สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เปลือกระบบผู้เชี่ยวชาญ บนฐานข้อมูลเชิงวัตถุ

An Expert System Shell based-on Object-Oriented Database

T042773

นายกอบโชค งามรัตนไพบูลย์
นายเกียรติศักดิ์ จิตทรงบุญ

ว363.1
ก343ป
2543
เลขหมู่.....
เลขทะเบียน 42773
วัน, เดือน, ปี 10 มิ.ย. 2545
.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปลือกระบบผู้เชี่ยวชาญ บนฐานข้อมูลเชิงวัตถุ

An Expert System Shell based-on Object-Oriented Database

โดย

นายกอบโชค งามรัตนไพบูลย์

นายเกียรติศักดิ์ จิตทรงบุญ

อาจารย์ที่ปรึกษา

รศ.ดร.ศุภมิตร จิตตะยโศธร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2543

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เปลือกระบบผู้เชี่ยวชาญ บนฐานข้อมูลเชิงวัตถุ

An Expert System Shell based-on Object-Oriented Database

ผู้จัดทำ

1. นายกอบโชค งามรัตนไพบูลย์ รหัสประจำตัว 40010027
2. นายเกียรติศักดิ์ จิตทรงบุญ รหัสประจำตัว 40010081

อาจารย์ที่ปรึกษา

(รศ.ดร.ศุภมิตร จิตตะยโศธร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปลือกระบบผู้เชี่ยวชาญ บนฐานข้อมูลเชิงวัตถุ

นายกอบโชค งามรัตนไพบูลย์ 40010027
นายเกียรติศักดิ์ จิตทรงบุญ 40010081
ปีการศึกษา 2543

บทคัดย่อ

จากการพัฒนาของเทคโนโลยีทางด้านระบบฐานข้อมูลและแนวคิดในแบบเชิงวัตถุได้มีความก้าวหน้ามากขึ้นเรื่อยๆ จนในปัจจุบันได้เริ่มมีผลิตภัณฑ์ในเชิงพาณิชย์ออกมาบ้างแล้ว และในอีกด้านหนึ่ง การวิจัยทางด้านระบบผู้เชี่ยวชาญก็ได้มีการนำเอาแนวคิดในแบบเชิงวัตถุมาพัฒนา อีกทั้งยังได้เริ่มมีการพัฒนาที่รวมเอาระบบทั้งสองข้างต้นนี้เข้าไว้ด้วยกัน ทำให้โครงงานนี้เล็งเห็นประโยชน์ที่จะนำเอาเทคโนโลยีแนวคิดในแบบเชิงวัตถุของทั้งสองด้านนี้มาประยุกต์เข้าด้วยกัน โดยให้ระบบผู้เชี่ยวชาญใช้ประโยชน์จากฐานข้อมูลที่มีอยู่ ซึ่งอาจเป็นข้อมูลที่เป็นแบบชนิดข้อเท็จจริงหรือกฎก็ได้ เพื่อให้การใช้งานระบบผู้เชี่ยวชาญเป็นไปอย่างมีประสิทธิภาพมากขึ้น

และในอีกส่วนหนึ่งของโครงงานนี้ที่ได้นำเสนอก็คือ ได้มีการเอาเทคโนโลยีในแบบเชิงวัตถุ มาประยุกต์ใช้ในการเชื่อมต่อระหว่างระบบผู้เชี่ยวชาญและระบบฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

An Expert System Shell based-on Object-Oriented Database

Kobchok Ngamratanapaibool

Kiattisak Chitsongboon

Assoc. Prof. Dr. Suphamit Chitayasothorn Advisor

ABSTRACT

From the result of evolutions in Object-Oriented concepts and database system technology. In present, there are some commercial Object-Oriented database management system's products that have been arisen. On the other hand, an expert system research use object oriented technology to develop part of knowledge base call frame base. This project is inspired by the idea of combining these technologies and utilizing them in expert systems by let them use data that have already stored in the database, which may be facts, or rules. For the purpose to archive higher usability of expert systems.

This project also presents the approach and the solution for interfacing between Object-Oriented database and rule-based system which data representations are different.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้คงไม่สามารถสำเร็จสมบูรณ์ลงได้ หากไม่ได้รับการช่วยเหลือจากบุคคลหลายๆ ฝ่ายด้วยกัน ทั้งที่เกี่ยวข้องกับการทำงานของนักศึกษาผู้จัดทำโครงงานโดยตรง และทางอ้อม ทางนักศึกษาผู้จัดทำโครงงานขอขอบพระคุณบุคคลต่างๆ ต่อไปนี้ :

- รศ.ดร.ศุภมิตร จิตตะยโศธร ที่รับเป็นอาจารย์ที่ปรึกษาของวิชาโครงงาน ตลอดจนช่วยดูแลเอาใจใส่ แนะนำและให้คำปรึกษาเกี่ยวกับแนวทางการทำงานของโครงงานมาโดยตลอด เวลาคิดอะไรไม่ออก ไปหาอาจารย์เพื่อขอคำปรึกษา ท่านก็ทำให้เราก็คิดออกได้อย่างง่ายดาย รวมถึงคำแนะนำส่วนตัว ขอขอบพระคุณท่านอาจารย์เป็นอย่างสูง
- อาจารย์บัณฑิต พัสยา สำหรับความช่วยเหลือ และคำแนะนำในเรื่องการใช้ซอฟแวร์ Tools ต่าง ๆ และแหล่งข้อมูลเพิ่มเติมโดยเฉพาะเนื้อหาของโครงงาน
- บิดา มารดา ผู้บังเกิดเกล้า ที่ชุบเลี้ยง สั่งสอน และดูแลจนทำให้มีวันนี้
- เจ้าหน้าที่ และบุคลากรที่เกี่ยวข้องในส่วนของห้องสมุดทั้งใน ส่วนของห้องสมุดคณะวิศวกรรมศาสตร์ สำนักหอสมุดกลาง ของสถาบันที่ทำให้นักศึกษาผู้จัดทำโครงงานรับรู้ว่ายังมีอะไรๆ ในห้องสมุดของสถาบันตัวเองให้ใช้ประโยชน์อีกมาก และยังช่วยอำนวยความสะดวกในการยืมหนังสือระหว่างห้องสมุดอีกด้วย
- เจ้าหน้าที่ห้องสมุดสถาบันเทคโนโลยีแห่งเอเชีย (AIT) ที่เปิดให้บุคคลที่แม้จะเป็นนักศึกษาของสถาบันการศึกษาอื่นๆ เข้าใช้บริการได้โดยไม่เสียค่าธรรมเนียมใด ๆ
- พี่ปุ้ม ที่ช่วยอำนวยความสะดวกในการช่วยจัดตารางเวลา ให้เราได้มีโอกาสในการเข้าพบอาจารย์ที่ปรึกษา
- เพื่อน ๆ ทุกๆ คนทั้งที่ทำงานอยู่ในห้องวิจัยเดียวกัน, ภาควิชาเดียวกัน และคนละภาควิชา รวมถึงรุ่นน้องที่คอยถามไถ่ถึงความคืบหน้าของโครงงานที่ทำอยู่เสมอ ๆ
- คนอื่นๆ ที่ยังเหลืออีกทุกๆ คนที่มีส่วนเกี่ยวข้องด้วย แต่ยังไม่ได้อ้างอิงถึงในส่วนข้างต้น ต้องขออภัยไว้ ณ ที่นี้ด้วย

กอบโชค งามรัตนไพบูลย์
เกียรติศักดิ์ จิตทรงบุญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทคัดย่อภาษาไทย		I
บทคัดย่อภาษาอังกฤษ		II
กิตติกรรมประกาศ		III
สารบัญ		IV
สารบัญภาพ		XII
สารบัญตาราง		XIII
บทที่	1 บทนำ	1
	1.1 ความสำคัญและที่มา	1
	1.2 วัตถุประสงค์ของโครงงาน	2
	1.3 ขอบเขตของโครงงาน	2
	1.4 วิธีดำเนินงาน	2
บทที่	2 ระบบฐานข้อมูลแบบต่าง ๆ	3
	2.1 บทนำ	3
	2.2 ระบบฐานข้อมูลอนุมาน	4
	2.2.1 โครงสร้างของระบบฐานข้อมูลอนุมาน	4
	2.2.2 รูปแบบของสถาปัตยกรรม	4
	2.3 ระบบฐานข้อมูลเชิงวัตถุ	5
	2.3.1 Object Identity	5
	2.3.2 Complex Value	6
	2.3.3 Method	6
	2.3.4 Class	6
	2.3.5 การสืบทอดคุณสมบัติ	6
	2.3.6 Polymorphism	8
	2.4 ระบบฐานข้อมูลอนุมานเชิงวัตถุ	9
บทที่	3 ระบบผู้เชี่ยวชาญ	12
	3.1 บทนำ	12
	3.2 โครงสร้างพื้นฐานของระบบผู้เชี่ยวชาญ	12
	3.3 รูปแบบการแทนความรู้ชนิดต่าง ๆ ที่ใช้ในระบบผู้เชี่ยวชาญ	14
	3.3.1 แบบของกฎ	14
	3.3.2 แบบใช้ข่ายความหมาย	15
	3.3.3 แบบใช้กรอบและแบบเชิงวัตถุ	15
	3.4 ข้อจำกัดของระบบผู้เชี่ยวชาญ	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การติดต่อของระบบผู้เชี่ยวชาญกับโปรแกรมอื่น ๆ 17
3.6 ปัญหาที่พบในกรณีที่ข้อมูลที่ระบบผู้เชี่ยวชาญเรียกใช้มีขนาดมาก ๆ 18
3.7 ตัวอย่างระบบผู้เชี่ยวชาญที่มีการติดต่อกับข้อมูลภายนอก 18
3.7.1 BERMUDA 18
3.7.2 Tanguy 19
3.7.3 ระบบฐานความรู้ในระบบสารสนเทศทางการแพทย์ใน 19
3.7.4 CLUES 20
3.7.5 GermWatcher 20
บทที่ 4 ระบบจัดการฐานข้อมูลเชิงวัตถุ Caché 21
4.1 คุณสมบัติด้านต่าง ๆ ของ Caché 21
4.2 องค์ประกอบของ Caché 22
4.3 Object Model 22
4.4 Class 23
4.4.1 Registered Class 24
4.4.2 Persistent Class 24
4.4.3 Embeddable Class 25
4.4.4 Non-Registered Class 26
4.4.5 Data Type Class 26
4.5 Caché กับ Inheritance 26
4.6 Caché กับ Multiple Inheritance 27
4.7 Caché กับ Relationship 27
4.8 Caché กับ Application 29
บทที่ 5 สถาปัตยกรรมของระบบ 30
5.1 บทนำ 30
5.2 รูปแบบของสถาปัตยกรรมแบบต่าง ๆ 31
5.2.1 สถาปัตยกรรมแบบแยกส่วน 31
5.2.1.1 แนวทางการใช้แบบ Interpreter 31
5.2.1.2 แนวทางการใช้แบบ Compiler 31
5.2.2 สถาปัตยกรรมแบบรวมเป็นส่วนเดียว 31
5.2.2.1 แบบทำการขยายส่วนของระบบผู้เชี่ยวชาญให้มีความสามารถของระบบฐานข้อมูล 32
5.2.2.2 แบบทำการขยายส่วนของระบบฐานข้อมูลให้มีความสามารถในการอนุมาน 32
บทที่ 6 สถาปัตยกรรมและการออกแบบในโครงงาน 33
6.1 บทนำ 33
6.2 ส่วนของระบบผู้เชี่ยวชาญ 33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.1 โครงสร้างของฐานความรู้ 33
6.2.2 ข้อเท็จจริง 35
6.2.3 กฎ 36
6.2.4 กลไกการอนุมาน 36
6.3 การอ้างอิงถึงฐานความรู้ 37
6.3.1 Frame 37
6.3.1.1 การอ้างอิงคลาส Persistent 37
6.3.1.2 การอ้างอิงคลาส Embeddable 38
6.3.2 ข้อเท็จจริง 38
6.4 กลไกการอนุมาน 38
6.4.1 ลักษณะของกฎ 38
6.4.1.1 If 39
6.4.1.2 Then 39
6.4.1.3 FramePointer 40
6.4.2 กลไกในการอนุมาน 40
6.5 ส่วนของตัวกลางในการติดต่อกับ Caché 41

บทที่ 7 การทดสอบระบบที่ได้ทำการสร้าง 42
7.1 บทนำ 42
7.2 ระบบผู้เชี่ยวชาญที่ทำการออกแบบและสร้างขึ้นในการทดสอบ 42
7.3 ตัวอย่างความรู้เรื่อง “ตากล้องมือใหม่” 43
7.3.1 การสร้างฐานความรู้เรื่อง “ตากล้องมือใหม่” 46
7.3.2 ข้อเท็จจริงเรื่อง “ตากล้องมือใหม่” 48
บทที่ 8 สรุปผล, วิจารณ์ และแนวทางการพัฒนาต่อ 49
8.1 สรุปผลงานวิจัย 49
8.2 วิจารณ์งานวิจัย 50
8.3 แนวทางการพัฒนาต่อ 51

บรรณานุกรม 52
ภาคผนวก ก คู่มือการใช้งาน 54
ภาคผนวก ข คู่มือสำหรับโปรแกรมเมอร์ 72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่ 2-1 การสืบทอดคุณสมบัติแบบลำดับชั้น 7
รูปที่ 3-1 โครงสร้างพื้นฐานของระบบผู้เชี่ยวชาญ 13
รูปที่ 3-2 ตัวอย่างการแทนความรู้ด้วยข่ายความหมาย 15
รูปที่ 3-3 การแทนความรู้ด้วยกรอบ 16
รูปที่ 4-1 การติดต่อกับระบบจัดการฐานข้อมูล Caché 21
รูปที่ 4-2 OREF และ OID 23
รูปที่ 4-3 ประเภทของ Class 24
รูปที่ 4-4 Embedded Object ในหน่วยความจำและดิสก์ 25
รูปที่ 4-5 ความสัมพันธ์ระหว่างคลาส Company และคลาส Employee 27
รูปที่ 6-1 สถาปัตยกรรมและขั้นตอนการติดต่อระหว่างระบบย่อย 33
รูปที่ 6-2 โครงสร้างของ Frame 34
รูปที่ 6-3 แสดงตัวอย่างของข้อเท็จจริงในระบบผู้เชี่ยวชาญ 35
รูปที่ 6-4 แสดงรูปแบบการเชื่อมต่อเข้าใช้งาน Caché 36
รูปที่ 6-5 แสดงกลไกในการอนุมานแบบเดินหน้า 41
รูปที่ 7-1 แสดงขั้นตอนในการกำหนดรายละเอียดของฐานความรู้ 46
รูปที่ 7-2 แสดงขั้นตอนในการกำหนดรายละเอียดของ Frame 46
รูปที่ 7-3 แสดงขั้นตอนในการกำหนดรายละเอียดของ Slots 47
รูปที่ 7-4 แสดงขั้นตอนในการกำหนดรายละเอียดของ Rules 47
รูปที่ 7-3 แสดงขั้นตอนการป้อนข้อเท็จจริง 48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่4-1 แสดงการเข้าถึงข้อมูลของ Caché 22
ตารางที่ 4-2 แสดงส่วนประกอบของ Caché 22
ตารางที่ 7-1 แสดงความรู้เรื่อง “ตากล้องมือใหม่” 43
ตารางที่ 7-2 แสดงรายละเอียดของฐานความรู้เรื่อง “ตากล้องมือใหม่” 43
ตารางที่ 7-3 แสดงข้อเท็จจริงเรื่อง “ตากล้องมือใหม่” 45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

จากการพัฒนาทางด้านเทคโนโลยีทางด้านฐานข้อมูลที่ได้พัฒนาขึ้นมาเรื่อยๆ จนในปัจจุบันได้มีผลิตภัณฑ์ในเชิงพาณิชย์ของระบบจัดการฐานข้อมูล (Database Management System: DBMS) เกิดขึ้นและใช้กันอย่างแพร่หลาย ซึ่งระบบจัดการฐานข้อมูลเหล่านี้ได้แบ่งเบาภาระหน้าที่ของผู้ที่มีหน้าที่ในการพัฒนาระบบสารสนเทศอย่างมีประสิทธิภาพ ตัวอย่างของความสามารถของระบบจัดการข้อมูลดังกล่าวนี้ ก็ได้แก่ ความสามารถในการจัดการเรื่องความซ้ำซ้อนของข้อมูล (Redundancy), ความสามารถในการดูแลความสอดคล้องกันของข้อมูล (Consistency), ความสามารถในการจัดการระเบียบข้อบังคับที่ใช้ในการจัดเก็บข้อมูล (Integrity), ความสามารถในการจัดการเรื่องของการจัดสรรข้อมูลให้สามารถใช้งานโดยโปรแกรมหลายๆ ตัวในเวลาเดียวกันได้ (Concurrency), และความสามารถในการจัดการเรื่องมาตรการด้านความปลอดภัยในการใช้ข้อมูลที่อยู่ในฐานข้อมูล (Security)

ไม่เพียงแต่ในเรื่องดังข้างต้นนี้เท่านั้นที่ได้มีการพัฒนา ในส่วนของแนวทางที่ใช้ในการจัดเก็บข้อมูลในฐานข้อมูลก็มีการพัฒนาและคิดค้นขึ้นมาอยู่เรื่อยๆ เช่นกัน ตัวอย่างหนึ่งของแนวทางเหล่านี้ได้แก่ แนวทางที่เป็นการนำเอาแนวความคิดเชิงวัตถุ (Object-Oriented Concept) ซึ่งมีจุดกำเนิดมาจากในส่วนของวิธีการพัฒนาและเขียนโปรแกรมก่อน ต่อมาก็มีการนำมาประยุกต์ใช้ในเรื่องของปัญญาประดิษฐ์ (Artificial Intelligence) จนในที่สุดก็เริ่มมีการนำมาใช้กับระบบฐานข้อมูล ซึ่งทำให้ระบบจัดการฐานข้อมูลที่ใช้วิธีการเก็บข้อมูลแบบนี้สามารถจัดการกับข้อมูลที่มีลักษณะที่ซับซ้อนได้ดีกว่าแบบเดิมซึ่งเป็นแบบ Relational สำหรับอีกแนวทางหนึ่งก็คือการนำเอาแนวทางที่ใช้ในระบบผู้เชี่ยวชาญเพื่อทำให้ระบบฐานข้อมูลมีความสามารถในการเก็บข้อมูลที่อยู่ในรูปแบบของกฎ (Rule) ได้ เป็นผลให้ระบบฐานข้อมูลชนิดนี้มีความสามารถในการอนุมาน (Deduction) ข้อมูลที่เป็นข้อเท็จจริงใหม่ๆ ได้ โดยไม่จำเป็นต้องมีการเก็บข้อมูลตัวนั้นจริงในฐานข้อมูลซึ่งในแนวทางนี้ก็ได้มีการพัฒนาขึ้นมาจนในปัจจุบันได้มีระบบที่เป็นต้นแบบอยู่มากมาย

นอกจากสองแนวทางหนึ่งดังข้างต้นแล้ว อีกแนวทางหนึ่งซึ่งเป็นแนวทางที่พยายามพัฒนาระบบฐานข้อมูลที่รวมเอาแนวทางทั้งสองเข้าไว้ด้วยกันก็มีการพัฒนาขึ้นด้วยเช่นกัน และก็ได้มีระบบต้นแบบให้เห็นกันแล้วด้วย ซึ่งสำหรับในโครงงานนี้จะนำรูปแบบของระบบจัดการฐานข้อมูลดังกล่าวมาประยุกต์ใช้ในระบบผู้เชี่ยวชาญ โดยมีวัตถุประสงค์เพื่อพยายามพัฒนาระบบสถาปัตยกรรมของระบบผู้เชี่ยวชาญที่มีประสิทธิภาพและความสามารถในการนำไปประยุกต์ในการใช้งานจริงได้อย่างกว้างขวางขึ้น โดยพัฒนาส่วนของระบบผู้เชี่ยวชาญให้สามารถใช้งานข้อมูล ข้อมูลที่อยู่ในฐานข้อมูลซึ่งในโครงงานนี้ได้ทำการพัฒนาขึ้นให้เป็นแบบเชิงวัตถุที่มีความสามารถในการอนุมานได้ด้วย เป็นการขยายความสามารถของระบบผู้เชี่ยวชาญซึ่งจากเดิมที่สามารถเก็บข้อมูลได้อย่างจำกัดและทำงานได้เพียงแค่ในหน่วยความจำหลักเท่านั้น อีกทั้งยังขาดความสามารถในการเก็บข้อมูลต่างๆ ไว้อย่างถาวร ทำให้ในการใช้งานระบบผู้เชี่ยวชาญบางครั้งค่อนข้างซ้ำซากและต้องทำการถามผู้ใช้ตามขั้นตอนต่างๆ ที่ซ้ำๆ กันซึ่งจริงๆ แล้วเป็นสิ่งที่ไม่จำเป็นเลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการพัฒนาระบบของโครงงานนี้จะเป็นการสร้างเปลือกระบบผู้เชี่ยวชาญ ซึ่งประยุกต์ด้วย Object Oriented Technology ให้มีสามารถติดต่อกับระบบฐานข้อมูล(Object Oriented Database)

1.2 วัตถุประสงค์ของโครงงาน

1.2.1. ศึกษาถึงทฤษฎี หลักการ และแนวทางการใช้งานฐานข้อมูลเชิงวัตถุ โดยเน้นการศึกษาไปที่ผลิตภัณฑ์ Caché ซึ่งเป็นโปรแกรมระบบจัดการฐานข้อมูลเชิงวัตถุ

1.2.2. ศึกษาถึงรูปแบบสถาปัตยกรรมที่เป็นการนำเอาระบบฐานข้อมูลมาประยุกต์ใช้ประโยชน์ในการทำงานของระบบผู้เชี่ยวชาญ และพยายามนำเสนอรูปแบบสถาปัตยกรรมแบบ Frame Based ที่สามารถลดปัญหาหรือข้อจำกัดที่มีอยู่ในระบบได้

1.2.3 ศึกษาถึงความเป็นไปได้ในการสร้างเปลือกระบบผู้เชี่ยวชาญให้สามารถเก็บอยู่ในระบบจัดการฐานข้อมูลเชิงวัตถุได้ โดยพยายามใช้ประโยชน์จากคุณสมบัติที่มีอยู่ของระบบฐานข้อมูลเชิงวัตถุ

1.2.4. นำเสนอวิธีการที่ใช้ในการแก้ปัญหาที่เกิดขึ้นการติดต่อกันระหว่างระบบผู้เชี่ยวชาญกับระบบฐานข้อมูลเชิงวัตถุซึ่งมีรูปแบบข้อมูลที่แตกต่างกัน

1.3 ขอบเขตของโครงงาน

โครงงานจะเน้นในด้านการศึกษาและออกแบบรูปแบบเปลือกระบบผู้เชี่ยวชาญ(Expert System Shell) ด้วย Object Oriented Technology สถาปัตยกรรมที่ใช้ในการติดต่อกันระหว่างระบบฐานข้อมูลและระบบผู้เชี่ยวชาญ โดยระบบจัดการฐานข้อมูลที่ทางโครงงานนี้ได้นำมาใช้จะเป็นระบบจัดการฐานข้อมูลเชิงวัตถุ คือ Caché และจะไม่เน้นในส่วนของการพัฒนาระบบเพื่อนำไปใช้งานจริง แต่จะทำการสร้างเพียงแค่ระบบต้นแบบเพื่อทดสอบว่าระบบที่ได้ทำการศึกษาและทำการออกแบบสามารถใช้งานได้จริง

1.4 วิธีการดำเนินงาน

1.4.1 ศึกษาทฤษฎีพื้นฐานต่าง ๆ หลักการและรูปแบบของฐานข้อมูลเชิงวัตถุ

1.4.2 ศึกษาทฤษฎีและหลักการของระบบผู้เชี่ยวชาญต่าง ๆ

1.4.3 ศึกษาการใช้งานระบบจัดการฐานข้อมูลเชิงวัตถุ Caché

1.4.3 นำ Caché มาประยุกต์ใช้งานกับระบบผู้เชี่ยวชาญเพื่อศึกษาผลการทำงานและความเป็นไปได้ที่จะนำเอาหลักการและรูปแบบทางสถาปัตยกรรมของฐานข้อมูลเชิงวัตถุมาใช้งาน

1.4.4 ออกแบบและทำการทดลองเพื่อหาข้อสรุป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ระบบฐานข้อมูลแบบต่างๆ

2.1 บทนำ

จากการวิจัยและพัฒนาทางด้านเทคโนโลยีทางด้านระบบฐานข้อมูลแบบ Relational ที่ได้มีการพัฒนามาช้านาน จนเริ่มมีผลิตภัณฑ์ที่พัฒนาขึ้นโดยมีวัตถุประสงค์เพื่อการใช้งานในเชิงพาณิชย์ออกมากันอย่างแพร่หลาย และมีการพัฒนาต่อมาเรื่อยๆ อย่างสม่ำเสมอ จนในปัจจุบัน ระบบจัดการฐานข้อมูลแบบ Relational (Relational Database Management System : RDBMS) ได้มีการใช้งานกันและเป็นที่รู้จักกันอย่างแพร่หลาย และการพัฒนาเพิ่มเติมต่อมาเรื่อยๆ จนในปัจจุบัน ผลิตภัณฑ์ในส่วนของระบบการจัดการฐานข้อมูลเหล่านี้ได้มีความสามารถต่างๆ ที่เป็นการเอื้อประโยชน์ในการพัฒนาระบบสารสนเทศขนาดใหญ่ๆ มากมายหลายประการและมีความน่าเชื่อถือ (Reliability) ในการทำงานที่สูงด้วย ตัวอย่างของความสามารถของระบบจัดการฐานข้อมูลที่มีอยู่ในปัจจุบันก็ได้แก่ ความสามารถในการจัดการเรื่องของความซ้ำซ้อนของข้อมูล (Redundancy) ความสามารถในการหลีกเลี่ยงและจัดการในเรื่องของความขัดแย้งกันของข้อมูล (Consistency) ความสามารถในการเรียกใช้ข้อมูลและการทำงานร่วมกันของโปรแกรมประยุกต์ใช้งานหลายๆ โปรแกรมในเวลาเดียวกัน (Concurrency) ความสามารถในการจัดการเพื่อรักษาความถูกต้องของตัวข้อมูล (Integrity) และความสามารถในการจัดตั้งมาตรการในการใช้งานต่างๆ เพื่อเหตุผลทางด้านความปลอดภัย (Security)

แต่ข้อจำกัดอย่างหนึ่งของระบบจัดการฐานข้อมูลส่วนใหญ่ในปัจจุบันที่เป็นแบบ Relational นี้ ก็คือความสามารถในการจัดการข้อมูลที่มีลักษณะที่ซับซ้อนซึ่งทางตัวแบบจำลองข้อมูล (Data model) ที่ระบบใช้อยู่นั้นออกแบบมาเพื่อให้สามารถจัดการจัดข้อมูลที่มีลักษณะง่ายๆ ไม่ซับซ้อนซึ่งหากเราต้องทำการเก็บข้อมูลที่มีลักษณะที่ซับซ้อนขึ้นมาแล้ว ในส่วนของทางระบบฐานข้อมูลก็จะต้องใการออกแบบโครงสร้างการเก็บข้อมูลที่ซับซ้อนขึ้นตามไปด้วยและตัวแบบจำลองข้อมูลเองก็ไม่ได้ออกแบบมาให้มีโครงสร้างหรือกลไกใดๆ เพื่อรองรับหรือจัดการในด้านความซับซ้อนเหล่านี้โดยตรง ทำให้ทางผู้พัฒนาและผู้ที่มีหน้าที่ดูแลในส่วนของฐานข้อมูลต้องมาจัดการเอง ทำให้ก่อให้เกิดความผิดพลาดต่างๆ ได้ง่าย และรูปแบบการเก็บข้อมูลต่างๆ ก็ยังไม่สามารถเก็บข้อมูลที่อยู่ในรูปแบบของกฎ (Rule) ได้

จากข้อจำกัดต่างๆ ที่ได้อธิบายมาแล้วข้างต้นนี้ ทำให้แนวทางการวิจัยในด้านของระบบฐานข้อมูลในระยะเวลาต่อมา ได้มุ่งเน้นไปในส่วนที่ใช้ในการจัดการข้อมูลที่ซับซ้อน และความสามารถในการอนุมานกฎ (Deduction) ซึ่งจากแนวทางในการวิจัยเหล่านี้ ก็ได้เกิดแนวความคิดในเรื่องของระบบฐานข้อมูลที่เป็นแบบเชิงวัตถุ (Object-Oriented Database) และระบบฐานข้อมูลอนุมาน (Deductive Database) ซึ่งในเนื้อหาของบทนี้จะประกอบด้วยเนื้อหาในส่วนระบบฐานข้อมูลทั้งสอง โดยในหัวข้อต่อไป จะอธิบายถึงระบบฐานข้อมูลอนุมานก่อน จากนั้นในหัวข้อต่อๆ ไป จะอธิบายถึงแนวความคิดเชิงวัตถุและการมาประยุกต์ใช้กับระบบฐานข้อมูล และสำหรับในลำดับสุดท้ายที่จะอธิบายในบทนี้ก็คือ แนวความคิดของระบบฐานข้อมูลที่มีการนำเอาแนวคิดของระบบฐานข้อมูลทั้งสองอย่างข้างต้น คือระบบฐานข้อมูลเชิงวัตถุ และระบบฐานข้อมูลอนุมานมารวมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งระบบฐานข้อมูลที่ได้มาใหม่นี้ก็มีชื่อเรียกว่าระบบฐานข้อมูลอนุมานเชิงวัตถุ (Deductive Object-Oriented Database)

2.2 ระบบฐานข้อมูลอนุมาน

ระบบฐานข้อมูลอนุมานเป็นระบบที่พยายามขยายความสามารถของเทคโนโลยีทางด้านฐานข้อมูล โดยการพยายามสร้างระบบที่มีความสามารถทั้งในส่วนที่เป็นลักษณะของฐานความรู้ (knowledge-base) และส่วนที่เป็นลักษณะของระบบการจัดการฐานข้อมูล ทำให้เราได้ระบบฐานข้อมูลมีความสามารถทางด้านการอนุมานหาข้อเท็จจริง (Fact) ใหม่ๆ จากข้อเท็จจริงที่มีการเก็บไว้ในส่วนของฐานข้อมูลซึ่งอาจมีการเก็บข้อมูลเหล่านั้นแยกไว้ต่างหากโดยไม่เกี่ยวข้องกับส่วนที่ใช้ในการทำ inference ซึ่งจะหมายถึงกระบวนการที่ใช้ในการหาข้อเท็จจริง โดยมีหลักการมาจากทฤษฎีทางด้านตรรกศาสตร์ (mathematical logic) และระบบฐานข้อมูลแบบนี้ก็จะใช้ทฤษฎีตรรกศาสตร์มาประยุกต์ใช้ในส่วนของการเก็บข้อมูลต่างๆ ด้วย ข้อมูลที่เก็บเหล่านี้ก็ได้แก่ ข้อเท็จจริง, กฎ, ข้อบังคับของข้อมูล (constraint) ซึ่งก็จะมีลักษณะที่คล้ายคลึงกับแบบจำลองข้อมูลแบบ Relational ที่สร้างขึ้นมาจากทฤษฎีทางด้านคณิตศาสตร์ด้วย ทำให้เป็นเกิดเป็นแนวทางการพัฒนาที่พยายามสร้างฐานข้อมูลที่มีความสามารถในการอนุมานขึ้นดังที่ได้อธิบายไปแล้ว

2.2.1 โครงสร้างของระบบฐานข้อมูลอนุมาน

ระบบฐานข้อมูลอนุมานสามารถพิจารณาแยกออกเป็น 2 ส่วน คือ

- intensional database ประกอบด้วย ส่วนของฐานความรู้ที่จะเก็บในส่วนของกฎทั่วไป (general rule) การอ้างสิทธิ รูปแบบลักษณะของข้อมูล และฐานความรู้ผู้เชี่ยวชาญ (expert knowledge) สำหรับในส่วนของฐานข้อมูลจะประกอบด้วยรูปแบบโครงสร้างของข้อมูล ข้อบังคับที่เกี่ยวข้องกับความถูกต้องของข้อมูล (integrity constraint) และรูปแบบของ view ที่จะเก็บไว้ในลักษณะต่างๆ ซึ่งจะเห็นได้ว่าสถานะของระบบในส่วนนี้จะไม่ค่อยเปลี่ยนแปลงมากนัก และค่อนข้างมีเสถียรภาพ
- extensional database คือ ส่วนของฐานข้อมูลที่เก็บข้อมูลในส่วนที่เราสนใจของระบบในปัจจุบัน ซึ่งจะมีการเปลี่ยนแปลงตลอดเวลา ข้อมูลต่างๆเหล่านี้ ได้แก่ ข้อมูลที่มาจากการเพิ่ม ลบ และแก้ไข จากตัวโปรแกรมที่เรียกใช้งานระบบฐานข้อมูล

2.2.2 รูปแบบสถาปัตยกรรม

ในการสร้างระบบฐานข้อมูลอนุมาน สิ่งแรกที่ต้องทำคือสร้างรูปแบบการเชื่อมต่อ (interface) ระหว่างระบบฐานข้อมูลกับภาษาที่ใช้ในการจัดการฐานข้อมูลซึ่งในที่นี้ก็คือ ภาษาประเภท logic programming โดยมีตัวประมวลผลทางภาษา (language processor) ทำหน้าที่เข้าถึงข้อมูลในส่วนที่เราต้องการ ซึ่งตัวประมวลผลทางภาษาอาจจะใช้วิธีการแปลงรูปแบบคำสั่งที่ส่งเข้ามาให้อยู่ในรูปแบบของภาษาที่ใช้จัดการฐานข้อมูล เช่น SQL ในกรณีที่สถาปัตยกรรมที่เราสร้างขึ้นมานั้นเป็นการสร้างระบบๆ หนึ่งขึ้นมาครอบระบบจัดการฐานข้อมูลที่มีอยู่เดิม เพื่อค้นหาข้อเท็จจริงนั้นในฐานข้อมูลต่อไป รูปแบบการแยกระบบจัดการฐานข้อมูลกับระบบผู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เชี่ยวชาญนี้เราเรียกว่า การรวมกันแบบหลวม (loosely coupled) ซึ่งสถาปัตยกรรมแบบนี้จะทำงานได้ดีถ้าทางฝั่งระบบผู้เชี่ยวชาญมีความต้องการข้อมูลจากฐานข้อมูลไม่บ่อยครั้งนัก แต่ถ้าหากทางฝั่งระบบผู้เชี่ยวชาญมีความต้องการข้อมูลจากฐานข้อมูลบ่อยจะทำให้การทำงานของระบบโดยรวมช้าลง

สำหรับรูปแบบสถาปัตยกรรมอีกแบบหนึ่งที่ได้มีการพัฒนาขึ้นเพื่อแก้ไขข้อเสียของรูปแบบสถาปัตยกรรมข้างต้น เป็นรูปแบบสถาปัตยกรรมที่ได้มีการสร้างให้กลไกของการเข้าถึงข้อมูลในระบบฐานข้อมูลกับกลไกของการทำ inference มีการทำงานร่วมกันอย่างมีประสิทธิภาพและเรียกสถาปัตยกรรมนี้ว่า การรวมกับอย่างเหนียวแน่น (tight coupling) โดยการชี้ดรรชนีและการเข้าถึงการทำงานย่อยของระบบฐานข้อมูลควรเป็นรูปแบบที่เป็นการเข้าถึงข้อมูลโดยตรง โดยไม่ต้องผ่านตัวกลางที่ทำหน้าที่ในการแปลงในการสืบค้นของระบบฐานข้อมูล อย่างไรก็ตามระบบจัดการฐานข้อมูลและระบบผู้เชี่ยวชาญก็ยังคงมีความเป็นอิสระอยู่ ยกเว้นการทำงานที่ประสานกันในระดับต่ำ

และเมื่อเข้าสู่การพัฒนาในรูปแบบต่อมา จะมีการรวมกันของระบบจนแยกไม่ออก นั่นคือระบบฐานข้อมูลและระบบการโปรแกรมเชิงตรรกไม่มีความเป็นอิสระต่อกัน ดังเช่น ภาษา logic-base ซึ่งทั้งการเชื่อมต่อและระบบต้องถูกออกแบบมาจากระบบฐานข้อมูลอนุมาน

2.3 ระบบฐานข้อมูลเชิงวัตถุ

แนวทางหนึ่งที่มีขึ้นเพื่อเป็นการขยายความสามารถเพื่อให้ระบบฐานข้อมูลจัดการข้อมูลที่มีลักษณะซับซ้อนได้ก็คือแนวทางการพัฒนาระบบฐานข้อมูลที่เป็นลักษณะในเชิงวัตถุ โดยสำหรับในแนวคิดเชิงวัตถุนี้มีต้นกำเนิดมาจากการเขียนโปรแกรมในเชิงวัตถุมาก่อน (Object-Oriented Programming) ซึ่งเมื่อนำมาประยุกต์ใช้กับระบบฐานข้อมูลแล้ว ก็มีการเปลี่ยนในส่วนของรายละเอียดปลีกย่อยเล็กน้อย แต่โดยรวมแล้วก็ยังมีลักษณะที่สำคัญๆ คล้ายคลึงกันอยู่ และยังมีความสามารถในส่วนของการทำงานของการจัดการด้านฐานข้อมูลที่ระบบจัดการฐานข้อมูลแบบ Relational มีเหมือนเดิม สำหรับลักษณะโดยรวมที่สำคัญๆ ของระบบฐานข้อมูลเชิงวัตถุสามารถอธิบายโดยสังเขปได้โดยแบ่งเป็นหัวข้อย่อยๆ ได้ดังนี้

2.3.1 Object Identity

สำหรับในระบบฐานข้อมูลที่เป็นแบบเน้นการแสดงค่า (Value-Oriented) จะใช้ค่าๆ หนึ่งที่อยู่ในฐานข้อมูลเป็นค่าหลัก (Key) ในการแทนถึงวัตถุใดวัตถุหนึ่ง ปัญหาที่พบเมื่อมีการแทนถึงวัตถุในลักษณะนี้ก็คือ ในกรณีที่ค่าหลักที่ใช้ในการแทนถึงวัตถุหรือสิ่งของใดๆ นั้นเป็นค่าที่มีการเปลี่ยนแปลงด้วย จะทำให้เกิดปัญหาของการแทนค่าถึงวัตถุตัวใดตัวหนึ่งเนื่องจากค่าที่ใช้ในการอ้างอิงนั้นเกิดการเปลี่ยนแปลงไปจากเดิม และต้องแก้ปัญหาด้วยการแก้ไขค่าหลักที่ใช้ทำการอ้างอิงนี้ทุกตัวเมื่อมีการเปลี่ยนแปลงเกิดขึ้นเพื่อให้การอ้างอิงถึงค่าของสิ่งใดสิ่งหนึ่งในฐานข้อมูลเป็นไปได้เหมือนเดิม

ในกรณีของระบบฐานข้อมูลเชิงวัตถุ จะแก้ปัญหานี้ด้วยการแทนค่าที่ใช้ในการอ้างอิงถึงวัตถุตัวนั้นเป็นค่าต่างหากค่าหนึ่ง ซึ่งจะเรียกค่านี้ว่า Object Identifier ลักษณะของค่านี้จะแตกต่างไปจากค่าที่เก็บอยู่ในฐานข้อมูลค่าอื่นๆ ก็คือ จะมีค่าที่ไม่ซ้ำกันเพื่อที่ระบบจัดการฐานข้อมูลใช้ค่าเหล่านี้ในการอ้างอิงถึง Object ตัวใดตัวหนึ่งที่เก็บอยู่ในฐานข้อมูลเป็นหลัก ซึ่งในอ้างอิงนี้เราอาจพบว่า จะมี Object สองตัวใดๆ ที่มีค่าเหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่มีการอ้างอิงโดยค่า Object Identifier คนละตัวก็ได้ ซึ่งค่านี้ ระบบฐานข้อมูลจะใช้ในการอ้างอิงถึง Object ตัวนั้นเป็นแบบถาวร คือ ค่าที่ใช้ในการอ้างอิงเหล่านี้จะถูกสร้างและทำลายโดยตัวระบบจัดการฐานข้อมูลเอง และจะไม่สามารถทำการเปลี่ยนแปลงค่าต่างๆ เหล่านี้ได้ ซึ่งจะแตกต่างจากค่าอื่นๆ ที่อยู่ในฐานข้อมูล ส่วนความสามารถที่ผู้ใช้ระบบฐานข้อมูลจะนำค่าเหล่านี้ไปใช้ในการอ้างอิงหรือใช้งานได้หรือไม่นั้น จะขึ้นอยู่กับระบบจัดการฐานข้อมูลที่ใช้ว่าจะเปิดโอกาสให้ผู้ใช้ได้เข้าถึงข้อมูลของค่าเหล่านี้หรือไม่ ซึ่งก็แตกต่างกันไปในแต่ละระบบ

2.3.2 Complex Value

สำหรับ Object แต่ละตัวจะมีค่าที่เกี่ยวข้องด้วยอยู่ค่าหนึ่ง ซึ่งอาจเป็นค่าที่มีลักษณะซับซ้อนก็ได้ ซึ่งค่านี้ก็เป็นค่าของ Object Identifier ก็ได้ เป็นผลให้สามารถมี Object สองตัวใดๆ ที่อ้างอิงถึง Object อีกตัวหนึ่งซึ่งเป็นตัวเดียวกันได้ โดยใช้ Object Identifier ที่มีค่าเหมือนกัน และเป็นผลให้เมื่อมีการเปลี่ยนแปลงข้อมูลในตัว Object นั้นๆ ก็จะไม่ส่งผลกระทบไปยัง Object ตัวอื่นๆ ที่มีการอ้างอิงถึง ซึ่งค่าที่เก็บอยู่ใน Object นี้สามารถเรียกได้อีกอย่างว่าเป็นสถานะ (State) ของ Object ตัวนั้นได้ด้วย

2.3.3 Method

Object ที่เก็บอยู่ในระบบฐานข้อมูลเชิงวัตถุนั้นจะมีวิธีการใช้งานโดยผ่านทาง Method นี้ โดย Method จะประกอบด้วย ชื่อ, รูปแบบในการใช้ (Signature), และส่วนที่เป็นตัวเนื้อของ Method

ในการใช้งานนี้จะต้องอ้างอิง Method ตามรูปแบบชื่อและรูปแบบในการใช้ให้ถูกต้องตามที่มีอยู่ ส่วนที่เป็นตัวเนื้อของ Method ซึ่งแสดงถึงรูปแบบการทำงานของ Method นั้นๆ โดยปกติจะถูกเขียนด้วยภาษาที่เป็นส่วนขยายเพิ่มเติมมาจากภาษาโปรแกรม

2.3.4 Class

จะหมายถึง กลุ่มของ Object ที่มีโครงสร้างภายในที่เหมือนๆ กัน นั่นคือ จะมีค่ารายละเอียดของค่า Attribute และรายละเอียดของ Method ที่เหมือนๆ กัน สำหรับตัว Object ใดๆ ที่เป็นของ Class ใด Class หนึ่งที่อยู่ในฐานข้อมูลจะเรียก Class นั้นว่าเป็น Class หลัก (Primary Class, Immediate Class หรือ Proper Class) ของ Object นั้นๆ ซึ่งจะเป็น Class ที่ Object ตัวนั้นมีความสัมพันธ์ในรูปแบบที่เป็น ตัวแทน (Instance-of) ของ Class นั้น และใน Class นี้จะเก็บลักษณะต่างๆ ของตัว Object ที่อยู่ใน Class ได้แก่ Attribute และ Method ต่างๆ ในบางกรณีอาจพบว่าเราสามารถให้ Object ตัวหนึ่ง ๆ เป็นตัวแทนของ Class มากกว่า 1 Class ได้ โดยการสืบทอดคุณสมบัติ (Inheritance) ซึ่งจะอธิบายต่อไป แต่อย่างไรก็ตาม Object ตัวนั้นจะต้องมี Class หลักได้เพียงตัวเดียว

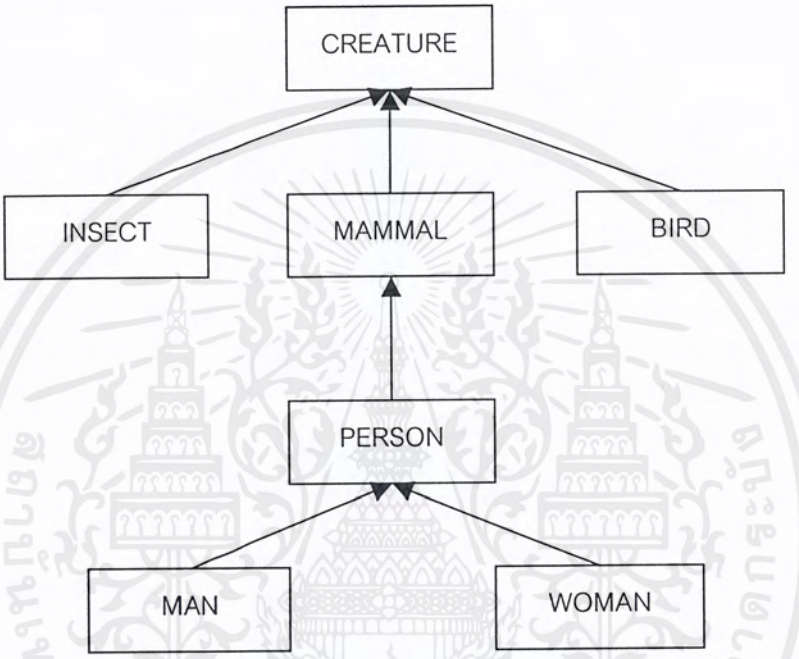
2.3.5 การสืบทอดคุณสมบัติ

ในระบบฐานข้อมูลเชิงวัตถุ แนวคิดของการสืบทอดคุณสมบัติทำให้ Object มี Attribute และ Method ที่เจาะจง(Specialized) ลงไปจาก Class ที่เป็น Class ต้น (Superclass หรือ Parent Class) ได้ โดยการสืบทอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Attribute และ Method จาก Class ดังกล่าวมายัง Class ของตัวจะเรียกว่า Subclass สำหรับ Object ที่อยู่ใน Class ใหม่นี้ ก็ยังถือว่าเป็นตัวแทนของ Class ต้นที่สืบทอดมา จากคุณสมบัติที่มีมาจากการสืบทอดนี้ทำให้เราได้ประโยชน์จากการที่เราสามารถเพิ่มศักยภาพทางด้านการนำกลับมาใช้ใหม่ (Reusability) และการขยายเพิ่มเติมคุณสมบัติ (Extensibility) ได้

ในส่วนต่อไปจะเป็นการแสดงตัวอย่างของลำดับชั้นของ Class (Class Hierarchies) ที่ได้มาจากการสืบทอดคุณสมบัติ โดยการสืบทอดจะเป็นในลักษณะแบบ Class ที่อยู่ในระดับที่ต่ำกว่าจะสืบทอดค่าคุณสมบัติต่างๆ จาก Class ที่อยู่สูงขึ้นไป ซึ่งจะแสดงได้ดังรูป



รูปที่ 2-1 การสืบทอดคุณสมบัติแบบลำดับชั้น

จากรูปจะเห็นว่ามี Class *Mammal Bird* และ *Insect* ซึ่งเป็น Class ย่อยของ Class *Creature* ส่วน Class Person เป็น Class ย่อยของ Class *Mammal* และ Class *Man* และ *Woman* เป็น Class ย่อยของ Class *Person* ในส่วนรายละเอียดของ Class ดังกล่าวข้างต้นนี้ เราสามารถเขียนตัวอย่างแสดงได้โดยยึดจากลำดับชั้นจากรูปได้ดังตัวอย่างข้างล่างที่จะแสดงต่อไปนี้ ซึ่งเขียนอยู่ในรูปที่ยึดหลักของไวยากรณ์ที่คล้ายๆกับภาษา C++

```
class creature
        properties
                type : string;
                weight : real;
                habitat : ( ... รูปแบบข้อมูลที่แสดงถึงลักษณะที่อยู่อาศัย ...);
                ...
        operations
                create() -> creature;
                predators (creature) -> set (creature);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            life_expectancy (creature) -> integer;
            ...
end creature.
class mammal
        inherit creature;
        properties
            Gestation_period : real;
        operations
            ...
end mammal.

class person
        inherit mammal;
        properties
            Surname, firstname : string;
            Date_of_birth : date;
            Origin : country;
end person.

class man
        inherit person;
        properties
            Wife : woman;
            ...
        operations
            ...
end man.

class woman
        inherit person;
        properties
            Husband : man;
            Maiden_name : string;
            ...
end woman.
```

2.3.6 Polymorphism

สำหรับความหมายของ Polymorphism โดยรวมแล้วก็คือ รูปแบบหนึ่งของการสืบทอดคุณสมบัติ โดยจะแบ่งออกได้เป็นสองลักษณะใหญ่ก็คือ ลักษณะแบบ Overloading จะหมายถึงการที่ Method สองตัวใดๆ ใน Class เดียวกันที่มีชื่อเหมือนกัน แต่ละรูปแบบการใช้งานที่แตกต่างกัน และแบบ Overiding ซึ่งหมายถึงความสามารถของโปรแกรมที่อ้างอิงถึง Attribute หรือ Method ที่มีชื่อเหมือนกัน แต่มีการเปลี่ยนแปลงความหมายไปเมื่อมีการสืบทอดค่าคุณสมบัตินั้นมาจาก Class ต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ระบบฐานข้อมูลอนุมานเชิงวัตถุ

สำหรับอีกแนวทางหนึ่งที่เป็นการขยายความสามารถของระบบจัดการฐานข้อมูลที่เป็นแบบ Relational ซึ่งมีการค้นคว้าวิจัยและพัฒนากันก็คือ แนวทางที่พยายามรวมเอาทั้งจุดเด่นของระบบฐานข้อมูลอนุมานและระบบฐานข้อมูลเชิงวัตถุเข้าด้วยกัน โดยทิศทางของการพัฒนาวิจัยจะเป็นการพยายามหาทฤษฎีที่ใช้อธิบายเรื่องของแนวความคิดเชิงวัตถุ ซึ่งแบ่งออกได้เป็นหลายแนวทางด้วยกัน เช่นการขยายความหมายของภาษาที่เป็น Logic Programming ที่มีอยู่ก่อนแล้ว เช่น Prolog ให้มีแนวความคิดในเชิงวัตถุด้วย โดยการใช้ลักษณะของภาษาที่เป็นแบบเชิงวัตถุ เช่น C++ หรือ Smalltalk เพิ่มเติมเข้ามา หรืออาจคิดทฤษฎีตรรกศาสตร์ใหม่ๆ เพิ่มเติมจากของเดิมเพื่อให้มีความสามารถในส่วนของที่เป็นเชิงวัตถุ และแนวทางอีกอย่างก็คือการคิดค้นทฤษฎีขึ้นมาใหม่ทั้งหมด สำหรับในส่วนของรายละเอียดของแนวทางเหล่านี้ จะขอไม่กล่าวถึงในรายงานฉบับนี้ เนื่องจากในโครงงานจะไม่เน้นในส่วนของการสร้างระบบฐานข้อมูลในส่วนนี้มากนัก แต่จะยกตัวอย่างหนึ่งของภาษาที่ใช้ในการจัดการระบบฐานข้อมูลชนิดนี้ ซึ่งจะเป็นหัวข้อสุดท้ายในบทนี้

- **F-logic [Kifer 1995]**

  สำหรับตัวอย่างหนึ่งของภาษาที่ใช้ในระบบฐานข้อมูลที่เป็นแบบอนุมานเชิงวัตถุ ซึ่งค่อนข้างเป็นที่รู้จักกันดี เนื่องจากมีการพัฒนาที่ยาวนานและ มีรูปแบบการกำหนดความหมายต่างๆ ที่รัดกุม (Well-Defined Semantics) นอกจากนี้ยังมีการพัฒนาต้นแบบเพื่อทดลองการใช้งานขึ้นมาแล้วด้วย ตามรายละเอียดใน [Kandzia, Schlepphorst 1997] แต่ยังมีคุณสมบัติในการทำงานของภาษาไม่ครบถ้วนทุกส่วน ซึ่งรูปแบบของภาษาเป็นการพัฒนาขึ้นมาใหม่ทั้งหมด โดยใช้ทฤษฎีทางด้านตรรกศาสตร์เป็นแนวทางในการพัฒนา สำหรับลักษณะของภาษาโดยทั่วไปมีรายละเอียดดังนี้

  - **Object Identity (OID)** สำหรับในภาษาจะใช้ OID ที่เป็นลักษณะแบบ Logical ซึ่งจะมีลักษณะเหมือนๆ กับ First-Order Term เช่น

    13 หรือ *_256* หรือ *john32*

    และสามารถใช้ฟังก์ชันในการสร้าง OID ที่มีความซับซ้อนมากขึ้นได้ เช่น

    father(mary) หรือ *head(csdept(stonybrook))*

  - **Attribute** จะแทนอยู่ในรูปแบบที่เรียกว่า Molecular Formula ซึ่งมีลักษณะตามตัวอย่าง

    john[name → "John Doe"; salary → 20000]
    father(mary)[address → "Main St. USA"; spouse → sally]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่างข้างต้น *john, father(mary)* และ *sally* เป็น OID ของ Object ของบุคคล, *"John Doe", "main St. USA"* เป็น OID ของ Object แบบสตริง (string) และ *20000* เป็น OID ของ Object แบบเลขจำนวนเต็ม (Integer) โดยค่า *name, salary, address* และ *spouse* จะเป็น Attribute มีข้อสังเกตว่าค่าแต่ละค่าที่เป็นสตริงและเลขจำนวนเต็มจะใช้ค่านั้นใช้เป็น OID ด้วย

- **Set-Valued Attribute** จากข้างต้น เรายังพบข้อจำกัดในการแสดงถึง Object ที่มีความซับซ้อนอยู่ ซึ่งจากตัวอย่างที่ผ่านมา เราไม่มีทางที่จะบอกว่า John มีลูกชื่อ Mary, Bob และ Alice เว้นเสียแต่เราจะทำการขยายความสามารถในการจัดการ Object ที่มีความซับซ้อน โดยอนุญาตให้ Attribute มีค่าเป็นเซต (Set-Valued) ได้ เช่น จากตัวอย่างที่แล้ว ถ้าเราต้องการเพิ่มค่า Attribute ที่มีค่าเป็นเซต คือ เซตของลูกๆ ก็สามารถทำได้โดยเขียนเป็น

  john[children ->> {mary, bob, alice}]

- **Method** สำหรับในภาษา F-logic จะมอง Method เป็นเหมือนกับค่า Attribute ที่มีค่าพารามิเตอร์

- **ลำดับชั้นของ Class** จะแสดงโดยการสร้างรูปแบบที่ใช้ในการอ้างอิงเพิ่มขึ้นมาใหม่มีลักษณะตามตัวอย่าง

  john : student หมายถึง *john* เป็น Object ของ Class *student*
  student :: person หมายถึง *student* เป็น Class ย่อยของ Class *person*

- **Higher-order Syntax** ถูกสร้างขึ้นมาเพื่อวัตถุประสงค์ต่างๆ ดังนี้
  1. ให้สามารถกำหนด (Defining), จัดการในเรื่องต่างๆ ของ Class และ Object โดยสามารถใช้ภาษาตัวเดียวกันได้
  2. ให้สามารถในการกำหนด Virtual Class (หรือ View) และคุณสมบัติต่างๆที่ใช้ในกฎอนุมาน
  3. ให้มีความสามารถในส่วนของการสืบทอดคุณสมบัติในส่วนของโครงสร้างข้อมูล
  4. ให้มีความสามารถในการตรวจสอบโครงสร้างข้อมูลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ก็ยังมีส่วนประกอบอื่นๆ อีกก็คือ เครื่องหมายของตัวที่ใช้ทำการเชื่อมต่อ (Connective) ได้แก่ AND, OR, NOT และ IF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ระบบผู้เชี่ยวชาญ

3.1 บทนำ

จากคำจำกัดความของโดยศาสตราจารย์ Edward Feigenbaum แห่งมหาวิทยาลัยสแตนฟอร์ด ซึ่งเป็นนักค้นคว้าชั้นแนวหน้าในสาขาวิชาปัญญาประดิษฐ์ (Artificial Intelligence) ได้กล่าวไว้ว่า ระบบผู้เชี่ยวชาญคือโปรแกรมคอมพิวเตอร์ที่มีความฉลาดด้วยการใช้ความรู้และขบวนการอนุมาน (Inference Procedure) ในการแก้ปัญหาที่มีความยุ่งยากขนาดถึงต้องใช้ประสบการณ์ความชำนาญของมนุษย์จึงจะแก้ไขได้

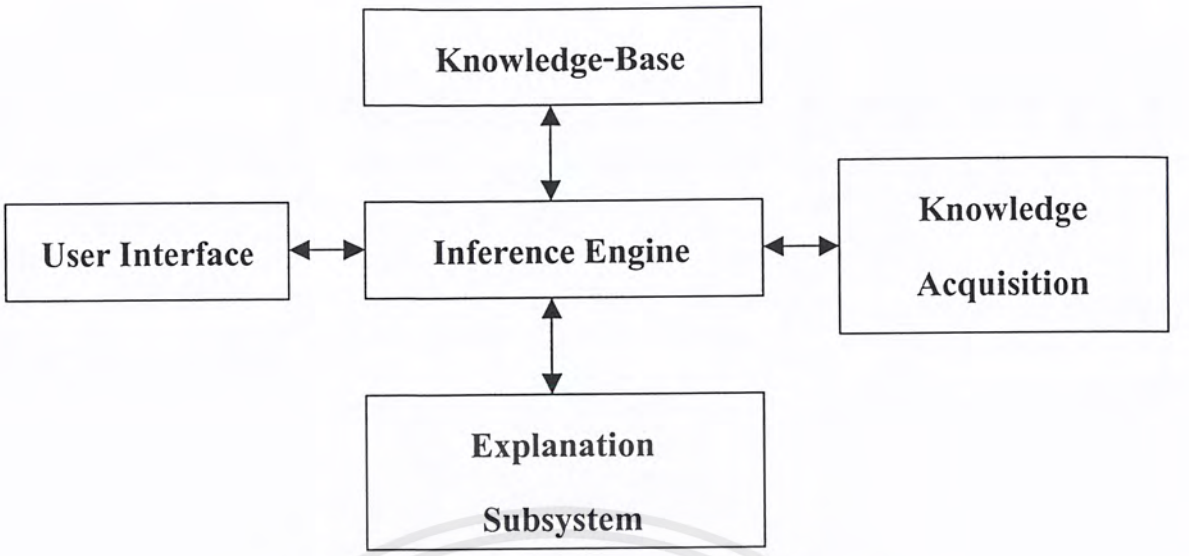
กล่าวคือ ระบบผู้เชี่ยวชาญ คือโปรแกรมคอมพิวเตอร์ที่เก็บทั้งความรู้เกี่ยวกับปัญหาที่จะแก้และขบวนการอนุมานเพื่อนำไปสู่ผลสรุปหรือคำตอบของปัญหานั้น ความรู้ที่เก็บมีทั้งความรู้ที่เป็นความจริงที่อาจจะถูกบันทึกไว้ในรูปของตำราหรือเอกสารทางวิชาการและความรู้ที่ได้จากประสบการณ์ที่อาจจะไม่อยู่ในรูปของตำราหรือเอกสารทางวิชาการ แต่จะต้องดึงมาจากผู้เชี่ยวชาญหรือผู้ชำนาญที่มีประสบการณ์นั้น

ปัญหาที่ระบบผู้เชี่ยวชาญจะแก้ส่วนใหญ่จะเป็นปัญหาที่ยุ่งยากและไม่ค่อยมีโครงสร้าง (Semi-Structured หรือ Ill-Structured Problem) ในปัญหาประเภทนี้คำตอบจะมีโอกาสเป็นได้หลายอย่าง ทั้งนี้ขึ้นอยู่กับสภาพขณะนั้นของปัญหาและข้อมูลที่เข้ามา ปัญหาประเภทนี้อาจจะอุปมาได้เหมือนกับการเล่นหมากรุก การเดินหมากครั้งต่อไปนั้นเดินได้หลายวิธีด้วยกันแต่ตัวหมากที่จะเดินดีที่สุดจะตัดสินจากสภาพของกระดานในขณะนั้นและหมากที่คิดว่าคู่ต่อสู้จะเดินในครั้งต่อไป ในการแก้ปัญหาประเภทนี้เรามักไม่สามารถจะกำหนดขั้นตอนในการแก้ปัญหาอย่างชัดเจนไว้ล่วงหน้าได้ ดังนั้นวิธีการแก้ปัญหาแบบมีที่มาซึ่งเป็นแบบเขียนโปรแกรมเป็นขั้นตอนการแก้ปัญหาหรืออัลกอริธึม (Algorithm) จึงไม่สามารถจะนำมาประยุกต์ใช้ในปัญหาประเภทนี้ได้ ระบบผู้เชี่ยวชาญถึงแม้จะเป็นโปรแกรมคอมพิวเตอร์ชนิดหนึ่ง แต่โครงสร้างและเทคนิคที่ใช้ในการสร้างหรือพัฒนาต่างจากของโปรแกรมที่มีมาและเป้าหมายในการประยุกต์ใช้ก็แตกต่างกัน การประยุกต์ใช้ระบบผู้เชี่ยวชาญที่ประสบความสำเร็จเท่าที่มีมาได้แก่ การวินิจฉัยโรค, การสำรวจทรัพยากรธรณี, การวิเคราะห์โครงสร้างสารอินทรีย์เคมี และการแนะนำระบบคอมพิวเตอร์ เป็นต้น

3.2 โครงสร้างพื้นฐานของระบบผู้เชี่ยวชาญ

ระบบผู้เชี่ยวชาญโดยทั่วไปจะประกอบด้วยส่วนประกอบพื้นฐาน 5 ส่วน ดังแสดงในรูปที่ 3-1 ส่วนที่เป็นหัวใจที่จะขาดเสียมิได้ คือ ฐานความรู้และเครื่องมืออนุมาน (Inference Engine) รายละเอียดโดยย่อของแต่ละส่วนสามารถอธิบายได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-1โครงสร้างพื้นฐานของระบบผู้เชี่ยวชาญ

- ฐานความรู้ (Knowledge Base)

ส่วนนี้เปรียบเสมือนกับข้อมูลในซอฟต์แวร์ธรรมดาหรือฐานข้อมูล (Database) ในระบบสารสนเทศ (Information Systems) เป็นส่วนที่ใช้เก็บความรู้ทุกประเภทไม่ว่าจะเป็นความรู้ที่ได้จากตำราหรือความรู้ที่ได้จากประสบการณ์ ปัญหาหลักของฐานความรู้ก็คือ การเลือกวิธีที่จะใช้ในการแสดงความรู้หรือโครงสร้างสำหรับเก็บความรู้ที่เหมาะสม ปัญหานี้เปรียบได้กับการเลือกโครงสร้างข้อมูลหรือโครงสร้างฐานข้อมูลที่เหมาะสมในระบบซอฟท์แวร์แบบธรรมดา

- เครื่องมืออนุมาน (Inference Engine)

ส่วนนี้เปรียบได้กับอัลกอริธึม เป็นส่วนที่ควบคุมการใช้ความรู้ในฐานความรู้เพื่อแก้ไขปัญหาอย่างมีประสิทธิภาพ วิธีการอนุมานมีหลายแบบแต่แยกประเภทใหญ่ๆ ได้ 2 ประเภท คือ อนุมานแบบเดินหน้า (Forward-Chaining Inference) และอนุมานแบบย้อนหลัง (Backward-Chaining Inference) ทั้งสองวิธีนี้ต่างก็มีข้อดีและข้อเสีย ทั้งนี้ขึ้นอยู่กับลักษณะของปัญหา ในระบบผู้เชี่ยวชาญบางระบบจะใช้วิธีอนุมานทั้งสองวิธีรวมกัน

- ส่วนดึงความรู้ (Knowledge Acquisition)

เป็นส่วนของระบบผู้เชี่ยวชาญที่ใช้ในการช่วยดึงเอาความรู้จากตำราหรือฐานข้อมูลและจากผู้เชี่ยวชาญ การดึงเอาความรู้จากตำราหรือฐานข้อมูลนั้นทำได้ไม่ยาก ถ้าหากเราสามารถจัดการความรู้จากแหล่งดังกล่าวให้เป็นระบบ และเข้ากันได้กับโครงสร้างของฐานความรู้ เราก็จะสามารถบรรจุความรู้เหล่านั้นเข้าไปในฐานข้อมูล แต่ทว่าการดึงเอาความรู้จากผู้เชี่ยวชาญนั้นทำได้ยาก จำเป็นต้องใช้เทคนิคต่างๆ เข้าช่วยหรือไม่ก็ทำให้ระบบผู้เชี่ยวชาญสามารถเรียนรู้ด้วยตนเองในบางส่วน ปัจจุบันการเรียนรู้ (Learning) เป็นหัวข้อค้นคว้าที่นักค้นคว้าในสาขาปัญญาประดิษฐ์ให้ความสนใจมากที่สุดสาขาหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ส่วนอธิบาย (Explanation Subsystem)

ส่วนนี้ทำหน้าที่อธิบายรายละเอียดของขั้นตอนการวินิจฉัยต่อผู้ใช้ว่าข้อสรุปหรือคำตอบนั้นได้มาอย่างไรและทำไม

- ส่วนติดต่อกับผู้ใช้ (User Interface)

เป็นส่วนที่ทำหน้าที่เป็นตัวกลางระหว่างผู้ใช้กับระบบเพื่อทำให้การสื่อสารระหว่างผู้ใช้กับระบบเป็นไปได้อย่างราบรื่น และช่วยให้ผู้ใช้ยอมรับระบบมากขึ้น

ในระบบผู้เชี่ยวชาญบางระบบจะไม่มีส่วนประกอบทั้งห้าส่วนดังกล่าวข้างต้น แต่ที่ขาดไม่ได้แน่ๆ คือ ฐานความรู้และเครื่องมืออนุมาน บางคนถึงกับกล่าวว่า Knowledge Base + Inference Engine = Expert System เลย

3.3 รูปแบบการแทนความรู้ (Knowledge Representation) ชนิดต่างๆ ที่ใช้ในระบบผู้เชี่ยวชาญ

สำหรับในระบบผู้เชี่ยวชาญจะมีรูปแบบการแทนความรู้ที่แตกต่างกันไปหลายแบบด้วยกัน ซึ่งรูปแบบต่างๆ ที่สามารถพบเห็นได้ พอจะอธิบายได้โดยสังเขปดังนี้ ซึ่งในปัจจุบันระบบผู้เชี่ยวชาญก็มิได้มีการใช้รูปแบบการแทนความรู้ที่อยู่ในลักษณะอันหนึ่งอันใดเพียงอย่างเดียวก็ได้ แต่อาจมีการใช้รูปแบบการแทนความรู้หลายๆ แบบเข้าด้วยกันหรืออาจมีการดัดแปลงรูปแบบการแทนความรู้แบบนี้เข้าด้วยกันด้านอื่นๆ ก็ได้ เช่น การใช้ Neural Network หรือทฤษฎี Fuzzy เป็นต้น

3.3.1 แบบของกฎ (Rule-Based)

จะเป็นการแสดงความรู้ในรูปแบบของกฎ ซึ่งแทนอยู่ในรูปแบบของ First Order Logic คือ จะแทนรูปแบบความรู้ต่างๆ ในลักษณะของ Horn Clause ซึ่งสามารถเขียนให้อยู่ในรูปแบบที่สามารถทำความเข้าใจได้ง่ายๆ ดังนี้

(Condition) → *(Action)* หรือเขียนในรูปแบบที่เข้าใจง่ายๆ ได้อีกแบบ คือ

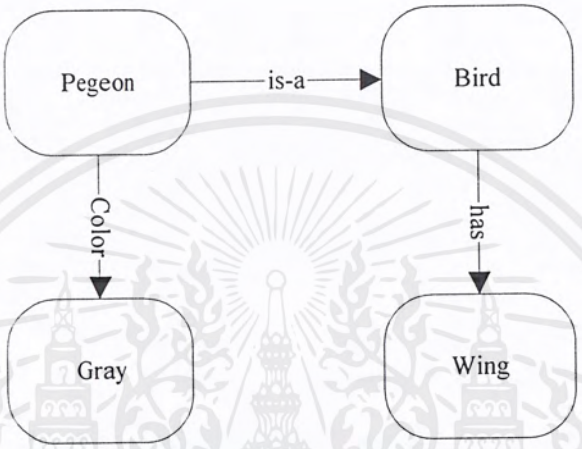
If (Condition) Then (Action)

คือ เมื่อพบว่าในส่วนของ *(Condition)* เป็นจริงแล้ว ก็ให้ทำในส่วนของ *(Action)* ซึ่งรูปแบบการแทนแบบนี้จะสามารถเว้นไว้และมีเฉพาะในส่วนของ *(Action)* ได้ด้วย ซึ่งถ้าเป็นการแทนแบบนี้จะหมายถึงการแทนความรู้ที่เป็นข้อเท็จจริง (Fact)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 แบบใช้ข่ายความหมาย (Semantic Network)

ในแบบนี้จะแทนความรู้โดยใช้การแสดงความสัมพันธ์ระหว่างสิ่งสองสิ่ง (Binary-Relation) ซึ่งอาจแทนได้ง่ายๆ ด้วยการใช้กราฟ โดยส่วนของ Node ในกราฟจะแสดงในส่วนที่เป็น Object, Concept, Event, Action และ State ของสิ่งที่เราต้องการสร้างเป็นฐานความรู้ ส่วนในส่วนของ Arc จะแสดงถึงความสัมพันธ์กันระหว่าง Node ต่างๆ ที่แทนถึงสิ่งที่ได้อธิบายไว้ในข้างต้นแล้ว ซึ่งถ้ารูปแบบของความสัมพันธ์มีทิศทางแล้วก็ต้องใช้ Arc แบบที่แสดงถึงทิศทางของความสัมพันธ์ด้วย (หมายถึงใช้ Arc ที่มีหัวลูกศร) ตัวอย่างของการแทนความรู้แบบนี้แสดงได้ดังรูปที่ 3-2



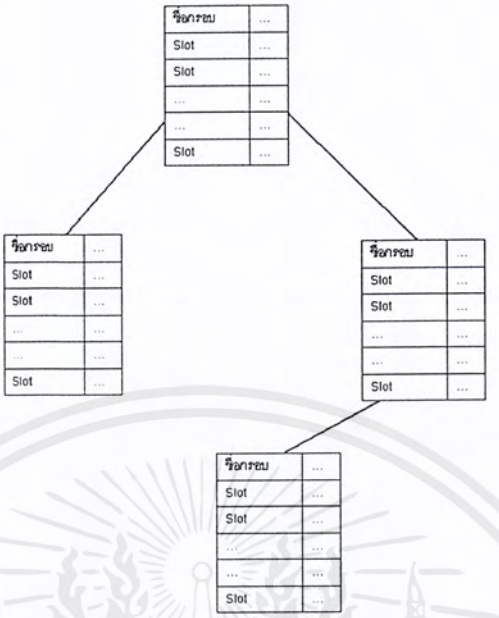
รูปที่ 3-2 ตัวอย่างการแทนความรู้ด้วยข่ายความหมาย

3.3.3 แบบใช้กรอบ (Frame-based) และแบบเชิงวัตถุ (Object-Oriented)

การแทนรูปแบบความรู้แบบกรอบนี้จะมีลักษณะแบบเป็นโครงสร้าง คือจะใช้แนวความคิดที่ว่า ในการแทนความรู้หนึ่งๆ นั้นจะประกอบไปด้วยสิ่งย่อยๆ หลายๆ อย่างประกอบเข้าด้วยกัน และในการอ้างอิงถึงความรู้นั้นก็จะอ้างอิงถึงส่วนย่อยๆ ต่างๆ เหล่านี้ทั้งหมด ซึ่งส่วนย่อยๆ ต่างๆ เหล่านี้จะถูกนำมารวมกันเป็นกรอบ (Frame) โดยการแสดงความรู้จะต้องมีกรอบที่แสดงถึงความรู้สามัญบรรจุอยู่ ซึ่งจะเป็นความรู้พื้นฐาน ในกรณีที่ไม่มีสิ่งบ่งบอกอย่างอื่นก็จะใช้ข้อมูลที่อยู่ในกรอบนี้ไปใช้ และสามารถมีกรอบได้หลายๆ กรอบโดยจะมีการแสดงถึงความสัมพันธ์ที่เกี่ยวข้องกันได้เช่นเดียวกับในกรณีของข่ายความหมาย โดยรูปแบบที่สำคัญจะเป็นการสืบทอดคุณสมบัติ คือ ในกรณีที่กรอบนั้นไม่ได้มีการกำหนดข้อมูลบางอย่างเป็นการจำเพาะก็จะใช้ข้อมูลที่สืบทอดมาจากกรอบอื่นๆ แทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างรูปแบบการแทนความรู้ด้วยกรอบสามารถแสดงได้โดยดูจากรูปที่ 3-3



รูปที่ 3-3 การแทนความรู้ด้วยกรอบ

สำหรับการแทนความรู้แบบเชิงวัตถุในระบบผู้เชี่ยวชาญนั้นเป็นผลมาจากการที่แนวความคิดเชิงวัตถุในวิทยาการคอมพิวเตอร์ในด้านอื่นๆ เป็นที่นิยมขึ้นมา จึงมีการพยายามนำมาประยุกต์เข้าด้วยกันกับการแทนความรู้ด้วยกรอบซึ่งก็มีลักษณะคล้ายๆ กันอยู่แล้วและมีการปรับเปลี่ยนไปมาโดยใช้หลักการของทั้งสองส่วนผสมผสานเข้าด้วยกัน จนบางครั้งเราสามารถเรียกรูปแบบต่างๆ ทั้งสองแบบนี้เข้ากันได้ แต่อย่างไรก็ดี ได้มีผู้ที่ชี้ให้เห็นถึงความแตกต่างของรูปแบบทั้งสองนี้ เช่นใน [Nikolopoulos 1997] โดยให้ดูในส่วนของการแบ่งแยกในส่วนของ Class และ Instance ที่ชัดเจนถ้าเป็นการแทนรูปแบบที่เป็นเชิงวัตถุ ในขณะที่ในแบบกรอบต้องมีการพิจารณาเป็นกรณี ๆ ไปว่ากรอบนั้นเป็น Class หรือว่าเป็น Instance ซึ่งบางครั้งกรอบที่ใช้แสดงความรู้ตัวเดียวกันจะเป็น Class ได้และก็อาจเป็น Instance ได้ด้วยเมื่อเราพิจารณาในอีกกรณี

3.4 ข้อจำกัดของระบบผู้เชี่ยวชาญ

ถึงแม้ว่าระบบผู้เชี่ยวชาญจะมีความสามารถในการทำงานเพื่อแก้ปัญหาหรือช่วยในการทำงานบางประเภทได้ยืดหยุ่นและดีกว่าโปรแกรมคอมพิวเตอร์อื่นๆ ก็ตาม แต่ระบบผู้เชี่ยวชาญที่ใช้กันอย่างแพร่หลายอยู่ในปัจจุบันนี้ก็มีข้อจำกัดที่เป็นอุปสรรคที่สำคัญของการพัฒนาตัวระบบก็คือ การที่ระบบผู้เชี่ยวชาญสามารถทำงานกับฐานความรู้ที่ต้องสามารถเก็บไว้ในหน่วยความจำหลักของเครื่องคอมพิวเตอร์เท่านั้น ทำให้ในการพัฒนาระบบผู้เชี่ยวชาญต้องทำกับระบบที่มีขนาดไม่ใหญ่เกินไป เพื่อที่จะให้สามารถเก็บฐานความรู้ไว้ในหน่วยความจำหลักของเครื่องคอมพิวเตอร์ได้ และในฐานความรู้ที่สร้างขึ้นมาจะประกอบไปด้วยรูปแบบความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

รู้ที่เป็นกฎเสียเป็นส่วนมาก ส่วนข้อมูลที่เป็นข้อเท็จจริงจะถูกกำจัดออกไปจากส่วนของฐานความรู้และจะทำการสอบถามเอาเองจากผู้ใช้ ซึ่งข้อเท็จจริงต่างๆ เหล่านี้เมื่อมีการถามจากผู้ใช้แล้ว ก็ไม่ได้มีการเก็บข้อมูลในส่วนนี้เอาไว้เพื่อนำไปใช้งานเมื่อมีการเรียกใช้งานระบบผู้เชี่ยวชาญในครั้งต่อๆ ไป ทำให้ในการใช้งานระบบผู้เชี่ยวชาญแต่ละครั้งต้องมีการใส่ข้อมูลเพื่อตอบคำถามเดิมๆ อยู่ตลอดเวลาทั้งๆ ที่บางคำถามก็ไม่จำเป็นหรือควรถามเพียงแค่ครั้งแรกที่มีการใช้งานระบบก็พอ ลักษณะการทำงานแบบนี้จะก่อให้เกิดผลเสียในแง่ต่างๆ ต่อไปนี้คือ

- โปรแกรมระบบผู้เชี่ยวชาญมีการถามผู้ใช้มากเกินไป ทำให้เมื่อมีการใช้งานระบบผู้เชี่ยวชาญตัวเดิมนั้นบ่อยครั้งมากเข้า จะก่อให้เกิดความเบื่อหน่ายในการใช้ได้ และทำให้การป้อนข้อมูลของผู้ใช้เพื่อนำไปประกอบการตัดสินใจของระบบเกิดความผิดพลาดขึ้นได้เนื่องจากผู้ใช้ไม่มีความตั้งใจที่จะตอบคำถามที่ระบบต้องการอย่างระมัดระวัง
- คำถามบางคำถามจะเป็นการถามถึงข้อมูลในอดีต ซึ่งจะเป็นข้อมูลที่จะเหมือนกันทุกครั้งหรือมีการเปลี่ยนแปลงที่ค่อนข้างน้อย และทุกครั้งที่มีการเรียกใช้งานระบบผู้เชี่ยวชาญจะต้องมีการถามถึงข้อมูลที่ไม่ค่อยมีการเปลี่ยนแปลงนี้อยู่ตลอดทุกครั้ง บางครั้งข้อมูลเหล่านี้ตัวผู้ใช้เองก็อาจจำไม่ได้เป็นสาเหตุทำให้เกิดเหตุการณ์ที่ผู้ใช้ให้ข้อมูลที่ผิดพลาดขึ้นได้
- คำถามบางอย่างที่ระบบผู้เชี่ยวชาญจะถามกับผู้ใช้บางคำถามจะเป็นการถามถึงข้อมูลที่แม้แต่ตัวผู้ใช้ระบบเองก็ไม่ทราบ คำถามที่อยู่ในข่ายกรณีนี้จะพบเห็นได้อยู่ทั่วไปในกรณีที่เป็นระบบผู้เชี่ยวชาญทางด้านการแพทย์ ซึ่งจะมีการอ้างอิงถึงข้อมูลบางอย่างที่เป็นรายละเอียดเฉพาะทางซึ่งบางครั้งตัวผู้ใช้ก็ไม่ได้มีการทราบมาก่อน เช่น ถ้าระบบผู้เชี่ยวชาญมีการถามถึงชื่อตัวยาที่ผู้ป่วยเคยได้รับมาก่อนหน้านี้ ตัวผู้ป่วยเองซึ่งเป็นผู้ใช้ระบบผู้เชี่ยวชาญก็อาจจะไม่ทราบก็ได้ เพราะรายละเอียดต่างๆ เหล่านี้ทางแพทย์ผู้สั่งยาจะไม่จำเป็นต้องมีการบอกกับผู้ป่วยก็ได้ เป็นต้น

3.5 การติดต่อของระบบผู้เชี่ยวชาญกับโปรแกรมอื่นๆ

จากการที่ตัวของโปรแกรมระบบผู้เชี่ยวชาญเองนั้นมีข้อจำกัดต่างๆ ดังที่ได้อธิบายมาแล้วข้างต้น ทางผู้สร้างโปรแกรมเปลือกระบบผู้เชี่ยวชาญ (Expert System Shell) ซึ่งจะหมายถึงโปรแกรมระบบผู้เชี่ยวชาญที่มีแต่ส่วนของกลไกเครื่องมืออนุมาน และจะเว้นในส่วนของฐานความรู้ให้ผู้พัฒนาระบบเชี่ยวชาญทำการพัฒนาและเขียนขึ้นมาเอง จึงได้มีการเพิ่มเติมความสามารถในการติดต่อกับโปรแกรมตัวอื่นๆ หรืออย่างน้อยที่สุด ก็ได้เพิ่มเติมในส่วนของการอ่านและเขียนข้อมูลจากแฟ้มข้อมูลได้ เพื่อช่วยเพิ่มประสิทธิภาพการในการใช้งานตัวระบบผู้เชี่ยวชาญได้ดีขึ้น โดยถ้าเป็นระบบผู้เชี่ยวชาญที่มีความสามารถอ่านหรือเขียนแฟ้มข้อมูลได้นั้น ก็จะทำให้ระบบผู้เชี่ยวชาญที่สร้างขึ้นมานั้นสามารถใช้งานได้กว้างขวางมากกว่าเดิม แต่ถ้าเป็นการติดต่อกับโปรแกรมอื่นๆ ซึ่งถ้าเป็นโปรแกรมประเภทกระดาษตาราง (Spread Sheet) ระบบผู้เชี่ยวชาญก็จะใช้ประโยชน์จากโปรแกรมตัวนี้ในการทำการวิเคราะห์ข้อมูลทางด้านสถิติ, ทำการสร้างรูปแบบรายงานเพื่อแสดงผลลัพธ์ของการวิเคราะห์ซึ่งจะมีรูปแบบที่หลากหลายกว่าการแสดงผลด้วยการใช้ตัวระบบผู้เชี่ยวชาญเอง และการเรียกดึงข้อมูลจากแฟ้มข้อมูลที่อยู่ในรูปแบบของกระดาษตารางที่ใช้ในงานของโปรแกรมนั้นเพื่อใช้ในการประมวลผลแทนการถามข้อมูลจากผู้ใช้ หรือถ้าเป็นโปรแกรมอีกแบบที่โปรแกรมเปลือกระบบผู้เชี่ยวชาญบางตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
42773

สามารถทำการติดต่อได้ก็คือโปรแกรมจัดการระบบฐานข้อมูลซึ่งทำให้โปรแกรมระบบผู้เชี่ยวชาญสามารถทำงานโดยดึงข้อมูลจากฐานข้อมูลที่เก็บไว้ โดยฐานข้อมูลตัวนี้สามารถถูกใช้งานโดยโปรแกรมอื่นๆ ได้ด้วยเช่นกัน รูปแบบข้อกำหนดที่ใช้ในการติดต่อกับส่วนภายนอกนี้ก็จะแตกต่างกันไป โดยจะขึ้นอยู่กับโปรแกรมเปลือกระบบผู้เชี่ยวชาญนั้นว่ามีลักษณะเป็นอย่างไร เช่น รูปแบบการแทนความรู้ (Knowledge Representation), ระบบปฏิบัติการที่โปรแกรมนั้นทำงาน, รูปแบบที่ใช้ในการติดต่อที่ทางผู้พัฒนาโปรแกรมเปลือกระบบผู้เชี่ยวชาญนั้นๆ กำหนดมา ซึ่งจะมีรายละเอียดที่ไม่เหมือนกันเลย

3.6 ปัญหาที่พบในกรณีที่ข้อมูลที่ระบบผู้เชี่ยวชาญเรียกใช้มีขนาดใหญ่มากๆ

จากหัวข้อที่แล้วจะพบว่าเราสามารถสร้างระบบผู้เชี่ยวชาญที่มีการเก็บข้อมูลต่างๆ ไว้เพื่อในการใช้งานครั้งหลังๆ จะได้นำข้อมูลเหล่านี้ที่ระบบผู้เชี่ยวชาญได้เคยถามผู้ใช้แล้วในอดีตมาใช้ประโยชน์ในการทำงานครั้งหลังๆ ได้ และไม่ต้องทำการถามผู้ใช้ซ้ำอีก ถ้าโปรแกรมเปลือกระบบผู้เชี่ยวชาญที่เราใช้นั้นมีความสามารถในการติดต่อกับส่วนภายนอกได้

อย่างไรก็ตาม ในกรณีที่ข้อมูลที่ระบบผู้เชี่ยวชาญจะต้องใช้ในการทำงานนั้นมีขนาดใหญ่มากๆ แล้วการเรียกข้อมูลจากภายนอกที่อยู่ในรูปแบบของการอ่าน/เขียน แฟ้มข้อมูลโดยตรง หรือการทำงานกับแฟ้มข้อมูลผ่านทางโปรแกรมกระดาษตารางนั้นจะพบกับความยากลำบากมากขึ้น เนื่องจากว่าทางโปรแกรมเปลือกระบบผู้เชี่ยวชาญหรือโปรแกรมกระดาษตารางมีความสามารถในส่วนของการจัดการข้อมูลที่ไม่ดีพอเนื่องจากโปรแกรมเหล่านี้ไม่ได้พัฒนาขึ้นมาเพื่อให้มีความสามารถในส่วนนี้เป็นหลักอยู่แล้ว ทำให้การพัฒนาโปรแกรมระบบผู้เชี่ยวชาญที่มีขนาดใหญ่ๆ จะทำได้อย่างยากลำบาก และเป็นการลำบากอีกเช่นกันถ้าเราจะนำแฟ้มข้อมูลต่างๆ เหล่านี้ที่ใช้ในการทำงานของระบบผู้เชี่ยวชาญมาใช้ประโยชน์โดยโปรแกรมอื่นๆ ด้วยแทนที่จะถูกใช้งานโดยโปรแกรมเพียงตัวเดียว แต่ถ้ามีการนำเอาระบบจัดการฐานข้อมูลมาใช้ประโยชน์ในด้านนี้ เราก็จะพบว่าปัญหาต่างๆ ที่พบเหล่านี้ก็จะสามารถจัดการได้ง่ายๆ โดยที่การใช้งานในส่วนของระบบผู้เชี่ยวชาญก็ยังสามารถขยายขอบเขตได้กว้างขวางและยืดหยุ่นได้มากกว่าเดิมด้วย เนื่องจากข้อมูลที่เก็บไว้ในฐานข้อมูลนั้นเราสามารถนำไปใช้งานในส่วนอื่นๆ ได้ด้วย

3.7 ตัวอย่างระบบผู้เชี่ยวชาญที่มีการติดต่อกับข้อมูลภายนอก

3.7.1 BERMUDA [Ioannidis et al. 1989]

เป็นตัวอย่างหนึ่งของต้นแบบที่นำความสามารถของระบบผู้เชี่ยวชาญมาใช้ร่วมกันกับระบบจัดการฐานข้อมูล ซึ่งทำให้ระบบที่ได้มีความสามารถทั้งในส่วนที่เป็นคุณสมบัติของระบบจัดการฐานข้อมูลและในส่วนของการทำงานที่เป็นข้อได้เปรียบในแบบของระบบผู้เชี่ยวชาญ โดยในส่วนของกฎจะเก็บไว้ที่ส่วนของ Front-End ที่จะทำงานติดต่อกับผู้ใช้ซึ่งสามารถมีได้หลายตัว และแต่ละตัวจะเรียกใช้ข้อมูลในส่วนของข้อเท็จจริงที่เก็บไว้ในส่วนของฐานข้อมูลซึ่งเป็นส่วนที่ใช้ร่วมกัน โดยการเรียกใช้จะผ่านตัวกลางตัวหนึ่งที่ใช้ในการจัดการในเรื่องของการจัดส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบการแทนความรู้ที่ใช้จะเป็นการแทนอยู่ในรูปของภาษา Prolog และมีการติดต่อกับฐานข้อมูลแบบ Relational โดยมีข้อดีอยู่ที่ผู้ใช้ไม่ต้องทำการจัดการในส่วนของการเรียกข้อมูลจากฐานข้อมูลเองเนื่องจากระบบจะทำหน้าที่จัดการให้เองโดยใช้เครื่องมือแบบ Preprocessor สำหรับข้อจำกัดของระบบนี้ก็คือในระหว่างการทำงานของระบบจะต้องไม่มีการเปลี่ยนแปลงใดๆ เกิดขึ้นในส่วนของฐานข้อมูล

3.7.2 Tanguy [Czejdo et al. 1993]

เป็นต้นแบบของฐานความรู้ที่เป็นการใช้รูปแบบการแทนความรู้แบบเป็น Object-Oriented และมีการแบ่งแยกฐานความรู้ออกเป็นส่วนๆ ด้วย โดยจะมีส่วนที่ทางระบบจะทำการบันทึกลงในหน่วยความจำสำรองด้วยในกรณีที่พิจารณาแล้วพบว่าข้อมูลในส่วนนั้นสามารถนำไปใช้งานในโปรแกรมอื่นๆ ได้หลายงาน

จุดเด่นของระบบฐานความรู้ระบบนี้ก็คือ มีการนำเอาแนวความคิดแบบ Object-Oriented มาใช้ประโยชน์ มีการนำเสนอวิธีการที่ใช้ในการจัดการข้อมูลที่เป็นแบบกลุ่มข้อมูลหรือเป็น Set ซึ่งภาษาโปรแกรมแบบ Object-Oriented ทั่วๆ ไป เช่น C++ ไม่สามารถทำการสนับสนุนโดยตรงได้ และการนำเสนอแนวคิดของการเก็บข้อมูลไว้เพื่อวัตถุประสงค์ในการใช้งานข้อมูลชุดนั้นจากหลายๆ โปรแกรมร่วมกัน รวมทั้งมีการนำเสนอ รูปแบบการเก็บกฎในโครงสร้างการแทนความรู้ที่เป็นแบบ Object-Oriented โดยเก็บให้อยู่ในรูปแบบของ Trigger ที่จะทำงานเมื่อได้รับ message ขึ้นมา

สำหรับข้อจำกัดของระบบนี้ก็คือ ข้อมูลในส่วนที่มีการบันทึกเก็บไว้นั้นไม่ได้จัดเก็บไว้ในระบบจัดการฐานข้อมูลทำให้การจัดการเกี่ยวกับตัวข้อมูลทำได้ไม่สะดวกเท่ากับการใช้ระบบจัดการฐานข้อมูลซึ่งรองรับการทำงานในส่วนนี้ได้มากกว่า และยังเป็นผลให้ขอบเขตการในการนำระบบไปใช้ประโยชน์ค่อนข้างจำกัดกว่าด้วย

3.7.3 ระบบฐานความรู้ในระบบสารสนเทศทางการแพทย์ใน [Abawajy,Shepard 1994]

เป็นการนำเอาฐานความรู้มาประยุกต์ใช้กับระบบฐานข้อมูลต่างๆ ที่มีอยู่เพื่อใช้ประโยชน์ในการสืบค้นเพื่อหาข้อมูลที่เกี่ยวข้อง โดยรูปแบบการแทนความรู้ที่ใช้ในฐานความรู้ชนิดนี้จะมีลักษณะเป็นข่ายความหมายที่มีการใช้ระบบฐานข้อมูลหลายๆ ระบบร่วมกัน โดยในการจัดสร้างระบบจะใช้ประโยชน์จากการพัฒนาในรูปแบบ Object-Oriented ทั้งในส่วนของฐานความรู้ (ใช้ C++) และระบบจัดการฐานข้อมูล (ONTOS) ที่ใช้เพื่อทำการสร้างฐานความรู้ที่เป็นรูปแบบข้างต้น

วิธีการพัฒนาระบบที่นำเสนอขึ้นมานี้จะทำการสร้างในส่วนของระบบฐานข้อมูลขึ้นมาก่อน จากนั้นจึงทำการสร้างส่วนของระบบฐานความรู้ขึ้นมาภายหลัง ซึ่งข้อดีของระบบนี้ก็คือ เรายังสามารถนำระบบฐานข้อมูลที่เป็นส่วนหนึ่งของระบบไปใช้ต่างหากแบบเป็นอิสระได้ แต่ก็ทำให้เกิดข้อเสียด้วยเช่นกัน เนื่องจากหากเราทำการเปลี่ยนแปลงโครงสร้างของฐานข้อมูลตัวใดตัวหนึ่งแล้วก็จะต้องทำการแก้ไขในส่วนของฐานความรู้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.4 CLUES [Telezadeh et al. 1995]

เป็นระบบผู้เชี่ยวชาญที่มีการพัฒนาขึ้นและใช้งานจริงเพื่อช่วยในการตัดสินใจในการปล่อยเงินกู้ เนื่องจากในการทำงานเพื่อพิจารณาปล่อยเงินกู้ให้แก่ลูกค้าแต่ละรายนั้นจะใช้เวลาในการพิจารณาที่ยาวนานเนื่องจากต้องทำการพิจารณาข้อมูลที่เกี่ยวข้องซึ่งมีรายละเอียดปลีกย่อยค่อนข้างมาก รูปแบบสถาปัตยกรรมของระบบก็เป็นการใช้รูปแบบการแทนความรู้ด้วยกฎเนื่องจากทางผู้พัฒนาเห็นว่าสามารถที่จะย้อนดูผลการพิจารณาได้ว่าที่ได้ผลการตัดสินใจออกนั้นเป็นแบบนั้นได้มาจากการพิจารณาอย่างไรซึ่งถือว่าเป็นข้อได้เปรียบของระบบผู้เชี่ยวชาญเมื่อเปรียบเทียบกับระบบอื่นๆ ที่สามารถตรวจสอบผลการทำงานได้ ในขณะที่หากเป็นการใช้เทคนิคแบบอื่นๆ เช่น Neural Network หรือ Genetic Algorithm การตรวจสอบความถูกต้องของผลที่ได้นั้นจะทำได้ลำบากหรือทำไม่ได้เลย ซึ่งระบบที่ได้พัฒนาขึ้นมานี้ก็มีการติดต่อกับฐานข้อมูลด้วยเพื่อทำการตรวจดูประวัติการกู้ยืมเงินของลูกค้าเพื่อใช้ประกอบการพิจารณาเพื่อตัดสินใจการปล่อยกู้ในครั้งปัจจุบัน

จุดเด่นของระบบนี้ก็คือจะมีโครงสร้างของระบบที่สามารถเรียกใช้งานจากระยะไกลได้ผ่านทางสายสัญญาณสื่อสารด้วยการใช้โมเด็ม โดยในส่วนของสาขาของเครือข่ายก็ทำการส่งข้อมูลที่ใช้ในการพิจารณาซึ่งจะอยู่ในรูปแบบของแฟ้มข้อมูลที่เป็นรายงานเกี่ยวกับรายละเอียดการขอกู้เข้ามาทำการประมวลผลที่ส่วนกลางซึ่งจะเก็บโปรแกรมระบบผู้เชี่ยวชาญที่ใช้ในการตัดสินใจรวมทั้งฐานข้อมูลที่เก็บข้อมูลที่ต้องใช้ในการพิจารณาด้วยจากนั้นก็จะทำการส่งเฉพาะข้อมูลส่วนที่เป็นผลการพิจารณากลับไปยังสาขาที่ทำการส่งข้อมูลมาในรูปแบบของแฟ้มรายงานซึ่งสามารถเลือกได้หลายแบบ

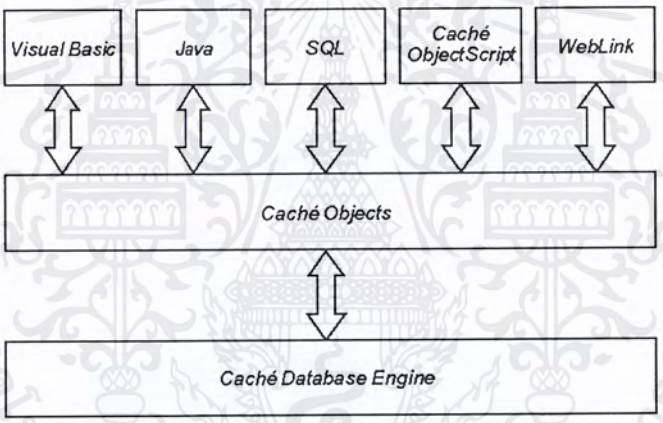
3.7.5 GermWatcher [Steib 1996]

เป็นระบบผู้เชี่ยวชาญเพื่อช่วยวินิจฉัยในงานทางด้านการแพทย์ที่พัฒนาขึ้นและใช้งานจริงเช่นเดียวกับ CLUES โดยในการพัฒนาช่วงแรกๆ นั้นจะยังไม่มีการใช้ระบบจัดการฐานข้อมูลมาใช้งานในระบบ ซึ่งทำให้เกิดปัญหาขึ้นเนื่องจากการบริหารข้อมูลทำได้ยากลำบาก ต่อมาจึงมีการนำระบบจัดการฐานข้อมูลมาใช้งานซึ่งก็ทำให้การบริหารข้อมูลต่างๆ ทำได้ดีขึ้น และยังช่วยให้ขอบเขตความสามารถในของระบบเพิ่มขึ้นด้วย ซึ่งก็เป็นตัวอย่างของข้อดีของการใช้ระบบจัดการฐานข้อมูลเข้ามาใช้ในงานร่วมกับระบบผู้เชี่ยวชาญที่ชัดเจนตัวอย่างหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ระบบจัดการฐานข้อมูลเชิงวัตถุ Caché

จากที่ได้กล่าวมาแล้วข้างต้นว่าในขั้นตอนการทำโครงงานชิ้นนี้จะศึกษาการประยุกต์ฐานข้อมูลเชิงวัตถุมาใช้งานกับฐานข้อมูลเชิงเวลา ระบบจัดการฐานข้อมูลเชิงวัตถุที่ใช้ (ODBMS) ในโครงงานได้แก่ **Caché** (อ่านว่า คา-เช่) พัฒนาโดย InterSystems Corporation ผู้จัดทำได้เลือกใช้ Caché เวอร์ชัน 4.0 ซึ่งเป็นระบบจัดการฐานข้อมูลที่สนับสนุนเทคโนโลยีเชิงวัตถุ (Object-Oriented Technology) สามารถจำลองโครงสร้างของข้อมูลที่มีความซับซ้อนให้คงสภาพตามที่เป็นจริงโดยไม่ต้องเปลี่ยนรูปร่างของโครงสร้างข้อมูล ทำให้ลดความสับสนในการกำหนดความสัมพันธ์ของข้อมูลไปได้มาก สำหรับการใช้งาน Caché สามารถทำงานได้บนหลากหลายแพลตฟอร์ม (Hardware Platform) และหลากหลายระบบปฏิบัติการ (Operating System) อาทิเช่น Window 95/98/NT, UNIX และ Linux เป็นต้น



รูปที่ 4-1 การติดต่อกับระบบจัดการฐานข้อมูล Caché

4.1 คุณสมบัติด้านต่าง ๆ ของ Caché

- Caché ใช้ ODBC ที่เรียกว่า Ultra Fast ODBC เพื่อให้สามารถใช้ภาษา SQL เป็นตัวติดต่อกับระบบฐานข้อมูลได้ และใช้ Visual Caché เพื่อให้ออบเจ็กต์ของ Caché สามารถใช้ได้กับ Visual Basic และเครื่องมือในการพัฒนา GUI อื่นๆ นอกจากนั้นยังมี Weblink ซึ่งทำให้สามารถเชื่อต่อฐานข้อมูล Caché ผ่านทางเครือข่ายอินเทอร์เน็ตได้ด้วย

- Caché Object Script เป็นส่วนที่ใช้ในการพัฒนาโปรแกรมให้เป็นแบบออบเจ็กต์ได้ เพื่อเพิ่มความยืดหยุ่นในการใช้งาน และยังสามารถใช้จาวาหรือ C++ ในการแก้ไขโปรแกรมได้อีกด้วย

- วิธีการเข้าถึงข้อมูล สามารถเข้าถึงข้อมูลได้ 3 แบบ ดังตารางที่ 4-1

- Transaction Database Engine ซึ่งเป็นตัวจัดการการทำงานของแอพพลิเคชั่น

- Distributed Caché Protocol ช่วยในการทำงานทางด้านการติดต่อของระบบเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Caché Multidimensional Access	เป็นการเข้าถึงข้อมูลแบบ Global variable ซึ่งจัดเก็บแบบ Multidimentional array
Caché SQL	เป็นเข้าถึงข้อมูลโดยผ่านทางภาษา SQL
Caché Objects	เป็นการเข้าถึงด้วย จาวา ซี++ และเทคโนโลยีในการพัฒนาทางด้านออบเจ็กต์อื่น ๆ

ตารางที่ 4-1 แสดงการเข้าถึงข้อมูลของ Caché

4.2 องค์ประกอบของ Caché

แสดงดังตารางที่ 4-2

Caché Object Server	ทำให้สามารถใช้งานออบเจ็กต์ของ Caché ได้โดยตรงผ่านทางจาวาหรือซี++
Caché SQL Server	สามารถเข้าถึงข้อมูลผ่านทาง SQL และ ODBC
Caché Studio	ใช้ในการพัฒนา GUI ในการกำหนดคลาสของออบเจ็กต์และสร้างส่วนประกอบของแอพพลิเคชัน
Caché Distributed Caché Protocol (DCP)	สามารถกระจายฐานข้อมูลทั่วระบบเครือข่ายในการตอบรับต่อความต้องการเพื่อให้ประสิทธิภาพดีที่สุด
Caché Object Script	เป็นภาษาในการเขียนโปรแกรมที่มีมาสำหรับแอพพลิเคชันด้านการประมวลผล ทรานแซคชั่นและควบคุมออบเจ็กต์ในระบบ
Caché Weblink	รองรับการเชื่อมต่อระหว่าง Caché กับเว็บเซิร์ฟเวอร์สำหรับประมวลผลทรานแซคชั่นผ่านทางเว็บ
Visual Caché	เป็นเครื่องมือสร้างฟอร์มวิชวลเบสิกของไมโครซอฟท์อย่างอัตโนมัติและยังให้การเชื่อมต่อระหว่างออบเจ็กต์ของ Caché กับวิชวลเบสิกได้อัตโนมัติ

ตารางที่ 4-2 แสดงส่วนประกอบของ Caché

4.3 Object Model

Caché เป็นระบบจัดการฐานข้อมูลเชิงวัตถุ ดังนั้นจึงจัดการและออกแบบระบบในเชิงวัตถุทั้งหมด ซึ่ง Caché Object Model สนับสนุนลักษณะดังต่อไปนี้

- คลาสหลักใน Caché จะเป็น Object Class และ Data Type Class (รวมทั้ง Literal ด้วย) โดย Object จะมีตัวชี้เฉพาะ (Unique Identifier) ส่วน Literal จะไม่มีตัวชี้เฉพาะ มีเพียงค่า (Value) เท่านั้น
- Object จะมีพฤติกรรมโดยอัตโนมัติซึ่งจัดการโดย Caché หรือถูกกำหนดโดยผู้ใช้ก็ได้
- Persistent Object สามารถถูกจัดเก็บในฐานข้อมูล Caché ได้

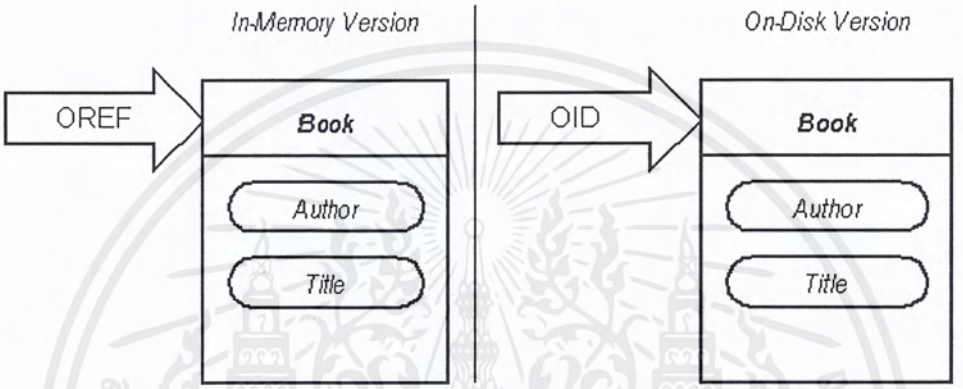
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอ้างอิงออบเจ็กต์

ในระบบจัดการของ Caché ออบเจ็กต์อาจอยู่ในดิสก์หรือในหน่วยความจำ ออบเจ็กต์ที่อยู่ในดิสก์จะเป็นออบเจ็กต์ที่ถูกจัดเก็บลงฐานข้อมูล ส่วนออบเจ็กต์ที่อยู่ในหน่วยความจำนั้นจะถูกโหลดดึงมาจากฐานข้อมูลและสามารถควบคุมได้ เราสามารถอ้างถึงออบเจ็กต์เหล่านี้ได้ 2 รูปแบบ ขึ้นอยู่กับว่าจะอ้างออบเจ็กต์นั้นหากมันถูกโหลดหรือถูกจัดเก็บอยู่ การอ้างอิงนั้นได้แก่

- OREF (Object Reference)
- OID (Object Identifier)

ดังแสดงในรูปที่ 4-2



รูปที่ 4-2 OREF และ OID

OREF เป็นการอ้างถึงออบเจ็กต์ที่อยู่ในหน่วยความจำ แต่ละครั้งที่ออบเจ็กต์ถูกโหลดขึ้นมาอยู่บนหน่วยความจำ มันอาจจะมีค่า OREF ที่แตกต่างกันได้

OID เป็นการอ้างถึงออบเจ็กต์ที่อยู่ในดิสก์ เป็นตัวชี้ออบเจ็กต์แบบ Persistent และมีค่าไม่ซ้ำ (Unique) ซึ่งถูกเก็บอยู่ในดิสก์ เมื่อออบเจ็กต์ถูกสร้างขึ้นและได้รับ OID แล้ว ก็จะมีค่านั้นไม่เปลี่ยนแปลง

เราสามารถใช้ OID ของออบเจ็กต์ Persistent นี้โหลดดึงออบเจ็กต์ขึ้นมาอยู่ในหน่วยความจำ จากนั้นระบบสามารถแทนออบเจ็กต์ตัวนั้นด้วย OREF ทำให้แอพพลิเคชันสามารถอ้างออบเจ็กต์และเข้าถึงค่าต่าง ๆ ของออบเจ็กต์ตัวนั้น ๆ ได้

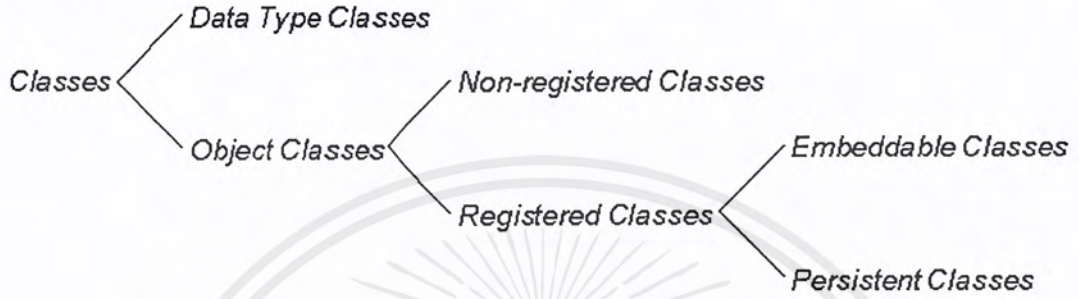
4.4 คลาส (Class)

คลาสใน Caché มี 5 ประเภท ได้แก่

- Registered Class
- Persistent Class
- Embeddable Class
- Non-Registered Class
- Data Type Class

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาสต่าง ๆ เหล่านี้ถูกจัดเป็น Data Type class และ Object class โดย Data Type class จะแทน Literal Value เช่น สตริง (String), ตัวเลข (Integer) หรือวันที่ (Date) เป็นต้น Object class ส่วนใหญ่จะสืบทอดคุณสมบัติมาจากคลาสของระบบที่เรียกว่า %RegisteredObject ซึ่งจัดการและมีเมธอดหลัก ๆ ของออบเจ็กต์ให้มา ส่วนคลาสที่ไม่ได้สืบทอดคุณสมบัติมาจาก %RegisteredObject เรียกว่า Non-Registed class ซึ่งคลาสชนิดนี้จะมีเพียงลักษณะพื้นฐานเท่านั้น เราจะต้องสร้างเมธอดให้แก่ออบเจ็กต์เหล่านี้เข้าไปเอง คลาสต่าง ๆ ใน Caché แสดงดังรูปที่ 4-3



รูปที่ 4-3 ประเภทของ Class

4.4.1 Registered Class

อินสแตนซ์ของคลาสนี้สืบทอดคุณสมบัติมาจาก %RegisteredObject หรือเรียกว่าเป็น registered object โดยออบเจ็กต์เหล่านี้จะมีเซตของเมธอดพื้นฐานทั้งหมดที่ใช้จัดการออบเจ็กต์ที่อยู่ในหน่วยความจำ ครั้งเมื่อ Registered object นี้ถูกโหลดสู่หน่วยความจำ เราสามารถอ้างถึงออบเจ็กต์นี้โดยผ่าน OREF ได้

4.4.2 Persistent Class

อินสแตนซ์ของคลาสนี้สืบทอดคุณสมบัติมาจาก %Persistent และถูกเรียกว่า persistent object ซึ่งออบเจ็กต์เหล่านี้เป็น registered object ด้วย ออบเจ็กต์นี้มีความสามารถที่จะจัดการและเก็บตัวมันเองลงในฐานข้อมูลได้ จากถูกโหลดสู่หน่วยความจำ ออบเจ็กต์อื่นที่มันอ้างถึงจะไม่ถูกโหลดมาด้วย เพราะออบเจ็กต์เป็นอิสระต่อกันและเพียงแต่ชี้หากันเท่านั้น

หาก persistent object ถูกใช้เป็น Attribute หนึ่งของออบเจ็กต์อื่น มันจะถูกชี้มากจากออบเจ็กต์นั้น ๆ ยกตัวอย่างเช่น ให้ Doctor เป็น persistent object และเรากำหนดให้ Attribute ในออบเจ็กต์ตัวหนึ่งมีชนิดเป็น Doctor จะได้เป็น

```
ATTRIBUTE TheDoc {TYPE = Doctor ;}
```

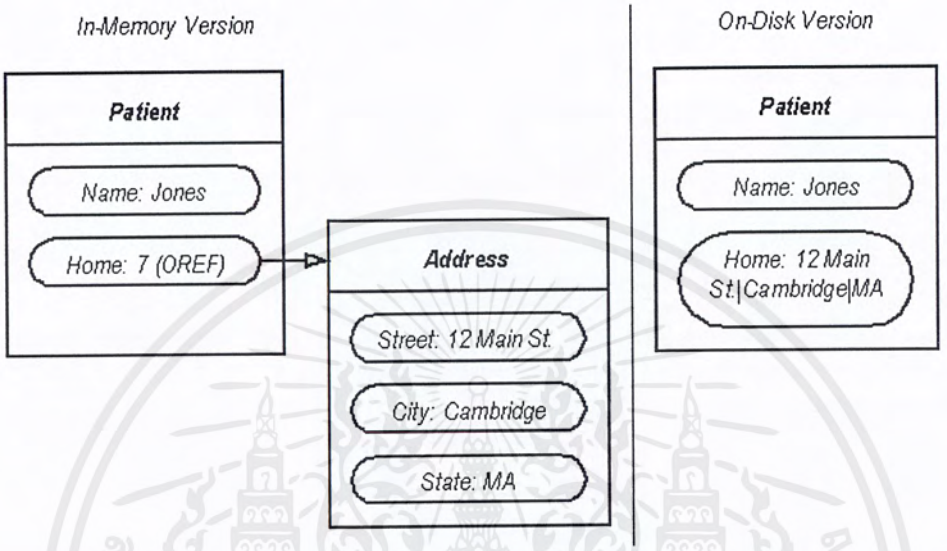
ซึ่ง Attribute TheDoc จะชี้ไปยังออบเจ็กต์ชนิด Doctor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.3 Embeddable Class

อินสแตนซ์ของคลาสนี้สืบทอดคุณสมบัติมาจาก %SerialObject และถูกเรียกว่า Embedded object ซึ่งออบเจ็กต์เหล่านี้อ้างอิงได้อย่างอิสระเมื่อถูกโหลดสู่หน่วยความจำ แต่เมื่อจัดเก็บลงฐานข้อมูลแล้ว ออบเจ็กต์เหล่านี้จะต้องฝังตัวอยู่ภายใน persistent object เท่านั้น

Embedded object มีรูปแบบ 2 ลักษณะแสดงดังรูปที่ 4-4 ได้แก่



รูปที่ 4-4 Embedded Object ในหน่วยความจำและดิสก์

- ในหน่วยความจำ, Embedded object ถูกแทนด้วยออบเจ็กต์อิสระคล้าย ๆ กับ persistent object หาก Attribute ใดอ้างชนิดเป็น Embedded object แล้วการจัดการในหน่วยความจำก็สามารถกำหนดเป็น OREF ได้
- ในดิสก์, Embedded object ถูกจัดเก็บเป็นเหมือน Attribute หนึ่งและถูกบรรจุลง persistent object ตัวหนึ่ง โดยมันจะไม่มี OID เป็นของตัวเองและไม่สามารถถูกอ้างโดยออบเจ็กต์ตัวอื่น ๆ อีกได้เลย ค่าของ Embedded object จะถูกเก็บลงฐานข้อมูลคล้าย ๆ เป็นเพียงสตริงใน Attribute ของออบเจ็กต์หลักเท่านั้น เหตุนี้จึงเรียกว่าเป็น SerialObject

ยกตัวอย่างเช่น

```
CLASS Address
{
        SUPER = %SerialObject ;
        ATTRIBUTE Street { TYPE = %String(MAXLEN=80) ; }
        ATTRIBUTE City { TYPE = %String(MAXLEN=3) ; }
        ATTRIBUTE State { TYPE = %String(MAXLEN=2) ; }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อกำหนดให้ออบเจ็กต์จากคลาส Address เป็น Embedded object attribute สำหรับ persistent object เช่นคลาส Patient จะได้

```
CLASS Patient
{
        SUPER = %Persistent ;
        PERSISTENT ;
        ATTRIBUTE Name { TYPE = %String ; }
        ATTRIBUTE Home { TYPE = Address ; }
}
```

4.4.4 Non-Registered Class

อินสแตนซ์ของคลาสนี้เป็นออบเจ็กต์ที่เหลือ ที่ไม่ได้สืบทอดคุณสมบัติมาจากคลาส %RegisteredObject อาจเกิดขึ้นจากการโปรแกรมของผู้ใช้ขึ้นเอง และเราจำเป็นต้องกำหนดค่าสำหรับ OREF ขึ้นมาเองเพราะระบบไม่ได้เตรียมหรือ register ค่าให้

4.4.5 Data Type Class

คลาสนี้กำหนดและควบคุมค่าของ Literal โดย data type ไม่มีตัวชี้อิสระเหมือนออบเจ็กต์ทั่วไปและไม่สามารถมีอินสแตนซ์ได้ ซึ่งมันเป็นเพียง Attribute ของออบเจ็กต์ซึ่งบรรจุมันเอาไว้ และตัวมันเองก็ไม่มี Property ด้วย Data Type class จะแทน Literal Value เช่น สตริง (String), ตัวเลข (Integer), วันที่ (Date), ตัวเลขทศนิยม (Float) และชนิดของข้อมูลพื้นฐานต่าง ๆ

4.5 Caché กับ Inheritance

Caché object model อนุญาตให้เราสามารถออกแบบคลาสด้วยการสืบทอดคุณสมบัติจากคลาสอื่นมาได้โดยการใช้คีย์เวิร์ด SUPER คลาสที่ได้จะมี specification เหมือนกับ Superclass ทุกประการรวมทั้ง Property, Method และ Class Parameter ด้วย ยกตัวอย่างเช่นคลาส Person และมีคลาส Employee มาทำการสืบทอดคุณสมบัติจากคลาส Person ดังนี้

```
CLASS Person
{
        SUPER = %Persistent ;
        PERSISTENT ;
        ATTRIBUTE Name { TYPE = %String ; }
        ATTRIBUTE DOB { TYPE = %Date ; }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
CLASS Employee
{
        SUPER = Person ;
        PERSISTENT ;
        ATTRIBUTE Salary { TYPE = %Integer ; }
        ATTRIBUTE Department { TYPE = %String ; }
}
```

4.6 Caché กับ Multiple Inheritance

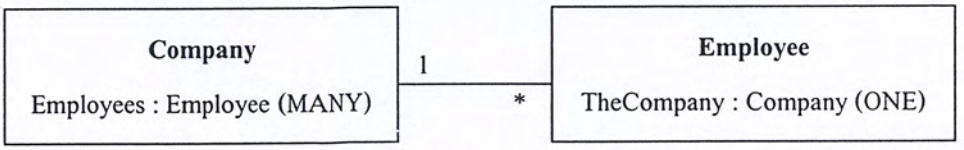
คลาสสามารถสืบทอดคุณสมบัติมาได้จากหลาย ๆ Superclass โดยใช้คีย์เวิร์ด SUPER เช่นเดียวกัน เมื่อทำการคอมไพล์คลาส หากพบว่าสมาชิกต่าง ๆ เช่น ชื่อเมธอดของ Superclass มีชื่อซ้ำกัน Caché จะกำหนดให้คลาสที่สืบทอดลำดับหลังถูกเรียกใช้งานก่อน ยกตัวอย่างเช่น คลาส X ทำการสืบทอดคุณสมบัติจากคลาส A, B และ C จะได้ว่า

```
CLASS X
{
        SUPER = A, B, C ;
}
```

ในกรณีนี้ค่าต่าง ๆ ของคลาส X จะสืบทอดคุณสมบัติและค่าต่าง ๆ จากคลาส A, B และ C ตามลำดับ (ดูจากลิสต์ก่อนหลัง) หากพบว่าคลาส B มีสมาชิกชื่อซ้ำกับคลาส A ที่ทำการสืบทอดคุณสมบัติมาแล้ว สมาชิกจากคลาส A จะถูกทับ (Override) ด้วยสมาชิกของคลาส B ทันที

4.7 Caché กับ Relationship

ความสัมพันธ์ (Relationship) เป็น Property ชนิดพิเศษซึ่งกำหนดความเกี่ยวข้องกันของออบเจ็กต์สองตัว เช่น คลาส Company มีความสัมพันธ์กับคลาส Employee ดังรูปที่ 4-5 ในกรณีนี้อาจมีออบเจ็กต์ Employee ศูนย์หรือมากกว่าหนึ่งตัวที่เกี่ยวข้องกับออบเจ็กต์ Company แต่ละตัว



รูปที่ 4-5 ความสัมพันธ์ระหว่างคลาส Company และคลาส Employee

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ Caché เวอร์ชัน 4.0 นี้ ความสัมพันธ์มีคุณลักษณะหลัก ๆ ดังนี้

- ความสัมพันธ์ถูกกำหนดขึ้นระหว่าง Persistent class 2 คลาสเท่านั้น
- ความสัมพันธ์เป็น bi-directional ซึ่งต้องกำหนดความสัมพันธ์ในทั้ง 2 ฝั่งคลาส
- สนับสนุนความสัมพันธ์แบบ One-to-Many

การกำหนดความสัมพันธ์

มีคีย์เวิร์ดที่เกี่ยวข้องอยู่ 3 ตัว ได้แก่

- TYPE เป็นชนิด (ชื่อคลาส) ของคลาสที่เกี่ยวข้อง และต้องเป็น Persistent Class
- INVERSE ชื่อของความสัมพันธ์ของอีกคลาสที่มีความสัมพันธ์ต่อกัน
- CARDINALITY มีค่าเป็น ONE หรือ MANY

จากรูปที่ 4-5 ได้ดังนี้

```
CLASS Company {
   SUPER = %Persistent ;
   PERSISTENT ;
   ATTRIBUTE Name { TYPE = %String ; }

   // a Company has MANY Employees
   RELATIONSHIP Employees { TYPE = Employee; INVERSE = the company; CARDINALITY
= MANY;}
}

CLASS Employee {
   SUPER = %Persistent ;
   PERSISTENT ;
   ATTRIBUTE Name { TYPE = %String ; }
   ATTRIBUTE Title { TYPE = %String ; }

   // an Employee has ONE Company
   RELATIONSHIP TheCompany { TYPE = Company; INVERSE = Employees; CARDINALITY
= ONE;}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8 Caché กับ Application

ในส่วนของการ Implementing the Application นั้น Caché ได้แบ่งออกเป็นขั้นตอนใหญ่ ดังนี้

- Setting Up a Namespace เป็นการกำหนดพื้นที่ในการจัดเก็บข้อมูลของ Application โดยที่จะเป็นการกำหนดทาง Logical เท่านั้น
- Creating the Application and its Classes สามารถสร้างคลาสโดยใช้ Caché Object Architecture สามารถกำหนดชื่อ คำอธิบายและชนิดของคลาสได้ สำหรับ Caché จะมี 3 คลาสที่ใช้ประจำ คือ Persistent Class (ออบเจ็กต์ที่ได้จะเป็นอิสระ เก็บข้อมูลได้เอง) , Embeddable Class (ออบเจ็กต์ของคลาสนี้จะอยู่ในออบเจ็กต์ของคลาสอื่นอีกที) และสุดท้ายคือ Data type Class เป็นคลาสที่ไม่ใช้เก็บข้อมูลแต่มีไว้เพื่อควบคุมการทำงานของแอพพลิเคชัน
- Properties เป็นการระบุข้อมูลของคลาสที่ได้สร้างไว้แล้วซึ่งจะต้องระบุชนิดของข้อมูลและเงื่อนไขต่าง ๆ ของ Property ไว้ด้วยเช่น unique , indexed , require เป็นต้น ขั้นตอนนี้ใช้ Caché Object Architecture ช่วยในการทำงาน
- Method หลังจากกำหนดค่า Properties ของคลาสแล้ว จะต้องทำการกำหนดเมธอดของคลาสต่าง ๆ ขึ้นมาโดยใช้ภาษา Caché ObjectScript เขียนขึ้นมาว่าจะให้คลาสที่เรากำหนดมานั้นสามารถทำงานหรือมีพฤติกรรมอย่างไรบ้าง
- Creating Query สร้างขึ้นเพื่อแสดงข้อมูลที่มีอยู่ในฐานข้อมูลของแอพพลิเคชันแทนที่จะเข้าไปดูข้อมูลโดยตรงกับออบเจ็กต์แต่ละตัวผ่านทาง OID โดยใช้ Caché SQL Query หรือ Caché ObjectScript

ในส่วนของการสร้าง Appilcation Front-end สำหรับผู้ใช้นั้น สามารถทำได้โดยใช้ ActiveX, C++, Delphi, PowerBuilder ซึ่งจะติดต่อกับ Caché ผ่านทาง Caché ObjectServer แต่วิธีที่นิยมและใช้งานง่ายที่สุดคือการสร้างผ่านทาง Visual Basic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สถาปัตยกรรมของระบบ

5.1 บทนำ

สำหรับในบทที่ผ่านๆ มานั้นจะเป็นการอธิบายถึงทฤษฎีต่างๆ ที่เกี่ยวข้อง และตัวอย่างของการประยุกต์ใช้ระบบผู้เชี่ยวชาญร่วมกับฐานข้อมูลบางแบบไปแล้ว สำหรับในบทนี้จะเป็นการนำเสนอและอธิบายรูปแบบสถาปัตยกรรมที่ใช้งานในโครงงานนี้ โดยก่อนอื่นในส่วนของบทนี้ จะขออธิบายสรุปถึงรูปแบบสถาปัตยกรรมต่างๆ ที่มีอยู่เสียก่อนในหัวข้อต่อไปของบทนี้ หลังจากนั้นจึงค่อยนำเสนอรูปแบบสถาปัตยกรรมที่เลือกใช้ รวมไปถึงหลักในการพิจารณาต่างๆ ที่เกี่ยวข้องเมื่อได้เลือกใช้รูปแบบสถาปัตยกรรมที่โครงงานได้นำเสนอ และตัวอย่างการนำไปใช้งานจริงด้วย

5.2 รูปแบบสถาปัตยกรรมแบบต่างๆ

สำหรับรูปแบบการแยกสถาปัตยกรรมของระบบที่มีการใช้ระบบผู้เชี่ยวชาญร่วมกันกับระบบจัดการฐานข้อมูลนั้น มีการจัดไว้อยู่หลายรูปแบบด้วยกัน ดังที่ได้มีการอธิบายย่อๆ ไว้จากบทก่อนๆ แล้ว สำหรับในส่วนนี้จะเป็นการอธิบายในส่วนที่เป็นรายละเอียดมากขึ้น โดยหลักการแบ่งรูปแบบจะแบ่งออกเป็นสองแบบใหญ่ๆ ด้วยกัน คือ

- แบบที่สร้างให้ระบบผู้เชี่ยวชาญและระบบฐานข้อมูลอยู่แยกจากกัน (Heterogeneous approach)
- แบบที่สร้างให้ระบบผู้เชี่ยวชาญและระบบฐานข้อมูลรวมอยู่ด้วยกัน (Homogeneous approach)

โดยในแต่ละรูปแบบสถาปัตยกรรมจะมีรายละเอียดต่างๆ เพิ่มเติมดังที่จะได้อธิบายดังนี้

5.2.1 สถาปัตยกรรมแบบแยกส่วน

สำหรับรูปแบบวิธีการแบบนี้จะเป็นการแยกส่วนของระบบผู้เชี่ยวชาญและฐานข้อมูลออกจากกัน โดยรูปแบบที่นิยมใช้กันก็คือการสร้างระบบผู้เชี่ยวชาญเป็นส่วนที่ทำการติดต่อกับผู้ใช้ ในส่วนนี้ซึ่งจะเก็บความรู้ประเภทกฎ ไว้ทั้งหมด และเมื่อมีการอ้างถึงความรู้ประเภทข้อเท็จจริงที่มีการเก็บไว้ในฐานข้อมูล ทางตัวระบบผู้เชี่ยวชาญจะมีการร้องขอข้อมูลจากฝั่งของฐานข้อมูลให้ส่งข้อมูลมา ซึ่งก็เปรียบได้กับการปรับปรุงส่วนของระบบผู้เชี่ยวชาญให้มีความสามารถในการเรียกใช้ข้อมูลจากหน่วยความจำสำรองนั่นเอง เรียกวิธีแบบนี้ว่าเป็นการรวมกันแบบหลวมๆ (Loosely Coupled)

อย่างไรก็ตาม ในการรวมแบบวิธีนี้ก็มีอุปสรรคอันเนื่องมาจากความแตกต่างของทฤษฎีและกลไกในการทำงานของระบบทั้งสองแบบที่แตกต่างกันเล็กน้อย คือสำหรับในทฤษฎีที่ใช้ในระบบผู้เชี่ยวชาญนั้น เป็นรูปแบบที่มีอัลกอริธึมแบบ Interpreter สำหรับทำงานกับข้อเท็จจริงเพียงทีละตัว (Tuple-Oriented Algorithm) และลำดับ (Order) ของการประมวลหาข้อเท็จจริงก็มีผลต่อการคำนวณหาคำตอบด้วย ในขณะที่ระบบฐานข้อมูลนั้น ตามทฤษฎีแล้ว การเก็บข้อมูลและการประมวลผลจะอยู่รูปของเซ็ต (Set) หรือรีเลชัน (Relation) ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประกอบด้วยข้อมูลที่อาจมีมากกว่าหนึ่งตัวและไม่มีการคำนึงถึงลำดับของกลุ่มข้อมูล การประมวลผลหาคำตอบของคำถามที่รับเข้ามาจึงเป็นการประมวลผลที่อิงกับข้อมูลชนิดเซ็ต (Set-Oriented Algorithm) ทางผู้ที่ทำการสร้างระบบโดยยึดเอาสถาปัตยกรรมรูปแบบนี้จึงต้องคิดวิธีการในการกำจัดข้อขัดแย้งที่เกิดขึ้นนี้เพื่อให้ระบบทั้งสองสามารถทำงานด้วยกันได้ แต่ผลที่ตามมาก็คือระบบจะมีประสิทธิภาพในการทำงานที่แย่มากเนื่องจากต้องมีการติดต่อกับแหล่งข้อมูลภายนอกทุกครั้งที่มีการต้องการเรียกใช้ข้อเท็จจริงเพื่อทำการประมวลผลซึ่งก็หมายความว่าต้อง มีการดึงข้อมูลจากดิสก์ซึ่งจะมีการทำงานที่ช้ามากเมื่อพิจารณาว่าเป็นกระบวนการที่ทำเพียงเพื่อดึงเอาข้อเท็จจริงเพียงตัวเดียวมาใช้ จะเห็นได้เลยว่าเป็นแนวทางที่ไม่คุ้มค่าเลย ดังนั้น หลักสำคัญของสถาปัตยกรรมในรูปแบบนี้จึงอยู่ที่การออกแบบส่วนที่ทำหน้าที่เชื่อมต่อ (Interface) กันระหว่างระบบทั้งสองเพื่อแก้ปัญหาที่เกิดขึ้นดังที่ได้อธิบายไว้แล้วให้มีการทำงานที่ก่อให้เกิดประสิทธิภาพสูงที่สุด แนวทางที่นำมาใช้ในการแก้ปัญหาจะแบ่งออกเป็น 2 แนวทางหลักๆ คือ แบบการใช้ Interpreter และ แบบการใช้ Compiler ซึ่งก็อาจมีบางระบบที่จะใช้แนวทางทั้งสองอย่างนี้ผสมผสานกัน

5.2.1.1 แนวทางแบบการใช้ Interpreter

ในแนวทางนี้จะมีการสร้างการกันหน่วยความจำหลักไว้ส่วนหนึ่งสำหรับเก็บข้อมูลที่ได้เรียกเข้ามาไว้ชั่วคราว โดยข้อมูลนี้อาจเป็นคำตอบของปัญหาย่อยๆ ที่จำเป็นต้องเก็บไว้ใช้ต่อไป ซึ่งถ้าระบบผู้เชี่ยวชาญต้องการใช้ข้อมูลนี้อีกครั้งก็ไม่ต้องเสียเวลาในการหาจากระบบฐานข้อมูลอีก

5.2.1.2. แนวทางแบบการใช้ Compiler

ในแนวทางนี้จะทำการดูรูปแบบของกฎก่อนว่าในการอนุมานคำตอบของกฎนั้น มีข้อเท็จจริงอะไรบ้างที่จำเป็นต้องดึงออกมาจากฐานข้อมูลภายนอก และจะพยายามรวบรวมข้อเท็จจริงที่ต้องการให้ได้มากที่สุดและส่ง Query ให้มีจำนวนน้อยที่สุด เพื่อให้จำนวนครั้งของการติดต่อกับฐานข้อมูลภายนอกเกิดขึ้นน้อยที่สุด วิธีนี้จะทำให้การประสิทธิภาพการทำงานของระบบเพิ่มขึ้นอย่างมาก แต่ในทางปฏิบัติแล้ว เป็นไปได้ยากที่จะสามารถทำการคอมไพล์กฎทั้งหมดให้เสร็จก่อนแล้วค่อยทำการส่ง Query ไป และถ้ามีการออกแบบที่ไม่ดีก็อาจทำให้เกิดการทำงานแบบ Infinite Loop ได้

5.2.2 สถาปัตยกรรมแบบรวมเป็นส่วนเดียว

ในระบบแบบนี้จะเป็นการพยายามรวมเอาส่วนของที่เก็บข้อมูลข้อเท็จจริงและส่วนที่เป็นกฎไว้ เรียกว่าเป็นการรวมกันอย่างเหนียวแน่น (Tightly-couple) ซึ่งจากการที่ในช่วงหลังๆ เทคโนโลยีทางด้านระบบจัดการฐานข้อมูลเชิงสัมพันธ์ได้มีการพัฒนาขึ้นเร็วกว่าทางด้านระบบผู้เชี่ยวชาญมาก มีการนำระบบไปประยุกต์ใช้งานในเชิงธุรกิจขนาดใหญ่อย่างได้ผลและมีการพัฒนาต่อให้สามารถทำงานเป็นระบบเครือข่าย เช่น โครงสร้างแบบ Client-Server ทำให้ในช่วงหลังๆ รูปแบบการพัฒนาของระบบฐานข้อมูลผู้เชี่ยวชาญส่วนใหญ่ได้พัฒนาให้อยู่ในแบบของสถาปัตยกรรมนี้ ซึ่งก็ได้เป็นรูปแบบหลักที่มีการพัฒนาและวิจัยอย่างต่อเนื่องและสม่ำเสมอมากกว่าแบบอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบสถาปัตยกรรมแบบนี้จะเป็นที่รู้จักกันทั่วไปในชื่อของระบบฐานข้อมูลอนุมาน (Deductive database) หรือระบบฐานข้อมูลชาญฉลาด (Intelligent Database) โดยสถาปัตยกรรมแบบนี้สามารถแบ่งย่อยลงไปได้อีกสองแบบตามวิธีการที่ใช้ในการพัฒนา คือ

5.2.2.1 แบบทำการขยายส่วนของระบบผู้เชี่ยวชาญให้มีความสามารถของระบบฐานข้อมูล (Elementary Deductive Database)

เป็นการสร้างส่วนขยายเพิ่มเติมจากภาษา Prolog ให้มีความสามารถในการจัดการด้านฐานข้อมูลได้เช่นเดียวกับที่ระบบจัดการฐานข้อมูลมี ได้แก่ ส่วนของภาษาที่ใช้ในการกำหนดรูปแบบของข้อมูล (Data Definition Language : DDL), ส่วนที่เกี่ยวข้องกับการดูแลความปลอดภัยของข้อมูล (Security), ส่วนที่เกี่ยวข้องกับความถูกต้องของข้อมูลและข้อบังคับในการจัดเก็บข้อมูล (Data integrity), ความสามารถในการประมวลผลข้อมูลหลายๆ กระบวนการในเวลาเดียวกัน (Concurrency), ความสามารถในการให้บริการผู้ใช้ได้หลายๆ คนพร้อมกัน (User sharing), ความสามารถในการสำรองข้อมูลและกู้ข้อมูลกลับคืนมากรณีที่ระบบเกิดความเสียหาย (Backup and recovery) ซึ่งนับว่าเป็นงานที่ค่อนข้างหนักทีเดียวในการสร้างระบบผู้เชี่ยวชาญให้มีความสามารถเหล่านี้โดยมีประสิทธิภาพที่ใกล้เคียงหรือเทียบเท่ากับระบบจัดการฐานข้อมูล

5.2.2.2 แบบทำการขยายส่วนของระบบฐานข้อมูลให้มีความสามารถในการอนุมาน (Advanced Deductive Database)

จากการที่ระบบผู้เชี่ยวชาญสามารถตอบคำถามบางรูปแบบซึ่งไม่สามารถหาคำตอบได้ด้วยการใช้ Query ของระบบฐานข้อมูลแบบ Relational ทั่วๆ ไปได้ โดยเฉพาะคำถามแบบ Recursive ทำให้มีการพยายามขยายความสามารถของระบบฐานข้อมูลที่มีอยู่เดิมให้มีความสามารถในส่วนนี้

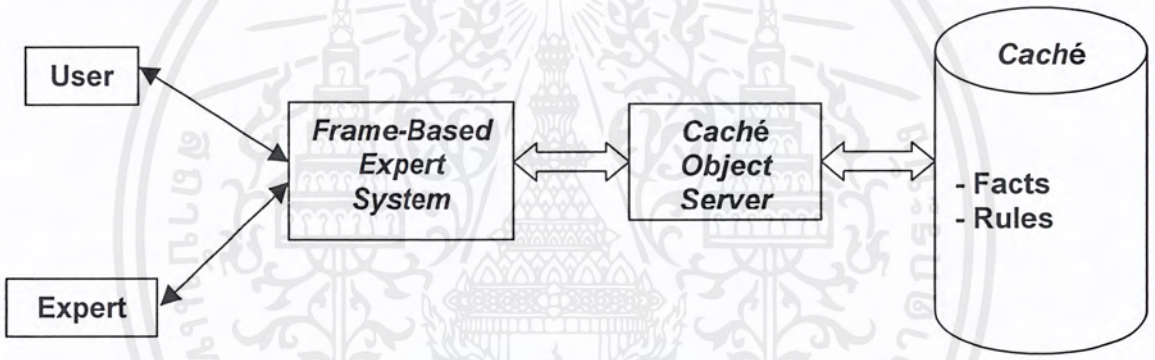
แต่เนื่องจากข้อจำกัดในเรื่องของรูปแบบทฤษฎีของระบบทั้งสองดังที่ได้อธิบายไว้ในข้างต้นแล้ว ทำให้เราไม่สามารถใช้ทฤษฎีทางตรรกศาสตร์ที่ใช้ในระบบผู้เชี่ยวชาญ คือ First-Order Predicate Logic ในการกำหนดส่วนของกฎที่ใช้อนุมานโดยตรงกับฐานข้อมูลได้ จึงมีการดัดแปลงทฤษฎีตรรกศาสตร์ดังกล่าวให้มีความสอดคล้องกับทฤษฎีของแบบจำลองข้อมูลแบบ Relational เพื่อสร้างเป็นภาษาที่ใช้ในการจัดการฐานข้อมูล โดยอาจมีการลดความสามารถทางด้านตรรกศาสตร์ลงด้วย เช่น Datalog ซึ่งมีลักษณะของภาษาตามที่อธิบายไว้ก่อนหน้านี้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
สถาปัตยกรรมและการออกแบบในโครงงาน

6.1 บทนำ

สำหรับในเนื้อหาของส่วนที่ผ่าน ๆ มาซึ่งได้กล่าวถึงแนวทางที่ใช้ในการออกแบบเรียบร้อยแล้ว สำหรับในบทนี้จะเป็นการนำรูปแบบสถาปัตยกรรมที่ใช้ในการติดต่อกันระหว่างระบบผู้เชี่ยวชาญและระบบจัดการฐานข้อมูลที่ได้ออกแบบไว้แล้วมาใช้ในการสร้างระบบต้นแบบเพื่อที่จะทำการทดสอบการทำงานและพิสูจน์ความถูกต้องของรูปแบบสถาปัตยกรรมการติดต่อระหว่างฐานข้อมูลและระบบผู้เชี่ยวชาญ โดยในบทนี้จะเป็นเนื้อหาจะเริ่มที่ส่วนของการพิจารณาวิธีที่จะใส่เฟรมเข้าไปในส่วนของระบบจัดการฐานข้อมูลก่อน จากนั้นก็จะพิจารณาถึงวิธีการที่จะทำให้สร้างระบบผู้เชี่ยวชาญตามที่เราได้ออกแบบไว้สามารถทำงานได้โดยจะพิจารณาในทุกๆ ส่วนทั้งในด้านของระบบผู้เชี่ยวชาญและด้านระบบจัดการฐานข้อมูล



รูปที่ 6-1 สถาปัตยกรรมและขั้นตอนการติดต่อกันระหว่างระบบย่อย

6.2 ส่วนของระบบผู้เชี่ยวชาญ (Frame-Based Expert System)

สำหรับในระบบผู้เชี่ยวชาญจะเป็นโปรแกรมที่สร้างขึ้นมาโดยมีลักษณะเป็นระบบผู้เชี่ยวชาญแบบ Frame-Based ซึ่งในโปรแกรมจะประกอบไปด้วยโครงสร้างของความรู้(โครงสร้างของความรู้จะประกอบไปด้วยความรู้และกฎต่าง ๆ) และ ข้อเท็จจริง ต่าง ๆ

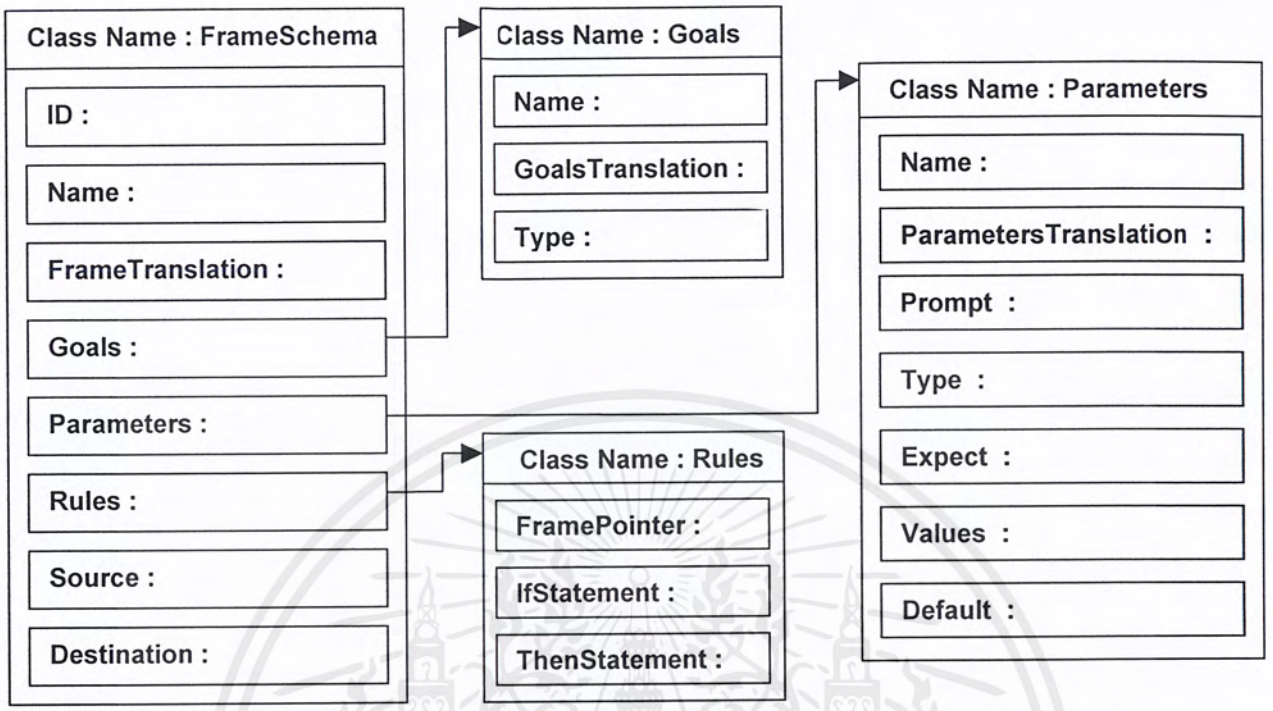
6.2.1 โครงสร้างของฐานความรู้

จากลักษณะโครงสร้างการแสดงความรู้แบบ Frame เราสามารถนำมาประยุกต์บนระบบจัดการฐานข้อมูลเชิงวัตถุได้โดยมีลักษณะดังนี้

- Frame จะเป็น Instance ของ Object
- Frame จะประกอบไปด้วย Attribute ต่าง ๆ ซึ่ง Attribute ต่าง ๆ จะเปรียบได้กับ Slots ต่าง ๆ
- Frame จะประกอบไปด้วย Rules ต่าง ๆ ซึ่งแต่ละ Rules จะชี้ไปยัง Frame ต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะโครงสร้างของ Frame จะเป็นดังรูปที่ 6-2



รูปที่ 6-2 โครงสร้างของ Frame

รายละเอียดของ Frame

- **FrameSchema Class** เป็น Persistent Class ที่จะมี ID เป็นของตัวเอง ซึ่งจะเก็บทุก Frame ในฐานความรู้
  - **ID** Type **Number** เป็น ID ที่มีไว้อ้างอิงซึ่งสร้างโดยตัวของ Caché เอง
  - **Name** Type **String** เก็บชื่อของ Frame
  - **FrameTranslation** Type **String** เก็บความหมายของ Frame
  - **Goals** Type **Goals** เป็น Embeddable Class ที่จะฝังตัวอยู่ใน Persistent ซึ่งไม่มี ID เป็นของตัวเอง
    - **Name** Type **String** เก็บชื่อของ Goals
    - **GoalsTranslation** Type **String** เก็บความหมายของ Goals
    - **Type** Type **String** ชนิดของ Goals
  - **Parameters** Type **Parameters**
    - **Name** Type **String** ชื่อของ Parameters
    - **ParametersTranslation** Type **String** ความหมายของ Parameters
    - **Prompt** Type **String** คำแนะนำของ Parameters
    - **Type** Type **String** ชนิดของ Parameters
    - **Expect** Type **String** ชนิดของค่าที่เป็นไปได้ของ Parameters
    - **Values** Type **String** ค่าที่เป็นไปได้ของ Parameters

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Default** Type **String** ค่าเริ่มต้นของ Parameters

- **Rules** Type **Rules**

  - **FramePointer** Type **String** เก็บชื่อของ Frame ที่ถูกชี้โดย ThenStatement

  - **IfStatement** Type **String** เก็บ Statement ในส่วนของ IF

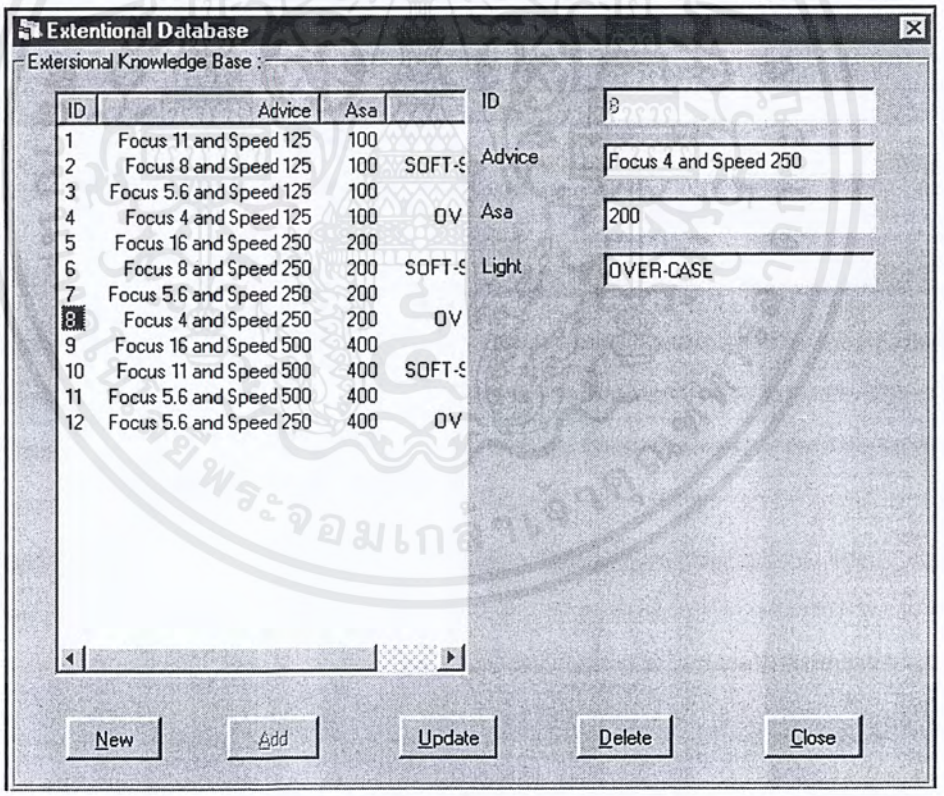
  - **ThenStatement** Type **String** เก็บ Statement ในส่วนของ Then

- **Source** Type **String** บอก Frame ที่เกี่ยวข้องกัน

- **Destination** Type **String** บอกลักษณะของ Frame ว่าเป็น Own หรือ Member ของ Source

6.2.2 ข้อเท็จจริง

ระบบผู้เชี่ยวชาญจะมีข้อเท็จจริงไว้บอ กถึงค่าที่เป็นจริงที่ผู้เชี่ยวชาญจะเป็นผู้กำหนดไว้ให้ ทำให้ระบบผู้เชี่ยวชาญสามารถตอบคำถามของผู้ใช้ระบบได้ในกรณีที่ไม่สามารถหาคำตอบจากกฎของระบบ ซึ่งข้อเท็จจริงนั้นจะมีหรือไม่มีในระบบผู้เชี่ยวชาญก็ได้ ลักษณะของข้อเท็จจริงจะเก็บเป็น Class หนึ่ง Class ซึ่งประกอบไปด้วย Attributes ต่าง ๆ ตามที่ผู้เชี่ยวชาญเป็นคนกำหนด หรือสามารถที่จะนำฐานข้อมูลข้อเท็จจริงจากภายนอกเข้ามาใช้งานก็ได้ ตัวอย่างของข้อเท็จจริงแสดงได้ดังรูปที่ 6-3



รูปที่ 6-3 แสดงตัวอย่างของข้อเท็จจริงในระบบผู้เชี่ยวชาญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.3 กฎ

กฎจะประกอบไปด้วยกฎที่บอกคำตอบให้กับผู้ใช้งานระบบและกฎที่บอกว่าจะต้องไป Frame ไหนต่อ ตัวอย่างของกฎที่บอกคำตอบให้กับผู้ใช้งานระบบก็เช่น

If Distance > 12 Then Darks = "Set Focus 11 and Speed 125"

ตัวอย่างของกฎที่บอกว่าจะต้องไป Frame ไหนต่อเช่น

If Light = Dark Then Advice="Set Focus 11 and Speed 250" AND CONSIDERFRAME Dark

ซึ่งในโครงงานนี้จะใช้คำว่า CONSIDERFRAME เป็นคำเฉพาะในกรณีที่ต้องการให้กฎชี้ไปยัง Frame ใด ๆ โดยจะต้องใส่ชื่อ Frame ที่ถูกชี้ไปต่อท้ายด้วย

6.2.4 กลไกการอนุมาน

ในโครงงานนี้ได้เลือกใช้กลไกในการอนุมานด้วยวิธีการอนุมานแบบเดินหน้า (Forward Chaining). ซึ่งจะเป็นการอนุมานจากความจริงที่มีอยู่เพื่อจะหาค่าที่ต้องการ ซึ่งหลักการในการอนุมานแบบเดินหน้านั้นแสดงได้ดังรูปที่ 6-4

รูปที่ 6.4 แสดงกลไกการอนุมานแบบเดินหน้า

ตัวอย่างในการกลไกการอนุมานแบบเดินหน้าเช่น

1) light-condition = soft-shadow

2) asa = 200

3) if light-condition = bright-sun and asa = 100
   then advice = "Aperture = f11 speed = 125"

4) if light-condition = soft-shadow and asa = 100
   then advice = "Aperture = f8 speed = 125"

5) if light-condition = bright-sun and asa = 200
   then advice = "Aperture = f11 speed = 250"

6) if light-condition = soft-shadow and asa = 200
   then advice = "Aperture = f8 speed = 250"

เมื่อกลไกการอนุมานพบความจริงในข้อ 1) และ 2) ความจริงนี้จะถูกนำเข้าไปเก็บไว้ในคลังความรู้ และกลไกการอนุมานจะทำการเปรียบเทียบค่าของนิพจน์ต่าง ๆ ตามลำดับที่อยู่ในฐานความรู้ ในที่นี้จะเริ่มด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเปรียบเทียบนิพจน์ light-condition กับกฎข้อต่าง ๆ ในที่นี้กฎข้อ 4) และ 6) จะมีค่าตามความจริงที่มีอยู่ เมื่อได้ค่าของ light-condition แล้วกลไกการอนุมานก็จะเปรียบเทียบค่าของ asa จากความจริง (2) asa = 200) กับกฎข้อ 4) และ 6) ซึ่งปรากฏว่าค่าที่ใช้ได้คือ ข้อ 6) ดังนั้นจะได้ค่าของ advice="Aperture = f8 and speed 250" ค่านี้จะถูกนำไปเก็บไว้ในคลังความรู้และแสดงผลออกมาทางหน้าจอ

6.3 การอ้างอิงถึงฐานความรู้

ฐานความรู้นั้นจะประกอบไปด้วยส่วนสำคัญ 2 ส่วนคือคือ Frame และข้อเท็จจริงต่าง ๆ ในที่นี้จะขอกล่าวถึงการใช้งานเพียงลักษณะคร่าว ๆ ส่วนรายละเอียดทั้งหมดนั้นจะอยู่ในส่วนของคู่มือสำหรับโปรแกรมเมอร์

6.3.1 Frame

ในการอ้างอิงถึง Frame นั้นจะประกอบไปด้วยรายละเอียดต่าง ๆ 4 ส่วนคือ

- รายละเอียดของ Frame
- Slots ของ Frame
- เป้าหมายของ Frame
- กฎของ Frame

ซึ่งทั้ง 4 ส่วนนี้จะแบ่งลักษณะการใช้งานเป็นเพียง 2 ส่วนคือการอ้างอิงคลาสที่เป็น Persistent ซึ่งก็คือรายละเอียดของ Frame ส่วน Slots, เป้าหมาย และกฎ จะเป็นการอ้างอิงคลาสที่เป็น Embeddable

6.3.1.1 การอ้างอิงคลาส Persistent

การอ้างอิงรายละเอียดของ Frame นั้นจะต้องทราบถึงชื่อของฐานความรู้และ ID ของ Frame ขั้นตอนในการใช้งานเป็นดังนี้

Domain คือ ชื่อ Class ที่จะอ้างอิงถึงในโครงงานนี้ก็คือชื่อของฐานความรู้

Oid คือ id ที่จะอ้างอิง

Caché

Set objopen = ##class(*Domain*).%OpenId(*Oid*)

Visual Basic

Dim Cache As CacheObject.Factory

Dim getinstance As Object

Set getinstance = Cache.OpenId(*Domain*,*Oid*)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3.1.2 การอ้างอิงคลาส Embeddable

ในที่นี้จะขอยกตัวอย่างการอ้างอิงถึง Slots เพียงอย่างเดียว ส่วนเป้าหมายและกฎก็ใช้งานในลักษณะเดียวกัน Slots จะเป็นการเก็บข้อมูลในลักษณะของ Lists ซึ่งจะต้องทราบถึงลำดับใน Lists ที่ต้องการอ้างอิง ลักษณะการใช้งานเป็นดังนี้

Caché

NameSlots1 = objopen.Slots1

Visual Basic

Dim NameSlots1 As String

NameSlots1 = getinstance.Slots1

จาก Code ข้างบนนี้จะเป็นเพียงการเข้าถึง Slots แต่ถ้าเราต้องการเข้าถึงรายละเอียดของ Slots ก็ให้ใช้คำสั่งในลักษณะนี้คือ

NameSlots1 = objopen.Slots1.Name หรือ

NameSlots1 = getinstance.Slots1.Name

แต่ถ้าเราต้องการทราบว่า Frame ประกอบไปด้วย Slots กี่ Slots ก็จะใช้คำสั่งดังนี้

Dim CountSlots As Integer

CountSlots = getinstance.Slots.Count

คำสั่งนี้จะใช้ได้เพียงใน Visual Basic

6.3.2 ข้อเท็จจริง

ในการกำหนดข้อเท็จจริงให้กับฐานความรู้ ระบบจะทำการสร้าง Class ขึ้นมาเพื่อทำการเก็บข้อเท็จจริง ดังนั้นในการอ้างอิงถึงข้อเท็จจริงนั้นจะมีลักษณะเหมือนกับการอ้างอิงลักษณะของ Frame ทุกประการ

6.4 กลไกการอนุมาน

ในโครงงานนี้ได้เลือกกลไกการอนุมานแบบเดินหน้า ซึ่งในหัวข้อที่ 6.2.4 ได้กล่าวถึงหลักการทำงานไปแล้ว ในส่วนนี้จะอธิบายถึงลักษณะของกฎ ที่เก็บเข้าไปไว้ใน Caché และกลไกในการอนุมาน

6.4.1 ลักษณะของกฎ

กฎจะประกอบไปด้วยส่วนสำคัญ 2 ส่วนคือ ส่วนของ If และส่วนของ Then ซึ่งในส่วนของ If จะเก็บเงื่อนไขของ Slots และค่าที่ผู้เชี่ยวชาญเป็นคนกำหนด และในส่วนของ Then นั้นจะเก็บเป้าหมาย(ถ้ามี) และ Frame ที่จะชี้ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4.1.1 If

ในส่วนของ If นั้นเราจะทำการเก็บโดยอ้างอิงถึงลักษณะของ Frame โดยที่จะเก็บเพียงตัวปฏิบัติการ (Operator =, >, <, >=, <=, <>) และค่าที่เป็นเงื่อนไข เพื่อให้เข้าใจได้ง่ายขึ้นจะยกตัวอย่างให้ดูดังนี้

Frame Aperture ประกอบไปด้วย Slots

1. Advice (เป้าหมาย)

2. Asa

3. Light

สมมติว่ากฎที่ผู้เชี่ยวชาญกำหนดเป็นดังนี้

If Asa = 600

เมื่อรับค่าจากมาแล้วโปรแกรมจะทำการตัดประโยคที่รับเข้ามาโดยใช้ตัวปฏิบัติการเป็นตัวแบ่ง ซึ่งจะได้แบ่งได้เป็นสองส่วนคือชื่อของ Slots และส่วนของตัวปฏิบัติกับการกับค่าที่เป็นเงื่อนไข จากนั้นโปรแกรมจะนำชื่อของ Slots ไปเทียบกับลักษณะของ Frame ว่าเป็น Slots ลำดับที่เท่าใด ในที่นี้จะได้ลำดับคือ 2 เพราะฉะนั้นโปรแกรมก็จะทำการจัดรูปแบบใหม่เป็นดังนี้

/=600/

แล้วนำค่าไปเก็บไว้ใน Caché

ถ้าในกรณีที่ Frame ไม่มี Goals

เช่น Frame Details ประกอบไปด้วย Slots

1. Name

2. Surname

ถ้ากฎที่รับเข้ามาคือ

If Name = “Name1”

ก็จะทำการจัดรูปแบบเป็นดังนี้

=”Name1”/

6.4.1.2 Then

Then จะเก็บเป้าหมาย(ถ้ามี) และ ชื่อของ Frame ที่จะชี้ไป ซึ่งในโครงงานนี้จะใช้คำพิเศษคือ “CONSIDERFRAME” ในการบอกว่ากฎนี้จะชี้ไปยัง Frame ใด เช่น

Then Advice = ”The camera must use flash” And Considerframe Dark

โปรแกรมก็จะทำการจัดรูปแบบให้เป็นดังนี้

=”The camera must use flash”/Dark ในกรณีที่ Frame มีเป้าหมาย

แต่ถ้า Frame ไม่มีเป้าหมายก็จะเก็บเพียง Dark

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4.1.3 FramePointer

FramePointer จะเก็บชื่อ Frame ที่ถูกชี้ด้วย Then ซึ่งจะทำให้เราสามารถทราบได้ Frame นั้นจะชี้ไปยัง Frame ใด ๆ ได้บ้าง แต่ถ้าในกรณีที่ Then ไม่ได้บอกว่าจะชี้ไปยัง Frame ใด FramePointer ก็จะชี้ไปยัง Frame ตัวเอง

6.4.2 กลไกในการอนุมาน

ในโครงงานนี้ได้เลือกกลไกในการอนุมานแบบเดินหน้าซึ่งลักษณะของการอนุมานแบบเดินหน้าได้แสดงให้ดูในหัวข้อที่ 6.2.4 แล้วแต่ในส่วนนี้จะบอกถึงวิธีในการค้นหาในโครงงาน

โดยจะเริ่มจาก Frame ที่เป็น Root Frame ก่อน

1.ตรวจสอบก่อนว่า Frame มีเป้าหมายหรือไม่

ถ้ามีเป้าหมายก็จะเริ่มนำ Slots ที่ 2 ขึ้นมาถามผู้ใช้งานระบบ

แต่ถ้าไม่มีเป้าหมายก็จะเริ่มนำ Slots ที่ 1 ขึ้นมาถามผู้ใช้งานระบบ

2.เมื่อรับค่าจากผู้ใช้งานระบบแล้วก็จะทำการนำค่าที่ได้รับจากผู้ใช้งานระบบมาทำการค้นหาข้อเท็จจริงที่เป็นไปได้ทั้งหมดมาเก็บไว้ในตัวแปร และทำการตรวจสอบค่าที่รับเข้ามาว่าตรงกับเงื่อนไขใดของกฎหรือไม่ ถ้าตรงกับเงื่อนไขของกฎและกฎได้ชี้ไปยัง Frame ถัดไปก็จะเปลี่ยนไปเริ่มที่ Frame ที่ถูกชี้และกลับไปยังเริ่มต้นที่ข้อ 1 ใหม่ ถ้าไม่ตรงกับเงื่อนไขใด ๆ ของกฎก็เข้าสู่ขั้นตอนที่ 3

3.อ่าน Slots ถัดไปของ Frame แล้วตรวจสอบว่าเป็น Slots สุดท้ายหรือไม่

ถ้าเป็น Slots สุดท้ายของระบบก็ไปขั้นตอนที่ 4

ถ้าไม่ได้เป็น Slots สุดท้ายของระบบก็นำ Slots ขึ้นมาถามผู้ใช้งานระบบแล้วไปขั้นตอนที่ 2

4.รับค่าจากผู้ใช้งานระบบแล้วทำการค้นหาข้อเท็จจริงกับกฎที่เป็นไปได้ทั้งหมด แล้วตรวจสอบว่า Frame นั้นมีเป้าหมายหรือไม่ ถ้ามีเป้าหมายก็ให้แสดงข้อเท็จจริงหรือกฎที่พบให้กับผู้ใช้งานระบบ แต่ถ้า Frame นั้นไม่มีเป้าหมายก็ต้องตรวจสอบก่อนว่ามีข้อเท็จจริงกับกฎหรือไม่ ถ้าไม่มีกฎก็ให้แสดงข้อเท็จจริงให้กับผู้ใช้งานระบบ แต่ถ้ามีเฉพาะกฎก็ให้ทำการตรวจสอบกฎว่าเป็นกฎที่ชี้ไปยัง Frame ถัดไปก็เปลี่ยนลำดับไปยัง Frame ที่ถูกชี้และกลับไปทำข้อ 1 ใหม่ แต่ถ้าเป็นกฎที่บอกคำตอบให้กับผู้ใช้ระบบ ก็ให้แสดงคำตอบให้กับผู้ใช้งานระบบ แต่ถ้ามีทั้งข้อเท็จจริงกับกฎก็ไม่สามารถสรุปได้

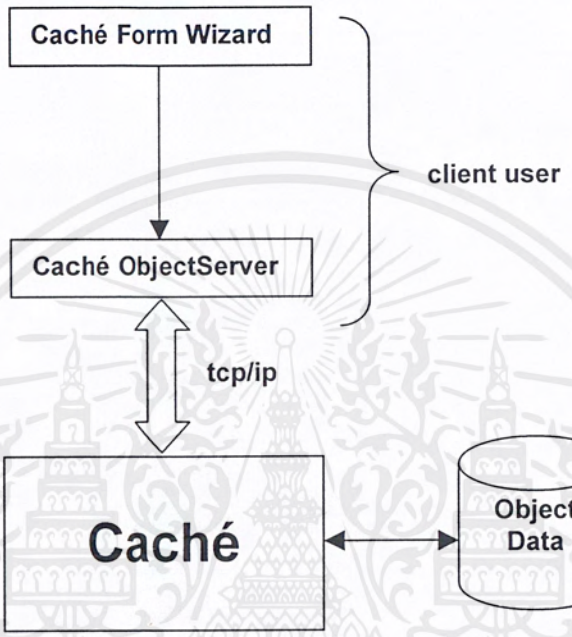
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.5 ส่วนของตัวกลางในการติดต่อกับ Caché (Caché Object Server)

ในการติดต่อเข้าถึงระบบจัดการฐานข้อมูลเชิงวัตถุ Caché นั้นสามารถติดต่อผ่านได้สองทางคือ

- ใช้ Caché Form Wizard
- ใช้ Caché Object Server

ในการเข้าถึงระบบผ่านทางรูปแบบนี้จะต้องมี Protocal TCP/IP ในการเข้าใช้งานซึ่งจะต้องทราบถึง IP และ Port ของ Caché



รูปที่ 6-5 แสดงรูปแบบการเชื่อมต่อเข้าใช้งาน Caché

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
การทดสอบระบบที่ได้ทำการสร้าง

7.1 บทนำ

หลังจากที่เราทำการวิเคราะห์ออกแบบโครงสร้างสถาปัตยกรรม, ออกแบบและสร้างระบบต้นแบบขึ้นมาเรียบร้อยแล้ว ขั้นตอนสุดท้ายก็คือทำการทดสอบระบบที่สร้างขึ้นมาเพื่อพิสูจน์ว่าระบบสามารถทำงานได้ถูกต้องและเป็นไปตามวัตถุประสงค์ที่ต้องการ โดยในส่วนนี้จะอธิบายในส่วนของการสร้างระบบผู้เชี่ยวชาญเพื่อทำการทดลองใช้ประโยชน์จาก Flame base expert system และทำการวิเคราะห์เปรียบเทียบ สำหรับรายละเอียดในบทนี้จะอธิบายถึงโครงสร้างของข้อมูลที่เก็บไว้ในฐานข้อมูลเพื่อใช้ในการทดสอบ และหลังจากนั้นจะเป็นการออกแบบสร้างระบบผู้เชี่ยวชาญที่จะนำข้อมูลที่อยู่ในฐานข้อมูลมาใช้ประโยชน์ในการประมวลผล

7.2 ระบบผู้เชี่ยวชาญที่ทำการออกแบบและสร้างขึ้นเพื่อใช้ในการทดสอบ

ในส่วนของระบบผู้เชี่ยวชาญที่จะทำการสร้างขึ้นมาจะไม่เน้นในเรื่องของกระบวนการถอดความรู้เท่าไรนัก เนื่องจากโครงงานนี้เป็นโครงงานวิจัย ที่ไม่เน้นในส่วนของการนำเอาระบบไปประยุกต์ใช้งานได้จริง ที่ต้องมีการตรวจสอบความถูกต้องของคำตอบที่ได้ ดังนั้นรูปแบบการให้คำตอบและเหตุผลที่ใช้ในการพิจารณาอาจจะผิดเพี้ยนไปจากความเป็นจริง ซึ่งระบบที่ใช้ในการทดสอบนี้จะเป็นระบบผู้เชี่ยวชาญที่ใช้ช่วยในการตัดสินใจเลือกค่าโฟกัสและค่า aperture ในการตั้งหน้ากล้องถ่ายรูป โดยการทำงานจะมีขั้นตอนโดยสังเขปดังนี้

ในส่วนของผู้เชี่ยวชาญทำการใส่ข้อมูลให้กับระบบ

- ผู้เชี่ยวชาญทำการออกแบบเฟรม แยกออกเป็นความรู้และเงื่อนไข
- ผู้เชี่ยวชาญทำการใส่ความรู้เข้าไปในระบบโดยแยกใส่เป็นเป้าหมายระบบ (Goals) ความรู้ (Parameters) และกฎ (Rules)

ในส่วนของผู้ใช้ทำการปรึกษาระบบ

- เลือกโดเมนของปัญหาที่จะปรึกษาระบบ
- กรอกข้อมูลเบื้องต้นให้กับระบบ
- ระบบจะทำการถามเมื่อต้องการข้อมูลเพิ่มเติม
- ระบบประมวลผลแล้วให้คำแนะนำออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3 ตัวอย่างความรู้เรื่อง "ตากล้องมือใหม่"

จากความรู้เกี่ยวกับการถ่ายภาพ ซึ่งเราสามารถนำมาเขียนเป็นตารางได้ดังนี้

ASA	LIGHT-CONDITION	DISTANCE	ADVICE
100	brigth-sun	-	f11 125
	soft-shadow	-	f8 125
	cloudy-bright	-	f5.6 125
	over-case	-	f4 125
	flash	<12	f8 125
		>12	f5.6 125
200	brigth-sun	-	f11 250
	soft-shadow	-	f8 250
	cloudy-bright	-	f5.6 250
	over-case	-	f4 250
	flash	<12	f11 60
		>12	f8 60
400	brigth-sun	-	f11 500
	soft-shadow	-	f8 500
	cloudy-bright	-	f5.6 500
	over-case	-	f5.6 250
	flash	<12	f16 60
		>12	f16 60

ตารางที่ 7-1 แสดงความรู้เรื่อง "ตากล้องมือใหม่"

จากความรู้เรื่องตากล้องมือใหม่และตารางที่ 7-1 ทำให้เรากำหนดรายละเอียดของ Frame และ Frame ที่จะมีอยู่ในฐานความรู้ได้ดังตารางที่ 7-2 และตารางที่ 7-3

Class CameraSetting		
FrameID :	1	2
Name :	APERTURE	DARK
FrameTranslation :	A program to help you to set the aperture of camera	When light condition is dark

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Goals : Name :	ADVICE	DARKS
GoalsTranslation :	The camera setting advisor	Camera setting for the dark condition
Type :	SINGELVALUED	SINGELVALUED
Parameters 1: Name :	ADVICE	DARKS
ParametersTranslation :	The camera setting advisor	Camera setting for the dark condition
Prompt :	-	The distance must be consider when light condition is dark
Type :	SINGLEVALUED	SINGLEVALUED
Expect :	-	-
Default :	-	-
Values :	-	-
Parameters 2: Name :	ASA	DISTANCE
ParametersTranslation :	Type of film speed	The distance between camera and object
Prompt :	What is the value of asa ?	What is the distance between object and camera ?
Type :	SINGLEVALUED	SINGLEVALUED
Expect :	User-defined	Number
Default :	200	-
Values :	100,200,400	1,100
Parameters 3: Name :	LIGHT	-
ParametersTranslation :	Light condition	-
Prompt :	What is the appropriate answer to described the light condition ?	-
Type :	SINGLEVALUED	-
Expect :	User-defined	-
Default :	SOFT-SHADOW	-
Values :	BRIGHT,SOFT-SHADOW, CLOUDY,DARK	-
Rules 1: FramePointer :	DARK	DARK
IfStatement :	LIGHT=DARK	DISTANCE >12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ThenStatement :	ADVICE="The camera must use flash" AND CONSIDERFRAME DARK	DARKS="Set focus to 11 and speed 125"
Source :	APERTURE	APERTURE
Destination :	OWNED	MEMBER

ตารางที่ 7-2 แสดงรายละเอียดของฐานความรู้ เรื่อง "ตากล้องมือใหม่"

ข้อเท็จจริง สามารถแสดงได้ด้วยตารางดังรูปที่ 7-3

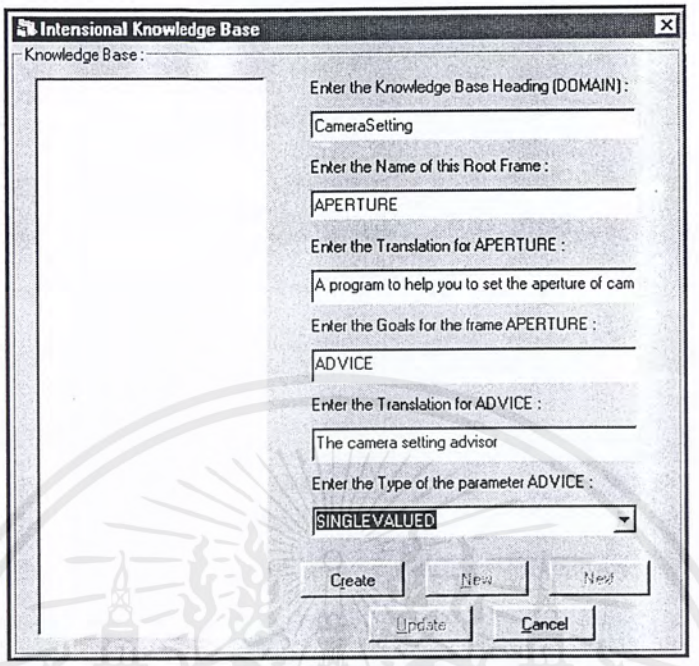
ID	ADVICE	ASA	LIGHT
1	Set focus to 11 and set aperture to 125	100	BRIGHT-SUN
2	Set focus to 8 and set aperture to 125	100	SOFT-SHADOW
3	Set focus to 5.6 and set aperture to 125	100	CLOUDY-BRIGHT
4	Set focus to 4 and set aperture to 125	100	OVER-CASE
5	Set focus to 11 and set aperture to 250	200	BRIGHT-SUN
6	Set focus to 8 and set aperture to 250	200	SOFT-SHADOW
7	Set focus to 5.6 and set aperture to 250	200	CLOUDY-BRIGHT
8	Set focus to 4 and set aperture to 250	200	OVER-CASE
9	Set focus to 11 and set aperture to 500	400	BRIGHT-SUN
10	Set focus to 8 and set aperture to 500	400	SOFT-SHADOW
11	Set focus to 5.6 and set aperture to 500	400	CLOUDY-BRIGHT
12	Set focus to 5.6 and set aperture to 250	400	OVER-CASE

ตารางที่ 7-3 แสดงข้อเท็จจริงเรื่อง "ตากล้องมือใหม่"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

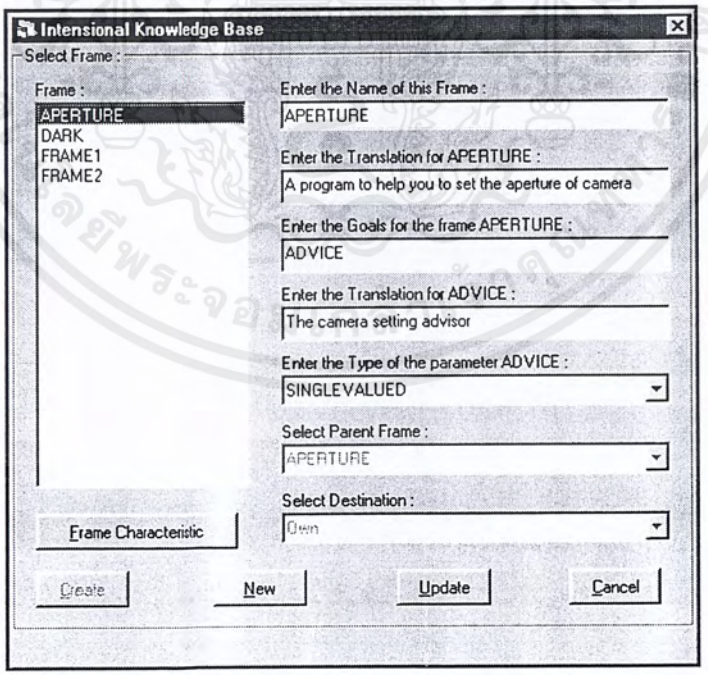
7.3.1 การสร้างฐานความรู้เรื่อง "ตากล้องมือใหม่"

การสร้างฐานความรู้นั้นจะเริ่มต้นจากการกำหนดรายละเอียดของฐานความรู้ก่อนดังรูปที่ 7-1



รูปที่ 7-1 แสดงขั้นตอนในการกำหนดรายละเอียดของฐานความรู้

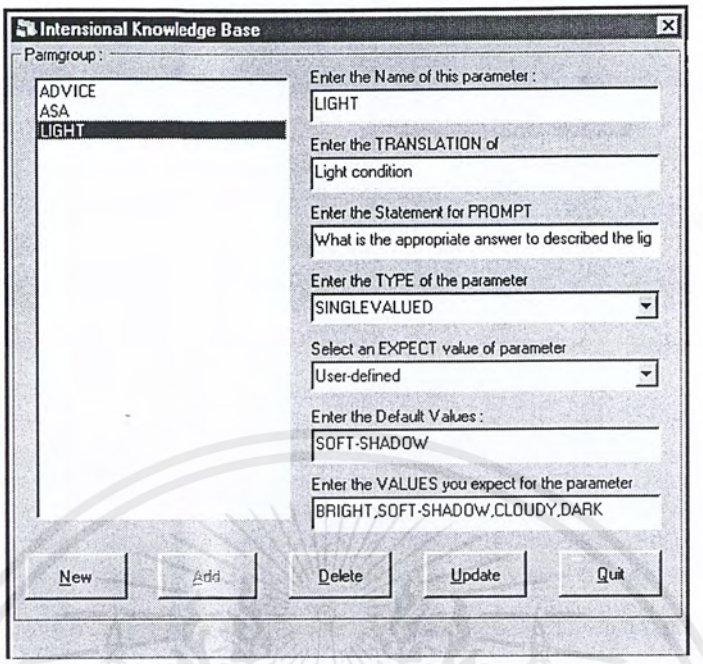
แล้วทำการกำหนดรายละเอียดของ Frame ต่าง ๆ ให้กับฐานความรู้ ดังรูปที่ 7-2



รูปที่ 7-2 แสดงขั้นตอนการกำหนดรายละเอียดของ Frame

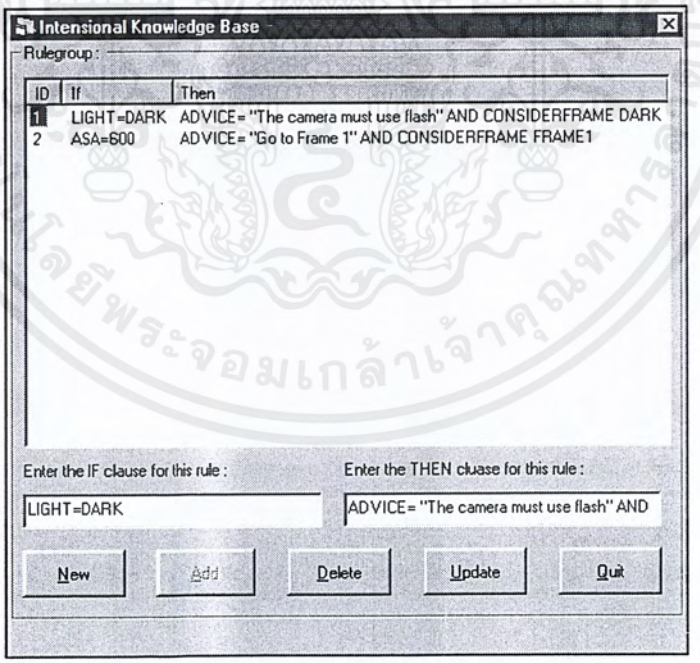
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้วทำการกำหนดรายละเอียดของ Slots ต่าง ๆ ให้กับฐานความรู้ดังรูปที่ 7-3



รูปที่ 7-3 แสดงขั้นตอนในการกำหนดรายละเอียดของ Slots

แล้วทำการกำหนดรายละเอียดของ Rules ต่าง ๆ ให้กับฐานความรู้ดังรูปที่ 7-4

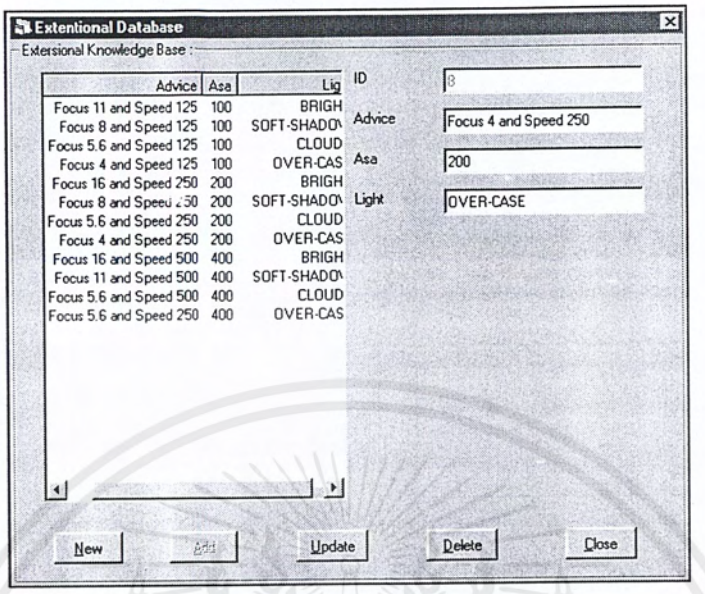


รูปที่ 7-4 แสดงขั้นตอนในการกำหนดรายละเอียดของ Rules

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3.2 ข้อเท็จจริงเรื่อง "ตากล้องมือใหม่"

จากตารางที่ 7-3 นำมาป้อนเข้าในข้อเท็จจริงให้กับระบบได้ดังรูปที่ 7-5



รูปที่ 7-5 แสดงขั้นตอนการป้อนข้อเท็จจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8
สรุปผล,วิจารณ์ และแนวทางการพัฒนาต่อ

8.1 สรุปผลงานวิจัย

จากวัตถุประสงค์ซึ่งเป็นที่มาของโครงงานวิจัยชิ้นนี้ ก็คือเพื่อหาวิธีการประยุกต์ใช้ข้อมูลที่เก็บอยู่ระบบจัดการฐานข้อมูลเพื่อนำมาใช้กับระบบผู้เชี่ยวชาญสำหรับในการให้ข้อมูลเพื่อช่วยให้การทำงานของระบบผู้เชี่ยวชาญมีประสิทธิภาพมากขึ้น คือ ไม่ต้องอาศัยตัวกลางในการติดต่อกันระหว่างระบบผู้เชี่ยวชาญและฐานข้อมูล และยังเป็นการลดการถามคำถามที่ไม่ควรจะถามผู้ใช้ เนื่องจากจะทำให้มีการถามคำถามเกิดขึ้นมากเกินไป และเกิดความซ้ำซาก เนื่องจากทุกครั้งที่เริ่มการทำงานของระบบผู้เชี่ยวชาญใหม่ก็จะต้องมีการถามคำถามนั้นอีก ซึ่งคำถามประเภทนี้ ก็คือ คำถามประเภทที่เป็นข้อมูลในอดีตซึ่งจะมีประโยชน์อยู่แล้วหากได้มีการเก็บเอาไว้ในฐานข้อมูล ซึ่งตามปกติตัวโปรแกรมเปลือกระบบผู้เชี่ยวชาญนั้นจะมีความสามารถในการติดต่อกับแหล่งข้อมูลภายนอกได้อย่างจำกัดอยู่แล้ว ซึ่งทางโครงงานก็ได้คิดค้นหาวิธีการที่ใช้ในการเชื่อมต่อกับระบบจัดการฐานข้อมูล เพื่อให้เราสามารถทำการบริหารข้อมูลได้ง่ายขึ้น และยังสามารถนำข้อมูลเหล่านั้นที่เก็บอยู่ในระบบฐานข้อมูลไปใช้ประโยชน์โดยโปรแกรมอื่นๆ ที่มีการเรียกใช้ฐานข้อมูลเช่นเดียวกัน

นอกจากนี้ การที่ระบบได้มีการถามคำถามจากผู้ใช้น้อยลงนั้น ก็ยังเป็นประโยชน์ คือ ช่วยให้ระบบผู้เชี่ยวชาญสามารถทำงานได้อย่างถูกต้องมากขึ้น เนื่องจากสาเหตุในกรณีดังต่อไปนี้

- การถามคำถามกับผู้ใช้มากๆ จะเป็นการเพิ่มโอกาสที่จะเกิดความผิดพลาดอันเนื่องมาจากผู้ใช้ตอบคำถามกับตัวโปรแกรมผิดๆ เนื่องจากเกิดความเบื่อหน่ายที่จะตอบคำถามกับโปรแกรม
- การถามคำถามบางอย่าง ซึ่งเป็นข้อมูลในอดีต บางครั้งอาจพบว่า ผู้ใช้เองอาจจำไม่ได้ เนื่องจากเวลาผ่านมานานมากเกินไป ทำให้มีการตอบคำถามที่ผิดๆ ได้
- คำถามบางคำถาม ซึ่งอาจจะเกี่ยวข้องกับตัวผู้ใช้ แต่ตัวผู้ใช้อาจไม่รู้ เนื่องจากอาจเป็นความรู้ที่เป็นความรู้เฉพาะทาง ซึ่งก็จะทำให้ผู้ใช้ตอบคำถามที่ผิดๆ ก็ได้

และในขอบเขตของโครงงานที่ได้มีการนำเสนอเปลือกระบบผู้เชี่ยวชาญ ให้มีความสามารถมากยิ่งขึ้นโดยมีการนำข้อมูลจากแหล่งอื่นมาเป็นข้อมูลให้กับระบบอีกด้วย เพื่อให้เกิดประสิทธิภาพมากยิ่งขึ้นโดยผู้เชี่ยวชาญไม่จำเป็นต้องกรอกข้อมูลใหม่ทุกครั้ง ในโครงงานได้ให้เปลือกระบบผู้เชี่ยวชาญมีความสามารถในการรับฐานข้อมูลจาก Microsoft Access

ด้วยเทคโนโลยี Object Oriented ทำให้สามารถเชื่อมระบบผู้เชี่ยวชาญซึ่งใช้ Frame based expert system กับทางด้านฐานข้อมูลซึ่งมีลักษณะเป็น Object Oriented Database ทำให้การเชื่อมต่อสามารถทำได้ง่ายและสะดวก กล่าวคือไม่จำเป็นต้องมีตัวกลางในการเชื่อมต่อระหว่างทั้งสองฝั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.2 วิจารณ์งานวิจัย

จากงานวิจัยที่ได้ทำการค้นคว้าและคิดค้นขึ้นมานั้น เมื่อเปรียบเทียบกับระบบอื่นๆ ที่มีอยู่ก่อนแล้วสิ่งที่ผลงานของงานวิจัยชิ้นนี้ที่แตกต่างและดีกว่าสถาปัตยกรรมของระบบผู้เชี่ยวชาญอื่นๆ ก็คือ

- ความสามารถในการติดต่อกันระหว่างระบบผู้เชี่ยวชาญแบบ Frame-based และระบบจัดการฐานข้อมูลเชิงวัตถุซึ่งมีลักษณะโครงสร้างข้อมูลที่คล้ายกัน กล่าวคือ ทางระบบผู้เชี่ยวชาญมีลำดับที่เฟรม สล๊อต(Slot) สล๊อต(Procedure) ทางด้านฐานข้อมูลก็มี OID ,Attribute และ method ตามลำดับ ทำให้ง่ายต่อการเชื่อมต่อ ไม่ต้องอาศัยตัวกลางเพื่อทำการแปรข้อมูลแต่อย่างใด
- ความสามารถในการใช้เก็บข้อมูลที่ซับซ้อน หรือข้อมูลขนาดใหญ่ เป็นการลดข้อด้วยของระบบผู้เชี่ยวชาญในปัจจุบัน
- ความสามารถในการติดต่อกับฐานข้อมูลภายนอก โดยให้ระบบผู้เชี่ยวชาญใช้ความสามารถของทางด้านฐานข้อมูลเป็นตัวจัดการ นำฐานข้อมูลภายนอกระบบให้กับระบบผู้เชี่ยวชาญ ลดความยุ่งยากในการกรอกข้อมูลใหม่ ลดการซ้ำซ้อนของการทำงาน
- ลดความยุ่งยากในการจัดการกับข้อมูลที่รับมาจากฐานข้อมูลซึ่งบางครั้งทางระบบจัดการฐานข้อมูลจะมีการส่งข้อมูลที่ไม่ได้เป็นค่าเดียวแต่จะเป็นเซ็ตของชุดข้อมูล โดยจะทำให้เป็นเสมือนว่าระบบผู้เชี่ยวชาญกำลังทำงานกับข้อมูลตัวเดียว และจะให้โปรแกรมที่เป็นตัวกลางสำหรับทำหน้าที่ในการติดต่อกับฐานข้อมูลเป็นตัวจัดการเอาเอง ทำให้สามารถใช้งานกับระบบผู้เชี่ยวได้หลากหลายรูปแบบขึ้น

แต่ส่วนที่ทำการพัฒนาขึ้นมานั้น ระบบสถาปัตยกรรมของการติดต่อระหว่างระบบผู้เชี่ยวชาญและฐานข้อมูลที่ได้ออกแบบในโครงงานนี้ไม่ได้มีความสามารถครอบคลุมในบางส่วน แต่พบว่ามีระบบผู้เชี่ยวชาญอื่นๆ ที่มีความสามารถครอบคลุมถึง คือ

- ความสามารถในการเปลี่ยนแปลงข้อมูลที่อยู่ในฐานข้อมูลโดยระบบผู้เชี่ยวชาญ
- ความสามารถในการติดต่อกับระบบฐานข้อมูลโดยที่ผู้พัฒนาโปรแกรมไม่จำเป็นต้องทราบในขณะที่ทำการพัฒนาระบบผู้เชี่ยวชาญ โดยจะถือว่าข้อมูลต่างๆ เหล่านั้นทางระบบผู้เชี่ยวจะทำการจัดการค้นหาเอาเองว่าจะดึงมาจากฐานข้อมูล หรือจะต้องถามจากผู้ใช้
- ความสามารถในการเรียกดูผลของการทำงานของระบบผู้เชี่ยวชาญที่เคยทำงานไว้ในอดีตซึ่งความสามารถนี้ต้องอาศัยความสามารถในการบันทึกผลการทำงานของระบบผู้เชี่ยวชาญลงในฐานข้อมูลด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.3 แนวทางการพัฒนาต่อ

สำหรับแนวทางในการพัฒนาต่อนั้นอาจแยกได้เป็นข้อ ๆ ดังนี้

- พัฒนาระบบผู้เชี่ยวชาญในด้านการออกแบบ Meta Rule เพื่อให้ระบบผู้เชี่ยวชาญมีความสามารถมากยิ่งขึ้น กล่าวคือ ในโครงงานการตัดสินใจของระบบผู้เชี่ยวชาญยังเป็นไปในลักษณะ Decision Tree ทำให้ระบบผู้เชี่ยวชาญไม่ยืดหยุ่นเท่าที่ควร
- พัฒนาในส่วนของระบบผู้เชี่ยวชาญให้มีความสามารถในการทำ Conflict Resolution กล่าวคือ หากในส่วนของกฎ มีความซ้ำซ้อนกันจะให้ระบบมีความสามารถในการตัดสินใจอย่างไร
- พัฒนาในส่วนที่ทำให้ระบบผู้เชี่ยวชาญมีความสามารถในด้านการเปลี่ยนแปลงแก้ไขข้อมูลที่อยู่ในฐานข้อมูลได้ ซึ่งจะทำให้ระบบนี้มีรูปแบบการใช้ประโยชน์ที่กว้างขึ้นไปอีก เช่น อาจใช้เป็น Intelligent Agent ที่ใช้ในการแก้ไขข้อมูลอะไรบางอย่างในฐานข้อมูล โดยสามารถปล่อยให้โปรแกรมทำงานเองแบบอัตโนมัติ โดยไม่ต้องให้คนมาทำหน้าที่นี้เอง
- พัฒนาในส่วนที่ให้ระบบผู้เชี่ยวชาญ สามารถรับหรือจัดการกับข้อมูลจากฐานข้อมูลซึ่งมีรูปแบบข้อมูลที่ซับซ้อนได้มากกว่าที่เป็นอยู่เพื่อช่วยอำนวยความสะดวกให้กับผู้เชี่ยวชาญ เช่น ข้อมูลที่เป็นแฟ้มข้อมูลแบบ Multimedia เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[วิลาศ, บุญเจริญ 2535] วิลาศ วูวงศ์, บุญเจริญ ศิริเนาวกุล, *ระบบผู้เชี่ยวชาญ*, ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ, 2535.

[วิทยา 2541] วิทยา วัชระวิทยากุล (เรียบเรียง), Kernighan, B.W., Ritchie, D.M., *ภาษาและการโปรแกรม C*, บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน), 2541.

[Abawajy,Shepherd 1994] Abawajy, J., Shepherd, M., *Framework for the Design of Coupled Knowledge/Data Base Medical Information Systems*, Seventh Annual IEEE Symposium on Computer-Based Medical System, Proceeding on, 1994, pp 1-5.

[Ceri,Ramakrishnan 1996] Ceri, S., Ramakrishnan, R. , *Rule in Database Systems*, ACM Computing Surveys, Vol. 28, No.1, March 1996, pp 109-111.

[Czejdo et al. 1993] Czejdo, B., Eick, C.F., Taylor, M., *Integrating Sets, Rules and Data in an Object-Oriented Environment*, IEEE Expert (Intelligent system & their application), Vol 8, No. 1, February 1993, pp 59- 66.

[Date 1995] Date,C.J., *An Introduction to Database Sysetms* , 6th edition, Addison-Wesley , 1995.

[Ioannidis et al 1989] Ioannidis, Y.E., Chen, J., Friedman, M.A., Tsangaris, M.M., *BERMUDA-An Architecture Perspective on Interfacing Prolog to a database Machine*, Expert database systems, the Second International Conference, Proceeding from, Benjamin/Cumming, 1989, pp 229-255.

[Kandzia, Schlepphorst 1997] Kandzia, P.-Th., Schlepphorst, C., *Florid – A Prototype for F-logic*, 12th German Workshop on Logic Programming (WLP'97), Proceeding from, 1997, pp 1-7.

[Kifer 1995] Kifer, M., *Deductive and Object Data Languages: A Quest for Integration*, Deductive and Object-Oriented Databases, 4th International Conference, Proceeding from, Published as Lecture Notes on Computer Science, Springer-Verlag, 1995.

[Liu 1999] Liu, M., *Deductive Database Languages: Problems and Solutions*, ACM Computing Surveys, Vol. 31, No. 1, March 1999, pp 27-62.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[Nikolopoulos 1997] Nikolopoulos, C., *Expert Systems*, Marcel Dekker,Inc., 1997

[Stanfield,Arvesen] Stanfield,S,. Arvesne,R. ,*Visual C++ 4 How-to*, Waite Group press, 1996

[Steib 1996] Steib, S., *Expert Database Systems*, CS530: Information Systems and Database Design, Lecture notes on, http://informatics.wustl.edu/marrs/cs530/Sherry1/Sherry1.ps, 1996

[Stroustrup 1986] Stroustrup, B., *The C++ Programming Language*, Addison-Wesley, 1986

[Telebzadeh et al. 1995] Telebzadeh, H., Mandutianu, S., Winner, C.F., *Countrywide Loan-Underwriting Expert System*, AI Magazine, Spring 1995, AAAI, pp 51-64.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.
คู่มือการใช้งาน

1. ความต้องการของระบบ

1.1 Caché

สำหรับในการติดตั้งโปรแกรมระบบจัดการฐานข้อมูลเชิงวัตถุ Caché จะติดตั้งได้ในระบบปฏิบัติการ Windows และ Unix/Linux และ VMS สำหรับระบบปฏิบัติการที่โครงงานนี้เลือกใช้คือ ระบบปฏิบัติการ Windows 95/98/NT ซึ่งจะต้องใช้ระบบคอมพิวเตอร์ขั้นต่ำดังนี้

- เครื่องคอมพิวเตอร์ PC ที่ใช้ CPU รุ่น Pentium ขึ้นไป
- หน่วยความจำของเครื่อง 32 MB เป็นอย่างต่ำ (แนะนำให้ใช้ขนาด 64 MB)
- พื้นที่ว่างบนฮาร์ดดิสก์ 100 MB เป็นอย่างต่ำ (แนะนำว่าควรมี 300 MB)
- ระบบปฏิบัติการ Microsoft Windows 95/98

สำหรับในการติดตั้งโปรแกรมเพื่อใช้งานนั้น จะต้องทำการติดตั้งจากแผ่น CD-ROM ซึ่งใช้ในการติดตั้งโปรแกรม หรือสามารถ Download ได้จาก www.e-dbms.com จากนั้นก็ให้ปฏิบัติตามขั้นตอนต่าง ๆ ซึ่งจะทำงานในรูปแบบของโปรแกรม Installation Wizard ไปเรื่อย ๆ จนเสร็จสมบูรณ์ก็เป็นอันเสร็จสิ้นขั้นตอนการติดตั้งโปรแกรม

1.2 Microsoft Visual Basic Library

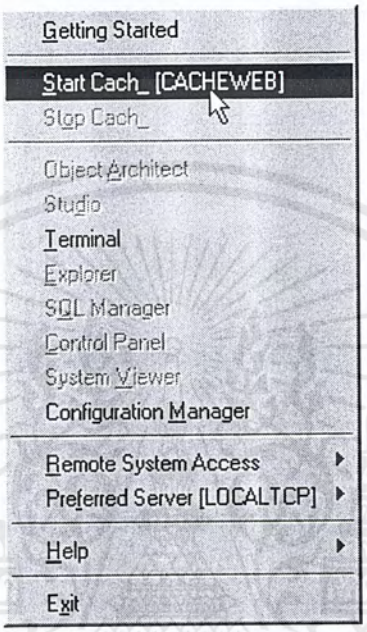
- Microsoft Visual Basic 6.0 Runtime Library (สามารถหาได้จาก www.microsoft.com)
- Microsoft Data Access Components 2.0 (สามารถหาได้จาก www.microsoft.com)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การใช้งานโปรแกรม

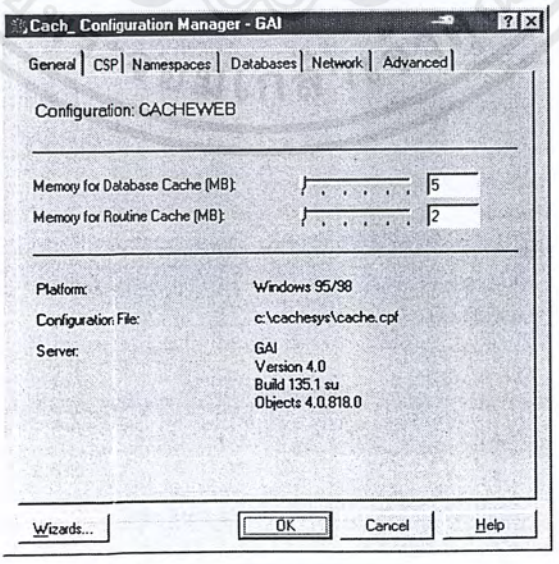
2.1 Import File ที่จำเป็นให้กับ Caché

หลังจากที่ได้ติดตั้ง Caché และ Microsoft Visual Basic 6.0 Runtime Library และ Microsoft Data Access Components 2.0 เรียบร้อยแล้ว ก็ให้เริ่มทำการเรียกโปรแกรม Caché โดยทำการ Click ขวาที่ Icon Caché ที่อยู่ใน Taskbar ( ) ทางด้านซ้ายสุดแล้วเลือก Start Caché ดังรูปที่ ก-1



รูปที่ ก-1 หน้าต่างเริ่มต้นของโปรแกรม Caché

โปรแกรมจะทำการเริ่มการให้บริการฐานข้อมูล และ Icon ใน Taskbar จะเปลี่ยนเป็นรูป ( ) จากนั้นให้ทำการ Click ขวาที่ Icon Caché แล้วเลือกที่ Configuration Manager จะปรากฏหน้าต่างดังรูปที่ ก-2



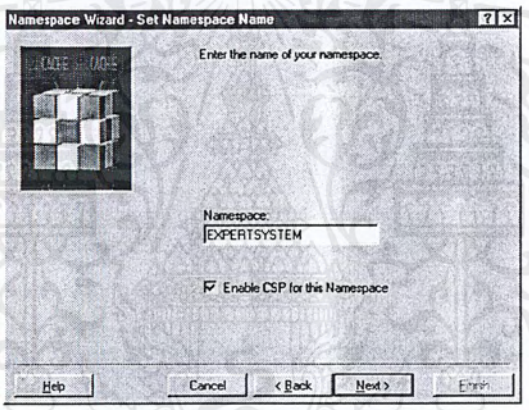
รูปที่ ก-2 หน้าต่างเริ่มต้นของการกำหนด Namespaces ของ Caché

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นให้ Click ที่ Tab Namespaces ดังรูปที่ ก-3

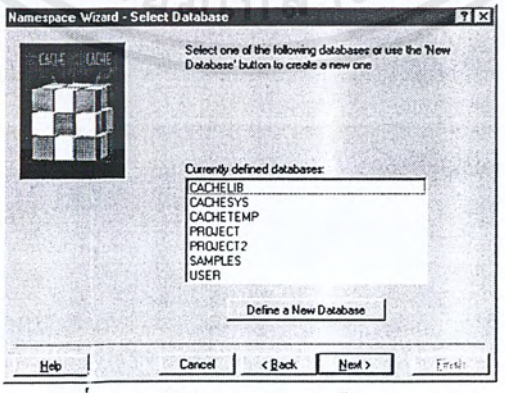
รูปที่ ก-3หน้าต่างแสดง Namespace

Click ที่ Add.. Button แล้วใส่ชื่อ Namespace เป็น **EXPERTSYSTEM** แล้ว Click Next ดังรูปที่ ก-4



รูปที่ ก-4 หน้าต่างแสดงการกำหนดชื่อให้กับ Namespace

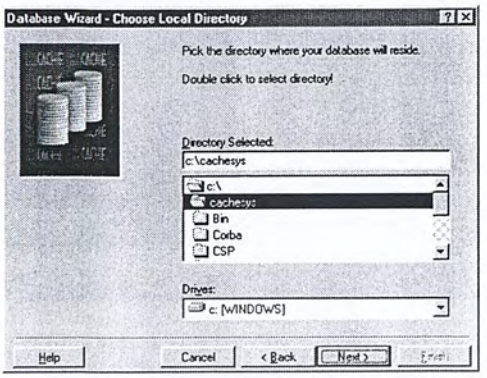
จะปรากฏหน้าต่างดังรูปที่ ก-5



รูปที่ ก-5 หน้าต่างแสดงการเลือก Database

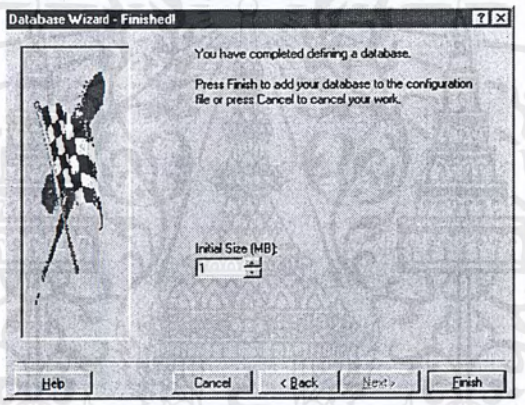
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Click ที่ Define a New Database แล้วทำการใส่ชื่อ **EXPERTSYSTEM** หรือจะใส่ชื่อเป็นอย่างอื่นก็ได้ แล้ว Click ที่ Next Button จะปรากฏหน้าต่างดังรูปที่ ก-6



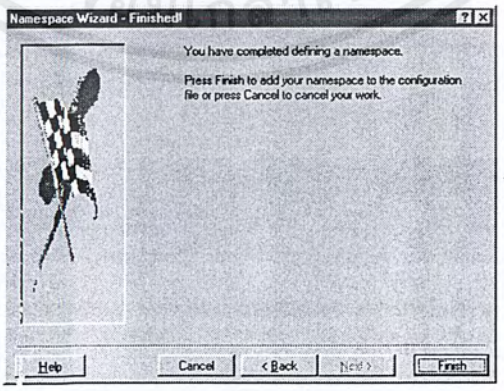
รูปที่ ก-6 หน้าต่างแสดงการเลือกเนื้อที่ที่ต้องการเก็บ Database

เลือก Drive และ Path ที่ต้องการจะทำการเก็บ Database ของ Caché แล้ว Click Next Button จะปรากฏหน้าต่างดังรูปที่ ก-7



รูปที่ ก-7 หน้าต่างแสดงการกำหนดเนื้อที่เริ่มต้น

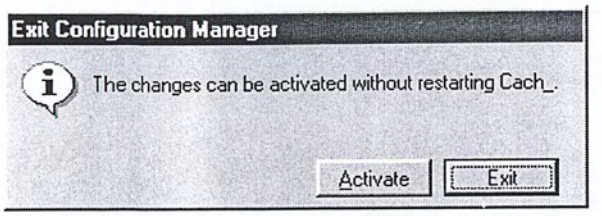
กำหนดค่าเริ่มต้น (Initial Size) ให้เท่ากับ 1 แล้ว Click Finish Button จะปรากฏหน้าต่างดังรูปที่ ก-8



รูปที่ ก-8 หน้าต่างแสดงขั้นตอนเสร็จสิ้นกระบวนการกำหนด Namespace

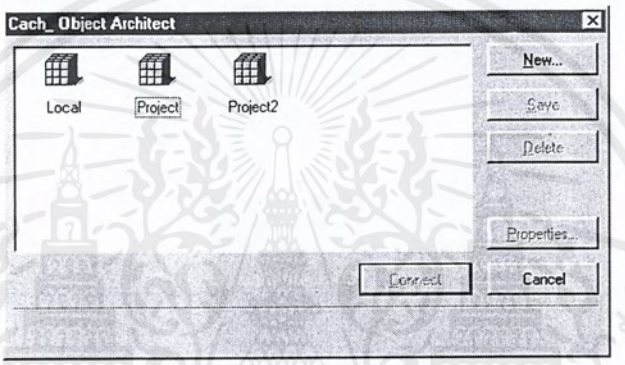
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Click ที่ Finish Button จะกลับมาที่หน้าต่างเดิมตามรูปที่ 3 Click ที่ OK button จะปรากฏหน้าต่างดังรูปที่ ก-9



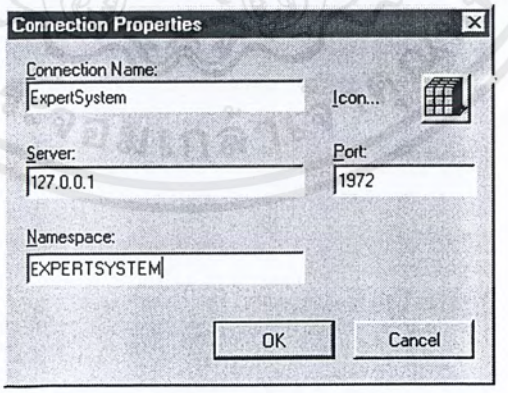
รูปที่ ก-9 หน้าต่างแสดงการกำหนดให้ Namespace ใช้งานได้เลยหรือไม่

Click ที่ Activate Button แล้วให้ทำการ Click ขวาที่ Icon Caché เลือก Object Architect จะปรากฏหน้าต่างดังรูปที่ ก-10



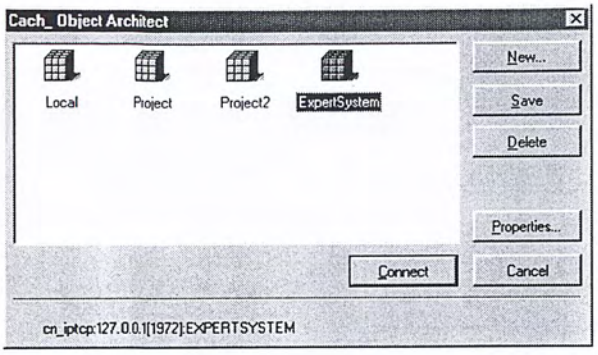
รูปที่ ก-10 หน้าต่างแสดงการกำหนดคีย์ลัดในการเข้าใช้งาน Namespace

Click ที่ New... Button จะปรากฏหน้าต่างดังรูปที่ 11 ให้ใส่ค่าดังรูปที่ ก-11 ตรงส่วนของ Connection Name: สามารถเปลี่ยนเป็นชื่ออื่นได้ แล้ว Click OK Button จะปรากฏหน้าต่างดังรูปที่ ก-12



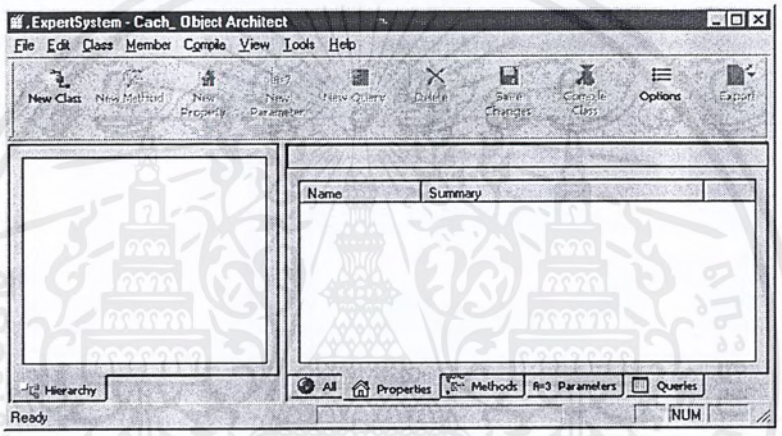
รูปที่ ก-11 หน้าต่างแสดงการกำหนดรายละเอียดในการเข้าใช้งาน Namespace

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



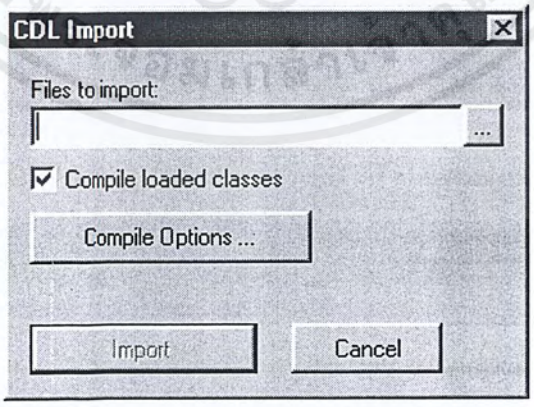
รูปที่ ก-12 หน้าต่างแสดงคีย์ลัดที่ได้กำหนดไว้

Click ที่ Connect จะปรากฏหน้าต่างดังรูปที่ ก-13



รูปที่ ก-13 หน้าต่างเริ่มต้นของ Caché **Object Architect**

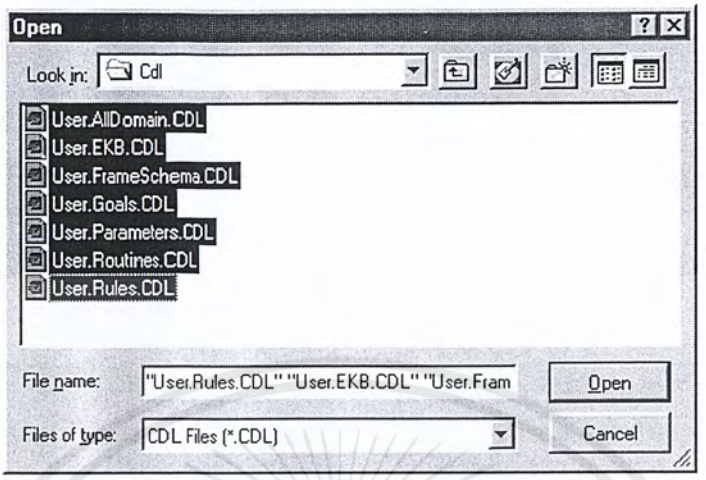
Click File -> Import -> Import CDL ... จะปรากฏหน้าต่างดังรูปที่ ก-14 ให้ Check Box ที่ Compile loaded classes



รูปที่ ก-14 หน้าต่างแสดงการนำเข้าข้อมูลจากภายนอก

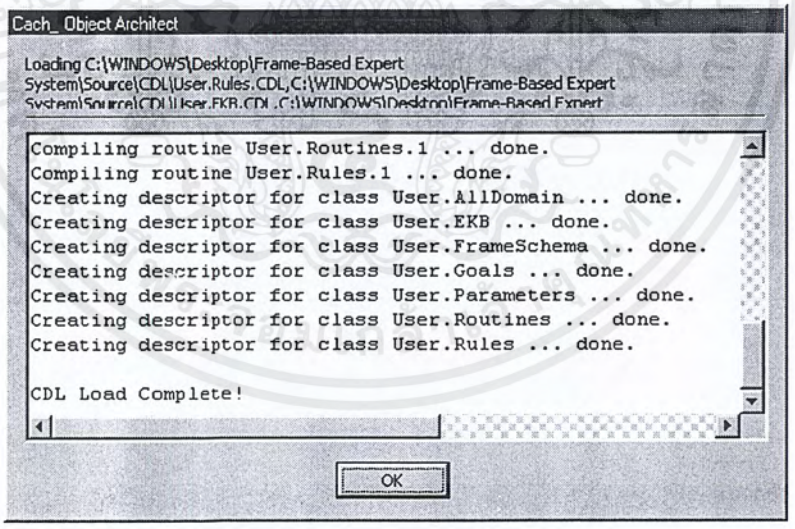
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Click ... Button แล้ว ไปยัง Folder CDL แล้ว กด Shift ค้างไว้แล้วเลือกทุก Files แล้ว Click Open Button ดังรูปที่ ก-15



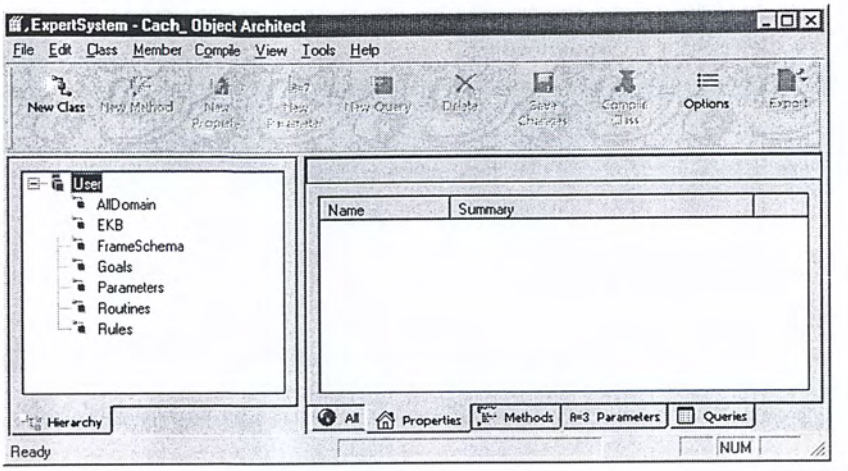
รูปที่ ก-15 หน้าต่างแสดงการเลือกไฟล์ที่จะนำเข้าสู่ Caché **Object Architect**

จะกลับมายังหน้าต่างดังรูปที่ ก-14 โดยใน TextBox Files to import : จะเปลี่ยนไป จากนั้น Click Import Button รอจนกระทั่งปรากฏหน้าต่างดังรูปที่ ก-16 ใน TextBox บรรทัดสุดท้ายจะปรากฏข้อความว่า CDL Load Complete ! แล้วให้กด OK Button จะกลับมายังหน้าต่างดังรูปที่ ก-17



รูปที่ ก-16 หน้าต่างแสดงรายละเอียดของการนำเข้าข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-17 หน้าต่างแสดงรายละเอียดของข้อมูลที่นำเข้าจากภายนอก

Click File -> Exit

2.2 เริ่มต้นใช้งานระบบ

ทำการ Run File **Frame-Based Expert System.exe** เข้าสู่ระบบจะปรากฏหน้าต่างเริ่มต้นดังรูปที่ ก-18

รูปที่ ก-18 หน้าต่างเริ่มต้นของระบบที่โครงงานได้พัฒนาขึ้น

ระบบที่โครงงานได้พัฒนาขึ้นมาจะประกอบไปด้วยส่วนสำคัญอยู่ 2 ส่วนคือ

- Develop ส่วนติดต่อกับผู้เชี่ยวชาญในการกำหนดองค์ประกอบต่าง ๆ ให้กับระบบ

Intensional Knowledge Base ส่วนของการกำหนดฐานความรู้ให้กับระบบ, โครงสร้างของ Frames, กำหนด Frame และ Rules ให้กับระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Extensional Knowledge Base ส่วนของการกำหนดข้อเท็จจริง (Facts) ให้กับระบบประกอบไปด้วย

- Import จาก Database ที่มีอยู่แล้ว
- สร้างขึ้นจาก Frame ที่อยู่ในระบบ

- Consult ส่วนติดต่อกับผู้ใช้งานระบบ

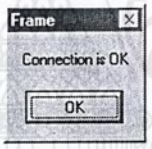
2.2.1 Develop

ผู้เชี่ยวชาญจะเข้าใช้งานระบบโดยผ่านทางขั้นตอนนี้ โดยจะประกอบไปด้วยส่วนของการกำหนดฐานความรู้ให้กับระบบ, โครงสร้างของ Frame การกำหนด Slots ต่าง ๆ ให้กับ Frames และกำหนด Rules ของแต่ละ Frame

2.2.1.1 Intensional Knowledge Base ส่วนของการกำหนดฐานความรู้ให้กับระบบ, โครงสร้างของ Frames, กำหนด Frame และ Rules ให้กับระบบ

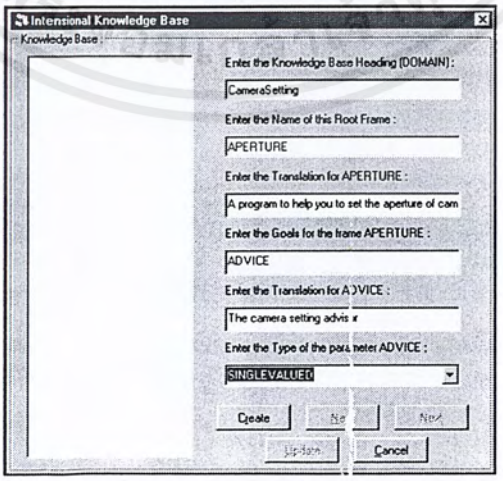
2.2.1.1.1 กำหนดโครงสร้างของฐานความรู้

Click ที่ Activities -> Develop จะปรากฏ Message Box บอกว่าการเชื่อมต่อกับระบบฐานข้อมูลเรียบร้อยหรือไม่ ถ้าเชื่อมต่อได้จะปรากฏหน้าต่างดังรูปที่ ก-19



รูปที่ ก-19 หน้าต่างแสดงการเชื่อมต่อระหว่างระบบที่ได้พัฒนาขึ้นกับระบบฐานข้อมูล

Click ที่ Knowledge Base -> Intensional Knowledge Base จะปรากฏหน้าต่างดังรูปที่ ก-20 ให้ทำการ Click ที่ New Button เพื่อกำหนดรายละเอียดต่าง ๆ ของฐานความรู้



รูปที่ก-20 หน้าต่างแสดงรายละเอียดของการกำหนดฐานความรู้

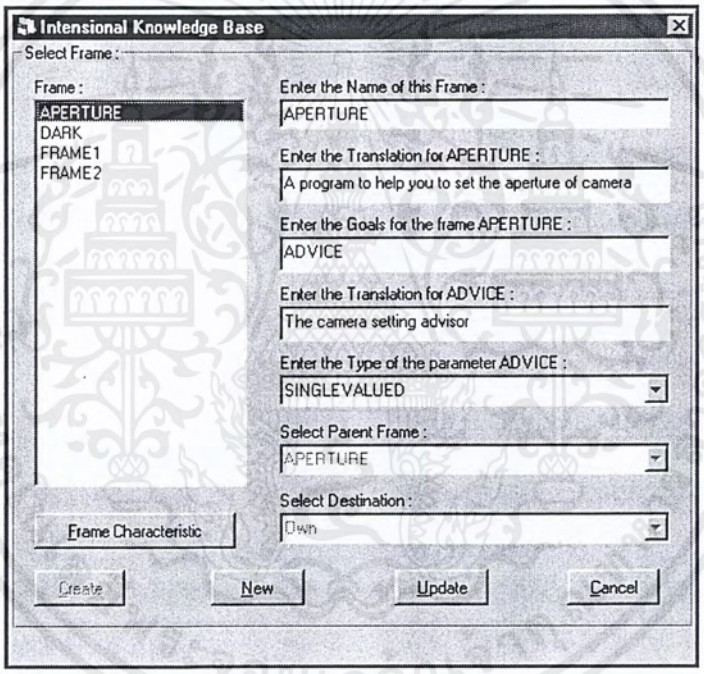
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อกำหนดโครงสร้างของฐานความรู้ ซึ่งประกอบไปด้วยการกำหนด Root Frame และกำหนด Goals ให้กับ Root Frame เรียบร้อยแล้ว Click Create Button เมื่อเสร็จแล้วก็จะปรากฏฐานความรู้ที่เราสร้างให้กับระบบขึ้นใน List ทางด้านซ้ายมือ

- Create สร้างฐานความรู้ให้กับระบบ
- New ล้างค่าที่ปรากฏอยู่บนหน้าจอเพื่อทำการกำหนดฐานความรู้ใหม่ให้กับระบบ
- Update แก้ไขรายละเอียดต่าง ๆ ของฐานความรู้
- Next เข้าสู่ขั้นตอนถัดไปของระบบคือการกำหนดโครงสร้างของ Frame ต่าง ๆ

2.2.1.1.2 กำหนดโครงสร้างของ Frame

Click ฐานความรู้ที่อยู่ใน List ทางด้านซ้ายมือแล้ว Click Next จะปรากฏหน้าต่างของการกำหนดแก้ไขโครงสร้างของ Frame ดังรูปที่ ก-21



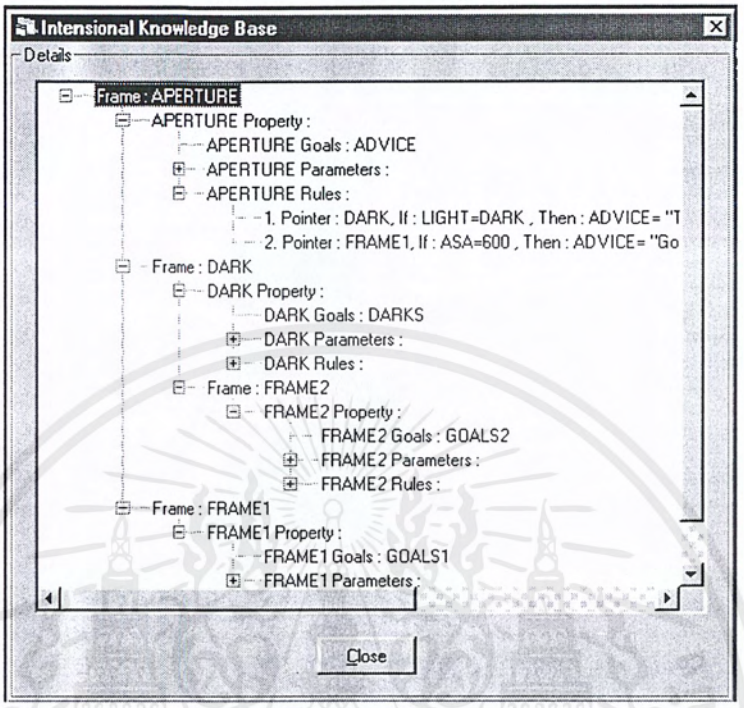
รูปที่ ก-21 หน้าต่างแสดงรายละเอียดการกำหนด Frame

- Create สร้าง Frame ใหม่ให้กับฐานความรู้
- New ล้างค่าที่ปรากฏอยู่ในหน้าจอเพื่อจะทำการสร้าง Frame ใหม่
- Update ทำการแก้ไขรายละเอียดของ Frame ต่าง ๆ
- Frame Characteristics กำหนดรายละเอียดของ Slots ต่าง ๆ และ Rules ให้กับ Frame

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.1.2.1 แสดงรายละเอียดของ Frame

Click ที่ Frame Characteristics เลือกที่ ...(Frame Name)....-Rules แล้ว Click OK จะปรากฏหน้าต่างดังรูปที่ ก-22

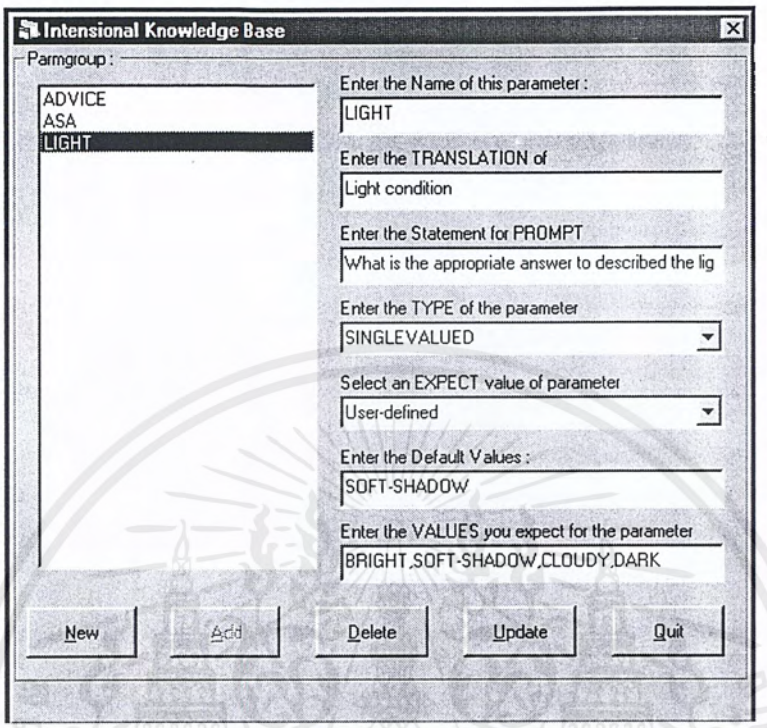


รูปที่ ก-22 หน้าต่างแสดงรายละเอียดของ Frame

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.1.2.2 กำหนด Slots ต่าง ๆ ของ Frame

Click ที่ Frame Characteristics เลือกที่ ...(Frame Name)....-Parameters แล้ว Click OK จะปรากฏหน้าต่างดังรูปที่ ก-23



รูปที่ ก-23 หน้าต่างแสดงรายละเอียดของการกำหนด Slots

- New ล้างค่าที่ปรากฏอยู่บนหน้าจอเพื่อจะทำการกำหนด Slots ใหม่ให้กับ Frame
- Add เพิ่ม Slots ใหม่ให้กับ Frame
- Delete ลบ Slots ที่เลือกออกจาก Frame
- Update แก้ไขรายละเอียดของ Slots

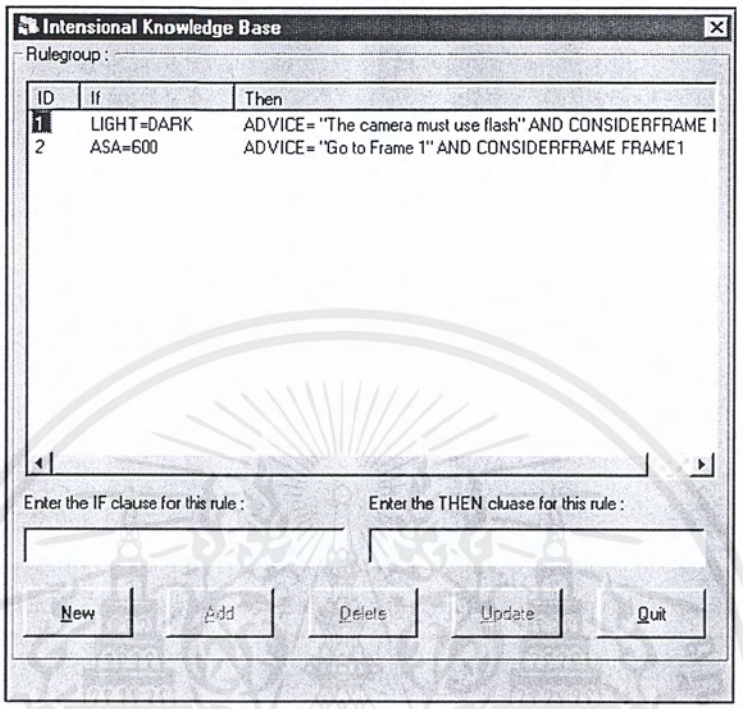
ข้อกำหนดเบื้องต้น

ในส่วนของการกำหนดค่าที่เป็นไปได้ของระบบใน TextBox "Enter the VALUES for expect the parameter" ค่าที่จะใส่ให้ใช้เครื่องหมาย " , " ในการกำหนดค่าหลาย ๆ ค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.1.2.3 กำหนด Rules ให้กับ Frame

Click ที่ Frame Characteristics เลือกที่ ...(Frame Name)....-Rules แล้ว Click OK จะปรากฏหน้าต่างดังรูปที่ ก-24



รูปที่ ก-24 หน้าต่างแสดงรายละเอียดการกำหนด Rules

- New ล้างค่าที่ปรากฏอยู่บนหน้าจอเพื่อจะทำการกำหนด Rules ใหม่ให้กับ Frame
- Add เพิ่ม Rules ใหม่ให้กับ Frame
- Delete ลบ Rules ที่เลือกออกจาก Frame
- Update แก้ไขรายละเอียดของ Rules

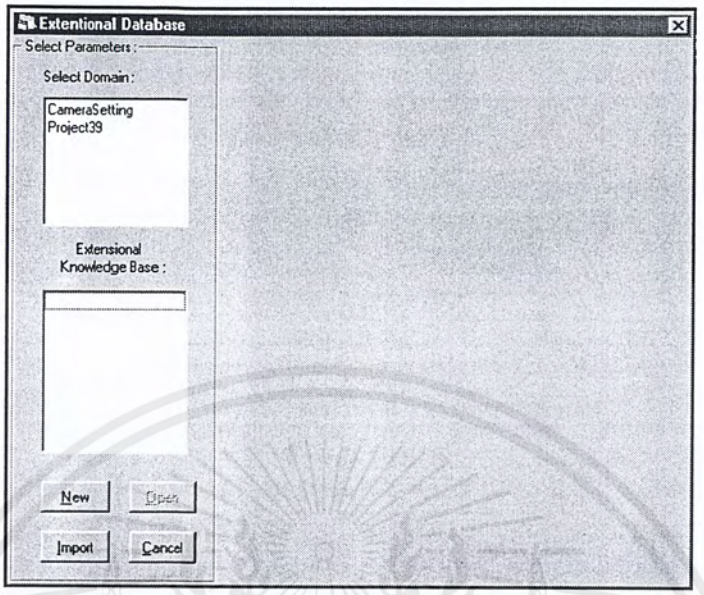
ข้อกำหนดเบื้องต้น

ในการกำหนดให้ Rules ชี้ไปยัง Frame ใด Frame หนึ่งให้ใช้คำว่า "AND CONSIDERFRAME" ตามด้วยชื่อ Frame ที่จะชี้ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.2 Extensional Knowledge Base ส่วนของการกำหนดข้อเท็จจริง (Facts) ให้กับระบบประกอบไปด้วย

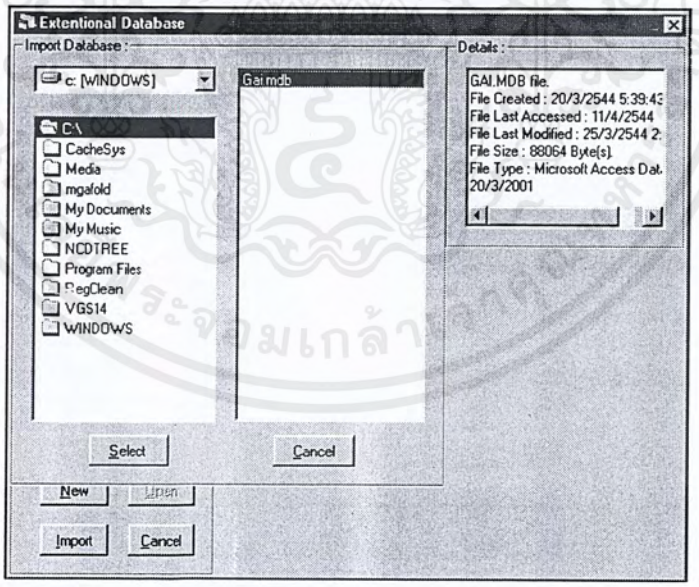
Click Knowledge Base -> Extensional Knowledge Base จะปรากฏหน้าต่างดังรูปที่ ก-25



รูปที่ ก-25 หน้าต่างเริ่มต้นของ Extensional Knowledge Base

2.2.1.2.1 Import จาก Database ภายนอก

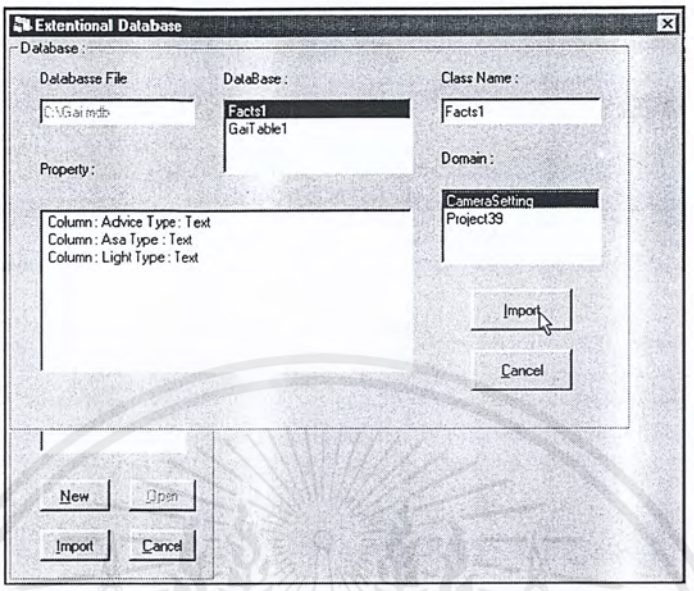
Click Import จะปรากฏหน้าต่างดังรูปที่ ก-26



รูปที่ ก-26 หน้าต่างแสดงการนำเข้าฐานข้อมูลจากภายนอก

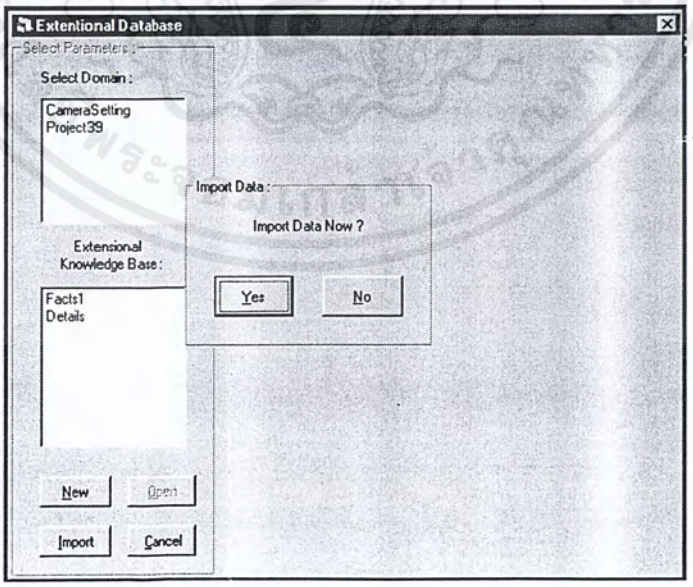
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือก Files ที่จะ Import แล้ว Click ที่ Select จะปรากฏหน้าต่างดังรูปที่ ก-27 ซึ่งแสดงรายละเอียดของ Table รายละเอียดของ Column ของแต่ละ Table ที่อยู่ใน Database จาก Files ที่เลือก



รูปที่ ก-27 หน้าต่างแสดงรายละเอียดของฐานข้อมูล

Click Table ที่ต้องการจะเลือกจาก List ภายใต้หัวข้อ Database ซึ่งในส่วนของ Property จะแสดง Column และ Type ของ Column ในส่วนของ Class Name เป็นส่วนของการกำหนดชื่อที่จะไว้ใช้อ้างอิงภายในฐานความรู้ ในส่วนของ Domain เป็นส่วนของการกำหนดว่าจะให้ Import เข้าไปไว้ในฐานความรู้ไหน เมื่อเลือกครบแล้วให้ Click Import จะปรากฏหน้าต่างดังรูปที่ ก-28 เพื่อถามว่าต้องการ Import Data เข้ามาหรือไม่

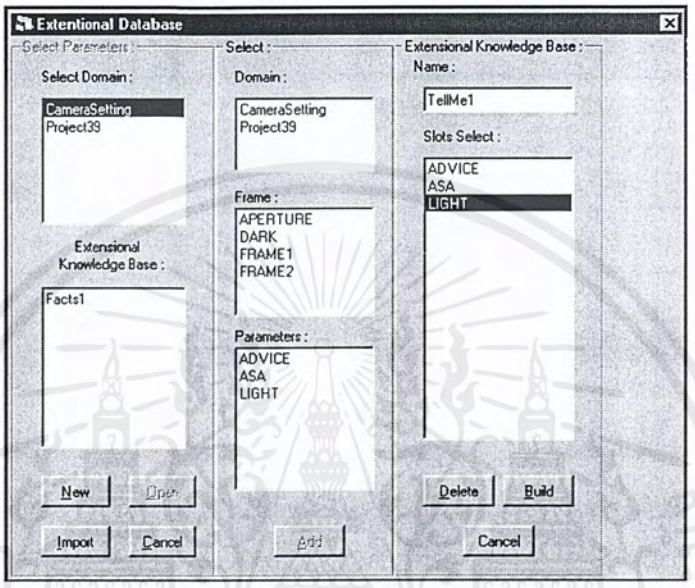


รูปที่ ก-28 หน้าต่างแสดงการติดต่อกับผู้ใช้งานระบบว่าต้องการนำข้อมูลเข้ามาด้วยหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.2.2 สร้างจาก Frame ที่มีอยู่ในฐานความรู้

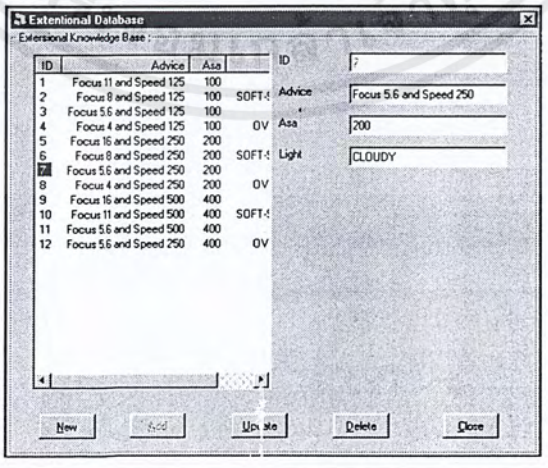
ให้ทำการ Click Domain ที่เราต้องการจะทำการสร้าง ก่อน จากนั้น Click New จะปรากฏหน้าต่างดังรูปที่ ก-29 ตรงส่วนของ Name ในหัวข้อ Extensional Knowledge Base เป็นส่วนของการกำหนดชื่อให้กับข้อเท็จจริงที่เราจะทำการสร้าง แล้วทำการเลือก Domain เลือก Frame แล้วเลือก Slots ที่เราต้องการ แล้ว Click Add เพื่อเพิ่มเข้าไปในข้อเท็จจริง เมื่อครบแล้วให้ Click Build ถ้า Slots ไหนที่ไม่ต้องการให้ Click ที่ Slots แล้ว Click Delete



รูปที่ ก-29 หน้าต่างแสดงรายละเอียดของการกำหนด Extensional Knowledge Base

2.2.1.2.3 แก้ไขเพิ่มเติมข้อเท็จจริง

Click Domain ใต้หัวข้อ Domain จะปรากฏ List ของ Extensional Knowledge Base แล้ว Click Extensional Knowledge Base ที่เราต้องการแก้ไขเพิ่มเติม แล้ว Click ที่ Open จะปรากฏหน้าต่างดังรูปที่ ก-30

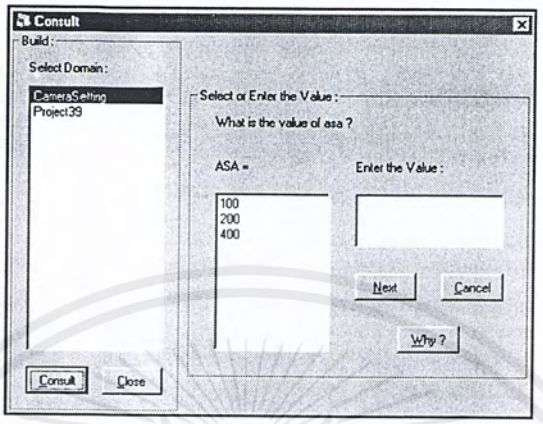


รูปที่ ก-30 หน้าต่างแสดงการเพิ่มเติมแก้ไขข้อเท็จจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

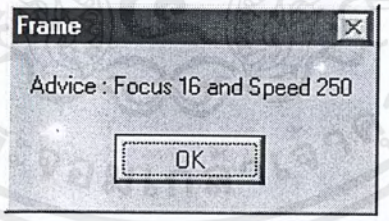
2.2.2 Consult

ผู้ใช้งานระบบจะเข้าใช้งานผ่านทางขั้นตอนนี้ Click Activities -> Consult จะปรากฏหน้าต่างดังรูปที่ ก-31 ซึ่งผู้ใช้งานจะต้องทำการเลือกก่อนว่าจะทำการปรึกษากับฐานความรู้ใด Click Consult ก็จะเริ่มเข้าสู่การให้คำปรึกษาจากระบบ



รูปที่ ก-31 หน้าต่างแสดงการให้คำปรึกษาของระบบ

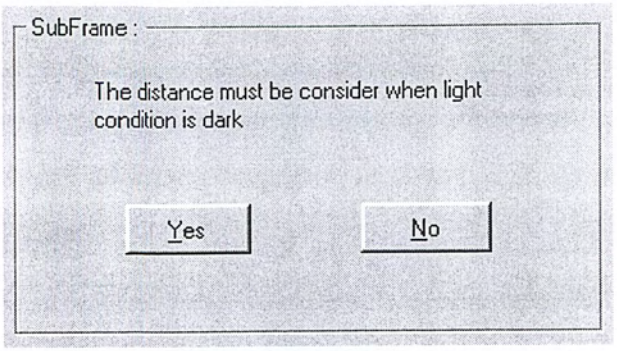
ในรายละเอียดของรูปที่ ก-31 จะประกอบไปด้วย 2 ส่วน คือส่วนทางด้านซ้ายมือจะเป็นการแสดงค่าที่ผู้เชี่ยวชาญได้กำหนดไว้ซึ่งผู้ใช้งานระบบจะเลือกคำตอบจากทางด้านนี้หรือผู้ใช้งานระบบจะใส่คำตอบของผู้ใช้เองทางด้านขวามือก็ได้ จากนั้นก็ให้ทำการ Click Next เพื่อเข้าสู่ขั้นตอนกระบวนการถัดไป แต่ถ้า User ต้องการรู้ถึงสาเหตุที่ทำไมระบบต้องถามก็ให้ทำการ Click Why ? ระบบก็จะแสดงคำตอบให้กับผู้ใช้ เมื่อระบบสอบถามผู้ใช้งานระบบจนครบกระบวนการแล้วก็จะแสดง Message Box แสดงคำตอบออกมาให้ผู้ใช้งานระบบดังเช่นรูปที่ ก-32



รูปที่ ก-32 หน้าต่างแสดงคำตอบของระบบ

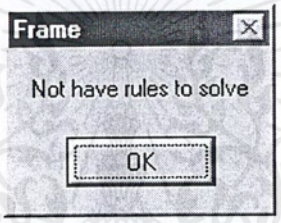
แต่ถ้าระบบได้รับคำตอบจากผู้ใช้งานระบบซึ่งเป็นเงื่อนไขของการไปยัง Frame ถัดไปก็จะปรากฏหน้าต่างดังรูปที่ ก-33 เพื่อถามความต้องการของผู้ใช้งานระบบว่าต้องการทำงานต่อไปหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-33หน้าต่างแสดงการติดต่อกับผู้ใช้ว่าต้องการดำเนินการต่อหรือไม่

แต่ถ้าระบบได้รับคำตอบจากผู้ใช้งานระบบจนครบเสร็จสิ้นแล้วแต่ไม่สามารถให้คำตอบกับผู้ใช้งานระบบได้ก็จะปรากฏหน้าต่างบอกกับผู้ใช้งานระบบดังเช่นรูปที่ ก-34



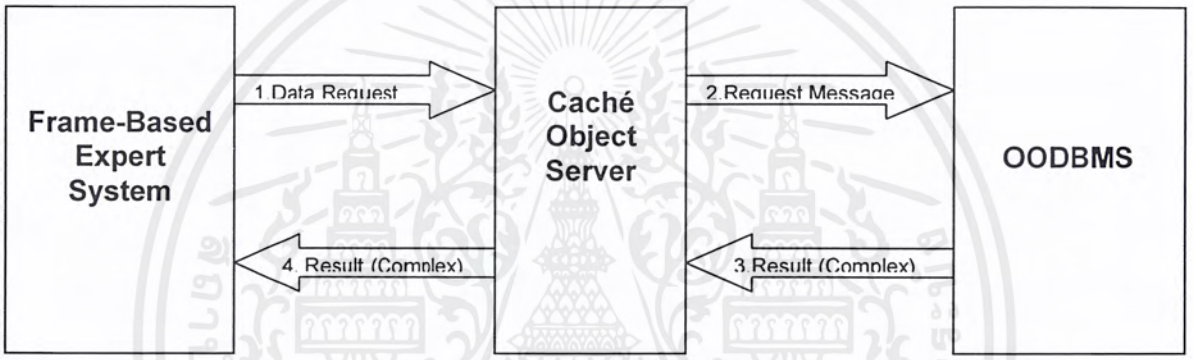
รูปที่ ก-34 หน้าต่างแสดงคำตอบของระบบว่าไม่สามารถให้คำตอบได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.
คู่มือสำหรับโปรแกรมเมอร์

1. ส่วนประกอบหลักของระบบ

สำหรับระบบที่สร้างขึ้นมาในโครงงานนี้จะแบ่งออกเป็นสามส่วนใหญ่ ๆ ด้วยกัน คือ ในส่วนของระบบผู้เชี่ยวชาญซึ่งเป็นส่วนที่ติดต่อโต้ตอบกับผู้ใช้และผู้เชี่ยวชาญ, ส่วนของตัวกลางในการแปลและประมวลผลคำสั่งระหว่างระบบผู้เชี่ยวชาญกับระบบจัดการฐานข้อมูล, และส่วนของระบบจัดการฐานข้อมูลที่จัดเก็บโครงสร้างของฐานความรู้ ข้อเท็จจริงและกฎต่าง ๆ ของระบบผู้เชี่ยวชาญ จากทั้งสามส่วนนี้ รูปแบบและขั้นตอนของการติดต่อจะเป็นดังรูปที่ ข-1



รูปที่ ข-1 สถาปัตยกรรมและขั้นตอนการติดต่อกันระหว่างระบบย่อย

จากรูปที่ ข-1 จะเห็นได้ว่าขั้นตอนที่ระบบผู้เชี่ยวชาญใช้ในการติดต่อจากฐานข้อมูลมีอยู่หลายขั้นตอนด้วยกันซึ่งรายละเอียดของแต่ละขั้นตอนมีดังนี้

1) ทางระบบผู้เชี่ยวชาญต้องการข้อมูลจากฐานข้อมูลและทำการส่งข้อความการร้องขอไปยังตัวกลางที่อยู่ระหว่างระบบผู้เชี่ยวชาญและระบบจัดการฐานข้อมูล

2) ตัวกลางจะทำการส่งข้อความที่ระบบผู้เชี่ยวชาญทำการร้องขอต่อไปยังระบบจัดการฐานข้อมูล

3) ระบบฐานข้อมูลได้รับสัญญาณการร้องขอจากตัวกลางที่ส่งมา และทำการส่งข้อมูลกลับไปยังตัวกลาง ซึ่งข้อมูลที่ส่งกลับมานี้จะมีลักษณะที่เป็นโครงสร้างข้อมูลที่ซับซ้อนอยู่

4) ทางตัวกลางจะทำการส่งข้อมูลคืนกลับไปยังระบบผู้เชี่ยวชาญโดยข้อมูลจะยังมีโครงสร้างข้อมูลที่ซับซ้อนอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ส่วนของระบบผู้เชี่ยวชาญ

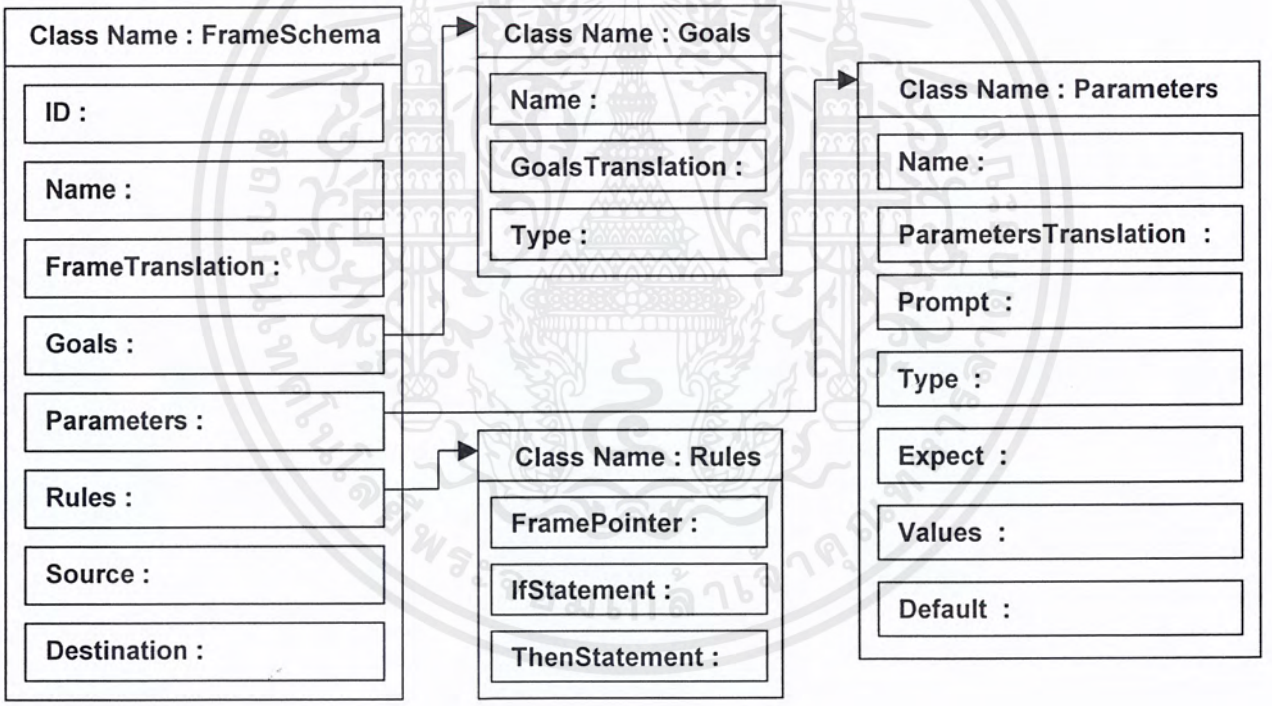
สำหรับในระบบผู้เชี่ยวชาญจะเป็นโปรแกรมที่สร้างขึ้นมาโดยมีลักษณะเป็นระบบผู้เชี่ยวชาญแบบ Frame-Based ซึ่งในโปรแกรมจะประกอบไปด้วยโครงสร้างของความรู้(โครงสร้างของความรู้จะประกอบไปด้วยความรู้และกฎต่าง ๆ) และ ข้อเท็จจริง ต่าง ๆ

2.1 โครงสร้างของฐานความรู้

จากลักษณะโครงสร้างการแสดงความรู้แบบ Frame เราสามารถนำมา implement บนระบบจัดการฐานข้อมูลเชิงวัตถุได้โดยมีลักษณะดังนี้

- Frame จะเป็น Instance ของ Object
- Frame จะประกอบไปด้วย Attribute ต่าง ๆ ซึ่ง Attribute ต่าง ๆ จะเปรียบได้กับ Slots ต่าง ๆ
- Frame จะประกอบไปด้วย Rules ต่าง ๆ ซึ่งแต่ละ Rules จะชี้ไปยัง Frame ต่าง ๆ

ลักษณะโครงสร้างของ Frame จะเป็นดังรูปที่ ข-2



รูปที่ ข-2 โครงสร้างของ Frame

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของ Frame

- **FrameSchema Class** เป็น Persistent Class ที่จะมี ID เป็นของตัวเอง ซึ่งจะเก็บทุก Frame ในฐานความรู้
    - **ID** Type **Number** เป็น ID ที่มีไว้อ้างอิง
    - **Name** Type **String** เก็บชื่อของ Frame
    - **FrameTranslation** Type **String** เก็บความหมายของ Frame
    - **Goals** Type **Goals** เป็น Embeddable Class ที่จะฝังตัวอยู่ใน Persistent ซึ่งไม่มี ID เป็นของตัวเอง
        - **Name** Type **String** เก็บชื่อของ Goals
        - **GoalsTranslation** Type **String** เก็บความหมายของ Goals
        - **Type** Type **String** ชนิดของ Goals
    - **Parameters** Type **Parameters**
        - **Name** Type **String** ชื่อของ Parameters
        - **ParametersTranslation** Type **String** ความหมายของ Parameters
        - **Prompt** Type **String** คำแนะนำของ Parameters
        - **Type** Type **String** ชนิดของ Parameters
        - **Expect** Type **String** ชนิดของค่าที่เป็นไปได้ของ Parameters
        - **Values** Type **String** ค่าที่เป็นไปได้ของ Parameters
        - **Default** Type **String** ค่าเริ่มต้นของ Parameters
    - **Rules** Type **Rules**
        - **FramePointer** Type **String** เก็บชื่อของ Frame ที่ถูกชี้โดย ThenStatement
        - **IfStatement** Type **String** เก็บ Statement ในส่วนของ IF
        - **ThenStatement** Type **String** เก็บ Statement ในส่วนของ Then
    - **Source** Type **String** บอก Frame ที่เกี่ยวข้องกัน
    - **Destination** Type **String** บอกลักษณะของ Frame ว่าเป็น Own หรือ Member ของ Source

2.2 ข้อเท็จจริง

ระบบผู้เชี่ยวชาญจะมีข้อเท็จจริงไว้บอ กถึงค่าที่เป็นจริงที่ผู้เชี่ยวชาญจะเป็นผู้กำหนดไว้ให้ ทำให้ระบบผู้เชี่ยวชาญสามารถตอบคำถามของผู้ใช้ระบบได้ในกรณีที่ไม่สามารถหาคำตอบจากกฎของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3 รายละเอียดของโปรแกรมที่โครงงานได้พัฒนาขึ้นในส่วนของ Microsoft Visual Basic

สำหรับในส่วนของระบบผู้เชี่ยวชาญนี้ทางโครงงานได้ใช้ภาษา Visual Basic 6.0 Professional Edition ในการพัฒนา และใช้ Caché เป็นระบบจัดการฐานข้อมูลเชิงวัตถุซึ่งลักษณะของโปรแกรมที่ได้ทำการพัฒนาขึ้นจะทำหน้าที่ในการรับคำสั่งหรือข้อมูลต่าง ๆ จากผู้ใช้และผู้เชี่ยวชาญแล้วทำการปรับเปลี่ยนให้อยู่ในรูปแบบที่กำหนดไว้แล้วทำการประมวลผลภายในตัวของโปรแกรมเองโดยจะมีการเรียกใช้และจัดเก็บข้อมูลต่าง ๆ กับระบบจัดการฐานข้อมูลเชิงวัตถุ ซึ่งส่วนประกอบสำคัญของระบบผู้เชี่ยวชาญที่ทำการพัฒนาจะประกอบไปด้วยส่วนต่าง ๆ ดังนี้

3.1 การเชื่อมต่อกันระหว่าง Caché และ Visual Basic

การเข้าใช้งาน Caché จะติดต่อผ่านทาง Caché Object Server โดย จะต้องรู้ค่าของ IP ที่กำหนด และ Namespace ที่ต้องการจะใช้ ซึ่ง IP ที่เป็นค่า Default ของ Caché คือ 127.0.0.1 Port 1972

- ทำการ References CacheObject ให้กับ Project ที่สร้าง

  VB Project เลือกที่ **Project/References** และเลือก **CacheObject**

- ประกาศตัวแปรให้เป็นชนิดของ Caché

  Dim Cache As CacheObject.Factory

  Set Cache=CreateObject("CacheObject.Factory")

  หรือ **Dim Cache As New CacheObject.Factory**

- ทำการเชื่อมต่อกับ Cache Server

  Cache.Connect("cn_iptcp:127.0.0.1[1972]:*Namespace***")**

  ตัวอย่างของ *Namespace* ในโครงงานนี้ได้ใช้เป็น "EXPERTSYSTEM" จึงประกาศเป็น

  Cache.Connect("cn_iptcp:127.0.0.1[1972]:EXPERTSYSTEM")

3.2 การแก้ไขข้อมูล

การแก้ไขข้อมูลภายในระบบฐานข้อมูลของ Caché แบ่งออกได้เป็น 2 ชนิดคือ

3.2.1 Persistent Class

เป็น Class ที่มี OID เป็นของตัวเองในแต่ละ Instance และข้อมูลจะถูกเก็บอยู่จริงในฐานข้อมูล

3.2.1.1 Insert

ในการเพิ่มเติมข้อมูลเข้าไปใน Class ที่เป็นแบบ Persistent นั้นมีขั้นตอนดังนี้

- ประกาศตัวแปรชนิด Object ขึ้นมารองรับ

  Dim insertinstance As Object

- ทำการสร้าง Instance ขึ้นมาใหม่ด้วยคำสั่ง

  Set insertinstance = Cache.New(*PersistentClass***)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PersistentClass เป็น Class ที่เราต้องการจะเพิ่มเติมข้อมูล ในที่นี้จะกำหนดเป็น FrameSchema

- กำหนดค่าให้กับ Properties ต่าง ๆ ใน FrameSchema

insertinstance.Name = *Value1*

insertinstance.FrameTranslation = *Value2*

insertinstance.Source = *Value3*

insertinstance.Destination = *Value4*

ในส่วนของ Properties Goals,Parameters และ Rules จะขอกล่าวถึงในหัวข้อที่ 3.2.2.1

- จัดเก็บข้อมูลที่เปลี่ยนแปลง

insertinstance.sys_Save

insertinstance.sys_Close

3.2.1.2 Delete

ในการลบข้อมูลออกจาก Class นั้นต้องทราบถึง ID ที่จะลบ ซึ่งมีขั้นตอนดังนี้

- ประกาศตัวแปรชนิด Object ขึ้นมารองรับ

Dim deleteinstance As Object

- ประกาศตัวแปรขึ้นมารองรับ Operation

Dim tmp As Variant

- เปิด Instance ด้วย ID ที่ต้องการจะลบออกจาก Class

Set deleteinstance = Cache.OpenId(FrameSchema,*ID*)

ID คือ ID ที่ต้องการจะลบออก

- ลบข้อมูลออกจาก Class

tmp = deleteinstance.sys_DeleteId(*ID*)

- จัดเก็บข้อมูลที่เปลี่ยนแปลง

deleteinstance.sys_Save

deleteinstance.sys_Close

3.2.1.3 Update

ในการแก้ไขข้อมูลที่อยู่ใน Class นั้นจะต้องทราบถึง ID ขั้นตอนนั้นจะคล้าย ๆ กับการ Insert แต่จะมีข้อแตกต่างตรงที่จะต้องทำการเปิดข้อมูลที่ต้องการแก้ไขขึ้นมาก่อน

- ประกาศตัวแปรชนิด Object ขึ้นมาเพื่อรับ Instance ที่ต้องการแก้ไข

Dim getinstance As Object

- ประกาศตัวแปรชนิด Object ขึ้นมารองรับค่าใหม่

Dim updateinstance As Object

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำการสร้าง Instance ขึ้นมาใหม่ด้วยคำสั่ง

Set updateinstance = Cache.New(FrameSchema)

- เปิด Instance ที่ต้องการแก้ไข

Set getinstance = Cache.OpenId(FrameSchema,ID)

- กำหนดค่าให้กับ Properties ต่าง ๆ

- ถ้าค่าที่เปลี่ยนแปลงเป็นค่าเก่า

updateinstance.Name = getinstance.Name

updateinstance.FrameTranslation = getinstance.FrameTranslation

- ถ้าค่าที่เปลี่ยนแปลงเป็นค่าใหม่

updateinstance.Name = *Value1*

updateinstance.FrameTranslation = *Value2*

- จัดเก็บข้อมูลที่เปลี่ยนแปลง

updateinstance.sys_Save

updateinstance.sys_Close

getinstance.sys_Close

* ถ้าข้อมูลที่เราเปิดขึ้นมาแต่ไม่มีการแก้ไขก็ Close เพียงอย่างเดียวได้เลย

3.2.2 Embeddable Class

เป็น Class ที่ไม่มี OID เป็นของตัวเอง และข้อมูลจะถูกฝังอยู่ใน Persistent Class ไม่สามารถเก็บอยู่ด้วยตัวของตัวเองได้ ในกรณีนี้ประกาศ Collection ใน Caché เป็นแบบ List

3.2.2.1 Insert

ในการเพิ่มเติมข้อมูลแบบ Embeddable Class นั้นมีขั้นตอนดังนี้

- ประกาศตัวแปรชนิด Object ขึ้นมารองรับทั้ง Persistent และ Embeddable

Dim insertinstance As Object

Dim embedinstance As Object

- ทำการสร้าง Instance ขึ้นมาใหม่ด้วยคำสั่งทั้ง Persistent และ Embeddable

Set insertinstance = Cache.New(*PersistentClass*)

Set embedinstance = Cache.New(*EmbeddableClass*)

PersistentClass เป็น Class ที่เราต้องการจะเพิ่มเติมข้อมูล ในที่นี้จะกำหนดเป็น FrameSchema

EmbeddableClass เป็น Class ที่เราต้องการจะเพิ่มเติมข้อมูล ในที่นี้จะกำหนดเป็น Rules

- ประกาศตัวแปรขึ้นมารองรับ Operation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dim tmp As Variant

- กำหนดค่าให้กับ Properties ต่าง ๆ ใน FrameSchema

Properties ที่เป็นแบบ Persistent

insertinstance.Name = *Value1*

insertinstance.FrameTranslation = *Value2*

insertinstance.Source = *Value3*

insertinstance.Destination = *Value4*

Properties ที่เป็นแบบ Embeddable

- กำหนดค่าให้กับ Embeddable Class ขึ้นมาก่อน

embedinstance.FramePointer = *Value5*

embedinstance.IfStatement = *Value6*

embedinstance.ThenStatement = *Value7*

- กำหนดค่าให้กับ Persistent ที่เป็นแบบ Embeddable

ถ้าต้องการเพิ่มค่าต่อท้ายเข้าไปใน List

tmp = insertinstance.Rules.Insert(embedinstance)

แต่ถ้าต้องการเพิ่มค่าไปในตำแหน่งใด ๆ

tmp = insertinstance.Rules.InsertAt(embedinstance,*OID*)

OID เป็นตำแหน่งที่ต้องการจะเพิ่มค่าเข้าไป

- จัดเก็บข้อมูลที่เปลี่ยนแปลง

insertinstance.sys_Save

insertinstance.sys_Close

embedinstance.sys_Close

* ถ้าเป็น Embeddable Class จะ Close ได้เพียงอย่างเดียวเท่านั้น เพราะข้อมูลที่จะเก็บจะฝังอยู่ใน Persistent Class แล้ว

3.2.2.2 Delete

ในการลบข้อมูลนั้นจะมีขั้นตอนคล้าย ๆ กับการ Delete ใน Persistent Class ข้อแตกต่างคือต้องทราบถึงตำแหน่งที่ต้องการจะลบออกจาก List ด้วย

- ประกาศตัวแปรชนิด Object ขึ้นมารองรับ

Dim deleteembed As Object

- ประกาศตัวแปรขึ้นมารองรับ Operation

Dim tmp As Variant

- เปิด Instance ด้วย ID ที่ต้องการจะลบออกจาก Class

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Set deleteembed = Cache.OpenId(FrameSchema,*ID*)

ID คือ ID ที่ต้องการจะลบออก

- ลบข้อมูลออกจาก List ณ ตำแหน่งที่ต้องการ

deleteembed.Rules.RemoveAt(*OID*)

OID คือ ID ที่ต้องการลบออกจาก List

- จัดเก็บข้อมูลที่เปลี่ยนแปลง

deleteinstance.sys_Save

deleteinstance.sys_Close

3.2.2.3 Update

ในการแก้ไขข้อมูลที่อยู่ใน Class นั้นจะต้องทราบถึง ID ของ Persistent และ ID ของ Embeddable ขั้นตอนนั้นจะคล้าย ๆ กับการ Insert แต่จะมีข้อแตกต่างตรงที่จะต้องทำการเปิดข้อมูลที่ต้องการแก้ไขขึ้นมาก่อน

- ประกาศตัวแปรชนิด Object ขึ้นมาเพื่อรับ Instance ที่ต้องการแก้ไข

Dim getinstance As Object

- ประกาศตัวแปรชนิด Object ขึ้นมารองรับค่าใหม่ทั้ง Persistent และ Embeddable

Dim updateinstance As Object

Dim embedinstance As Object

- ทำการสร้าง Instance ขึ้นมาใหม่ด้วยคำสั่งทั้ง Persistent และ Embeddable

Set updateinstance = Cache.New(FrameSchema)

Set embedinstance = Cache.New(Rules)

- เปิด Instance ที่ต้องการแก้ไข

Set getinstance = Cache.OpenId(FrameSchema,ID)

- กำหนดค่าให้กับ Properties ต่าง ๆ

Properties ที่เป็นแบบ Persistent

- ถ้าค่าที่เปลี่ยนแปลงเป็นค่าเก่า

updateinstance.Name = getinstance.Name

updateinstance.FrameTranslation = getinstance.FrameTranslation

- ถ้าค่าที่เปลี่ยนแปลงเป็นค่าใหม่

updateinstance.Name = *Value1*

updateinstance.FrameTranslation = *Value2*

Properties ที่เป็นแบบ Embeddable

updateinstance.Rules.SetAt(embedinstance,*OID*)

OID เป็น ID ตำแหน่งของ List ที่เราต้องการแก้ไข

- จัดเก็บข้อมูลที่เปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

updateinstance.sys_Save

updateinstance.sys_Close

getinstance.sys_Close

* ถ้าข้อมูลที่เราเปิดขึ้นมาแต่ไม่มีการแก้ไขก็ Close เพียงอย่างเดียวได้เลย

embedinstance.sys_Close

*ถ้าเป็น Embeddable Class จะ Close ได้เพียงอย่างเดียวเท่านั้น เพราะข้อมูลที่จะเก็บจะฝังอยู่ใน Persistent Class แล้ว

3.3 การเรียกข้อมูลด้วย SQL Statement

ในการเรียกข้อมูลด้วย SQL Statement ผ่านทาง VB นั้นมีขั้นตอนดังนี้

- ประกาศตัวแปรชนิด Object ขึ้นมารองรับ

Dim queryinstance As Object

- ทำการสร้าง Instance ขึ้นมาใหม่ด้วยคำสั่ง

Set queryinstance = Cache.DynamicSQL(*SqlStatement*)

SqlStatement เป็นชุดคำสั่ง SQL เช่น Select * from *PersistentClass*

PersistentClass เป็น Class ที่เราต้องการจะเพิ่มเติมข้อมูล ในที่นี้จะกำหนดเป็น FrameSchema

- ทำการประมวลผล Sql Statement

queryinstance.Execute

ข้อมูลที่ได้ออกมาจะคล้าย ๆ กับตารางใน Relational

ถ้าเราต้องการเลื่อนไปยัง Tuple ถัดไป ก็ให้ใช้คำสั่ง

queryinstance.Next

ถ้าอยู่ที่ Tuple สุดท้ายแล้ว คำสั่ง **queryinstance.Next** จะเป็น **False** ซึ่งเราสามารถนำไปใช้งานได้

3.4 รายละเอียดของ Code ในโครงงาน

ในโครงงานที่ได้พัฒนาด้วย Microsoft Visual Basic จะประกอบไปด้วย Form ทั้งหมด 4 Form ประกอบไปด้วย

- MDIMain เป็นหน้าจอเริ่มต้นสำหรับโปรแกรม
- formDevelop เป็นส่วนของการกำหนดโครงสร้างของฐานความรู้, โครงสร้างของ Frame, Slots และ Rules ต่าง ๆ
- formDatabase เป็นส่วนของการจัดการทางด้านข้อเท็จจริงทั้งการดึงข้อมูลจาก Database ภายนอก และจากโครงสร้างของฐานความรู้
- formCousult เป็นส่วนของการประมวลผลในการโต้ตอบกับผู้ใช้งานระบบ

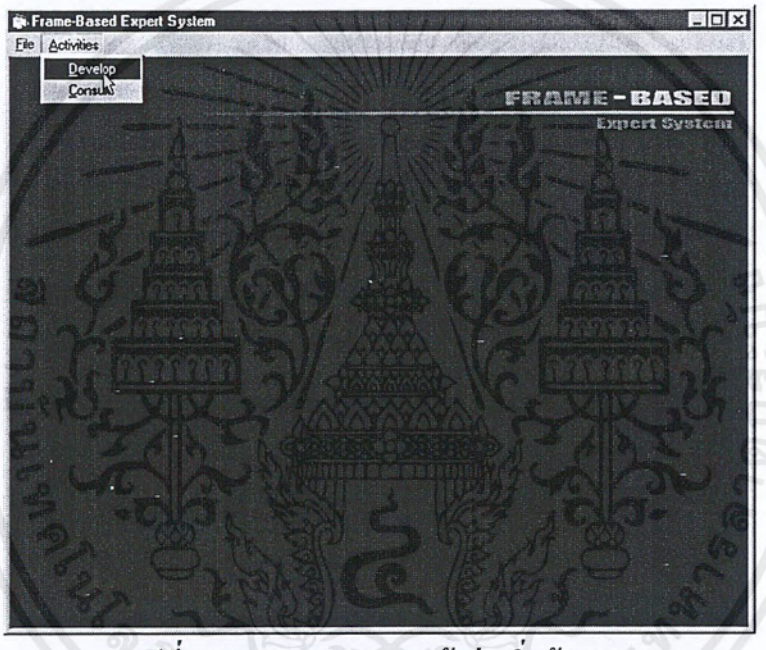
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 MDIMain

ใน Form นี้จะเป็นส่วนในการเข้าใช้งานระบบและออกจากระบบ โดยการเข้าใช้งานระบบจะแบ่งออกเป็น 2 ส่วนคือ ผู้เชี่ยวชาญจะเข้าใช้งานผ่านทาง Develop และ ผู้ใช้ระบบจะเข้าใช้งานผ่านทาง Consult Form นี้หน้าต่างจะเป็นดังรูปที่ ข-3

Code ใน Form นี้จะเป็นส่วนจัดการการ Load Form ที่เกี่ยวข้องต่าง ๆ เท่านั้น ด้วย Event mnu_Click เช่น

```
Private Sub mnuActConsult_Click()
  Load formConsult
End Sub
```



รูปที่ ข-3 MDIMain Form หน้าต่างเริ่มต้นของระบบ

3.4.2 formDevelop

ใน Form นี้จะเป็นส่วนของการจัดการทางด้านการติดต่อกับ Caché, การกำหนดโครงสร้างของ Frame, การกำหนด Slost, การกำหนด Rules, แสดงรายละเอียดของ Frame

- การติดต่อกับ Caché

```
Private Sub ConnectToCache()
  If cache.Connect("cn_iptcp:127.0.0.1[1972]:EXPERTSYSTEM") Then
    MsgBox ("Connect to Cache OK")
  Else
    MsgBox ("Can't Connect to Cache")
  End If
```

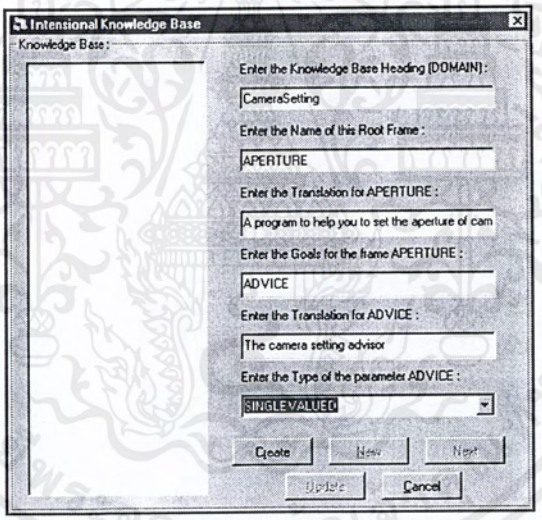
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub
```

- การ Load Combo Box และ List Box ต่าง ๆ เช่น

```
Private Sub LoadTypeGoalsCombo(selectcombo As Object)
   Set pers = cache.Static(TCGoals)
   displays = pers.GoalsTypeDisplay
   delimd = Left(displays, 1)
   types = Split(displays, delimd)
   selectcombo.Clear
   For i = 1 To UBound(types)
      selectcombo.AddItem types(i)
   Next i
End Sub
```

- การจัดการทางด้านฐานความรู้ (Domain) ดังรูปที่ ข-4



รูปที่ ข-4 หน้าต่างแสดงรายละเอียดของการกำหนดฐานความรู้

รายละเอียดของ Code จะมี Procedure ต่าง ๆ ที่สำคัญคือ

DomainProcess(tagDomain)

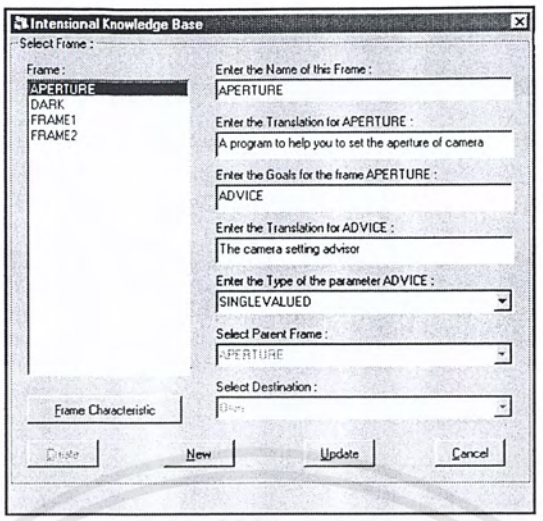
เป็นส่วนของการจัดการทางด้านการตรวจสอบความถูกต้องของข้อมูล การเพิ่มเติมแก้ไขข้อมูลเข้าไปใน Caché ซึ่งจะมี **tagDomain** บอกว่าเป็นการเพิ่มเติมหรือการแก้ไข

lstDomain_Click()

เมื่อผู้เชี่ยวชาญ Click ที่ List ของ Domain ก็จะมีส่วนจัดการในการด้านการค้นหาข้อมูล และการกำหนดค่าตัวแปรต่าง ๆ ให้กับ Procedure ถัด ๆ ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การจัดการ Frame ดังรูปที่ ข-5



รูปที่ ข-5 หน้าต่างแสดงรายละเอียดการกำหนด Frame

รายละเอียดของ Code จะมี Procedure ต่าง ๆ ที่สำคัญคือ

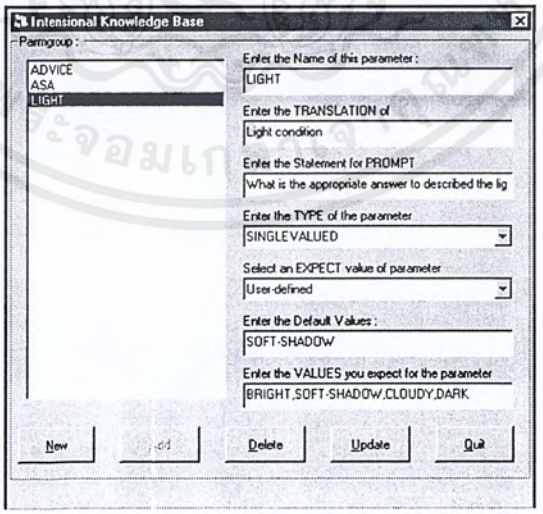
FrameProcess(tagFrame)

เป็นส่วนของการจัดการทางด้านการตรวจสอบความถูกต้องของข้อมูล การเพิ่มเติมแก้ไขข้อมูลเข้าไปใน Caché ซึ่งจะมี **tagFrame** บอกว่าเป็นการเพิ่มเติมหรือการแก้ไข

lstFrameList_Click()

เมื่อผู้เชี่ยวชาญ Click ที่ List ของ Frame ก็จะมีส่วนจัดการในการด้านการค้นหาข้อมูล และการกำหนดค่าตัวแปรต่าง ๆ ให้กับ Procedure ถัด ๆ ไป

- การจัดการ Slots ดังรูปที่ ข-6



รูปที่ ข – 6 หน้าต่างแสดงรายละเอียดของการกำหนด Slots

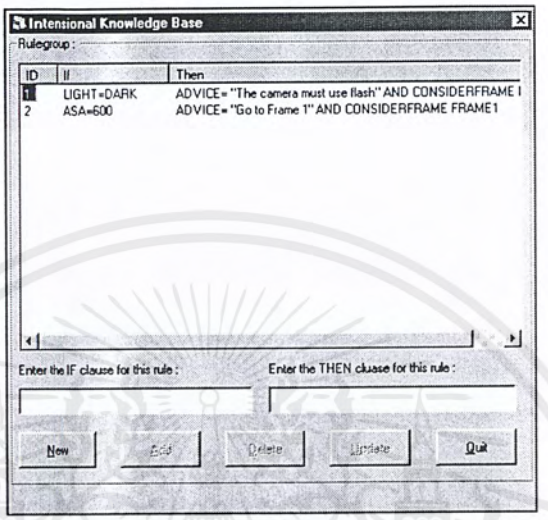
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของ Code จะมี Procedure ต่าง ๆ ที่สำคัญคือ

ParameterProcess(tagParameter)

เป็นส่วนของการจัดการทางด้านการตรวจสอบความถูกต้องของข้อมูล การเพิ่มเติมแก้ไขข้อมูลเข้าไปใน Caché ซึ่งจะมี **tagParameter** บอกว่าเป็นการเพิ่มเติมหรือการแก้ไข

- การจัดการ Rules ดังรูปที่ ข - 7



รูปที่ ข-7 หน้าต่างแสดงรายละเอียดการกำหนด Rules

รายละเอียดของ Code จะมี Procedure ต่าง ๆ ที่สำคัญคือ

RuleProcess(tagRule)

เป็นส่วนของการจัดการทางด้านการตรวจสอบความถูกต้องของข้อมูล การเพิ่มเติมแก้ไขข้อมูลเข้าไปใน Caché ซึ่งจะมี **tagRule** บอกว่าเป็นการเพิ่มเติมหรือการแก้ไข

ManageRule(ByRef IfString,ByRef ThenString)

เป็นการจัดการแปลงข้อมูลที่รับเข้ามาให้อยู่ในรูปแบบที่กำหนด ซึ่งมีรายละเอียดดังนี้

- IfStatement

ในการเก็บค่านั้นจะเก็บเฉพาะ Operation และ Value เข้าไปไว้ใน Caché โดยจะใช้คำว่า AND ในการแบ่ง โดยแทนที่ด้วยเครื่องหมาย "/" ในการแบ่งให้ตรงกับ Slots ต่าง ๆ ของ Frame เช่น

Frame Aperture จะมี Slots **Advice,Asa,Light**

ซึ่งลักษณะการเก็บใน Caché จะเป็นดังนี้

//

ถ้าข้อมูลที่รับเข้ามาเป็น Asa=600 AND Light = Dark ก็จะจัดรูปแบบให้เป็นดังนี้

/=600/=Dark

- ThenStatement

ในการรับค่านั้นจะใช้คำว่า AND ในการแบ่งแล้วแทนที่ด้วยเครื่องหมาย "/"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่นข้อมูลที่รับเข้ามาเป็น ADVICE = “Set focus to 11” AND CONSIDEFFRAME Frame จะจัดรูปแบบให้เป็นดังนี้

=”Set focus to 11”/Frame

* CONSIDERFRAME เป็นคำเฉพาะในการที่บอกให้โปรแกรมชี้ไปยัง Frame ใด ๆ

- แสดงรายละเอียดของ Frame

LoadTreeView(tmp)

เป็นส่วนของการจัดการในการแสดงรายละเอียดของ Frame ให้อยู่ในรูปของ Tree

- การจัดการเชื่อมโยง Rule ไปยัง Frame ต่าง ๆ

RelateFrame

เป็นส่วนของการจัดการ Property **FramePointer** ให้ชี้ไปยัง Frame ที่ถูกต้อง

3.4.3 formDatabase

Form นี้จะจัดการเกี่ยวกับข้อเท็จจริง ทั้งการนำเข้าจากฐานข้อมูลอื่น ๆ และโครงสร้างของข้อเท็จจริงจาก Frame ที่สร้างและแก้ไขเพิ่มเติมข้อเท็จจริง

- การนำเข้าข้อมูลจากฐานฐานข้อมูลภายนอก

โปรแกรมที่ทางโครงงานนี้ได้พัฒนาขึ้นมาจะสนับสนุนฐานข้อมูลจากภายนอกเพียง Microsoft Access ซึ่งถ้าต้องการให้สนับสนุนฐานข้อมูลจากแหล่งอื่นก็สามารถทำได้โดยอาศัยหลักการจากโครงงานนี้ ใน Form นี้จะมี Procedure หลัก ๆ ที่สำคัญดังนี้

GetTableAndColumn

ทำหน้าที่อ่านข้อมูลจากฐานข้อมูลว่าประกอบไปด้วยตารางอะไรบ้างและในแต่ละตารางประกอบไปด้วย Column อะไรบ้างซึ่งตารางทั้งหมดที่อ่านได้จะประกอบไปด้วยตารางที่ผู้ใช้สร้างและตารางของระบบแต่จะใช้เพียงเฉพาะตารางที่ผู้ใช้สร้างเท่านั้น

GetTypeDB

ทำหน้าที่อ่านค่าชนิดของ Column ของแต่ละตาราง ซึ่งค่าของชนิดที่อ่านได้จะเป็นตัวเลขซึ่งจะต้องทำการเปรียบเทียบกับชนิดที่แท้จริง ใน Microsoft Access จะเป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Text	202
Memo	206
Number	3,2
Date/time	7
Currency	6
AutoNumber	3
YESNO	11
OLE Object	205

ตารางที่ ข-1 แสดงค่าที่อ่านได้และชนิดของ Column ต่าง ๆใน Microsoft Access

cmdDBImport_Click

ในส่วนของ Procedure นี้จะทำการนำโครงสร้างของตารางที่ผู้ใช้งานระบบเลือกแล้วทำการสร้าง Class เข้าไปไว้ใน Caché ด้วย Method ใน Caché รายละเอียดของในส่วนนี้จะอยู่ในหัวข้อที่ 4

cmdImportYes_Click

Procedure นี้จะอ่านข้อมูลที่อยู่ในตารางทั้งหมดแล้วนำเข้าไปสร้างไว้ใน Caché

3.4.4 formConsult

form นี้จะทำหน้าที่รับข้อมูลจากผู้ใช้งานมาระบบมาประมวลผลเพื่อหาคำตอบให้กับผู้ใช้งานระบบ ประกอบไปด้วย Procedure ที่สำคัญ ๆ ดังนี้

ManageVars

จะสร้าง Array ขึ้นมาเพื่อเก็บรายละเอียดทั้งหมดของ Frame ประกอบไปด้วย ชื่อ Frame, Rules ที่จะชี้ไปยัง Frame อื่น ๆ, Frame ที่ถูกชี้

RestoreRules

อ่านค่าที่เก็บไว้ใน Cache แล้วจัดรูปแบบให้อยู่ในรูปที่ผู้ใช้งานระบบเข้าใจได้ง่าย

ManageConsult

ทำหน้าที่ในการหาคำตอบให้กับผู้ใช้ระบบและทำการแสดงรายละเอียดของ Slots ที่ต้องนำมาถามผู้ใช้งานระบบ ใน Procedure นี้จะมี Procedure ที่เกี่ยวข้องอยู่หลาย Proceduer ประกอบไปด้วย

- ShowParameter

นำรายละเอียดของ Slots ขึ้นมาแสดงให้กับผู้ใช้ระบบ

- FindEKB

รับค่าจากผู้ใช้งานระบบแล้วนำไปค้นในข้อเท็จจริงเพื่อหาข้อเท็จจริงทั้งหมดที่เป็นไปได้

- FindRules

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

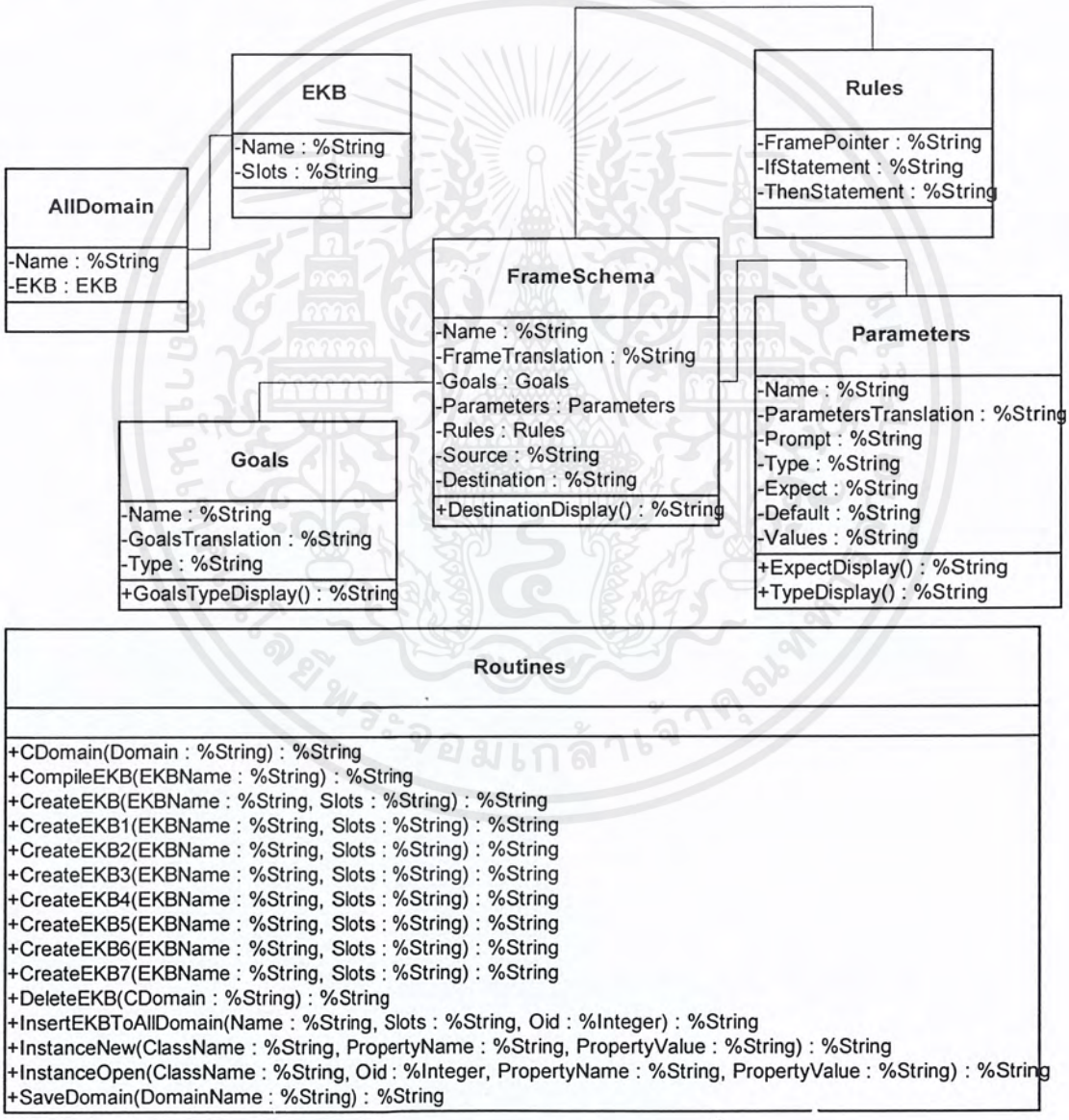
รับค่าจากผู้ใช้งานระบบมาค้นหา Rules ที่เกี่ยวข้อง ซึ่งใน Procedure นี้จะมี Procedure ที่เกี่ยวข้องคือ

- CompareAndOperation

จะนำค่าที่รับจากผู้ใช้งานระบบมาเปรียบเทียบกับค่าที่อ่านได้จากฐานข้อมูลเพื่อหาข้อสรุปของ Rules ทั้งหมดที่เกี่ยวข้องและเป็นไปได้

4 รายละเอียดของโปรแกรมที่โครงงานได้พัฒนาขึ้นในส่วนของ Caché และรายละเอียดของ Class

ในส่วนนี้จะไม่ขอกล่าวถึงการติดตั้งและการใช้งาน Caché แต่จะอธิบายถึงรายละเอียดของแต่ละ Class ว่าประกอบไปด้วยอะไรบ้างและแต่ละ Class เกี่ยวข้องกันยังไงดังรูปที่ ข-8



รูปที่ ข-8 แสดงรายละเอียดของ Class ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 รายละเอียดของ Class AllDomain

- **AllDomain** Class เป็น Class แบบ Persistent ซึ่งจะเก็บว่าใน Namespace จะประกอบไปด้วยฐานความรู้อะไรบ้างและในแต่ละฐานความรู้จะประกอบไปด้วย Class ของข้อเท็จจริงอะไรบ้าง

Properties

- **Name** : **%String** เก็บชื่อของฐานความรู้
- **EKB** : **EKB** เก็บรายละเอียดของข้อเท็จจริงว่าประกอบไปด้วย Slots อะไรบ้างแต่ไม่ได้เก็บข้อมูลจริง ๆ เก็บเพียงแค่โครงสร้าง

4.2 รายละเอียดของ Class EKB

- **EKB** Class เป็น Class แบบ Embeddable ที่จะเก็บชื่อของ Class ของข้อเท็จจริงและรายละเอียดของโครงสร้างของข้อเท็จจริง

Properties

- **Name** : **%String** เก็บชื่อของข้อเท็จจริง
- **Slots** : **%String** เก็บรายละเอียดของ Class ของข้อเท็จจริงว่าประกอบด้วย Slots อะไรบ้าง

4.3 รายละเอียดของ Class FrameSchema

- **FrameSchema** Class เป็น Class แบบ Persistent และเป็น Class ต้นแบบที่จะเก็บรายละเอียดปลีกย่อยต่าง ๆ ของ Frame

Properties

- **Name** : **%String** เก็บชื่อของ Frame
- **FrameTranslation** : **%String** เก็บความหมายของ Frame
- **Goals** : **Goals** เก็บรายละเอียดของเป้าหมายของ Frame
- **Parameters** : **Parameters** เก็บรายละเอียดของ Slots ต่าง ๆ ลักษณะการเก็บจะเป็น List
- **Rules** : **Rules** เก็บรายละเอียดของ Rules ต่าง ๆ ลักษณะการเก็บจะเป็น List
- **Source** : **%String** เก็บชื่อของ Frame ที่เกี่ยวข้อง
- **Destination** : **%String** บอกความสัมพันธ์ของ Frame ที่เกี่ยวข้องว่าเป็น Own หรือ Member

Methods

- **DestinationDisplay()** เป็น Method ที่แสดงรายละเอียดที่เก็บอยู่ใน Property Destination

4.4 รายละเอียดของ Class Goals

- **Goals** Class เป็น Class แบบ Embeddable และเป็น Class ต้นแบบในการเก็บรายละเอียด

Properties

- **Name** : **%String** เก็บชื่อของเป้าหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **GoalsTranslation** : %String เก็บรายละเอียดของเป้าหมาย
- **Type** : %String เก็บชนิดของเป้าหมายว่าเป็นแบบ YES/NO, SINGLEVALUED, MULTIVALUED หรือ ASK-ALL

Methods

- **GoalsTypeDisplay()** เป็น Method ที่แสดงรายละเอียดที่เก็บอยู่ใน Property Type

4.5 รายละเอียดของ Class Parameters

- **Parameters** Class เป็น Class แบบ Embeddable และเป็น Class ต้นแบบในการเก็บรายละเอียด

Properties

- **Name** : %String เก็บชื่อของ Slots
- **ParametersTranslation** : %String เก็บรายละเอียดของ Slots
- **Prompt** : %String เก็บคำถามสำหรับตอบผู้ใช้งานระบบถึงเหตุผลที่ถาม
- **Type** : %String เก็บชนิดของ Slots ว่าเป็นแบบ YES/NO, SINGLEVALUED, MULTIVALUED หรือ ASK-ALL
- **Expect** : %String เก็บชนิดของค่าที่เป็นไปได้ว่าเป็นแบบ SINGLE-LINE-INPUT, MULTI-LINE-INPUT, INTEGER, NUMBER, POSITIVE-NUMBER หรือ User-defined
- **Default** : %String เก็บค่าปกติของ Slots
- **Values** : %String เก็บค่าที่เป็นไปได้ทั้งหมด

Methods

- **ExpectDisplay()** เป็น Method ที่แสดงรายละเอียดที่เก็บอยู่ใน Property Expect
- **TypeDisplay()** เป็น Method ที่แสดงรายละเอียดที่เก็บอยู่ใน Property Type

4.6 รายละเอียดของ Class Rules

- **Rules** Class เป็น Class แบบ Embeddable และเป็น Class ต้นแบบในการเก็บรายละเอียด

Properties

- **FramePointer** : %String เก็บชื่อของ Frame ที่ถูกชี้ด้วย Property ThenStatement
- **IfStatement** : %String เก็บประโยคที่อยู่ในส่วนของ If ด้วยรูปแบบที่กำหนด
- **ThenStatement** : %String เก็บประโยคที่อยู่ในส่วนของ Then ด้วยรูปแบบที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 รายละเอียดของ Class Routines

- Routines Class เป็น Class แบบ Persistent ที่เป็น Class กลางซึ่งเก็บ Methods ทั้งหมด

Methods

- CDomain(Domain : %String) สำหรับสร้าง Class สำหรับเก็บ ทุก ๆ Frame ในฐานความรู้ และทำการ Compile เพื่อให้ใช้งานได้
- CompileEKB(EKBName : %String) สำหรับทำการ Compile EKB Class ที่ได้สร้างไว้แล้ว
- CreateEKB(EKBName : %String,Slots : %String) สำหรับสร้าง Class EKB และสร้าง Property
- CreateEKB1(EKBName : %String,Slots : %String) สำหรับเพิ่ม Property เข้าไปใน EKB Class ในกรณีที่ชนิดของ Property เป็น Number
- CreateEKB2(EKBName : %String,Slots : %String) สำหรับเพิ่ม Property เข้าไปใน EKB Class ในกรณีที่ชนิดของ Property เป็น Currency
- CreateEKB3(EKBName : %String,Slots : %String) สำหรับเพิ่ม Property เข้าไปใน EKB Class ในกรณีที่ชนิดของ Property เป็น Date
- CreateEKB4(EKBName : %String,Slots : %String) สำหรับเพิ่ม Property เข้าไปใน EKB Class ในกรณีที่ชนิดของ Property เป็น YES/NO
- CreateEKB5(EKBName : %String,Slots : %String) สำหรับเพิ่ม Property เข้าไปใน EKB Class ในกรณีที่ชนิดของ Property เป็น String
- CreateEKB6(EKBName : %String,Slots : %String) สำหรับเพิ่ม Property เข้าไปใน EKB Class ในกรณีที่ชนิดของ Property เป็น Memo
- CreateEKB7(EKBName : %String,Slots : %String) สำหรับเพิ่ม Property เข้าไปใน EKB Class ในกรณีที่ชนิดของ Property เป็น Stream
- DeleteEKB(CName : %String) สำหรับลบ Class ออกจากฐานข้อมูล
- InsertEKBToAllDomain(Name : %String,Slots : %String,Oid : %Integer) สำหรับเก็บรายละเอียดของ EKB Class เข้าไว้ใน AllDomain Class
- InstanceNew(ClassName :%String,PropertyName :%String,PropertyValue :%String) สำหรับเพิ่ม Instance เข้าไปใน Class
- InstanceOpen(ClassName :%String,Oid :%Integer,PropertyName :%String, PropertyValue :%String) สำหรับแก้ไขค่าของ Instance ที่ตำแหน่ง Oid
- SaveDomain(DomainName :%String) สำหรับเพิ่ม Instance เข้าไปไว้ใน AllDomain Class

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้