most such expert system shells have internal frame-based knowledge bases. They are internal in the sense that frames are loaded and stored in the main memory of the expert system during consultation sessions. The knowledge bases do not have the advanced data management facilities such as indexing, query optimization, concurrency control and recovery control which are common in modern database management systems (DBMS).

During a consulting session, there are facts that are obtained from the user interactively and facts that are obtained from inferences. Inference rules in conventional expert systems are executed by the inference engine on the expert system's machine. It mainly uses facts obtained interactively from users. Facts from external databases are sometimes loaded into the knowledge base when required by the inference process. In simple systems, relations that contain both relevant and irrelevant facts are loaded into the knowledge base. In more advanced systems only related facts are loaded. In both cases, there are no inferences on the external database (or knowledge base) side.

Our approach is different. We propose an architecture that includes an inference engine on the external knowledge base side as well as one on the expert system shell side. This approach enables inferences to be performed on the external knowledge base side so that only the inference results are sent back to the expert system instead of sending facts several times during an inference process performed by the expert system's inference engine.

Due to the similarity between the frame and the object relational concepts, a prototype system is implemented on an object relational database using Oracle 10g DBMS.

2 Frames

Frame was introduced by Marvin Minsky in 1974 [2]. It is a knowledge representation that has both the data structure and inference capability. It is suitable for the representation of concepts and classifications. It is also suitable for the representation of a taxonomy hierarchy [3][4].

A frame comprises a frame name, slots or attributes of the frame and facets [5][6]. For clarification, frames can be classified into class frames and instance frames. Class frames are used to describe group of objects or class of objects and can also be organized into taxonomy. A class frame therefore has parent and children as common slots. Slots from a parent frame can be inherited to its children. Instance frames, on the other hand, describe particular object instances. They are the leaf node of the taxonomy and have no children [4][5][6][7][8].

Facets are used to control slot values and corresponding operations. It can be used to establish initial slot value, slot data type, possible value range and next activity to be performed. Validation rules, trigger operations and derivation rules are common facets as well.

Frames are similar to objects in object programming and databases but the most important difference between them is the former lacks the encapsulation property. Objects are encapsulated but frames are not. Attributes of an object class cannot be seen from object users. They can only see method signatures. Objects without encapsulation which is well known under the name "object relational" are therefore a perfect match of the frame concept. This is the reason why an object relational DBMS is employed in this research project.

3 Coupling between expert system and databases

As mentioned earlier, the coupling between an expert system and a database system extends the capability of the former to have access to larger databases [9]. There are several systems that have the capability such as the Perk database [10], EcoCyc [9], PARKA-DB [11] and Sophia [12]. All of them employ relational databases to record facts.

The Perk database connects to a database using OKBC operations [10] and refers to frames in the database by loading frames into the main memory for inferences. Its later versions employ object views and uses indexes to point to the required frames. EcoCyc system and PARKA-DB keep frames in relational database and load them into the main memory when required. Sophia also keeps frames in relational database and use SQL query for frames loading. All of them do not have the inference capability on the database side. Only tuples of relations are transfer to the expert systems.

In the following section, we propose an architecture which employs an inference engine on the database side. Frames are kept in an object relational database. The expert system can request derived facts which can be obtained as a result from the inference that takes place on the database side. Since the database side has frames and the inference capability, we therefore call it an external knowledge base.

4 The System Architecture

Fig.1 shows the architecture of the FORXDB (Frame-based Object Relational Expert Database system). The novel feature of this architecture is that it has an inference engine on the knowledge base side (the server side) as well as the client side. Frames are on both the knowledge server side and expert system (client) side. The client-side frames are those that involve user interaction and fact acquisition. Inference on this side can lead to a reference to the facts on the external knowledge server that can be inferred from other server-side frames.

The client expert system side comprises the user interfaces both for the expert and the user, the frame-based internal knowledge base, the knowledge base maintenance tool and the client-side inference engine. The external knowledge server comprises an inference engine, an object relational DBMS, the frame-based knowledge base and other databases that belong to other information systems. These systems are the data sources of the external knowledge base.

In a consultation session, the user consults the client-side frame-based expert system. The main purpose of this client-side inference is to gather current information on the subject matter. The inference rule in the first frame will choose the most suitable next frame to go to. Several frames may be visited on this client side before the client expert system gathers enough information which are to be sent to the knowledge base system on the server side for further inference.

Inference on the external knowledge base side focuses on using existing facts which are already stored in the knowledge base and other relevant information systems. The result of this knowledge base inference is sent back to the client-side expert system and finally to the user.

562 WSEAS TRANSACTIONS ON COMPUTERS Issue 3, Vol. 5, March 2006 ISSN: 1109-2750

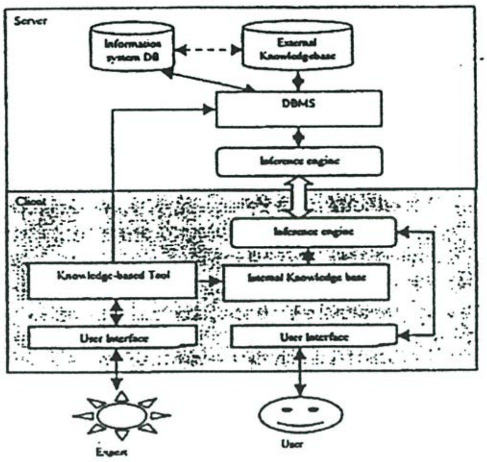


Fig. 1 The FORXDB System Architecture

5 Implementation Issues

5.1 The Meta Tables

The FORXDB on frames is kept in system tables. Fig. 2 shows an Object Role Model (ORM) [13] diagram that describes frames. Corresponding meta tables with some sample data are shown in Fig. 3.

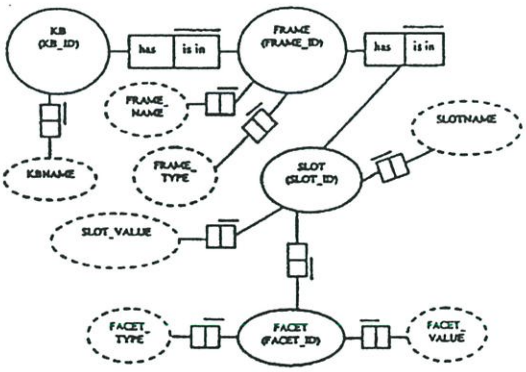


Fig.2 An ORM schema that describe frames

KB

KB ID	KBNAME
100	Sensor

FRAME

FRAME ID	KB ID	FRAME NAME	FRAME TYPE
1	100	Thermostat	Class
2	100	Thermostat1	Instance

SLOT

SLOT ID	FRAME ID	SLOTNAME	SLOT VALUE
1	1	Air Conditioner	
2	1	Furnance	
3	1	Mode	
4	1	Setting	
5	1	Temperature	
6	1	Room	
7	2	Parent	Thermostat
9	1	Children	Thermostat1

FACET

FACET ID	SLOT ID	FACET TYPE	FACET VALUE
1	3	Default	Heat
2	4	Default	68
3	5	Default	65

Fig.3 Meta tables obtained from Fig.2 and some sample data

5.2 External Database Data Source

In this project, data from existing information systems are used as the data source to the FORXDB system. Inferences are performed on frames which refer to other frames until a fact is found in a database. Methods are employed in order to refer to facts on the data source without copying them permanently to the frames. Thus avoid data inconsistency when updates are made to the data source.

In this project, Oracle 10g object-relational DBMS is used to manage the external knowledge base. Since Oracle 10g has a feature called dblink which allows an Oracle DBMS to refer to other Oracle databases and use SQL data manipulation statements directly on them. In the case that the external databases are not Oracle, a utility called OCA (Oracle Open Client Adapter) can be used to retrieve facts from them. In the implementation of the FORXDB system, only fact retrieval is required so we are well-equipped with data access tools.

Facts which can be obtained from an external source are predefined by the expert. For example, patient records in hospital information systems are needed for a medical expert system consultation. If the required records are not available, then the expert system will ask the user interactively. Frames that interact with external databases are instance frames. Their corresponding class frames have attached procedures in the facet. The instance frames actually inherit these procedures from them.

5.3 Object Relational Database

An object relational database supports non-atomic attributes. A table needs not represent a relation in the traditional sense. Multi-valued attributes, composite attributes and simple attributes are allowed. User-defined data types (UDT) are also allowed. These UDT comprises other attributes and methods and are fully encapsulated.

The concept of types and instances are clearly distinct. Unlike some simple relational database implementation that mixed the concepts of relational schema and relation in to a table, object relational allowed types (or row types) to be separately declared from table instances. Fig.4a shows a sample table Student with a composite attribute NAME and a multi-valued attribute COURSE.

Student

ID	NAME		COURSE
	FIRST	LAST	
st031	Jane	Hunter	Economy
			Planning
st072	Richard	White	Computers in Engineering

Fig 4 An object relational table

The corresponding row type and table definitions together with some sample insertions are shown below:

```
CREATE ROW TYPE Student (
 id CHAR(5),
 name ROW (first VARCHAR(12), last
VARCHAR(20)),
 course SET (VARCHAR(128) NOT NULL)
);
CREATE TABLE students OF TYPE Student;
INSERT INTO students
VALUES (
 'st031',
 ROW('Jane', 'Hunter'),
 SET('Economy', 'Planning')
);
INSERT INTO students
VALUES (
 'st072',
 ROW('Richard', 'White'),
 SET('Computers in Engineering')
);
```

Subtype hierarchies and methods are also supported. The following SQL codes show the creation of the super type EMPLOYEE and subtypes PROGRAMMER and REPRESENTATIVE.

```
CREATE TYPE EMPLOYEE AS OBJECT (
 NAME    VARCHAR2(20),
 SALARY  NUMBER(6,2)
) NOT FINAL;
CREATE TYPE PROGRAMMER UNDER
EMPOLYEE (
 LANGUAGE  VARCHAR2(12),
 PROJECT   VARCHAR2(30)
);
CREATE TYPE REPRESENTATIVE UNDER
EMPLOYEE (
 REGION  VARCHAR2(30)
);
CREATE TABLE employees OF EMPLOYEE;
CREATE TABLE programmers OF
PROGRAMMER;
CREATE TABLE representatives OF
REPRESENTATIVE;
```

The object relational table is neither a relation nor an object since it allows non-atomic attribute without encapsulation. The encapsulated part is a user-defined data type.

6 A Prototype Medical Diagnosis Expert System

Medical diagnosis is a process that requires skilled and qualified physicians. In remote areas where doctors are not available when required, other less qualified health care personals may have to do the job. A medical diagnosis expert system will be an invaluable tool in such situations. Patient's records, symptoms and illness history are essential information for the diagnosis process. It is clear that some information should be available on external databases (such as the patient's records) and others have to be obtained from patient's interviews.

A frame-based expert system uses set of rules for inference. Rules are facets in slots. There is a trigger mechanism that checks activities on a slot. If add, if need, if change and if remove are typical facets which take action when slot value is manipulated. In the FORXDB system, forward chaining is employed. Since the system has 2 inference engines, one on the client side and another one on the knowledge base (knowledge server) side, inference can be done on both sides. The client side first takes care of user interaction to

obtain current information interactively while the server side takes care of the inference from recorded or historical information.

In the medical diagnosis system, the client-side inference engine guides patient through relevant questions based on the given symptoms. Fig.5 shows an instance frame Basic Question1 which is created from a class frame Basic Question. The user specifies the earache symptom and the corresponding slot value is set to true. The If Change attached procedure of the slot will refer to the frame instance "symptoms to earache" which leads to further information on the patient. Finally, patient information frame instances are created. It will be submitted to the knowledge base system on the server side for further inference.

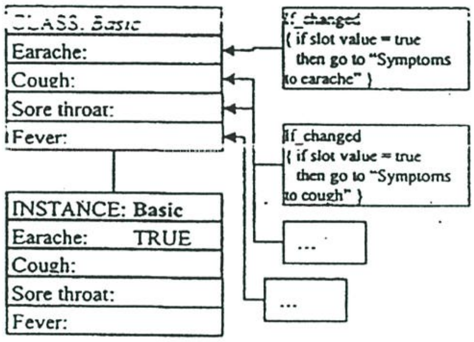


Fig. 5 Sample frames on the client side.

There are two techniques that are employed for the server side inference. The first one is to keep the frame instances from the client side as server side frame instances. Methods on the slots will direct inference to the final goal. The second technique is to keep information from the client as parameters of store procedures which in turn, carry on the inference. This second approach is suitable for small amount of information is received from the client side. Fig. 6 summarizes the inference activity in the frame system.

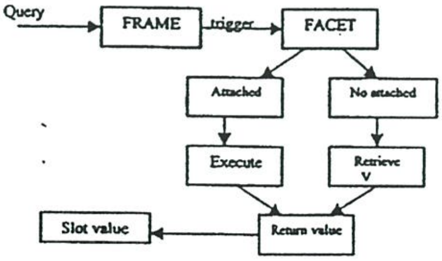


Fig. 6 Frame inference mechanism

Frames on the client expert system side guide the patient's interview process and send information obtained to the server whose frames access external databases and perform diagnosis. Appendix A shows server-side frames of ear-related problems and appendix B shows client-side frames of earache cases.

8 Conclusions

The paper presents a frame-based expert system architecture that has an inference engine on both the client consulting expert system and on the knowledge base. Inferences that are performed on the client side are mainly user's interviews and interactive fact gathering. Inferences on the knowledge base side are performed based on already known facts recorded on the databases. Frames are implemented using object relational database technology.

References:

[1] Peter D. Karp., *The Design Space of Frame Knowledge Representation System.*, SRI AI Center Technical Note #520, 1993.

[2] Marvin Minsky, *A Framework for Representing Knowledge*, Reprinted in The Psychology of Computer Vision, P. Winston (Ed.), McGraw-Hill, 1975.

[3] Negnevitsky, M., *Artificial Intelligence: A Guide to Intelligent Systems*, Addison Wesley, Harlow, England, 2002.

[4] Richard Fikes and Tom KehLer, The Role of frame-based representation in reasoning, *Communications of the ACM*, 28(9), 1985

[5] Natalya F. Noy, Mark A. Musen, Jose L.V. Mejino, Cornelius Rosses, Pushing the Envelope : Challenges in a Frame-Based Representation of Human Anatomy, *Data & Knowledge Engineering*, Volume 48 Issue 3 (ACM), March 2004

[6] Durkin J., *Expert Systems : Design and Development*, Macmillan Inc., 1994.

[7] Benjamin Kuipers, *Algernon for expert system*, Draft document in Computer Science Department University of Texas at Austin, 18 January 1994.

[8] Kamran Parsaye, *Expert systems for experts*, John Wiley & Sons, Inc., 1988.

[9] P.D. Karp et al., *The EcoCyc Database*, Nucleic Acids Research, 2002, Vol. 30, No. 1 56-58, Oxford University Press.

[10] Gang Luo and Vinay K. Chaudhri, *Implementing OKBC Knowledge Model Using Object Relational Capabilities of Oracle 8*, Technical Report.

[11] M.P. Evett, J.A. Hendler, and L. Spector, *Parallel Knowledge Representation on the Connection Machine*, Journal of Parallel and Distributed Computing, 22:168-184, 1994

[12] Neil F. Abernethy, Russ B. Altman., Sophia : Providing Basic Knowledge Service with A Common DBMS, *Proceedings of the 5th KRDB Workshop*, 1998.

[13] Nijssen, G.M. and Halpin, T.A, Conceptual Schema and Relational Database Design A Fact Oriented Approach, Prentice Hall, 1989

[14] Patrick Henry Winston, *Artificial Intelligence*, Addision Wesley, 1992

[15] Oracle Documentation, http://otn.oracle.com/documentation/content.html

[16] Dimitris Metaxas and Timos Sellis, A database implementation for large frame-based systems, *Second International Conference on Data and Knowledge Systems for Manufacturing and Engineering*, 1989.

[17] Peter Lucas and Linda van der gang, *Principle of expert systems*, Addison-Wesley, 1991.

[18] Robert Keller, *Expert system technology: Development and Application*, Yourdon Press, 1987.

[19] Stuart E. Savory, *Expert systems for professionals*, ELLIS HORWOOD Limited, 1990.

[20] Neil F. Abernethy, Julie J. Wu, Micheal Hewett, and Russ B Altman, *Sophia: A Flexible, Web-Based Knowledge Server*, Stanford University Medical Center, 1999.

[21] Chao Chen, Philip D. Udo-Inyang, Frederick C. Schmitt., *Integration of a Database Management System and a Knowledge-Based Expert System in Construction: A Review*, 1994.

566 WSEAS TRANSACTIONS ON COMPUTERS Issue 3, Vol. 5, March 2006 ISSN: 1109-2750

Appendix A: Some server-side frames of ear-related problems.

frame	slot	slot value	facet	facet value
basilar skull fracture	Parent	-		
	Earache	yes	default	yes
	Head injury	yes	default	yes
	Cerebrospinal fluid otorrhea	yes	default	yes
Secondary otalgia (referred pain)	Parent	-		
	Earache	yes	default	yes
	Pain on pulling the auricles	no	default	no
Primary otalgia (Disease of the Ear)	Parent	-		
	Children	External Ear ,Middle Ear, Inner Ear		
	Earache	yes	default	yes
	Pain on pulling the auricles	yes	if changed	if slot value = no then go to "Secondary otalgia"
External Ear	Parent	Primary otalgia		
	Children	otomycosis ,acute cellulitis ,furunculosis, [illegible], [illegible] polychondritis		
	Pain and swelling of external ear	yes	default	yes
			if changed	if slot value = no then go to "Middle Ear"
Middle Ear	Parent	Primary otalgia		
	Children	AOM, OME, Trauma to TM		
	Otitis media	yes	default	yes
			if changed	if slot value = no then go to "Inner Ear"
Inner Ear	Parent	Primary otalgia		
	Children	Bell palsy, Temporal bone tumor		
Otomycosis	Parent	External Ear		
	Characteristic of ear wax	{white plaque ,black spore}*	constraint	{white plaque ,black spore}
Acute cellulitis	Parent	External Ear		
	Infection and inflammation	yes	default	yes
Furunculosis	Parent	External Ear		
	Small firm tender red nodule in skin	yes	default	yes
Herpes	Parent	External Ear		
	Itch	yes	default	yes
	Burning pain	yes	default	yes
	Small red bumps or blisters appear around auricles and mouth	yes	default	yes
Relapsing polychondritis	Parent	External Ear		
	inflammation and deterioration of cartilaginous tissue	yes	default	yes
	inflammation of other connective tissue	{hearing loss ,eyes inflammation, joint inflammation}*	constraint	{hearing loss ,eyes inflammation, joint inflammation}
	Serology test	negative	default	negative
Ear Trauma	Parent	External Ear	Ear Trauma	Parent
	Trauma to auricles	yes	default	yes
Foreign body	Parent	External Ear	Foreign body	Parent
	Foreign body	yes	default	yes
Impacted cerumen	Parent	External Ear	Impacted cerumen	Parent
	partial loss of hearing	yes	default	yes
	a sensation of fullness in the ear	yes	default	yes
Acute otitis media (AOM)	Parent	Middle Ear		
	Pus in the middle ear	yes	default	yes
	Infection	yes	default	yes

WSEAS TRANSACTIONS ON COMPUTERS Issue 3, Vol. 5, March 2006 ISSN: 1109-2750 567

			If changed	If slot value = no then go to "OME"
	redness of the eardrum	yes	default	yes
Otitis media with effusion (OME)	Parent	Middle Ear		
	Glue ear	yes	default	yes
	Presence of fluid in the middle ear	yes	default	yes
	Infection	no	default	no
			If changed	If slot value = yes then go to "AOM"
Trauma to TM	Parent	Middle Ear		
	Trauma to eardrum	{Barotrauma, Blunt trauma, Laceration}*	constraint	{Barotrauma, Blunt trauma, Laceration}
Bell palsy	Parent	Inner Ear		
	Unilateral or bilateral acute facial nerve palsy	yes	default	yes
Temporal bone tumor	Parent	Inner Ear		
	-	-		

* one of them

Appendix B: Some client-side frames of earache cases

frame	slot	slot value	facet	facet value
Basic Question	Earache		if changed	if slot value = yes then go to "Symptoms to earache"
	Cough		if changed	if slot value = yes then go to "Symptoms to cough"
	Sore throat		if changed	if slot value = yes then go to "Symptoms to sore throat"
	Fever		if changed	if slot value = yes then go to "Symptoms fever"
Symptoms to earache	Earache	yes	default	yes
	Severe head injury		if changed	if slot value = yes then go to "Emergency"
	Pain on pulling the auricles		if changed	if slot value = yes then go to "Primary otalgia"
Emergency	Cerebrospinal fluid otorrhea		if changed	if slot value = no then go to "Symptoms to earache" and set "Severe head injury" = no else admit
Primary otalgia	Characteristic of external ear		default	normal
			constraint	{Normal ,Inflammation}
	Characteristic of ear wax		default	normal
			constraint	{normal, white, black}*
	Itche		default	no
			constraint	yes/no
	Inflammation		default	no
			constraint	yes/no
	Characteristic of eardrum		default	normal
			constraint	{normal ,redness of the eardrum ,glue ear ,perforation of eardrum}*
	symptoms about facial nerve		default	normal
			constraint	{normal ,Unilateral or bilateral acute facial nerve palsy}

ประวัติผู้เขียน

ชื่อ-นามสกุล	นางสาวชุลีรัตน์ รัตนประทีป
วัน เดือน ปีเกิด	11 สิงหาคม พ.ศ. 2522 ที่จังหวัดอุบลราชธานี
ที่อยู่	56/59 หมู่บ้านเจษฎา1 ถ.พหลโยธิน แขวงสีกัน เขตดอนเมือง กรุงเทพฯ 10210 โทร 0-2523-7442
ประวัติการศึกษา	2544 ปริญญาตรีวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ เกียรตินิยมอันดับ (2) มหาวิทยาลัยเทคโนโลยีสุรนารี
ความชำนาญเฉพาะด้าน	1.) ระบบการจัดการฐานข้อมูล 2.) ระบบผู้เชี่ยวชาญ