

บทที่ 4

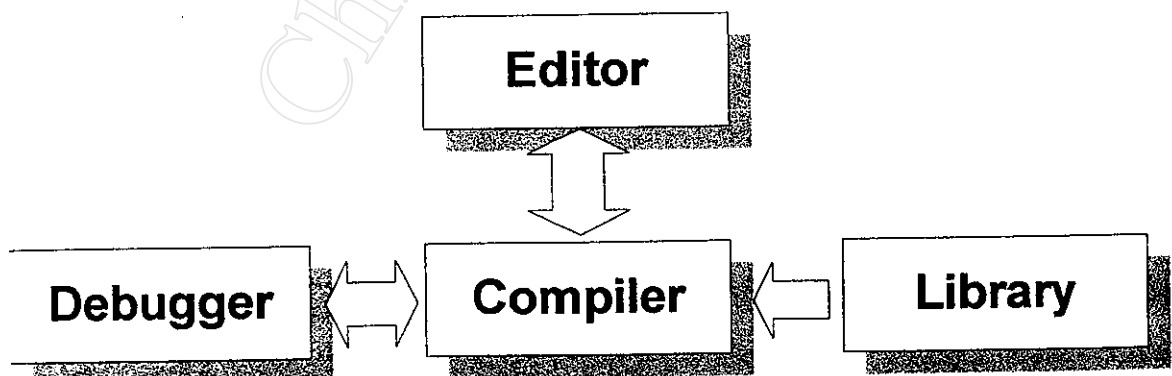
การออกแบบและพัฒนาเปลือกระบบผู้เชี่ยวชาญแบบพีซี

ในงานวิจัยนี้สามารถแบ่งงานออกแบบและพัฒนาระบบ ได้เป็น 2 ส่วนใหญ่ ๆ คือ ส่วนของการพัฒนาเปลือกระบบผู้เชี่ยวชาญแบบพีซี และส่วนของการพัฒนาระบบผู้เชี่ยวชาญแบบพีซีเพื่อการวินิจฉัยโรคทางคลินิก ที่ใช้แสดงตัวอย่างการประยุกต์ใช้งานเปลือกระบบผู้เชี่ยวชาญแบบพีซีที่พัฒนาขึ้น

4.1 การออกแบบเปลือกระบบผู้เชี่ยวชาญแบบพีซี

เปลือกระบบผู้เชี่ยวชาญแบบพีซี คือเครื่องมือที่ใช้ช่วยอำนวยความสะดวกให้กับวิศวกรความรู้ในการนำไปใช้พัฒนาเป็นระบบผู้เชี่ยวชาญแบบพีซี ดังนั้นเปลือกระบบผู้เชี่ยวชาญที่ดีจึงจำเป็นต้องมีเครื่องมืออำนวยความสะดวกต่าง ๆ ให้มากพอและครบถ้วน เพื่อให้วิศวกรความรู้เลือกเครื่องมือเหล่านี้ไปใช้งานได้ตามที่ต้องการ

โครงสร้างของเปลือกระบบผู้เชี่ยวชาญแบบพีซีที่จะพัฒนาขึ้นนี้ จะประกอบไปด้วยส่วนสำคัญ 4 ส่วนด้วยกันดังรูปที่ 4.1 คือ



รูปที่ 4.1 แสดงโครงสร้างของเปลือกระบบผู้เชี่ยวชาญ

4.1.1 ส่วนบรรณาธิการ

ในส่วนบรรณาธิการ (Editor) นี้จะเป็นส่วนที่ใช้สำหรับสร้างและแก้ไขโปรแกรมระบบผู้เชี่ยวชาญแบบพีซี โดยมีคุณลักษณะคล้ายกับโปรแกรมประมวลคำ (Word Processor) ซึ่งจะอำนวยความสะดวกในการเขียนโปรแกรม คัดลอกหรือเพิ่มเติม แก้ไขโปรแกรมข้อมูลต่าง ๆ ลงไปในส่วนบรรณาธิการนี้ได้ และนอกจากนั้นยังประกอบไปด้วยสิ่งอำนวยความสะดวกอื่น ๆ อีก เช่น การจัดเก็บข้อมูล การเรียกโปรแกรมที่เก็บไว้ขึ้นมาแก้ไข เป็นต้น สำหรับส่วนบรรณาธิการนี้ เราจะใช้ตัวบรรณาธิการของโปรแกรมไมโครซอฟต์วิซวลชี

4.1.2 ส่วนแปลโปรแกรม

ในส่วนของการแปลโปรแกรม (Compiler) นี้ จะถูกใช้เพื่อทำการแปลคำสั่งของโปรแกรมที่เราเขียนขึ้นมาด้วยภาษาระดับสูง (High Level Language) ใด ๆ ให้เป็นภาษาเครื่อง (Machine Language) เพื่อให้เครื่องคอมพิวเตอร์สามารถทำงานตามโปรแกรมที่เราพัฒนาขึ้นได้ ซึ่งงานหลัก ๆ ของส่วนแปลโปรแกรมนี้จะมีอยู่ 2 ส่วนด้วยกันคือ ส่วนวิเคราะห์รหัสต้นกำเนิด (Source Code) ซึ่งจะทำหน้าที่ในการตรวจสอบการใช้คำสั่ง รวมไปถึงรูปแบบของคำสั่งว่าถูกต้องตามวากยสัมพันธ์ของโปรแกรมหรือไม่ และส่วนที่สองคือส่วนสังเคราะห์ ซึ่งจะทำการแปลรหัสต้นกำเนิดให้เป็นโปรแกรมประสมค์ (Object Program) เพื่อนำไปแปลงเป็นภาษาเครื่องต่อไป สำหรับส่วนของการแปลโปรแกรมนี้ เราจะใช้ตัวแปลโปรแกรมของไมโครซอฟต์วิซวลชี

4.1.3 ส่วนแก้ไขจุดบกพร่อง

ในส่วนของการแก้ไขจุดบกพร่อง (Debugger) นี้ จะถูกใช้เมื่อวิศวกรความรู้ที่สร้างระบบผู้เชี่ยวชาญแบบพีซีพบว่าโปรแกรมที่สร้างขึ้นมาไม่ถูกต้อง เช่น อาจจะทำงานไม่ได้ตามที่วางเป้าหมายไว้ หรือคำตอบที่ได้ไม่ครบถ้วนบริบูรณ์ โดยวิศวกรความรู้จะใช้ส่วนนี้เพื่อทำการค้นหาความผิดพลาดของโปรแกรมที่เกิดขึ้น สำหรับส่วนแก้ไขจุดบกพร่องนี้ เราจะใช้ของไมโครซอฟต์วิซวลชี

4.1.4 ส่วนไลบรารี

ในส่วนของไลบรารี (Library) นี้จะเป็นส่วนที่ใช้เก็บชุดคำสั่งสำหรับทำงานย่อยต่าง ๆ ในการพัฒนาระบบผู้เชี่ยวชาญแบบฟัซซี่ ซึ่งผู้ที่พัฒนาเปลี่ยนระบบผู้เชี่ยวชาญจะต้องจัดเตรียมคำสั่งต่าง ๆ ไว้อย่างเพียงพอและครบถ้วน เพื่ออำนวยความสะดวกให้กับผู้ที่พัฒนาระบบผู้เชี่ยวชาญให้สามารถจัดการกับค่าของตัวแปรในลักษณะที่เป็นภาษามนุษย์ได้ และมีระบบการตัดสินใจโดยใช้ค่าตัวแปรที่เป็นภาษามนุษย์ร่วมกับตัวแปรที่เป็นตัวเลขได้ ซึ่งจะให้ผลของการตัดสินใจที่ถูกต้องมากยิ่งขึ้น สำหรับในส่วนไลบรารีนี้เราจะทำการเพิ่มเติมชุดคำสั่งที่จะช่วยอำนวยความสะดวกให้กับวิศวกรความรู้ในการพัฒนาระบบผู้เชี่ยวชาญแบบฟัซซี่ โดยออกแบบให้มีชุดคำสั่งทั้งหมด 40 คำสั่ง แยกเก็บไว้ตามคลาสต่าง ๆ ได้ 7 คลาสด้วยกันคือ

- คลาสการให้เหตุผลแบบฟัซซี่ (Fuzzy Reasoning Class) ในคลาสนี้จะประกอบไปด้วยฟังก์ชันสมาชิก เพื่อใช้สำหรับการให้เหตุผลกับความรู้ในแบบฟัซซี่ ซึ่งจะประกอบไปด้วยฟังก์ชันดังนี้

(1) ฟังก์ชันของฟัซซี่ลอจิก

CFuzlogic::setval(float& h, float& l)

เป็นฟังก์ชันที่ใช้สำหรับเซตค่าข้อมูลให้กับตัวแปรในคลาส

โดยที่

h : หมายถึงตัวแปรทางด้านค่าสูง (high value)

l : หมายถึงตัวแปรทางด้านค่าต่ำ (low value)

CFuzlogic::getlowval()

เป็นฟังก์ชันที่ใช้สำหรับส่งค่าข้อมูลทางด้านค่าต่ำออกมา

CFuzlogic::gethighval()

เป็นฟังก์ชันที่ใช้สำหรับส่งค่าข้อมูลทางด้านค่าสูงออกมา

CFuzlogic::getshare(const float& input)

เป็นฟังก์ชันที่ใช้สำหรับให้ค่าตัวเลขที่สัมพันธ์กับสมาชิกในแบบฟัซซี่ลอจิก

โดยที่

input : ค่าตัวเลขที่ป้อนเข้ามาเพื่อหาค่าสมาชิก

(2) ฟังก์ชันของฟัซซีเซต

CFuzset::setname(char*n)

เป็นฟังก์ชันสำหรับป้อนชื่อหมวดหมู่ให้กับข้อมูลแบบฟัซซีเซต

CFuzset::getname()

เป็นฟังก์ชันสำหรับส่งค่าข้อมูลที่เป็นชื่อของหมวดหมู่ออกมา

CFuzset::setval(float& h , float& m , float& l)

เป็นฟังก์ชันที่ใช้สำหรับเซตค่าข้อมูลให้กับตัวแปรในคลาส

โดยที่

h : หมายถึงตัวแปรทางด้านค่าสูง (high value)

m : หมายถึงตัวแปรในช่วงกึ่งกลาง (medium value)

l : หมายถึงตัวแปรทางด้านค่าต่ำ (low value)

CFuzset::getlowval()

เป็นฟังก์ชันที่ใช้สำหรับส่งค่าข้อมูลทางด้านค่าต่ำออกมา

CFuzset::getmidval()

เป็นฟังก์ชันที่ใช้สำหรับส่งค่าข้อมูลทางด้านค่ากลางออกมา

CFuzset::gethighval()

เป็นฟังก์ชันที่ใช้สำหรับส่งค่าข้อมูลทางด้านค่าสูงออกมา

CFuzset::getshare(const float& input)

เป็นฟังก์ชันที่ใช้สำหรับให้ค่าตัวเลขที่สัมพันธ์กับสมาชิกในแบบฟัซซีเซต

โดยที่

input : ค่าตัวเลขที่ป้อนเข้ามาเพื่อหาค่าสมาชิก

(3) ฟังก์ชันของตัวเลขแบบฟัซซี

CFuznum::setval(float& h , float& m , float& l)

เป็นฟังก์ชันที่ใช้สำหรับเซตค่าข้อมูลให้กับตัวแปรในคลาส

โดยที่

h : หมายถึงตัวแปรทางด้านค่าสูง (high value)

m : หมายถึงตัวแปรในช่วงกึ่งกลาง (medium value)

l : หมายถึงตัวแปรทางด้านค่าต่ำ (low value)

CFuznum::getlowval()

เป็นฟังก์ชันที่ใช้สำหรับส่งค่าข้อมูลทางด้านค่าต่ำออกมา

CFuznum::getmidval()

เป็นฟังก์ชันที่ใช้สำหรับส่งค่าข้อมูลทางด้านค่ากลางออกมา

CFuznum::gethighval()

เป็นฟังก์ชันที่ใช้สำหรับส่งค่าข้อมูลทางด้านค่าสูงออกมา

CFuznum::getshare(const float& input)

เป็นฟังก์ชันที่ใช้สำหรับให้ค่าของตัวเลขที่ป้อนเข้ามา ให้ออกมาเป็นค่าทางด้านภาษาศาสตร์

โดยที่

input : ค่าตัวเลขที่ป้อนเข้ามาเพื่อหาค่าทางด้านภาษาศาสตร์

- คลาสของส่วนอนุมาน (Inference Engine Class) ในคลาสนี้จะประกอบไปด้วยฟังก์ชันสมาชิกที่ใช้ในการอนุมานเพื่อวินิจฉัยหาเหตุผลของปัญหา ซึ่งจะประกอบไปด้วยฟังก์ชันดังนี้

(1) กลไกการอนุมานแบบเดินหน้า

Forward_chaining(char* fact , char* antecedent)

โดยที่

fact : ความรู้ที่เป็นข้อเท็จจริง

antecedent : ความรู้ที่ถูกเก็บไว้ในฐานความรู้ในส่วนของ antecedent

(2) กลไกการอนุมานแบบย้อนกลับ

Backward_chaining(char* fact , char* conclusion)

โดยที่

fact : ความรู้ที่เป็นข้อเท็จจริง

conclusion : ความรู้ที่ถูกเก็บไว้ในฐานความรู้ในส่วนของ conclusion

- คลาสของการแสดงความรู้ (Knowledge Representation Class) ในคลาสนี้จะประกอบไปด้วยฟังก์ชันที่ใช้ในการแปลงความรู้ที่มีอยู่ในฐานความรู้ให้อยู่ในรูปแบบที่สามารถนำความรู้นั้นไปใช้งานได้ง่าย ดังนี้คือ

- (1) แบบฐานกฎ
- (2) แบบโครงข่ายอรรถศาสตร์
- (3) แบบเฟรม

- คลาสของฐานความรู้ (Knowledge Base Class) ในคลาสนี้จะประกอบไปด้วยฟังก์ชันที่
ใช้เก็บความรู้ที่ได้รับจากผู้เชี่ยวชาญ ดังนี้คือ

- (1) แบบโครงสร้างต้นไม้

CTree::GetTree_Node(int Index)

เป็นฟังก์ชันสำหรับให้ค่าของโหนด สำหรับโครงสร้างแบบต้นไม้

CTree::GetTree_Left(int Index)

เป็นฟังก์ชันสำหรับให้ค่าของโหนดทางซ้ายมือ สำหรับโครงสร้างแบบต้นไม้

CTree::GetTree_Right(int Index)

เป็นฟังก์ชันสำหรับให้ค่าของโหนดทางขวามือ สำหรับโครงสร้างแบบต้นไม้

- (2) แบบเชื่อมโยง

CIList::Get_Pointer(int IPoint)

เป็นฟังก์ชันสำหรับให้ค่าตัวชี้สำหรับข้อมูล

CIList::Get_PointerNext(int IPoint)

เป็นฟังก์ชันสำหรับให้ค่าตัวชี้ตัวต่อไปสำหรับข้อมูล

CIList::Ins_LinkList(int* IPointNext, int* IPointIns , int* IPointInsNext)

เป็นฟังก์ชันสำหรับการเพิ่มข้อมูลเข้าไปในลิงค์ลิสต์

โดยที่

IPointNext : ค่าตัวชี้ของข้อมูลที่ตัวแทรกต้องการแทรกก่อน

IPointIns : ค่าตัวชี้ของตัวแทรก

IPointInsNext : ค่าตัวชี้ตัวต่อไปของข้อมูลในลิสต์

CIList::Del_LinkList(int* IPoint , int* IPointNext)

เป็นฟังก์ชันสำหรับการลบข้อมูลในลิงค์ลิสต์

โดยที่

IPoint : ค่าตัวชี้ของข้อมูลก่อนหน้าข้อมูลที่ต้องการลบจากลิสต์

IPointNext : คำชี้ของข้อมูลตัวต่อไปในลิสต์

- คลาสของการค้นหาข้อมูล (Search Class) ในคลาสนี้จะประกอบไปด้วยฟังก์ชันที่ใช้ในการค้นหาข้อมูล ซึ่งมีหลายวิธีด้วยกัน ดังนี้คือ

(1) การค้นหาแบบลึกก่อน

CDepFirst (int Index , int stark_right , int Last_node)

โดยที่

Last_node : ค่าของโหนดตัวสุดท้ายในฐานะความรู้

Stack_right : ค่าของโหนดทางด้านขวามือของโหนดย่อย

Index : ค่าของรูตโหนดในแต่ละโหนดย่อย

(2) การค้นหาแบบกว้างก่อน

CBreFirst (int Index)

โดยที่

Index : ค่าของโหนดลูกในแต่ละโหนดย่อย

(3) การค้นหาแบบดีที่สุดก่อน

CBestFirst (int Index , int value_right , int value_left)

โดยที่

Index : ค่าของโหนดในแต่ละโหนดย่อย

Value_right : คุณค่าของโหนดทางด้านขวามือ

Value_left : คุณค่าของโหนดทางด้านซ้ายมือ

- คลาสของการเรียงข้อมูล (Sort Class) ในคลาสนี้จะประกอบไปด้วยฟังก์ชันสมาชิกเพื่อใช้ในการจัดเรียงข้อมูล ตามที่วิศวกรความรู้ต้องการ ซึ่งสามารถจัดเรียงได้ทั้งข้อมูลที่เป็นตัวเลขและข้อมูลที่เป็นตัวอักษร นอกจากนั้นยังสามารถระบุบอกเพื่อจัดการกับข้อมูลว่าต้องการให้เรียงจากน้อยไปมาก หรือเรียงจากมากไปน้อยได้อีกด้วย ดังนี้

(1) แบบควิกซอร์ต

quick_sort(int* values, int start , int end_list, int order)

quick_sort(char values[][200], int start , int end_list, int order)

โดยที่

values : อาร์เรย์ของข้อมูลที่ต้องการจัดเรียง
 start : ตัวชี้ค่าแรกของลิสต์ในอาร์เรย์
 end_list : ตัวชี้ค่าสุดท้ายของลิสต์ในอาร์เรย์
 order : ลำดับการเรียงข้อมูลที่ต้องการ
 0 → เรียงจากน้อยไปมาก
 1 → เรียงจากมากไปน้อย

(2) แบบเชลล์

shell_sort (int * values, int num_elements , int order)
 shell_sort (char values[][200], int num_elements , int order)

โดยที่

values : อาร์เรย์ของข้อมูลที่ต้องการจัดเรียง
 num_elements : จำนวนของข้อมูลที่มีอยู่ในอาร์เรย์
 order : ลำดับของการเรียงข้อมูลที่ต้องการ
 0 → เรียงจากน้อยไปมาก
 1 → เรียงจากมากไปน้อย

(3) แบบเรียงลำดับผสาน

simple_merge(int* k , int first , int second , int third)

โดยที่

k : อาร์เรย์ของข้อมูลที่ต้องการจัดเรียง
 first : ตัวชี้ค่าแรกของข้อมูลในอาร์เรย์หนึ่ง
 second : ตัวชี้ค่าแรกของข้อมูลในอาร์เรย์สอง
 third : ตัวชี้ค่าสุดท้ายของข้อมูลในอาร์เรย์

- คลาสของภาษาธรรมชาติ (Natural Language Class) ในคลาสนี้จะประกอบไปด้วยฟังก์ชันที่ช่วยทำให้ระบบผู้เชี่ยวชาญ ฯ มีความยืดหยุ่นในการให้ข้อมูลในรูปประโยคที่มนุษย์เข้าใจมากยิ่งขึ้น ดังนี้คือ

(1) การวิเคราะห์คำศัพท์

(2) การวิเคราะห์วากยสัมพันธ์

CSyntax::seps_data(char* value[] , char list[])

โดยที่

list : อาร์เรย์ของข้อมูลที่ต้องการนำมาแตกเป็นคำ

value : อาร์เรย์ของข้อมูลที่แตกเป็นคำแล้ว เพื่อนำไปใช้งานต่อไป

CSyntax::Syntax_Analysis(char* analy[] , char data[] , int Index)

โดยที่

Index : ตัวชี้ของข้อมูลในอาร์เรย์ที่ต้องการเก็บ

Analy : ข้อมูลที่ผ่านการวิเคราะห์ประโยคแล้ว

Data : ชนิดของข้อมูล

(3) การวิเคราะห์ความหมาย

CSemantic::Sem_Analysis(char* analy[] , char semantic[][100])

โดยที่

analy : อาร์เรย์ของข้อมูลที่ผ่านการวิเคราะห์ประโยคแล้ว

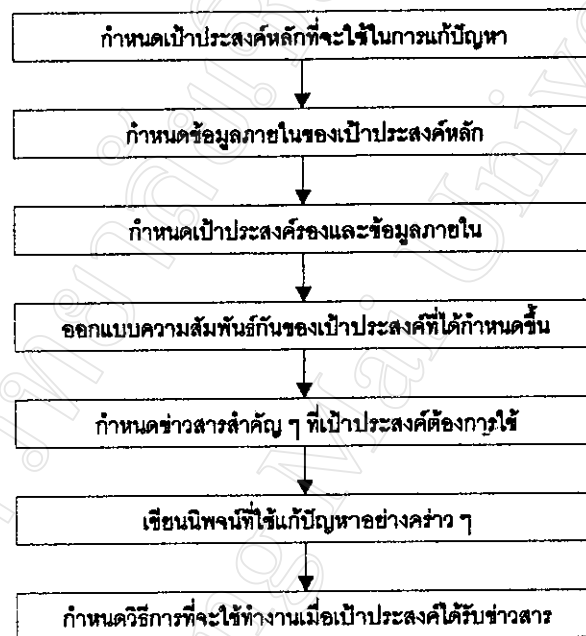
Semantic : อาร์เรย์ของประโยคที่ผ่านการวิเคราะห์ความหมายมาแล้ว

การออกแบบคลาส

คลาสเป็นหลักการสำคัญอย่างหนึ่งของวิธีการโปรแกรมเชิงประสงค์ (Object – Oriented Programming) ซึ่งเป็นวิธีในการพัฒนาซอฟต์แวร์ โดยมุ่งเน้นที่จะเพิ่มประสิทธิภาพในการผลิต หลักการพื้นฐานของการพัฒนาซอฟต์แวร์เชิงประสงค์ก็คือ การแก้ปัญหาจะทำได้ด้วยการส่งข่าวสาร (Message) ให้กับประสงค์ (Object) ซึ่งจะต้องมีการออกแบบประสงค์และข่าวสารขึ้นมาให้เสียก่อน

ขั้นตอนแรกของการออกแบบคลาสคือการกำหนดเป้าประสงค์หลักที่จะใช้ในการแก้ไข ปัญหา ว่าเราต้องการสร้างเป้าประสงค์หลักอะไร จากนั้นจึงกำหนดข้อมูลภายในของเป้าประสงค์หลัก เพื่อให้ได้ผลลัพธ์ที่ต้องการ หากว่าเป้าประสงค์หลักสามารถแตกแยกย่อยเป็นเป้าประสงค์รองได้ ก็ให้ทำการแตกเป้าประสงค์หลักออกเป็นเป้าประสงค์รอง หลาย ๆ เป้า แล้วจึงกำหนดข้อมูลภายในของเป้าประสงค์รองต่อไป การออกแบบคลาสที่มีเป้าประสงค์รองหลาย ๆ เป้าประสงค์นั้น โปรแกรมเมอร์จำเป็นต้องทำการจัดความสัมพันธ์กันของเป้าประสงค์เหล่านั้นให้มีความสัมพันธ์กัน ส่วนใดที่ทำหน้าที่

เหมือนกันให้จัดไว้เป็นเป้าประสงค์ใหม่ หลังจากที่โปรแกรมเมอร์ทำการออกแบบความสัมพันธ์ของเป้าประสงค์ที่ได้กำหนดขึ้นมาแล้ว ต่อไปก็จะเป็นการกำหนดข่าวสารที่เป้าประสงค์แต่ละตัวต้องการใช้ในการทำงาน เมื่อเป้าประสงค์ได้รับข่าวสาร มันก็จะให้ผลลัพธ์ออกมาตามที่ต้องการ หลังจากนั้นเราจึงนำเอาข้อมูลดังกล่าวมาเขียนเป็นคลาสตามที่ต้องการ ซึ่งขั้นตอนการออกแบบคลาสนี้สามารถเขียนเป็นผังงานได้ดังรูปที่ 4.2



รูปที่ 4.2 แสดงถึงผังงานการออกแบบคลาส

องค์ประกอบ	ประเภทของ	ตัวแปร	ผลลัพธ์
การให้เหตุผลแบบพีชชี	พีชชีเซต พีชชีลอจิก ตัวเลขพีชชี	ค่าตัวเลขที่เป็นตัวแปรทางภาษาศาสตร์ “ “	ตัวแปรทางภาษาศาสตร์ “ “
ส่วนอนุมาน	แบบเดินหน้า แบบย้อนกลับ	ข้อมูลที่เป็นเหตุ ข้อมูลที่เป็นผล	ข้อมูลที่เป็นผล ข้อมูลที่เป็นเหตุ
การแสดงความรู้	ฐานกฎ โครงข่ายอรรถศาสตร์ เฟรม	ค่าในส่วนของเงื่อนไข ค่าของโหนด ค่าแนวคิด หรือหลักการ	ค่าในส่วนของกรกระทำ ค่าความสัมพันธ์ใน โหนด ค่าในสล็อต
การค้นหาข้อมูล	แบบลึกก่อน แบบกว้างก่อน แบบกว้างก่อน	ข้อมูล “ “	เส้นทางของข้อมูล “ “
การเรียงข้อมูล	แบบลำดับผสม แบบเซลล์ แบบควิกซอร์ต	ข้อมูล “ “	การจัดเรียงข้อมูล “ “
ภาษาธรรมชาติ	คำศัพท์ วากยสัมพันธ์ ความหมาย	คำศัพท์ ประโยค ประโยค	ชนิดของคำศัพท์ ผลวิเคราะห์ประโยค ผลวิเคราะห์ความหมาย
ฐานความรู้	คั่นไม้ แบบเชื่อมโยง	โหนด ตัวชี้	ค่าในโหนด “

ตารางที่ 4.1 แสดงถึงการกำหนดข่าวสารและผลลัพธ์ ของคลาสที่พัฒนาขึ้น

จากตารางที่ 4.1 จะเป็นการแตกคลาสต่าง ๆ ที่ได้ออกแบบไว้ โดยทำการกำหนดข่าวสาร และผลลัพธ์ต่าง ๆ ที่ต้องการเพื่อนำไปใช้ในการสร้างคลาสต่อไป

การสร้างคลาสที่ได้ทำการออกแบบไว้นี้ เราจะใช้โปรแกรมของไมโครซอฟต์วิซวลซี ซึ่งโปรแกรมดังกล่าวจะมีโปรแกรมวิซาร์ด เพื่อช่วยโปรแกรมเมอร์ในการจัดสร้างคลาสที่ต้องการได้ง่าย ซึ่งคลาสที่ได้จัดทำขึ้นนี้สามารถจะทำการคอมไพล์ให้เป็นไฟล์ที่มีนามสกุลเป็น LIB เมื่อต้องการใช้

งานคลาสที่ได้จัดทำไว้แล้วนี้ เราเพียงแต่เพิ่มเติมคำสั่ง (Include) ไฟล์ที่มีนามสกุล .H และทำการบวกเพิ่มไฟล์ที่มีนามสกุล .LIB นี้เข้าไปในโปรเจกต์ของเรา เราก็สามารถใช้งานคลาสดังกล่าวได้

4.2 การทดสอบเพื่อปรับปรุงและการประเมินผลการทำงานของเปลือกระบบผู้เชี่ยวชาญ

กระบวนการทดสอบเปลือกระบบผู้เชี่ยวชาญแบบพีซี จะใช้วิธีการนำเอาเปลือกระบบผู้เชี่ยวชาญ ๑ ที่ได้จัดทำขึ้น มาสร้างเป็นระบบผู้เชี่ยวชาญเพื่อการวินิจฉัยโรคทางคลินิกเบื้องต้น ซึ่งระบบผู้เชี่ยวชาญที่เราสร้างขึ้นมาเพื่อทดสอบนี้ เราจะเน้นเฉพาะในส่วนของการวินิจฉัยโรคสำหรับผู้ป่วยที่มีอาการไข้เป็นหลัก ดังนั้นกระบวนการทดสอบของเราจะแบ่งออกเป็น 2 ขั้นตอนหลัก ๆ คือ การทดสอบเพื่อปรับปรุงเปลือกระบบผู้เชี่ยวชาญ และการทดสอบเพื่อประเมินผลการทำงานของเปลือกระบบผู้เชี่ยวชาญเมื่อถูกนำไปพัฒนาเป็นระบบผู้เชี่ยวชาญจริง

4.2.1 การทดสอบเพื่อปรับปรุงเปลือกระบบผู้เชี่ยวชาญ

การทดสอบในขั้นตอนนี้ จะเป็นการตรวจสอบระบบจากวิศวกรความรู้ เพื่อแก้ไขปรับปรุงเปลือกระบบผู้เชี่ยวชาญ ๑ ให้มีความสมบูรณ์ภายหลังจากที่วิศวกรความรู้ได้พัฒนาเปลือกระบบผู้เชี่ยวชาญต้นแบบเสร็จสิ้นลงในแต่ละส่วนแล้ว โดยขั้นตอนการทดสอบจะแบ่งออกได้อีก 2 ส่วนคือ

- 1) การตรวจสอบและแก้ไขจุดบกพร่อง (Debug) ในแต่ละฟังก์ชันการทำงานของคลาตในระบบ ๑ โดยวิศวกรความรู้
- 2) การตรวจสอบและแก้ไขจุดบกพร่องในการทำงานของระบบ เมื่อวิศวกรความรู้นำเอาฟังก์ชันต่าง ๆ ในคลาต มาประกอบเป็นระบบผู้เชี่ยวชาญตามที่ต้องการ

4.2.2 การทดสอบเพื่อประเมินผลการทำงานของเปลือกระบบผู้เชี่ยวชาญ

การทดสอบเพื่อประเมินผลการทำงานของเปลือกระบบผู้เชี่ยวชาญ ๑ ต้นแบบที่พัฒนาขึ้นในที่นี้ได้ดำเนินการทดสอบโดยนำเอาเปลือกระบบผู้เชี่ยวชาญนี้ไปพัฒนาเป็นระบบผู้เชี่ยวชาญเพื่อการวินิจฉัยโรคทางคลินิกเบื้องต้น โดยจะเน้นเฉพาะในส่วนของผู้ป่วยที่มีอาการไข้เป็นหลัก ซึ่งกระบวนการทั้งหมดจะประกอบไปด้วย 3 ขั้นตอนหลัก ๆ คือ

1) การสกัดเอาความรู้ในการวินิจฉัยโรคจากตำราทางการแพทย์ เกี่ยวกับหลักการวินิจฉัยและรักษาโรค

2) ออกแบบสถาปัตยกรรมของระบบผู้เชี่ยวชาญเพื่อการวินิจฉัยโรคทางคลินิกเบื้องต้น ดังรูปที่ 4.3 โดยมีรายละเอียดดังนี้

- การแทนความรู้ในฐานความรู้จะใช้ระบบฐานกฎ โดยมีกฎทั้งหมด 94 กฎ
- ใช้กลไกการอนุมานเป็นแบบเดินหน้า โดยมีการใช้พีซีเข้ามาช่วยในการตัดสินใจหาข้อสรุปจากกฎที่มีอยู่

ใจหาข้อสรุปจากกฎที่มีอยู่

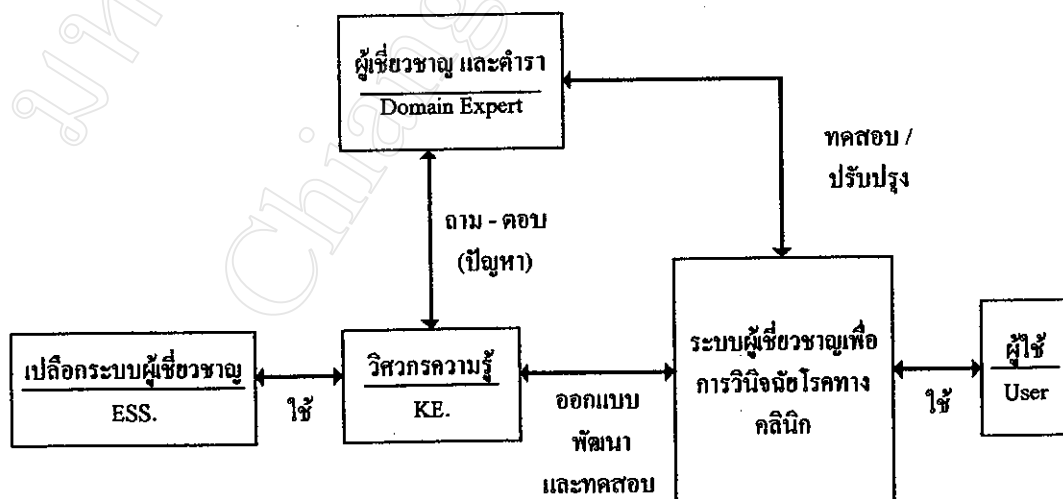
- การค้นหาความรู้ในฐานความรู้จะใช้แบบลึกก่อน
- การเก็บความรู้ไว้ในฐานความรู้ จะใช้วิธีเก็บแบบรูปคั่นไม้
- ส่วนที่ใช้ติดต่อกับผู้ใช้งานจะสร้างในลักษณะของกราฟฟิก โดยออกแบบให้เป็นรูปหน้าต่างและเมนู

เป็นรูปหน้าต่างและเมนู

- ภาษาที่ใช้ในการโต้ตอบระหว่างระบบฯ กับผู้ใช้งานจะใช้ลักษณะของภาษาธรรมชาติ

ธรรมชาติ

3) การทดสอบเพื่อแก้ไขปรับปรุงและประเมินผลการทำงานของเปลือกระบบ ฯ



รูปที่ 4.3 แสดงถึงวิศวกรรมความรู้ของการพัฒนาระบบผู้เชี่ยวชาญเพื่อการวินิจฉัยโรคทางคลินิก

4.3 การนำปณิธานของผู้เชี่ยวชาญไปพัฒนาเป็นระบบผู้เชี่ยวชาญเพื่อการวินิจฉัยโรคทางคลินิก

แนวคิดของการวินิจฉัยโรคทางคลินิกเบื้องต้น

โดยทั่วไป แพทย์หรือผู้ทำการรักษาจะอาศัยข้อมูลจากการซักถามอาการ การตรวจร่างกาย และการตรวจทางห้องปฏิบัติการ เข้าช่วยในการพิจารณาหรือวินิจฉัยโรค และกำหนดแนวทางการดูแลรักษา การซักถามอาการหรือที่ทางการแพทย์เรียกว่า “ การซักประวัติการเจ็บป่วย “ (History taking) จึงมีความสำคัญต่อกระบวนการตรวจรักษาโรค กล่าวกันว่าโรคที่พบเห็นในชีวิตประจำวันกว่าครึ่งหนึ่ง สามารถวินิจฉัยได้จากการซักถามอาการเพียงอย่างเดียว

ในทางการแพทย์ ได้แบ่งประวัติการเจ็บป่วยออกเป็น 9 ส่วนด้วยกันคือ

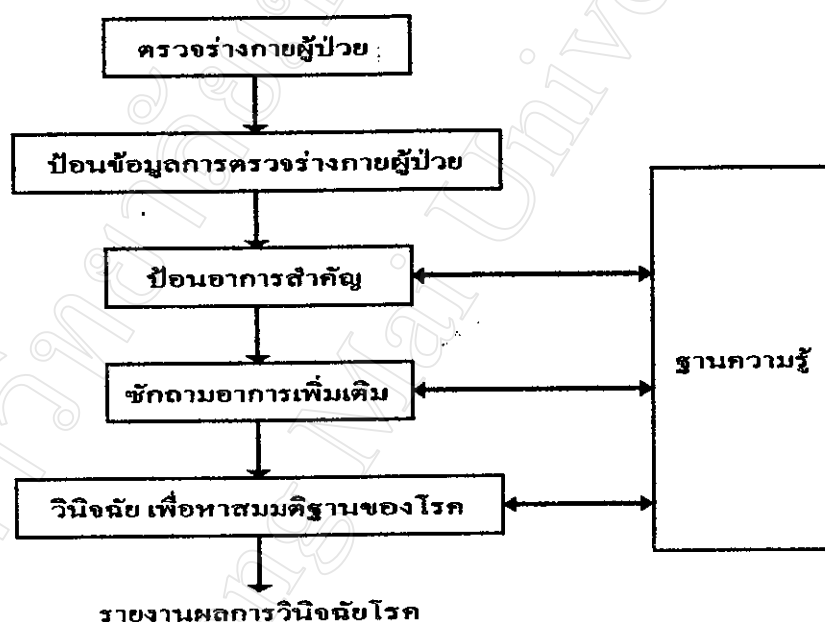
- 1) อาการสำคัญ (อ.ส.)
- 2) ประวัติการเจ็บป่วยในปัจจุบัน (ป.ป.)
- 3) ประวัติการเจ็บป่วยในอดีต (ป.อ.)
- 4) ประวัติการเจ็บป่วยในครอบครัว (ป.ค.)
- 5) ประวัติการเจ็บป่วยในคนข้างเคียง (ป.ข.)
- 6) ประวัติส่วนตัว (ป.ส.)
- 7) ประวัติในเด็ก (ป.ด.)
- 8) ประวัติในผู้หญิง (ป.ญ.)
- 9) ประวัติตามระบบ (ป.ร.)

ประวัติส่วนที่ถือว่าเป็นหัวใจสำคัญก็คือ อาการสำคัญกับประวัติการเจ็บป่วยในปัจจุบัน ส่วนประวัติข้ออื่น ๆ เพียงแต่เป็นส่วนประกอบเท่านั้น

สำหรับระบบผู้เชี่ยวชาญๆ เพื่อการวินิจฉัยโรคทางคลินิกในที่นี้จะใช้อาการสำคัญ และประวัติการเจ็บป่วยในปัจจุบัน มาเป็นหลักในการวินิจฉัย ซึ่งอาการสำคัญนี้จะหมายถึงอาการหลักๆ ที่นำผู้ป่วยมาพบกับหมอ โดยมากมักจะเป็นอาการเพียง 1-2 อย่าง เช่น ปวดศีรษะ เป็นไข้ ปวดท้อง หรือ ท้องเดิน เป็นต้น ส่วนประวัติการเจ็บป่วยในปัจจุบัน จะหมายถึงประวัติอาการต่างๆ ของผู้ป่วย และการรักษาที่ผู้ป่วยได้รับนับตั้งแต่เริ่มไม่สบายมา

การวินิจฉัยโรคเป็นกระบวนการแก้ปัญหาทางคลินิกโดยใช้ความรู้และประสบการณ์ที่มีอยู่ประกอบกับวิธีวินิจฉัยโรค มาวิเคราะห์และสังเคราะห์ข้อมูลที่ได้ เพื่อให้เกิดข้อสรุปหรือคำตอบว่าผู้ป่วยเป็นโรคอะไร การวินิจฉัยโรคเป็นส่วนที่สำคัญที่สุดในกระบวนการแก้ปัญหาผู้ป่วย เพราะถ้าการวินิจฉัยโรคผิดพลาด แพทย์จะพยากรณ์โรค และรักษาโรคผิดทางไปด้วย

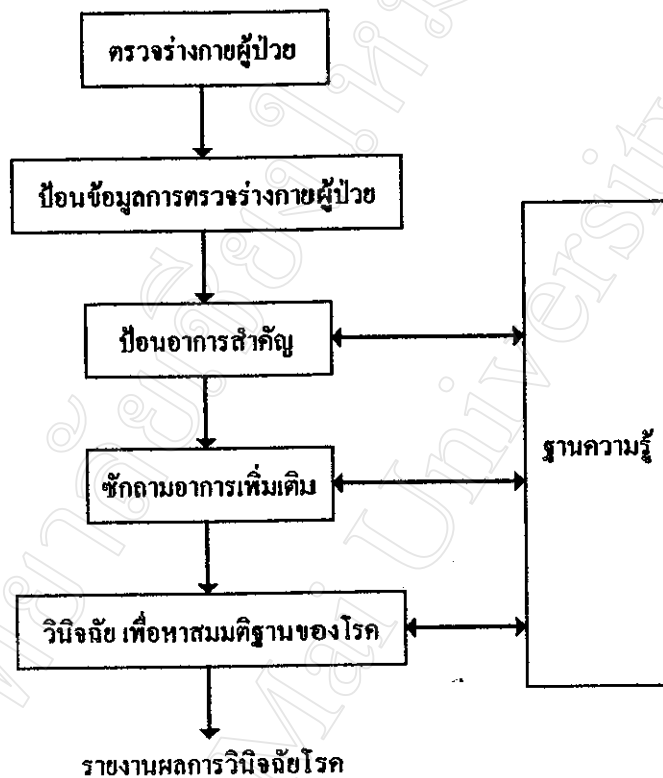
จากการพิจารณากระบวนการวินิจฉัยโรคของแพทย์ เราสามารถกำหนดเป็นแนวทางในการสร้างแบบจำลองกระบวนการวินิจฉัยโรคของระบบผู้เชี่ยวชาญค้นแบบได้ดังรูปที่ 4.4



รูปที่ 4.4 แสดงถึงแบบจำลองกระบวนการวินิจฉัยโรคของระบบผู้เชี่ยวชาญค้นแบบ

สถาปัตยกรรมของระบบผู้เชี่ยวชาญ เพื่อการวินิจฉัยโรคทางคลินิก

สถาปัตยกรรมของระบบผู้เชี่ยวชาญการวินิจฉัยโรคทางคลินิก มีโครงสร้างประกอบไปด้วยองค์ประกอบที่สำคัญ 7 ส่วน ดังรูปที่ 4.5 คือ



รูปที่ 4.5 แสดงสถาปัตยกรรมของระบบผู้เชี่ยวชาญการวินิจฉัยโรคทางคลินิก

1) ฐานความรู้ : เป็นส่วนที่ใช้เก็บรวบรวมความรู้ สำนวนการวินิจฉัยโรค โดยเก็บไว้ในรูปของกฎ

2) ส่วนวินิจฉัยโรค : เป็นส่วนที่มีการใช้กลไกการอนุมานความรู้ที่ใช้ในการวินิจฉัยโรคของระบบ การวินิจฉัยโรคในส่วนนี้จะใช้กระบวนการอนุมานเป็นแบบเค้นหน้า ซึ่งจะนำเอาความรู้ในรูปของกฎ จากฐานความรู้ และข้อเท็จจริงที่ได้จากผู้ใช้รวมไปถึง ผลการตรวจร่างกาย การซักถามอาการสำคัญ และการซักถามประวัติเพิ่มเติม เพื่อนำมาใช้ประกอบการวินิจฉัย

3) ส่วนเชื่อมโยงกับผู้ใช้ : เป็นส่วนที่ทำหน้าที่ประมวลผลในเรื่องของภาษาและเป็นส่วนที่เชื่อมโยงระหว่างผู้ใช้กับระบบฯ ให้ผู้ใช้สามารถติดต่อกับระบบฯ ได้ง่ายขึ้น

4) หน่วยความจำใช้งาน : เป็นหน่วยเก็บบันทึกข้อมูลชั่วคราวของระบบ เพื่อใช้เก็บข้อเท็จจริงที่ได้จากข้อมูลเบื้องต้นของผู้ป่วย ผลการตรวจร่างกาย การซักถามอาการที่สำคัญ การซักถามประวัติเพิ่มเติม และข้อเท็จจริงที่ได้มาในระหว่างการอนุมานความรู้

การออกแบบฐานความรู้

ความรู้สำหรับการวินิจฉัยโรคทางคลินิก ซึ่งนำมาเข้ารหัสให้อยู่ในรูปของกฎที่เหมาะสมกับกลไกการอนุมาน เป็นความรู้ที่ได้มาจากการทางด้านการแพทย์ เป็นหลัก การออกแบบฐานความรู้ของระบบผู้เชี่ยวชาญนี้จะแบ่งออกได้เป็น 2 ส่วนความลักษณะการแทนความรู้ของระบบ คือ การออกแบบฐานความรู้ในรูปกฎฟัซซี่ และการออกแบบฐานความรู้ในรูปกฎโครงคักชัน

การออกแบบฐานความรู้ในรูปกฎฟัซซี่ : ความรู้ในส่วนนี้จะมีความสัมพันธ์กับข้อมูลเบื้องต้นที่ได้จากการตรวจร่างกายของผู้ป่วยเฉพาะในส่วนของการใช้ปรอทวัดไข้ ซึ่งการออกแบบฐานความรู้ส่วนนี้ จะนำเอาความรู้ข้างต้นมาแปลงส่งเข้าสู่กระบวนการแทนความรู้ในรูปกฎฟัซซี่ ซึ่งประกอบด้วย การนิยามตัวแปรทางภาษา การนิยามฟัซซี่เซต และการนิยามกฎฟัซซี่

พิจารณาความรู้เกี่ยวกับอุณหภูมิของร่างกาย โดยการวัดทางปากสามารถแทนความรู้ในรูปกฎฟัซซี่ ได้ดังนี้

- ชื่อของตัวแปรทางภาษาคือ อุณหภูมิ

- เทอมของเซต คือ

T (อุณหภูมิ) = {ต่ำ, ปกติ, ค่อนข้างสูง, สูง, ค่อนข้างสูงมาก, สูงมาก}

- เอกภพสัมพันธ์ของอุณหภูมิร่างกายมีค่าเป็นองศาเซลเซียส

$$U = [35, 45]$$

กำหนดค่าระดับการเป็นสมาชิกของฟัซซี่เซต โดยใช้ฟังก์ชันรูปประฆัง ดังสมการ 2.6 และนิยามฟัซซี่เซตของอุณหภูมิในรูปฟังก์ชันการเป็นสมาชิก ดังนี้

$$\mu_{\text{ต่ำ}}(X) = 1 - S(X; 36.10, 36.15, 36.20)$$

$$\mu_{\text{ปกติ}}(X) = \begin{cases} s(x; 36.10, 36.15, 36.20) & ; x < 36.70 \\ 1 - s(x; 37.20, 37.25, 37.30) & ; x \geq 36.70 \end{cases}$$

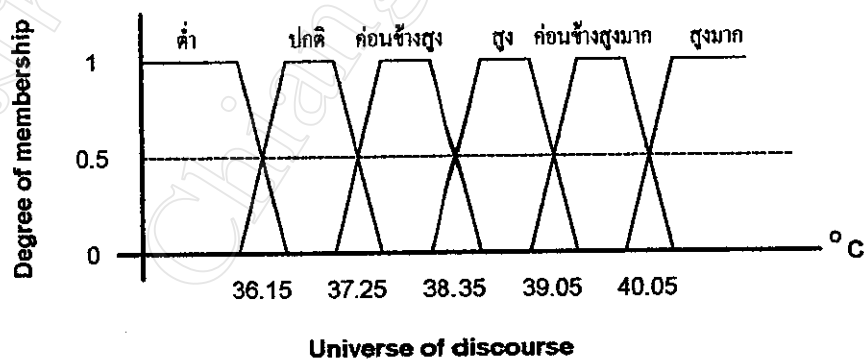
$$\mu_{\text{ค่อนข้างสูง}}(X) = \begin{cases} s(x; 37.20, 37.25, 37.30) & ; x < 37.80 \\ 1 - s(x; 38.30, 38.35, 38.40) & ; x \geq 37.80 \end{cases}$$

$$\mu_{\text{สูง}}(X) = \begin{cases} s(x; 38.30, 38.35, 38.40) & ; x < 38.70 \\ 1 - s(x; 39.00, 39.05, 39.10) & ; x \geq 38.70 \end{cases}$$

$$\mu_{\text{ค่อนข้างสูงมาก}}(X) = \begin{cases} s(x; 39.00, 39.05, 39.10) & ; x < 39.55 \\ 1 - s(x; 40.00, 40.05, 40.10) & ; x \geq 39.55 \end{cases}$$

$$\mu_{\text{สูงมาก}}(X) = S(X; 40.00, 40.05, 40.10)$$

เราสามารถแทนค่าของอุณหภูมิทั้งหมดบนเอกภพสัมพัทธ์ที่กำหนดไว้และเขียนเป็นกราฟแสดงฟังก์ชันเซตของอุณหภูมิได้ดังรูป 4.6



รูปที่ 4.6 กราฟแสดงฟังก์ชันเซตของอุณหภูมิ

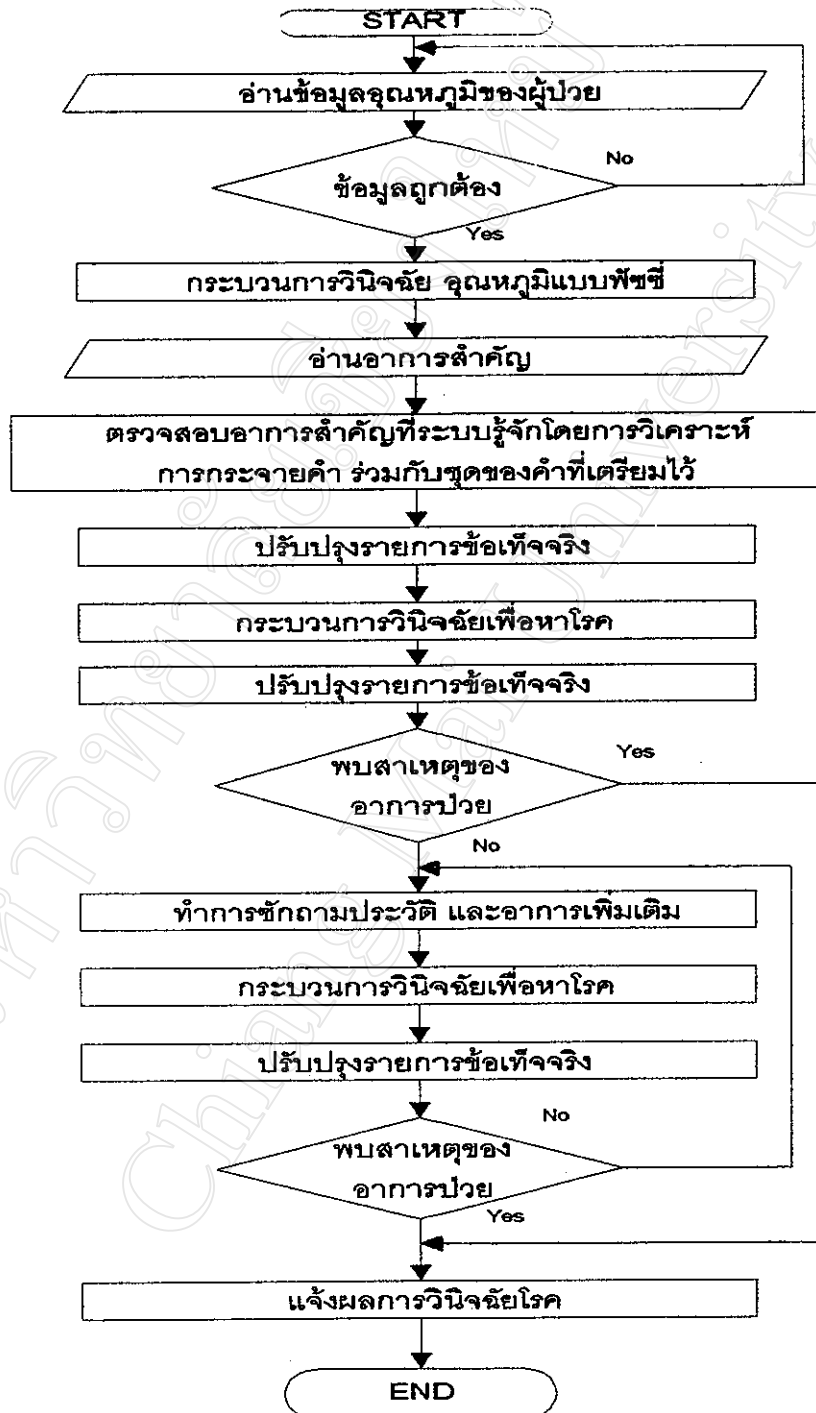
การออกแบบฐานความรู้ในรูปกฎโพรคักชัน : ความรู้ในส่วนนี้จะมีความสัมพันธ์กับข้อมูลเบื้องต้นที่ได้จากการซักถามอาการสำคัญที่นำผู้ป่วยมารับการรักษา ซึ่งการออกแบบฐานความรู้ส่วนนี้จะนำเอาความรู้ที่ได้ดังกล่าวข้างต้นมาแปลง แล้วส่งเข้าสู่กระบวนการแทนความรู้ในรูปกฎโพรคักชัน ซึ่งกฎในฐานความรู้นี้จะได้รับการออกแบบให้มีโครงสร้างเป็นรายการของกฎ โดยที่แต่ละกฎจะประกอบไปด้วย 2 ส่วนคือ ส่วนเงื่อนไข และส่วนกระทำ ส่วนทั้งสองนี้จะประกอบด้วยอนุประโยคที่เป็นข้อพิสูจน์ และอนุประโยคที่เป็นข้อสรุป ตามลำดับ อนุประโยคนี้สามารถกำหนดให้มีเพียงหนึ่งประโยค หรือกำหนดให้มีหลาย ๆ ประโยคก็ได้ ขึ้นอยู่กับการออกแบบของวิศวกรความรู้

การออกแบบส่วนวินิจฉัยโรค

ส่วนวินิจฉัยโรคนี้ เป็นส่วนอนุมานความรู้ของระบบผู้เชี่ยวชาญ ซึ่งได้รับการออกแบบการจัดวางโครงสร้าง และกลไกการทำงาน ให้มีลักษณะใกล้เคียงกับกระบวนการวินิจฉัยและรักษาโรคของแพทย์ ในส่วนการวินิจฉัยโรคนี้ จะเป็นการอนุมานเพื่อหาว่าผู้ป่วยเป็นโรคอะไร ลักษณะของการวินิจฉัยจะเริ่มต้นจากข้อเท็จจริงของอาการและการแสดงของโรค ซึ่งเป็นปัญหาที่ผู้ป่วยไม่ทราบคำตอบล่วงหน้า และคำตอบที่เป็นไปได้มีจำนวนมาก การออกแบบกลไกการอนุมานในส่วนนี้จะกำหนดให้ใช้กระบวนการอนุมานเป็นแบบเดินหน้า จากขั้นตอนของผังงานรูปที่ 4.7 กระบวนการแก้ปัญหาของผู้ป่วยของระบบผู้เชี่ยวชาญ เริ่มต้นที่การอ่านข้อมูลซึ่งเป็นผลจากการตรวจร่างกาย นั่นคือ อุณหภูมิของผู้ป่วย(วัดทางปาก)จากนั้นระบบจะทำการตรวจสอบความถูกต้องของข้อมูล และทำการวินิจฉัยเพื่อทำการบันทึกข้อมูลเกี่ยวกับระดับอุณหภูมิของผู้ป่วย โดยทำการบันทึกอุณหภูมิเป็นแบบพีชชี จากนั้นจะทำการซักถามอาการสำคัญที่เป็นสาเหตุให้ผู้ป่วยมาขอรับการรักษา และตรวจสอบว่าอาการนั้นเป็นอาการสำคัญที่ระบบรู้จักหรือไม่ ถ้าปรากฏว่าระบบไม่รู้จักอาการดังกล่าว ระบบจะทำการแจ้งและแสดงข้อความอธิบายให้ผู้ใช้ทราบ ถ้าปรากฏว่ารู้จัก ระบบจะนำอาการนี้เข้าสู่กระบวนการวินิจฉัยเพื่อหาโรคเป็นลำดับถัดไป

การออกแบบส่วนเชื่อมโยงกับผู้ใช้

ส่วนเชื่อมโยงกับผู้ใช้ นี้ เป็นตัวกลางที่ใช้ในการติดต่อสื่อสารกันระหว่างระบบกับผู้ใช้ ซึ่งในระบบผู้เชี่ยวชาญนี้ จะให้ผู้ใช้ทำการป้อนรายละเอียดต่าง ๆ ลงในช่องที่กำหนดให้ โดยผู้ใช้



รูปที่ 4.7 แสดงผังงาน ขั้นตอนวิธีการวินิจฉัยโรคของระบบผู้เชี่ยวชาญค้นแบบ

สามารถป้อนข้อมูลเข้ามาเป็นภาษาไทยได้ แล้วระบบจะนำข้อมูลที่ได้อ้างกล่าวไปวิเคราะห์ตามกระบวนการ จากนั้นจึงให้คำตอบออกมา

4.3.1 ผลการทดลอง

จากการทดสอบเพื่อประเมินผลการทำงานของเปลือกระบบผู้เชี่ยวชาญต้นแบบที่พัฒนาขึ้น ด้วยการนำเอาเปลือกระบบผู้เชี่ยวชาญนี้ไปพัฒนาเป็นระบบผู้เชี่ยวชาญเพื่อการวินิจฉัยโรคทางคลินิกเบื้องต้น ที่เน้นเฉพาะในส่วนของผู้ป่วยที่มีอาการ ไข้เป็นหลัก โดยจัดสร้างฐานความรู้ในรูปแบบฐานกฎจำนวน 94 กฎ พบว่าเปลือกระบบผู้เชี่ยวชาญต้นแบบสามารถนำมาพัฒนาเป็นระบบผู้เชี่ยวชาญได้ ซึ่งฟังก์ชันต่าง ๆ ที่มีในแต่ละคลาสจะช่วยอำนวยความสะดวกให้กับวิศวกรความรู้ ในการนำเอาฟังก์ชันดังกล่าวมาประกอบกันเป็นระบบผู้เชี่ยวชาญ ทำให้การพัฒนากระบวนผู้เชี่ยวชาญกระทำได้ง่ายและรวดเร็วขึ้น โดยที่วิศวกรความรู้ไม่จำเป็นต้องมีความเชี่ยวชาญในการเขียนโปรแกรมมากมาย ก็สามารถทำการพัฒนาระบบผู้เชี่ยวชาญขึ้นมาได้