



บทที่ 3

การออกแบบโครงสร้างระบบผู้เชี่ยวชาญ

การออกแบบโครงสร้างระบบผู้เชี่ยวชาญ

จากการวิเคราะห์ปัญหาและแนวทางในการสร้างโปรแกรมโครงสร้างระบบผู้เชี่ยวชาญที่กล่าวมาแล้วนั้น ทำให้การออกแบบสามารถแบ่งออกเป็นส่วนต่างๆ คือ การออกแบบรูปแบบของกฎและ attribute ที่จะใช้, การออกแบบหน้าที่ของโครงสร้างระบบผู้เชี่ยวชาญ ทั้งในส่วนจัดการกับความรู้และส่วนให้คำปรึกษา, การออกแบบโครงสร้างข้อมูลของระบบ, และการออกแบบการทำงานของโครงสร้างระบบผู้เชี่ยวชาญ

1. รูปแบบของกฎ

นอกจากกฎจะมีส่วนเงื่อนไขและส่วนสรุป ซึ่งประกอบด้วยอนุประโยคแล้ว ยังจะต้องมีส่วนสำคัญอีกส่วนหนึ่งคือส่วนที่เป็นคำแนะนำหรือข้อสรุปสำหรับแจ้งให้ผู้ใช้ทราบในระหว่างทำการปรึกษา อันเป็นความต้องการเนื่องจากลักษณะพิเศษของงาน ซึ่งคำแนะนำหรือข้อสรุปนี้ ผู้ใช้ความรู้เป็นผู้กำหนดขึ้นและต้องใส่ไว้ในฐานความรู้ด้วย ดังนั้นกฎที่ใช้จึงมีรูปแบบดังนี้

```
RULE: XXXXX          IF:  1. (A-V)1
                               2. (A-V)2
                               .
                               .
                               .
                               THEN: 1. (A-V)a
                                       2. (A-V)b
                                       .
                                       .
```

CONCLUSION TEXT: คำแนะนำหรือข้อสรุปสำหรับแจ้งให้ผู้ใช้ทราบในระหว่างทำการปรึกษา

เลขที่ประจำกฎนั้นเป็นเลขจำนวนเต็มตั้งแต่ 1 ขึ้นไปและไม่เกินค่ามากที่สุดของเลขจำนวนเต็มที่คอมพิวเตอร์กำหนดไว้ ทั้งนี้เพื่อใช้เป็นคีย์ฟิลด์ (key field) ในการเข้าถึง (access) กฎข้อต่างๆ ได้รวดเร็ว

ส่วนเงื่อนไขและส่วนสรุปสามารถจะมีอนุประโยคจำนวนเท่าไรก็ได้ แต่ต้องไม่เกินค่ามากที่สุดของเลขจำนวนเต็มเช่นกัน และต้องไม่เป็นอนุประโยคที่เป็นนิเสธ (negation) ถ้ามีเกิน 1 อนุประโยคจะต้องเชื่อมต่อกันด้วยตัวกระทำทางตรรก "และ" (logical AND) เท่านั้น ดังนั้นในกรณีที่ต้องการจะใช้ตัวกระทำทางตรรก "หรือ" (logical OR) ในส่วนเงื่อนไข ก็ให้แยกกฎข้อนั้นออกเป็นกฎข้อย่อยๆ ดังนี้

IF: 1. (A-V)₁ OR
2. (A-V)₂

THEN: 1. (A-V)_x

แยกเป็น

IF: 1. (A-V)₁

THEN: 1. (A-V)_x

และ

IF: 1. (A-V)₂

THEN: 1. (A-V)_x

ในส่วนสรุปนอกจากจะประกอบด้วยอนุประโยคแบบ A-V ซึ่งเป็นอนุประโยคธรรมดาแล้ว ยังสามารถจะมีอนุประโยคแบบพิเศษที่สามารถเรียกใช้คำสั่งชนิดภายนอกของดอสได้ ในที่นี้จะขอเรียกอนุประโยคแบบนี้ว่าอนุประโยคคำสั่ง (command clause) เพื่อเป็นช่องทางสำหรับให้ระบบสามารถติดต่อกับภายนอกได้ด้วยการใช้คำสั่งชนิดภายนอกของดอส (*.com หรือ *.exe) โดยสามารถจะมีอาร์กิวเมนต์ได้ไม่เกิน 1 ตัว คำสั่งนี้จะปฏิบัติงานเมื่อกฎข้อที่มีอนุประโยคคำสั่งถูกทดสอบแล้วได้ผลเป็นจริงและมีไฟล์คำสั่งที่ระบุอยู่ในไดเรกทอรีปัจจุบัน (current directory) เท่านั้น ดังนั้นรูปแบบของอนุประโยคคำสั่งจึงเป็น "#EXTERNAL command [arg]" แทนที่จะเป็น A-V แบบธรรมดา

สำหรับคำแนะนำหรือข้อสรุปนั้นสามารถจะมีความยาวได้ไม่เกิน 2048 ตัวอักษรและจะต้องประกอบด้วยอักขระที่พิมพ์ได้ (printable) เท่านั้น

2. รูปแบบของ attribute

ในระหว่างการปรึกษา ระบบจะต้องตั้งคำถามเพื่อสอบถามข้อเท็จจริงซึ่งเป็นผลการทดสอบจากผู้ใช้ คำถามที่ระบบใช้ถามนั้นจะเป็นคำถามประจำ attribute ซึ่งผู้ใช้ความรู้เป็นผู้กำหนดขึ้นและใส่ไว้ในฐานความรู้นั้นด้วย ดังนั้น attribute นอกจากจะประกอบด้วย value ที่เป็นไปได้ทั้งหมดแล้ว ยังมีคำถามประจำตัวอีกต่างหาก รูปแบบของ attribute จึงเป็นดังนี้

ATTRIBUTE NAME: attribute-1

VALUES: 1. value-1
 2. value-2

.

.

QUESTION: คำถามที่ใช้ถามเมื่อระบบต้องการทราบค่าของ attribute-1 จากผู้ใช้

เพื่อให้การเข้าถึง attribute และ value ทำได้อย่างรวดเร็วจึงใช้ชื่อของมันเป็นคีย์ฟิลด์ของระเบียบขึ้น ดังนั้นชื่อของ attribute และ value จึงถูกกำหนดให้มีความยาวไม่เกิน 28 ตัวอักษร และกำหนดให้ attribute 1 ตัวสามารถจะมี value ได้ไม่เกิน 100 ตัว ส่วนคำถามนั้นจะมีความยาวได้ไม่เกิน 2048 ตัวอักษร แต่ไม่ควรยาวเกิน 1 บรรทัดเพื่อความสวยงาม และจะต้องเป็นตัวอักษรที่พิมพ์ได้เท่านั้น

3. การออกแบบหน้าที่ (function) ของโครงระบบผู้เชี่ยวชาญ

ผู้วิจัยได้แบ่งหน้าที่ของโครงระบบผู้เชี่ยวชาญออกเป็น 3 ส่วนคือ ส่วนเตรียมพร้อมที่จะใช้หรือเลิกใช้ฐานความรู้, ส่วนจัดการกับความรู้อื่นๆ และ ส่วนให้คำปรึกษา

3.1 การออกแบบหน้าที่ของส่วนเตรียมพร้อมที่จะใช้หรือเลิกใช้ฐานความรู้

การเตรียมพร้อมที่จะใช้ฐานความรู้เป็นช่วงเริ่มต้นที่เรียกใช้โปรแกรม เพื่อรอให้ผู้ใช้ระบุชื่อของฐานความรู้ที่ต้องการใช้ จึงเป็นระยะก่อนที่ผู้ใช้จะเลือกใช้ส่วนจัดการกับความรู้อื่นๆหรือส่วนให้คำปรึกษา เมื่อผู้ใช้ได้ระบุชื่อฐานความรู้แล้ว ระบบจะต้องทำการเปิดไฟล์ต่างๆของฐานความรู้ และทำ

การสำรองไฟล์ทั้งหมดไว้ถ้าผู้ใช้ต้องการ แล้วให้ผู้ใช้เลือกใช้งานระบบต่อไป ส่วนการเลิกใช้งานฐานความรู้ นั้น หลังจากที่ใช้ได้ จัดการกับความรู้หรือปรึกษาเสร็จแล้วและต้องการเลิกใช้ฐานความรู้ นั้น ระบบ ก็จะต้องทำการปิดไฟล์ต่างๆให้เรียบร้อย แล้วเตรียมพร้อมที่จะทำงานในรอบ ใหม่ต่อไป

3.2 การออกแบบหน้าที่ของส่วนจัดการกับความรู้

หน้าที่ในการจัดการกับความรู้แบ่งออกเป็น 4 หน้าที่ด้วยกันคือ หน้าที่ในการแสดงความรู้(list knowledge), หน้าที่ในการรับความรู้ใหม่เข้าสู่ฐานความรู้(add knowledge), หน้าที่ในการลบความรู้บางส่วนออกจากฐานความรู้(delete knowledge), และหน้าที่ในการแก้ไขความรู้ในฐานความรู้(update knowledge) ซึ่งแต่ละหน้าที่ควรจะมีความสามารถดังนี้

3.2.1 การแสดงความรู้ที่มีอยู่ในฐานความรู้ จะต้องสามารถทำหน้าที่ดังต่อไปนี้

3.2.1.1 แสดงเนื้อหาของกฎ(list rule) ก่อนที่จะระบุเลขที่ของกฎที่ต้องการให้แสดง ผู้ใช้สามารถเลือกให้แสดงออกทาง เครื่องพิมพ์พร้อมกับหน้าจอได้ การแสดงกฎสามารถที่จะเลือกกฎเฉพาะข้อที่ละ ข้อ หรือให้แสดงกฎทุกข้อที่มีอยู่ในฐานความรู้เรียงตามลำดับเลขที่ของกฎก็ได้ ในกรณีที่ผู้ใช้เลือกให้แสดงกฎทุกข้อก็สามารถสั่งให้ระบบหยุดรอหลังจากที่แสดงกฎ แต่ละข้อไปแล้ว และในระหว่างหยุดรอก็สามารถที่จะยุติการแสดงกฎได้ด้วย

3.2.1.2 แสดง attribute จะต้องทำได้ ในลักษณะเดียวกันกับการแสดงกฎคือ แสดงทีละตัวหรือทุกตัวก็ได้ นอกจากนี้จะต้องมีการช่วยเหลือให้แก่ผู้ใช้ในขณะที่รอให้ผู้ใช้ระบุชื่อของ attribute โดยการแสดงรายชื่อ attribute ที่มีอยู่ให้ผู้ใช้เลือกได้

3.2.1.3 แสดงอนุประโยคที่มีใช้อยู่ทั้งหมดใน ฐานความรู้ เป็นการนำเอาอนุประโยคทั้งหมดที่ถูกใช้โดยกฎต่างๆมาแสดง และจะแสดงทีละหนึ่งหน้าจอแล้วหยุดรอ เพื่อให้ผู้ใช้สามารถเลือกที่จะให้แสดง ต่อหรือเลิกแสดงได้

3.2.1.4 แสดง value ทั้งหมดที่มีอยู่ในฐาน ความรู้เรียงตามลำดับตัวอักษร ทำในลักษณะเดียวกับการแสดงอนุประโยค

3.2.2 การรับความรู้ใหม่เข้าสู่ฐานความรู้ จะต้องสามารถทำหน้าที่ดังต่อไปนี้

3.2.2.1 รับ attribute ใหม่เข้าสู่ฐานความรู้ ก่อนนำเข้าจะต้องตรวจสอบให้แน่ใจก่อนว่าไม่ซ้ำกับ attribute เดิมที่มีอยู่แล้ว การรับ attribute ใหม่ให้รับ value และคำถามของมันด้วย

3.2.2.2 รับกฎใหม่เข้าสู่ฐานความรู้ กฎที่จะนำเข้าต้องมีเลขที่กฎไม่ซ้ำกับเลขที่ที่มีอยู่แล้ว การรับกฎใหม่ให้รับทั้งส่วนเงื่อนไข, ส่วนสรุป, และคำแนะนำหรือข้อสรุปด้วย โดยที่กฎ 1 ข้ออาจจะมีไม่ครบทั้ง 3 ส่วนก็ได้ ส่วนเงื่อนไขและส่วนสรุปให้ประกอบด้วยอนุประโยคซึ่งแต่ละอนุประโยคผู้ใช้จะเป็นผู้เลือก attribute ก่อน โดย attribute ที่เลือกจะต้องมีอยู่ในฐานความรู้ก่อนแล้ว จากนั้นระบบจะแสดง value ของ attribute นั้นให้ผู้ใช้เลือกเพื่อเข้าคู่เป็นอนุประโยค A-V

สำหรับการนำเข้าอนุประโยคคำสั่งให้ใช้คำสั่งวน(reserved word) คำว่า "#EXTERNAL" ใส่แทนชื่อของ attribute ในการสร้างอนุประโยค จากนั้นระบบจะให้ใส่ชื่อไฟล์คำสั่งชนิดภายนอกของดอส พร้อมทั้งอาร์กิวเมนต์ของมัน(ถ้ามี) โดยให้เว้นวรรคแยกระหว่างคำสั่งกับอาร์กิวเมนต์

3.2.3 การลบความรู้บางส่วนออกจากฐานความรู้ จะต้องสามารถทำหน้าที่ดังต่อไปนี้

3.2.3.1 ลบ attribute ออกจากฐานความรู้ที่ละตัว โดยผู้ใช้เป็นผู้ระบุชื่อ attribute ที่ต้องการจะลบทิ้ง ในกรณีที่ผู้ใช้ต้องการความช่วยเหลือ ระบบต้องสามารถแสดงรายชื่อ attribute ที่มีอยู่เรียงตามลำดับให้ผู้ใช้เลือก หาก attribute นั้นยังคงถูกใช้โดยกฎบางข้อ ระบบจะไม่ทำการลบ

3.2.3.2 ลบกฎออกจากฐานความรู้ที่ละข้อ โดยผู้ใช้เป็นผู้ระบุเลขที่กฎที่ต้องการจะลบทิ้ง 2 อย่างนี้ก่อนทำการลบให้แสดงเนื้อหาของมันเสียก่อนแล้วจึงถามซ้ำอีกครั้งก่อนที่จะลบจริง

3.2.4 การแก้ไขความรู้ในฐานความรู้ ต้องสามารถทำหน้าที่ดังต่อไปนี้

3.2.4.1 แก้ไข attribute ผู้ใช้เป็นผู้ระบุชื่อ attribute ที่ต้องการจะแก้ไข ในกรณีที่ผู้ใช้ต้องการความช่วยเหลือ ในระหว่างการระบุชื่อ attribute ที่ต้องการ ระบบจะต้องสามารถแสดงรายชื่อที่มีอยู่ให้ผู้ใช้เลือกได้ แล้วแสดงเนื้อหาของ attribute นั้น การแก้ไขทำ

ได้ 3 ส่วนคือ แก๊ไขชื่อ, แก๊ไขส่วน value, และแก๊ไขส่วนคำถาม การแก๊ไขส่วน value นั้นทำได้ 2 แบบคือเพิ่มหรือลบ value การเพิ่ม value เข้าไปผู้ใช้สามารถเลือกลำดับที่ของ value ใหม่ที่จะเพิ่มเข้าไปว่าจะให้เป็นตัวที่เท่าไรของ attribute นี้ และในกรณีที่ผู้ใช้ต้องการความเหลือในขณะใส่ชื่อของ value ใหม่ ระบบต้องแสดงรายชื่อที่มีอยู่เรียงตามลำดับให้ผู้ใช้เลือก และ value ที่จะเพิ่มเข้าไปต้องไม่ซ้ำกับของเดิมที่ attribute นั้นมีอยู่แล้ว สำหรับการลบ value ผู้ใช้ต้องระบุชื่อโดยตรงเท่านั้น

3.2.4.2 แก๊ไขกฎ ผู้ใช้เป็นผู้ระบุเลขที่กฎที่ต้องการจะแก๊ไข การแก๊ไขทำได้ 3 ส่วนคือ ส่วนเงื่อนไข, ส่วนสรุป, และคำแนะนำหรือข้อสรุป การแก๊ไขส่วนเงื่อนไขและส่วนสรุปมีลักษณะที่เหมือนกันคือเพิ่มหรือลบอนุประโยคทีละ 1 อนุประโยค การเพิ่มอนุประโยคผู้ใช้ต้องระบุตำแหน่งของอนุประโยคที่จะเพิ่มเข้าไป แล้วระบุชื่อ attribute ที่ต้องการซึ่งต้องมีอยู่ในฐานความรู้ จากนั้นก็เลือก value ที่ต้องการ ส่วนการลบอนุประโยคผู้ใช้ต้องระบุตำแหน่งของอนุประโยคที่ต้องการจะลบทิ้ง

3.3 การออกแบบหน้าที่ของส่วนให้คำปรึกษา

ส่วนให้คำปรึกษาจะต้องมีความสามารถในการทำหน้าที่ต่อไปนี้

3.3.1 แสดงข้อสรุป หรือคำแนะนำให้แก่ผู้ใช้เป็นครั้งคราว โดยการใช้ความรู้ที่มีอยู่ในฐานความรู้มาอนุมานร่วมกับข้อเท็จจริงที่เกิดขึ้นในระหว่างปรึกษา ซึ่งข้อสรุปหรือนำแนะนำนี้ก็คือส่วนหนึ่งของกฎนั่นเอง และจะแสดงให้เห็นต่อเมื่อการทดสอบกฎข้อนั้นได้ผลเป็นจริงเท่านั้น

3.3.2 ตั้งคำถามและรับคำตอบจากผู้ใช้ การตั้งคำถามจะเกิดขึ้นเมื่อระบบต้องการทราบข้อเท็จจริงบางอย่าง นั่นคือระบบต้องการทราบค่าของ attribute ที่ระบบกำลังสนใจ เพื่อนำไปใช้ในการทดสอบกฎ ดังนั้นระบบจะต้องเอาคำถามประจำ attribute มาแสดงพร้อมกับตัวเลือกซึ่งเป็น value ทั้งหมดของ attribute นั้น แล้วให้ผู้ใช้เลือกตอบโดยใส่ตัวเลขที่กำกับหน้า value ที่ต้องการเป็นคำตอบ ระบบจะนำ value นั้นมาเก็บไว้เป็นข้อเท็จจริง เพื่อทำการทดสอบต่อไปว่าตรงกับกฎที่กำลังทดสอบอยู่หรือไม่

3.3.3 ต้องสามารถรับคำสั่งพิเศษในระหว่างการถามตอบ แทนที่ผู้ใช้จะตอบคำถามของระบบ อาจจะมีป้อนคำสั่งพิเศษให้กับระบบแทนคำตอบ เพื่อให้ระบบทำงานบางอย่างก่อนที่ผู้ใช้จะตอบคำถามที่ระบบถามมา

คำสั่งที่จำเป็นสำหรับโครงสร้างของผู้เชี่ยวชาญที่จะสร้างขึ้นมี 4 คำสั่งดังนี้

3.3.3.1 คำสั่งให้แสดงเหตุผลของการตั้งคำถาม(why) เมื่อระบบได้รับคำสั่งนี้จะต้องแสดงเนื้อหาของกฎที่กำลังทดสอบอยู่และแสดงเลขที่ของกฎที่เกี่ยวข้องกับการตั้งคำถามนั้น จากนั้นรอจนกว่าผู้ใช้หรือคนที่มาตอบคำถาม

3.3.3.2 คำสั่งให้ทำการบันทึก(save) ข้อเท็จจริงที่ได้มาตั้งแต่เริ่มปรึกษาเก็บไว้ในไฟล์ เมื่อได้รับคำสั่งนี้จะต้องรอให้ผู้ใช้ระบุชื่อไฟล์ที่จะทำการบันทึก เมื่อผู้ใช้ระบุชื่อไฟล์แล้วก็ทำการบันทึกแล้วจบการปรึกษา

3.3.3.3 คำสั่งให้ทำการดึงข้อมูล(load)ที่บันทึกไว้กลับมาใช้ในการปรึกษา เมื่อได้รับคำสั่งนี้จะต้องรอให้ผู้ใช้ระบุชื่อของไฟล์ที่จะทำการอ่าน เมื่อผู้ใช้ระบุชื่อไฟล์แล้วจึงทำการอ่านเพื่อนำเอาข้อเท็จจริงต่างๆจากไฟล์เข้าสู่ระบบแล้วนำไปใช้งาน โดยจะยกเลิกข้อเท็จจริงเดิมที่มีอยู่ในระบบ แล้วทำการปรึกษาต่อ

3.3.3.4 คำสั่งให้ยุติการปรึกษา(quit) เป็นคำสั่งที่สั่งให้ระบบหยุดและจบการปรึกษากันที่

นอกจากคำสั่งต่างๆดังกล่าวมาแล้ว ยังมีคำสั่งพิเศษเพิ่มเติมอีก 6 คำสั่งด้วยกัน ซึ่งจะอำนวยความสะดวกแก่ผู้ใช้ในระหว่างการปรึกษา ตลอดจนอาจใช้เพื่อการตรวจสอบการทำงานของระบบด้วยก็ได้ คำสั่งพิเศษเพิ่มเติมมีดังต่อไปนี้

3.3.3.5 คำสั่งขอความช่วยเหลือ(help) สำหรับใช้ในกรณีที่ตัวเลือกของคำถามมีจำนวนมากเกิน 1 หน้าจอจนทำให้ผู้ใช้ไม่สามารถเห็นตัวเลือกทั้งหมดได้พร้อมกัน เมื่อระบบได้รับคำสั่งนี้จะต้องแสดงตัวเลือกในรูปแบบของเมนูที่มีแถบสว่างเลื่อนได้(hi-light bar)ให้ผู้ใช้เลือกโดยสามารถเปลี่ยนหน้าเมนูได้

3.3.3.6 คำสั่งให้ทำการแสดงเนื้อหาของกฎ(list rule) เมื่อได้รับคำสั่งนี้จะต้องรอให้ผู้ใช้ระบุเลขที่ของกฎที่ต้องการจะให้เห็น เมื่อผู้ใช้ระบุเลขที่กฎแล้วก็ให้ทำการแสดงเนื้อหาของกฎข้อนั้น

3.3.3.7 คำสั่งให้แสดงข้อเท็จจริงที่มีอยู่ในระบบ(summarize) เมื่อได้รับคำสั่งนี้จะต้องนำเอาข้อเท็จจริงต่างๆในระบบมาแสดงในรูปแบบของอนุประโยค A-V แล้วทำการปรึกษาต่อ

3.3.3.8 คำสั่งให้ย้อนกลับ 1 คำถาม(step back) เมื่อได้รับคำสั่งนี้จะต้องย้อนกลับไปถามคำถามที่ผ่านมา เพื่อเปิดโอกาสให้ผู้ใช้งานสามารถแก้ไขคำตอบที่ตอบไปแล้วได้

3.3.3.9 คำสั่งให้แสดงผลการทดสอบกฎ (trace on) เมื่อได้รับคำสั่งนี้ระบบจะต้องแสดงเลขที่ของกฎที่กำลังถูกทดสอบ รวมทั้งผลการทดสอบของกฎข้อนั้นด้วย

3.3.3.10 คำสั่งยกเลิกการแสดงผลการทดสอบกฎ(trace off) เมื่อได้รับคำสั่งนี้ระบบจะเลิกแสดงผลการทดสอบกฎ นอกเหนือจากคำสั่งพิเศษซึ่งเป็นคำสั่งที่เรียกใช้ภายในระบบแล้ว ผู้ใช้สามารถเรียกใช้คำสั่งภายนอกระบบคือ คำสั่งชนิดภายนอกของดอสที่สามารถมีอาร์กิวเมนต์ได้ 1 ตัว ซึ่งนับได้ว่าเป็นการติดต่อระหว่างระบบ(system interface) คำสั่งภายนอกจะถูกเรียกใช้ต่อเมื่อกฎข้อนั้นได้ถูกทดสอบและผลการทดสอบเป็นจริง ซึ่งจะมีผลทำให้การปรึกษาถูกพักไว้ชั่วคราว แล้วระบบจะโอนการควบคุมการทำงานของโปรแกรมให้กับคำสั่งภายนอกที่ถูกเรียกใช้นั้น และเมื่อสิ้นสุดการทำงานของคำสั่งภายนอกก็จะกลับมาทำการปรึกษาต่อไป ทั้งนี้การใช้งานคำสั่งภายนอกขึ้นอยู่กับความจุของหน่วยความจำหลักที่เหลืออยู่ ดังนั้นหากหน่วยความจำหลักไม่เพียงพอก็ไม่สามารถใช้งานได้ และอาจทำให้การทำงานของระบบจบลงแบบผิดปกติ(abnormal terminated) อีกด้วย

4. การออกแบบโครงสร้างข้อมูลระบบ

โครงสร้างข้อมูลของระบบแบ่งออกเป็น 2 ส่วนคือ โครงสร้างข้อมูลของฐานความรู้(data structure of knowledge base) และ โครงสร้างข้อมูลหน่วยความจำใช้งานของส่วนให้คำปรึกษา(data structure of working memory of consultant)

เพื่อความเข้าใจกับโครงสร้างข้อมูลที่จะกล่าวถึงต่อไปนี้ ผู้วิจัยได้กำหนดชนิดของข้อมูล(data type) ขึ้น 5 แบบด้วยกันคือ BYTE, NUMBER, ADDR, POINTER และ STRING ซึ่งจากนี้ไปการกล่าวถึงชนิดของข้อมูลจะมีเพียง 5 แบบนี้เท่านั้นเพื่อให้เป็นที่เข้าใจตรงกัน รายละเอียดของชนิดของข้อมูลทั้ง 5 แบบมีดังต่อไปนี้

ชนิดของข้อมูลแบบ BYTE มีขนาด 1 ไบท์ ใช้เก็บเลขจำนวนเต็ม(-255 ถึง 256) หรือตัวอักษร 1 ตัว

ชนิดของข้อมูลแบบ NUMBER มีขนาด 2 ไบท์ ใช้เก็บเลขจำนวนเต็มบวก(1 ถึง 32767)

ชนิดของข้อมูลแบบ ADDR มีขนาด 2 ไบท์ ใช้เก็บเลขจำนวนเต็มมีค่าเริ่มต้นจาก 0 (0 ถึง 32767)

ชนิดของข้อมูลแบบ POINTER มีขนาดขึ้นอยู่กับโมเดลของคอมไพเลอร์(compiler model) ใช้เก็บที่อยู่(address)ของหน่วยความจำหลัก

ชนิดของข้อมูลแบบ STRING[n] มีขนาด n ไบท์ ใช้เก็บตัวอักษร

4.1 โครงสร้างข้อมูลของฐานความรู้

ฐานความรู้ของระบบจะต้องประกอบไปด้วย value, attribute, อนุประโยค, กฎ และส่วนที่เป็นข้อความ(ทั้งคำถามและข้อสรุป) และเนื่องจากฐานความรู้ของระบบจะเป็นข้อมูลแบบสถิตย์(static data)คือ ในขณะที่ทำการปรึกษาจะไม่มี การเปลี่ยนแปลงเกิดขึ้นในฐานความรู้ จะมีเพียงการนำเอามาใช้เท่านั้น ดังนั้นเพื่อให้ง่ายและสะดวกในการพัฒนาฐานความรู้จึงออกแบบให้ฐานความรู้และโครงสร้างข้อมูลของมันถูกเก็บไว้ในดิสค์อยู่ในไดเรกตอรีย่อย(sub-directory) และกำหนดให้ชื่อของฐานความรู้เป็นชื่อของไดเรกตอรีย่อยที่เก็บความรู้ นั้น ซึ่งประกอบไปด้วย 6 ไฟล์ข้อมูล(data file) คือ ATTRIBUTE, VALUE, CLAUSE, RULE, NUMBER, และ TEXT โดยแต่ละไฟล์เป็นไฟล์ที่มีขนาดของระเบียนคงที่(fixed-length record) ถูกเก็บในลักษณะของไฟล์แบบรีเลทีฟ(relative record file)ซึ่งสามารถอ้างอิงถึงระเบียนได้ด้วยเลขที่ของมันโดยเริ่มต้นจากหมายเลข 1 ดังนั้นจึงเป็นเหตุหนึ่งที่ทำให้ระเบียนในไฟล์มีจำนวนจำกัด สำหรับการวิจัยนี้จะใช้เลขจำนวนเต็มบวก(positive integer)ในการระบุหมายเลขของระเบียน นั่นคือแต่ละไฟล์ข้อมูลจะมีจำนวนระเบียนได้ไม่เกิน 32767 ระเบียน หมายเลขของระเบียน ATTRIBUTE, VALUE, และ CLAUSE ยังถูกใช้เป็นเลขที่ประจำตัวของ attribute, value, และ clause นั้นๆด้วย ตัวอย่างเช่น attribute ที่ถูกเก็บอยู่ในระเบียนหมายเลข 1 เลขที่ของ attribute นั้นจะเป็นเลขที่ 1, value ที่ถูกเก็บอยู่ในระเบียนหมายเลข 5 เลขที่ของ value นั้นจะเป็นเลขที่ 5, อนุประโยคที่เก็บอยู่ในระเบียนหมายเลข 9 เลขที่ของอนุประโยคนั้นจะเป็นเลขที่ 9 เป็นต้น

ATTRIBUTE

text *	val_list	clause_list	question_list
--------	----------	-------------	---------------

VALUE

text *	attr_list
--------	-----------

NORMAL CLAUSE

type = 0	attr_num	val_num	rule_list	chain_list
----------	----------	---------	-----------	------------

COMMAND CLAUSE

type = 1	command	arg	rule_list	unused
----------	---------	-----	-----------	--------

RULE

num *	if_list	then_list	conclude_list
-------	---------	-----------	---------------

NUMBER

num	next
-----	------

TEXT

text	next
------	------

* = KEY FIELD

รูปที่ 3.1 โครงสร้างของไฟล์ข้อมูลของฐานความรู้

นอกจากไฟล์ข้อมูลแล้ว ฐานความรู้ยังประกอบด้วยไฟล์ดัชนี(index file)อีก 3 ไฟล์คือ ไฟล์ดัชนีของ ATTRIBUTE, VALUE และ RULE เพื่อให้สามารถเข้าถึงระเบียนของไฟล์ ATTRIBUTE และ VALUE ได้ 2 วิธีคือโดยใช้ชื่อหรือหมายเลขประจำระเบียน สำหรับระเบียนของกฎให้ใช้หมายเลขกฎเป็นดัชนีในการเข้าถึงตัวระเบียนเพียงอย่างเดียว

รูปที่ 3.1 แสดงโครงสร้างของไฟล์ข้อมูลของฐานความรู้ แต่ละไฟล์มีรายละเอียดดังต่อไปนี้

4.1.1 TEXT เป็นไฟล์ที่ใช้เก็บข้อความต่างๆ ทั้งข้อความคำถามของ attribute และข้อความคำแนะนำหรือข้อสรุปของกฎต่างๆ การเข้าถึงระเบียนของไฟล์นี้ทำได้วิธีเดียวคือใช้หมายเลขประจำระเบียนเท่านั้น แต่ละระเบียนมีขนาด 30 ไบท์ ใน 1 ระเบียนมี 2 ฟิลด์คือ

4.1.1.1 text ชนิดของข้อมูลเป็นแบบ STRING[28] คือฟิลด์ที่ใช้เก็บส่วนของข้อความ 1 ส่วน

4.1.1.2 next ชนิดของข้อมูลเป็นแบบ ADDR คือฟิลด์ที่ใช้เก็บหมายเลขระเบียนของข้อความส่วนถัดไป ในกรณีที่ไม่มีข้อความถัดไปให้มีค่าเป็น 0 (NULL)

การเก็บข้อความเป็นลักษณะลิงค์ลิสต์โดยจะตัดแบ่งข้อความออกเป็นส่วนๆในกรณีที่มีความยาวเกิน 28 ไบท์ ข้อความแต่ละส่วนยาว 28 ไบท์จะเก็บเป็น 1 ระเบียน โดยมีฟิลด์ next เป็นตัวบอกหมายเลขของระเบียนข้อความส่วนถัดไป

4.1.2 NUMBER เป็นไฟล์ที่ใช้เก็บลิงค์ลิสต์ของเลขที่ของ attribute, value, อนุประโยค, และกฎ การเข้าถึงระเบียนของไฟล์นี้ทำได้วิธีเดียวคือใช้หมายเลขประจำระเบียนเท่านั้น แต่ละระเบียนมีขนาด 4 ไบท์ ใน 1 ระเบียนมี 2 ฟิลด์คือ

4.1.2.1 num ชนิดของข้อมูลเป็นแบบ NUMBER เป็นฟิลด์ที่ใช้เก็บเลขที่ของ attribute, value, อนุประโยค, หรือกฎ

4.1.2.2 next ชนิดของข้อมูลเป็นแบบ ADDR เป็นฟิลด์ที่ใช้เก็บหมายเลขระเบียนของเลขที่ถัดไปในลิงค์ลิสต์ หากเป็นระเบียนสุดท้ายของลิงค์ลิสต์ให้มีค่าเป็น 0 (NULL)

4.1.3 ATTRIBUTE เป็นไฟล์ที่ใช้เก็บระเบียนของ attribute แต่ละระเบียนมีขนาด 34 ไบต์ การเข้าถึงระเบียนของไฟล์นี้ทำได้ 2 วิธีคือ ใช้เลขที่ของ attribute ซึ่งจะมีค่าเท่ากับหมายเลขประจำระเบียน หรือใช้ชื่อของ attribute ไปทำการค้นหาหมายเลขประจำระเบียนจากไฟล์ดัชนี แล้วใช้หมายเลขประจำระเบียนนั้น ใน 1 ระเบียนประกอบด้วยข้อมูล 4 필ด์คือ

4.1.3.1 text ชนิดของข้อมูลเป็นแบบ STRING[28] ใช้เก็บชื่อของ attribute เนื่องจากฟิลด์นี้เปรียบเสมือนคีย์ฟิลด์ของระเบียน ดังนั้นข้อมูลในฟิลด์นี้จะต้องไม่ซ้ำกับของระเบียนอื่นในไฟล์เดียวกันนี้

4.1.3.2 val_list ชนิดของข้อมูลเป็นแบบ ADDR ใช้เก็บหมายเลขระเบียนของเลขที่ของ value ตัวแรกของ attribute นี้ เลขที่ของ value ทั้งหมดของ attribute จะถูกเก็บเป็นลิงค์ลิสต์ในไฟล์ NUMBER

4.1.3.3 clause_list ชนิดของข้อมูลเป็นแบบ ADDR ใช้เก็บหมายเลขระเบียนของเลขที่ของอนุประโยคแรกที่ใช้ attribute นี้อยู่ เลขที่ของอนุประโยคทั้งหมดที่ใช้ attribute นี้จะถูกเก็บเป็นลิงค์ลิสต์ในไฟล์ NUMBER

4.1.3.4 question_list ชนิดของข้อมูลเป็นแบบ ADDR ใช้เก็บหมายเลขระเบียนของข้อความคำถามส่วนแรก คำถามจะถูกตัดแบ่งเป็นส่วนๆ เก็บเป็นลิงค์ลิสต์ในไฟล์ TEXT

4.1.4 VALUE เป็นไฟล์ที่ใช้เก็บระเบียนของ value แต่ละระเบียนมีขนาด 30 ไบต์ การเข้าถึงระเบียนของไฟล์นี้ทำได้ 2 วิธี ในลักษณะเดียวกับไฟล์ ATTRIBUTE ใน 1 ระเบียนประกอบด้วยข้อมูล 2 필ด์คือ

4.1.4.1 text ชนิดของข้อมูลเป็นแบบ STRING[28] ใช้เก็บชื่อของ value และเนื่องจากฟิลด์นี้เปรียบเสมือนคีย์ฟิลด์ของระเบียน ดังนั้นข้อมูลในฟิลด์นี้จะต้องไม่ซ้ำกับของระเบียนอื่นในไฟล์เดียวกันนี้

4.1.4.2 attr_list ชนิดของข้อมูลเป็นแบบ ADDR ใช้เก็บหมายเลขระเบียนของเลขที่ของ attribute ที่ใช้ value นี้อยู่ เลขที่ของ attribute ทั้งหมดที่ใช้ value นี้จะถูกเก็บเป็นลิงค์ลิสต์ในไฟล์ NUMBER

4.1.5 CLAUSE เป็นไฟล์ที่ใช้เก็บระเบียนของ
 อนุประโยค แต่ละระเบียนมีขนาด 9 ไบต์ การเข้าถึงระเบียนของไฟล์นี้ทำได้
 วิธีเดียวคือใช้เลขที่ของอนุประโยคซึ่งมีค่าเท่ากับหมายเลขประจำระเบียนของ
 อนุประโยค ระเบียนของอนุประโยคแบ่งเป็น 2 ประเภทคืออนุประโยคธรรมดา
 (normal clause) และอนุประโยคคำสั่ง (command clause) โดยเก็บ
 ระเบียนทั้ง 2 ประเภทนี้ไว้ในไฟล์เดียวกัน

4.1.5.1 อนุประโยคธรรมดาคืออนุประโยคที่
 เกิดจากการจับคู่ของ attribute และ value ใน 1 ระเบียนประกอบด้วย
 ข้อมูล 5 필ด์คือ

4.1.5.1.1 type ชนิดของ
 ข้อมูลเป็นแบบ BYTE มีค่าเป็น 0 บอกให้รู้ว่าเป็นระเบียนของอนุประโยค
 ธรรมดา

4.1.5.1.2 attr_num ชนิด
 ของข้อมูลเป็นแบบ NUMBER ใช้เก็บเลขที่ของ attribute ที่อนุประโยคนี้อาศัยอยู่

4.1.5.1.3 val_num ชนิด
 ของข้อมูลเป็นแบบ NUMBER ใช้เก็บเลขที่ของ value ที่อนุประโยคนี้อาศัยอยู่

4.1.5.1.4 rule_list ชนิด
 ของข้อมูลเป็นแบบ ADDR ใช้เก็บหมายเลขระเบียนของเลขที่กฎที่ใช้อนุประโยค
 นี้อยู่ เลขที่ของกฎทั้งหมดที่ใช้อนุประโยคนี้อาจถูกเก็บเป็นลิงค์ลิสต์ในไฟล์ NUMBER

4.1.5.1.5 chain_list
 ชนิดของข้อมูลเป็นแบบ ADDR ใช้เก็บหมายเลขระเบียนของเลขที่ของกฎที่มี
 อนุประโยคนี้อาศัยอยู่ในส่วนเงื่อนไข เลขที่ของกฎทั้งหมดเหล่านี้จะถูกเก็บเป็น
 ลิงค์ลิสต์ในไฟล์ NUMBER

4.1.5.2 อนุประโยคคำสั่งเป็นอนุประโยคที่
 ปรากฏได้เฉพาะในส่วนสรุปของกฎ ใน 1 ระเบียนประกอบด้วยข้อมูล 5 필ด์คือ

4.1.5.2.1 type ชนิดของ
 ข้อมูลเป็นแบบ BYTE มีค่าเป็น 1 บอกให้รู้ว่าเป็นอนุประโยคคำสั่ง

4.1.5.2.2 command ชนิด
 ของข้อมูลเป็นแบบ ADDR ใช้เก็บหมายเลขระเบียนของคำสั่งซึ่งถูกเก็บไว้ใน
 ไฟล์ TEXT

4.1.5.2.3 arg ชนิดของ
 ข้อมูลเป็นแบบ ADDR ใช้เก็บหมายเลขระเบียนของอาร์กิวเมนต์ของคำสั่ง

และอาร์กิวเมนต์นี้ก็จะถูกเก็บไว้ในไฟล์ TEXT

4.1.5.2.4 rule_list เช่นเดียวกับฟิลด์ rule_list ของอนุประโยคธรรมดา

4.1.5.2.5 unused มีขนาด 2 ไบต์ เป็นฟิลด์ที่ไม่มีการใช้งาน เพื่อทำให้ความยาวของระเบียบอนุประโยคคำสั่งเท่ากับของอนุประโยคธรรมดาเท่านั้น

4.1.6 RULE เป็นไฟล์ที่ใช้เก็บระเบียบของกฎ แต่ละระเบียบมีขนาด 8 ไบต์ การเข้าถึงระเบียบของไฟล์นี้ให้ทำโดยใช้เลขที่ของกฎไปทำการค้นหาหมายเลขประจำระเบียบจากไฟล์ดัชนีแล้วใช้หมายเลขประจำระเบียบนั้น ที่เป็นเช่นนี้เพราะต้องการให้ผู้ใช้งานสามารถจะกำหนดเลขที่ของกฎได้เอง ในแต่ละระเบียบประกอบด้วยข้อมูล 4 ฟิลด์คือ

4.1.6.1 num ชนิดของข้อมูลเป็นแบบ NUMBER ใช้เก็บเลขที่ของกฎ

4.1.6.2 if_list ชนิดของข้อมูลเป็นแบบ ADDR ใช้เก็บหมายเลขระเบียบของเลขที่ของอนุประโยคแรกของส่วนเงื่อนไข เลขที่ของอนุประโยคทั้งหมดของส่วนเงื่อนไขจะถูกเก็บเป็นลิงค์ลิสต์ในไฟล์ NUMBER

4.1.6.3 then_list เช่นเดียวกับฟิลด์ if_list แต่เป็นของส่วนสรุป

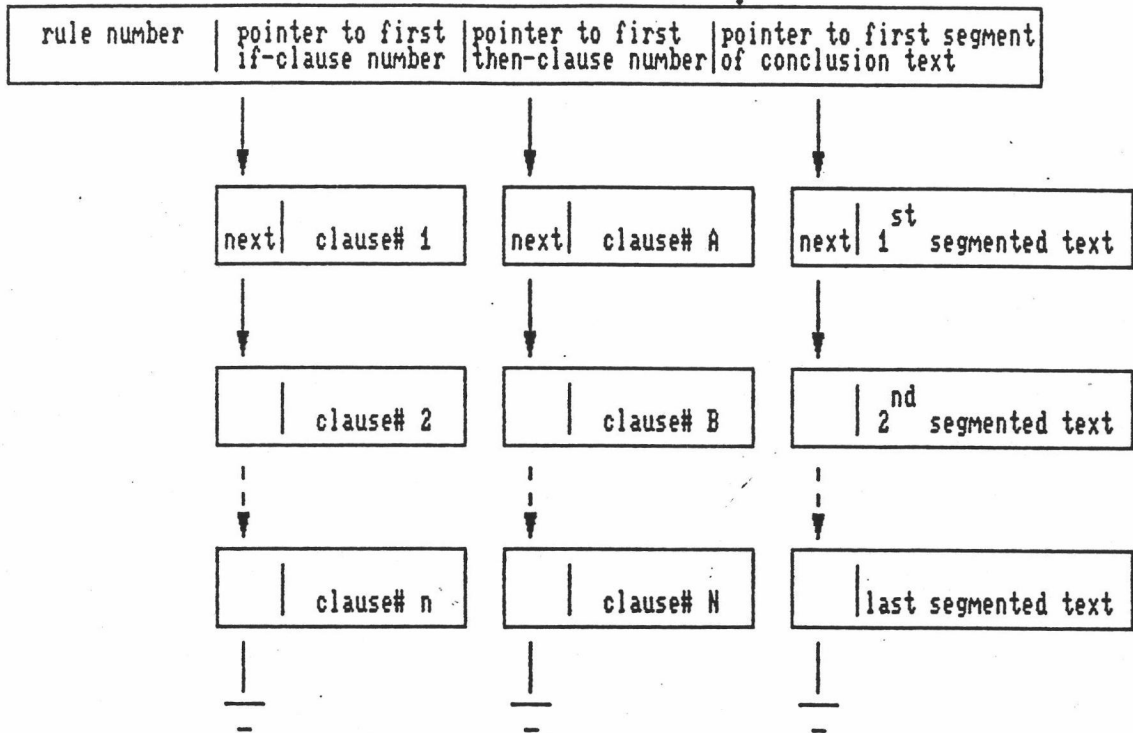
4.1.6.4 conclude_list ชนิดของข้อมูลเป็นแบบ ADDR ใช้เก็บหมายเลขระเบียบของข้อความคำแนะนำหรือข้อสรุปส่วนแรก ข้อความนี้จะถูกตัดแบ่งเป็นส่วนๆ เก็บเป็นลิงค์ลิสต์ในไฟล์ TEXT

4.2 โครงสร้างข้อมูลของกฎ (data structure of a rule)

รูปที่ 3.2 แสดงโครงสร้างข้อมูลของกฎ 1 ข้อซึ่งประกอบด้วย เลขที่ประจำกฎ, ส่วนเงื่อนไข, ส่วนสรุป, และข้อความที่เป็นคำแนะนำหรือข้อสรุป โดยระเบียบของกฎจะถูกเก็บไว้ในไฟล์ RULE

เนื่องจากทั้งส่วนเงื่อนไขและส่วนสรุปสามารถจะประกอบด้วยหลายอนุประโยค ดังนั้นโครงสร้างข้อมูลของสองส่วนนี้จึงเป็นลิงค์ลิสต์ (linked list) โดยจะเก็บเลขที่ของอนุประโยคและตัวบอกตำแหน่งของเลขที่อนุประโยคถัดไปไว้เป็น 1 ระเบียบของไฟล์ NUMBER เลขที่ของอนุประโยคนี้ระบบจะเป็นผู้กำหนดให้โดยจะมีค่าเท่ากับหมายเลขระเบียบของมัน และผู้ใช้จะไม่ทราบว่าอนุประโยคนั้นมีเลขที่เท่าไร ผิดกับเลขที่ของกฎซึ่งผู้ใช้จะ

เป็นผู้กำหนดขึ้นเองโดยจะต้องเป็นเลขจำนวนเต็มบวกและจะซ้ำกันไม่ได้

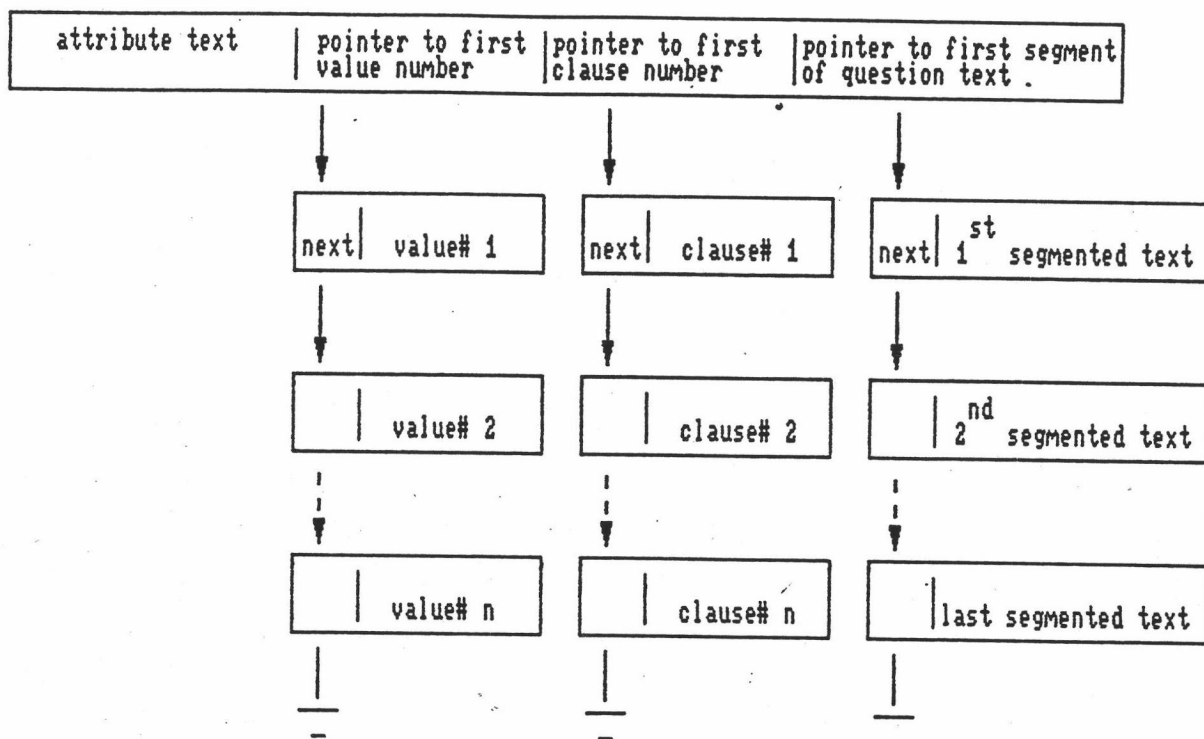


รูปที่ 3.2 โครงสร้างข้อมูลของกฎ 1 ข้อ

สำหรับข้อความนั้นมีความยาวได้ตั้งแต่ 0-2048 ตัวอักษร ถ้าหากจะเก็บข้อความเป็นระเบียบที่มีขนาด 2048 ตัวอักษรเท่ากันสำหรับกฎทุกข้อก็จะทำให้เสียเนื้อที่ดิสค์ในกรณีที่กฎบางข้อมีข้อความสั้นๆ เพื่อเป็นการประหยัดเนื้อที่จึงให้ตัดแบ่งข้อความออกเป็นส่วนๆ แต่ละส่วนมีขนาดเท่ากันและจะมีตัวบอกตำแหน่งของข้อความส่วนถัดไปรวมเป็น 1 ระเบียบ เก็บไว้ในไฟล์ TEXT

4.3 โครงสร้างข้อมูลของ attribute (data structure of an attribute)

รูปที่ 3.3 แสดงโครงสร้างข้อมูลของ attribute 1 ตัว ประกอบด้วย value ซึ่งมีได้หลายตัวและข้อความที่เป็นคำถาม โดยระเบียบของ attribute จะถูกเก็บไว้ในไฟล์ ATTRIBUTE ซึ่งระบบจะกำหนดเลขที่ของ attribute ให้ในลักษณะเดียวกับเลขที่ของอนุประโยค



รูปที่ 3.3 โครงสร้างข้อมูลของ attribute 1 ตัว

attribute แต่ละตัวอาจจะมีย่อประกอบที่ใช้มันอยู่ และเพื่อให้สามารถอ้างอิงถึงย่อประกอบที่ใช้ attribute นี้ จึงเก็บเลขที่ของย่อประกอบเหล่านั้นไว้เป็นลิงค์ลิสต์ในลักษณะเดียวกันกับส่วนเงื่อนไขและส่วนสรุปของกฎ โดยเก็บไว้ในไฟล์ NUMBER เหมือนกัน

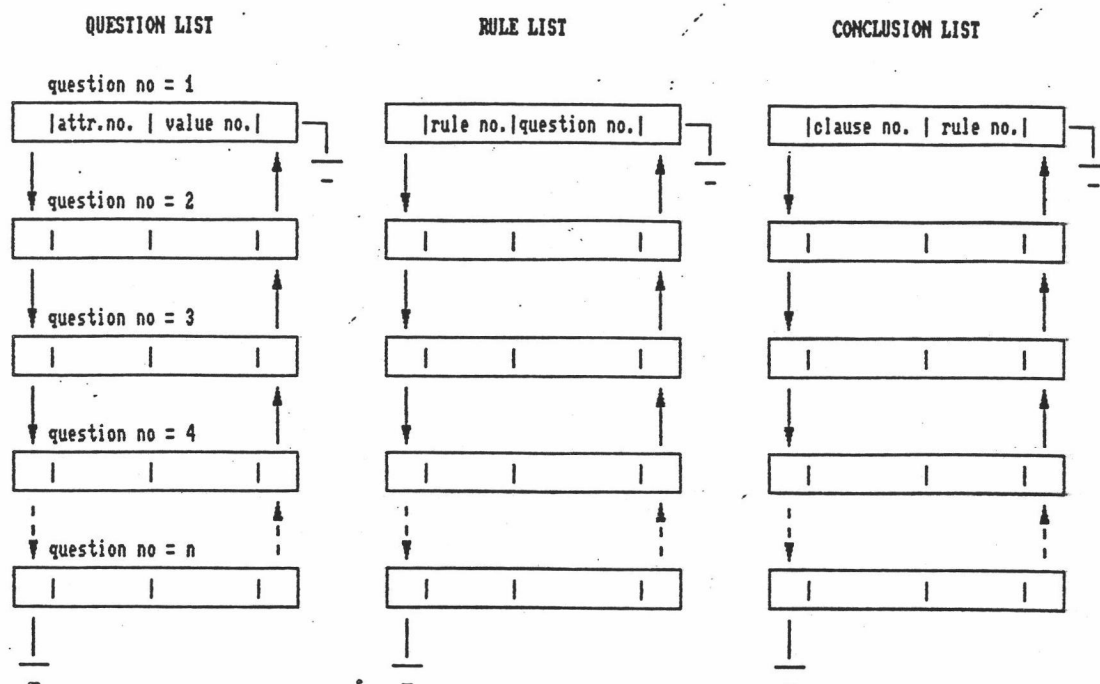
ข้อความที่เป็นคำถามจะถูกเก็บในลักษณะเช่นเดียวกันกับข้อความที่เป็นคำแนะนำหรือข้อสรุปต่างๆและเก็บไว้ในไฟล์ TEXT เช่นเดียวกัน

ส่วน value นั้นก็จะถูกเก็บเป็นลิงค์ลิสต์ โดยจะเก็บเลขที่ของ value และตัวบอกตำแหน่งของเลขที่ value ถัดไปไว้เป็น 1 ระเบียบของไฟล์ NUMBER สำหรับเลขที่ value นั้นระบบจะเป็นผู้กำหนดให้ในลักษณะเดียวกับเลขที่ของย่อประกอบ

4.4 โครงสร้างข้อมูลหน่วยความจำใช้งานของส่วนให้คำปรึกษา

ในระหว่างทำการปรึกษาระบบจะใช้หน่วยความจำส่วนหนึ่งสำหรับเก็บข้อมูลต่างๆเพื่อใช้ในการอนุมาน หน่วยความจำใช้งานนี้ประกอบด้วย

3 ลิงค์ลิสต์แบบ 2 ทิศทาง(doubly linked list)คือ QUESTION LIST, RULE LIST, และ CONCLUSION LIST ดังในรูปที่ 3.4



รูปที่ 3.4 โครงสร้างข้อมูลหน่วยความจำใช้งานของส่วนให้คำปรึกษา

4.4.1 QUESTION LIST เป็นลิงค์ลิสต์ที่ใช้เก็บ

ข้อเท็จจริงที่ระบบได้จากการสอบถามผู้ใช้ ทุกคำตอบที่ผู้ใช้ตอบคำถามของระบบ จะถูกเก็บไว้ในลิสต์นี้ ดังนั้นการตอบคำถาม 1 ครั้งจะทำให้จำนวนโหนดในลิสต์เพิ่มขึ้น 1 โหนด กำหนดให้โหนดแรกของลิสต์เป็นโหนดหมายเลข 1 และเพิ่มขั้นที่ละ 1 ในโหนดถัดไป แต่ละโหนดประกอบด้วยข้อมูล 4 ส่วนคือ

4.4.1.1 ส่วนที่เก็บที่อยู่ของโหนดถัดไป

ชนิดของข้อมูลเป็นแบบ POINTER ในโหนดสุดท้ายของลิงค์ลิสต์ส่วนนี้จะมีค่าเป็น 0 (NULL)

4.4.1.2 ส่วนที่เก็บเลขที่ของ attribute

ซึ่งเป็น attribute ที่ระบบต้องการทราบค่า value ของมันโดยการสอบถามจากผู้ใช้ ชนิดของข้อมูลเป็นแบบ NUMBER

4.4.1.3 ส่วนที่เก็บเลขที่ของ value

จะเก็บเลขที่ของ value คำตอบ ระบบจะใช้ value ที่ผู้ใช้ตอบไปค้นหาเลขที่ของ

มันแล้วเอามาเก็บไว้ที่ส่วนนี้ ชนิดของข้อมูลเป็นแบบ NUMBER

4.4.1.4 ส่วนที่เก็บที่อยู่ของ โหนดก่อนหน้า
ชนิดของข้อมูลเป็นแบบ POINTER ในโหนดแรกของลิสต์ส่วนนี้จะมีค่าเป็น 0
(NULL)

4.4.2 RULE LIST เป็นลิสต์ที่ใช้เก็บเลขที่ของ
กฎที่ถูกทดสอบแล้วและได้ผลเป็นจริง แต่ละโหนดประกอบด้วยข้อมูล 4 ส่วนคือ

4.4.2.1 ส่วนที่เก็บที่อยู่ของ โหนดถัดไป
ลักษณะเหมือนกับของ QUESTION LIST

4.4.2.2 ส่วนที่เก็บเลขที่ของกฎที่ถูกทดสอบ
แล้วและได้ผลเป็นจริง ชนิดของข้อมูลเป็นแบบ NUMBER

4.4.2.3 ส่วนที่เก็บหมายเลขของ โหนดใน
QUESTION LIST ที่ทำให้ผลการทดสอบเงื่อนไขท้ายสุดของกฎนั้นเป็นจริง ชนิด
ของข้อมูลเป็นแบบ NUMBER

4.4.2.4 ส่วนที่เก็บที่อยู่ของ โหนดก่อนหน้า
ลักษณะเหมือนกับของ QUESTION LIST

4.4.3 CONCLUSION LIST เป็นลิสต์ที่ใช้เก็บ
เลขที่ของอนุประโยคในส่วนสรุปของกฎที่ถูกทดสอบแล้วและได้ผลเป็นจริง แต่ละ
โหนดประกอบด้วยข้อมูล 4 ส่วนคือ

4.4.3.1 ส่วนที่เก็บที่อยู่ของ โหนดถัดไป
ลักษณะเหมือนกับของ QUESTION LIST

4.4.3.2 ส่วนที่เก็บเลขที่ของอนุประโยคใน
ส่วนสรุปของกฎที่ถูกทดสอบแล้วและได้ผลเป็นจริง ชนิดของข้อมูลเป็นแบบ
NUMBER

4.4.3.3 ส่วนที่เก็บเลขที่ของกฎที่เป็นเจ้าของ
อนุประโยคนั้น ชนิดของข้อมูลเป็นแบบ NUMBER

4.4.3.4 ส่วนที่เก็บที่อยู่ของ โหนดก่อนหน้า
ลักษณะเหมือนกับของ QUESTION LIST