การพัฒนาต้นแบบระบบผู้เชี่ยวชาญแบบเฟรม โดยใช้ฐานข้อมูล RDF/XML

FRAME-BASED RDF/XML PROTOTYPE EXPERT SYSTEM DEVELOPMENT

นภัทร์ ประภากร
NAPAT PRAPAKORN

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2552

KMITL-2009-EN-M-070-060

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาต้นแบบระบบผู้เชี่ยวชาญแบบเฟรม โดยใช้ฐานข้อมูล RDF/XML

FRAME-BASED RDF/XML PROTOTYPE EXPERT SYSTEM DEVELOPMENT

นภัทธ์ ประภากร

NAPAT PRAPAKORN

เลขหมู่.....
เลขทะเบียน 105115
วัน.เดือน.ปี 16 พ.ย. 2552

.b.....
.i.....

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2552

KMITL-2009-EN-M-070-060

FRAME-BASED RDF/XML PROTOTYPE EXPERT SYSTEM DEVELOPMENT

NAPAT PRAPAKORN

A THESIS SUBMITTED IN PARTAIL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN COMPUTER ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF THECHNOLOGY LADKRABANG

2009

KMITL-2009-EN-M-070-060

COPYRIGHT 2009

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การพัฒนาต้นแบบระบบผู้เชี่ยวชาญแบบเฟรมโดยใช้ฐานข้อมูล RDF/XML

Thesis Title Frame-based RDF/XML Prototype Expert System Development

นักศึกษา นายนภัทธ์ ประภากร

รหัสประจำตัว 47060813

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา วิศวกรรมคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์ รศ.ดร.ศุภมิตร จิตตะยโศธร

หมายเลขวิทยานิพนธ์ KMITL-2009-EN-M-070-060

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
ผศ.ดร.สมศักดิ์	วลัยรัชต์	
ผศ.ดร.ชุติเมษฎ์	ศรีนิลทา	
รศ.ดร.เอื้อน	ปิ่นเงิน	
ดร.อรัญญา	วลัยรัชต์	
รศ.ดร.ศุภมิตร	จิตตะยโศธร	

วัน / เดือน/ ปี ที่สอบ วันพฤหัสบดีที่ 21 พฤษภาคม พ.ศ. 2552 เวลา 14.00-16.00 น.

สถานที่สอบ ณ อาคาร A ชั้น 3 ห้องประชุม 1

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์ รับรองแล้ว

(รองศาสตราจารย์ ดร.กอบชัย เดชหาญ)

คณบดี คณะวิศวกรรมศาสตร์

วันที่ 21 พฤษภาคม พ.ศ. 2552

หัวข้อวิทยานิพนธ์	การพัฒนาต้นแบบระบบผู้เชี่ยวชาญแบบเฟรมโดยใช้ฐานข้อมูล RDF/XML
นักศึกษา	นาย นภัทธ์ ประภากร
รหัสนักศึกษา	47060813
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
พ.ศ.	2552
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.ดร.ศุภมิตร จิตตะยโศธร

บทคัดย่อ

ระบบผู้เชี่ยวชาญคือโปรแกรมคอมพิวเตอร์ที่เลียนแบบพฤติกรรมการแก้ปัญหาของมนุษย์ โดยในระบบผู้เชี่ยวชาญส่วนใหญ่ จะเก็บกฎที่ใช้สำหรับการวินิจฉัย ไว้ในฐานความรู้ และสอบถามข้อเท็จจริงจากผู้ใช้ ขณะใช้งาน เพื่อให้กลไกการวินิจฉัย นำไปวินิจฉัยเพื่อหาคำตอบ แต่การสอบถามข้อเท็จจริงจากผู้ใช้ทุกครั้ง ย่อมทำให้เกิดความไม่สะดวกในการใช้งาน อีกทั้งยังอาจทำให้ได้รับข้อเท็จจริงที่ไม่ถูกต้อง หรือ อาจไม่ได้ข้อเท็จจริงเลย เนื่องจากผู้ใช้ไม่สามารถให้ข้อมูลที่ถูกต้องได้ จึงได้มีการนำระบบฐานข้อมูลเข้ามาเชื่อมต่อกับระบบผู้เชี่ยวชาญในส่วนของฐานความรู้ เพื่อเป็นการแก้ปัญหาดังกล่าว

แต่อย่างไรก็ตาม การเรียกใช้ข้อเท็จจริงจากระบบฐานข้อมูล เพื่อนำมาประกอบการวินิจฉัยของระบบผู้เชี่ยวชาญ ในบางครั้งจำเป็นต้องมีการส่งข้อมูลเป็นจำนวนมาก ซึ่งจะทำให้เกิดความล่าช้าในการส่งข้อมูลผ่านเครือข่ายระหว่างการวินิจฉัย และเป็นการเพิ่มภาระการประมวลผลให้กับเปลือกระบบผู้เชี่ยวชาญ วิทยานิพนธ์ฉบับนี้จึงได้นำเสนอการพัฒนาต้นแบบระบบผู้เชี่ยวชาญ ที่มีโครงสร้างแตกต่างออกไป โดยจะมีกลไกลการวินิจฉัยเพิ่ม ที่ส่วนของฐานความรู้ เพื่อให้ฐานความรู้สามารถวินิจฉัยข้อเท็จจริงที่เก็บไว้ และส่งเพียงคำตอบที่ระบบผู้เชี่ยวชาญต้องการ กลับไปยังระบบผู้เชี่ยวชาญได้

นอกจากนี้ วิทยานิพนธ์ฉบับนี้ ยังได้นำเสนอการสร้างฐานความรู้ ด้วยฐานข้อมูล RDF/XML ที่เป็นการจัดเก็บข้อมูลที่ใช้ทรัพยากรน้อย และสามารถรองรับการพัฒนาการจัดเก็บข้อมูลไว้บนเว็บได้ โดยในฐานความรู้ของระบบต้นแบบ จะใช้การแทนความรู้แบบเฟรม ซึ่งถูกนำมาใช้ในระบบผู้เชี่ยวชาญที่มีขนาดใหญ่และมีความซับซ้อน อย่างกว้างขวาง

วิทยานิพนธ์ฉบับนี้จึงได้นำเสนอการพัฒนาต้นแบบระบบผู้เชี่ยวชาญแบบเฟรม โดยใช้ฐานข้อมูล RDF/XML เป็นฐานความรู้ ที่เพิ่มกลไกการวินิจฉัยที่ส่วนของฐานความรู้ เพื่อช่วยลดปริมาณการรับส่งข้อมูลระหว่างระบบผู้เชี่ยวชาญ และฐานความรู้ โดยระบบต้นแบบที่พัฒนาขึ้น จะเป็นระบบผู้เชี่ยวชาญการวิเคราะห์เลือกโปรโมชันสำหรับโทรศัพท์เคลื่อนที่

Thesis Title	Frame-based RDF/XML Prototype Expert System Development
Student	Mr. Napat Prapakorn
Student ID.	47060813
Degree	Master of Engineering
Program	Computer Engineering
Year	2009
Thesis Advisor	Assoc. Prof. Dr. Suphamit Chittayasothorn

ABSTRACT

Expert system is a program that imitates human's problem solving process. Most expert systems collect facts by questioning user during a consultation session then use rules stored in a knowledge base to inference those facts for solving problems. Therefore, asking too many questions to a user may result in inaccurate facts or no fact at all. To prevent this facts acquisition problem, a database management system is presented. With a coupling between expert system and database management system, facts are retrieved from a database management system without asking a user.

However, retrieving many existing facts will result in a lot of communication traffic during inference and also a processing load in an expert system shell. This thesis proposes an alternative architecture of expert system which has an inference engine on its external knowledge base side. With such architecture, inferences can be performed on an external knowledge base then sending the result, not only simple facts, to an expert system shell for any further inferences.

Moreover, this thesis also proposes a knowledge base development using RDF/XML database, a recently emerged, low resource technology for representing data on the World Wide Web, and using frames, which are widely used by large and complex expert systems, as its knowledge representation.

This thesis presents a frame-based RDF/XML prototype expert system development. An inference engine on an external knowledge base side is also added to decrease communication traffic when transferring a large collection of data during inference. The prototype system will be developed based on a mobile promotion consultant problem solving.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดี ด้วยความเมตตาในการให้คำปรึกษาและแนะนำตลอดทุกขั้นตอนการดำเนินการจาก รศ.ดร.ศุภมิตร จิตตะยโสธร ซึ่งเป็นอาจารย์ที่ปรึกษาวิทยานิพนธ์ ข้าพเจ้ารู้สึกซาบซึ้งในความเมตตาของท่านมากและขอกราบขอบพระคุณเป็นอย่างสูง

วิทยานิพนธ์ฉบับนี้จะไม่สามารถสำเร็จลุล่วงได้ หากปราศจากแรงผลักดันและความเชื่อมั่นจากครอบครัวอันเป็นที่รักยิ่งของข้าพเจ้า ซึ่งเป็นแรงบันดาลใจและกำลังใจให้ข้าพเจ้าในการทำวิทยานิพนธ์ฉบับนี้อย่างเสมอมา

ข้าพเจ้าขอกราบขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์ และคณาจารย์ทุกท่านในภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ ที่ให้ความกรุณาประสิทธิประสาทความรู้และตักเตือนแนะนำให้ข้าพเจ้าสามารถดำเนินการทำวิทยานิพนธ์จนสำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณ บมจ. โทเทิล แอคเซส คอมมิวนิเคชัน ที่ให้ความอนุเคราะห์ข้อมูล และให้คำปรึกษาเกี่ยวกับการเลือกโปรโมชันสำหรับโทรศัพท์เคลื่อนที่

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ทุกคน ที่คอยห่วงใย และให้ความช่วยเหลือข้าพเจ้ามาโดยตลอด และขอขอบคุณ คุณเมธาวี พันเจริญ สำหรับความช่วยเหลืออย่างเต็มที่ในการสอบวิทยานิพนธ์

สุดท้ายนี้ คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน ข้าพเจ้าหวังเป็นอย่างยิ่งว่าวิทยานิพนธ์ฉบับนี้จะเป็นประโยชน์ต่อผู้ศึกษา

นภัทธ์ ประภากร

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VI

สารบัญรูป VII

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วรรณกรรมที่เกี่ยวข้อง 3

1.3 แนวทางการแก้ปัญหา 3

1.4 วัตถุประสงค์ 4

1.5 สมมติฐานของการศึกษา 4

1.6 ขอบเขตการศึกษา 5

1.7 ขั้นตอนการศึกษา 5

1.8 โครงสร้างของวิทยานิพนธ์ 6

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 7

2.1 ระบบผู้เชี่ยวชาญ 7

2.2 การแทนความรู้ 11

2.3 กลไกการวินิจฉัยบนระบบผู้เชี่ยวชาญ 16

2.4 ภาษา RDF 24

2.5 การติดต่อสื่อสาร และสืบค้นข้อมูลด้วยภาษา SPARQL 32

บทที่ 3 สถาปัตยกรรมของระบบต้นแบบ 39

3.1 สถาปัตยกรรมของระบบต้นแบบที่นำเสนอ 39

3.2 การติดต่อสื่อสารระหว่างไคลเอนท์กับเซิร์ฟเวอร์ 42

สารบัญ (ต่อ)

หน้า

บทที่ 4 รายละเอียดเกี่ยวกับการพัฒนาระบบต้นแบบ 43

4.1 การเขียนโครงสร้างของเฟรมบนฐานข้อมูล RDF/XML 43

4.2 กลไกการวินิจฉัยของระบบ 46

4.3 การติดต่อสื่อสารและแลกเปลี่ยนข้อมูล RDF ด้วยภาษา SPARQL 50

บทที่ 5 การพัฒนาต้นแบบระบบผู้เชี่ยวชาญเพื่อวิเคราะห์เลือกโปรโมชันสำหรับโทรศัพท์เคลื่อนที่ ...52

5.1 ขั้นตอนการทำงานของระบบ 53

5.2 การสร้างเฟรมใช้แทนความรู้ 54

5.3 การวินิจฉัยหาคำตอบของต้นแบบระบบผู้เชี่ยวชาญที่พัฒนาขึ้น 60

5.4 การทดสอบระบบต้นแบบที่พัฒนาขึ้น 62

บทที่ 6 สรุปผลและข้อเสนอแนะ 67

6.1 สรุปผลการทดสอบระบบต้นแบบ 67

6.2 ข้อเสนอแนะ 68

เอกสารอ้างอิง 69

ภาคผนวก 71

ภาคผนวก ก. เฟรมต้นแบบที่จัดเก็บในฐานข้อมูล RDF/XML 72

ภาคผนวก ข. งานวิจัยที่ได้รับการตีพิมพ์ 77

ประวัติผู้เขียน.....90

สารบัญตาราง

ตารางที่ หน้า

5.1 แสดงเฟรมที่เกี่ยวกับโปรโมชันสำหรับระบบรายเดือนและลักษณะการใช้งานที่เหมาะสม..... 54

5.2 เฟรมคำถามเกี่ยวกับลักษณะการใช้งานของลูกค้า..... 57

สารบัญรูป

รูปที่ หน้า

2.1 ส่วนประกอบหลักของระบบผู้เชี่ยวชาญ 8
2.2 สถาปัตยกรรมของระบบฐานความรู้....9
2.3 ตัวอย่างโครงสร้างของเฟรม 13
2.4 ตัวอย่างโครงสร้าง และความสัมพันธ์ของเฟรมรถยนต์....15
2.5 สถานะของเนื้องาน สเต็กเป้าหมาย และสเต็กของกฎ I....17
2.6 สถานะของเนื้องาน สเต็กเป้าหมาย และสเต็กของกฎ II 17
2.7 สถานะของเนื้องาน สเต็กเป้าหมาย และสเต็กของกฎ III 18
2.8 สถานะของเนื้องาน สเต็กเป้าหมาย และสเต็กของกฎ IV....19
2.9 สถานะของเนื้องาน สเต็กเป้าหมาย และสเต็กของกฎ V....20
2.10 สถานะของเนื้องานก่อนที่จะเริ่มกระบวนการวินิจฉัย....21
2.11 สถานะของเฟรม A และ B ก่อนที่จะถูกกระตุ้นด้วยดีมอน 23
2.12 สถานะของเฟรม A และ B หลังจากถูกกระตุ้นด้วยดีมอน 23
2.13 การอธิบายทรัพยากรในภาษา RDF โดยใช้ ซับเจกต์, เพรดิเคต และ อ็อบเจกต์....25
2.14 การนำเสนอประโยคภาษา RDF ด้วยกราฟ....27
2.15 การเขียนทริปเพิล แทนกราฟ....27
2.16 การเขียน คำเติมหน้า แทน URI 28
2.17 เอกสาร RDF/XML....29
2.18 การแสดง class และ subclass ในเอกสาร RDF/XML....30
2.19 แสดงความสัมพันธ์แนวคิดของ RDF schema....31
2.20 แสดงข้อมูลสำหรับประกอบการอธิบายการสืบค้นด้วยภาษา SPARQL....32
2.21 แสดงการสืบค้นด้วยคำสั่ง SELECT ในภาษา SPARQL....33
2.22 แสดงผลลัพธ์จากคำสั่ง SELECT ในภาษา SPARQL....33
2.23 แสดงผลลัพธ์จากคำสั่ง SELECT ในภาษา SPARQL ในรูปแบบ XML....34
2.24 แสดงการสืบค้นด้วยคำสั่ง CONSTRUCT ในภาษา SPARQL....35
2.25 แสดงผลลัพธ์จากคำสั่ง CONSTRUCT ในภาษา SPARQL ในรูปของทริปเพิล....35
2.26 แสดงผลลัพธ์จากคำสั่ง CONSTRUCT ในภาษา SPARQL ในรูปแบบ RDF/XML....35
2.27 แสดงการสืบค้นด้วยคำสั่ง ASK ในภาษา SPARQL....36
2.28 แสดงผลลัพธ์จากคำสั่ง ASK ในภาษา SPARQL ในรูปแบบ XML 36

สารบัญรูป (ต่อ)

รูปที่ หน้า

2.29 อินเตอร์เฟซ SparqlQuery....37
2.30 การผูกโปรโตคอลกับ HTTP....38
3.1 แสดงสถาปัตยกรรมของระบบ....40
3.2 แสดงสถาปัตยกรรมของระบบฝั่งเซิร์ฟเวอร์....41
3.3 แสดงสถาปัตยกรรมของระบบฝั่งไคลเอนท์....41
4.1 แผนภาพโครงสร้างข้อมูลแบบเฟรม....44
4.2 RDF กราฟ แสดงโครงสร้างฐานข้อมูล RDF/XML เพื่อใช้จัดเก็บเฟรม....45
4.3 ตัวอย่างเฟรมต้นแบบ ในฐานข้อมูล RDF/XML....45
4.4 อัลกอริธึมในการหาเฟรมเป้าหมายด้วยการจับคู่....47
4.5 แสดงการทำงานของกระบวนการหาค่าของสล๊อตภายในเฟรม....49
4.6 อัลกอริธึมในการเรียกดูค่าของสล๊อต....49
4.7 อัลกอริธึมสำหรับการหาเฟรมแม่....49
4.8 อัลกอริธึมสำหรับการดึงค่าจากสล๊อต....50
4.9 การส่งคำสั่งภาษา SPARQL ผ่านโปรโตคอล HTTP....50
4.10 HTTP Trace ส่วนผลลัพธ์จากการสืบค้น....51
5.1 แสดงขั้นตอนการทำงานของระบบวิเคราะห์เลือกโปรโมชัน....54
5.2 เฟรมลูกค้าที่ฝั่งไคลเอนท์....61
5.3 ส่วนผลลัพธ์การสอบถามโปรโมชันที่เหมาะสมไปยังเซิร์ฟเวอร์ที่ไม่มีกลไกการวินิจฉัย....65
5.4 ผลลัพธ์การสอบถามโปรโมชันที่เหมาะสมไปยังเซิร์ฟเวอร์ที่มีกลไกการวินิจฉัย....66

บทที่ 1
บทนำ

1.1 ความเป็นมา และความสำคัญของปัญหา

ระบบผู้เชี่ยวชาญ เป็นสาขาหนึ่งในการศึกษาด้านปัญญาประดิษฐ์ และเป็นโปรแกรมคอมพิวเตอร์ที่มีพฤติกรรมเลียนแบบความฉลาดของมนุษย์ ซึ่งจะประกอบไปด้วยสองส่วน คือส่วนฐานความรู้ (Knowledge Base) และส่วนกลไกการวินิจฉัย (Inference Engine) ที่เรียกว่า เปลือกระบบผู้เชี่ยวชาญ (Expert System Shell)

โดยส่วนมาก เปลือกระบบผู้เชี่ยวชาญจะมีโครงสร้างฐานความรู้อยู่ภายใน ซึ่งจะต้องมีการโหลดข้อมูลจากฐานความรู้เข้ามายังหน่วยความจำหลักระหว่างกระบวนการวินิจฉัย เพื่อทำการวินิจฉัยแก้ปัญหาอีกที ในระหว่างกระบวนการวินิจฉัยนั้น ข้อเท็จจริงจะได้มาจากสองส่วน ส่วนแรกคือ ข้อเท็จจริงที่ได้มาจากผู้ใช้ป้อนเข้าสู่ระบบในลักษณะโต้ตอบไปมา และส่วนที่สองคือข้อเท็จจริง ที่ได้จากการวินิจฉัยแก้ปัญหา โดยทั่วไป กฎในการวินิจฉัยในระบบผู้เชี่ยวชาญจะถูกกระทำโดยกลไกการวินิจฉัยของระบบผู้เชี่ยวชาญ ซึ่งใช้ข้อเท็จจริงที่ได้จะมาจากการสอบถามโต้ตอบกับผู้ใช้เสียเป็นส่วนใหญ่ แต่เนื่องจากระบบผู้เชี่ยวชาญบางครั้งมีการถามผู้ใช้บ่อยเกินไป ทำให้เมื่อมีการใช้งานบ่อยครั้งเข้า ย่อมก่อให้เกิดความไม่สะดวกในการใช้ได้ และยังทำให้การป้อนข้อมูลของผู้ใช้เพื่อนำไปประกอบการวินิจฉัยเกิดความผิดพลาดขึ้นได้ อีกทั้งบางคำถามอาจจะเกี่ยวข้องกับข้อมูลในอดีต หรือข้อมูลที่ผู้ใช้เองอาจไม่ทราบ ด้วยเหตุนี้จึงได้มีงานวิจัยหลายงานพยามนำเสนอแนวทางการพัฒนาระบบผู้เชี่ยวชาญเพื่อแก้ปัญหาดังกล่าว

1.2 วรรณกรรมที่เกี่ยวข้อง

เพื่อเป็นการลดปัญหาในการสอบถามข้อเท็จจริงที่มีอยู่แล้วจากผู้ใช้บ่อยครั้ง ได้มีงานวิจัยหลายงาน นำเสนอโครงสร้างของระบบผู้เชี่ยวชาญ ที่นำระบบฐานข้อมูล ซึ่งเป็นระบบที่มีประสิทธิภาพในการจัดเก็บข้อมูล มาเชื่อมต่อกับฐานความรู้ เพื่อให้เปลือกระบบผู้เชี่ยวชาญสามารถนำข้อเท็จจริงที่เก็บไว้ในระบบฐานข้อมูล มาประกอบการวินิจฉัยได้

ปี ค.ศ. 1988

- งานวิจัย [19] ได้นำเสนอโครงสร้างของระบบผู้เชี่ยวชาญ ที่มีการเชื่อมต่อกับระบบฐานข้อมูล เพื่อนำข้อเท็จจริงที่มีอยู่แล้ว มาใช้ประกอบการวินิจฉัย

- งานวิจัย [20] ได้นำเสนอภาษาที่ใช้ในการสืบค้นข้อเท็จจริงจากฐานข้อมูล ที่นำมาเชื่อมต่อกับระบบผู้เชี่ยวชาญแบบแน่น (Tight Coupling) เพื่อลดปัญหาในด้านประสิทธิภาพและความสัมพันธ์เชิงความหมายของข้อเท็จจริงที่ได้จากการสืบค้น

ปี ค.ศ. 1989

- งานวิจัย [21] ได้นำเสนอการเชื่อมต่อระบบผู้เชี่ยวชาญกับระบบฐานข้อมูล โดยใช้วิธีการเชื่อมต่อแบบหลวม (Loose Coupling) ในการพัฒนาส่วนเชื่อมต่อ
- งานวิจัย [22] ได้นำเสนอโครงสร้างของระบบผู้เชี่ยวชาญ ที่เชื่อมต่อกับระบบฐานข้อมูล ที่ใช้การแทนความรู้แบบกฎ และใช้ความรู้เชิงความหมายประกอบกับความรู้เชิงสามัญในการวินิจฉัยเพื่อหาคำตอบ

ปี ค.ศ. 1990

- งานวิจัย [23] ได้นำเสนอการวินิจฉัยโดยใช้กฎ ในระบบผู้เชี่ยวชาญที่เชื่อมต่อกับระบบฐานข้อมูล ซึ่งใช้การแทนความรู้แบบกฎ ที่จัดเก็บอยู่ในระบบฐานข้อมูลด้วย

ปี ค.ศ. 1991

- งานวิจัย [24] ได้นำเสนอการพัฒนาระบบผู้เชี่ยวชาญโดยการใช้ตรรกศาตร์คลุมเครือ (Fuzzy Logic) และประเด็นเกี่ยวกับข้อมูลเชิงเวลา มาพิจารณาตัดสินการใช้กระบวนการต่างๆ ของระบบผู้เชี่ยวชาญ

ปี ค.ศ. 1993

- งานวิจัย [25] ได้นำเสนอการพัฒนาระบบผู้เชี่ยวชาญ ที่มีการเชื่อมต่อกับระบบฐานข้อมูล และใช้เทคนิคการวินิจฉัยแบบ ฟอร์เวิร์ดเชนนิง (Forward Chaining) ผสมกับแบคเวิร์ดเชนนิง (Backward Chaining)

ปี ค.ศ. 1994

- งานวิจัย [26] ได้นำเสนอสถาปัตยกรรมของระบบผู้เชี่ยวชาญแบบกระจาย (Distributed Expert System) พร้อมทั้งภาษา และโครงกฎ (Meta-rule) ที่ใช้ในการควบคุมการทำงานของกฎ ในกระบวนการวินิจฉัย

ปี ค.ศ. 2005

- งานวิจัย [27] ได้นำเสนอการออกแบบและวิธีการสร้างระบบผู้เชี่ยวชาญที่เชื่อมต่อกับฐานข้อมูลแบบคลุมเครือ (Dynamic Fuzzy Database) เพื่อให้ระบบสามารถแก้ไขปัญหาที่มีความคลุมเครือและ ขึ้นกับปัจจัยเชิงเวลาได้อย่างมีประสิทธิภาพ

จะเห็นได้ว่า มีวรรณกรรมมากมาย นำเสนอการนำระบบฐานข้อมูลเข้ามาใช้ร่วมกับระบบผู้เชี่ยวชาญ เพื่อพัฒนาขีดความสามารถของระบบผู้เชี่ยวชาญ และอีกทั้งเป็นการช่วยลดปัญหาเกี่ยวการสอบถามข้อเท็จจริง ที่มีอยู่แล้วจากผู้ใช้ได้ และนอกจากนี้ยังรองรับขนาดของฐานความรู้ที่มีแนวโน้มจะเพิ่มมากขึ้นเรื่อยๆอีกด้วย แต่อย่างไรก็ตาม การเรียกใช้ข้อเท็จจริง จากระบบฐานข้อมูลมาประกอบการวินิจฉัยในบางครั้ง มีความจำเป็นต้องใช้ข้อเท็จจริงจำนวนมาก ทำให้ระบบต้องโหลดข้อมูลจำนวนมากจากระบบฐานข้อมูล เข้ามาที่หน่วยความจำหลักระหว่างการวินิจฉัย เช่น ระบบผู้เชี่ยวชาญการใช้งานโทรศัพท์เคลื่อนที่ หากมีความจำเป็นต้องใช้ข้อมูลประวัติการโทรออกของลูกค้ามาประกอบการวินิจฉัย ระบบจะต้องโหลดข้อมูลประวัติการโทรออกของลูกค้า ซึ่งมีจำนวนมาก จากฐานข้อมูลที่เก็บข้อมูลดังกล่าวไว้ มาที่หน่วยความจำหลัก ซึ่งจะทำให้เกิดความล่าช้าในการส่งข้อมูล และเป็นการทำให้เกิดภาระในการประมวลผลของเปลือกระบบผู้เชี่ยวชาญอีกด้วย

1.3 แนวทางการแก้ปัญหา

จากการศึกษาวรรณกรรมที่เกี่ยวข้องต่างๆแล้วพบว่า หากฐานความรู้สามารถส่งเพียงคำตอบที่เปลือกระบบผู้เชี่ยวชาญต้องการ กลับไปให้เปลือกระบบผู้เชี่ยวชาญได้ โดยคำตอบอาจเป็นข้อมูลทางสถิติ หรือปัจจัยความเหมาะสมต่างๆ ซึ่งทำให้เปลือกระบบผู้เชี่ยวชาญสามารถทำการวินิจฉัยต่อได้ทันที ไม่ใช่เป็นการโหลดข้อเท็จจริงที่เกี่ยวข้องทั้งหมด มาไว้ที่เปลือกระบบผู้เชี่ยวชาญ แล้วค่อยทำการวินิจฉัย ก็จะสามารถลดปัญหาในด้านความล่าช้าในการส่งข้อมูล และเป็นการช่วยลดภาระในการประมวลผลของเปลือกระบบผู้เชี่ยวชาญได้อีกด้วย

วิทยานิพนธ์ฉบับนี้จึงได้นำเสนอการพัฒนาต้นแบบระบบผู้เชี่ยวชาญที่แตกต่างออกไป โดยเพิ่มกลไกการวินิจฉัยที่ฐานความรู้ เพื่อเป็นการลดปัญหาด้านความล่าช้าในการส่งข้อมูล เมื่อมีการเรียกใช้ข้อเท็จจริงจากฐานความรู้ และเพื่อเป็นการช่วยลดภาระในการประมวลผลที่ฝั่งเปลือกระบบผู้เชี่ยวชาญ โดยวิทยานิพนธ์ฉบับนี้ จะใช้ฐานข้อมูล RDF/XML เป็นฐานความรู้ เนื่องจากฐานข้อมูล RDF/XML เป็นรูปแบบการจัดเก็บข้อมูลที่กำลังเป็นที่นิยม และใช้ทรัพยากรไม่มากในการจัดเก็บ อีกทั้งยังสามารถนำข้อมูล ไปเก็บไว้บนเว็บได้โดยง่าย เพื่อที่จะสามารถเป็นแนวทางในการพัฒนาให้ระบบผู้เชี่ยวชาญสามารถใช้ฐานความรู้ที่อยู่บนเว็บได้ในอนาคตอีกด้วย

และจากการศึกษาพบว่า การแทนความรู้แบบเฟรม (Frame-based Knowledge Representation) เป็นเทคนิคในการแทนความรู้ ที่ได้รับการยอมรับว่าสามารถสร้างตัวแทนความรู้ที่มีความซับซ้อนและมีขนาดใหญ่ได้ดี ซึ่งมีความเหมาะสมสำหรับฐานความรู้ที่ใช้ในระบบผู้เชี่ยวชาญขนาดใหญ่ได้เป็นอย่างดี ดังจะเห็นได้จากความนิยมในการนำหลักการนี้มาพัฒนา และวิจัยระบบผู้เชี่ยวชาญต่างๆ

ในปัจจุบัน [1-3] วิทยานิพนธ์ฉบับนี้ จึงได้เลือกใช้การแทนความรู้แบบเฟรม เป็นการแทนความรู้ในฐานความรู้ของระบบต้นแบบที่พัฒนาขึ้น

1.4 วัตถุประสงค์

วิทยานิพนธ์ฉบับนี้ มีวัตถุประสงค์เพื่อพัฒนาต้นแบบระบบผู้เชี่ยวชาญ ซึ่งมีการพัฒนาในส่วนต่างๆ ดังต่อไปนี้

1) พัฒนาต้นแบบระบบผู้เชี่ยวชาญ ด้วยการเพิ่มกลไกการวินิจฉัย ให้สามารถวินิจฉัยข้อมูลได้ทั้งสองฝั่ง คือฝั่งเปลือกระบบผู้เชี่ยวชาญที่ทำหน้าที่ติดต่อกับผู้ใช้ และฝั่งฐานความรู้ เพื่อเป็นการลดความความล่าช้าในการส่งข้อมูลจำนวนมาก และหลายครั้ง ระหว่างเปลือกระบบผู้เชี่ยวชาญ กับฐานความรู้ และลดภาระในการประมวลผลของฝั่งเปลือกระบบผู้เชี่ยวชาญ

2) พัฒนาฐานความรู้ ให้จัดเก็บอยู่ในฐานข้อมูล RDF/XML ซึ่งเป็นการจัดเก็บข้อมูลที่กำลังเป็นที่นิยม และใช้ทรัพยากรในการจัดเก็บไม่มาก

3) พัฒนาให้ระบบสามารถเรียกใช้ข้อเท็จจริงที่มีอยู่แล้วจากฐานข้อมูล RDF/XML ได้ เพื่อเป็นการลดปัญหาความไม่สะดวกและความผิดพลาดในการสอบถามข้อเท็จจริงที่มีอยู่แล้วจากผู้ใช้

1.5 สมมติฐานของการศึกษา

ในวิทยานิพนธ์ฉบับนี้ ระบบต้นแบบที่พัฒนาขึ้นจะต้องมีความสามารถในการทำการวินิจฉัยได้ทั้งสองทาง คือ ส่วนเปลือกระบบผู้เชี่ยวชาญ ซึ่งจะมีการพัฒนาให้มีฐานความรู้อยู่ภายในด้วย โดยในวิทยานิพนธ์ฉบับนี้ จะเรียกส่วนนี้ว่า ส่วนไคลเอนท์ และ ส่วนฐานความรู้ที่จัดเก็บอยู่ในฐานข้อมูล RDF/XML ซึ่งจะมีกลไกการวินิจฉัยอยู่ด้วยเช่นกัน โดยในวิทยานิพนธ์ฉบับนี้ จะเรียกส่วนนี้ว่า เซิร์ฟเวอร์ และในส่วนของเซิร์ฟเวอร์นี้จะต้องสามารถส่งผลลัพธ์ที่ผ่านการวินิจฉัยเรียบร้อยแล้วกลับมายังไคลเอนท์ได้ เมื่อได้รับคำถามจากไคลเอนท์

โดยวิทยานิพนธ์ฉบับนี้ จะนำเสนอต้นแบบระบบผู้เชี่ยวชาญการวิเคราะห์เลือกโปรโมชันสำหรับโทรศัพท์เคลื่อนที่ เป็นกรณีศึกษา

1.6 ขอบเขตของการศึกษา

วิทยานิพนธ์ฉบับนี้ กำหนดขอบเขตของการศึกษาเอาไว้ดังต่อไปนี้

1) ศึกษาการออกแบบระบบผู้เชี่ยวชาญที่ใช้รูปแบบการแทนความรู้แบบเฟรม

2) ศึกษาการจัดเก็บข้อมูลในฐานข้อมูล RDF/XML

3) ศึกษาการวินิจฉัยแบบเฟรมบนเอกสารRDF/XMLและการสืบค้นข้อมูลด้วยภาษา SPARQL

4) ศึกษาการติดต่อสื่อสารและแลกเปลี่ยนข้อมูลระหว่างเซิร์ฟเวอร์และไคลเอนท์

5) พัฒนาระบบผู้เชี่ยวชาญต้นแบบเป็นกรณีศึกษา ซึ่งจะใช้ระบบการวิเคราะห์เลือกโปรโมชันสำหรับโทรศัพท์เคลื่อนที่ โดยระบบต้นแบบจะต้องมีความสามารถดังต่อไปนี้

5.1) สามารถทำการวินิจฉัยข้อมูลที่ผู้ใช้ป้อนเข้ามาเกี่ยวกับลักษณะการใช้งานโทรศัพท์เคลื่อนที่ รวมถึงข้อมูลอื่นๆที่เกี่ยวข้องได้

5.2) สามารถวินิจฉัยข้อมูลในฐานข้อมูล RDF/XML ที่ฝั่งเซิร์ฟเวอร์ได้

5.3) สามารถติดต่อสื่อสารและแลกเปลี่ยนข้อมูลระหว่างเซิร์ฟเวอร์และไคลเอนท์ได้

1.7 ขั้นตอนของการศึกษา

เพื่อให้การดำเนินการศึกษา และพัฒนาเป็นไปด้วยความเรียบร้อย วิทยานิพนธ์ฉบับนี้ได้แบ่งขั้นตอนการดำเนินการไว้ดังนี้

1) ศึกษาการแทนความรู้แบบเฟรม และกลไกการวินิจฉัยแบบเฟรม

2) ศึกษาการจัดเก็บข้อมูลในฐานข้อมูล RDF/XML

3) ศึกษาการติดต่อสื่อสารเซิร์ฟเวอร์และไคลเอนท์

4) ศึกษากระบวนการวิเคราะห์เลือกโปรโมชันสำหรับโทรศัพท์เคลื่อนที่ เพื่อเป็นกรณีศึกษา

5) พัฒนาฐานความรู้ต้นแบบ ที่ใช้การแทนความรู้แบบเฟรมในฐานข้อมูล RDF/XML ด้วยโปรแกรม Protégé

6) พัฒนาต้นแบบระบบผู้เชี่ยวชาญสำหรับงานการวิเคราะห์เลือกโปรโมชันสำหรับโทรศัพท์เคลื่อนที่เป็นกรณีศึกษา

1.8 โครงสร้างของวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้แบ่งเนื้อหาออกเป็น 6 บท ซึ่งแต่ละบทประกอบด้วยเนื้อหาดังนี้

บทที่ 1 กล่าวถึงความเป็นมา และความสำคัญของปัญหา วรรณกรรมที่เกี่ยวข้อง แนวทางการแก้ปัญหา วัตถุประสงค์ของการศึกษา สมมติฐานของการศึกษา รวมไปถึงขอบเขตของการศึกษา และขั้นตอนของการศึกษา

บทที่ 2 กล่าวถึงทฤษฎีเกี่ยวกับเฟรมและการแทนความรู้แบบเฟรม กลไกการวินิจฉัยในระบบเฟรม ทฤษฎีของการนำเสนอข้อมูลในรูปแบบเอกสาร RDF/XML รวมถึง การสืบค้นข้อมูล ในรูปแบบเอกสาร RDF/XML ด้วยภาษา SPARQL และการติดต่อสื่อสารและแลกเปลี่ยนข้อมูลระหว่างเซิร์ฟเวอร์และไคลเอนท์

บทที่ 3 กล่าวถึงสถาปัตยกรรมของระบบที่นำเสนอ ซึ่งจะกล่าวถึง การทำงานของระบบที่ฝั่งเซิร์ฟเวอร์และฝั่งไคลเอนท์ รวมไปถึงกลไกการวินิจฉัยของระบบ

บทที่ 4 กล่าวถึงขั้นตอนในการสร้างระบบต้นแบบเพื่อเป็นกรณีศึกษา พร้อมทั้งวิธีการสร้างเฟรมต้นแบบ และ เทคนิคในการวินิจฉัยบนระบบที่สร้างขึ้น ด้วยการสืบค้นข้อมูลจากฐานข้อมูล RDF/XML

บทที่ 5 กล่าวถึงระบบต้นแบบที่เป็นกรณีศึกษา เพื่อทดสอบการทำงานของระบบที่ได้พัฒนาไว้ข้างต้น

บทที่ 6 สรุปผลการทดสอบ ข้อเสนอแนะ และแนวทางการพัฒนาระบบ ต่อไปในอนาคต

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 ระบบผู้เชี่ยวชาญ

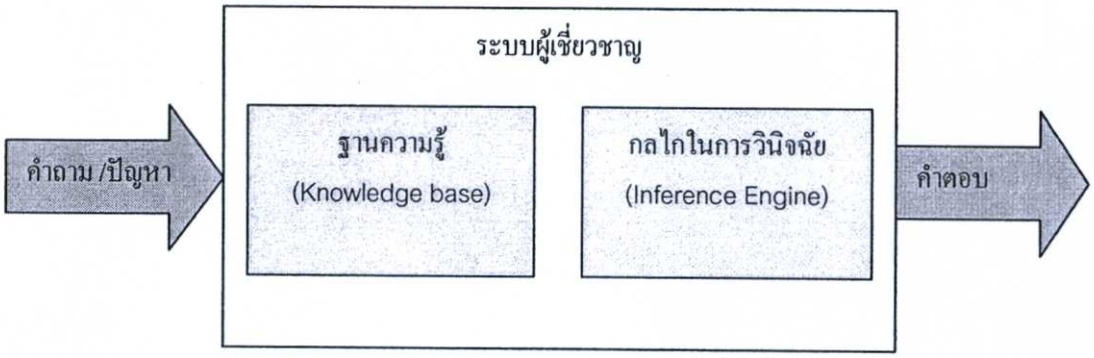
ระบบผู้เชี่ยวชาญ คือ โปรแกรมคอมพิวเตอร์ที่ถูกออกแบบมาเพื่อทำหน้าที่เสมือนเป็นผู้เชี่ยวชาญ ในการแก้ปัญหาในขอบเขตนั้นๆ ซึ่งโปรแกรมจะใช้ความรู้ในขอบเขตที่ถูกกำหนดเข้าไปในนั้นและระบุแผนการควบคุมความรู้นั้นเพื่อให้ได้มาซึ่งคำตอบหรือวิธีการแก้ปัญหา โดยระบบผู้เชี่ยวชาญส่วนมากมักถูกเรียกว่าระบบฐานความรู้ด้วยเหมือนกัน ระบบผู้เชี่ยวชาญนั้นจะไม่เรียกว่าเป็นโปรแกรม แต่เป็นระบบ เนื่องจากระบบผู้เชี่ยวชาญจะรายล้อมไปด้วยส่วนประกอบหลายส่วน เช่น ฐานความรู้ กลไกการวินิจฉัย ส่วนอธิบาย เป็นต้น ซึ่งส่วนประกอบเหล่านี้จะมีผลต่อกันและกันในการเลียนแบบกระบวนการแก้ปัญหาโดยที่ได้รับการยอมรับจากผู้เชี่ยวชาญในขอบเขตนั้นๆ[8]

ระบบผู้เชี่ยวชาญจะจำลองกระบวนการตัดสินใจโดยใช้ข้อเท็จจริงและความรู้ ซึ่งต้องการมากกว่าความรู้เดียวในกระบวนการตัดสินใจ และส่วนประกอบหลักของระบบผู้เชี่ยวชาญประกอบไปด้วยสองส่วนคือ [9]

ฐานความรู้ - ซึ่งเป็นกลุ่มของความรู้ที่ต้องใช้ในการแก้ปัญหา และ

กลไกการวินิจฉัย - ซึ่งใช้ตรวจสอบข้อเท็จจริงที่มีอยู่, เลือกแหล่งความรู้ที่เหมาะสมจากฐานความรู้ จับคู่ข้อเท็จจริงกับความรู้ และก่อให้เกิดข้อเท็จจริงเพิ่มเติม

แต่เนื่องจากปัญหามีเพิ่มขึ้นเรื่อยๆ และข้อเท็จจริงจำนวนมากได้มาจากข้อเท็จจริงที่มีอยู่แล้ว ดังนั้นในการทำงานของระบบ การตัดสินใจหรือข้อเท็จจริงจะเกิดขึ้นต่อๆกันไปเป็นลำดับโดยใช้ข้อเท็จจริงที่มีอยู่ ซึ่งกระบวนการดังกล่าวนี้จะใช้ข้อเท็จจริงที่มีอยู่ในปัจจุบัน และความรู้ที่อยู่ในฐานความรู้เพื่อก่อให้เกิดข้อเท็จจริงเพิ่มเติมต่อไปเรื่อยๆเป็นห่วงโซ่ จนกว่าข้อเท็จจริงที่เป็นเป้าหมายจะถูกค้นพบ กลไกนี้ถูกเรียกว่าการวินิจฉัย ซึ่งมีได้หลายแนวทางในการที่ความรู้ในรูปแบบของกฎจะถูกทำการวินิจฉัย ดังนั้นกลไกที่ใช้ควบคุมสามารถประกอบไปด้วยแนวทางการวินิจฉัยได้หลากหลายแนวทาง ดังนั้น ฐานความรู้ และ กลไกการวินิจฉัย จึงเป็นส่วนประกอบหลักของระบบฐานความรู้ [10] และส่วนประกอบหลักของระบบผู้เชี่ยวชาญดังแสดงในรูปที่ 2.1



รูปที่ 2.1 ส่วนประกอบหลักของระบบผู้เชี่ยวชาญ

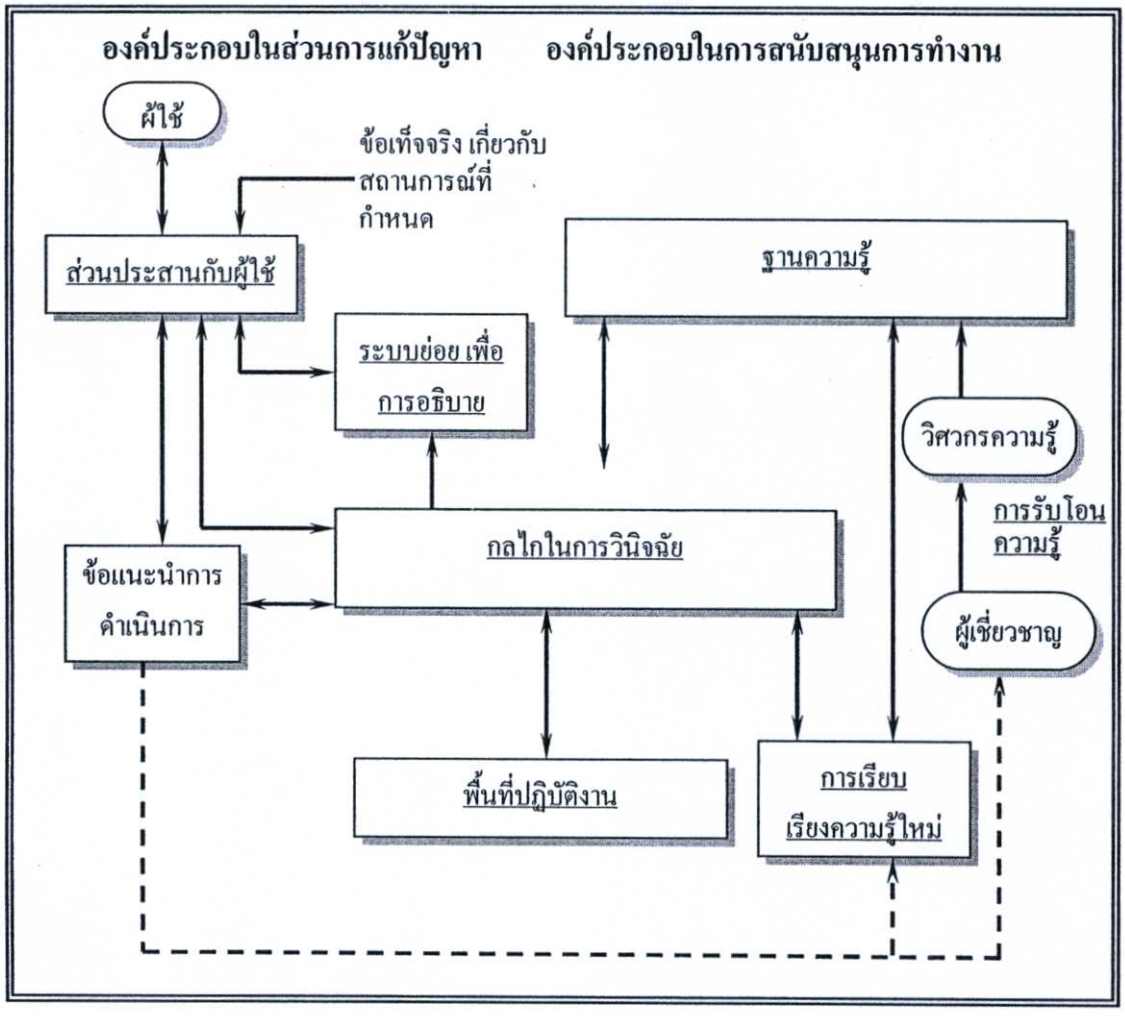
2.1.1 สถาปัตยกรรมของระบบผู้เชี่ยวชาญ

จากที่ได้อธิบายในหัวข้อข้างต้น ฐานความรู้และกลไกการวินิจฉัยจะประกอบไปด้วยกลไกหลายกลไก เมื่อระบบผู้เชี่ยวชาญเริ่มกระบวนการวินิจฉัย จะมีการร้องขอให้จัดเก็บข้อเท็จจริงเพื่อใช้งานในอนาคต ซึ่งกลุ่มของข้อเท็จจริงที่ถูกตั้งขึ้นมาจะแทนเนื้องาน(context) เช่น สถานะปัจจุบันของปัญหาที่กำลังถูกแก้ไข ดังนั้นส่วนประกอบนี้ส่วนมากจะถูกเรียกว่า เนื้องาน (context) หรือหน่วยความจำปฏิบัติงาน (working memory)

เมื่อใดก็ตามที่ระบบผู้เชี่ยวชาญมีการตัดสินใจ ระบบควรทราบว่าการตัดสินใจนั้นเกิดขึ้นได้อย่างไร นอกจากนี้เมื่อระบบผู้เชี่ยวชาญมีการร้องขอข้อเท็จจริง ระบบก็ควรทราบว่าเหตุใดข้อเท็จจริงเหล่านั้นจึงถูกร้องขอ และเมื่อระบบผู้เชี่ยวชาญใช้ความรู้ที่มีประกอบกับเนื้อหาของปัญหาในการตอบคำถาม ระบบก็ควรทราบว่าคำตอบเกิดขึ้นได้อย่างไร ซึ่งส่วนนี้ก็เป็นส่วนประกอบภายในของระบบผู้เชี่ยวชาญเช่นกัน

กระบวนการในการรวบรวม บริหารจัดการ และคอมไพล์ความรู้ พร้อมทั้งสร้างความรู้ในรูปแบบของฐานความรู้นั้นเป็นงานที่ต้องใช้ความพยายาม เพราะงานนี้ไม่ได้มีเพียงส่วนของการสร้างระบบเท่านั้น ทว่าจะรวมไปถึงการอัพเดทฐานความรู้ หรือเพิ่มความรู้เข้าไปในระบบอยู่ตลอดเวลา ซึ่ง จะพิจารณาจากการเพิ่มขึ้นของความรู้ในขอบเขตของความรู้นั้นๆ ทั้งนี้เครื่องอำนวยความสะดวกในส่วนการรับโอนความรู้ ซึ่งจะทำหน้าที่เสมือนเป็นส่วนประสานงานระหว่างผู้เชี่ยวชาญหรือวิศวกรความรู้กับฐานความรู้ สามารถนำมาใช้เป็นองค์ประกอบภายในของระบบผู้เชี่ยวชาญ เพื่อความสะดวกยิ่งขึ้นได้

ผู้ใช้ในระบบผู้เชี่ยวชาญจะติดต่อกับระบบด้วยการป้อนข้อมูล กำหนดข้อเท็จจริง และสังเกตสถานะของการแก้ปัญหา ในส่วนติดต่อกับผู้ใช้จะเป็นส่วนที่เป็นรูปแบบและข้อความที่แสดงให้ผู้ใช้ทำการติดต่อกับระบบได้ โดยรูปที่ 2.2 แสดงสถาปัตยกรรมของระบบฐานความรู้และองค์ประกอบในฐานความรู้ รวมทั้งวิธีการที่แต่ละองค์ประกอบจะติดต่อซึ่งกันและกัน



รูปที่ 2.2 สถาปัตยกรรมของระบบฐานความรู้

ดังที่ได้กล่าวมาแล้ว ฐานความรู้และกลไกการวินิจฉัยจะเป็นส่วนที่สำคัญที่สุดของระบบฐานความรู้ ดังนั้นในหัวข้อถัดไปจะอธิบายรายละเอียดเกี่ยวกับเทคโนโลยีทางด้านฐานความรู้และกลไกการวินิจฉัย

2.1.2 ฐานความรู้ (Knowledge base)

ฐานความรู้จะมีความรู้ในขอบเขตของปัญหาที่เฉพาะเจาะจงเพื่อการแก้ปัญหานั้นๆ โดยฐานความรู้ จะถูกสร้างขึ้นโดยวิศวกรความรู้ ซึ่งจะได้มาจากการสอบถามผู้เชี่ยวชาญและทำการเรียบเรียงความรู้ในรูปแบบที่ระบบสามารถนำไปใช้งานได้ วิศวกรความรู้ต้องมีเทคโนโลยีในการแทนความรู้และควรรู้ว่าจะพัฒนาระบบผู้เชี่ยวชาญอย่างไร ซึ่งก่อนที่จะตัดสินใจว่าโครงสร้างของ

ฐานความรู้จะเป็นแบบใด วิศวกรความรู้ควรจะตัดสินใจได้ว่าควรจะใช้การแทนความรู้แบบใด จึงจะเหมาะสม รวมไปถึงความเหมาะสมในการเลือกใช้เทคนิคนั้นๆภายใต้สถานการณ์ต่างๆ

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

ในการพัฒนาระบบผู้เชี่ยวชาญจะเกี่ยวข้องกับความรู้ในรูปแบบเหล่านี้ เนื่องจากความรู้ในรูปแบบเหล่านี้เป็นขอบเขตของกลไกที่ตัดสินธรรมชาติของความรู้ที่อยู่ภายในนั้น ยกตัวอย่างเช่น การพิจารณาเกี่ยวกับการวินิจฉัยสภาพอันตรายในการก่อสร้าง ซึ่งจะต้องหาว่าผนังมีความร้าวเป็นอย่างไร รวมถึงปัญหาที่สามารถทำให้ผนังร้าวได้ ทั้งนี้ระบบต้องมีความรู้เกี่ยวกับตำแหน่งของผนัง ชนิดของผนัง ความหนาของผนัง อัตราส่วนที่ใช้ในการผสมปูน ชนิดของสิ่งก่อสร้าง เป็นต้น ในส่วนของชนิดของผนังนั้น อาจมีการร้องขอ ความรู้เกี่ยวกับฐานราก ชนิดของดิน รายละเอียดของแบบคานและเสา เป็นต้น และเพิ่มเติมจากข้างต้น ความรู้เกี่ยวกับการใช้ข้อมูลที่มีอยู่ในการหาเหตุผลของการร้าวที่ผนังก็อาจถูกร้องขอด้วยเช่นกัน ซึ่งส่วนแรกของความรู้ หรือความรู้เชิงพรรณาที่ได้กล่าวถึงข้างต้นจะอธิบายเหตุการณ์ ในขณะที่ความรู้ส่วนที่สอง หรือความรู้เชิงกระบวนการจะอธิบายถึงการใช้ความรู้ในเพื่อให้ได้มาซึ่งการตัดสินใจ ดังนั้นในการพัฒนาระบบฐานความรู้จะต้องรู้ถึงความแตกต่างระหว่างการแทนความรู้แต่ละอย่าง และผลกระทบที่เกิดขึ้นจากการเลือกใช้การแทนความรู้นั้น

ในการพัฒนาระบบผู้เชี่ยวชาญนั้นจะเกี่ยวข้องกับงานอย่างเช่น การรับโอนความรู้จากผู้เชี่ยวชาญในขอบเขตงานนั้นๆ การจัดการเอกสารและการรวบรวมความรู้ การผลิตเครือข่ายความรู้ (Knowledge Net) เพื่อตรวจสอบความสัมพันธ์ระหว่างแหล่งความรู้ที่แตกต่างกัน การตรวจสอบความถูกต้องในความรู้ และสุดท้ายคือการแปลงเครือข่ายความรู้ไปเป็นโปรแกรมโดยใช้เครื่องมือที่เหมาะสม ซึ่งโปรแกรมที่ได้มานั้นจะถูกเรียกว่าระบบผู้เชี่ยวชาญ ซึ่งเป็นระบบอย่างเป็นทางการสำหรับการจัดเก็บข้อเท็จจริงและความสัมพันธ์ระหว่างข้อเท็จจริงเหล่านั้น รวมไปถึงวิธีการในการใช้ข้อเท็จจริงเหล่านั้น โดยทั่วๆไประบบผู้เชี่ยวชาญจะมีความรู้เกี่ยวกับวัตถุ ความสัมพันธ์ระหว่างววัตถุ เหตุการณ์ ความสัมพันธ์ระหว่างเหตุการณ์ และความสัมพันธ์ระหว่าง

วัตถุกับเหตุการณ์ นอกจากนี้ชนิดของกลไกการค้นหาอาจต้องถูกร้องขอเพื่อใช้ในการขับเคลื่อนระบบ ซึ่งการการตัดสินใจเลือกการแทนความรู้นั้นจะขึ้นอยู่กับชนิดของแอพพลิเคชันที่จะสร้าง และวิศวกรความรู้ต้องตัดสินว่าส่วนไหนของความรู้ ควรถูกแทนด้วยการแทนความรู้ในรูปแบบใด ซึ่งจะขึ้นอยู่กับลักษณะของความรู้และ การใช้ความรู้นั้นอย่างมีประสิทธิภาพ

ฐานความรู้และกลไกการวินิจฉัยจะเป็นส่วนที่สำคัญที่สุดของระบบฐานความรู้ ดังนั้นในหัวข้อถัดไปจะอธิบายรายละเอียดเกี่ยวกับเทคโนโลยีทางด้านฐานความรู้และกลไกการวินิจฉัย

2.2 การแทนความรู้ (Knowledge Representation)

ในเวลาที่ผ่านมา ได้มีการศึกษาวิจัย และกำหนดรูปแบบการสร้างตัวแทนความรู้ไว้หลายรูปแบบ ซึ่งสามารถจัดเป็นกลุ่มใหญ่ ที่ได้รับความนิยมได้ 3 รูปแบบดังนี้ แบบใช้กฎ (Production Rule) แบบเครือข่ายความหมาย (Semantic Network) และแบบใช้เฟรม (Frame) ซึ่งในวิทยานิพนธ์ฉบับนี้ จะขอกล่าวถึงเฉพาะแบบใช้กฎ และแบบใช้เฟรมเท่านั้น เนื่องจากในการสร้างตัวแทนความรู้แบบใช้กฎนั้นเป็นวิธีพื้นฐานที่นิยมใช้ในการพัฒนาระบบผู้เชี่ยวชาญ และการสร้างตัวแทนความรู้แบบใช้เฟรม เป็นรูปแบบที่นิยมใช้ในระบบผู้เชี่ยวชาญที่มีฐานความรู้ขนาดใหญ่ ซึ่งเป็นรูปแบบที่วิทยานิพนธ์ฉบับนี้เลือกใช้เป็นแนวทางในการพัฒนาต้นระบบผู้เชี่ยวชาญ

2.2.1 การแทนความรู้แบบกฎ (Production Rules)

เป็นรูปแบบที่ไม่ซับซ้อนแต่มีประสิทธิภาพในการแทนความรู้ ซึ่งจะมีความยืดหยุ่นในการผสมผสานความรู้เชิงพรรณากับความรู้เชิงกระบวนการ โดยในการแทนความรู้แบบกฎนี้จะประกอบไปด้วยเซตของเหตุ (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

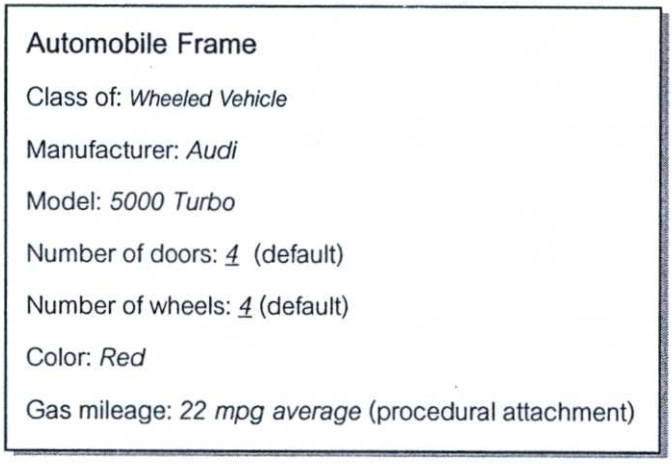
IF the object has two openings
AND opening at top is larger than that at the bottom
AND enough space is available on left side
AND space available on right side is too small
THEN take the pipe by the left side of the object
AND take pipe into the object from top opening

เหตุการณ์ที่ระบุในกฎที่หนึ่งและสองนั้นเรียบง่าย แต่กฎในข้อที่สามนั้นซับซ้อน ค่าที่ถูกใส่ให้กับตัวแปรสามารถถูกใช้แทนข้อเท็จจริงในกฎข้อที่หนึ่งและสอง แต่ในกรณีกฎข้อที่สี่ วัตถุและคุณสมบัติของกฎต้องถูกกำหนดในการแทนข้อเท็จจริง ซึ่งในกรณีทั้งหมดข้างต้น เมื่อส่วนเหตุ (IF) ของกฎนั้นเป็นจริงโดยข้อเท็จจริงที่ถูกจัดเก็บในส่วนเนื้อหา หรือข้อเท็จจริงนั้นถูกป้อนเข้ามาโดยผู้ใช้ การกระทำที่ระบุไว้ในส่วนผล (THEN) จะถูกกระทำและกฎจะทำงาน โดยทั่วไปแล้วฐานความรู้จะประกอบไปด้วยกฎหลายๆกฎ ซึ่งกฎจะถูกรวบรวมอยู่ในฐานกฎ (rule bases)

2.2.2 การแทนความรู้แบบเฟรม (Frames)

การแทนความรู้แบบเฟรมได้ถูกเสนอเป็นครั้งแรกโดย Marvin Minsky ในปี 1974 [13] และได้มีการนำเสนอแนวคิดนี้ในงานวิจัยจนถึงปัจจุบัน [1] แนวคิดของเฟรมคือโครงสร้างข้อมูล (data structure) สำหรับรวบรวมข้อมูลชนิดต่างๆของแนวคิดหรือความรู้ (typical knowledge) และเรียกว่าโครงสร้างแบบนี้ว่าเฟรม [12, 14] ดังนั้นเฟรมจึงเป็นการแทนความรู้แบบหนึ่ง ซึ่งมีทั้งโครงสร้างข้อมูลในการแทนความรู้และความสามารถในการวินิจฉัยด้วย นอกจากนี้เฟรมยังเหมาะสำหรับเป็นตัวแทนของแนวคิดในการจำแนกประเภท (classification) และยังเหมาะที่จะเป็นตัวแทนของอนุกรมวิธาน (Taxonomy) หรือการจำแนกหมวดหมู่แบบลำดับขั้นด้วย (Hierarchy) [12, 15] โดยในเอนทิตี้ (Entity) หนึ่งของเฟรมจะรวบรวมความรู้ที่จำเป็นทั้งหมดเกี่ยวกับวัตถุหรือแนวคิดนั้น

2.2.2.1 โครงสร้างของเฟรม โครงสร้างของเฟรมประกอบไปด้วย ชื่อของเฟรมที่ใช้อ้างถึงเฟรม, สล๊อตหรือ แอททริบิวต์ของเฟรม และ ฟาเซท[1, 14] ทั้งนี้เฟรมมีหลากหลายความหมาย ซึ่งสามารถอ้างถึงวัตถุเฉพาะเจาะจง (Particular Object) หรือ อ้างถึงกลุ่มของวัตถุที่มีคุณสมบัติเหมือนกัน (Group of Similar Object) ก็ได้ เพื่อให้สามารถแยกแยะความแตกต่างนี้ เฟรมจึงสามารถจำแนกได้เป็น คลาสเฟรม และ อินสแตนซ์เฟรม



รูปที่ 2.3 ตัวอย่างโครงสร้างของเฟรม

คลาสเฟรม : จะใช้เพื่ออธิบายกลุ่มของวัตถุ หรือ คลาสของวัตถุ และสามารถนำไปรวบรวมแบบเป็นอนุกรมวิธาน (Taxonomy) ได้ด้วย ซึ่ง คลาสเฟรม จะมี สล๊อตที่เหมือนกันในทุกๆ คลาสเฟรม คือ สล๊อตแม่และ สล๊อตลูก ยกเว้น รูธคลาส ที่ไม่มี สล๊อตแม่ โดยที่ สล๊อตอื่นๆ จาก เฟรมแม่สามารถสืบทอดไปยังเฟรมลูกได้ด้วยการ สืบทอด (Inheritance)

อินสแตนซ์เฟรม : จะใช้ในการอธิบายถึงวัตถุเฉพาะเจาะจง ซึ่งส่วนมากเป็น โนดสุดท้าย ของ อนุกรมวิธาน และเป็นเฟรมที่ไม่มี สล๊อตลูก อีกแล้ว เป็น เฟรมที่ใช้ในการอ้างถึง วัตถุเฉพาะเจาะจง โดยที่ อินสแตนซ์เฟรม จะมีเพียง สล๊อตที่บอกความสัมพันธ์ เป็นสล๊อตแม่เท่านั้น และรับถ่ายทอดคุณสมบัติหรือ สล๊อตต่างๆ มาจาก คลาสเฟรม โดยที่อาจจะมีการเปลี่ยนแปลงค่าใน สล๊อตที่ อินสแตนซ์เฟรม ก็ได้ เพราะฉะนั้น ค่าของ สล๊อตใน อินสแตนซ์เฟรม จึงไม่จำเป็นต้องมีค่าเดียวกันกับ คลาสเฟรม เสมอไป ในอีกนัยหนึ่งอาจกล่าวได้ว่า คลาสเฟรม นั้นจะทำหน้าที่เสมือนเป็นต้นแบบของ อินสแตนซ์เฟรม นั่นเอง [1, 2, 14, 15, 16]

สล๊อต : ในแต่ละเฟรมจะประกอบไปด้วย สล๊อตหรือ แอททริบิวต์เพื่อบอกคุณสมบัติของเฟรมนั้นๆ ในเฟรมหนึ่งๆอาจมีได้หลาย สล๊อตและบางทีอาจกล่าวได้ว่า แนวคิดของเฟรมนั้นจะถูกกำหนดโดยการรวบรวม สล๊อตเข้าด้วยกัน [12] ในแต่ละ สล๊อตจะอธิบายถึงคุณสมบัติ หรือการดำเนินการของเฟรม และในแต่ละ สล๊อตจะมีค่าของ สล๊อตซึ่งเรียกว่า ค่าในสล๊อต หรือ ฟิลเลอร์ [15] ซึ่งค่าของ สล๊อตนั้นอาจเป็นได้ทั้งค่าโดยปริยาย (Default Value), เป็นพอยน์เตอร์ (Pointer) ชี้ไปยังเฟรมอื่น หรือเป็นเซตของกระบวนการที่ค่านั้นได้มาก็ได้ [11, 14]

ฟาเซท : ในแต่ละ สล๊อตจะมี ฟาเซทซึ่งอาจมีมากกว่า หนึ่ง ฟาเซทในหนึ่ง สล๊อต เพื่อใช้ควบคุมค่าของ สล๊อตและตัวดำเนินการใน สล๊อตนั้น ฟาเซทยังสามารถนำมาใช้เพื่อกำหนดค่าเริ่มต้นให้กับ ค่าของสล๊อต, กำหนดชนิดของข้อมูล, กำหนดขอบเขตของค่าเป็นไปได้ และกำหนดd กระบวนการที่จะกระทำการต่อไปได้ โดยชนิดของฟาเซทจะมีดังนี้ [14]

Value Type : เป็น ฟาเซทประเภทพื้นฐานที่อธิบายชนิดของค่าใน สล๊อตเป็นการกำหนดประเภทของค่าใน 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 ฟาเซทแบบกระบวนการที่ระบุกระบวนการที่จะเกิดขึ้น เมื่อมีการเพิ่มค่าเข้ามาในสล๊อตบางครั้งเราอาจเรียกว่าเป็น ดีมอน (demons)

If needed : เป็น ฟาเซทประเภทเดียวกับ *If added* แต่จะทำงานเมื่อไม่มีการระบุค่าให้ค่าของสล๊อต ที่มีความต้องการ เพื่อใช้ประมวลผลขณะนั้น

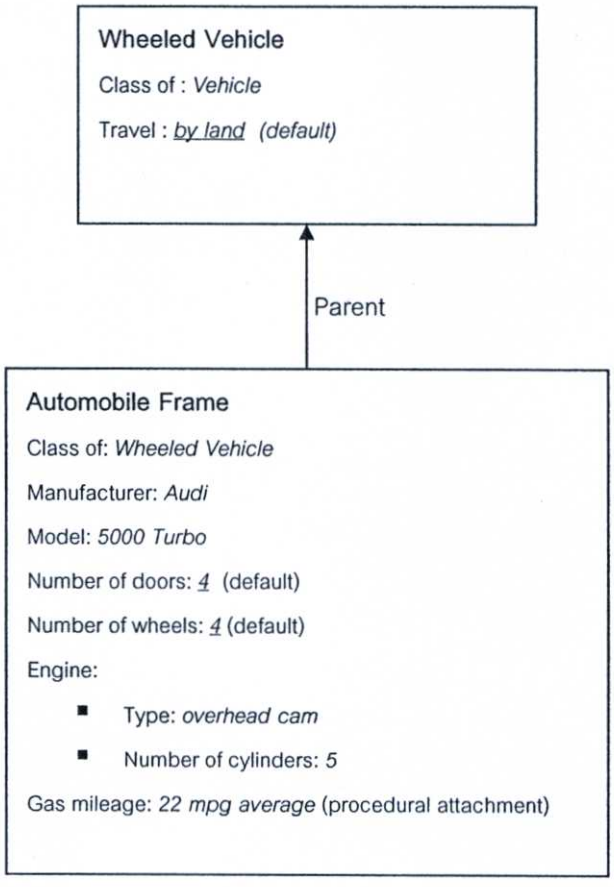
If removed : เป็น ฟาเซทประเภทเดียวกับ *If added* แต่จะทำงานเมื่อค่าภายใน สล๊อตถูกลบออก

ฟาเซทสามารถมีโครงสร้างเป็นกฎได้ ซึ่งหมายความค่าของ สล๊อตหรือ ฟาเซท สามารถกำหนดให้เป็นค่าในลักษณะหลายค่า (Multi-Value) ได้ ดังนั้นจึงสรุปได้ว่าค่าของ ฟาเซทนั้น ไม่จำเป็นต้องมีคุณสมบัติอะตอมมิค (Atomic) นั่นเอง

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

นอกจากนี้ เนื่องจาก ฟาเซทเป็นกฎที่ใช้ในการเพิ่มค่าเข้าไปได้ด้วย ทำให้เมื่อมีค่าที่เพิ่มเข้ามาใหม่ใน ฟิลเลอร์ แล้ว ฟาเซทจะมีหน้าที่ไปตรวจสอบค่าใน สล๊อตอื่นๆ หรือเมื่อมีการร้องขอดูค่าใน ฟิลเลอร์ ของ สล๊อตนั้นๆ ฟาเซทก็สามารถร้องขอค่าของ ฟิลเลอร์ ที่ สล๊อตของเฟรมแม่ได้ ดังนั้น ฟาเซทจึงมีความสามารถในการวินิจฉัยได้ด้วย ซึ่งการวินิจฉัยแบบนี้เป็นการวินิจฉัยแบบสืบทอด ดังนั้นจึงกล่าวได้ว่าระบบการแทนความรู้แบบเฟรม จะสามารถทำการวินิจฉัยโดยการสืบทอดโดยอัตโนมัตินั่นเอง

ในรูปที่ 2.4 เป็นตัวอย่างโครงสร้างและความสัมพันธ์ของเฟรม โดยที่ เฟรมรถยนต์ (Automobile Frame) จะมีความสัมพันธ์กับเฟรมเครื่องยนต์ที่มีล้อ (Wheeled Vehicle) โดยเฟรมรถยนต์จะเป็นเฟรมลูกของ ของเฟรมเครื่องยนต์ที่มีล้อ เป็นต้น



รูปที่ 2.4 ตัวอย่างโครงสร้างและความสัมพันธ์ของเฟรมรถยนต์

2.2.2.2 การใช้เทคนิคการแทนความรู้แบบเฟรมในงานวิจัยทั่วไป สำหรับเทคนิคการแทนความรู้แบบเฟรมนี้ เป็นที่ได้รับความสนใจ และนิยมนำไปใช้พัฒนาในงานวิจัยหลายด้าน โดยเฉพาะงานวิจัยที่ต้องการสร้างตัวแทนความรู้สำหรับความรู้ที่มีโครงสร้างซับซ้อน เพราะเป็นที่ยอมรับกันโดยทั่วไปว่า หลักแนวคิดในแบบเฟรม ซึ่งมองรายละเอียดโครงสร้างความรู้ในรูปแบบใกล้เคียงกับวัตถุ จะสามารถรองรับการประมวลผลความรู้ที่ใกล้เคียงกับโลกแห่งความเป็นจริงได้มากขึ้น ถึงแม้ยังจะพบข้อจำกัดบางประการอยู่ [1] แต่ก็ยังมีนักวิจัยที่พยายามพัฒนารูปแบบการแทนความรู้แบบเฟรมขึ้นมาอยู่เสมอ ดังจะเห็นได้จากการพยายามนำเทคนิคการแทนความรู้แบบเฟรมนี้มาพัฒนางานทั้งด้านการแพทย์ [1, 16], งานวิศวกรรม [14] , การออกแบบพัฒนาซอฟท์แวร์ [4] รวมถึงการพัฒนาระบบผู้เชี่ยวชาญด้านอื่นๆ โดยทั่วไป [2] โดยเครื่องมือการพัฒนาส่วนใหญ่ที่

ใช้มักจะอยู่ในรูปของภาษา, เครื่องมือ หรือเปลือกระบบผู้เชี่ยวชาญที่ยึดมาตรฐานตามภาษาด้านปัญญาประดิษฐ์ในกลุ่มภาษา ลิสพ์ (LISP)หรือ โปรลอก (Prolog) เป็นต้น ซึ่งเครื่องมือที่ได้รับความนิยมส่วนใหญ่ได้แก่ Smalltalk, KEE, Kappa-PC, KL-One, Protégé เป็นต้น

2.3 กลไกการวินิจฉัยบนระบบผู้เชี่ยวชาญ (Inference Mechanismsm)

เป็นส่วนควบคุมหรือเทคนิคการค้นหาซึ่งจะค้นหาทั้งฐานความรู้เพื่อการตัดสินใจ ฐานความรู้จะเป็นส่วน state space และกลไกการวินิจฉัยจะเป็นกระบวนการค้นหา กระบวนการวินิจฉัยจะใช้สัญลักษณ์โดยการเลือกกฎที่สอดคล้องกับสัญลักษณ์ของข้อเท็จจริงและเรียกใช้กฎเพื่อให้ได้ข้อเท็จจริงใหม่ ซึ่งกระบวนการนี้จะกระทำต่อเนื่องเป็นห่วงโซ่จนกว่าจะพบเป้าหมาย ในระบบผู้เชี่ยวชาญนั้นกระบวนการวินิจฉัยสามารถถูกกระทำได้หลายทาง ซึ่งกระบวนการวินิจฉัยที่นิยมใช้กันมากสามวิธีคือ แบคเวิร์ดเชนนิง ฟอร์เวิร์ดเชนนิง และ แบบผสมโดยที่ แบคเวิร์ดเชนนิง คือ กระบวนการที่ขับเคลื่อนโดยเป้าหมาย (Goal-driven Process) ในขณะที่ ฟอร์เวิร์ดเชนนิง เป็นกระบวนการที่ขับเคลื่อนโดยข้อมูล (Data-driven Process)

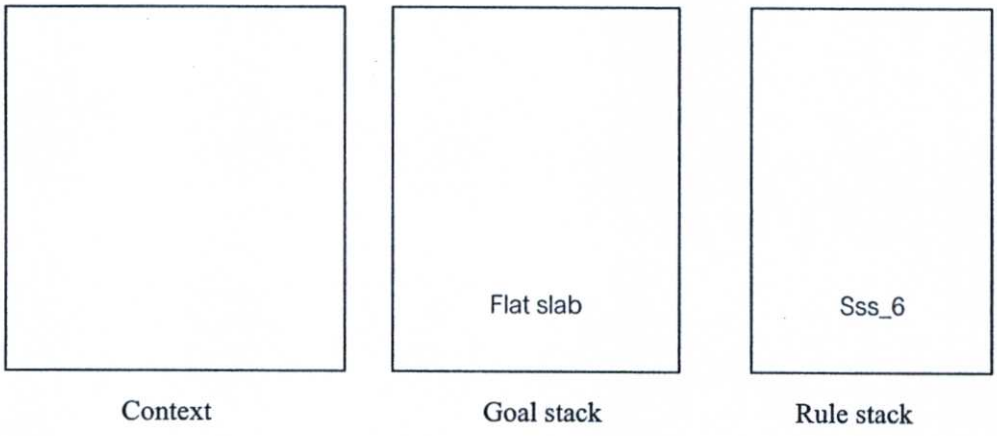
2.3.1 แบคเวิร์ดเชนนิง (Backward Chaining)

เป็นกระบวนการที่ขับเคลื่อนโดยเป้าหมาย โดยการตั้งเป้าหมายที่อยู่ในฐานความรู้ขึ้นมาเป็นตัวแปร และกระบวนการวินิจฉัยจะหยุดทำงานเมื่อได้ค่าของตัวแปร โครงสร้างข้อมูลที่ใช้ในระหว่างกระบวนการวินิจฉัยคือส่วนหน่วยความจำปฏิบัติงาน (Working Memory) หรือ เนื้องาน (Context) ส่วน สแต็กของกฎ (Rule Stack) และส่วนสแต็กของเป้าหมาย (Goal Stack) ซึ่งเมื่อไหร่ก็ตามที่เกิดการกระทำอย่างหนึ่งในกระบวนการวินิจฉัย เช่น การเลือก (Select) การจับคู่ (Match) และการเรียกใช้กระบวนการ (Execute) เกิดขึ้น ข้อมูลที่ใช้ในส่วนกระบวนการวินิจฉัยเหล่านั้นจะถูกเปลี่ยนแปลง การศึกษาถึงผลกระทบของการกระทำบนโครงสร้างข้อมูลระหว่างกระบวนการวินิจฉัยเป็นทางเลือกที่ดีกว่าในการศึกษากระบวนการของกลไกการวินิจฉัย ดังนั้นการทำงานของกระบวนการวินิจฉัยจะถูกอธิบายผ่านการเปลี่ยนแปลงที่เกิดขึ้นกับส่วนเนื้องาน ส่วนสแต็กของกฎ และส่วนสแต็กของเป้าหมาย ซึ่งในส่วนสแต็กของกฎ และสแต็กเป้าหมายจะเป็นโครงสร้างข้อมูลชั่วคราวที่สร้างขึ้นเพื่อจัดเก็บ

โดยเมื่อเริ่มกระบวนการในส่วนของเนื้องานจะว่างเปล่าและตัวแปรเป้าหมายจะถูกดันเข้าไปในสแต็กเป้าหมาย จากนั้นกระบวนการจะทำการเลือกกฎแรกในฐานกฎซึ่งมีตัวแปรเป้าหมายในส่วน THEN และดันเข้าไปในสแต็กของกฎ ซึ่งตัวอย่างอย่างกฎข้อที่ 6 จะเป็นกฎแรกที่มีตัวแปร

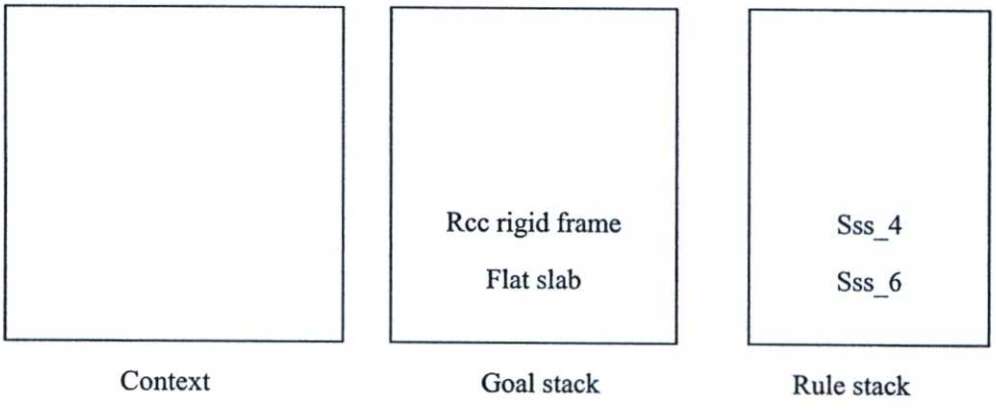
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เป้าหมายในส่วนผล ดังนั้นกฎข้อที่ 6 จะถูกดันเข้าไปในสแต็กของกฎและตัวแปรเป้าหมายจะอยู่ในสแต็กเป้าหมาย สถานะของเนื้องาน สแต็กเป้าหมาย และสแต็กของกฎแสดงในรูปที่ 2.5 ดังนี้



รูปที่ 2.5 สถานะของเนื้องาน สแต็กเป้าหมาย และสแต็กของกฎ I

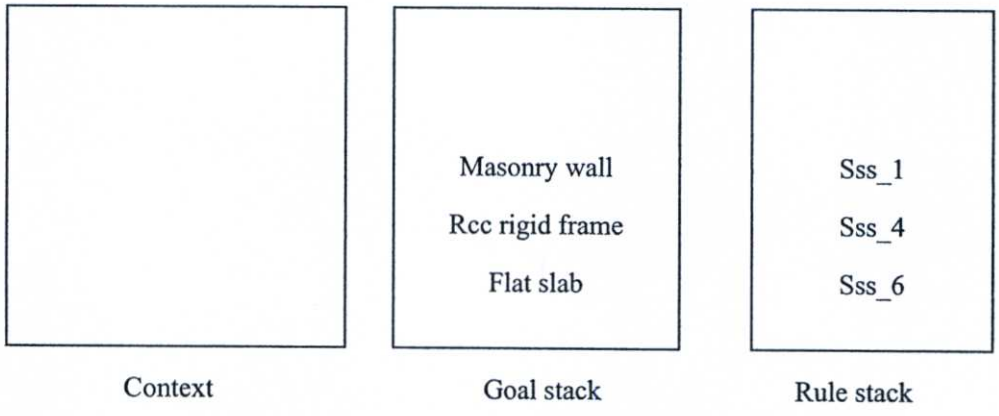
จากนั้น กฎข้อที่ 6 เป็นกฎแรกที่มีตัวแปรเป้าหมายในส่วนของผล ซึ่งเมื่อกฎข้อที่ 6 ถูกเลือกขึ้นมา ระบบก็จะเริ่มทำการจับคู่กับเหตุในส่วนที่ตัวแปรเป้าหมายปรากฏขึ้นในกฎซึ่งเงื่อนไขแรกคือ "structural system IS rcc rigid frame" การจับคู่กฎสามารถกระทำต่อไปได้ก็ต่อเมื่อข้อเท็จจริงนี้ได้ถูกตั้งไว้แล้ว ดังนั้นกระบวนการวินิจฉัยจะกำหนดให้ตัวแปร "structural system" เป็นเป้าหมายปัจจุบัน ซึ่งจะถูกกำหนดโดยการดันตัวแปรเข้าไปในสแต็กเป้าหมายและ ดันกฎแรกในลิสต์ที่มีตัวแปรเป้าหมายในส่วน THEN เข้าไปในสแต็กของกฎด้วย ดังนั้นกฎข้อที่ 4 จะถูกเลือกสำหรับการหาค่า โดยสถานะของเนื้องานและสแต็กทั้งสองแสดงในรูปที่ 2.6



รูปที่ 2.6 สถานะของเนื้องาน สแต็กเป้าหมาย และสแต็กของกฎ II

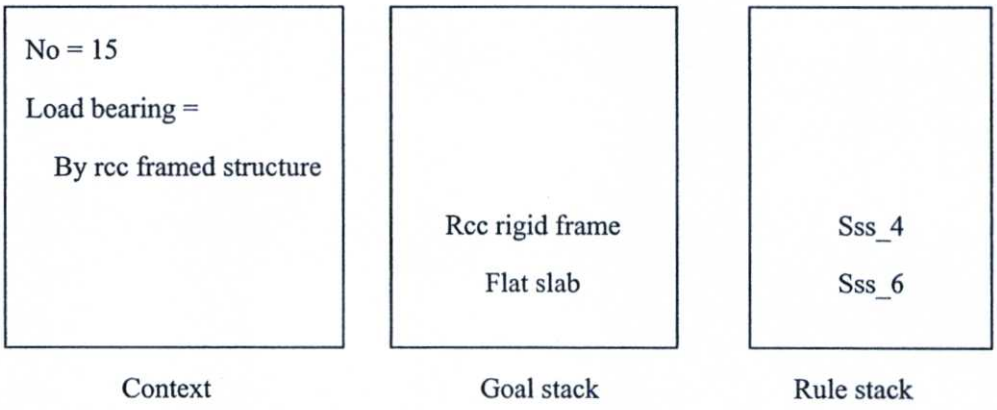
105115

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



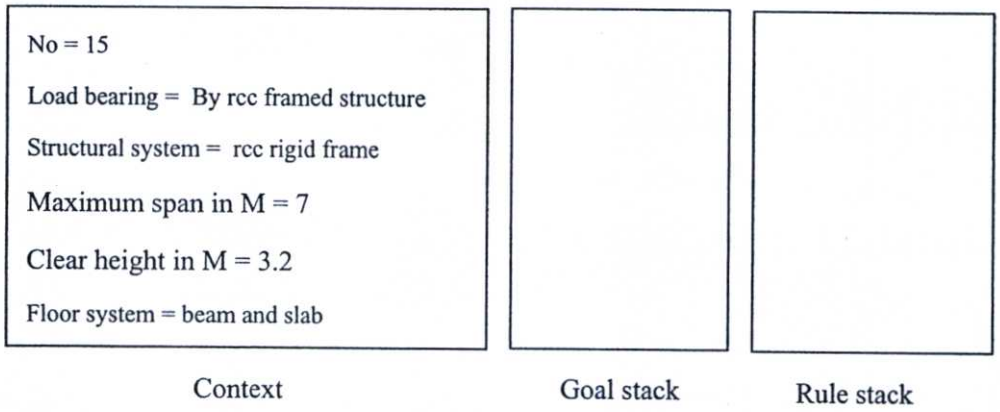
รูปที่ 2.7 สถานะของเนื้องาน สแต็กเป้าหมาย และสแต็กของกฎ III

เงื่อนไขแรกของกฎข้อที่ 1 นั้นคือ "no of stories" ซึ่งร้องขอค่าให้ผู้ใช้ป้อนค่าสำหรับตัวแปรนี้ ซึ่งสมมติให้ผู้ใช้ป้อนค่าเป็น 15 ซึ่งเป็นข้อเท็จจริงแรกและจะถูกจัดเก็บอยู่ในเนื้องาน 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 ซึ่งทำการผลลัพธ์ที่ได้จากเงื่อนไขนี้มีค่าเป็นเท็จและกฎถูกปฏิเสธและถูกดึงออกจากสแต็ก โดยที่กฎถัดไปซึ่งมีเป้าหมายปัจจุบันในส่วนของผลจะถูกเลือกขึ้นมาแทน ในที่นี้คือกฎข้อที่ 7 ซึ่งจะถูกปฏิเสธด้วยเหมือนกันเนื่องจากเงื่อนไขข้อที่สองเป็นเท็จ ดังนั้นกฎข้อที่ 8 จะถูกเลือกจะใส่เข้าไปในสแต็กกฎ ซึ่งเงื่อนไขสองอันแรกเป็นจริงจะผู้ใช้จะต้องใส่ค่าตัวแปรในเงื่อนไขที่สาม โดยให้ผู้ใช้ป้อนค่าเข้ามาเป็น 3.2 ซึ่งจะได้ค่าเป็นจริงและกฎจะถูกใช้ ซึ่งจะได้ผลลัพธ์เป็นค่า 'beam and slab' สำหรับตัวแปรเป้าหมาย ซึ่งเมื่อกฎถูกใช้ทั้งสองสแต็กจะถูกดึงออกจะข้อเท็จจริงใหม่จะถูกเพิ่มเข้ามายังเนื้องาน ซึ่งตอนนี้ทั้งสองสแต็กว่างเปล่าและกระบวนการวินิจฉัยก็สิ้นสุดลงโดยสถานะของเนื้องานของตัวอย่างนี้แสดงในรูปที่ 2.9



รูปที่ 2.9 สถานะของเนื้องาน สแต็กเป้าหมาย และสแต็กของกฎ V

ในส่วนการอธิบายการดำเนินงานของระบบผู้เชี่ยวชาญจะมีกลไกสำหรับการสอบถามเนื้องานในการได้มาซึ่งค่าของตัวแปร (what?) และรู้ว่าข้อเท็จจริงเกิดขึ้นได้อย่างไร (how?) คำตอบสำหรับคำถามแรกจะได้มาจากส่วนเนื้องานโดยตรง ในขณะที่คำตอบที่ได้จากคำถามที่สองจะต้องการการอ้างอิงถึงฐานความรู้ โดยกระบวนการที่เกี่ยวข้องกับคำถามว่าเกิดขึ้นได้อย่างไรจะค้นหากฎซึ่งต้องการข้อเท็จจริงในส่วน THEN และแสดงกฎซึ่งแสดงให้เห็นว่าข้อเท็จจริงเกิดขึ้นได้อย่างไร การได้มาซึ่งคำตอบของคำถามแล้วถ้า (what if?) จะมีประโยชน์อย่างมากถ้าผู้ใช้ของระบบต้องการทราบกระบวนการตัดสินใจโดยการเปลี่ยนค่าที่ป้อนเข้าไป ซึ่งสามารถทำได้ในสองขั้นตอน ขั้นแรกข้อเท็จจริงเหล่านั้นที่ขึ้นอยู่กับการเปลี่ยนแปลงตัวแปรต้องถูกยกเลิกก่อน จากนั้นจึงจะถูกกำหนดขึ้นใหม่โดยกระบวนการวินิจฉัยจากจุดนั้น ซึ่งจะค่อนข้างซับซ้อนและต้องมีการรักษาเส้นทางการวินิจฉัยที่เสร็จสิ้นแล้วด้วย โดยจะขึ้นอยู่กับข้อเท็จจริงที่อยู่ในเนื้องาน กระบวนการวินิจฉัยจะสร้างเครือข่ายที่เกี่ยวข้องกันทั้งหมดจาก TMS (Truth Maintenace System) ซึ่งเครือข่ายนี้จะช่วยในการย้อนกลับ (Backtrack) จากขั้นตอนใดๆไปยังขั้นตอนก่อนหน้าโดยยกเลิกการกระทำที่เกี่ยวข้อง ซึ่งจะได้ผลลัพธ์ในการลบข้อเท็จจริงต่างๆออกไปจากส่วนเนื้องาน ซึ่งการย้อนกลับนี้จะกระทำต่อไปเรื่อยๆจนกว่าจะถึงส่วนก่อตั้งตัวแปรข้อเท็จจริง โดยค่าของตัวแปรดังกล่าวจะถูกเปลี่ยนแปลงด้วยคำถามแล้วถ้า กระบวนการวินิจฉัยจะเริ่มใหม่จากจุดนั้น และจะต่อเนื่องไปเรื่อยๆจนกว่าเป้าหมายสุดท้ายจะเกิดขึ้น และเนื่องจากความยากที่เกิดขึ้นในการบำรุงรักษาเครือข่ายที่ขึ้นต่อกันของการตัดสินใจ ทำให้เครื่องมือที่ใช้ในเชิงพาณิชย์ต่างๆ ไม่มีส่วนนี้

บรรลุเป้าหมายได้ หรือถ้าไม่เพียงพอก็จะหยุดโดยแจ้งว่าไม่สามารถบรรลุเป้าหมายได้ด้วยข้อมูลที่มีอยู่ ดังนั้นกระบวนการแบบฟอร์เวิร์ด จะเหมาะสมเมื่อมีสถานะเริ่มต้นน้อยๆแต่มีได้หลากหลายเป้าหมายเท่านั้น ซึ่งระบบผู้เชี่ยวชาญขนาดใหญ่ จะมีได้หลากหลายสถานเริ่มต้น และหลายเป้าหมาย ดังนั้นผู้ใช้อาจไม่สามารถให้ข้อมูลที่จำเป็นทั้งหมดก่อนได้ และผู้ใช้อาจไม่สามารถมองออกว่าการไหลของกระบวนการวินิจฉัยเป็นอย่างไร ซึ่งลักษณะนี้กลไกการวินิจฉัยแบบผสมอาจเหมาะสมกว่า โดยกลไกการวินิจฉัยแบบผสมจะเริ่มจากการทำงานแบบ ฟอร์เวิร์ด ก่อนเพื่อให้ได้ข้อเท็จจริงบางส่วนจากนั้นจะใช้การทำงานแบบแบคเวิร์ด เพื่อพิสูจน์ข้อเท็จจริงเหล่านั้น ซึ่งกระบวนการแบบผสมนี้กระบวนการแบบฟอร์เวิร์ด จะถูกเรียกขึ้นมาเมื่อข้อเท็จจริงถูกตั้งขึ้น ผู้ใช้จะไม่ต้องป้อนข้อมูลทั้งหมดตั้งแต่เริ่มต้น แต่จะป้อนข้อมูลส่วนหนึ่งเข้ามาก่อนแล้วป้อนอีกครั้งเมื่อมีการร้องขอ จากนั้นจึงจะเรียกใช้กระบวนการแบบแบคเวิร์ด เพื่อพิสูจน์ข้อเท็จจริงเหล่านั้นไปสู่เป้าหมาย

ถ้าเฟรมถูกใช้กับกฎในการแทนความรู้ ข้อเท็จจริงที่เกี่ยวข้องกับตัวแปรที่ถูกในเฟรมจะถูกเก็บในเนื้องานแยกออกมาต่างหากเรียกว่าฐานเฟรม โดยฐานเฟรมสามารถมีได้หลายเฟรมอยู่ภายในซึ่งชนิดของสล๊อตมีได้ทั้ง concrete หรือ abstract และเฟรมยังสามารถถูกเชื่อมโยงโดยใช้ความสัมพันธ์อย่างมีความหมายซึ่งสามารถสร้างเครือข่ายความหมายสำหรับแทนความรู้แบบพรรณาได้ แต่โดยทั่วๆไปนั้นการทำเชนนิง (Chaining) จะถูกกระทำบนตัวแปรที่อยู่ในฐานกฎและไม่อยู่ในตัวแปรเฟรม ดังนั้นทุกๆตัวแปรกที่กำหนดในกระบวนการวินิจฉัยจะถูกกำหนดเป็นตัวแปรฐานกฎทั้งหมด

2.3.4 การวินิจฉัยโดยใช้ความรู้ในการแทนความรู้แบบเฟรม

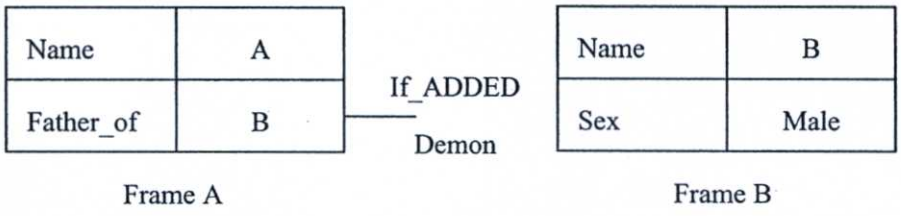
ในกรณีที่มีชนิดของการวินิจฉัยที่แตกต่างกันสามารถทำได้ด้วยการแทนความรู้แบบเฟรม การวินิจฉัยเพียงอย่างเดียวจะไม่มีรูปแบบควบคุมวิธีการทั้งหมดในการแก้ปัญหาอย่างในกรณีการวินิจฉัยบนฐานกฎ แต่การวินิจฉัยในลักษณะนี้จะแสดงลักษณะความฉลาดของมนุษย์หลายอย่างที่ใช้ในกระบวนการแก้ปัญหา ลักษณะอย่างหนึ่งคือสามัญสำนึก ดังตัวอย่างต่อไปนี้ซึ่งแสดงการจำลองสามัญสำนึกโดยใช้เฟรม

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

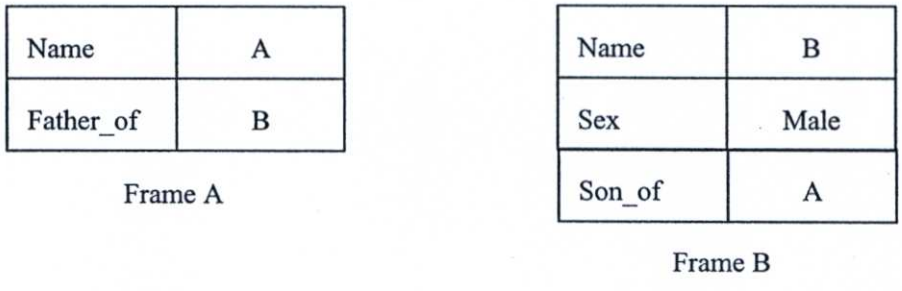
ลักษณะพื้นฐานสองประการของระบบการจัดการเฟรมคือเฟรมจะถูกสร้างระหว่างเวลาปฏิบัติการ (Run Time) และสล๊อตกับค่าของสล๊อตจะถูกเพิ่มเข้าไประหว่างกระบวนการวินิจฉัย

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 ซึ่งดีมอน (กระบวนการทำงาน) ชนิด if_added จะถูกผูกติดอยู่กับสล๊อต father_of ของเฟรม A ซึ่งดีมอนนี้จะถูกกระตุ้นขึ้นมาเมื่อค่าถูกใส่เข้าไปในสล๊อต รูปที่ 2.11 จะแสดงสถานะของเฟรมก่อนถูกกระตุ้นด้วยดีมอน

เมื่อค่าถูกใส่เข้าไปในสล๊อต father_of ของเฟรม A ดีมอนที่ผูกติดอยู่กับสล๊อตจะถูกปล่อยออกมาอย่างอัตโนมัติซึ่งจะเพิ่มสล๊อตที่ชื่อว่า son_of ให้กับเฟรม B และใส่ค่าของสล๊อตเป็น A ให้กับมัน ความรู้ในเรื่องความสัมพันธ์จะถูกห่อหุ้มไว้ในส่วนดีมอนที่ผูกติดอยู่กับสล๊อต status ของเฟรมหลังจากการอ้างของดีมอนที่ผูกติดอยู่กับสล๊อต father_of ของเฟรม A ดังแสดงในรูปที่ 2.12



รูปที่ 2.11 สถานะของเฟรม A และ B ก่อนที่จะถูกกระตุ้นด้วยดีมอน



รูปที่ 2.12 สถานะของเฟรม A และ B หลังจากถูกกระตุ้นด้วยดีมอน

วิศวกรความรู้ผู้ออกแบบฐานความรู้จะผูกดีมอนเข้ากับสล๊อตซึ่งจะถูกกระตุ้นโดยอัตโนมัติเมื่อมีเกิดเหตุการณ์ในการใส่ค่าเข้าไปในสล๊อต ซึ่งแบบจำลองการวินิจฉัยนี้มาจากสามัญสำนึก และผลกระทบเดียวกันสามารถถูกกระทำได้โดยการผูกดีมอนที่ชื่อว่า if_needed กับสล๊อต son_of ของ

เฟรม B ซึ่งดีมอนนี้จะถูกปลุกขึ้นมาถ้ามีการอ้างถึงสล๊อต son_of ของเฟรม B ซึ่งข้อได้เปรียบของวิธีการนี้คือว่าค่าจะถูกวินิจฉัยและเพิ่มเข้าไปในสล๊อตก็ต่อเมื่อมีการอ้างถึงสล๊อตนั้น ดีมอนอีกชนิดหนึ่งคือ if_deleted ซึ่งดีมอนชนิดนี้จะถูกปลุกขึ้นมาถ้าสล๊อตถูกลบออกไปจากเฟรม ซึ่งดีมอนนี้จะมีประโยชน์อย่างมากในปัญหาวิศวกรรมหลายๆปัญหา ซึ่งเนื้องานจะถูกอัพเดตระหว่างกระบวนการแก้ปัญหาที่มาจากการเปลี่ยนแปลงสมมติฐานหรือมาจากค่าที่ถูกคำนวนขึ้นมา ชนิดของการวินิจฉัยที่หลากหลายนี้สามารถถูกจำลองโดยแนวคิดของดีมอน

ระบบผู้เชี่ยวชาญที่ใช้วิธีการแทนความรู้แบบเฟรมส่วนใหญ่นั้น มักจะใช้เทคนิคการสืบทอดในการวินิจฉัย และแก้ปัญหาบนพื้นฐานของความรู้ที่จัดเก็บไว้ในฐานความรู้ โดยลักษณะการทำงานของกลไกการวินิจฉัยนั้น มีทั้งรูปแบบในการใช้กระบวนการเป็นหลัก ซึ่งเป็นการจัดเก็บกระบวนการต่างๆ ไว้ที่ ค่าของฟาเซท ใน แต่ละสล๊อต และจะมีตัวแปลภาษา (Interpreter) คอยกระทำ กระบวนการ เหล่านั้นอีกครั้ง หรืออาจใช้ กระบวนการ (Method) ที่ถูก ผูกติด (Attached) ไว้ใน ฟาเซทของ สล๊อตเป็นตัวดำเนินการเองก็ได้

ในงานวิจัยนี้การวินิจฉัยระบบฐานความรู้นั้นจะมีการวินิจฉัย คือ การใช้ การสืบทอดกับอีกวิธีหนึ่งคือการ ใช้การติดต่อสื่อสารระหว่าง เฟรมผ่านทางการใช้ กระบวนการที่ผูกติดอยู่กับสล๊อต ซึ่งส่วนใหญ่ที่ใช้อยู่คือกระบวนการ IF-NEEDED และ IF-CHANGED หรือก็คือการใช้ การส่งผ่านข้อความ (Message Passing) ระหว่างอินสแตนซ์เฟรม [2, 14]

2.4 ภาษา RDF (Resource Description Framework)

RDF เป็นภาษาในการนำเสนอข้อมูล ที่ถูกออกแบบมาเพื่อนำเสนอข้อมูลเกี่ยวกับทรัพยากรบนเว็บ โดยมีวัตถุประสงค์หลักคือ เพื่อนำเสนอโครงสร้างข้อมูลของทรัพยากรต่างๆ ที่เกี่ยวข้องกับเว็บ เช่น หัวเรื่อง, ผู้แต่ง, วันที่ปรับปรุงแก้ไข, ลิขสิทธิ์ และ การอนุญาตใช้งานเอกสารบนเว็บ หรือแม้แต่ตารางเวลาในการใช้ทรัพยากรร่วมกัน เป็นต้น อย่างไรก็ตาม การจำกัดความคำว่า “ทรัพยากรบนเว็บ” ยังครอบคลุมไปถึง สิ่งต่างๆ ที่สามารถระบุไว้บนเว็บได้ แม้ว่าสิ่งนั้นจะไม่อาจค้นคืนจากเว็บโดยตรงได้ ยกตัวอย่างเช่น ข้อมูลเกี่ยวกับสินค้าบนเว็บซื้อขายสินค้า อันได้แก่ คุณลักษณะ, ราคา และ สถานะ การซื้อขาย หรือ คำอธิบายรูปแบบการนำส่งข้อมูลที่ผู้ใช้งานเว็บเลือกใช้ เป็นต้น

ภาษา RDF ยังเหมาะสมสำหรับนำเสนอข้อมูลในลักษณะที่ข้อมูลเหล่านั้นมีความจำเป็นต้องนำไปประมวลผลโดยโปรแกรมประยุกต์ มากกว่าที่จะนำไปแสดงผลเพียงอย่างเดียว เนื่องด้วยภาษา RDF มีรูปแบบกลางในการชี้แจงข้อมูล จึงทำให้เราสามารถแลกเปลี่ยนข้อมูลระหว่างแอพพลิเคชันต่างๆ ด้วย ภาษา RDF ได้ โดยไม่สูญเสียความหมายของข้อมูลเหล่านั้นไป และเนื่องด้วยภาษา RDF มีรูปแบบกลางในการชี้แจงข้อมูลนี้เอง ผู้ออกแบบแอพพลิเคชันต่างๆ จึงสามารถ

ออกแบบตัวแปลงข้อมูล และส่วนในการประมวลผล ให้สามารถแลกเปลี่ยนข้อมูลระหว่างแอพพลิเคชันที่แตกต่างกันได้ หรือกล่าวคือข้อมูลเหล่านี้สามารถทำให้อยู่ในรูปแบบที่แอพพลิเคชันอื่นๆ นอกเหนือจากแอพพลิเคชันที่สร้างข้อมูลนี้ขึ้น ใช้งานได้ [5]

2.4.1 การนำเสนอข้อมูล ด้วย ภาษา RDF

ภาษา RDF อิงแนวความคิดว่า สิ่งที่เราต้องการอธิบาย ต่างก็ประกอบไปด้วยคุณสมบัติ (Property) ซึ่งมี ค่า (Value) ต่างกันไป ดังนั้น เราจึงสามารถอธิบายทรัพยากร (Resource) ต่างๆ ได้ด้วยการระบุ คุณสมบัติ และ ค่า เหล่านั้นในรูปของประโยค (Statement) เป็นภาษาอังกฤษ ซึ่งถือเป็นภาษากลางในการสื่อสารของมนุษย์ภาษาหนึ่ง โดยภาษา RDF บัญญัติคำศัพท์เฉพาะเกี่ยวกับการอธิบายสิ่งต่างๆ ไว้ดังนี้

ซับเจกต์ (Subject) – คือ ส่วนของประโยค ที่กล่าวถึง ทรัพยากร หรือ สิ่งต่างๆ ที่เราต้องการอธิบาย ด้วยประโยคในภาษา RDF

เพรดิเคต (Predicate) – คือ ส่วนของประโยค ที่กล่าวถึง คุณสมบัติ หรือ คุณลักษณะ ของ ทรัพยากร หรือ สิ่งต่างๆ ที่เราต้องการอธิบายด้วย ประโยคในภาษา RDF

อ็อบเจกต์ (Object) – คือ ส่วนของประโยค ที่ระบุถึง ค่า ของ คุณสมบัติ หรือ คุณลักษณะ ของ ทรัพยากร หรือ สิ่งต่างๆ ที่ต้องการอธิบาย ด้วยประโยคในภาษา RDF

ยกตัวอย่างเช่น

> **http://www.example.org/index.html** has a **creator** whose value is **John Smith**.
>
> ซับเจกต์ ก็คือ URL **http://www.example.org/index.html**
> เพรดิเคต ก็คือ คำว่า **creator**
> อ็อบเจกต์ ก็คือ วลี **John Smith**

รูปที่ 2.13 การอธิบายทรัพยากรในภาษา RDF โดยใช้ ซับเจกต์, เพรดิเคต และ อ็อบเจกต์

ถึงแม้ ภาษาอังกฤษจะเหมาะสมที่จะเป็นสื่อกลางในการติดต่อสื่อสาร แต่ภาษา RDF ยังคงต้องการให้ ประโยคในภาษา RDF อยู่ในรูปแบบที่เหมาะสม สำหรับการประมวลผลด้วยคอมพิวเตอร์ อีกด้วย ทั้งนี้ ภาษา RDF จึงต้องการคุณสมบัติเพิ่มเติมอีก 2 ประการคือ

- รูปแบบตัวบ่งชี้ที่สามารถประมวลผลด้วยคอมพิวเตอร์ได้ สำหรับการบ่งชี้ ซับเจกต์, เพรดิเคต และ อ็อบเจกต์ ในประโยคของภาษา RDF โดยไม่ก่อให้เกิดความสับสนกับตัวบ่งชี้ที่คล้ายกัน ที่อาจมีการใช้งานบนเว็บอื่นๆ

- ภาษาที่คอมพิวเตอร์ประมวลผลได้ สำหรับนำเสนอประโยคภาษา RDF เหล่านี้ และแลกเปลี่ยนข้อมูลระหว่างคอมพิวเตอร์

ทั้งนี้สถาปัตยกรรมของเว็บ ในปัจจุบัน สามารถรองรับความต้องการทั้งสองข้อดังกล่าวได้แล้ว โดยเว็บ จะมีรูปแบบของตัวบ่งชี้ ที่เรียกว่า Uniform Resource Locator (URL) ซึ่งเป็นสายอักขระ (Character string) ที่อธิบายทรัพยากรบนเว็บ ด้วยการแสดงถึงกลไกหลักในการเข้าถึง ทรัพยากรบนเว็บนั้น หรือกล่าวคือ แสดงถึงตำแหน่งที่ตั้งบนเครือข่าย แต่อย่างไรก็ตาม การเก็บข้อมูลเกี่ยวกับสิ่งที่ไม่มี ตำแหน่งที่ตั้งบนเครือข่าย (URL) ก็เป็นเรื่องสำคัญอีกเรื่องหนึ่งเช่นกัน ทั้งนี้ สถาปัตยกรรมของเว็บได้มีการกำหนดตัวบ่งชี้ที่เฉพาะเจาะจงน้อยลงสำหรับกรณีนี้ ที่เรียกว่า Uniform Resource Identifier (URI) ไว้เช่นกัน โดยที่ URL จัดว่าเป็น URI ประเภทหนึ่งนั่นเอง จึงจะเห็นได้ว่า URI เป็นตัวบ่งชี้ที่ถูกสร้างขึ้นเพื่ออ้างถึงสิ่งต่างๆดังนี้

- สิ่งที่สามารถเข้าถึงได้บนเครือข่าย เช่น เอกสารอิเล็กทรอนิกส์, รูปภาพ หรือแม้แต่บริการต่างๆบนเครือข่าย
- สิ่งที่ไม่สามารถเข้าถึงได้บนเครือข่าย เช่น บุคคล, องค์กร และ หนังสือในห้องสมุด
- แนวคิดที่เป็นนามธรรมต่างๆ ที่ยังไม่เป็นรูปธรรม เช่น แนวคิดเกี่ยวกับ “ผู้ผลิต”

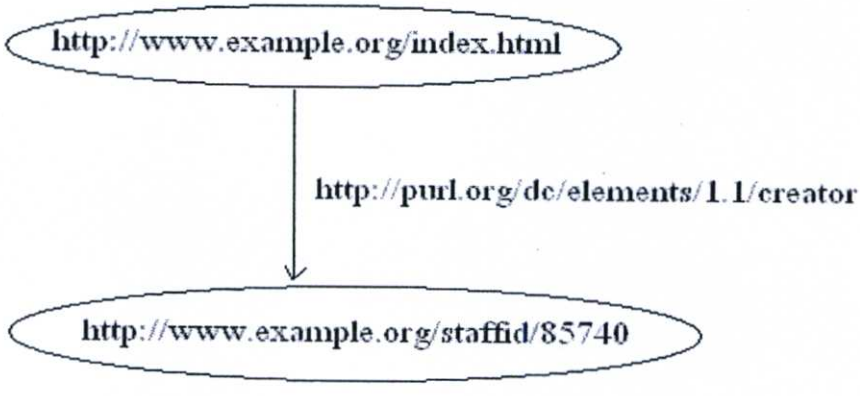
และด้วยคุณสมบัติต่างๆของ URI นี้เอง ภาษา RDF จึงได้นำ URI มาใช้เป็นกลไกพื้นฐานในการกล่าวถึง ซับเจกต์, เพรดิเคต และ อ็อบเจกต์ ใน ประโยคภาษา RDF หรือหากจะอธิบายให้ชัดเจนยิ่งขึ้น อาจกล่าวได้ว่า ภาษา RDF ใช้ URI Reference ในการอธิบายสิ่งต่างๆ โดยที่ URI Reference คือ URI ที่มีตัวบ่งชี้ส่วนย่อย (Fragment Identifier) เพิ่มเติมต่อท้าย URI ตัวอย่างเช่น http://www.example.org/index.html#section2 เป็น URI Reference ที่ประกอบด้วย URI **http://www.example.org/index.html** และ ตัวบ่งชี้ส่วนย่อย **section2** โดยมี เครื่องหมาย # เป็นตัวคั่นระหว่าง URI และ ตัวบ่งชี้ส่วนย่อยนั่นเอง และโดยส่วนมากแล้ว ในภาษา RDF จะนิยม กล่าวถึงสิ่งต่างๆ โดยอ้างถึง URI ทั้งสิ้น

ในส่วนของการนำเสนอข้อมูล ภาษา RDF ใช้ภาษา Extensible Markup Language (XML) เพื่อนำเสนอ ประโยคภาษา RDF ในลักษณะที่คอมพิวเตอร์สามารถประมวลผลได้ เนื่องจาก ภาษา XML ถูกออกแบบมาเพื่อให้ทุกคนสามารถกำหนดรูปแบบเอกสารของตนเอง และบันทึกข้อมูลใน

เอกสารตามรูปแบบนั้นๆ ได้ ภาษา RDF จึงได้กำหนดรูปแบบเอกสารด้วยภาษา XML ที่เรียกว่า RDF/XML ขึ้น เพื่อนำเสนอข้อมูลในภาษา RDF และเพื่อแลกเปลี่ยนข้อมูลดังกล่าวระหว่างคอมพิวเตอร์ได้

2.4.2 การนำเสนอประโยคภาษา RDF ด้วยกราฟ

จากที่ได้กล่าวถึงความสัมพันธ์ระหว่าง ซับเจกต์, เพรดิเคต และ อ็อบเจกต์ ในภาษา RDF ไปแล้ว ภาษา RDF ยังสามารถนำเสนอในรูปแบบกราฟ ได้อีกด้วย โดยในการนำเสนอ ประโยคในภาษา RDF ด้วยกราฟนั้น จะกำหนดให้ใช้ โนด (node) แทน ซับเจกต์ และ อ็อบเจกต์ และให้ใช้อาร์ค (arc) แทน เพรดิเคต โดยจะ ชี้จาก โนดที่เป็น ซับเจกต์ ไปยัง โนดที่เป็น อ็อบเจกต์ ตามตัวอย่างในรูปที่ 2.13 เราสามารถเขียนให้อยู่ในรูปของกราฟได้ดังรูปที่ 2.14 ซึ่งจะเห็นว่า มีการใช้ URI แทน คำว่า creator และ John Smith



รูปที่ 2.14 การนำเสนอประโยคภาษา RDF ด้วยกราฟ

ในบางกรณี ที่มีข้อจำกัดในการวาดกราฟ เราสามารถเขียนกราฟ ให้อยู่ในรูปของ ทริปเพิล (triple) ได้ ด้วยการเขียนเรียงตามลำดับดังนี้ ซับเจกต์ เพรดิเคต อ็อบเจกต์ โดยหากเป็นการเขียน URI จะต้องเขียนอยู่หลังสัญลักษณ์วงเล็บแหลมเปิด “<” และ ใส่สัญลักษณ์วงเล็บแหลมปิด “>” หลัง URI ตัวอย่างเช่น

```
< http://www.example.org/index.html >  < http://purl.org/dc/elements/1.1/creator >  < http://www.example.org/staffid/85740 >
```

รูปที่ 2.15 การเขียนทริปเพิล แทนกราฟ

แต่ในการเขียน URI ซึ่งมีความยาวมาก อาจทำให้เกิดความไม่สะดวก ภาษา RDF จึงกำหนดให้ใช้ คำเติมหน้า (prefix) แทนการเขียน URI ได้ โดยให้ประกาศ คำเติมหน้าที่ต้องการใช้แทน URI จากนั้น จึงสามารถเขียน คำเติมหน้า ตามด้วยเครื่องหมายทวิภาค ":" ได้ โดยไม่ต้องใส่เครื่องหมายวงเล็บแหลม โดยจากตัวอย่างในรูปที่ 2.15 จะสามารถ ใช้คำเติมหน้าได้ ดังนี้

@prefix **ex:** , namespace URI: **http://www.example.org/**

@prefix **dc:** , namespace URI: **http://purl.org/dc/elements/1.1/**

@prefix **exstaff:** , namespace URI: **http://www.example.org/**

ex:index.html	**dc:creator**	**exstaff:85740**

รูปที่ 2.16 การเขียน คำเติมหน้า แทน URI

2.4.3 การนำเสนอข้อมูลด้วยรูปแบบเอกสาร RDF/XML

เพื่อให้คอมพิวเตอร์สามารถประมวลผลประโยคภาษา RDF ได้ ภาษา RDF จึงได้กำหนดรูปแบบเอกสารด้วยภาษา XML ขึ้น โดยมีแท็ก (Tag) พื้นฐาน คือ

1. *<rdf:RDF>* สำหรับประกาศว่า ข้อมูลภายในแท็กนี้ มีรูปแบบเป็น RDF/XML ซึ่งจะมีแอททริบิวต์ (attribute) ที่สำคัญดังต่อไปนี้
   - xmlns:(ชื่อคำเติมหน้า) : สำหรับประกาศคำเติมหน้า
2. *<rdf:Description>* สำหรับแทนโนดและอาร์คต่างๆ โดยจะเขียนอยู่ภายในแท็กที่เป็น ชื่อของโนด หรือ อาร์ค อีกทีหนึ่ง และในการแสดงแทนโนด จะมีแอททริบิวต์ ที่สำคัญดังต่อไปนี้
   - *rdf:about* : สำหรับโนดที่มีข้อมูลเป็น URI และ มีอาร์คชี้ไปยังโนดอื่น ได้แก่ ซับเจกต์โนดทั้งหมด และ อ็อบเจกต์โนดบางโนด
   - *rdf:resource* : สำหรับอ็อบเจกต์โนด ที่มีข้อมูลเป็น URI โดยที่ไม่มีอาร์คชี้ไปยังโนดอื่นๆ

ในการเขียนจะเขียนในลักษณะเป็นลำดับชั้น โดย เขียนแท็กแทนอ็อบเจกต์โนด อยู่ภายในแท็กแทนเพรดิเคตอาร์ค และ แท็กแทนเพรดิเคตอาร์ค อยู่ภายใน แท็กแทนซับเจกต์โนด อีกทีหนึ่ง โดยจากตัวอย่างในรูปที่ 2.15 เราสามารถเขียนให้อยู่ในรูปของเอกสาร RDF/XML ได้ดังรูปที่ 2.17

```
<?xml version="1.0"?>
<rdf:RDF  xmlns: ex="http://www.example.org/"
          xmlns:dc= "http://purl.org/dc/elements/1.1"
          xmlns:exstaff="http://www.example.org/staffid/" >
          <rdf:Description rdf:about= "http://www.example.org/index.html">
                    <dc:creator>
                         <rdf:Description>
                                   <exstaff:85740 rdf:resource=
"http://www.example.org/staffid/85740"/>
                         </rdf:Description>
                    </dc:creator>
          </rdf:Description>
</rdf:RDF>
```

รูปที่ 2.17 เอกสาร RDF/XML

นอกจากนี้ ภาษา RDF ยังสามารถกำหนดชนิดของทรัพยากรต่างๆโดยการกำหนด โครงในภาษา RDF เพื่อความสมบูรณ์ของข้อมูลยิ่งขึ้น ได้อีกด้วย [5]

2.4.4 โครงในภาษา RDF (RDF Schema)

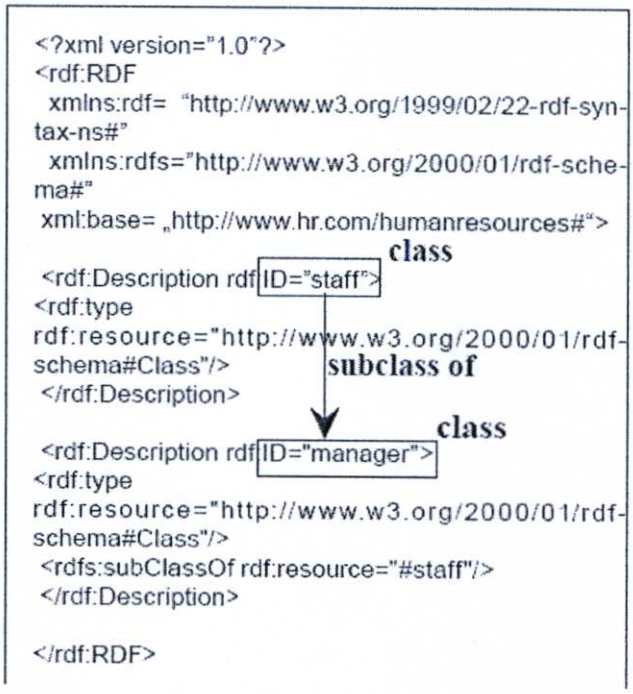
โครงในภาษา RDF เป็นตัวจัดแบ่งชนิดของทรัพยากรต่างๆ สำหรับภาษา RDF โดย โครงในภาษา RDF หรือ RDFs นั้น เป็นเทคนิคที่ใช้ในการสร้าง object model จากข้อมูลจริงที่ถูกอ้างถึงและบอกว่าสิ่งนั้นมีความหมายจริงๆเป็นอย่างไร

ในการกำหนด RDFs นั้นผู้ใช้สามารถทำการกำหนดทรัพยากรด้วย คลาส , คุณสมบัติ และ ค่าได้ เนื่องจากแนวคิดของ RDF class นั้นคล้ายกับแนวคิดของคลาส ใน การเขียนโปรแกรมเชิงวัตถุ เช่น Java และ C++ โดยคลาสนั้นเป็นโครงสร้างของสิ่งที่มีความคล้ายกันและสามารถถ่ายทอดคุณสมบัติได้ ทำให้สามารถกำหนดให้ทรัพยากรต่างๆ เป็น อินสแตนซ์ ของ คลาส และ ซับคลาส ของ คลาสได้ ตัวอย่างเช่น โครงในภาษา RDF นั้น ทำการกำหนดอินสแตนซ์ของ คลาสใดคลาสหนึ่ง หรือ มากกว่านั้นได้ และยิ่งไปกว่านั้นยังสามารถจัดการคลาส ให้เป็นรูปแบบลำดับชั้นได้ ตัวอย่างเช่น คลาส First_Line_Manager อาจถูกกำหนดให้เป็น ซับคลาส ของ Manager ซึ่งคือ ซับ

คลาส ของ Staff ซึ่งจะหมายความว่าทรัพยากรใดๆ ที่เป็นอินสแตนซ์ของ คลาส Staff นั้น จะถือว่าเป็นอินสแตนซ์ของ คลาส First_Line_Manager ด้วยเช่นกัน

RDFs property สามารถมองเป็นแอททริบิวต์ ของ คลาส ได้ ซึ่ง RDFs property นี้ อาจได้รับมาจากการถ่ายทอดจาก RDF property อื่น หรือเป็นตัวกำหนดค่าที่เป็นไปได้ของค่าต่างๆก็ได้ โดยมีวัตถุประสงค์หลักในการนำไปประยุกต์ใช้เพื่อกำหนดคุณสมบัติของคลาสนั้นเอง ตัวอย่างเช่น domain constraint ถูกใช้เพื่อที่กำหนดค่าที่เป็นไปได้ ของคลาส เป็นต้น ซึ่ง RDFs จะถูกใช้ในการประกาศ คำศัพท์ต่างๆนั่นเอง

Box 1.



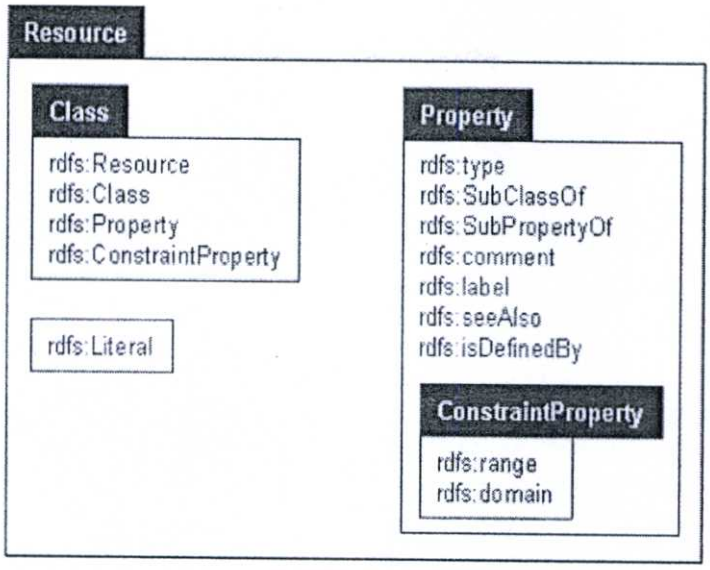

```
<?xml version="1.0"?>
<rdf:RDF
 xmlns:rdf= "http://www.w3.org/1999/02/22-rdf-syn-
tax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-sche-
ma#"
 xml:base= „http://www.hr.com/humanresources#“>
 <rdf:Description rdf:ID="staff">
<rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Class"/>
 </rdf:Description>

 <rdf:Description rdf:ID="manager">
<rdf:type
rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Class"/>
 <rdfs:subClassOf rdf:resource="#staff"/>
 </rdf:Description>

</rdf:RDF>
```

รูปที่ 2.18 การแสดง class และ subclass ในเอกสาร RDF/XML

ใน ประโยคภาษา RDF จาก Box 1 ในรูปที่ 2.18 เป็นตัวอย่างของ RDFs ซึ่งถูก คลาส และลูกหลานนำไปใช้ โดย rdfs:Class ซึ่งมีลักษณะเหมือนคลาส ในการเขียนโปรแกรมเชิงวัตถุ เมื่อมีการกำหนด schema ของคลาสใหม่ คลาสนั้นจะต้องมี rdf:type ซึ่งค่าของมันก็คือทรัพยากรนั่นเอง ทั้งนี้ rdfs:Class ทุกสิ่งที่อธิบายโดย RDF expressions จะถูกเรียกว่า ทรัพยากรและจะถูกพิจารณาเพื่อที่ใช้เป็นอินสแตนซ์ ของ class rdfs:Resource โดยส่วนประกอบอื่นของ RDFs นั้นจะแสดงในรูป 2.19



รูปที่ 2.19 แสดงความสัมพันธ์แนวคิดของ โครงในภาษา RDF

- rdfs:Datatype คือ คลาส ของ data types และกำหนดความสามารถของ data types
- rdfs:Literal คือ คลาส ของค่าสัญพจน์ต่างๆ (literal) เช่น strings และ integers
- rdfs:subClassOf คือ คุณสมบัติการส่งผ่านที่เฉพาะความสัมพันธ์ระหว่าง subset-superset กับ คลาส
- rdfs:subPropertyOf คืออินสแตนซ์ของ rdf : Property ใช้เพื่อที่ระบุว่าคุณสมบัตินั้นเป็นคุณสมบัติ พิเศษของคุณสมบัติอันอื่นหรือไม่
- rdfs:comment คือ คำอธิบายของทรัพยากรที่มนุษย์สามารถอ่านได้
- rdfs:label คือ ชื่อของทรัพยากร แบบที่มนุษย์สามารถอ่านได้และแสดงในรูปแบบ string
- rdfs:seeAlso ระบุว่าทรัพยากรนั้นอาจจะมีข้อมูลเพิ่มเติมเกี่ยวกับ ซับเจกต์
- rdfs:isDefinedBy เป็น sub property ของ rdfs: seeAlso และ ระบุการกำหนดซับเจกต์
- rdfs:member เป็น super-property ของทุกสมาชิกที่อยู่
- rdfs:range แสดงชี้ให้เห็นว่าคลาสนั้นค่าของ property ต้องเป็นสมาชิก
- rdfs:domain ชี้ให้เห็นว่าคลาส ที่เป็นสมาชิกนั้นมีคุณสมบัติอย่างไร
- rdfs:Container คือที่เก็บทรัพยากรต่างๆ
- rdfs:ContainerMemberShipProperty เป็นคลาส ที่ถูกใช้เป็นสถานะของทรัพยากรต่างๆที่เป็น สมาชิกของcontainer

จากที่กล่าวมาทั้งหมดนี้เป็นเพียงพื้นฐานในการนำเสนอข้อมูลในรูปแบบเอกสารRDF/XML เท่านั้น ในการเขียนเอกสาร RDF/XML นั้นแท้จริงแล้วยังสามารถลดทอนการเขียนในบางขั้นตอนได้ ซึ่งวิธีการเหล่านี้สามารถศึกษาได้จากเว็บไซต์ของ W3C (http://www.w3c.org/TR/rdf-primer/)

2.5 การติดต่อสื่อสาร และการสืบค้นข้อมูล ด้วยภาษา SPARQL

ในการติดต่อสื่อสารเพื่อแลกเปลี่ยนข้อมูลในรูปแบบเอกสาร RDF/XML นั้น จำเป็นต้องมีภาษาและโปรโตคอลที่ใช้ในการติดต่อสื่อสารระหว่างระบบ เพื่อเป็นมาตรฐานในการแลกเปลี่ยนข้อมูล ซึ่งภาษาและโปรโตคอลสำหรับการสืบค้นและแลกเปลี่ยนข้อมูลภาษา RDF ที่เป็นที่นิยมและถูกใช้เป็นมาตรฐานในขณะนี้คือ ภาษา SPARQL [6]

2.5.1 การสืบค้นข้อมูลด้วย ภาษา SPARQL

ในขณะนี้ในการสืบค้นข้อมูล RDF เราสามารถใช้ภาษา SPARQL ได้ เนื่องจากภาษา SPARQL ถูกออกแบบมาเพื่อสืบค้นข้อมูลระหว่างแหล่งข้อมูลแหล่งต่างๆ ไม่ว่าจะอยู่ในรูปแบบของ RDF หรือ ถูกจัดให้อยู่ในรูปแบบ RDF ด้วยแอพพลิเคชันอื่นก็ตาม โดยภาษา SPARQL นี้สามารถที่จะสืบค้น RDF กราฟในรูปแบบที่กำหนดได้ และให้ผลลัพธ์ในรูปของ RDF กราฟ หรือเซตของข้อมูลก็ได้ ทั้งนี้การสืบค้นข้อมูลด้วยภาษา SPARQL จะอ้างอิงจาก RDF กราฟ ในลักษณะของ ทริปเพิล ซึ่งเราสามารถประกาศใช้คำนำหน้าได้เช่นกัน และในการสืบค้น จะใช้การกำหนดตัวแปรเพื่อหาผลลัพธ์ ที่ตรงกับรูปแบบ RDF กราฟที่กำหนดได้ ซึ่งจะใช้ข้อมูลในรูปที่ 2.20 ซึ่งเป็นข้อมูลใน ไฟล์ชื่อ TestData.rdf เป็นข้อมูลสำรับอธิบายการสืบค้นในบทนี้

```
@prefix dc: <http://purl.org/dc/elements/1.1/>
@prefix book: <http://example.org/book/>
@prefix writer: <http://example.org/writer/>

book:book1      dc:title     "Harry Potter and the Deadly Hallows"
book:book2      dc:title     "The Lord of the Ring"
writer:ID001    dc:name      "J.K. Rowling"
writer:ID002    dc:name      "J.R.R. Tolkien"
book:book1      dc:auther    writer:ID001
book:book2      dc:auther    writer:ID002
```

รูปที่ 2.20 แสดงข้อมูลสำหรับประกอบการอธิบายการสืบค้นด้วยภาษา SPARQL

2.5.1.1 การสืบค้นด้วยคำสั่ง SELECT การสืบค้นด้วยคำสั่ง SELECT ในภาษา SPARQL จะให้ผลลัพธ์เป็นการผูกค่าตัวแปรที่ตรงกับรูปแบบที่กำหนด ซึ่งจะแบ่งออกเป็น สองส่วนหลัก คือ ส่วนของ SELECT ซึ่งจะเป็นส่วนที่ระบุตัวแปรในการสืบค้นที่ ซึ่งจะเป็นตัวกำหนด ผลลัพธ์ในการสืบค้น รวมถึงแหล่งที่มาของข้อมูล และ ส่วนของ WHERE ซึ่งจะเป็นส่วนที่ระบุรูปแบบของ RDF กราฟ ที่ต้องการสืบค้น โดยในรูปที่ 2.21 จะแสดงถึง ตัวอย่างคำสั่ง SELECT และ รูปที่ 2.22 จะแสดงถึง ผลลัพธ์ที่ได้ในลักษณะค่าตัวแปร

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>

SELECT ?myTitle
FROM <TestData.rdf>

WHERE
{
    ?anyBook        dc:title        ?myTitle
}
```

รูปที่ 2.21 แสดงการสืบค้นด้วยคำสั่ง SELECT ในภาษา SPARQL

myTitle
"Harry Potter and the Deadly Hallows"
"The Lord of the Ring"

รูปที่ 2.22 แสดงผลลัพธ์จากคำสั่ง SELECT ในภาษา SPARQL

จากผลลัพธ์ในรูปที่ 2.21 นั้น เราสามารถจัดให้อยู่ในรูปของเอกสาร XML ได้ด้วย ดังตัวอย่างในรูปที่ 2.23

```
<?xml version= "1.0"?>
<sparql xmlns= "http://www.w3c.org/2005/sparql-results#">
    <head>
        <variable name= "myTitle"/>
    </head>
    <results>
        <result>
            <binding name= "myTitle">
                <literal>Harry Potter and the Deadly Hollows</literal>
            </binding>
        </result>
        <result>
            <binding name= "myTitle">
                <literal>The Lord of the Ring</literal>
            </binding>
        </result>
    </results>
</sparql>
```

รูปที่ 2.23 แสดงผลลัพธ์จากคำสั่ง SELECT ในภาษา SPARQL ในรูปแบบ XML

2.5.1.2 การสืบค้นด้วยคำสั่ง CONSTRUCT การสืบค้นด้วยคำสั่ง CONSTRUCT ในภาษา SPARQL จะให้ผลลัพธ์ เป็น RDF กราฟ ในรูปของ ทริปเพิล ซึ่งเราสามารถจัดให้อยู่ในรูปของ RDF/XML ได้ ซึ่งจะแบ่งออกเป็น สองส่วนหลัก คือ ส่วนของคำสั่ง CONSTRUCT ซึ่งจะเป็นการกำหนด รูปแบบของ กราฟที่เป็นผลลัพธ์ ซึ่งค่าที่จะนำมาสร้างกราฟนี้ จะได้จาก ส่วนที่สอง ที่เป็นส่วนของ คีย์เวิร์ด WHERE ซึ่งจะเป็นการระบุเงื่อนไข เพื่อเลือกข้อมูลที่จะนำมาสร้าง กราฟผลลัพธ์ของการสืบค้นด้วยคำสั่ง CONSTRUCT โดยในรูปที่ 2.24 จะแสดงถึง ตัวอย่างคำสั่ง CONSTRUCT และรูปที่ 2.25 จะแสดงถึง ผลลัพธ์ที่ได้ในรูปของทริปเพิล ซึ่งเราสามารถจัดให้อยู่ในรูปของ เอกสาร RDF/XML ได้ดังรูปที่ 2.26

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>

CONSTRUCT ?myAuther dc:write ?myTitle

WHERE
{
      ?myBookdc:title          ?myTitle
      ?myBookdc:auther ?myAuther
}
```

รูปที่ 2.24 แสดงการสืบค้นด้วยคำสั่ง CONSTRUCT ในภาษา SPARQL

```
@prefix dc: <http://purl.org/dc/elements/1.1/>
@prefix writer: <http://example.org/writer/>

writer:ID001        dc:write    "Harry Potter and the Deadly Hallows"
writer:ID002        dc:write    "The Lord of the Ring"
```

รูปที่ 2.25 แสดงผลลัพธ์จากคำสั่ง CONSTRUCT ในภาษา SPARQL ในรูปของทริปเพิล

```
<rdf:RDF
        xmlns:rdf= "http://www.w3c.org/1999/02/22-rdf-syntax-ns#"
        xmlns:dc= "http://purl.org/dc/elements/1.1/"
        xmlns:writer= "http://example.org/writer/"
>
    <rdf:Description rdf:about= "http://example.org/writer/ID001" >
        <dc:write>Harry Potter and the Deadly Hollows</dc:write>
    </rdf:Description>
    <rdf:Description rdf:about= "http://example.org/writer/ID002" >
        <dc:write>The Lord of the Ring</dc:write>
    </rdf:Description>
</rdf:RDF>
```

รูปที่ 2.26 แสดงผลลัพธ์จากคำสั่ง CONSTRUCT ในภาษา SPARQL ในรูปแบบ RDF/XML

2.5.1.3 การสืบค้นด้วยคำสั่ง ASK การสืบค้นด้วยคำสั่ง ASK ในภาษา SPARQL จะเป็นการตรวจสอบว่ามี RDF กราฟ ในรูปแบบที่กำหนดอยู่หรือไม่ โดยจะให้ผลลัพธ์ เป็น boolean มีค่าเป็น true เมื่อมี RDF กราฟตรงตามรูปแบบที่กำหนด และ มีค่าเป็น false เมื่อไม่มี RDF กราฟตรงตามรูปแบบที่กำหนด โดยรูปที่ 2.27 จะแสดงถึงตัวอย่างคำสั่ง ASK ในภาษา SPARQL

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>

ASK { ?anyBook    dc:title      "The Lord of the Ring" }
```

รูปที่ 2.27 แสดงการสืบค้นด้วยคำสั่ง ASK ในภาษา SPARQL

หากลองพิจารณาจากข้อมูลแล้ว จะพบว่ามี RDF กราฟ ในรูปแบบที่กำหนด ดังนั้น ผลลัพธ์จากการสืบค้นนี้ จะให้ค่าเป็น true ซึ่งแสดงในรูปของเอกสาร XML ดังรูปที่ 2.27

```
<?xml version= "1.0"?>
<sparql xmlns= "http://www.w3c.org/2005/sparql-results#">
        <head></head>
        <results>
            <boolean>true</boolean>
        </results>
</sparql>
```

รูปที่ 2.28 แสดงผลลัพธ์จากคำสั่ง ASK ในภาษา SPARQL ในรูปแบบ XML

2.5.2 การส่งคำสั่ง SPARQL ด้วยเว็บเซอร์วิซ

ภาษา SPARQL ได้ถูกกำหนดโปรโตคอล (Protocol) ที่เป็นมาตรฐานในการรับส่งข้อมูลไว้โดยองค์กร W3C ซึ่งให้คำจำกัดความไว้ในรูปของเว็บเซอร์วิซ ด้วยภาษา WSDL 2.0 (WSDL2) [18] ดังแสดงในรูปที่ 2.28

ซึ่งโปรโตคอลดังกล่าว จะประกอบด้วยอินเตอร์เฟซเดียวคือ SparqlQuery ซึ่งมีเพียงปฏิบัติการเดียวคือ query แต่ทั้งนี้ การที่จะทำให้โปรโตคอลนี้ใช้งานได้ จำเป็นจะต้องทำการผูกโปรโตคอล (Protocol binding) เสียก่อน โดยทำการผูกกับโปรโตคอล HTTP ซึ่งเป็นโปรโตคอลที่

นิยมใช้กันอย่างแพร่หลาย ซึ่งการผูกโปรโตรคอลนี้ ได้มีการอธิบายไว้ด้วยภาษา WSDL2 เช่นกัน ดังตัวอย่างในรูปที่ 2.29

```
<!-- Abstract SparqlQuery Interface -->
<interface name="SparqlQuery"
styleDefault="http://www.w3.org/2006/01/wsdl/style/iri
">

   <!-- the Interface Faults -->
   <fault name="MalformedQuery" element="st:malformed-
query"/>
   <fault name="QueryRequestRefused"
element="st:query-request-refused"/>

   <!-- the Interface Operation -->
   <operation name="query"
pattern="http://www.w3.org/2006/01/wsdl/in-out">

      <documentation>The operation is used to convey
queries and their results from clients to services and
back
      again.</documentation>

      <input messageLabel="In" element="st:query-
request"/>
      <output messageLabel="Out" element="st:query-
result"/>

      <!-- the interface faults are out faults -->
      <outfault ref="tns:MalformedQuery"
messageLabel="Out"/>
      <outfault ref="tns:QueryRequestRefused"
messageLabel="Out"/>
   </operation>

</interface>
```

รูปที่ 2.29 อินเตอร์เฟซ SparqlQuery

```
<!-- the HTTP GET binding for query operation -->
 <binding name="queryHttpGet"
interface="tns:SparqlQuery"
       type="http://www.w3.org/2006/01/wsdl/http"
       whttp:version="1.1">

   <fault ref="tns:MalformedQuery" whttp:code="400"/>
   <fault ref="tns:QueryRequestRefused"
whttp:code="500"/>

   <operation ref="tns:query"
        wsdlx:safe="true"
         whttp:method="GET"
         whttp:faultSerialization="*/*"
         whttp:inputSerialization="application/x-www-
form-urlencoded"

   whttp:outputSerialization="application/sparql-
results+xml, application/rdf+xml, */*" />
 </binding>

 <!-- the HTTP POST binding for query operation -->
 <binding name="queryHttpPost"
interface="tns:SparqlQuery"
          type="http://www.w3.org/2006/01/wsdl/http"
          whttp:version="1.1">

     <fault ref="tns:MalformedQuery"
whttp:code="400"/>
     <fault ref="tns:QueryRequestRefused"
whttp:code="500"/>

   <operation ref="tns:query"
        wsdlx:safe="true"
         whttp:method="POST"
         whttp:faultSerialization="*/*"
         whttp:inputSerialization="application/x-www-
form-urlencoded, application/xml"

   whttp:outputSerialization="application/sparql-
results+xml, application/rdf+xml, */*" />
 </binding>
```

รูปที่ 2.30 การผูกโปรโตคอลกับ HTTP

บทที่ 3
สถาปัตยกรรมของระบบต้นแบบ

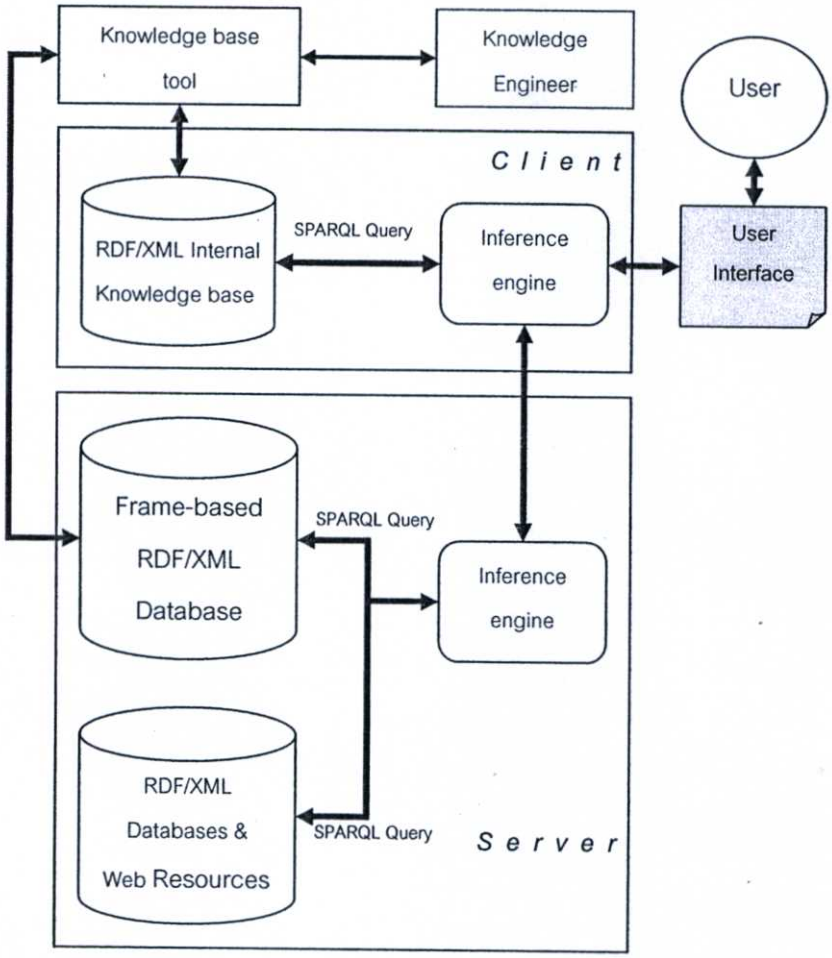
ในบทนี้นำเสนอสถาปัตยกรรมแบบใหม่ในการพัฒนาต้นแบบระบบผู้เชี่ยวชาญ ซึ่งมีฐานความรู้เป็นฐานข้อมูล RDF/XML ที่จัดเก็บโครงสร้างของเฟรม ซึ่งเป็นการแทนความรู้ของระบบไว้ โดยจะกล่าวถึงส่วนนี้ว่า เซิร์ฟเวอร์ และมีเปลือกระบบผู้เชี่ยวชาญ ที่เพิ่มฐานความรู้ภายในเข้าไป โดยจะกล่าวถึงส่วนนี้ว่า ไคลเอนท์ ทั้งนี้ สถาปัตยกรรมของต้นแบบระบบผู้เชี่ยวชาญที่นำเสนอ สามารถทำการวินิจฉัยได้ทั้งที่ฝั่งเซิร์ฟเวอร์ และฝั่งไคลเอนท์ เพื่อลดปัญหาในการโหลดข้อมูลที่เกี่ยวข้องทั้งหมดจากฝั่งเซิร์ฟเวอร์ มายังไคลเอนท์เพื่อทำการวินิจฉัย แต่จะเป็นการถามหาคำตอบจากฝั่งเซิร์ฟเวอร์ได้เลย ซึ่งหากจำเป็นต้องทำการวินิจฉัยเพิ่มเพื่อหาคำตอบ ระบบก็จะทำการวินิจฉัยที่ฝั่งเซิร์ฟเวอร์ แล้วจึงส่งข้อมูลที่ได้รับการวินิจฉัยระดับหนึ่งแล้วให้กับไคลเอนท์อีกที

วิทยานิพนธ์ฉบับนี้ ใช้โปรแกรม Protégé ในการสร้างฐานความรู้ในรูปแบบเอกสาร RDF/XML โดยมีการสร้างเฟรมต้นแบบขึ้น เพื่อเป็นกรณีศึกษา ทั้งนี้อาจจะมีการเรียกใช้ข้อมูลจากแหล่งข้อมูลอื่นๆ ที่ถูกเก็บอยู่ในรูปแบบเอกสาร RDF/XML ด้วยเช่นกัน ซึ่งสามารถทำได้โดยการอ้างถึงค่าในสล๊อตของเฟรมไปยังแหล่งข้อมูลซึ่งเก็บค่านั้นๆ ไว้

3.1 สถาปัตยกรรมของระบบต้นแบบที่นำเสนอ

จากรูปที่ 3.1 แสดงสถาปัตยกรรมของระบบต้นแบบ ซึ่งมีคุณสมบัติที่สำคัญคือ ระบบสามารถที่จะทำการวินิจฉัยข้อมูลได้ทั้งจากฝั่งเซิร์ฟเวอร์ และฝั่งไคลเอนท์ ทั้งนี้การแทนความรู้ของทั้งสองส่วนจะใช้การแทนความรู้แบบเฟรม ซึ่งจะถูกเก็บอยู่ในฐานข้อมูล RDF/XML โดยในส่วนไคลเอนท์ นั้นจะเป็นส่วนที่สื่อสารโต้ตอบกันระหว่างระบบผู้เชี่ยวชาญกับผู้ใช้ เพื่อรวมรวมข้อเท็จจริงที่ได้จากการที่ผู้ใช้ป้อนเข้ามายังระบบ ซึ่งการวินิจฉัยในส่วนนี้สามารถจะอ้างไปถึงข้อเท็จจริงที่อยู่บนฐานความรู้ ที่เป็นส่วนเซิร์ฟเวอร์ได้ และเมื่อส่วนเซิร์ฟเวอร์ทำการวินิจฉัยเรียบร้อย ก็จะสามารถส่งผลลัพธ์สุดท้ายกลับไปยังฝั่งไคลเอนท์ ได้

ในส่วนไคลเอนท์นี้ จะประกอบไปด้วย ส่วนติดต่อกับผู้ใช้ ฐานความรู้ภายใน และ กลไกการวินิจฉัยบนฝั่งไคลเอนท์ ในส่วนของเซิร์ฟเวอร์นั้นจะประกอบไปด้วย ฐานความรู้ที่ใช้การแทนความรู้แบบเฟรมในฐานข้อมูล RDF/XML กลไกการวินิจฉัยบนฝั่งเซิร์ฟเวอร์ และฐานข้อมูลที่เกี่ยวข้องอื่นๆ (ถ้ามี)



รูปที่ 3.1 แสดงสถาปัตยกรรมของระบบ

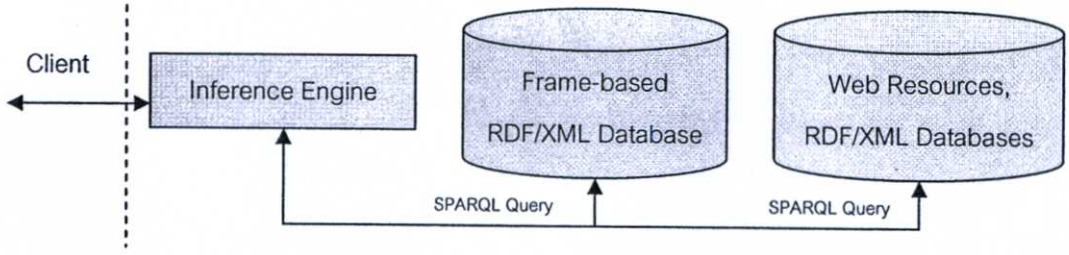
ในระหว่างกระบวนการตอบคำถามนั้น ผู้ใช้จะติดต่อกับระบบผู้เชี่ยวชาญด้วยการถามตอบกับไคลเอนท์ ซึ่งใช้การวินิจฉัยแบบเฟรม โดยมีจุดประสงค์หลักของการวินิจฉัยที่ส่วนนี้คือการรวบรวมข้อเท็จจริงปัจจุบันในหัวข้อนั้นๆ ซึ่งกฎการวินิจฉัยในเฟรมแรกจะเลือกเฟรมที่เหมาะสมที่จะกระทำต่อไป ซึ่งอาจมีหลายเฟรมที่ส่วนไคลเอนท์นี้ เมื่อส่วนนี้ทำการรวบรวมข้อมูลได้เพียงพอแล้ว จะส่งข้อมูลไปยังกลไกการวินิจฉัยบนฝั่งเซิร์ฟเวอร์เพื่อทำการวินิจฉัยต่อไป

การวินิจฉัยบนฝั่งเซิร์ฟเวอร์ จะเน้นไปที่การใช้ข้อเท็จจริงที่มีอยู่แล้วซึ่งได้ถูกจัดเก็บเป็นเฟรมไว้ในฐานความรู้และอาจมีการเรียกใช้ข้อเท็จจริง จากแหล่งข้อมูลอื่นๆเพิ่มเติมด้วย โดยคำตอบที่ได้จากการวินิจฉัยบนฝั่งนี้จะถูกส่งกลับไปยังระบบผู้เชี่ยวชาญที่ฝั่งไคลเอนท์ และจากนั้นจึงส่งคำตอบไปยังผู้ใช้

ในการติดต่อสื่อสารระหว่างไคลเอนท์ กับเซิร์ฟเวอร์ นั้น วิทยานิพนธ์ฉบับนี้ใช้การติดต่อกันในรูปแบบของการเรียกใช้เว็บเซอร์วิซ โดยจะเป็นการส่งค่าการสืบค้นและจัดการข้อมูลด้วยภาษา SPARQL ซึ่งเป็นภาษาที่ใช้สำหรับการสืบค้นและจัดการฐานข้อมูล RDF/XML

3.1.1 ฝั่งเซิร์ฟเวอร์

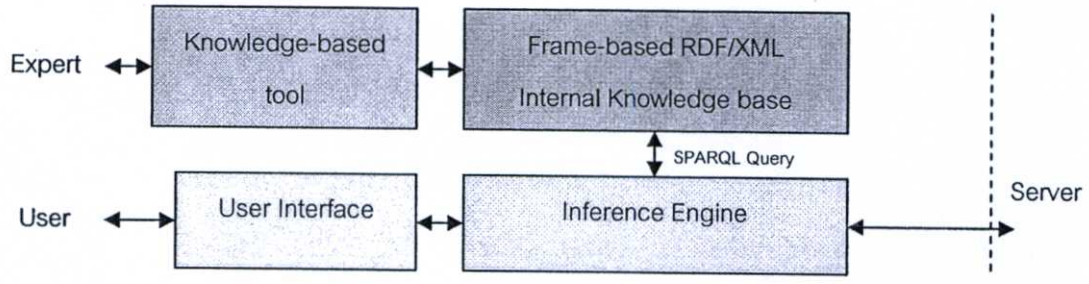
ในส่วนสถาปัตยกรรมของระบบที่ฝั่งเซิร์ฟเวอร์ ในรูปที่ 3.2 นั้น วิทยานิพนธ์ฉบับนี้ได้เพิ่มส่วนกลไกการวินิจฉัย ที่ฝั่ง เซิร์ฟเวอร์ซึ่งเป็นฐานความรู้ด้วย โดยที่ฝั่งเซิร์ฟเวอร์จะรับข้อมูลจากฝั่งไคลเอนท์ ที่ถูกส่งมาด้วยภาษา SPARQL จากนั้นกลไกการวินิจฉัยที่ส่วนเซิร์ฟเวอร์นี้จะเป็นส่วนดำเนินการวินิจฉัยเพื่อหาคำตอบส่งกลับไปให้ส่วนไคลเอนท์ ทั้งนี้ ที่ฝั่งเซิร์ฟเวอร์ อาจมีการสืบค้นข้อมูลจากแหล่งข้อมูลอื่นเช่น ทรัพยากรบนเว็บต่างๆ หรือฐานข้อมูล RDF/XML อื่นๆที่เกี่ยวข้อง เป็นต้น



รูปที่ 3.2 แสดงสถาปัตยกรรมของระบบฝั่งเซิร์ฟเวอร์

3.1.2 ฝั่งไคลเอนท์

ที่ฝั่งไคลเอนท์นี้จะทำการรับข้อเท็จจริงจากผู้ใช้เป็นหลักและสามารถทำการวินิจฉัยข้อเท็จจริงที่ฝั่งไคลเอนท์ ได้เลยถ้าข้อมูลที่ได้รับจากผู้ใช้มีเพียงพอ โดยที่การทำงานของฝั่งไคลเอนท์นั้นผู้ใช้จะทำการสอบถามถึงคำตอบที่ต้องการ จากระบบ จากนั้นระบบจะทำการสอบถามข้อเท็จจริงที่เกี่ยวข้องจากผู้ใช้ซึ่งถ้าข้อเท็จจริงจากผู้ใช้มีไม่เพียงพอ หรือข้อเท็จจริงที่ต้องใช้ประกอบการวินิจฉัยอยู่ที่ฝั่งเซิร์ฟเวอร์ ระบบก็จะทำการสอบถามไปยังเซิร์ฟเวอร์ โดยส่งไปในรูปแบบของภาษา SPARQL แล้วจากนั้นจึงรอรับคำตอบที่ได้จากเซิร์ฟเวอร์ เมื่อทางฝั่งไคลเอนท์ได้รับคำตอบที่ต้องการแล้ว ก็จะทำการวินิจฉัยต่อไปหรือส่งผลลัพธ์ที่ได้กลับไปยังผู้ใช้ได้ดังรูปที่ 3.3



รูปที่ 3.3 แสดงสถาปัตยกรรมของระบบฝั่งไคลเอนท์

3.2 การติดต่อสื่อสารระหว่างไคลเอนท์กับเซิร์ฟเวอร์

ในการวินิจฉัยของระบบ จะกระทำโดยการเรียกเฟรมขึ้นมาทีละเฟรม และเชื่อมต่อเฟรมไปเรื่อยๆ เพื่อหาผลลัพธ์ซึ่งอาจต้องมีการส่งข้อมูลไปวินิจฉัยที่ฝั่งเซิร์ฟเวอร์ หรืออาจต้องมีการสืบค้นข้อมูลจากแหล่งข้อมูลอื่นอีกที ซึ่งในการส่งข้อมูลไปวินิจฉัยที่ฝั่งเซิร์ฟเวอร์นั้น วิทยานิพนธ์ฉบับนี้ใช้วิธีการเรียกใช้เว็บเซอร์วิซในการส่งคำสั่งภาษา SPARQL ในการติดต่อสื่อสารระหว่างไคลเอนท์และ เซิร์ฟเวอร์ โดยฝั่งไคลเอนท์จะทำการเรียกใช้เว็บเซอร์วิซ พร้อมทั้งส่งค่าพารามิเตอร์ (Parameter) เป็นคำสั่งภาษา SPARQL ไปยังเซิร์ฟเวอร์ เพื่อให้เซิร์ฟเวอร์นำข้อเท็จจริงไปวินิจฉัยเพื่อหาคำตอบ และส่งกลับมาให้ไคลเอนท์ และถ้าหากมีข้อเท็จจริงที่จำเป็นอยู่บนแหล่งข้อมูลอื่นๆ ก็อาจมีการการสืบค้นข้อมูลด้วยภาษา SPARQL เพื่อดึงเฉพาะข้อมูลที่เฟรมนั้นต้องการออกมา แล้วทำการวินิจฉัยต่อไปได้

ทั้งนี้ ข้อมูลจากแหล่งข้อมูลอื่นจะถูกใช้เป็นข้อมูลภายนอกของระบบ ซึ่งกลไกการวินิจฉัยจะถูกกระทำบนเฟรม และจะอ้างไปยังเฟรมอื่นๆจนกว่าข้อเท็จจริงจะถูกค้นพบในฐานความรู้ ซึ่งกระบวนการจะถูกกระทำเพื่อที่จะอ้างถึงข้อเท็จจริงที่อยู่ในแหล่งข้อมูลภายนอกโดยไม่จำเป็นต้องคัดลอกข้อมูลอย่างถาวรมาไว้ในเฟรมแต่จะเป็นการเรียกดูข้อมูลเมื่อจำเป็นต้องใช้เท่านั้น ดังนั้นจึงสามารถหลีกเลี่ยงปัญหาความซ้ำซ้อนของข้อมูล เมื่อมีการอัพเดตข้อมูลต้นทางได้และจากการที่ระบบผู้เชี่ยวชาญของเราใช้ภาษา SPARQL ในการสืบค้นข้อมูล ทำให้เราสามารถสืบค้นข้อมูลจากฐานข้อมูล RDF/XML หรือทรัพยากรบนเว็บต่างๆ ได้โดยง่าย เนื่องจากเป็นความสามารถของภาษา SPARQL อยู่แล้ว

ยกตัวอย่างเช่น การวิเคราะห์เลือกโปรโมชันสำหรับโทรศัพท์เคลื่อนที่ที่เหมาะสม หากต้องการทราบว่า มีโปรโมชันใดที่รองรับความต้องการของผู้ใช้บริการโทรศัพท์เคลื่อนที่ บ้าง ระบบก็จะทำการสร้าง ตัวอย่างของโปโรโมชันสำหรับโทรศัพท์เคลื่อนที่ เป็นอินซแตนซ์เฟรม (Instance Frame) ขึ้นมา แล้วมีข้อมูลให้ผู้ใช้ กรอกข้อมูลเบื้องต้นของผู้ใช้บริการโทรศัพท์เคลื่อนที่ ซึ่งอาจเป็นลักษณะการใช้งานโทรศัพท์เคลื่อนที่ของผู้ใช้บริการ หรืออาจเป็นข้อมูลเกี่ยวกับบริการเสริมที่ต้องการอื่นๆก็ได้ จากนั้น ระบบจะทำการเลือกเฟรมของโปรโมชันที่ใกล้เคียงขึ้นมา ซึ่งค่าของสล๊อต ในเฟรม นั้นๆ จะอ้างไปยังเฟรม อื่นๆ หรือ แหล่งข้อมูล อื่นๆ อีกที และจะทำการ ใส่ค่าของสล๊อต ของอินสแตนซ์เฟรม นั้น เพิ่มเข้าไปยัง ฐานความรู้จนกว่าจะมีข้อมูลครบถ้วน จนได้โปรโมชันที่มีความใกล้เคียงกับความต้องการของ ผู้ใช้บริการโทรศัพท์เคลื่อนที่ มากที่สุด จึงจะส่งค่ากลับไปยังระบบผู้เชี่ยวชาญฝั่งไคลเอนท์อีกที เพื่อใช้ในการวิเคราะห์ข้อมูลเพิ่มเติมต่อไป

บทที่ 4
รายละเอียดเกี่ยวกับการพัฒนาระบบต้นแบบ

การพัฒนาต้นแบบระบบผู้เชี่ยวชาญ ได้แบ่งการทำงานออกเป็น 3 ส่วน ได้แก่ การสร้างฐานความรู้บนฐานข้อมูล RDF/XML การพัฒนากลไกการวินิจฉัยบนฝั่งเซิร์ฟเวอร์และไคลเอนท์ และ การติดต่อสื่อสารแลกเปลี่ยนข้อมูล RDF ด้วยภาษา SPARQL

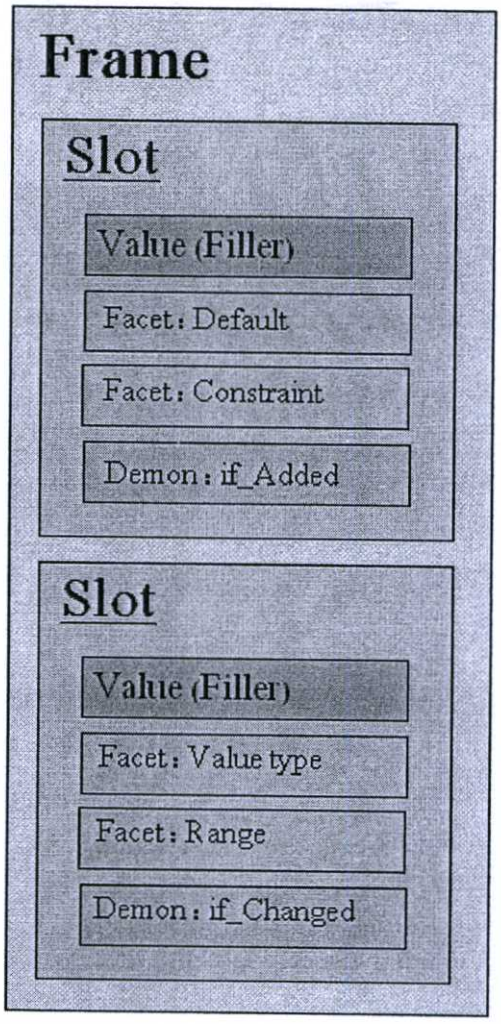
4.1 การเขียนโครงสร้างของเฟรม บนฐานข้อมูล RDF/XML

เนื่องจากวิทยานิพนธ์ฉบับนี้ นำเสนอรูปแบบการแทนความรู้แบบเฟรม โดยใช้ฐานข้อมูล RDF/XML ดังนั้นในวิทยานิพนธ์ฉบับนี้จึงได้ทำการศึกษาโครงสร้างข้อมูลของเฟรม เพื่อที่จะได้สร้างเฟรมต้นแบบ หรือ รูธเฟรม (Root Frame) ขึ้น บนฐานข้อมูล RDF/XML ได้

4.1.1 โครงสร้างข้อมูลประเภท เฟรม

เฟรม เป็นโครงสร้างที่เกิดมาจากแนวคิดในการออกแบบโดยยึดหลักของการออกแบบตามวัตถุ หรือหลักการ (Concept) โดยเฟรมจะเป็นสิ่งที่รวบรวมเอาตัวบ่งบอกคุณสมบัติของวัตถุเอาไว้เป็นเพียงเสมือนกรอบที่ถูกวาดไว้เพื่อรักษาการรวมกลุ่มกันของตัวบ่งบอกคุณสมบัติเท่านั้น แนวคิดนี้จึงถูกเรียกว่าเฟรม ในขณะที่ตัวบ่งบอกคุณสมบัติของวัตถุทั้งหลายจะถูกจัดให้เป็นเสมือนวัตถุหน่วยย่อยๆ ที่อยู่ภายในตัวเอง โดยจะมีคุณสมบัติต่างๆ รวมถึงเมธอด (Method) โพรซิเยอร์ (Procedure) และดีมอนที่เกี่ยวข้องกับตัวเองไว้ด้วยกัน หรือกล่าวอีกนัยหนึ่งคือ เมธอด โพรซิเยอร์ หรือดีมอนต่างๆที่มีความเกี่ยวข้องกับตัวบ่งบอกคุณสมบัติใด จะถูกรวบรวมไว้กับตัวบ่งบอกคุณสมบัตินั้นเอง ทำให้หน่วยย่อยที่ทำหน้าที่เป็นตัวบ่งบอกคุณสมบัติของวัตถุนั้นถูกเรียกว่าสล๊อต ดังแสดงให้เห็นในรูปที่ 4.1

จากข้อมูลเบื้องต้นด้านโครงสร้างของเฟรม ทำให้พบว่า แต่ละเฟรม จะประกอบไปด้วยองค์ประกอบหลักที่สำคัญ คือ สล๊อต ซึ่งสล๊อตเหล่านี้ จะมีค่าของสล๊อต ที่เรียกว่า ฟิลเลอร์ และอาจมี ตัวกำหนดค่าในสล๊อต ที่เรียกว่า ฟาเซท ได้หลายประเภท ซึ่งจะมีลักษณะแตกต่างกันไป และนอกจากนี้ เฟรม จะประกอบไปด้วยเฟรมแม่ และ เฟรมลูก ยกเว้น รูธเฟรม ที่จะมีเพียงเฟรมลูกเท่านั้น และ ลีฟเฟรม (Leaf Frame) จะมีเพียงเฟรมแม่เท่านั้น

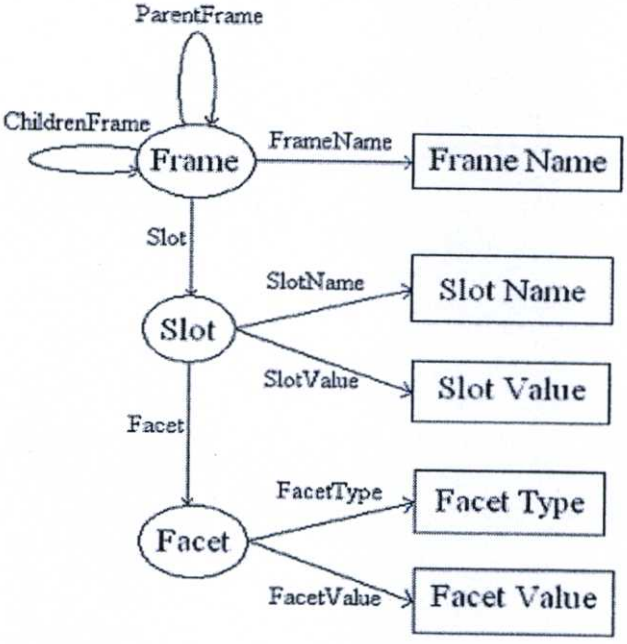


รูปที่ 4.1 แผนภาพตัวอย่างโครงสร้างข้อมูลแบบเฟรม

4.1.2 การจัดเก็บเฟรมในฐานข้อมูล RDF/XML

ในการที่จะจัดเก็บโครงสร้างแบบเฟรม ในฐานข้อมูล RDF/XML นั้น ควรทำการศึกษาโครงสร้างของเฟรม ในรูปของ RDF กราฟ เพื่อแสดงถึงคุณสมบัติต่างๆของ เฟรมต้นแบบก่อน จากนั้นจึงทำการออกแบบ RDF กราฟ เพื่อนำไปแปลงเป็น เอกสาร RDF/XML ที่สามารถเก็บข้อมูลของเฟรมได้ ซึ่งจากโครงสร้างของเฟรม ที่ได้กล่าวไปนั้น พบว่า เฟรม ประกอบไปด้วยโครงสร้างหลักๆ 3 ส่วน คือ เฟรม สล๊อต และ ฟาเซท รวมถึง ดีมอน ซึ่งจัดเป็น ฟาเซทประเภทหนึ่งด้วยเช่นกัน

วิทยานิพนธ์ฉบับนี้จึงได้นำเสนอการออกแบบ RDF กราฟ แสดงโครงสร้างของฐานข้อมูล RDF/XML เพื่อใช้เก็บการแทนความรู้แบบเฟรม ดังแสดงในรูปที่ 4.2



รูปที่ 4.2 RDF กราฟ แสดงโครงสร้างฐานข้อมูล RDF/XML เพื่อใช้จัดเก็บเฟรม

จะเห็นว่า โครงสร้างของฐานข้อมูล RDF/XML ที่นำเสนอ ประกอบไปด้วย ซับเจกต์ทั้งหมดสามซับเจกต์ คือ เฟรม สล็อต และ ฟาเซท โดยในแต่ละซับเจกต์ จะมีเพรดิเคต และ อ็อบเจกต์ ตามความสัมพันธ์ที่แสดงใน รูปที่ 4.2 ซึ่งจาก RDF กราฟ นี้ สามารถนำไปเขียนให้อยู่รูปของฐานข้อมูล RDF/XML ได้ ดังแสดงในภาคผนวก ก. โดยในรูปที่ 4.3 แสดงตัวอย่างของเอกสาร RDF/XML ที่ได้

```
<?xml version="1.0"?>
<rdf:RDF    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
    xmlns:owl="http://www.w3.org/2002/07/owl#">
...
   <owl:Ontology rdf:about="Frame"/>
   <owl:Class rdf:ID="Frames">
      <rdfs:subClassOf><owl:Restriction>
            <owl:onProperty rdf:resource="#FrameName"/>
         </owl:Restriction></rdfs:subClassOf>
   </owl:Class>
...
</rdf:RDF>
```

รูปที่ 4.3 ตัวอย่างเฟรมต้นแบบ ในฐานข้อมูล RDF/XML

4.2 กลไกการวินิจฉัยของระบบ

ในระบบผู้เชี่ยวชาญแบบเฟรมนั้นมีการวินิจฉัยได้หลายรูปแบบ เช่น การวินิจฉัยโดยการสืบทอด (Inference by Inheritance) เป็นการตอบคำถามด้วยการหาเฟรม ไม่ว่าจะเป็นคลาสเฟรม หรืออินสแตนซ์เฟรม ที่มีสล๊อตและค่าของสล๊อตที่ต้องการ ซึ่งบางครั้งการสืบทอดสามารถถูก นิยามเป็นกระบวนการ ได้ [4] ตัวอย่างการวินิจฉัยโดยการสืบทอด เช่น Does Shania Twain have a spleen? จากคำถามนี้ถ้าหากพิจารณาไปยัง คลาสเฟรม ของ อินสแตนซ์เฟรม Shania Twain จะพบว่าเป็นสมาชิกของคลาสเฟรม woman จะสรุปได้ว่า Shania Twain is woman และเมื่อพิจารณาขึ้นไปอีกจะพบว่า 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 ว่าคู่ใดตรงกันมากที่สุด ตัวอย่างเช่น What kind of singer is Shania Twain? คำถามนี้ระบบจะทำการจับคู่ อินสแตนซ์เฟรมของ Shania Twain เข้ากับ คลาสเฟรมต่างๆเหล่านี้ คือ opera-singer, country-singer, rock-singer, และ pop-singer ซึ่งถ้าคุณสมบัติของอินสแตนซ์เฟรม Shania Twain ตรงกันกับ คลาสเฟรม pop-singer มากที่สุด ก็จะสามารถสรุปได้ว่า Shania Twain ถูกจำแนกให้เป็น pop singer

ระบบผู้เชี่ยวชาญที่ใช้วิธีการแทนความรู้แบบเฟรมส่วนใหญ่นั้น มักจะใช้เทคนิคการจำแนกประเภท และการสืบทอด ในการวินิจฉัย และแก้ปัญหาบนพื้นฐานของความรู้ที่จัดเก็บไว้ในฐานความรู้ ดังนั้นในวิทยานิพนธ์ฉบับนี้จึงใช้กลไกการวินิจฉัยสองแบบคือ การจำแนกโดยการจับคู่และการวินิจฉัยโดยการสืบทอด

4.2.1 ขั้นตอนการวินิจฉัยของระบบ

ระบบต้นแบบที่พัฒนาขึ้นจะใช้กระบวนการหาเฟรมเป้าหมายด้วยการจับคู่ โดยระบบจะให้ผู้ใช้กรอกข้อมูลเข้ามาแล้วทำการจับคู่ข้อมูลที่ได้รับกับเฟรม เพื่อระบุว่าเป็นอินสแตนซ์ของคลาสเฟรมใด โดยจะมีขั้นตอนดังนี้

ฝั่งไคลเอนต์

1. ระบบจะทำการเปรียบเทียบค่าที่ป้อนเข้ามา กับเฟรมตั้งต้นที่มีอยู่
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. จับคู่สล๊อตในเฟรมนั้นกับข้อมูลที่ได้มา
5. ถ้ามีการร้องขอข้อมูลเพิ่มเติมจากในฐานข้อมูลภายนอก ให้ไปดึงข้อมูลเข้ามา
6. ถ้ามีการร้องขอให้ผู้ใช้ป้อนข้อมูลเพิ่มเติม ส่งคำถามไปยังฝั่งไคลเอนต์เพื่อให้ผู้ใช้ตอบคำถาม แล้วส่งคำตอบมายังฝั่งเซิร์ฟเวอร์
7. ถ้าเฟรมเป้าหมายสอดคล้องกับข้อมูลทั้งหมด ก็ส่งเฟรมนั้นเป็นคำตอบกลับไปยังผู้ใช้
8. ถ้าข้อมูลไม่สอดคล้องกับเฟรมเป้าหมาย ให้เลือกเฟรมเป้าหมายใหม่จากลิสต์ goal_list แล้วทำการจับคู่เฟรมอีกครั้งจนกว่าจะได้คำตอบ หรือจนกว่าไม่มีเฟรมเหลืออยู่ใน goal_list อีกแล้ว

จากขั้นตอนการวินิจฉัยของระบบข้างต้นเราสามารถเขียนอัลกอริธึมเพื่อทำการวินิจฉัยโดยรวมของระบบได้ดังนี้

```
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
```

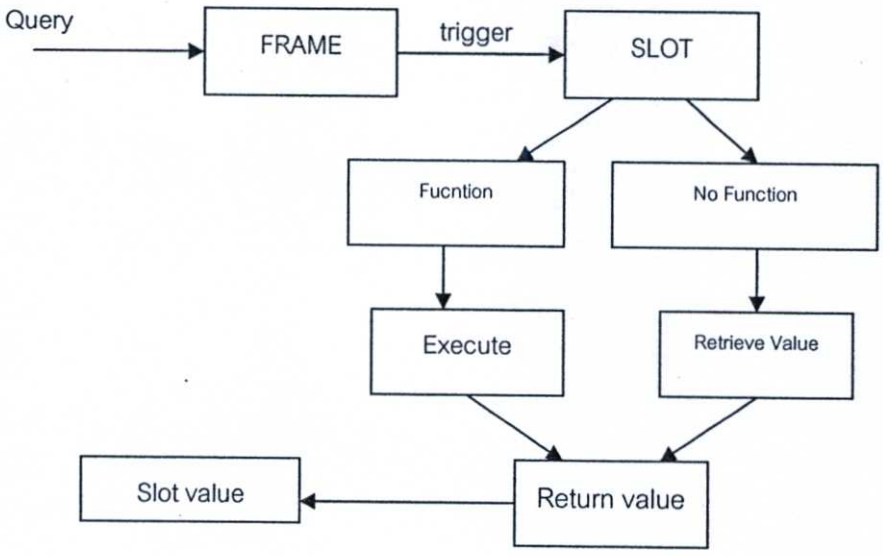
รูปที่ 4.4 อัลกอริธึมในการหาเฟรมเป้าหมายด้วยการจับคู่

จากขั้นตอนการวินิจฉัยข้างต้นกลไกการวินิจฉัยนี้จะกระทำก็ต่อเมื่อได้มีการส่งคำถามมาจากฝั่งไคลเอนท์เพื่อให้เซิร์ฟเวอร์ทำการวินิจฉัยต่อโดยที่ข้อมูลทางฝั่งไคล์เอนต์จะทำการวินิจฉัยไประยะหนึ่งก่อนเพื่อให้ผู้ใช้ได้ตอบคำถาม ซึ่งเป็นการวินิจฉัยแบบฟอร์เวิร์ด (Forward Reasoning) ซึ่งเมื่อได้ข้อมูลมาพอสมควร ระบบจะทำการตั้งเป้าหมายแล้วค้นหาว่าข้อมูลที่ได้มาตรงกับเป้าหมายใดมากที่สุด ซึ่งจะเข้าสู่กระบวนการวินิจฉัยเพื่อจับคู่เฟรมที่สอดคล้องที่สุด ซึ่งเป็นการวินิจฉัยแบบแบคเวิร์ด (Backward Reasoning) ดังนั้นในวิทยานิพนธ์ฉบับนี้จะใช้การวินิจฉัยแบบผสม คือมีการวินิจฉัยแบบฟอร์เวิร์ดไประยะหนึ่งก่อน จากนั้นจึงทำการวินิจฉัยแบบแบคเวิร์ดต่อไปจนพบเป้าหมายแล้วส่งผลลัพธ์หรือคำตอบที่ได้กลับไปยังผู้ใช้

4.2.2 การพัฒนากลไกการวินิจฉัยที่ฝั่งฐานความรู้

วิทยานิพนธ์ฉบับนี้จะใช้ภาษา ASP.NET C# เป็นเครื่องมือในการพัฒนากลไกการวินิจฉัยที่ฝั่งฐานความรู้ที่อยู่ในฐานข้อมูล RDF/XML ซึ่งสร้างขึ้นโดยใช้ โปรแกรม Protégé ทั้งนี้ เทคนิคในการวินิจฉัยนั้นจะมีอยู่ด้วยกันสองวิธี หนึ่งคือ รับเฟรมจากฝั่งไคลเอนท์เข้ามาเก็บไว้ในเฟรมบนฝั่งเซิร์ฟเวอร์ซึ่งกระบวนการจะถูกเรียกขึ้นมาจากการเปลี่ยนแปลงค่าในสล๊อต และสองคือ เก็บข้อมูลที่ได้จากเฟรมบนฝั่งไคลเอนท์มาเป็น พารามิเตอร์ของฟังก์ชันที่เก็บไว้ ซึ่งจะทำการวินิจฉัยโดยใช้ฟังก์ชัน ต่อๆไป ซึ่งวิธีที่สองนี้จะเหมาะสำหรับข้อมูลที่มาจากฝั่งไคลเอนท์มีจำนวนไม่มาก โดยตัวที่จะใช้เรียก ฟังก์ชัน จะอยู่ในส่วนของโปรแกรมซึ่งค่าที่ได้จากการประมวลผลจะถูก ส่งกลับไปยังไคลเอนท์ต่อไป

ในส่วนของกระบวนการต่างๆ จะมีการทำงานดังนี้คือ เมื่อรับคำสั่งจากผู้ใช้ หรือระบบผู้เชี่ยวชาญฝั่งไคลเอนท์ กระบวนการต่างๆ จะทำการค้นหาค่าของสล๊อตภายในเฟรม และเมื่อได้เฟรมที่ต้องการก็จะไปตรวจสอบที่สล๊อตว่าได้มีฟังก์ชัน สำหรับเหตุการณ์นั้นๆหรือไม่ จากนั้นจึงรับค่าที่ได้จากกระบวนการในส่วนนี้ ไปยังใส่ไว้ในฟิลเลอร์ซึ่งเป็นค่าของสล๊อตในอินสแตนซ์เฟรมอีกที โดยที่การใส่ค่าในสล๊อต อาจมีกระบวนการที่กระทำภายหลังจากที่ใส่ค่าแล้วต่อไปก็ได้ ในรูปที่ 4.5 เป็นการแสดงการทำงานของกระบวนการหาค่าสล๊อตภายในเฟรม โดยอันดับแรกจะมีการป้อนค่าหรือเรียกดูค่าในส่วนของฟิลเลอร์ของอินสแตนซ์เฟรม ระบบจะดูว่ามีกระบวนการที่ต้องกระทำหลังป้อนค่าหรือก่อนเรียกดูค่าหรือไม่ ถ้ามีก็จะไปเรียกใช้ฟังก์ชัน โดยอ้างอิงจากกระบวนการต่างๆขึ้นมา ซึ่งอาจมีการร้องขอให้มีการป้อนข้อมูลที่สล๊อตอื่นได้ ซึ่งในการเรียกดูค่าจะมีอัลกอริธึมดังแสดงในรูปที่ 4.6 โดยจะเป็นการหาค่าสล๊อตจากเฟรมปัจจุบันไล่ขึ้นไปยังเฟรมแม่ตามลำดับขั้นและจะมีการเรียกฟังก์ชันที่มีอัลกอริธึมตามในรูปที่ 4.7 และ 4.8 ตามลำดับ



รูปที่ 4.5 แสดงการทำงานของกระบวนการหาค่าของสล๊อตภายในเฟรม

```
Function:       get_all_slot_value
Input:          frame_id, slot_name
Output:         answer
1.  answer:=null;
2.  parent:=null;
3.  answer = call get_slot_value;
4.  if answer = null
5.      parent = call find_parent;
6.      while parent is not null AND answer = null
7.          parent = call find_parent;
8.      end loop
9.  return answer;
```

รูปที่ 4.6 อัลกอริธึมในการเรียกดูค่าของสล๊อต

```
Function:       find_parent
Input:          frame, slot_name
Output:         parent
1.  begin
2.      parent := null;
3.      SELECT parent INTO parent FROM  parent_tab
4.      if  no_data_found then
5.          parent := null;
6.      return parent
7.  end;
```

รูปที่ 4.7 อัลกอริธึมสำหรับการหาเฟรมแม่

```
Function :  get_slot_value
Input:      frame, slot_name
Output:     answer
1. answer:= null;
2. begin
3.          SELECT id INTO v_slot_id FROM   slots_tab
4.          if v_slot_id is not null then
5.          begin
6.                SELECT  'value' INTO ret
7.                when no_data_found
8.                      SELECT 'default' INTO ret
9.                      when no_data_found
10.                           SELECT if_needed INTO ret
11.                           when no_data_found
12.                                 answer = NULL;
13.  end if;
14.  answer ret;
15.end;
```

รูปที่ 4.8 อัลกอริธึมสำหรับการดึงค่าจากสล๊อต

4.3 การติดต่อสื่อสารและแลกเปลี่ยนข้อมูล RDF ด้วยภาษา SPARQL

วิทยานิพนธ์ฉบับนี้ ใช้ภาษา SPARQL ในการสืบค้นข้อมูลจากฐานข้อมูล RDF/XML ซึ่งการส่งค่าคำสั่งภาษา SPARQL ระหว่างเซิร์ฟเวอร์ กับไคลเอนท์ ระบบจะใช้วิธีการเรียกใช้เว็บเซอร์วิซ ซึ่งถูกกำหนดมาตรฐานโดยองค์กร W3C โดยมีการกำหนดโปรโตคอล SPARQL ขึ้น และนำโปรโตคอล SPARQL ไปผูกกับ โปรโตคอล HTTP อีกที โดยในรูปที่ 4.9 จะแสดงตัวอย่างการส่งคำสั่ง SPARQL ผ่านโปรโตคอล HTTP โดยที่ EncodedQuery คือคำสั่งภาษา SPARQL และผลลัพธ์ของการสืบค้นก็จะถูกส่งกลับมาในรูปของเอกสาร XML ตัวอย่างเช่นในรูปที่ 4.10

```
GET /sparql/?query=EncodedQuery HTTP/1.1
Host: www.example
User-agent: my-sparql-client/0.1
```

รูปที่ 4.9 การส่งคำสั่งภาษา SPARQL ผ่านโปรโตคอล HTTP

```
HTTP/1.1 200 OK
Date: Fri, 06 May 2005 20:55:12 GMT
Server: Apache/1.3.29 (Unix) PHP/4.3.4 DAV/1.0.3
Connection: close
Content-Type: application/sparql-results+xml

<?xml version="1.0"?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <variable name="book"/>
    <variable name="who"/>
  </head>
  <results distinct="false" ordered="false">
    <result>
      <binding name="book"><uri>http://www.example/book/book5</uri></binding>
      <binding name="who"><bnode>r29392923r2922</bnode></binding>
    </result>
    ...
    <result>
      <binding name="book"><uri>http://www.example/book/book6</uri></binding>
      <binding name="who"><bnode>r8484882r49593</bnode></binding>
    </result>
  </results>
</sparql>
```

รูปที่ 4.10 HTTP Trace ส่วนผลลัพธ์จากการสืบค้น

วิทยานิพนธ์ฉบับนี้ พัฒนาต้นแบบระบบผู้เชี่ยวชาญขึ้น โดยเลือกใช้การแทนความรู้แบบเฟรมในการนำเสนอความรู้ และใช้ฐานข้อมูล RDF/XML ในการจัดเก็บความรู้ของระบบ ซึ่งกลไกการวินิจฉัยโดยใช้เฟรม จะกระทำโดยการเชื่อมต่อเฟรม ด้วยค่าในสล๊อต และกระบวนการที่ผูกติดไว้ โดยจะทำการเชื่อมต่อเฟรมไปเรื่อยๆ จนกว่าจะได้คำตอบที่ต้องการ นอกจากนี้ โครงสร้างของระบบต้นแบบที่พัฒนาขึ้น จะมีส่วนกลไกการวินิจฉัยทั้งในส่วนของเซิร์ฟเวอร์และในส่วนของไคลเอนท์ ซึ่งกลไกการวินิจฉัยทั้งสองส่วนนี้ จะติดต่อสื่อสารกัน โดยการส่งคำสั่งภาษา SPARQL ซึ่งเป็นภาษาในการสืบค้น และจัดการข้อมูล RDF ในรูปของการใช้งานเว็บเซอร์วิซ

บทที่ 5
การพัฒนาต้นแบบระบบผู้เชี่ยวชาญ เพื่อวิเคราะห์เลือกโปรโมชันสำหรับโทรศัพท์เคลื่อนที่

การวิเคราะห์เลือกโปรโมชันสำหรับโทรศัพท์เคลื่อนที่ เป็นกระบวนการที่ต้องอาศัยผู้เชี่ยวชาญคือนักสถิติ ซึ่งในการวิเคราะห์ลักษณะการใช้งานโทรศัพท์เคลื่อนที่ และความต้องการของผู้ใช้บริการโทรศัพท์เคลื่อนที่ในเบื้องต้นนั้น เพื่อทำไปสรุปหรือวิเคราะห์ความเหมาะสมว่าโปรโมชันสำหรับโทรศัพท์เคลื่อนที่ใด มีความเหมาะสม และ สอดคล้องกับความต้องการของผู้ใช้บริการโทรศัพท์เคลื่อนที่มากที่สุด และหากพิจารณาจากลักษณะการใช้งานโทรศัพท์เคลื่อนที่ของผู้ใช้บริการที่ผ่านมาแล้ว ค่าใช้บริการโทรศัพท์เคลื่อนที่ของผู้ใช้บริการจะมีแนวโน้มเป็นอย่างไร เพื่อเป็นข้อมูลประกอบการตัดสินใจเลือกใช้โปรโมชันสำหรับโทรศัพท์เคลื่อนที่ดังกล่าวให้แก่ผู้ใช้บริการโทรศัพท์เคลื่อนที่ โดยในบทนี้จะกล่าวถึงวิธีการวินิจฉัยโดยสังเขป รวมถึงการสร้างระบบ

ในงานวิจัยนี้เราได้กำหนดผู้ใช้ เป็นบุคคลากรที่ได้รับการอบรมมารยาท และทักษะในการสอบถาม และ ให้ข้อมูลแก่ผู้ใช้บริการโทรศัพท์เคลื่อนที่เกี่ยวกับโปรโมชันมาแล้ว ซึ่งอาจเป็นพนักงานคอลเซ็นเตอร์ หรือเจ้าหน้าที่ฝ่ายต่างๆที่เกี่ยวข้อง เนื่องจากข้อมูลเกี่ยวกับโปรโมชันสำหรับโทรศัพท์เคลื่อนที่ เป็นข้อมูลจำนวนมาก ไม่ว่าจะเป็นโปรโมชันสำหรับโทรศัพท์เคลื่อนที่ระบบรายเดือน โปรโมชันสำหรับโทรศัพท์เคลื่อนที่ระบบเติมเงิน หรือแม้แต่ โปรโมชันสำหรับโทรศัพท์เคลื่อนแบบองค์กรก็ตาม ซึ่งข้อมูลเหล่านี้จะมีการเพิ่มเติม ปรับเปลี่ยน และยกเลิก อย่างสม่ำเสมอ ส่งผลให้ ระบบคอลเซ็นเตอร์ในปัจจุบัน ต้องมีการจัดอบรม ให้ความรู้ รวมไปถึงประชาสัมพันธ์ความเปลี่ยนแปลงเกี่ยวกับข้อมูลเหล่านี้ให้กับพนักงานคอลเซ็นเตอร์และ เจ้าหน้าที่ที่เกี่ยวข้อง เป็นประจำ แต่อย่างไรก็ตาม พนักงานคอลเซ็นเตอร์ส่วนมาก ก็ยังขาดทักษะในด้านการวิเคราะห์เกี่ยวกับโปรโมชัน ด้วยเหตุนี้จึงทำให้ การแจ้งข้อมูลต่างๆ ให้กับลูกค้า หรือในที่นี้คือผู้ใช้บริการโทรศัพท์เคลื่อนที่ โดยเฉพาะอย่างยิ่ง การให้คำแนะนำเกี่ยวกับโปรโมชัน มีความผิดพลาดเกิดขึ้นอยู่เสมอ และนอกเหนือจากปัญหาในด้าน ทักษะการวิเคราะห์เกี่ยวกับโปรโมชันแล้ว ยังมีปัญหาในด้านของสิทธิ์ในการเข้าถึงข้อมูล ที่พนักงานคอลเซ็นเตอร์บางระดับ อาจไม่สามารถเรียกดูข้อมูลประวัติการใช้งานโทรศัพท์เคลื่อนที่ของลูกค้าได้อีกด้วย

ดังนั้นระบบผู้เชี่ยวชาญนี้ จึงมีประโยชน์อย่างมากในการช่วยวิเคราะห์เลือกโปรโมชันสำหรับโทรศัพท์เคลื่อนที่ของลูกค้า เพื่อเป็นข้อมูลให้ผู้ใช้ ในที่นี้คือ พนักงานคอลเซ็นเตอร์ สามารถ

แนะนำลูกค้าได้อย่างถูกต้องและเหมาะสม มากที่สุด โดยไม่จำเป็นต้องทราบรายละเอียดการใช้งานโทรศัพท์เคลื่อนที่ของลูกค้า ส่วนที่เป็นข้อมูลส่วนบุคคล

ในการวิเคราะห์เลือกโปรโมชัน สิ่งที่ต้องการทราบคือ ลักษณะการใช้งานของลูกค้าและ ความต้องการของลูกค้าเกี่ยวกับการใช้บริการโทรศัพท์เคลื่อนที่ เพื่อให้เห็นภาพมากขึ้น จึงขอยกตัวอย่างการวิเคราะห์เลือกโปรโมชันสำหรับโทรศัพท์เคลื่อนที่ระบบรายเดือนเป็นหลัก

5.1 ขั้นตอนการทำงานของระบบ

ระบบผู้เชี่ยวชาญนี้จะทำการวิเคราะห์เบื้องต้นจาก ความต้องการของลูกค้าที่จำเป็น โดยฝั่งไคลเอนท์จะทำการวินิจฉัย เพื่อพิจารณาข้อมูลที่เกี่ยวข้อง ว่าจะต้องถามอะไรเพิ่มเติมอีกบ้าง แล้วส่งข้อมูลไปวินิจฉัยบนฝั่งเซิร์ฟเวอร์จากนั้นจึงส่งผลการวินิจฉัยกลับมาว่าเป็นมีโปรโมชันใดที่มีความเหมาะสมบ้าง ดังนั้นบนฝั่งเซิร์ฟเวอร์จะเป็นเฟรมที่เกี่ยวกับโปรโมชันต่างๆ พร้อมทั้งลักษณะการใช้งานโทรศัพท์เคลื่อนที่ ที่เหมาะสมกับโปรโมชันนั้นๆ ส่วนฝั่งไคลเอนท์จะเป็นเฟรมที่เกี่ยวข้องกับการถามคำถาม และการเลือกคำถามออกมาถามลูกค้า โดยจะสรุปเป็นกระบวนการได้จากรูปที่ 5.1 ซึ่งจะแสดงขั้นตอนการทำงานของระบบดังที่กล่าวไว้ดังนี้

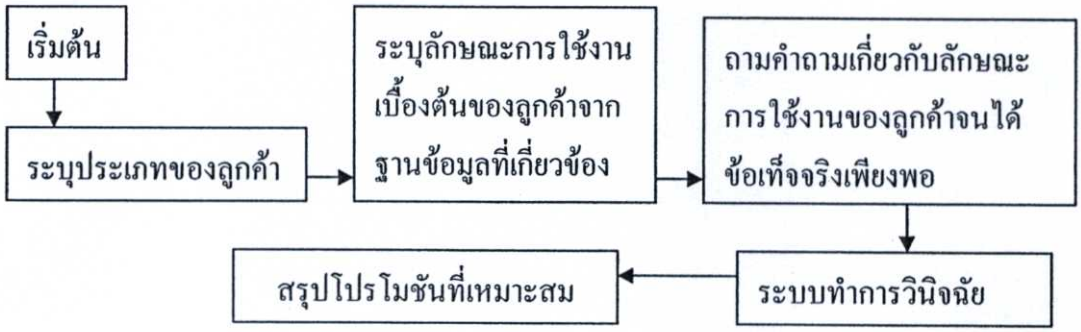
1) ระบุประเภทของลูกค้า ว่าเป็น ลูกค้าปัจจุบัน หรือเป็นลูกค้าใหม่

2) ระบบจะทำการเรียกดูข้อมูลลักษณะการใช้งานของลูกค้าจากฐานข้อมูลที่เกี่ยวข้องโดยอัตโนมัติ แต่หากเป็นลูกค้าใหม่ ระบบจะให้ตอบคำถาม เพื่อระบุลักษณะการใช้งานเบื้องต้น เช่น โทรบ่อย โทรนาน หรือ เน้นการใช้งานบริการเสริม เป็นต้น

3) ระบบจะถามความคำถามเกี่ยวกับลักษณะการใช้งานเบื้องต้นของลูกค้า โดยให้พนักงานสอบถาม แล้วป้อนข้อมูลเข้าไป จนกว่าจะได้ข้อเท็จจริงเพียงพอต่อการวินิจฉัย

4) ระบบส่งข้อมูลที่ได้ไปยังเซิร์ฟเวอร์เพื่อทำการวินิจฉัย

5) สรุปว่ามีโปรโมชันใดที่เหมาะสมบ้าง และแสดงรายละเอียดของโปรโมชัน



รูปที่ 5.1 แสดงขั้นตอนการทำงานของระบบวิเคราะห์เลือกโปรโมชัน

5.2 การสร้างเฟรมเพื่อใช้ แทนความรู้ในระบบต้นแบบ

ในการสร้างเฟรมนั้น เราจะแบ่งออกเป็น 2 ฝั่ง คือ ฝั่งเซิร์ฟเวอร์และฝั่งไคลเอนท์โดยในฝั่งเซิร์ฟเวอร์จะเป็นเฟรมเกี่ยวกับโปรโมชันสำหรับโทรศัพท์เคลื่อนที่ทั้งหมด โดยในที่นี้คือเฟรมเกี่ยวกับโปรโมชันสำหรับโทรศัพท์เคลื่อนที่ระบบรายเดือนทั้งหมด ส่วนในฝั่งไคลเอนท์จะเป็นเฟรมที่เกี่ยวข้องกับการถามลักษณะการใช้งาน และความต้องการของลูกค้า โดยจะเรียงลำดับตามเหมาะสม เช่น ถ้าลูกค้าเป็นลูกค้าใหม่ ต้องการหาโปรโมชันที่เหมาะสม ระบบก็จะถามความต้องการของลูกค้าก่อน เพราะเป็นลูกค้าใหม่ จึงยังไม่มีประวัติการใช้งาน แล้วจึงอาจมีการสอบถามถึง ลักษณะการใช้งานของลูกค้า หากระบบจำเป็นต้องทราบ เป็นต้น ตารางที่ 5.1 จะแสดงเฟรมเกี่ยวกับโปรโมชันสำหรับโทรศัพท์เคลื่อนที่ระบบรายเดือน และลักษณะการใช้งานที่เหมาะสม และ ตารางที่ 5.2 จะแสดงเฟรม คำถามเกี่ยวกับความต้องการ และ ลักษณะการใช้งานของลูกค้า

ตารางที่ 5.1 แสดงเฟรมที่เกี่ยวกับโปรโมชันระบบรายเดือน และลักษณะการใช้งานที่เหมาะสม

No.	frame	slot	slot value	facet	facet value
P1		Parent	-		
		Children	Pos Regular, Pos Vary, Pos Time, Pos Number		
	Postpaid Promotion	Cust. Type	Postpaid	constraint	{Postpaid, Prepaid}
				If_changed	If slot value = Prepaid then go to "Prepaid Pro"

ตารางที่ 5.1 (ต่อ)

P2	Pos Regular	Parent	Postpaid Promotion		
		Children	Feel Free 20, Feel Free 40, Feel Free60, Feel Free 90		
		Call Behavior	Regular	Constraint	{regular, vary, time-relate, num-relate }
P3	Pos Vary	Parent	Postpaid Promotion		
		Children	Vary Duration, Good Short Chat, Lite 100		
		Call Behavior	Vary	Constraint	{regular, vary, time-relate, num-relate }
P4	Pos Time	Parent	Postpaid Promotion		
		Children	Good One 12 Hr., Sode		
		Call Behavior	Time-relate	Constraint	{regular, vary, time-relate, num-relate }
P5	Pos Number	Parent	Postpaid Promotion		
		Children	Number No Limit, Number Limit		
		Call Behavior	Num-relate	Constraint	{regular, vary, time-relate, num-relate }
P6	Feel Free 20	Parent	Pos Regular		
		Call Frequecy	Rare	Constraint	{rare, normal, frequent, very frequent}
P7	Feel Free 40	Parent	Pos Regular		
		Call Frequecy	Normal	constraint	{rare, normal, frequent, very frequent}
P8	Feel Free 60	Parent	Pos Regular		
		Call Frequecy	Frequent	constraint	{rare, normal, frequent, very frequent}

ตารางที่ 5.1 (ต่อ)

P9	Feel Free 90	Parent	Pos Regular		
		Call Frequecy	Very frequent	constraint	{rare, normal, frequent, very frequent}
P10	Lite 100	Parent	Pos Vary		
		Call Duration	Long	constraint	{short, long, vary}
P11	Good Short Chat	Parent	Pos Vary		
		Call Duration	Short	constraint	{short, long, vary}
P12	Vary Duration	Parent	Pos Vary		
		Children	Half 500, Half 700, Half 900, Half 1200, Half 1800, Half 2400, Half 4500		
		Call Duration	Vary	constraint	{short, long, vary}
P13	Half 500	Parent	Vary Duration		
		Estimated Cost	250-349		
P14	Half 700	Parent	Vary Duration		
		Estimated Cost	350-449		
P15	Half 900	Parent	Vary Duration		
		Estimated Cost	450-599		
P16	Half 1200	Parent	Vary Duration		
		Estimated Cost	600-899		
P17	Half 1800	Parent	Vary Duration		
		Estimated Cost	900-1199		
P18	Half 2400	Parent	Vary Duration		
		Estimated Cost	1200-1799		
P19	Half 4800	Parent	Vary Duration		
		Estimated Cost	1800+		
P20	Good One 12 Hr.	Parent	Pos Time		
		Time Preferred	Day	Constraint	{Day, Night, Both}
P21	Sode	Parent	Pos Time		
		Time Preferred	{Night, Both}*	Constraint	{Day, Night, Both}

ตารางที่ 5.1 (ต่อ)

P22	Number No Limit	Parent	Pos Number		
		Children	Good Friend, Good 3 Friends		
		Limited Time	Unlimited		
P23	Number Limited	Parent	Pos Number		
		Children	Lite Love Love, Four Mo		
		Limited Time	60		
P24	Good Friend	Parent	Number No Limit		
		Special No.	1		
P25	Good 3 Friends	Parent	Number No Limit		
		Special No.	>1		
P26	Lite Love Love	Parent	Number Limited		
		Special No.	1		
P24	Good Friend	Parent	Number Limited		
		Special No.	>1		

ตารางที่ 5.2 เฟรมคำถามเกี่ยวกับ ลักษณะการใช้งานของลูกค้า

No	frame	slot	slot value	facet	facet value
C1	Cust. Profile	Mobile No.		Default	New Customer
		Cust. Type		Constraint	{Postpaid, Prepaid}
				default	Postpaid
				If_changed	If slot value = Prepaid then go to "Prepaid Question" else go to "Postpaid Question"

ตารางที่ 5.2 (ต่อ)

C2	Postpaid Question	Call Behavior		Constraint	{regular, vary, time-relate, num-relate}
				If_changed	If slot value = regular then go to "Pos Regular Question"
				If_changed	If slot value = vary then go to "Pos Vary Question"
				If_changed	If slot value = time-relate then go to "Pos Time Question"
				If_changed	If slot value = num-relate then go to "Pos Num Question"
		Promotion Calc		If_changed	Go to "Calc"
C3	Pos Regular Question	Call Frequency		Constraint	{rare, normal, frequent, very frequent}
C4	Pos Vary Question	Call Duration		Constraint	{short, long, vary}
				If_changed	If slot value = vary then go to "Pos Duration Vary Question"
C5	Pos Duration Vary Question	Estimated Cost			
C6	Pos Time Question	Time Preferred		Constraint	{Day, Night, Both}

ตารางที่ 5.2 (ต่อ)

C7	Pos Num Question	No. of Special Number			
		Limited time			
C8	Calc	Promotion Name			
		Use > 3 month	yes	default	Yes
				If_changed	If slot value = no then go to "Estimated Usage"
C9	Estimated Usage	Avg. each call duration			
		Avg. usage / day			
		Call behavior		Constraint	{regular, vary, time-relate, num-relate }

จากเฟรมใน ตาราง 5.1 เราจะทำการจัดเก็บไว้ที่ฝั่งเซิร์ฟเวอร์โดยที่จะรับค่าความต้องการของลูกค้า และ ลักษณะการใช้งานของลูกค้า มาจากฝั่งไคลเอนท์และทำการวินิจฉัยพร้อมสรุปโปรโมชันที่เหมาะสมส่งกลับไปให้ โดยในฝั่งไคลเอนท์ก็จะมีเฟรมที่ใช้ถามความต้องการต่างๆ และลักษณะการใช้งาน ของลูกค้า ดังนี้

จากตารางที่ 5.2 ผู้ใช้ สามารถเลือกกรอกเฉพาะความต้องการที่ลูกค้าแจ้ง และลักษณะการใช้งานที่ทราบได้ จากนั้นเมื่อได้ข้อมูลความต้องการและลักษณะการใช้งานของลูกค้าแล้ว ระบบจะสร้างเฟรมขึ้นมาเพื่อแสดงความต้องการ และลักษณะการใช้งานของลูกค้า พร้อมทั้งหมายเลขโทรศัพท์เคลื่อนที่ ที่อ้างอิงถึงลูกค้า ในกรณีเป็นลูกค้าเดิม และเมื่อกรอกข้อมูลแล้ว จึงส่งข้อมูลไปยังฝั่งเซิร์ฟเวอร์เพื่อประมวลผลต่อไป

5.3 การวินิจฉัยหาคำตอบของต้นแบบระบบผู้เชี่ยวชาญที่พัฒนาขึ้น

ในการวินิจฉัยเพื่อหาคำตอบ ของระบบต้นแบบที่พัฒนาขึ้น จะมีขั้นตอน ดังต่อไปนี้

1) เมื่อทราบข้อมูลเบื้องต้น ลักษณะการใช้งานและความต้องการของลูกค้าแล้ว ระบบจะดูว่าลูกค้ามีความเหมาะสมกับโปรโมชันกลุ่มใดบ้าง แล้วเรียกคำถามที่ต้องถามเพิ่มเติมเกี่ยวกับลักษณะการใช้งานที่เกี่ยวข้อง

2) เมื่อผู้ใช้กรอกข้อมูลที่เกี่ยวข้องแล้ว จะทำการส่งค่าทั้งหมดไปยัง ฝั่งเซิร์ฟเวอร์เพื่อทำการวินิจฉัยแยกเฉพาะกลุ่ม

3) เมื่อฝั่งเซิร์ฟเวอร์รับเฟรมที่บอกลักษณะการใช้งาน และความต้องการอื่นๆของลูกค้า ก็จะทำการเปรียบเทียบเพื่อหาโปรโมชันที่มีลักษณะสอดคล้อง โดยไล่หาที่เฟรมของโปรโมชันต่างๆ แล้วสร้าง อินสแตนซ์ ขึ้นมาเพื่อเปรียบเทียบลักษณะความต้องการและลักษณะการโทรของลูกค้ากับ ลักษณะการใช้งานที่เหมาะสมกับโปรโมชัน จนได้โปรโมชันนั้นๆ ออกมา ซึ่งลักษณะการใช้งานที่เหมาะสมโดยรวมต่างๆ จะ สืบทอด มาจากคลาสเฟรม

4) ในกรณี ที่ผู้ใช้ อาจกรอกข้อมูลไม่ครบถ้วน เนื่องจากต้องอ้างอิงถึงประวัติการใช้งานที่ผ่านมา เช่น ประวัติการใช้งานโทรศัพท์เคลื่อนที่ย้อนหลัง 3 เดือน หรือประวัติการสมัครบริการเสริม ระบบจะทำการดึงข้อมูลเพิ่มเติม จากแหล่งข้อมูลอื่นๆ อย่างเช่น ฐานข้อมูลลูกค้า เพื่อทำการวินิจฉัยต่อไป

5) เมื่อฝั่งเซิร์ฟเวอร์ทำการวินิจฉัย และสรุปผลการวิเคราะห์เลือกแล้ว จะส่งผลลัพธ์ ของโปรโมชันที่เหมาะสม กลับไปให้ผู้ใช้ ที่อยู่บนฝั่งไคลเอนท์ เพื่อแจ้งข้อมูลแก่ลูกค้าต่อไปได้

ในกรณีตัวอย่างนี้ สมมุติลูกค้า เป็นลูกค้าใหม่ สนใจใช้งานโทรศัพท์เคลื่อนที่ระบบรายเดือน โดยเมื่อสอบถามลักษณะการใช้งานจากลูกค้าแล้ว ลูกค้าแจ้งว่าลูกค้ามีนิสัยการโทรไม่แน่นอน บางวันโทรเยอะ บางวันโทรน้อย และในการโทรแต่ละครั้ง คุยนาน และไม่ต้องการใช้งานบริการเสริมอื่นๆ ทางฝั่งไคลเอนท์จะได้ข้อสรุปคร่าวๆ ว่า ลูกค้าคนนี้ เหมาะกับโปรโมชันระบบรายเดือนแบบไม่เน้นบริการเสริม แล้วจะทำการสร้างเฟรม ลูกค้า ซึ่งจะมี สล๊อต ดังนี้ หมายเลขอ้างอิง, หมายเลขโทรศัพท์เคลื่อนที่, ชื่อลูกค้า, พฤติกรรมการใช้งาน, ลักษณะระยะเวลาการใช้งาน ดังรูปที่ 5.2 ดังนี้

frame	slot	slot value
Customer	Mobile Number	
	Customer Type	Postpaid
	Call Behavior	Vary
	Call Duration	Long

รูปที่ 5.2 เฟรมลูกค้าที่ฝั่งไคลเอนท์

ขั้นตอนการไป สืบค้นค่าเพื่อเปรียบเทียบนั้น จะเริ่ม เปรียบเทียบ จาก คลาสแม่ ก่อน และไล่ไปตามอนุกรมวิธานไปจนถึง โปรโมชันที่ระบุ โดยเราจะใช้ ฟังก์ชันที่เก็บไว้ใน ตาราง ฟาเซท เป็นตัว ชี้นำในการค้นหาซึ่งเราจะรู้ คลาสแม่ที่จะเริ่มค้นหาได้จาก สล๊อตพื้นฐานที่อยู่ในเฟรมลูกค้าที่ได้มาจากทางฝั่งไคลเอนท์นั่นเอง จากนั้น ในส่วนของสล๊อต Call Behavior และ Call Duration จะเป็นลักษณะการใช้งานของลูกค้า ซึ่งจะเอาไปเปรียบเทียบกับสล๊อต ของเฟรม อื่นๆ ที่เป็น เฟรมลูกต่อไป

ในงานวิจัยนี้ เราสร้างฟังก์ชันซึ่งจะทำงานเมื่อมีการทริกเกอร์ ด้วยเงื่อนไข ที่ให้มา ซึ่งในเงื่อนไข จะประกอบไปด้วย if_needed, if_changed, constraint และ default ซึ่งเป็นชนิดของ ฟาเซทนั่นเอง ซึ่งในส่วนฟังก์ชันต่างๆ จะถูกเก็บไว้ที่ฝั่งเซิร์ฟเวอร์ และส่วนกฎ ต่างๆ ที่จะกระทำเมื่อมีเหตุการณ์ที่กำหนดไว้ จะเก็บเป็น ชื่อฟังก์ชั่นไว้ในค่าของฟาเซท แล้วตัวทริกเกอร์ จะมีเงื่อนไข เป็นตัวกำหนดว่าจะทำการเรียกใช้ฟังก์ชั่น หรือเรียกดู ข้อมูลขึ้นมา ตามชื่อฟังก์ชั่น ซึ่งวิธีนี้จะทำให้เราสามารถเปรียบเทียบค่าของเฟรมต่อๆ ไปได้ โดยใช้กลไกการเรียกใช้งานฟังก์ชัน ที่ผูกติดไว้กับฟาเซทนี้เอง

เมื่อฝั่งไคลเอนท์ส่งเฟรมข้างต้น ไปยังเซิร์ฟเวอร์เพื่อวิเคราะห์หาโปรโมชันที่สอดคล้องกับความต้องการของลูกค้า ทางฝั่งเซิร์ฟเวอร์ก็จะไล่เปรียบเทียบเฟรมตามลักษณะการใช้งานและความต้องการของลูกค้า แล้วค้นหา ไปยังเฟรมต่างๆ ที่มีคุณสมบัติที่เกี่ยวข้อง ซึ่ง จะตัดเฟรมที่มีคุณสมบัตินอกเหนือจากที่กำหนดมาทิ้งไป ดังเช่นในตัวอย่างนี้ ตัววินิจฉัยจะไล่ไปยังเฟรมลูกของโปรโมชันสำหรับโทรศัพท์เคลื่อนที่ระบบรายเดือน อย่างเช่น กลุ่มโปรโมชันเหมารายวัน ซึ่งพบว่ามีคุณสมบัติคือ เหมาะสำหรับใช้งานสม่ำเสมอ ซึ่งไม่ตรงกับ ลักษณะการใช้งานของลูกค้า ก็จะตัดเฟรมนี้ทิ้ง ไปหาเฟรมใหม่ ซึ่งเป็น เฟรมลูกของ โปรโมชันระบบรายเดือนอีกทีซึ่งก็คือ กลุ่มโปรโมชันเหมาทั้งเดือน ซึ่งมีคุณสมบัติคือ เหมาะสำหรับการใช้งานไม่แน่นอน ซึ่งมีคุณสมบัติตรงกับลักษณะการใช้งานของลูกค้า และจากนั้น กลไกการวินิจฉัย ก็จะไล่ไปยังเฟรมลูกของ กลุ่มโปรโมชันเหมาทั้งเดือน ต่อไปอีก ซึ่ง เหมือนกับที่วินิจฉัยก่อนหน้านี้ จนสุดท้าย ก็จะได้โปรโมชันที่เหมาะสม คือ โปรโมชัน Lite 100 ซึ่งในกรณีนี้ มีโปรโมชันที่เหมาะสมเพียงโปรโมชันเดียว

5.4 การทดสอบระบบต้นแบบที่พัฒนาขึ้น

ในการทดสอบระบบต้นแบบที่พัฒนาขึ้น จะแบ่งการทดสอบระบบต้นแบบที่พัฒนาขึ้นออกเป็นสองส่วนคือ ส่วนแรกเพื่อทดสอบความถูกต้องในการทำงานของระบบที่พัฒนาขึ้น โดยจะให้ผู้ใช้กรอกลักษณะการใช้งาน และความต้องการของลูกค้า เท่าที่ทราบ แล้วให้ระบบทำการวินิจฉัยโดยอาจถามลักษณะการใช้งานของลูกค้าเพิ่มเติม หรืออาจไปสืบค้นข้อมูลขึ้นมาจากประวัติการใช้งานของลูกค้าเพื่อทำการวินิจฉัย คำตอบที่ได้จะเป็นโปรโมชันที่เหมาะสมกับลักษณะการใช้งานและสอดคล้องกับความต้องการของลูกค้า และหากเป็นลูกค้าปัจจุบัน ระบบจะเปรียบเทียบข้อมูลกับโปรโมชันปัจจุบันด้วย และการทดสอบระบบต้นแบบที่พัฒนาขึ้นส่วนที่สองเพื่อเปรียบเทียบการทำงานของระบบ ที่ใช้กลไกการวินิจฉัยสองฝั่ง คือที่ฝั่งเปลือกระบบผู้เชี่ยวชาญและฝั่งฐานความรู้ภายนอก ซึ่งเป็นโครงสร้างของระบบที่นำเสนอ กับการใช้กลไกการวินิจฉัยที่ฝั่งเปลือกระบบผู้เชี่ยวชาญเพียงอย่างเดียว เพื่อพิสูจน์ว่า โครงสร้างที่นำเสนอ สามารถปริมาณในการรับส่งข้อมูลระหว่างฝั่งเซิร์ฟเวอร์และไคลเอนท์ได้จริงหรือไม่ โดยจะทดสอบระบบต้นแบบที่พัฒนาขึ้นโดยการเปรียบเทียบ การถามคำถามไปยัง เซิร์ฟเวอร์ที่ไม่มีกลไกการวินิจฉัยโดยในที่นี้จะกำหนดให้เป็นเซิร์ฟเวอร์ที่มีฐานข้อมูลเป็นฐานข้อมูลสัมพันธ์ทั่วๆไป กับ การถามคำถามไปยังเซิร์ฟเวอร์ของระบบที่นำเสนอ

การทดสอบระบบต้นแบบที่พัฒนาขึ้นส่วนแรก จะทดสอบระบบต้นแบบที่พัฒนาขึ้นโดยสมมติว่า มีลูกค้าที่ต้องการสมัครโปรโมชัน เป็นลูกค้าใหม่ ที่ต้องการใช้โทรศัพท์เคลื่อนที่ระบบรายเดือน โดยลูกค้ามีนิสัยการใช้งานโทรศัพท์เคลื่อนที่แบบไม่แน่นอน และ มีลักษณะการใช้งานโทรศัพท์เคลื่อนที่แบบ โทรนาน ซึ่งในครั้งแรก ระบบจะถามเพียงแค่ว่า ลูกค้าสนใจใช้งานโปรโมชันกลุ่มใด จากนั้น เมื่อได้คำตอบแล้ว ระบบจึงจะวินิจฉัยต่อว่าต้องถามคำถามใดเพิ่มเติม ซึ่งจะ มีขั้นตอนการวินิจฉัยดังนี้

ฝั่งไคลเอนต์

1. ระบบจะทำการเปรียบเทียบค่าที่ได้ป้อนเข้ามา กับเฟรมข้อมูลลูกค้า ที่มีอยู่จะได้มา หนึ่งเฟรมอยู่ในลิสต์ goal_list (คือเฟรมโปรโมชันระบบรายเดือน)
2. จากนั้นจะเลือกเฟรมโปรโมชันระบบรายเดือน (C2 จากตารางที่ 5.2) ขึ้นมาก่อนและตั้งเฟรมโปรโมชันระบบรายเดือนให้เป็นเป้าหมาย
3. เมื่อได้เป้าหมายแล้วก็มาดูลักษณะของโปรโมชันระบบรายเดือน
4. จากเฟรมเป้าหมาย C2 ระบบจะให้ผู้ใช้ตอบคำถามว่า ลูกค้ามีพฤติกรรมการใช้งานโทรศัพท์เคลื่อนที่เป็นอย่างไร
5. ซึ่งในตัวอย่างนี้จะตอบว่า ไม่แน่นอน

6. จากนั้นทริกเกอร์ในสล๊อตนี้ไปเรียกเฟรมถัดมา นั่นก็คือเฟรมโปรโมชันระบบรายเดือนแบบใช้งานไม่แน่นอน ขึ้นมา (C4 จากตารางที่ 5.2)

7. จากนั้นก็ให้ผู้ใช้กรอกข้อมูลเกี่ยวกับลูกค้าเพิ่มเติม คือ ระยะเวลาในการโทรแต่ละครั้ง ซึ่งในที่นี้ ลูกค้าให้ข้อมูลว่า โทรนาน

8. จากนั้นเมื่อระบบรู้แล้วว่าโปรโมชันที่เหมาะสมกับลูกค้า เป็นโปรโมชันระบบรายเดือน ก็จะส่งข้อมูลลูกค้า พร้อมทั้งลักษณะการใช้งานต่างๆ ตามที่ผู้ใช้ได้ป้อนเข้ามาไปยังฝั่งเซิร์ฟเวอร์ (ตามตัวอย่างในรูปที่ 5.2)

ฝั่งเซิร์ฟเวอร์

1. ได้รับข้อมูลว่าเป็นโปรโมชันระบบรายเดือน ที่มีลักษณะการใช้งานที่ไม่แน่นอน และมีระยะเวลาในการโทรนาน

2. ระบบที่ฝั่งเซิร์ฟเวอร์ จะเปรียบเทียบลักษณะของโปรโมชัน โดยเลือกเฟรมเป้าหมายมาไว้ในลิสต์ goal_list ก่อน

3. ซึ่งในตัวอย่างนี้จากเฟรมโปรโมชันระบบรายเดือน จะมีเฟรมลูกอยู่สี่เฟรม ซึ่งก็คือ เฟรมโปรโมชันรายเดือนแบบใช้งานสม่ำเสมอ, โปรโมชันระบบรายเดือนแบบใช้งานไม่แน่นอน, โปรโมชันระบบรายเดือนแบบเน้นใช้งานตามช่วงเวลา และ โปรโมชันระบบรายเดือนเน้นใช้งานบางหมายเลข (P2, P3, P4, P5 ในตารางที่ 5.1 ตามลำดับ)

4. ในตอนแรกระบบจะทำการเลือกเฟรมโปรโมชันระบบรายเดือนแบบใช้งานสม่ำเสมอ คือ P2 ออกมาจาก goal_list ก่อน

5. เมื่อพิจารณาเฟรม P2 จะพบว่ามีค่าในสล๊อต พฤติกรรมการใช้งาน เป็น สม่ำเสมอ

6. ระบบจะทำการเปรียบเทียบค่าของสล๊อตกับเฟรมที่ถูกส่งเข้ามาที่ฝั่งเซิร์ฟเวอร์

7. ซึ่งในตัวอย่างนี้จะพบว่าค่าในสล๊อต พฤติกรรมการใช้งาน ไม่ตรงกับเฟรมที่ส่งเข้ามา ก็จะปฏิเสธเฟรมนี้แล้วไปเลือกเฟรมเป้าหมายใหม่ซึ่งก็คือเฟรม P3 มาแทน

8. เมื่อได้เฟรมโปรโมชันระบบรายเดือนแบบใช้งานไม่แน่นอน (P3) แล้ว ระบบก็จะเปรียบเทียบค่าสล๊อตในเฟรม P3 กับค่าที่ถูกส่งเข้ามาซึ่งในตัวอย่างนี้จะมีค่าตรงกันคือ ไม่แน่นอน

9. ระบบจะทำการตรวจสอบว่าเฟรมที่ได้มานี้มีเฟรมลูกอีกหรือไม่

10. ซึ่งในตัวอย่างนี้เฟรม P3 จะมีเฟรมลูกคือ (P10, P11, P12 จากตารางที่ 5.1 ตามลำดับ)

11. ระบบจะทำการเก็บเฟรมนั้นเพิ่มเข้าไปในลิสต์เป้าหมาย goal_list ในลักษณะของสแต็ก (คือเข้าไปที่หลังออกมาก่อน)

12. ระบบจะเลือกเฟรม P10 ขึ้นมาก่อนแล้วทำการจับคู่สล๊อตของเฟรม กับค่าที่มี

13. จากตัวอย่างนี้ เฟรม P10 จะมีสล๊อตเพิ่มขึ้นมาอีกหนึ่งสล๊อต คือระยะเวลาในการโทร มีค่าเป็น คุยนาน ซึ่งจะตรงกับค่าที่มี

14.และเมื่อระบบทำการตรวจสอบ ก็พบว่าเฟรม P10 ไม่มีเฟรมลูกอีกแล้ว

15.ระบบก็จะส่งเฟรมที่ได้ กลับไปยังฝั่งไคล์เอนท์ เป็นคำตอบว่า โปรโมชันใดมีความเหมาะสม ซึ่งก็คือเฟรม P10 หรือโปรโมชัน Lite 100 นั่นเอง

จากผลการทดสอบระบบต้นแบบที่พัฒนาขึ้นส่วนแรกนี้ ทำให้พบว่าระบบมีความสามารถในการวิเคราะห์หา โปรโมชันที่เหมาะสมกับลักษณะการใช้งานและความต้องการของลูกค้าได้ และสามารถส่งผลลัพธ์ที่ถูกต้องกลับไปยังฝั่งไคลเอนท์ได้ถูกต้อง แต่ผู้ใช้จะต้องป้อนข้อมูลเข้ามาในระบบพอสมควรก่อน จึงจะสามารถทำการวินิจฉัยต่อไปได้ ถ้าผู้ใช้ไม่ป้อนข้อมูลเข้ามาในระบบเลย แต่ถามถึงโปรโมชันที่มีความเหมาะสม ระบบจะไม่สามารถตอบคำถามของผู้ใช้ได้ เพราะระบบไม่มีข้อมูลที่จะใช้เพื่อตั้งเป็นเป้าหมายได้

จากผลการทดสอบต้นแบบที่พัฒนาขึ้นส่วนแรกจะสรุปได้ว่า ระบบจะทำการวินิจฉัยได้ ก็ต่อเมื่อมีข้อเท็จจริงเริ่มต้นบางส่วนถูกป้อนเข้าไปในระบบก่อน จากนั้นระบบถึงจะสามารถทำการวินิจฉัยแล้วทำการหาคำตอบต่อไปได้ หรือสามารถถามคำถามเพิ่มเติมกับผู้ใช้ได้ หรือไปหาข้อเท็จจริงที่ผู้ในฐานข้อมูลภายนอกอื่นได้

การทดสอบระบบต้นแบบที่พัฒนาขึ้นส่วนที่สอง จะทำโดยการถามคำถามไปยัง เซิร์ฟเวอร์เพื่อวินิจฉัย โดยจะส่งคำถามไปยัง เซิร์ฟเวอร์ สองเซิร์ฟเวอร์ ที่มีข้อมูลเหมือนกัน แต่เซิร์ฟเวอร์หนึ่งจะไม่มีกลไกการวินิจฉัย และ อีกเซิร์ฟเวอร์หนึ่ง จะมีกลไกการวินิจฉัย ซึ่งคำถามที่จะส่งไปยังเซิร์ฟเวอร์ทั้งสองคือ คำถามว่า ลักษณะการใช้งานโทรศัพท์เคลื่อนที่ของลูกค้าปัจจุบันหมายเลขโทรศัพท์ 081-444-1278 ซึ่งมีทรานแซคชัน (Transaction) การโทรออกเก็บไว้ในฐานความรู้ต้นแบบทั้งสิ้น 647 ทรานแซคชัน ว่ามีลักษณะการใช้งานเหมาะสมกับโปรโมชันใด

โดยการถามคำถามไปยังเซิร์ฟเวอร์ที่ไม่มีกลไกการวินิจฉัย ระบบจะต้องถามหาข้อเท็จจริงที่เกี่ยวข้องทั้งหมด เพื่อนำกลับมาประกอบการวินิจฉัยที่ฝั่งไคลเอนท์ อีกที โดยในการถามหาข้อเท็จจริง จะทำโดยการของการเรียกใช้งานเว็บเซอร์วิซ โดยการใส่พารามิเตอร์เป็น คำสั่งภาษา SQL ดังนี้ Find_Facts(“SELECT * FROM CallOut_Detail WHERE MOBNUMBER= ‘0814441278’”) จากนั้นเซิร์ฟเวอร์จะทำการตอบกลับไปยังไคลเอนท์เป็นผลลัพธ์ที่ได้จากการใช้คำสั่ง SQL ที่ส่งมาโดยผลลัพธ์ จะอยู่ในรูปของเอกสาร XML ซึ่งจะมีข้อมูลเป็นจำนวนมาก ตามระยะเวลาการใช้งานของลูกค้าหมายเลขนี้ เพื่อให้ไคลเอนท์ นำไปวินิจฉัยต่ออีกที โดย ในรูปที่ 5.3 จะแสดงข้อมูลเพียงส่วนหนึ่งเพื่อเป็นตัวอย่างเท่านั้น

```
...
<ROW>  <MOBNUMBER>0814441278</ MOBNUMBER >
       <DESNUM>0814457898</DESNUM>
       <TIMECALLOUT>21052009110455</TIMECALLOUT>
       <CALLDURATION>17.24</CALLDURATION></ROW>
<ROW>  <MOBNUMBER>0814441278</ MOBNUMBER >
       <DESNUM>0815457823</DESNUM>
       <TIMECALLOUT>21052009114012</TIMECALLOUT>
       <CALLDURATION>18.46</CALLDURATION></ROW>
<ROW>  <MOBNUMBER>0814441278</ MOBNUMBER >
       <DESNUM>026913863</DESNUM>
       <TIMECALLOUT>21052009121023</TIMECALLOUT>
       <CALLDURATION>29.04</CALLDURATION></ROW>
<ROW>  <MOBNUMBER>0814441278</ MOBNUMBER >
       <DESNUM>0865559991</DESNUM>
       <TIMECALLOUT>21052009151414</TIMECALLOUT>
       <CALLDURATION>32.11</CALLDURATION></ROW>
<ROW>  <MOBNUMBER>0814441278</ MOBNUMBER >
       <DESNUM>0866797979</DESNUM>
       <TIMECALLOUT>21052009171947</TIMECALLOUT>
       <CALLDURATION>68.24</CALLDURATION></ROW>
<ROW>  <MOBNUMBER>0814441278</ MOBNUMBER >
       <DESNUM>036419578</DESNUM>
       <TIMECALLOUT>21052009213417</TIMECALLOUT>
       <CALLDURATION>16.51</CALLDURATION></ROW>
...
```

รูปที่ 5.3 ส่วนผลลัพธ์การสอบถามโปรโมชันที่เหมาะสมไปยังเซิร์ฟเวอร์ที่ไม่มีกลไกการวินิจฉัย

ในกรณีเดียวกัน สำหรับ ฐานความรู้ที่มีกลไกการวินิจฉัย ไคลเอนท์จะส่งคำถามไปในลักษณะของการเรียกใช้งานเว็บเซอร์วิซ โดยการใส่พารามิเตอร์เป็น หมายเลขโทรศัพท์ ดังนี้ Find_Promotion("0814441278") จากนั้น กลไกการวินิจฉัยที่ฝั่งเซิร์ฟเวอร์จะทำการนำหมายเลขโทรศัพท์ ไปสร้างเฟรมลักษณะคาร์ใช้งานโทรศัพท์ของหมายเลขโทรศัพท์นี้ โดยค่าต่างๆของสล๊อต จะถูกดึงมาจากฐานข้อมูล RDF/XML ที่เก็บข้อมูลประวัติการโทรออก เมื่อทำการวินิจฉัย จนได้เฟรม โปรโมชันที่เหมาะสมแล้ว ระบบจะทำการส่งค่า ชื่อโปรโมชันที่เหมาะสม กลับไปให้ไคลเอนท์ในรูปของ เอกสาร XML ดังรูปที่ 5.4

```
<?xml version= "1.0"?>
<sparql xmlns= "http://www.w3c.org/2005/sparql-results#">
    <head>
        <variable name= "Promotion"/>
    </head>
    <results>
        <result>
            <binding name= "Promotion">
                <literal>Lite100</literal>
            </binding>
        </result>
    </results>
</sparql>
```

รูปที่ 5.4 ผลลัพธ์การสอบถามโปรโมชันที่เหมาะสมไปยังเซิร์ฟเวอร์ที่มีกลไกการวินิจฉัย

จากการทดสอบระบบต้นแบบ ในส่วนที่สองนี้ พบว่าการที่โครงสร้างของระบบผู้เชี่ยวชาญ มีกลไกการวินิจฉัยเพิ่มที่ฝั่งฐานความรู้ จะเป็นการช่วยลดปริมาณในการส่งข้อมูลระหว่างไคลเอนท์และเซิร์ฟเวอร์ได้ เป็นอย่างมาก เนื่องจากมีปริมาณข้อมูลในการส่งที่น้อยกว่ามาก

สรุปการทดสอบต้นแบบระบบผู้เชี่ยวชาญที่พัฒนาขึ้น โดยเพิ่มกลไกการวินิจฉัยที่ฝั่งฐานความรู้ พบว่าระบบสามารถทำการวินิจฉัยได้ถูกต้อง และมีการส่งข้อมูลระหว่างฐานข้อมูลและเปลือกระบบผู้เชี่ยวชาญในปริมาณที่น้อยกว่าระบบผู้เชี่ยวชาญแบบเดิม มาก

บทที่ 6
สรุปผล และข้อเสนอแนะ

จากวัตถุประสงค์หลักของวิทยานิพนธ์ฉบับนี้ เพื่อนำเสนอการพัฒนาต้นแบบระบบผู้เชี่ยวชาญแบบเฟรม โดยใช้ฐานข้อมูล RDF/XML ที่เพิ่มกลไกการวินิจฉัย ในส่วนฐานความรู้ เพื่อให้สามารถวินิจฉัยข้อมูลได้ทั้งฝั่งเปลือกระบบผู้เชี่ยวชาญ และฝั่งฐานความรู้ภายนอก โดยผู้วิจัยได้นำเอาระบบการวิเคราะห์โปรโมชันสำหรับโทรศัพท์เคลื่อนที่ มาสร้างเป็นระบบผู้เชี่ยวชาญ เพื่อเป็นกรณีศึกษาให้กับแนวคิดที่ผู้วิจัยได้นำเสนอไว้ ซึ่งจากการดำเนินการวิจัยทำให้ผู้วิจัยสามารถสรุปผล และข้อเสนอแนะเพื่อเป็นแนวทางในการนำแนวคิดจากวิทยานิพนธ์ฉบับนี้ไปพัฒนาต่อในอนาคต โดยมีรายละเอียดดังต่อไปนี้

6.1 สรุปผลการทดสอบระบบต้นแบบ

วิทยานิพนธ์ฉบับนี้นำเสนอการพัฒนาต้นแบบระบบผู้เชี่ยวชาญแบบเฟรม โดยใช้ฐานข้อมูล RDF/XML เพิ่มกลไกการวินิจฉัยในส่วนฐานความรู้ภายนอก เพื่อให้สามารถวินิจฉัยข้อมูลได้ทั้งฝั่งเปลือกระบบผู้เชี่ยวชาญ และฝั่งฐานความรู้ภายนอก ซึ่งจากสมมติฐานพบว่าสามารถพัฒนาต้นแบบระบบผู้เชี่ยวชาญโดยใช้การแทนความรู้แบบเฟรมบนฐานข้อมูล RDF/XML ได้

จากการนำเฟรม ไปจัดเก็บในฐานข้อมูล RDF/XML นั้นพบว่า โครงสร้างของ ฐานข้อมูล RDF/XML สามารถสร้างให้มีความคล้ายคลึงกับหลักการเชิงวัตถุ ซึ่งเอื้อประโยชน์ต่อการประมวลผลจากเฟรมได้ ดังนั้นวิทยานิพนธ์ฉบับนี้จึงสามารถนำหลักการจัดเก็บข้อมูลด้วยฐานข้อมูล RDF/XML มาจัดเก็บข้อมูลการแทนความรู้แบบเฟรมได้โดยง่าย

ผลการทดสอบ ระบบต้นแบบที่พัฒนาขึ้นสามารถทำการวินิจฉัยได้ถูกต้อง และในการที่ฝั่งฐานความรู้สามารถวินิจฉัยข้อมูลได้ด้วย เป็นการช่วยลดปริมาณในการรับส่งข้อมูลระหว่างเซิร์ฟเวอร์และไคลเอนท์เป็นอย่างมาก ตรงตามที่ตั้งสมมติฐานไว้ โดยในการพัฒนาต้นแบบระบบผู้เชี่ยวชาญการวิเคราะห์โปรโมชันสำหรับโทรศัพท์เคลื่อนที่นี้ ได้ศึกษาจากผู้เชี่ยวชาญทางด้านสถิติของบริษัทผู้ให้บริการเครือข่ายโทรศัพท์เคลื่อนที่ขนาดใหญ่แล้วพบว่า ระบบต้นแบบสามารถตอบคำถามในลักษณะของผู้เชี่ยวชาญได้ แต่ยังไม่สามารถทำการวินิจฉัยได้ถูกต้องร้อยเปอร์เซ็นต์ตามแบบผู้เชี่ยวชาญ

อย่างไรก็ตาม วิทยานิพนธ์ฉบับนี้พัฒนาต้นแบบระบบผู้เชี่ยวชาญบนฐานข้อมูล RDF/XML ซึ่งเป็นการจัดเก็บข้อมูลที่มีแนวโน้มที่จะเป็นรูปแบบการจัดเก็บข้อมูลที่ได้รับความนิยมอย่างกว้างขวางในอนาคต อีกทั้งเป็นการจัดเก็บข้อมูลที่รองรับการจัดเก็บข้อมูลบนเว็บอีกด้วย จึงมีความ

เป็นไปได้ในการที่จะพัฒนาระบบผู้เชี่ยวชาญแบบเฟรมบนฐานข้อมูล RDF/XML นี้ให้จัดเก็บและสามารถสืบค้นข้อมูลบนอุปกรณ์ไร้สายต่างๆในอนาคตอย่างมีประสิทธิภาพได้

จากการทดสอบสมมติฐานดังกล่าวสามารถสรุปได้ว่าแนวคิดในการพัฒนาต้นแบบระบบผู้เชี่ยวชาญแบบเฟรมบนฐานข้อมูล RDF/XML ใช้งานได้ และการที่นำการจัดเก็บข้อมูลในฐานข้อมูล RDF/XML มาประยุกต์ใช้ ทำให้เป็นแนวทางในการพัฒนาต้นแบบระบบผู้เชี่ยวชาญต่อไปในอนาคตได้เป็นอย่างดี

6.2 ข้อเสนอแนะ

วิทยานิพนธ์ฉบับนี้ได้มีการจัดเก็บเฟรมไว้ที่ฝั่งไคลเอนต์และฝั่งเซิร์ฟเวอร์ในฐานข้อมูล RDF/XML โดยที่ทั้งฝั่งไคลเอนท์ และเซิร์ฟเวอร์เป็นเครื่องคอมพิวเตอร์ธรรมดา ซึ่งสามารถประยุกต์ใช้ การเก็บข้อมูลในรูปแบบเอกสาร RDF/ XML ในการจัดเก็บเฟรมบนอุปกรณ์ไร้สายต่างๆ เช่น PDA หรือ โทรศัพท์เคลื่อนที่ได้ ในอนาคต

นอกจากนี้ เนื่องจากในวิทยานิพนธ์ฉบับนี้มีการวินิจฉัยได้สองทางซึ่งฝั่งไคลเอนต์จะเป็นส่วนติดต่อกับผู้ใช้และส่งข้อมูลไปยังเซิร์ฟเวอร์เพื่อทำการวินิจฉัยต่อในลักษณะที่ไคลเอนต์จะรู้อยู่แล้วว่าควรจะติดต่อไปยังเซิร์ฟเวอร์ไหนซึ่งสามารถพัฒนาต่อยอดให้ไคลเอนต์สามารถค้นหาได้โดยอัตโนมัติว่ามีเซิร์ฟเวอร์ใดบ้างที่สามารถทำการวินิจฉัยต่อได้

ในวิทยานิพนธ์ฉบับนี้ระบบต้นแบบนั้นถูกออกแบบเพื่อทดสอบสมมติฐานเท่านั้น

เอกสารอ้างอิง

[1] Nata lya 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] S uphamit Chittayasothorn and Chuleerat Rattanaprateep, **Expert Database System Architecture and Implementation on Object Relational Databases**, WSEAS Transactions on Computers Issue 3, Vol.5, March 2006

[5] W3C **"RDF Primer"** [Online] Available: http://www.w3.org/TR/REC-rdf-syntax/

[6] W3C **"SPARQL Query Language for RDF"** [Online] Available: http://www.w3.org/TR/rdf-sparql-query/

[7] Stanford University **"Protégé Documentation"** [Online] Available: http://protege.stanford.edu/doc/users.html

[8] Peter Lucas and Linda van d er 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] W3C **"SPARQL Protocol for RDF"** http://www.w3.org/TR/rdf-sparql-protocol/

[19] Thierry Barsalou, M.D. "An Object-Based Architecture for Biomedical Expert Database Systems", **Proceedings of the Annual Symposium on Computer Application in Medical Care**, November 9, 1988, P 572–578.

[20] Tsur, S. "LDL-a technology for the realization of tightly coupled expert database systems", **IEEE Expert**, Vol. 3, Issue 3, 1988

[21] Kerschberg, L. "The role of loose coupling in expert database system architectures", **Proceedings. Fifth International Conference on Data Engineering**, 6-10 Feb 1989, Page(s): 255-256.

[22] Yoon, S.-C. Henschen, L.J. "Intelligent control in expert database systems using a heuristic approach", **Proceedings. IEEE International Symposium on Intelligent Control**, 1989. Page(s): 399 - 405

[23] Ruparel, B. "Designing and implementing intelligent database applications-a case study", **Proceedings. IEEE Conference on Managing Expert System Programs and Projects**, 10-12 Sep 1990, Page(s): 223-229

[24] Bechtold, R.T. "Knowledge attributes: fuzzy application of temporal constraints in active expert database systems", **Proceedings of the IEEE/ACM International Conference on Developing and Managing Expert System Programs,** 30 Sept.-2 Oct. 1991, Page(s):29-35

[25] Szabo, M.E. "Inference strategies for expert database systems", **Canadian Conference on Electrical and Computer Engineering**, 14-17 Sept. 1993, Vol.1, Page(s): 5 - 8

[26] Dewan, H.M. Stolfo, S.J. "Meta-level control of rule execution in a parallel and distributed expert database system", **Research Issues in Data Engineering, 1994. Proceedings Fourth International Workshop on Active Database Systems.** 14-15 Feb. 1994 , Page(s): 105 – 114

[27] Zhu Yanqin Li Fanzhang Hu Yuemei "The design and application of dynamic fuzzy expert database system", **IEEE International Conference on Granular Computing,** 25-27 July 2005,Volume: 1, Page(s): 378 - 381

ภาคผนวก

ภาคผนวก ก.

เฟรมต้นแบบที่จัดเก็บในฐานข้อมูล RDF/XML

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
   <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
   <!ENTITY swrl "http://www.w3.org/2003/11/swrl#" >
   <!ENTITY swrlb "http://www.w3.org/2003/11/swrlb#" >
   <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
   <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
   <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
   <!ENTITY protege "http://protege.stanford.edu/plugins/owl/protege#" >
   <!ENTITY xsp "http://www.owl-ontologies.com/2005/08/07/xsp.owl#" >
]>
<rdf:RDF xmlns="http://www.owl-ontologies.com/Ontology1233987595.owl#"
    xml:base="http://www.owl-ontologies.com/Ontology1233987595.owl"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
    xmlns:swrl="http://www.w3.org/2003/11/swrl#"
    xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
    xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:owl="http://www.w3.org/2002/07/owl#">
   <owl:Ontology rdf:about="Frame"/>
   <owl:Class rdf:ID="Facets">
      <rdfs:subClassOf rdf:resource="&owl;Thing"/>
      <rdfs:subClassOf>
         <owl:Restriction>
            <owl:onProperty rdf:resource="#FacetValue"/>
            <owl:cardinality rdf:datatype="&xsd;int">1</owl:cardinality>
         </owl:Restriction>
      </rdfs:subClassOf>
   </owl:Class>
   <owl:DatatypeProperty rdf:ID="FacetType">
```

```
    <rdfs:domain rdf:resource="#Facets"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="FacetValue">
    <rdfs:domain rdf:resource="#Facets"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="FrameName">
    <rdfs:domain rdf:resource="#Frames"/>
    <rdfs:range rdf:resource="&xsd;string"/>
  </owl:DatatypeProperty>
 <owl:Class rdf:ID="Frames">
    <rdfs:subClassOf>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:onProperty rdf:resource="#hasChildren"/>
            <owl:someValuesFrom rdf:resource="#Frames"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty rdf:resource="#hasParent"/>
            <owl:someValuesFrom rdf:resource="#Frames"/>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty rdf:resource="#hasSlot"/>
            <owl:someValuesFrom rdf:resource="#Slots"/>
          </owl:Restriction>
        </owl:unionOf>
      </owl:Class>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="#FrameName"/>
        <owl:cardinality rdf:datatype="&xsd;int">1</owl:cardinality>
```

```
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="&owl;Thing"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="hasChildren">
    <rdfs:domain rdf:resource="#Frames"/>
    <owl:inverseOf rdf:resource="#hasParent"/>
    <rdfs:range rdf:resource="#Frames"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasFacet">
    <rdfs:domain rdf:resource="#Slots"/>
    <owl:inverseOf rdf:resource="#isFacetOf"/>
    <rdfs:range rdf:resource="#Facets"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasParent">
    <rdfs:domain rdf:resource="#Frames"/>
    <owl:inverseOf rdf:resource="#hasChildren"/>
    <rdfs:range rdf:resource="#Frames"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasSlot">
    <rdfs:domain rdf:resource="#Frames"/>
    <owl:inverseOf rdf:resource="#isInFrame"/>
    <rdfs:range rdf:resource="#Slots"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isFacetOf">
    <rdfs:domain rdf:resource="#Facets"/>
    <owl:inverseOf rdf:resource="#hasFacet"/>
    <rdfs:range rdf:resource="#Slots"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isInFrame">
    <rdfs:domain rdf:resource="#Slots"/>
    <owl:inverseOf rdf:resource="#hasSlot"/>
```

```
    <rdfs:range rdf:resource="#Frames"/>
  </owl:ObjectProperty>
  <owl:DatatypeProperty rdf:ID="SlotName">
    <rdfs:domain rdf:resource="#Slots"/>
    <rdfs:range rdf:resource="&xsd;string"/>
  </owl:DatatypeProperty>
<owl:Class rdf:ID="Slots">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="#SlotValue"/>
        <owl:maxCardinality rdf:datatype="&xsd;int">1</owl:maxCardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="#SlotName"/>
        <owl:cardinality rdf:datatype="&xsd;int">1</owl:cardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="&owl;Thing"/>
  </owl:Class>
  <owl:DatatypeProperty rdf:ID="SlotValue">
    <rdfs:domain rdf:resource="#Slots"/>
  </owl:DatatypeProperty>
</rdf:RDF>
```

ภาคผนวก ข.

งานวิจัยที่ได้รับการตีพิมพ์

WORLD SCIENTIFIC and ENGINEERING
ACADEMY and SOCIETY
http://www.wseas.org

Professor Charles A. Long, WSEAS President, Professor Emeritus, University of Wisconsin, Stevens Point, Wisconsin, USA	WSEAS, A. I. Theologou 17-23, 15773, Zographou, Athens, GREECE
WSEAS, Philippe Dondon, ENSEIRB rue A Schweitzer 33400, Talence, FRANCE	WSEAS, Yiming Li, Microelectronics & Inform. Research Center, National Chiao Tung Univ., Hsinchu 300, TAIWAN
WSEAS, Valeri M. Mladenov, Technical University of Sofia Sofia-1000, BULGARIA	WSEAS, Petr Ekel, Av. Dom Jose Gaspar, 500, 30535-610 - Belo Horizonte – MG, BRAZIL
WSEAS, Alexander Zemliak, Physics & Mathem. Dept., Puebla Auton. University, MEXICO	WSEAS, Minh Hung Le, School of Electrical and Computer Engineering, International Business College, AUSTRALIA

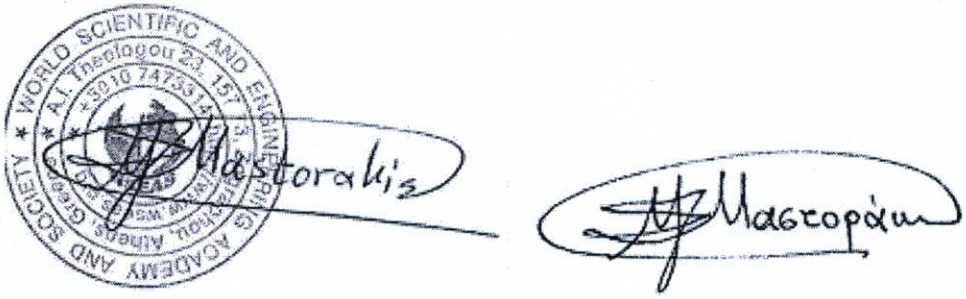
May 13, 2009

RE: Paper 29-230 An RDF-based Distributed Expert System

Dear Napat Prapakorn and Suphamit Chittayasothorn,

We are pleased to inform you that your paper has been accepted and will be published in the *WSEAS Transactions on Computers* www.wseas.org. The WSEAS Journals participate in EI Compendex, ACM, IET (IEE), SCOPUS, ASM, ACM, ACS, CSA, ELSEVIER , ZENTRALBLATT, MATHSCINET, DPP,EI, CSBA, Ulrigh, DEST, EBSCO, EMBASE, GEOBASE, BIOBASE, BIOTECHNOBASE, FLUIDEX , OceanBase, Also, except the WSEAS Books and proceedings as high-quality periodical Publications participate in SCI , ISI and in EI Compendex, ACM, IET (IEE), SCOPUS, ASM, ACM, ACS, CSA, ELSEVIER , ZENTRALBLATT, MATHSCINET, DPP,EI, CSBA, Ulrigh, DEST, EBSCO, EMBASE, GEOBASE, BIOBASE,BIOTECHNOBASE, FLUIDEX , OceanBase. More details: www.wseas.us

Best Regards



Prof. N. Mastorakis
* Professor, Technical University of Sofia, BULGARIA
http://elfe.tu-sofia.bg/elfe/staff.htm
http://elfe.tu-sofia.bg/elfe/curriculum4.htm
Honorary Professor, University of Cluj, ROMANIA
http://outstanding.wseas.us
Biography:
http://www.wseas.org/mastorakis/CV_Mastorakis.doc

An RDF-based Distributed Expert System

NAPAT PRAPAKORN*, SUPHAMIT CHITTAYASOTHORN**
Department of Computer Engineering
King Mongkut's Institute of Technology Ladkrabang
Faculty of Engineering, Bangkok 10520
THAILAND
Napat.Prapakorn@gmail.com*, suphamit@kmitl.ac.th**

ABSTRACT – An expert system or knowledge-based system comprises of a knowledge base and an inference engine in which their expertise knowledge is represented. The knowledge can be called upon when needed to solve a problem by the inference engine. In a large expert system, the knowledge base can be represented using frames. Hence they are called frame-based expert system or frame-based system. To avoid too many communication traffics during inferences, a distributed expert system, an expert system with an inference engine on its external knowledge base side is presented. It is an expert system which has an RDF external knowledge base for improved flexibility and mobility in knowledge sharing. This research presents a design and implementation of a Frame-based RDF expert system which has a RDF/XML database as its external knowledge base. The external knowledge base uses frame as its knowledge representation stored in RDF/XML format so that it can be placed on the WWW (World Wide Web) which is, ideally, accessible from anywhere. With this capability, an expert system will be enriched with flexibility and mobility in knowledge sharing.

Key-Words: - Frames, Expert system, Knowledge base, RDF, RDF/XML, Ontology

1 Introduction

Frames are widely used as the knowledge representation of large, complex expert systems [1]. Such expert system comprises internal knowledge base and inference engine in which their knowledge base is used to represent their expertise knowledge as frames. To avoid too many communication traffics during inferences, a database system is presented to store knowledge as an external knowledge base along with an inference engine of its own. With a tight coupling between an expert system and an external knowledge base, inferences can be performed on the external knowledge base then the inference result, and not only simple facts, are sent to a client expert system for further inferences [2]. In some distributed systems, mobile agents are used to reduce the communication problem [13]. However, the systems may not be very efficient in a resource-limited environment due to its lack of flexibility and mobility in knowledge sharing.

To improve flexibility and mobility, a Resource Description Framework (RDF) is presented. RDF is a standard for representing information about resources in the World Wide Web. RDF is intended for situations in which the information needs to be processed by application, rather than being displayed to people. Moreover, RDF also provides a common framework for expressing information between applications without loss of meaning [3]. With RDF concepts, we could simply store our expertise knowledge on the WWW which is, ideally, accessible from anywhere. Thus, this research proposes a frame-based RDF expert system, an RDF-based distributed expert system, as an alternative expert system with flexibility and mobility, using a collection of RDF/XML databases as its knowledge base.

2 Resource Description Framework (RDF)

A Resource Description Framework (RDF) is a standard developed by W3C for representing information about resources in the World Wide Web. RDF uses a Uniform Resource Identifier Reference (URIref), a URI together with an optional fragment identifier at the end, as its mechanism for identifying resources. With the capability of URIref, not only network-accessible resources but things that are not network-accessible such as human beings, corporations or even abstract concepts that do not physically existed can also be identified by RDF.

A complete detail of RDF is available on the Website: http://www.w3.org/RDF/. In this section, we briefly review the RDF concepts.

2.1 RDF Statement

RDF is based on the idea that things being described have properties which have values. RDF describes things by making statements that specify those properties and values. RDF statement is divided into three parts; the part that identifies the thing being described is called the "Subject"; the part that identifies the property or characteristic of the subject is called the "Predicate"; the part that identifies the value of that property is called the "Object" [3]. For example:

http://www.example.org/index.html has a creator whose value is John Smith.

The statement can be divided into three parts:

"http://www.example.org/index.html" is the subject.
"creator" is the predicate.
"John Smith" (who is identified by http://www.example.org/staffid/85740) is the object.

2.2 RDF Conceptual Model

RDF models statements as nodes and arcs in a graph. In this notation, RDF statements are represented by:

- A node for the subject
- A node for the object
- An arc for the predicate, directed from the subject node to the object node

So the RDF statement above would be represented by the graph shown in Fig.1:

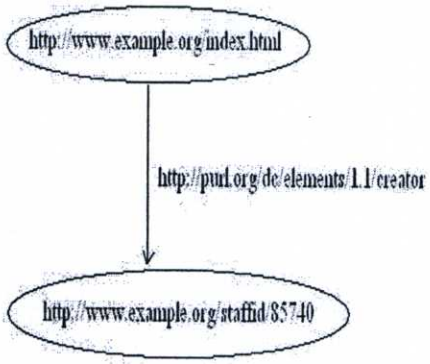


Fig.1 RDF graph [3]

Sometimes it is not convenient to draw graphs, so an alternative way of writing down the statements, called triples, is also used. In a triples notation, each statement in the graph is written as a simple triple of subject, predicate, object, respectively [3]. For example, the statement shown in Fig. 1 would be written in the tripples notation as:

<http://www.example.org/index.html>
<http://purl.org/dc/elements/1.1/creator>
<http://www.example.org/staffid/85740>

2.3 An XML syntax for RDF: RDF/XML

RDF also provides an XML-based syntax, called RDF/XML, for recording and exchanging these graphs. Fig. 2 is an RDF in RDF/XML corresponding to the graph in Fig.1:

```
1.<?xml version="1.0"?>
2.<rdf:RDF xmlns:ex="http://www.example.org/"
3.         xmlns:dc=
    "http://purl.org/dc/elements/1.1"
4          xmlns:exstaff=
    "http://www.example.org/staffid/" >
5.         <rdf:Description rdf:about=
    "http://www.example.org/index.html">
6.           <dc:creator>
7.             <rdf:Description>
8.                <exstaff:85740
    rdf:resource=
    "http://www.example.org/staffid/85740"/>
9.             </rdf:Description>
10.          </dc:creator>
11.    </rdf:Description>
12.</rdf:RDF>

(Line numbers are added to help in explaining the
example.)
```

Fig. 2 RDF/XML Example

Line 1, <?xml version="1.0"?>, is the XML declaration, which indicates that the following content is XML, and the version of XML. Line 2 begins an rdf:RDF element. This indicates that the following XML content (starting here and ending with the </rdf:RDF> in line 12) is intended to represent RDF. Following the rdf:RDF on this same line is an XML namespace declaration, represented as an xmlns attribute of the rdf:RDF start-tag. This declaration specifies that all tags in this content

prefixed with rdf: are part of the namespace identified by the URIref http://www.w3.org/1999/02/22-rdf-syntax-ns#. URIrefs beginning with the string http://www.w3.org/1999/02/22-rdf-syntax-ns# are used for terms from the RDF vocabulary. Line 3 and Line 4 specifies another XML namespace declaration. The ">" at the end of line 4 indicates the end of the rdf:RDF start-tag. Lines 1-4 are basically necessary to indicate that this is RDF/XML content, and to identify the namespaces being used within the RDF/XML content. Lines 5-11 provide the RDF/XML for the specific statement shown in Fig.1. An obvious way to talk about any RDF statement is to say it is a description, and that it is about the subject of the statement (in this case, about http://www.example.org/index.html), and this is the way RDF/XML represents the statement. The rdf:Description start-tag in line 5 indicates the start of a description of a resource, and goes on to identify the resource the statement is about (the subject of the statement) using the rdf:about attribute to specify the URIref of the subject resource. Line 6 provides a property element, with dc:creator as its tag, to represent the predicate and object of the statement. The content of this property element is the object of the statement, with exstaff:85740 as its tag using the rdf:resource attribute to specify the URIref of the object resource with no further sub properties. The property element is nested within the containing rdf:Description element, indicating that this property applies to the resource specified in the rdf:about attribute of the rdf:Description element. Line 11 indicates the end of the first rdf:Description element. Finally, Line 12 indicates the end of the rdf:RDF element started on line 2.

2.4 RDF Schema

RDF provides a way to express simple statements about resources, using named properties and values. However, RDF user communities also need the ability to define the vocabularies (terms) they intend to use in those statements, specifically, to indicate that they are describing specific kinds or classes of resources, and will use specific properties in describing those resources. For example, some applications might need to describe classes such as ex:Person and ex:Company, and properties such as ex:age, ex:jobTitle, and ex:numberOfEmployees. RDF itself cannot define such application-specific classes and properties. Instead, such classes and properties are described as an RDF vocabulary, using extensions to RDF provided by the RDF Vocabulary Description Language, called RDF Schema [3]. The basic concept of RDF Schema and its property are illustrated in Fig. 3. More details of each element are described in Appendix A and Appendix B in Appendix section.

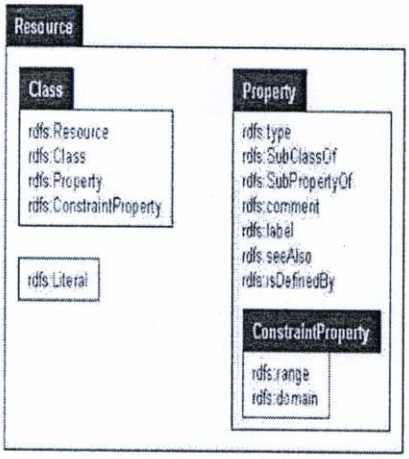


Fig. 3 Concept of RDF Schema

3 SPARQL Query Language for RDF

With an RDF approach, W3C also propose SPARQL as a standard query language for RDF. SPARQL can be used to express queries across diverse data sources, whether the data is stored natively as RDF documents or viewed as RDF documents via a middleware. SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. SPARQL also supports extensible value testing and constraining queries by source RDF graph. The results of SPARQL queries can be results sets or RDF graphs and also can be serialized into XML format.

Most forms of SPARQL query contain a set of triple patterns called a basic graph pattern. Triple patterns are like RDF triples except that each of the subject, predicate and object may be a variable. A basic graph pattern matches a subgraph of the RDF data when RDF terms from that subgraph may be substituted for the variables and the result is RDF graph equivalent to the subgraph [11].

In this section, we briefly illustrate the SPARQL basics using the data source in Fig. 4. A complete detail of SPARQL Query Language for RDF can be

found on the Website: http://www.w3.org/TR/rdf-sparql-query/.

@prefix dc: <http://purl.org/dc/elements/1.1/>
@prefix book: <http://example.org/book/>
@prefix writer: <http://example.org/writer/>

book:book1	dc:title	"Harry Potter and the Deadly Hallows"
book:book2	dc:title	"The Lord of the Ring"
writer:ID001	dc:name	"J.K. Rowling"
writer:ID002	dc:name	"J.R.R. Tolkien"
book:book1	dc:auther	writer:ID001
book:book2	dc:auther	writer:ID002

Fig. 4 Data Source used for illustrating SPARQL basics

3.1 SPARQL "SELECT" Query

The SELECT query will result in variables binding results. The SELECT query is mainly used to find values of declared variables that matches RDF graphs in WHERE clause. An example of SELECT query will be illustrated in Fig. 5 and its result will be illustrated in Fig. 6 which can be serialized into XML format as illustrates in Fig. 7.

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>

SELECT   ?myTitle

WHERE
{
   ?anyBook     dc:title     ?myTitle
}
```

Fig. 5 SPARQL "SELECT" Query Example

myTitle
"Harry Potter and the Deadly Hallows"
"The Lord of the Ring"

Fig. 6 "SELECT" Query's result

```
<?xml version="1.0"?>
<sparql xmlns="http://www.w3c.org/2005/sparql-results#">
    <head>
        <variable name="myTitle"/>
    </head>
    <results>
        <result>
            <binding name="myTitle">
                <literal>Harry Potter and the Deadly Hollows</literal>
            </binding>
        </result>
        <result>
            <binding name="myTitle">
                <literal>The Lord of the Ring</literal>
            </binding>
        </result>
    </results>
</sparql>
```

Fig. 7 "SELECT" Query result in XML Format

3.2 SPARQL "CONSTRUCT" Query

The CONSTRUCT query will result in RDF graphs. The CONSTRUCT query is mainly use to build graph based on a template which is used to generate RDF triples based on the results of matching the graph pattern in WHERE clause. An example of CONSTRUCT query will be illustrated in Fig.8 and its result will be illustrated in Fig. 9 which can also be serialized into XML format as illustrates in Fig. 10.

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>

CONSTRUCT ?myAuther dc:write ?myTitle

WHERE

{

    ?myBook      dc:title      ?myTitle

    ?myBook      dc:auther     ?myAuther

}
```

Fig. 8 SPARQL "CONSTRUCT" Query Example

@prefix dc: <http://purl.org/dc/elements/1.1/>

@prefix writer: <http://example.org/writer/>

writer:ID001	dc:write	"Harry Potter and the Deadly Hallows"
writer:ID002	dc:write	"The Lord of the Ring"

Fig. 9 SPARQL "CONSTRUCT" Query's result

```
<rdf:RDF xmlns:rdf=
"http://www.w3c.org/1999/02/22-rdf-syntax-ns#"
        xmlns:dc=
"http://purl.org/dc/elements/1.1/"
        xmlns:writer=
"http://example.org/writer/">
    <rdf:Description rdf:about=
"http://example.org/writer/ID001">
        <dc:write>Harry Potter and the Deadly
Hollows</dc:write>
    </rdf:Description>
    <rdf:Description rdf:about=
"http://example.org/writer/ID002">
        <dc:write>The Lord of the
Ring</dc:write>
    </rdf:Description>
</rdf:RDF>
```

Fig. 10 "CONSTRUCT" Query result in XML Format

3.3 SPARQL "ASK" Query

The ASK query will result in a boolean value which can also be serialized into XML format. The main purpose of this type of query is to test wheter or not a graph that match a specified graph pattern exists. An example of ASK query will be illustrated in Fig.11 and its result in XML format will be illustrated in Fig. 12

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>

ASK { ?anyBook  dc:title  "The Lord of the Ring" }
```

Fig. 11 SPARQL "ASK" Query Example

```
<?xml version="1.0"?>
<sparql xmlns="http://www.w3c.org/2005/sparql-results#">
    <head></head>
    <results>
        <boolean>true</boolean>
    </results>
</sparql>
```

Fig. 12 "ASK" Query result in XML Format

4 SPARQL Protocol for RDF

The SPARQL Protocol for RDF uses Web Service Description Language 2.0 (WSDL2) to describe how to convey SPARQL queries to a SPARQL query processing service and returning the query results to the entity that requested them. SPARQL Protocol can be described in two ways: first, as an abstract interface independent of any concrete realization, implementation, or binding to another protocol; second, as HTTP and SOAP bindings of this interface.

The proposed system architecture will implemented this protocol using HTTP Binding technique. SPARQL Protocol contains one interface, SparqlQuery, which in turn contains one operation, query. SPARQL Protocol is described abstractly with WSDL2 in terms of a web service that implements its interface, types, faults, and operations, as well as by HTTP bindings [11].

5 Frames

Frame was introduced by Marvin Minsky in 1974 [4]. It is a knowledge representation which has both data structure and inference capability. Frame is suitable for representing knowledge with concepts and classifications or a taxonomy hierarchy [5], [6].

A frame comprises a frame name, slots or attributes of the frame and facets [7], [8]. For clarification, frames can be classified into class frames and instance frames. Class frames are used to describe groups of objects or classes 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 [2], [6]-[10].

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 facet as well [2].

With the RDF concept of describing things by identifying its properties and properties' value, frames can simply be represented on the World Wide Web by RDF/XML.

6 System Architecture

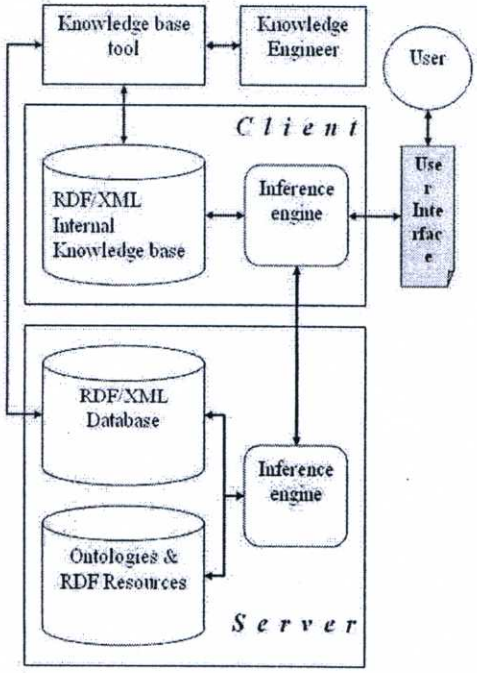


Fig.13 FRDFX System Architecture

Fig.13 shows the architecture of the FRDFX (Frame-based Resource Description Framework Expert System). This architecture allows the knowledge base side (server side) to represent their expertise knowledge with frames and simply share them on the World Wide Web as desired. Moreover, both of the knowledge base side and the expert system side (the client side) has their own inference engine to avoid a lot of communication traffic during inferences. The frames on the expert system side are those which involve user's interaction and acquire fact. Inference on this side can lead to a reference to the facts on the external knowledge base which can be inferred from other frames on server side [2].

The client-sided expert system comprises user interfaces for user and expert, the internal knowledge base and the client-sided inference engine. The external knowledge server comprises the server-sided inference engine, the RDF/XML external knowledge base and also ontologies and other RDF resources.

The main purpose of client-sided inference is to gather current information on the subject matter by gathering facts from the user during a consultation session and inference. 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-sided expert system gathers enough information which is to be sent to the knowledge base system on the server side for further inferences [2] using an implemented web service technology [14] including a SPARQL Protocol for RDF along with a SPARQL Query Language for RDF [11], [12].

On the other hand, server-sided inference mainly focuses on using existing facts which are already stored in the knowledge base and other relevant knowledge from ontologies and other RDF resources. The result of this server-sided inference will be sent back to the client-sided expert system and finally to the user [2].

7 Implementation

Frames on the knowledge base are kept in RDF/XML documents. Fig.14 shows RDF graph which representing frame's infrastructure.

From RDF graph representing Frame in Fig.14, we can create RDF/XML which represents a prototype frame as shown in Appendix C.

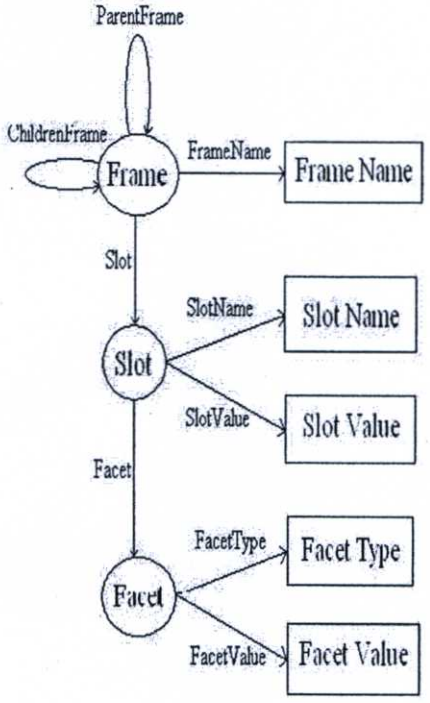


Fig.14 RDF graph representing Frame

8 Conclusions

This paper presents an alternative architecture of frame-based knowledge base and expert system with improved flexibility and mobility. Storing frames in RDF/XML format provides the capability of knowledge sharing on the World Wide Web which is, ideally, accessible from anywhere. The architecture uses web service technology implementing a SPARQL Protocol for RDF and SPARQL Query Language for RDF to communicate between servers and clients. For further research, RDF/XML knowledge base can be migrated to mobile server to improve even more flexibility and mobility in knowledge sharing.

References:

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

[2] Suphamit Chittayasothorn and Chuleerat Rattanaprateep, *Expert Database System Architecture and Implementation on Object Relational Databases*, WSEAS Transactions on Computers Issue 3, Vol.5, March 2006.

[3] *RDF Primer*, W3C Recommendation 10 February 2004

[4] Marvin Minsky, *A Framework for Representing Knowledge*, Reprinted in The Pschology of Computer Vision, P. Winston (Ed.), McGrawHill, 1975.

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

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

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

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

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

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

[11] *SPARQL Protocol for RDF*, W3C Recommendation 15 January 2008

[12] *SPARQL Query Language for RDF*, W3C Recommendation 15 January 2008

[13] Nutlada Rattanavijai and Suphamit Chittayasothorn, *Mobile Agents Based Distributed Database Statistics Collection*, WSEAS Transactions on Computers, Vol.6, June 2007.

[14] Saeed Araban and Leon Sterling, *Quality of service for web services*, WSEAS Transactions on Computers, Vol. 3, October 2004

Appendix A: RDF classes

Class name	comment
rdfs:Resource	The class resource, everything.
rdfs:Literal	The class of literal values, e.g. textual strings and integers.
rdf:XMLLiteral	The class of XML literals values.
rdfs:Class	The class of classes.
rdf:Property	The class of RDF properties.
rdfs:Datatype	The class of RDF datatypes.
rdf:Statement	The class of RDF statements.
rdf:Bag	The class of unordered containers.
rdf:Seq	The class of ordered containers.
rdf:Alt	The class of containers of alternatives.
rdfs:Container	The class of RDF containers.
rdfs:ContainerMembershipProperty	The class of container membership properties, rdf:_1, rdf:_2, ..., all of which are sub-properties of 'member'.
rdf:List	The class of RDF Lists.

Appendix B: RDF properties

Property name	comment	domain	range
rdf:type	The subject is an instance of a class.	rdfs:Resource	rdfs:Class
rdfs:subClassOf	The subject is a subclass of a class.	rdfs:Class	rdfs:Class
rdfs:subPropertyOf	The subject is a subproperty of a property.	rdf:Property	rdf:Property
rdfs:domain	A domain of the subject property.	rdf:Property	rdfs:Class
rdfs:range	A range of the subject property.	rdf:Property	rdfs:Class
rdfs:label	A human-readable name for the subject.	rdfs:Resource	rdfs:Literal
rdfs:comment	A description of the subject resource.	rdfs:Resource	rdfs:Literal
rdfs:member	A member of the subject resource.	rdfs:Resource	rdfs:Resource
rdf:first	The first item in the subject RDF list.	rdf:List	rdfs:Resource
rdf:rest	The rest of the subject RDF list after the first item.	rdf:List	rdf:List
rdfs:seeAlso	Further information about the subject resource.	rdfs:Resource	rdfs:Resource
rdfs:isDefinedBy	The definition of the subject resource.	rdfs:Resource	rdfs:Resource
rdf:value	Idiomatic property used for structured values.	rdfs:Resource	rdfs:Resource
rdf:subject	The subject of the subject RDF statement.	rdf:Statement	rdfs:Resource
rdf:predicate	The predicate of the subject RDF statement.	rdf:Statement	rdfs:Resource
rdf:object	The object of the subject RDF statement.	rdf:Statement	rdfs:Resource

Appendix C: RDF/XML Frame Representation

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
   <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
   <!ENTITY swrl "http://www.w3.org/2003/11/swrl#" >
   <!ENTITY swrlb "http://www.w3.org/2003/11/swrlb#" >
   <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
   <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
   <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
   <!ENTITY protege "http://protege.stanford.edu/plugins/owl/protege#" >
   <!ENTITY xsp "http://www.owl-ontologies.com/2005/08/07/xsp.owl#" >
]>
<rdf:RDF xmlns="http://www.owl-ontologies.com/Ontology1233987595.owl#"
   xml:base="http://www.owl-ontologies.com/Ontology1233987595.owl"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
   xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
   xmlns:swrl="http://www.w3.org/2003/11/swrl#"
   xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
   xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:owl="http://www.w3.org/2002/07/owl#">
   <owl:Ontology rdf:about="Frame"/>
   <owl:Class rdf:ID="Facets">
      <rdfs:subClassOf rdf:resource="&owl;Thing"/>
      <rdfs:subClassOf>
         <owl:Restriction>
            <owl:onProperty rdf:resource="#FacetValue"/>
            <owl:cardinality rdf:datatype="&xsd;int">1</owl:cardinality>
         </owl:Restriction>
      </rdfs:subClassOf>
   </owl:Class>
   <owl:DatatypeProperty rdf:ID="FacetType">
      <rdfs:domain rdf:resource="#Facets"/>
   </owl:DatatypeProperty>
   <owl:DatatypeProperty rdf:ID="FacetValue">
      <rdfs:domain rdf:resource="#Facets"/>
   </owl:DatatypeProperty>
   <owl:DatatypeProperty rdf:ID="FrameName">
      <rdfs:domain rdf:resource="#Frames"/>
      <rdfs:range rdf:resource="&xsd;string"/>
   </owl:DatatypeProperty>
   <owl:Class rdf:ID="Frames">
      <rdfs:subClassOf>
         <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
               <owl:Restriction>
                  <owl:onProperty rdf:resource="#hasChildren"/>
                  <owl:someValuesFrom rdf:resource="#Frames"/>
               </owl:Restriction>
               <owl:Restriction>
                  <owl:onProperty rdf:resource="#hasParent"/>
                  <owl:someValuesFrom rdf:resource="#Frames"/>
```

```
            </owl:Restriction>
            <owl:Restriction>
              <owl:onProperty rdf:resource="#hasSlot"/>
              <owl:someValuesFrom rdf:resource="#Slots"/>
            </owl:Restriction>
          </owl:unionOf>
        </owl:Class>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:onProperty rdf:resource="#FrameName"/>
          <owl:cardinality rdf:datatype="&xsd;int">1</owl:cardinality>
        </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf rdf:resource="&owl;Thing"/>
    </owl:Class>
    <owl:ObjectProperty rdf:ID="hasChildren">
      <rdfs:domain rdf:resource="#Frames"/>
      <owl:inverseOf rdf:resource="#hasParent"/>
      <rdfs:range rdf:resource="#Frames"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:ID="hasFacet">
      <rdfs:domain rdf:resource="#Slots"/>
      <owl:inverseOf rdf:resource="#isFacetOf"/>
      <rdfs:range rdf:resource="#Facets"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:ID="hasParent">
      <rdfs:domain rdf:resource="#Frames"/>
      <owl:inverseOf rdf:resource="#hasChildren"/>
      <rdfs:range rdf:resource="#Frames"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:ID="hasSlot">
      <rdfs:domain rdf:resource="#Frames"/>
      <owl:inverseOf rdf:resource="#isInFrame"/>
      <rdfs:range rdf:resource="#Slots"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:ID="isFacetOf">
      <rdfs:domain rdf:resource="#Facets"/>
      <owl:inverseOf rdf:resource="#hasFacet"/>
      <rdfs:range rdf:resource="#Slots"/>
    </owl:ObjectProperty>
    <owl:ObjectProperty rdf:ID="isInFrame">
      <rdfs:domain rdf:resource="#Slots"/>
      <owl:inverseOf rdf:resource="#hasSlot"/>
      <rdfs:range rdf:resource="#Frames"/>
    </owl:ObjectProperty>
    <owl:DatatypeProperty rdf:ID="SlotName">
      <rdfs:domain rdf:resource="#Slots"/>
      <rdfs:range rdf:resource="&xsd;string"/>
    </owl:DatatypeProperty>
    <owl:Class rdf:ID="Slots">
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:onProperty rdf:resource="#SlotValue"/>
          <owl:maxCardinality
```

```
rdf:datatype="&xsd;int">1</owl:maxCardinality>
        </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:onProperty rdf:resource="#SlotName"/>
          <owl:cardinality rdf:datatype="&xsd;int">1</owl:cardinality>
        </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf
rdf:resource="&owl;Thing"/>
    </owl:Class>
    <owl:DatatypeProperty rdf:ID="SlotValue">
      <rdfs:domain rdf:resource="#Slots"/>
    </owl:DatatypeProperty>
</rdf:RDF>
```

ประวัติผู้เขียน

ชื่อ-นามสกุล	นายนภัทธ์ ประภากร
วัน เดือน ปีเกิด	วันที่ 25 กรกฎาคม พ.ศ. 2525 ที่ จังหวัดกรุงเทพมหานครฯ
ที่อยู่	12/60 ซอยอินทามระ 33 แยก 1 ถนน สุทธิสารฯ
	แขวงดินแดน เขตดินแดง กรุงเทพฯ 10320
	โทร 0-2691-9930
ประวัติการศึกษา	2546 ปริญญาตรีวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมระบบควบคุม
	สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง
ความชำนาญเฉพาะด้าน	1.) ระบบการจัดการฐานข้อมูล
	2.) ระบบผู้เชี่ยวชาญ
ประสบการณ์การทำงาน	
พ.ศ. 2548-2549	โปรแกรมเมอร์ บริษัท Promt Now
พ.ศ. 2550-2551	วิศวกร แผนก IT บมจ. โทเทิล แอคเซส คอมมิวนิเคชัน