บทที่ 5
การพัฒนาเพื่อใช้งานจริง และ การทดสอบ

5.1 เครื่องมือที่ใช้ในการพัฒนา

ในหัวข้อการค้นคว้าแบบอิสระในครั้งนี้ ใช้การเขียนโปรแกรมด้วยภาษาจาวา ในการพัฒนา ทั้งนี้ด้วยเหตุผลของความต้องการที่จะนำไปใช้ในสภาพแวดล้อมแบบอินเทอร์เน็ต และใช้งานร่วมกับเปลือกระบบผู้เชี่ยวชาญที่พัฒนาโดยใช้ภาษาจาวา (Jess-Java expert system shell)

ความเฉลียวฉลาด รอบรู้ของระบบผู้เชี่ยวชาญฯ ใช้เปลือกระบบผู้เชี่ยวชาญที่พัฒนาโดยใช้ภาษาจาวา เป็นกลไกในการอนุมาน ทำหน้าที่ในการประมวลฐานความรู้ของระบบ การให้คำตอบ คำแนะนำ และการจัดเตรียมคำถามสำหรับผู้ใช้งาน

ไฟล์ psres.clp เป็นไฟล์ที่เก็บฐานความรู้ของระบบ ถูกเขียนให้อยู่ในรูปแบบของ CLIPS script ซึ่งสามารถนำมาใช้กับ Jess ได้

เทคโนโลยีเซิร์ฟเลท ใช้ในการทำให้โปรแกรมที่เขียนโดยภาษาจาวา ทำการคอมไพล์เป็นคลาส ให้สามารถใช้งานได้ในสภาพแวดล้อมแบบอินเทอร์เน็ต ผลที่ได้จากการทำงาน จะถูกแปลงให้อยู่ในรูปของไฟล์ HTML เพื่อไปแสดงที่เว็บเบราเซอร์ของผู้ใช้งาน

5.2 กฎที่ใช้ในระบบ

5.2.1 ตัวอย่างของกฎที่ใช้ในระบบผู้เชี่ยวชาญฯ

ไฟล์ psres.clp เป็นไฟล์ที่ใช้เก็บฐานความรู้ของระบบฯ ประกอบด้วย การกำหนดค่าความจริง และการกำหนดกฎที่ใช้ในการอนุมาน ในตัวอย่างต่อไปนี้ เป็นส่วนหนึ่งของกฎที่อยู่ในฐานความรู้ของระบบผู้เชี่ยวชาญฯ

```
(defrule first-query
        (initial-fact)
        =>
        (new EsMainPanel "What type of Problem?"  5
```

```
                (create$
                        "Antenna" "(equip antenna)"
                        "Transmitter" "(equip tx)"
                        "Receiver" "(equip rx)"
                        "Radar Head Processing" "(equip rhp)"
                        "Display System" "(equip display)" )
                        (fetch "ReteControl") ))

(defrule lru-66-67
        (or  (lru 66)  (lru 67) )
        =>
        (new EsQueryPanel "Open the PLD/AS Cabinet door and \\
                on the EDR with the fail LED lit<br>
                verify if at least one of the two led EN1 EN2 is lit"  2
          (create$
                "OK" "(edr-fail-ok)"
                "NOT OK" "(edr-fail-nok)" )
        (fetch "ReteControl") ))

(defrule edr-fail-ok
        (edr-fail-ok)
        =>
        (new EsAnsPanel "Replace the concerned EDR pcb."
        (fetch "ReteControl")))

(defrule edr-fail-nok
        (edr-fail-nok)
        =>
        (new EsQueryPanel "How many encoder are installed?"  2
          (create$
```

```
        "One" "(one-encoder)"
        "Two" "(two-encoder)" )
    (fetch "ReteControl")))
```

จากตัวอย่าง ข้างต้น ประกอบด้วย คำสั่งของ Jess ดังนี้

1. (defrule) เป็นคำสั่งที่ใช้ในการกำหนดกฎในฐานความรู้ของระบบผู้เชี่ยวชาญฯ

2. (fetch) เป็นคำสั่งที่ใช้ในการเรียกใช้กลไกอนุมานของ Jess เพื่อให้ปฏิบัติงานอย่างใดอย่างหนึ่ง เช่น การสั่งให้ Jess ทำงาน หยุดทำงาน การเพิ่ม-ลบค่าความจริง เป็นต้น ซึ่งแล้วแต่ว่าเซิร์ฟเลทใดที่เรียกใช้

กฎทั้ง 4 กฎ ที่นำมาแสดงให้เห็นนี้ เป็นตัวอย่างรูปแบบของกฎที่เป็นมาตรฐานของระบบผู้เชี่ยวชาญฯ ที่ได้ทำการพัฒนาขึ้น ดังที่เคยกล่าวไว้ในตอนต้นแล้วว่า รูปแบบกฎของ Jess จะเหมือนกับรูปแบบในการเขียนกฎของ CLIPS คือ ประกอบด้วย สองส่วนคือ ส่วน if part หรือ LHS และส่วน then part หรือ RHS ในส่วน LHS จะประกอบด้วย ค่าความจริง (fact pattern) และในส่วน RHS ประกอบด้วย คำสั่งต่าง ๆ (action) กฎจะได้รับการกระตุ้น (trigger) หรือได้รับการเตรียมพร้อมปฏิบัติตามคำสั่ง เมื่อค่าความจริงในส่วน LHS ตรงกับค่าความจริงในหน่วยความจำขณะทำงาน (working memory)

กฎ first-query เป็นกฎที่ได้รับการกระตุ้นให้ทำงานเป็นกฎแรก ทั้งนี้เพราะว่า ส่วน LHS คือ (initial-fact) ค่าความจริง (initial-fact) นี้ จะปรากฏขึ้นในหน่วยจำขณะทำงาน เมื่อมีการเรียกใช้คำสั่ง (reset) ของ Jess ดังนั้น กฎข้อนี้ จะพร้อมที่จะทำงานทันทีที่ได้รับคำสั่งให้กลไกอนุมานของ Jess (Jess-engine) ทำงาน

ส่วนรายการกิจกรรมในส่วน RHS ของกฎข้อนี้ เป็นการสร้างออบเจ็ค EsMainPanel และส่งค่าพารามิเตอร์ต่าง ๆ เพื่อใช้สำหรับออบเจ็คนี้ ได้แก่ ข้อความส่วนหัวที่ปรากฏใน EsMainPanel จำนวนของข้อความ รายการของข้อความที่ตรงตามจำนวนที่กำหนด พร้อมกับค่าความจริงที่ถูกแนบพร้อมไปกับข้อความนั้น และพารามิเตอร์ตัวสุดท้าย คือ การสั่งให้มีการเพิ่มค่าความจริงในหน่วยจำขณะทำงาน

กฎ lru-66-67 ในส่วน LHS คือ (or (lru 66) (lru 67)) ซึ่งหมายความว่ากฎจะได้รับการกระตุ้น เมื่อมีค่าความจริง (lru 66) หรือ (lru 67) อย่างใดอย่างหนึ่งหรือ ทั้งสอง ปรากฏอยู่ในหน่วยความจำขณะทำงาน

ส่วนรายการกิจกรรมในส่วน RHS ของกฎข้อนี้ เป็นการสร้างออบเจ็ค EsQueryPanel และส่งค่าพารามิเตอร์ต่าง ๆ เพื่อใช้สำหรับออบเจ็คนี้ ได้แก่ ข้อความส่วนหัวที่ปรากฏใน EsQueryPanel จำนวนของข้อความ รายการของข้อความที่ตรงตามจำนวนที่กำหนด พร้อมกับค่าความจริงที่ถูกแนบพร้อมไปกับข้อความนั้น และพารามิเตอร์ตัวสุดท้าย คือ การสั่งให้มีการเพิ่มค่าความจริงในหน่วยจำขณะทำงาน (Working Memory)

กฎ edr-fail-ok ในส่วน LHS คือ (edr-fail-ok) ซึ่งหมายความว่ากฎจะได้รับการกระตุ้น เมื่อมีค่าความจริง (edr-fail-ok) ปรากฏอยู่ในหน่วยความจำขณะทำงาน

ส่วนรายการกิจกรรมในส่วน RHS ของกฎข้อนี้ เป็นการสร้างออบเจ็ค EsAnsPanel ซึ่งเป็นออบเจ็คสำหรับแสดงคำแนะนำในการปฏิบัติงานให้กับผู้ใช้งาน เมื่อระบบผู้เชี่ยวชาญฯ ได้รับข้อมูลจากผู้ใช้งานเพียงพอกับความต้องการแล้ว

รูปแบบของกฎที่ใช้ในการเสนอคำตอบ และคำแนะนำจะใช้รูปแบบนี้ทั้งหมด

รายละเอียดของไฟล์ psres.clp ทั้งหมด ซึ่งเป็นไฟล์ฐานความรู้ของระบบผู้เชี่ยวชาญฯ แสดงไว้ในภาคผนวก ข

5.2.2 การเพิ่ม-ลบ และการแก้ไขกฎในฐานความรู้

การเพิ่ม – ลบ และการแก้ไขกฎในฐานความรู้ สามารถทำได้โดยการใช้เท็กเอดิเตอร์ทั่ว ๆ ไป เช่น โปรแกรม NOTEPAD โปรแกรม WRITE หรือแม้กระทั่งโปรแกรม editor ใน MS-DOS mode ก็สามารถทำได้เช่นกัน เนื่องจากระบบผู้เชี่ยวชาญฯที่พัฒนาขึ้นมาใช้งานนี้ ได้ทำการติดตั้งไว้ในระบบอินทราเน็ตของกองวิศวกรรมระบบติดตามอากาศยาน ศูนย์ควบคุมการบินพิษณุโลก ดังนั้น ผู้ที่จะสามารถทำการแก้ไขไฟล์ฐานความรู้ของระบบผู้เชี่ยวชาญฯได้ จะต้องได้รับอนุญาตจากผู้ดูแลระบบอินทราเน็ต หรือได้รับสิทธิ์ให้ทำการแก้ไขไฟล์ฐานความรู้ได้

ในขั้นตอนของการพัฒนาระบบผู้เชี่ยวชาญฯในครั้งนี้ ไฟล์ฐานความรู้ ใช้ชื่อว่า psres.clp ถูกเรียกใช้จากตำแหน่ง "c:\tomcat41\webapps\ROOT\expert" การเพิ่มปัญหาเข้าไปในฐานความรู้ ผู้ที่ทำการเพิ่มเติมจะต้องมีความรู้เข้าใจหลักการทำงานเบื้องต้นของ Jess อยู่บ้าง ซึ่งจะได้อธิบาย โดยใช้ตัวอย่างการเพิ่มปัญหาใหม่ให้กับระบบผู้เชี่ยวชาญฯ พอสังเขป ดังนี้

ในที่นี้ สมมุติว่าผู้ใช้งานต้องการที่จะเพิ่มปัญหาเกี่ยวกับอุปกรณ์ใหม่ให้กับระบบผู้เชี่ยวชาญฯ โดยจะเรียกอุปกรณ์ในส่วนนี้ว่า "New System" สิ่งแรกที่ผู้ใช้งานต้องทำการแก้ไขคือส่วนของ Main Section ในไฟล์ psres.clp

Main Section เดิม คือ

```
;*************************************
;       MAIN SECTION
;*************************************
(defrule first-query
 (initial-fact)
=>
 (new EsMainPanel "What type of Problem?"  5
  (create$
   "Antenna" "(equip antenna)"
   "Transmitter" "(equip tx)"
   "Receiver" "(equip rx)"
   "Radar Head Processing" "(equip rhp)"
   "Display System" "(equip display)"
)
  (fetch "ReteControl"))
)
```

จุดแรกที่ต้องทำการแก้ไข คือ เลข 5 ให้เปลี่ยนเป็นเลข 6 เพื่อให้ออบเจ็ค EsMainPanel รู้ว่าจะต้องแสดงตัวเลือกทั้งหมด 6 ตัวเลือก และเพิ่มข้อความ "New System" "(equip new-system)" ภายในวงเล็บ create$ จะได้ กฎ first-query ใหม่ ดังนี้

```
;*************************************
;       MAIN SECTION
;*************************************
(defrule first-query
 (initial-fact)
=>
 (new EsMainPanel "What type of Problem?"  6
```

```
(create$
 "Antenna" "(equip antenna)"
 "Transmitter" "(equip tx)"
 "Receiver" "(equip rx)"
 "Radar Head Processing" "(equip rhp)"
 "Display System" "(equip display)"
 "New System" "(equip new-system)"
)
 (fetch "ReteControl"))
)
```

จากนั้น ผู้ใช้งานต้องทำการเพิ่มกฎใหม่ให้กับระบบผู้เชี่ยวชาญฯ โดยกฎที่เพิ่มขึ้นจะต้องมีส่วน LHS เป็น (equip new-system) เพื่อให้ระบบผู้เชี่ยวชาญฯเลือกกฎที่เพิ่มขึ้นใหม่นี้ไปปฏิบัติ เช่น

```
(defrule new-system-rule "Add rule example" ;;This is a comment
 (equip new-system)
=>
 (new EsQueryPanel "Hi! How are you?" 2
  (create$
  "Fine, Thanks." "(fine-thank)"  ;; Factcomponent #1
   "Wonderful!" "(wonderful)"    ;; Factcomponent #2
)
  (fetch "ReteControl"))
)
```

เป็นการเพิ่มกฎใหม่ที่มีชื่อกฎว่า new-system-rule ข้อความที่อยู่ในเครื่องหมายคำพูดเป็นการอธิบายว่าเป็นกฎที่ใช้เพื่ออะไร ซึ่งในส่วนนี้จะมีหรือไม่มีก็ได้ ถัดมาเป็นส่วน LHS มี fact pattern เป็น (equip new-system) ดังนั้น กฎ new-system-rule จะได้รับการแอคติเวชัน (activation) เพื่อให้พร้อมทำงานเมื่อมีค่าความจริง (equip new-system) ปรากฏในหน่วยความจำขณะทำงาน และส่วน RHS จะปฏิบัติงานเมื่อมีคำสั่งให้กลไกการอนุมานทำงาน

เครื่องหมาย => ใช้แทนคำว่า then ส่วนที่อยู่หลัง then เรียกว่าเป็น RHS หรืออาจเรียกว่าเป็นส่วน action นั่นคือเมื่อกลไกอนุมานทำงาน ส่วน RHS จะได้รับการนำไปปฏิบัติในทันที ในที่นี้ เป็นการสร้างออบเจคของหน้าคำถาม EsQueryPanel จากตัวอย่างเป็นการสร้าง หน้าคำถามแสดงข้อความ "Hi! How are you?" และมีคำตอบให้เลือก 2 คำตอบ คือ "Fine, Thank." และ "Wonderful!" มีลักษณะเป็นปุ่มกด ทั้งสองปุ่มกดนี้มีค่าความจริงถูกผูกค่าติดไปกับปุ่ม คือ ค่าความจริง "(fine-thank)" และ "(wonderful)" ตามลำดับ ให้สังเกตว่า ค่าความจริงจะต้องมีวงเล็บอยู่ด้วย เพื่อนำไปใช้เป็นค่าความจริงของระบบต่อไป ถ้าเราต้องการให้เกิดการโต้ตอบกับผู้ใช้งาน เมื่อผู้ใช้งานกดปุ่มใดปุ่มหนึ่ง เช่น ถ้าผู้ใช้กดปุ่ม "Fine,Thank." ให้ระบบแสดงคำว่า "Have a nice day." หรือ ถ้าผู้ใช้กด "Wonderful" ให้ระบบแสดงคำว่า "Excellent!" จะต้องเขียนกฎขึ้นมารองรับค่าความจริงทั้งสองอีกสองกฎ ดังนี้

```
(defrule fine-thank-rule
 (fine-thank)
=>
 (new EsAnsPanel "Have a nice day"
  (fetch "ReteControl"))
)

(defrule wonderful-rule
 (wonderful)
=>
 (new EsAnsPanel "Excellent!"
  (fetch "ReteControl"))
)
```

สำหรับการลบกฎก็จะใช้หลักการเดียวกันนี้ คือ แทนที่จะเพิ่ม ก็ให้ทำการลบออก อย่างไรก็ตามการแก้ไขกฎในฐานความรู้ จะต้องกระทำด้วยความรอบคอบ หากว่าเกิดการผิดพลาด เช่น พิมพ์ขาด หรือ เกิน จะทำให้ระบบผู้เชี่ยวชาญไม่ทำงาน หรือทำงานได้ไม่ตรงกับความต้องการ ดังนั้น การเพิ่มหรือลบ กฎในฐานความรู้ของระบบผู้เชี่ยวชาญฯ จึงควรเป็นหน้าที่ของผู้ที่ทำหน้าที่เป็นวิศวกรความรู้ (Knowledge Engineer) หรือ เป็นผู้ที่มีหน้าที่รับผิดชอบโดยตรงในการแก้ไข

ฐานความรู้ นอกจากนี้ การเพิ่มกฎในฐานความรู้ กฎที่เพิ่มเข้าไปจะต้องได้รับการพิสูจน์แล้วว่าสามารถทำงานได้ตรงกันกับแนวทางในการปฏิบัติงานของผู้เชี่ยวชาญที่เป็นมนุษย์

5.3 การสร้างเซิร์พเลทหลักของระบบ

เซิร์ฟเลทหลักของระบบ มีอยู่ 2 ไฟล์ ทำหน้าที่เป็นตัวกลางในการติดต่อระหว่างผู้ใช้งานกับกลไกการอนุมานของระบบผู้เชี่ยวชาญ คือ EsServlet.java และ EsReteControl.java

ไฟล์ EsServlet.java เป็นส่วนของการจัดการการร้องขอจากเว็บเบราเซอร์ ให้เป็นไปตามลำดับเพื่อป้องกันการผิดพลาดของกลไกการอนุมาน เมธอดที่มีการสร้างไว้ในไฟล์นี้ มีดังนี้

public void init(ServletConfig Conf) throws ServletException

public void doPost(HttpServletRequest req, HttpServletResponse res)
throws IOException

private void startNewSession(HttpServletRequest req, HttpServletResponse res) throws
IOException

private void continueSession(HttpServletRequest req, HttpServletResponse res, String action)
throws IOException

รายละเอียดการทำงานของเมธอดภายในคลาส

1. public void init(ServletConfig conf) throws ServletException ทำหน้าที่เริ่มต้นการทำงาน ของเซิร์ฟเลท มีส่วนตรวจสอบข้อผิดพลาด (Exception Handling) โดยอาศัยคลาส ServletException

2. public void doPost(HttpServletRequest req, HttpServletResponse res) throws IOException ทำหน้าที่รับข้อมูลที่ส่งมาจากฟอร์มของไฟล์ HTML ด้วยรูปแบบ POST มีอาร์กิวเมนต์ 2 ตัว คือ HttpServletRequest สำหรับอ้างข้อมูลที่ส่งมาจากผู้ใช้ ด้วยตัวแปร req และ HttpServletResponse สำหรับส่งข้อมูลกลับไปยังผู้ใช้ด้วยตัวแปร res ในคลาสนี้มีส่วนตรวจสอบข้อผิดพลาด โดยอาศัย IOException ในคลาสนี้จะมีตรวจสอบการร้องขอที่รับมาจาก

ผู้ใช้งานด้วยว่าเป็นการร้องขอแบบใด เป็นการร้องขอสำหรับปัญหาใหม่ หรือว่าเป็นการร้องขอของปัญหาเดิมที่เกี่ยวข้องกัน

3. private void startNewSession(HttpServlet req, HttpServletResponse res) throws IOException ทำหน้าที่ในการ รีเซทกลไกการอนุมาน และหน่วยความจำในขณะทำงานใหม่ทั้งหมด เพื่อให้มีความพร้อมสำหรับการทำงานของกลไกการอนุมาน ในกรณีที่การร้องขอนั้นเป็นการเริ่มต้นการแก้ปัญหา

4. private void continueSession(HttpServlet req, HttpServletResponse res) throws IOException ทำหน้าที่เพิ่มค่าความจริงที่รับมาจากผู้ใช้งานและเรียกใช้กลไกอนุมาน เพื่อหากฎที่สอดคล้องกับค่าความจริงที่รับเข้ามาใหม่ ต่อเนื่องกันไปจนกว่ากลไกการอนุมานสามารถหาคำตอบให้กับผู้ใช้งานได้

ไฟล์ EsReteControl.java เป็นไฟล์ที่ทำหน้าที่จัดการเกี่ยวกับการทำงานของกลไกการอนุมานที่อยู่เบื้องหลัง โดยมี EsServlet เป็นผู้สั่งการว่าจะให้ทำงานอะไร เมธอดที่สร้างไว้ในคลาสนี้ มีดังนี้

public EsReteControl(HttpServletResponse res)

public void run()

public void resetRete(ServletOutputStream sos)

public void assertFact(String n_querySignature, String n_label, ServletOutputStream sos)

public boolean getSeqError()

public ServletOutputStream getClient()

public HttpServletResponse getResponse()

public void setAssertions(Hashtable n_a)

public void setQuerySignature(String qs)

public void setSession(HttpServletResponse res)

ในที่นี้ จะขออธิบายการทำงานของเมธอดหลัก แบบคร่าว ๆ เพื่อให้เข้าใจการทำงานของระบบ รายละเอียดของโปรแกรมทั้งหมด สามารถศึกษาได้จาก ภาคผนวก ก

1. public ReteControl(HttpServletResponse res) เป็นเมธอดที่จะเริ่มต้นทำงานทันทีที่มีการเรียกใช้งานหรือสร้างอินแสตนซ์ของคลาส EsReteControl ภายในเมธอดนี้จะเป็นการสร้างกลไกการอนุมานสำหรับใช้ในคลาสนี้ทั้งหมด ดังจะได้กล่าวต่อไป

2. public void run() ทำหน้าที่เป็นคำสั่งที่สั่งให้กลไกการอนุมาน Jess ทำงาน โดยภายในเมธอดนี้ จะมีการตรวจสอบด้วยว่าเป็นการเรียกใช้ครั้งแรก การสั่งให้ Jess ทำงาน การเพิ่มค่าความจริงให้กับ Jess หรือ การสั่งรีเซท Jess ถ้าเป็นการเรียกใช้งานครั้งแรก จะมีการอ่านไฟล์ฐานความรู้ของระบบผู้เชี่ยวชาญฯ เข้ามาเก็บในหน่วยความจำของระบบ ซึ่งในหน่วยความจำจะมีค่าความจริง และกฎต่าง ๆ ของระบบผู้เชี่ยวชาญฯ เพื่อเป็นการเตรียมพร้อมปฏิบัติงาน

3. public void resetRete(ServletOutputStream sos) เป็นเมธอดที่ใช้ในการเซ็ทแฟลก (boolean Flag) เพื่อบอกกับ ReteControl ให้รีเซทกลไกการอนุมานของ Jess ให้เริ่มต้นทำงานใหม่

4. public void assertFact(String n_querySignature, String n_label, ServletOutputStream sos) เป็นเมธอดที่ใช้ในการเซ็ทแฟลก (boolean Flag) เพื่อบอกกับ ReteControl ให้เพิ่มค่าความจริงในหน่วยความจำขณะทำงานของ Jess

5.4 การสร้างเซิร์ฟเลทที่ใช้ในการติดต่อกับผู้ใช้งาน

เซิร์ฟเลท ที่ใช้ในการติดต่อกับผู้ใช้งาน มีอยู่ 4 ไฟล์ คือ

1. EsMainPanel.java ทำหน้าที่สร้างไฟล์ HTML ที่ใช้ในการแสดงหน้าจอหลักของระบบผู้เชี่ยวชาญฯ มีปุ่มกดให้ผู้ใช้เลือกกด แต่ละปุ่มแทนการเลือกรายละเอียดของอุปกรณ์แต่ละส่วน เมธอดในคลาสนี้ มี 2 เมธอด คือ

public EsMainPanel(String query, int items, String[] factComponents, EsReteControl reteCon)

public void sendPanel(EsReteControl reteCon)

เมธอด EsMainPanel รับค่าพารามิเตอร์ 4 ค่า คือ

-query เป็นข้อความที่แสดงที่ MainPanel

- items เป็นจำนวนของข้อความและค่าความจริงที่ใช้สำหรับ MainPanel

-factComponents เป็นค่าตัวแปรของข้อความที่ต้องการให้ปรากฏเป็นตัวเลือกพร้อมกับค่าความจริงที่จะผูกติดไปกับตัวเลือกนั้น

-reteCon เป็นออบเจคของ EsReteControl

ส่วนที่สร้างไฟล์ HTML คือ เมธอด sendPanel ซึ่งจะสร้างไฟล์ HTML ที่แสดงหน้าจอหลักของระบบผู้เชี่ยวชาญฯ

2. EsSelectPanel.java ทำหน้าที่สร้างไฟล์ HTML ที่ใช้ในการแสดงหน้าจอที่เป็นลักษณะของการแจกแจงรายการ (drop-down lists) ให้ผู้ใช้งานเลือกรายการที่ต้องการ และกดปุ่ม submit เมธอดในคลาสนี้ มี 2 เมธอด คือ

public EsSelectPanel(String query, int items, String[] factComponents, EsReteControl reteCon)

public void sendPanel(EsReteControl reteCon)

เมธอด EsSelectPanel รับค่าพารามิเตอร์ 4 ค่า คือ

- query เป็นข้อความที่แสดงที่ SelectPanel
- items เป็นจำนวนของข้อความและค่าความจริงที่ใช้สำหรับ SelectPanel
- factComponents เป็นค่าตัวแปรของข้อความที่ต้องการให้ปรากฏเป็นตัวเลือกพร้อมกับค่าความจริงที่จะผูกติดไปกับตัวเลือกนั้น
- reteCon เป็นออบเจคของ EsReteControl

ส่วนที่สร้างไฟล์ HTML คือ เมธอด sendPanel ซึ่งจะสร้างไฟล์ HTML ที่แสดงรายการตัวเลือกแบบแจกแจง ให้ผู้ใช้งานเลือกเพียงรายการเดียว

3. EsQueryPanel.java ทำหน้าที่สร้างไฟล์ HTML ที่ใช้ในการแสดงหน้าจอหลักของระบบผู้เชี่ยวชาญฯ มีปุ่มกดให้ผู้ใช้เลือกกดแต่ละปุ่มแทนการเลือกคำตอบที่เกี่ยวข้องกับคำถามนั้น เมธอดในคลาสนี้ มี 2 เมธอด คือ

public EsQueryPanel(String query, int items, String[] factComponents, EsReteControl reteCon)

public void sendPanel(EsReteControl reteCon)

เมธอด EsQueryPanel รับค่าพารามิเตอร์ 4 ค่า คือ

- query เป็นข้อความที่แสดงที่ QueryPanel
- items เป็นจำนวนของค่าความจริงที่ใช้สำหรับ QueryPanel
- factComponents เป็นค่าตัวแปรของข้อความที่ต้องการให้ปรากฏเป็นคำตอบพร้อมกับค่าความจริงที่จะผูกติดไปกับคำตอบนั้น
- reteCon เป็นออบเจคของ EsReteControl

ส่วนที่สร้างไฟล์ HTML คือ เมธอด sendPanel ซึ่งจะสร้างไฟล์ HTML ที่แสดงคำถามและคำตอบให้ผู้ใช้งานเลือกตอบเพียงคำตอบเดียว

4. EsAnsPanel. java ทำหน้าที่สร้างไฟล์ HTML ที่ใช้ในการแสดงหน้าจอหลักของระบบผู้เชี่ยวชาญฯ มีปุ่มกดให้ผู้ใช้เลือกกด แต่ละปุ่มแทนการเลือกรายละเอียดของอุปกรณ์แต่ละส่วน เมธอดในคลาสนี้ มี 2 เมธอด คือ

public EsAnsPanel(String query, EsReteControl reteCon)

public void sendPanel(EsReteControl reteCon)

เมธอด EsAnsPanel รับค่าพารามิเตอร์ 2 ค่า คือ

- query เป็นข้อความที่แสดงที่ AnsPanel
- reteCon เป็นออบเจคของ EsReteControl

ส่วนที่สร้างไฟล์ HTML คือ เมธอด sendPanel ซึ่งจะสร้างไฟล์ HTML ที่แสดงคำแนะนำที่ได้จากระบบ เพื่อให้ผู้ใช้งานปฏิบัติตาม

5.5 การสร้างไฟล์ประกอบอื่น ๆ

5.5.1 startes.html เป็นไฟล์ที่ทำหน้าที่นำผู้ใช้งานเข้าสู่การทำงานของระบบผู้เชี่ยวชาญ ในไฟล์นี้จะมีการส่งข้อมูลโดยใช้เว็บเบราเซอร์ ผ่านทางแท็กฟอร์ม ใช้รูปแบบการส่งแบบ POST โดยกำหนดแท็กฟอร์มให้มีแอททริบิวต์ method เป็น POST

```
<form name="startup" method="POST" action="/servlet/EsServlet">
    <input type="hidden" name="action" value="start">
    <input type="submit" value="Start Expert System" name="submit">
</form>
```

เมื่อผู้ใช้งานกดปุ่ม "Start Expert System" ที่เว็บเบราเซอร์ ที่เว็บเซิร์ฟเวอร์จะมีการเรียกใช้งานเซิร์ฟเลท EsServlet โดยค่าตัวแปรที่ส่งให้เซิร์ฟเลท คือ "start" ผ่านทาง input type แบบ hidden เพื่อสั่งให้ระบบผู้เชี่ยวชาญเริ่มทำงาน

5.5.2 assist.html ทำหน้าที่ในการอธิบายวิธีการในการหาค่ารหัสของการวินิจฉัยเบื้องต้นที่ได้จากส่วนควบคุมการทำงานของระบบ เรียกว่าเป็น LRU code (Line Replacement Unit Code) เพื่อที่ผู้ใช้งานจะสามารถนำค่าที่ได้จากส่วนควบคุมฯ มาใช้ในการหาคำแนะนำวิธีการในการแก้ปัญหาจากระบบผู้เชี่ยวชาญฯ ต่อไป

5.6 การทดสอบการใช้งาน และผลการทดสอบ

การทดสอบการทำงานของระบบ แบ่งการทดสอบออกเป็น 2 ส่วน คือ การทดสอบความถูกต้องของคำแนะนำที่ได้จากระบบผู้เชี่ยวชาญฯ ในทุกปัญหาที่มีอยู่ในฐานความรู้ และการเปรียบเทียบระยะเวลาที่ใช้ในการปฏิบัติงานของผู้ใช้งาน เมื่อเทียบกับเวลาที่ใช้ในการปฏิบัติงานของผู้เชี่ยวชาญที่เป็นมนุษย์ เมื่อทำการแก้ไขข้อขัดข้องแบบเดียวกัน

5.6.1 การทดสอบความถูกต้องของคำแนะนำที่ได้จากระบบผู้เชี่ยวชาญฯ

การทดสอบความถูกต้องของคำแนะนำที่ได้จากระบบผู้เชี่ยวชาญฯ ทำการทดสอบโดยผู้เชี่ยวชาญที่เป็นมนุษย์ ที่มีความเชี่ยวชาญในแต่ละระบบย่อย ของเรดาร์ปฐมภูมิ รุ่น ATCR-33S ผู้ทดสอบได้ทำการทดสอบฐานความรู้ของระบบผู้เชี่ยวชาญฯ โดยตรวจสอบวิธีการในการแก้ไขปัญหาทีละปัญหาจนครบ อ้างอิงผลลัพธ์ที่ได้กับเอกสารคู่มือทางเทคนิคของเรดาร์ปฐมภูมิ และเอกสารบันทึกการแก้ไขข้อขัดข้องของระบบเรดาร์ปฐมภูมิ กองวิศวกรรมระบบติดตามอากาศยาน

ศูนย์ควบคุมการบินพิบูลโลก ทั้งนี้ เพื่อเป็นการรับประกันว่า คำแนะนำที่ผู้ใช้งานได้รับจากระบบผู้เชี่ยวชาญฯ เป็นคำแนะนำที่ถูกต้อง และ เชื่อถือได้

1) ฐานความรู้ที่ทำการทดสอบ : สายอากาศ เครื่องส่ง และเครื่องรับ
จำนวนปัญหาที่ทดสอบ : สายอากาศ 5 ปัญหา
เครื่องรับ 65 ปัญหา
เครื่องส่ง 6 ปัญหา
ผู้ทดสอบ : นายคมสัน สุระจิตร์ ผู้เชี่ยวชาญระบบเครื่องรับ-ส่ง และสายอากาศ
ผลการทดสอบ : ถูกต้อง ตรงตามข้อมูลในเอกสารทุกปัญหา

2) ฐานความรู้ที่ทำการทดสอบ : ระบบประมวลผลข้อมูล และ การแสดงผล
จำนวนปัญหาที่ทำการทดสอบ : ระบบประมวลผลข้อมูล 4 ปัญหา
ระบบการแสดงผล 3 ปัญหา
ผู้ทดสอบ : นายวิสุทธิ์ ศรีสมบูรณ์ ผู้เชี่ยวชาญระบบประมวลผล และระบบการแสดงผล
ผลการทดสอบ : ถูกต้อง ตรงตามข้อมูลในเอกสารทุกปัญหา

5.6.2 การทดสอบความเร็วในการปฏิบัติงาน

จากการทดสอบการทำงานของระบบผู้เชี่ยวชาญฯ โดยการสมมุติเหตุการณ์ว่า เกิดการขัดข้องกับระบบเรดาร์ ทำให้ไม่สามารถให้แสดงเป้าวัตถุเคลื่อนที่บนจอเรดาร์ให้กับเจ้าหน้าที่ควบคุมจราจรทางอากาศได้ตามปกติ สาเหตุของการขัดข้องเกิดจากอุปกรณ์ในส่วนของระบบสายอากาศ การทดสอบการใช้งานจริง จะใช้วิธีการเปรียบเทียบการปฏิบัติงานของผู้เชี่ยวชาญ กับการปฏิบัติงานของผู้ใช้งานทั่วไป โดยใช้ระบบผู้เชี่ยวชาญฯ ช่วยให้คำแนะนำในการปฏิบัติงานแล้วใช้การจับเวลาที่ใช้ในการปฏิบัติงาน จนกระทั่งพบสาเหตุของการขัดข้องที่แท้จริงที่ผู้ใช้งานสามารถทำการแก้ไขได้ สำหรับตัวอย่างวิธีการหาสาเหตุข้อขัดข้องโดยใช้ระบบผู้เชี่ยวชาญฯ อย่างละเอียดสำหรับการแก้ไขข้อขัดข้องนี้ แสดงไว้ในภาคผนวก ง

<u>การแก้ไขข้อขัดข้อง ของผู้เชี่ยวชาญที่เป็นมนุษย์</u>

วัน/เวลา ที่ทำการทดสอบ : *28 เมษายน 2546 เวลา 14.00 น.*

ผู้ปฏิบัติงาน : *นายคมสัน สุระจิตร์ ผู้เชี่ยวชาญเครื่องรับ-ส่งเรดาร์*

ขั้นตอนการปฏิบัติงาน

1. เกิด alarm ที่จอควบคุมการทำงานของระบบเรดาร์ ในส่วนของสายอากาศ
2. ผู้ปฏิบัติงานตรวจสอบ alarm code ที่ปรากฏบนจอควบคุมฯ
3. ผู้ปฏิบัติงานเปิดคู่มือทางเทคนิคของระบบสายอากาศเพื่อหาจุดทดสอบสัญญาณในอุปกรณ์ที่เกี่ยวข้อง
4. วัดสัญญาณ ที่จุดทดสอบ
5. ผู้ปฏิบัติงานตรวจสอบพบว่า ข้อขัดข้องเกิดจากสายสัญญาณที่ต่อจากเครื่องวัดมุมบนสายอากาศมายังอุปกรณ์ด้านล่าง

วัน/เวลา สิ้นสุดการทดสอบ : *28 เมษายน 2546 เวลา 14.20 น.*

รวมเวลาที่ใช้ในการปฏิบัติงาน **20** นาที

การแก้ไขข้อขัดข้อง ของผู้ใช้งานทั่วไปโดยอาศัยคำแนะนำจากระบบผู้เชี่ยวชาญฯ

วัน/เวลา ที่ทำการทดสอบ : *28 เมษายน 2546 เวลา 15.00 น.*

ผู้ปฏิบัติงาน : *นายสุเทพ อินทรเทพ ช่างเรดาร์ (ผู้ใช้งาน)*

ขั้นตอนการปฏิบัติงาน

1. เกิด alarm ที่จอควบคุมการทำงานของระบบเรดาร์ ในส่วนของสายอากาศ
2. ผู้ปฏิบัติงานตรวจสอบ alarm code ที่ปรากฏบนจอควบคุมฯ
3. ผู้ปฏิบัติงานเลือกปัญหาในส่วนของสายอากาศของระบบผู้เชี่ยวชาญฯ เลือก alarm code ที่ตรงกับ alarm ที่ปรากฏบนจอควบคุม
4. ปฏิบัติตามคำแนะนำในการตรวจสอบที่ปรากฏบนเว็บเบราเซอร์ และตอบคำถาม
5. ผู้ปฏิบัติงานตรวจสอบพบว่า ข้อขัดข้องเกิดจากสายสัญญาณที่ต่อจากเครื่องวัดมุมบนสายอากาศมายังอุปกรณ์ด้านล่าง

วัน/เวลา สิ้นสุดการทดสอบ : *28 เมษายน 2546 เวลา 15.27 น.*

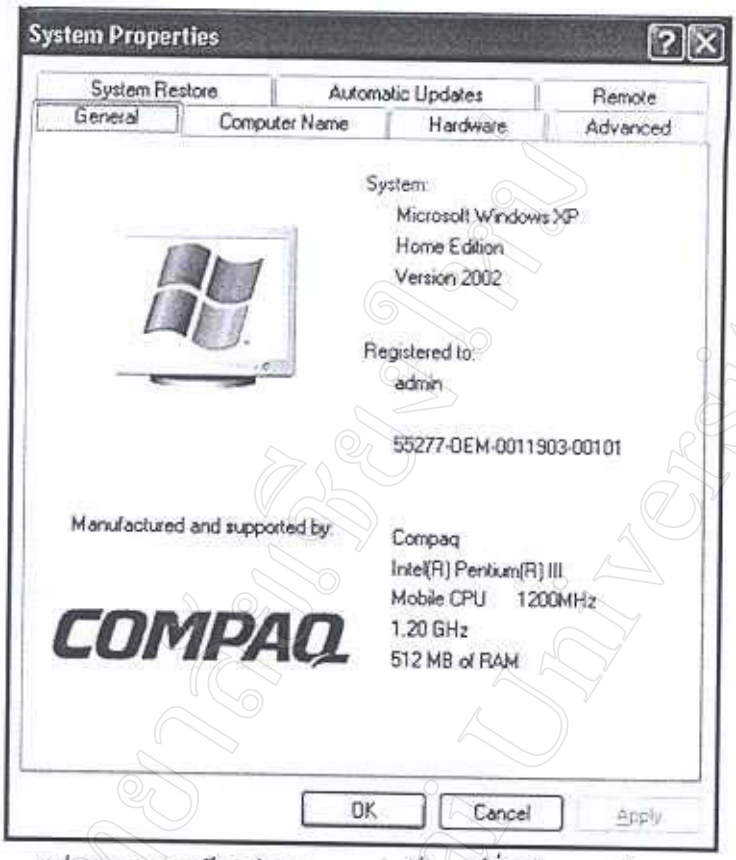
รวมเวลาที่ใช้ในการปฏิบัติ **27** นาที

ผลการทดสอบพบว่า การแก้ไขปัญหาข้อขัดข้องโดยใช้ระบบผู้เชี่ยวชาญฯ ที่ได้พัฒนาขึ้นนี้ เพื่อแก้ไขปัญหาข้อขัดข้องของระบบสายอากาศ ระบบผู้เชี่ยวชาญฯ สามารถช่วยให้ผู้ปฏิบัติงานสามารถทำการแก้ไขข้อขัดข้อง โดยใช้ระยะเวลาปฏิบัติงานใกล้เคียงกับการปฏิบัติงานของผู้เชี่ยวชาญที่เป็นมนุษย์ แสดงให้เห็นว่า ระบบผู้เชี่ยวชาญฯที่ได้พัฒนาขึ้นนี้สามารถช่วยให้คำแนะนำให้กับผู้ปฏิบัติงานได้ โดยไม่ต้องอาศัยความช่วยเหลือจากผู้เชี่ยวชาญที่เป็นมนุษย์ แต่ทั้งนี้ ผู้ใช้งานจะต้องมีความรู้พื้นฐานเกี่ยวกับระบบเรดาร์ เพื่อที่จะสามารถเข้าใจศัพท์ทางเทคนิคที่ระบบผู้เชี่ยวชาญฯ แสดงไว้ในคำแนะนำได้ อย่างไรก็ตาม ประโยชน์ที่ได้รับจากระบบผู้เชี่ยวชาญฯ อีกส่วนหนึ่งได้แก่ ความชำนาญที่ผู้ปฏิบัติงานจะได้รับ ซึ่งคล้ายกับเป็นการฝึกทักษะในการปฏิบัติงานให้กับผู้ปฏิบัติงาน และเป็นการทบทวนความรู้ที่มีอยู่เดิมของผู้ใช้งานได้อีกด้วย

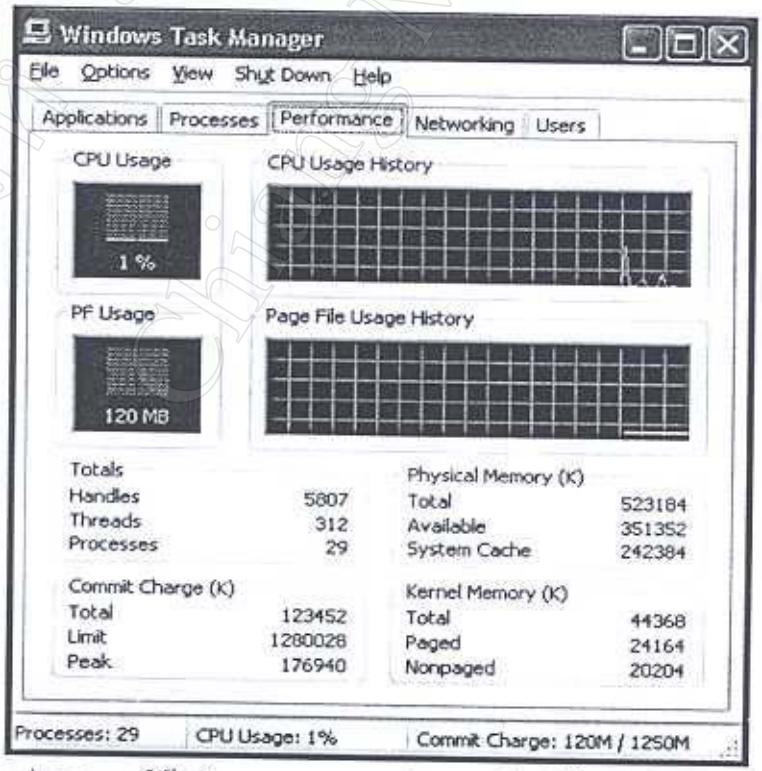
5.7 ความสิ้นเปลืองทรัพยากรของระบบในการทำงาน

การตรวจสอบความสิ้นเปลืองทรัพยากรของระบบฮาร์ดแวร์ ในการทำงานของระบบผู้เชี่ยวชาญฯ โดยการใช้โปรแกรม Windows Task Manager ตรวจดูการใช้งานหน่วยความจำ และการใช้งานหน่วยประมวลผล (CPU) ของอุปกรณ์ฮาร์ดแวร์ที่มีรายละเอียดจำเพาะ ดังแสดงให้เห็นในรูป 5.1 พบว่า การใช้งานหน่วยความจำ และการใช้งานหน่วยประมวลผลทั้งก่อนการเรียกใช้ระบบผู้เชี่ยวชาญฯ และระหว่างการเรียกใช้งานที่ต้องมีการนำค่าความจริงและกฎทั้งหมดของระบบผู้เชี่ยวชาญฯไว้ในหน่วยความจำ มีความแตกต่างกันเห็นได้ชัดในส่วนของหน่วยความจำทางกายภาพ (Physical Memory) และการใช้ Page File ดังรูป 5.2 และ รูป 5.3 การใช้ทรัพยากรของระบบส่วนใหญ่จะถูกใช้โดยทอมแคทเซิร์ฟเวอร์ เพื่อให้บริการข้อมูลกับผู้ใช้งานในเครือข่าย และอีกส่วนหนึ่ง จะถูกใช้โดย Java Virtual Machine เพื่อให้เซิร์ฟเลทสามารถทำงานได้ตามความต้องการ รูป 5.4 แสดงโปรเซสต่าง ๆ ในขณะเรียกใช้ระบบผู้เชี่ยวชาญฯ

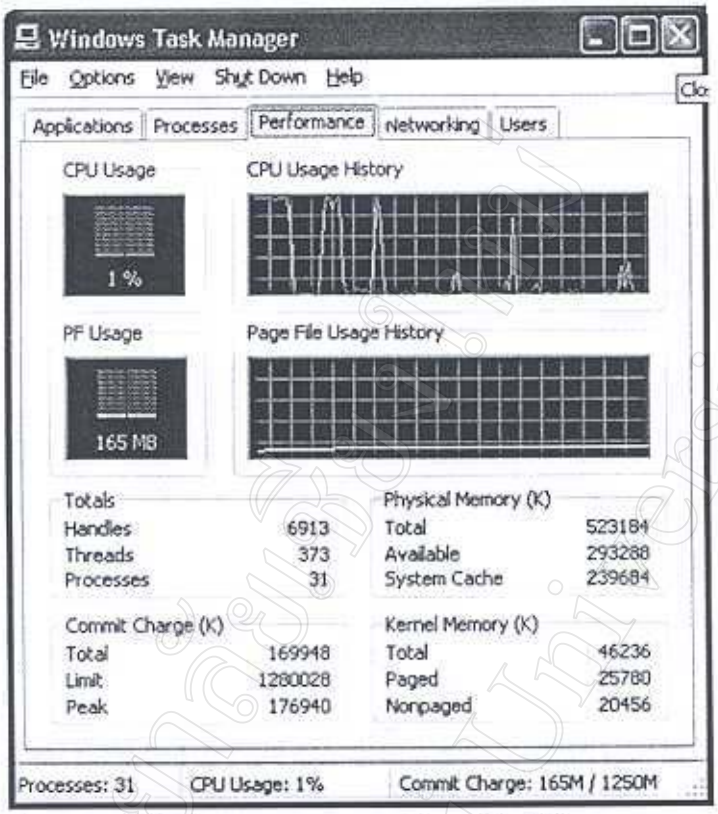
อย่างไรก็ตาม ความสิ้นเปลืองทรัพยากรของระบบ ยังถือว่าอยู่ในระดับที่สามารถยอมรับได้ เพื่อแลกกับประโยชน์ที่ได้รับ แต่ทั้งนี้ อุปกรณ์คอมพิวเตอร์ที่นำมาใช้ จะต้องมีหน่วยประมวลผลที่มีประสิทธิภาพสูง และมีหน่วยความจำในระบบที่มากพอสมควร เพื่อช่วยให้การทำงานของระบบเป็นไปอย่างรวดเร็ว และไม่สร้างความอึดอัดให้กับผู้ใช้งาน



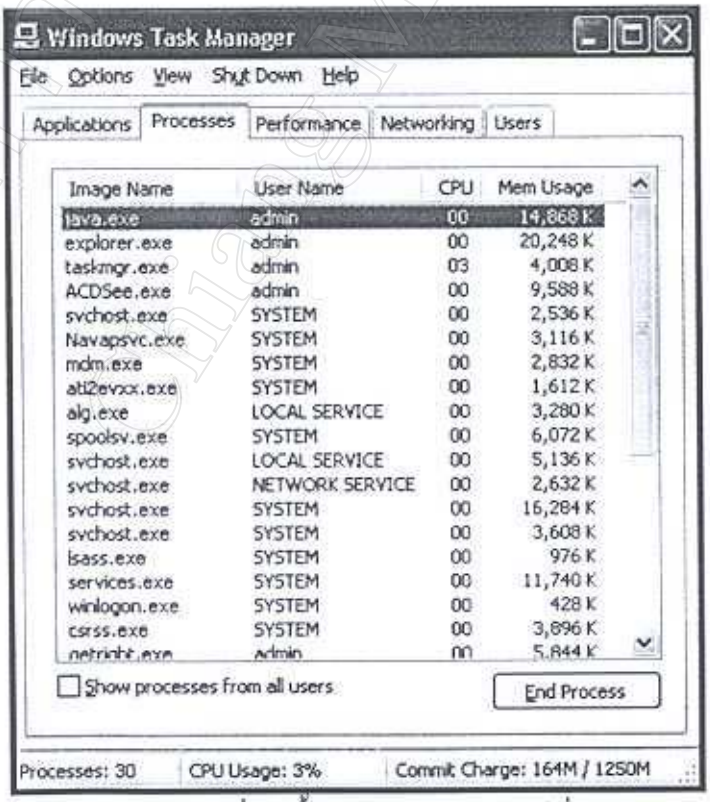
รูป 5.1 รายละเอียดจำเพาะของฮาร์ดแวร์ที่ใช้ในการพัฒนา



รูป 5.2 การใช้ทรัพยากรของระบบก่อนการเรียกใช้งานทอมแคทฯ



รูป 5.3 การใช้ทรัพยากรของระบบหลังการเรียกใช้งานทอมแคทฯ



รูป 5.4 โปรเซสที่เกิดขึ้นขณะเรียกใช้ระบบผู้เชี่ยวชาญฯ