บทที่ 4

GES : ระบบเปลือกผู้เชี่ยวชาญ

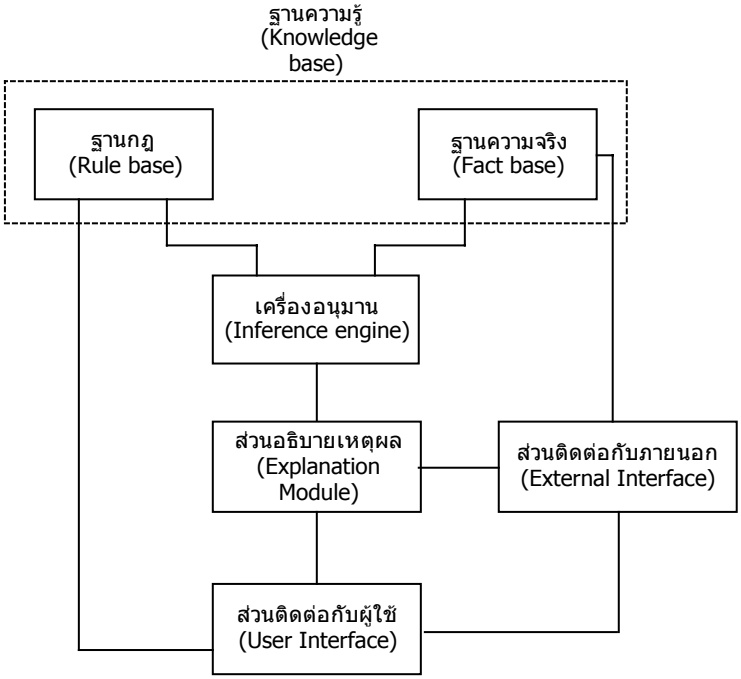
ระบบเปลือกผู้ชำนาญการ คือ โปรแกรมคอมพิวเตอร์ที่ทำหน้าที่แก้ปัญหาโดยอาศัยกระบวนการอนุมาน (Inference Procedure) โดยในตอนต้นยังไม่สามารถแก้ปัญหาเรื่องใดๆ ได้ จนกว่าจะมีการป้อนความรู้เข้าไปให้กับระบบเปลือกผู้ชำนาญการ หลังจากนั้นระบบเปลือกผู้ชำนาญการจะกลายเป็นระบบผู้ชำนาญการที่สามารถแก้ปัญหาเรื่องใดเรื่องหนึ่งโดยเฉพาะได้ ซึ่งจะมีความสามารถเทียบเท่ากับผู้เชี่ยวชาญในสาขานั้นๆ

GES ย่อมาจากคำว่า General Expert Shell เป็นระบบเปลือกผู้เชี่ยวชาญทั่วไป ที่สามารถนำไปประยุกต์ใช้งานในด้านต่างๆ ระบบเปลือกผู้เชี่ยวชาญ GES นี้ถูกพัฒนาขึ้นในปี พ.ศ. 2543 พัฒนาขึ้นโดยใช้ภาษา LISP โดย ถิรวัฒน์ หงส์บิน และคณะ จากมหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่ การใช้งานระบบเปลือกผู้เชี่ยวชาญ ทำได้โดยการแทนความรู้อยู่ในรูปแบบของกฎ ความจริง และมีการอนุมานทั้งแบบเดินหน้า แบบย้อนหลัง และแบบผสม

4.1 สถาปัตยกรรมของระบบเปลือกผู้เชี่ยวชาญ GES

สถาปัตยกรรมของระบบเปลือกผู้เชี่ยวชาญ GES แสดงดังในภาพประกอบ 4-1 องค์ประกอบของระบบเปลือกผู้เชี่ยวชาญ GES ประกอบไปด้วยส่วนหลักที่สำคัญ 5 ส่วนดังนี้

- *ฐานความรู้ (Knowledge Base)*
- *เครื่องอนุมาน (Inference Engine)*
- *ส่วนอธิบายเหตุผล (Explanation Module)*
- *ส่วนติดต่อกับผู้ใช้ (User Interface)*
- *ส่วนติดต่อกับภายนอก (External Interface)*



ภาพประกอบ 4-1 สถาปัตยกรรมของระบบเปลือกผู้เชี่ยวชาญ

4.1.1 ฐานความรู้

ฐานความรู้ประกอบด้วยความจริง (Facts) และ กฎ (Production Rules) ที่เกี่ยวข้องกับปัญหาที่ต้องการแก้ไข ฐานความรู้ของระบบผู้ชำนาญการเป็นการรวบรวมความรู้เฉพาะทางที่ได้จากเอกสาร ตำรา จากการสอบถามจากผู้เชี่ยวชาญ และข้อมูลความรู้ที่ได้ในระหว่างการถามโต้ตอบระหว่างระบบกับผู้ใช้

4.1.2 เครื่องอนุมาน

เครื่องอนุมานทำหน้าที่เป็นเครื่องมือในการควบคุมระบบและเป็นส่วนที่ใช้ในการหาข้อสรุป เครื่องอนุมานเป็นส่วนที่สำคัญที่สุดส่วนหนึ่งของระบบผู้ชำนาญการสำหรับใช้ในการหาเหตุผลแล้วจึงสรุปออกมาเป็นคำตอบหรือคำแนะ

4.1.3 ส่วนอธิบายเหตุผล

เมื่อผู้ใช้ปรึกษาระบบชำนาญการและได้คำตอบออกมา เพียงคำตอบอย่างเดียว ผู้ใช้อาจไม่มั่นใจว่าคำตอบนั้นจะนำไปแก้ปัญหาได้จริงแค่ไหน การอธิบายเหตุผลให้ผู้ใช้ทราบอาจทำให้ผู้ใช้นำคำตอบที่ได้ไปใช้อย่างมั่นใจยิ่งขึ้น

4.1.4 ส่วนติดต่อกับผู้ใช้

ส่วนติดต่อกับผู้ใช้เป็นส่วนที่ใช้สำหรับการถ่ายทอดข้อมูลการโต้ตอบระหว่างผู้ใช้ระบบ และข้อมูลดังกล่าวจะถูกนำไปเก็บไว้ในฐานข้อมูลแบบไดนามิก การที่โปรแกรมจะใช้งานได้ง่าย หรือยากเพียงใดขึ้นกับการออกแบบในส่วนนี้

4.1.5 ส่วนติดต่อกับภายนอก

ส่วนติดต่อกับภายนอกเป็นส่วนที่ระบบต้องการติดต่อกับโปรแกรมภายนอกหรือภายในระบบเอง เช่น การคำนวณ การแสดงภาพ การแสดงข้อความส่วนอธิบาย การสั่งพิมพ์ เป็นต้น ซึ่งในแต่ละกฎสามารถเรียกการแสดงได้ทั้งในส่วนเงื่อนไขและผลสรุป เป็นต้น

4.2 การจัดการฐานความรู้ (Knowledge Base Management)

สำหรับระบบผู้ชำนาญการเพื่อช่วยวิเคราะห์การประหยัดพลังงานไฟฟ้าในโรงงาน แบ่งฐานความรู้ออกได้เป็น 2 ส่วน คือ ฐานข้อมูลสถิต (Static Database) ฐานข้อมูลชนิดนี้จะเป็นฐานข้อมูลที่อยู่คงที่ในโปรแกรมซึ่งจะไม่แปรเปลี่ยนขณะใช้งาน และฐานข้อมูลไดนามิก (Dynamic Database) เป็นข้อมูลที่ระบบได้รับจากการโต้ตอบกับผู้ใช้ และฐานข้อมูลแบบไดนามิกนี้ระบบจะเก็บไว้ชั่วคราวจนกว่าการอนุมานจะเสร็จสิ้นลงก็จะทำการลบฐานข้อมูลแบบไดนามิกนี้

4.2.1 ความจริง (Fact)

ความจริง คือ สิ่งที่เป็นจริง ปฏิเสธไม่ได้ ในฐานความรู้ของระบบจำเป็นต้องมีฐานความจริงเริ่มต้น การเขียนความจริงในระบบมีรูปแบบของไวยากรณ์ ดังแสดงในภาพประกอบ 4-2

```
Fact   n
((Type-of-Object Object) (Attribute) Comparator (Type-of-Value Value))
```

ภาพประกอบ 4-2 ไวยากรณ์ของการเขียนความจริง

จากไวยากรณ์ข้างต้นสามารถอธิบายได้ดังนี้

(1) Type-of-Object เป็นการอ้างถึงความรู้หนึ่ง สิ่งที่อ้างถึงต้องบ่งชี้ชนิดของสิ่งนั้นและกำหนด type สำหรับทุก ๆ object เพื่อความสะดวกในการแยกชนิดของสิ่งที่กล่าวถึง

(2) Object ในการอ้างถึงความรู้หนึ่ง สิ่งที่ถูกอ้างถึงมีการกำหนดชื่อของ object ของความรู้นั้นนั่นเอง สามารถแบ่งเป็นได้ 2 แบบ คือ

- Normal form คือ Object รูปแบบธรรมดา ซึ่งค่าของ Object จะเป็นลักษณะเป็นAtomic (คำ)
- Variable คือ Object รูปแบบตัวแปร ซึ่งค่าของ Object จะถูกหาค่าในการอนุมาน โดยมีรูปแบบ (? Name_variable)

(3) Attribute หมายถึงสถานะของวัตถุที่เราอ้างถึงทั้งในกฎและความจริง มีได้ 2 แบบ คือ

- Atomic (คำ)
- ประโยค

(4) Comparator คือ ในการอ้างถึงความรู้หนึ่ง สิ่งที่ถูกอ้างจะมีการเปรียบเทียบกับค่า Value โดยใช้ส่วนประกอบ Comparator เป็นตัวเปรียบเทียบ

- ในระบบจะกำหนดค่าของ Comparator ไว้ 6 ชนิด คือ
  - = มีค่า เท่ากับ
  - /= มีค่า ไม่เท่ากับ
  - > มีค่า มากกว่า
  - >= มีค่า มากกว่าเท่ากับ
  - < มีค่า น้อยกว่า
  - <= มีค่า น้อยกว่าเท่ากับ
- กรณีที่ ค่า type value เป็น Atomic ค่าของ Comparator มี 2 ค่า คือ = , /=

(5) Type-of-Value ในการอ้างถึงความรู้หนึ่ง สิ่งที่ถูกอ้างจะมีค่าผลลัพธ์ (Value) เท่าใด โดยในส่วนของ type of value มีได้ 2 แบบคือ

- Numerical ชนิดตัวเลข มีค่ากับตัวเลขทางคณิตศาสตร์
- Atomic ชนิดคำ แปลความหมายตามคำนั้น ไม่มีค่าทางคณิตศาสตร์

(6) Value ในการอ้างถึงความรู้หนึ่ง สิ่งที่ถูกอ้างจะมีค่าผลลัพธ์เท่าใด

- ในกรณีของกฎ ภายในมีได้ 2 แบบ คือ

ก. Normal form คือ Object รูปแบบธรรมดา ซึ่งค่าของ Object จะเป็นลักษณะเป็นAtomic (คำ)

ข. Variable คือ Object รูปแบบตัวแปร ซึ่งค่าของ Object จะถูกหาค่าในการอนุมาน โดยมีรูปแบบ (? Name_variable)

- ในกรณีของความจริง ซึ่ง Argument ชนิด Value ของแต่ละความจริง เป็นได้ 1 แบบเท่านั้นคือ Normal form ในการสร้างความจริงจะมีตัวแปรภายในความจริงไม่ได้

4.2.2 กฎ (Rule)

กฎ คือ การแสดงความรู้ในรูปของ If-Then หรือเรียกว่า Production Rules อยู่ในรูปแบบของเงื่อนไขและข้อสรุป เป็นการแสดงความรู้ในรูปแบบของกลุ่มความจริง ถ้าระบบทำการตรวจสอบว่าถูกต้องตรงตามเงื่อนไข ก็จะดำเนินการตามส่วนของข้อสรุป การเขียนกฎในโปรแกรม มีรูปแบบของไวยากรณ์ดังแสดงในภาพประกอบ 4-3

```
identify   n
mane_rule
 if        (Condition 1)
           (Condtion 2)
           ...
           (Condition n)
then       (Conclusion 1)
           (Conclusion 2)
           ...
           (Conclusion n)
end-of-rule
```

ภาพประกอบ 4-3 ไวยากรณ์ของการเขียนกฎ

รูปแบบของการเขียนกฎ

- จำนวนของ Condition เป็นเท่าไรก็ได้
- จำนวนของ Conclusion เป็นเท่าไรก็ได้
- ภายใน Conclusion สามารถเรียกส่วนติดต่อผู้ใช้หรือส่วนติดต่อภายนอกได้

4.2.3 ตัวแปร (Variable)

ในการอ้างถึงความรู้หนึ่ง ผู้ใช้ต้องการทราบค่า โครงสร้างของการสร้างตัวแปรให้กับระบบเปลือกผู้เชี่ยวชาญ GES ดังแสดงในภาพประกอบ 4-4

(? Name-variable)

ภาพประกอบ 4-4 โครงสร้างของการสร้างตัวแปร

โดยค่าที่เป็นตัวแปร (Variable) จะเรียกใช้ภายในกฎเท่านั้น โดยจะอยู่ภายในส่วนของ Object และ Value

4.2.4 การติดต่อกับส่วนอื่นๆ หรือส่วน Action

ในการอ้างถึงความรู้หนึ่งภายในกฎ หรือมีการเรียก Action ในกรณีที่ต้องการติดต่อกับภายนอกหรือภายในระบบ เช่น ใช้เมื่อต้องการคำนวณ สั่งพิมพ์ เป็นต้น ซึ่งจะมีการเรียกใช้ในส่วนเงื่อนไขหรือ ข้อสรุปของกฎ โดยมีโครงสร้างไวยากรณ์ของการเรียก Action ดังแสดงในภาพประกอบ 4-5

(? Name-Action)

ภาพประกอบ 4-5 ไวยากรณ์ของการเรียก Action

การเรียก Action มีด้วยกัน 2 ประเภทคือ

- Internal Action เป็นลักษณะการเรียกใช้งาน Action ภายใน ที่อยู่นอกระบบผู้เชี่ยวชาญ โดยใช้ภาษา LISP ในการเขียนฟังก์ชัน มีลักษณะไวยากรณ์ดังภาพประกอบ 4-6

```
(name_action (function_action) "Path_of_file_function_action")
```

ภาพประกอบ 4-6 ไวยากรณ์ของการเรียก Internal Action

โดย name_action คือ ชื่อของ Action

function_action คือ ชื่อของฟังก์ชัน ภายในมีคำสั่งที่ต้องการให้ระบบทำตามคำสั่งนั้น

“Path_of_file_function_action” คือ ชื่อไฟล์โดยมีการบอกเส้นทางที่เก็บฟังก์ชันที่ได้เขียนไว้

- External Action เป็นลักษณะ Action ภายใน ที่อยู่นอกระบบผู้เชี่ยวชาญ ลักษณะการเขียนไวยากรณ์ของการเรียก External Action ดังแสดงในภาพประกอบ 4-7 โดยระบบที่ติดต่อมี 2 ประเภท คือ

ก. OS Command เป็นโปรแกรมในระบบปฏิบัติการ Linux เช่น Kview, Kwrite, C language

ข. Execute file เป็นไฟล์ที่สามารถทำงานได้ทันทีที่เรียกใช้

```
(name_action "Path_of_file_action" OS)
```

ภาพประกอบ 4-7 ไวยากรณ์ของการเรียก External Action

โดย name_action คือ ชื่อของ Action

function_action คือ ชื่อของฟังก์ชัน ภายในมีคำสั่งที่ต้องการให้ระบบทำตามคำสั่งนั้น

“Path_of_file_action” คือ ชื่อไฟล์โดยบอกเส้นทางของไฟล์ Execute file

4.3 การสร้างฐานความรู้ใหม่

การสร้างฐานความรู้ใหม่แบ่งออกได้เป็น 2 ประเภท คือ

4.3.1 การสร้างฐานความรู้ใหม่ โดยการโหลดไฟล์จากภายนอก (Load file External)

การสร้างฐานความรู้ใหม่โดยการโหลดไฟล์ข้อมูลที่ผู้ใช้เขียนขึ้นมานั้น มีข้อดีคือผู้ใช้สามารถสร้างกฎได้มากและรวดเร็วกว่าการป้อนโดยผ่านระบบ แต่ว่าผู้ใช้จะต้องมีความรู้เกี่ยวกับไวยากรณ์ของระบบของกฎและความจริงเสียก่อน จากนั้นก็เขียนกฎและความจริงในโปรแกรม Text Editor ทั่วไป และนำมาโหลดเข้าระบบโดยผ่าน ส่วนระบบติดต่อผู้ใช้ส่วนนี้ ระบบจะทำการตรวจสอบไวยากรณ์ของกฎและความจริงให้ถูกต้อง ก่อนที่จะบันทึกลงเป็นฐานความรู้ให้กับระบบต่อไป ซึ่งส่วนนี้จะแบ่งออกเป็นสองส่วน คือ Load Rule File เป็นส่วนที่ใช้ในการโหลดไฟล์กฎที่ผู้ใช้เขียนขึ้น ตามไวยากรณ์ของกฎที่กำหนดไว้ เพื่อให้ระบบตรวจสอบความถูกต้องของกฎ ก่อนที่จะบันทึกเป็นฐานกฎในระบบ Load Fact File เป็นส่วนที่ใช้ในการโหลดไฟล์ความจริงที่ผู้ใช้เขียนขึ้น ตามไวยากรณ์ของความจริงที่กำหนดไว้ เพื่อให้ระบบตรวจสอบความถูกต้องของกฎ ก่อนที่จะบันทึกเป็นฐานกฎในระบบต่อไป

4.3.2 การสร้างฐานความรู้ใหม่โดยการป้อนผ่านระบบ

การสร้างฐานความรู้ใหม่โดยการป้อนฐานความรู้ผ่านระบบจะมีข้อดีคือ ผู้ใช้ไม่จำเป็นต้องมีความรู้เกี่ยวกับไวยากรณ์ของกฎและความจริงของระบบก็สามารถสร้างกฎและความจริงขึ้นมาใช้งานได้ โดยการป้อนข้อมูลตามที่ระบบต้องการผ่านส่วนที่ช่วยสร้างกฎและความจริง และกฎที่ได้จะมีไวยากรณ์ที่ถูกต้อง แต่มีข้อเสียคือ การสร้างกฎแบบนี้จะช้าเมื่อเทียบกับแบบแรก

4.4 การปรับปรุงฐานความรู้ (Improvement of Knowledge Base)

การปรับปรุงฐานความรู้ เป็นการปรับปรุงความรู้ในฐานความรู้เพื่อให้ระบบที่พัฒนาขึ้นนั้นมีความรู้ที่ถูกต้องและทันสมัยอยู่เสมอในการแก้ปัญหา โดยเปลือกระบบผู้เชี่ยวชาญ “GES” ได้เตรียมส่วนที่ใช้ในการปรับปรุงฐานความรู้โดยการใช้งานจากระบบเปลือกที่สามารถปรับปรุงฐานความรู้ต่างๆ ได้ ซึ่งส่วนนี้จะช่วยให้การพัฒนาระบบผู้เชี่ยวชาญให้มีความสะดวกมากขึ้น คือ ผู้พัฒนาระบบสามาระที่จะปรับปรุงความรู้ได้ทันทีในระบบ โดยไม่ต้องสร้างฐานความรู้ใหม่ทั้งระบบ ซึ่งส่วนของการปรับปรุงความรู้นี้สามารถจะปรับปรุงได้ทั้งกฎและความจริงในระบบ

4.4.1 การปรับปรุงฐานกฎ (Improvement of Rule Base)

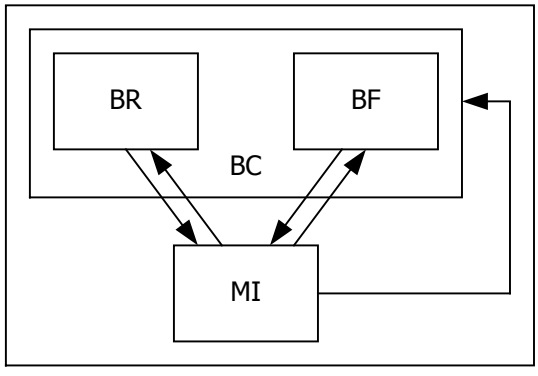
ในการปรับปรุงฐานกฎในฐานความรู้ เราสามารถที่จะเพิ่ม ลบ หรือแก้ไขส่วนไหนของกฎก็ได้ อีกทั้งในการแก้ไขสามารถที่จะแก้ไขกฎในส่วนของเงื่อนไข (Condition) หรือส่วนของข้อสรุป (Conclusion) ส่วนใดส่วนหนึ่งหรือทั้งสองส่วน โดยผู้พัฒนาสามารถที่จะเลือกแก้ไขกฎใดกฎหนึ่งก็ได้

4.4.2 การปรับปรุงฐานความจริง (Improvement of Fact Base)

การปรับปรุงฐานความจริงในฐานความรู้เริ่มต้น เราสามารถที่จะ เพิ่ม ลบ หรือแก้ไขความจริงได้ โดยที่ในผู้พัฒนาสามารที่จะเลือกแก้ไขตรงส่วนไหนของระบบก็ได้ เมื่อผู้พัฒนาระบบปรับปรุงความรู้เสร็จ ระบบจะทำการแก้ไขความสัมพันธ์ระหว่างความจริงกับกฎใหม่ เพื่อให้ถูกต้องตามฐานความรู้ใหม่ที่ถูกปรับปรุงขึ้น

4.5 การอนุมาน (Utilization of System)

เครื่องอนุมานทำหน้าที่เป็นเครื่องมือในการควบคุมระบบและเป็นส่วนที่ใช้ในการหาข้อสรุป เครื่องอนุมานเป็นส่วนที่สำคัญที่สุดส่วนหนึ่งของระบบผู้ชำนาญการสำหรับใช้ในการหาเหตุผลแล้วจึงสรุปออกมาเป็นคำตอบหรือคำแนะนำ ในภาพประกอบ 4-8 แสดงการติดต่อระหว่างเครื่องอนุมานกับฐานความรู้ ระบบเปลือกผู้เชี่ยวชาญ GES สามารถใช้การอนุมานพื้นฐานทั้งการอนุมานแบบเดินหน้า การอนุมานแบบถอยหลัง และการอนุมานแบบผสมได้



ภาพประกอบ 4-8 การติดต่อระหว่างเครื่องอนุมานกับฐานความรู้

4.5.1 การอนุมานแบบเดินหน้า (Forward Chaining Inference)

การอนุมานแบบเดินหน้าของ “GES” นี้จะทำการอนุมานโดยยึดหลักการอนุมานแบบเดินหน้าดังพื้นฐานในบทที่ 3 ซึ่ง “GES” จะเริ่มต้นอนุมานจากความสัมพันธ์ระหว่างความจริงกับกฎที่สร้างขึ้น เพื่อนำไปหาความจริงใหม่ที่ได้จากข้อสรุปภายในกฎที่มีเงื่อนไขที่ถูกต้อง

4.5.2 การอนุมานแบบถอยหลัง (Backward Chaining Inference)

การอนุมานแบบถอยหลังของ GES จะทำการอนุมานโดยจะเริ่มต้นจากเป้าประสงค์ที่ต้องการแล้วนำไปหากฎที่จำเป็นต้องนำมาใช้ในการอนุมาน จากนั้นจะนำกฎเหล่านั้นไปทำการอนุมานแบบเดินหน้าต่อไป เพื่อให้ได้เป้าประสงค์ที่ต้องการ

4.5.3 การอนุมานแบบผสม (Mix Chaining Inference)

การอนุมานแบบผสมของ GES จะทำการอนุมานโดยยึดหลักการอนุมานแบบผสม ซึ่ง GES จะรับเป้าประสงค์จากผู้ใช้ไปทำการอนุมานแบบถอยหลัง ซึ่งหากเป้าประสงค์เป็นจริง ระบบจะนำเส้นทางของการอนุมานแบบถอยหลัง ไปทำการอนุมานแบบเดินหน้า เพื่อหาความจริงใหม่

4.6 ส่วนแสดงฐานความรู้ในระบบ (Show Knowledge Base)

ส่วนแสดงฐานความรู้ในระบบ เป็นส่วนหนึ่งของระบบเปลือกผู้เชี่ยวชาญ GES ที่ผู้ใช้สามารถที่จะเรียกดูว่ากฎ และความจริงที่มีในระบบมีอะไรบ้าง โดยส่วนนี้สามารถที่จะแสดงกฎที่และความจริงที่มีอยู่ในระบบได้ และบางครั้งผู้ใช้สามารถที่จะค้นหาคำที่ต้องการในฐานความรู้ในระบบโดยระบุคำต่างๆ ของกฎเพื่อให้ระบบเปลือกค้นหาให้ก็ได้

4.7 ส่วนอธิบายการกำหนดค่าต่างๆ (Explanation about Utilization)

ในระบบเปลือกผู้เชี่ยวชาญ GES ส่วนอธิบายการกำหนดค่าต่างๆ เป็นส่วนที่ใช้สำหรับการอธิบายการใช้งานโปรแกรมในส่วนต่างๆ เช่นการเขียนกฎ หรือความจริงให้ถูกต้องตามหลักไวยากรณ์ต้องเขียนอย่างไร การเรียกใช้งานส่วนติดต่อกับภายนอกทำอย่างไร ตลอดจนอื่นๆ ในระบบเปลือกที่ผู้ใช้ต้องการคำอธิบาย ซึ่งส่วนนี้เปรียบเสมือน Help ในระบบนั้นเอง