บทที่ 8
ระบบผู้เชี่ยวชาญ
(Expert System)

ในบทนี้จะเป็นการกล่าวถึงระบบผู้เชี่ยวชาญ (Expert System) ในส่วนของพื้นฐาน, ข้อดีและข้อเสีย, ความเหมาะสมในการประยุกต์ใช้และความสัมพันธ์ที่มีต่อโปรแกรมอื่นๆ

8.1 ระบบผู้เชี่ยวชาญคืออะไร?

ก่อนที่เราจะแก้ปัญหาอะไรก็ตาม เราจะต้องรู้ขอบเขตของปัญหาที่เราสนใจ (Domain) เสียก่อนซึ่งสิ่งที่เราพิจารณานั้น มักจะเป็นเรื่องทั่วไป เช่นเดียวกับระบบปัญญาประดิษฐ์ (Artificial Intelligence)อย่างไรก็ตาม เนื่องจากความซับซ้อนที่เกิดขึ้นในตอนแรกๆ ของระบบปัญญาประดิษฐ์ซึ่งยังมีความเชื่อในคำกล่าวแบบเดิม "หากปัญหายังไม่ได้รับการแก้ไข นั่นคือปัญหาของระบบปัญญาประดิษฐ์" นี่เป็นแนวคิดที่ได้รับความนิยมอย่างมากในช่วงปี 1970 เมื่อได้มีขั้นตอนการทำการวิจัยระบบปัญญาประดิษฐ์ทั้งระบบขึ้น อย่างไรก็ดี ณ วันนี้ ก็ยังมีปัญหาที่เกิดขึ้นในโลกแห่งความเป็นจริงมากมายรวมทั้งกระบวนการทางธุรกิจที่ต้องใช้ระบบปัญญาประดิษฐ์ในการแก้ไข

วิธีการแก้ปัญหาของระบบปัญญาประดิษฐ์ในเรื่องทั่วไป อย่างเช่น การแปลภาษาการตีความประโยคและ สิ่งอื่น ๆ ที่ยังไม่เคยแก้ปัญหาสิ่งที่จำเป็นที่สุดในการแก้ปัญหาก็คือขอบเขตของปัญหา ยกตัวอย่างเช่น การสร้างระบบภาษา โดยทั่วไปนั้น มันอาจไม่ใช่เรื่องที่ยาก หากประโยคที่เราป้อนลงไปถูกต้องตามรูปแบบโครงสร้างของประโยค : คำนาม กริยาและกรรม โดยปกติการแก้ปัญหาในลักษณะนี้จะทำงานร่วมกับซอฟต์แวร์ประเภทระบบฐานข้อมูล และตารางคำนวณ

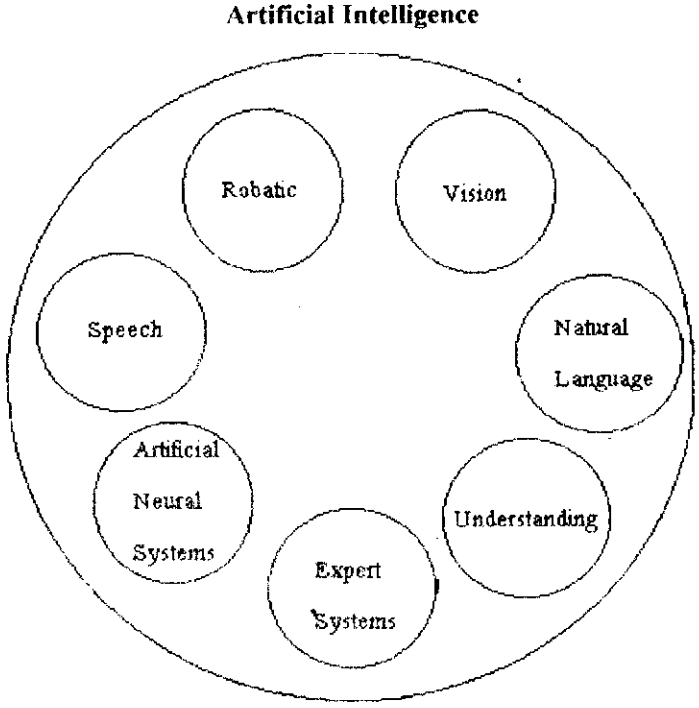
ดังรูป 7.1, ระบบปัญญาประดิษฐ์เกี่ยวข้องกับงานหลายประเภท ระบบผู้เชี่ยวชาญก็เป็นหนึ่งในวิธีแก้ปัญหาที่นำเอาหลักของ AI มาใช้ ซึ่งได้รับความนิยมอย่างมาก Mr.Edward Feigenbaum ผู้เชี่ยวชาญจากมหาวิทยาลัยสแตนด์ฟอร์ด เป็นบุคคลในยุคแรกๆที่ลงมือทำการวิจัยในเรื่องของเทคโนโลยีที่ใช้ในระบบผู้เชี่ยวชาญและได้มีการให้คำจำกัดความของระบบผู้เชี่ยวชาญไว้ว่า "เป็นความชาญฉลาดของโปรแกรมคอมพิวเตอร์ซึ่งต้องใช้ความรู้และกระบวนการวินิจฉัยต่างๆ

CT 488 137

เรื่องของเทคโนโลยีที่ใช้ในระบบผู้เชี่ยวชาญและได้มีการให้คำจำกัดความของระบบผู้เชี่ยวชาญไว้ว่า "เป็นความชาญฉลาดของโปรแกรมคอมพิวเตอร์ซึ่งต้องใช้ความรู้และกระบวนการวินิจฉัยต่างๆ

ในการแก้ปัญหาซึ่งเป็นเรื่องยากพอสมควรในการกำหนดเกณฑ์ความเชี่ยวชาญของมนุษย์ที่ใช้ในการคิดค้นหาทางแก้ไขปัญหา" ซึ่งก็คือ ระบบผู้เชียวชาญนั้น หมายถึง ระบบคอมพิวเตอร์ซึ่งพยายามเลียนแบบความสามารถในการตัดสินใจแบบมนุษย์นั่นเอง

ข้อดีของการแก้ปัญหาด้วยระบบผู้เชียวชาญก็คือก็คือการแก้ปัญหาได้อย่างถูกต้องตรงจุด เพื่อเป็นการยืนยันความสำเร็จของระบบดังกล่าวอาจดูได้จากระบบงานต่าง ๆ เช่น ธุรกิจ การแพทย์ วิทยาศาสตร์ วิศวกรรม ที่ได้นำเอาระบบผู้เชียวชาญไปใช้



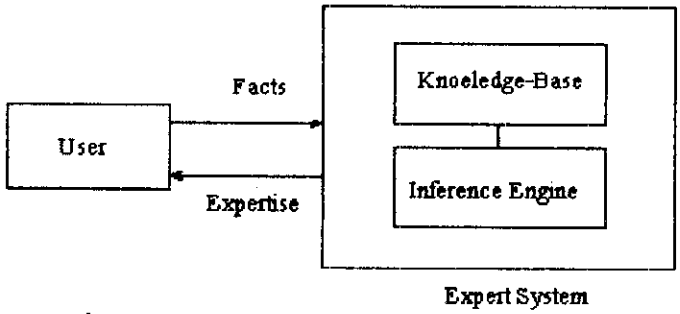
รูปที่ 7.1

ระบบผู้เชี่ยวชาญเป็นแขนงหนึ่งของระบบปัญญาประดิษฐ์ซึ่งมีการใช้อย่างแพร่หลาย ผู้เชี่ยวชาญ คือ บุคคลผู้ซึ่งมี ความรู้ ความชำนาญในเรื่องใดเรื่องหนึ่ง ซึ่งก็หมายความว่า ผู้เชี่ยวชาญเป็นบุคคลที่มีความรู้ ความสามารถ อย่างที่บุคคลทั่วๆไปไม่มี และเนื่องด้วยความเป็นผู้เชี่ยวชาญนี้เอง ทำให้เขาสามารถแก้ไขปัญหาในบางจุดที่คนทั่วไปไม่สามารถแก้ไขได้ ในเรื่องของระบบผู้เชี่ยวชาญนั้นยังรวมไปถึง ความเชี่ยวชาญในเรื่องของระบบภาษา, โปรแกรม และอุปกรณ์ด้วย

ความรู้ในเรื่องระบบผู้เชี่ยวชาญนั้น อาจจะเป็นได้ทั้ง ความชำชองหรือความรู้ที่มี ซึ่งปกติแล้วมักจะเจออยู่ตามหน้าหนังสือ, นิตยสาร หรือ บุคคลที่มีความรู้ ในส่วนของระบบผู้เชี่ยวชาญ

ระบบความรู้พื้นฐาน (knowledge-base system หรือ knowledge-base expert system ซึ่งทั้งสองตัวนี้มีความหมายเหมือนกัน) ผู้คนโดยส่วนใหญ่มักจะใช้ระบบนี้แบบง่ายๆ เพราะว่ามันใช้ระยะเวลาไม่นานในการวิเคราะห์ ถึงแม้ว่าผู้ใช้จะไม่มีความรู้ในเรื่องระบบนี้อย่างลึกซึ้งก็ตามที

รูป 7.2 เป็นรูปภาพแนวคิดพื้นฐานของระบบความรู้พื้นฐาน ผู้ใช้จะมีการส่งข้อมูลเข้าสู่ระบบเพื่อรอผลตอบรับหรือคำแนะนำของระบบผู้เชี่ยวชาญ ในส่วนภายในของระบบนั้น ประกอบไปด้วยสองส่วนประกอบหลักๆ คือ ส่วนบรรจุข้อมูลพื้นฐาน (Knowledge-base container)และส่วนของประมวลผลสรุป (Inference Engine)



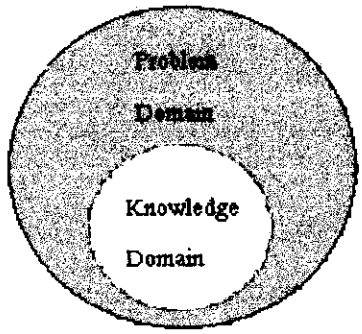
รูปที่ 7.2 Basic Concept of an Expert System Function

ประโยชน์ของระบบความรู้พื้นฐานนั้น ยังถูกออกแบบเพื่อให้เกิดกิจกรรมที่เป็นส่วนช่วยในกระบวนการคิดของมนุษย์ ตัวช่วยนี้ได้ถูกออกแบบมาจากแนวความคิดของระบบผู้เชี่ยวชาญและเช่นเดียวกัน ยิ่งถ้ามีข้อมูลเพิ่มลงไปในส่วนของระบบนี้แล้ว ผลตอบสนองก็จะยิ่งแสดงผลได้อย่างมีประสิทธิภาพมากยิ่งขึ้น การพัฒนาตัวช่วยทางความคิดนี้อาจเป็นประโยชน์ในการกำหนดเกณฑ์ในส่วนการแสดงผลของระบบผู้เชี่ยวชาญ อีกนัยหนึ่ง ยังช่วยให้เกิดความเร็วในการแก้ปัญหาเพื่อเป็นการประหยัดเวลาของระบบผู้เชี่ยวชาญอีกด้วย

ความรู้ของผู้เชี่ยวชาญจะมีขอบเขตปัญหาเพียงอย่างใดอย่างหนึ่ง เกี่ยวกับเทคนิคในการแก้ปัญหา ขอบเขตของปัญหาที่มีลักษณะค่อนข้างพิเศษ เช่น ปัญหาในเรื่อง เภสัชกรรม, การเงิน, วิทยาศาสตร์และวิศวกรรม ซึ่งทั้งสี่ตัวนี้ผู้เชี่ยวชาญเท่านั้นจึงจะสามารถแก้ปัญหาได้อย่างมีประสิทธิภาพ ระบบผู้เชี่ยวชาญ มันก็เหมือนกับ ความเชี่ยวชาญของมนุษย์นั่นแหละครับ โดยปกติแล้วมันถูกออกแบบมาเพื่อขอบเขตปัญหาเพียงแบบเดียว โปรดจำไว้ว่า ระบบความเชี่ยวชาญจะถูกออกแบบมาเพื่อปัญหาใดปัญหาหนึ่งเท่านั้น

ความรู้ของระบบนี้จะใช้ในการแก้ไขปัญหาเพียงอย่างใดอย่างหนึ่งเท่านั้น เราเรียกว่าขอบเขตความรู้ (Knowledge Domain) ยกตัวอย่างเช่น ระบบความชำนาญทางการแพทย์นั้นถูก

ออกแบบมาเพื่อวินิจฉัยโรค เพื่อหาที่มาของอาการป่วยซึ่งก็คือ การติดโรคนั่นเอง ในกรณีนี้ขอบเขตทางความคิดก็คือ ความรู้ทางด้านเภสัชกรรมและความรู้ในเรื่องของ โรค, อาการป่วยและการรักษา จากรูป 7.3 เป็นภาพความสัมพันธ์ระหว่างขอบเขตของปัญหากับขอบเขตของความรู้ เราจะสังเกตว่าขอบเขตทางความรู้ทั้งหมดนั้นจะอยู่ภายใต้ขอบเขตของปัญหา ส่วนที่อยู่ภายนอกขอบเขตทางความรู้นั้นแสดงถึงความรู้ที่ไม่มีความเกี่ยวข้องกับปัญหาเลยแม้แต่น้อย โปรดจำไว้ว่าระบบผู้เชี่ยวชาญนั้นจะปราศจากซึ่งความรู้ในทุกๆ เรื่อง ถ้าเราไม่ได้ทำการติดตั้งโปรแกรมเพื่อสร้างขอบเขตทางความรู้ให้กับระบบ



รูปที่7.3 A possible Problem and Knowledge Domain Relationship

ระบบผู้เชี่ยวชาญได้ให้เหตุผลหรือลงความเห็นไปในทิศทางเดียวกับความชำนาญของมนุษย์ซึ่งก็คือ ทางแก้ของปัญหานั่นเอง นั่นเป็นการแสดงถึง การให้ความจริงบางอย่างและผลสรุปที่ตามมาก็จะแสดงผลออกมาเอง ยกตัวอย่างเช่น ถ้าแฟนของคุณไม่พูดกับคุณเป็นระยะเวลาหนึ่งเดือน คุณอาจจะสรุปได้ว่า แฟนของคุณคงไม่มีเรื่องอะไรที่จะพูด อย่างไรก็ตาม นี่เป็นเพียงหนึ่งในผลสรุปที่เป็นไปได้เท่านั้น

เช่นเดียวกันกับเทคโนโลยีใหม่ๆ เรายังคงต้องเรียนรู้อีกมากเกี่ยวกับระบบผู้เชี่ยวชาญ ในตาราง 7.1 ได้มีการสรุปภาพความแตกต่างของผู้ที่มีส่วนเกี่ยวข้องกับเทคโนโลยี ในตารางนี้ช่างเทคนิคอาจเป็น วิศวกรหรือผู้ออกแบบซอฟต์แวร์ และเทคโนโลยีนั้น อาจเป็นในส่วนของตัวอุปกรณ์หรือระบบที่ใช้ในอุปกรณ์ ในการแก้ปัญหาใดๆก็ตาม ทุกๆคำถามจำเป็นต้องมีคำตอบเฉกเช่นเดียวกับอุปกรณ์อื่นๆ ระบบผู้เชี่ยวชาญอาจจะเหมาะสมหรือไม่เหมาะสมกับการใช้ก็เป็นได้เช่นเดียวกัน ยิ่งเรามีประสบการณ์ในเรื่องของระบบผู้เชี่ยวชาญ ก็จะมองเห็นการนำเอาระบบดังกล่าวมาประยุกต์ใช้ได้อย่างเหมาะสม

Person	*Question*
Manager	**What can I use it for?**
Technologist	**How can I best implement it?**
Researcher	**How can I extend it?**
Consumer	**How will it help me?** **Is it worth the trouble and expense?** **How reliable is it?**

Table 7.1
Differing Views of Technology

8.2 ข้อดีของระบบผู้เชี่ยวชาญ

ระบบผู้เชี่ยวชาญ มีจุดที่น่าสนใจดังต่อไปนี้

- เพิ่มความเหมาะสมในเรื่องประโยชน์ใช้สอย - ความเชี่ยวชาญในส่วนของระบบนั้นต้องเหมาะสมกับตัวอุปกรณ์คอมพิวเตอร์ โปรดสังเกตว่า ในทุกสถานการณ์จริง ระบบผู้เชี่ยวชาญเป็นจะตัวสร้างผลผลิตทางความชำนาญอย่างมากมายเลยทีเดียว
- ลดต้นทุน - มีต้นทุนต่อผู้ใช้ระบบค่อนข้างต่ำ
- ลดความอันตราย - ระบบผู้เชี่ยวชาญสามารถนำไปประยุกต์ใช้ได้ทุกๆสถานการณ์ ซึ่งในบางสถานการณ์นั้น อาจเป็นอันตรายต่อมนุษย์
- มีความเป็นมาตรฐาน – ความชำนาญของระบบคอมพิวเตอร์นั้นเป็นสิ่งที่คงทนไม่เปลี่ยนแปลง ซึ่งจะแตกต่างกับ ความชำนาญของมนุษย์ซึ่งบางทีอาจเกิดการหยุด, ยกเลิกกลางคัน หรือ ความตาย สิ่งเหล่านี้จะทำให้ผลลัพธ์ที่ได้เกิดความไม่แน่นอน
- ความเชี่ยวชาญที่หลากหลาย - ความเชี่ยวชาญที่หลากหลายนั้นอาจถูกนำมาใช้พร้อมๆ กัน หรือนำมาใช้ต่อเนื่องกันได้ ไม่ว่าจะในเวลาใดก็ตาม ระดับความเชี่ยวชาญเกิดมาจากการรวมความชำนาญที่หลากหลายเข้าไว้ด้วยกัน ซึ่งมีมากกว่าความเชี่ยวชาญเพียงอย่างเดียวของมนุษย์
- เพิ่มความน่าเชื่อถือ - ระบบผู้เชี่ยวชาญนั้นจะสามารถเพิ่มความมั่นใจให้กับเราได้ว่า การตัดสินใจที่ถูกต้องนั้นเกิดขึ้นจากการเสนอแนวคิดอย่างอื่น ในกรณีที่เกิดความความขัดแย้งระหว่างความเชี่ยวชาญของมนุษย์ วิธีนี้อาจจะไม่ค่อยดีนักถ้าระบบผู้เชี่ยวชาญถูกกำหนด

ให้มีความชำนาญเพียงอย่างใดอย่างหนึ่ง ระบบผู้เชี่ยวชาญนั้นควรจะมีความเห็นที่สอดคล้องกันกับผู้ใช้ระบบ เนื่องจากในบางกรณีระบบ หรือผู้ใช้ระบบ ก็อาจมีข้อผิดพลาด

- แจกแจงรายละเอียด – ระบบผู้เชี่ยวชาญจะแสดงรายละเอียดของทุกๆสาเหตุซึ่งจะนำไปสู่ผลสรุป บางทีมนุษย์ไม่อยากหรือไม่เต็มใจที่จะทำในส่วนของการแจกแจงรายละเอียดอาจเป็นเพราะว่า เหนื่อยหรือไม่อยากเสียเวลาในส่วนนี้ ซึ่งการแจกแจงรายละเอียดนี้เป็นการสร้างความมั่นใจว่าการตัดสินใจนั้นถูกต้องแล้ว
- มีการตอบสนองอย่างรวดเร็ว – มีการตอบสนองที่รวดเร็วแต่การที่ระบบผู้เชี่ยวชาญสามารถตอบสนองได้อย่างรวดเร็วนั้น ขึ้นอยู่กับตัวซอฟต์แวร์และตัวอุปกรณ์ที่ใช้ด้วยเช่นกัน ซึ่งอาจจะส่งผลให้ตอบสนองได้รวดเร็วขึ้นกว่าความเชี่ยวชาญของมนุษย์เลยทีเดียว
- ความมั่นคงและสามารถตอบสนองได้ตลอดเวลา – นี่เป็นส่วนสำคัญเวลาเจอสถานการณ์อันเร่งด่วนเมื่อยาม ที่ความเชี่ยวชาญของมนุษย์ไม่สามารถจัดการให้เกิดประสิทธิภาพสูงสุดได้
- ฐานข้อมูลที่มีศักยภาพ – ระบบผู้เชี่ยวชาญสามารถเข้าถึงฐานข้อมูลได้อย่างยอดเยี่ยม

8.3 หลักการทั่วๆ ไปของระบบผู้เชี่ยวชาญ

ข้อมูลของระบบผู้เชี่ยวชาญจะแสดงข้อมูลเป็นแบบตัวเลข วิธีส่วนใหญ่ที่ใช้ คือ รูปแบบของ ถ้า...ก็ (IF THEN) ตัวอย่างเช่น IF the light is red THEN stop ถ้าไฟแดงเราก็จะหยุด

ในปัจจุบัน ความรู้พื้นฐานของระบบผู้เชี่ยวชาญได้ถูกขยายขอบเขตทางความรู้ให้มากขึ้น ระบบใหญ่ๆ จะประกอบไปด้วยหลักการต่างๆ มากมาย อย่างเช่น XCON/R1 ซึ่งเป็นระบบของบริษัท Digital Equipment ที่รู้เรื่องราวต่างๆ มากมายมากกว่าคนทั่วๆไป ระบบย่อยๆ สำหรับงานพิเศษโดยเฉพาะก็ถูกสร้างขึ้นด้วยหลักการต่างๆ แม้ว่าระบบย่อยๆ นั้นแม้อาจจะไม่สามารถทำงานในระดับของผู้เชี่ยวชาญได้ แต่มันจะถูกออกแบบให้มีข้อดีในด้านเทคโนโลยีเพื่อให้แสดงในความรู้ที่เจาะลึก สำหรับข้อมูลระบบย่อยๆนั้นเราจะสามารถพบในหนังสือ บทความ หรือ สิ่งตีพิมพ์อื่นๆ

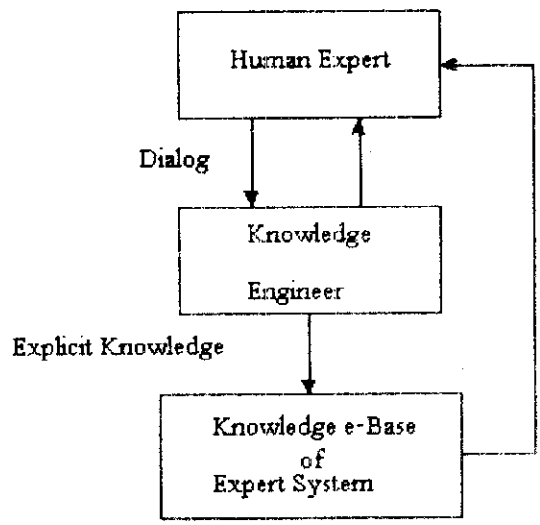
กระบวนการของการค้นคิดระบบผู้เชี่ยวชาญ ได้ถูกเรียกว่า ความรู้ทางด้านวิศวกรรม โดยความรู้ดังกล่าวได้อ้างถึงความรู้ที่ได้มาจากบุคคลที่เชี่ยวชาญ หรือ แหล่งข้อมูลอื่นๆ

หลักที่ได้พัฒนาโดยระบบผู้เชี่ยวชาญ (ซึ่งสามารถดูภาพประกอบที่ 7.4) ความรู้ทางด้านวิศวกรรมเริ่มมาจากการมีการแลกเปลี่ยนความรู้กันของบุคคลขนกลายเป็นความรู้ที่มีความชัดเจน

มากขึ้น หลักการนี้คล้ายกับระบบการออกแบบในโปรแกรมธรรมดาทั่วไป ผู้เชี่ยวชาญได้ประเมิน ระบบผู้เชี่ยวชาญและได้มีการวิเคราะห์ความรู้ กระบวนการนี้ผู้เชี่ยวชาญจะทำการประเมินและตรวจซ้ำๆ จนกระทั่งการแสดงผลของระบบทำได้

ระบบผู้เชี่ยวชาญมีการออกแบบที่แตกต่างจากโปรแกรมอื่นตรงที่ว่า ปัญหาทั่วไปไม่มีการแก้ปัญหาอย่างเป็นขั้นตอน และยังขึ้นอยู่กับความน่าเชื่อถือของผลสรุปที่ได้รับ การแก้ปัญหาที่สมเหตุสมผลนั้นเป็นสิ่งที่เราคาดหวังไว้ และระบบยังต้องสามารถอธิบายอย่างสมเหตุสมผลและสามารถที่จะตรวจสอบได้

ความจริง การอธิบายที่ซับซ้อนอาจจะออกแบบให้ผู้ใช้สามารถเข้าไปสำรวจดูในหลายๆทางของ "What If...." เป็นประเภทของคำถาม เรียกว่า เป็นข้อสมมุติฐานที่สามารถเป็นไปได้



รูปที่ 7.4 Development of an Expert System

ระบบผู้เชี่ยวชาญบางระบบ แม้จะยอมให้เข้าไปเรียนรู้ได้โดยการ ยกตัวอย่าง หลักการหาข้อเท็จจริงซึ่งได้ข้อมูลมาจากจากตารางข้อมูล การที่เราจะนำความรู้ของระบบของผู้เชี่ยวชาญมาสร้างเป็นหลักการนั้นไม่ใช่เรื่องง่าย โดยเฉพาะอย่างยิ่ง ถ้าความรู้นั้นยังไม่มีการจัดเรียงให้เป็นระบบ บางทีอาจจะไม่สอดคล้องกัน มีความไม่ชัดเจน

ความเชี่ยวชาญของมนุษย์ยังได้มีการขยายความรู้และความสามารถ ซึ่งทำให้เราสามารถเข้าถึงปัญหาที่พวกเราอาจละเลยไป ความเชี่ยวชาญของมนุษย์ยังเป็นที่รู้กันว่า "เมื่อมีความรู้ความรู้จะทำให้เกิดกฎใหม่ ๆ" ถ้าระบบผู้เชี่ยวชาญที่ถูกออกแบบมานั้น ไม่มีความชัดเจน แล้ว

ยังนำมาใช้จัดการกับปัญหาที่ไม่มีความแน่นอน ระบบก็จะทำงานช้าถึงแม้ข้อมูลที่นำมาจัดการอาจไม่เหมาะสมหรือไม่สมบูรณ์ก็ตาม

ระบบผู้เชี่ยวชาญได้ให้คำแนะนำ เฉกเช่นเดียวกับความเชี่ยวชาญของมนุษย์ควรที่จะค่อยๆ ลดระดับขอบเขตสิ่งที่ละเลยแทนที่จะลดลงทันที ข้อจำกัดในเรื่องข้อปฏิบัติของระบบผู้เชี่ยวชาญในปัจจุบันนี้ยังขาดสาเหตุหรือที่มาของความรู้ ซึ่งก็คือ ระบบผู้เชี่ยวชาญไม่ได้เข้าใจในเรื่องของสาเหตุและผลลัพธ์ที่เกิดขึ้นอย่างแท้จริงเท่าใดนัก มันอาจจะง่ายกว่าถ้าระบบผู้เชี่ยวชาญใช้เพียงความรู้ในเรื่องนั้นๆ เป็นฐาน ซึ่งอาจได้รับมาจากประสบการณ์และการเรียนรู้มากกว่าการที่ใช้ความรู้ที่ลึกซึ้งเป็นฐานของโครงสร้าง,หน้าที่,และพฤติกรรม ของสิ่งๆหนึ่ง ยกตัวอย่างเช่น มันจะเป็นเรื่องง่ายขึ้นต่อโปรแกรมของระบบผู้เชี่ยวชาญ ในการออกคำสั่ง หรือกำหนด ปริมาณยาแก้ปวดให้แก่บุคคลที่มีอาการปวดหัว มากกว่าทุกๆ โปรแกรม ความรู้ประเภทวิทยาศาสตร์,กายวิภาคศาสตร์ และประสาทวิทยา ที่เกี่ยวกับร่างกายมนุษย์ ระบบข้อมูลเหล่านี้อาจใช้ระบบเวลาตอบสนองค่อนข้างนาน

ความรู้ที่ได้จากการเรียนรู้มักไม่ค่อยประสบความสำเร็จ เมื่อเปรียบเทียบกับวิธีการแก้ปัญหาเป็นขั้นตอน มันจะทำให้ได้ผลลัพธ์ที่แน่นอนกว่า ความรู้ที่ได้มาจากการเรียนรู้ หรือการทดลองนั้นมีลักษณะ เช่นเดียวกันกับ "กฎหัวแม่โป้ง" ซึ่งสิ่งที่รู้มานั้นอาจจะถูกหรือผิดก็ได้ เพราะไม่ได้รับการยืนยัน อย่างไรก็ตาม ในบางสถานการณ์ ความรู้ประเภทนี้ก็เหมาะสมแก่การนำมาใช้แก้ปัญหาเพราะจะช่วยลดต้นทุนในเรื่องของเวลาได้เป็นอย่างดี

อีกปัญหาหนึ่งของระบบผู้เชี่ยวชาญ ณ วันนี้ คือขอบเขตความรู้ยังมีข้อจำกัดรูปแบบของระบบผู้เชี่ยวชาญสามารถนำสิ่งที่เราคิดว่าเหมือนกันหรืออุปมาว่าเหมือนกันนั้น นำมาใช้เป็นเหตุผลได้ถึงแม้ว่าสามารถทำได้ก็ตาม แต่ในความจริงเป็นปัญหาของการเปลี่ยนความรู้ของมนุษย์ให้อยู่ในรูปของระบบผู้เชี่ยวชาญเป็นสิ่งที่จำเป็นหรือที่เราเรียกเสมอว่า ความรู้ผ่านทางคอขวด มันเป็นการบรรยายลักษณะเพราะว่า ความรู้ผ่านทางคอขวดนั้น เป็นการลดโครงสร้างของระบบให้เล็กลงจนสามารถผ่านทะลุคอขวดเข้าไปได้

8.4 ลักษณะของระบบผู้เชี่ยวชาญ

- ระบบผู้เชี่ยวชาญได้ถูกออกแบบมาให้มีลักษณะต่อไปนี้
- มีประสิทธิภาพในการทำงาน – ระบบจะต้องมีประสิทธิภาพในการตอบสนองในสถานการณ์ เฉพาะหน้าได้ดีเทียบเท่าหรือดีกว่า สถานการณ์ปกติ ซึ่งหมายความว่าการให้คำแนะนำโดยระบบนั้นจะต้องมีคุณภาพที่สูงมาก
- ตอบสนองในระยะเวลาที่เหมาะสม – ระบบจะต้องทำงานให้ถูกเวลา เพื่อให้เกิดการตัดสินใจ ดังนั้นการทำงานของระบบมักจะใช้รูปแบบของการลดเวลาการทำงานในกรณีที่ต้องแก้สถานการณ์เฉพาะหน้า โดยระบบจะตอบสนองภายในเวลาที่กำหนด
- มีความน่าเชื่อถือ
- สามารถเข้าใจได้ – ระบบ ควรที่จะอธิบายขั้นตอนสาเหตุ เพื่อให้เกิดความเข้าใจในทางปฏิบัติมากกว่าการให้คำตอบไปเลยหรือที่เรียกกันว่า กล่องดำ (black box) การอธิบายรายละเอียดโดยระบบผู้เชี่ยวชาญนั้น ควรเป็นไปในทิศทางเดียวกับมนุษย์ที่ซึ่งพวกเราสามารถอธิบายถึงสาเหตุได้ สิ่งนี้จึงเป็นสิ่งที่สำคัญมาก

เหตุผลที่ 1 การที่ระบบได้มีการอธิบายรายละเอียดนั้น เพื่อเป็นการสร้างความเข้าใจเพื่อให้มนุษย์ตรวจสอบสาเหตุหรือที่มาได้

เหตุผลที่ 2 การที่ระบบมีความสามารถในการอธิบายรายละเอียดนั้นเป็นการพัฒนาระบบเพื่อเป็นการตรวจสอบและแก้ไขข้อผิดพลาดของความรู้ที่ไม่ถูกต้อง อาจเกิดมาจากการพิมพ์ผิดหรือเกิดมาจากความเข้าใจผิดระหว่างความรู้ของระบบกับความรู้ของผู้ใช้งาน ระบบผู้เชี่ยวชาญควรทำงานในลักษณะของโปรแกรมคู่ขนานที่ซึ่งเป็นกฎความรู้ที่อิสระ มีลักษณะดังนี้

1.มีความยืดหยุ่น - เนื่องจากปริมาณความรู้ของระบบผู้เชี่ยวชาญนั้นมีจำนวนมาก ดังนั้นสิ่งสำคัญในการทำให้เกิดกลไกที่ประสิทธิภาพนั้น อาจทำการเพิ่มเติม, ปรับเปลี่ยน หรือ ลบความรู้ของระบบได้

* แสดงทุกๆ สาเหตุและสมมุติฐานที่ขัดแย้งกัน -สมมุติฐานเป็นสิ่งที่เราต้องทำการพิสูจน์ในสถานการณ์ที่เกิดปัญหาจริงขึ้นนั้น อาจมีหลายหลายสมมุติฐาน เราจำเป็นต้องทำการพิสูจน์ หาความจริงในสถานการณ์ที่ซับซ้อนให้ได้

*แสดงทุกๆสมมุติฐานที่สามารถอธิบายได้จากหลักฐานการสังเกต

*อธิบายทุกๆ ผลลัพธ์ของ ทุกๆสมมุติฐาน

*ให้คำทำนายหรือคาดคะเน สิ่งที่จะเกิดขึ้นถ้าสมมุติฐานเป็นจริง

*เป็นการแสดงว่า คำถามที่ผู้ใช้ได้ถามโปรแกรมไปนั้น เป็นการถามเพื่อเป็นข้อมูลในอนาคต ลักษณะของคำถามมักจะเป็นการถามหาสาเหตุที่มาคล้ายๆ กับการวินิจฉัยโรค ซึ่งมีต้นทุนสูงและใช้เวลานานในทางปฏิบัติ

*เป็นการแสดงถึงความรู้ของโปรแกรม

การแสดงความรู้เกี่ยวกับกฎเกณฑ์สมมุติฐานจะถูกแสดงออกโดยความรู้ และ ความรู้จะถูกแสดงออกโดยหลักฐานซึ่งก็คือเป็นสิ่งที่ถูกต้อง ความรู้สามารถเพิ่มพูนขึ้นได้อย่างง่ายๆ ในระบบของกฎพื้นฐาน ซึ่งก็คือความรู้จะเพิ่มขึ้นทีละเล็กทีละน้อย โดยการเพิ่มกฎเข้าไป ซึ่งทำให้การปฏิบัติและความถูกต้องของระบบสามารถตรวจสอบได้อย่างต่อเนื่อง การเพิ่มระดับความรู้อย่างรวดเร็วนั้นเราเรียกว่า Rapid Prototyping ซึ่งความรู้ของผู้ใช้ สามารถแสดงผลการทำงานของระบบต้นแบบได้อย่างรวดเร็วและระบบนี้ยังแสดงให้เหตุข้อผิดพลาดหรือช่องโหว่ต่างๆ เพื่อทำการแก้ไขให้ถูกต้องโดยทันที

8.5 การพัฒนาระบบผู้เชี่ยวชาญทางด้านเทคโนโลยี

ปัญญาประดิษฐ์มีสาขาที่เกี่ยวข้องมากมายเช่น วิชาเกี่ยวกับการพูด ทัศนคติ เครื่องยนต์กลไก ภาษาซึ่งสามารถเข้าใจและเรียนรู้ได้ รากฐานของระบบผู้เชี่ยวชาญถูกจัดให้เป็นระเบียบโดยเฉพาะ รากฐานที่สำคัญอย่างหนึ่งก็คือสาขาวิชากระบวนการประมวลของมนุษย์ ที่เรียกว่า Cognitive Science กระบวนการการรับรู้ตัวนี้ศึกษาเกี่ยวกับกระบวนการการประมวลผล หรือกระบวนการแก้ปัญหาของมนุษย์

การศึกษาเกี่ยวกับกระบวนการการรับรู้มีความสำคัญมาก คือ ถ้าเราพยายามที่จะทำให้คอมพิวเตอร์เอาอย่างมนุษย์ บ่อยครั้งที่มนุษย์นั้นไม่สามารถที่จะอธิบายการแก้ปัญหาได้ ถ้ามนุษย์ยังไม่สามารถที่จะอธิบายว่าจะแก้ปัญหาอย่างไร นั้นก็เป็นไปไม่ได้ที่จะสามารถเปลี่ยนความรู้ดังกล่าวให้กลายเป็นความรู้พื้นฐานของระบบ ในกรณีนี้ มีความเป็นไปได้อยู่ทางเดียวคือให้โปรแกรมนั้นๆ ให้มันเรียนรู้การเลียนแบบผู้เชี่ยวชาญด้วยตัวของมันเอง โปรแกรมเหล่านี้ขึ้นอยู่กับการหาความจริงด้วยการสังเกตข้อเท็จจริงและการจำลองระบบประสาท

การแก้ปัญหาของมนุษย์ และ การผลิต

การพัฒนาระบบผู้เชี่ยวชาญทางด้านเทคโนโลยี จากตาราง 7.2 ได้มีการสรุปส่วนสำคัญของการพัฒนาให้มีระบบผู้เชี่ยวชาญแบบใหม่ๆ ความเป็นไปได้ คือ วันที่เริ่มต้นโครงการ โครงการมากมายที่แพร่หลายมามากมายหลายปี การพัฒนาเหล่านี้ไปครอบคลุมเนื้อหาในบทนี้ และในบทอื่นๆ หนังสือคู่มือที่ดีที่สุดสำหรับการศึกษาระบบเบื้องต้นคือ หนังสือคู่มือปัญญาประดิษฐ์ (Handbook of Artificial Intelligence) หลังจากปี 1950 และปี 1960 โปรแกรมจำนวนมากได้ถูกเขียนขึ้นโดยมีจุดประสงค์เพื่อใช้ในการแก้ปัญหาทั่วๆ ไปโปรแกรมที่ได้รับความนิยมมากที่สุดคือ General Problem Solver ซึ่ง เนเวลล์ และ ไซมอน เป็นผู้ประดิษฐ์ขึ้น และได้พูดถึงผลงานมากมายของพวกเขาจำนวน 920 หน้าในหนังสือเล่มนี้ เนเวลล์ และไซมอน ได้สาธิตผลลัพธ์ที่สำคัญข้อหนึ่ง ซึ่งเป็นปัญหาที่มนุษย์นั้นจะเจอและต้องแก้ไขบ่อยมาก ก็คือ ถ้า ก็.... (IF THEN.....) เป็นประเภทหลักการการผลิต ยกตัวอย่าง เช่น ถ้าดูเหมือนฝนจะตก ก็ควรที่จะพกร่มไปด้วย หรือ ถ้าสามีหรือภรรยาของคุณอารมณ์ไม่ดี ก็จะทำให้ไม่มีความสุข ถ้ามันเป็นหลักการเล็กๆ แล้วสามารถนำมาประยุกต์ให้เข้ากับข้อมูลหรือความรู้ได้ เราจะเรียกว่า Chunk โดยที่ Chunk จะจัดระเบียบจุดที่บกพร่องแล้วมีการเชื่อมโยงไปยังข้อมูลของ Chunk ที่เกี่ยวข้อง ความจำของคนเราก็เป็นทฤษฎีหนึ่งที่ถูกจัดระบบโดย Chunk ตัวอย่างเช่น

IF the car doesn't run and the fuel gauge reads empty THEN fill the gas tank

ถ้ารถไม่ทำงานและน้ำมันก็หมด ก็เติมน้ำมัน

หลักการของเนเวลล์ และ ไซมอนได้รับความนิยมมากเพราะเราสามารถตรวจสอบได้อย่างสมเหตุสมผลกัน นักจิตวิทยาก็ได้ใช้หลักการนี้เพื่อที่จะสามารถอธิบายการประมวลผลของมนุษย์ ความคิดพื้นฐาน คือ การป้อนความรู้สึกเข้าไปกระตุ้นสมอง เมื่อหลักการที่เหมาะสมได้เข้าไปกระตุ้นส่วนความจำระยะยาวก็จะมีการตอบสนองที่เหมาะสมกลับมา ส่วนความจำระยาวนั้นเป็นส่วนที่เก็บข้อมูล ยกตัวอย่างเช่น

IF there is flame THEN there is a fire. ถ้ามีเปลวไฟก็อาจจะเกิดไฟไหม้

IF there is smoke THEN there may be a fire. ถ้ามีควันก็อาจจะเกิดไฟไหม้

IF there is a siren THEN there may be a fire. ถ้ามีเสียงหวอก็อาจจะเกิดไฟไหม้

หมายเหตุ หลักการสองหลักสุดท้ายอาจจะเป็นการแสดงผลให้เห็นไม่ค่อยชัดเจนนัก บางทีอาจจะไม่เกิดไฟไหม้ก็ได้ อาจจะเป็นแค่ควันในอากาศ ในทำนองเดียวกัน เสียงหวอก็ไม่ได้พิสูจน์ว่าจะเกิดไฟไหม้ อาจจะแค่ดังขึ้นมาเฉยๆ

CT 488

ว่าจะเกิดไฟไหม้ อาจจะแค่ดังขึ้นมาเฉยๆ

ความจำระยะยาวประกอบไปด้วยหลักการง่ายๆมากมาย อย่างโครงสร้าง IF THEN... ในทางกลับกัน ส่วนของความจำระยะสั้น ถูกใช้เก็บข้อมูลแค่ชั่วคราวระหว่างการแก้ปัญหาเท่านั้น แม้ว่าส่วนของความจำระยะยาวสามารถที่จะรองรับข้อมูลเป็นร้อยๆ พันๆ อย่างเช่น คุณลองนึกตัวเลขบางตัวไว้ในใจ คนส่วนมากมองเห็นตัวเลขได้มากถึง 4 -7 ตัวภายในครั้งเดียว แน่นอนเราอาจสามารถที่จะจำตัวเลขได้มากกว่า 4 -7 ตัว และตัวเลขเหล่านั้นก็จะถูกเก็บในส่วนของความจำระยะยาว

ทฤษฎีหนึ่งที่เสนอส่วนของความจำระยะสั้นเช่นเรื่องของตัวเลข ที่มีส่วนกระตุ้นให้เกิดการกระตือรือร้นและคำนึงถึง การแก้ปัญหาของมนุษย์ การขยายตัวของชั้งค์ ในความคิด เช่น การที่เรากำลังคิดหรือมีสติกับสิ่งใดสิ่งหนึ่งนั้น เราอาจจะพูดกับตัวเอง ด้วยเสียง "อืม....." เพื่อเป็นการกระตุ้นความคิด

ส่วนประกอบอื่นๆที่จำเป็นสำหรับการแก้ปัญหา คือ ตัวประมวลผลการรับรู้ (Cognitive Processor) Cognitive Processor พยายามที่จะคิดค้นหรือหาหลักการที่จะมาเป็นตัวกระตุ้น หลักการเดียวเท่านั้นที่เหมาะกับการทำงานให้เกิดผล cognitive processor ตัวนี้จะเป็นตัวเลือกหลักการเพียงหนึ่งซึ่งหลักการที่ดีที่สุดจากหลายๆ หลักการ แล้วเลือกไปปฏิบัติ อย่างตัวอย่างทั้งสองประโยคนี้ก็อาจเป็นผลที่ถูกแสดงออก

IF there is a fire THEN leave ถ้าไฟไหม้ก็ให้ออกจากพื้นที่ตรงนั้น

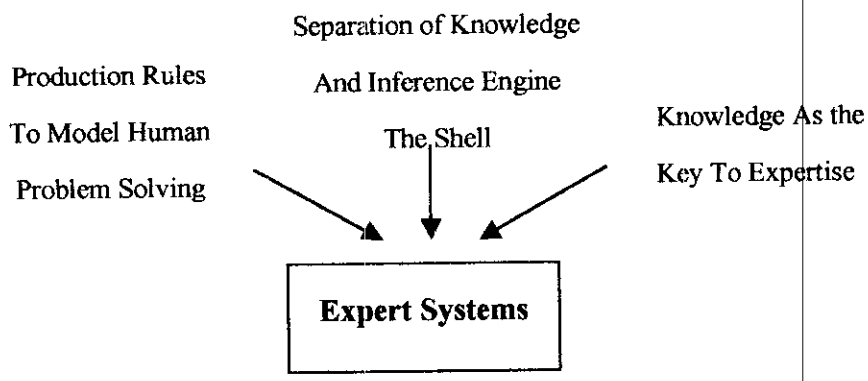
IF my clothes are burning THEN put out the fire ถ้าเสื้อกำลังถูกเผาก็ให้ดับไฟ

จากนั้นหลักการอันนั้นก็จะถูกเลือกออกมาปฏิบัติงานก่อนหลักการอื่นๆ Inference engine ของระบบผู้เชี่ยวชาญแบบใหม่ก็จะตรงกันกับ Cognitive Processor

เนเวลล์ และ ไซมอนได้สร้างแบบการแก้ปัญหาในส่วนความจำระยะยาว,ส่วนของความจำระยะสั้น และ Cognitive Processor ซึ่งเป็นระบบผู้เชี่ยวชาญขั้นพื้นฐานแบบใหม่

หลักการเหล่านี้ก็คล้ายกับระบบการผลิตประเภทหนึ่ง หลักการระบบผลิตขั้นพื้นฐานเป็นที่วิธีที่คนส่วนใหญ่นิยมใช้ การออกแบบระบบผู้เชี่ยวชาญนั้นก็จะต้องมีข้อมูลที่สำคัญจำนวนหนึ่งหรือหลักการย่อยหลายหลักการเป็นองค์ประกอบ การที่หลักการย่อยๆ หรือเล็กๆ มากเกินไปแล้วปราศจากการอ้างอิงหลักการอื่นๆ ก็จะทำให้เข้าใจยาก หรือถ้ามีการอ้างอิงถึงกันมากเกินไปก็ทำให้การทำงานนั้นยากต่อการดัดแปลงแก้ไขเพราะจะทำให้เกิดผลกระทบกัน

จนกระทั่งกลางยุคปี 60 การค้นคว้าเกี่ยวกับเรื่องปัญญาประดิษฐ์ เป็นการผลิตระบบความชาญฉลาด (Intelligent Systems) แม้ว่าGeneral Problem Solver จะมีการอธิบายโดยการยกตัวอย่าง



รูปที่ 7.5 Convergence of important factors to create the model Rule-based Expert System

ปี 1980 บริษัทใหม่เริ่มต้นที่จะนำเอาระบบผู้เชี่ยวชาญของห้องวิจัยในมหาวิทยาลัยและการผลิตผลิตภัณฑ์ต่างๆ ซอฟต์แวร์ที่มีประสิทธิภาพสูงสำหรับพัฒนาระบบผู้เชี่ยวชาญ อย่างเช่น Automated Reasoning Tool (ART) บริษัท Inference Knowledge Engineer Tool (KEE) โดยบริษัท Intelli และ Rule master โดยบริษัท Radian ฮาร์ดแวร์ตัวใหม่ถูกพัฒนาเพื่อซอฟต์แวร์ที่มีความเร็วกว่าเดิม เช่นบริษัท Symbolic และ LMI แนะนำคอมพิวเตอร์ อ้างถึง ระบบซ้อน LISP เพราะพวกเขาได้ออกแบบ สำหรับ LISP รากฐานภาษาของระบบพัฒนาซอฟต์แวร์ผู้เชี่ยวชาญ ในระบบซับซ้อน LISP เป็นภาษาท้องถิ่น ระบบการทำงาน และรหัสพื้นฐานทั้งหมดจะถูกนำไปปฏิบัติ LISP

CLIPS

โชคไม่ดีที่ เทคโนโลยีสูงๆ มีราคาแพงมาก ขณะที่ระบบ LISP จะมีประสิทธิภาพและให้ผลดีกว่าระบบ และ ซอฟต์แวร์ ราคา 100,000 ดอลลาร์ การสร้างปัญญาประดิษฐ์ในห้องทดลองกับอีก 6 โปรแกรม ราคา 500,000 ดอลลาร์ การคัดค้านในกลางปี 80 โดย การพัฒนาซอฟต์แวร์ให้มีประสิทธิภาพ เช่น CLIPS โดย NASA CLIPS : C คือ ความเร็วและสิ่งที่เคลื่อนย้ายได้และยังสามารถใช้งานได้อย่างมีประสิทธิภาพซึ่งเรียกว่า Rete Algorithm C ตัวอื่นๆเป็นตัวรับรองมาตรฐาน Kernigan และ Richie ภาษา C สามารถใช้ในการติดตั้ง CLIPS CLIPS ที่ถูกติดตั้งบน IBM PC และ VAX , Hewlett Packard , Sun Cray และคอมพิวเตอร์แบบอื่นๆ อีกมากมาย เรื่องของ Macintosh ยังสนับสนุน Macintosh เชื่อมโยงกับหน้าต่างและเมาส์ ความเร็วของ CLIPS บนไมโครคอมพิวเตอร์ 80386 หรือ 68020 ซึ่งติดต่อหับฐานของ LISP และระบบ LISP

CT 488 151

8.6 การประยุกต์ระบบผู้เชี่ยวชาญและโดเมน

โปรแกรมคอมพิวเตอร์ธรรมดา ได้ถูกใช้แก้ปัญหาหลายประเภทโดยทั่วไป ปัญหาเหล่านั้นจะมีการแก้ปัญหาที่เป็นขั้นตอน เช่น FORTRAN, Pascal, Ada สาขา ธุรกิจ และ วิศวกรรม , การคำนวณเลข เป็นส่วนสำคัญ ในทางกลับกัน ระบบผู้เชี่ยวชาญเป็นการออกแบบที่สำคัญสำหรับ symbolic reasoning

LISP และ PROLOG ได้ถูกใช้ ในการจัดการsymbolic ซึ่งจะมีเป้าหมายทั่วไปมากกว่าระบบผู้เชี่ยวชาญเชลล์ นั่นก็ไม่ได้หมายความว่ามันจะเป็นไปไม่ได้ที่จะสร้างระบบผู้เชี่ยวชาญในระบบ LISP และ PROLOG ในความเป็นจริงแล้วระบบผู้เชี่ยวชาญมากมายนั้นได้สร้างขึ้นกับ LISP และ PROLOG

PROLOG มีข้อดีอยู่มากสำหรับระบบการวินิจฉัย มันมีความสะดวกมากกว่าและมีประสิทธิภาพที่จะสร้างระบบผู้เชี่ยวชาญขนาดใหญ่กับเชลล์และโปรแกรมที่มีประโยชน์ใช้สอยที่มีการออกแบบโดยเฉพาะสำหรับระบบผู้เชี่ยวชาญการสร้างแทนที่การ กลับมาประดิษฐ์อีกครั้ง (re-inventing the wheel) ทุกครั้งที่มีระบบผู้เชี่ยวชาญใหม่ๆ มา มันจะมีประสิทธิภาพกับการใช้เครื่องมือที่ออกแบบมาโดยเฉพาะ สำหรับ ระบบผู้เชี่ยวชาญการสร้างซึ่งจะมีประสิทธิภาพ มากกว่าการใช้อุปกรณ์ทั่วๆ ไป

การประยุกต์ระบบผู้เชี่ยวชาญ

ข้อมูลเกือบทั้งหมดของระบบผู้เชี่ยวชาญจะถูกนำมาประยุกต์ มีบางระบบที่นำมาออกแบบเพื่องานวิจัย ขณะที่ระบบอื่นๆถูกนำไปใช้ทางด้านธุรกิจและการทำงานในโรงงานอุตสาหกรรม มีระบบ XCON ของบริษัท ดิจิตอล อีควิปเมนต์ ที่นำมาใช้ประจำในทางธุรกิจ ระบบนี้มีชื่อเก่าว่า R1 เคยได้รับการพัฒนา โดย จอห์น แมคเดอร์มอท จาก มหาวิทยาลัย คาร์เนจี – เมลลอน ระบบนี้เป็นโครงสร้างของระบบผู้เชี่ยวชาญ สำหรับระบบคอมพิวเตอร์ของบริษัท ดิจิตอล อีควิปเมนต์

โครงสร้างของระบบคอมพิวเตอร์ หมายถึง เวลาที่มีลูกค้าจะสินค้า ส่วนทั้งหมดฝั่งขวาคือ ซอฟต์แวร์,ฮาร์ดแวร์ และ เอกสารประกอบก็คือ การจัดส่งหรือเสบียง สำหรับระบบใหญ่ๆ ระบบของลูกค้าคือโครงสร้างที่โรงงานและมีรับรองการทดสอบที่เป็นการความต้องการของลูกค้าไม่เหมือนการซื้อโทรทัศน์หรือคอมพิวเตอร์ที่ใช้ในบ้าน มีทางเลือกมากมายและการติดต่อกับ

คอมพิวเตอร์ระบบใหญ่ๆ การป้อนข้อมูลเข้าระบบใหญ่ มันไม่ใช่แค่ส่งตามจำนวนของไปตามจำนวน แต่เราต้องมีการตรวจสอบสภาพการทำงานว่าของชิ้นนั้นไม่มีปัญหา

ระบบ XCON เป็นอีกระบบหนึ่งที่มีประสิทธิภาพมากที่สุด สำหรับการใช้งานในชีวิตประจำวันและ ช่วยลดค่าใช้จ่ายให้กับบริษัท ดิจิตอล อีควิปเมนท์ เป็นหลายล้านดอลลาร์ต่อปี ช่วยประหยัดเวลาการสั่ง และสามารถที่จะแก้ไขคำสั่งให้ถูกต้อง XCON สมารถสามารถที่จะทำค่าเฉลี่ยของคำสั่งภายสองนาที ซึ่งเร็วกว่ามนุษย์แน่นอน มนุษย์สามารถที่จะแก้ไขคำสั่งให้ถูกต้องได้แค่ 70 เปอร์เซ็นต์ต่อครั้ง แต่ สำหรับ XCON นั้นสามารถทำได้ถึง 98 เปอร์เซ็นต์ และนี่ก็เป็นจุดสำคัญในการเอาระบบของคอมพิวเตอร์เข้ามาช่วยในการทำงาน

ระบบผู้เชี่ยวชาญเป็นร้อยๆ ระบบได้ถูกสร้างขึ้นและตีพิมพ์ในวารสารคอมพิวเตอร์ หนังสือ และ ในการประชุม พื้นฐานข้อมูลบรรยายเกี่ยวกับสิ่งตีพิมพ์ การประยุกต์ข้อมูลที่มีหลายประเภท สามารถสังเกต และ แสดงตามตาราง 7.3

Class	General Area
Configuration	Assemble proper components of a system in the proper way.
Diagnosis	Infer underlying problems based on observed evidence.
Instruction	Intelligent teaching so that a student can ask *Why, How* and *What if* type questions just as if a human was teaching.
Interpretation	Explain observed data.
Monitoring	Compares observed data to expected data to judge performance.
Planning	Devise actions to yield a desired outcome.
Prognosis	Predict the outcome of a given situation.
Remedy	Prescribe treatment for a problem.
Control	Regulate a process. May require interpretation, diagnosis, monitoring, planning, prognosis, and remedies

ตาราง 7.3 Broad Classes of Expert Systems

CT 488 153

ตัวอย่างของระบบผู้เชี่ยวชาญถูกแสดงผ่านทางตาราง 7.4 จนถึง 7.9 และส่วนอ้างอิงท้ายบท

Name	Chemistry
CRYSALIS	Interpret a protein's 3-D structure
DENDRAL	Interpret molecular structure
TQMSTUNE	Remedy Triple Quadruple Mass Spectrometer (keep it tuned)
CLONER	Design new biological molecules
MOLGEN	Design gene-cloning experiments
SECS	Design complex organic molecules
SPEX	Plan molecular biology experiments

ตาราง 7.4 Chemistry Expert System

Name	Electronics
ACE	Diagnosis telephone network faults
IN-ATE	Diagnosis oscilloscope faults
NDS	Dagnose national communication net
EURISKO	Design 3-D microelectronics
PALLADIO	Design and test new VLSI circuits
REDESIGN	Redesign digital circuits to new
CADHELP	Instruct for computer aided design
SOPHIE	Instruct circuit fault diagnosis

ตาราง 7.5 Electronics Expert Systems

Name	Medicine
PUFF	Diagnosis lung disease
VM	Monitors intensive-care patients
ABEL	Diagnosis acid-base/electrolytes
AI/COAG	Diagnosis blood disease
AI/RHEUM	Diagnosis rheumatoid disease
CADUCEUS	Diagnosis internal medicine disease
ANNA	Monitor digitalis therapy
BLUEBOX	Diagnosis / remedy depression
MYCIN	Diagnosis / remedy bacterial infections
ONCOCIN	Remedy / manage chemotherapy patients
ATTENDING	Instruct in anesthetic management
GUIDON	Instruct in bacterial infections

ตาราง 7.6 Medical Expert Systems

Name	Engineering
REACTOR	Diagnosis / remedy reactor accidents
DELTA	Diagnosis / remedy GE locomotives
STEAMER	Instruct operation – steam powerplant

ตาราง 7.7 Engineering Expert Systems

Name	Geology
DIPMETER	Interpret dipmeter logs
LITHO	Interpret oil well log data
MUD	Diagnosis / remedy drilling problems
PROSPECTOR	Interpret geologic data for mineral

ตาราง 7.8 Geology Expert Systems

CT 488

Name	Computer Systems
PTRANS	Prognosis for managing DEC computers
BDS	Diagnosis bad parts in switching net
XCON	Configure DEC computer system
XSEL	Configure DEC computer sales order
XSITE	Configure customer site for DEC computers
YES/MVS	Monitor / control IBM MVS operating system
TIMM	Diagnosis DEC computers

ตาราง 7.9 Computer Expert Systems

ตัวแปรที่เหมาะสมสำหรับระบบผู้เชี่ยวชาญ

ความสำคัญต่อการตัดสินใจก่อนที่จะเริ่มสร้างระบบผู้เชี่ยวชาญ คือ ถ้าระบบผู้เชี่ยวชาญเป็นตัวอย่างที่เหมาะสม สิ่งที่น่าสนใจอย่างหนึ่งคือ ไม่ว่าจะเป็นระบบผู้เชี่ยวชาญที่สามารถใช้เป็นตัวอย่างทางเลือก อย่างเช่น โปรแกรมทั่วๆ ไป ตัวแปรที่สำคัญสำหรับระบบผู้เชี่ยวชาญก็ขึ้นอยู่กับกรณีของตัวเลข (Number of factor)

โปรแกรมทั่วๆ ไปสามารถที่จะแก้ปัญหาได้อย่างมีประสิทธิภาพได้หรือไม่ ?

ถ้าคำตอบว่า ได้ ดังนั้นระบบผู้เชี่ยวชาญก็ไม่ใช่ทางเลือกที่ดีที่สุดแล้ว อย่างเช่น การพิจารณาวินิจฉัยปัญหา ถ้าอาการทั้งหมดไม่สามารถปฏิบัติงานได้อย่างปกติ จึงต้องตัดสินใจจุดบกพร่อง ระบบผู้เชี่ยวชาญเป็นระบบที่เหมาะสมมากที่สุดสำหรับสถานการณ์ที่ไม่มีทางแก้ปัญหาอย่างมีประสิทธิภาพ กรณีนี้เรียกว่า โครงสร้างของปัญหาที่ยุ่งยาก (Ill – structured problems) และได้เสนอให้เป็นการแก้ปัญหาที่ดีวิธีหนึ่ง

อย่างเช่นในกรณีของคนคนหนึ่งที่คิดจะท่องเที่ยว นอกจากคนส่วนใหญ่จะมีการวางแผนการเดินทางแล้ว จากตารางรายการที่ 7.10 ก็เป็นโครงสร้างของปัญหาที่ยุ่งยาก เป็นการถามคำถามของผู้นำเที่ยว

คำถามของผู้นำเที่ยว	คำตอบ
มีอะไรให้ช่วยไหม ? คุณต้องการไปที่ไหน ? มีสถานที่อยากไปเป็นพิเศษรึเปล่า ? คุณมีเงินมากเท่าไหร่ ? คุณพอที่จะหาเงินได้บ้างไหม? คุณต้องการที่จะไปเมื่อไหร่ ?	ฉันกำลังคิดว่าอยากจะไปที่ไหนสักแห่ง ฉันก็ไม่รู้ว่าจะไปที่ไหนดี ฉันแค่อยากไปเที่ยวเท่านั้น ไม่มีสถานที่ที่อยากจะไปเป็นพิเศษ ฉันมีเงินไม่พอที่จะไปเที่ยว ฉันไม่รู้ว่าจะหาเงินได้อย่างไร ฉันต้องการไปเร็วนี้

ตาราง 7.10

ตัวอย่างของ โครงสร้างปัญหาที่ยุ่งยาก

กรณีนี้มันเป็นการยกตัวอย่างแบบง่ายๆ อย่างที่คุณเห็น โครงสร้างของปัญหาที่ยุ่งยากไม่ยอมให้มีการแก้ปัญหาอย่างเป็นขั้นตอน เพราะมีความเป็นไปได้หลายทาง

ในภาวะที่ต้องประสพปัญหาเกี่ยวกับตัวโครงสร้างซึ่งเป็นอันตรายต่อระบบผู้เชี่ยวชาญนั้น มีผลสะท้อนถึงการคิดค้นขั้นตอนในการแก้ปัญหา ซึ่งก็คือ เป็นการพัฒนาของระบบผู้เชี่ยวชาญในการค้นพบขั้นตอนในการแก้ปัญหาที่ไม่เคยพบเห็นหรือสิ่งที่ไม่เคยรู้มาก่อน ซึ่งเป็นกฎข้อบังคับตายตัวในการปฏิบัติให้เกิดผลลัพธ์ที่แน่นอน ในส่วนของกฎข้อบังคับในเรื่องของส่วนควบคุมโครงสร้างของระบบผู้เชี่ยวชาญ จะไม่สนใจข้อดีในส่วนของระบบผู้เชี่ยวชาญซึ่งเกี่ยวเนื่องกับส่วนข้อมูลที่คาดไม่ถึงที่ซึ่งไม่เป็นไปตามรูปที่กำหนดไว้ (Parrello 88) ซึ่งก็คือระบบผู้เชี่ยวชาญนี้มีโอกาสที่จะมีปฏิกิริยากับข้อมูล แต่ไม่ว่าจะอะไรก็ตาม ระบบผู้เชี่ยวชาญกับระบบควบคุมทั้งหลายนั้น ส่วนใหญ่เป็นตัวชี้นำในเรื่องของขั้นตอนที่แตกต่างกันและอาจจะเป็นตัวเลือกที่ดีสำหรับโปรแกรมที่ใช้สอยทั่วๆ ไป

*ขอบเขตของตัวแปรหลักที่เหมาะสม ? มันมีความสำคัญต่อการจำกัดความหมายของระบบผู้เชี่ยวชาญที่ถูกคาดหวังให้มีประสิทธิภาพ สมมติถ้าคุณต้องการที่สร้างระบบผู้เชี่ยวชาญขึ้นมาสักระบบหนึ่งให้วินิจฉัยอาการปวดศีรษะ ข้อมูลทางการแพทย์ก็จะเป็นข้อมูลพื้นฐาน อย่างไรก็ตาม การมีความเข้าใจของอาการปวดศีรษะอย่างละเอียดนั้น คุณก็ควรที่จะมีข้อมูลทางวิทยาศาสตร์ทางเคมี แล้วก็ วิชาเคมี แล้วก็ โมเลกุลที่มีการศึกษาจากกระบวนการทางชีววิทยาโดยอาศัยหลักการและวิธีทางฟิสิกส์ และอันที่สี่ก็คือฟิสิกส์ทางนิวเคลียร์ ตัวแปรอื่นๆก็ อย่างเช่น วิธี

การเรียนรู้การควบคุมการทำงานของจิตและกาย, จิตวิทยา, จิตเวชศาสตร์, การออกกำลังกาย, โยคะ, การจัดการกับความเครียด ก็อาจเป็นข้อมูลที่ตรงกับอาการปวดศีรษะ คุณจะหยุดการเพิ่มข้อมูลเมื่อไหร่ ? เมื่อข้อมูล หรือ ตัวแปร เพิ่มมากขึ้น ก็จะเกิดเป็นความซับซ้อนของระบบผู้เชี่ยวชาญที่มากขึ้น

งานของความชำนาญในระดับเดียวทั้งหมด ก็กลายเป็นงานชิ้นใหญ่ จากประสบการณ์การทำงานของระบบแบบเป็นทีมนั้นยาก โดยเฉพาะเมื่อเวลาที่มีการเสนอหัวข้อที่ขัดแย้งกัน ถ้าเรารู้ข้อกำหนดในระดับเดียวกัน จากนั้นก็สามารถที่จะจัดการวางแผนระบบผู้เชี่ยวชาญให้มีความเชี่ยวชาญหลายทางมากขึ้น ระบบความเชี่ยวชาญหลายทางเหล่านี้เรียกว่า HEARSAY II และ HEARSAY III

* ระบบผู้เชี่ยวชาญมีความจำเป็นหรือไม่ ? แม้ว่ามันเป็นประสบการณ์ที่ยิ่งใหญ่สำหรับการสร้างระบบผู้เชี่ยวชาญ แต่ถ้าไม่มีใครสนใจที่จะใช้มัน มันก็จะไม่มีความหมายอะไร ถ้ามีผู้เชี่ยวชาญ มันก็ยากที่จะตัดสินความรู้ระบบผู้เชี่ยวชาญกับคนที่ขาดความเชี่ยวชาญ เช่นเดียวกับเวลาที่ผู้เชี่ยวชาญหรือผู้ใช้ไม่ต้องการใช้ระบบ แม้ว่ามันจะจำเป็นก็ตาม

การจัดการต้องมีการรับรองแม้ว่าว่าระบบผู้เชี่ยวชาญจะถูกวิพากษ์วิจารณ์มากกว่าโปรแกรมทั่วไป เพราะว่า ระบบผู้เชี่ยวชาญเป็นเทคโนโลยีตัวใหม่ คนก็มีความรู้ในการใช้น้อยและไม่แน่ว่ามันจะมีประสิทธิภาพมากพอ อย่างไรก็ตามขอบเขตของระบบผู้เชี่ยวชาญก็น่าที่จะได้รับการรับรองมากกว่านี้ เพราะว่าระบบผู้เชี่ยวชาญก็สามารถที่จะแก้ปัญหาได้ แต่ โปรแกรมทั่วไปไม่สามารถที่จะแก้ปัญหาได้อย่างระบบผู้เชี่ยวชาญ แม้ว่าจะมีความเสี่ยงแต่มันก็เป็นระบบที่ทำงานได้ดี

* ต้องมีผู้เชี่ยวชาญอย่างน้อยหนึ่งคนที่จะเป็นคนประสานงานหรือไม่ ? ต้องเป็นผู้เชี่ยวชาญจริงๆ และต้องมีความกระตือรือร้นมาก ผู้เชี่ยวชาญไม่ทุกคนที่จะมีการตรวจสอบข้อมูลที่ผิดพลาด แม้ว่ามีผู้เชี่ยวชาญหลายด้านที่จะร่วมมือกันพัฒนา จำกัดขอบเขตความฉลาดของผู้เชี่ยวชาญที่เกี่ยวข้อง ความเชี่ยวชาญที่แตกต่างกันก็มีการแก้ปัญหาที่ต่างกัน บางทีพวกผู้เชี่ยวชาญก็นำข้อสรุปที่ต่างกันมารวมๆ กัน พยายามที่จะหาหลายวิธีมาแก้ปัญหาให้เป็นหนึ่งวิธีแม้ว่ามีความขัดแย้งกันอยู่ภายใน

* ผู้เชี่ยวชาญสามารถอธิบายข้อมูลที่เป็นความเข้าใจในรูปแบบข้อมูลทางด้านวิศวกรรม ? ถ้าผู้เชี่ยวชาญร่วมมือกันก็จะเกิดความยุ่งยากของข้อมูลที่ต่างกันอย่างแน่นอน ตัวอย่างง่ายๆ ของความยุ่งยาก เช่นคุณสามารถอธิบายความหมายว่า คุณสามารถเคลื่อนไหวนิ้วของคุณได้อย่างไร

คุณก็ตอบได้ว่า เกิดจากการเกร็งกล้ามเนื้อของนิ้วมือ และ คำถามต่อไปก็คือ แล้วจะเกร็งกล้ามเนื้อส่วนนิ้วมือได้อย่างไร ความยุ่งยากแบบอื่นๆ ในการสื่อสารระหว่างผู้เชี่ยวชาญกับข้อมูลทางด้านวิศวกรรมที่ไม่มีความรู้ทางด้านข้อมูลทางเทคโนโลยีของระบบผู้เชี่ยวชาญ ปัญหานี้ก็คือ คำศัพท์ทางการแพทย์โดยเฉพาะ มันอาจจะเวลาในการทำความเข้าใจข้อมูลทางวิศวกรรมเป็นปีหรือมากกว่านั้นกว่าที่จะฟังพวกผู้เชี่ยวชาญเค้าพูดกันหรือกว่าจะแปลเป็นรหัสข้อมูลของคอมพิวเตอร์ได้

* ข้อมูลที่แก้ปัญหาส่วนใหญ่จะเป็นตัวช่วยแก้ปัญหาและความไม่แน่นอนจริงหรือไม่ ? ระบบผู้เชี่ยวชาญจะเหมาะสมเมื่อความรู้ของระบบนี้ใช้ในการแก้ปัญหาและมีความไม่แน่นอน บางทีข้อมูลนั้นเป็นพื้นฐานประสบการณ์ ซึ่งเรียกว่า ความรู้ประสบการณ์ (experiential knowledge) และผู้เชี่ยวชาญอาจจะมีวิธีมากมายที่จะลองใช้ อีกทางหนึ่ง อาจเป็นการวิธีทดลองและวิธีที่ผิดมากกว่า วิธีที่ขึ้นอยู่กับวิธีตรรกะ และ การแก้ปัญหาที่เป็นขั้นตอน อย่างไรก็ตามผู้เชี่ยวชาญยังคงแก้ปัญหาได้เร็วกว่าผู้ที่ไม่มีความรู้เลย การประยุกต์ที่ดีสำหรับระบบผู้เชี่ยวชาญ ถ้าปัญหาสามารถแก้ได้ง่ายๆ โดยวิธีตรรกะ และ การแก้ปัญหาที่เป็นขั้นตอนมันเป็นการแก้ปัญหาที่ดีที่สุดโดยโปรแกรมทั่วๆไป

8.7 ภาษา, ลักษณะภายนอกและเครื่องมือ

พื้นฐานการตัดสินใจในเวลาที่ต้องเผชิญกับปัญหาคือการตัดสินใจนั้นดีที่สุดหรือยังในบางทีประสบการณ์เป็นตัวที่เหมาะสมตัวแรกที่จะถูกเลือกนำมาใช้เป็นแบบอย่าง ประสบการณ์ยังเป็นสิ่งที่นิยมใช้ในรูปแบบการทำธุรกิจ ในการพิจารณาเพื่อที่จะเลือกนั้นจะเลือกตัวที่เคยใช้มาก่อน เช่น โปรแกรมที่ใช้งานทั่วๆไป ถ้าในกรณีที่โปรแกรมเหล่านี้ไม่สามารถนำมาใช้ได้เต็มประสิทธิภาพ เราก็จะเปลี่ยนไปใช้โปรแกรมอื่นๆ เช่น ระบบปัญญาประดิษฐ์ เป็นต้น

ถึงแม้ว่า ระบบผู้เชี่ยวชาญนั้น จะเป็นเพียงส่วนหนึ่งของระบบปัญญาประดิษฐ์แต่ก็ยังมีภาษาพิเศษสำหรับระบบผู้เชี่ยวชาญที่แตกต่างกับภาษาที่ใช้ในระบบปัญญาประดิษฐ์ เช่น LISP และ PROLOG ในขณะที่ได้มีการพัฒนาสิ่งเหล่านี้ เช่น IPL-II, SAIL, CONNIVER, KRL และ Smalltalk ซึ่งนิยมใช้กันอย่างแพร่หลาย

ระบบภาษาของระบบปัญญาประดิษฐ์มีความสามารถในการสั่งงานที่สูงกว่าภาษาอื่นๆ เช่น LISP หรือ ภาษา C เพราะว่ามันใช้งานง่ายกว่าทำให้ภาษาของระบบนี้เหมาะสมอย่างมากสำหรับการใช้เขียนระบบผู้เชี่ยวชาญ

หน้าที่หลักๆที่แตกต่างกันระหว่างภาษาของระบบผู้เชี่ยวชาญและภาษาโปรแกรมทั่วๆ ไป คือการนำเสนอ ระบบภาษาโดยทั่วไปจะเน้นในเรื่องความยืดหยุ่นและเทคนิคสำคัญๆ ในการนำเสนอข้อมูล ภาษาแบบใหม่ๆ เช่น Modula-2 และ Ada ถูกสร้างขึ้นเพื่อช่วยเหลือในเรื่องของ data abstraction โดยการจัดสรรโครงสร้างสำหรับการทำส่วนสรุป เช่น เกณฑ์วัดและรูปแบบข้อมูลและวิธีการที่ใช้ในการจัดการข้อมูลที่มีความเกี่ยวพันกันอย่างเหนียวแน่น ขั้นตอนของระบบผู้เชี่ยวชาญยอมรับข้อมูล abstract 2 ระดับ คือ data abstraction และ knowledge abstraction

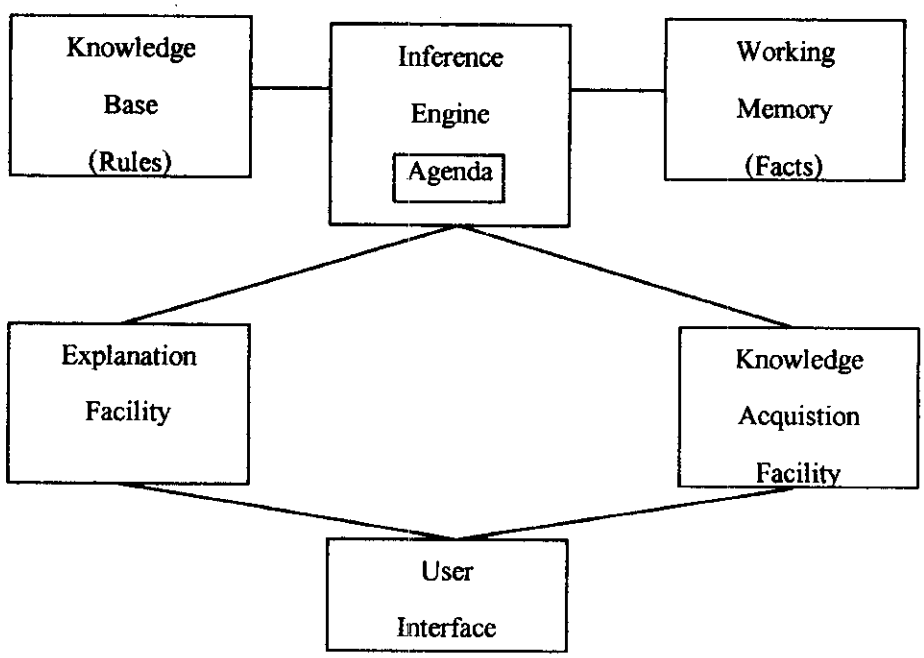
ภาษาของระบบมีการแบ่งแยกข้อมูลออกจากการจัดการข้อมูล ตัวอย่างของการแบ่งแยกคือ ข้อมูลที่เป็นความจริง (data abstraction) และ กฎเกณฑ์ (knowledge abstraction) ความแตกต่างในเรื่องที่เราสนใจจะนำไปสู่การออกแบบวิธีที่แตกต่างออกไป เนื่องด้วยความสัมพันธ์ของข้อมูลและความรู้ในเรื่องกระบวนการของภาษา นักเขียนโปรแกรมต้องระวังในเรื่องของผลลัพธ์ของการปฏิบัติ อย่างไรก็ตามความชัดเจนในการแบ่งแยกข้อมูลออกจากความรู้ในเรื่องของภาษาระบบผู้เชี่ยวชาญ

* ภาษา ผู้แปลคำสั่งจะเขียนให้อยู่ในรูปแบบของโครงสร้างประโยค ระบบผู้เชี่ยวชาญจะวิเคราะห์ในเรื่องประโยคของภาษาขึ้นอยู่กับการปฏิบัติ ประโยคอาจเป็นตัวส่งไปและส่งกลับภายใต้คำนามของภาษา LISP ไม่ใช่ ภาษาระบบผู้เชี่ยวชาญ ในขณะ PROLOG นั้นเป็นภาษาระบบผู้เชี่ยวชาญ แต่เราก็สามารถใช้ LISP ในการเขียนภาษาระบบผู้เชี่ยวชาญได้หรือเขียนภาษาของระบบปัญญาประดิษฐ์โดยใช้ PROLOG ได้

ลักษณะของระบบผู้เชี่ยวชาญมีหลากหลาย เช่น การนำเสนอความรู้, ส่งต่อ หรือ ย้อนกลับในระบบลูกโซ่, สนับสนุนในส่วนของความไม่แน่นอน, เหตุผลทางสมมติฐาน, ให้ความสะดวกในเรื่องคำอธิบาย และอื่นๆ ทางที่ดีที่สุด ในการเรียนรู้เทคโนโลยีระบบผู้เชี่ยวชาญ คือ การพัฒนาระบบให้เรียนรู้ภาษาได้ง่าย และสร้างความคุ้นเคยกับอุปกรณ์ ถ้าคุณมีความจำเป็น

8.8 องค์ประกอบของระบบผู้เชี่ยวชาญ

องค์ประกอบของระบบผู้เชี่ยวชาญแบบหนึ่ง จากรูป 7.6 หลักการพื้นฐานของระบบข้อมูลพื้นฐาน ประกอบด้วยข้อมูลตัวแปรที่ต้องการการแก้ปัญหาอย่างมีหลักการ ขณะที่หลักการเป็นการแก้ปัญหาที่เป็นขั้นตอนที่นิยมสำหรับการแทนที่ข้อมูล ระบบผู้เชี่ยวชาญประเภทอื่นๆ ใช้แทนที่ต่างกันไป



รูปที่ 7.6 Structure of a Rule-Based Expert System

ระบบผู้เชี่ยวชาญประกอบด้วย

- การติดต่องานของผู้ใช้ : ระบบเครื่องกลไกโดย ผู้ใช้และการติดต่อระบบผู้เชี่ยวชาญ
- การอธิบายความสะดวก : อธิบายระบบอย่างมีเหตุผลกับผู้ใช้
- การทำงานของหน่วยความจำ : ข้อมูลพื้นฐานทั่วไปของการใช้โดยอาศัยกฎ
- ข้อสรุป : การสรุปจาก0การตัดสินใจกฎที่น่าพอใจ
- ระเบียบวาระการประชุม (Agenda): รายการของหลักการที่สร้างขึ้นโดยข้อสรุปของเครื่องยนต์กลไก ที่เป็นแบบที่น่าพอใจจากการทำงานของหน่วยความจำ
- ข้อมูลที่ได้มาจากความสะดวก : วิธีอัตโนมัติสำหรับผู้ใช้ระบบข้อมูลมากกว่าวิธีของข้อมูลทางวิศวกรรม
- ข้อมูลที่ได้มาจากความสะดวกเป็นระบบลักษณะที่เลือกได้ ในอุปกรณ์บางระบบผู้เชี่ยวชาญเหมือน KEE และ ชั้นหนึ่ง (First คลาส) อุปกรณ์สามารถที่จะเรียนรู้โดยกฎหลักความจริงผ่านการยกตัวอย่าง และ หลักการทั่วไปอัตโนมัติ อย่างไรก็ตาม ตัวอย่างทั่วไปจากตารางหรือประเภทตารางคำนวณเหมาะสมกว่าการโครงสร้างการตัดสินใจ

หลักการทั่วๆ ไปถูกสร้างขึ้นโดยข้อมูลทางวิศวกรรมที่มีความสามารถมากมายซับซ้อนมากกว่าหลักการง่ายๆจากหลักการจริง

ข้อมูลพื้นฐานยังเรียกว่า การผลิตหน่วยความจำ (Production memory) ในหลักการระบบผู้เชี่ยวชาญพื้นฐาน เป็นตัวอย่างแบบง่ายๆ การตัดสินใจเวลาที่จะข้ามถนน การผลิตสำหรับหลักการสองหลักการที่มีลูกศรหมายความว่าระบบนั้นจะแสดงการทำงานทางขวาของลูกศร ถ้าเงื่อนไขทางซ้ายนั้นเป็นจริง

ไฟแดง → หยุด

ไฟเขียว → ไป

การผลิตสามารถแสดง ถ้า ก็..... (IF THEN...) ได้

หลักการ : แดง_ไฟ

ถ้า

ไฟแดง

ก็

หยุด

หลักการ : เขียว_ไฟ

ถ้า

ไฟเขียว

ก็

ไป

แต่ละหลักการเป็นการบ่งชี้โดยชื่อ ตามชื่อคือส่วนหลักการ ถ้า หน่วยของหลักการระหว่างส่วน ถ้า และ ก็ ที่ถูกเรียกก็มีหลายชื่อ เช่น เกิดขึ้นก่อน (Antecedent) , ส่วนของเงื่อนไข (conditional part) , ส่วนแบบแผน (pattern part) หรือ ฝั่งซ้ายมือ (left-hand-side) (LHS)

ไฟเขียว

ถูกเรียกว่าเงื่อนไขตัวประกอบ หรือ แบบแผน

ในหลักการพื้นฐาน ข้อสรุปของกลไกกำหนดที่เป็นหลักการมาก่อน ถ้ายันอื่นเป็นที่น่าพอใจ วิธีสรุปทั่วไปสองวิธีที่ใช้กันอย่างแพร่หลายคือ การส่งต่อ และ ส่งกลับ เป็นวิธีการแก้ปัญหาของระบบผู้เชี่ยวชาญวิธีอื่นๆ ใช้สำหรับระบบแบบเฉพาะที่ต้องการการวิเคราะห์, การลด

ปัญหา, การกลับสู่วิถีทางเดิม, การวางแผนเพื่อทำการทดสอบ , การวางแผนเป็นลำดับขั้น , การควบคุมการจัดการ

การส่งต่อเป็นการชี้แจงอย่างมีเหตุผล ไปสู่การสรุปผลจากข้อเท็จจริง เช่น ถ้าคุณเห็นว่าฝนกำลังจะตกก่อนออกจากบ้าน(ข้อเท็จจริง)ดังนั้นก็ควรที่จะพกร่มไปด้วย(ข้อสรุป)

การส่งกลับมีการชี้แจงอย่างมีเหตุผลเกี่ยวกับการกลับกันจากข้อสันนิษฐาน , การสรุปอย่างมีศักยภาพจากการพิสูจน์ , ข้อเท็จจริงที่สนับสนุนข้อสันนิษฐาน ตัวอย่าง ถ้าคุณไม่ได้มองออกไปข้างนอกและมีบางคนเข้ามาพร้อมกับรองเท้าและร่มเปียกๆ คุณก็คิดได้ว่าสงสัยฝนกำลังตก การหาข้อสนับสนุนของข้อสันนิษฐานของคุณได้ก็คือ การถามคนคนนั้นว่าฝนตกหรือไม่ ถ้าคำตอบนั้นคือใช่ ก็แสดงว่าข้อสันนิษฐานนั้นเป็นถูกต้องและเป็นจริง อย่างที่กล่าวมาก่อนหน้านี้แล้ว ข้อสันนิษฐานนั้นสามารถแสดงถึงความสงสัยและต้องการทดสอบ สันนิษฐานสามารถเป็นตัวเชื่อมระหว่างวัตถุประสงค์กับข้อพิสูจน์

ขึ้นอยู่กับการออกแบบ ข้อสรุปของเครื่องยนต์กลไกไม่ว่าจะส่งต่อ หรือ ส่งกลับ อย่างเช่น OPS5 และ CLIPS ที่ถูกออกแบบสำหรับการส่งต่อ ขณะที่ EMYCIN แสดงการส่งกลับ ข้อสรุปของกลไกบางประเภท ก็เสนอทั้งสองอย่าง เช่น ART และ KEE ตัวเลือกของข้อสรุปของเครื่องยนต์กลไกขึ้นอยู่กับประเภทของปัญหา การวินิจฉัยปัญหาจะ แก้ปัญหาได้ดีกว่าการส่งกลับ ขณะที่การคาดเดา , การใช้เครื่องรับฟังสัญญาณเพื่อทดสอบคุณภาพของการส่งสัญญาณโทรทัศน์และการควบคุมจะดีกว่าการส่งต่อ

การทำงานของหน่วยความจำสามารถบรรจุข้อเท็จจริงที่ได้รับการพิจารณาสถานภาพของไฟจราจร เช่น ไฟเขียว หรือ ไฟแดง ข้อเท็จจริงทั้งคู่จะทำงานอยู่ในหน่วยความจำ ณ เวลาเดียวกัน ถ้าสัญญาณไฟกำลังทำงานปกติก็จะมีข้อเท็จจริงเพียงหนึ่งตัวเท่านั้นที่จะอยู่ในหน่วยความจำ อย่างไรก็ตาม มันมีความเป็นไปได้ที่ข้องเท็จจริงทั้งสองอาจอยู่ในการทำงานของหน่วยความจำ ถ้ามีการทำงานที่ผิดของไฟสัญญาณสังเกตว่ามีความแตกต่างในเรื่องของความรู้พื้นฐานกับการทำงานของหน่วยความจำ ข้อเท็จจริงจะไม่มีปฏิกิริยากับข้อเท็จจริงตัวอื่นๆ ข้อเท็จจริงที่ว่าสัญญาณไฟเป็นสีเขียวจะไม่มีผลต่อข้อเท็จจริงที่ว่าสัญญาณไฟเป็นสีแดง นอกเหนือจากนี้ความรู้ของเราในเรื่องสัญญาณไฟจราจรอาจกล่าวได้ว่าข้อเท็จจริงทั้งสองเกิดขึ้นพร้อมกัน ดังนั้นก็จะเกิดข้อผิดพลาดในเรื่องของสัญญาณไฟ

ถ้ามีข้อเท็จจริงที่ว่าสัญญาณไฟเป็นสีเขียวอยู่ในการทำงานของหน่วยความจำ ตัวสรุปผลจะสังเกตได้ว่าข้อเท็จจริงตัวนี้อยู่ในสภาวะที่เหมาะสม ในส่วนของการแสดงสัญญาณไฟสีเขียว

และใส่ส่วนนี้ลงไปในหน่วยบันทึก ถ้าหลักการมีหลายรูปแบบ ดังนั้นทุกๆรูปแบบจะต้องเหมาะสมสำหรับการเป็นหลักการที่บันทึกอยู่ในหน่วยบันทึก อาจจะมีบางรูปแบบที่มีความเหมาะสมโดยการกำหนดข้อยกเว้นของข้อเท็จจริงในส่วนการทำงานของหน่วยของจำ หลักการที่นิยมใช้เป็นรูปแบบที่เหมาะสมจะถูกกล่าวว่าเป็นตัวกระตุ้นหรือตัวเร่ง หลักเกณฑ์ที่เป็นตัวกระตุ้นหลายๆ ข้ออาจอยู่ในหน่วยการบันทึก ณ เวลาเดียวกัน ในกรณีนี้ ตัวสรุปต้องทำการสรุปหลักการขึ้นมาหนึ่งข้อเพื่อนำไปใช้ ในส่วนของการนำไปปฏิบัติมาจากระบบสรีระวิทยาที่เกี่ยวกับระบบประสาท การศึกษาเกี่ยวกับระบบการตอบสนอง เซลล์ตอบตอบสนองแต่ละตัว หรือ ที่เรียกว่า Neuron จะกระจายสัญญาณไฟฟ้า เมื่อถูกกระตุ้น ไม่ว่าจะมีตัวกระตุ้นมากเท่าใดก็ตาม ก็ไม่สามารถทำให้ neuron ถูกนำกลับมาใช้ได้อีกครั้งในช่วงระยะสั้นๆ ปรากฏการธรรมชาตินี้เรียกว่า การหักเห หรือการเปลี่ยนทิศทาง ซึ่งก็คือ ถ้าหลักการไฟสีเขียวถูกนำมาใช้หลายๆครั้ง ระบบผู้เชี่ยวชาญจะไม่สามารถบรรลุการทำงานได้

มีหลายวิธีที่ถูกนำมาใช้ในเรื่องของการเปลี่ยนทิศทาง หนึ่งในรูปแบบของภาษาระบบผู้เชี่ยวชาญ เราเรียกว่า ระบบ OPS5 แต่ละข้อเท็จจริงจะถูกแบ่งแยกออกอย่างชัดเจนซึ่งเราเรียกว่า time tag เมื่อมันถูกบรรจุอยู่ในการทำงานของหน่วยความจำ หลังจากหลักการได้ถูกนำมาใช้ผลสรุปจะไม่ถูกนำมาใช้ในข้อเท็จจริงนั้นๆอีกครั้ง เพราะว่ามันได้ถูกใช้งานไปแล้ว

หลักการของ ก็ เป็นรายการของการกระทำไปสู่ความสำเร็จ ส่วนนี้เป็นที่รู้จักเช่น ผลที่ตามมา หรือ ฝั่งขวามือ (RHS) เมื่อสัญญาณเป็นไฟแดงการกระทำที่ต้องทำคือ หยุด ก็เหมือนกับเมื่อสัญญาณเป็นไฟเขียวการกระทำที่ต้องทำคือ ไป การกระทำโดยเฉพาะทั่วไปรวม ข้อเงื่อนไข หรือการเปลี่ยนแปลงของข้อเท็จจริงจากการทำงานของหน่วยความจำหรือการประทับผล รูปแบบของการกระทำเหล่านี้ขึ้นอยู่กับการสร้างประโยคของระบบภาษาผู้เชี่ยวชาญ เช่น OPS , ART, และ CLIPS การกระทำได้เพิ่มข้อเท็จจริงตัวใหม่ลงไปเรียกว่า "หยุด" ในการทำงานของหน่วยความจำก็จะเป็น (แสดงการหยุด) เพราะว่า LISP ภาษาของมันถูกออกแบบให้อยู่ในวงเล็บ

ข้อสรุปของการทำงานเป็นวงจรหมุนเวียน มีหลายชื่อที่ใช้ในการอธิบายวงหมุนเวียน เช่น วงจรรู้จักการกระทำ (recognize-act cycle) วงจรเลือกการปฏิบัติ (select-execute cycle) วงจรสถานการณ์ตอบสนอง (situation-response cycle) และ วงจรสถานการณ์การกระทำ (situation-action cycle) ไม่ว่าจะเป็นชื่ออะไรสำหรับวงจร ข้อสรุปมีการทำงานซ้ำๆ จนกว่า จะได้เกณฑ์ที่แน่นอน วงจรของงานสำหรับ OPS5 ประเภทของระบบผู้เชี่ยวชาญเชลล์ ได้ถูกแสดงรหัสปลอมเป็นการตัดสินใจที่ขัดแย้งกัน, การกระทำ, การแข่งขัน และ ตรวจตราความไม่สมบูรณ์

ขณะที่ไม่เสร็จ

การตัดสินใจที่ขัดแย้งกัน : ถ้ามีการกระตุ้น ก็เลือกสิ่งที่มีประสิทธิภาพสูงสุดหนึ่งข้อ

การกระทำ : จากการแสดงการเลือกตัวกระตุ้นของ RHS การเปลี่ยนแปลงการทำงานของ หน่วยความจำมีผลกระทบทันทีในวงจร การเคลื่อนที่ของตัวกระตุ้นก็จะทำให้เกิดการทำงานจากการบันทึก

การแข่งขัน : การบันทึกข้อมูลใหม่ๆ จากการตรวจสอบ ถ้า LHS ของหลักการต่างๆนั้นเป็นที่น่าพอใจถ้ามีการกระตุ้นมัน เคลื่อนที่ตัวกระตุ้น LHS ของหลักการต่างๆก็ไม่เป็นที่ต้องการอีกแล้ว

ตรวจตราความไม่สมบูรณ์ : ถ้าความไม่สมบูรณ์ถูกแสดงหรือมีการแหกกฎ ก็ถือว่าแสร็จจบ

จบ

ยอมรับคำสั่งของผู้ใช้คนใหม่

หลักการมากมายอาจเป็นตัวกระตุ้นและเพิ่มขั้นตอนระหว่างหนึ่งวงจร โดยเฉพาะการกระตุ้นที่อยู่ด้านซ้ายของ ขั้นตอนจากวงจรอันแรก นอกจากหลักการเหล่านั้นถูกทำให้ไม่มีความสามารถ เพราะว่า LHS ของพวกมัน ไม่ค่อยเป็นที่น่าพอใจ จำนวนตัวกระตุ้นของ agenda มีหลากหลายในการทำงาน ขึ้นอยู่กับโปรแกรมการกระตุ้นอาจจะอยู่บน agenda เสมอ แต่มันไม่เคยถูกเลือกในการนำไปปฏิบัติ เหมือนกับหลักการบางหลักการอาจจะไม่เคยได้รับการกระตุ้นเลย ในกรณีเหล่านี้ จุดประสงค์ของหลักการก็คือควรที่จะนำมาตรวจดูอีกครั้ง เพราะว่าหลักการบางทีไม่จำเป็นหรือรูปแบบถูกออกแบบมาไม่ถูกต้อง

ข้อสรุปของกลไกปฏิบัติงานการกระทำของตัวกระตุ้นที่สูงสุดบน ขั้นตอนจากนั้นรอจนกระทั่งตัวกระตุ้นจะหลุดออกไป ตารางต่างๆ ก่อนหน้านี้ได้ถูกออกแบบในระบบผู้เชี่ยวชาญ การขัดแย้งของ ขั้นตอนเกิดขึ้นเมื่อการกระตุ้นที่แตกต่างกันมีที่มาเหมือนกันและข้อสรุปของเครื่องยนต์กลไก ต้องตัดสินใจการทำงาน เซลล์ที่แตกต่างกันก็มีปัญหาที่ต่างกันอย่างตัวอย่างของเนเวลล์และไซมอน หลักการเหล่านั้นเข้าไปในระบบที่ไม่สามารถทำงานได้ ใน CPS5 หลักการจะซับซ้อนมากกว่า ART และ CLIPS หลักการมีข้อบกพร่องเหมือนอันก่อน นอกจากจะกำหนดข้อแตกต่างโดยข้อมูลทางวิศวกรรม

ตอนนี้การควบคุมย้อนกลับไปคำสั่งขั้นสูงสุด (Top level) สำหรับผู้ใช้ให้คำแนะนำระบบผู้เชี่ยวชาญเชลล์ ขั้นสูงสุดเป็นวิธีบกพร่องที่ผู้ใช้ติดต่อสื่อสารกับระบบผู้เชี่ยวชาญและการสั่งงาน "ยอมรับคำสั่งของผู้ใช้คนใหม่" ขั้นสูงสุดยอมรับคำสั่งใหม่ ขั้นสูงสุดเป็นผู้ใช้ที่ติดต่อกับเชลล์ ขณะที่การประยุกต์ระบบผู้เชี่ยวชาญอยู่ภายใต้การพัฒนา ผู้ใช้ที่มีความเชี่ยวชาญมากกว่าจะถูกออก

แบบสำหรับระบบผู้เชี่ยวชาญให้สะดวกต่อการทำงาน เช่น ระบบผู้เชี่ยวชาญอาจจะมีผู้ใช้ติดต่อสำหรับการควบคุมการผลิต

8.9 ระบบการผลิต

หนึ่งในรูปแบบของระบบผู้เชี่ยวชาญที่นิยมมากที่สุด ในปัจจุบัน คือ ระบบหลักการพื้นฐานเป็นสิ่งที่ได้รับความนิยมด้วยเหตุผลดังต่อไปนี้

- *ลักษณะโดยทั่วไป : ทำให้ง่ายในการสรุปความรู้และขยายระบบผู้เชี่ยวชาญโดยมีการพัฒนาที่สูงขึ้น
- *สะดวกต่อการอธิบาย : มันง่ายต่อการสร้างหลักการการอธิบาย ซึ่งหลักการที่เกิดขึ้นมาก่อนจะกำหนดอย่างชัดเจนว่าอะไรคือสิ่งสำคัญ ในการกระตุ้นให้เกิดหลักการ โดยการเก็บข้อมูลว่าหลักการไหนที่ใช้ไม่ได้ การอธิบายสามารถนำแสนอเหตุผลลูกโซ่ ซึ่งจะนำไปสู่การสรุปที่แน่นอน
- *ความคล้ายคลึงในเรื่องกระบวนการรับรู้ของมนุษย์ : พื้นฐานในงานของเนเวลล์และไซมอน หลักการเกิดขึ้นมาเพื่อที่มนุษย์จะได้ในการแก้ปัญหา ในประโยค ถ้า ก็... เป็นการนำเสนอหลักการที่ทำให้การอธิบายโครงสร้างของความรู้ได้ง่ายขึ้น ซึ่งคุณพยายามค้นหาความจริงจากพวกมัน ข้อดีอื่นๆของหลักการถูกบรรยายใน Hayes Roth 85

รูปแบบการผลิตที่จุดกำเนิดย้อนหลังไปในปี 1940 เนื่องด้วยความสำคัญของหลักการ พื้นฐานมันคุ้มค่าต่อการพัฒนาแนวคิดนี้ขึ้นมา แล้วคุณจะรู้ว่าทำไม ระบบนี้ถึงมีประโยชน์ อย่างมากต่อระบบผู้เชี่ยวชาญ

ระบบหลังการผลิต

ระบบการผลิตถูกใช้ในรูปของสัญลักษณ์ทางตรรกศาสตร์ โดย Post 43 ผู้ที่เป็นคนคิดค้นชื่อขึ้นมา

เขาพิสูจน์ความสำคัญและความมหัศจรรย์ในผลลัพธ์ ซึ่งทุกๆระบบทางคณิตศาสตร์หรือตรรกศาสตร์สามารถถูกเขียนเป็นรูปแบบในรูปของระบบหลักการการผลิต ผลลัพธ์ของการจัดตั้งหลักการ การผลิตที่มีประสิทธิภาพสูง เพื่อเป็นการแสดงถึงระดับความรู้หลักมากกว่าข้อจำกัดภายใต้การเขียนหลักการซ้ำมีการนำมาใช้ในส่วนของภาษาศาสตร์ เป็นหนทางในการหาไวยากรณ์ของภาษา ภาษาคอมพิวเตอร์ปกติเราพบว่ามีการใช้

Backus-Naur Form (BNF) ของหลักการการผลิต

พื้นฐานความคิดแบบ Post คือระบบคณิตศาสตร์หรือตรรกศาสตร์ สามารถถูกเขียนเป็นกลุ่มของหลักการที่กำหนดค่า การเปลี่ยนสัญลักษณ์หนึ่งที่ให้อยู่ในรูปของอีกกลุ่มสัญลักษณ์ได้อย่างไร

แนวคิดนี้ยังควรสมเหตุสมผลกับโปรแกรมและระบบผู้เชี่ยวชาญกลุ่มสัญลักษณ์เป็นข้อมูลในตอนแรกและผลลัพธ์ที่ได้มาบางส่วนจะเปลี่ยนไปอยู่ในรูปของข้อมูล

ตัวอย่างที่นิยมมากที่สุดคือ ถ้ากลุ่มของวัตถุดิบ คือ คนไข้ตัวร้อน ผลลัพธ์ควรเป็น กินยาแก้ไข้ สังเกตว่าไม่มีความหมายแนบติดมากับสองกลุ่มนี้ ซึ่งก็คือ การโอนถ่าย ขึ้นอยู่กับ การสร้างประโยคและไม่ว่าความหมายของคำใดๆ ก็ตามหรือความเข้าใจว่าอะไรคือสิ่งที่แสดงถึง ตัวร้อน, ยาแก้ไข้ และคนป่วย มนุษย์รู้สิ่งเหล่านี้จากโลกของความเป็นจริง ระบบหลังการผลิตเพียงแค่เปลี่ยนจากสิ่งหนึ่งไปเป็นอีกสิ่งหนึ่ง

หลักการการผลิตสำหรับการผลิตตัวอย่างนี้
เงื่อนไข เหตุ (Antecedent) → ผล (consequent)
คนไข้ที่มีไข้ตัวร้อน → กินยาลดไข้
เราสามารถรวมเข้าด้วยกันได้ด้วยหลักของ ถ้า ก็..... (IF THEN....)
ถ้าคนไข้มีไข้ก็ให้กินยาลดไข้
หลักการการผลิตสามารถหาสาเหตุที่เกิดได้มากกว่า 1 อย่าง
คนที่ตัวร้อนและไข้เกินกว่า 102 → ไปพบแพทย์

สังเกตตัวเชื่อมพิเศษ และ (And) ไม่ใช่ส่วนหนึ่งของ ประโยค ตัวเชื่อม และเป็นตัวบ่งชี้ว่า มีหลายสาเหตุหรือหลายเงื่อนไข

ระบบหลังการผลิต ประกอบไปด้วยกลุ่มของของหลักการการผลิตดังต่อไปนี้
1. รถจะสตาร์ทไม่ติด → ตรวจดูแบตเตอรี่
2. รถจะสตาร์ทไม่ติด → ตรวจดูน้ำมัน
3. ตรวจดูแบตเตอรี่ และ แบตเตอรี่เสีย → เปลี่ยนแบตเตอรี่ใหม่
4. ตรวจดูน้ำมัน และ น้ำหมดหมด → เติมน้ำมัน

ถ้ามี ประโยค ว่า รถจะสตาร์ทไม่ติด หลักการข้อที่ 1 และ 2 อาจนำมาใช้เพื่อสร้างประโยค ตรวจดูแบตเตอรี่ และตรวจดูน้ำมัน อย่างไรก็ตาม ไม่มีหลักการข้อบังคับใดๆ ให้

CT 488 167

ประโยค ตรวจดูแบตเตอรี่ และตรวจดูน้ำมัน อย่างไรก็ตาม ไม่มีหลักการข้อบังคับใดๆ ให้

ประยุกต์หลักการทั้งคู่ เข้ากับ ประโยค มีเพียงหลักการตัวเดียวที่สามารถใช้ได้ อาจใช้ได้ทั้งคู่ หรือ อาจจะใช้ไม่ได้เลย ถ้ามี ประโยค ตัวอื่นๆ แบตเตอรี่เสีย และ ประโยค ตรวจดูแบตเตอรี่ ดังนั้น หลักการข้อที่ 3 อาจนำมาประยุกต์ใช้ เพื่อสร้าง ประโยค เปลี่ยนแบตเตอรี่ใหม่

4. ตรวจดูน้ำมัน และ แบตเตอรี่หมด → เติมเชื้อเพลิง

2. รถสตาร์ทไม่ติด → ตรวจดูน้ำมัน

1. รถสตาร์ทไม่ติด → ตรวจดูแบตเตอรี่

3. ตรวจดูแบตเตอรี่และแบตเตอรี่เสีย → เปลี่ยนแบตเตอรี่ใหม่

ถึงแม้ว่าหลักการหลังการผลิตจะมีประโยชน์ต่อระบบพื้นฐานของผู้เชี่ยวชาญ แต่มันก็ยังไม่เหมาะสมกับการนำมาเขียนโปรแกรมทดสอบ

ข้อจำกัดของระบบนี้คือ ขาดกลยุทธ์การควบคุม เพื่อเป็นแนวทางในการนำหลักเกณฑ์มาประยุกต์ใช้ ระบบจะอนุญาตให้หลักเกณฑ์ควรใช้อย่างไร

มันเหมือนเป็นสิ่งที่คล้ายคลึงกัน สมมติว่าคุณไปห้องสมุดเพื่อหาหนังสือเล่มหนึ่งบนระบบผู้เชี่ยวชาญ ณ ห้องสมุด คุณเริ่มสุ่มหาหนังสือบนชั้นวางที่คุณต้องการ ถ้าห้องสมุดนั้นมีขนาดใหญ่ มันก็จะใช้เวลานานในการค้นหาหนังสือที่คุณต้องการ ถึงแม้ว่าคุณหากลุ่มหนังสือบนระบบผู้เชี่ยวชาญ การสุ่มหากลุ่มหนังสือครั้งต่อไปของคุณ สามารถทำให้คุณเจอกลุ่มหนังสือที่มีความแตกต่างกัน

สถานการณ์จะยิ่งเลวร้ายลง ถ้าคนต้องข้อมูลจากหนังสือเล่มแรก เพื่อช่วยให้คุณหาหนังสือเล่มที่สองที่คุณต้องการ การสุ่มหาหนังสือเล่มที่ต้องใช้เวลาหานานเช่นกัน

วิธีของ Markov

ขั้นต่อไปเป็นการประยุกต์หลักเกณฑ์ที่คิดค้นโดย Markov ผู้กำหนด ควบคุมโครงสร้างสำหรับระบบการผลิต วิธีของ Markov เป็นการเรียงลำดับกลุ่มการผลิตซึ่งจะถูกใช้ในวัตถุดิบหลักในสายอักขระ

ถ้าหลักเกณฑ์ที่สำคัญที่สุดไม่สามารถประยุกต์ใช้ได้ ก็จะใช้ตัวต่อไป

วิธีของ Markov จะใช้ไม่ได้ผลก็ต่อเมื่อ

การผลิตครั้งล่าสุด ไม่สามารถประยุกต์ให้เข้ากันสายอักขระได้

จุดสุดท้ายของการผลิต ณ ช่วงเวลานั้น

วิธีของ Markov ยังประยุกต์ออกเป็นสานอักขระย่อยๆ ได้อีก

อีกหนึ่งความหมายที่มักใช้เป็นประจำสำหรับโปรแกรมลำดับขั้น คือ โปรแกรมต่อเนื่อง อย่างไรก็ตามในส่วนของโปรแกรมต่อเนื่องมีความหมายที่มีข้อจำกัดค่อนข้างมาก ตั้งแต่ ภาษาของโปรแกรมใหม่ๆ ที่สนับสนุนและโปรแกรมส่วนใหญ่ไม่มีการจำกัดในเรื่องของความต่อเนื่อง ลักษณะภายนอกที่แตกต่างของรูปแบบขั้นตอน คือ นักโปรแกรมต้องกำหนดลงไปเลยว่า ทางแก้ปัญหานั้นต้องตีความอย่างไร เป้าหมายของโปรแกรมที่ไม่มีลำดับขั้น คือ การที่นักโปรแกรมกำหนดว่าอะไรคือเป้าหมายและกำหนดระบบเก่าจะบรรลุเป้าหมายได้อย่างไร

Imperative Programming

ในส่วนของ imperative และ statement -- oriented มีความหมายเหมือนกันภาษา เช่น FORTRAIN , Ada , Pascal , Modula-2 ,COBOL และ BASIC

Imperative language ยังเป็นตัวสนับสนุนที่ดีเยี่ยม มอบหมายงานที่ต้องปฏิบัติและทบทวนสิ่งเหล่านี้เป็น การทำงานในระดับต่างๆ ซึ่งภาษาสมัยใหม่พยายามที่จะแอบซ่อนไว้ โดยการจัดเตรียมรูปลักษณ์ เช่น การทำซ้ำ , ลำดับขั้นตอน , มาตรฐานที่สามารถวัดได้ , แพ็คเก็ตและอื่นๆ Imperative language ยังเน้นในส่วนของ โครงสร้างควบคุมที่ตายตัวและมีการจัดการโปรแกรมแบบ บน-ล่าง

ปัญหาที่น่าวิตกที่เกิดขึ้นกับทุกๆภาษา คือความยากในการพิสูจน์ความถูกต้องของโปรแกรม จากจุดยืนของระบบปัญญาประดิษฐ์ ปัญหาอื่นๆ คือ Imperative language ไม่ใช่ตัวที่มีประสิทธิผลสูงสุดเพราะว่า Imperative language เป็นเหมือนแม่พิมพ์ที่ตายตัวกับคอมพิวเตอร์ที่มีสถาปัตยกรรม Von Neuman เรามีภาษาที่สามารถสนับสนุนได้เป็นอย่างดีแต่ไม่ใช่การจัดการสัญลักษณ์ อย่างไรก็ตาม Imperative language เช่น C และ Ada ยังคงถูกใช้เป็นฐานของภาษา ในการเขียนโครงสร้างระบบผู้เชี่ยวชาญ ภาษาเหล่านี้และโครงสร้างที่สร้างขึ้นจากภาษาเหล่านี้ทำงานได้อย่างมีประสิทธิภาพและรวดเร็ว ตามจุดประสงค์การใช้งานคอมพิวเตอร์ทั่วไปมากกว่าการสร้างโครงสร้างโดยการใช้ LISP

เนื่องจากผลลัพธ์ที่เกิดขึ้นโดยปกติ Imperative language ไม่มีประสิทธิผลอย่างยิ่ง สำหรับการทำงานโดยตรงของระบบผู้เชี่ยวชาญ โดยเฉพาะอย่างยิ่ง หลักการพื้นฐานที่เป็นปัญหาที่เกิดขึ้นการพิจารณาปัญหาของการส่งผ่านข้อมูลของปัญหากับหลักการนับร้อยนับพัน ยกตัวอย่างเช่น ระบบ XCON ใช้โดย บริษัท ดิจิตอล อีควิปเมนท์ ทำโครงสร้าง ระบบคอมพิวเตอร์ ในปัจจุบันมีหลักการประมาณ 7,000 ข้อ ในเรื่องพื้นฐานความรู้ ในช่วงแรกที่ไม่ประสบความสำเร็จถูกทำเป็นรหัสใน FORTRAN และ BASIC ก่อนที่จะประสบความสำเร็จ

รหัสใน FORTRAN และ BASIC ก่อนที่จะประสบความสำเร็จ

การกำหนดทางโดยตรงในความรู้ของ Imperative language ต้องใช้ประโยค ถ้า ก็... ถึง 7,000 ประโยคหรือกรณีที่ยาวมากๆ รูปแบบของการกำหนดนี้จะนำเสนอปัญหาหลักๆ ของหลักการทั้ง 7,000 ข้อ ที่จำเป็นจะต้องค้นหาเพื่อเป็นรูปแบบการจับคู่บนวงจร สังเกตว่าการทำสรุปและการแยกแยะนั้นยังคงต้องทำการกำหนดใน Imperative language

ประสิทธิผลของโปรแกรมสามารถพัฒนาได้ ถ้าหลักการถูกจัดเรียงลำดับดังนั้นหลักการส่วนใหญ่จะถูกนำไปปฏิบัติตั้งแต่ตอนเริ่มต้น อย่างไรก็ตาม ยังคงต้องทำการลบและปรับปรุง เช่นเดียวกัน วิธีที่ดีกว่าในการพัฒนาประสิทธิผลควรสร้าง โครงสร้างของหลักการเป็นรูปแบบเพื่อเวลาในการค้นหาว่ากฎไหนควรนำมาใช้ ยิ่งไปกว่านั้น มันควรสร้างเป็นแบบอัตโนมัติ โดย ใช้รากฐานของคอมพิวเตอร์ เป็นรูปแบบและหลักการประโยค ถ้า ก็..... มันยังมีประโยชน์ใช้ในการช่วยเหลือมากในประโยค ถ้า ก็.... ซึ่งเป็นการนำเสนอความรู้และอำนาจการทดสอบของรูปแบบการจับคู่ ดังนั้นตัววิเคราะห์ไวยากรณ์จึงเป็นสิ่งที่ต้องการ ในการวิเคราะห์โครงสร้างและเป็นตัวแปลหรือเรียบเรียงให้เกิดประโยค ถ้า ก็..... ประโยคใหม่ขึ้น

เมื่อทุกๆ เทคนิค สำหรับการพัฒนาประสิทธิผลได้ถูกนำมาใช้แล้ว ผลลัพธ์ก็คือ ระบบผู้เชี่ยวชาญจะสามารถทำหน้าที่พิเศษโดยเฉพาะได้ ถ้าตัวสรุป ตัววิเคราะห์ไวยากรณ์และตัวแปลถูกพัฒนาให้ใช้งานง่ายขึ้นในระบบผู้เชี่ยวชาญ มันจะทำการประกอบโครงสร้างของระบบผู้เชี่ยวชาญเข้าด้วยกัน แน่นอนว่า แทนการทำสิ่งเหล่านี้จากการขุดค้นนั้น ในโลกปัจจุบันง่ายขึ้นกว่าเดิมมากเพียงแค่ใช้โครงสร้างที่มีซึ่งก็คือ เอกสาร และส่วนทดสอบอื่นๆ

หน้าที่ของโปรแกรม

ลักษณะโดยธรรมชาติหน้าที่ของโปรแกรมเป็น ตัวอย่างอธิบายโดยภาษา เช่น LISP และ APL ซึ่งเป็นสิ่งที่แตกต่างกันจากภาษาในประโยคกับความน่าเชื่อถือบนโครงสร้างการควบคุมที่ซับซ้อนและการออกแบบระบบบน-ล่าง

แนวคิดพื้นฐานของหน้าที่ของโปรแกรมคือ การรวม หน้าที่ทั่วไป ให้เกิดผลที่ดีในเรื่องความสามารถในการทำงาน นี่เป็นสิ่งสำคัญในแนวคิดแบบ ล่าง-บน สิ่งที่เป็นสิ่งที่ตรงกันข้าม กับ บน-ล่าง ใน Imperative language หน้าที่ของโปรแกรม คือ ศูนย์กลางของหน้าที่ ในทางคณิตศาสตร์ หน้าที่เป็นส่วนจัดการหรือหลักการ ซึ่งเป็นเหมือนแผนที่สมาชิกของสมาชิกหนึ่งกลุ่ม ตัวแปร อยู่ในรูปของกลุ่มอีกกลุ่มหนึ่ง ตัวแปรร่วม

เริ่มจากซ้าย ตัวอย่างเช่น ระบบการผลิต ประกอบไปด้วย หลักการเดียวกัน

AB → HIJ

เพื่อประยุกต์ให้เข้ากับวัตถุดิบของสายอักขระ BABKAB จะทำให้เกิดสายอักขระใหม่ GHIJKAB และผลสุดท้ายก็จะเป็น GHIJRHIJ

สัญลักษณ์พิเศษ (Λ) แทน null ประโยค เช่น

A → Λ

ลบทุกๆ ตัวอักษร A ที่เกิดขึ้นในสายอักขระ

สัญลักษณ์พิเศษที่แสดงโดยตัวอักษรเดี่ยวใดๆ ก็ตามถูกบ่งชี้โดย กรณีที่ตัวอักษรน้อยกว่า a,b,c และอื่นๆ

สัญลักษณ์เหล่านี้เสนอผ่านทางตัวอักษรเดี่ยวและเป็นส่วนที่สำคัญของภาษาระบบผู้เชี่ยวชาญแบบใหม่ ยกตัวอย่างเช่น

AxB → BxA

จะกลับตัวอักษรกัน A และ B

ตัวอักษรกรีก เช่น α,β และอื่นๆ ถูกใช้เป็นเครื่องหมายพิเศษในสายอักขระ ตัวอักษรกรีก ถูกนำมาใช้เพราะ มีความแตกต่างจากพยัญชนะโดยทั่วไป

ตัวอย่างวิธีของ Markov ในการย้ายตัวหนังสือตัวแรกของวัตถุดิบในสายอักขระ ไปสู่สุดท้ายตามตารางดังนี้ หลักการจะถูกเรียบเรียงในรูปของอันดับแรกสุด

1) ถัดไปจากตัวสูงสุด2) และลำดับถัดไปเรื่อยๆ

1) $\alpha xy \rightarrow y\alpha x$
2) $\alpha \rightarrow \Lambda$
3) $\Lambda \rightarrow \alpha$

สำหรับวัตถุดิบในสายอักขระ ABC ดูได้จากตาราง 7.11

CT 488

Rule	*Success or Failure*	*String*
1	F	ABC
2	F	ABC
3	S	□ABC
1	S	B□AC
1	S	BC□A
1	F	BC□A
2	S	BCA

สังเกตสัญลักษณ์ α เปรียบเหมือนกับตัวแปรชั่วคราวในภาษาของโปรแกรม อย่างไรก็ตามแทนที่การยึดเอามูลค่ามันไว้ α ถูกใช้แทนที่ตัวที่ยึด เพื่อเป็นการกำหนดกระบวนการในวัตถุดิบสายอักขระ เป็นงานเสร็จแล้วครั้งหนึ่ง ตัว α ถูกทำลาย โดยหลักการที่สอง

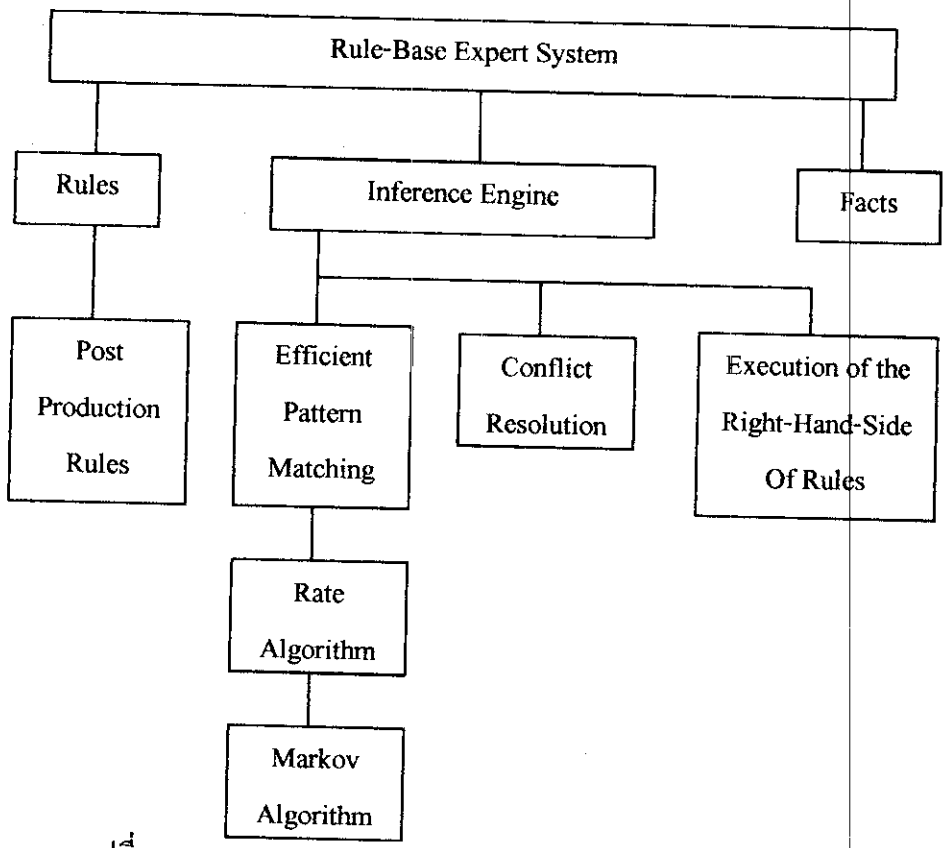
วิธีของ Rete

สังเกตว่ามีความชัดเจนในกลยุทธ์ควบคุมในวิธี Markov กับการเรียงลำดับหลักการ ตราบใดที่หลักการแรกสุดถูกนำมาใช้ ถ้าไม่แล้ววิธีของ Marcov ก็จะใช้หลักการลำดับถัดลงมา ถึงแม้วิธีของ Marcov สามารถใช้เป็นพื้นฐานของระบบผู้เชี่ยวชาญ แต่ก็ไม่เหมาะสมกับระบบที่มีหลายหลักการ ปัญหาในเรื่องของประสิทธิภาพกลายเป็นสิ่งที่สำคัญ ถ้าเราต้องการสร้างระบบผู้เชี่ยวชาญ สำหรับปัญหาที่แท้จริงที่ประกอบไปด้วย หลักการนับร้อยนับพัน ถ้าผู้ใช้ต้องรอระยะเวลาการตอบสนองที่ค่อนข้างนาน ระบบนั้นก็จะไม่นำมาใช้ สิ่งที่เราต้องการคือ วิธีที่รู้เกี่ยวกับทุกๆ หลักการและสามารถประยุกต์ได้ทุกๆหลักการ ปราศจาก การที่เราต้องพยายามใช้แต่ละหลักการอย่างต่อเนื่องกัน

ทางแก้ปัญหานี้คือวิธีของ Rete ถูกคิดค้นขึ้น โดย Clarles L.Forgy ที่มหาวิทยาลัย Carnegie-Mellon ในปี 1979 สำหรับวิทยานิพนธ์บน OPS โครงสร้างของระบบผู้เชี่ยวชาญ วิธีของ Rete เป็นรูปแบบตัวจับคู่ที่เร็วมาก โดยการจัดเก็บข้อมูลเกี่ยวกับหลักการในเครือข่าย แทนที่การจับคู่ข้อเท็จจริงที่ขัดแย้งกันทุกๆหลักการ ในวงจรวิธีของ Rete เพียงแค่มองหา เพื่อเปลี่ยนให้เข้ากันในทุกๆ วงจร ทำให้เกิดความเร็วในการจับคู่ข้อเท็จจริงที่เกิดขึ้น ตั้งแต่ข้อมูลที่คงที่ไม่เปลี่ยนแปลงจากวงจรไปสู่อีกวงจรสามารถละทิ้งไปได้

หัวข้อนี้จะกล่าวถึงในบทที่ว่าด้วย CLIPS

จากรูป 7.7 จะสรุปเทคโนโลยีที่เป็นกฎพื้นฐานของระบบผู้เชี่ยวชาญไว้ดังนี้



รูปที่ 7.7 Foundations of Modem Rule-based Expert Systems

รุ่นภาษาของ OPS ที่แตกต่างกันได้ถูกพัฒนาขึ้น เช่น OPS 2 , OPS 3 และ OPS 5 และยังมีรุ่นใหม่ที่พัฒนาโดย Forgy คือ OPS 83 ซึ่งมีระดับความเร็วที่สูงมาก อย่างไรก็ตาม OPS 83 ยังมีความแตกต่างจาก OPS 5 ในเรื่องรูปแบบการสร้างประโยคและมันเป็นส่วนหนึ่งของความเร็วในการปฏิบัติงาน

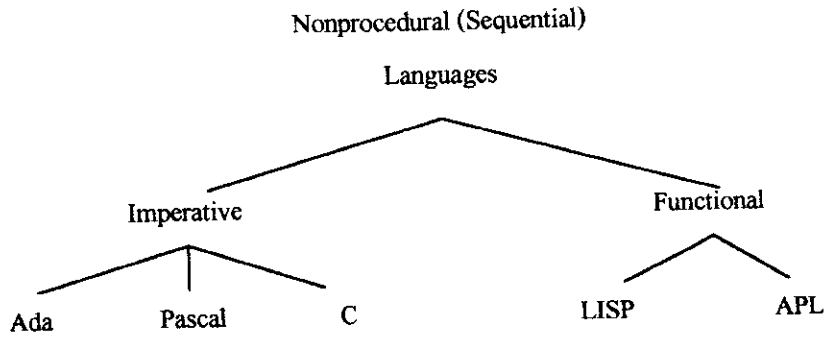
8.10 ขั้นตอนที่มีรูปแบบ

รูปแบบของโปรแกรมสามารถแบ่งแยกเป็นแบบมีขั้นตอนและไม่มีขั้นตอน รูปที่ 7.8 แสดงถึงเทคนิคการแบ่งประเภท ของขั้นตอนที่มีรูปแบบในรูปของภาษา รูปที่ 7.9 แสดงถึงเทคนิคการแบ่งแยกประเภทสำหรับรูปแบบที่ไม่มีขั้นตอน ในรูปนี้มีการวาดความสำพันธ์ของระบบผู้เชี่ยวชาญที่มีต่อรูปแบบอื่นๆ รูปนี้พิจารณาเพียงแนวทางและไม่ยึดติดคำนิยามตายตัว บางส่วนของรูปแบบและภาษา มีลักษณะซึ่งสามารถแทนที่ได้มากกว่าหนึ่งระดับ ยกตัวอย่างเช่น บางคน

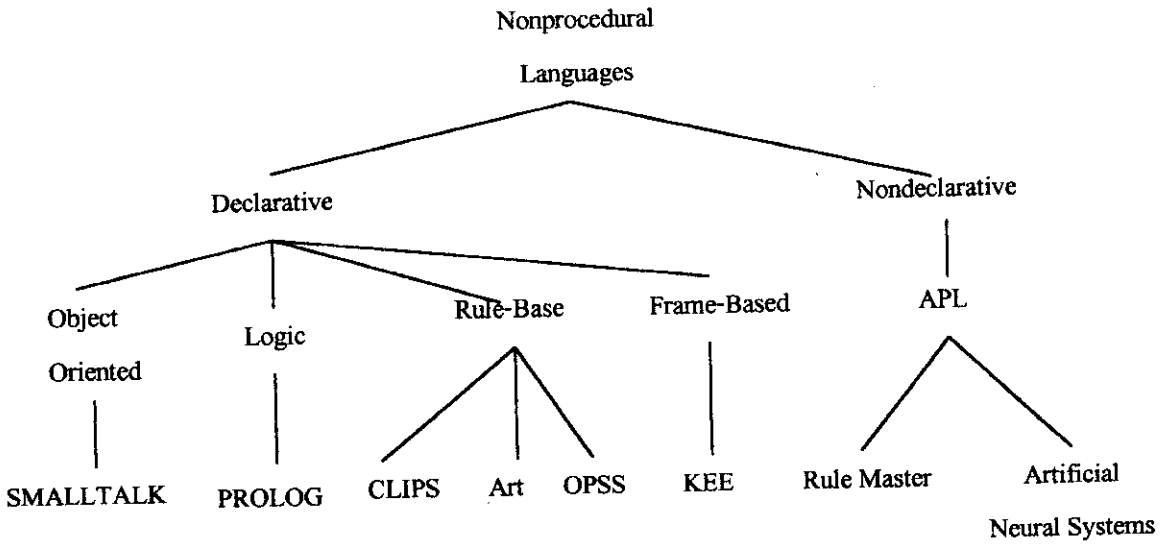
ของรูปแบบและภาษา มีลักษณะซึ่งสามารถแทนที่ได้มากกว่าหนึ่งระดับ ยกตัวอย่างเช่น บางคน

CT 488

พิจารณาหน้าที่การทำงานของโปรแกรมรูปแบบขั้นตอน ในขณะที่บางคนกล่าวถึงมันเป็นรูปแบบการอธิบาย



รูปที่ 7.8 Procedural Languages



รูปที่ 7.9 Nonprocedural Languages

คำว่า Algorithm แปลว่า มันเป็นวิธีในการแก้ปัญหาเป็นลำดับดับขั้น การทำงานแบบที่ในโปรแกรมเป็นขั้นตอนของโปรแกรม ในรูปของขั้นตอนของโปรแกรม ลำดับขั้นของโปรแกรมแบบแผนของโปรแกรมเหล่านี้มีความหมายเดียวกันในรูปของโปรแกรมที่ไม่มีระบบปัญญาประดิษฐ์ แนวคิดโดยทั่วไปของลำดับขั้นตอนโปรแกรม คือ กระบวนการที่เป็นลำดับประโยคต่อประโยค ถ้าไม่มีส่วนปลีกย่อยที่ถูกพบ

ตัวอย่างของคำนิยามของหน้าที่

Cube (x) = x* x* x where x is a real number and cube is a function with real values

สามส่วนของคำนิยามของหน้าที่ คือ

ในส่วนของการรวมกัน x* x* x

ตัวแปร , จำนวนจริง

ตัวแปรร่วม, จำนวนจริง

สัญลักษณ์ = หมายถึง เท่ากับ หรือ เปรียบเสมือน จากบันทึกดังต่อไปนี้ คือทางลัดของการเขียนแผนที่ จาก ตัวแปรของจำนวนจริง

สัญลักษณ์ คือ R ถึง ตัวแปรร่วมของจำนวนจริง

Cube : R → R

บันทึกทั่วๆ ไปของหน้าที่ F ซึ่งแผนที่จาก ตัวแปร S ถึง ตัวแปร T คือ F: S→T (Gersting82) ขอบเขตของภาพลักษณ์ F(s) S คือ องค์ประกอบของ S สำหรับกรณีของ cube function ภาพลักษณ์ของ s คือ s* s* s และขอบเขตของกลุ่มคือทุกๆจำนวนจริง ขอบเขตของตัวแปรร่วมคือสิ่งเดียวกันกับ cube function อย่างไรก็ตาม มันอาจไม่ถูกต้องนักสำหรับหน้าที่อื่นๆ เช่น square function xxx กับ ตัวแปร และตัวแปรร่วมของจำนวนจริง ตั้งแต่ขอบเขตของ square function เป็นเพียงจำนวนที่ไม่ติดลบ ของบเขต ตัวแปรร่วมไม่เหมือนกัน การใช้กลุ่มบันทึก ขอบเขตของหน้าที่สามารถเขียนได้เป็น

$R = \{f(s) / s \in s\}$

วงเล็บปีกกา {} แสดงถึงกลุ่ม เครื่องหมายที่เป็นเส้น / เป็นการบ่งบอกถึงตำแหน่งประโยคข้างบน สามารถอ่านได้ว่า ขอบเขต R เท่ากับกลุ่มของ f(s) ที่ทุกๆ องค์ประกอบ s เป็นสมาชิก S การจัดการคือการเรียงกลุ่มเป็นคู่ (s,t) ที่ $s \in S$ $t \in T$ และ $t = f(s)$ ทุกๆสมาชิกของ S ต้องมีหนึ่งตัวและมีเพียงหนึ่งองค์ประกอบของ T อย่างไรก็ตาม การที่มี tหลายๆตัวอาจมีการจัดการรวมกับ s เพียงตัวเดียว ยกตัวอย่างง่ายๆ ทุกๆจำนวนบวก n ต้องมี 2 square roots ±

หน้าที่อาจให้คำอธิบายได้อีกอย่างหนึ่ง

Factorial (n) = n*factorial(n-1)

Where n is an integer and

Factorial is an integer function.

175

Recursive function มักใช้ภาษา เช่น LISP แนวทางคณิตศาสตร์และการแสดงออกนั้น เป็นการอ้างอิงที่เปิดเผย เพราะว่า ความหมายทั้งหมด คือ การกำหนดจากส่วนของตัวมันเอง ไม่มีการส่งเสริมกันและกันเป็นสิ่งที่เกี่ยวข้องกันระหว่างส่วนดังเช่น ตัวอย่าง พิจารณาการอธิบายฟังก์ชั่น x+(2*x) อะไรถูกแทนค่าด้วย x ถึงแม้ฟังก์ชั่นอื่นสามารถแทน x และผลก็เป็นเหมือนเดิม ตัวอย่าง เช่น ให้ h(y) เป็นฟังก์ชั่น ดังนั้น h(y)+(2*h(y)) ยังคงเท่ากับ 3*h(y) ตอนนี้พิจารณาประโยคข้างล่าง ในส่วนของภาษา Pascal

Sum : = f(x)+x

ถ้าตัวแปร x คือตัวที่ผ่านไปโดย การกล่าวถึงและค่าของมันเปลี่ยนไปในฟังก์ชั่นเรียกว่า f(x) และมูลค่าของ x คืออะไร ? มันก็ขึ้นอยู่กับว่าผู้เรียบเรียงเขียนไว้อย่างไร ค่าของ x อาจเป็นค่าปกติ ถ้ามันถูกเก็บไว้บนกลุ่ม หรือ ค่าใหม่ถ้า x ไม่ถูกเก็บ แหล่งอื่นๆที่ทำให้เกิดความสับสน อาจเกิดขึ้นถ้าผู้เรียบเรียง ได้อธิบายจากขวาไปซ้าย ในขณะที่ผู้อื่นวิเคราะห์จากซ้ายไปขวา ในกรณีนี้ f(x)+x จะไม่เหมือนกับ x+f(x) ในความแตกต่างของผู้เรียบเรียง ถึงแม้ว่าใช้ภาษาเดียวกัน ผลข้างเคียงอื่นๆ อาจเกิดขึ้นเนื่องจากตัวแปรกว้างมาก ดังนั้นจึงแตกต่างจากฟังก์ชั่นคณิตศาสตร์ ฟังก์ชั่นของโปรแกรมจะไม่มีการอ้างอิงแบบเปิดเผย ฟังก์ชั่นของภาษาของโปรแกรม ถูกสร้างขึ้นเพื่ออ้างอิงแบบเปิดเผย ฟังก์ชั่นทั้ง 5 ส่วนคือ

Data object : สำหรับฟังก์ชั่นภาษาในการทำงาน

Primitive function : ทำงานบน Data object

Function forms : การสังเคราะห์ฟังก์ชั่นใหม่จากฟังก์ชั่นอื่นๆ

Application operation : ฟังก์ชั่นที่ย้อนค่ากลับ

Naming Procedures : แยกแยะฟังก์ชั่นใหม่

ฟังก์ชั่นภาษาโดยปกติแล้ว จะทำงานเป็นตัวแปลสำหรับโครงสร้างและตอบสนองแก่ผู้ใช้โดยทันที ใน LISP (List processing) Data object เป็นสัญลักษณ์ที่แสดงออก (s-expression) ซึ่งเป็นทั้ง รายการหรือหน่วยเล็กๆ ในขณะที่ APL(A programming language) เป็นลำดับ

ตัวอย่าง รายการคือ

(milk eggs cheese)

(shopping (groceries(milk eggs cheese)clothes(pants))) ()

รายการว่าง (empty list),() ไม่มีองค์ประกอบภายในซึ่งเรียกว่า ศูนย์ (nil) ต้องมีบาง Primitive function อยู่โดยภาษาเป็นพื้นฐานสำหรับการสร้างฟังก์ชั่นที่ซับซ้อน ในรุ่นดั้งเดิมของ LISP สร้างโดย John Mc Carthy ในปี 1960 มี Primitive ค่อนข้างน้อย ดังตาราง 1-12

Primitive CAR,CDR,CPR และ CTR เป็นคำย่อจากพยัญชนะตัวแรก หลังจากกำหนดลงทะเบียนอุปกรณ์ เมื่อแรกเริ่มทำงานใน LISP,CPR และ CTR เป็นตัวที่ล้ำสมัยที่สุดตอนนี้ แต่ตัวย่อของตัวอื่นๆ ยังคงเดิม เช่นเดียวกัน การนำเสนอ เป็น ตัวบ่งชี้ของ LISP ตัวบ่งชี้เป็น ฟังก์ชั่นพิเศษ ซึ่งจะแสดงผลตอบกลับในรูปของถูกและผิด

รุ่นดั่งเดิมของ LISP เรียกว่า LISP บริสุทธิ์ เพราะว่ามันเป็นฟังก์ชั่นที่บริสุทธิ์ อย่างไรก็ตาม มันก็ยังไม่มีประสิทธิผล สำหรับในการเขียนโปรแกรม Non-function ถูกเพิ่มเข้ามา เพื่อเพิ่มประสิทธิผลให้ LISP ในการเขียนโปรแกรม ตัวอย่างเช่น SET เป็นตัวมอบหมายงาน ในขณะที่ LET และ PROG สามารถใช้สร้างตัวแปรภายในและทำให้เกิดผลของ 3 –expression ถึงอย่างไรก็ตาม การกระทำเหล่านี้เหมือนฟังก์ชั่น แต่มันไม่ใช่ฟังก์ชั่นในเชิงคณิตศาสตร์

Function	*Predicates*
QUOTE	ATOM
CAR	EQ
CDT	NULL
CPR	
CTR	
CONS	
EVAL	
COND	
LAMBDA	
DEFINE	
LABEL	

ตารางที่ 7.12 Original LISP Primitives and Functions

ตั้งแต่ถูกสร้างขึ้นมา LISP เคยเป็นผู้นำในภาษา ระบบปัญญาประดิษฐ์ในสหรัฐอเมริกา โครงสร้างระบบผู้เชี่ยวชาญรุ่นดั้งเดิมนั้นถูกเขียนโดย LISP เพราะว่า มันง่ายต่อการทดสอบ

อย่างไรก็ตาม คอมพิวเตอร์ทั่วๆ ไป ไม่ใช้ LISP เพราะในเรื่องประสิทธิผลและการสร้างโดยการใช้ LISP นั้นค่อนข้างจะแย่มาก เพื่อที่จะครอบคลุมปัญหานี้ หลายบริษัทเสนออุปกรณ์ที่ออกแบบมาใช้งานกับ LISP อุปกรณ์ที่ใช้กับ LISPนี้สามารถใช้ได้อย่างสมบูรณ์ อย่างไรก็ตาม มันมีต้นทุนที่สูงพอสมควร

ปัญหาเรื่องต้นทุนสูง มีผลกระทบทั้งการพัฒนาและปัญหาการจัดส่ง การพัฒนามันยังไม่เพียงพอ ถ้าไม่สามารถจัดส่งเพื่อการใช้งานได้ เพราะว่าต้นทุนสูง การพัฒนาที่ดี ต้องพัฒนาในส่วนของความเร็ว, อำนาจ, ขนาด, น้ำหนัก และสภาพแวดล้อม การใส่รหัสลงไปใน ROM อาจทำให้เกิดปัญหากับระบบปัญญาประดิษฐ์และระบบผู้เชี่ยวชาญที่ต้องการอุปกรณ์พิเศษในการทำงาน ปัญหาอื่นๆ คือ การฝังของระบบปัญญาประดิษฐ์ในภาษาโปรแกรม เช่น C, Ada, Pascal และ FORTRAN

สิ่งหนึ่งที่ควรพิจารณาเป็นหลักในการเลือกภาษาระบบปัญญาประดิษฐ์ควรเป็นภาษาที่ถูกเขียนขึ้นโดยอุปกรณ์ ด้วยเหตุผลที่ว่า เคลื่อนย้ายง่าย,มีประสิทธิผล และรวดเร็ว

8.11 Nonprocedural Paradigms

Nonprocedural Paradigms ไม่ขึ้นอยู่กับนักจัดโปรแกรมที่ให้รายละเอียดในการแก้ไขปัญหาว่าควรแก้อย่างไร สิ่งที่ตรงข้ามของ Nonprocedural Paradigms ที่ระบุรายละเอียด "อย่างไร" ฟังก์ชั่นหรือการบรรยายรายละเอียด "อะไร" ให้ระบบทำงานให้บรรลุผล

โปรแกรมการอธิบาย (Declarative Programming)

โปรแกรมการอธิบายแบ่งจุดประสงค์จากวิธีที่จุดประสงค์เคยได้รับ ผู้ใช้อธิบายว่าจุดประสงค์ขณะที่ภายใต้โครงสร้างของอุปกรณ์ที่พยายามที่จะทำให้จุดประสงค์บรรลุตามเป้าหมาย จำนวนของโปรแกรมภาษาที่เป็นแบบอย่างและการร่วมมือ ได้สร้างขึ้นเพื่อให้ผลของตัวอย่างการอธิบายบรรลุผล

Object-oriented Programming

ตัวอย่าง Object-oriented เป็นอีกกรณีของตัวอย่างที่สามารถเป็นส่วนสำคัญต่อการพิจารณาและเป็นเพียงส่วนที่อธิบาย ในปัจจุบันส่วนของ Object-oriented เคยใช้ทางที่ต่างกัน 2 ทาง การแสดงออกแบบ Object-orientedกำลังพัฒนาเหมือนหลักการโปรแกรม ในโปรแกรมภาษาพิจารณา เช่น Ada และ Modula-2 พื้นฐานความคิดเป็นการออกแบบโปรแกรมโดยการพิจารณาข้อมูลที่ใช้ในโปรแกรมเหมือนกับเป้าหมายและการทำงานของอุปกรณ์

นี้ก็เป็นสิ่งตรงกันข้ามการออกแบบ top-down ที่ทำงานอย่างประณีตเป็นขั้นตอนของโครงสร้างการควบคุมโปรแกรม ความจริง Object-oriented เป็นสิ่งสำคัญต่อการเรียกตัวออกแบบ bottom – up แต่โชคไม่ดีที่ กลุ่มของ bottom-up ไม่เคยตอบกลับมาให้ Object-oriented เช่นการออกแบบ Object-orientedการพิจารณาการเขียนโปรแกรมการจัดการเกี่ยวกับบัญชีกับ interative mean (Riley 87)

วิธีออกแบบ Object-oriented มีความเหมาะสมดีกับโปรแกรมที่มีโครงสร้างที่ไม่สมบูรณ์ มันจะไม่เหมาะสมกับโปรแกรมที่มีโครงสร้างสมบูรณ์ เช่น บัญชีค้างจ่าย
สำหรับกรณีบัญชีค้างจ่ายสามารถอธิบายเป็นขั้นตอนได้ดังนี้
ได้รับข้อมูล time date
บัญชีสำหรับเวลาเจ็บป่วยและวันพักร้อน
เพิ่มชั่วโมงการทำงาน x rate ของการจ่าย
ทำงานนอกเวลา
เพิ่มโบนัสหรือคอมมิสชั่น ถ้าเหมาะสม
บวกค่าเสียภาษี, ค่าประกัน, หนี้สิน และการปลดเกษียณ
จ่ายเช็คที่ค้างจ่าย

การออกแบบ Object-oriented ไม่ต้องการภาษาสำคัญ เช่น FORTRAN, BASIC, C และอื่นๆ อีก
ภาษาเช่น Ada และ Modula-2 รับรอง เพราะภาษามีลักษณะสรุปและแพ็คเก็ต โปรแกรม Object-oriented เป็นต้นแบบการใช้ภาษา เช่น smalltalk ที่ออกแบบพิเศษ ตอนนี้บ่อยครั้งเคยหมายถึงโปรแกรม Object-oriented แม้ว่าภาษาจะไม่มีจุดประสงค์จริง รับรอง

Smalltalk มีลักษณะรับรองเป้าหมายในการสร้างภาษา Smalltalk ถูกถ่ายทอดจาก SIMULA 67 ภาษาพัฒนาสำหรับการลอกเลียนแบบ SIMULA67 แนะนำหลักการของคลาส คลาสเป็นประเภทหนึ่งที่ให้

หลักการสำคัญอื่นๆ มาจาก SIMULA67 เป็นสิ่งที่สืบทอดมา ใน SIMULA67 subคลาสเป็นนิยามในการสืบทอดส่วนของ คลาส เช่น คลาส อาจอธิบายส่วนประกอบของ objects ที่สามารถใช้เป็นจำนวนมากและคลาสอื่นๆ ที่มีจำนวนที่ซับซ้อน ตอนนี้ subคลาส สามารถอธิบาย objects ง่ายขึ้น

นั่นก็คือ objects ต่างๆ สืบทอดสิ่งต่างจาก คลาส ที่อยู่เหนือพวกมัน ซึ่งเรียกว่า super classes หลักการของการถ่ายทอดสามารถขยายการจัดการ objects ที่ objectsสามารถถ่ายทอดจาก คลาส ของพวกมันต่อๆ ไป การถ่ายทอดเป็นประโยชน์มากๆตั้งแต่ objects สามารถที่จะถ่ายทอดคุณสมบัติของคลาสได้

โปรแกรมตรรกะ

หนึ่งในการประยุกต์ปัญญาประดิษฐ์ของคอมพิวเตอร์อยู่บนการพัฒนาทฤษฎีตรรกะกับโปรแกรมนักทฤษฎีตรรกะของเนเวลล์และไซมอน โปรแกรมเป็นรายงานตัวแรกที่ Dartmouth Conference รายงานปัญญาประดิษฐ์ในปี 1956 และสาเหตุความรู้สึก เพราะว่าคอมพิวเตอร์อิเล็คโทรนิกส์ช่วงต้นๆ ได้ถูกใช้แต่ในการคำนวณตัวเลข

ทางด้านทฤษฎีตรรกะและความสำเร็จของมัน หนังสือมาที่เนเวลล์และไซมอนตั้งใจจะเขียนหลักการการแก้ปัญหาทั่วๆไป ขณะที่ทฤษฎีตรรกะเป็นความหมายจากการพัฒนาทฤษฎีคณิตศาสตร์ GPS ที่ถูกออกแบบให้แก้ปัญหาปัญหาตรรกะประเภทต่างๆ รวมทั้งเกมและปริศนาต่างๆ เช่น หมากรุก, Tower of Hanoi, Hissionaries and Cannibals และ cryptarithmetic ตัวอย่างของ cryptarithmetic (ความลับของคณิตศาสตร์) ปริศนาคือ

DONALD

GERALD

ROBERT

และมันก็รู้ว่า D=5 object เป็น ตัวเลขบอกค่าคณิตศาสตร์ของตัวหนังสืออื่นๆในระยะ 0-9

GPS เป็นการแก้ปัญหาโปรแกรมอันดับแรกก็เพื่อที่จะแยกข้อมูลของการแก้ปัญหาจากข้อมูลของตัวแปร การแก้ปัญหาเ็ นขั้นตอนของการแบ่งแยกข้อมูลของการแก้ปัญหาที่ชัดเจนจากข้อมูลของตัวแปร ที่ถูกใช้เป็นระบบข้อมูลพื้นฐาน

การทดลองยังมีการปรับปรุงทางด้านทฤษฎีเครื่องจักรอยู่ตลอด ต้นปี 1970 ได้ถูกค้นพบการคำนวณในกรณีพิเศษของเครื่องจักร, การอนุมานแบบตรรกะ (Kowalski 1985) เมื่อลูกโซ่ย้อนกลับได้ถูกประยุกต์ในรูปแบบของ "conclusion if conclusion" มันมีประสิทธิภาพพอสำหรับการพิสูจน์ทฤษฎีสำคัญๆ conditions สามารถเป็นความคิดของรูปแบบการเสนอไปเข้าคู่กับการสร้างหลักการที่อภิปรายกันก่อนหน้านี้ ประโยคแสดงในรูปแบบซึ่งถูกเรียกว่า Horn clauses หลังจาก Alfred Horn คนที่สำรวจพวกมัน ในปี 1972 ภาษา PROLOG ได้ถูกสร้างขึ้นโดย Kowalski, Colmerauer และ Roussel อุปกรณ์ของโปรแกรมตรรก โดยลูกโซ่ย้อนกลับใช้ Horn clauses

ลูกโซ่ย้อนกลับสามารถที่จะใช้ได้ทั้งแสดงข้อมูลในการอธิบายและยังควบคุมกระบวนการต่างๆ อีก ตัวอย่าง ลูกโซ่ย้อนกลับทำงานโดยการอธิบาย subgoals ต้องเป็นที่น่าพอใจ ถ้าจุดประสงค์แรกๆเป็นที่น่าพอใจ ถ้า subgoals มีลักษณะที่ชำรุดเป็น subgoalsที่เล็กกว่าก็จะออกจากไป

ตัวอย่างของการอธิบายข้อมูลที่เป็นพื้นฐาน

All men are mortal

Socrates is a man

สามารถแสดงใน Horn clauses

Someone is mortal

If someone is a man

Socrates is a man

If (in all cases)

จากประโยคเกี่ยวกับ Socrates ถ้าเงื่อนไข if เป็นจริงในกรณีทั้งหมด อีกทางหนึ่ง ข้อมูลเกี่ยวกับ Socrates ไม่เรียกร้องรูปแบบเพื่อให้เหมาะสมกับมัน ทางกลับกันกับกรณีที่ต้องตาย (mortal) คนคนนั้นต้องเป็นผู้ชายคนหนึ่ง สำหรับรูปแบบของของเงื่อนไข if นี้เป็นที่น่าพอใจ

หมายเหตุ Horn clauses สามารถเชื่อมต่อได้ เช่น กระบวนการที่บอกความน่าพอใจของจุดประสงค์ นั่นก็คือ การตัดสินใจถ้าคนคนนั้นต้องตาย ก็จำเป็นที่ต้องตัดสินใจถ้าคนคนนั้นก็ต้องเป็นผู้ชาย มันค่อนข้างที่จะซับซ้อนมาก

A car needs gas, oil and inflated tired to run

สามารถแสดงใน Horn clauses ได้คือ

X is a car and runs

If x has gas and

If x has oil and

If x has inflated tires

หมายเหตุ จะตัดสินใจปัญหาอย่างไร if a car will run ยังไปลด 3 subproblem หรือ 3 subgoals นี้ก็คือข้อสนับสนุนมีเงื่อนไขอธิบายข้อมูลตามตัวอย่างด้านล่าง

The fuel gauge shows not-empty if a car has gas

The dipstick shows not-empty if a car had oil

The air pressure gauge shows at least 20 if a car has

inflated tires

The fuel gauge shows not-empty

The dipstick reads empty

The air pressure gauge shows at least 15

ตัวอย่างข้างต้นสามารถแปลให้เป็น Horn clauses ได้คือ

x has gas

if the fuel gauge shows not-empty

x has oil

if the dipstick shows not-empty

x has inflated tires

if the air pressure gauge shows at least 20

Fuel gauge is not empty

if (in all cases)

Dipstick reads empty

Air pressure is at least 15

if (in all cases)

จากประโยคเหล่านี้ ทฤษฎีสามารถพิสูจน์ว่ารถคันจะไม่วิ่งเพราะว่ามันไม่มีน้ำมันและแรงดันอากาศไม่เพียงพอ

ข้อดีของระบบลูกโซ่ย้อนกลับเป็นการทำงานแบบคู่ขนาน การทำงานแบบคู่ขนานสามารถเร่งความเร็วการทำงานของโปรแกรมและอีกหนึ่งเหตุผลที่ชาวญี่ปุ่นเลือก PROLOG เป็นโปรแกรมภาษาสำหรับ Fifth Generation Computer project (Moto-Oka 1982) มีโครงการหนึ่งที่จะถูกพัฒนาไปทำให้เกิดคอมพิวเตอร์อีกรุ่นหนึ่งสำหรับปี 1990กับแบบตัวอย่างความสามารถพิเศษของมัน เครื่องจักรเหล่านี้เป็นโปรแกรมอธิบายเกี่ยวกับทางด้านเทคนิคต่างๆ แต่การเข้าไปนั้นเป็นการยาก โครงการของ Fifth Generation ได้ออกมาแสดงในปี 1981 ที่ญี่ปุ่น กับงบประมาณ 850 ล้านดอลลาร์ จุดประสงค์ 26 อันแรกของโครงการรวมถึงการแปลภาษาทั่วไป ภาษาญี่ปุ่น – ภาษาอังกฤษ และ vice-versa, การแก้ปัญหาและข้อสรุปอีกนับพันครั้งเร็วกว่าเครื่องจักรทั่วๆ ไป

สามารถที่จะเขียนข้อความเป็นตัวเขียน , เข้าใจรูปภาพ , ข้อมูลพื้นฐานขนาดใหญ่ที่ฉลาด และสามารถสนทนาในภาษาทั่วๆ ไปได้

PROLOG เป็นอะไรที่ได้มากกว่าภาษา

เป็นตัวเชื่อม หรือ inference engine

ข้อมูลพื้นฐาน (ทั้งข้อเท็จจริงและหลักการต่างๆ)

รูปแบบการจับคู่เรียกว่า unification

ระบบเครื่องกล backtracking

รุ่นของ PROLOG ที่ซับซ้อน เช่น Turbo PROLOG จาก Borland

ตัวอย่างของลูกโซ่ย้อนกลับ สนับสนุน ถ้าคุณมีเงินสดหรือบัตรเครดิต คุณสามารถที่ซื้อน้ำมันเพื่อที่จะให้รถทำงาน ถ้าคุณมีเงินสดหรือบัตรเครดิตที่สามารถจะซื้อน้ำมันนั้นก็เป็นสิ่งที่น่าพอใจ หมายเหตุที่ขาดข้อเท็จจริงที่จะพิสูจน์เป้าหมายก็เป็นแค่ผลกระทบ แม้ว่าบางทีประสิทธิภาพน้อยกว่าการค้นข้อเท็จจริง เช่น Dipstick reads empty ไม่ว่าจะค้นข้อเท็จจริงหรือขาดข้อเท็จจริงก็เป็นเป้าหมายที่ไม่น่าพอใจ

Backtracking และ รูปแบบการจับคู่ ไม่เป็นที่ต้องการของปัญหา จากนั้น โปรแกรมเมอร์ต้องทำงานเพิ่มขึ้นหรือ รหัสที่ต่างภาษาต่างกัน ข้อดีข้อหนึ่งของโปรแกรมตรรกะที่ทำงานแบบเฉพาะกิจ นั้นก็คือการเรียกโดยเฉพาะของปัญหา Horn clause ผลิตโปรแกรม อันนี้จะแตกต่างจากโปรแกรมทั่วไป ภาษาที่จำเป็น เช่น Ada

ระบบผู้เชี่ยวชาญ

ระบบผู้เชี่ยวชาญสามารถที่จะตัดสินโปรแกรมการอธิบายได้ ยกตัวอย่างหลักการพื้นฐานของระบบผู้เชี่ยวชาญ หลักการต่างๆกลายมาเป็นการกระตุ้นและใส่บน agenda ถ้า LHs ของมันจับคู่กับข้อเท็จจริง คำสั่งที่หลักการเข้าไปแล้วนั้น ไม่เกิดผลใดๆ ดังนั้น คำสั่งที่ใช้ในการกระตุ้นโปรแกรมจึงไม่ตายตัว ความแตกต่างระหว่างระบบผู้เชี่ยวชาญกับโปรแกรมทั่วไป มีดังนี้

CT 488

Characteristic	*Conventional Program*	*Expert System*
Control by ...	Statement order	Inference engine
Control and data	Implicit integration	Explicit separation
Control Strength	Strong	Weak
Solution search	Algorithm	Rules and inference
Problem solving	Small or none	Large
Input	Algorithm is correct	Rules
Unexpected input	Assumed correct	Incomplete, incorrect
Output	Difficult to deal with	Very responsive
Explanation	Always correct	Varies with problem
Applications	None	Usually
Execution	Numeric, file, and text	Symbolic reasoning
Program design	Generally sequential	Opportunistic rules
Modifiability	Structured design	Little or no structure
Expansion	Difficult	Reasonable
	Done in major jumps	Incremental

ตาราง 7.13 Some Typical Differences between Conventional Programs and Expert Systems

ระบบผู้เชี่ยวชาญถูกออกแบบให้มีความเกี่ยวข้องกับความไม่แน่นอน เหตุเพราะว่า เป็นอุปกรณ์ตัวหนึ่งที่ดีที่สุดที่เราต้องค้นหาในส่วนของ ความเกี่ยวข้องกับความไม่แน่นอน ความไม่แน่นอนอาจจะถูกป้อนข้อมูลเข้าสู่ระบบผู้เชี่ยวชาญ ข้อมูลพื้นฐานของมัน อันดับแรกอาจจะดูเหมือนสร้างความประหลาดใจกับผู้คนที่เคยใช้โปรแกรมทั่วไป อย่างไรก็ตาม ความรู้โดยส่วนใหญ่แล้ว เป็นสิ่งที่ช่วยแก้ปัญหา ซึ่งหมายถึง มันอาจจะทำงานได้อย่างอย่างถูกต้อง ณ ช่วงเวลาหนึ่งเท่านั้น การป้อนข้อมูลที่ไม่ถูกต้อง ไม่สมบูรณ์ ไม่สมประกอบและอาจมีข้อเสียอื่นๆ ขึ้นอยู่กับการป้อนข้อมูลและข้อมูลพื้นฐานระบบผู้เชี่ยวชาญ อาจจะประมวลผลได้ถูกต้อง อาจเป็นคำตอบที่ดี คำตอบที่ผิด หรือ ไม่มีคำตอบเลย ขณะที่ในช่วงแรกๆ นั้น อาจจะตกใจที่ไม่มีตัวเลือกนั้นไม่มีคำตอบให้ ในบางครั้ง สิ่งที่สำคัญที่สุด ที่ควรจะจำไว้ นั้นคือระบบผู้เชี่ยวชาญที่ดีจะไม่แสดงผลที่แย่กว่าตัวผู้แก้ปัญหา บางทีผู้เชี่ยวชาญอาจจะทำงานได้ดีกว่า ถ้าเรารู้วิธีการแก้ปัญหาที่ถูกใช้ที่ดีกว่าระบบผู้เชี่ยวชาญที่เราเคยใช้ สิ่งสำคัญคือ เลือกสิ่งที่ดีที่สุดมาใช้

Nondeclarative Program

รูปแบบการแก้ปัญหาที่เป็นขั้นตอนได้รับความนิยมเป็นอย่างมาก วิธีแนวใหม่นี้ถูกนำไปประยุกต์ใช้อย่างแพร่หลาย มันสามารถใช้ได้ทั้งตัวมันเองหรือนำไปใช้ร่วมกับตัวอื่น

Induction-based Programming

การนำระบบปัญญาประดิษฐ์มาประยุกต์เป็นสิ่งที่น่าดึงดูดและน่าสนใจคือ Induction-based Programming ในรูปแบบนี้จะเรียนรู้โดยตัวอย่าง รูปแบบหนึ่งของการประยุกต์รูปแบบนี้ถูกนำมาใช้ในเรื่องข้อ การเข้าถึงฐานข้อมูล แทนที่ผู้ใช้จะต้องทำการกำหนดค่าเพื่อที่จะทำการค้นหา มันเพียงแค่เลือกตัวอย่างสักหนึ่งตัวอย่างหรือมากกว่า ที่มีความเหมาะสมกับลักษณะเหล่านั้น โปรแกรมฐานข้อมูลได้แสดงถึงลักษณะของข้อมูลและการค้นหาที่เข้ากันได้กับฐานข้อมูล

เครื่องมือของระบบผู้เชี่ยวชาญ เช่น 1st-คลาส และ KDS เสนอวิธีการเรียนรู้แบบแตกแขนง โดยการทำความเข้าใจตัวอย่างและกรณีศึกษาและกฎต่างๆ

8.12 Artificial Neural Systems

แบบอย่างการพัฒนาแบบใหม่ในเรื่องเกี่ยวกับโปรแกรมเกิดขึ้นในปี 1980 เรียกว่า Artificial Neural System (ANS) ขึ้นอยู่กับว่าสมองจะมีกระบวนการในการประมวลผลข้อมูลอย่างไร แบบนี้บางทีเรียกว่า Connectionism เพราะว่ารูปแบบการแก้ปัญหาของมัน ทำโดยการกระตุ้นระบบเครือข่ายระบบประสาท มีนักวิจัยหลายๆคนทำการวิเคราะห์เครือข่ายโยงใยของระบบนี้ เครือข่ายโยงใยจะเชื่อมต่อทำให้เกิดประสิทธิภาพสูงด้านหน้า-หลัง ของระบบผู้เชี่ยวชาญซึ่งมี ข้อกำหนดในเรื่อง ปริมาณของวัตถุดิบจากเครื่องตรวจจับซึ่งทำให้เกิดการตอบสนอง ณ เวลานั้น

ปัญหาการเดินทางของพนักงานขาย

ANS มีจุดที่น่าสังเกตในเรื่อง การประสบความสำเร็จในเรื่องเวลาการตอบสนองที่มีต่อรูปแบบที่ซับซ้อนในการแยกแยะปัญหา ในกรณีหนึ่ง เครือข่ายโยงใยระบบประสาทบนเครื่องไมโครคอมพิเตอร์นั้นจะได้รับวิธีการแก้ปัญหาที่ดีมากในการแก้ปัญหาเรื่องของการเดินทางของพนักงานขายในเวลา 0.1 วินาที เปรียบเทียบกับทางแก้ที่เหมาะสมที่สุดซึ่งมีข้อกำหนดต้องการเวลาเป็นชั่วโมงของเวลาการทำงาน CPU บน เมนเฟรม (Port 86) ปัญหาการเดินทางของพนักงานขายเป็นสิ่งที่สำคัญเพราะว่า มันเป็นปัญหาที่คลาสสิค ที่พบได้ในรากสัญญาณในระบบโทรทัศน์

Optimizing routing เป็นสิ่งสำคัญในเรื่องการเดินที่ที่ใช้เวลาน้อยที่สุดและยังทำให้เกิดประสิทธิผลและความเร็ว

ปัญหาพื้นฐานของการเดินทางของพนักงานขายเป็นการคำนวณทางที่สั้นที่สุดผ่านทางรายชื่อเมืองต่างๆ ตาราง 7.14 แสดงถึงเส้นทางที่เป็นไปได้สำหรับเมือง 1-4 สังเกตว่าจำนวนเส้นทางเป็นสัดส่วนในรูปของ แฟคตอเรียล ของจำนวนเมืองลบหนึ่ง , (N-1)!

Number of Cities	Routes
1	1
2	1-2-1
3	1-2-3-1
	1-3-2-1
4	1-2-3-4-1
	1-2-4-3-1
	1-3-2-4-1
	1-3-4-2-1
	1-4-2-3-1
	1-4-3-2-1

ตาราง 7.14 Traveling Salesman Problem Routes

ในขณะที่มี 9! = 362880 สำหรับสิบเมือง มี29! = 8.8E30 เส้นทางที่เป็นไปได้สำหรับ 30 เมือง ปัญหาการเดินทางของพนักงานขายที่นิยมใช้เป็นตัวอย่างของ combinatorial explosion เพราะว่า จำนวนเส้นทางที่เป็นไปได้เพิ่มขึ้นอย่างรวดเร็วซึ่งไม่มีการทดสอบทางแก้สำหรับจำนวนเมืองที่แท้จริง ถ้ามันใช้เวลา 1 ชั่วโมงของเมนเฟรมของเวลาการทำงาน CPU ในการแก้ปัญหาของ 30 เมือง มันจะใช้เวลา 30 ชั่วโมงสำหรับ 31 เมืองและ 330 ชั่วโมง สำหรับ 32 เมือง

เครือข่ายโยงใยระบบประสาทยังสามารถแก้ปัญหาในกรณี 10 เมืองเร็วเทียบเท่ากับกรณี 30 เมือง ในขณะที่คอมพิวเตอร์ทั่วๆไปอาจใช้ระยะเวลาที่นานกว่า สำหรับกรณี 10 เมือง ระบบนี้จะหาเส้นทางที่ดีที่สุด 1 – 2 ทางและสำหรับกรณี 3 เมือง ระบบจะหาทางเลือกที่ดีที่สุด 10,000,000 ทาง ถึงแม้ว่าระบบนี้อาจจะไม่ได้ให้คำตอบที่เหมาะสมที่สุดแต่มันก็ให้ข้อดีในเรื่องของเวลา ในหลายๆกรณี 99.9999999999999999%เป็นคำตอบที่ถูกต้องหนึ่งในพันของหนึ่งวินาทีซึ่งดีกว่าคำตอบที่ถูกต้อง 100% ใน 30 ชั่วโมง

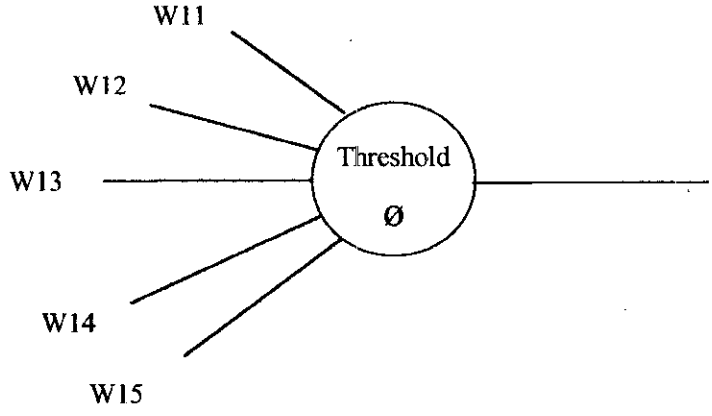
องค์ประกอบ ของ ANS

ANS เป็นระบบสัญญาณคอมพิวเตอร์พื้นฐานที่มีองค์ประกอบการใช้งานทั่วๆ ไป มีลักษณะเป็นแบบขนาน องค์ประกอบของกระบวนการการทำงานนั้นเป็นแบบหลักพีชคณิตธรรมดามาก หน้าที่หลักของ ANS คือ น้ำหนักของแต่ละองค์ประกอบ น้ำหนักเป็นตัวแสดงตัวจัดเก็บข้อมูลในระบบ

รูปแบบ Artificial neuron แสดงอยู่ในรูป 1-10 ระบบเซลล์ประสาทอาจมีหลายส่วนนำเข้าแต่จะมีผลลัพธ์ได้เพียงหนึ่งเดียว สมองของมนุษย์ประกอบไปด้วยเซลล์สมอง 1010 และ 1 เซลล์ประสาทอาจมีการเชื่อมต่อกับเซลล์ตัวอื่นๆนับพัน ส่วนนำเข้าสัญญาณไปสู่เซลล์ประสาท คูณ ด้วยน้ำหนัก และ บวกกับผลรวมของส่วนนำเข้าเซลล์ประสาททั้งหมด ซึ่งแทนด้วย I

$$I = \text{Neuron Input}_i = \sum_j W_{ij} I_j$$

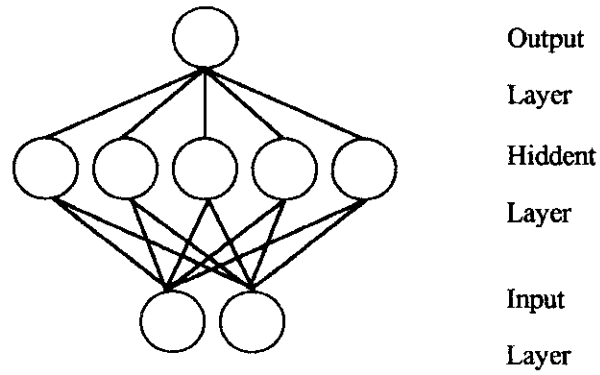
$$\text{Neuron Output} = 1/(1+e^{-(I-\emptyset)})$$



รูปที่ 7.10 A Neuron Processing Element

ผลลัพธ์เซลล์ประสาทอยู่ในรูปของ sigmoid function ของส่วนนำเข้า sigmoid แทนเซลล์ประสาทที่แท้จริงซึ่งถูกจำกัดสำหรับส่วนนำเข้าที่เล็กมากๆ หรือ ใหญ่มากๆ sigmoid เรียกว่า activation function และโดยปกติแล้วฟังก์ชั่น คือ (1+e – x) – 1 แต่ละเซลล์ประสาทยังมีความเกี่ยวข้องกับ threshold value แทนสัญลักษณ์โดย Ø ซึ่งเป็นส่วนที่ถอดออกมาจากส่วนนำเข้าทั้งหมด (I) รูป 1-11 แสดงถึง ANS ซึ่งสามารถคำนวณ exclusive-OR (XOR) ของส่วนนำเข้าโดยการใช้เทคนิคที่เรียกว่า back-propagation

XOR ให้ผลลัพธ์ที่ถูกต้องเมื่อส่วนนำเข้าไม่ถูกทั้งหมดและไม่ผิดทั้งหมด จำนวนของ node ในชั้นผิวที่ถูกซ่อนอยู่จะเปลี่ยนแปลง ขึ้นอยู่กับการประยุกต์ใช้และการออกแบบ



รูปที่ 7.11 A Back-propagation Net

มีประมาณ 13 ความรู้ที่เซลล์ประสาทได้รับการเรียนรู้เป็นลำดับ เช่น counter-propagation และ back-propagation เพื่อเป็นการฝึกฝนเครือข่ายโยงใยของเซลล์ประสาท นักโปรแกรมทำการกำหนดเครือข่ายโดยการส่งส่วนนำเข้าและผลลัพธ์ที่มีความสอดคล้องกัน เครือข่ายจะเรียนรู้ได้โดยอัตโนมัติ ทำการปรับปรุงในเรื่องของน้ำหนักในเครือข่ายซึ่งเชื่อมต่อกับเซลล์ประสาท การฝึกฝนเครือข่ายเพื่อทำให้เกิดการตอบสนองที่ถูกต้องอาจใช้เวลาเป็นชั่วโมงหรือเป็นวันๆ ขึ้นอยู่กับจำนวนรูปแบบที่เครือข่ายต้องเรียนรู้, ทั้งฮาร์ดแวร์ และ ซอฟต์แวร์ อย่างไรก็ตาม ถ้าเรียนรู้ได้แล้วครั้งหนึ่ง, ระบบการตอบสนองจะเป็นไปอย่างรวดเร็ว

ถ้าซอฟต์แวร์ที่เ็ นตัวกระตุ้น มีความเร็วไม่เพียงพอ, ANS สามารถสร้างวงจรเพื่อให้เกิดการตอบสนองที่ทันเวลา เครือข่ายที่เคยฝึกฝนแล้วกำหนดน้ำหนักมาแล้วครั้งหนึ่ง, ถึงจะสามารถสร้างวงจรได้

AT&T และ บริษัทอื่นๆ กำลังสร้างการทดลองวงจรระบบเครือข่ายเซลล์ประสาท ที่บรรจุไปด้วยเซลล์นับร้อย ในอีกไม่กี่ปีข้างหน้า วงจรอาจจะถูกสร้างเพื่อบรรจุเซลล์นับพันเลยก็ได้

ลักษณะของ ANS

สถาปัตยกรรมของ ANS นั้นแตกต่างจากสถาปัตยกรรมของเครื่องคอมพิวเตอร์ทั่วๆ ไปเป็นอย่างมาก ในคอมพิวเตอร์ทั่วๆไปนั้น มีความเป็นไปได้ในเรื่องของการแยกความความสัมพันธ์ของข้อมูลกับเซลล์ความจำ ตัวอย่างเช่น จำนวนของสังคมสงเคราะห์ สามารถจัดเก็บเป็น รหัส

ASCII ในกลุ่มเซลล์ความจำที่ติดกัน โดยการยกตัวอย่างเนื้อหาของกลุ่มที่ติดกันนี้ จำนวนสังคมสงเคราะห์สามารถทำการฟื้นฟูใหม่ได้โดยตรง การฟื้นฟูนี้มีความเป็นไปได้เพรามีความสัมพันธ์แบบ 1ต่อ1ระหว่างลักษณะของจำนวนสังคมสงเคราะห์และเซลล์ความจำซึ่งบรรจุรหัส ASCII ของลักษณะตัวนั้นๆ

ANS เป็นรูปแบบหลังจากข้อมูลถูกนำเสนอโดยน้ำหนัก อย่างไรก็ตาม ไม่มีความสัมพันธ์โดยตรงระหว่างน้ำหนักที่กำหนดและสิ่งของข้อมูลที่กำหนดเก็บ การนำเสนอของข้อมูลมีความคล้ายคลึงกับภาพสามมิติ ซึ่งเส้นทุกเส้นที่ประกอบขึ้นเป็นภาพสามมิติทำหน้าที่เป็น เหมือนซี่ถูกกรงที่สร้างขึ้นเพื่อจัดเก็บเมื่อแสงเลเซอร์ตัดผ่าน

เครื่องข่ายโยงใยเซลล์ประสาทเป็นตัวเลือกที่ดีเมื่อมีข้อมูลที่เป็นประสบการณ์มาก่อนซึ่งทำเกิดความแม่นยำและรวดเร็ว ANS มีข้อดีหลายอย่างเปรียบเทียบกับระบบจัดเก็บของคอมพิวเตอร์ทั่วๆ ไป

การจัดเก็บยกเลิกข้อที่ผิดพลาด ในส่วนของเครือข่ายสามารถเคลื่อนย้ายและมีเพียงการลดคุณภาพของการจัดเก็บข้อมูล จะเกิดขึ้นเพราะว่าข้อมูลได้มีการกระจายออกไป

คุณภาพของการจัดเก็บลดลงในสัดส่วนของปริมาณที่เครือข่ายเคลื่อนย้าย

ข้อมูลดิบถูกจัดเก็บอยู่ในรูปแบบของ associative memory เป็นหนึ่งในในส่วนของข้อมูลที่มีความเหมาะสมแก่การนึกถึงในส่วนที่ข้อมูลถูกจัดเก็บไว้แล้ว ส่วนนำเข้าที่มีปัญหาก็ยังถูกพบเห็นได้ในข้อมูลที่สมบูรณ์แล้ว

เครือข่ายสามารถคาดการณ์และแก้ไข จากข้อมูลที่จัดเก็บได้

เครือข่ายมีคุณสมบัติที่สามารถปรับเปลี่ยนได้

เหล่านี้เป็นลักษณะที่ทำให้เกิด ANS เป็นที่ดึงดูดมากสำหรับหุ่นยนต์บนอวกาศ, อุปกรณ์เติมน้ำมัน, อุปกรณ์ใต้น้ำ, กระบวนการควบคุมและอื่นๆ ที่ซึ่งจำเป็นต่อสิ่งที่ต้องใช้เป็นระยะเวลานานๆแล้วขาดการบำรุงรักษา ANS มีต้นทุนในเรื่องค่าบำรุงรักษาที่ต่ำเพราะมันสามารถปรับเปลี่ยนได้

การพัฒนาในเรื่องเทคโนโลยี ANS

จุดกำเนิดของ ANS เริ่มต้นขึ้นกับรูปแบบการคำนวณของเซลล์ประสาทโดย McCulloch และ Pitts ในปี 1943 คำอธิบายของเรื่องการเรียนรู้โดยเซลล์ประสาทถูกให้คำจำกัดความไว้โดย Hebb ในปี 1949 ในเรื่องการเรียนรู้แบบ Hebb นั้น ประสิทธิผลของเซลล์ประสาทในเรื่องการกระตุ้นเซลล์ประสาทอื่นๆ เพิ่มขึ้นกับ firing ในส่วนของ firing หมายถึง เซลล์ประสาทห่อยๆ

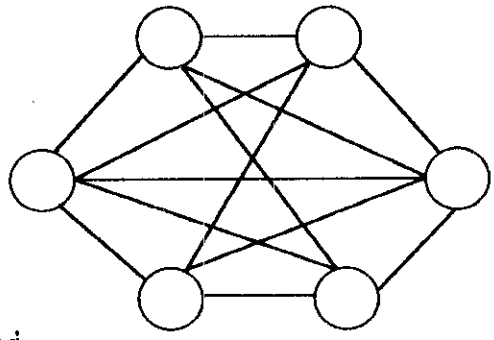
ปล่อยกระแสไฟฟ้าเพื่อกระตุ้นเซลล์ประสาทอื่นๆที่เชื่อมต่อกัน มีหลักฐานในเรื่องของการเชื่อมต่อกันระหว่างเซลล?ประสาทเรียกว่า synapses ใน ANS น้ำหนักของการเชื่อมต่อระหว่างเซลล์ประสาทเปลี่ยนแปลงโดยการกระตุ้นให้เกิดการเปลี่ยนแปลงในเซลล์ประสาท

ในปี 1961 Rosenblatt ได้ประกาศหนังสือที่มีอิทธิพลที่เกี่ยวกับระบบเซลล์ประสาทแบบใหม่ เรียกว่า perceptron หมายถึงอุปกรณ์พิเศษซึ่งแสดงถึงประสิทธิภาพสำหรับการเรียนรู้และการแยกแยะรูปแบบ ซึ่งประกอบไปด้วยชั้นผิวเซลล์ประสาท 2 ชั้น นักวิจัยได้เริ่มเข้ามาวิจัยในเรื่องของ ANS ในปี 1960

ในยุคแรกๆ ของ perceptron เริ่มหมดยุคช่วงปลายปี 1969 เมื่อ Minsky และ Papert พิมพ์หนังสือ เรียกว่า Perceptron ซึ่งแสดงถึงข้อจำกัดของทฤษฎีนี้ว่าเป็นเหมือนเครื่องคำนวณทั่วๆ ไป มันคำนวณค่าทางตรรกศาสตร์ได้ 14 ถึง 16 แบบ ซึ่งหมายถึงว่า Perceptron ไม่ใช่อุปกรณ์ที่ใช้ในการคำนวณทั่วๆ ไป ในกรณีนี้ พวกเขาได้ทำการพิสูจน์แล้วว่า Percetron ไม่สามารถแยกแยะ exclusive-OR ได้ อย่างไรก็ตาม พวกเขาก็ไม่ได้จริงจังกับการสืบค้นชั้นผิวของ ANS มากนักแต่พวกเขาทัศนคติที่ไม่ดีเพราะว่าไม่สามารถแก้ไขปัญหาของ XOR ได้ งบประมาณของรัฐบาลที่ใช้ในการทำวิจัย ANS หยุดการใช้สัญลักษณ์ใกล้เคียงกับระบบบั ญญาประดิษฐ์โดยการใช้ภาษา เช่น LISP และ ขั้นตอนในการแก้ปัญหา วิธีใหม่ของการนำเสนอสัญลักษณ์ข้อมูลระบบปัญญาประดิษฐ์โดยขอบเขตที่คิดค้นโดย Minsky และกลายเป็นตัวที่นิยมในปี 1970

ยังมีการทำวิจัยในเรื่อง ANS ในสัดส่วนเล็กๆ ในปี 1970 อย่างไรก็ตาม ก็มีงานที่น่าสนใจเกิดขึ้นซึ่งก็คืองานของ Hopfield ในปี 1982 เขานำ ANS มาใช้เป็นทฤษฎีพื้นฐานในบริษัท ซึ่งสามารถแก้ปัญหาได้หลากหลาย โครงสร้างทั่วไปของเครือข่าย Hopfield อยู่ในรูปที่ 7.12 เขาได้แสดงถึงว่า ANS สามารถแก้ไขปัญหาการเดินทางของพนักงานขายได้อย่างไรในเวลาคงที่ เป็นการเปรียบเทียบกับ combinatorial explosion โดยลำดับขั้นตอนในการแก้ปัญหา รูปแบบวงจรไฟฟ้าของ ANS สามารถแก้ไขปัญหาการเดินทางของ พนักงานขายใน 1μ วินาที

ANS สามารถแก้ปัญหา XOR ในเครือข่าย back-propagation เช่นเดียวกับที่รู้จักเป็นกฎ generalize data เครือข่าย back-propagation ปกติแล้วจะทำเป็นแบบ 3ชั้นเครือข่าย ส่วนนำเข้าระหว่างชั้นและผลลัพธ์ เรียกว่า hidden layer เพราะว่ามีเพียงส่วนนำเข้าและผลลัพธ์ของชั้นผิวที่สามารถมองเห็นได้ ทฤษฎีสำคัญที่ได้ผลมาจากคณิตศาสตร์ คือ Kolmogorov Theorem สามารถแปลงเครือข่าย 3 ชั้น กับส่วนนำเข้า n และ 2n+1 เซลล์ประสาทใน hidden layer สามารถร่างรูปแบบหน้าที่ต่อเนื่องได้



รูปที่ 7.12 A Hopfield Artificial Neural Net

การประยุกต์เทคโนโลยีของ ANS

ตัวอย่างสำคัญการเรียนรู้โดย Back-propagation ที่แสดงออกโดยเครือข่ายเซลล์ประสาทที่เรียนรู้การออกเสียงที่ถูกต้องของคำจากภาษา ANS ถูกนำมาฝึกโดยถูกนำมาใช้โดย Digital Equipment Corp. เป็นอุปกรณ์แปลงคำพูดเป็นคำบรรยายเรียกว่า DECTalk
ANS ได้สอนตัวมันเองในทักษะการออกเสียงที่เทียบเท่ากันชั่วข้ามคืนโดยการฟังการออกเสียงแบบง่ายๆจากคำพูด โดยที่ไม่มีโปรแกรมเกี่ยวกับภาษาใน ANS เลย

จากการสำรวจ ANS อยู่ภายใต้ขอบเขต โดยอุปกรณ์ไฟฟ้าและคอมพิวเตอร์แบบออฟติค ระบบเครือข่ายเซลล์ประสาทนำมาใช้ในแนวทางใหม่เป็นส่วนประกอบทางสายตาโดยที่มีความรวดเร็วที่สูงมากกว่าอุปกรณ์ไฟฟ้า อุปกรณ์เกี่ยวกับสายตาของ ANS เป็นที่ดึงดูดเนื่องมาจากธรรมชาติที่คล้ายคลึงกันของแสง ซึ่งก็คือ แสงไม่ใช่ตัวขัดขวางการเดินทางของตัวอื่นๆ ภาพถ่ายจำนวนมหาศาลสามารถสร้างขึ้นได้ง่ายโดยอุปกรณ์เกี่ยวกับสายตาเช่น กระจก, เลนส์, ระยะทางของแสง ซึ่งสามารถเป็นหน้าที่สายตาของสายตาเซลล์ประสาท คอมพิวเตอร์เกี่ยวกับสายตาถูกออกแบบมาเฉกเช่น ANS เพื่อให้เกิดผลต่างกับอันอื่นๆ

การพัฒนาเชิงพาณิชย์ใน ANS

จำนวนบริษัทใหม่ๆและที่มีอยู่เดิมนั้นได้มีการพัฒนา ANS และตัวสินค้า Nestro markets มีสินค้าประเภท ANS เรียกว่า NestroWriter ซึ่งสามารถแยกแยะ ลายมือส่วนนำเข้าและเปลี่ยนเป็นคำพูดโดยการใช้คอมพิวเตอร์ส่วนบุคคล บริษัทอื่นๆ เช่น TRW, SAIC, HNC, Synaptics, Neural Tech, Revelation Research and Texas Instrument ที่มีความหลากหลายใน ANS ที่เป็นตัวกระตุ้นและตัวเร่งเพื่อให้เกิดการเรียนรู้ที่รวดเร็ว

8.13 ตัวเชื่อมต่อระบบผู้เชี่ยวชาญและการพิสูจน์การเรียนรู้

มีความเป็นไปได้ที่จะสร้างระบบผู้เชี่ยวชาญในการใช้กับ ANS ในหนึ่งระบบ ANS เป็นข้อมูลพื้นฐานจากตัวอย่างการฝึก เช่น ยารักษาโรค (Gallant88) ในระบบผู้เชี่ยวชาญพยายามที่จะแบ่งประเภทของโรคจากอาการต่างๆ Inference engine ที่เรียกว่า MACIE (Matrix Controlled Inference Engine) ที่ถูกออกแบบที่ใช้กับ ข้อมูลพื้นฐาน ANS ระบบใช้การส่งต่อเป็นแบบลูกโซ่เพื่อที่จะสร้างข้อสรุปและลูกโซ่ย้อนกลับเพื่อที่จะตั้งคำถามกับผู้ใช้สำหรับข้อมูลเงื่อนไขต่างที่ต้องการจะผลิตการแก้ปัญหา แม้ว่า ANS มันจะไม่สามารถด้วยตัวมันเองได้ว่าทำไมน้ำหนักของมันถูกจัดเป็นค่าที่แน่นอน MACIE สามารถที่จะเชื่อมต่อกับ ANS และ ทำให้เกิดหลักการ IF... THEN... เพื่อที่จะอธิบายข้อมูลต่างของมัน

ระบบผู้เชี่ยวชาญ ANS เช่น ที่ใช้ inductive learning นั่นก็คือ ระบบเหนี่ยวนำข้อมูลจากตัวอย่าง การเหนี่ยวนำเป็นกระบวนของการสรุปกรณีทั่วๆไปจาก ข้อพิเศษๆ นอกจาก ANS มีตัวเลขเกี่ยวกับการเหมาะกับระบบผู้เชี่ยวชาญเซลล์ที่มีความชัดเจนทำให้เกิดหลักจากตัวอย่าง จุดประสงค์ของเรียนรู้ของการเหนี่ยวนำการลดหรือการกำจัดข้อมูลคอขวด การจำกัดขอบเขตของความรู้บนระบบผู้เชี่ยวชาญ เวลาการพัฒนาอาจจะถูกลดและความน่าเชื่อถืออาจเป็นการเพิ่มขึ้นถ้าระบบเหนี่ยวนำหลักการที่ไม่ใช่ความรู้โดยมนุษย์

บทสรุป

ในบทนี้เรามีการทบทวนปัญหาและมีการพัฒนาระบบผู้เชี่ยวชาญ ปัญหาเหล่านี้ที่ระบบผู้เชี่ยวชาญถูกใช้สำหรับเสิ่งทั่วๆไปไม่ใช่สำหรับการแก้ปัญหาโดยโปรแกรมทั่วไปเพราะว่า พวกมันขาดความรู้หรือ ขั้นตอนการแก้ปัญหาที่มีประสิทธิภาพ ระบบผู้เชี่ยวชาญเป็นข้อมูล พื้นฐานระบบผู้เชี่ยวชาญสามารถใช้ได้อย่างมีประสิทธิภาพสำหรับปัญหาของโลกแห่งความจริงที่โครงสร้างมีจุดบกพร่องและโดยความหมายอื่นๆยากต่อการแก้ปัญหา

ข้อดีและข้อเสียเปรียบของระบบผู้เชี่ยวชาญที่ถูกอภิปรายในทางเลือกการแก้ปัญหาตัวแปรที่เหมาะสมสำหรับการประยุกต์ระบบผู้เชี่ยวชาญได้ให้เกณฑ์การเลือกการประยุกต์ที่เหมาะสม

ความสำคัญของระบบผู้เชี่ยวชาญเซลล์ที่ถูกอภิปรายกับการอ้างอิงหลักการพื้นฐานของระบบผู้เชี่ยวชาญ วงจร Inference engine พื้นฐานถูกอธิบายและมีตัวอย่างอธิบายจากตัวอย่างหลักการง่ายๆ ท้ายนี้ความสัมพันธ์ของระบบผู้เชี่ยวชาญถึงขั้นตอนการแก้ปัญหาถูกอธิบายในส่วนของตัวแปรที่เหมาะสมของแต่ละขั้นตอนการแก้ปัญหา จุดสำคัญของทั้งนี้เป็นหลักการที่ระบบ

ผู้เชี่ยวชายสามารถที่จะตรวจสอบอุปกรณ์ทั่วๆไปที่เหมาะสมในการประยุกต์และไม่เหมาะสมกับระบบอื่น บทสุดท้ายจะอธิบายลักษณะและความเหมาะสมของระบบผู้เชี่ยวชาญในข้อมูลที่มากกว่านี้