บทที่ 2

ทฤษฎี

ภาษาโปรล็อก (PROLOG LANGUAGE)

ภาษาคอมพิวเตอร์ที่พัฒนากันมาจากอดีตจนถึงปัจจุบันนี้ ส่วนใหญ่จะเป็นภาษาที่มีการกำหนดลำดับขั้นตอนของการทำงานที่แน่นอนตายตัว ซึ่งเราเรียกกันว่าเป็น PROCEDURAL PROGRAMMING LANGUAGES เช่น ภาษาฟอร์ทราน ภาษาเบสิค ภาษาซี เป็นต้น ในการใช้งานภาษาเหล่านี้ ผู้ที่จะเขียนโปรแกรมจะต้องกำหนดลำดับขั้นตอน สำหรับการแก้ปัญหาขึ้นเสียก่อน จากนั้นจึงเริ่มเขียนโปรแกรมให้สามารถทำงานตามลำดับขั้นตอนที่เราได้กำหนดไว้แล้วอย่างแน่นอน โดยไม่มีความยืดหยุ่นในการแปรเปลี่ยนลำดับขั้นตอนได้ แต่ก็ยังมีกลุ่มของปัญหาอีกกลุ่มหนึ่ง ซึ่งไม่สามารถที่จะใช้ภาษาคอมพิวเตอร์ประเภทนี้ในการช่วยแก้ปัญหาได้ หรือสามารถกระทำได้ยากและไม่มีประสิทธิภาพ ซึ่งกลุ่มของปัญหาประเภทนี้จัดอยู่ในจำพวกของปัญญาประดิษฐ์ (ARTIFICIAL INTELLIGENCE, AI) ดังนั้น จึงได้มีการพัฒนาภาษาคอมพิวเตอร์ขึ้นมา เพื่อที่จะใช้กับการแก้ปัญหาประเภทนี้ เช่น ภาษาโปรล็อก ภาษาลิส เป็นต้น

ภาษาโปรล็อกย่อมาจากคำว่า PROGRAMMING IN LOGIC เป็นการเขียนโปรแกรมโดยวิธีทางตรรกวิทยา มีคุณลักษณะตรงกันข้ามกับพวก PROCEDURAL PROGRAMMING ก็คือความสามารถในการวินิจฉัย และสรุปความเห็นโดยการชักเหตุผลจากข้อเท็จจริง อาศัยฐานความรู้ (KNOWLEDGE BASE) ที่มีอยู่ ประกอบกับกลไกการวินิจฉัย (INFERENCE ENGINE) ที่เราจะสร้างให้กับโปรแกรมขึ้นมา โดยอาศัยการทำงานทางด้านตรรกวิทยา ดังนั้นภาษาโปรล็อกจึงเหมาะสำหรับการนำไปใช้แก้ปัญหาที่มีโครงสร้างไม่แน่นอนตายตัว และมีการเก็บข้อมูลในลักษณะที่เปลี่ยนแปลงไปมา ในทางกลับกันก็ไม่เหมาะที่จะนำไปใช้กับปัญหาที่มีการคำนวณ หรือทำงานตามลำดับขั้นตอนที่แน่ชัดแล้ว ตัวอย่างความเหมาะสมของการใช้ภาษาโปรล็อก เช่น

-ระบบผู้เชี่ยวชาญ (EXPERT SYSTEM)

-การโต้ตอบด้วยภาษาธรรมชาติ (NATURAL LANGUAGE PROCESSING)

-การสร้างเกมที่ต้องใช้การชักเหตุผล
-การประยุกต์ใช้กับหุ่นยนต์อุตสาหกรรม (ROBOTICS)
-โปรแกรมอื่น ๆ

<u>หลักการทำงานขั้นต้นของภาษาโปรล็อก</u>

โปรล็อกเป็นภาษาที่ใช้ตรรกวิทยาในการตัดสินใจ ฉะนั้นส่วนใหญ่ของโปรแกรมจึงเป็นการรวบรวมฐานความรู้เกี่ยวกับเรื่องใดเรื่องหนึ่งโดยเฉพาะ ฐานความรู้นี้อาจจะเรียกได้ว่าเป็นฐานข้อมูล (DATABASE) และในโปรล็อกเราจะเขียนฐานข้อมูลในรูปแบบของข้อเท็จจริงและกฎ (FACTS AND RULES) โดยโปรล็อกสามารถบรรยายข้อเท็จจริงให้อยู่ในรูปของสัญลักษณ์ของความสัมพันธ์ได้ ตัวอย่างเช่น ถ้าต้องการจะบรรยายข้อเท็จจริงที่ว่า เบรครถมีเสียง

The brake has noise.

จะสามารถแสดงในรูปแบบของโปรล็อกได้เป็น

has(brake,noise).

รูปแบบที่เราแสดงเป็นข้อเท็จจริงในภาษาโปรล็อกข้างบนนี้เรียกว่า อนุประโยค (CLAUSE) ตัวอย่างนี้ has มีชื่อเรียกว่าความสัมพันธ์ (RELATION) ส่วน brake และ noise เรียกว่าวัตถุ(OBJECT) ถ้าอนุประโยคนี้ไม่รวมเครื่องหมายมหัพภาพ (.) จะเรียกว่า เพรดิเคท (PREDICATE) และคำที่อยู่ในวงเล็บของเพรดิเคทเรียกว่าอาร์กิวเมนท์ (ARGUMENT)

โปรแกรมโปรล็อกจะประกอบด้วยส่วนต่าง ๆ คือ

1. ส่วนโปรแกรมที่เชื่อมโยงกัน (INCLUDE)
2. ส่วนโดเมน (DOMAINS)
3. ส่วนฐานข้อมูล (DATABASE)
4. ส่วนเพรดิเคท (PREDICATES)
5. ส่วนเป้าประสงค์ (GOAL)
6. ส่วนอนุประโยค (CLAUSES)

1. ส่วนโปรแกรมที่เชื่อมโยงกัน (INCLUDE) เป็นการบ่งบอกการนำโปรแกรมอื่นที่ต้องการมาใช้ร่วมกับโปรแกรมที่กำลังทำงานอยู่

2. ส่วนโดเมน (DOMAINS) เป็นการที่จะบอกหรือจำแนกว่าวัตถุในแต่ละเพรดิเคท

เป็นวัตถุชนิดใด เช่น วัตถุประเภทเลขจำนวนเต็ม วัตถุประเภทเลขจำนวนจริง วัตถุประเภทสัญลักษณ์ วัตถุประเภทสตริง วัตถุประเภทไฟล์ วัตถุประเภทตัวอักษรหรือตัวอักขระ และวัตถุประเภทลิสท์ (จำนวนเทอมหรือคำหลาย ๆ คำที่นำมาเรียงกัน)

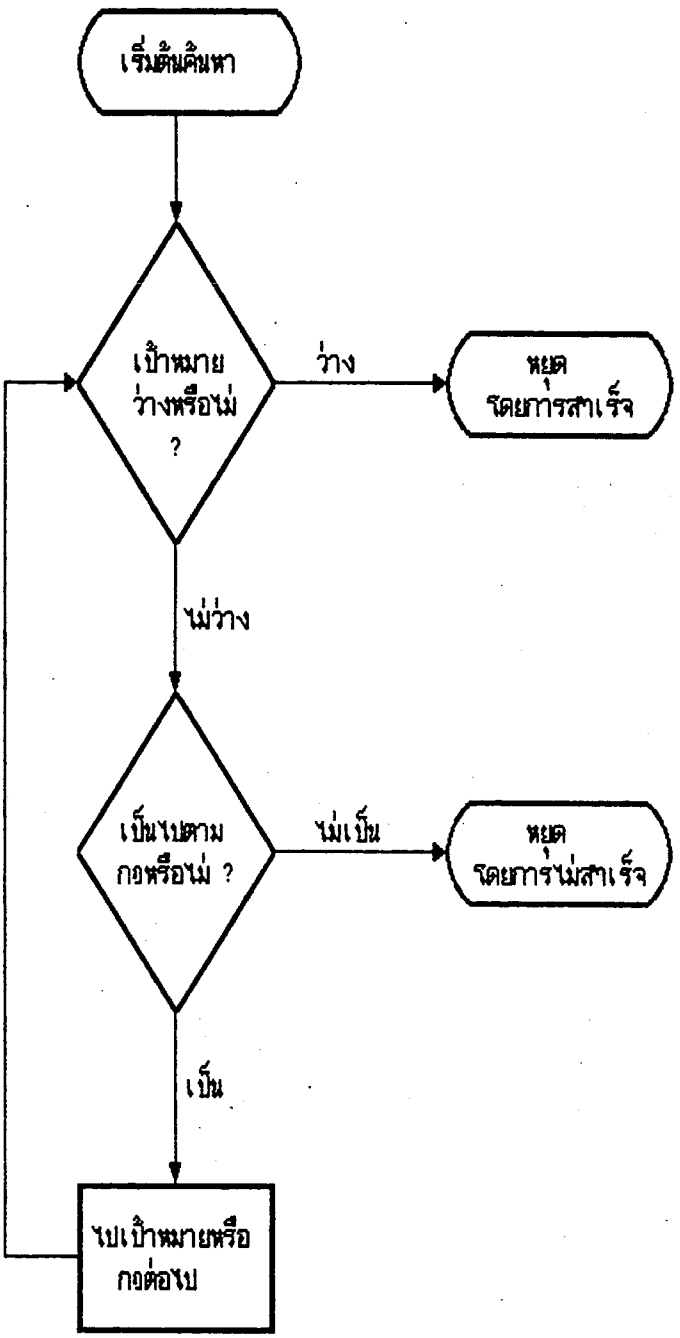
3. ส่วนฐานข้อมูล (DATABASE) เป็นการประกาศและแบ่งแยกฐานข้อมูลที่จะนำมาใช้เป็นส่วน ๆ ไม่ให้ปนกัน

4. ส่วนเพรดิเคท (PREDICATE) เป็นส่วนที่ประกาศรูปแบบของความสัมพันธ์และอาร์กิวเมนท์ต่าง ๆ

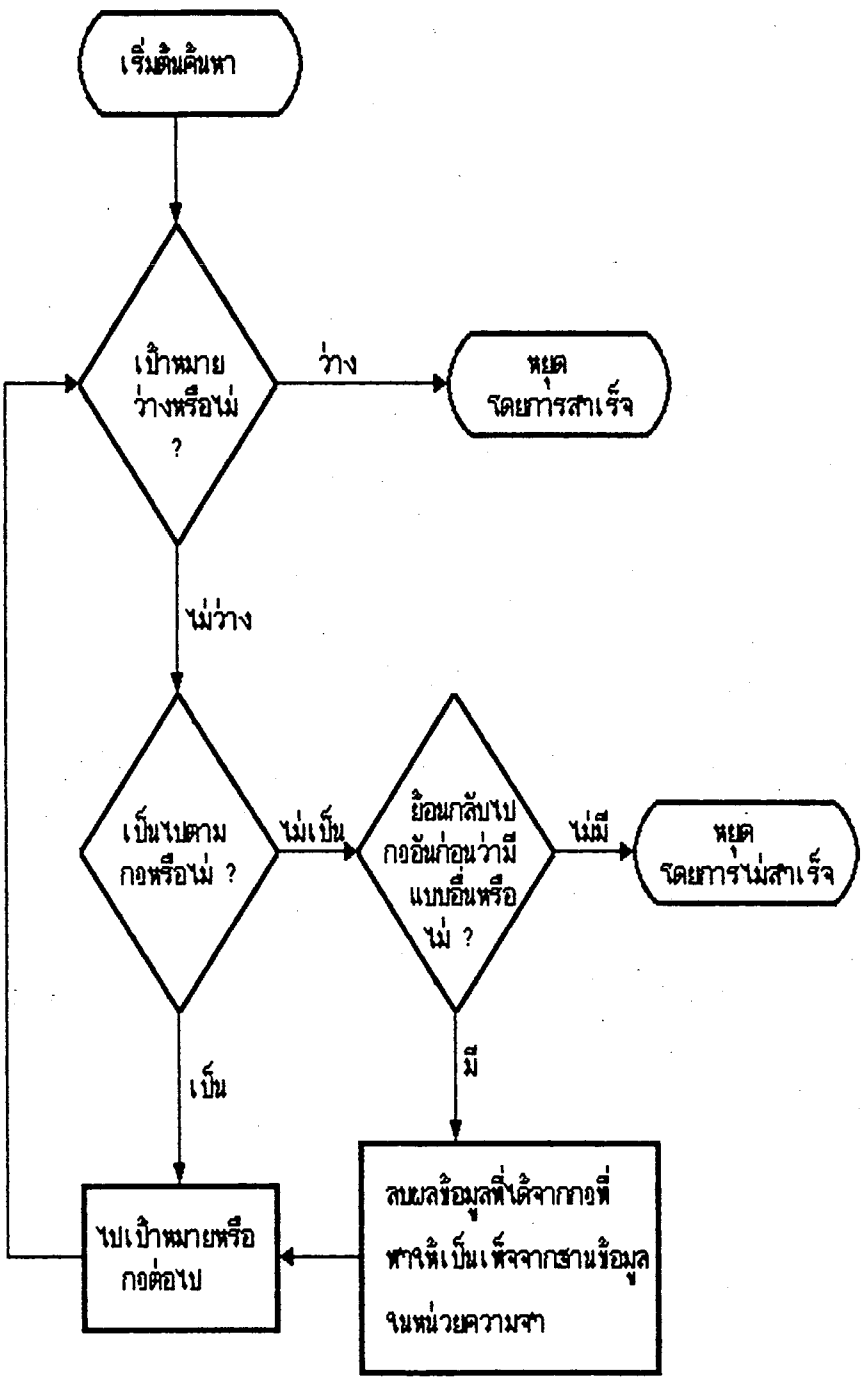
5. ส่วนเป้าประสงค์ (GOAL) เป็นส่วนที่จะบอกวัตถุประสงค์หรือเป้าหมายที่ต่อการดำเนินในโปรแกรม อาจจะมีหลาย ๆ เป้าหมายก็ได้

6. ส่วนอนุประโยค (CLAUSES) เป็นส่วนประกอบของข้อเท็จจริงและกฎต่าง ๆ ซึ่งกฎนั้นคือประโยคที่บ่งบอกให้ทราบว่า ข้อเท็จจริงข้อหนึ่งจะเป็นจริงหรือไม่จะขึ้นอยู่กับข้อเท็จจริงอื่น ๆ ที่ตามมา

ขั้นตอนการทำงานพื้นฐานของโปรล็อก เมื่อกดปุ่มรันโปรแกรมโปรล็อกจะทำการอ่านในส่วนที่เป็นเป้าประสงค์ ถ้าไม่มีส่วนเป้าประสงค์ในโปรแกรม ก็จะมีการให้ป้อนเป้าประสงค์ที่ต้องการเข้าไป เมื่อมีเป้าประสงค์แล้วโปรแกรมก็จะทำการจับคู่ระหว่างเป้าประสงค์กับข้อเท็จจริงหรือกฎที่อยู่ในฐานข้อมูล ถ้าสามารถจับคู่ได้ก็จะไปทำเป้าประสงค์ต่อไปจนกระทั่งหมดเป้าประสงค์ แล้วโปรแกรมก็จะหยุดด้วยความสำเร็จหรือเป็นจริง แต่ถ้าไม่สามารถจับคู่ได้ก็จะหยุดด้วยความไม่สำเร็จและให้คำตอบเป็นเท็จ ดังในรูปที่ 2.1 ซึ่งแสดงขั้นตอนการทำงานพื้นฐานของโปรล็อก ในการทำงานโปรล็อกจะตรวจสอบข้อเท็จจริงทีละข้อจากข้างบนลงมาข้างล่าง และจากซ้ายไปขวา แต่เนื่องด้วยโปรแกรมโปรล็อกจะมีคุณสมบัติหรือวิธีการหาคำตอบที่สำคัญที่เรียกว่า การย้อนกลับ (BACKTRACKING) ซึ่งการย้อนกลับจะเกิดขึ้นในกรณีที่เป้าประสงค์จับคู่กับข้อเท็จจริงหรือกฎมาเรื่อย ๆ จนกระทั่งมาถึงกฎที่จับคู่ไม่ได้ ก็จะทำการย้อนกลับไปที่กฎก่อนหน้านี้ ว่ามีทางอื่นหรือวิธีอื่นหรือไม่ ถ้าไม่มีโปรแกรมก็ย้อนกลับขึ้นไปอีกเรื่อย ๆ จนกระทั่งถึงกฎเริ่มแรก และถ้ายังจับคู่ไม่ได้อีกโปรแกรมก็จะหยุดและให้คำตอบเป็นเท็จ แต่ถ้ากฎก่อนหน้านี้มีทางอื่นก็จะทำไปทางนั้นเรื่อย ๆ จนกระทั่งหมดเป้าหมายที่ให้ไว้ โปรแกรมก็จะหยุดด้วยความสำเร็จหรือเป็นจริง ขั้นตอนการทำงานพื้นฐานของโปรล็อกด้วยการย้อนกลับสามารถแสดงได้โดยรูปที่ 2.2



รูปที่ 2.1 แสดงขั้นตอนการทำงานพื้นฐานของโปรล็อก



รูปที่ 2.2 ขั้นตอนการทำงานพื้นฐานของโปรล็อกด้วยการย้อนกลับ

ระบบผู้เชี่ยวชาญ (EXPERT SYSTEM)

ระบบผู้เชี่ยชาญเป็นศาสตร์แขนงหนึ่งในสาขาปัญญาประดิษฐ์ ซึ่งหมายถึงสิ่งชาญฉลาดที่ถูกประดิษฐ์ขึ้นมา สำหรับโปรแกรมระบบผู้เชี่ยวชาญจะหมายถึงโปรแกรมคอมพิวเตอร์ที่ได้รับการสร้างขึ้น เพื่อเลียนแบบพฤติกรรมในการวินิจฉัยและการตัดสินใจของผู้เชี่ยวชาญในการแก้ปัญหาต่าง ๆ โดยโปรแกรมจะรวบรวมข้อมูลและความรู้จากผู้เชี่ยวชาญที่เป็นมนุษย์ นำมาเป็นข้อมูลในการพิจารณา หรือให้คำปรึกษาแนะนำแก่บุคคลที่ต้องการปรึกษา ด้วยวิธีการถามโต้ตอบระหว่างโปรแกรมหรือเครื่องคอมพิวเตอร์กับผู้ใช้โปรแกรม

โครงสร้างของระบบผู้เชี่ยวชาญนั้นจะมีความแตกต่างกันไม่มากมัก มักจะประกอบด้วยส่วนสำคัญ ๆ อยู่ 5 ส่วน ดังแสดงในรูปที่ 2.3 ซึ่งมีดังนี้

1.ส่วนปฏิภาคกับผู้ใช้โปรแกรม (USER INTERFACE) เป็นส่วนช่วยให้ผู้ใช้สามารถติดต่อกับระบบผู้เชี่ยวชาญได้สะดวกรวดเร็ว และมีประสิทธิภาพมากยิ่งขึ้น ซึ่งโดยปกติแล้วข้อมูลหรือความรู้ต่าง ๆ ที่ใช้ในระบบผู้เชี่ยวชาญมักจะอยู่ในรูปที่ผู้ใช้เข้าใจได้ยาก หรือแปลความหมายยาก ซึ่งอาจจะทำให้เกิดความผิดพลาดในการสื่อสารระหว่างผู้ใช้กับระบบ ดังนั้นจึงมีการสร้างส่วนการปฏิภาคกับผู้ใช้นี้ขึ้นมาช่วย โดยจะทำหน้าที่เป็นตัวกลางในการรับข้อมูลจากผู้ใช้ แล้วแปลงข้อมูลนั้นให้อยู่ในรูปแบบที่ระบบสามารถนำไปใช้ได้ พร้อมทั้งส่งให้กับระบบนำไปเป็นข้อมูลหรือความรู้ในการประมวลต่อไป และเป็นตัวกลางในการรับข้อมูลข่าวสารจากระบบแล้วแปลงให้เป็นรูปแบบที่ผู้ใช้สามารถเข้าใจได้ดี ดังนั้นส่วนนี้จึงต้องประกอบด้วยระบบการประมวลผลภาษาที่ใกล้เคียงกับภาษาธรรมชาติ (NATURAL LANGUAGE) เพื่อให้ผู้ใช้สามารถตอบโต้ได้ใกล้เคียงเหมือนกับการปรึกษากับผู้เชี่ยวชาญที่เป็นมนุษย์มากที่สุด โดยปกติแล้วส่วนปฏิภาคกับผู้ใช้นี้ จะประกอบด้วย 2 ส่วนย่อยคือ ส่วนรับข้อมูล (DATA INPUT) ซึ่งมักจะอยู่ในรูปการถาม-ตอบ หรือเมนูให้เลือก อีกส่วนคือการรายงานผล (REPORTING) ซึ่งอาจจะออกมาในรูปข้อสรุปข้อมูลข่าวสาร กราฟ รูปภาพ หรือเป็นข้อมูลเพื่อที่จะส่งต่อไปยังระบบอื่น ๆ ที่ต่อเนื่องกัน

2. กลไกวินิจฉัย (INFERENCE ENGINE) เป็นส่วนสำคัญที่สุดส่วนหนึ่งของระบบผู้เชี่ยวชาญ บางครั้งอาจเรียกว่าเครื่องอนุมาน ซึ่งจะทำหน้าที่เป็นตัวประมวลผลความรู้จากข้อเท็จจริงและกฎในฐานความรู้ที่มีอยู่มาวินิจฉัยหาข้อสรุป คำตอบ คำให้การปรึกษาตามที่ต้องการในขอบเขตของระบบนั้น ๆ

ระบบผู้เชี่ยวชาญมีกลไกการวินิจฉัยตัดสินใจขั้นพื้นฐานอยู่ 3 แบบ คือ
-กลไกวินิจฉัยแบบย้อนกลับ (BACKWARD-CHAINING METHOD)
-กลไกวินิจฉัยแบบไปข้างหน้า (FOREWARD-CHAINING METHOD)
-กลไกวินิจฉัยแบบกฎ-ค่าความสำคัญ (RULE-VALUE METHOD)

2.1 กลไกวินิจฉัยแบบย้อนกลับ (BACKWARD-CHAINING METHOD) กลไกวินิจฉัยแบบนี้จะเริ่มต้นโดยกำหนดเป้าหมายหรือข้อสรุปสมมุติขึ้นมา แล้วจึงพยายามที่จะค้นหาข้อมูลที่จะสนับสนุนหรือทำการทดสอบในส่วนที่เป็นเงื่อนไขของข้อสรุปที่สมมุตินั้น ๆ หากข้อมูลที่สนับสนุนและเงื่อนไขของข้อสรุปสมมุตินั้นเป็นจริง ข้อสรุปสมมุตินั้นก็จะเป็นคำตอบ และหากว่าข้อมูลที่สนับสนุนหรือเงื่อนไขของข้อสรุปสมมุตินั้นไม่เป็นจริง ก็จะทำการกำหนดเป้าหมายหรือข้อสรุปสมมุติตัวอื่นขึ้นมาใหม่และทำการทดสอบต่อไปจนกว่าจะได้คำตอบที่ต้องการ หรือจนกว่าจะหมดความรู้ในฐานความรู้ ดังนั้นกลไกแบบนี้อาจจะเรียกว่า OBJECT-DRIVEN METHOD หรือ GOAL-DRIVEN METHOD เนื่องจากในการวินิจฉัยจะมุ่งไปที่เป้าหมายก่อน แล้วจึงค่อยหาหลักฐานมาสรุปเป็นคำตอบภายหลัง

2.2 กลไกวินิจฉัยแบบไปข้างหน้า (FOREWARD-CHAINING METHOD) กลไกวินิจฉัยแบบนี้จะตรงกันข้ามกับกลไกวินิจฉัยแบบย้อนกลับ กล่าวคือจะเริ่มต้นด้วยการกำหนดข้อเท็จจริงหรือหลักฐานสนับสนุนขึ้นมาจำนวนหนึ่ง จากหลักฐานที่มีอยู่นี้ระบบจะนำไปใช้ในการค้นหาเป้าหมายหรือข้อสรุป ถ้าหากเป้าหมายหรือข้อสรุปใดที่มีข้อเท็จจริงหรือหลักฐานตรงกับที่สนับสนุน ก็จะถูกนำมาพิจารณาเป็นเป้าหมาย ข้อสรุป หรือเพื่อในไปพิจารณาเป้าหมายหรือข้อสรุปอื่น ๆ ต่อไป กลไกแบบนี้อาจจะเรียกว่า DATA-DRIVEN METHOD เนื่องจากเป็นกลไกที่เริ่มต้นจากข้อมูลที่เป็นข้อเท็จจริงหรือหลักฐานสนับสนุนก่อน แล้วนำข้อมูลนี้ไปหาข้อสรุปที่หลัง

2.3 กลไกวินิจฉัยแบบกฎ-ค่าความสำคัญ (RULE-VALUE METHOD) กลไกวินิจฉัยแบบนี้เป็นการปรับปรุงมาจากกลไกวินิจฉัยแบบย้อนกลับและแบบไปข้างหน้า ซึ่งแบบนี้จะถามคำถามหรือข้อมูลที่มีความสำคัญมากที่สุดก่อน เพื่อที่จะได้เข้าสู่เป้าหมายที่เป็นคำตอบที่ต้องการเสียตั้งแต่แรกเริ่ม แล้วจึงค่อยถามที่มีความสำคัญรองลงมาเรื่อย ๆ ดังนั้นแบบนี้จะให้ประสิทธิภาพสูงกว่าและควรที่ให้คำตอบได้เร็วกว่า 2 แบบแรก แต่จะมีความยากในการสร้างด้วยเหตุผล 2 ประการ คือ ในทางปฏิบัติข้อมูลในฐานความรู้จะมีมากจนกระทั่งไม่สามารถที่จะจัดลำดับความสำคัญได้อย่างเหมาะสม และอีกประการคือจะต้องสร้างฐานความรู้ที่มีการให้น้ำหนักข้อเท็จจริงในแต่ละข้อแตกต่างกันไปตามความสำคัญ ซึ่งทำให้การสร้างฐานความรู้นี้ยากขึ้น

3. ฐานความรู้ (KNOWLEDGE BASE) ฐานความรู้เป็นฐานข้อมูลเกี่ยวกับความรู้ในการแก้ปัญหาหนึ่ง ๆ ที่ต้องการนำมาสร้างเป็นระบบผู้เชี่ยวชาญตามจุดมุ่งหมายที่ตั้งไว้ ซึ่งฐานความรู้นี้จะประกอบได้ด้วยข้อเท็จจริงและกฎต่าง ๆ ซึ่งรวบรวมมาจากการสอบถามผู้เชี่ยวชาญ เอกสารตำรา และอื่น ๆ นำมารวมไว้และจัดให้อยู่ในรูปแบบที่กลไกการวินิจฉัยของระบบจะนำไปใช้ได้ ซึ่งฐานความรู้นี้จะรวมถึงความรู้ที่ได้ในระหว่างการถามโต้ตอบระหว่างระบบกับผู้ใช้ด้วย ฐานความรู้นี้จะประกอบไปด้วยข้อเท็จจริงและกฎต่าง ๆ ซึ่งข้อเท็จจริงเป็นความรู้ระบุถึงข้อมูลความเป็นจริงในปัญหาหนึ่ง ๆ ในรูปของประโยคบอกเล่า สามารถกำหนดค่าความจริงได้อย่างแน่นอน ซึ่งในโปรแกรมจะกล่าวเป็นความสัมพันธ์ระหว่างวัตถุ ดังเช่น เบรคมีเสียง ก็จะเขียนเป็น มี(เบรค,เสียง) เป็นต้น ส่วนกฎเป็นความรู้ที่แสดงความสัมพันธ์ของข้อเท็จจริงภายใต้ความเป็นเงื่อนไขหรือเป็นเหตุเป็นผลต่อกัน ไม่สามารถระบุถึงค่าความจริงได้ในทันทีทันใด จะเป็นจริงหรือเป็นเท็จก็จะขึ้นอยู่กับค่าความจริงของส่วนที่นำมาสัมพันธ์กัน หรือมีเงื่อนไขที่เกี่ยวข้องกัน เช่น ถ้าน้ำมันหมดเครื่องยนต์ก็จะไม่ทำงาน คอมเพรสเซอร์ของระบบแอร์จะทำงานก็ต่อเมื่อพัดลมที่คอลย์เย็นทำงาน เป็นต้น

ตามสถานภาพของฐานความรู้จะสามารถแบบออกได้เป็น 2 ประเภท คือ ฐานความรู้สถิตย์ (STATIC DATABASE) เป็นฐานความรู้ที่บรรจุอยู่ในระบบหรือในโปรแกรมไม่มีการเปลี่ยนแปลงข้อมูลใดในขณะที่เราใช้งานการปรึกษาอยู่ จนกว่าจะมีการแก้ไขโปรแกรม กับฐานความรู้ไดนามิค (DYNAMIC DATABASE OR ACTIVE DATABASE) เป็นฐานความรู้ในรูปของข้อเท็จจริงที่สามารถเปลี่ยนแปลงได้ในขณะที่กำลังใช้งานระบบผู้เชี่ยวชาญ

4. การแสวงหาความรู้ (KNOWLEDGE ACQUISTION) เป็นการค้นหาความรู้ที่จะนำมาใช้ในระบบผู้เชี่ยวชาญ ซึ่งอาจจะได้มาจากตำรา รายงาน ประสบการณ์ หรือจากการได้รับการถ่ายทอดมาจากผู้เชี่ยวชาญในหัวข้อปัญหาด้านวิชานั้น ๆ โดยจะมีวิศวกรความรู้ (KNOWLEDGE ENGINEER) เป็นผู้นำความรู้เหล่านี้มาดำเนินการให้อยู่ในรูปแบบที่เหมาะสมเพื่อใช้กับโปรแกรม

5. สิ่งอำนวยความสะดวกคำอธิบาย (EXPLANATION FACILITY) เป็นส่วนที่ให้คำอธิบายการให้เหตุผล และวิธีการแก้ปัญหาแก่ผู้ใช้ เพื่อให้ผู้ใช้มีความรู้สึกมั่นใจ และเข้าใจขั้นตอนต่าง ๆ ได้ดียิ่งขึ้น โดยปกติแล้วการอธิบายเหตุผลและวิธีการแก้ปัญหาต่าง ๆ ดังกล่าวนั้นจะกำหนดในขอบเขตจำเพาะ และเป็นคำอธิบายสั้น ๆ พอเข้าใจได้ง่าย

กลยุทธ์การแก้ปัญหา (SEARCH STRATEGY)

กลยุทธ์การแก้ปัญหาในระบบผู้เชี่ยวชาญ คือ วิธีการดำเนินการที่จะได้มา ซึ่งคำตอบหรือข้อสรุปตามต้องการ โดยที่มีความนิยมมี 2 วิธี คือ กลยุทธ์การแก้ปัญหาแนวลึก (DEPTH-FIRST SEARCH) และกลยุทธ์การแก้ปัญหาแนวทางกว้าง (BREADTH-FIRST SEARCH) ซึ่งจะได้กล่าวถึงรายละเอียดดังต่อไปนี้

1. กลยุทธ์การแก้ปัญหาแนวทางลึก จะเป็นการดำเนินไปสู่เป้าหมายโดยสำรวจไปในทางเดียวก่อนจนได้คำตอบ หรือหากไม่ได้คำตอบก็จะย้อนขึ้นมาแล้วสำรวจลงไปในทางใหม่ทำเช่นนี้เรื่อย ๆ จนได้คำตอบหรือหมดข้อมูล เพื่อให้เข้าใจการทำงานของกลยุทธ์การแก้ปัญหาแนวทางลึกได้ดียิ่งขึ้น จะแสดงแนวทางการแก้ปัญหาให้เห็น ดังรูปที่ 2.4

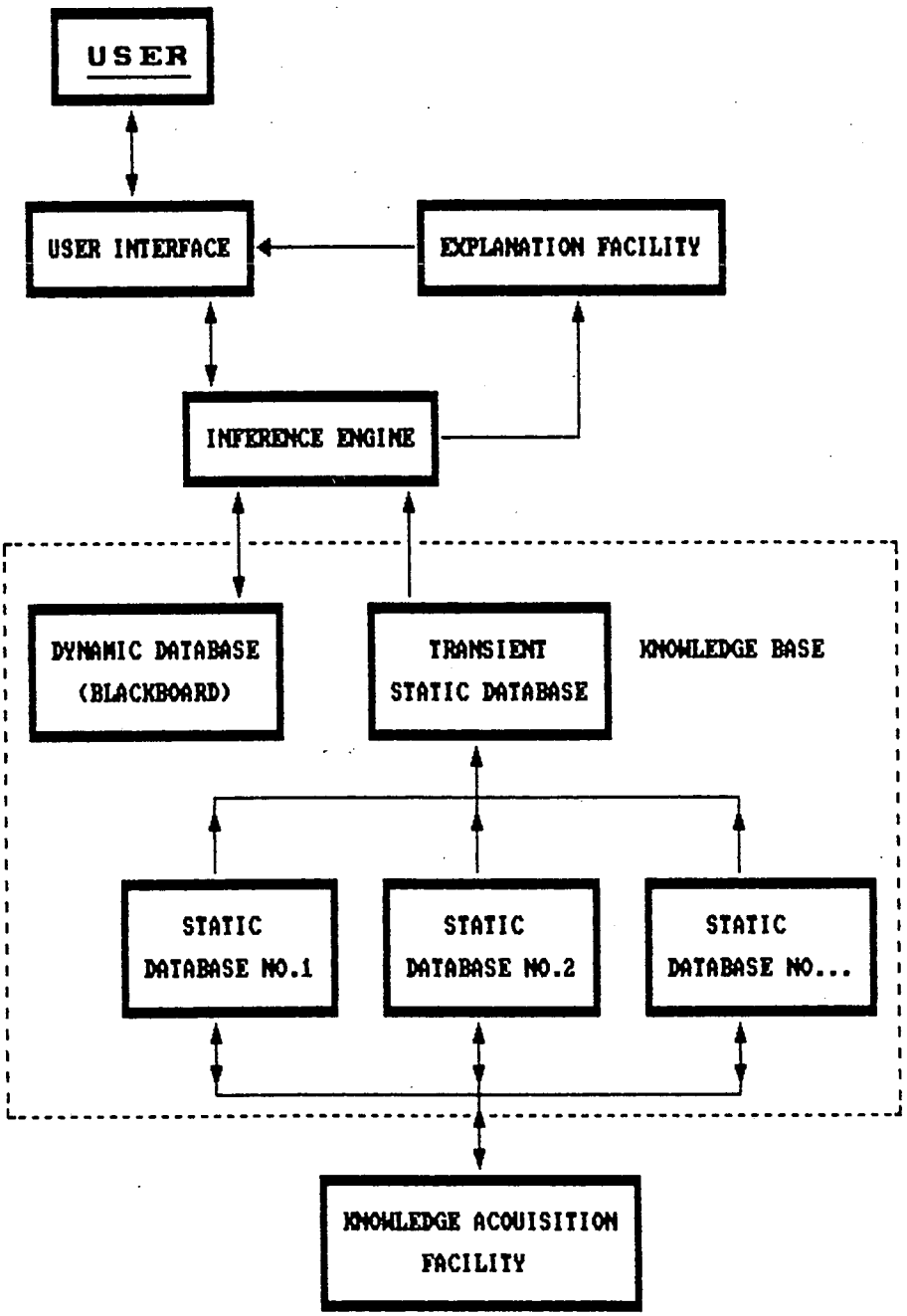
ซึ่งจากรูปที่ 2.4 สมมุติเป้าหมายที่ต้องการคือ F ซึ่งจะเป็นคำตอบของการค้นหานี้ กลยุทธ์การแก้ปัญหาแนวทางลึกจะดำเนินการค้นหาเป้าหมายตามแนวลูกศรดังรูป โดยเริ่มต้นที่โนด (NODE) เริ่มต้น ซึ่งในที่นี้คือโนด A ไปโนดที่ต่ำกว่าโนด A ตัวแรก คือ โนด B มาเป็นเป้าหมาย แล้วตรวจดูว่าเป็นคำตอบหรือไม่ ถ้าโนด B ไม่ใช่คำตอบ ก็จะเลือกโนดที่ต่ำกว่าโนด B ตัวแรก คือ โนด D มาเป็นเป้าหมายแล้วตรวจดูว่าเป็นคำตอบหรือไม่ ถ้าโนด D ไม่ใช่คำตอบ แต่โนด D เป็นโนดปลายสุด ดังนั้นไม่มีโนดที่ต่ำกว่าโนด D อีกแล้วก็จะย้อนกลับขึ้นไปที่โนด B แล้วเลือกโนดที่ต่ำกว่าโนด B ตัวถัดมาทางขวามือ ก็คือโนด E มาเป็นเป้าหมาย แล้วตรวจดูว่าเป็นคำตอบหรือไม่ ถ้าโนด E ไม่ใช่คำตอบ แต่โนด E เป็นโนดสุดท้ายก็จะย้อนขึ้นไปที่โนด B ใหม่ แล้วหาโนดที่ต่ำกว่าโนด B มาเป็นเป้าหมาย แต่โนดที่ต่ำกว่าโนด B หมด ก็จะย้อนกลับขึ้นไปยังโนดที่สูงขึ้นไปอีกหนึ่งขั้น คือโนด A แล้วเลือกโนดที่ต่ำกว่าโนด A ตัวถัดมา คือโนด C มาเป็นเป้าหมาย แล้วตรวจสอบว่าเป็นคำตอบหรือไม่ ถ้าโนด C ไม่มีคำตอบก็จะเลือกโนด F เป็นเป้าหมายและตรวจสอบคำตอบอีก ซึ่งในที่นี้สมมุติให้โนด F เป็นเป้าหมาย ก็จะได้คำตอบออกมา หากสมมุติว่าโนด F ไม่ใช่คำตอบ ก็จะเลือกโนด G มาตรวจสอบอีก และถ้ายังไม่ใช่คำตอบก็จะหยุด และจะไม่มีคำตอบในกรณีนี้ เนื่องจากหมดข้อมูล

2. กลยุทธ์การแก้ปัญหาแนวทางกว้าง จะเป็นวิธีการดำเนินไปสู่เป้าหมายโดยการสำรวจไปในทุกเส้นทางในระดับลึกทีละขั้นพร้อม ๆ กันทุกเส้นทาง จนได้คำตอบหากไม่ได้คำตอบ ก็จะสำรวจลงลึกไปอีกหนึ่งขั้นในแต่ละเส้นทางอีก ทำเช่นนี้ไปเรื่อย ๆ จนได้คำตอบ

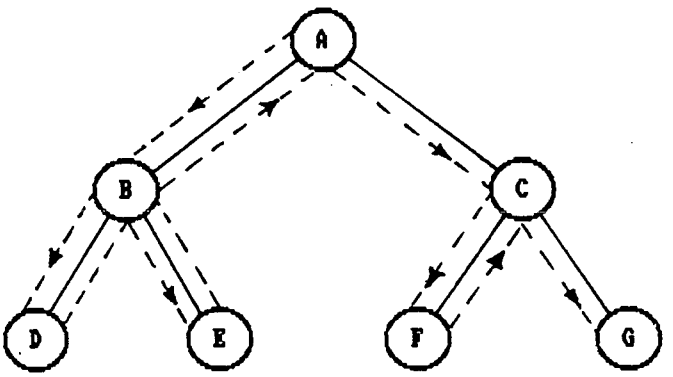
หรือหมดข้อมูล เพื่อให้เข้าใจการทำงานของกลยุทธ์การแก้ปัญหาในแนวทางกว้างให้ดียิ่งขึ้น จะแนวทางแก้ไขปัญหาให้เห็น ดังรูปที่ 2.5

ซึ่งจากรูปที่ 2.5 สมมุติให้เป้าหมายที่ต้องการ คือ F เช่นเดียวกับการแก้ปัญหาในแนวลึกที่ผ่านมา การค้นหาคำตอบ F ของกลยุทธ์การแก้ปัญหาในแนวทางกว้างจะดำเนินการค้นหาเป้าหมายตามแนวลูกศรในรูป โดยมีขั้นตอนดังนี้ คือ เริ่มต้นที่โนด (NODE) เริ่มต้นซึ่งในที่นี้คือโนด A ถ้าโนด A ไม่ใช่คำตอบ และไม่มีโนดอื่นในระดับ A ก็จะไปโนดที่ต่ำกว่าโนด A ตัวแรก คือ โนด B มาเป็นเป้าหมาย แล้วตรวจดูว่าเป็นคำตอบหรือไม่ ถ้าโนด B ไม่ใช่คำตอบ ก็จะเลือกโนดในระดับเดียวกันกับโนด B ซึ่งก็คือ โนด C มาเป็นเป้าหมายแล้วตรวจดูว่าเป็นคำตอบหรือไม่ ถ้าโนด C ไม่ใช่คำตอบ แต่โนด C เป็นโนดสุดท้ายในระดับเดียวกับโนด B ดังนั้นก็จะไปเลือกโนดที่ต่ำกว่าอีกหนึ่งระดับ ซึ่งตัวแรกก็คือโนด D มาเป็นเป้าหมาย แล้วตรวจดูว่าเป็นคำตอบหรือไม่ ถ้าโนด D ไม่ใช่คำตอบ ก็จะเลือกโนดในระดับเดียวกัน คือโนด E มาเป็นเป้าหมาย แล้วตรวจสอบว่าเป็นคำตอบหรือไม่ ถ้าโนด E ไม่มีคำตอบก็จะเลือกโนด F เป็นเป้าหมายและตรวจสอบคำตอบอีก ซึ่งในที่นี้สมมุติให้โนด F เป็นเป้าหมาย ก็จะได้คำตอบออกมา หากสมมุติว่าโนด F ไม่ใช่คำตอบ ก็จะเลือกโนด G มาตรวจสอบอีก และถ้ายังไม่ใช่คำตอบก็จะหยุด และจะไม่มีคำตอบในกรณีนี้ เนื่องจากหมดข้อมูล

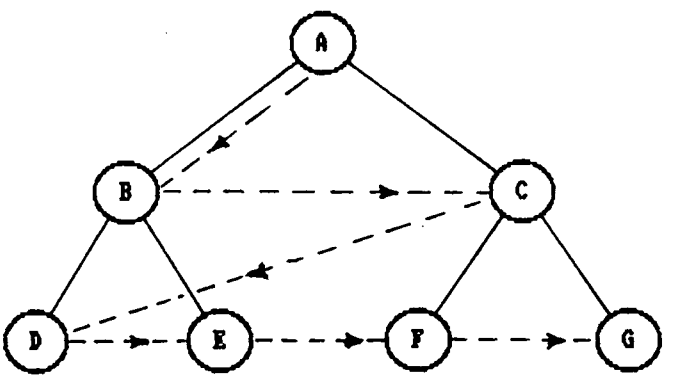
กลยุทธ์การแก้ปัญหาทั้ง 2 วิธีที่กล่าวมา การค้นหามักจะค้นหาโดยไม่มีสิ่งบ่งบอกถึงความแน่นอนในการเลือกเส้นทางไปสู่คำตอบ เป็นการทดลองหาเส้นทางอย่างเดาสุ่มเท่านั้น จึงเป็นกรรมวิธีที่ค่อนข้างจะมืดมน ขาดแนวทางที่จะบีบขอบเขตของการค้นหาให้แคบลง ทำให้เสียเวลามากในการค้นหา โดยเฉพาะถ้าฐานความรู้ที่เป็นขอบเขตของการค้นหานั้นมีขนาดใหญ่มาก จากสาเหตุที่กล่าวมานี้ในบางครั้งจึงมีการใช้กลยุทธ์การแก้ปัญหาจากประสบการณ์เข้ามาช่วยในการค้นหาคำตอบ ซึ่งจะช่วยให้การค้นหาคำตอบเป็นไปได้อย่างรวดเร็วขึ้น แต่อย่างไรก็ตาม การนำกลยุทธ์การแก้ปัญหาจากประสบการณ์มาใช้นั้นจะมีความยุ่งยากมากขึ้นในการสร้างระบบผู้เชี่ยวชาญ และการสื่อความหมายในกระบวนการการค้นหาคำตอบต่อผู้ใช้ไม่ดีเท่าที่ควร การติดตามขั้นตอนการแก้ปัญหายุ่งยาก จึงยังไม่เป็นที่นิยมมากนัก



รูปที่ 2.3 แสดงโครงสร้างของระบบผู้เชี่ยวชาญ



รูปที่ 2.4 แสดงกลยุทธ์การแก้ปัญหาแนวทางลึก



รูปที่ 2.5 แสดงกลยุทธ์การแก้ปัญหาแนวทางกว้าง

เปลือกระบบผู้เชี่ยวชาญ (EXPERT SYSTEM SHELL)

จากที่ผ่านมาจะพบว่า ระบบผู้เชี่ยวชาญจะประกอบด้วยองค์ประกอบพื้นฐานที่สำคัญ 2 ส่วน คือ กลไกวินิจฉัยและฐานความรู้ จึงเกิดมีแนวความคิดในการแยกพัฒนาแต่ละส่วนของระบบผู้เชี่ยวชาญออกจากกัน โดยในส่วนแรกคือ กลไกวินิจฉัยนั้นได้ถูกพัฒนาออกมาในรูปของเปลือกระบบผู้เชี่ยวชาญ โดยได้รวมเอาองค์ประกอบย่อย ๆ ของระบบผู้เชี่ยวชาญมาไว้ด้วยกัน เช่น ส่วนติดต่อกับผู้ใช้ ส่วนอำนวยความสะดวกในการอธิบาย เป็นต้น อีกส่วนหนึ่งคือ ฐานความรู้ ก็จะถูกแยกไปไว้อีกที่หนึ่งต่างหาก โดยสามารถเปลี่ยนแปลง ลด หรือเพิ่มเติมได้อย่างอิสระ โดยมีข้อจำกัดแต่เพียงเรื่องโครงสร้างข้อมูลจะต้องอยู่ในแบบเดิมที่กลไกวินิจฉัยในเปลือกระบบผู้เชี่ยวชาญสามารถนำไปใช้ได้ ดังนั้นจะเห็นได้ว่าเปลือกระบบผู้เชี่ยวชาญหนึ่ง ๆ จะสามารถนำไปใช้กับฐานความรู้ได้หลาย ๆ ด้าน และเป็นการสร้างให้ระบบผู้เชี่ยวชาญใช้ได้กว้างขวางมากยิ่งขึ้น เพียงแต่สร้างฐานความรู้ในด้านที่ต้องการขึ้นมาใช้กับเปลือกระบบผู้เชี่ยวชาญอันเดิมเท่านั้น ก็จะได้ระบบผู้เชี่ยวชาญใหม่ขึ้นมา

ข้อได้เปรียบของเปลือกระบบผู้เชี่ยวชาญ คือ สามารถสร้างระบบผู้เชี่ยวชาญด้านต่าง ๆ ขึ้นมาใหม่ได้ตามความต้องการของผู้ใช้ โดยไม่ต้องสร้างระบบผู้เชี่ยวชาญทั้งระบบขึ้นมาใหม่ คงเพียงแต่สร้างฐานความรู้ขึ้นมาเท่านั้น แต่ระบบนี้จะมีข้อด้อย คือ การปฏิภาคกับผู้ใช้ไม่ดีเท่าที่ควร เพราะระบบนี้ถูกสร้างขึ้นมาใช้กับความรู้หลายด้านไม่จำเพาะเจาะจง และระบบจะใช้ได้กับฐานความรู้ที่สามารถจัดให้อยู่ในรูปโครงสร้าง ตามที่กลไกวินิจฉัยในเปลือกระบบผู้เชี่ยวชาญเข้าใจและนำไปใช้ได้เท่านั้น

ค่าความแน่นอน (CERTAINTY FACTOR)

เราอยู่ในโลกของความไม่แน่นอน ในแต่ละครั้งของเหตุผลและการวินิจฉัยตัดสินใจนั้นต้องเผชิญกับความไม่แน่นอน เช่น การตัดสินใจในการลงทุน การวินิจฉัยโรค การวินิจฉัยข้อขัดข้องของระบบเครื่องกล เป็นต้น แม้ว่าจะมีข้อมูลรายละเอียดที่ได้มาเพื่อการตัดสินใจแต่ละขั้นตอนแล้ว และได้ผลลัพธ์สุดท้าย ก็อาจจะไม่ถูกต้องก็ได้ หรืออาจจะได้ผลลัพธ์หลายค่าก็ได้ ดังนั้น ระบบผู้เชี่ยวชาญที่มีวัตถุประสงค์ลอกเลียนแบบพฤติกรรมของมนุษย์ควรจะมีการเพิ่มค่าความแน่นอนเข้าไปในการตัดสินใจ โดยใช้เหตุผลและความเป็นจริงที่เหมาะสมของข้อมูลที่มี ดังเช่นผู้เชี่ยวชาญที่เป็นมนุษย์

บ่อยครั้งเหตุการณ์ที่เกี่ยวกับความไม่แน่นอน เราไม่สามารถบอกว่าเหตุการณ์ที่ตัดสินใจนั้นเป็นจริงโดยสมบูรณ์ แต่เราจะใช้ค่าความน่าจะเป็น (PROBABILITY) มาช่วย โดยเราจะบอกว่ามีเหตุผลเชื่อว่าเหตุการณ์นั้นมี "ความน่าจะเป็น" จริงเท่าไร ซึ่งเราจะบอกให้อยู่ในรูปของค่าความแน่นอน (CERTAINTY FACTOR, CF) โดยค่าความแน่นอนจะมีได้หลายรูปแบบ เช่น มีค่าเป็นน้ำหนักของความเชื่อถือ 1,2,3,... หรือมีค่าอยู่ระหว่าง 0 ถึง 1 หรือมีค่าเป็นเปอร์เซ็นต์ เป็นต้น ในรูปของค่าความแน่นอนที่เป็นเปอร์เซ็นต์นั้น ถ้าค่าความแน่นอนมีค่าเท่ากับ 100 % ก็แสดงว่าเหตุการณ์จะเกิดขึ้นแน่นอน แล้วถ้ามีค่าความแน่นอนเท่ากับ 0 % แสดงว่าเหตุการณ์นั้นจะไม่เกิดขึ้น ดังนั้นค่าความแน่นอนนี้ก็จะช่วยในการตัดสินใจได้ดีขึ้น

ในการหาค่าความแน่นอนนั้นจะต้องใช้ทฤษฎีของเซตและความน่าจะเป็นมาประกอบด้วย ดังนั้นเราจึงควรที่จะทราบ

<u>กลุ่มผลการทดลอง</u> (SAMPLE SPACE)

<u>นิยาม</u> กลุ่มผลการทดลองของการทดลองหนึ่ง ๆ คือ เซตของผลลัพธ์ที่จะเป็นไปได้ทั้งหมดของการทดลองนั้น ซึ่งเขียนสัญญลักษณ์แทนด้วย "S"

เมื่อเราพิจารณาการทดลองเชิงสุ่มหนึ่ง ๆ ผลลัพธ์ที่จะเป็นไปได้ทั้งหมดของการทดลองเชิงสุ่มนั้น ๆ ก็คือ กลุ่มผลการทดลองของการทดลองนั้น ๆ และแต่ละผลลัพธ์ของการทดลองที่ได้เราเรียกว่า "OUTCOME"

<u>เหตุการณ์</u> (EVENTS)

<u>นิยาม</u> เหตุการณ์ คือ ซับเซท (SUBSET) ของกลุ่มผลการทดลอง มักจะใช้อักษรภาษาอังกฤษตัวใหญ่ จำนวนสมาชิกจะต้องไม่เกินจำนวนสมาชิกที่มีอยู่ในกลุ่มผลการทดลอง ทั้งนี้ขึ้นอยู่กับความสนใจ หรือข้อจำกัดซึ่งเรากำหนดไว้

<u>ความน่าจะเป็น</u> (PROBABILITY)

ความน่าจะเป็นของเหตุการณ์ คือ ตัวเลขที่กำหนดให้แก่แต่ละเหตุการณ์ โดย

แทนตัวเลขนี้ด้วย P_i เรียกตัวเลขนี้ว่า ความน่าจะเป็นที่จะเกิดผลลัพธ์ของเหตุการณ์ (i) ซึ่งมีค่าระหว่าง 0 ถึง 1 การกำหนดความน่าจะเป็นให้แก่แต่ละเหตุการณ์นั้นทำได้ 2 วิธี คือ

ก. วิธีปรนัย (OBJECTIVE VIEW) วิธีวัดแบบนี้จะได้ความน่าจะเป็นอยู่ในรูปเชิงปรนัย ซึ่งมี 2 แบบด้วยกัน คือ การศึกษาตัวแบบทางกายภาพและกลไกของการทดลองเชิงสุ่มนั้น ๆ แล้วจึงกำหนดค่า P_i ให้แก่แต่ละเหตุการณ์ โดยใช้สมมุติฐานบางประการ เช่น การโยนเหรียญ 1 เหรียญ ความน่าจะเป็นที่จะออกหัวเท่ากับ 0.5 ความน่าจะเป็นที่จะออกก้อยเท่ากับ 0.5 เป็นต้น ซึ่งแบบนี้จะเรียกว่า INTRINSIC MODEL APPROACH ส่วนอีกรูปแบบนั้นเป็นการกำหนดค่า P_i โดยอาศัยข้อมูลจากการทดลอง ในกรณีที่เราไม่มีข้อเท็จจริงที่เพียงพอเกี่ยวกับตัวแบบทางกายภาพของการทดลองเชิงสุ่มหนึ่ง ๆ เราไม่สามารถที่จะกำหนดค่า P_i โดยแบบที่แล้วได้ เช่น การหยิบลูกบอลจากกล่องนั้น ถ้าเราไม่ทราบว่าลูกบอลแต่ละสีมีจำนวนเท่าใด เราก็ไม่สามารถกำหนดค่า P_i ให้แต่ละเหตุการณ์ ดังนั้นต้องใช้ข้อมูลจากสถิติของการทดลอง เป็นต้น

ข. วิธีอัตนัย (SUBJECTIVE VIEW) วิธีวัดแบบนี้จะได้ความน่าจะเป็นเชิงจิตวิสัย (SUBJECTIVE PROBABILITY) ยึดเอาตัวบุคคลเป็นหลักและใช้ระดับความเชื่ออย่างมีเหตุผลมาเป็นประโยชน์ในการกำหนดความน่าจะเป็น ซึ่งมีการทดลองเชิงสุ่มบางประเภทเราไม่สามารถที่จะวัดความน่าจะเป็นแบบวิธีปรนัยโดยวิธีใดวิธีหนึ่งได้ เนื่องจากการทดลองเชิงสุ่มนั้น ๆ เราไม่ทราบข้อเท็จจริงเกี่ยวกับตัวแบบหรือกลไกที่ทำให้เกิดการทดลองเชิงสุ่มนั้น และในขณะเดียวกันก็ไม่สามารถที่จะทำการทดลองกับการทดลองเชิงสุ่มเหล่านั้นซ้ำกับหลาย ๆ ครั้งภายใต้สภาวะการณ์เดียวกันได้ เราจึงวัดความน่าจะเป็นแบบวิธีปรนัยไม่ได้ ดังนั้น วิธีที่เราจะวัดความน่าจะเป็นได้ก็คือ ใช้วิธีประมาณค่า P_i โดยการชั่งใจของเราเองโดยวิธีอัตนัย ซึ่งการกำหนดความน่าจะเป็นแบบนี้เป็นการกำหนดเชิงจิตวิสัย คือใช้ความรู้สึกของตนเองเป็นเครื่องกำหนด การกำหนดแบบนี้จะถูกต้องตามความน่าจะเป็นที่แท้จริง หรือใกล้เคียงความน่าจะเป็นที่แท้จริงเท่าใดนั้น ซึ่งไม่สามารถพิสูจน์ได้ในเฉพาะกรณีนี้ แต่จากประสบการณ์ที่ผ่านมา แสดงให้เห็นว่าการใช้ความน่าจะเป็นเชิงจิตวิสัยมาช่วยประกอบในการตัดสินใจ จะได้ผลประโยชน์บ้าง ดีกว่าที่จะไม่ใช้ความน่าจะเป็นเลียเลย เช่น นักศึกษาสามารถที่จะประมาณความน่าจะเป็นที่จะสอบได้เกรดใด ๆ นั้น น่าจะได้จากการประมาณสติปัญญา ความสามารถของตนเอง การมาเรียนโดยสม่ำเสมอหรือไม่ เป็นต้น

การนับมูลฐาน (FUNDAMENTAL COUNTING)

ถ้าในการทดลองใด ๆ สามารถแยกการะทำออกได้เป็น k ขั้นตอน ขั้นตอนแรกมีหนทางที่จะเกิดขึ้นได้ n_1 เป็นหนทางที่แตกต่างกัน ขั้นตอนที่สองมีหนทางที่จะเกิดขึ้นได้ n_2 เป็นหนทางที่แตกต่างกันไป จนถึงขั้นที่ k มีหนทางที่จะเกิดขึ้นได้ n_k เป็นหนทางที่แตกต่างกัน จำนวนหนทางทั้งหมดที่จะเป็นไปได้ จะเท่ากับ $n_1*n_2*...*n_k$ หนทาง ดังนั้นในการถามอาการข้อขัดข้องของสาเหตุต่าง ๆ จะมีคำตอบอยู่ 2 อย่าง คือ ใช้หรือไม่ ถ้ามีอาการในสาเหตุหนึ่ง ๆ จำนวน n อาการ การนับวิธีที่จะเกิดอาการแตกต่างกันไปจะได้จำนวน 2^n วิธี แต่ถ้าไม่นับว่าไม่เกิดอาการทั้งหมดจะได้จำนวน 2^n-1 วิธี หรือจะใช้ทฤษฎีว่าด้วยการจัดหมู่ก็ได้ โดยการคิดว่ามีอาการในสาเหตุหนึ่ง ๆ จำนวน n อาการ จะเกิดขึ้นมา 1 อาการมีกี่วิธี จะเกิดขึ้นมา 2 อาการมีกี่วิธี คิดเรื่อย ๆ มาจนกระทั่งว่าจะเกิด n อาการมีกี่วิธี เอาจำนวนวิธีทั้งหมดมารวมกันก็จะได้จำนวนวิธีทั้งหมด ซึ่งเท่ากับ 2^n-1 (ไม่นับในกรณีที่ไม่มีอาการเกิดขึ้น)