บทที่ 2

ทฤษฎี แนวคิด ของระบบผู้เชี่ยวชาญ

ความหมายของระบบผู้เชี่ยวชาญ

ศาสตราจารย์ Edward Feigenbaum แห่งมหาวิทยาลัยสแตนฟอร์ด ได้ให้คำจำกัดความของระบบ ผู้เชี่ยวชาญ (Expert System) ว่า **ระบบผู้เชี่ยวชาญ** คือ โปรแกรมคอมพิวเตอร์ที่มีความฉลาดด้วยการใช้ ความรู้และขบวนการอนุมาน (inference procedure) ในการแก้ปัญหาที่ยุ่งยากขนาดที่ต้องใช้ประสบการณ์ ความชำนาญการของมนุษย์จึงจะแก้ได้ ระบบผู้เชี่ยวชาญก็คือระบบคอมพิวเตอร์ที่เลียนแบบ (emulate) ความสามารถในการตัดสินใจของมนุษย์ผู้มีความเชี่ยวชาญ การเลียนแบบจะกระทำการตัดสินใจเหมือนคนจริง มากกว่าวิธีการจำลอง (simulation) จะเป็นโปรแกรมคอมพิวเตอร์ที่เก็บทั้งความรู้ที่เกี่ยวกับปัญหาที่จะแก้ไข และกระบวนการอนุมานเพื่อนำไปสู่ผลสรุปหรือคำตอบของปัญหานั้น ความรู้ที่เก็บมามีทั้งความรู้ที่เป็นจริงที่อาจถูกบันทึกไว้ในรูปของตำราหรือเอกสารทางวิชาการ และ ความรู้ที่ได้จากประสบการณ์ ที่อาจไม่อยู่ในรูปเอกสาร ตำรา แต่ต้องดึงมาจากผู้เชี่ยวชาญที่มีประสบการณ์ ซึ่งหมายถึงผู้ที่มีความรู้ความชำนาญในสาขาใดสาขาหนึ่งโดยเฉพาะ เป็นพิเศษกว่าคนอื่น

ระบบผู้เชี่ยวชาญจะเป็นระบบการแก้ไขปัญหาที่ยุ่งยาก และไม่มีโครงสร้างที่แน่นอน คำตอบจะมีโอกาสเป็นได้หลายแบบ ตามลักษณะของปัญหาและข้อมูลที่มีอยู่ ซึ่งไม่สามารถจะกำหนดขั้นตอนในการแก้ไขไว้ล่วงหน้าได้ ต้องอาศัยความรู้ประสบการณ์และสภาพของปัญหาในขณะนั้นรวมกัน จึงจะแก้ไขได้ การสร้างโปรแกรมระบบผู้เชี่ยวชาญจะมีเทคนิคแตกต่างไปจากการสร้างโปรแกรมทั่วไป และการประยุกต์ใช้ก็แตกต่างกัน

Wolfgram, Dear และ Galbraith (1987) ได้สรุปลักษณะเฉพาะ ของระบบผู้เชี่ยวชาญไว้ดังนี้

1. นำมาใช้แก้ไขปัญหาที่มีความยาก ระดับที่ต้องการผู้เชี่ยวชาญมาแก้ปัญหาให้

2. สามารถแก้ไขปัญหาโดยใช้สัญญาลักษณ์

3. สามารถนำเอาปัจจัยที่มีความคงที่ และความน่าจะเป็นไปใช้ในการแก้ไขปัญหาที่ไม่มีความแน่นอน (uncertainty) ได้

4. ผลลัพธ์ของระบบผู้เชี่ยวชาญจะเป็นคำตอบดีที่สุดที่เป็นไปได้ พร้อมคำอธิบายและข้อเสนอแนะ มากกว่าที่เป็นคำตอบที่เป็นปริมาณหรือเป็นตัวเลขที่แน่นอน

5. จะนำไปแก้ปัญหาที่ต่อเนื่องกัน ซึ่งอาจจะไม่ทราบข้อมูล และไม่ทราบส่วนประกอบอื่นๆ ของปัญหาเลย

6. ระบบผู้เชี่ยวชาญจะหาคำตอบโดยใช้สัญญาลักษณ์ และการแก้ไขปัญหาแบบศึกษาสำนึก (heuristic approach)

7. ผลลัพธ์ของระบบผู้เชี่ยวชาญอาจมีหลายคำตอบ หรือคำตอบเดียว หรือ ไม่มีคำตอบเลยก็ได้ ระบบจะใช้ความรู้จากฐานความรู้ไปดำเนินการค้นหาคำตอบเอง

8. มีความสามารถสร้างเส้นทางการอนุมานเพื่อการอธิบายและการพินิจพิจารณาหลักฐานการตรวจสอบ (audit trail) จากผู้ใช้ระบบ และการติดต่อกับผู้ใช้จะเป็นภาษาธรรมชาติ (natural language) มีลักษณะเป็นคำโต้ตอบ (dialogue)

9. มีความยืดหยุ่นในการสร้าง สามารถรับตัวแปรที่เป็นแบบพลวัต (dynamic variable) ได้ ซึ่งเป็นแปรที่มีการเปลี่ยนแปลงค่าตลอดเวลา

10. ระบบผู้เชี่ยวชาญทำงานแบบเรียกซ้ำ (recursive) เมื่อมีการอ่านข้อมูลสารสนเทศ (information) ไปครั้งหนึ่งแล้ว ก็สามารถนำกลับมาใช้ได้อีกโดยไม่จำเป็นต้องอ่านซ้ำ

11. การทำงานของระบบผู้เชี่ยวชาญจะไปตามสถานการณ์ความสัมพันธ์ระหว่างข้อมูลที่ใส่เข้าไป กับ สภาพแวดล้อมการทำงานในขณะนั้น

12. ระบบผู้เชี่ยวชาญต้องมีการปรับปรุงสภาพการทำงานของระบบให้เป็นมาตรฐานอยู่ตลอดเวลาโดยผู้เชี่ยวชาญ หรือวิศวกรรมความรู้ (knowledge engineer)

ความเป็นมาของระบบผู้เชี่ยวชาญ

Wolfgram, Dear และ Galbraith (1987) ได้กล่าวว่าระบบผู้เชี่ยวชาญเริ่มต้นมีขึ้นเมื่อ ค.ศ.1965 โดย Joshua Lederberg นักเคมีรางวัลโนเบิลแห่งมหาวิทยาลัยสแตนฟอร์ดได้สร้างระบบ DENDRAL ที่ใช้ขั้นตอนวิธี (algorithm) ในการวิเคราะห์ สารสนเทศจาก spectoscopic analysis เพื่อหาโครงสร้างโมเลกุลขององค์ประกอบทางเคมี โดยใช้การค้นหาแบบก่อนกำเนิดและทดสอบ (generate and test) ใช้เวลาการพัฒนาไม่ต่ำกว่า 15 ปี กลายมาเป็นระบบ META-DENDRAL เพิ่มการรวบรวมข้อมูลความรู้จากนักเคมีที่เป็นผู้เชี่ยวชาญ เขียนด้วยโปรแกรมภาษา LISP(LISt Programming language) และถือว่าเป็นตัวอย่างที่ดีของระบบฐานแห่งกฎ (rule-based system) แทนความรู้แบบประโยคของกฎการผลิต (production rule) คือ IF-THEN

ในปี ค.ศ. 1970 ระบบ CADAUCEUS ได้พัฒนาขึ้นที่มหาวิทยาลัยพิทส์เบิอร์ก เป็นระบบที่ช่วยแพทย์ในการวินิจฉัยโรคภายใน (internal diseases) 17 ปีต่อมาระบบนี้ได้รวบรวมความรู้เฉพาะในเรื่องใดเรื่องหนึ่งที่เกี่ยวข้องกันได้ถึง 85%ของทั้งหมด ใช้กลยุทธ์ในการแก้ไขปัญหาแบบ ล่างขึ้นบน (bottom-up) แล้วกลับเป็นแบบ บนลงล่าง (top-down)

ระบบ MACSYMA เป็นระบบผู้เชี่ยวชาญทางคณิตศาสตร์ พัฒนาขึ้นที่ MIT ในโตรงการชื่อMAX จนถึงปี ค.ศ. 1971มีความสามารถในการทำหาอนุพันธ์ (differentiate) และหาปริพันธ์ (integrate) โดยใช้สัญญาลักษณ์ และสามารถลดรูปนิพจน์ทางคณิตศาสตร์ได้อย่างดี MACSYMA ประกอบด้วยกฎที่ได้รับจากผู้เชี่ยวชาญทางคณิตศาสตร์หลายร้อยกฎ แต่ละกฎจะแสดงวิธีการเปลี่ยนรูปนิพจน์ไปยังอีกนิพจน์หนึ่งที่สมมูลกัน สำหรับการแก้ปัญหา และเป็นการหาวิธีการเชื่อมโยงของกฎซึ่งทำการเปลี่ยนนิพจน์เริ่มต้นไปเป็นอีกนิพจน์หนึ่งที่มีความกระทัดรัดมากกว่า ระบบนี้เขียนด้วยภาษา LISP

ระบบ MYCIN พัฒนาที่มหาวิทยาลัยสแตนฟอร์ดในปี ค.ศ.1972 เป็นระบบผู้เชี่ยวชาญที่มีชื่อเสียงและเป็นที่รู้จักมากที่สุดระบบหนึ่ง ระบบ MYCIN จะให้คำปรึกษาเกี่ยวกับการวินิจฉัยโรค และการรักษาเฉพาะโรคติดเชื้อทางเลือด ประกอบด้วยฐานความรู้ที่เป็นกฎไม่ต่ำกว่า 400 กฎ ในปี ค.ศ. 1976 มหาวิทยาลัยสแตนฟอร์ดได้พัฒนาระบบ TIERESIAS ซึ่งเป็นเครื่องมือการแสวงหาความรู้ (knowledge acquisition) ที่ช่วยในการนำข้อมูลเข้าและการปรับปรุงฐานความรู้ของ

MYCIN โดยใช้ประโยชน์จากอภิความรู้ (mata knowledge) ประมาณปี ค.ศ. 1978 ได้มีวิวัฒนาการจาก MYCIN เป็น EMYCIN ระบบ EMYCIN จะประกอบด้วยโครงสร้างเชิงตรรกะทั้งหมดของ MYCIN ถือได้ว่าเป็นจุดเริ่มต้นของ เปลือกระบบผู้เชี่ยวชาญ (expert system shell) หรือเป็นโปรแกรมที่ประกอบด้วยโครงสร้างเชิงตรรกะ และ กลยุทธ์ในการคิด แต่ไม่มีฐานความรู้เรื่องใดอยู่เลย

ระบบ PUFF เป็นระบบผู้เชี่ยวชาญให้คำปรึกษาในการวินิจฉัยโรคการทำงานของปอด (pulmonary function diseases) ระบบ PUFF ได้นำระบบ EMYCIN มาใส่ฐานความรู้เรื่องการวินิจฉัยโรคการทำงานของปอด ระบบ PUFF จะบันทึกการทำงานของปอดโดยตรงจากเครื่องมือวัดการทดสอบการทำงานของปอด (pulmonary function test) ใช้การแทนความรู้ (knowledge representation) แบบระบบฐานแห่งกฎ (rules-based system) และกลยุทธ์แบบลูกโซ่ไปข้างหน้า (forward chaining)

ระบบ PROSPECTOR พัฒนาจากสถาบันการวิจัยแห่งสแตนฟอร์ด โดยใช้ระบบฐานแห่งกฎให้คำปรึกษาแก่นักธรณีวิทยาในการหาแหล่งแร่ มีโครงสร้างข้อมูลแบบข่ายงานความหมาย (semantic networks) เขียนด้วยภาษา LISP

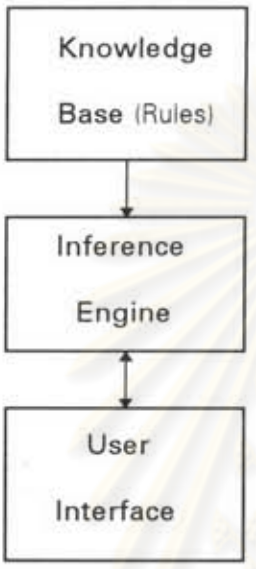
ใน ค.ศ. 1980 บริษัทดิจิทอล อีคิบเมนท์ (Digital equipment) ได้พัฒนาระบบ XCON ที่กลายเป็นระบบผู้เชี่ยวชาญระบบแรกที่ประสพความสำเร็จในภาคธุรกิจ ระบบ XCON จะให้คำตอบแก่ลูกค้าในเรื่องโครงแบบ (configuration) ที่ยากๆของระบบคอมพิวเตอร์ของ VAX

ระบบ HEARSAY-II คือระบบเข้าใจคำพูด (speech understanding system) ที่พัฒนาโดยมหาวิทยาลัยคาร์เนกีเมลลอน เป็นระบบแรกที่สามารถเข้าใจการสนทนาติดต่อกันของมนุษย์ รวมเป็นคำศัพท์ทั้งหมดประมาณ 1,000 คำ แม้ว่าจะเทียบได้เท่ากับเด็กอายุ 10 ขวบ แต่ก็ยังมีการพัฒนาต่อไปให้ความเข้าการสนทนามากขึ้น

ปัจจุบันได้มีการสร้างเปลือกระบบผู้เชี่ยวชาญ ซึ่งเครื่องมือที่ช่วยให้การพัฒนาระบบผู้เชี่ยวชาญทำได้ง่ายขึ้น เปลือกระบบผู้เชี่ยวชาญที่เป็นที่รู้จักได้แก่ OPS จากมหาวิทยาลัยคาร์เนกีเมลลอน AGE จากมหาวิทยาลัยสแตนฟอร์ด และที่พัฒนาอยู่บนเครื่องคอมพิวเตอร์ส่วนบุคคลได้แก่ M.1 , EXSYS , GURU, PC plus.

โครงสร้างพื้นฐานของระบบผู้เชี่ยวชาญ

ระบบผู้เชี่ยวชาญประกอบด้วยโครงสร้างพื้นฐาน 3 อันตามรูปที่ 2.1 คือ



รูปที่ 2.1 แสดงโครงสร้างพื้นฐานของระบบผู้เชี่ยวชาญ

1. **ฐานความรู้** (Knowledge Base)

ฐานความรู้ เป็นที่เก็บความรู้สารสนเทศ (Information) ทั้งหมด ที่จะใช้ในระบบผู้เชี่ยวชาญ เพื่อใช้หาคำตอบของปัญหา ฐานความรู้ ไม่เพียงเป็นที่เก็บความรู้แบบฐานข้อมูลเท่านั้น แต่ยังเก็บความสัมพันธ์ ของสารสนเทศไว้ด้วย ประกอบด้วย 2 ส่วนที่เกี่ยวข้องกัน

1.1 **ข้อมูลความจริง** (Facts)

ข้อมูลความจริงจะเก็บความจริง(Facts)ของปัญหาทั้งหมด ทั้งที่มีความสัมพันธ์และไม่มีความสัมพันธ์ เป็นบริเวณที่เก็บของ ข้อมูลสารสนเทศที่จำเป็น และไม่มีการเปลี่ยนแปลงความรู้เกี่ยวปัญหา

1.2 **ข้อมูลความสัมพันธ์** (Relationships)

ข้อมูลความสัมพันธ์เป็นที่เก็บความสัมพันธ์ของปัญหาและคำตอบในรูปการแทนความรู้(Knowledge Representation) แบบต่างๆ ตามความเหมาะสมของเรื่องที่จะศึกษาเช่น ข่ายงานความหมาย (Semantic networks) กรอบ (Frames) ระบบฐานแห่งกฎ (Rules-based system) และแคลคูลัสภาคแสดง (Predicate calculus) ซึ่งเป็นข้อมูลที่เป็นหัวใจสำคัญ ของระบบผู้เชี่ยวชาญ

2. **เครื่องอนุมาน** (Inference Engine)

เครื่องอนุมานเป็นส่วนที่ควบคุมการใช้ความรู้ในฐานความรู้ เพื่อแก้ไขปัญหาอย่างมีประสิทธิภาพโดยการเลือกวิธีการค้นหา (search methods) ซึ่งมี 2 ประเภทคือการค้นหาแบบไม่มีสิ่งชี้นำ (Blind searches) กับ การค้นหาแบบศึกษาสำนึก (Heuristic searches) แล้วจัดกระทำกับฐานความรู้ เพื่อให้ได้คำตอบที่ต้องการ โดยมีการควบคุมทิศทางการค้นหาซึ่งมีหลายวิธีได้แก่ แบบลูกโซ่ไปข้างหน้า (Forward Chaining) แบบลูกโซ่ย้อนกลับ(Backward Chaining)

3. **ระบบการติดต่อกับผู้ใช้** (User Interface)

ระบบการติดต่อกับผู้ใช้ เป็นการติดต่อระหว่างผู้ใช้ระบบกับตัวโปรแกรมระบบผู้เชี่ยวชาญ โดยการถามตอบ หรือใส่ข้อความ หรือเลือกตอบจากตัวเลือกที่กำหนดให้ การติดต่อกับผู้ใช้มี 2 รูปลักษณะคือ

3.1 ใส่ข้อมูล (Data Entry)

ผู้ใช้ระบบจะให้ข้อมูลที่บอกสภาพของปัญหาที่เกิดขึ้น ด้วยการตอบคำถาม หรือใส่ข้อความ หรือเลือกคำตอบจากรายการที่กำหนดให้เลือก ทำไปจนกระทั่งได้คำตอบ

3.2 แสดงผลลัพธ์ (Output)

เมื่อโปรแกรมได้ทำการค้นหาคำตอบเสร็จ จะแสดงผลให้ทราบว่า ได้คำตอบหรือไม่ ถ้าได้คำตอบ จะมีคำอธิบาย เหตุผล(Why) ที่ใช้ในการสรุปคำตอบ คำอธิบายเพิ่มเติม ว่าจากคำตอบที่ได้ควรทำอย่างไร (How) ต่อไปอีก หรือคำอธิบายข้อสรุปที่มีคำตอบมากกว่า 1 คำตอบ

การประยุกต์ใช้ระบบผู้เชี่ยวชาญ

Hayes-Roth (1983) และ Waterman (1986) ได้จำแนกประเภทของงานที่มีการประยุกต์ใช้ระบบผู่เชี่ยวชาญดังนี้

1. **การแปลความหมาย** (Interpretation)

ระบบผู้เชี่ยวชาญจะวิเคราะห์ข้อมูลที่ได้รับแล้วทำการแปลความหมายของสภาพการณ์ เช่นการแปลความหมายของข้อมูลที่วัดได้ในโรงงานเคมีเพื่อแสดงสถานะของการดำเนินการ ระบบ CRYSALIS เป็นการแสดงความหมายเป็นโครงสร้างสามมิติของโปรตีน

2. **การวินิจฉัย** (Diagnosis)

เป็นการเฝ้าสังเกตระบบการทำงานที่ผิดปรกติแล้วทำการหาสาเหตุของความผิดปรกตินั้น เช่น สังเกตอาการของผู้ป่วยที่เปลี่ยนแปลงไปแล้วทำการหาสาเหตุ ระบบ AI/COAG เป็นระบบวินิจฉัยเกี่ยวโรคเลือด ระบบ AI/RHEUM เป็นการวินิจฉัยเกี่ยวกับโรคไขข้ออักเสบ ระบบ ACE เป็นการวินิจฉัยข้อผิดพลาดในเครือข่ายโทรศัพท์

3. **การตรวจจับ** (Monitoring)

เป็นการสังเกตุพฤติกรรมของระบบที่คาดว่าจะแสดงออกมาให้เห็น เช่น ระบบ VM เป็นการตรวจจับอาการต่างๆของผู้ป่วยในไอซียู ระบบ ANNA เป็นการสังเกตุพฤติกรรมของผู้ป่วยโรคหัวใจที่รักษาด้วยยา ดิจิเทลลิส

4. **การแก้ไข** (Debugging)

เป็นการให้ข้อเสนอแนะการแก้ไขข้อผิดพลาดของระบบปรกติ เช่น แนะนำว่าจะปรับระบบคอมพิวเตอร์อย่างไร จึงจะลดความล่าช้าในการทำงานลงได้

5. **การซ่อมแซม** (Repair)

เป็นการหาสาเหตุและแก้ไขการทำงานของระบบตามระยะเวลาและแผน ที่ได้กำหนดไว้ เช่นการซ่อมบำรุงระบบคอมพิวเตอร์

6. **การคาดการณ์** (Prediction)

เป็นการวิเคราะห์ข้อมูลที่ผ่านมาเพื่อคาดการณ์เหตุการในอนาคตล่วงหน้า เช่นการทำนายสภาพอากาศอีก 1 สัปดาห์ข้างหน้า ระบบ PTRANS เป็นการพยากรณ์สำหรับการบริหารจัดการคอมพิวเตอร์ของDEC

7. **การออกแบบ** (Design)

เป็นการหาข้อกำหนดลักษณะเฉพาะตามวัตถุประสงค์ที่ต้องการ เช่น ระบบ EURISKO เป็นการออกแบบสามมิติของวงจรไฟฟ้าจุลภาค ระบบ PALLADIO เป็นการออกแบบและตรวจสอบวงจร VLSI

8. **การวางแผน**(Planning)

เป็นการเตรียมขั้นตอนการปฏิบัติให้บรรลุตามวัตถุประสงค์ที่กำหนดไว้ เช่น ระบบ SPEX เป็นการวางแผนการทดลองทางชีววิทยาเกี่ยวกับโมเลกุล

9. **การสอน** (Instruction)

เป็นการสอนเชิงปัญญา ที่ให้ผู้เรียนได้สอบถาม ทำไม อย่างไร และอะไรถ้า (what-if) เสมือนมีผู้เชี่ยวชาญมาสอนให้จริงๆ เช่น ระบบ CADHELP เป็นการสอนการใช้โปรแกรมออกแบบ ระบบ STEAMER เป็นการสอนการควบคุมโรงไฟฟ้าพลังน้ำ ระบบ GUIDON เป็นการสอนในเรื่องการติดเชื้อแบคทีเรีย ระบบ SOPHIE เป็นการสอนวิธีการวินิจฉัยข้อผิดพลาดของวงจรไฟฟ้า

10. **การควบคุม** (Control)

เป็นการบังคับ ควบคุมการทำงานของระบบทั้งหมดให้เป็นไปตามวัตถุประสงค์ เช่น การควบคุมจราจร ระบบ YES/VMS ใช้ควบคุมระบบปฏิบัติการ VMS ของบริษัทไอบีเอม

ขั้นตอนการพัฒนาระบบผู้เชี่ยวชาญ

การสร้างหรือพัฒนาระบบผู้เชี่ยวชาญ มีขั้นตอนหลักการอย่างเดียวกับการสร้างโปรแกรมปรกติ แตกต่างกันในรายละเอียดเฉพาะขั้นตอนการพัฒนาต้นแบบและการสร้าง ดังนี้

1. **กำหนดลักษณะของปัญหาและบทนิยาม** (Identification and definition)

เป็นการกำหนดลักษณะและขอบเขตของปัญหา ในเรื่องความต้องการของผู้เชี่ยวชาญ ทรัพยากรที่จำเป็นอันได้แก่ เงินทุน อุปกรณ์ เครื่องคอมพิวเตอร์ โปรแกรม บุคลากรที่มีเกี่ยวข้องกับระบบ และเวลาที่ใช้ในการพัฒนา ต้องกำหนดเป้าหมายของการพัฒนาระบบผู้เชี่ยวชาญขึ้นมาอย่างชัดเจน ต้องแน่ใจว่าปัญหาที่มีอยู่สามารถแก้ไขได้โดยใช้ระบบผู้เชี่ยวชาญ ปัญหาบางอย่างง่ายหรือยากเกินไปไม่เหมาะสมกับการใช้ระบบผู้เชี่ยวชาญ การให้บทนิยามของคำที่ใช้ต้องมีความชัดเจนเป็นสากลที่เข้าใจได้ตรงกัน ดังนั้นจึงควรคำนึงถึงสิ่งต่อไปนี้คือ

1. ทรัพยากรที่มีอยู่อันได้แก่ เครื่องคอมพิวเตอร์ อุปกรณ์ที่เกี่ยวข้อง เงินทุน บุคลากรผู้เชี่ยวชาญในเรื่องจะพัฒนา และระยะเวลาที่มีอยู่

2. เป้าหมายของการสร้างระบบผู้เชี่ยวชาญ ต้องทราบว่า จะนำไปใช้ในงานประเภทไหน มักต้องมีความชัดเจน และเจาะจงเรื่องใดเรื่องหนึ่งโดยเฉพาะ

3. มีเกณฑ์ในการเก็บข้อมูล การแทนความรู้ และการใช้คำต่างๆ ให้เข้าใจตรงกันในเรื่องเดียวกันได้

2. **การพัฒนาต้นแบบ** (Development of prototype)

เป็นขั้นตอนที่ต้องหาแนวคิดพื้นฐานที่จะแทนความรู้ (knowledge representation) กฎต่างๆ ความสัมพันธ์ของกฎ สามารถเข้าใจระบบสายงานของสารสนเทศที่จำเป็นต่อการอธิบายกรรมวิธีในการแก้ปัญหา ต้องเลือกตัวเครื่องและโปรแกรมให้เหมาะสม ต้องจัดรูปแบบการแทนความรู้ สร้างโครงสร้างของระบบที่เป็นต้นแบบขึ้นมา ให้มีขั้นตอนการทำงานครบ สร้างเป็นโมเดลที่สามารถทดสอบการทำงานได้ เพื่อจะได้ตัดสินว่าจะพัฒนาให้ต้นแบบไปเป็นของจริงหรือจะหยุดการพัฒนาต่อไปได้ จึงต้องมีการคำนึงถึงสิ่งต่อไปนี้คือ

1. ต้องค้นหาความรู้ ความเข้าใจ การไหลเวียนของสารสนเทศในระบบที่ทำการแก้ปัญหา

หอสมุดกลาง จุฬาลงกรณ์มหาวิทยาลัย

2. พัฒนาความรู้ใหม่ จากความรู้ที่มีอยู่เดิม ด้วยวิธีการของอภิความรู้ (meta knowledge)

3. สร้างวิธีการแก้ไขปัญหาโดยใช้ฐานความรู้ที่มีอยู่ และการอนุมาน

4. เลือกใช้เครื่องคอมพิวเตอร์ อุปกรณ์ และโปรแกรม ให้เหมาะสมกับปัญหาที่จะทำการแก้ไข

5. ระบุเงื่อนไขบังคับของระบบผู้เชี่ยวชาญ เช่นการแทนความรู้ ลักษณะของปัญหาที่จะนำไปใช้

3. **การสร้าง** (Construction)

นำกฎที่มีอยู่ในต้นแบบมาแจงรายละเอียดให้ครบ สร้างความสัมพันธ์ให้ครบทุกกรณี และจัดความรู้ใหม่ นำความรู้จากปฐมภูมิมาจัดให้เป็นฐานความรู้ (knowledge acquisition) มีการสร้างตัวประสานกับผู้ใช้ (user interface) ทั้งการนำข้อมูลเข้า การปรับปรุงและรักษาสภาพข้อมูลรวมถึงการให้คำปรึกษา และข้อเสนอแนะในการแก้ไขปัญหา

4. **การทดสอบและประเมินค่า** (Test and evaluation)

ระบบผู้เชี่ยวชาญต้องถูกตรวจสอบหลายครั้งโดยหลายวิธี ทุกอย่างจะถูกตรวจสอบตั้งแต่ส่วนที่ติดต่อกับผู้ใช้ระบบส่วนหน้า ต้องมีความชัดเจน ติดต่อสื่อสารกันได้ ง่ายต่อการใช้งาน จนถึงความถูกต้องของผลสรุปของปัญหา มีการตรวจสอบเทียบผลลัพธ์จากผู้เชี่ยวชาญกับระบบผู้เชี่ยวชาญควรได้ผลสรุปตรงกันอย่างน้อย 90 % และเมื่อมีการปรับปรุงแก้ไขระบบใหม่ก็ต้องมีการตรวจสอบอีกจนกว่าจะแน่ใจว่าสมบูรณ์จึงจะนำไปในงานได้

5. **การรวมเบ็ดเสร็จและทำให้เกิดผล** (Integration and Implementation)

เป็นการนำระบบผู้เชี่ยวชาญที่สร้างขึ้น ไปรวมเข้ากับการปฏิบัติงานประจำวัน ทำให้เกิดการเปลี่ยนแปลงวิธีการปฏิบัติงาน ต้องมีการศึกษา ฝึกอบรมใหม่ อาจทำให้เกิดการต่อต้านขึ้นได้ ถ้าผู้ใช้ไม่มีการยอมรับระบบผู้เชี่ยวชาญ ก็ทำให้การพัฒนาระบบทั้งหมดที่สร้างมาสูญเปล่าไป ดังนั้นการจะนำระบบผู้เชี่ยวชาญไปใช้ในชีวิตประจำวัน จำเป็นต้องทำด้วยความระมัดระวัง มีการวางแผนที่ดี มีการฝึกอบรม มีเอกสารให้ศึกษา และมีการสนับสนุนเป็นอย่างดี จึงจะประสบความสำเร็จ

6. **การรักษาสภาพ** (Maintenance)

ในรายละเอียดของระบบผู้เชี่ยวชาญ จะไม่มีความสมบูรณ์ เนื่องจากการเปลี่ยนแปลงสภาพแวดล้อมของสารสนเทศที่มีอยู่ตลอดเวลา และขึ้นอยู่กับความสามารถของระบบผู้เชี่ยวชาญในการปรับปรุงแก้ไขกฎใหม่และศึกษาสำนึกใหม่ การรักษาสภาพของระบบผู้เชี่ยวชาญคือการทบทวน การดัดแปร การขยายเพิ่มเติม และการยกระดับฐานความรู้ ส่วนเครื่อง ส่วนอุปกรณ์และโปรแกรม

การแทนความรู้ (Knowledge Representation)

1. **ข่ายงานความหมาย** (Semantic networks)

ข่ายงานความหมายเป็นโครงสร้างการแทนความรู้ที่ใช้เป็นพื้นฐานสำหรับการแทนความรู้แบบอื่น ๆ ข่ายงานความหมายเองไม่ได้ถูกนำมาใช้แทนความรู้โดยตรงเนื่องจากการขาดกฎโครงสร้างที่เป็นลักษณะเฉพาะ

ข่ายงานความหมายคือการรวบรวมวัตถุ (object) ที่เป็นบัพ (node) ที่มีความเกี่ยวพันธ์กันโดยความสัมพันธ์ บัพจะเป็นแนวคิด (concept) และสภาพการณ์ของวัตถุมักแทนด้วยวงกลม จุด หรือกล่อง ส่วนความสัมพันธ์ (relationship) ระหว่างบัพ จะแทนด้วยเส้นตรงหรือเส้นโค้งต่อเชื่อมระหว่างบัพ

ข้อดีของข่ายงานความหมายคือสามารถที่จะเพิ่ม หรือแก้ไข หรือลบ ทั้งบัพและเส้นเชื่อม ที่ตำแหน่งไหน ก็ทำได้ง่าย สามารถถ่ายทอดทายาทความสัมพันธ์ (inheritance) จากบัพหนึ่งไปยังบัพอื่น ได้ทั้งแบบลำดับชั้น (hierarchy) และแบบอธิบายความรู้เป็นคุณสมบัติ (property)

ข้อเสียของข่ายงานความหมายคือ ไม่มีโครงสร้างการแทนความรู้ ที่เป็นทางการ (formal) ไม่มีกฎมาตรฐานที่จะกำหนดลักษณะเฉพาะของบัพและความสัมพันธ์ ทำให้การแทนความรู้ของข่ายงานความหมายมีหลายรู้แบบ

ข่ายงานความหมายที่ใช้ในการสร้างระบบผู้เชี่ยวชาญคือ วัตถุ-ลักษณะประจำ-ค่า (object-attribute-value) วัตถุจะเป็นเอนทิตี (entity) ของแนวคิดแทนด้วยบัพ ลักษณะประจำ (attribute) เป็นคุณสมบัติทั่วไปของวัตถุแทนด้วยเส้นตรง ส่วนค่าจะเป็นค่าข้อมูลเฉพาะ

2. **กรอบ** (Frames)

กรอบมีส่วนประกอบที่รวบรวมช่อง (slots) ที่เก็บลักษณะประจำ (attribute) ที่อธิบายวัตถุ ชั้นของวัตถุ (class of objects) สภาพการณ์ การกระทำ และเหตุการณ์ต่างๆไว้ กรอบจะเก็บลักษณะที่สำคัญ เพื่อแทนโครงสร้างของความสัมพันธ์ที่เป็นประโยชน์โดยการจับ (capture) ซึ่งเป็นวิธีเดียวกันกับการได้ข้อมูลมาของฐานความรู้ในระบบผู้เชี่ยวชาญนั่นเอง

กรอบแตกต่างจากข่ายงานความหมายตรง ค่า (value) เป็นกลุ่มค่าที่รวมกันเป็นหน่วยเดียวเรียกว่า กรอบ แต่จะสามารถอธิบายสถาพการณ์ หรือวัตถุที่มีความสลับซับซ้อน หรือเป็นชุดของเหตุการณ์ทั้งหมดในหน่วยเดียวได้

กรอบหนึ่งกรอบจะประกอบด้วยความรู้หลายๆส่วน แต่ละส่วนคือช่อง ช่องแต่ละช่องจะมีคุณสมบัติเฉพาะของมันเอง เช่นเป็นประโยคที่ประกาศไว้(declarative statement)สำหรับการนำข้อมูลลักษณะประจำเข้ามาสู่กรอบ อาจเป็นค่าที่ได้กำหนดไว้แน่นอน (default values) ก็ได้ อีกลักษณะหนึ่งจะเป็น ประโยคกระบวนงาน (procedure statement) ที่เป็นชุดของคำสั่งซึ่งเมื่อถูกกระทำการ(excute) แล้วจะได้ผลเป็นความรู้สำหรับใส่ในช่องของกรอบ

ข้อดีของการแทนความรู้ด้วยกรอบคือ จะได้ฐานความรู้ที่รวบรัดและรัดกุม ทำให้ประหยัดเวลาในการค้นหาสารสนเทศเฉพาะที่ต้องการได้

การแทนความรู้ด้วยกรอบ มีความเหมาะสม ที่จะใช้ในกรณีต่อไปนี้คือ โดเมนของความรู้จะไม่เป็นอิสระแต่เกี่ยวข้องกัน เช่นโดเมนสามารถแทนกลุ่มของกระบวนการที่ การนำเข้ากระบวนการหนึ่ง ขึ้นอยู่กับผลที่ได้จากกระบวนการอื่น ๆ

3. **ระบบฐานแห่งกฎ** (Rules-based system) **หรือ กฎการผลิต** (Production rules)

กฎการผลิตเป็นกฎแบบมีเงื่อนไข IF-THEN และ IF-THEN-ELSE ที่จำลองสภาพของปัญหาหรือ สถานการณ์ของปัญหาที่กำหนดขึ้นประโยค IF อธิบาย วัตถุ(object) สภาพและตำแหน่ง ถ้า ประโยค IF เป็นจริงประโยค THEN ของกฎการผลิตจะทำงาน ถ้ากฎการผลิตมีประโยค ELSE ประกอบด้วย เมื่อประโยค IF ผิดประโยค ELSE ของกฎการผลิตจะทำงาน

ประโยค IF เรียกว่า ข้อตั้ง (Premise) ส่วนประโยค THEN และประโยค ELSE เรียกว่าการกระทำ (action) กฎการผลิต มี รูปแบบดังนี้

IF [premises] THEN [actions] ELSE [actions]

กฎจะทำงาน เมื่อเงื่อนไขทั้งหมดในข้อตั้งของกฎนั้นเป็นไปตามสภาพปัจจุบัน กฎโปรดักชันถูกเรียกว่าไฟรด์(fired) หรือกระทำการเมื่อมีการกระทำเกิดขึ้น กฎโปรดักชันโดยทั่วไปจะแทนความรู้ด้วยโครงสร้างของ วัตถุ-ลักษณะประจำ-ค่า (Object-Attribute-Value) หรือ โอวีเอ(O-A-V) โดยที่ข้อตั้งและการกระทำสามารถแสดงในโครงสร้างของ โอวีเอได้

กฎการผลิตที่เป็นระบบฐานแห่งกฎ (Rule-based system) จะมีลักษณะโครงสร้างโดยทั่วไป 3 ส่วนดังนี้ คือ

1. ฐานแห่งกฎ (Rule base) ซึ่งประกอบด้วยเซตของกฎการผลิต
2. โครงสร้างข้อมูล (data structure) บริบท (context) ซึ่งแสดงลักษณะปัญหาเฉพาะเรื่องไว้ในฐานความรู้
3. เครื่องอนุมาน (Inference engine) เป็นตัวควบคุมการทำงานของระบบ

ข้อดีของระบบฐานแห่งกฎ (Rule-based systems)

1. การใช้ส่วนจำเพาะ (Modular) คือกฎการผลิตแต่ละกฎจะสามารถ เพิ่ม ลบ หรือเปลี่ยนแปลง ได้ โดยไม่มีผลกระทบต่อความสัมพันธ์กับกฎข้ออื่น เพราะแต่กฎมีความสมบูรณ์ในตัวเอง และตัวแปรที่ใช้ในกฎหนึ่งมีขอบเขตแค่ภายในกฎนั้นเท่านั้น
2. เป็นเอกรูป (Uniform) คือการแทนความรู้เป็นรูปแบบเดียวกันตลอด ทำให้เข้าใจได้ง่าย และรูปแบบ IF-THEN เป็นรูปแบบที่มนุษย์มีความคุ้นเคยอยู่แล้ว
3. เป็นธรรมชาติ (Natural) คือกฎการผลิตที่สร้างขึ้น จะมีโครงสร้าง แบบเดียวกับความคิดของมนุษย์ ในการแก้ไขปัญหา

ข้อเสียของระบบฐานแห่งกฎคือ ยากต่อการบันทึกความสัมพันธ์ระหว่างความรู้และผลกระทบซึ่งกันและกันของความรู้ที่สัมพันธ์กัน การมีส่วนจำเพาะสูงทำให้บันทึกโครงสร้างของความรู้ได้ยาก

4. **แคลคูลัสภาคแสดง** (Predicate calculus)

แคลคูลัสภาคแสดงมีพื้นฐานอยู่บนความจริง(truth) และกฎของการอนุมาน (rules of inference) ทางตรรกะที่ใช้สัญญาลักษณ์แสดง แคลคูลัสภาคแสดงเป็นส่วนขยายของตรรกศาสตร์ประพจน์ (propositional logic) ไม่เพียงแต่ข้อสรุปว่าเป็นความจริง(truth) หรือเป็นความเท็จ(false) เท่านั้น แต่ยังเป็นประโยคแทนลักษณะเฉพาะของวัตถุอีกด้วย

ตรรกวิทยาเป็นศาสตร์ที่เก่าแก่มีมาตั้งแต่สมัยกรีกโบราณ ส่วนใหญ่เป็นสิ่งที่นักปรัชญาศึกษาค้นคว้า ถูกนำมาใช้ในการแก้ไขปัญหาและได้รับการพัฒนาอย่างต่อเนื่อง ทำให้ตรรกวิทยาเป็นศาสตร์เกี่ยวกับการคิดอย่างสมเหตุสมผล ที่มีรากฐานสมบูรณ์และถูกต้องอย่างเคร่งครัด มีคำศัพท์และมีภาษาเฉพาะ

แคลคูลัสภาคแสดงจะเกี่ยวข้องกับ ตัวแปร (variable) ภาคแสดง (predicate) การเชื่อมต่อประโยค (sentential connectives) ตัวบ่งคุณลักษณะ (qualifier) และฟังก์ชัน(function) ตัวแปรจะแทนวัตถุ สิ่งของ และประโยคในคำถาม ภาคแสดงจะอธิบายความเกี่ยวพันธ์กันของตัวแปรที่พิจารณานำมาสร้างเป็นประโยค การเชื่อมต่อประโยคจะถูกนำมาใช้กับประโยคที่มีความสลับซับซ้อน ได้แก่ และ(and) หรือ(or) ไม่(or) ส่อให้เห็น(implies) และสมมูล (equivalent) ตัวบ่งคุณลักษณะจะกระทำกับตัวแปรในทุกวัตถุ ฟังก์ชันจะคล้ายกับภาคแสดงแต่จะคืนวัตถุที่เกี่ยวข้องให้

แคลคูลัสภาคแสดงมีข้อดีดังนี้

1. มีรากฐานทางทฤษฎีที่ดีถูกต้องและเคร่งครัด เหมาะสมอย่างยิ่งต่อการจัดความรู้ และการรับเอาความรู้

2. การใช้ส่วนเฉพาะ (modular) ประโยคสามารถ เพิ่ม ลบ หรือดัดแปลงได้โดยไม่มีผลกระทบต่อประโยคอื่นเช่นเดียวกับกฎการผลิต

แคลคูลัสภาคแสดงมีข้อเสียดังนี้

1. เมื่อจำนวนความจริง (fact) ในฐานความรู้เพิ่มมากขึ้น จำนวนวิธีการเกี่ยวพันธ์กันของความจริงจะยิ่งเพิ่มขึ้นเป็นเลขชี้กำลัง(exponential)

2. แม้ว่าการแทนความรู้แบบแคลคูลัสภาคแสดงจะง่ายก็ตาม แต่ ในการหาวิธีการแก้ไขปัญหา อาจจะไม่ชัดเจน หรือหาคำตอบไม่ได้

วิธีการค้นหา (Search Methods)

เมื่อปัญหาถูกนำเข้าสู่โมเดลการแทน(representation model) คอมพิวเตอร์ต้องการกลยุทธ์ในการแก้ปัญหา การค้นหา (searching) เป็นวิธีการที่สำคัญอย่างหนึ่งที่นำมาใช้ในการค้นหาคำตอบ การค้นหาจะเกี่ยวข้องกับการเลือกวิถี(path) ในฐานความรู้ที่นำไปสู่คำตอบ ถ้าเลือกวิถีไม่ถูกต้องก็จะเปลี่ยนไปเป็นวิถีใหม่ จนกว่าจะพบคำตอบ การค้นหาแบ่งเป็น 2 ประเภทคือ

1. **การค้นหาแบบไม่มีสิ่งชี้นำ** (Blind searches)

การค้นหาแบบไม่มีสิ่งชี้นำ เป็นการค้นหาที่ไม่มีความรู้ที่จะช่วยแนะแนวทางในการตัดสินใจในการค้นหา ไม่มีโดเมนสารสนเทศที่จำเป็นต่อการค้นหา เป็นการค้นหาที่ใช้เวลานานและค่าใช้จ่ายสูง ได้แก่

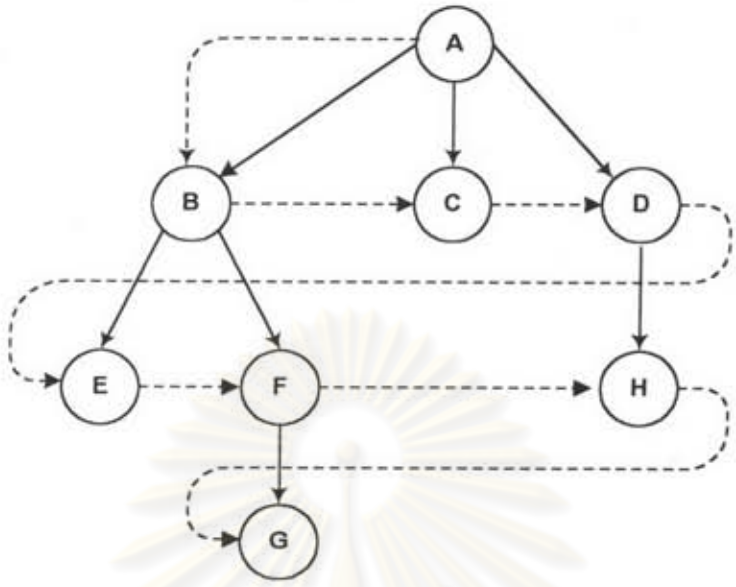
1.1 **การค้นแบบแจงกรณี** (Exhaustive)

เป็นวิธีการค้นหาที่ใช้วิถี(path) ทั้งหมดทุกกรณีที่จะเป็นไปได้ นำมาวิเคราะห์เป็นต้นไม้การตัดสินใจหรือเครือข่ายการตัดสินใจ ซึ่งใช้เวลาและค่าใช้จ่ายสูง วิธีการนี้จะไม่เป็นจริงหรือเป็นไปไม่ได้ในบางปัญหาที่มีขนาดใหญ่ แต่ก็ใช้ในกรณีที่ต้องการความแน่นอน และไม่มีปัญหาเรื่องค่าใช้จ่าย

1.2 **การค้นในแนวกว้าง** (Breath-first)

เป็นการค้นหาที่ไม่ใช้ข้อมูลอะไรมาเป็นเกณฑ์ในการค้นหา คือ จากบัพเริ่มต้นให้สร้างทางเลือกทุกทางที่เป็นไปได้ แล้วทำการตรวจสอบดูว่าบัพที่เกิดจากทางเลือกที่สร้างขึ้นนั้นเป็นคำตอบหรือบัพเป้าหมายหรือไม่ ถ้าเป็นก็หยุดการค้นหา แต่ถ้าไม่เป็นก็สร้างวิถีจากทุกบัพที่ตรวจสอบแล้วลงไปอีกระดับหนึ่งแล้วทำการตรวจสอบอีก ทำเช่นนี้จนกว่าจะพบบัพเป้าหมายหรือจนกระทั่งหมดบัพค้นหา นิยมใช้กับกลยุทธ์การควบคุมแบบลูกโซ่ไปข้างหน้า (Foreward chaining) จากตัวอย่างรูปที่ 2.2 บัพเริ่มต้นเป็น A และบัพคำตอบเป็น G จะมีการเลือก 8 ขั้น จาก A ไป B ไป C ไป D ไป E ไป F ไป H จนถึง G

ข้อดีของการค้นในแนวกว้างคือ ไม่มีการเข้าลูพแบบวนเวียนไม่สิ้นสุด เมื่อผ่านไปเวลาหนึ่งการค้นหาจะสิ้นสุดลง และถ้าในต้นไม้การค้นหา มีคำตอบอยู่ การค้นในแนวกว้างให้ประกันว่าจะพบคำตอบนั้น



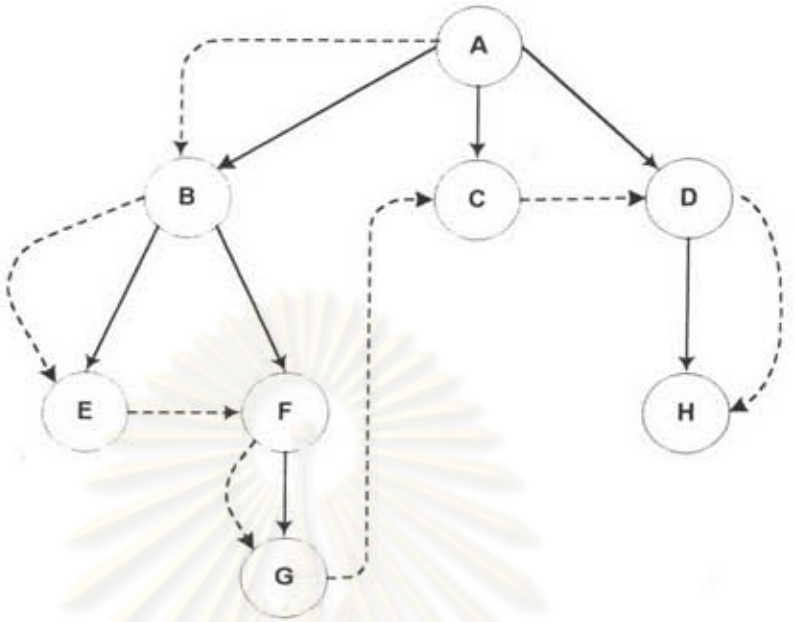
Step	Choices	Selection
1	(ABCDEFGH)	A
2	(BCDEFGH)	B
3	(CDEFGH)	C
4	(DEFGH)	D
5	(EFGH)	E
6	(FGH)	F
7	(GH)	H
8	(G)	G

รูปที่ 2.2 แสดงการค้นในแนวกว้าง (Breadth - first search)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

1.3 **การค้นในแนวลึก** (Depth-first)

เป็นการค้นหาที่ไม่ใช้ข้อมูล แต่มีกฎเกณฑ์ตายตัวในการค้นหาคำตอบคือ จากบัพในต้นไม้การค้นหาให้เลือกวิถีที่อยู่ซ้ายสุดก่อน แล้วเลื่อนไปยังบัพที่อยู่ใต้ไปหนึ่งระดับทำเช่นนี้จนกว่าจะพบบัพที่เป็นคำตอบ หรือลงไปจนถึงระดับล่างสุดที่กำหนดไว้แล้วจึงย้อนกลับ (backtack) ขึ้นมาหนึ่งระดับเพื่อตรวจวิถีที่อยู่ขวามือถัดไป ทำเช่นนี้จนกว่าจะพบคำตอบ นิยมใช้กับกลยุทธ์การควบคุมแบบลูกโซ่ย้อนกลับ (Backward chaining) จากตัวอย่างรูปที่ 2.3 บัพเริ่มต้นคือ A และบัพคำตอบคือ H จะมีการเลือก 8 ขั้น จาก A ไป B ไป E ไป F ไป G ไป Cไป D จนถึง H



Step	Choices	Selection
1	(ABCDEFGH)	A
2	(BCDEFGH)	B
3	(CDEFGH)	E
4	(CDFGH)	F
5	(CDGH)	G
6	(CDH)	C
7	(DH)	D
8	(H)	H

รูปที่ 2.3 แสดงการค้นในแนวลึก (Depth - first search)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

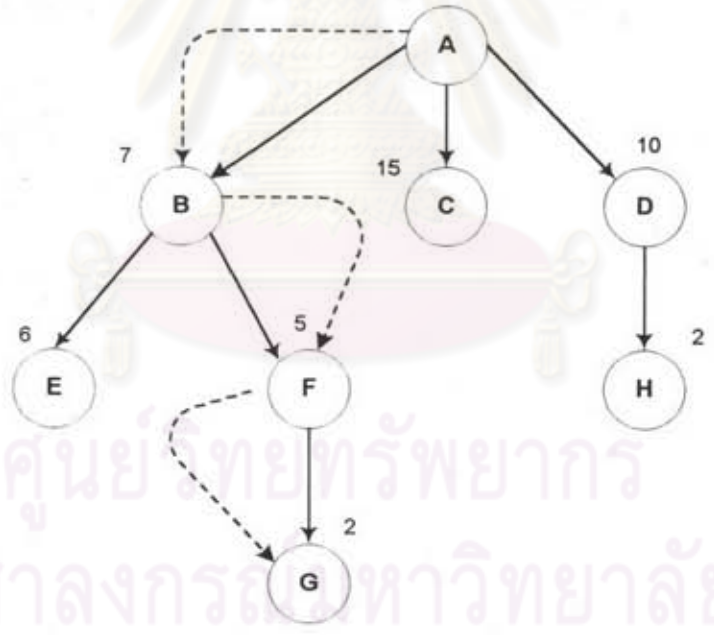
ข้อดีของการค้นในแนวลึกคือ ใช้เนื้อที่หน่วยความจำน้อย เพราะต้องเก็บข้อมูลเกี่ยวกับบัพเพียงจำนวนความลึกของการค้นหาเท่านั้น และสามารถทำการย้อนกลับได้ จะมีโอกาสพบคำตอบโดยไม่ต้องตรวจสอบบัพจำนวนมาก ทำให้ประหยัดเวลาการค้นหา ถ้ามีหลายคำตอบและต้องการเพียงคำตอบเดียว การค้นในแนวลึกสามารถหยุดได้ทันทีเมื่อพบคำตอบใดคำตอบหนึ่ง แต่มีข้อเสียตรงการค้นหาอาจเป็นไปอย่างไม่สิ้นสุด ในกรณีที่มีวงวน(loop) ในต้นไม้การค้นหา หรือกรณีที่ไม่มีการกำหนดความลึกของการค้นหา ในทั้งสองกรณีจะเสียเวลาค้นหาโดยไม่ได้คำตอบอะไรเลย และไม่รับประกันว่าจะให้คำตอบแก่เรา

2. **การค้นหาแบบศึกษาสำนึก** (Heuristic searches)

การค้นหาแบบศึกษาสำนึก เป็นการค้นหาแบบแข็งเกร่ง โดยการวิเคราะห์โดเมนสารสนเทศเฉพาะเพื่อการเลือกวิถี(path)ที่ดีกว่า จะช่วยลดขนาดลงไปมาก สารสนเทศบริบท(contextual information) จะถูกใช้ในการเลือกวิถีการค้นหาบัพถัดไป ช่วยหาตัวบัพตัวถัดไปแทนการเลือกดูทั้งหมดทุกกรณี ได้แก่

2.1 **การค้นแบบปีนเขา** (Hill climbing)

เป็นการค้นหาแบบเดียวกับการค้นหาในแนวลึกแต่การเลือกวิถีไม่เป็นไปตามลำดับแต่ขึ้นกับการประมาณค่า หาบัพต่อไปที่อยู่ใกล้คำตอบมากที่สุด จะเหมาะสมกับกรณีที่มีวิธีการบอกระยะทางที่ห่างจากเป้าหมายได้ ดังแสดงตัวอย่างในรูปที่ 2.4 เมื่อบัพเริ่มต้นเป็น A และบัพคำตอบเป็น G



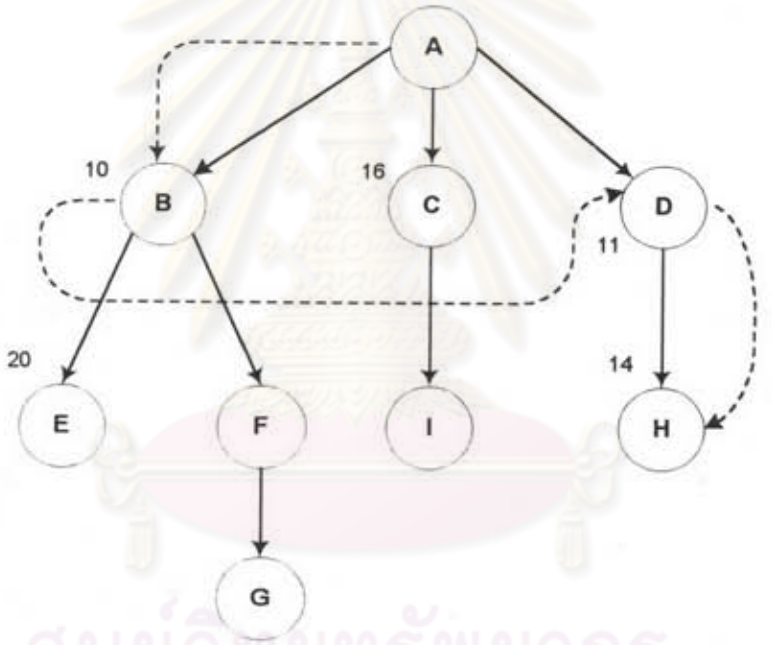
Step	Choices	Selection
1	(A)	A
2	(BCD)	B
3	(EF)	F
4	(G)	G

รูปที่ 2.4 แสดงการค้นแบบปีนเขา (Hill - climbling search)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

2.2 **การค้นแบบดีที่สุดก่อน** (Best-first)

การค้นหาแบบดีที่สุดก่อนเป็นการค้นหาที่ใช้ข้อมูลช่วยประกอบในการเลือกทางเลือกต่อไปข้อมูลที่ใช้ คือ คุณค่าของแต่ละบัพ การค้นหาแบบดีที่สุดเอาคุณค่าของแต่ละบัพไปผสมกับการค้นหาแบบกว้างก่อน โดยใช้คุณค่าของแต่ละบัพเป็นตัวกำหนดลำดับในการตรวจสอบในแต่ละระดับความลึก แทนที่จะเป็นลำดับจากซ้ายไปขวา หรือจากบัพแรกไปบัพหลังตามลำดับการสร้างบัพอย่างในการค้นหาแบบกว้างก่อน ดังแสดงตัวอย่างในรูปที่ 2.5 เมื่อบัพเริ่มต้นเป็น A และบัพคำตอบเป็น G



Step	Choices	Selection
1	(A)	A
2	(BCD)	B
3	(EFCD)	F
4	(EFCH)	G

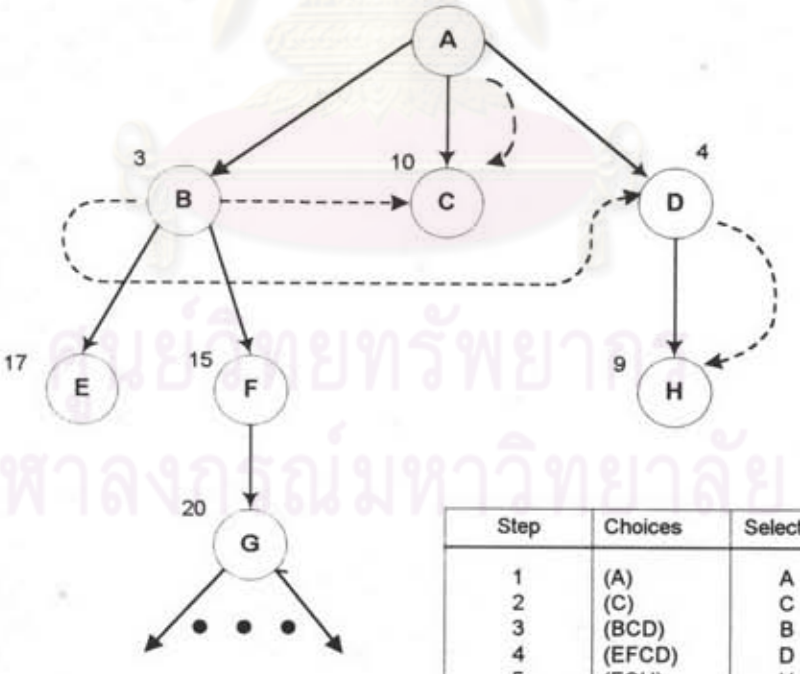
รูปที่ 2.5 แสดงการค้นแบบดีที่สุดก่อน (Best - first search)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

โดยทั่วไปการค้นหาแบบดีที่สุดก่อนมักจะหาทางเลือกที่ดีที่นำไปสู่บัพเป้าหมาย ประเด็นที่สำคัญในการค้นหาแบบดีที่สุดก่อนคือการกำหนดวิธีการประเมินคุณค่าของแต่ละบัพ ถ้ากำหนดได้ดีจะทำให้การค้นหามีประสิทธิภาพ

2.3 **การค้นแบบขยายและจำกัดเขต** (Branch-and-bound)

เป็นการค้นหาที่แต่ละขั้นตอน จะหาวิถีสำรองใหม่ที่สั้นที่สุด แล้วก็ขยายไปยังระดับถัดไป หาวิถีสำรองใหม่แบบที่สั้นที่สุดอีก ทำซ้ำไปทุกวิถี การค้นหาจะเสร็จสิ้นสมบูรณ์เมื่อวิถีสำรองที่สั้นที่สุด ยาวกว่า วิถีหลักที่สั้นที่สุด ยกตัวอย่างรูปที่ 2.6 บัพเริ่มต้นคือ A บัพเป้าหมายคือ H จะเริ่มจากบัพ A สร้างวิถีสำรองที่สั้นที่สุดคือไปบัพ C จากบัพ Cระหว่าง B,D เลือกไปบัพ B จากบัพ Bระหว่างบัพE,F,C,D เลือกบัพ D จากบัพ D ระหว่าง F,C,H เลือกบัพ H วิถีสำรองจาก A ไปC ไปB ไปD และไปH ยาวกว่าวิถีหลัก จาก A ไปD ไปH ดังนั้นวิถีที่เหมาะสมที่สุดคือ A-D-H



Step	Choices	Selection
1	(A)	A
2	(C)	C
3	(BCD)	B
4	(EFCD)	D
5	(FCH)	H
6	(GCH)	H

รูปที่ 2.6 แสดงการค้นแบบขยายและจำกัดเขต (Branch - and - bound)

2.4 **การค้นแบบขั้นตอนวิธี เอสตาร์** (A* algorithm)

การค้นหาแบบ A* เป็นการค้นหาที่ใช้ข้อมูลช่วยประกอบในการเลือกทางเลือกต่อไปที่คล้ายกับการค้นหาแบบดีที่สุดก่อนคือ ใช้การค้นหาแบบกว้างก่อนเป็นหลัก แต่แทนที่จะใช้ข้อมูลการประเมินคุณค่าของแต่ละบัพอย่างเดียวดังในการค้นหาแบบดีที่สุดก่อน การค้นหาแบบ A* ใช้ข้อมูลเกี่ยวกับค่าใช้จ่ายในการกระทำที่เลือกประกอบด้วย ถ้าหากหน่วยมาตราที่ใช้ในการประเมินคุณค่าและในการคำนวณค่าใช้จ่ายเป็นหน่วยเดียวกันหรือสามารถเปลี่ยนแปลงให้เป็นหน่วยเดียวกันได้ เราก็สามารถรวมค่าคุณค่าและค่าใช้จ่ายเข้าด้วยกัน แล้วใช้ค่ารวมนี้เป็นตัวนำในการค้นหาแบบกว้างก่อน

<u>กลยุทธ์การควบคุม</u> (Control Strategies)

ภายหลังจากที่เลือกวิธีการค้นหาคำตอบแล้ว สภาพปัญหาที่แตกต่างกัน จำเป็นต้องเลือกใช้วิธีการหาเหตุผลที่แตกต่าง สำหรับการตัดสินใจว่าจะต้องทำอะไรต่อไป วิธีการควบคุมการหาเหตุผลที่สำคัญมี 2 แบบคือ

1. **<u>แบบลูกโซ่ไปข้างหน้า</u>** (FORWARD CHAINING)

เป็นการเริ่มต้นค้นหาจากข้อมูล(data-driven)ของข้อกำหนด แล้วค้นหาฐานความรู้ทางเลือกที่เหมาะสมเป็นลำดับไปจนกระทั่งพบคำตอบ บางทีเรียกว่า การประมวลผลจากล่างขึ้นบน (bottom-up processing) หรือ อะไร-ถ้า ("what-if") ระบบจะเริ่มจากความจริง(fact) ทำการค้นหากฎ(rule) โดยตรวจสอบกับข้อตั้ง(premise)ของแต่ละกฎ ถ้าข้อตั้งตรงกันทั้งหมดก็จะเลือกการกระทำของกฎนั้นเป็นคำตอบ การหาคำตอบแบบลูกโซ่ไปข้างหน้า(forward chaining) เหมาะสำหรับปัญหาที่มีสภาวะเริ่มต้นง่ายๆ แต่มีเป้าหมายที่ยุ่งยากซับซ้อน หรือ มีหลายเป้าหมาย ที่เป็นไปได้ เช่น การคาดการณ์(prediction) การออกแบบ(design) และ การวางแผน(planning)

2. **<u>แบบลูกโซ่ย้อนกลับ</u>** (Backward chaining)

เป็นวิธีการควบคุมการหาเหตุผล ที่เริ่มจากเป้าหมาย มาทำการค้นหากฎในฐานความรู้ ที่มีค่าความจริงในส่วนการกระทำของกฎตรงกับค่าความจริงที่เป็นเป้าหมาย แล้วทำการทดสอบเงื่อนไขของกฎข้อนั้นกับค่าจริงในหน่วยความจำ ถ้าส่วนเงื่อนไขของกฎข้อนั้นเป็นจริง ก็

เป็นการพิสูจน์ค่าความจริงในส่วนกระทำของกฎข้อนั้นเป็นจริง และจะนำค่าความจริงที่ได้ไปเก็บไว้ในหน่วยความจำ แต่ถ้าไม่มีค่าความจริงในหน่วยความจำที่จะใช้ทดสอบส่วนเงื่อนไขของกฎข้อนั้น ก็จะนำค่าความจริงในส่วนเงื่อนไขของกฎข้อนั้นมาทำเป็นเป้า-หมายย่อย แล้วกลับไปเริ่มต้นนำค่าความจริงที่เป็นเป้าหมายย่อยมาทำการค้นหากฎข้ออื่นๆ ในฐานความรู้ต่อไป ทำเช่นนี้จนกระทั่งสามารถพิสูจน์ค่าความจริงที่เป็นเป้าหมายได้ว่าเป็นจริงหรือเท็จ วิธีการควบคุมการหาเหตุผลแบบลูกโซ่ย้อนกลับ เหมาะสำหรับปัญหาที่มีเป้าหมายจำนวนน้อย แต่มีจุดเริ่มต้นจำนวนมาก อันได้แก่ การวินิจฉัย(diagnosis) การตรวจจับ(monitoring) และการซ่อมแซม(repair)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย