บทที่ 2
ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

2.1 ระบบผู้เชี่ยวชาญ

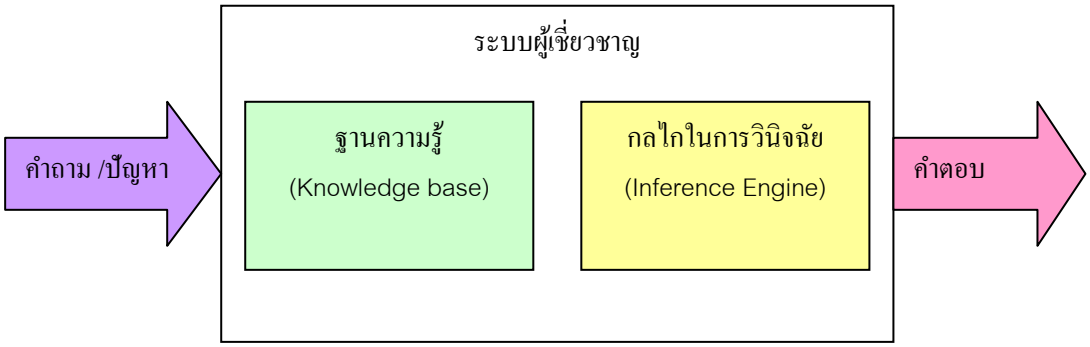
เป็นโปรแกรมคอมพิวเตอร์ที่ถูกออกแบบมาเพื่อทำหน้าที่เหมือนเป็นผู้เชี่ยวชาญในการแก้ปัญหาในขอบเขตนั้นๆ ซึ่งโปรแกรมจะใช้ความรู้ในขอบเขตที่ถูกโค็ดเข้าไปในนั้นและระบุแผนการควบคุมความรู้นั้นเพื่อให้ได้มาซึ่งคำตอบหรือวิธีการแก้ปัญหา โดยระบบผู้เชี่ยวชาญส่วนมากมักถูกเรียกว่าระบบฐานความรู้ด้วยเหมือนกัน ระบบผู้เชี่ยวชาญนั้นจะไม่เรียกว่าเป็นโปรแกรม แต่เป็นระบบเพราะมันรายล้อมไปด้วยส่วนประกอบหลายส่วน อย่างเช่น ฐานความรู้ กลไกการวินิจฉัย ส่วนอธิบาย เป็นต้น ซึ่งส่วนประกอบเหล่านี้จะมีผลต่อกันและกันในการเลียนแบบกระบวนการแก้ปัญหาโดยที่ได้รับการยอมรับจากผู้เชี่ยวชาญในขอบเขตนั้นๆ[8]

ระบบผู้เชี่ยวชาญจะจำลองกระบวนการตัดสินใจโดยใช้ข้อเท็จจริงและความรู้ ซึ่งต้องการมากกว่าความรู้เดียวในกระบวนการตัดสินใจ และส่วนประกอบหลักของระบบผู้เชี่ยวชาญประกอบไปด้วยสองส่วนคือ [9]

ฐานความรู้ - ซึ่งเป็นกลุ่มของความรู้ที่ต้องใช้ในการแก้ปัญหา และ

กลไกการ*วินิจฉัย* - ซึ่งใช้ตรวจสอบข้อเท็จจริงที่มีอยู่ เลือกแหล่งความรู้ที่เหมาะสามจากฐานความรู้ จับคู่ข้อเท็จจริงกับความรู้ และก่อให้เกิดข้อเท็จจริงเพิ่มเติม

แต่เนื่องจากปัญหามีเพิ่มขึ้นเรื่อยๆ และข้อเท็จจริงจำนวนมากได้มาจากข้อเท็จจริงที่มีอยู่แล้ว ดังนั้นในแอพพลิเคชั่นการตัดสินใจหรือข้อเท็จจริงจะเกิดขึ้นต่อๆกันไปเป็นลำดับโดยใช้ข้อเท็จจริงที่มีอยู่ ซึ่งกระบวนการนี้จะใช้ข้อเท็จจริงปัจุบันจะความรู้ที่อยู่ในฐานความรู้เพื่อก่อให้เกิดข้อเท็จจริงเพิ่มเติมต่อไปเรื่อยๆเป็นห่วงโซ่ จนกว่าข้อเท็จจริงที่เป็นเป้าหมายจะถูกค้นพบ กลไกนี้ถูกเรียกว่าการวินิจฉัย ซึ่งมีได้หลายแนวทางในการที่ความรู้ในรูปแบบของกฎจะถูกทำการวินิจฉัย ดังนั้นกลไกที่ใช้ควบคุมสามารถประกอบไปด้วยแนวทางการวินิจฉัยได้หลากหลายแนวทาง ดังนั้น ฐานความรู้ และ กลไกการวินิจฉัย จึงเป็นส่วนประกอบหลักของระบบฐานความรู้ [10] และส่วนประกอบหลักของระบบผู้เชี่ยวชาญดังแสดงในรูปที่ 2.1



รูปที่ 2.1 ส่วนประกอบหลักของระบบผู้เชี่ยวชาญ

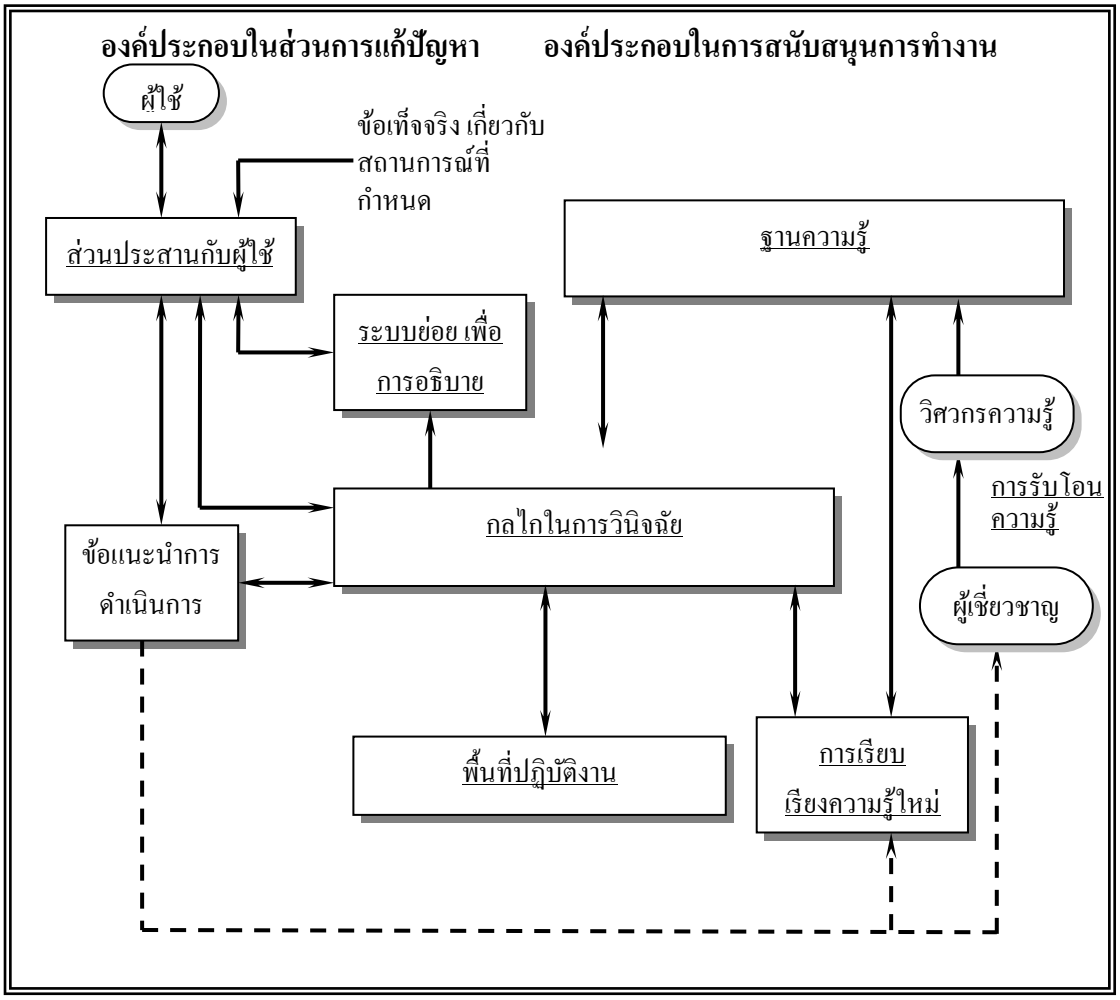
2.1.1 สถาปัตยกรรมของระบบผู้เชี่ยวชาญ

จากที่ได้อธิบายในห้อข้อข้างต้น ฐานความรู้และกลไกการวินิจฉัยจะประกอบไปด้วยกลไกหลายกลไก เมื่อระบบผู้เชี่ยวชาญเริ่มกระบวนการวินิจฉัย มันจะถูกร้องขอให้จัดเก็บข้อเท็จจริงเพื่อใช้งานในอนาคต ซึ่งเซตของข้อเท็จจริงที่ถูกตั้งขึ้นมาจะแทนเนื้องาน(context) เช่น สถานะปัจจุบันของปัญหาที่กำลังถูกแก้ไข ดังนั้นส่วนประกอบนี้ส่วนมากจะถูกเรียกว่า เนื้องาน (context) หรือหน่วยความจำปฏิบัติงาน (working memory)

เมื่อไหร่ก็ตามที่ผู้เชี่ยวชาญใช้การตัดสินใจ เราก็อยากจะรู้ว่าผู้เชี่ยวชาญตัดสินใจได้อย่างไร นอกจากนี้เมื่อผู้เชี่ยวชาญถามถึงข้อมูล เราก็อยากจะรู้ว่าทำไมข้อมูลนั้นจึงถูกร้องขอ ผู้เชี่ยวชาญจะใช้ความรู้ที่พวกเขามีและเนื้อหาของปัญหาในการตอบคำถามอย่างเช่นว่า ทำไมถึงตัดสินใจอย่างนั้น? หรือว่าทำไมต้องการข้อมูลนี้? ซึ่งส่วนนี้ก็เป็นส่วนประกอบภายในของระบบผู้เชี่ยวชาญเหมือนกัน

กระบวนการในการรวบรวม บริหารจัดการ และคอมไพล์ความรู้ พร้อมทั้งสร้างมันในรูปแบบของฐานความรู้นั้นเป็นงานที่ต้องใช้ความพยายาม เพราะมันไม่ได้จบลงในส่วนของการสร้างระบบ ฐานความรู้จะต้องถูกอัพเดต หรือไม่ก็ถูกเพิ่มเข้าไป ขึ้นอยู่กับการเพิ่มขึ้นของความรู้ในขอบเขตของความรู้นั้นๆ เครื่องอำนวยความสะดวกในส่วนการรับโอนความรู้ ซึ่งจะทำหน้าที่เหมือนเป็นส่วนประสานงานระหว่างผู้เชี่ยวชาญหรือวิศวกรความรู้กับฐานความรู้นั้นสามารถเป็นองค์ประกอบภายในของระบบผู้เชี่ยวชาญได้

ผู้ใช้ในระบบผู้เชี่ยวชาญต้องติดต่อกับระบบในการป้อนข้อมูล กำหนดข้อเท็จจริง และสังเกตสถานะของการแก้ปัญหา ในส่วนติดต่อกันผู้ใช้จะเป็นส่วนที่เป็นรูปแบบและข้อความที่แสดงให้ผู้ใช้ทำการติดต่อกับระบบได้ รูปที่ 2.2 แสดงสถาปัตยกรรมของระบบฐานความรู้และองค์ประกอบในฐานความรู้ รวมทั้งวิธีการที่แต่ละองค์ประกอบจะติดต่อซึ่งกันและกัน



รูปที่ 2.2 สถาปัตยกรรมของระบบฐานความรู้

ดังที่ได้กล่าวมาแล้ว ฐานความรู้และกลไกการวินิจฉัยจะเป็นส่วนที่สำคัญที่สุดของระบบฐานความรู้ ดังนั้นในหัวข้อถัดไปจะอธิบายรายละเอียดเกี่ยวกับเทคโนโลยีทางด้านฐานความรู้และกลไกการวินิจฉัย

2.1.2 ฐานความรู้ (Knowledge base)

ฐานความรู้จะมีความรู้ในขอบเขตของปัญหาที่เฉพาะเจาะจงเพื่อการแก้ปัญหานั้นๆ ฐานความรู้ถูกสร้างขึ้นโดยวิศวกรความรู้ที่ได้มาจากการสอบถามผู้เชี่ยวชาญและทำการเรียบเรียงความรู้ในรูปแบบที่ระบบสามารถนำไปใช้งานได้ วิศวกรความรู้ต้องมีเทคโนโลยีในการแทนความรู้และควรรู้ว่าจะพัฒนาระบบผู้เชี่ยวชาญอย่างไร ซึ่งก่อนที่จะตัดสินใจว่าโครงสร้างของฐานความรู้จะเป็นแบบไหน วิศวกรความรู้ควรจะเลือกได้ว่าน่าจะใช้การแทนความรู้แบบใดและความเหมาะสมในการเลือกใช้เทคนิคนั้นๆภายใต้สถานการณ์ต่างๆ

ความรู้ในการแก้ปัญหานั้นสามารถจำแนกได้เป็น 3 กลุ่มหลักๆ คือ ความรู้ที่ผ่านการคอมไพล์แล้ว (compiled knowledge) ความรู้เชิงคุณภาพ (qualitative knowledge) และความรู้เชิงปริมาณ (quantitative knowledge) ซึ่งความรู้แบบคอมไพล์จะเป็นได้ทั้งผลจากประสบการณ์ของผู้เชี่ยวชาญในขอบเขตนั้นๆ ความรู้ที่ได้จากคู่มือ ข้อมูลเก่า รายการมาตรฐาน และอื่นๆ ส่วนความรู้เชิงคุณภาพจะประกอบไปด้วยกฎตายตัว ทฤษฎีคร่าวๆ โมเดลของกระบวนการอย่างมีเหตุผล และสันชาตญาณ ส่วนความรู้เชิงปริมาณจะจัดการกับเทคนิคบนพื้นฐานของทฤษฎีทางคณิตศาสตร์หรือเทคนิคในทางจำนวนเป็นต้น ซึ่งความรู้เชิงคุณภาพและความรู้ที่ผ่านการคอมไพล์แล้วสามารถถูกจำแนกได้อีกเป็น 2 กลุ่ม คือ ความรู้เชิงพรรณา (declarative knowledge) และความรู้เชิงกระบวนการ(procedural knowledge) ความรู้เชิงพรรณนาจะเกี่ยวข้องกับความรู้บนคุณสมบัติทางกายภาพของขอบเขตของปัญหา ในขณะที่ความรู้เชิงกระบวนการจะเกี่ยวข้องกับเทคนิคการแก้ปัญหา

ในการพัฒนาระบบผู้เชี่ยวชาญจะเกี่ยวข้องกับความรู้ในรูปแบบเหล่านี้ เพราะมันเป็นขอบเขตของแอพพลิเคชันที่ตัดสินธรรมชาติของความรู้ที่อยู่ภายในนั้น ยกตัวอย่างเช่น พิจารณาเกี่ยวกับการวินิจฉัยสภาพอันตรายในการก่อสร้าง ซึ่งจะต้องหาว่าผนังมีความร้าวอย่างไร เพื่อหาสามารถที่ทำให้ผนังร้าว ระบบต้องมีความรู้เกี่ยวกับตำแหน่งของผนัง ชนิดของผนัง ความหนาของผนัง อัตราส่วนที่ใช้ในการผสมปูน ชนิดของสิ่งก่อสร้าง เป็นต้น ในส่วนของชนิดของผนังนั้นความรู้เกี่ยวกับฐานราก ชนิดของดิน รายละเอียดของแบบคานและเสา เป็นต้นจะถูกร้องขอ และเพิ่มเติมจากข้างต้น ความรู้เกี่ยวกับการใช้ข้อมูลที่มีอยู่ในการหาเหตุผลของการร้าวที่ผนังก็จะถูกร้องขอด้วย ซึ่งส่วนแรกของความรู้ (ซึ่งก็คือความรู้เชิงพรรณา) ที่ได้กล่าวถึงข้างต้นจะอธิบายเหตุการณ์ ในขณะที่ความรู้ส่วนที่สอง (ความรู้เชิงกระบวนการ) จะอธิบายถึงการใช้ความรู้ในเพื่อให้ได้มาซึ่งการตัดสินใจ ดังนั้นในการพัฒนาระบบฐานความรู้ จะต้องรู้ถึงความแตกต่างระหว่างการแทนความรู้แต่ละอย่าง และผลกระทบระหว่างมัน

ในการพัฒนาระบบผู้เชี่ยวชาญนั้นจะเกี่ยวข้องกับงานอย่างเช่น การรับโอนความรู้จากผู้เชี่ยวชาญในขอบเขตงานนั้นๆ การจัดการเอกสารและการรวบรวมความรู้ การผลิตเครือข่ายความรู้ (knowledge net) เพื่อตรวจสอบความสัมพันธ์ระหว่างแหล่งความรู้ที่แตกต่างกัน การตรวจสอบความถูกต้องในความรู้ และสุดท้ายคือการแปลงเครือข่ายความรู้ไปเป็นโปรแกรมโดยใช้เครื่องมือที่เหมาะสม ซึ่งโปรแกรมที่ได้มานั้นจะถูกเรียกว่าระบบผู้เชี่ยวชาญ ซึ่งเป็นระบบอย่างเป็นทางการสำหรับการจัดเก็บข้อเท็จจริงและความสัมพันธ์ระหว่างมันและวิธีการในการใช้มัน โดยทั่วๆไประบบผู้เชี่ยวชาญจะมีความรู้เกี่ยวกับวัตถุ ความสัมพันธ์ระหว่างมัน เหตุการณ์ ความสัมพันธ์ระหว่างเหตุการณ์ และความสัมพันธ์ระหว่างวัตถุกับเหตุการณ์ นอกจากนี้ชนิดของกลไกการค้นหาอาจต้องถูกร้องขอเพื่อใช้ในการขับเคลื่อนระบบ ซึ่งการการตัดสินใจเลือกการแทนความรู้นั้นจะขึ้นอยู่กับชนิดของแอพพลิเคชันที่จะสร้าง และวิศวกรความรู้ต้องตัดสินว่าส่วนไหน

ของความรู้ควรถูกแทนในรูปแบบใดได้นั้นขึ้นอยู่กับลักษณะของความรู้และประสิทธิภาพในการใช้มัน

ฐานความรู้และกลไกการวินิจฉัยจะเป็นส่วนที่สำคัญที่สุดของระบบฐานความรู้ ดังนั้นในหัวข้อถัดไปจะอธิบายรายละเอียดเกี่ยวกับเทคโนโลยีทางด้านฐานความรู้และกลไกการวินิจฉัย

2.2 การแทนความรู้ (Knowledge Representation)

ในเวลาที่ผ่านมา ได้มีการศึกษาวิจัย และกำหนดรูปแบบการสร้างตัวแทนความรู้ไว้หลายรูปแบบ แต่เราสามารถจัดเป็นกลุ่มใหญ่ ที่ได้รับความนิยมได้ 3 รูปแบบได้แก่ แบบใช้กฎ (Production Rule) แบบเครือข่ายความหมาย (Semantic Network) และแบบใช้เฟรม (Frame) ซึ่งในวิทยานิพนธ์เล่มนี้ จะขอกล่าวถึงเฉพาะแบบใช้กฎ และแบบใช้เฟรมเท่านั้น เนื่องจากในการสร้างตัวแทนความรู้แบบใช้กฎนั้นเป็นวิธีพื้นฐานที่นิยมใช้ในการพัฒนาระบบผู้เชี่ยวชาญมากที่สุด และการสร้างตัวแทนความรู้แบบใช้เฟรม เป็นรูปแบบที่วิทยานิพนธ์เล่มนี้จะนำมาเป็นแนวทางในการพัฒนาระบบผู้เชี่ยวชาญเชิงวัตถุสัมพันธ์

2.2.1 การแทนความรู้แบบกฎ (Production Rules) เป็นรูปแบบที่ไม่ซับซ้อนแต่ทรงพลังสำหรับการแทนความรู้ ซึ่งมันจะมีความยืดหยุ่นในการผสมผสานความรู้เชิงพรรณากับความรู้เชิงกระบวนการ ซึ่งในการแทนความรู้แบบกฎนั้นจะประกอบไปด้วยเซตของเหตุ (antecedents) และเซตของผล (consequents) ซึ่งเหตุจะระบุเงื่อนไขหรือเหตุการณ์ และผลเป็นเซตของการกระทำที่เกิดจากเหตุนั้น ตัวอย่างเช่น

IF span of beam is known
THEN depth of beam is (span of bema/12)
IF flow is open channel flow
AND Fround number > 1
THEN flow is supercritical
IF flow is open channel flow
AND Fround number = 1
THEN flow is critical
IF the object has two openings
AND opening at top is larger than that at the bottom
AND enough space is available on left side
AND space available on right side is too small
THEN take the pipe by the left side of the object

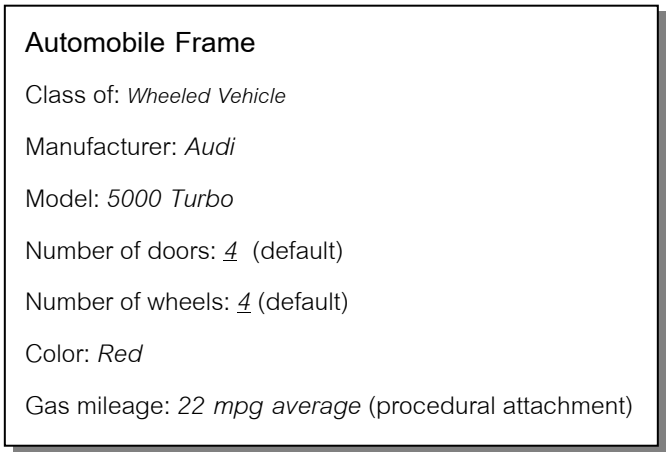
AND take pipe into the object from top opening

เหตุการที่ระบุในกฎที่หนึ่งและสองนั้นเรียบง่าย แต่กฎในข้อที่สามนั้นซับซ้อน ค่าที่ถูกใส่ให้กับตัวแปรสามารถถูกใช้แทนข้อเท็จจริงในกฎข้อที่หนึ่งและสองแต่ในกรณีกฎข้อที่สี่ วัตถุและคุณสมบัติของมันต้องถูกำหนดในการแทนข้อเท็จจริง ในกรณีทั้งหมดข้างต้น เมื่อส่วนเหตุ (IF) ของกฎนั้นเป็นจริงโดยข้อเท็จจริงที่ถูกจัดเก็บในส่วนเนื้อหา หรือข้อเท็จจริงนั้นถูกป้อนเข้ามาโดยผู้ใช้ การกระทำที่ระบุไว้ในส่วนผล (THEN) จะถูกกระทำและกฎจะทำงาน โดยทั่วไปแล้วฐานความรู้จะประกอบไปด้วยกฎหลายๆกฎ ซึ่งตามเหตุผลแล้วกฎจะถูกรวบรวมอยู่ในฐานกฎ (rule bases)

2.2.2 การแทนความรู้แบบเฟรม (Frames)

การแทนความรู้แบบเฟรมได้ถูกเสนอเป็นครั้งแรกโดย Marvin Minsky ในปี 1974 [13] และได้มีการนำเสนอแนวคิดนี้ในงานวิจัยจนถึงปัจจุบัน(ปี 2004) [1] แนวคิดของเฟรมคือโครงสร้างข้อมูล (data structure) สำหรับรวบรวมข้อมูลชนิดต่างๆของแนวคิดหรือความรู้ (typical knowledge) และเรียกว่าโครงสร้างแบบนี้ว่าเฟรม [12][14] ดังนั้นเฟรมเป็นการแทนความรู้แบบหนึ่ง ซึ่งมีทั้งโครงสร้างข้อมูลในการแทนความรู้และความสามารถในการวินิจฉัยด้วย มันยังเหมาะสำหรับเป็นตัวแทนของแนวคิดการจำแนกประเภทด้วย (classification) และยังเหมาะที่จะเป็นตัวแทนของอนุกรมวิธานหรือการจำแนกหมวดหมู่แบบลำดับขั้นด้วย (taxonomy hierarchy) [12][15] ใน เอนทิตี้ (entity) หนึ่งของเฟรมจะรวบรวมความรู้ที่จำเป็นทั้งหมดเกี่ยวกับวัตถุหรือแนวคิดนั้น

2.2.2.1 โครงสร้างของเฟรม โครงสร้างของเฟรมประกอบไปด้วย ชื่อของเฟรมที่ใช้อ้างถึงเฟรม, สล็อตหรือ แอททริบิวต์ของเฟรม และ ฟาเซท[1][14] เฟรมมักจะมีหลากหลายความหมาย บางทีก็อ้างถึงวัตถุเฉพาะเจาะจง (particular object) และ บางทีก็อ้างถึง กลุ่มของวัตถุที่มีคุณสมบัติเหมือนกัน (group of similar object) เพื่อแยกแยะความแตกต่างนี้ จึงแยกชนิดของ เฟรมออกเป็น คลาสเฟรม และ อินสแตนท์เฟรม



รูปที่ 2.3 ตัวอย่างโครงสร้างของเฟรม

คลาสเฟรม : จะใช้เพื่ออธิบายกลุ่มของวัตถุ หรือ คลาสของวัตถุ และสามารถถูกรวบรวมแบบเป็นอนุกรมวิธาน (taxonomy) ได้ด้วย ซึ่ง คลาสเฟรม จะมี สล็อตที่เหมือนกันในทุกๆ คลาสเฟรม คือ สล๊อดแม่และ สล๊อตลูก ยกเว้น รูธคลาส ที่ไม่มี สล๊อดแม่โดยที่ สล็อตอื่นๆ จากเฟรมแม่สามารถถูกสืบทอดไปยังเฟรมลูกได้ด้วยการ สืบทอด

อินสแตนท์เฟรม : จะใช้ในการอธิบายถึงวัตถุเฉพาะเจาะจง ซึ่งส่วนมากเป็น โหนดสุดท้าย ของ อนุกรมวิธาน และเป็นเฟรมที่ไม่มี สล๊อตลูก อีกแล้ว เป็น เฟรมที่ใช้ในการอ้างถึง วัตถุเฉพาะเจาะจง หรือ อินสแตนท์เฟรม จะมีเพียง สล็อตที่บอกความสัมพันธ์ เป็นสล๊อตแม่เท่านั้น และรับถ่ายทอดคุณสมบัติหรือ สล็อตต่างๆ มาจาก คลาสเฟรม โดยที่อาจจะมีการเปลี่ยนแปลงค่าในสล็อตที่ อินสแตนท์เฟรม ก็ได้ เพราะฉะนั้น ค่าของ สล็อตใน อินสแตนท์เฟรม จึงไม่จำเป็นต้องมีค่าเดียวกันกับ คลาสเฟรม เสมอไป ในอีกนัยหนึ่งอาจกล่าวได้ว่า คลาสเฟรม นั้นจะทำหน้าที่เสมือนเป็นต้นแบบของ อินสแตนท์เฟรม นั่นเอง [1][2][14][15][16]

สล็อต: ในแต่ละเฟรมจะประกอบไปด้วย สล็อตหรือ แอททริบิวต์เพื่อบอกคุณสมบัติของเฟรมนั้นๆ ในเฟรมหนึ่งๆอาจมีได้หลาย สล็อตและบางทีอาจกล่าวได้ว่า แนวคิดของเฟรมนั้นจะถูกกำหนดโดยการรวบรวม สล็อตเข้าด้วยกัน [12] ในแต่ละ สล็อตจะอธิบายถึงคุณสมบัติ หรือการดำเนินการของเฟรม และในแต่ละ สล็อตจะมีค่าของ สล็อตซึ่งเรียกว่า ค่าในสล็อต หรือ ฟิลเลอร์ [15] ซึ่งค่าของ สล็อตนั้นอาจเป็นได้ทั้งค่าโดยปริยาย (default value), เป็นพอยด์เตอร์ ชี้ไปยังเฟรมอื่น, หรือเซตของกระบวนการที่ค่านั้นได้มาก็ได้ [11][14]

ฟาเซท: ในแต่ละ สล็อตจะมี ฟาเซทซึ่งอาจมีมากกว่า หนึ่ง ฟาเซทในหนึ่ง สล็อตเพื่อใช้ควบคุมค่าของ สล็อตและตัวดำเนินการใน สล็อตนั้น ฟาเซทยังสามารถใช้เพื่อกำหนดค่าเริ่มต้นให้กับ ค่าของสล๊อต, กำหนดชนิดของข้อมูล, กำหนดขอบเขตของค่าเป็นไปได้, และกำหนดกิจกรรมที่จะกระทำการต่อไปได้ ชนิดของแต่ละ ฟาเซทมีดังนี้ [14]

Value Type : เป็น ฟาเซทประเภทพื้นฐานที่อธิบายชนิดของค่าใน สล๊อตเป็นการกำหนด type restriction ของค่าใน ค่าของสล๊อต ซึ่ง ฟาเซทนี้จะสามารถมีได้แค่ ค่าเดียว โดยจะเป็นสัญลักษณ์เดี่ยวๆหรือ ลิสต์ ของสัญลักษณ์ เช่น integer, string หรือเป็นลิสต์ ของ ค่าที่สามารถเป็นไปได้ เป็นต้น

Default : เป็นค่าที่ถูกนำมาใช้ เมื่อ ค่าของสล๊อต นั้นว่าง หรือไม่มีการกำหนดค่าใดๆให้

Constraint : เป็น ฟาเซทที่กำหนดค่าที่อนุญาตให้มีได้

Minimum cardinality : เป็น ฟาเซทที่กำหนดจำนวนค่าของ ค่าของสล๊อต ว่ามีได้อย่างน้อยสุด N ค่า

Maximum cardinality : เป็น ฟาเซทที่กำหนดจำนวนค่าของ ค่าของสล๊อต ว่ามีได้อย่างมากที่สุด N ค่า

Range : เป็น ฟาเซทที่ทำหน้าที่ระบุขอบเขตของข้อมูลที่จะปรากฏใน ค่าของสล๊อต เช่น เลขจำนวนเต็ม, เลขทศนิยม หรือ 10 ถึง 100 เป็นต้น

If added facet : นี้เป็น ฟาเซทแบบกระบวนการที่ระบุการปฏิบัติที่จะเกิดขึ้น เมื่อมีการเพิ่มค่าเข้ามาในสล็อตบางครั้งเราอาจเรียกว่าเป็น ดีมอน (demons)

If needed : เป็น ฟาเซทประเภทเดียวกับ ถ้าถูกเพิ่มเติม แต่จะทำงานเมื่อไม่มีการระบุค่าให้ค่าของสล๊อต ที่มีความต้องการ เพื่อใช้ประมวลผลขณะนั้น

If removed : เป็นกระบวนการที่จะปฏิบัติ เมื่อค่าภายใน สล็อตถูกลบออก

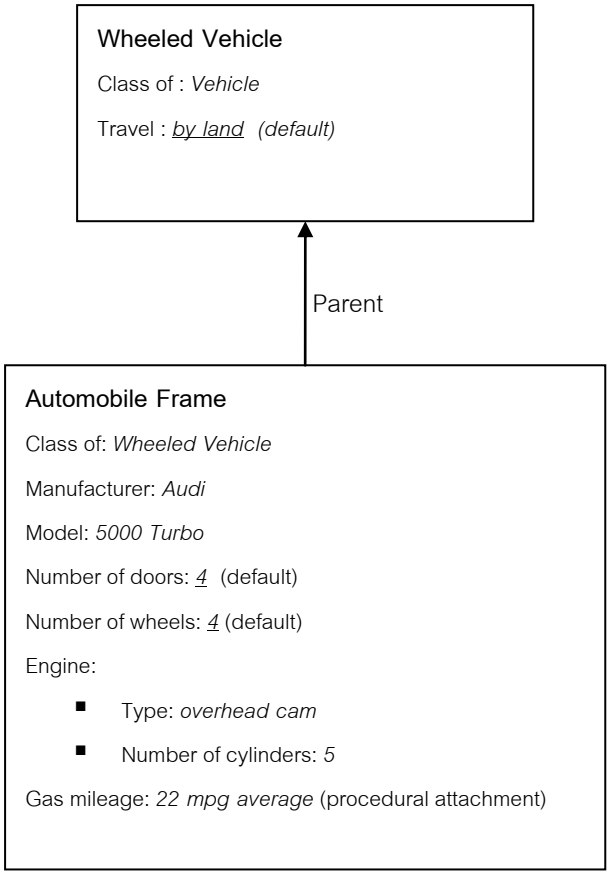
ฟาเซทสามารถมีโครงสร้างเป็นกฎได้ นั่นหมายความว่าเราสามารถกำหนดค่าของสล็อตหรือ ฟาเซทในลักษณะหลายค่า (Multi-Value) ได้ ดังนั้นอาจสรุปได้ว่าค่าของ ฟาเซทนั้น ไม่จำเป็นต้องมีคุณสมบัติอะตอมมิค (Atomic) นั่นเอง

ซึ่งจากที่ได้กล่าวมาข้างต้น สรุปได้ว่า ฟาเซทนั้นเป็น การตรวจสอบค่า อย่างหนึ่งของเฟรม ที่ทำการควบคุมและกำหนด ค่าของสล๊อต หรือ ฟิลเลอร์ ให้เป็นไปตามข้อกำหนด ซึ่งทำหน้าที่เสมือนเป็นกฎที่ใช้ในการจัดการเกี่ยวกับค่าที่เพิ่ม, ลบ, หรือ ปรับปรุง ค่าใน ฟิลเลอร์

นอกจากนี้ เนื่องจาก ฟาเซทเป็นกฎที่ใช้ในการเพิ่มค่าเข้าไปได้ด้วย ทำให้เมื่อมีค่าที่เพิ่มเข้ามาใหม่ใน ฟิลเลอร์ แล้ว ฟาเซทจะมีหน้าที่ไปตรวจสอบค่าใน สล็อตอื่นๆ หรือเมื่อมีการร้องขอดูค่าใน ฟิลเลอร์ ของ สล็อตนั้นๆ ฟาเซทก็สามารถร้องขอค่าของ ฟิลเลอร์ ที่ สล็อตของเฟรมแม่ได้ ดังนั้น ฟาเซทจึงมีความสามารถในการวินิจฉัยได้ด้วย ซึ่งการวินิจฉัยแบบนี้เป็นการวินิจฉัยแบบสืบทอด ดังนั้นจึงกล่าวได้ว่าระบบการแทนความรู้แบบเฟรม จะสามารถทำการวินิจฉัยโดยการสืบทอดโดยอัตโนมัตินั่นเอง

ในรูปที่ 2.4 เป็นตัวอย่างโครงสร้างและความสัมพันธ์ของเฟรม โดยที่ เฟรมรถยนต์ (automobile frame) จะมีความสัมพันธ์กับเฟรมเครื่องยนต์ที่มีล้อ (wheeled vehicle) โดยเฟรมรถยนต์จะเป็นเฟรมลูกของ ของเฟรมเครื่องยนต์ที่มีล้อ เป็นต้น

เฟรมนั้นมีความคล้ายคลึงกันกับวัตถุในภาษาทางด้านโปรแกรมเชิงวัตถุ และ ฐานข้อมูลเชิงวัตถุ แต่ข้อแตกต่างที่สำคัญระหว่าง เฟรม กับภาษาทางด้านวัตถุ คือ มันไม่มีคุณสมบัติการห่อหุ้มวัตถุ (encapsulation) ในภาษาทางด้านวัตถุนั้น วัตถุจะถูกห่อหุ้มรายละเอียดภายในไว้ แต่เฟรมนั้นไม่ เพราะฉะนั้นคุณสมบัติของคลาสของวัตถุจะไม่สามารถถูกมองเห็นได้จากผู้ใช้ จะเห็นเพียงแต่กระบวนการเท่านั้น รูปแบบทางด้านวัตถุที่ไม่มีคุณสมบัติของการห่อหุ้ม นั้นเป็นที่รู้จักกันดีในชื่อของ รูปแบบเชิงวัตถุสัมพันธ์ ซึ่งเข้ากันได้ดียิ่งกับรูปแบบของเฟรม งานวิจัยนี้จึงใช้ระบบการจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์



รูปที่ 2.4 ตัวอย่างโครงสร้างและความสัมพันธ์ของเฟรมรถยนต์

2.2.2.2 การใช้เทคนิคการแทนความรู้แบบเฟรมในงานวิจัยทั่วไป สำหรับเทคนิคการการแทนความรู้แบบเฟรมนี้ เป็นที่ได้รับความสนใจ และนิยมนำไปใช้พัฒนาในงานวิจัยหลายด้าน โดยเฉพาะงานวิจัยที่ต้องการสร้างตัวแทนความรู้สำหรับความรู้ที่มีโครงสร้างซับซ้อน เพราะเป็นที่ยอมรับกันโดยทั่วไปว่า หลักแนวคิดในแบบเฟรม ซึ่งมองรายละเอียดโครงสร้างความรู้ในรูปแบบใกล้เคียงกับวัตถุ จะสามารถรองรับการประมวลผลความรู้ที่ใกล้เคียงกับโลกแห่งความเป็นจริงได้มากขึ้น ถึงแม้ยังจะพบข้อจำกัดบางประการอยู่ [1] แต่ก็ยังมีนักวิจัยที่พยายามพัฒนารูปแบบ

การแทนความรู้แบบเฟรมขึ้นมาอยู่เสมอ ดังจะเห็นได้จากการพยายามนำเทคนิคการแทนความรู้แบบเฟรมนี้มาพัฒนางานทั้งด้านการแพทย์ [1][16], งานวิศวกรรม [14] , การออกแบบพัฒนาซอฟท์แวร์ [4] รวมถึงการพัฒนาระบบผู้เชี่ยวชาญด้านอื่นๆ โดยทั่วไป [2] โดยเครื่องมือการพัฒนาส่วนใหญ่ที่ใช้มักจะอยู่ในรูปของภาษา, เครื่องมือ หรือเปลือกระบบผู้เชี่ยวชาญที่ยึดมาตรฐานตามภาษาด้านปัญญาประดิษฐ์ในกลุ่มภาษา ลิสพ์ (LISP)หรือ โปรลอก (Prolog) เป็นต้น ซึ่งเครื่องมือที่ได้รับความนิยมส่วนใหญ่ได้แก่ Smalltalk, KEE, Kappa-PC, KL-One, Protégé-2000 เป็นต้น

2.3 กลไกการวินิจฉัยบนระบบผู้เชี่ยวชาญ (Inference Mechanisms)

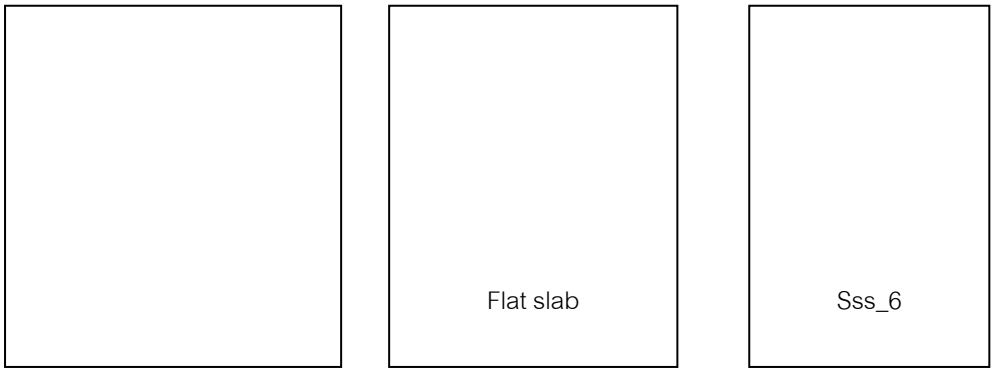
เป็นส่วนควบคุมหรือเทคนิคการค้นหาซึ่งจะค้นหาทั้งฐานความรู้เพื่อการตัดสินใจ ฐานความรู้จะเป็นส่วน state space และกลไกการวินิจฉัยจะเป็นกระบวนการค้นหา กระบวนการวินิจฉัยจะใช้สัญลักษณ์โดยการเลือกกฎที่สอดคล้องกับสัญลักษณ์ของข้อเท็จจริงและรันกฎเพื่อให้ได้ข้อเท็จจริงใหม่ ซึ่งกระบวนการนี้จะกระทำต่อเนื่องเป็นห่วงโซ่จรกว่าจะพบเป้าหมาย ในระบบผู้เชี่ยวชาญนั้นกระบวนการวินิจฉัยสามารถถูกกระทำได้หลายทาง ซึ่งกระบวนการวินิจฉัยที่นิยมใช้กันมากสองวิธีคือ backward chaining และ forward chaining โดยที่ backward chaining คือ กระบวนการที่ขับเคลื่อนโดยเป้าหมาย (goal-driven process) ในขณะที่ forward chaining เป็นกระบวนการที่ขับเคลื่อนโดยข้อมูล (data-driven process)

2.3.1 Backward chaining

เป็นกระบวนการที่ขับเคลื่อนโดยเป้าหมาย โดยการตั้งเป้าหมายที่อยู่ในฐานความรู้ขึ้นมาเป็นตัวแปร และกระบวนการวินิจฉัยจะหยุดทำงานเมื่อได้ค่าของตัวแปร โครงสร้างข้อมูลที่ใช้ในระหว่างกระบวนการวินิจฉัยคือส่วนหน่วยความจำปฏิบัติงาน (working memory) หรือ เนื้องาน (context) ส่วน สแต็กของกฎ (rule stack) และส่วนสแต็กของเป้าหมาย (goal stack) ซึ่งเมื่อไหร่ก็ตามที่เกิดการกระทำอย่างหนึ่งในกระบวนการวินิจฉัย เช่น การเลือก(select) การจับคู่(match) และการรันกระบวนการ(execute) เกิดขึ้น ข้อมูลที่ใช้ในส่วนกระบวนการวินิจฉัยเหล่านั้นจะถูกเปลี่ยนแปลง การศึกษาถึงผลกระทบของการกระทำบนโครงสร้างข้อมูลระหว่างกระบวนการวินิจฉัยเป็นทางที่ดีกว่าในการศึกษากระบวนการของกลไกการวินิจฉัย ดังนั้นการทำงานของกระบวนการวินิจฉัยจะถูกอธิบายผ่านการเปลี่ยนแปลงที่เกิดขึ้นกับส่วนเนื้องาน ส่วนสแต็กของกฎ และส่วนสแต็กของเป้าหมาย ซึ่งในส่วนสแต็กของกฎ และสแต็กเป้าหมายจะเป็นโครงสร้างข้อมูลชั่วคราวที่สร้างขึ้นเพื่อจัดเก็บ

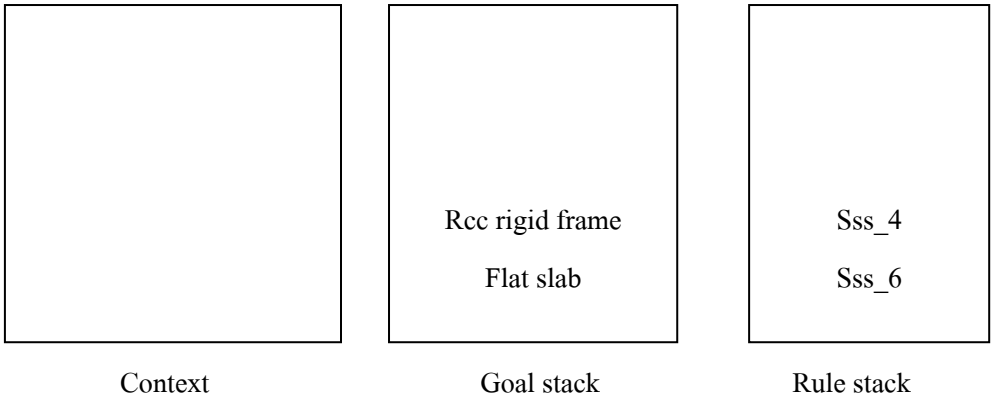
โดยเมื่อเริ่มกระบวนการในส่วนของเนื้องานจะว่างเปล่าและตัวแปรเป้าหมายจะถูกดันเข้าไปในสแต็กเป้าหมาย จากนั้นกระบวนการจะทำการเลือกกฎแรกในฐานกฎซึ่งมีตัวแปรเป้าหมายในส่วน THEN และดันเข้าไปในสแต็กของกฎ ซึ่งตัวอย่างอย่างกฎข้อที่ 6 จะเป็นกฎแรกที่มีตัวแปร

เป้าหมายในส่วนผล ดังนั้นกฎข้อที่ 6 จะถูกดันเข้าไปในสแต็กของกฎและตัวแปรเป้าหมายจะอยู่ในสแต็กเป้าหมาย สถานะของเนื้องาน สแต็กเป้าหมาย และสแต็กของกฎแสดงในรูปที่ 2.5 ดังนี้



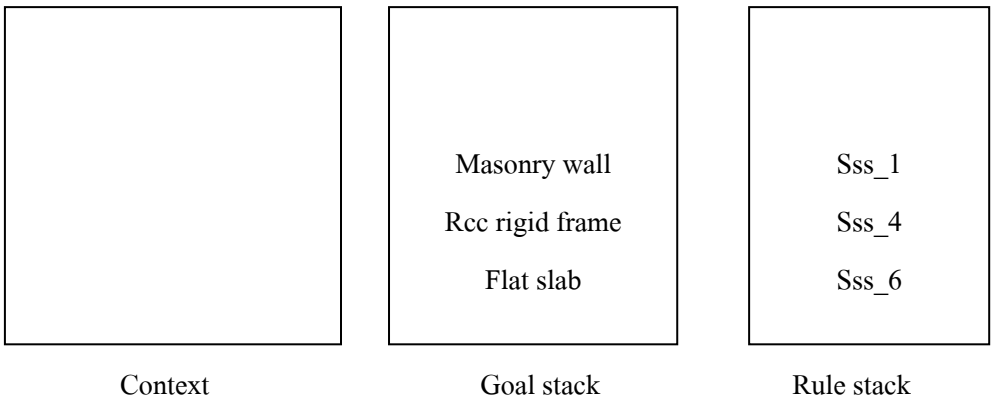
รูปที่ 2.5 สถานะของเนื้องาน สแต็กเป้าหมาย และสแต็กของกฎ I

จะเป็นหากฎข้อที่ 6 เป็นกฎแรกที่มีตัวแปรเป้าหมายในส่วนของผล ซึ่งเมื่อกฎข้อที่ 6 ถูกเลือกขึ้นมา มันจะเริ่มทำการจับคู่กับเหตุในส่วนที่มันปรากฏขึ้นในกฎซึ่งเงื่อนไขแรกคือ "structural system IS rcc rigid frame" การจับคู่กฎสามารถกระทำต่อไปได้ก็ต่อเมื่อข้อเท็จจริงนี้มันได้ถูกตั้งไว้แล้ว ดังนั้นกระบวนการวินิจฉัยจะเซตตัวแปร "structural system" ให้เป็นเป้าหมายปัจจุบัน ซึ่งมันจะถูกเซตโดยการดันตัวแปรเข้าไปในสแต็กเป้าหมายและยังได้ดันกฎแรกในลิสต์ที่มีตัวแปรเป้าหมายในส่วน THEN เข้าไปในสแต็กของกฎด้วย ดังนั้นกฎข้อที่ 4 จะถูกเลือกสำหรับการหาค่าโดยสถานะของเนื้องานและสแต็กทั้งสองแสดงในรูปที่ 2.6



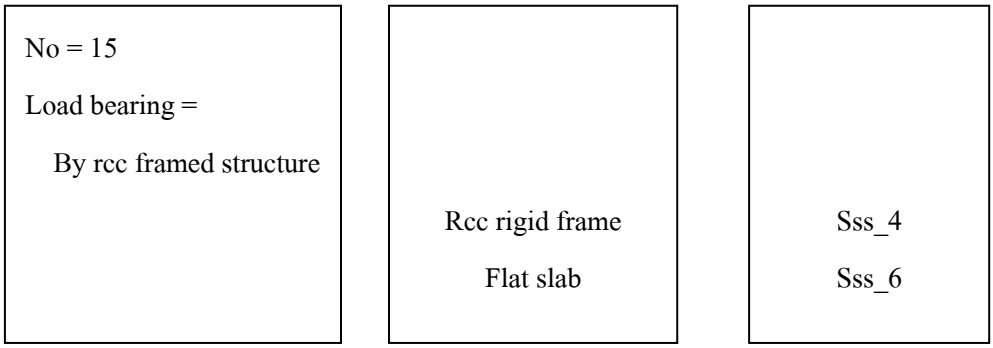
รูปที่ 2.6 สถานะของเนื้องาน สแต็กเป้าหมาย และสแต็กของกฎ II

ตอนนี้กระบวนการวินิจฉัยจะเริ่มตรวจสอบเงื่อนไขแรกของกฎข้อที่ 4 ซึ่งก็คือ "load bearing IS by rcc framed structure" ซึ่งจากผลที่ได้ตัวแปร "load bearing" จะกลายเป็นเป้าหมายปัจจุบันและถูกดันเข้าไปในสแต็กเป้าหมายและกฎข้อที่ 1 จะถูกดันเข้าไปในสแต็กของกฎดังแสดงในรูปที่ 2.7



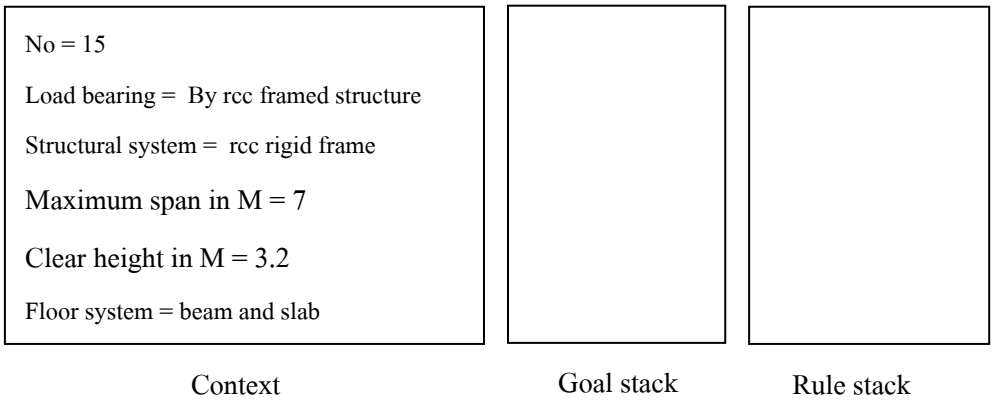
รูปที่ 2.7 สถานะของเนื้องาน สแต็กเป้าหมาย และสแต็กของกฎ III

เงื่อนไขแรกของกฎข้อที่ 1 นั้นคือ "no of stories" ซึ่งร้องขอค่าให้ผู้ใช้ป้อนค่าสำหรับตัวแปรนี้ ซึ่งสมมติให้ผู้ใช้ป้อนค่าเป็น 15 ซึ่งเป็นข้อเท็จจริงแรกและมันจะถูกจัดเก็บอยู่ในเนื้องาน ทำให้เงื่อนไขของกฎข้อที่ 1 คือ "no of stories <= 5" เป็นเท็จ และกฎข้อที่ 1 จะไม่ถูกใช้และถูกปฏิเสธไปโดยการดึงออกจากสแต็กของกฎ กระบวนการวินิจฉัยจะไปยังกฎถัดไปที่มีผลเหมือนกัน (ซึ่งก็คือกฎข้อที่ 2) และดันเข้าไปในสแต็กของกฎและเริ่มการหาค่าในส่วนเงื่อนไขแรกของเหตุ ซึ่งก็เป็นเท็จเหมือนกัน สแต็กของกฎก็จะถูกดึงออกและกฎถัดไปซึ่งมีตัวแปรเป้าหมาย ('load bearing') ในส่วนผลซึ่งก็คือกฎข้อที่ 3 จะถูกดันเข้าไปในสแต็กซึ่งในที่นี้เงื่อนไขในส่วนเหตุจะสอดคล้องกับส่วนเนื้องานและได้ค่าเป็นจริงดังนั้นกฎจะถูกใช้ ซึ่งจะทำให้ได้เป้าหมายปัจจุบันและข้อเท็จจริงที่ได้จะถูกเก็บไว้ในส่วนเนื้องาน ซึ่งเมื่อเป้าหมายปัจจุบันถูกตั้งขึ้นและกฎถูกใช้ไปแล้ว ทั้งสแต็กของกฎและสแต็กเป้าหมายจะถูกดึงออกดังแสดงในรูปที่ 2.8



รูปที่ 2.8 สถานะของเนื้องาน สแต็กเป้าหมาย และสแต็กของกฎ IV

และในขณะนี้เป้าหมายปัจจุบันคือส่วนที่อยู่บนสุดของสแต็กเป้าหมาย ซึ่งก็คือ 'structural system' และกฎปัจจุบันคือกฎข้อที่ 4 ซึ่งเงื่อนไขของกฎเป็นจริงโดยการเปรียบเทียบกับข้อเท็จจริงในส่วนเนื้องานซึ่งจะได้ข้อเท็จจริงใหม่คือ 'structural system IS rcc rigid frame' ได้ถูกเพิ่มเข้าไปในส่วนเนื้องาน และสแต็กทั้งสองจะถูกดึงออกและส่วนเป้าหมายปัจจุบันจะกลายเป็น 'floor system' โดยกฎปัจจุบันก็คือกฎข้อที่ 6 เงื่อนไขแรกของกฎปัจจุบันจะเป็นจริงและเงื่อนไขที่สองจะถูกหยิบขึ้นมา ซึ่งเป็นตัวแปรในสถานะเริ่มต้นของฐานความรู้โดยผู้ใช้ต้องการการป้อนค่าเข้าไป สมมติว่าผู้ใช้ป้อนค่า 7 เข้ามา ซึ่งทำการผลลัพธ์ที่ได้จากเงื่อนไขนี้มีค่าเป็นเท็จและกฎถูกปฏิเสธ และถูกดึงออกจากสแต็ก โดยที่กฎถัดไปซึ่งมีเป้าหมายปัจจุบันในส่วนของผลจะถูกเลือกขึ้นมาแทน ในที่นี้คือกฎข้อที่ 7 ซึ่งจะถูกปฏิเสธด้วยเหมือนกันเนื่องจากเงื่อนไขข้อที่สองเป็นเท็จ ดังนั้นกฎข้อที่ 8 จะถูกเลือกจะใส่เข้าไปในสแต็กกฎ ซึ่งเงื่อนไขสองอันแรกเป็นจริงจะผู้ใช้จะต้องใส่ค่าตัวแปรในเงื่อนไขที่สาม โดยให้ผู้ใช้ป้อนค่าเข้ามาเป็น 3.2 ซึ่งจะได้ค่าเป็นจริงและกฎจะถูกใช้ซึ่งจะได้ผลลัพธ์เป็นค่า 'beam and slab' สำหรับตัวแปรเป้าหมาย ซึ่งเมื่อกฎถูกใช้ทั้งสองสแต็กจะถูกดึงออกจะข้อเท็จจริงใหม่จะถูกเพิ่มเข้ามายังเนื้องาน ซึ่งตอนนี้ทั้งสองสแต็กมันว่างเปล่าและกระบวนการวินิจฉัยก็สิ้นสุดลงโดยสถานะของเนื้องานของตัวอย่างนี้แสดงในรูปที่ 2.9



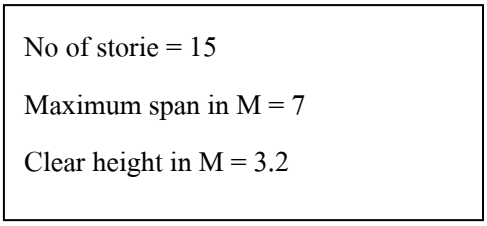
รูปที่ 2.9 สถานะของเนื้องาน สแต็กเป้าหมาย และสแต็กของกฎ V

ในส่วนการอธิบายการดำเนินงานของระบบผู้เชี่ยวชาญจะมีกลไกสำหรับการสอบถามเนื้องานในการได้มาซึ่งค่าของตัวแปร (what?) และรู้ว่าข้อเท็จจริงเกิดขึ้นได้อย่างไร (how?) คำตอบสำหรับคำถามแรกจะได้มาจากส่วนเนื้องานโดยตรง ในขณะที่คำตอบที่ได้จากคำถามที่สองจะต้องการการอ้างอิงถึงฐานความรู้ โดยกระบวนการที่เกี่ยวข้องกับคำถามว่าเกิดขึ้นได้อย่างไรจะค้นหากฎซึ่งต้องการข้อเท็จจริงในส่วน THEN และแสดงกฎซึ่งแสดงให้เห็นว่าข้อเท็จจริงเกิดขึ้นได้อย่างไร การได้มาซึ่งคำตอบของคำถามแล้วถ้า (what if?) จะมีประโยชน์อย่างมากถ้าผู้ใช้ของระบบต้องการรู้กระบวนการตัดสินใจโดยการเปลี่ยนค่าที่ป้อนเข้าไป ซึ่งสามารถทำได้ในสองขั้นตอน ขั้นแรกข้อเท็จจริงเหล่านั้นที่ขึ้นอยู่กับการเปลี่ยนแปลงตัวแปรต้องถูกยกเลิกก่อน จากนั้นมันต้องถูกกำหนดใหม่โดยกระบวนการวินิจฉัยจากจุดนั้น ซึ่งจะค่อนข้างซับซ้อนและต้องมีการรักษาเส้นทางการวินิจฉัยที่เสร็จสิ้นแล้วด้วยขึ้นอยู่กับข้อเท็จจริงที่อยู่ในเนื้องาน กระบวนการวินิจฉัยจะสร้างเครือข่ายที่ขึ้นกันทั้งหมดจาก TMS (Truth Maintenace System) ซึ่งเครือข่ายนี้จะช่วยในการย้อนกลับ (backtrack) จากขั้นตอนใดๆไปยังขั้นตอนก่อนหน้าโดยยกเลิกการกระทำที่เกี่ยวข้อง ซึ่งจะได้ผลลัพธ์ในการลบข้อเท็จจริงต่างๆออกไปจากส่วนเนื้องาน ซึ่งการย้อนกลับนี้จะกระทำต่อไปเรื่อยๆจนกว่าจะถึงส่วนก่อตั้งตัวแปรข้อเท็จจริง โดยค่าของมันจะถูกเปลี่ยนแปลงด้วยคำถามแล้วถ้า (what if?) กระบวนการวินิจฉัยจะเริ่มใหม่จากจุดนั้นจะต่อเนื่องไปเรื่อยๆจนกว่าเป้าหมายสุดท้ายจะเกิดขึ้น และเพราะว่าความยากที่เกิดขึ้นในการบำรุงรักษาเครือข่ายที่ขึ้นต่อกันของการตัดสินใจดังนั้นเครื่องมือที่ใช้ในเชิงพาณิชย์จึงไม่มีส่วนนี้

2.3.2 Forward chaining

เป็นกระบวนการวินิจฉัยโดยขับเคลื่อนข้อมูล ผู้ใช้ของระบบต้องให้ข้อมูลทั้งหมดก่อนที่จะเริ่มกระบวนการวินิจฉัยซึ่งกลไกการวินิจฉัยจะพยายามที่จะตั้งข้อเท็จจริงที่อยู่ใน

ฐานความรู้จนกว่าเป้าหมายจะถูกค้นพบ พิจารณาฐานกฎเดียวกัน ผู้ใช้จะให้ข้อมูลและสถานะของเนื้องานก่อนที่จะเริ่มกระบวนการวินิจฉัยดังแสดงในรูปที่ 2.10



รูปที่ 2.10 สถานะของเนื้องานก่อนที่จะเริ่มกระบวนการวินิจฉัย

กระบวนการวินิจฉัยจะเลือกกฎแรกในฐานกฎขึ้นมาและปฏิเสธมันถ้าเงื่อนไขมันเป็นเท็จ จากนั้นมันจะไปยังกฎที่สองซึ่งก็ถูกปฏิเสธเช่นกัน แต่ส่วนเงื่อนไขในกฎข้อที่สามนั้นเป็นจริง และกฎจะถูกใช้ได้ผลเป็นข้อเท็จจริงใหม่ที่ถูกเพิ่มเข้ามาในส่วนเนื้องาน กฎข้อที่ 4 ก็ถูกใช้แต่กฎข้อที่ 5, 6, 7 จะถูกปฏิเสธและสุดท้ายกฎข้อที่ 8 จะถูกใช้เพื่อให้ได้เป้าหมายและสิ้นสุดกระบวนการวินิจฉัย ในฐานกฎขนาดใหญ่จะมีการทำซ้ำก่อนที่จะเกิดเป้าหมาย ลำดับของกฎซึ่งอยู่ในฐานกฎจะทำหน้าที่สำคัญในแนวทางการวินิจฉัยในกระบวนการนี้ ในขณะที่ลำดับจะไม่มีบทบาทใน backward chaining แต่ลำดับที่เงื่อนไขถูกลิสต์อยู่ในกฎนั้นสำคัญใน backward chaining ลำดับคำถามจะถามผู้ใช้เพื่อรอบรับลำดับเหล่านี้ ดังนั้นก่อนที่จะสร้างฐานกฎ วิศวกรความรู้ควรตัดสินว่าจะใช้ backward chaining หรือ forward chaining ในการวินิจฉัย

วิธีการวินิจฉัยแบบที่สามคือกลไกการวินิจฉัยที่ผสมกันระหว่างกระบวนการ backward chaining และ forward chaining ซึ่งกระบวนการแบบ backward นั้นจะเหมาะสมถ้ามีเป้าหมายน้อยแต่มีสถานะเริ่มต้นเยอะ ลำดับของข้อมูลที่ถูกร้องขอขึ้นกับการไหลของกระบวนการวินิจฉัย ส่วนในกระบวนการแบบ forward นั้นทุกๆข้อมูลที่ผู้ใช้รู้ต้องถูกป้อนเข้ามาในระบบก่อนและระบบไม่ต้องรอให้ผู้ใช้ป้อนเข้ามาในระบบอีก ซึ่งถ้าข้อมูลมันเพียงพอก็จะบรรลุเป้าหมาย หรือไม่ก็จะหยุดโดยบอกว่าไม่สามารถบรรลุเป้าหมายได้ด้วยข้อมูลที่มีอยู่ ดังนั้นกระบวนการแบบ forward จะเหมาะสมเมื่อมีสถานะเริ่มต้นน้อยๆแต่มีได้หลากหลายเป้าหมายเท่านั้น ซึ่งระบบผู้เชี่ยวชาญขนาดใหญ่ จะมีได้หลากหลายสถานเริ่มต้น และหลายเป้าหมาย ดังนั้นผู้ใช้อาจไม่สามารถให้ข้อมูลที่จำเป็นทั้งหมดก่อนได้ และเขาอาจไม่สามารถมองออกว่าการไหลของกระบวนการวินิจฉัยเป็นอย่างไร ซึ่งลักษณะนี้กลไกการวินิจฉัยแบบผสมอาจเหมาะสม โดยมันจะเริ่มจากการทำงานแบบ forward ก่อนเพื่อให้ได้ข้อเท็จจริงบางส่วนจากนั้นจะใช้การทำงานแบบ backward เพื่อพิสูจน์มัน ซึ่งกระบวนการแบบผสมคือ forward ซึ่ง backward จะถูกเรียกขึ้นมาเมื่อข้อเท็จจริงถูกตั้งขึ้น ผู้ใช้จะ

ไม่ต้องป้อนข้อมูลทั้งหมดตั้งแต่เริ่มต้น แต่จะป้อนข้อมูลส่วนหนึ่งเข้ามาก่อนแล้วป้อนอีกครั้งเมื่อมีการร้องขอ

ถ้าเฟรมถูกใช้กับกฎในการแทนความรู้ ข้อเท็จจริงที่เกี่ยวข้องกับตัวแปรที่ถูกในเฟรมจะถูกเก็บในเนื้องานแยกออกมาต่างหากเรียกว่าฐานเฟรม ฐานเฟรมสามารถมีได้หลายเฟรมอยู่ภายในซึ่งชนิดของสล๊อตมีได้ทั้ง concrete หรือ abstract และเฟรมยังสามารถถูกเชื่อมโยงโดยใช้ความสัมพันธ์อย่างมีความหมายซึ่งสามารถสร้างเครือข่ายความหมายสำหรับแทนความรู้แบบพรรณา แต่โดยทั่วๆไปนั้น chaining จะถูกกระทำบนตัวแปรที่อยู่ในฐานกฎและไม่อยู่ในตัวแปรเฟรม ดังนั้นทุกๆตัวแปรกที่กำหนดในกระบวนการวินิจฉัยจะถูกกำหนดเป็นตัวแปรฐานกฎทั้งหมด

2.3.3 การวินิจฉัยโดยใช้ความรู้ในการแทนความรู้แบบเฟรม

ในกรณีที่มีชนิดของการวินิจฉัยที่แตกต่างกันสามารถทำได้ด้วยการแทนความรู้แบบเฟรม การวินิจฉัยเท่านั้นจะไม่มีรูปแบบควบคุมวิธีการทั้งหมดในการแก้ปัญหาอย่างในกรณีการวินิจฉัยบนฐานกฎ แต่มันจะแสดงลักษณะที่ฉลาดหลายๆอย่างที่มนุษย์ใช้ในกระบวนการของการแก้ปัญหา ลักษณะอย่างหนึ่งคือสามัญสำนึก ดังตัวอย่างต่อไปนี้ซึ่งแสดงการจำลองสามัญสำนึกโดยใช้เฟรม

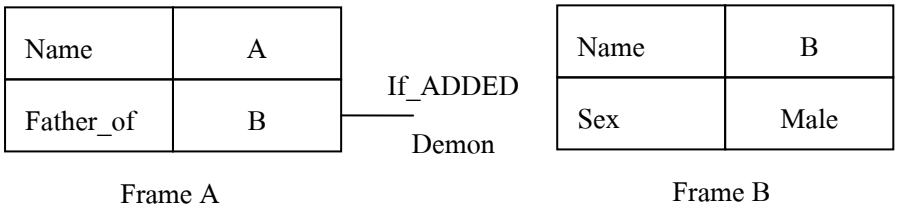
พิจารณาสถานการณ์ซึ่งมีบางคนบอกเราว่า A เป็นพ่อของ B และ B เป็นผู้ชาย จากนั้นอ้างถึงลูกชายของ A ซึ่งจะทำให้เราคิดได้ทันทีว่า B เป็นลูกชายของ A และไม่มีข้อกำหนดว่า B เป็นลูกชายของ A แต่ด้วยสามัญสำนึก (โดยมีพื้นฐานบนความรู้ที่เรามีเรื่องความสัมพันธ์ระหว่างพ่อแม่และลูก) เราจะสามารถสร้างข้อกำหนดได้ว่า B เป็นลูกชายของ A และในสถานการณ์ที่คล้ายคลึงกัน ซึ่งก็คือการวินิจฉัยซึ่งมีพื้นฐานมาจากสามัญสำนึกสามารถถูกจำลองสถานการณ์ในโปรแกรมคอมพิวเตอร์โดยใช้เทคนิคซึ่งเกี่ยวข้องกับเฟรมและการจัดการเฟรมและทำได้อย่างไรดังนี้

ลักษณะพื้นฐานสองประการของระบบการจัดการเฟรมคือเฟรมจะถูกสร้างระหว่างเวลารัน (run time) และสล๊อตกับค่าของมันจะถูกเพิ่มเข้าไประหว่างกระบวนการวินิจฉัย

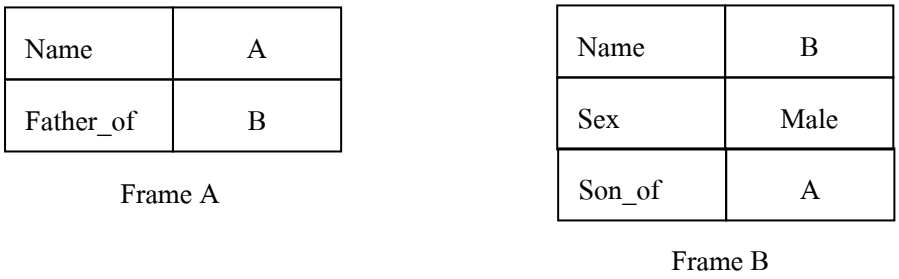
สมมติให้ A และ B เป็นเฟรมสองเฟรมซึ่งมีสองสล๊อตคือ name และ father_of ซึ่งสำหรับ A และมีสล๊อต name และ sex สำหรับเฟรม B โดยใช้สล๊อต name ถูกใส่ค่าเข้าไปเป็น A และ B ตามลำดับและสล๊อต sex ของเฟรม B ถูกใส่ค่าเข้าไปเป็น male โดยการระบุข้อกำหนด A เป็นพ่อของ B จะสำเร็จด้วยการกำหนดให้ใส่ค่า B ในสล๊อต father of ในเฟรม A ซึ่งเดมอน (กระบวนการทำงาน) ชนิด if_added จะถูกผูกติดอยู่กับสล๊อต father_of ของเฟรม A ซึ่งเดมอนนี้จะถูกกระตุ้นขึ้นมาเมื่อค่าถูกใส่เข้าไปในสล๊อต รูปที่ 2.11 จะแสดงสถานะของเฟรมก่อนถูกกระตุ้นด้วยเดมอน

เมื่อค่าถูกใส่เข้าไปในสล๊อต father_of ของเฟรม A เดมอนที่ผูกติดอยู่กับสล๊อตจะถูกปล่อยออกมาอย่างอัตโนมัติซึ่งจะเพิ่มสล๊อตที่ชื่อว่า son_of ให้กับเฟรม B และใส่ค่าของสล๊อตเป็น

A ให้กับมัน ความรู้ในเรื่องความสัมพันธ์จะถูกห่อหุ้มไว้ในส่วนเดมอนที่ผูกติดอยู่กับสล๊อต status ของเฟรมหลังจากการอ้างของเดมอนที่ผูกติดอยู่กับสล๊อต father_of ของเฟรม A ดังแสดงในรูปที่ 2.12



รูปที่ 2.11 สถานะของเฟรม A และ B ก่อนที่จะถูกกระตุ้นด้วยเดมอน



รูปที่ 2.12 สถานะของเฟรม A และ B หลังจากถูกกระตุ้นด้วยเดมอน

วิศวกรความรู้ผู้ออกแบบฐานความรู้จะผูกเดมอนเข้ากับสล๊อตซึ่งจะถูกกระตุ้นโดยอัตโนมัติเมื่อมีเกิดเหตุการณ์ในการใส่ค่าเข้าไปในสล๊อต ซึ่งแบบจำลองการวินิจฉัยนี้มาจากสามัญสำนึก และผลกระทบเดียวกันสามารถถูกกระทำได้โดยการผูกเดมอนที่ชื่อว่า if_needed กับสล๊อต son_of ของเฟรม B ซึ่งเดมอนนี้จะถูกปลุกขึ้นมาถ้ามีการอ้างถึงสล๊อต son_of ของเฟรม B ซึ่งข้อได้เปรียบของวิธีการนี้คือว่าค่าจะถูกวินิจฉัยและเพิ่มเข้าไปในสล๊อตก็ต่อเมื่อมีการอ้างถึงสล๊อตนั้น เดมอนอีกชนิดหนึ่งคือ if_deleted ซึ่งเดมอนชนิดนี้จะถูกปลุกขึ้นมาถ้าสล๊อตถูกลบออกไปจากเฟรม ซึ่งเดมอนนี้จะมีประโยชน์อย่างมากในปัญหาวิศวกรรมหลายๆปัญหา ซึ่งเนื้องานจะถูกอัพเดตระหว่างกระบวนการแก้ปัญหาที่มาจากการเปลี่ยนแปลงสมมติฐานหรือมาจากค่าที่ถูกคำนวนขึ้นมา ชนิดของการวินิจฉัยที่หลากหลายนี้สามารถถูกจำลองโดยแนวคิดของเดมอน

ระบบผู้เชี่ยวชาญที่ใช้วิธีการแทนความรู้แบบเฟรมส่วนใหญ่นั้น มักจะใช้เทคนิคการสืบทอด (inheritance) ในการวินิจฉัย และแก้ปัญหาบนพื้นฐานของความรู้ที่จัดเก็บไว้ในฐานความรู้ โดยลักษณะการทำงานของกลไกการวินิจฉัยนั้น มีทั้งรูปแบบในการใช้กระบวนการเป็นหลัก ซึ่งเป็นการจัดเก็บกระบวนการต่างๆ ไว้ที่ ค่าของฟาเซท ใน แต่ละสล็อต และจะมีตัวแปลภาษา

(Interpreter) คอย กระทำ กระบวนการ เหล่านั้นอีกครั้ง หรืออาจใช้ กระบวนการ (method) ที่ถูก ผูก ติด (attached) ไว้ใน ฟาเซทของ สล็อตเป็นตัวดำเนินการเองก็ได้

ในงานวิจัยนี้การวินิจฉัยระบบฐานความรู้นั้นจะมีการวินิจฉัย คือ การใช้ การสืบทอดกับอีกวิธีหนึ่งคือการ ใช้การติดต่อสื่อสารระหว่าง เฟรมผ่านทางการใช้ กระบวนการที่ผูกติดอยู่กับสล๊อต ซึ่งส่วนใหญ่ที่ใช้อยู่คือกระบวนการ IF-NEEDED และ IF-CHANGED หรือก็คือการใช้ การส่งผ่านข้อความ (message passing) ระหว่างอินสแตนท์เฟรม [2][14]

2.4 หลักการการโปรแกรมเชิงวัตถุ (Object-oriented Programming)

การเขียนโปรแกรมเชิงวัตถุ (Object-oriented programming, OOP) คือหนึ่งในรูปแบบการเขียนโปรแกรมคอมพิวเตอร์ ที่ให้ความสำคัญกับ วัตถุ ซึ่งสามารถนำมาประกอบกันและนำมาทำงานรวมกันได้โดยการแลกเปลี่ยนข่าวสารเพื่อนำมาประมวลผลและส่งข่าวสารที่ได้ไปให้ วัตถุอื่นๆที่เกี่ยวข้องเพื่อให้ทำงานต่อไป

แนวคิดการเขียนโปรแกรมแบบดังเดิมมักนิยมใช้การเขียนโปรแกรมเชิงกระบวนการ (Procedural Programming) ซึ่งให้ความสำคัญกับขั้นตอนกระบวนการที่ทำ โดยแบ่งโปรแกรมออกเป็นส่วนๆตามลำดับขั้นตอนการทำงาน แต่แนวคิดการเขียนโปรแกรมเชิงวัตถุนั้นให้ความสำคัญกับ ข้อมูล(data) และ พฤติกรรม(behavior) ของวัตถุ และความสัมพันธ์กันระหว่างวัตถุกันมากกว่า

2.4.1 แนวคิดที่สำคัญของการเขียนโปรแกรมเชิงวัตถุ

- *คลาส (Class)* - ประเภทของวัตถุ เป็นการกำหนดว่าวัตถุจะประกอบไปด้วยข้อมูล(data) หรือคุณสมบัติ(property) และพฤติกรรม(behavior)หรือการกระทำ(method) อะไรบ้าง ซึ่งคลาสเป็นโครงสร้างพื้นฐานของการเขียนโปรแกรมเชิงวัตถุ
- *วัตถุ (Object)* - โดยมากจะเรียกว่า อ็อบเจกต์ คือ ตัวตน(instance) ของ คลาส(เช่น นายทักษิณ, นายสนธิ) ซึ่งจะเกิดขึ้นระหว่าง run-time โดยแต่ละอ็อบเจกต์ จะมีข้อมูลเฉพาะของตัวเองทำให้อ็อบเจกต์แต่ละอ็อบเจกต์ของคลาสซึ่งใช้ source code เดียวกันมีคุณลักษณะและคุณสมบัติที่แตกต่างกัน
- *การห่อหุ้ม (Encapsulation)* - การปิดบังข้อมูล เป็นวิธีการกำหนดสิทธิในการเข้าถึงข้อมูล หรือการกระทำกับ อ็อบเจกต์ ของ คลาสนั้นๆ ทำให้แน่ใจได้ว่าข้อมูลของอ็อบเจกต์นั้นจะถูกเปลี่ยนแปลงแก้ไขผ่านทาง methods หรือ properties ที่อนุญาตเท่านั้น (เช่น การกำหนดตำแหน่งทางการเมือง เป็น public method ที่ผู้อื่นสามารกระทำได้ ส่วนการลาออกจากตำแหน่ง เป็น private method ที่มีแต่ อ็อบเจกต์ ของ คลาส เท่านั้นที่จะสามารถทำได้ แต่การกดดันและการขับไล่สามารถสร้าง data ที่อาจจะส่งผลเกิดการลาออกได้เช่นกัน)

• *การสืบทอด (Inheritance)* - การสืบทอดคุณสมบัติ เป็นวิธีการสร้าง คลาสย่อย ที่เรียกว่า ซับคลาส (subclass) ซึ่งจะเป็นกำหนดประเภทของวัตถุให้จำเพาะเจาะจงขึ้น ซึ่ง ซับคลาส จะได้รับถ่ายทอดคุณสมบัติต่างๆมาจากคลาสหลักด้วย (เช่น คลาส มนุษย์ สืบทอดมากจาก คลาส สิ่งมีชีวิต)

• *นามธรรม (Abstraction)* - นามธรรม เป็นการแสดงถึงคุณลักษณะและพฤติกรรมของ object เท่าที่จำเป็นต้องรับรู้และใช้งาน โดยซ่อนส่วนที่เหลือเอาไว้เพื่อไม่ให้เกิดความสับสน เช่น ตามปกติแล้ว นายทักษิณ จัดเป็นตัวตนของ คลาส มนุษย์ ซึ่งจะมีพฤติกรรม การกระทำทุกอย่างที่ตามที่กำหนดไว้ตามโครงสร้างของ คลาส มนุษย์ แต่ในบางกรณีที่นำไปใช้งาน เราไม่ต้องการให้เกิดการสับสนต่อการใช้งานหรือการจัดประเภทมากเราสามารถจัดการหรือใช้งาน อ็อบเจกต์ นายทักษิณ ให้อยู่ในรูปของสิ่งมีชีวิต ก็ได้

• *ภาวะที่มีหลายรูปแบบ (Polymorphism)* - ภาวะที่มีหลายรูปแบบ เป็นวิธีการกำหนดรูปแบบการกระทำที่เหมือนกันแต่ได้ผลที่แตกต่างการ เช่น การเปล่งเสียง เป็น method หลักของคลาส สิ่งมีชีวิต ซึ่งมีคลาส มนุษย์ และคลาสสุนัข เป็น ซับคลาส แต่ผลของการเปล่งเสียงของอ็อบเจกต์จากคลาสทั้งสองจะออกมาไม่เหมือนกัน

2.5 เปรียบเทียบหลักการเชิงวัตถุกับเฟรม

ในปัจจุบันนี้ภาษาทางด้านปัญญาประดิษฐ์ได้มีส่วนขยายทางด้านวัตถุซึ่งสามารถใช้เป็นเครื่องมือทางด้านปัญญาประดิษฐ์ได้ และระบบเฟรมก็มีแนวโน้มที่จะมีคุณลักษณะของการโปรแกรมเชิงวัตถุมากขึ้น ในหัวข้อนี้จะทำการเปรียบเทียบโครงสร้างของเฟรมกับหลักการเชิงวัตถุ ซึ่งข้อแตกต่างระหว่างเฟรมกับวัตถุก็คือ

รูปประโยค (Syntax):

เฟรมนั้นโดยทั่วไปแล้วจะเขียนง่ายกว่าวัตถุ เพราะว่ารูปประโยคของวัตถุจะขึ้นอยู่กับระบบที่ใช้ แต่รูปประโยคของเฟรมจะขึ้นอยู่กับภาษามากกว่าระบบ

การสืบทอด (inheritance):

ในเฟรมล้วนๆนั้นการสืบทอดจะไม่ตายตัว (dynamic) เช่น ค่าของสล๊อตในเฟรมแม่จะไม่ถูกแพร่ลงมา มันคือกลไกการสืบทอดซึ่งจะค้นหาผ่านเส้นทางการสืบทอดแต่ละครั้งที่ต้องการค่าในสล๊อต แต่วัตถุนั้นโดยทั่วไปแล้วอินสแตนท์จะถูกใส่ค่าให้เรียบร้อยแล้วเมื่อมันถูกสร้างขึ้นมาเพื่อเพิ่มประสิทธิภาพ อย่างไรก็ตามเทคนิคผสมได้ถูกใช้ในระบบเฟรมเพื่อหลีกเลี่ยงการเสียเวลาของการสืบทอดแบบไม่ตายตัวแต่ต้องไม่ลืมว่าในระบบเฟรมโดยปกตินั้นจะมีลำดับขั้นตามการเฉพาะเจาะจง เช่น เราจะมีการสืบทอดระหว่าง "วัตถุ" ซึ่งจะเฉพาะเจาะจงด้วยโครงสร้างของมัน แต่

ในระบบเชิงวัตถุนั้นส่วนมากจะอ้างถึงลำดับขั้นตามพฤติกรรมที่คล้ายคลึงกัน เช่น ถ้าเราต้องการที่จะเพิ่มคลาสใหม่เข้ามา เราจะเพิ่มมันเข้าไปกับวัตถุซึ่งมีพฤติกรรมที่คล้ายคลึงกัน (ที่มันกว้างกว่า) แต่ไม่จำเป็นต้องมีโครงสร้างคล้ายคลึงกัน

การห่อหุ้ม (encapsulation):

ในภาษาเชิงวัตถุส่วนใหญ่นั้น ข้อมูลและการควบคุมจะถูกห่อหุ่มไว้ในวัตถุเป็นเสมือนกล่องดำซึ่งมีการคอมไพล์ข้อมูล การเปลี่ยนแปลง และการหาเหตุผลรวมอยู่ในนั้น แต่ในระบบเฟรมจะไม่มีการห่อหุ้ม

แต่ว่ามันก็จริงเพียงส่วนหนึ่งเท่านั้น เพราะระบบเฟรมจะถูกใช้ด้วยการโปรแกรมเชิงการเข้าถึง เช่น เราจะเข้าถึงสล๊อตและกระบวนการอาจจะไม่ถูกใช้ก็ได้ (มันถูกซ่อนไว้) วัตถุจะถูกเข้าถึงด้วยกระบวนการและสล๊อตจะถูกซ่อนไว้ เพราะฉะนั้นเราจะเข้าถึงโครงสร้างของเฟรมและถามถึงพฤติกรรมของวัตถุ

จากที่ได้กล่าวมาแล้วถึงข้อแตกต่างระหว่างทั้งสองหลักการ ซึ่งข้อแตกต่างอยู่ที่จุดประสงค์ของมันซึ่งหลักการเชิงวัตถุจะเป็นเครื่องมือในส่วน softward engineering ในขณะที่เฟรมจะเป็นเครื่องมือทางด้านปัญญาประดิษฐ์สำหรับการแทนความรู้ การเปรียบเทียบเฟรมกับหลักการเชิงวัตถุ เฟรมนั้นจะมีกระบวนการดังนี้

- การเข้าถึงข้อมูลในระดับคลาส : ซึ่งชื่อของสล๊อตและขอบเขต (domain) เป็นหัวใจสำคัญ
- การป้อนค่าของสล๊อตแบบไดนามิก
- การโปรแกรมด้วยการขับเคลื่อนเหตุการณ์ (even-driven programming) โดยใช้เดมอน
- รองรับสล๊อตที่มีค่าเป็นเซต ซึ่งจะถูกใช้ในการแทนความสัมพันธ์ หรือคาดินาลลิตี้มากกว่า 1
- การสร้างคลาสแบบไดนามิก

ทั้งเฟรมและหลักการเชิงวัตถุจะเป็นโครงสร้างแบบลำดับขั้นซึ่งมีการสืบทอดและสล๊อตเดมอน ดังนั้นสล๊อตของเฟรมจะมีเดมอน (และไม่มีกระบวนการ) และหลักการเชิงวัตถุจะมีกระบวนการ (แต่ไม่มีสล๊อตและไม่มีเดมอน)

สรุปเฟรมเป็นแนวคิดทางด้านปรัชญาของการแทนความรู้และการจัดการโดยมนุษย์ ซึ่งเฟรมเป็นแนวคิดโครงร่างซึ่งต้องมีการเติมเต็มด้วยรายละเอียดในการอธิบายวัตถุเฉพาะในภายหลัง ในขณะที่ หลักการเชิงวัตถุเป็นแนวคิดทางการโปรแกรมซึ่งเป็นสัญลักษณ์ของการจัดการโดยโปรแกรม ซึ่งวัตถุจะอ้างถึงโปรแกรมของคลาสซึ่งถูกควบคุมโดยพฤติกรรมที่ได้อธิบายไว้ก่อนแล้วโดยคลาส

ในหลักการเชิงวัตถุนั้นส่วนมากจะสร้างรูปแบบที่เรียกว่า is-a hierarchy ซึ่งการสืบทอดจะเป็นการกระจายวิธีการ(method) และข้อมูลโครงสร้างของวัตถุซึ่งส่วนมากแล้วการกระจายการสืบทอดจะอยู่ในเวลาการกำหนด (definetion time) และเมื่อโปรแกรมถูกสร้างด้วยหลักการเชิงวัตถุถูกรันขึ้นมาการสืบทอดจะไม่ตายตัว

ส่วนในระบบเฟรมนั้นการสืบทอดจะถูกใช้ในการขยายค่าโดยปริยายของสล๊อต (default slot value) โดยทั่วๆไปการสืบทอดจะอยู่ในเวลารัน (run time) ซึ่งจะอนุญาตให้ค่าโดยปริยายสามารถถูกเปลี่ยนแปลงได้ในระหว่างการปฏิบัติการ นอกจากนี้ระบบเฟรมอาจยังอนุญาตการใช้เส้นทางการสืบทอดมากกว่าจะเป็นการใช้ is-a hierarchy

ข้อแตกต่างในการสืบทอดจะถูกเชื่อมโยงกับความแตกต่างในการเข้าถึงของโปรแกรมในการสืบทอดชนิดด้วยซึ่งในหลักการเชิงวัตถุนั้นอินสแตนซ์เท่านั้นที่จะเป็นวัตถุของระบบแต่ในระบบเฟรมนั้นชนิดของเฟรมจะไม่แตกต่างกับอินสแตนซ์ของมันนักดังนั้นจึงสามารถที่จะเข้าถึงชนิดของเฟรมได้โดยตรง

หลักการเชิงวัตถุถูกออกแบบสำหรับการโปรแกรมและลักษณะเชิงวัตถุจะถูกใช้ในการโปรแกรมให้มีประสิทธิภาพ ซึ่งแตกต่างกับในระบบเฟรมที่ถูกออกแบบสำหรับแทนความรู้โดยในส่วนการโปรแกรมนั้นไม่ถูกเน้นให้มีความสำคัญมากนัก

2.6 ระบบการจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์

ก่อนที่จะได้พูดถึงพื้นฐานสถาปัตยกรรมระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ จำเป็นจะต้องทำการเปรียบเทียบระบบฐานข้อมูลเชิงสัมพันธ์ กับระบบฐานข้อมูลเชิงวัตถุ เพื่อให้เห็นความแตกต่างของเทคโนโลยีทั้ง 2 แบบ และเป็นพื้นฐานในการศึกษาระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ ที่เกิดจากการผสมผสานทั้ง 2 เทคโนโลยีเข้าด้วยกัน โดยการเปรียบเทียบจะแบ่งออกเป็น 3 ส่วนคือ โครงสร้างข้อมูล, ภาษาที่ใช้ในการจัดการฐานข้อมูล และกลไกที่ใช้ในการเข้าถึงข้อมูล เพื่อให้เข้าใจถึงหลักการของระบบฐานข้อมูลทั้ง 2 แบบ

2.6.1 ฐานข้อมูลเชิงสัมพันธ์ (Relational Database)

ในการศึกษาฐานข้อมูลเชิงสัมพันธ์นี้จะสามารถแบ่งการศึกษาออกเป็น 3 ส่วนคือ ส่วนโครงสร้างข้อมูล, ส่วนภาษาที่ใช้ในการจัดการฐานข้อมูล และกลไกที่ใช้ในการเข้าถึงข้อมูล

2.6.1.1 โครงสร้างข้อมูล ถ้ากล่าวโดยสรุปแล้ว โครงสร้างข้อมูลของระบบฐานข้อมูลเชิงสัมพันธ์จะจัดเก็บข้อมูลเป็นเซตของรีเลชั่น (Relation) ซึ่ง รีเลชั่น จะประกอบไปด้วยเซตของทูเพิล (Tuple) ซึ่งมีแอตทริบิวต์แบบเดียวกัน และในส่วนชนิดข้อมูลในแต่ละแอตทริบิวต์

จะต้องเหมือนกัน และแต่ละแอตทริบิวต์ จะมีได้เพียงหนึ่งชนิดเท่านั้น และข้อมูลที่ถูกจัดเก็บในแต่ละแอตทริบิวต์ จะมีค่าได้เพียงค่าเดียว (คุณสมบัติ Atomic) [18]

2.6.1.2 ภาษาที่ใช้ในการจัดการฐานข้อมูล ระบบฐานข้อมูลเชิงสัมพันธ์ใช้ภาษา SQL (Structured Query Language) สำหรับการสร้าง (create), เปลี่ยนแปลง (modify), ค้นคืน (retrieve) และ เข้าถึง (manipulate) ข้อมูลในระบบฐานข้อมูลเชิงสัมพันธ์ ภาษา SQL ได้ถูกกำหนดให้เป็นภาษามาตรฐานโดย ANSI (American National Standards Institute) และ ISO (International Organization for Standardization) [19]

2.6.1.3 กลไกที่ใช้ในการเข้าถึงข้อมูล การดำเนินการทั้งหมดในการเข้าถึงข้อมูลจะใช้ค่าของข้อมูลที่ถูกจัดเก็บไว้ใน แต่ละ ทูเพิล โดยที่ไม่สามารถสนับสนุนการอ้างอิงจาก ทูเพิล หนึ่งไปยังอีก ทูเพิล หนึ่งได้ และผลการ คิวรี จะอนุญาตให้เกิดการปฏิบัติการต่างๆ กับข้อมูลได้ครั้งละ 1 ทูเพิล เท่านั้น ซึ่งกลไกนี้จะใช้ทั้งในการทำ คิวรี หรือการ เปลี่ยนแปลงแก้ไขข้อมูล

2.6.2 ฐานข้อมูลเชิงวัตถุ (Object-Oriented Database)

ถึงแม้ว่าฐานข้อมูลเชิงวัตถุนี้จะยังไม่มีมาตรฐานอย่างเป็นทางการ แต่ในทางปฏิบัติแล้วบริษัทผู้พัฒนาซอฟท์แวร์ระบบฐานข้อมูลเชิงวัตถุได้อ้างอิงเอามาตรฐานที่ Morgan Kaufmann ได้ตีพิมพ์ไว้ ซึ่งการจัดทำมาตรฐานนี้ได้รับการสนับสนุนจาก Object Database Management Group (ODMG) มาตรฐานดังกล่าวที่ใช้คือ ODMG-V2.0 [20] โดยที่เนื้อหาของมาตรฐานจะเน้นหนักไปในเรื่องของความสัมพันธ์ระหว่างวัตถุ ที่จะสามารถสนับสนุนการเขียนโปรแกรมด้วยภาษาเชิงวัตถุได้ รวมถึงการจัดเก็บวัตถุต่างๆ ในฐานข้อมูล

2.6.2.1 โครงสร้างข้อมูล การใช้งานข้อมูลในฐานข้อมูลเชิงวัตถุจะอยู่ในรูปแบบของ คลาส ซึ่งเป็นแนวความคิดพื้นฐานของทฤษฎีทางวัตถุโดยที่ในแต่ละ คลาส จะประกอบด้วย แอททริบิวต์ (attribute) และกระบวนการ (method) รวมไปถึงเงื่อนไขความถูกต้อง (Integrity Constrain) ของ คลาส นั้นๆ ซึ่งจะมีการกำหนด Object Identifier (OID) เอาไว้ เพื่อแบ่งแยกความแตกต่างระหว่าง คลาส นอกจากนี้ยังสนับสนุนการทำงานแบบ ห่อหุ้มข้อมูล (encapsulation) และ สืบทอด (inheritance) ด้วย สำหรับชนิดของข้อมูลนั้นสามารถรองรับชนิดข้อมูลแบบ Abstract Data Type ได้ จากมาตรฐานของ ODMG-V2.0 [20]

เนื่องจากฐานข้อมูลเชิงวัตถุนี้เกิดขึ้นจากการรวมเอาพื้นฐานแนวความคิดเชิงวัตถุ กับหลักการด้านภาษาสำหรับการโปรแกรมเชิงวัตถุ และความสามารถด้านฐานข้อมูลเข้าด้วยกัน ผลลัพธ์ที่ได้คือทำให้เกิดความเข้ากันได้เป็นอย่างดีระหว่างโครงสร้างข้อมูลภายในฐานข้อมูล กับโครงสร้างข้อมูลที่ใช้ในการพัฒนาโปรแกรม ซึ่งอำนวยความสะดวกในการใช้ภาษาสำหรับการ

โปรแกรมเชิงวัตถุต่างๆ เพื่อพัฒนาโปรแกรมที่ใช้งานร่วมกับฐานข้อมูลเชิงวัตถุ ทำให้สามารถลดจำนวนโค้ดที่ใช้ในการเขียนโปรแกรมต่างๆลงได้

2.6.2.2 ภาษาที่ใช้ในการจัดการฐานข้อมูล ระบบฐานข้อมูลเชิงวัตถุจะใช้ภาษาสำหรับการโปรแกรมเชิงวัตถุ ทั้งในการจัดการฐานข้อมูล และการพัฒนาโปรแกรม (เช่น C++, SmallTalk, Java) เนื่อง วัตถุในแอบพริเคชั่น (Application Object) กับ วัตถุที่ถูกจัดเก็บ (Stored Object) ในระบบฐานข้อมูลชนิดนี้มีความสัมพันธ์กันโดยตรง ดังนั้นการจัดการกับฐานข้อมูล และการคิวรี จึงสามารถทำได้ด้วยภาษาสำหรับการโปรแกรมเชิงวัตถุดังกล่าว นอกจากนั้นแล้ว ตามมาตรฐาน ODMG-93 ได้มีการกำหนดภาษา ขึ้นมาอีกภาษาหนึ่งคือ OQL (Object Query Language) อย่างไรก็ดีในมาตรฐานใหม่ของ SQL 3 จะได้มีบางส่วนที่จะได้เพิ่มความสามารถเหล่านี้ลงไป ซึ่งผลจากการพัฒนา OQL จะสามารถสร้างความหลากหลายของผลลัพธ์ที่เกิดขึ้นจากการทำคิวรีได้ อันได้แก่ ผลลัพธ์ที่อยู่ในรูปของ อะตอม, โครงสร้าง, วัตถุ หรือแม้แต่ เซตของวัตถุ [20]

2.6.2.3 กลไกที่ใช้ในการเข้าถึงข้อมูล การสร้าง และ แก้ไขข้อมูลในระบบฐานข้อมูลเชิงวัตถุจะใช้การเข้าถึงโดยตรงจากภาษาสำหรับการโปรแกรมเชิงวัตถุในลักษณะของภาษาหลัก (Native Language) สำหรับวัตถุ ที่ถูกสร้างขึ้นในระบบฐานข้อมูลเชิงวัตถุนี้จะถูกกำหนด OID ให้โดยอัตโนมัติ และ OID นี้จะมีความเฉพาะ (Unique) ไปตลอดอายุการใช้งานวัตถุ นั้นๆ นอกจากนี้ในตัว วัตถุแต่ละ วัตถุยังสามารถจัดเก็บ OID ของ วัตถุตัวอื่น เพื่อสร้างการอ้างอิงแบบลอจิก (Logical Reference) ซึ่งตัวอ้างอิงนี้เอง จะเป็นประโยชน์ในการสร้างความสัมพันธ์ระหว่างวัตถุที่เกิดขึ้นตามโลกแห่งความเป็นจริงที่สามารถแบ่งตามกฎเกณฑ์ได้

จะเห็นได้จากการเปรียบเทียบดังกล่าวว่า ระบบฐานข้อมูลทั้ง 2 แบบข้างต้น มีความแตกต่างกันอย่างมาก ทั้งทางโครงสร้างข้อมูล ที่เกิดจากแนวความคิดที่ต่างกัน รวมถึงภาษาที่ใช้ในการจัดการฐานข้อมูล และกลไกที่ใช้ในการเข้าถึงข้อมูล ที่ทำให้เกิดผลลัพธ์ในการใช้งานข้อมูลที่ต่างกัน

2.6.3 ฐานข้อมูลเชิงวัตถุสัมพันธ์ (Object-Relational Database)

เป็นที่ทราบกันดีอยู่แล้วว่าการพัฒนาโปรแกรมในยุคปัจจุบัน มีแนวโน้มไปทางการโปรแกรมเชิงวัตถุมากขึ้น ทำให้เกิดความต้องการในการใช้ข้อมูลที่มีความซับซ้อนมากขึ้นตามไปด้วย ซึ่งระบบฐานข้อมูลเชิงสัมพันธ์แบบเดิมไม่สามารถรองรับได้เต็มที่ จึงได้เกิดแนวคิดที่จะมีการเพิ่มความสามารถในการจัดการข้อมูลในรูปแบบวัตถุให้แก่ระบบฐานข้อมูลเชิงสัมพันธ์ ระบบฐานข้อมูลใหม่นี้เรียกว่า ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ [19][21] ซึ่งได้รับความสนใจเป็นอย่างมาก จนทำให้เกิดแนวคิดและวิธีการที่หลากหลายในการพัฒนาระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ขึ้น [21] โดยเฉพาะเมื่อ Michael Stonebraker ได้เขียนหนังสือชื่อ Object-Relational DBMSs, The Next

Great Wave. ที่ถูกตีพิมพ์โดยสำนักพิมพ์ Morgan Kaufman และได้ร่วมกับทีมวิจัยของตนพัฒนาระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ตัวอย่างขึ้นชื่อ Postgres (ซึ่งก็คือ Informix ในปัจจุบัน) [21] จนทำให้บริษัทผู้ผลิตซอฟท์แวร์ระบบการจัดการฐานข้อมูลอื่นๆ เช่น IBM, Unisys, Oracle และ UniSQL สนใจที่จะนำแนวความคิดดังกล่าวมาพัฒนาซอฟท์แวร์ระบบการจัดการฐานข้อมูลของตน โดยแนวทางในการพัฒนาระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ หรือการเพิ่มความสามารถในการจัดการข้อมูลแบบ วัตถุให้แก่ระบบฐานข้อมูลเชิงสัมพันธ์

ต่อมาในภายหลัง ANSI จึงได้เริ่มดำเนินการเพื่อเพิ่มมาตรฐานด้านเชิงวัตถุ ให้แก่มาตรฐาน SQL ที่มีอยู่ โดยการดำเนินการดังกล่าวอยู่ในความรับผิดชอบของทีมพัฒนาชุด X3H2 ของ ANSI ซึ่งได้ใช้วิธีการเพิ่มส่วนขยายเชิงวัตถุ (Object Extension) ให้แก่ระบบฐานข้อมูลเชิงสัมพันธ์โดยตรงเป็นแนวทางในการพัฒนามาตรฐานไปสู่ SQL 3 (SQL-1999) [19]

2.6.3.1 มาตรฐาน SQL 3 เพื่อให้สามารถเปรียบเทียบความแตกต่างของระบบฐานข้อมูล เชิงวัตถุสัมพันธ์ กับระบบฐานข้อมูลเชิงสัมพันธ์ และระบบฐานข้อมูลเชิงวัตถุ จึงได้สรุปคุณสมบัติของระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ (ตามมาตรฐาน SQL 3) ออกเป็น 3 ส่วน ดังนี้

2.6.3.1.1 โครงสร้างข้อมูล

ดังที่กล่าวมาแล้วว่าการพัฒนามาตรฐาน SQL 3 เพื่อรองรับการใช้งานข้อมูลประเภทวัตถุของระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ ใช้วิธีการเพิ่ม ส่วนขยายเชิงวัตถุ ให้แก่ระบบฐานข้อมูลเชิงสัมพันธ์ ซึ่งการดำเนินการในส่วนนี้ถือเป็นหัวใจของมาตรฐาน SQL 3 โดยการเพิ่ม ส่วนขยายเชิงวัตถุดังกล่าวคือ การเพิ่มชนิดข้อมูลแบบใหม่ให้แก่ระบบฐานข้อมูลเชิงสัมพันธ์ โดยที่โครงสร้างการจัดเก็บข้อมูลยังถูกแสดงอยู่ในรูปแบบของตารางเช่นเดิม ชนิดข้อมูลแบบใหม่นี้ถูกเรียกว่า Abstract Data Type (ADT) ซึ่งจะมีลักษณะคล้าย คลาส ในหลักการโปรแกรมเชิงวัตถุ กล่าวคือ ADT จะถูกใช้เพื่อรวมกลุ่มของข้อมูลที่มีความสัมพันธ์กันทั้งในส่วนของคุณสมบัติของข้อมูล และสถานการณ์ที่จะเกิดขึ้นกับข้อมูลเหล่านั้น โดยที่ชนิดข้อมูลแบบ ADT นี้ จะสนับสนุนคุณสมบัติ การห้อหุ้ม หรือการมีความสัมพันธ์กับ ADT กลุ่มอื่นในรูปแบบของ Suptype-Supertype ทั้งยังมีคุณสมบัติสืบทอดในการถ่ายทอดคุณสมบัติของ แอททริบิวต์มาจาก ADT ที่เป็น Supertype ของตนด้วย นอกจากนี้แล้วยังสามารถกำหนด การดำเนินการ (Operation) และ คำสั่ง (Function) ที่มีความสัมพันธ์กับข้อมูล เพื่อใช้ใน การสร้างและการใช้งานอินเดกซ์, จับเก็บและค้นคืนข้อมูล ตามคุณลักษณะพิเศษของข้อมูลแต่ละประเภทได้ เช่น ข้อมูลที่เป็นมัลติมีเดียต่างๆ ที่จะต้องมีวิธีการ ค้นคืน ที่แตกต่างออกไปจากปกติเป็นต้น ซึ่ง การดำเนินการ และ คำสั่ง ของ ADT เหล่านี้สามารถนำมาใช้ ร่วมกับการค้นหา (Search Predicate) ในการทำคิวรีข้อมูลได้

สำหรับชนิดข้อมูล ที่ ADT สนับสนุนมีทั้ง Row Type (หรือชนิดของแถวที่อยู่ในตาราง), Distinct Type (หรือการสร้าง User-Defined Type จากชนิดที่ต่างกัน ที่ระบบมีอยู่) และ การสร้างต้นแบบชนิดข้อมูล (Type Template) ขณะเดียวกันยังมีการเตรียมเครื่องมือภายในระบบเอาไว้สนับสนุนการสร้างชนิดข้อมูลในรูปแบบ Collection Type จำพวก LIST หรือ SET อีกด้วย ซึ่งจะเห็นได้ว่า ADT จะสามารถรองรับการใช้งานข้อมูลที่มีโครงสร้างซับซ้อนมากให่ดีกว่าเดิมได้ [19]

2.6.3.1.2 ภาษาที่ใช้ในการจัดการข้อมูล

เพื่อสนับสนุนการใช้งานข้อมูล ADT ในระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ จึงได้มีการเพิ่มเติมความสามารถในการเข้าถึงข้อมูลดังกล่าวให้แก่ภาษา SQL ประเด็นสำคัญของส่วนที่เพิ่มเติมใหม่นี้คือการสนับสนุนการใช้งานข้อมูลในรูปแบบ วัตถุที่รวมการจัดการข้อมูลแบบ Nested Object, set-valued Attribute, ความสามารถในการใช้ การดำเนินการ หรือคำสั่ง ของ วัตถุร่วมกับการค้นหาในการคิวรีข้อมูลแบบ ADT ได้ โดยยังยึดเอา SQL เป็นภาษาหลักในการทำจัดการฐานข้อมุลและการทำคิวรี สำหรับผลของการทำคิวรี นั้น ข้อมูลจะยังถูกแสดงในรูปแบบของตารางอยู่ อย่างไรก็ดีในอนาคตได้มีการวางแผนที่จะพัฒนามาตรฐานใหม่เป็น SQL 4 ที่สามารถแสดงผลการทำคิวรี ให้อยู่ในรูปของ วัตถุด้วยการผสมผสานเทคนิคของ OQL เข้ามา หรือการใช้เทคนิคที่ใกล้เคียง

นอกจากนี้มาตรฐาน SQL 3 ยังมีส่วนของการเพิ่ม Procedural Extension (SQL/PSM) ซึ่งการดำเนินการดังกล่าวได้ครอบคลุมไปถึงการพัฒนารูปแบบคำสั่ง SQL ที่ใช้ในการทำ Stored Procedure และ User-Defined Function เพื่อใช้งานร่วมกับฐานข้อมูลด้วย กล่าวคือ กระบวนการ และ คำสั่ง เหล่านี้จะสามารถสร้างขึ้นได้จากภาษา SQL เอง ซึ่งมีการพัฒนารูปแบบคำสั่งใหม่ที่ใกล้เคียงกับภาษาสำหรับการโปรแกรมทั่วไปในยุคปัจจุบัน ได้แก่ความสามารถในการใช้คำสั่งเงื่อนไขประเภท IF/THEN/ELSE, การใช้ลูป แบบ WHILE และ DO/UNTIL หรือการใช้ CASE เป็นต้น ทั้งยังให้ โปรแกรมเมอร์ สามารถสร้างฟังก์ชั่นไลบราลี ด้วยภาษาอื่น เพื่อใช้งานร่วมกับภาษา SQL ที่มีอยู่ได้ ซึ่งจะเพิ่มขีดความสามารถในการทำคิวรีได้ดียิ่งขึ้น

อีกส่วนหนึ่งที่มีความสำคัญในมาตรฐาน SQL 3 ก็คือส่วนขยายเชิงสัมพันธ์ (relational extension) ซึ่งจะทำให้โปรแกรมเมอร์สามารถสร้าง Recursive Query, สนับสนุน Query Expression ที่สามารถใช้งานร่วมกันได้ รวมไปถึงการพัฒนามุมมอง (View) ให้ดีขึ้นจากมาตรฐานเดิม ซึ่งจะสามารถสนับสนุนระบบการทำงาน ในส่วนของ Trigger และ Integrity Constrain ได้

เนื่องจากภาษาหลักที่ใช้ในการจัดการฐานข้อมูลยังคงเป็น SQL และระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ได้ถูกพัฒนาขึ้นมาจากระบบฐานข้อมูลเชิงสัมพันธ์ ดังนั้นกลไกในการเข้าถึงข้อมูลส่วนใหญ่จะยังคงเป็นเช่นเดียวกับระบบฐานข้อมูลเชิงสัมพันธ์อยู่ ส่วนการเพิ่มความสามารถในการ

เข้าถึงข้อมูลประเภท ADT หรือการเข้าถึงข้อมูลในรูปแบบ วัตถุนั้นยังมีปัญหาเมื่อต้องการเข้าถึงโดยตรงจากภาษาการโปรแกรมเชิงวัตถุ ทำให้ยังจำเป็นที่จะต้องมีการแปลงรูปให้กลับมาอยู่ในลักษณะตารางเสียก่อน

2.7 ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ในเชิงพาณิชย์

ปัจจุบันมีบริษัทผู้ผลิตซอฟท์แวร์ระบบการจัดการฐานข้อมูลจำนวนมากที่หันมาให้ความสนใจกับระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ โดยเฉพาะเมื่อบริษัทที่เป็นผู้นำในกลุ่มได้เริ่มพัฒนาผลิตภัณฑ์ของตน เพื่อการรองรับการใช้งานข้อมูลที่เป็น วัตถุตามกระแสความนิยมของแนวคิดเชิงวัตถุมากขึ้น ในขณะที่มาตรฐาน SQL 3 ยังไม่เสร็จสมบูรณ์ ทำให้การดำเนินการพัฒนาระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ของแต่ละบริษัทจะมความแตกต่างกันออกไป ซึ่งบางทีอาจทำให้เกิดความสับสนในรูปแบบของเทคโนโลยีของแต่ละผลิตภัณฑ์ ดังนั้นงานวิจัยนี้จึงได้ยกตัวอย่างผลิตภัณฑ์ซอฟท์แวร์ระบบการจัดการฐานข้อมูล ในกลุ่มที่เป็นระบบฐานข้อมูลเชิงวัตถุสัมพันธ์เพื่อมาศึกษาเปรียบเทียบระหว่างผลิตภัณฑ์ต่างๆ เหล่านั้น กับมาตรฐาน SQL 3 ที่กำหนดขึ้น โดยผลิตภัณฑ์ที่เลือกมา คือออราเคิล ซึ่งถือว่าเป็นซอฟท์แวร์ระบบการจัดการฐานข้อมูล ที่ได้รับความนิยม และการยอมรับอย่างกว้างขวาง ในท้องตลาดผลิตภัณฑ์ซอฟท์แวร์

2.7.1 ผลิตภัณฑ์ของออราเคิล

ออราเคิลเป็นอีกผลิตภัณฑ์หนึ่งที่มีการพัฒนาระบบการจัดการฐานข้อมูลของตนเพื่อรองรับการขยายตัวในการใช้งานที่ซับซ้อนขึ้น [7][21] โดยเฉพาะข้อมูลแบบ วัตถุซึ่ง ออราเคิลสนับสนุนการทำงานดังกล่าว โดยการเพิ่มชนิดข้อมูลแบบ วัตถุเข้ามาในระบบฐานข้อมูล ซึ่ง ข้อมูลแบบวัตถุ (Object Type) นี้จะใช้ในการกำหนดชนิดของข้อมูลแบบใหม่ที่แตกต่างไปจากเดิมที่มีอยู่ รวมทั้งสามารถสร้างกระบวนการ สำหรับการทำงานกับ วัตถุต่างๆ ที่สร้างขึ้นได้ โดย Collection ของ วัตถุจะสามารถแสดงอยู่ได้ทั้งในรูปของโครงสร้างแบบ Array หรือ Nested Table สำหรับข้อมูลแบบวัตถุ ของออราเคิล จะสามารถเทียบได้กับ Row Type ของมาตรฐาน SQL 3 กล่าวคือข้อมูลแบบวัตถุนั้นสามารถเป็นได้ทั้งระดับ Column Type และ Table Type โดยที่ออราเคิล ยังคงใช้ภาษา PL/SQL เป็นที่มีการพัฒนาขึ้นตามมาตรฐาน SQL 3 เป็นภาษาหลักในการจัดการ และ คิวรีข้อมูลอยู่เช่นเดิม

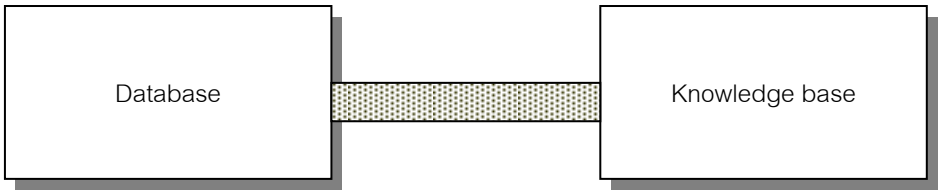
2.8 การบูรณาการระบบฐานความรู้กับระบบฐานข้อมูล

จากที่ได้กล่าวถึงโครงสร้าง และเทคนิคในการแทนความรู้แบบเฟรม ตลอดจนข้อขัดข้อง และข้อจำกัดต่างๆ ในการพัฒนาระบบผู้เชี่ยวชาญที่จำเป็นต้องใช้เครื่องมือเฉพาะ ด้วยสาเหตุนี้เองทำให้มีนักวิจัยจำนวนหนึ่ง ต้องการที่จะลดข้อจำกัด และจัดการกับปัญหาที่เกิดขึ้นเหล่านี้ โดยความพยายามนำระบบฐานข้อมูลเข้ามาช่วยเหลือ ซึ่งจะได้กล่าวถึงแนวการบูรณาการระบบฐานความรู้กับระบบฐานข้อมูล เพื่อช่วยเพิ่มความสามารถของระบบฐานความรู้สำหรับระบบผู้เชี่ยวชาญ และจะได้ยกตัวอย่างงานวิจัยที่เกี่ยวข้องบางส่วนมาอธิบายพอสังเขปดังนี้ [22]

2.8.1 สถาปัตยกรรมการบูรณาการระบบฐานความรู้กับระบบฐานข้อมูล

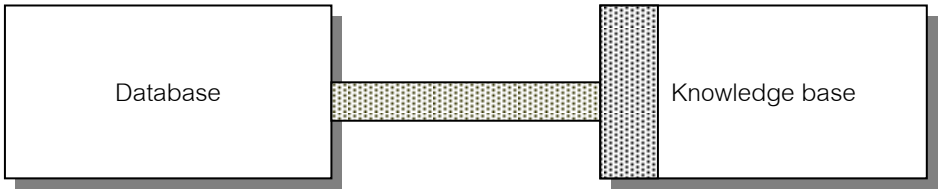
ในการประยุกต์ใช้ระบบฐานข้อมูลร่วมกับระบบฐานความรู้ สำหรับระบบผู้เชี่ยวชาญนั้น มีแนวทางการดำเนินการอยู่หลายลักษณะ บางระบบก็เป็นเพียงการใช้ฐานข้อมูลอย่างเรียบง่ายเพื่อเก็บความรู้แทนการใช้ไฟล์ข้อมูลปกติเท่านั้น ซึ่งจากแนวทางต่างๆ สามารถจำแนกประเภทสถาปัตยกรรมในการพัฒนาระบบฐานความรู้ด้วยการใช้ระบบฐานข้อมูลได้ 5 รูปแบบดังนี้

2.8.1.1 การสร้างการเชื่อมต่อระหว่างฐานข้อมูล และฐานความรู้ ในรูปแบบนี้นักพัฒนาระบบจะสร้างระบบฐานข้อมูลเพื่อเก็บข้อมูลเบื้องต้น ที่ใช้ในการดำเนินระบบงานทั่วไปไว้ต่างหาก และสร้างการเชื่อมต่อในการดึงข้อมูลจากระบบฐานข้อมูลไปสร้างระบบฐานความรู้อีกส่วนหนึ่ง เพื่อใช้สำหรับการวินิจฉัยระบบผู้เชี่ยวชาญโดยเฉพาะ สำหรับการเชื่อมต่อนั้นอาจเป็นการสร้างฟังก์ชันพิเศษขึ้นเอง หรือการใช้เทคโนโลยีจำพวก ODBC ก็ได้ ในบางครั้งเราอาจเรียกวิธีนี้ว่าการเชื่อมต่อแบบหลวมๆ (Loose Coupling) ดังจะเห็นตามลักษณะดังรูปที่ 2.13



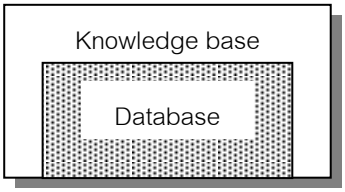
รูปที่ 2.13 เทคนิคแบบการสร้างการเชื่อมต่อ

2.8.1.2 สร้างส่วนเชื่อมประสานภายในฐานความรู้ไปยังฐานข้อมูล ในลักษณะการสร้างส่วนเชื่อมประสานภายในฐานความรู้ไปยังฐานข้อมูลนั้น จะเป็นการสร้างฟังก์ชันการเชื่อมประสานไว้ภายในระบบฐานความรู้ มีหน้าที่คอยเข้าถึงข้อมูลที่ต้องการภายในระบบฐานข้อมูลที่ใช้ ดังรูปที่ 2.14



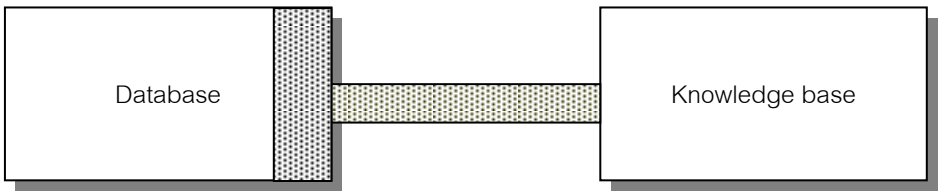
รูปที่ 2.14 เทคนิคแบบการสร้างส่วนเชื่อมประสานภายในฐานความรู้ไปยังฐานข้อมูล

2.8.1.3 สร้างฟังก์ชันงานฐานข้อมูลภายในฐานความรู้ วิธีแบบการสร้างฟังก์ชันงานฐานข้อมูลไว้ภายในฐานความรู้ดังที่เห็นตามแผนภาพในรูปที่ 2.15 เป็นวิธีที่มีความซับซ้อนในการสร้าง เนื่องจากฟังก์ชันงานในการบริหารจัดการข้อมูลในรูปแบบฐานข้อมูลนั้นมีความซับซ้อน ทำให้การใช้ระบบฐานความรู้มาสร้างฟังก์ชันดังกล่าวจะเกิดความซับซ้อนมากขึ้นไปอีก เพราะปกติระบบฐานความรู้ จะมิได้ให้ความสำคัญกับวิธีการจัดเก็บ และบริหารจัดการข้อมูลมากนัก



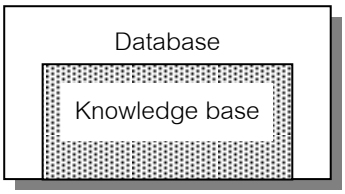
รูปที่ 2.15 เทคนิคแบบการสร้างฟังก์ชันงานฐานข้อมูลภายในฐานความรู้

2.8.1.4 สร้างส่วนเชื่อมประสานภายในฐานข้อมูลไปยังฐานความรู้ วิธีนี้เป็นวิธีที่คล้ายกับข้อ 2.2.1.2 แต่ต่างกันตรงที่ฟังก์ชันการเชื่อมประสานข้อมูลจะอยู่ฝั่งฐานข้อมูล ดังรูปที่ 2.16



รูปที่ 2.16 เทคนิคแบบการสร้างสร้างส่วนเชื่อมประสานภายในฐานข้อมูลไปยังฐานความรู้

2.8.1.5 สร้างฟังก์ชันงานฐานความรู้ภายในฐานข้อมูล เป็นเทคนิคที่ได้รับความนิยมอีกรูปแบบหนึ่ง ด้วยการใช้ระบบฐานข้อมูลที่จัดเก็บข้อมูลที่ต้องการอยู่แล้ว และสร้างฟังก์ชันงาน หรือทำการสร้างตัวแทนความรู้แบบที่ต้องการลงไป โดยอาศัยความสามารถในด้านการจัดการ และบริหารข้อมูลของระบบฐานข้อมูลเป็นหลัก วิธีนี้นอกจากจะมีความสะดวก เพียงการกำหนดการสร้างตัวแทนความรู้ที่เหมาะสมบนฐานข้อมูลแล้ว ยังสะดวกในเรื่องการจัดใช้งานเครื่องมือ เนื่องจากระบบฐานข้อมูลมีความสามารถในการดำเนินการ ทั้งการจัดเก็บ, การค้นคืน และการเข้าถึงข้อมูลอยู่แล้ว ดังแสดงให้เห็นในรูปที่ 2.17



รูปที่ 2.17 เทคนิคแบบการสร้างฟังก์ชันงานฐานความรู้ภายในฐานข้อมูล

2.8.2 งานวิจัยที่ใช้การบูรณาการระบบฐานความรู้กับระบบฐานข้อมูล

โดยส่วนมากแล้วในการพัฒนาฐานความรู้โดยการบูรณาการระบบฐานความรู้กับระบบฐานข้อมูลนั้นระบบงานที่เลือกใช้เทคนิคการแทนความรู้แบบเฟรมที่พบ กลับเลือกที่จะใช้ระบบฐานข้อมูลเชิงสัมพันธ์เพื่อทำการจัดเก็บความรู้ ทั้งในรูปแบบที่พยายามแมปโครงสร้างข้อมูลของเฟรม ให้มาอยู่ในรูปเชิงสัมพันธ์โดยตรง [5] หรือการแมปโครงสร้างของเฟรม ให้มาอยู่ในโครงสร้างเชิงวัตถุเสียก่อน แล้วจึงค่อยแมปจากโครงสร้างเชิงวัตถุให้มาอยู่ในรูปเชิงสัมพันธ์ [23] ด้วยจุดประสงค์ที่ต้องการรักษาความหมาย (Semantic) ของความรู้ไว้ให้มากที่สุด โดยจำเป็นจะต้องเลือกโมเดลเชิงวัตถุที่มีความสามารถในการอำนวยความสะดวกในการแมปโครงสร้างเชิงวัตถุให้มาอยู่ในรูปเชิงสัมพันธ์ เพื่อลดความยุ่งยากในการดำเนินงาน

การเชื่อมโยงกันระหว่างระบบฐานความรู้กับระบบฐานข้อมูลเป็นการขยายความสามารถของระบบฐานความรู้ให้สามารถรองรับฐานความรู้ขนาดใหญ่ได้ [4] ซึ่งในปัจจุบันได้มีเครื่องมือที่ใช้เชื่อมต่อทั้งสองระบบเข้าด้วยกัน อย่างเช่น Perk [3], EcoCyc [4], Sophia [5] และ PARKA-DB [24]

2.8.2.1 Perk

ระบบนี้จะทำงานในลักษณะเป็น backend ของระบบเฟรมที่ชื่อว่า Ocelot ภายใต้รูปแบบ OKBC โดยจะจัดเก็บข้อเท็จจริงไว้ที่ฐานข้อมูล แล้วใช้คำสั่งรูปแบบ OKBC เพื่อเข้าถึง

เฟรมที่ต้องการ โดยการสร้างมุมมองเชิงวัตถุ ขึ้นมา เพื่อเรียกเฟรมที่เกี่ยวข้อง ขึ้นมาในลักษณะของเฟรม แล้วนำเฟรมที่ได้มาทำการวินิจฉัย ที่ Ocelot ซึ่งอยู่บนฝั่งไคล์เอนท์ อีกที

2.8.2.2 EcoCyc

เป็นเครื่องมือที่ใช้ในการสร้างระบบฐานความรู้ และใช้ระบบฐานความรู้เชิงสัมพันธ์เป็นส่วนจัดเก็บข้อเท็จจริง ระบบนี้จะใช้กลไลการจัดเก็บแบบฝั่งแน่น (persistent storage mechanisim) ซึ่งเฟรมจะถูกจัดเก็บอยู่ในระบบฐานข้อมูลและถูกบีบอัดให้เป็นข้อความ ASCII ก่อน จากนั้นเมื่อมีการเข้าถึงเฟรมจะทำการแปลงกลับให้เป็นเฟรมและจะถูก paged in เข้าไปยังหน่วยความจำอีกที

2.8.2.3 Sophia

จะจัดเก็บการแทนความรู้แบบเฟรม บนฐานข้อมูลเชิงวัตถุสัมพันธ์ โดยใช้ MS-Acess97 ในการจัดเก็บความรู้แบบเฟรม และจะทำการค้นคืนค่าขึ้นมาโดยใช้ คำสั่ง SQL ในการโหลดเฟรมภายใต้รูปแบบ OKBC

2.8.2.4 PARKA-DB

จะเก็บการแทนความรู้แบบเฟรม โดยใช้ระบบการจัดการฐานข้อมูลเชิงสัมพันธ์ และจะโหลดเข้าไปในหน่วยความจำหลักอีกที

ซึ่งจากระบบที่ได้ทำการศึกษามาข้างต้น ระบบดังกล่าวไม่มีกลไกการวินิจฉัยที่ฝั่งฐานข้อมูลที่จัดเก็บฐานความรู้เลย แต่จะเป็นในลักษณะจัดเก็บข้อเท็จจริง หรือเฟรม ไว้ที่ฐานข้อมูล แล้วทำการโหลดข้อเท็จจริงขึ้นมาทำการวินิจฉัยอีกที

2.9 การติดต่อสื่อสารระหว่างระบบฐานความรู้

ในการติดต่อสื่อสารเพื่อแลกเปลี่ยนความรู้ระหว่างระบบฐานความรู้ได้นั้นจำเป็นต้องมีภาษาและโปรโตคอลที่ใช้ในการติดต่อสื่อสารระหว่างระบบเพื่อเป็นมาตรฐานในการแลกเปลี่ยนความรู้ระหว่างระบบฐานความรู้ ซึ่งภาษาและโปรโตคอลที่เป็นที่นิยมและถูกใช้เป็นภาษามาตรฐานในการติดต่อสื่อสารระหว่างระบบฐานความรู้คือภาษา KQML [6]

2.9.1 รูปแบบภาษา KQML (KQML String Syntax)

KQML หรือ Knowledge Query and Manipulation Language เป็นภาษาและโปรโตคอลสำหรับการติดต่อสื่อสารระหว่างซอฟต์แวร์เอเจ้นท์และระบบฐานความรู้ ซึ่งถูกพัฒนาขึ้นมาในปี 1994 โดย Tim Finin [6] โดยเป็นส่วนหนึ่งของ ARPA Knowledge Sharing Effort ซึ่งมีจุดมุ่งหมายในการพัฒนาเทคนิคสำหรับสร้างฐานความรู้ขนาดใหญ่ที่สามารถแชร์ความรู้และสามารถนำความรู้

กลับมาใช้ใหม่ได้ซึ่งแนวคิดแรกเริ่มคือเป็นส่วนติดต่อของระบบฐานความรู้และในภายหลังได้ถูกนำเสนอใหม่เป็นภาษาที่ใช้ในการติดต่อสื่อสารระหว่างเอเจ้นท์

รูปแบบข้อความ KQML (KQML message) และโปรโตคอลสามารถถูกใช้เพื่อติดต่อกับระบบอัจฉริยะ (intelligent system) ได้ ซึ่งอาจจะติดต่อกันโดยใช้โปรแกรมหรือโดยระบบอัจฉริยะตัวอื่นๆ ในข้อความ KQML นั้น "performatives" จะเป็นส่วนบ่งบอกการดำเนินการบนฐานความรู้อื่นๆ

BNF ในรูปที่ 2.18 [6] กำหนดนิยามสำหรับ <ascii>, <alphabetic>, <numberic>, <double-quote>, <backslash>, และ <whitespace> ในนิยามนี้ “*” หมายถึงจำนวนใดๆของเหตุการณ์ที่เกิดขึ้น และ “-“ หมายถึง set difference หมายเหตุ <performative> เป็น specialization ของ <expression>

```
<performative> ::= (<word> {<whitespace> :<word> <whitespace>
<expression>}*)

<expression> ::= <word> | <quotation> | <string> |
   (<word> {<whitespace> <expression>}*)

<word> ::= <character><character>*

<character> ::= <alphabetic> | <numeric> | <special>

<special> ::=    <     | > | = | + | - | * | / | & | ^ | ~ | _ |
                 @     | $ | % | : | . | ! | ?

<quotation> ::= ‘<expression> | ‘<common-expression>

<comma-expression> ::= <word> | <quotation> | <string> | ,<comma-expression>
                           (<word> {<whitespace> <comma-expression>}*)

<string> ::= “<stringchar>*” | #<digit><digit>*”<ascii>*

<stringchar> ::= \<ascii> | <ascii>-\-<double-qoute>
```

รูปที่ 2.18 KQML string syntax ในรูปแบบ BNF

2.9.2 Reserved Performative Parameters

performative จะรับ parameter ที่ถูกระบุถึงโดย keyword ดังตารางที่ 2.1 [6]

:sender <word>

:receiver <word>

Parameter พวกนี้แสดง ผู้ส่งและผู้รับจริงๆ ของ performative ซึ่งแตกต่างจาก ผู้รับและผู้ส่ง แบบ virtual ใน :from และ :to parameter ของ networking performative

ตารางที่ 2.1 สรุปค่าพารามิเตอร์ที่เป็นคีย์เวิร์ด และความหมายของมัน

Keyword	*Meaning*
:content	The information about which the performative expresses an attitude
:force	Whether the sender will ever deny the meaning of the performative
:in-reply-to	The expected label in a reply
:language	The name of representation language of the :content parameter
:ontology	The name of the ontology (e.g., set of term definitions) used in the **:content** parameter
:receiver	The actual receiver of the performative
:reply-with	Whether the sender expects a reply, and if so, a label for the reply
:sender	The actual sender of the performative

:reply-with <expression>

:in-reply-to <expression>

ถ้า <expression> เป็น nil หรือ parameter นี้มันขาดหายไปจาก performative แล้วผู้ส่งจะไม่คาดหวังให้มีการตอบกลับ ถ้า <expression> เป็น t แล้วผู้ส่งจะคาดหวังการตอบกลับ มิฉะนั้น ผู้ส่งคาดหวังการตอบกลับที่อยู่ใน :in-reply-to parameter ด้วยค่าที่เหมือนกับ <expression>

:content <expression>

:language <word>

:ontology <word>

:content parameter เป็นการแสดงถึง “direct object” ของ performative ตัวอย่างเช่น ถ้า performative name เป็น tell แล้ว :content จะเป็น sentence ที่ได้ถูกบอกกล่าว <expression>

ใน :content parameter ต้องเป็น valid expression ใน representation language ที่ถูกระบุโดย :language parameter หรือ KQML ถ้า :language parameter ไม่ปรากฏ นอกจากนี้ ค่าคงที่ที่ถูกใช้ใน expression ต้องเป็น subset ของที่ซึ่งถูกกำหนดโดย ontology named ของ :ontology parameter หรือ standard ontology สำหรับ representation language ถ้า :ontology parameter ไม่ปรากฏ

:force <word>

ถ้าค่าของ parameter นี้เป็น permanent แล้วผู้ส่งจะการันตีว่ามันจะไม่ปฏิเสธความหมายของ performative ค่าอื่นๆจะบ่งชี้ว่า ผู้ส่งอาจปฏิเสธความหมายในอนาคต (parameter นี้มีอยู่เพื่อช่วยให้เอเจ้นท์ หลีกเลี่ยง overhead ที่ไม่จำเป็น)