

บทที่ 4

รายละเอียดเกี่ยวกับการสร้างระบบ

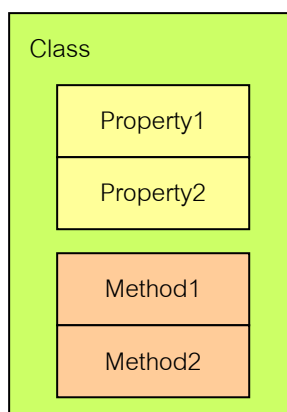
งานวิจัยนี้ได้แบ่งการทำงานออกเป็น 4 ส่วน ได้แก่ การสร้างระบบฐานความรู้ภายนอกบนระบบฐานข้อมูล กลไกการวินิจฉัยบนพื้นฐานความรู้ภายนอก การเชื่อมต่อระบบฐานความรู้ภายนอกกับระบบสารสนเทศอื่นๆ การติดต่อสื่อสารระหว่างไคลเอนต์กับเซิร์ฟเวอร์ด้วยภาษา KQML

4.1 เปรียบเทียบหลักการเชิงวัตถุสัมพันธ์กับเฟรม

เนื่องจากฐานข้อมูลเชิงวัตถุสัมพันธ์เป็นการขยายความสามารถในด้านวัตถุให้กับฐานข้อมูลเชิงสัมพันธ์ ประกอบกับรูปแบบการแทนความรู้แบบเฟรม กับแนวคิดเชิงวัตถุมีความใกล้เคียงกันดังนั้นในงานวิจัยนี้จึงได้ทำการศึกษาโครงสร้างข้อมูลเชิงวัตถุเปรียบเทียบกับโครงสร้างข้อมูลแบบเฟรมเสียก่อน แล้วเปรียบเทียบความเป็นไปได้ที่จะนำหลักการแนวคิดเชิงวัตถุมาใช้กับเฟรม โดยใช้ฐานข้อมูลเชิงวัตถุสัมพันธ์

4.1.1 โครงสร้างข้อมูลเชิงวัตถุ

โครงสร้างข้อมูลเชิงวัตถุโดยทั่วไป เป็นโครงสร้างข้อมูลที่ยึดหลักการออกแบบโดยอาศัยแนวคิดทางวัตถุสิ่งของทั้งที่จับต้องได้และไม่ได้ โดยจะสร้างโครงสร้างข้อมูลที่รวบรวมเอาตัวบ่งบอกคุณสมบัติของวัตถุนั้นๆ เข้าไว้ด้วยกัน ในลักษณะการเ็นแคปซูล (Encapsulate) โดยภายในตัววัตถุเองอาจประกอบด้วยเมธอด หรือโพรซีเจอร์ ที่เกี่ยวข้องกับวัตถุรวมกันไว้ ดังแสดงให้เห็นในรูปที่ 4.1

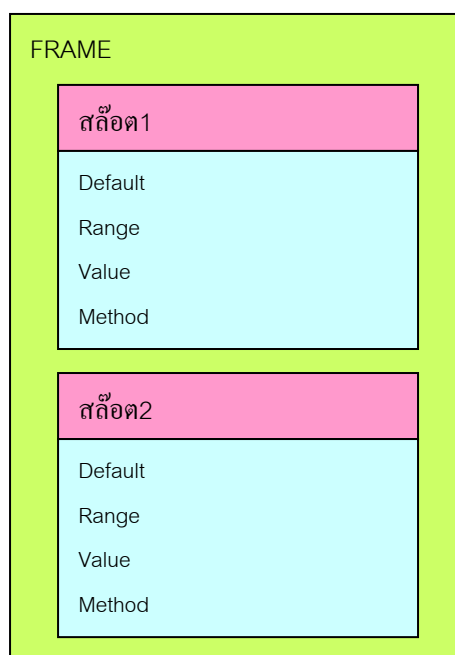


รูปที่ 4.1 แผนภาพโครงสร้างข้อมูลแบบวัตถุ

4.1.2 โครงสร้างข้อมูลแบบเฟรม

เป็นโครงสร้างที่เกิดมาจากแนวคิดในการออกแบบโดยยึดหลักของการออกแบบตามวัตถุ หรือคอนเซ็ปต์ เช่นเดียวกับแนวคิดเชิงวัตถุหากแต่เฟรมจะให้ความสำคัญกับองค์ประกอบในระดับตัวบ่งบอกคุณสมบัติของวัตถุมากกว่า กล่าวคือเฟรมจะมองสิ่งที่มีหน้าที่รวบรวมเอาตัวบ่งบอกคุณสมบัติของวัตถุเป็นเพียงเสมือนกรอบที่ถูกวาดไว้เพื่อรักษาการรวมกลุ่มกันของตัวบ่งบอกคุณสมบัติเท่านั้น นั่นจึงเป็นสาเหตุที่แนวคิดนี้ถูกเรียกว่าเฟรม

ซึ่งในขณะที่ตัวบ่งบอกคุณสมบัติของวัตถุทั้งหลายจะถูกให้ความสนใจเป็นเสมือนวัตถุหน่วยย่อยๆ ที่ภายในตัวเองจะมีคุณสมบัติสำหรับตัวเองรวมถึงเมธอด โพรซีเยอร์ และเดมอนที่เกี่ยวข้องกับตัวเองไว้ด้วยกัน หรือกล่าวอีกนัยหนึ่งคือโพรซีเยอร์หรือเดมอนต่างๆที่มีความเกี่ยวข้องกับตัวบ่งบอกคุณสมบัติใด จะถูกรวบรวมไว้กับตัวบ่งบอกคุณสมบัตินั้นเอง ทำให้หน่วยย่อยที่ทำหน้าที่เป็นตัวบ่งบอกคุณสมบัติของวัตถุนั้นถูกเรียกว่าสล็อต ดังแสดงให้เห็นในรูปที่ 4.2



รูปที่ 4.2 แผนภาพโครงสร้างข้อมูลแบบเฟรม

จากข้อมูลเบื้องต้นด้านโครงสร้างข้อมูลแบบวัตถุ และเฟรม ทำให้พบว่า มีความแตกต่างในด้านการให้ความสำคัญระดับกับข้อมูลดังที่ได้กล่าวมาแล้วข้างต้นทำให้โครงสร้างข้อมูลปกติของระบบฐานข้อมูลเชิงวัตถุไม่สามารถรองรับการจัดเก็บและการบริหารจัดการกับข้อมูลที่อยู่ในโครงสร้างแบบเฟรมได้โดยตรงเป็นสาเหตุให้งานวิจัยโดยส่วนใหญ่ที่ต้องการพัฒนาระบบ

ผู้เชี่ยวชาญต้องเลือกใช้วิธีการแมปหรือแปลงโครงสร้างข้อมูลแบบเฟรมให้มาอยู่ในรูปวัตถุก่อนจะจัดเก็บ

สำหรับงานวิจัยในวิทยานิพนธ์ฉบับนี้มีความต้องการที่จะประยุกต์ใช้โครงสร้างข้อมูลเชิงวัตถุสัมพันธ์ให้สามารถจัดเก็บข้อมูลที่ถูกสร้างด้วยการแทนความรู้แบบเฟรมได้โดยตรงเพราะโครงสร้างข้อมูลเชิงวัตถุนั้นจะมีการห่อหุ้มข้อมูล (encapsulation) แต่ในโครงสร้างข้อมูลแบบเฟรมนั้นไม่มีการห่อหุ้มข้อมูล และด้วยข้อจำกัดในความแตกต่างด้านการให้ความสำคัญระดับข้อมูลนี้ทำให้งานวิจัยนี้จึงเลือกใช้ระบบฐานข้อมูลเชิงวัตถุสัมพันธ์

4.2 การสร้างระบบฐานความรู้ในระบบฐานข้อมูลเชิงวัตถุสัมพันธ์

จากผลการศึกษาความแตกต่างระหว่างโครงสร้างข้อมูลเชิงวัตถุ และเฟรมตามที่ได้อธิบายไว้ข้างต้นแล้ว จะพบว่าข้อแตกต่างหลักอยู่ที่การให้ความสำคัญในระดับของข้อมูลที่ต่างกัน กล่าวคือ หลักการเชิงวัตถุจะให้ความสำคัญเน้นที่ระดับวัตถุหรือคอนเซ็ปต์ แต่เฟรมจะให้ความสำคัญในระดับที่ต่ำกว่า คือระดับตัวบ่งบอกคุณสมบัติของวัตถุ หรือคอนเซ็ปต์ ทำให้ความรู้ที่ใช้เทคนิคการแทนความรู้แบบเฟรมจะไม่สามารถจัดเก็บลงในโครงสร้างข้อมูลเชิงวัตถุได้โดยตรง ดังนั้นจึงมีความจำเป็นจะต้องดัดแปลง หรือสร้างโครงสร้างใหม่ภายในระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ ที่มีรูปแบบเหมือน หรือใกล้เคียงกับ เฟรมมากที่สุด

จากสาเหตุข้างต้น ทำให้งานวิจัยนี้เลือกใช้ฐานความรู้เชิงวัตถุสัมพันธ์ในการจัดเก็บโครงสร้างแบบเฟรม ซึ่งมีความสามารถเข้าในระดับตัวบ่งบอกคุณสมบัติของเฟรมได้ และในขณะเดียวกัน ก็สามารถจะใช้มุมมองในรูปแบบเชิงวัตถุ ได้ด้วย

ในสถาปัตยกรรมของระบบที่สร้างขึ้นนั้น เพื่อที่จะเก็บฐานความรู้ในระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ นั้นเราต้องกำหนดการแปลงจากโมเดลความรู้ (knowledge model) ของระบบฐานความรู้แบบเฟรมไปเป็นโมเดลข้อมูล (data model) ของระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ เสียก่อน

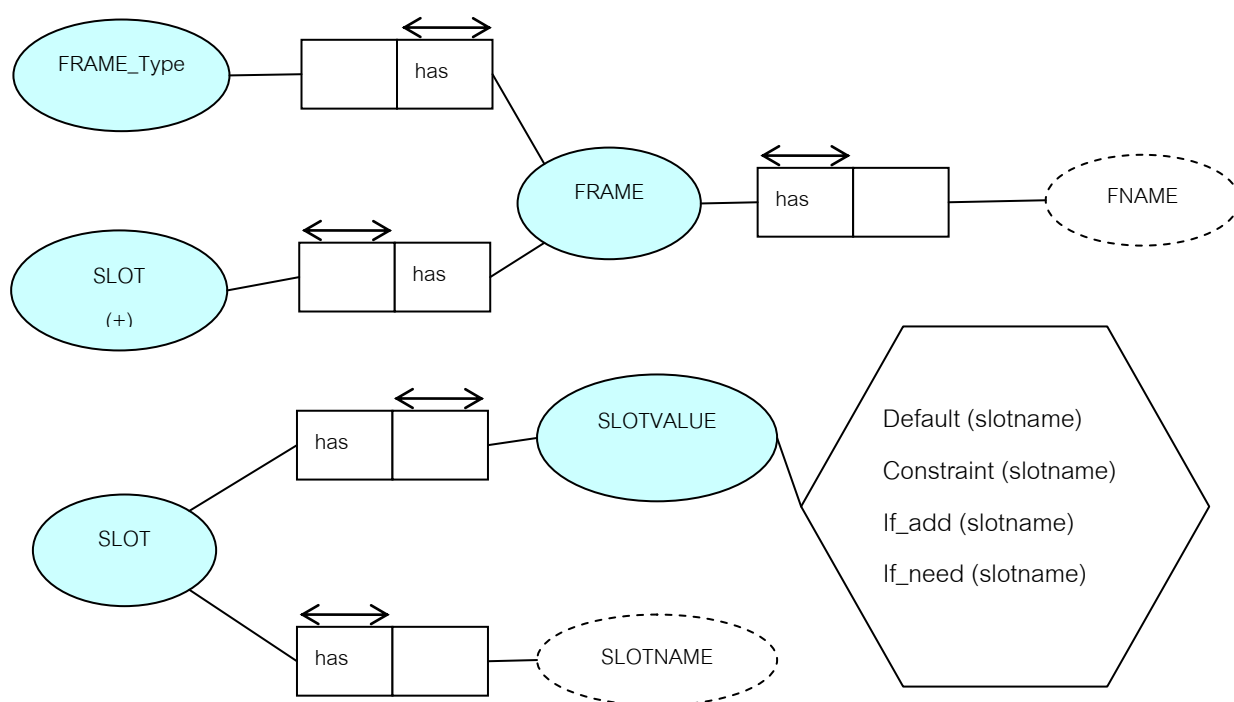
4.2.1 การจัดเก็บเฟรมในฐานข้อมูลเชิงวัตถุสัมพันธ์

เนื่องจากการที่จะจัดเก็บโครงสร้างแบบเฟรม ในฐานข้อมูลเชิงวัตถุสัมพันธ์นั้น เราจะต้องทำการแปลงโครงสร้างแบบเฟรมให้สามารถจัดเก็บอยู่ในฐานข้อมูลเชิงวัตถุสัมพันธ์ให้ได้เสียก่อน ซึ่งจากการศึกษาพบว่า ในการจัดเก็บฐานความรู้ในระบบฐานข้อมูลนั้น ไม่จำเป็นต้องเป็นฐานความรู้ชนิดเดียวกันนั้น ที่สามารถถูกจัดเก็บได้ เพื่อที่ว่าเวลาเรียกข้อมูลจากฐานความรู้ขึ้นมา หรืออ้างอิงไปยังอีกฐานความรู้หนึ่ง จะทำให้เกิดความซับซ้อนในขั้นตอนการดำเนินงาน

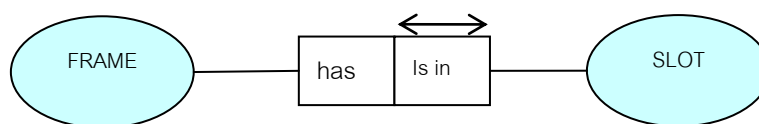
งานวิจัยนี้ยังได้ทำเสนอการออกแบบตารางที่ใช้เก็บฐานความรู้แบบเฟรม โดยใช้ NIAM [25] สำหรับออกแบบฐานข้อมูลในการจัดเก็บเฟรมต่างๆ ซึ่ง NIAM เป็น fact-based conceptual

schema model ซึ่งประกอบไปด้วย ชนิดของเอนทิตี (entity type), ชนิดของเลเบล (label type), ชนิดของข้อเท็จจริง (fact type) และ ชนิดของการอ้างอิง (reference type) นอกจากนี้ก็ควบคุมความถูกต้องของข้อมูล (static integrity constraints) ก็สามารถถูกแสดงบน conceptual schema diagram ได้ด้วย และ มีส่วนขยายเพื่อที่จะรองรับรูปแบบของฐานข้อมูลเชิงวัตถุได้ด้วย ดังนี้ [26] [27]

1. ชนิดของเอนทิตีที่สามารถเกิดขึ้นได้โดยไม่จำเป็นต้องมี ตัวบ่งบอกค่าเฉพาะ (unique identifier) ซึ่งเรียกว่า เอนทิตีชนิดซับซ้อน (complex entity type) ซึ่ง เอนทิตีชนิดซับซ้อนนี้จะแสดงถึง คลาสของวัตถุ ซึ่งตัวบ่งบอกอินสแตนท์ของวัตถุ (object instance identifier) จะถูกสร้างขึ้นจากระบบสร้าง ids ของวัตถุขึ้นมาเอง
2. เพิ่ม สัญลักษณ์ สำหรับกระบวนการ
3. เพื่อที่จะสนับสนุน วิธีการแบบบนลงล่าง (top-down) แนวคิดของ main schema และ sub schema ได้ถูกนำเสนอขึ้นด้วย โดยที่ sub schema จะอธิบายถึง ชนิดของข้อเท็จจริง และ ชนิดของการอ้างอิง ซึ่งเป็นไฟเวท และถูกห่อหุ้ม ภายใน เอนทิตีชนิดซับซ้อนส่วน main schema จะอธิบายถึง ชนิดของข้อเท็จจริง ของ เอนทิตีชนิดซับซ้อน



รูปที่ 4.3 A Frame sub schema Using NIAM



รูปที่ 4.4 A Frame main schema Using NIAM

จากรูปที่ 4.3 และรูปที่ 4.4 จะได้ว่า ระบบนี้ในฐานข้อมูลหนึ่งจะมีเฟรมได้หลายเฟรมซึ่งในแต่ละเฟรมจะมีชื่อของเฟรม และ ชนิดของเฟรม รวมทั้งมีเซต ของ สล็อตหรือ แอตทริบิวต์ บรรจุอยู่ภายใน โดยที่ในแต่ละ สล็อตจะประกอบไปด้วย ชื่อของสล็อต และค่าของสล็อตซึ่งค่าของสล็อต จะถูกกำหนดโดย ฟาเซท ซึ่งเป็น ข้อกำหนด หรือ กระบวนการ สำหรับควบคุมค่าใน สล็อต ดังรูป

จากรูปที่ 4.3 และรูปที่ 4.4 จะได้ schema ออกมา ดังนี้

FRAME	SLOT
FNAME: string FRAME_TYPE: string Has : set (SLOT)	SLOTNAME: string Has: set(SLOTVALUE)
	Default (slotname) Constraint (slotname) If_add (slotname) If_need (slotname)

รูปที่ 4.5 Object Schema ของเฟรม

จากรูปที่ 4.5 จะได้ว่าระบบนี้มีสองวัตถุ คือ เฟรม และ สล็อตโดยใน วัตถุ เฟรม จะเก็บชื่อของเฟรม และชนิดของเฟรม พร้อมทั้ง collection ของ สล็อตส่วน สล็อตวัตถุ ก็จะมีชื่อ และ collection ของ ค่าสล็อต โดยมี กระบวนการไว้ควบคุม ค่าของสล็อต ซึ่งก็คือฟาเซท นั่นเอง

และจากตัวอย่างข้างต้น เราสามารถทำการแปลงให้อยู่ในรูปแบบ nested table ได้ในรูปแบบที่ 4.6 ดังนี้

FRAME			
FRAMENAME	FRAMETYPE	SLOTNAME	SLOTVALUE
			SLOTVALUE
		SLOTNAME	SLOTVALUE
			SLOTVALUE

รูปที่ 4.6 Object-Relational Schema ของเฟรม

ระบบฐานข้อมูลออรากเคิล 10g นั้นเป็นระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ โดยระบบจะทำการเพิ่มความสามารถของระบบฐานข้อมูลเชิงสัมพันธ์ ด้วยการเพิ่มความสามารถเชิงวัตถุ บนระบบฐานข้อมูลเชิงสัมพันธ์อีกที ซึ่งเมื่อมีการติดต่อกับฐานข้อมูลมันจะทำการติดต่อกับข้อมูลเสมือนว่าข้อมูลนั้นๆถูกจัดเก็บแบบวัตถุและด้วยความสามารถแบบวัตถุที่เพิ่มเข้ามาทำให้สามารถจัดเก็บข้อมูลวัตถุในฐานข้อมูลได้ และกระบวนการสามารถถูกเรียกโดยตรงจากภาษา SQL ได้ ซึ่งอย่างไรก็ตามระบบก็ยังต้องแปลงข้อมูลเกี่ยวกับวัตถุไปเป็นข้อมูลในตารางอยู่ดีและจัดการกับข้อมูลเหมือนกับฐานข้อมูลเชิงสัมพันธ์ ในขณะที่เดียวกันเมื่อมีการค้นคืนข้อมูลออกมามันก็จะแปลงจากข้อมูลตารางมาเป็นวัตถุที่ซับซ้อน (complex object) ดังนั้นระบบฐานข้อมูลเชิงวัตถุสัมพันธ์จึงสามารถลดการเกิดความซ้ำซ้อนได้จากการสร้างตารางข้อมูลเชิงสัมพันธ์ที่ไม่มีความซ้ำซ้อน แล้วใช้มุมมองเชิงวัตถุในการเรียกดูเฟรมขึ้นมาแบบเชิงวัตถุซึ่งวิธีนี้ระบบสามารถเข้าถึงข้อมูลทั้งในแบบเชิงวัตถุและเชิงสัมพันธ์ได้ทั้งสองแบบตามการใช้งาน

ซึ่งในฐานข้อมูลออรากเคิล 10g เราสามารถจัดเก็บเฟรมได้ดังนี้คือ

1. สร้างตารางที่จัดเก็บข้อมูลของฟาเซ็ทขึ้นมาโดยสร้างเป็น collection type ของวัตถุ
2. สร้างตารางที่จัดเก็บสต็อคโดยที่ภายในแต่ละแถวของสต็อคจะมีตารางของฟาเซ็ทอยู่
3. สร้างชนิดเชิงวัตถุที่เป็นเฟรมซึ่งประกอบไปด้วยชื่อของเฟรมซึ่งแต่ละเฟรมจะมีตารางของสต็อคซึ่งในแต่ละสต็อคจะมีตารางของฟาเซ็ท

```
CREATE OR REPLACE TYPE t_facets_row AS OBJECT
```

```
( id NUMBER(10),
```

```
slot_id NUMBER(10),
```

```
name VARCHAR2(50),
```

```
facet_value VARCHAR2(50)
```

```

)

CREATE OR REPLACE TYPE t_facets_tab AS TABLE OF t_facets_row

CREATE OR REPLACE TYPE t_slots_row AS OBJECT (

    id      NUMBER(10),

    frame_id NUMBER(10),

    name     VARCHAR2(50),

    facets   t_facets_tab

)

CREATE OR REPLACE TYPE t_slots_tab AS TABLE OF t_slots_row

CREATE OR REPLACE TYPE t_frame_row AS OBJECT

( id      NUMBER,

  name    VARCHAR2 (50),

  frametype VARCHAR2(50),

  slots   t_slots_tab

)

```

4.3 กลไกการวินิจฉัยของระบบ

ในระบบผู้เชี่ยวชาญแบบเฟรมนั้นมีการวินิจฉัยได้หลายรูปแบบ เช่น การวินิจฉัยโดยการสืบทอด (inference by inheritance) เป็นการตอบคำถามด้วยการหาเฟรม ไม่ว่าจะเป็นคลาสเฟรมหรืออินสแตนซ์เฟรม ที่มีสล็อตและค่าของสล็อตที่ต้องการ ซึ่งบางครั้งการสืบทอดสามารถถูกนิยามเป็นกระบวนการ ได้ [4] ตัวอย่างการวินิจฉัยโดยการสืบทอด เช่น Does Shania Twain have a spleen? จากคำถามนี้ถ้าเราไล่ไปยัง คลาสเฟรม ของ อินสแตนซ์เฟรม Shania Twain จะพบว่าป็นสมาชิกของคลาสเฟรม woman จะสรุปได้ว่า Shania Twain is woman และเมื่อไล่ขึ้นไปอีกจะพบว่า a woman is a kind of person ในคลาสเฟรม person ก็สรุปได้ว่า persons have spleens ดังนั้นเราสามารถสรุปได้ว่าโดยการวินิจฉัยโดยการสืบทอด Shania Twain has a spleen เป็นต้น

การจำแนกโดยการจับคู่ (Classification by matching) เป็นการจำแนก อินสแตนซ์เฟรม โดยการจับคู่ค่าใน อินสแตนซ์เฟรม กับ ค่าใน คลาสเฟรม ที่เกี่ยวเนื่องกัน ว่าอันไหนตรงที่สุด ตัวอย่างเช่น What kind of singer is Shania Twain? คำถามนี้ระบบจะทำการจับคู่ อินสแตนซ์เฟรม ของ Shania Twain เข้ากับ คลาสเฟรมต่างๆเหล่านี้ คือ opera-singer, country-singer, rock-singer, และ pop-singer ซึ่งถ้าคุณสมบัติของอินสแตนซ์เฟรม Shania Twain มันตรงกันกับ คลาสเฟรม pop-singer มากที่สุด ก็จะสรุปได้ว่า Shania Twain ถูกจำแนกให้เป็น pop singer

ระบบผู้เชี่ยวชาญที่ใช้วิธีการแทนความรู้แบบเฟรมส่วนใหญ่ นั้น มักจะใช้เทคนิคการจำแนกประเภท (Classification) และ inheritance ในการวินิจฉัย และแก้ปัญหาบนพื้นฐานของความรู้ที่จัดเก็บไว้ในฐานความรู้ ดังนั้นในงานวิจัยนี้จึงใช้กลไกการวินิจฉัยสองแบบคือ pattern matching และการสืบทอด

4.3.1 ขั้นตอนการวินิจฉัยของระบบ

กระบวนการวินิจฉัยนี้จะมีอยู่ด้วยกันสองแบบ แบบแรกคือให้ผู้ใช้กรอกข้อมูลเข้ามาแล้วทำการจับคู่ข้อมูลที่ได้รับกับเฟรม เพื่อระบุว่าเป็นอินสแตนท์ของคลาสเฟรมใด ส่วนแบบที่สองจะเป็นการถามว่าเฟรมที่ป้อนเข้ามาเป็นอินสแตนท์ของเฟรมที่ระบุหรือไม่ ซึ่งกระบวนการจะมีขั้นตอนดังนี้

ฝั่งไคลเอนต์

1. ระบบจะทำการเปรียบเทียบค่าที่ได้ป้อนเข้ามา กับเฟรมอาการแรกเริ่มที่มีอยู่
2. จากนั้นจะเลือกเฟรม ขึ้นมาก่อนและตั้งเฟรมเป็นเป้าหมาย
3. เมื่อได้เป้าหมายแล้วก็จะใส่ค่าในสล็อตของเฟรม
4. และในเฟรมเป้าหมายนี้ ถ้าไม่มีข้อเท็จจริงในเฟรมก็จะให้ผู้ใช้ตอบคำถาม
5. ทริกเกอร์ในสล็อตนี้ไปเรียกเฟรมถัดมาเมื่อมีการใส่ค่าเข้าไป
6. จากนั้นเมื่อได้รับข้อมูลจากผู้ใช้ครบถ้วนแล้วระบบจะส่งคำถามไปยังฝั่งเซิร์ฟเวอร์

ฝั่งเซิร์ฟเวอร์

1. ฝั่งเซิร์ฟเวอร์เมื่อรับเฟรมเข้ามาก็จะทำการเลือกเฟรมเป้าหมาย
2. เมื่อได้ลิสต์ของเป้าหมายแล้วก็เก็บไว้ในลิสต์ goal_list
3. เลือกเฟรมใน goal_list มาหนึ่งเฟรม โดยลบเฟรมออกจากลิสต์ goal_list ด้วย
4. จับคู่อาการในเฟรมนั้นกับข้อมูลที่ได้มา
5. ถ้ามีการร้องขอข้อมูลเพิ่มเติมจากในฐานข้อมูลภายนอก ให้ไปดึงข้อมูลจากฐานข้อมูลภายนอกเข้ามา
6. ถ้ามีการร้องขอให้ผู้ใช้ป้อนข้อมูลเพิ่มเติม ส่งคำถามไปยังฝั่งไคลเอนต์เพื่อให้ผู้ใช้ตอบคำถาม แล้วส่งคำตอบมายังฝั่งเซิร์ฟเวอร์
7. ถ้าเฟรมเป้าหมายสอดคล้องกับข้อมูลทั้งหมด ก็ส่งเฟรมนั้นเป็นคำตอบกลับไปยังผู้ใช้
8. ถ้าข้อมูลไม่สอดคล้องกับเฟรมเป้าหมาย ให้เลือกเฟรมเป้าหมายใหม่จากลิสต์ goal_list แล้วทำการจับคู่เฟรมอีกครั้งจนกว่าจะได้คำตอบ หรือจนกว่าไม่มีเฟรมเหลืออยู่ใน goal_list อีกแล้ว

จากขั้นตอนการวินิจฉัยของระบบข้างต้นเราสามารถเขียนอัลกอริทึมเพื่อทำการวินิจฉัยโดยรวมของระบบได้ดังนี้

Function:	find frame
Input:	facts
Output:	answer
1. find frame that match facts	
2. loop	
3. find children of matched frame	
4. push into goal stack list	
5. until no children of frame found	
6. pop frame from goal stack	
7. bind slot variable with fact	
8. match frame with current goal frame	
9. return matched frame	

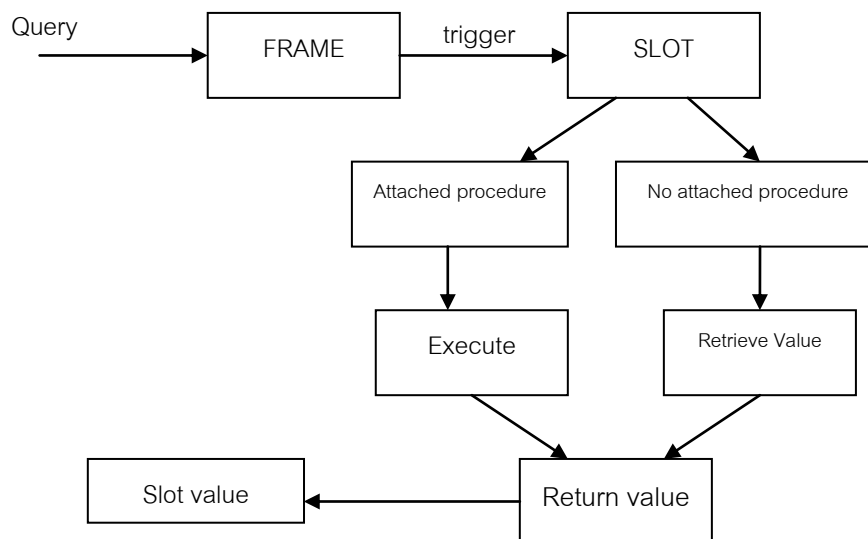
รูปที่ 4.7 อัลกอริทึมในการวินิจฉัยของระบบ

จากขั้นตอนการวินิจฉัยข้างต้นกลไกการวินิจฉัยนี้จะกระทำต่อเมื่อได้มีการส่งคำถามมาจากฝั่งไคลเอนต์เพื่อให้เซิร์ฟเวอร์ทำการวินิจฉัยต่อโดยที่ข้อมูลทางฝั่งไคลเอนต์จะทำการวินิจฉัยไประยะหนึ่งก่อนเพื่อให้ผู้ใช้ได้ตอบคำถาม (forward reasoning) ซึ่งเมื่อได้ข้อมูลมาพอสมควร ระบบจะทำการตั้งเป้าหมายแล้วค้นหาว่าข้อมูลที่ได้มาตรงกับเป้าหมายใดมากที่สุด ซึ่งจะเข้าสู่กระบวนการวินิจฉัยเพื่อจับคู่เฟรมที่สอดคล้องที่สุด (backward reasoning) ดังนั้นในงานวิจัยนี้การวินิจฉัยจะเป็นแบบผสม คือมีการ forward reasoning ไประยะหนึ่งก่อนจากนั้นถึงทำการวินิจฉัยแบบ backward reasoning ต่อไปจนพบเป้าหมายแล้วส่งผลลัพธ์หรือคำตอบกลับไปยังผู้ใช้ได้

4.3.2 กลไกการวินิจฉัยที่ฝั่งฐานความรู้ภายนอก

งานวิจัยนี้จะใช้โปรแกรมที่เขียนด้วยภาษาจาวา เป็นตัวติดต่อกับฐานข้อมูล ผ่านทาง JDBC driver ของระบบการจัดการฐานข้อมูล ออราเคิล 10g โดยเทคนิคในการวินิจฉัยนั้นจะมีอยู่ด้วยกันสองวิธี หนึ่งคือ รับเฟรมจากฝั่งไคลเอนต์เข้ามาเก็บไว้ในเฟรมบนฝั่งเซิร์ฟเวอร์ซึ่งกระบวนการจะถูกเรียกขึ้นมาจากการเปลี่ยนแปลงค่าในสล็อต และสองคือ เก็บข้อมูลที่ได้จากเฟรมบนฝั่งไคลเอนต์มาเป็น พารามิเตอร์ของ store procedure ซึ่งจะทำการวินิจฉัยโดยใช้ store procedure ต่อไป ซึ่งวิธีที่สองนี้จะเหมาะสำหรับข้อมูลที่มาจากฝั่งไคลเอนต์มีจำนวนไม่มาก โดยที่ในส่วนฐานข้อมูลฝั่งเซิร์ฟเวอร์ซึ่งเราเก็บฐานความรู้ไว้ จะมีการเก็บ store procedure ไว้ในระบบการจัดการฐานข้อมูล เพื่อกระทำการตามสถานการณ์ โดยตัวที่จะใช้เรียก ฟังก์ชัน จะอยู่ในส่วนของโปรแกรมซึ่งค่าที่ได้จากการประมวลผลจะถูก ส่งกลับไปยังไคลเอนต์ต่อไป

ซึ่งค่าของ สล็อต จะถูกเรียกใช้เมื่อต้องการ ซึ่งในส่วนของ attached procedure นี้จะแตกต่างกันไปในแต่ละทิวเฟิลซึ่งบางที อาจต้องเรียกต่อเนื่องไปเรื่อยๆ จนกว่าจะได้คำตอบ แล้วจึงประมวลค่าอีกครั้งเพื่อส่งค่าที่ได้กลับไปยังฟังก์ชันโคลเอนท์อีกที



รูปที่ 4.8 แสดงการทำงานของกระบวนการภายในเฟรม

ในส่วนของการบวนการจะมีความทำงานดังนี้คือเมื่อรับคำสั่งจากผู้ใช้หรือระบบผู้เชี่ยวชาญฟังก์ชันโคลเอนท์นั้นจะทำการค้นหาค่าในเฟรมและเมื่อได้เฟรมที่ต้องการก็จะไปตรวจสอบที่สล็อตว่าได้มี stored procedure สำหรับเหตุการณ์นั้นๆหรือไม่ จากนั้นจึงรับค่าที่ได้ทำการกระทำในส่วนนี้ ไปยังใส่ไว้ในฟิลเลอร์ซึ่งเป็นค่าของสล็อตในเฟรมอินสแตนซ์อีกที โดยที่การใส่ค่าในสล็อตก็อาจจะมีกระบวนการที่กระทำหลังจากที่ใส่ค่าแล้วต่อไปก็ได้ ในรูปที่ 4.7 เป็นการแสดงการทำงานของกระบวนการภายในเฟรมโดยอันดับแรกจะมีการป้อนค่าหรือเรียกดูค่าในส่วนของคุณค่าของสล็อตของเฟรมอินสแตนซ์โปรแกรมจะดูว่าได้มีกระบวนการที่ต้องกระทำหลังป้อนค่าหรือก่อนเรียกดูค่าหรือไม่ ถ้ามีก็จะไปเรียก store procedure โดยอ้างอิงจาก attached procedure ขึ้นมา ซึ่งกระบวนการนี้อาจมีการร้องขอให้มีการป้อนข้อมูลที่สล็อตอื่นได้ ซึ่งในรูปที่ 4.8 จะมีอัลกอริทึมดังนี้

```

Function:    get_all_slot_value
Input:      frame_id, slot_name
Output:     answer
1.  answer:=null;
2.  parent:=null;
3.  answer = call get_slot_value;
4.  if answer = null
5.      parent = call find_parent;
6.      while parent is not null AND answer = null
7.          parent = call find_parent;
8.      end loop
9.  return answer;

```

รูปที่ 4.9 อัลกอริทึมในการหาค่าของสล็อต

กระบวนการในรูปที่ 4.9 จะเป็นการหาค่าสล็อตจากเฟรมปัจจุบันไล่ขึ้นไปยังเฟรมพ่อแม่ตามลำดับชั้นและจะมีการเรียกฟังก์ชันในรูปที่ 4.10 และ 4.11 ตามลำดับดังนี้

```

Function :  get_slot_value
Input:    frame, slot_name
Output:   answer
1.  answer:= null;
2.  begin
3.      SELECT id INTO v_slot_id FROM slots_tab
4.      if v_slot_id is not null then
5.          begin
6.              SELECT 'value' INTO ret
7.              when no_data_found
8.              SELECT 'default' INTO ret
9.              when no_data_found
10.             SELECT if_needed INTO ret
11.             when no_data_found
12.                 answer = NULL;
13.          end if;
14.  answer ret;
15.end;

```

รูปที่ 4.10 อัลกอริทึมสำหรับการดึงค่าจากสล็อต

```

Function:    find_parent
Input:      frame, slot_name
Output:     parent
1.  begin
2.      parent := null;
3.      SELECT parent INTO parent FROM parent_tab
4.      if no_data_found then
5.          parent := null;
6.      return parent
7.  end;

```

รูปที่ 4.11 อัลกอริทึมสำหรับการหาเฟรมแม่

4.4 การดึงข้อมูลสารสนเทศจากฐานข้อมูลอื่นเข้ามายังระบบเฟรม

ในงานวิจัยนี้ เราจะทำการอ้างถึงข้อมูลจากฐานข้อมูลที่มีอยู่เข้ามายังฝั่งฐานข้อมูล ที่เป็นระบบผู้เชี่ยวชาญโดยการเข้าถึงเฟรมที่มีอยู่ แล้วเรียกต่อๆ กัน ไปจนไปถึงข้อมูลจริงๆ ซึ่งอยู่บนฐานข้อมูลอื่น โดยมีกระบวนการที่อ้างไปถึงข้อเท็จจริง ที่อยู่ในฐานข้อมูล อื่น ในเฟรม นั้นๆ ซึ่งไม่จำเป็นต้องทำการ โหลดข้อมูลจากฐานข้อมูลที่มีอยู่แล้วเข้ามายังเฟรม ก่อน แต่จะเป็นการอ้างถึงข้อมูลที่เกี่ยวข้อง ซึ่งก็คือข้อเท็จจริง จากฐานข้อมูลอื่นๆ มายังระบบเฟรม ที่อยู่ใน ฐานข้อมูลของออราเคิล อีกทั้ง โดยที่ข้อมูลจากฐานข้อมูลตัวอื่น ที่เราเชื่อมต่อกับระบบผู้เชี่ยวชาญแบบเฟรม นี้ ไม่จำเป็นต้องมีการเปลี่ยนแปลงรูปแบบใดๆ และผู้ใช้ ของระบบนั้นสามารถเข้าถึงข้อมูลได้ตามปกติ ซึ่งระบบที่เราวิจัยนี้ระบบฐานข้อมูลที่เก็บฐานความรู้บนฝั่งเซิร์ฟเวอร์จะทำหน้าที่เสมือนกับเป็นตัวระบบผู้เชี่ยวชาญ ที่เชื่อมต่อกับฐานข้อมูลอีกที โดยระบบของเราจะเป็นระบบผู้เชี่ยวชาญ ที่เป็นระบบจัดการฐานข้อมูลด้วย ทำให้การเชื่อมต่อกับระบบฐานข้อมูลอื่นถูกกระทำโดยง่ายและเป็นการใช้ฐานข้อมูลที่มีอยู่แล้วให้มีประสิทธิภาพด้วย

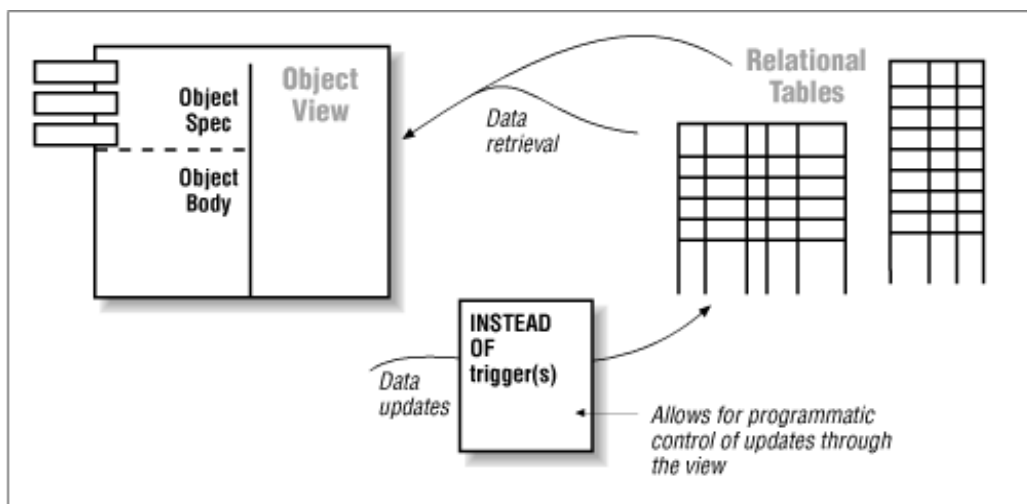
และด้วยเหตุที่เราต้องเชื่อมต่อกับระบบผู้เชี่ยวชาญนี้ กับระบบฐานข้อมูลอื่นๆ การเลือกใช้ฐานข้อมูลเชิงวัตถุสัมพันธ์ ก็เป็นประโยชน์ซึ่งทำให้เราสามารถเชื่อมโยงกับระบบฐานข้อมูลอื่นที่มีอยู่แล้วที่ส่วนใหญ่จะเป็นระบบฐานข้อมูลเชิงสัมพันธ์ หรือบางทีอาจจะเป็นฐานข้อมูลเชิงวัตถุสัมพันธ์ก็ได้ เพราะระบบฐานข้อมูลเชิงวัตถุสัมพันธ์ นี้สามารถรองรับการเชื่อมต่อกับระบบฐานข้อมูลทั้งสองแบบ และกระทำได้โดยง่าย

ซึ่งในงานวิจัยนี้เราได้ใช้คุณสมบัติที่มีอยู่แล้วของออราเคิล 10g นั่นคือ dblink ทำให้สามารถเชื่อมต่อกับฐานข้อมูลอื่นที่เป็นฐานข้อมูลของออราเคิลเหมือนกันได้และสามารถทำการจัดการ (DML) กับฐานข้อมูลนั้นได้เลย แต่ในกรณีที่เป็นการเชื่อมต่อกับฐานข้อมูลอื่นที่ไม่ใช่ออราเคิลเหมือนกันอย่างเช่น MySQL เป็นต้นในระบบของเราจะเป็นเพียงการดึงข้อมูลขึ้นมาเพื่อให้ได้ข้อเท็จจริงที่ต้องการเท่านั้นโดยอาจจะใช้ OCA (Oracle Open Client Adapter) สำหรับเชื่อมต่อกับฐานข้อมูลอื่นและในขั้นตอนนี้ส่วนใหญ่เราต้องการเพียงข้อเท็จจริงเพิ่มเติมเท่านั้นจึงไม่มีความจำเป็นที่จะต้องเปลี่ยนแปลงข้อมูลของฐานข้อมูลที่เชื่อมต่ออยู่ด้วยเพราะฉะนั้นจึงเป็นเพียงการสร้างฟังก์ชัน SQL เพื่อทำการคิวรีข้อมูลจากฐานข้อมูลอื่นขึ้นมาเท่านั้น

ในงานวิจัยนี้การติดต่อกับฐานข้อมูลภายนอกอื่นๆจะจำลองรูปแบบการเรียกใช้งานของระบบผู้เชี่ยวชาญที่จัดเก็บเฟรมอยู่ในฐานข้อมูลแต่ว่าระบบที่สร้างขึ้นนี้จะต้องไม่ไปยุ่งเกี่ยวกับโครงสร้างที่มีอยู่แล้วในฐานข้อมูลภายนอกอื่นๆดังนั้นในงานวิจัยนี้จึงได้มีการทำตารางเสมือนขึ้นมาเพื่อสร้างมุมมองเชิงวัตถุ (object view) เพื่อมองข้อมูลภายนอกเป็นลักษณะเฟรม และทำให้ฐานความรู้ที่จัดเก็บอยู่ในระบบฐานข้อมูลสามารถเรียกข้อมูลขึ้นมาใช้งานได้

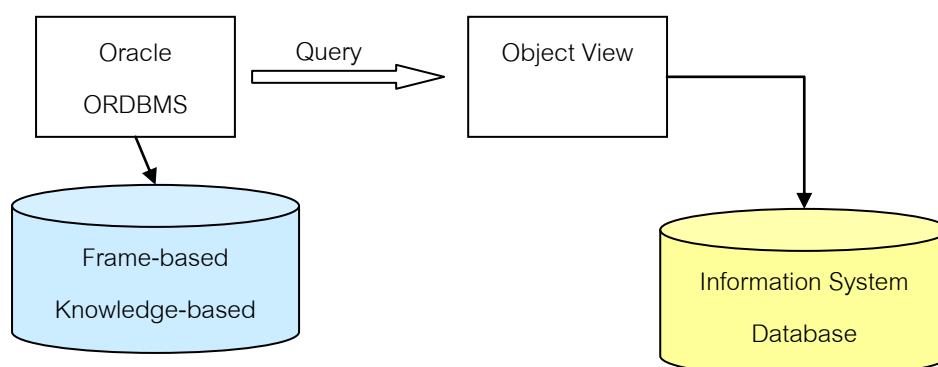
ในรูปที่ 4.12 [7] จะแสดงการเรียกดูข้อมูลจากฐานข้อมูลโดยผ่านมุมมองเชิงวัตถุในฐานข้อมูลออราเคิลตัวอื่น โดยที่ ข้อมูลที่ต้องถามไปที่ฐานข้อมูลอื่นจะมีการไปคิวรี จากฐานข้อมูล

ก่อนซึ่งถ้ายังไม่เคยมีค่าในระบบ ก็จะทำการสอบถาม ผู้ใช้ต่อไป เช่นข้อมูลทางการแพทย์ ที่จะถามถึงประวัติคนไข้ในฐานะข้อมูลที่เป็นเวชระเบียน ซึ่งถ้าคนไข้ไม่เคยมีประวัติมาก่อน เมื่อทำการ คิวรี แล้วไม่ได้คำตอบ ก็จะสอบถามผู้ใช้ถึงประวัติคนไข้ เพื่อเป็นข้อมูลไว้อ้างถึงข้อเท็จจริงในระบบผู้เชี่ยวชาญในภายหลังได้ เพราะฉะนั้นตัวฐานข้อมูลที่เป็นระบบผู้เชี่ยวชาญ ของงานวิจัยนี้จะเก็บ SQL คิวรี หรือกระบวนการที่เกี่ยวข้องผูกไว้กับเฟรมเพื่ออ้างถึงข้อเท็จจริงนั้นเพื่อเชื่อมไปยังข้อเท็จจริงที่อยู่ในฐานข้อมูลอื่นต่อไป



รูปที่ 4.12 แสดงการใช้มุมมองเชิงวัตถุกับฐานข้อมูลภายนอก

เฟรมที่จะถามจากฐานข้อมูลนั้นจะเป็นอินสแตนซ์เฟรม ส่วนคลาสเฟรมจะมี attached procedure ซึ่งจะสืบทอด attached procedure นี้มายังอินสแตนซ์เฟรม ซึ่งเมื่อต้องการจะค้นคืน ค่าขึ้นมา ก็จะไปค้นหาข้อเท็จจริงจากฐานข้อมูลก่อน



รูปที่ 4.13 แสดงการเข้าถึงข้อมูลของฐานข้อมูลอื่นจากเฟรม

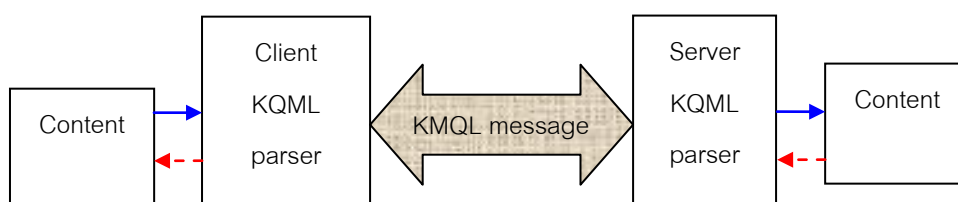
จากรูปที่ 4.13 เราอธิบายได้ว่า ผู้ใช้ได้กรอกข้อมูลลงไป สมมติว่ามีข้อมูลอยู่แล้วเป็นคนไข้เก่า ก็จะกรอกเพียงแค่เลขที่บัตรประชาชนเท่านั้น เมื่อส่งข้อมูลไปทำการวินิจฉัยบนฝั่งเซิร์ฟเวอร์จะส่งค่าเลขที่บัตรประชาชนไปด้วย เพื่อว่าถ้ามีการร้องขอข้อมูลส่วนตัวจากฐานข้อมูลอื่นในภายหลัง ก็จะไปดึงข้อมูลขึ้นมาได้เลยโดยไม่ต้องถามผู้ใช้อีกครั้ง แต่ในขณะเดียวกันถ้าข้อมูลเป็นประวัติคนไข้ใหม่ผู้ใช้ก็ต้องขึ้นประวัติคนไข้ใหม่ โดยอาจให้เจ้าหน้าที่ทางส่วนเวชระเบียนป้อนข้อมูลต่างๆ ลงบนฐานข้อมูลให้เรียบร้อยก่อน แล้วจึงมาทำการวินิจฉัยเหมือนกับเป็นคนไข้เก่าปกติ

4.5 โพรโทคอลที่ใช้ในการติดต่อสื่อสารระหว่างไคลเอนท์กับเซิร์ฟเวอร์

ในการติดต่อสื่อสารระหว่างผู้เชี่ยวชาญอย่างมีประสิทธิภาพ จะต้องประกอบไปด้วย

- ภาษากลางในการติดต่อสื่อสาร (common language)
- ความเข้าใจเกี่ยวกับความรู้ที่จะแลกเปลี่ยน (exchange) แบบเดียวกัน
- ความสามารถในการแลกเปลี่ยนอะไรก็ตามที่อยู่ใน (a) กับ (b)

ในงานวิจัยนี้ เราเลือกใช้ KQML เป็นโพรโทคอลที่ใช้ในการติดต่อสื่อสารระหว่างไคลเอนท์กับเซิร์ฟเวอร์ เนื่องจากเป็นที่นิยมใช้อย่างแพร่หลาย ในการติดต่อสื่อสารระหว่างเอเจนต์ ซึ่งในการติดต่อสื่อสาร ระหว่าง เอเจนต์นั้น ข้อมูลที่จะถูกส่งไป จะต้องถูกแปลงให้อยู่ในรูปแบบภาษา KQML



รูปที่ 4.14 โพรโทคอลที่ใช้ในการติดต่อสื่อสารระหว่างไคลเอนท์กับ เซิร์ฟเวอร์

จากรูปที่ 4.14 จะแสดงถึงการแปลงข้อมูลที่ต้องการจะส่งให้อยู่ในรูปแบบ KQML เสียก่อนด้วย KQML parser ให้อยู่ในรูปแบบข้อความ KQML และส่งข้อความนั้น ไปยังผู้รับ ซึ่งก็คือเซิร์ฟเวอร์ โดยเมื่อได้รับข้อความแล้วจะต้องทำการแปลงข้อความ KQML ให้ออกมาเป็นข้อมูลเสียก่อน แล้วนำข้อมูลนั้นไปทำการวินิจฉัยต่อไป เมื่อทำการวินิจฉัยเสร็จสิ้นแล้ว ก็ส่งคำตอบที่ได้กลับไปยังฝั่งไคลเอนท์ด้วยวิธีการเดิม

เฟรมที่ได้จากฝั่งไคลเอนท์นี้จะอยู่ในรูปแบบของ script ในภาษาจาวา และเมื่อเราจะทำการส่งไปยังเซิร์ฟเวอร์ที่เกี่ยวข้อง เราต้องทำการเข้ารหัสเฟรมนี้ให้อยู่ในรูปแบบ ข้อความ KQML ดังนี้

(ask-one

```
:sender client1
:content "get-value(dog,blood-type)"
:receiver server1
:reply-with knowledge-base1
:language LISP
:ontology animal-kingdom)
```

ในข้อความนี้เนืองาน ก็คือส่วนของเฟรมทางฝั่งไคลเอนท์ที่เราทำการส่งไป ontology คือ ขอบเขตของความรู้ ส่วนที่เราต้องการจะถามเกี่ยวกับเรื่องนั้นๆ ซึ่งในที่นี้ เกี่ยวข้องกับฐานความรู้เกี่ยวกับอาณาจักรสัตว์ ผู้รับ คือ server1 และกระบวนการคิวรี ถูกเขียนขึ้นโดยใช้ภาษาจาวา ค่าของ parameter ต่อไปนี้คือ sender, receiver และ reply-with จะเป็นส่วน communication layer ในส่วน performative name (ในที่นี้คือ ask-one), language และ ontology จะเป็นส่วน message layer และส่วน เนืองาน จะเป็นส่วน content layer ใน KQML

ทางฝั่งเซิร์ฟเวอร์จะเก็บข้อมูลไว้ใน store procedure ต่างหาก และจะได้ตามเฟรม dog ก่อนแล้ว ค้นหาไปยังเฟรมต่างๆ ที่มีสล็อตเกี่ยวข้อง ซึ่ง เมื่อได้ผลลัพธ์แล้ว ทางฝั่งเซิร์ฟเวอร์ก็จะส่งกลับไปให้ไคลเอนท์ที่ทำการร้องขอ และจะทำการ encode ให้อยู่ในรูปแบบข้อความ KQML ดังนี้

(tell

```
:sender server1
:content "warm"
:receiver client1
:in-reply-to knowledge-base1
:language sql
:ontology animal-kingdom)
```