บทที่ 5

ระบบเปลือกผู้เชี่ยวชาญ GES

ระบบเปลือกผู้เชี่ยวชาญ คือ โปรแกรมคอมพิวเตอร์ที่ทำหน้าที่แก้ปัญหาโดยอาศัยกระบวนการอนุมาน (inference procedure) โดยในตอนต้นยังไม่สามารถแก้ปัญหาเรื่องใด ๆ ได้ จนกว่าจะมีการป้อนความรู้เข้าไป หลังจากนั้นระบบจะกลายเป็นระบบผู้เชี่ยวชาญ (expert system) ที่สามารถแก้ปัญหาเรื่องใดเรื่องหนึ่งโดยเฉพาะได้ ซึ่งจะมีความสามารถเทียบเท่ากับผู้เชี่ยวชาญในสาขานั้น ๆ

GES ย่อมาจากคำว่า General expert shell เป็นระบบเปลือกผู้เชี่ยวชาญทั่วไป ที่สามารถนำไปประยุกต์ใช้งานในด้านต่าง ๆ ระบบเปลือกผู้เชี่ยวชาญ GES นี้ถูกพัฒนาขึ้นจากโปรแกรมภาษา LISP ในปี พ.ศ.2543 โดยนักศึกษาภาควิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่ การใช้งานระบบเปลือกผู้เชี่ยวชาญทำได้โดยการแทนความรู้ให้อยู่ในรูปแบบของกฎ ความจริง และมีการอนุมานทั้งแบบเดินหน้า แบบย้อนหลัง และแบบผสม

จากที่กล่าวมาจะเห็นว่า ระบบเปลือกผู้เชี่ยวชาญเป็นส่วนที่สำคัญมากในการพัฒนาระบบผู้เชี่ยวชาญ ในงานวิจัยนี้จะใช้ระบบเปลือกผู้เชี่ยวชาญ GES ซึ่งเป็นระบบเปลือกผู้เชี่ยวชาญทั่วไปที่สามารถนำไปประยุกต์ใช้งานในด้านต่าง ๆ และสามารถสร้างฐานความรู้ใหม่ ๆ ในภายหลังได้ตามความต้องการ ทำให้ฐานความรู้นั้นมีความทันสมัยอยู่เสมอ

5.1 สถาปัตยกรรมของระบบเปลือกผู้เชี่ยวชาญ GES

สถาปัตยกรรมของระบบเปลือกผู้เชี่ยวชาญ GES ซึ่งแสดงในภาพประกอบ 5-1 มีส่วนประกอบหลักที่สำคัญ 5 ส่วนดังนี้

(1) ฐานความรู้

ฐานความรู้ในระบบเปลือกผู้เชี่ยวชาญ GES จะทำการเก็บความรู้ในรูปของกฎและความจริงซึ่งผ่านการแปลงจากภาษาธรรมชาติมาเป็นภาษาที่ระบบสามารถเข้าใจได้ โดยในส่วนนี้ผู้ใช้หรือผู้พัฒนาระบบสามารถที่จะแก้ไขดัดแปลงระบบได้ง่ายโดยผ่านการทำงานของระบบเปลือกที่มีเมนูต่าง ๆ ให้ผู้ใช้เลือกใช้ โดยสามารถเพิ่ม ลบ หรือแก้ไขดัดแปลงกฎได้โดยสะดวก

(2) การอนุมาน

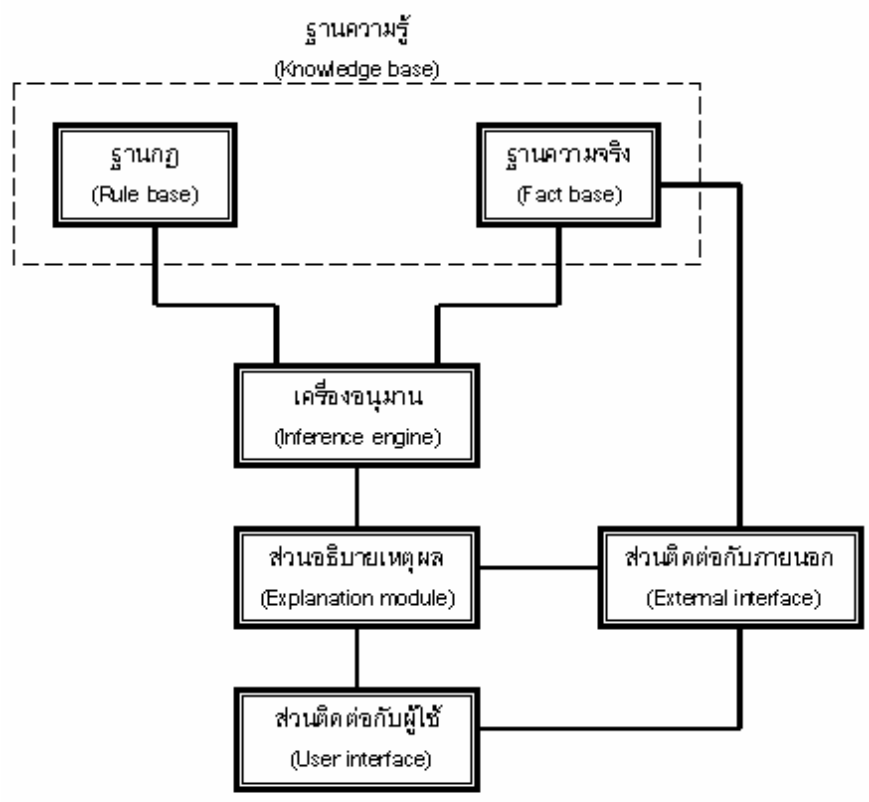
เป็นส่วนที่ทำหน้าที่ควบคุมการใช้ความรู้ในฐานความรู้เพื่อแก้ไขปัญหาได้อย่างมีประสิทธิภาพโดยการอนุมานของระบบเปลือกผู้เชี่ยวชาญ GES นี้จะมีทั้งการอนุมานแบบเดินหน้า การอนุมานแบบถอยหลัง และการอนุมานแบบผสม

(3) ส่วนติดต่อกับภายนอก

เป็นส่วนที่ทำหน้าที่ติดต่อกับโปรแกรมภายนอกระบบ โดยที่ระบบเปลือกผู้เชี่ยวชาญ GES สามารถติดต่อกับโปรแกรมภายนอก ซึ่งอาจจะเป็นโปรแกรมที่สามารถทำงานได้ทันที (Executable file) และระบบยังสามารถส่งค่าอินพุตและรับค่าเอาท์พุตของโปรแกรมภายนอกได้อีกด้วย เช่น การสั่งพิมพ์ การคำนวณ เป็นต้น

(4) ส่วนอธิบายเหตุผล

เมื่อผู้ใช้ปรึกษาระบบผู้เชี่ยวชาญและได้คำตอบออกมาเพียงคำตอบเดียว ผู้ใช้อาจไม่มั่นใจว่าคำตอบนั้นจะนำไปแก้ปัญหาได้จริงแค่ไหน การอธิบายเหตุผลให้ผู้ใช้ทราบอาจทำให้ผู้ใช้นำคำตอบที่ได้ไปใช้อย่างมั่นใจขึ้น



ภาพประกอบ 5-1 สถาปัตยกรรมของระบบเปลือกผู้เชี่ยวชาญ

(5) ส่วนติดต่อกับผู้ใช้

เป็นส่วนที่ผู้ใช้ทั่วไปและผู้พัฒนาระบบสามารถติดต่อหรือโต้ตอบกับระบบเปลือก โดยระบบเปลือกผู้เชี่ยวชาญ GES สามารถติดต่อกับผู้ใช้โดยผ่านระบบเมนูของระบบเปลือกที่ให้ผู้ใช้สามารถเลือกทำงานอย่างใดอย่างหนึ่งได้ เช่น การพัฒนาส่วนของฐานความรู้ การเลือกการอนุมาน การแสดงข้อมูลในฐานความรู้ตลอดจนการแก้ไขดัดแปลงต่าง ๆ

5.2 การจัดการฐานความรู้ (Knowledge base management)

ในระบบผู้เชี่ยวชาญจำเป็นต้องมีฐานความรู้เพื่อนำมาอนุมานในระบบ ซึ่งฐานความรู้ได้มาจากการป้อนของผู้ใช้หรือผู้พัฒนาระบบ โดยแบ่งฐานความรู้ออกเป็น 2 ประเภท คือ ฐานกฎ และฐานความจริง ดังนั้นผู้ใช้จำเป็นต้องเข้าใจการพัฒนาฐานความรู้ให้กับระบบ เพื่อให้ระบบทำงานได้อย่างถูกต้อง

5.2.1 การแทนความรู้ (Knowledge representation)

การแทนความรู้ของระบบเปลือกผู้เชี่ยวชาญ GES นี้จะใช้การแทนในรูปแบบของกฎ (Rule of production) และความจริง (Fact) เนื่องจากความรู้ทั่วไปไม่ว่าจะเป็นเรื่องอะไรก็ตามจะอยู่ในลักษณะของกฎและความจริง และการแทนความรู้ในรูปแบบนี้ยังง่ายต่อการทำความเข้าใจอีกด้วย ดังนั้น GES จึงเป็นระบบเปลือกผู้เชี่ยวชาญที่มีความยืดหยุ่นในการแทนความรู้และสามารถสร้างความรู้ได้โดยง่าย

(1) ฐานความจริง

ความจริง คือ สิ่งที่เป็นจริง ปฏิเสธไม่ได้ ในฐานความรู้ของระบบนั้นจำเป็นต้องมีฐานความจริงเริ่มต้นเสมอ หรือระบบมีฐานความจริงอยู่ก่อนแล้ว โดยมีโครงสร้างดังนี้

Fact N
((Type_of_object Object) (Attribute) Comparator (Type_of_value Value))

เมื่อ Type_of_object คือ ชนิดของสิ่งที่สนใจ
Object คือ ชื่อของสิ่งที่สนใจ
Attribute คือ ความสัมพันธ์หรือคุณลักษณะของสิ่งที่สนใจ
Comparator คือ การเปรียบเทียบสิ่งที่สนใจกับค่าผลลัพธ์
Type_of_value คือ ชนิดของค่าผลลัพธ์
Value คือ ค่าผลลัพธ์

ตัวอย่าง

```
Fact 1
((show  menu)(picture) = (atomic true))
Fact 2
((wire  main_line)(dimension in sq.mm.) = (numerical 150))
```

(2) ฐานกฎ

กฎ คือ การแสดงความรู้ในรูปแบบของ IF-THEN statement หรือเรียกว่า Production rules ซึ่งประกอบด้วยส่วนของเงื่อนไข (Condition) และ ข้อสรุป (Conclusion) ถ้าระบบทำการตรวจสอบว่าถูกต้องตรงตามเงื่อนไขก็จะดำเนินการตามส่วนของข้อสรุป โดยมีโครงสร้างดังนี้

Identify N
name_of_rule
If
(Condition 1)
(Condition 2)
……...
Then
(Conclusion 1)
(Conclusion 2)
……...
End-of-rule

หลักการเขียนกฎ

- ส่วนของ condition มีจำนวนเท่าไรก็ได้
- ส่วนของ conclusion มีจำนวนเท่าไรก็ได้ ซึ่งภายใน conclusion สามารถเรียกส่วนติดต่อผู้ใช้หรือส่วนติดต่อภายนอกได้

ตัวอย่าง

```
Identify 1
Find_CB
If
  ((wire phase_A)(current in Amp.) = (numerical (? Lp_A)))
  ((wire phase_B)(current in Amp.) = (numerical (? Lp_B)))
Then
```

```
  (@ max_current_phase)
  ((current CB)(current in Amp.) = (numerical (? i_CB))))
End-of-rule
```

(3) ตัวแปร (Variable)

ในการอ้างถึงความรู้หนึ่ง ซึ่งผู้ใช้ต้องการทราบค่า มีโครงสร้างของการสร้างตัวแปรดังนี้

(? Name_variable)

โดยค่าที่เป็นตัวแปรจะเรียกใช้ภายในกฎเท่านั้น ซึ่งจะเป็นได้ทั้ง Object และ Value

(4) การติดต่อกับส่วนอื่น ๆ หรือส่วน Action

ในการอ้างถึงความรู้หนึ่งภายในกฎ หรือมีการเรียก Action ในกรณีที่ต้องการติดต่อกับภายนอกและภายในระบบ เช่น ใช้เมื่อต้องการคำนวณ สั่งพิมพ์ เป็นต้น ซึ่งจะมีการเรียกใช้ในส่วนเงื่อนไข หรือข้อสรุปของกฎ โดยมีโครงสร้างดังนี้

(@ Name_action)

การเรียก Action มีด้วยกัน 2 ประเภทคือ

1. Internal Action เป็นลักษณะการเรียกใช้งาน Action ที่อยู่ภายในระบบผู้เชี่ยวชาญ โดยใช้ภาษา LISP ในการเขียนฟังก์ชั่น

(Name_action (function_action) “Path_of_file_ function_action”)

เมื่อ name_action คือ ชื่อของ Action

function_action คือ ชื่อของฟังก์ชั่น ภายในมีคำสั่งที่ต้องการให้ระบบทำตามคำสั่งนั้น

“Path_of_file_ function_action” คือ ชื่อไฟล์โดยมีการบอกเส้นทางที่เก็บฟังก์ชั่นที่ได้เขียนไว้

ตัวอย่าง

ต้องการ Internal Action เพื่อนำไปเรียกในส่วนของ condition หรือ conclusion ของกฎ คือ

(@ bonding_jumper)

- กำหนดชื่อของ Action และส่วนประกอบต่าง ๆ ในไฟล์ action.txt ของระบบผู้เชี่ยวชาญ

  (bonding_jumper (fn_bonding_jumper) "./engine_expert/utili/call_action.lsp")

- ทำการสร้างไฟล์ที่กำหนด

ในกรณีที่เป็น action ที่มีการกำหนด Internal Action และฟังก์ชั่นที่เขียนขึ้นใช้ภาษา LISP เช่น ในไฟล์ call_action.lsp มีฟังก์ชั่นชื่อ fn_bonding_jumper ดังตัวอย่างไว้แล้วข้างต้น

```
(defun fn_bonding_jumper ()
   (setq bj (round (/ (* 12.5 d_ml) 100)))
   (setq ans (list 'd_dj bj))
   (setq *binding-stream*(cons(cons ans(car *binding-
   stream*)) (cdr *binding-stream*)))
   (fn_set_variable_macth (reverse (first *binding-
stream*)))
)
```

- ใส่ path ของไฟล์ call_action.lsp จากตัวอย่างคือ ./engine_expert/utili/call_action.lsp

2. External Action เป็นลักษณะการเรียกใช้งาน Action ที่อยู่นอกระบบผู้เชี่ยวชาญ โดยระบบที่ติดต่อมี 2 แบบ คือ

- OS Command เป็นโปรแกรมในระบบปฏิบัติการ Linux เช่น Kview, Kwrite, C Language
- Executable file เป็นไฟล์ที่สามารถรันได้ทันทีที่เรียกใช้

(Name_action "Path_of_file_action" OS)

เมื่อ name_action คือ ชื่อของ Action

function_actionคือ ชื่อของฟังก์ชั่นภายในมีคำสั่งที่ต้องการให้ระบบทำตามคำสั่งนั้น

"Path_of_ file_action" คือ ชื่อไฟล์โดยบอกเส้นทางของไฟล์ Execute file

ตัวอย่าง

ต้องการ External Action แบบ OS command เพื่อนำไปเรียกในส่วนของ condition หรือ conclusion ของกฎ คือ *(@ table_wire)*

- กำหนดชื่อของ Action และส่วนประกอบต่าง ๆ ในไฟล์ action.txt ของระบบผู้เชี่ยวชาญ

  (table_wire "kview ./engine_expert/data_action/test1.jpg" OS)

- ใส่โปรแกรมที่ต้องการติดต่อ ในตัวอย่างเป็นการเรียกโปรแกรม kview ในการดูไฟล์ test1.jpg

5.2.2 การสร้างฐานความรู้ใหม่

การสร้างฐานความรู้ใหม่แบ่งออกได้เป็น 2 ประเภท คือ

1. การสร้างฐานความรู้ใหม่โดยการโหลดไฟล์จากภายนอก

การสร้างฐานความรู้ใหม่โดยการโหลดไฟล์ข้อมูลที่ผู้ใช้เขียนขึ้นมานั้น มีข้อดีคือผู้ใช้สามารถสร้างกฎได้มากและรวดเร็วกว่าการป้อนโดยผ่านระบบ แต่ผู้ใช้จะต้องมีความรู้เกี่ยวกับไวยากรณ์ในการสร้างกฎและความจริงเสียก่อน จากนั้นก็เขียนกฎและความจริงในโปรแกรม text editor ทั่วไปแล้วนำมาโหลดเข้าระบบ ระบบจะทำการตรวจสอบไวยากรณ์ของกฎและความจริงให้ถูกต้องก่อนที่จะบันทึกเป็นฐานความรู้ให้กับระบบต่อไป

- Load rule file เป็นส่วนที่ใช้ในการโหลดไฟล์กฎภายนองระบบที่ผู้ใช้เขียนขึ้นตามไวยากรณ์ที่กำหนดไว้
- Load fact file เป็นส่วนที่ใช้ในการโหลดไฟล์ความจริงภายนอกระบบที่ผู้ใช้เขียนขึ้นตามไวยากรณ์ที่กำหนดไว้

2. การสร้างฐานความรู้ใหม่โดยการป้อนผ่านระบบ

การสร้างฐานความรู้ใหม่โดยการป้อนฐานความรู้ผ่านระบบจะมีข้อดีคือ ผู้ใช้ไม่จำเป็นต้องมีความรู้เกี่ยวกับไวยากรณ์ในการสร้างกฎและความจริงของระบบก็สามารถสร้างกฎและความจริงขึ้นมาใช้งานได้ โดยการป้อนข้อมูลตามที่ระบบต้องการผ่านส่วนที่ช่วยสร้างกฎและความจริง แต่มีข้อเสียคือการสร้างฐานความรู้แบบนี้จะใช้เวลามากกว่าเมื่อเทียบกับแบบแรก

5.2.3 การปรับปรุงฐานความรู้

การปรับปรุงฐานความรู้ เป็นการปรับปรุงความรู้ในฐานความรู้เพื่อให้ระบบที่พัฒนาขึ้นนั้นมีความรู้ที่ถูกต้องและทันสมัยอยู่เสมอในการแก้ปัญหา โดยระบบเปลือกผู้เชี่ยวชาญ GES ได้เตรียมส่วนที่ใช้ในการปรับปรุงฐานความรู้โดยการใช้งานจากระบบเปลือกที่สามารถปรับปรุงฐานความรู้ต่าง ๆ ได้

ซึ่งส่วนนี้จะช่วยให้การพัฒนาระบบผู้เชี่ยวชาญให้มีความสะดวกมากขึ้น คือผู้พัฒนาระบบสามารถที่จะปรับปรุงฐานความรู้เดิมได้ทันที โดยที่ไม่ต้องสร้างฐานความรู้ใหม่ทั้งระบบ ซึ่งส่วนของการปรับปรุงฐานความรู้นี้สามารถจะปรับปรุงได้ทั้งกฎและความจริงในระบบ เมื่อผู้พัฒนาระบบปรับปรุงความรู้แล้ว ระบบจะทำการแก้ไขความสัมพันธ์ระหว่างความจริงกับกฎใหม่ เพื่อให้ถูกต้องตามฐานความรู้ใหม่ที่ถูกปรับปรุงขึ้น

5.3 การอนุมานฐานความรู้ (Utilization of System)

เครื่องอนุมานทำหน้าที่เปรียบเสมือนสมองของระบบที่ใช้ในการหาข้อสรุป เครื่องอนุมานจึงเป็นส่วนที่สำคัญที่สุดส่วนหนึ่งของระบบผู้เชี่ยวชาญสำหรับใช้ในการหาเหตุผลแล้วจึงสรุปออกมาเป็นคำตอบหรือคำแนะนำ ระบบเปลือกผู้เชี่ยวชาญ GES สามารถใช้การอนุมานพื้นฐานทั้งการอนุมานแบบเดินหน้า การอนุมานแบบถอยหลัง และการอนุมานแบบผสมได้

5.3.1 การอนุมานแบบเดินหน้า

การอนุมานแบบเดินหน้าของระบบเปลือกผู้เชี่ยวชาญ GES จะเริ่มต้นอนุมานจากความสัมพันธ์ระหว่างความจริงกับกฎที่สร้างขึ้นในฐานความรู้ โดยจะเปรียบเทียบกับเงื่อนไขของกฎ ซึ่งกฎที่มีเงื่อนไขถูกต้องก็จะถูกปฏิบัติการในส่วนของข้อสรุป กฎต่าง ๆ จะถูกอนุมานในลักษณะนี้เรื่อย ๆ จนกว่าจะได้คำตอบหรือบรรลุเป้าหมาย ซึ่งเป็นกระบวนการสร้างฐานความรู้ขึ้นมาใหม่

ข้อดีของการอนุมานแบบเดินหน้า คือ จะทำการค้นหาทุก ๆ คำตอบหรือเป้าประสงค์ที่เป็นไปได้ แต่ในบางครั้งอาจจะไม่ได้คำตอบหรือเป้าประสงค์ที่ต้องการ ดังนั้นจึงเป็นการเสียเวลามากต่อระบบที่ต้องค้นหาคำตอบของแต่ละปัญหา

5.3.2 การอนุมานแบบย้อนหลัง

การอนุมานแบบย้อนหลังระบบเปลือกผู้เชี่ยวชาญ GES จะเริ่มต้นอนุมานจากเป้าประสงค์ที่ต้องการแล้วนำไปหากฎที่จำเป็นต้องนำมาใช้ในการอนุมาน ซึ่งกฎดังกล่าวจะต้องมีข้อสรุปเหมือนกับเป้าประสงค์ หลังจากนั้นจะทำการตรวจสอบเงื่อนไข ถ้าเงื่อนไขของกฎเป็นจริงตามข้อเท็จจริงที่มีอยู่ในฐานความรู้ เป้าประสงค์ดังกล่าวก็จะบรรลุผลเป็นจริง แต่ถ้ามีบางเงื่อนไขหรือทุกเงื่อนไขของกฎไม่เป็นจริงตามข้อเท็จจริงในฐานความรู้ เงื่อนไขเหล่านี้ก็จะกลายเป็นเป้าประสงค์ย่อย ๆ เพื่ออนุมานในขั้นต่อไป โดยจะดำเนินไปในแบบลูกโซ่ จนกว่าจะพบกฎที่มีเงื่อนไขจากข้อเท็จจริงและทำให้เป้าประสงค์ย่อยเหล่านั้นบรรลุผล

ข้อดีของการอนุมานแบบย้อนหลัง คือ จะค้นหาเฉพาะเป้าประสงค์ที่ต้องการเท่านั้นจึงใช้เวลาน้อยกว่า แต่ในบางครั้งอาจจะหาเป้าหมายไม่ได้เลย เนื่องจากเงื่อนไขในการนำไปสู่เป้าประสงค์นั้นไม่ถูกต้อง

5.3.3 การอนุมานแบบผสม

การอนุมานแบบผสมจะเป็นลักษณะการอนุมานที่รวมกันระหว่างการอนุมานแบบเดินหน้าและการอนุมานแบบย้อนหลัง กล่าวคือ เริ่มต้นระบบจะรับเป้าประสงค์จากผู้ใช้ไปทำการอนุมานแบบถอยหลัง ซึ่งหากเป้าประสงค์เป็นจริง ระบบจะนำเส้นทางของการอนุมานนั้นไปทำการอนุมานแบบเดินหน้า เพื่อให้ได้ความจริงใหม่ตามเส้นทางของการอนุมานแบบย้อนหลัง

5.4 ส่วนแสดงฐานความรู้ในระบบ (Show knowledge base)

ส่วนแสดงฐานความรู้ในระบบ เป็นส่วนหนึ่งของระบบเปลือกผู้เชี่ยวชาญ GES ที่ผู้ใช้สามารถเรียกดูว่ากฎและความจริงที่มีในระบบมีอะไรบ้าง โดยส่วนนี้สามารถที่จะแสดงกฎและความจริงบางส่วนหรือทั้งหมดที่มีอยู่ในระบบได้ บางครั้งผู้ใช้สามารถที่จะค้นหาคำที่ต้องการในฐานความรู้ในระบบ โดยระบุคำที่ต้องการเพื่อให้ระบบเปลือกค้นหาก็ได้

5.5 ส่วนอธิบายการการใช้งาน (Explanation about utilization)

ในระบบเปลือกผู้เชี่ยวชาญ GES ส่วนอธิบายการกำหนดค่าต่าง ๆ เป็นส่วนที่ใช้สำหรับการอธิบายการใช้งานโปรแกรมในส่วนต่าง ๆ เช่น การเขียนกฎหรือความจริงให้ถูกต้องตามหลักไวยากรณ์ การเรียกใช้งานส่วนติดต่อกับภายนอก และส่วนอื่น ๆ ในระบบเปลือกผู้เชี่ยวชาญที่ผู้ใช้ต้องการคำอธิบาย ซึ่งส่วนนี้เปรียบเสมือน Help ในระบบนั่นเอง