บทที่ 2
ความรู้เกี่ยวกับระบบฐานความรู้

2.1 บทนำ

ปัญญาประดิษฐ์ (Artificial Intelligence: AI) เป็นแขนงวิชาหนึ่งของศาสตร์ทางคอมพิวเตอร์ที่มุ่งเน้นในด้านการทำความเข้าใจ เกี่ยวกับวิธีการทำให้คอมพิวเตอร์สามารถแสดงความฉลาดออกมาได้ เช่น การคิด การหาเหตุผล การรับรู้ เป็นต้น สถาบันที่ทำงานวิจัยทางด้านปัญญาประดิษฐ์ในระยะแรก ได้แก่ Massachusetts Institute of Technology เมื่อปี 2506 นีเวล (Newell) ได้ทำการค้นคว้าเพื่อหาวิธีการออกแบบระบบการแก้ปัญหา (Problem Solving System) สำหรับคอมพิวเตอร์ขึ้น หลังจากนั้นเทคนิคต่างๆ ทางด้านปัญญาประดิษฐ์ก็ได้ถูกพัฒนาขึ้นมา ระบบผู้เชี่ยวชาญ (Expert System: ES) ต้นแบบหลายตัวจึงถูกสร้างขึ้น

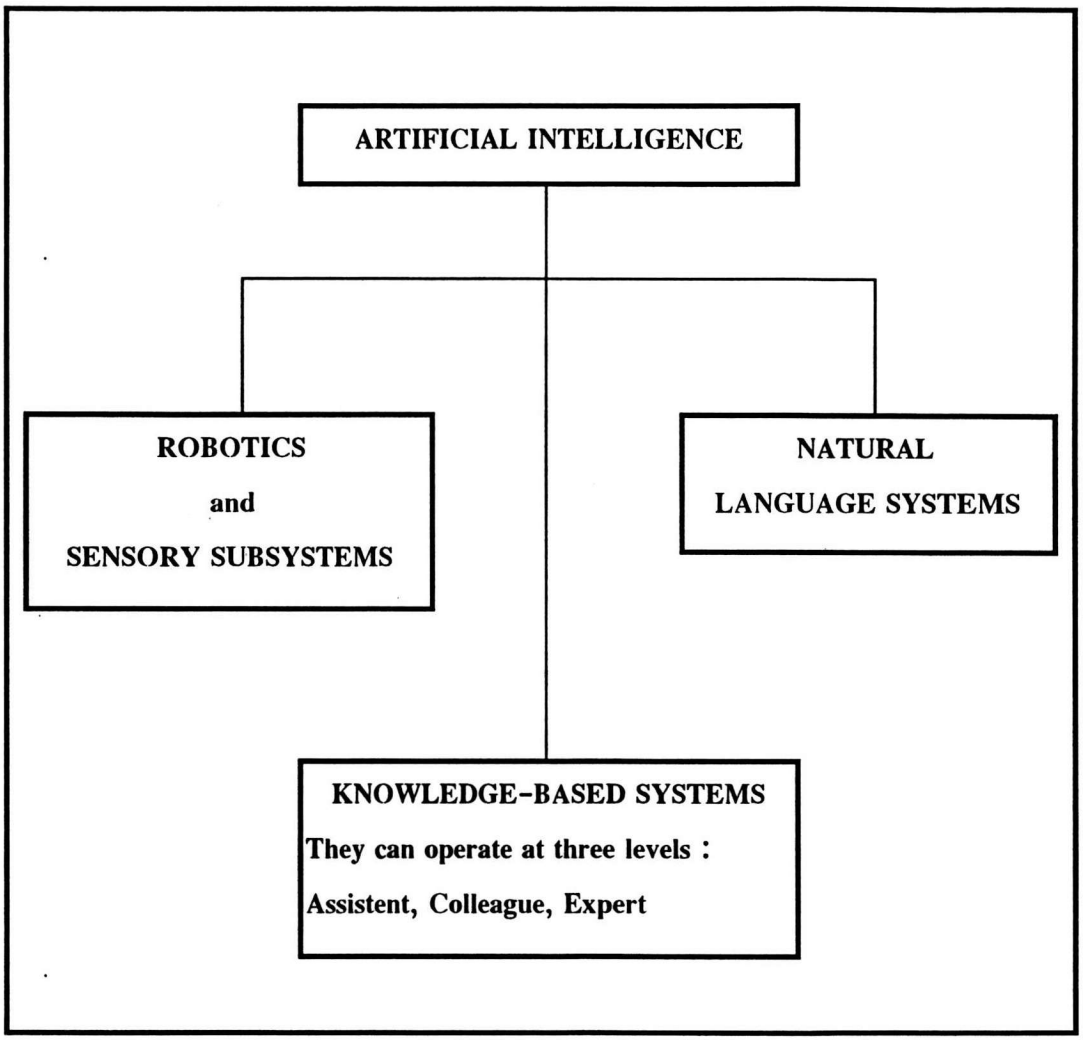
ระบบผู้เชี่ยวชาญเป็นสาขาใหม่ที่เพิ่งเกิดขึ้น ตามหลักการแล้วระบบผู้เชี่ยวชาญไม่ใช่เรื่องที่ยากมาก แต่เนื่องจากระบบผู้เชี่ยวชาญได้เปลี่ยนแนวคิดบางอย่างจากวิธีการเขียนโปรแกรมแบบเดิม จึงทำให้ดูเหมือนเป็นเรื่องที่ยากและน่าอัศจรรย์ ระบบผู้เชี่ยวชาญจะถูกใช้เพื่อทำงานที่ซับซ้อนมากๆ ซึ่งในอดีตงานประเภทนี้จะสามารถทำได้ก็ต้องอาศัยผู้เชี่ยวชาญที่เป็นมนุษย์เท่านั้น ด้วยวิธีการประยุกต์ใช้งานด้านปัญญาประดิษฐ์ ระบบผู้เชี่ยวชาญจะรับเอาความรู้พื้นฐานซึ่งมนุษย์เป็นผู้ใส่ให้มาทำการประเมินผลเช่นเดียวกับการที่มนุษย์แก้ปัญหาที่ซับซ้อน

สิ่งที่เรียกได้ว่าเป็นความสำเร็จทางด้านปัญญาประดิษฐ์ คือการพัฒนาระบบผู้เชี่ยวชาญที่เกิดขึ้นเมื่อประมาณกลางทศวรรษ 1960 เมื่อนักวิจัยทางด้านปัญญาประดิษฐ์หันมาสนใจการสร้างโปรแกรมที่มีประสิทธิภาพสูง และการแก้ปัญหาพิเศษเฉพาะด้าน โดยอาศัยความรู้ที่จะต้องรวบรวมจากผู้เชี่ยวชาญ

2.2 นิยามของระบบฐานความรู้

ระบบฐานความรู้เป็นส่วนหนึ่งในแขนงวิชาปัญญาประดิษฐ์ (Bowerman and Glover 1988; Charniack and McDermott 1985; Feigenbaum, McCorduck, and Nii 1988; Harmon and

Sawyer 1990; Martin and Oxman 1988; Mockler 1989; Patterson 1990) ระบบนี้สามารถทำหน้าที่เป็นผู้ช่วย, ผู้ร่วมงาน หรือระดับผู้เชี่ยวชาญ ดังรูปที่ 2.1



รูปที่ 2.1 แขนงวิชาของปัญญาประดิษฐ์

ระบบฐานความรู้แบบผู้เชี่ยวชาญ (Knowledge-Based Expert System; KBES) เป็นโปรแกรมคอมพิวเตอร์ที่ถูกออกแบบขึ้นมาเพื่อทำหน้าที่ในการเป็นผู้เชี่ยวชาญ และให้คำปรึกษาแก่มนุษย์ในเรื่องต่างๆ ซึ่งเป็นส่วนหนึ่งในแขนงวิชาปัญญาประดิษฐ์ (L. Dym and E. Levitt , 1991) การทำงานของระบบจะต้องอาศัยผู้เชี่ยวชาญรวบรวมความจริงและกฎของเรื่องนั้นๆ มาเป็นความรู้แล้วเก็บไว้ในรูปของฐานข้อมูลที่เรียกว่า ฐานความรู้ (Knowledge Base) ระบบฐานความรู้นี้สามารถกรองข้อมูลที่จำเป็นจากผู้ใช้โดยการถามคำถามที่เกี่ยวข้องกับ

ปัญหาในระหว่างการปรึกษา ระบบยังสามารถตอบคำถามที่ถูกถามจากผู้ใช้ว่า "ทำไมข้อมูลนี้จึงมีความจำเป็น" และสามารถอธิบายขั้นตอนการให้เหตุผลที่ใช้เพื่อให้ได้ข้อสรุปนั้น ความรู้ที่เก็บมีทั้งความรู้ที่เป็นความจริงที่อาจจะถูกบันทึกไว้ในรูปของตำรา หรือเอกสารทางวิชาการ และความรู้ที่ได้จากประสบการณ์ที่อาจจะไม่อยู่ในรูปของตำรา หรือเอกสารทางวิชาการ แต่จะต้องดึงออกมาจากผู้เชี่ยวชาญ หรือผู้ชำนาญที่มีประสบการณ์นั้น

2.3 ความแตกต่างระหว่างโปรแกรมแบบขั้นตอนดั้งเดิม และโปรแกรมระบบฐานความรู้

ความแตกต่างระหว่างโปรแกรมแบบขั้นตอนดั้งเดิม (Conventional algorithmic programming) และโปรแกรมระบบฐานความรู้ (Knowledge-Based system programming) ควรพูดถึงในลักษณะของการโปรแกรม ดังแสดงในตาราง 2.1 ในโปรแกรมแบบดั้งเดิมซึ่งมักเรียกว่า **โปรแกรมขั้นตอน** นั้นเราต้องบอกคอมพิวเตอร์ถึงสิ่งที่ต้องทำภายหลังที่ใส่ข้อมูล ดังนั้นในโปรแกรมขั้นตอนนั้นวิธีการเป็นการแสดงถึง เราต้องการให้ทำอย่างไร ในทางตรงกันข้ามโปรแกรมฐานความรู้นั้นเราแสดงความรู้ที่ชัดเจนเกี่ยวกับโดเมน นั่นคือ เป็นการแสดงถึง เรารู้อะไร โดยปราศจากการตัดสินใจล่วงหน้าว่าจะใช้ความรู้นั้นอย่างไร

ข้อแตกต่างระหว่างระบบฐานความรู้ และโปรแกรมแบบดั้งเดิมที่สำคัญข้อหนึ่ง คือ โปรแกรมแบบดั้งเดิมมีส่วนที่เป็นข้อมูลอยู่รวมกับส่วนที่เป็นโครงสร้างของโปรแกรม แต่สำหรับโปรแกรมระบบฐานความรู้นั้นส่วนของความรู้ที่ใช้ในการแก้ปัญหาจะอยู่แยกออกไปต่างหากจากส่วนที่เป็นการควบคุมซึ่งทำให้ระบบฐานความรู้มีความคล่องตัว และโปร่งใส เพราะส่วนที่เก็บความรู้สามารถตรวจสอบ แก้ไข ปรับปรุงได้ง่ายขึ้น ดังรูปที่ 2.2

2.4 โครงสร้างของระบบฐานความรู้

ส่วนประกอบของระบบฐานความรู้แสดงดังรูปที่ 2.3 ซึ่งรายละเอียดโดยย่อของส่วนประกอบแต่ละส่วนอธิบายได้ดังนี้

ก. **ฐานความรู้** (Knowledge Base) เป็นส่วนที่ใช้เก็บความรู้ทุกประเภทไม่ว่าจะเป็นความรู้ที่ได้จากตำรา หรือความรู้ที่ได้จากประสบการณ์

ข. **เครื่องอนุมาน** (Inference Engine) เป็นส่วนที่ควบคุมการใช้ความรู้ในฐานความรู้เพื่อแก้ไขปัญหาอย่างมีประสิทธิภาพ

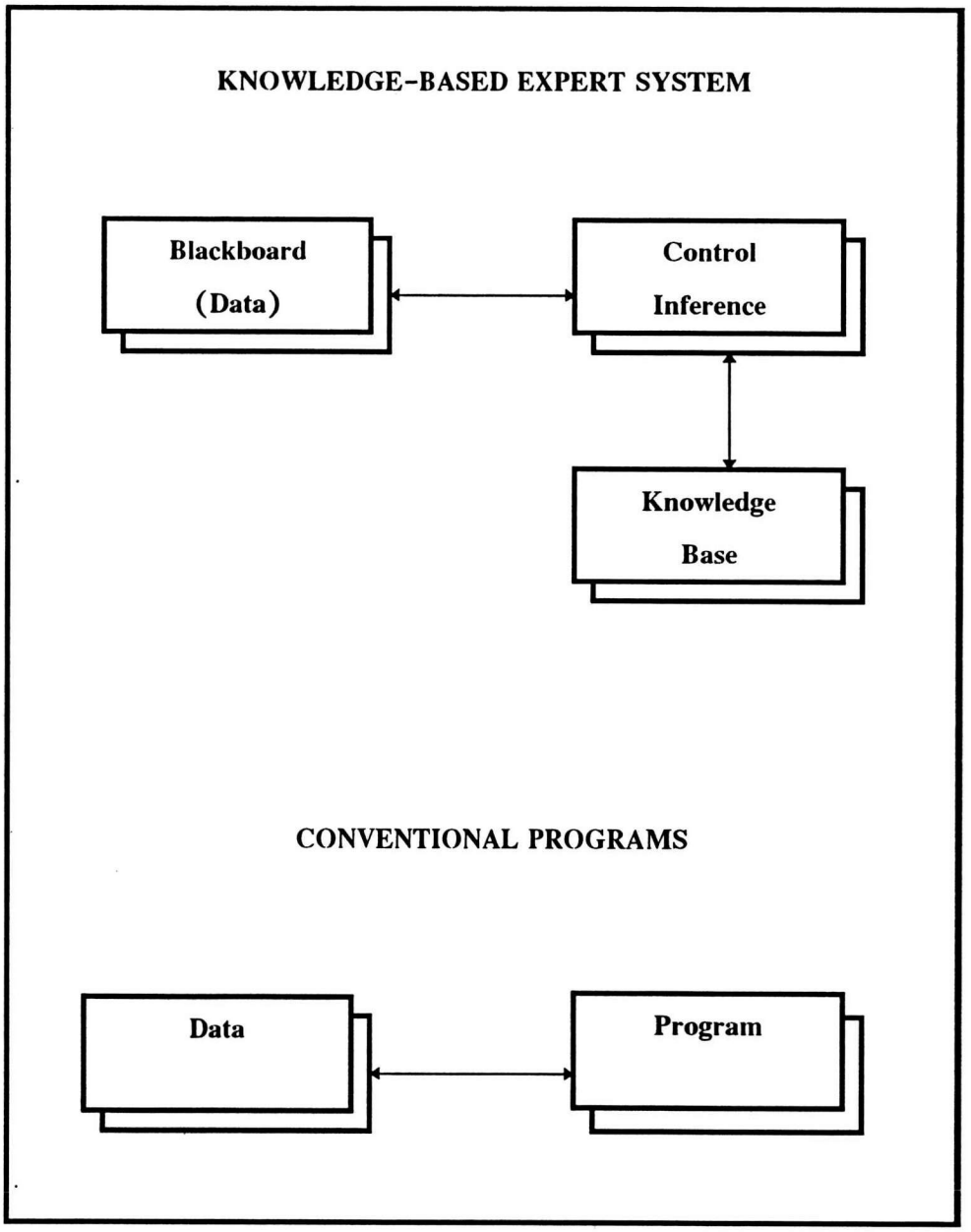
ค . ส่วนดึงความรู้ (Knowledge Acquisition Source) เป็นส่วนที่ช่วยในการดึงเอาความรู้จากตำราหรือฐานข้อมูล และจากผู้เชี่ยวชาญ

ง. ส่วนอธิบาย (Explanation Facility) ส่วนนี้ทำหน้าที่อธิบายรายละเอียดของขั้นตอนการวินิจฉัยต่อผู้ใช้ว่าข้อสรุป หรือคำตอบนั้นได้มาอย่างไร และทำไม

จ. ส่วนติดต่อกับผู้ใช้ (User Interface) เป็นส่วนที่เป็นตัวกลางระหว่างผู้ใช้กับระบบเพื่อทำให้การสื่อสารระหว่างผู้ใช้กับระบบเป็นไปได้อย่างราบรื่น และช่วยให้ผู้ใช้ยอมรับระบบมากขึ้น

ตาราง 2.1 เปรียบเทียบองค์ประกอบของโปรแกรมแบบดั้งเดิม (วิธีการ) และโปรแกรมฐานความรู้

โปรแกรมแบบดั้งเดิม	โปรแกรมฐานความรู้
มีการแสดง และใช้ข้อมูล	มีการแสดง และใช้ความรู้
ส่วนที่ควบคุมโปรแกรมอยู่รวมกับส่วนที่เป็นความรู้	ส่วนที่ควบคุมโปรแกรมแยกออกจากส่วนที่เป็นความรู้
กระบวนการที่เป็นวิธีการ หรือขั้นตอน	กระบวนการตัดสินใจ, คิดหาเหตุผล
กระบวนการทางตัวเลข	กระบวนการทางสัญญลักษณ์
การจัดการกับฐานข้อมูล	การจัดการกับฐานความรู้
ใช้กับกรณีที่ข้อมูลครบถ้วนสมบูรณ์ ไม่มีความขัดแย้ง	ใช้กับกรณีที่ข้อมูล หรือความรู้ไม่ครบถ้วนสมบูรณ์ หรืออาจมีความขัดแย้งได้
ไม่สามารถให้คำอธิบายระหว่างที่ปฏิบัติการ	การให้คำอธิบายระหว่างปฏิบัติการเป็นองค์ประกอบหนึ่งของระบบ



รูปที่ 2.2 ระบบผู้เชี่ยวชาญ และโปรแกรมแบบดั้งเดิม

ส่วนที่ใช้เก็บความรู้และส่วนที่ทำหน้าที่ในการแก้ปัญหาไม่จำเป็นที่จะต้องถูกสร้างขึ้นพร้อมกัน ด้วยเหตุผลดังกล่าวนี้ ในระบบผู้เชี่ยวชาญโดยทั่วไปจึงถูกออกแบบมาให้เป็นระบบที่มีเฉพาะส่วนที่เป็นเครื่องอนุมาน โดยที่เว้นฐานความรู้เอาไว้ให้ผู้ใช้สามารถพัฒนาต่อเอง เรียกระบบผู้เชี่ยวชาญที่มีลักษณะนี้ว่า เปลือกระบบผู้เชี่ยวชาญ (Expert System Shell) หรือเครื่องมือที่ใช้ในการพัฒนาระบบผูเชี่ยวชาญ (Experrt System Building Tool: ESBT) การพัฒนาระบบฐานความรู้โดยใช้เปลือกระบบผู้เชี่ยวชาญนี้ วิศวกรความรู้ (Knowledge Engineer: KE) จะเป็นผู้ที่นำความรู้จากผู้เชี่ยวชาญมนุษย์มาสร้างเป็นฐานความรู้ให้กับระบบฐานความรู้

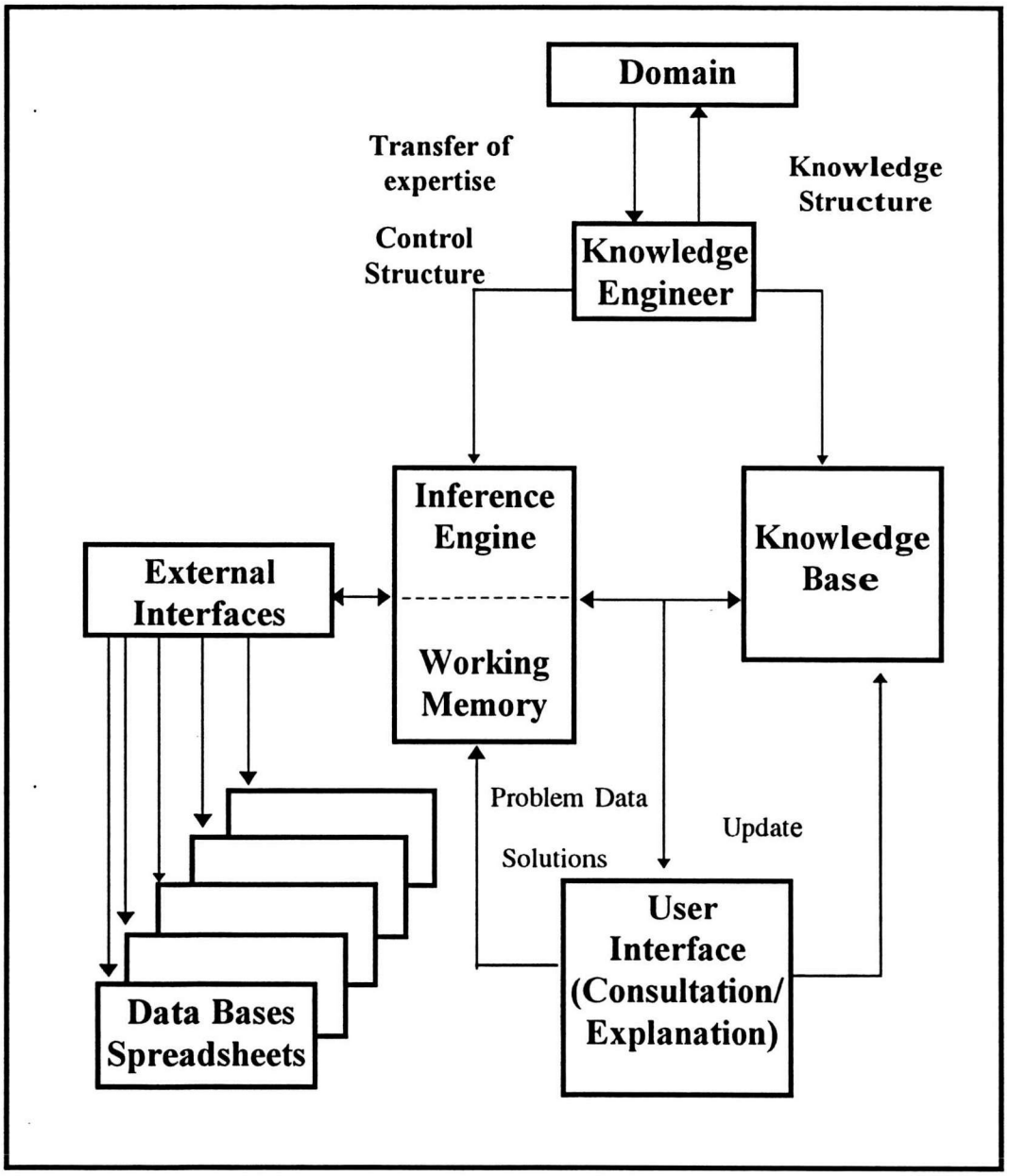
2.5 การแสดงความรู้ (Knowledge Representation)

มนุษย์เราเมื่อพบกับปัญหาที่ไม่เคยพบมาก่อน หรือปัญหาที่มีลักษณะพิเศษ ส่วนใหญ่แล้วจะสามารถปรับตัวสนองปัญหาเหล่านั้นได้ สาเหตุหลักที่ทำให้มนุษย์เรามีความสามารถเช่นนี้คงจะเป็นเพราะมนุษย์มีความรู้พื้นฐานที่เกี่ยวข้องทั้งโดยตรง และทางอ้อมกับปัญหาเหล่านั้นอยู่ ในทำนองคล้ายกันถ้าหากจะสร้างโปรแกรมที่มีความยืดหยุ่นสามารถแก้ปัญหาที่ยุ่งยากซับซ้อนได้ นอกจากโปรแกรมนั้นจะต้องประกอบด้วยอัลกอริธึมที่ดี และมีประสิทธิภาพแล้วยังจำเป็นจะต้องมี “ความรู้” ด้วย “ความรู้” นี้ใช้เป็นทั้งข้อมูลในการแก้ปัญหา และเป็นตัวชี้นำสำหรับอัลกอริธึมต่างๆ ความรู้ที่มนุษย์เรามีอยู่นั้นมีหลายรูปแบบ ถึงแม้จะไม่รู้ว่าความรู้เหล่านี้ถูกเก็บอยู่ในสมองในรูปโครงสร้างแบบใด แต่การที่จะทำให้คอมพิวเตอร์สามารถใช้ความรู้ได้ เราจำเป็นจะต้องบันทึกความรู้ในรูปแบบโครงสร้างใดโครงสร้างหนึ่งเข้าไปในคอมพิวเตอร์ปัญหานี้เรียกว่า การแสดงความรู้ การแสดงความรู้นี้เป็นหัวใจสำคัญของการสร้างระบบความรู้ และมีความสัมพันธ์อย่างลึกซึ้งกับการอนุมาน, การรับเอา และการจัดการความรู้ ความรู้ถึงแม้จะมีหลายรูปแบบ แต่พอจะแยกออกเป็นสามประเภทใหญ่ๆ ได้ดังนี้

ก. ความรู้ที่บอกความจริง, ความสัมพันธ์, ลักษณะ หรือคุณสมบัติ (Declarative Knowledge)

ข. ความรู้ที่บอกขั้นตอนหรือวิธีการ (Task-level Knowledge)

ค. ความรู้ที่เกี่ยวกับความรู้ (Meta-level Knowledge)



รูปที่ 2.3 ส่วนประกอบระบบผู้เชี่ยวชาญ

ในการใส่ความรู้ให้กับระบบผู้เชี่ยวชาญนั้น วิศวกรความรู้จะเป็นผู้ที่นำความรู้มาแสดงลงในฐานความรู้ซึ่งเรียกว่า การแสดงความรู้ (Knowledge Representation) การแสดงความรู้นั้นมีอยู่หลายแบบแต่จะขอกล่าวถึงเฉพาะที่ใช้ในระบบฐานความรู้นี้ ได้แก่

2.5.1 การแสดงความรู้โดยกฎ (IF - THEN Rules) มีชื่อเรียกอีกอย่างหนึ่งว่า การแสดงความรู้ในรูประบบการผลิต (Production System) มีหลักเกณฑ์พื้นฐานง่ายๆ คืออาศัยรูป

ประโยคของ IF (*ส่วนเงื่อนไข*) THEN (*ส่วนข้อสรุป หรือส่วนปฏิบัติการ*) โครงสร้าง IF-THEN ของกฎเชื่อมระหว่างคู่ของวัตถุ (Objects) กับคุณลักษณะ (Attributes) ดังแสดงข้างล่าง

IF สถานการณ์ (Situation) THEN กระทำ (Actions)

IF สถานบท (Premise) THEN ข้อสรุป (Conclusions)

IF อินพุท (Input) THEN เอาท์พุท (Output)

IF ข้อนำ (Antecedent) THEN ข้อตาม (Consequent)

สำหรับกรณีที่มีกฎมากกว่าหนึ่งกฎสามารถนำมารวมกันโดยใช้ **AND หรือ OR**

ตัวอย่าง การแสดงความรู้โดยกฎ :

IF แดดออก AND ท้องฟ้าสีฟ้า THEN ไปเที่ยวทะเล

IF แดดออก OR ท้องฟ้าสีฟ้า THEN ไปเที่ยวทะเล

IF ฝนตก AND ท้องฟ้าสีดำ THEN อยู่กับบ้าน

2.5.2 การแสดงความรู้โดยเฟรม (Frame) **หรือออปเจกต์** (Object) ในเฟรมจะมีการบันทึกข้อมูลเกี่ยวกับสภาพเหตุการณ์ วัตถุ หรือความคิด และการบันทึกความสัมพันธ์ต่างระดับระหว่างสิ่งต่างๆ เหล่านั้น ซึ่งโครงสร้างข้อมูลเป็นดังนี้

< Frame> : : = <Frame Name>

<Slot Name> <Slot Name>

.....

<Slot Name> <Slot Value>

ความสัมพันธ์ต่างระดับระหว่างเฟรมนั้นสามารถ*ถ่ายทอดคุณสมบัติ* กล่าวคือ เฟรมที่อยู่ระดับต่ำกว่าจะมีคุณสมบัติของเฟรมที่อยู่ระดับสูงกว่า

ตามความเป็นจริงแล้วไม่มีการแสดงความรู้ชนิดใดเลยที่กล่าวมาข้างต้นที่สามารถจัดการกับปัญหาของโลกแห่งความจริงที่ซับซ้อนนี้ได้อย่างน่าพอใจทั้งหมด ด้วยเหตุนี้ จึงมีผู้คิดค้นที่จะพยายามรวมวิธีการแสดงความรู้เข้าด้วยกันให้อยู่ในรูปแบบของ การแสดงความรู้แบบไฮบริด ซึ่งเป็นการขจัดข้อเสียเปรียบของวิธีการแสดงความรู้แต่ละวิธี

2.6 การหาเหตุผลหรือการแก้ปัญหาโดยระบบฐานความรู้

2.6.1 โครงสร้างของระบบการผลิต ระบบการผลิต (Production System) จะประกอบด้วยระบบใหญ่ 3 ส่วน คือ

ก. **ฐานกฎ (Rule Base)** หรือ Production Memory มีหน้าที่เก็บความรู้ที่อยู่ในรูปของกฎ

ข. **ส่วนตีความ (Interpreter)** หรือส่วนอนุมาน **(Inference)** เป็นส่วนที่ทำหน้าที่ในการควบคุมการกระทำ (Action) ของกฎ

ค. **Working Memory** เป็นส่วนที่ทำหน้าที่ในการเก็บความจริงที่ได้มีการอนุมานมาก่อนหน้า

ในการปฏิบัติแต่ละครั้งจะประกอบด้วยวงจรดังต่อไปนี้

ก. Pattern Matching : เป็นการทำการเปรียบเทียบ

ข. Conflict Resolution: เป็นส่วนที่ทำหน้าที่ในการเลือกกฎ

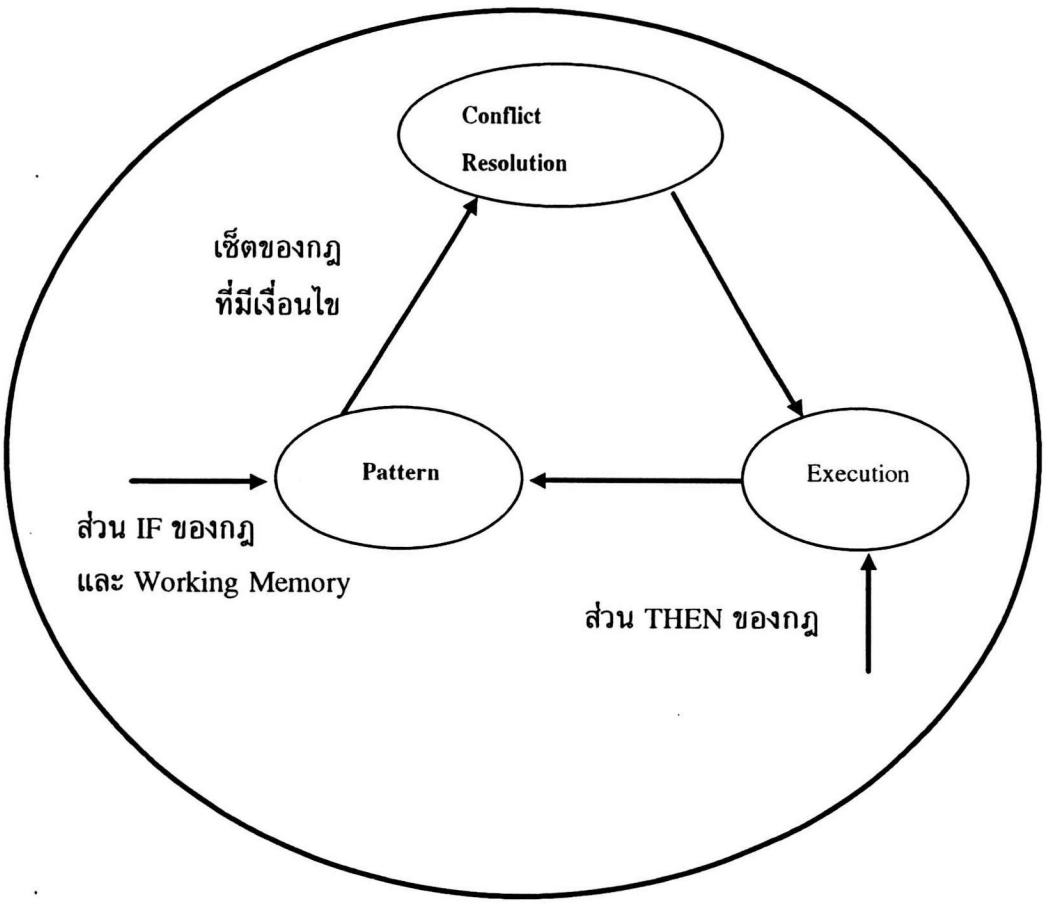
ค. Execution: เป็นส่วนที่ทำหน้าที่ตามการสั่งการของกฎข้อที่ได้เลือกมาแล้ว

วงจรปฏิบัติการของระบบการผลิตนี้ แสดงดังรูปที่ 2.4 ลักษณะของการตีความนี้จะเริ่มด้วยการเปรียบเทียบในส่วนที่เป็นดาต้าเซ็ตใน Working Memory กับฐานความรู้ที่อยู่ในฐานกฎ ในกรณีนี้จะเป็นไปได้ที่มีกฎหลายข้อที่เมื่อเปรียบเทียบแล้วตรงกัน ในกรณีเช่นนี้ส่วนของ Conflict Resolution จะทำหน้าที่ในการเลือกกฎข้อที่ถูกต้องซึ่งในการเลือกกฎนี้จะมีกระบวนการของการอนุมานแบบเดินหน้า ซึ่งเป็นการหาเหตุผลจากความจริง (Fact) ไปหาเป้าหมาย (Goal) และการอนุมานแบบย้อนกลับ ซึ่งเป็นการหาเหตุผลจากเป้าหมายไปสู่ความจริง เมื่อได้กฎข้อที่ต้องการแล้วก็เอ็กซ์คิวต์ (Execute) เมื่อเอ็กซ์คิวต์เสร็จแล้วก็จะวนกลับไปทำ Pattern Matching ใหม่และจะวนไปเรื่อยๆ จนกระทั่งได้คำตอบที่แท้จริง ซึ่งกระบวนการดังกล่าวมาทั้งหมดนี้ เรียกว่า **การอนุมาน (Inference)**

2.6.2 การอนุมาน (Inference)

การอนุมาน คือ กระบวนการในการค้นหาความจริงจากความจริงที่มีอยู่แล้ว (Known Fact) ในคลังความรู้หรือความจริงที่สามารถหาได้จากผู้ใช้ ในการอนุมานของระบบฐานความรู้นั้นต้องอาศัยเครื่องอนุมาน (Inference Engine) ซึ่งเป็นส่วนของโปรแกรมในระบบฐานความรู้ที่ทำหน้าที่ดังกล่าว นอกจากนั้นเครื่องอนุมานยังทำหน้าที่ในการควบคุมการทำงานของระบบฐานความรู้ด้วย โดยปกติเครื่องอนุมานจะมีหน้าที่หลักคือการกำหนดทิศทางในการหาเหตุผลโดยมีหน่วยควบคุมเป็นผู้ทำหน้าที่โดยตรง

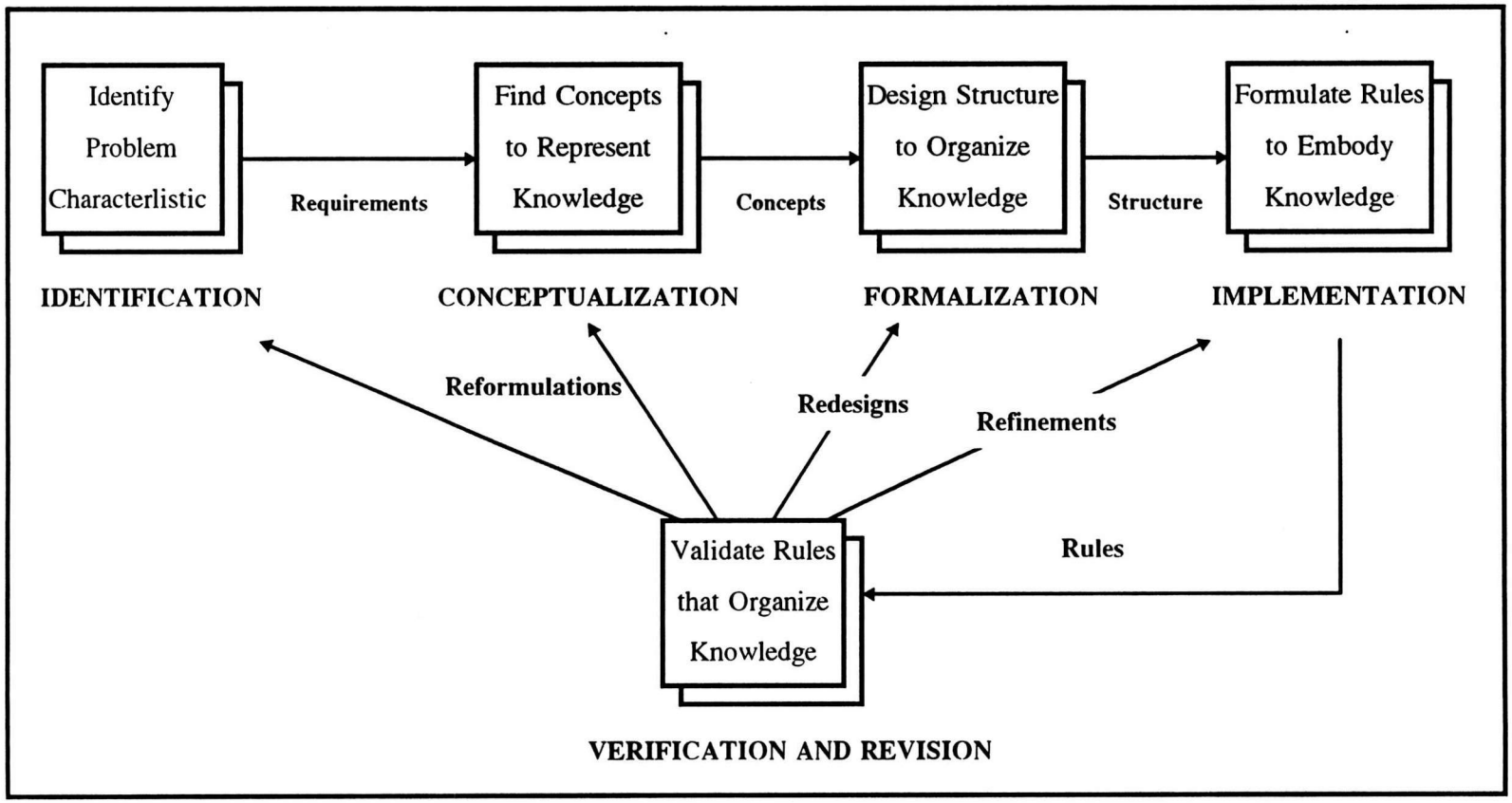
ระบบฐานความรู้ด้วย โดยปกติเครื่องอนุมานจะมีหน้าที่หลักคือการกำหนดทิศทางในการหาเหตุผลโดยมีหน่วยควบคุมเป็นผู้ทำหน้าที่โดยตรง



รูปที่ 2.4 วงจรปฏิบัติการของระบบการผลิต (Production System)

การหาเหตุผลของเครื่องอนุมานสามารถแบ่งออกเป็นชนิดใหญ่ๆ ได้ดังนี้

ก. **การอนุมานแบบเดินหน้า (Forward Chaining)**: จากรูปที่ 2.4 แสดงวงจรการปฏิบัติการของระบบการผลิตภายใต้การอนุมานแบบเดินหน้า กล่าวคือ ระบบการผลิต (Production System) จะเริ่มต้นจาก Working Memory ทำการค้นหากฎที่มีเงื่อนไขครบ แล้วจึงปฏิบัติการตามกฎนั้น ระบบการผลิตจะปฏิบัติการซ้ำๆกันเช่นนี้จนกว่าจะได้คำตอบ หรือ



รูปที่ 2.5 แผนภาพขั้นตอนในวิศวกรรมความรู้

ในกรณีที่ค้นไม่พบข้อมูลดังกล่าว อาจจะต้องทำการย้อนรอย (Back Tracking) และลองเปลี่ยนเป้าหมายย่อยระหว่างทางเสียใหม่ กานอนุมานแบบนี้เรียกอีกชื่อหนึ่งว่า **Goal - Driving Inference หรือ Top - Down Inference**

ค. การหาแหตุผลโดยนิรนัย (Deductive Reasoning) เป็นกระบวนการหาเหตุผลจากข้อมูลโดยทั่วไปที่เกี่ยวกับคลาส (Class) ของออบเจ็คต์ (Object) หรือเหตุการณ์ (Events) ไปสู่ข้อมูลเฉพาะที่เกี่ยวกับสมาชิกของคลาส (Class)

ง. การหาเหตุผลโดยอุปนัย (Inductive Reasoning) เป็นการหาเหตุผลจากข้อสรุปที่ขึ้นกับข้อเท็จจริงจำเพาะ ตัวอย่างเช่น ข้อมูลจำเพาะเกี่ยวกับสมาชิกแต่ละตัวของคลาส หรือเหตุการณ์ที่นำไปสู่การค้นหาเกี่ยวกับคลาสทั้งหมด

จ. การหาเหตุผลแบบโมโนโทนิก (Monotonic Reasoning) เป็นระบบของการหาเหตุผลที่อยู่บนพื้นฐานที่ว่าเมื่อความจริงใดๆ ที่ได้รับการพิจารณาแล้ว ความจริงนั้นๆ จะไม่สามารถเปลี่ยนแปลงไปได้ตลอดของการหาเหตุผลครั้งนั้น หรืออาจกล่าวได้ว่า จำนวนความจริงที่ได้รับการพิสูจน์แล้วว่าถูกต้อง ณ เวลาหนึ่งๆ จะมีค่าของความน่าเชื่อถือเพิ่มขึ้นอยู่เสมอ และจะไม่ลดลง

ฉ. การหาเหตุผลแบบนอนโมโนโทนิก (Nonmonotonic Reasoning) การหาเหตุผลแบบนี้จะตรงข้ามกับการหาเหตุผลแบบโมโนโทนิก คือความจริงใดที่ได้รับการสรุปมาก่อนหน้าแล้ว สามารถถูกตรวจสอบแก้ไขได้ ความจริงนั้นอาจจะไม่เป็นความจริงไปตลอดช่วงของการอนุมานก็ได้

2.7 การพัฒนาระบบฐานความรู้

การพัฒนาระบบฐานความรู้นั้นมีขั้นตอนต่างๆ ที่มากมายและซับซ้อน ถ้าหากว่าการพัฒนาระบบฐานความรู้จะเริ่มต้นที่การพัฒนาระบบซอฟต์แวร์ ซึ่งประกอบด้วยเครื่องอนุมานฐานความรู้ หน่วยติดต่อกับผู้ใช้และอื่นๆ ผู้พัฒนาระบบจะต้องเสียเวลามากในการออกแบบเขียนโปรแกรมและสร้างความรู้ให้กับระบบ ปัจจุบันเทคโนโลยีทางด้านซอฟต์แวร์ได้ทำให้การพัฒนาระบบฐานความรู้ง่ายขึ้น ระบบฐานความรู้ถูกสร้างให้มีเฉพาะโครงภายนอกที่เรียกว่า เปลือกระบบฐานความรู้ เปลือกระบบฐานความรู้นี้จะเป็นระบบที่สามารถพัฒนาฐานความรู้ทั่วไปทีหลังตามความต้องการของผู้ใช้ได้

กระบวนการและกิจกรรมของการสร้างระบบฐานความรู้โดยทั่วไปเรียกว่า วิศวกรรมความรู้ (Knowledge Engineering) ส่วนผู้ที่ทำหน้าที่ในการพัฒนาระบบฐานความรู้ เรียกว่า วิศวกรความรู้ (Knowledge Engineer : KE) ซึ่งมีหน้าที่ในการสร้างระบบฐานความรู้ โดยทำการศึกษา และนำความรู้จากแหล่งต่างๆ เช่น หนังสือ หรือความรู้จากผู้เชี่ยวชาญที่เป็นมนุษย์มาทำการจัดเป็นระบบ และสร้างให้อยู่ในรูปแบบที่เหมาะสมเป็นฐานความรู้ให้กับระบบฐานความรู้ โดยใช้เครื่องมือช่วยพัฒนาระบบฐานความรู้ หน้าที่ที่สำคัญอีกประการหนึ่งของวิศวกรความรู้ก็คือ การตรวจสอบและบำรุงรักษาระบบ การตรวจสอบนั้นเป็นการทดสอบการทำงานของระบบว่าการทำงานเป็นไปตามจุดประสงค์หรือไม่ สำหรับการบำรุงรักษาก็คือ การปรับปรุงฐานความรู้ให้ทันสมัยขึ้นตามกาลเวลาที่เปลี่ยนแปลงไป การสร้างระบบฐานความรู้ให้สำเร็จถือว่าเป็นงานที่ยากที่สุดงานหนึ่ง เพราะกระบวนการระหว่างความรู้ภายนอกซึ่งผู้เชี่ยวชาญมนุษย์มีอยู่ และความรู้ภายในซึ่งระบบฐานความรู้ต้องมีนั้นเป็นกระบวนการแบบไดนามิก และเป็นการเปลี่ยนแปลงที่เกิดซ้ำๆ กัน รูปที่ 2.5 เป็นภาพแนวคิดของวิศวกรรมความรู้ที่เริ่มต้นด้วยขั้นตอนการวิเคราะห์ปัญหา, การหาแนวคิด, การออกแบบระบบ, การสร้างระบบ, การทดสอบและการปรับปรุงระบบ รายละเอียดของแต่ละขั้นตอนจะกล่าวถึงโดยลำดับ ดังต่อไปนี้

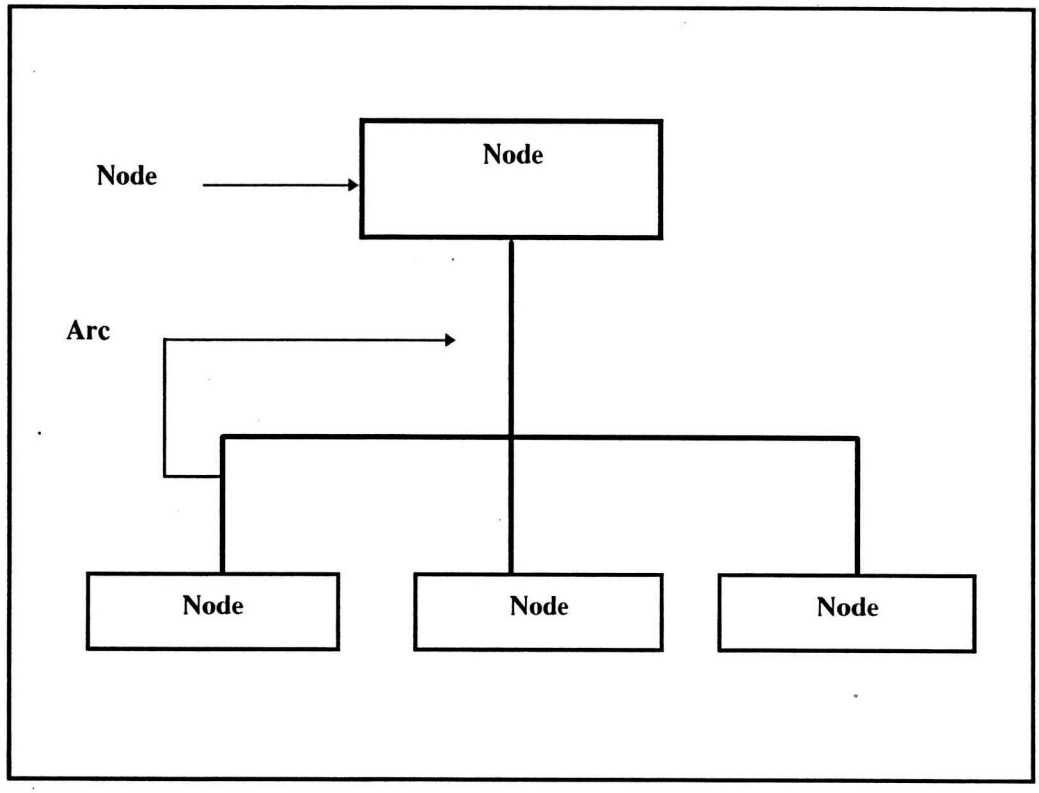
2.7.1 การวิเคราะห์ปัญหา (Identification Stage)

การวิเคราะห์ปัญหาของระบบนั้น ผู้พัฒนาระบบจะต้องทำความเข้าใจเบื้องต้นกับปัญหาเหล่านั้น และจะต้องมีการเตรียมการสำหรับการแก้ปัญหานั้นด้วย เรื่องที่สำคัญสำหรับการวิเคราะห์ปัญหาที่จะต้องพิจารณาก็คือ ความจำเป็นหรือความเหมาะสมของการใช้ระบบฐานความรู้ การมีความรู้ความเข้าใจปัญหาอย่างถูกต้อง การจัดขั้นตอนสำหรับการแก้ปัญหาและการกำหนดรูปแบบของการให้คำปรึกษา

2.7.2 การหาแนวคิด (Conceptualization Stage)

หลังจากที่เข้าใจขอบเขตของปัญหาแล้วขั้นต่อไปก็คือการหาแนวคิด, ความสัมพันธ์และกลไกการควบคุมที่อธิบายถึงความรู้ในการแก้ปัญหาของผู้เชี่ยวชาญ และกลยุทธ์ที่ใช้ในโดเมน ในขั้นตอนนี้วิศวกรความรู้ควรพิจารณาเกี่ยวกับการแสดงความรู้ และวิธีการอนุมานซึ่งจะต้องเหมาะกับเครื่องมือที่ช่วยในการพัฒนาระบบด้วย

โดเมน ในขั้นตอนนี้วิศวกรความรู้ควรพิจารณาเกี่ยวกับการแสดงความรู้ และวิธีการอนุมาน ซึ่งจะต้องเหมาะกับเครื่องมือที่ช่วยในการพัฒนาระบบด้วย



รูปที่ 2.6 แผนภาพแสดงโครงสร้างต้นไม้

2.7.3 การออกแบบระบบ (Formalization Stage)

การออกแบบระบบเป็นขั้นตอนหลังจากการเขียนแนวความคิดของความรู้ทั้งหมดที่เราจะสร้างโดยเริ่มต้นจากเป้าหมายซึ่งก็คือคำตอบของการให้คำปรึกษานั่นเอง และคำตอบนี้จะมีอยู่หลายๆ คำตอบ ซึ่งระบบผู้เชี่ยวชาญฐานความรู้จะเป็นผู้เลือกให้สอดคล้องกับลักษณะเฉพาะของปัญหา วิธีการหนึ่งที่จะสามารถช่วยในการจัดขั้นตอนในการแก้ปัญหาได้ดีคือ การจัดความรู้ในรูปแบบของต้นไม้ (Tree) องค์ประกอบของโครงสร้างแบบต้นไม้จะประกอบด้วยโนด (Node) และอาร์ก (Arc) สำหรับโนดจะแทนความหมายที่จะแสดงในฐานความรู้ และอาร์กจะเป็นส่วนที่เชื่อมความสัมพันธ์ของโนดต่างๆ ที่เกี่ยวพันกันเข้าด้วยกัน ซึ่งแสดงดังรูปที่ 2.6 ในส่วนของความสัมพันธ์ระหว่างโนดจะกำหนดให้โนดตัวที่อยู่ในระดับที่สูงกว่า

2.7.4 การสร้างต้นแบบ (Implementation Stage)

ในการสร้างต้นแบบนั้นจะมีการแสดงความรู้เฉพาะตอนขึ้นมา โดยการจำกัดโดเมนของความรู้ให้แคบลง การสร้างระบบต้นแบบนี้มีจุดประสงค์เพื่อหาความเป็นไปได้ของการสร้างระบบ และหาหนทางในการแก้ปัญหาก่อนที่จะสร้างระบบจริง ระบบต้นแบบนี้จะเป็นต้นแบบเพื่อใช้ในการทดสอบว่าการแก้ปัญหาที่ได้ทำการออกแบบมานั้นถูกต้องหรือไม่

2.7.5 การทดสอบ และปรับปรุงระบบ (Verification and Revision Stage)

ทำการตรวจสอบระบบโดยผู้เชี่ยวชาญ และวิศวกรความรู้อย่างละเอียด โดยการนำเงื่อนไขต่างๆ ที่ได้วางไว้ในการสร้างระบบต้นแบบมาทำการทดสอบ และตรวจโดยผู้เชี่ยวชาญเพื่อดูว่าเงื่อนไขต่างๆ ที่ทดสอบนั้นถูกต้องหรือไม่ ถ้าหากว่าระบบต้นแบบมีความคลาดเคลื่อนจากที่วางระบบเอาไว้ก็จะต้องวกกลับไปทำการออกแบบระบบต้นแบบใหม่ วิศวกรความรู้จะต้องตรวจสอบเงื่อนไขต่างๆ ของการอนุมานให้ครบถ้วน และผู้เชี่ยวชาญจะต้องตรวจสอบความรู้ทุกอย่างที่มีอยู่ในระบบว่าตรงกับตวามเป็นจริงหรือไม่ ถ้าหากเกิดความผิดพลาดขึ้นวิศวกรความรู้จะต้องเป็นผู้แก้ไขกฎหรือข้อมูลต่างๆ ในฐานความรู้

2.8 เครื่องมือช่วยพัฒนาระบบฐานความรู้

เครื่องมือที่ช่วยพัฒนาระบบฐานความรู้ (Knowledge-Based System Building Tools) หรือที่เรียกกันทั่วไปว่า **เปลือกระบบฐานความรู้** จะมีลักษณะพิเศษในการใช้งานที่ต่างกัน ทั้งนี้ขึ้นอยู่กับวิธีการที่ใช้ในการสร้างเครื่องมือเหล่านี้ สำหรับงานวิจัยนี้ใช้ **Smart Element Version 2.0** เป็นเครื่องมือในการพัฒนาระบบฐานความรู้

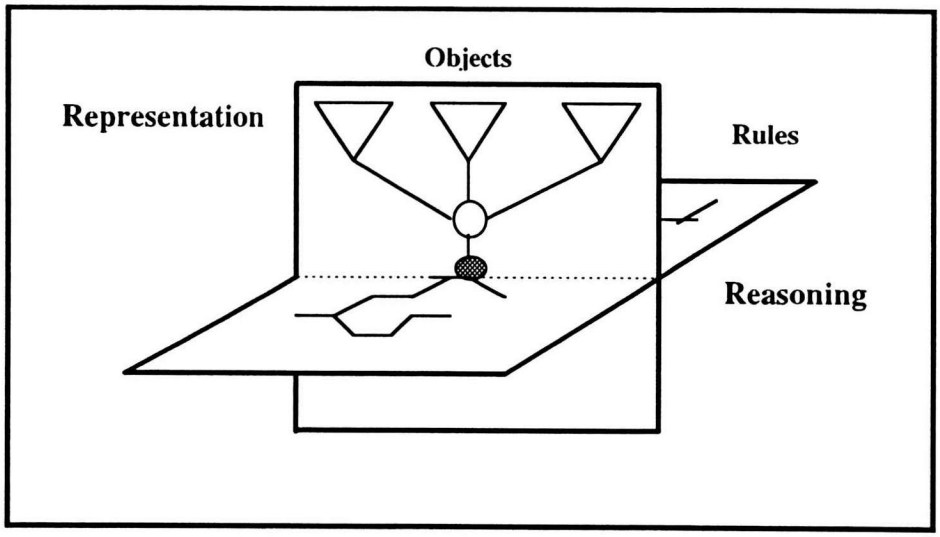
Smart Element Version 2.0 เป็นเปลือกระบบฐานความรู้ที่ใช้งานบนไมโครคอมพิวเตอร์ที่ถูกออกแบบให้มีองค์ประกอบเชิงวัตถุและกฎรวมกับการติดต่อกับผู้ใช้เชิงภาพ (Graphical User Interface; GUI) องค์ประกอบทั้งสองนี้ติดต่อกันโดยภาษาสคริพต์ (Script Language) ซึ่งทำให้ผู้พัฒนาระบบสามารถกำหนดคุณลักษณะของการติดต่อและเชื่อมองค์ประกอบทางจอภาพไปยังวัตถุ Smart Element นี้เป็นระบบไฮบริด (Hybrid System) ดังรูปที่ 2.7 นั่นคือมีความสามารถรองรับได้ทั้งระบบการหาเหตุผล (Reasoning System) และการแสดงเชิงออบเจกต์ (Object-Oriented Representation) ซึ่งคุณลักษณะของระบบในแง่ต่างๆ ของทั้งสองด้านนี้อธิบายได้ดังต่อไปนี้

2.8.1 การแสดงความรู้ใน Smart Element (Knowledge Representation in Smart Element)

Smart Element มีลักษณะเด่นของการแสดงความรู้มากมาย การแสดงความรู้ใน Smart Element มีสองแบบที่สำคัญ คืออออปเจกต์และกฎ โดเมนของการแสดงความรู้จำลองให้อยู่ในเทอมของออบเจกต์, คลาส และสมบัติ โดยสมบัติเฉพาะของออปเจกต์และคลาส เรียกว่าสล็อท (Slot) สล็อทนี้ใช้เก็บข้อมูลทั้งหมด และสล็อทมีคุณลักษณะของเมตาสล็อทซึ่งอธิบายค่าลักษณะเฉพาะของสล็อท นอกจากนี้ Smart Element ยังมีกฎซึ่งประกอบด้วยโดเมนความรู้ทั้งหมด กฎควบคุมสล็อทได้ดีพอๆ กับโครงสร้างของออปเจกต์ และคลาส การเปรียบเทียบแบบอย่าง (Pattern Matching) และการตีความ (Interpretations) ทำให้สามารถอ้างถึงออปเจกต์ที่ถูกพิจารณาในระหว่างการปฏิบัติการ การเชื่อมกับกฎนั้น Smart Element มีระเบียบวิธี (Method) และการส่งผ่านข้อความ (Massing Passing) เพื่อกำหนดฮิวริสติกที่เป็นเชิงออปเจกต์ในธรรมชาติ นอกจากนี้ Smart Element ยังมีการสืบทอดได้หลายอย่าง เช่น สมบัติ, ค่า, วิธีการ และเมตาสล็อท ซึ่งสามารถสืบทอดลงมาตามลำดับออปเจกต์ ส่วนการสืบทอดขึ้นมาตามลำดับออเจกต์นั้นใช้ได้เฉพาะกับสมบัติและค่าเท่านั้น การแสดงความรู้ใน Smart Element พอจะกล่าวโดยสังเขป ได้ดังนี้

ก. **โครงสร้างกฎ (Rule Structure)** Smart Element ใช้กฎในการแสดงการหาเหตุผล ซึ่งจะทำการหาเหตุผลบนลำดับขั้นของออปเจกต์ กฎเก็บความรู้ที่จำเป็นในการแก้ปัญหาเฉพาะและแสดงถึงสิ่งต่างๆ ดังนี้ ความสัมพันธ์, ฮิวริสติก, ความรู้ที่เป็นวิธีการดำเนินการ และโครงสร้างของความรู้ตามเวลา ตามรูปที่ 2.7 แสดงระนาบของความรู้ที่เป็นโครงสร้างกฎ รูปแบบของกฎที่ใช้ใน Smart Element เป็นดังนี้

IF (*conditions*) **THEN** (*hypothesis*) **THEN DO** (*actions*) or **ELSE DO** (*actions*)



รูปที่ 2.7 แสดงระนาบของกฎและออปเจกต์ซึ่งเป็นระบบไฮบริด

ไดอะแกรมแสดงกฎและส่วนประกอบของกฎใน Smart Element แสดงดังรูปที่ 2.8 รูปสี่เหลี่ยมผืนผ้าด้านซ้ายมือที่แบ่งด้วยเส้นแนวนอนแสดงส่วนซ้ายมือของกฎ (Left-Hand Side; LHS) เป็นส่วนเงื่อนไข (Conditions) และรูปสี่เหลี่ยมผืนผ้าด้านขวามือที่แบ่งออกเป็นสองส่วน คือส่วนขวามือของกฎ (Right-Hand Side; RHS) ซึ่งเป็นส่วนสมมติฐาน (Hypothesis) และ ส่วนการกระทำ (Actions) ของกฎ

ข. โครงสร้างออปเจกต์ (Object Structure) ; แสดงความรู้ที่เป็นเหตุผลโดยกฎ โครงสร้างของคลาส และออปเจกต์ แสดงดังรูปที่ 2.9 และความหมายของส่วนประกอบโครงสร้างออปเจกต์เป็นดังนี้

ออบเจกต์ (Object) เป็นส่วนที่อธิบายตัวแปรในฐานความรู้ สิ่งใดๆ ก็สามารถเป็นออบเจกต์ได้

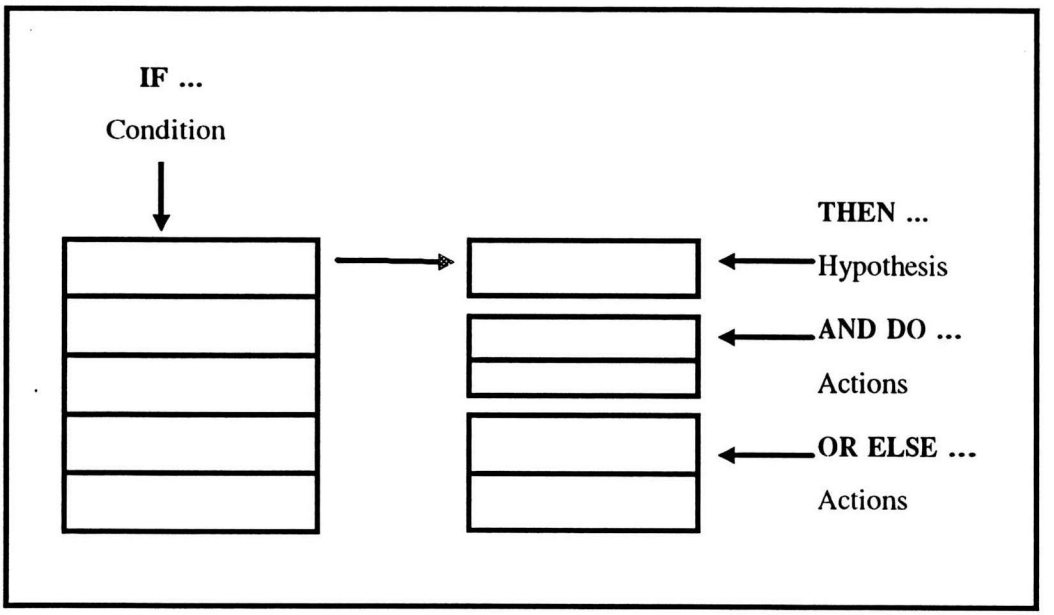
สมบัติ (Property) เป็นลักษณะเฉพาะซึ่งสามารถใช้ร่วมกับออปเจกต์ หรือคลาส

คลาส (Class) เป็นการรวบรวมออปเจกต์ซึ่งมักจะมีการใช้สมบัติร่วมกัน

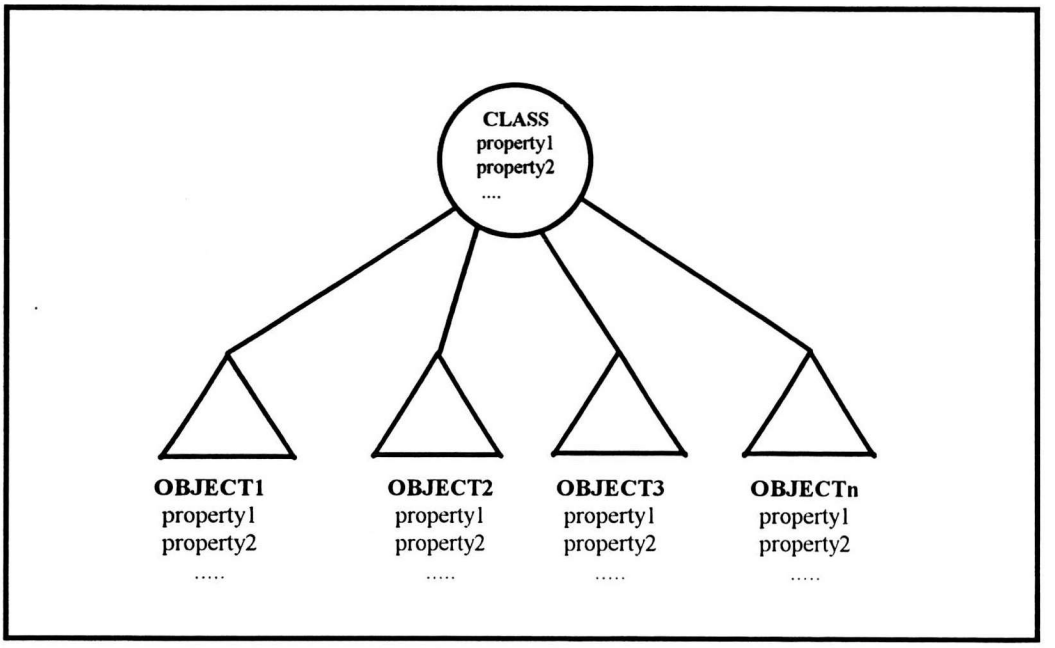
ค. วิธีการ (Methods) อธิบายพฤติกรรมเฉพาะของสล็อท, ออปเจกต์ หรือเซ็ตของออปเจกต์

ง. การสืบทอด (Inheritance) กลไกการแสดงความรู้ต่างๆ ที่กล่าวมาข้างต้นเป็นประโยชน์ในแง่ของโครงสร้างแต่การสืบทอดเป็นสิ่งที่ให้อรรถประโยชน์ต่อรูปแบบของการแสดง การสืบทอดมี 3 แบบซึ่งอยู่ภายใต้การควบคุมของผู้พัฒนาระบบ ได้แก่ การสืบทอด

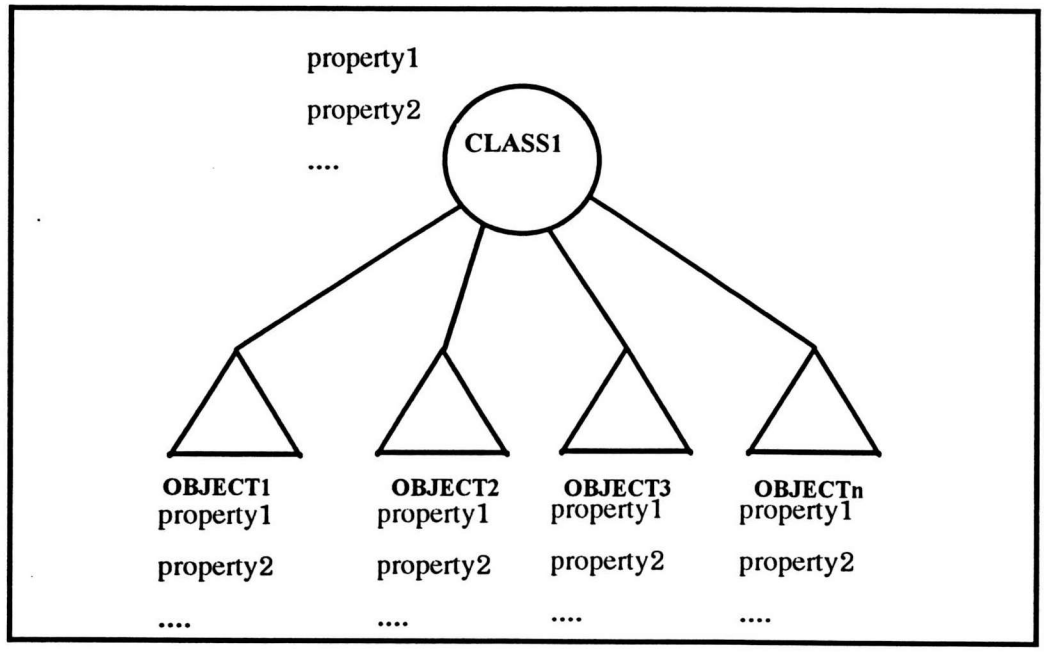
สมบัติ, การสืบทอดค่า และการสืบทอดวิธีการ รูปที่ 2.10 แสดงการสืบทอดสมบัติของคลาสโดยออปเจกต์



รูปที่ 2.8 แสดงส่วนประกอบของกฎ



รูปที่ 2.9 โครงสร้างคลาสและออบเจกต์



รูปที่ 2.10 การสืบทอดสมบัติโดยออปเจกต์ของคลาสหนึ่ง

จ. โครงสร้างไดนามิก (Dynamic Structures) กลไกการแสดงที่กล่าวมาข้างต้นกำหนดสภาวะมากมายในการอธิบายระบบที่ซึ่งนิยามความสัมพันธ์ทั้งหมดที่ชัดเจนเมื่อสร้างระบบฐานความรู้ แต่จะเกิดอะไรขึ้นถ้าออบเจ็คต์และความสัมพันธ์ที่มีอยู่นั้นไม่ทราบลำดับความสำคัญ ในการแก้ปัญหานี้ระบบจึงต้องมีการสร้างไดนามิกออบเจ็คต์ และการเชื่อมไดนามิก ในที่นี้ออปเจกต์สามารถสร้างได้โดยใช้ตัวดำเนินการ CreateObject ที่ปรากฏในกฎหรือวิธีการ (Method)

ฉ. การตีความ (Interpretations) บ่อยครั้งที่สล็อทที่ต้องการทดสอบในกฎหรือภาวะของวิธีการที่ส่งไปยังกิจวัตรเฉพาะ หรือส่งไปยังฟังก์ชันนั้นไม่ทราบค่าก่อนที่จะทำการอนุมาน ในสถานะเช่นนี้จึงมีความจำเป็นต้องสร้างออปเจกต์, คลาส และสมบัติในขณะปฏิบัติการมากกว่าที่จะกำหนดในกฎ

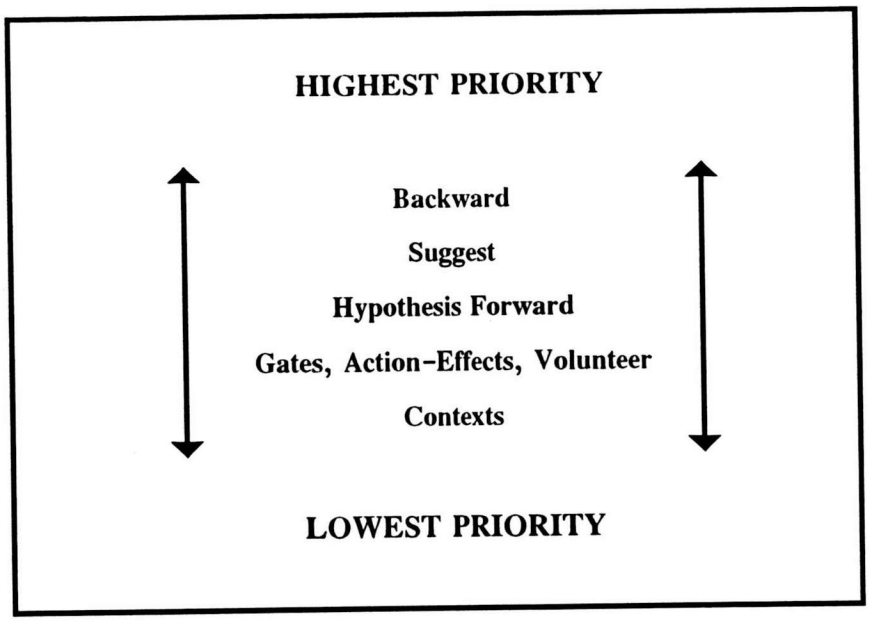
ช. การเปรียบเทียบแบบอย่าง (Pattern Matching) เป็นวิธีการหนึ่งในการทดสอบค่าของสล็อทโดยไม่ต้องกำหนดโดยตรง การเปรียบเทียบแบบอย่างนี้จะสร้างออปเจกต์ที่เป็นของคลาสหรือออปเจกต์แม่ (Parent Class or Parent Object)

2.8.2 การกระบวนการเครื่องอนุมาน (Inference Engine Processing)

กลไกที่ทำการกำหนดรายเหตุการณ์ที่เกิดในระหว่างการกระบวนการระบบคือระเบียบวาระ (Agenda) ข้อแตกต่างระหว่างโปรแกรมแบบดั้งเดิม (Classic Programming) และโปรแกรมฐานระเบียบวาระ (Agende-Based Programming) คือ โปรแกรมแบบดั้งเดิมนั้นโปรแกรมถูกเอ็คซิคิวท์ด้วยขั้นตอนวิธีในขณะที่โปรแกรมฐานระเบียบวาระสามารถแก้ไขรายเหตุการณ์โดยการเพิ่มเหตุการณ์ใหม่ที่มีสมบัติต่างๆ กันที่เปลี่ยนแปลงอยู่เสมอ ระเบียบวาระ (Agenda) คือกลไกไดนามิกที่เป็นเครื่องของ Smart Element ซึ่งจัดเตียมตัวกลางการแปลงระหว่างการสังเกตการณ์เหตุการณ์และการปฏิบัติการที่ระบบปฏิบัติ

การประเมินค่ากฎ (Rule Evaluation) ยุทธวิธีที่กำหนดไว้ในการประเมินค่ากฎนั้นเริ่มประเมินเงื่อนไขจากบนไปล่าง อย่างไรก็ตามยุทวิธีที่กำหนดไว้นี้สามารถแก้ไขโดยใช้การกำหนดลำดับความสำคัญการอนุมานของสล็อท เงื่อนไขที่มีลำดับความสำคัญในการอนุมานของสล็อทสูงสุดจะถูกกระบวนการก่อน แล้วเงื่อนไขที่มีลำดับความสำคัญลำดับต่อมาก็จะถูกกระบวนการต่อไป เงื่อนไขต่างๆ ภายในกฎหนึ่งกฎเชื่อมเข้าด้วยกันโดยใช้ตัวการกระทำ AND (ดังนั้น เงื่อนไขทุกเงื่อนไขในหนึ่งกฎต้องเป็นจริงทั้งหมดจึงจะทำให้กฎนั้นเป็นจริง) ส่วนกฎหลายๆ กฎที่มีสมมติฐาน (Hypothesis) เดียวกันเชื่อมกันด้วยตัวการกระทำ OR ดังนั้นถ้ามีกฎบางกฎที่ประเมินค่าแล้วเป็นเท็จแต่มีกฎหนึ่งกฎที่ประเมินค่าเป็นไม่ทราบค่าสมมติฐานนั้นก็จะถูกประเมินว่าไม่ทราบค่าด้วย และถ้ามีกฎเพียงหนึ่งกฎที่อยู่ในสมมติฐานเดียวกันประเมินค่าได้เป็นจริงแล้วสมมติฐานนั้นก็จะถูกประเมินค่าว่าเป็นจริงเช่นกัน

กลไกการอนุมาน (Inferencing Mechanisms) กลไกการค้นหาเพื่ออนุมานใน Smart Element มี 7 ชนิดดังนี้ แบบลูกโซ่ย้อนกลับ (Backward Chaining), แบบลูกโซ่เดินหน้า (Forward Chaining), การแนะ (Suggesting), เกต (Gate), การเดินหน้าของสมมติฐาน (Hypothesis Forward), การโวลันเทีย (Volunteering), การเชื่อมบริบท (Context Links) แต่ละกลไกการค้นหาเหล่านี้ช่วยในการขยายการค้นหา เพื่อให้ได้ข้อสรุปที่ตรงกันโดยไม่ต้องทำการประเมินค่าทุกกฎในฐานความรู้ ลำดับความสำคัญของกลไกการค้นหาของระเบียบวาระใน Smart Element แสดงดังรูปที่ 2.11 ซึ่งจะอธิบายแต่ละกลไกการค้นหาเพื่อการอนุมานเหล่านี้ตามลำดับ ดังนี้



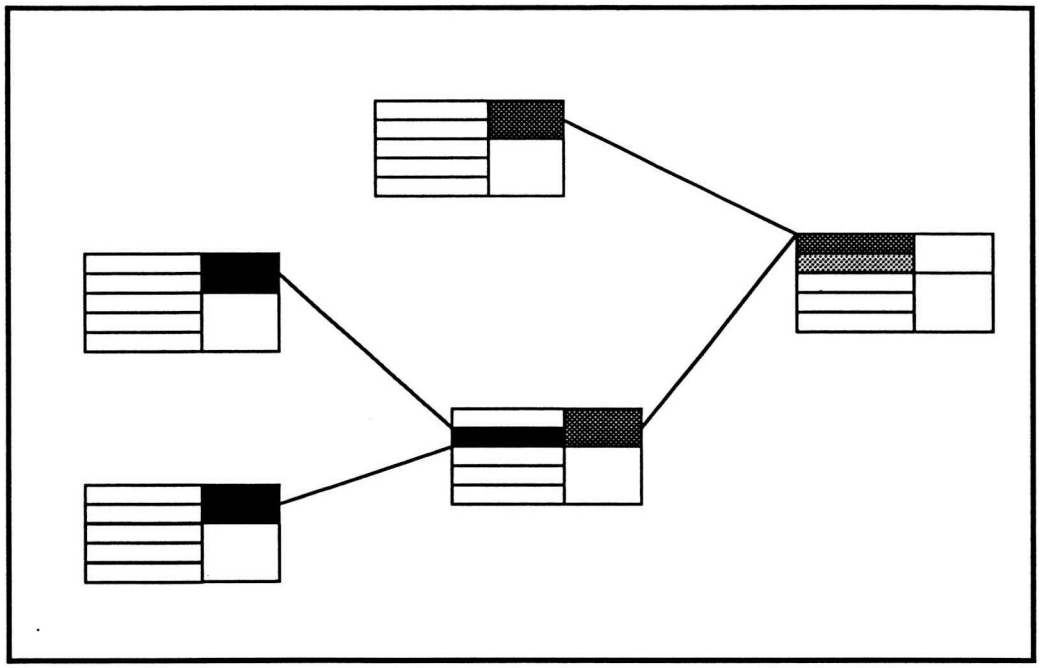
รูปที่ 2.11 ลำดับความสำคัญของกลไกการอนุมาน

ก. **แบบลูกโซ่ย้อนกลับ (Backward Chaining)** เป็นเหตุการณ์ที่มีลำดับความสำคัญสูงสุดซึ่งขึ้นอยู่กับการประเมินค่าของสมมติฐานเท่านั้น รูปที่ 2.12 แสดงกลไกการอนุมานแบบลูกโซ่ย้อนกลับหลายระดับ

ข. **การแนะ (Suggesting)** การแนะสมมติฐานจากระหว่างหน้าการพัฒนาเป็นการวางสมมติฐานนั้นบนระเบียบวาระ (Agenda) เพื่อทำการหาค่าทันทีทันใด การแนะ (Suggest) สมมติฐานเป็นการบอกระบบว่าสมมติฐานนั้นเป็นเป้าหมายที่สำคัญ และเมื่อเป็นเช่นนี้ก็ควรจะทำการตรวจสอบเท่าที่เป็นไปได้ การแนะสมมติฐานมีลำดับความสำคัญเหนือสมมติฐานที่สร้างโดยกลไกการค้นหาเพื่ออนุมานแบบอื่นๆ ยกเว้นกลไกการอนุมานแบบลูกโซ่ย้อนหลัง

ค. **การเดินหน้าของสมมติฐาน (Hypothesis Forward)** เป็นผลจากการตรวจสอบของเป้าหมายย่อยซึ่งมีทิศทางการค้นหา เพื่อการอนุมานตรงกันข้ามกับสมมติฐานสิ้นสุด แสดงดังรูปที่ 2.13

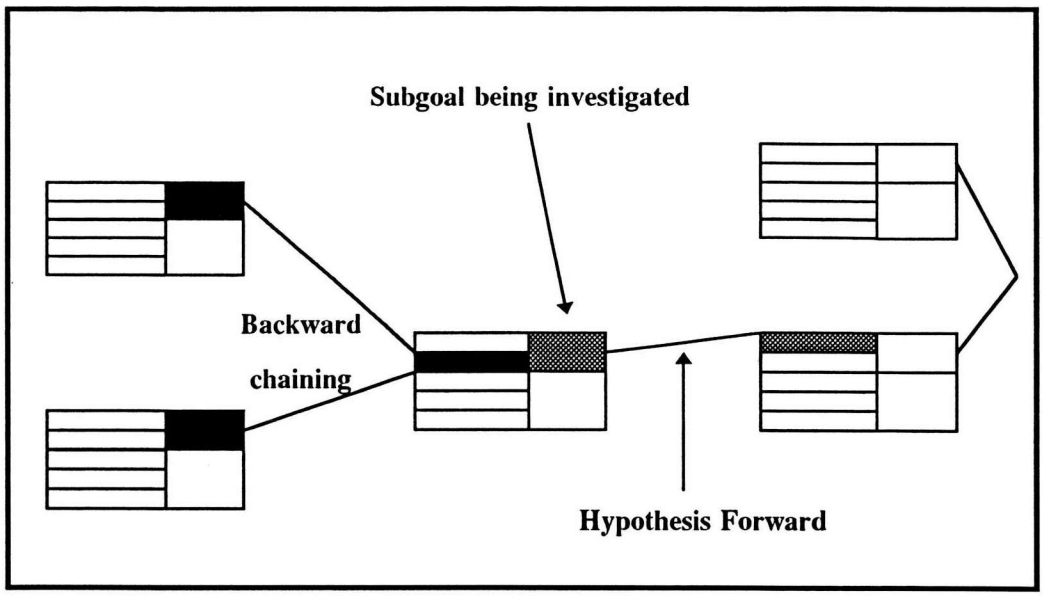
ง. **เกต (Gate)** เกตจะถูกสร้างขึ้นระหว่างการประเมินค่าเงื่อนไขของกฎซึ่งขึ้นอยู่กับการวิเคราะห์โครงสร้างของกฎ มีผลให้เกิดการแทรกสมมติฐานใหม่บนระเบียบวาระ (Agenda) เช่นในกรณีที่กฎสองกฎมีการใช้สล็อทในเงื่อนไขร่วมกัน ดังรูปที่ 2.14



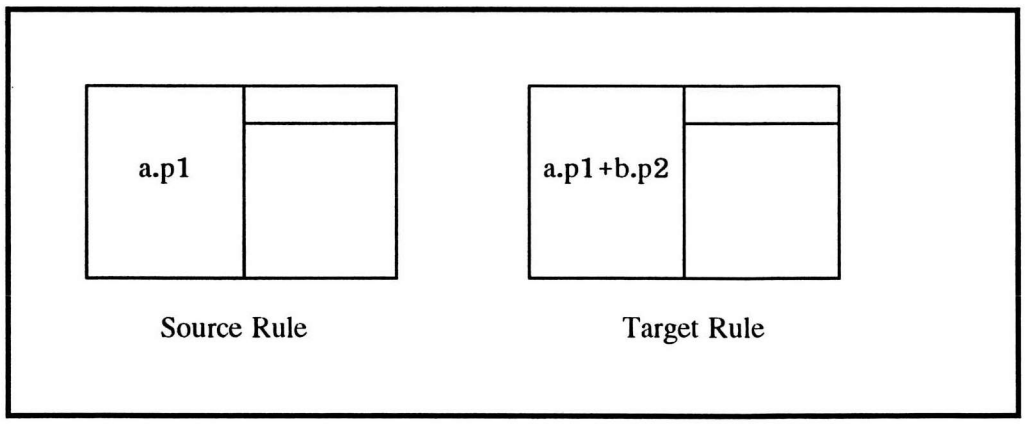
รูปที่ 2.12 ลูกโซ่ย้อนกลับหลายระดับ (Multiple Level Backward Chaining)

จ. แบบลูกโซ่เดินหน้า (Forward Chaining) ซึ่งเกิดขึ้นในกฎหรือวิธีการ (Method) ได้สองกรณีคือในการประเมินค่าเงื่อนไขว่าเป็นจริงแล้วจึงกระทำ (True Actions) และอีกกรณีหนึ่งคือเมื่อประเมินค่าเงื่อนไขว่าเป็นเท็จแล้วจึงกระทำ (False Actions) นั่นคือเริ่มต้นจากค่าของสล็อท และทำงานต่อไปข้างหน้ายังกฎทั้งหมดที่มีเงื่อนไขอ้างถึงสล็อทนั้น ดังรูปที่ 2.15

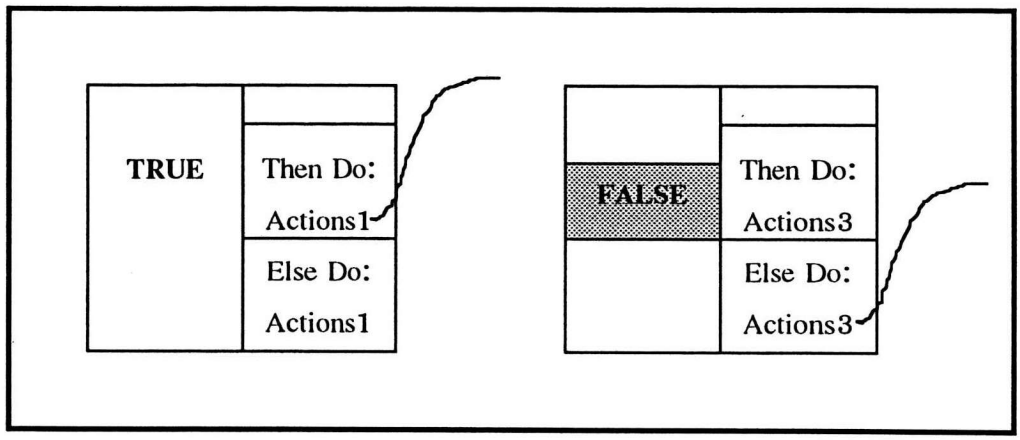
ฉ. แบบโวลันเทีย (Volunteer) เป็นการตั้งค่าของสล็อทที่แน่นอนในระหว่างหน้าการพัฒนาให้เป็นค่าเฉพาะ นั่นคือเริ่มจากค่าของข้อมูลที่มีอยู่หนึ่งหรือมากกว่าแล้วขยายต่อไป ดังรูปที่ 2.16



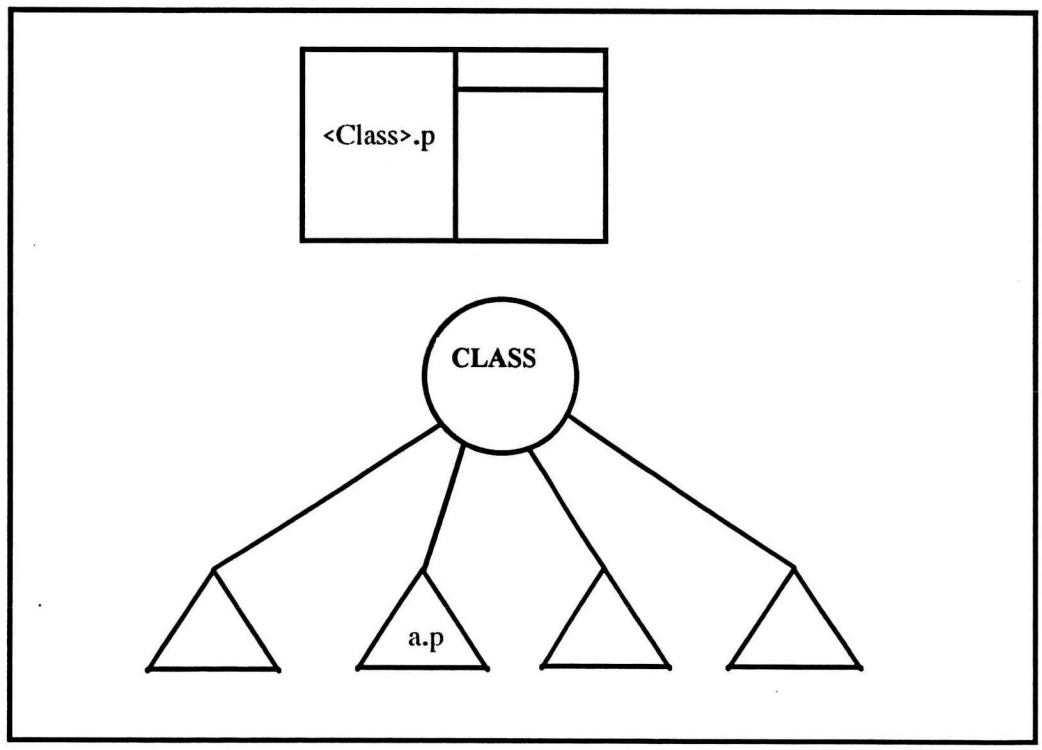
รูปที่ 2.13 การเดินหน้าของสมมติฐาน (Hypothesis Forward)



รูปที่ 2.14 กลไกการอนุมานแบบเกต



รูปที่ 2.15 ผลของการกระทำแบบเดินหน้า



รูปที่ 2.16 การโวลันเทียไปยังการเปรียบเทียบแบบอย่าง

ช. บริบท (Context) ในการควบคุมการขยายจากเกาะความรู้หนึ่งซึ่งเป็นกลุ่มของกฎและออปเจกต์ที่มีเงื่อนไข และสมมติฐานตัดระหว่างกันกับกลุ่มของเงื่อนไข, สมมติฐาน และการกระทำอื่นๆ ไปยังเกาะความรู้หนึ่งๆ นั้นไม่สามารถใช้กลไกการควบคุมทั้งหมดที่กล่าวมาข้างต้นได้ ดังนั้น จึงต้องใช้เส้นโยงบริบท (Context Links) หรือที่เรียกกันทั่วไปว่า เส้นโยงชั่วคราว (Weak Links) เพื่อขยายการควบคุมระหว่างเกาะความรู้ ซึ่งจะทำการเชื่อมสมมติฐานสองสมมติฐานหรือมากกว่าเข้าด้วยกัน แสดงดังรูปที่ 2.17

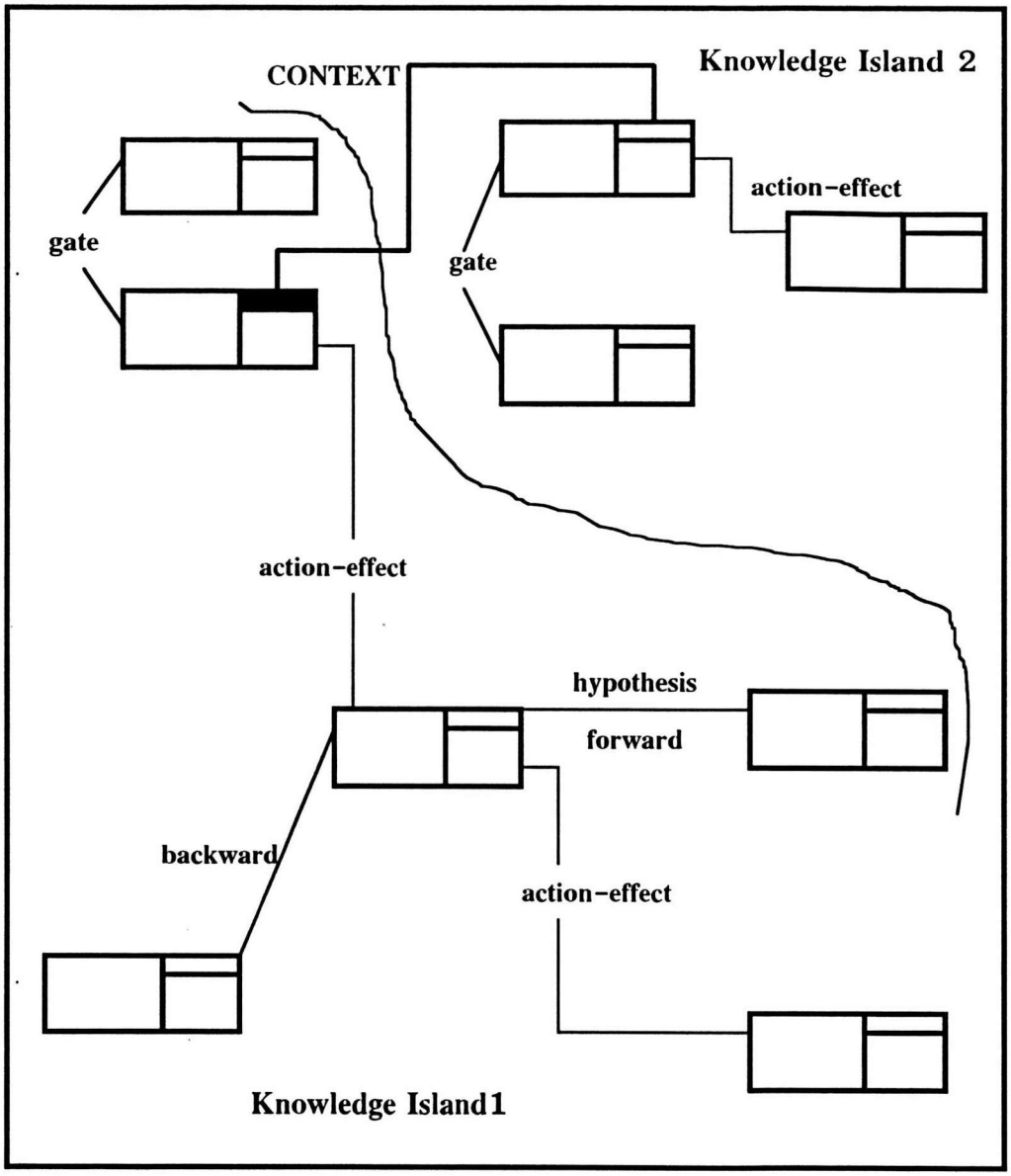
นอกจากนี้ยังมีกระบวนการปรับปรุงสถานะใหม่ (Revision) ซึ่งสามารถเปลี่ยนแปลงแก้ไขค่าของสถานะนั้นๆ ได้ ซึ่งพฤติกรรมแบบนี้เรียกว่า นอนโมโนโทนิก (Nonmonotonic) นั่นคือ เมื่อกระแส (Fired) ในการกระทำ (Actions) ด้านขวา (RHS) ของกฎแล้วจะมีทั้งการทริกเกอร์ เพื่อประเมินค่ากฎที่2 และเป็นผลให้กฎที่1 เปลี่ยนค่าตัวเองด้วย ในกระบวนการความรู้ของ Smart Element นั้น กฎใน Smart ELement เป็นแบบสมมาตร นั่นคือ ในกฎหนึ่งกฎอาจใช้พฤติกรรมในการหาเหตุผลได้ทั้ง 2 แบบรวมกัน คือ แบบนิรนัย (Deductive) หรือแบบลูกโซ่ย้อนกลับ (Backward) รวมกับแบบอุปนัย (Inductive) หรือแบบลูกโซ่เดินหน้า (Forward)

2.8.3 การติดต่อกับผู้ใช้ (Graphic User Interface; GUI)

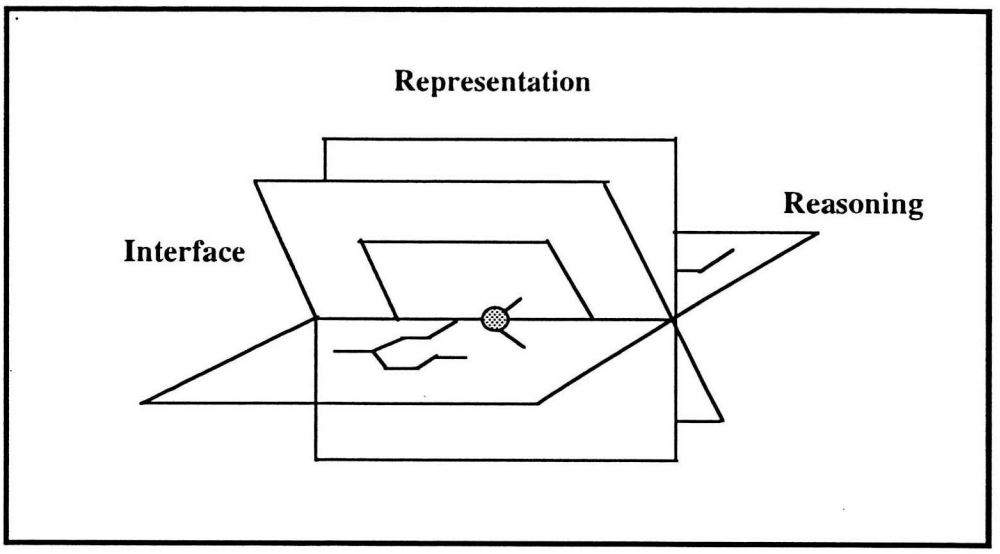
จากรูปที่ 2.7 ได้แสดงถึงระนาบเหตุผลและระนาบการแสดงความรู้ที่ตัดกัน ซึ่งระบบจะมุ่งสนใจที่จุดตัดกันนี้เพื่อตัดสินค่าข้อมูลที่ได้จากผู้ใช้ที่สัมพันธ์กับการหาเหตุผล ระบบอาจต้องสร้างเหตุการณ์ของคำถาม เพื่อค้นหาข้อมูลจากผู้ใช้ในกรณีที่ข้อมูลในการหาเหตุผลนั้นไม่เพียงพอ ดังนั้น จึงมีความจำเป็นต้องเพิ่มระนาบของส่วนการติดต่อกับผู้ใช้ ดังรูปที่ 2.18

Smart Element สามารถสร้างแอพพลิเคชันของการติดต่อกับผู้ใช้เชิงภาพ เพื่อให้เป็นวงจรส่วนหน้า (Front end) ของฐานความรู้ได้โดยใช้ Open Editor ซึ่งมีกลุ่มของตัวแก้ไข (Editors) ที่มีเครื่องมือมากมายในการพัฒนาการติดต่อกับผู้ใช้ของแอพพลิเคชัน องค์ประกอบของหน้าต่างที่ผู้ใช้ใช้ในการติดต่อกับแอพพลิเคชัน เพื่อสร้างเป็นหน้าต่างที่ต้องการก็สามารถสร้างได้ง่ายโดยการเลือกองค์ประกอบของหน้าต่างจากกรอบของวิดเจทใน Open Editor ที่เรียกว่า กล่องเครื่องมือ (Toolkit) ตัวแก้ไข (Editors) ที่กล่าวถึงนี้นอกจากใช้ในการพัฒนาฐานความรู้แล้วยังสามารถใช้แสดงภาพโดยรวมของตัวแก้ไขการติดต่อกับผู้ใช้

I20503738



รูปที่ 2.17 แสดงเส้นโยงบริบท (Context Link) ระหว่างเกาะความรู้



รูปที่ 2.18 การตัดกันของระนาบการแสดงความรู้, ระนาบการหาเหตุผล และระนาบการติดต่อกับผู้ใช้

ทางภาพด้วยที่เรียกรวมกันว่า แหล่งตัวแก้ไข (Resource Editors) ซึ่งแหล่งตัวแก้ไขนี้มีประโยชน์ในการจัดแผนงานหน้าต่าง และกำหนดส่วนประกอบของตัวเชื่อมประสาน แหล่งตัวแก้ไขที่ใช้กันอยู่แบ่งเป็นกลุ่มๆ ดังนี้

ก. **หน้าต่าง (Windows)** หมายถึงการติดต่อสื่อสารระหว่างแอพพลิเคชันกับผู้ใช้บั้นปลายโดยผ่านหน้าต่าง หน้าต่างจะรวมกลุ่มฟังก์ชันของแอพพลิเคชันให้อยู่ภายในส่วนที่กั้นไว้เป็นห้องๆ ที่สามารถแก้ไขได้

ข. **วิดเจท (Widgets)** องค์ประกอบหน้าต่างชั้นสูงนี้ทำให้ผู้ใช้บั้นปลายกระทำต่อกันกับหน้าต่าง องค์ประกอบหน้าต่างเหล่านี้ ได้แก่ กล่องเลือก (Choice Boxes), กล่องรายการ (List Boxes), ตัวแก้ไขตัวหนังสือ (Text Edits), ตัวเลือก (Browser), ปุ่ม (Bottons), เมนู (Menus) และฉลาก (Labels)

ค. **แหล่งสคริพท์ (Script Resources)** เป็นองค์ประกอบพิเศษที่ทำให้สามารถสร้างสคริพท์ในการควบคุมแนวทางของวิดเจท, หน้าต่าง และอะตอมที่กระทำระหว่างกันระหว่างการกระบวนการความรู้

ง. แหล่งภาพ (Graphic Resources) องค์ประกอบชั้นต่ำเหล่านี้ทำให้สามารถเพิ่มรูปภาพต่างๆ ภายในหน้าต่าง และวิดเจท แหล่งภาพรวมถึงกรอบ (Borders), สี และรูปแบบพื้นหลัง (Background Patterns)

จ. แหล่งเชิงตรรกศาสตร์ (Logical Resources) องค์ประกอบนี้ทำให้สามารถกำหนดตัวเชื่อมประสานกับผู้ใช้ที่พิเศษต่อวิดเจท รวมทั้งเคอร์เซอร์ (Cursors), แบบอักษร (Fonts), ไอคอน (Icons), ตัวเชื่อมคีย์ (Key Binding), รายการคีย์ (Key Lists) และสตริง

2.8.4 องค์ประกอบของแอพพลิเคชันใน Smart Element

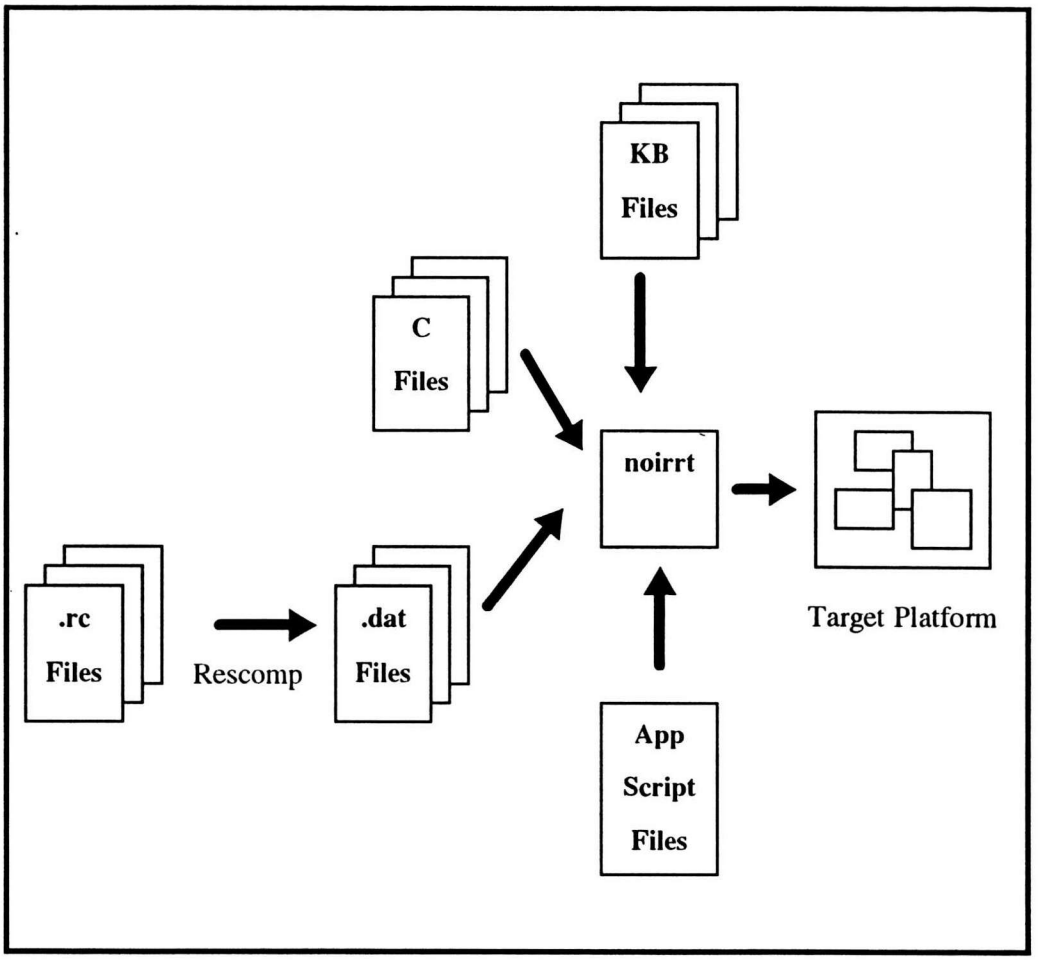
นอกจากจะมีไฟล์ฐานความรู้แล้วนั้นแอพพลิเคชันสุดท้ายจะประกอบด้วยหลายไฟล์ที่มีวากยสัมพันธ์ (Syntax) ต่างๆ กันไปขึ้นอยู่กับชนิดของไฟล์ ซึ่งไฟล์ของแอพพลิเคชันมีดังนี้

ก. ไฟล์เริ่มต้นให้ปฏิบัติการ ซึ่งมีเพียงหนึ่งไฟล์เป็นไฟล์ที่จัดการเกี่ยวกับการควบคุมการเริ่มต้น เรียกว่า แอพพลิเคชันสคริพท์ (Application Script)

ข. กลุ่มไฟล์ของทรัพยากรวงจรส่วนหน้าเชิงภาพ (Resource File ; .RC)

ค. ไฟล์รหัส (Code File) ซึ่งประกอบด้วยภารกิจภายนอก (ตัวเลือก)

ซึ่งไฟล์ดังกล่าวข้างต้นจะใช้ร่วมกันกับไฟล์ฐานความรู้ (Knowledge-Based File ; .KB) เพื่อแสดงพฤติกรรมของวงจรส่วนหน้าของแอพพลิเคชันฐานความรู้ในการส่งไปยังผู้ใช้ (ดูรูปที่ 2.19) ในที่สุดการปฏิบัติการแอพพลิเคชันให้สำเร็จนั้นเพียงแค่ใช้แอพพลิเคชันของ **NOIR Runtime** (NEXPERT Open Interface Runtime)



รูปที่ 2.19 แอพพลิเคชันของฐานความรู้ใน Smart Element