พ่อสมุดกลาง ศูนย์วิทยทรัพยากร จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 ออนโทโลจี (Ontology)

ออนโทโลจี [4] เป็นคำศัพท์ที่ยืมมาจากสาขาวิชาปรัชญา ซึ่งเดิมใช้หมายถึง "รายการอย่างมีแบบแผนของสิ่งที่มีอยู่" ("A Systematic Account of Existence") โดยคำว่าสิ่งที่มีอยู่ในที่นี้แสดงให้เห็นว่าจะต้องสามารถแสดงแทนสิ่งนั้นด้วยถ้อยคำได้ เมื่อนำออนโทโลจีมาใช้ในทางคอมพิวเตอร์ คำจำกัดความของออนโทโลจีที่ได้รับการยอมรับอย่างแพร่หลายคือ "ข้อกำหนดเกี่ยวกับแนวคิด" ("The Specification of a Conceptualization") ซึ่งนำมาเป็นข้อกำหนดในการใช้คำศัพท์เพื่อแสดงแทนสิ่งต่างๆ ในโดเมนที่สนใจ (Domain of Interest) ในรูปของอ็อบเจกต์ และความสัมพันธ์ระหว่างอ็อบเจกต์ เช่น การให้นิยามแก่คลาส อินสแตนซ์ (Instance) ความสัมพันธ์ พรอพเพอร์ตี กฎ (Rule) และอ็อบเจกต์อื่นๆ

ออนโทโลจีถูกออกแบบให้รองรับการแสดงแทนความรู้ (Knowledge Representation) สำหรับระบบการทำงานร่วมกัน (Sharing) และการนำกลับมาใช้ใหม่ (Reuse) [26] โดยอธิบายในรูปของข้อความที่คนทั่วไปเข้าใจได้ด้วยภาษาสำหรับแสดงแทนความรู้ (Knowledge Representational Language) ซึ่งมีความชัดเจนและเที่ยงตรงมากกว่าการใช้ภาษาธรรมชาติ (Natural Language) เพื่อให้โปรแกรมสามารถนำออนโทโลจีไปใช้ประโยชน์ได้ ซึ่งการอธิบายนี้จะอาศัยรูปแบบตามโครงสร้างของออนโทโลจีซึ่งประกอบด้วย [4,26]

1. คลาสหรือแนวคิด (Concept) สำหรับแสดงแทนกลุ่มของเอนทิตี (Entity) ในโดเมน สามารถมีสับคลาส (Subclass) เพื่อแสดงแทนแนวคิดที่เฉพาะเจาะจงกว่าของซูเปอร์คลาส (Superclass) ได้ ซึ่งเรียกว่าการแบ่งประเภท (Taxonomy)
2. ความสัมพันธ์หรือพรอพเพอร์ตี (Property) สำหรับแสดงแทนการติดต่อระหว่างคลาสในโดเมน สามารถมีสับพรอพเพอร์ตี (Subproperty) เพื่อแสดงแทนความสัมพันธ์ที่เฉพาะเจาะจงกว่าได้เช่นกัน
3. สัจพจน์ (Axiom) สำหรับกำหนดเงื่อนไขบังคับให้กับคลาส หรือกำหนดอาร์กิวเมนต์ (Argument) ของความสัมพันธ์
4. อินสแตนซ์ (Instance) สำหรับกำหนดอินสแตนซ์ของคลาส

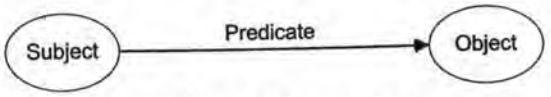
ตัวอย่างเช่น ถ้าต้องการอธิบายว่า "A student has a Name" สามารถทำได้โดยกำหนดให้ "Student" และ "Name" เป็นคลาส กำหนดความสัมพันธ์ระหว่างคลาสทั้งสองเป็น "has a" แล้วเติมอินสแตนซ์ให้กับทั้งสองคลาส อย่างไรก็ตาม แนวคิดถือเป็นเรื่องนามธรรมสำหรับการอธิบายโลกใน

มุมมองอย่างง่าย เพื่อจุดประสงค์ในการใช้งานอย่างใดอย่างหนึ่ง ดังนั้นแต่ละคนอาจสร้างออนโทโลจีที่แตกต่างกันเพื่ออธิบายเรื่องเดียวกัน [27]

2.1.2 อาร์ดีเอฟ (RDF: Resource Description Framework)

อาร์ดีเอฟ [28] เป็นภาษามาตรฐานเชิงความหมาย (W3C Recommendation) เพื่อนำมาใช้เป็นเฟรมเวิร์คในการแสดงแทนทรัพยากร (Resource) อันได้แก่สารสนเทศต่างๆ ที่อยู่บนเว็บ โดยได้รับการออกแบบให้แสดงแทนสารสนเทศได้อย่างยืดหยุ่นและมีเงื่อนไขบังคับน้อยที่สุด แต่ยังคงความสามารถในการนำสารสนเทศเหล่านั้นมาใช้งานร่วมกัน ทั้งนี้จุดมุ่งหมายหลักของภาษาอาร์ดีเอฟก็คือการสร้างแบบจำลองข้อมูลอย่างง่ายที่สามารถใช้งานกับความหมายเชิงรูปนัย (Formal Semantics) และรองรับการอนุมาน (Inference) โดยใช้ไวยากรณ์แบบภาษาเอ็กซ์เอ็มแอล (XML)

โครงสร้างทางไวยากรณ์ของอาร์ดีเอฟจะอธิบายทรัพยากรใดๆ ในรูปของกลุ่มของทริปเปิล (Triple) ซึ่งประกอบด้วยซับเจกต์ (Subject) เพรดิเคท (Predicate) หรือพรอพเพอร์ตี และอ็อบเจกต์ (Object) แสดงอยู่ในรูปของโหนดและเส้นเชื่อมระหว่างโหนด เรียกว่า อาร์ดีเอฟกราฟ (RDF Graph) ดังรูปที่ 2.1 ทั้งนี้ ทิศทางของเส้นเชื่อมมีความสำคัญมากเพราะจะชี้ไปยังปลายทางที่เป็นอ็อบเจกต์เสมอ ส่วนโหนดนั้นสามารถเป็นได้ทั้งซับเจกต์และอ็อบเจกต์



รูปที่ 2.1 อาร์ดีเอฟกราฟ

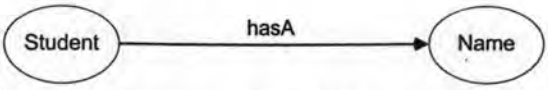
จากอาร์ดีเอฟกราฟจะสามารถบอกได้ว่าซับเจกต์มีความสัมพันธ์บางอย่างกับอ็อบเจกต์โดยความสัมพันธ์ดังกล่าวจะระบุไว้ที่เพรดิเคท

นอกเหนือจากการอธิบายในรูปแบบของอาร์ดีเอฟกราฟ ภาษาอาร์ดีเอฟยังอาจแสดงแทนได้ในรูปของอาร์ดีเอฟทริปเปิล (RDF Triple) ซึ่งนิยมเขียนในรูปของซับเจกต์ เพรดิเคทและอ็อบเจกต์เรียงต่อกันตามลำดับ หรืออาร์ดีเอฟ/ เอ็กซ์เอ็มแอล (RDF/ XML) โดยการแสดงแทนอาร์ดีเอฟตามโครงสร้างพื้นฐานของภาษาเอ็กซ์เอ็มแอล

2.1.3 อาวล์ (OWL: Web Ontology Language)

อาวล์ [29-30] เป็นภาษามาตรฐานเชิงความหมาย (W3C Recommendation) ที่นิยมใช้ประกาศออนโทโลจีเมื่อต้องการให้สามารถนำออนโทโลจีไปใช้งานร่วมกันได้ อาวล์ถูกพัฒนาขึ้นจากภาษาดีเอเอ็มแอลพลัสออยล์ (DAML + OIL: DARPA Agent Markup Language + Ontology Inference Layer) และพัฒนาขยายต่อจากการอธิบายด้วยภาษาอาร์ดีเอฟ/ เอ็กซ์เอ็มแอล จึงทำให้เป็นที่นิยมในการเขียนออนโทโลจีที่แสดงด้วยภาษาอาวล์ในรูปของอาร์ดีเอฟกราฟ ซึ่งสามารถแปลงให้อยู่ในรูปของอาร์ดีเอฟทริปเปิลเพื่อนำไปใช้ได้โดยง่าย ตัวอย่างเช่น การเขียนออนโทโลจีเพื่อแสดงข้อความ "A

Student has a Name." สามารถทำได้ในรูปของอาร์ดีเอฟกราฟ อาร์ดีเอฟทริปเปิลและอาร์ดีเอฟ/ เอ็กซ์เอ็มแอลได้ตามรูปที่ 2.2 ถึงรูปที่ 2.4 ตามลำดับ



รูปที่ 2.2 การเขียนแทนออนโทโลจีในรูปของอาร์ดีเอฟกราฟ

```
<Student> <hasA> <Name>
```

รูปที่ 2.3 การเขียนแทนออนโทโลจีในรูปของอาร์ดีเอฟทริปเปิล

```
<owl:Class rdf:ID="Student"/>
<owl:Class rdf:ID="Name"/>
<owl:ObjectProperty rdf:ID="hasA">
   <rdf:domain rdf:resource="#Student"/>
   <rdf:range rdf:resource="#Name"/>
</owl:ObjectProperty>
```

รูปที่ 2.4 การเขียนแทนออนโทโลจีในรูปของอาร์ดีเอฟ/ เอ็กซ์เอ็มแอล

ทั้งนี้อาวล์ได้รับการออกแบบเพื่อให้เป็นภาษาที่มีความสามารถในการอธิบายคลาสและความสัมพันธ์ระหว่างคลาสอันเป็นส่วนประกอบหลักพื้นฐานของออนโทโลจี เพื่อนำไปใช้งานในเอกสารและแอพลิเคชันบนเว็บ ซึ่งประโยชน์สำคัญที่จะได้รับจากการใช้ภาษาอาวล์คือ

- สามารถแสดงแทนข้อมูลของโดเมนใดๆ โดยการกำหนดเป็นคลาสและความสัมพันธ์ระหว่างคลาสที่เกิดขึ้นในโดเมนนั้น
- สามารถกำหนดอินสแตนซ์ที่เกี่ยวข้องให้กับคลาสและความสัมพันธ์ที่ได้กำหนดไว้ก่อนแล้วได้
- สามารถใส่กฎเพิ่มเติมเพื่อทำการอนุมานเกี่ยวกับคลาสและอินสแตนซ์ได้

นอกจากนี้ จุดเด่นของอาวล์ซึ่งทดแทนข้อจำกัดของอาร์ดีเอฟคือการที่อาวล์อนุญาตให้กำหนดลักษณะเฉพาะของความสัมพันธ์ (Property Characteristic) เช่น เป็นความสัมพันธ์แบบ Transitive, Symmetric, Functional หรือ InverseFunctional และข้อจำกัดของความสัมพันธ์ (Property Restriction) เช่น ข้อจำกัดด้านตัวบ่งปริมาณ (Quantifier Restriction) ข้อจำกัดด้านจำนวน (Cardinality Restriction) ในการอธิบายข้อมูลได้ ทำให้สามารถใช้ภาษาอาวล์ในการอธิบายว่า "นักเรียนแต่ละคนจะต้องมีชื่อเพียงชื่อเดียว" หรือ "นักเรียนทุกคนจะต้องศึกษาในคณะใดคณะหนึ่ง" ได้ โดยทั่วไปแล้ว องค์ประกอบพื้นฐานของอาวล์จะเกี่ยวข้องกับคลาส พรอพเพอร์ตีและอินสแตนซ์ ซึ่งส่วนประกอบที่มีการใช้งานในวิทยานิพนธ์นี้ได้แก่

1. คลาส แสดงด้วยข้อความ Class หรือ rdf:subClassOf เช่น

```
<owl:Class rdf:ID="Student"/>
```

2. พรอพเพอร์ตี แสดงด้วยข้อความ ObjectProperty, DatatypeProperty, rdfs:subPropertyOf, rdfs:domain หรือ rdfs:range เช่น

```
<owl:ObjectProperty rdf:ID="hasStudentName">
   <rdf:domain rdf:resource="#Student"/>
   <rdf:range rdf:resource="#StudentName"/>
</owl:ObjectProperty>
```

3. ชนิดของข้อมูล (Datatype) สามารถเลือกกำหนดและใช้ได้ตามชนิดข้อมูลที่มีในเอ็กซ์เอ็มแอลสกีมา (XML Schema) ซึ่งอ้างอิงได้จาก http://www.w3.org/2001/XMLSchema

4. ข้อจำกัดเกี่ยวกับพรอพเพอร์ตี ได้แก่ ข้อจำกัดด้านตัวบ่งปริมาณ allValueFrom ($\forall$) และ someValueFrom ($\exists$) เป็นต้น

5. การสมมูลกัน (Equivalence) ระหว่างคลาส ใช้เมื่อต้องการระบุว่าคลาสทั้งสองคลาสหมายถึงสิ่งเดียวกัน แสดงด้วยข้อความ equivalentClass เช่น

```
<owl:Class rdf:ID="Student">
   <owl:equivalentClass rdf:resource="#GradStudent"/>
</owl:Class>
```

6. ยูเนียน (Union) ใช้เมื่อต้องการระบุว่าคลาสหนึ่งกลุ่มเป็นส่วนประกอบของอีกคลาสหนึ่ง แสดงแทนด้วยข้อความ unionOf เช่น

```
<owl:Class rdf:ID="Address">
   <owl:unionOf rdf:parseType="Collection">
         <owl:Class rdf:about="#HomeNo"/>
         <owl:Class rdf:about="#HomeCity"/>
   </owl:unionOf >
</owl:Class>
```

ในปัจจุบันดับเบิลยูทรีซี (W3C) ซึ่งเป็นผู้กำหนดมาตรฐานภาษาอาวล์ได้แบ่งอาวล์ออกเป็น 3 ระดับตามความสามารถในการอธิบายออนโทโลจีจากน้อยไปมากคือ

1. อาวล์ไลท์ (OWL Lite) เป็นภาษาย่อยระดับต้นของอาวล์ สามารถรองรับการจัดแบ่งลำดับชั้นของคลาส และการใช้งานข้อบังคับเบื้องต้นของอาวล์ได้

2. อาวล์ดีแอล (OWL DL) เป็นภาษาย่อยระดับกลางของอาวล์ โดยดีแอลในที่นี้หมายถึงการอธิบายเชิงตรรกะ (Description Logic: DL) อาวล์ดีแอลสามารถอธิบายออนโทโลจีได้ละเอียดกว่าอาวล์ไลท์ และสามารถรองรับการใส่กฎเพิ่มเติมเพื่อทำการอนุมานได้

3. อาวล์ฟูลล์ (OWL Full) เป็นภาษาย่อยระดับสูงสุดของอาวล์ นิยมใช้เมื่อต้องการอธิบายออนโทโลจีให้มีความละเอียดสูงสุดโดยไม่จำเป็นต้องยึดติดกับไวยากรณ์ของอาร์ดีเอฟอีกต่อไป

2.1.4 การอนุมาน (Inference)

การอนุมาน [31] เป็นคุณสมบัติอย่างหนึ่งของภาษาอาวล์ที่เปิดให้ผู้ใช้สามารถใส่กฎ (Rule) เพิ่มเติมเพื่อให้ได้ข้อเท็จจริง (Fact) ใหม่เกี่ยวกับคลาสและอินสแตนซ์ที่ยังไม่ถูกแสดงไว้ในออนโทโลจีได้ ประโยชน์เพิ่มเติมที่ได้จากการอนุมานคือ ทำให้สามารถตรวจสอบความไม่ต้องกัน (Inconsistency) ที่อาจเกิดขึ้นในการสร้างออนโทโลจี และช่วยให้ทราบว่าคลาสใดเป็นสับคลาสกันบ้าง การอนุมานมาตรฐานที่มีใช้ในภาษาอาวล์ดีแอลมี 4 แบบได้แก่

1. การตรวจสอบ Subsumption (Subsumption Checking) เป็นการทดสอบว่า คลาสใหม่ที่สร้างขึ้นมาภายหลังโดยการกำหนดความหมายของคลาส จะมีคลาสใดที่มีอยู่เดิมเป็นสับคลาส หรือเป็นสับคลาสของคลาสใดที่มีอยู่เดิมบ้าง
2. การตรวจสอบความสมมูล (Equivalence Checking)
3. การตรวจสอบความต้องกัน (Consistency Checking) เป็นการทดสอบว่าคลาสสามารถมีอินสแตนซ์ได้จริงโดยไม่ขัดแย้งกับคลาสอื่นหรือไม่
4. การตรวจสอบการสร้างกรณีตัวอย่าง (Instantiation Checking)

อย่างไรก็ตาม หลักการอนุมานที่ใช้ในออนโทโลจีและในภาษาอาวล์เป็นการอนุมานแบบเปิด (Open-World Reasoning) ซึ่งมีความแตกต่างจากการอนุมานแบบปิด (Closed-World Reasoning) ที่ใช้ในงานวิจัยด้านอื่นๆ ลักษณะสำคัญของการอนุมานแบบเปิดคือ ทุกข้อความจะเป็นจริงจนกว่าจะได้รับการกำหนดหรือสามารถพิสูจน์ว่าข้อความนั้นเป็นเท็จ (ในขณะที่การอนุมานแบบปิดจะให้ทุกข้อความที่ไม่ถูกกำหนดว่าเป็นจริงเป็นข้อความเท็จ) ดังนั้นการใช้ออนโทโลจีเพื่อแสดงแทนความหมายใดๆ จำเป็นต้องกำหนดเงื่อนไขให้ชัดเจนเพื่อป้องกันความผิดพลาดของการอนุมานในภายหลัง

2.1.5 เมตาโมเดลของนิยามของออนโทโลจี (Ontology Definition MetaModel)

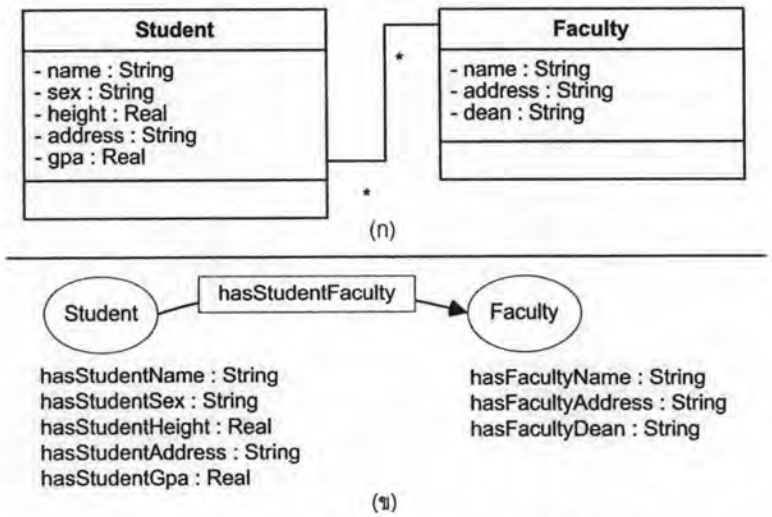
ตามที่โอเอ็มจี (OMG: Object Management Group) ได้ประกาศอาร์เอฟพี (RFP: Request For Proposal) ให้ผู้สนใจส่งเอกสารข้อเสนอเรื่อง Ontology Definition MetaModel (ODM) เพื่อสร้างเป็นข้อกำหนดเรื่องคำนิยามของออนโทโลจี ซึ่งสามารถใช้อธิบายออนโทโลจีด้วยเมตาโมเดลเอ็มโอเอฟรุ่นที่ 2 (MOF2 MetaModel) และนำมาใช้ในการแสดงแทนความรู้ รวมทั้งรองรับการทำงานแบบหลายออนโทโลจีได้ อาร์เอฟพีดังกล่าวได้กำหนดให้เอกสารข้อเสนอแต่ละชุดต้องระบุเมตาโมเดลของนิยามของออนโทโลจี ซึ่งสามารถแสดงความหมายของออนโทโลจีได้ในรูปของยูเอ็มแอล โดยใช้ข้อกำหนดของเมตาโมเดลเอ็มโอเอฟ ยูเอ็มแอล (UML) และโอซีแอล (OCL) รุ่นที่ 2.x รวมทั้งการแมประหว่างเมตาโมเดลกับยูเอ็มแอลโพรไฟล์รุ่นที่ 2 (UML2 Profile) การแมประหว่างโอดีเอ็ม (ODM) กับอาวล์และอื่นๆ

ด้วยข้อกำหนดของโอเอ็มจีนี้ ผู้วิจัยจึงได้นำเอกสารข้อเสนอในส่วนการแสดงความหมายของออนโทโลจีในรูปของยูเอ็มแอลมาใช้ประยุกต์เป็นแนวทางการแปลงระหว่างแบบจำลองข้อมูลเชิงวัตถุกับ

ออนโทโลจีได้ โดยจากผู้ยื่นข้อเสนอสี่ราย ผู้วิจัยเลือกใช้เอกสารข้อเสนอของดีเอสทีซี (DSTC) [21] ในส่วนการเทียบระหว่างออนโทโลจีที่แสดงด้วยภาษาอาวล์บนอาร์ดีเอฟสกีมา (RDF Schema) กับยูเอ็มแอลมาอ้างอิงดังนี้

1. Class ใช้เพื่อกำหนดกลุ่มอ็อบเจกต์ที่เกี่ยวข้องกันเนื่องจากใช้คุณสมบัติบางอย่างร่วมกัน เทียบได้กับคลาสในยูเอ็มแอล
2. rdfs:subClassOf ใช้เพื่อกำหนดสับคลาสให้กับคลาสใดคลาสหนึ่ง เทียบได้กับสับคลาสในยูเอ็มแอล
3. rdfs:Property ใช้แสดงความสัมพันธ์ระหว่างอินสแตนซ์หรือจากอินสแตนซ์ไปยังค่าของข้อมูล แบ่งเป็น
   - ObjectProperty ใช้แสดงความสัมพันธ์ระหว่างอินสแตนซ์ของคลาสสองคลาส เทียบได้กับความสัมพันธ์แบบ Association ในยูเอ็มแอล
   - DatatypeProperty ใช้แสดงความสัมพันธ์จากอินสแตนซ์ของคลาสไปยังค่าของข้อมูล เทียบได้กับแอททริบิวต์ในยูเอ็มแอล
4. rdfs:subProperty ใช้เพื่อกำหนดสับพรอพเพอร์ตีให้กับพรอพเพอร์ตีที่มีอยู่ เทียบได้กับความสัมพันธ์แบบ Sub-association หรือ Sub-type ในยูเอ็มแอล
5. rdfs:domain ใช้เพื่อกำหนดว่าพรอพเพอร์ตีที่สร้างขึ้นใช้งานกับอ็อบเจกต์ในโดเมนใดได้บ้าง เทียบได้กับต้นทางความสัมพันธ์แบบ Association ของคลาสในยูเอ็มแอล
6. rdfs:range ใช้เพื่อกำหนดว่าพรอพเพอร์ตีที่สร้างขึ้นใช้งานกับอ็อบเจกต์ในเรนจ์ใดได้บ้าง เทียบได้กับปลายทางความสัมพันธ์แบบ Association ของคลาสในยูเอ็มแอล

ตัวอย่างการแปลงแบบจำลองข้อมูลเชิงวัตถุที่แสดงด้วยยูเอ็มแอลคลาสไดอะแกรมให้แสดงในรูปของออนโทโลจีแสดงไว้ในรูปที่ 2.5



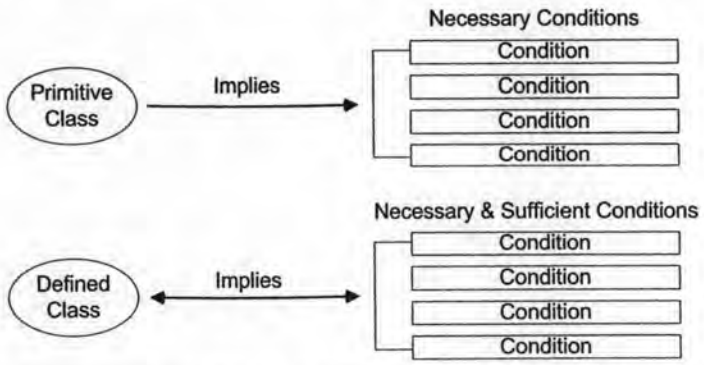
รูปที่ 2.5 ตัวอย่างการแปลงแบบจำลองข้อมูลเชิงวัตถุ (ก) เป็นออนโทโลจีในรูปของอาร์ดีเอฟกราฟ (ข)

2.1.6 โพรเทเจ (Protégé)

โพรเทเจ [32-33] เป็นเครื่องมือสำหรับสร้างออนโทโลจีเพื่อใช้เป็นแบบจำลองข้อมูลของโดเมนหรือแอพพลิเคชันเชิงฐานความรู้ใดๆ (Knowledge-Based Application) โดยมีโพรเทเจ-อาวล์ (Protégé-OWL) ซึ่งเป็นส่วนขยายของโพรเทเจทำหน้าที่สนับสนุนการใช้งานอาวล์ในการสร้างออนโทโลจี

ตามข้อกำหนดจากดับเบิลยูทรีซี อาวล์ออนโทโลจีที่สร้างขึ้นควรจะประกอบด้วยการอธิบายคลาส ความสัมพันธ์ระหว่างคลาสและอินสแตนซ์ที่เกี่ยวข้อง และเมื่อนำออนโทโลจีที่สร้างขึ้นมาพิจารณาร่วมกับ ความหมายเชิงรูปนัยจากอาวล์ (OWL Formal Semantics) ยังช่วยบอกให้ทราบได้ว่าจะได้รับข้อมูลอื่นๆ เพิ่มขึ้นตามมาได้อย่างไร เช่น การหาข้อเท็จจริงที่ไม่ได้แสดงโดยตรงในออนโทโลจีที่สร้าง แต่เป็นผลมาจากความหมายอื่นประกอบกัน ทั้งนี้ ความหมายของคลาสที่กำหนดในโพรเทเจ-อาวล์สามารถแบ่งได้เป็น 2 ประเภท [33-34] แสดงไว้ตามรูปที่ 2.6 คือ

1. คลาสแบบ Primitive (Primitive Class) หรือ Partial Class กำหนดโดยใช้เงื่อนไขจำเป็น (Necessary Condition) เพื่อระบุว่าอินสแตนซ์ใดจะเป็นสมาชิกของคลาสนั้นได้จะต้องมีคุณสมบัติตรงตามเงื่อนไขเหล่านี้
2. คลาสแบบ Defined (Defined Class) หรือ Complete Class กำหนดโดยใช้เงื่อนไขจำเป็นและเพียงพอ (Necessary & Sufficient Condition) เพื่อระบุว่าอินสแตนซ์ใดจะเป็นสมาชิกของคลาสนั้นได้จะต้องมีคุณสมบัติตรงตามเงื่อนไขเหล่านี้ และในทางกลับกันหากอินสแตนซ์ใดมีคุณสมบัติตรงตามเงื่อนไขเหล่านี้แสดงว่าอินสแตนซ์นั้นเป็นสมาชิกของคลาส



รูปที่ 2.6 ลักษณะของคลาสแบบ Primitive และคลาสแบบ Defined ที่สร้างจากโพรเทเจ-อาวล์

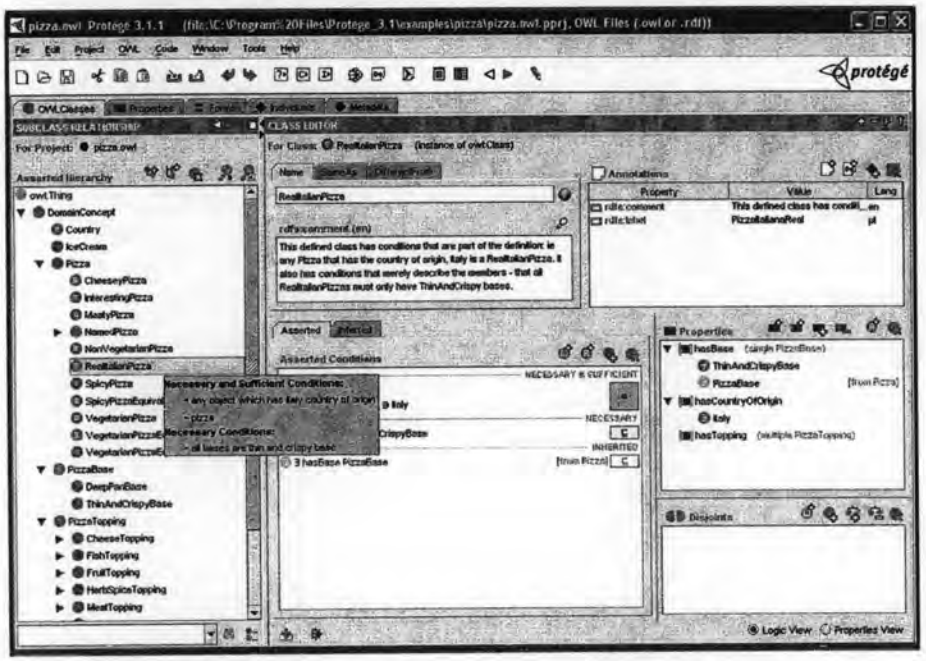
ด้วยความสามารถของโปรแกรมโพรเทเจ-อาวล์ ผู้ใช้จะสามารถสร้างออนโทโลจีได้โดยง่ายและได้รับความสะดวกในการจัดการกับออนโทโลจี ดังนี้

1. เรียกใช้ (Load) และบันทึก (Save) ออนโทโลจีในรูปแบบอาวล์และอาร์ดีเอฟ
2. ตรวจแก้ (Edit) และแสดงผลคลาสและความสัมพันธ์ที่สร้างไว้
3. กำหนดลักษณะเฉพาะเชิงตรรกะของคลาส (Logical Class Characteristic) ในรูปของนิพจน์ภาษาอาวล์ (OWL Expression)

4. เรียกใช้เครื่องมืออนุมาน (Inference Engine) อื่นได้

5. แก้ไขอินสแตนซ์ของอาวล์เพื่อนำไปใช้งานกับเว็บเชิงความหมายได้

ตัวอย่างหน้าจอการทำงานของโปรแกรมโพรเทเจ-อาวล์แสดงไว้ในรูปที่ 2.7 โดยสามารถศึกษาวิธีการใช้งานโปรแกรมโดยละเอียดได้จากเอกสาร "A practical guide to building OWL ontologies using the Protégé-OWL plugin and CO-ODE tools [33]"



รูปที่ 2.7 หน้าจอการทำงานของโปรแกรมโพรเทเจ-อาวล์

2.1.7 จีนา (Jena)

จีนา [35] เป็นส่วนต่อประสานโปรแกรมประยุกต์หรือเอพีไอภาษาจาวา (Java API) เพื่อสร้างและดำเนินการกับอาร์ดีเอฟกราฟ ตัวอย่างเช่น การสร้างอาร์ดีเอฟกราฟ การกำหนดทรัพยากรและพรอพเพอร์ตีให้กับอาร์ดีเอฟกราฟ การแสดงรายการของทรัพยากร การเข้าถึงข้อมูลในอาร์ดีเอฟกราฟ การค้นคืนค่าจากอาร์ดีเอฟกราฟ และการค้นหาอาร์ดีเอฟกราฟ ดังนั้น จีนาจึงได้รับความนิยมนำมาใช้เป็นชุดเครื่องมือ (Toolkit) สำหรับพัฒนาโปรแกรมประยุกต์ในงานด้านเว็บเชิงความหมาย

ถึงแม้ว่าจุดประสงค์หลักของจีนาคือการนำไปใช้งานกับอาร์ดีเอฟก็ตาม จีนายังสามารถทำงานกับออนโทโลจีที่แสดงแทนด้วยภาษาอื่นๆ ได้แก่ อาร์ดีเอฟเอส (RDFS: Resource Description Framework Schema) อาวล์และดีเอเอ็มแอลพลัสออยล์ได้ ซึ่งจีนาจะทำหน้าที่จัดเตรียมตัวต่อประสานการโปรแกรม (Programming Interface) ไปยังออนโทโลจีเพื่อสนับสนุนการพัฒนาโปรแกรมโดยไม่คำนึงว่าออนโทโลจีดังกล่าวถูกแสดงแทนด้วยภาษาใด

นอกจากนี้ จีนายังมีความสามารถรองรับการทำงานของออนโทโลจีร่วมกับฐานข้อมูลเชิงสัมพันธ์ (Relational Database) [36] และการอนุมานโดยใช้กฎ [37] ได้อีกด้วย

2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง

2.2.1 งานวิจัยเกี่ยวกับการรวมแบบจำลองข้อมูลเชิงวัตถุ

งานวิจัยเกี่ยวกับการรวมแบบจำลองข้อมูลเชิงวัตถุมีอยู่มากมายในปัจจุบัน ซึ่งแต่ละงานวิจัยมีจุดเด่นแตกต่างกันไปขึ้นกับการนำไปใช้งาน เนื่องจากผู้วิจัยต้องการนำขั้นตอนวิธีการรวมแบบจำลองข้อมูลเชิงวัตถุมาประยุกต์ใช้เป็นขั้นตอนวิธีการรวมออนโทโลจี ซึ่งขั้นตอนวิธีที่ต้องการควรจะครอบคลุมถึงการวิเคราะห์ในเชิงความหมายและความสัมพันธ์ระหว่างคลาสอันเป็นลักษณะสำคัญของออนโทโลจี รวมทั้งจะส่งผลต่อส่วนของการออกแบบออนโทโลจีเพื่อแสดงแบบจำลองข้อมูลเชิงวัตถุ และขั้นตอนวิธีการรวมออนโทโลจี โดยผู้วิจัยได้เลือกงานวิจัย [20] ซึ่งมีรายละเอียดดังนี้

2.2.1.1 วิธีการรวมแบบจำลองข้อมูลเชิงวัตถุโดยใช้วิทยาการศึกษาสำนึกและการวิเคราะห์ความสัมพันธ์ระหว่างคลาส [20]

งานวิจัยนี้นำเสนอวิธีการรวมแบบจำลองข้อมูลเชิงวัตถุหนึ่งคู่ โดยพิจารณาจากความสัมพันธ์ระหว่างคลาสและความหมายที่เพิ่มให้กับแบบจำลองข้อมูล แล้วกำหนดเป็นค่าความสัมพันธ์เพื่อวิเคราะห์ว่าแบบจำลองข้อมูลรวมควรเป็นอย่างไร หลังจากนั้นจะนำแบบจำลองข้อมูลรวมที่ได้จากวิธีการนี้ไปเปรียบเทียบกับแบบจำลองข้อมูลรวมที่สร้างขึ้นโดยผู้เชี่ยวชาญ งานวิจัยได้นำแนวคิดเรื่องค่าวิทยาการศึกษาสำนึก (Heuristics) มาใช้ปรับหาค่าความสัมพันธ์ที่เหมาะสมเพื่อกำหนดให้กับความสัมพันธ์แต่ละรูปแบบในการรวมแบบจำลองข้อมูลว่าควรเป็นค่าใด โดยมีขั้นตอนดังนี้

1. ขั้นตอนการเตรียมแบบจำลองข้อมูล ประกอบด้วย 2 ขั้นตอนย่อยคือ
   - เพิ่มความหมายให้กับแบบจำลองข้อมูล ได้แก่ การเพิ่มความหมายให้กับคลาส และการเพิ่มความหมายให้กับแอททริบิวต์
   - ระบุคำที่มีความหมายเหมือนกันหรือคำที่เป็นส่วนประกอบกัน
2. ขั้นตอนการเปรียบเทียบแบบจำลองข้อมูล ประกอบด้วย 3 ขั้นตอนย่อยคือ
   - เปรียบเทียบความหมายของคลาสหรือความหมายแอททริบิวต์ซึ่งได้ระบุเพิ่มเติมไว้ในขั้นตอนที่ 1 โดยคิดจากชื่อของความหมายและค่าของความหมายที่กำหนดให้กับคลาสหรือแอททริบิวต์แต่ละคู่ทีละคู่จนครบ ผลที่ได้จากขั้นตอนนี้คือระดับความสัมพันธ์ระหว่างคลาสหรือแอททริบิวต์แต่ละคู่
   - เปรียบเทียบแอททริบิวต์ พิจารณาจากชื่อของแอททริบิวต์ ความหมายของแอททริบิวต์ ชนิดข้อมูลของแอททริบิวต์ ข้อกำหนดบูรณภาพของแอททริบิวต์ (Integrity Constraint) และระดับความสัมพันธ์ระหว่างแอททริบิวต์ที่ได้จากขั้นตอนก่อนหน้านี้ ผลที่ได้จากขั้นตอนนี้คือระดับความเหมือนหรือแตกต่างเมื่อเทียบกับแอททริบิวต์อื่น และคำแนะนำว่าสามารถรวมแอททริบิวต์นั้นเข้ากับแอททริบิวต์อื่นได้หรือไม่

- เปรียบเทียบคลาส กำหนดค่าความสัมพันธ์ระหว่างคลาสโดยพิจารณาได้จากระดับความสัมพันธ์ระหว่างคลาสจากขั้นตอนการเปรียบเทียบความหมาย ร่วมกับระดับความเหมือนหรือแตกต่างระหว่างชุดแอททริบิวต์ในคลาสนั้นที่ได้จากขั้นตอนการเปรียบเทียบแอททริบิวต์ ผลที่ได้คือค่าความสัมพันธ์ระหว่างคลาส ซึ่งในที่นี้จะแบ่งเป็น 5 รูปแบบคือ แบบเท่ากัน (Equivalence) แบบซูเปอร์คลาส (Superclass) แบบสับคลาส (Subclass) แบบพี่น้อง (Sibling) และแบบแตกต่างกัน (Disjoint)

3. ขั้นตอนการรวมแบบจำลองข้อมูล ประกอบด้วย 3 ขั้นตอนย่อยคือ
   - รวมความหมายของคลาสหรือแอททริบิวต์ โดยการรวมความหมายของคลาสจะเกิดขึ้นเมื่อคลาสคู่นั้นมีความสัมพันธ์ระหว่างคลาสแบบเท่ากัน แบบซูเปอร์คลาสหรือแบบสับคลาส ส่วนการรวมความหมายของแอททริบิวต์จะเกิดขึ้นเมื่อมีการรวมแอททริบิวต์คู่ใดเข้าด้วยกัน ซึ่งความหมายรวมที่ได้จะต้องเป็นผลจากการยูเนียน (Union) ระหว่างความหมายเดิม
   - รวมแอททริบิวต์ ทำได้โดยการรวมความหมายของแอททริบิวต์ตามขั้นตอนก่อนหน้านี้ (ซึ่งในบางกรณีอาจต้องสร้างฟังก์ชันเพื่อแปลงความหมายด้วย) ตรวจสอบชนิดข้อมูลของแอททริบิวต์และสร้างฟังก์ชันเพื่อปรับให้ตรงกัน แล้วพิจารณาข้อกำหนดบูรณภาพของแอททริบิวต์ เพื่อหาค่าที่เป็นได้ทั้งหมดของแอททริบิวต์ พร้อมทั้งพิจารณาว่ามีแอททริบิวต์ใดหรือไม่ที่ค่าของแอททริบิวต์ขึ้นกับแอททริบิวต์ที่ถูกแปลงค่า หากมีจะต้องเพิ่มฟังก์ชันการแปลงค่าแอททริบิวต์นั้นด้วย
   - รวมคลาส พิจารณาจากค่าความสัมพันธ์ระหว่างคลาสทั้ง 5 รูปแบบที่ได้จากการเปรียบเทียบคลาสและลำดับความสำคัญของความสัมพันธ์ที่คลาสหนึ่งในแบบจำลองข้อมูลหนึ่งมีผลต่อคลาสต่างๆ ในอีกแบบจำลองหนึ่ง โดยมีลำดับการรวมคลาสเริ่มจากคลาสที่มีความสัมพันธ์แบบเท่ากัน แบบซูเปอร์คลาส / แบบสับคลาส และแบบพี่น้องตามลำดับ ส่วนความสัมพันธ์แบบแตกต่างกันจะแสดงว่าทั้งสองคลาสไม่เกี่ยวข้องกันจึงไม่นำมารวม จากนั้นจึงพิจารณาจากในแบบจำลองข้อมูลรวมที่ได้ว่า มีคลาสคู่ใดที่มีความสัมพันธ์แบบแอกกรีเกชัน (Aggregation) ปรากฏอยู่หรือไม่ ถ้ามีจะทำการรวมแอททริบิวท์ในคลาสเหล่านั้นที่มีความเกี่ยวข้องกันเข้าด้วยกันแทน

ผลลัพธ์ที่ได้จากงานวิจัยนี้คือข้อเสนอแนะว่า ในการรวมแบบจำลองข้อมูลเชิงวัตถุสองชุดเข้าด้วยกัน ควรจะได้แบบจำลองข้อมูลรวมเป็นอย่างไร แล้วจึงนำคำแนะนำดังกล่าวมาสร้างเป็นวิว (View) บนฐานข้อมูลเชิงวัตถุเดิม เพื่อเป็นส่วนติดต่อระหว่างผู้ที่ต้องการใช้งานฐานข้อมูลทั้งสองร่วมกัน กับการสอบถามไปยังฐานข้อมูลจริง

ผู้วิจัยได้นำวิธีการรวมแบบจำลองข้อมูลเชิงวัตถุจากงานวิจัยนี้มาประยุกต์ใช้กับออนโทโลจี เพื่อพัฒนาเป็นวิธีการรวมออนโทโลจีซึ่งจะนำมาใช้กับงานของผู้วิจัย

2.2.2 งานวิจัยเกี่ยวกับการนำออนโทโลจีมาใช้ในการรวมแบบจำลองข้อมูล

งานวิจัยเกี่ยวกับการนำออนโทโลจีมาใช้ในการรวมแบบจำลองข้อมูลอาศัยความสามารถของออนโทโลจีในการทำหน้าที่เป็นตัวกลาง (Mediator) ติดต่อระหว่างผู้ใช้กับแบบจำลองข้อมูลหรือระหว่างแบบจำลองข้อมูลแต่ละชุดกับแบบจำลองข้อมูลรวมหรือทั้งสองหน้าที่ร่วมกัน ในงานวิจัย [5] ได้แบ่งแนวทางในการนำออนโทโลจีมาใช้เพื่อการรวมข้อมูลออกเป็น 3 แนวทางตามวิธีการนำออนโทโลจีมาใช้อธิบายแบบจำลองข้อมูลคือ แนวคิดแบบออนโทโลจีเดียว แนวคิดแบบหลายออนโทโลจี และแนวคิดแบบผสม ตัวอย่างงานวิจัยที่นำแนวคิดทั้งสามไปใช้ ได้แก่

2.2.2.1 Semantic Data Integration in Hierarchical Domains [7]

งานวิจัยนี้เป็นตัวอย่างการนำแนวคิดแบบออนโทโลจีเดียวมาใช้ โดยลักษณะสำคัญของแนวคิดแบบออนโทโลจีเดียวคือ การสร้างโกลบอลออนโทโลจีหนึ่งชุดเพื่อใช้อธิบายแบบจำลองข้อมูลทั้งหมดร่วมกัน มีจุดประสงค์สำคัญในการรวมข้อมูลจากหลายแหล่งข้อมูล และนำมาแสดงให้ผู้ใช้เห็นเสมือนว่าข้อมูลทั้งหมดถูกจัดเก็บไว้ในแหล่งข้อมูลเดียว โดยซ่อนความแตกต่างอันเป็นผลจากการออกแบบแหล่งข้อมูลจากผู้ใช้งานข้อมูล งานวิจัยนี้ยกตัวอย่างการจัดการแหล่งข้อมูลที่แสดงแทนด้วยเอกสารเอ็กซ์เอ็มแอล (XML: eXtensible Markup Language) แบบโดเมนเชิงลำดับชั้น (Hierarchical Domain) ซึ่งการจัดเรียงจะแตกต่างกันไปในแต่ละแหล่งข้อมูล ด้วยแนวคิดนี้ข้อมูลแต่ละฟิลด์ในทุกแหล่งข้อมูลจะถูกแมปไปยังโกลบอลออนโทโลจีที่สร้างขึ้นที่ส่วนกลาง

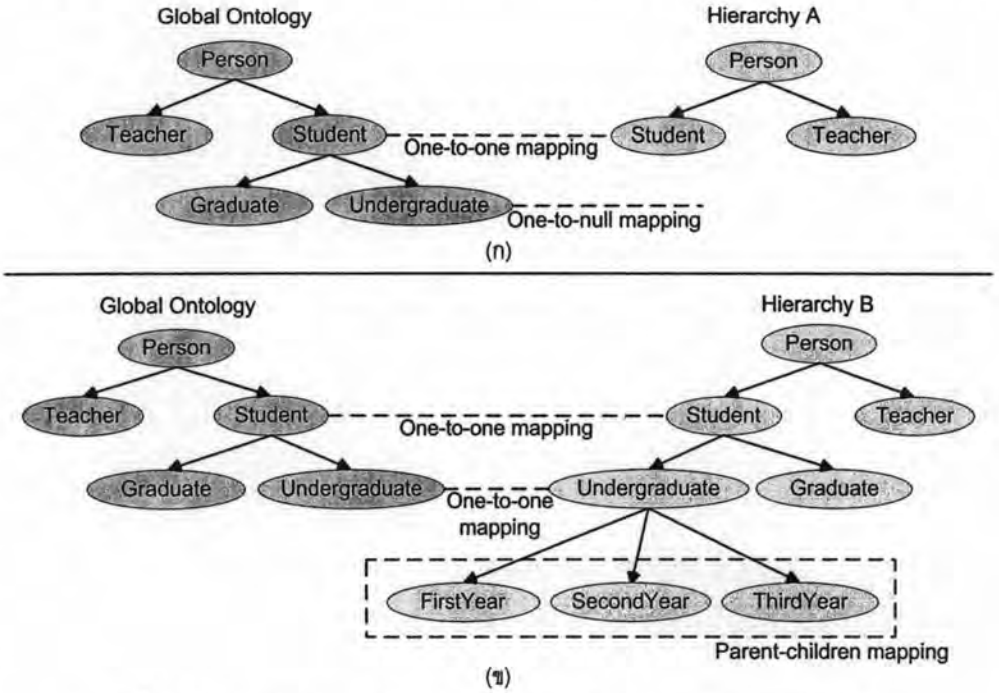
งานวิจัยนี้เสนอแนวคิดในการแมประหว่างแหล่งข้อมูลกับโกลบอลออนโทโลจีดังรูปที่ 2.8 เริ่มต้นจากการสร้างโกลบอลออนโทโลจีขึ้นมาหนึ่งชุด หากลำดับชั้น (Hierarchy) ใดของข้อมูลตรงกับที่กำหนดไว้ในโกลบอลออนโทโลจี การแมปจะเกิดขึ้นแบบหนึ่งต่อหนึ่ง (One-to-one Mapping) (รูปที่ 2.8(ก) และ (ข)) หากไม่มีลำดับชั้นตามที่โกลบอลออนโทโลจีกำหนดไว้ การแมปจะเป็นแบบหนึ่งต่อศูนย์ (One-to null Mapping) (รูปที่ 2.8(ก)) และหากลำดับชั้นใดของข้อมูลมิได้ถูกกำหนดไว้ในโกลบอลออนโทโลจี จะกำหนดให้แมปไปยังลำดับชั้นที่สูงกว่าแทน เรียกว่าการแมปแบบ Parent-children (Parent-children Mapping) (รูปที่ 2.8(ข))

งานวิจัยนี้แสดงให้เห็นได้อย่างชัดเจนว่า แม้แนวคิดแบบออนโทโลจีเดียวจะทำให้ง่ายต่อการจัดการคือมีผู้ออกแบบออนโทโลจีหนึ่งคนสร้างโกลบอลออนโทโลจีไว้ แล้วให้เจ้าของข้อมูลแต่ละคนเลือกแมปข้อมูลของตนไปยังออนโทโลจีดังกล่าว แต่การออกแบบโกลบอลออนโทโลจีเป็นไปได้ยาก เนื่องจากจำเป็นต้องเข้าใจภาพรวมของข้อมูลทั้งหมดเพื่อให้ออนโทโลจีที่ได้ถูกต้อง ครบถ้วน และสามารถรองรับการนำไปใช้ได้จริง นอกจากนั้นการแมปแบบ Parent-children แต่ละครั้งจะทำให้สูญเสียความหมายของข้อมูลในลำดับชั้นนั้นไปทันที

ดังนั้น แม้ว่าผู้วิจัยจะเลือกนำวิธีการแมปจากโกลบอลออนโทโลจีไปยังแหล่งข้อมูลมาใช้เป็นต้นแบบในการแมปจากออนโทโลจีรวมที่สร้างขึ้นไปยังแบบจำลองข้อมูลก็ตาม ผู้วิจัยยังให้

หอสมุดกลาง ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ความสำคัญกับที่มาของออนโทโลจีรวมที่จะต้องสร้างขึ้นเพื่อให้ครอบคลุมความหมายทั้งหมดของแบบจำลองข้อมูลเดิมทุกตัว และมิให้ความหมายของข้อมูลในลำดับชั้นใดๆ สูญหายระหว่างการรวม



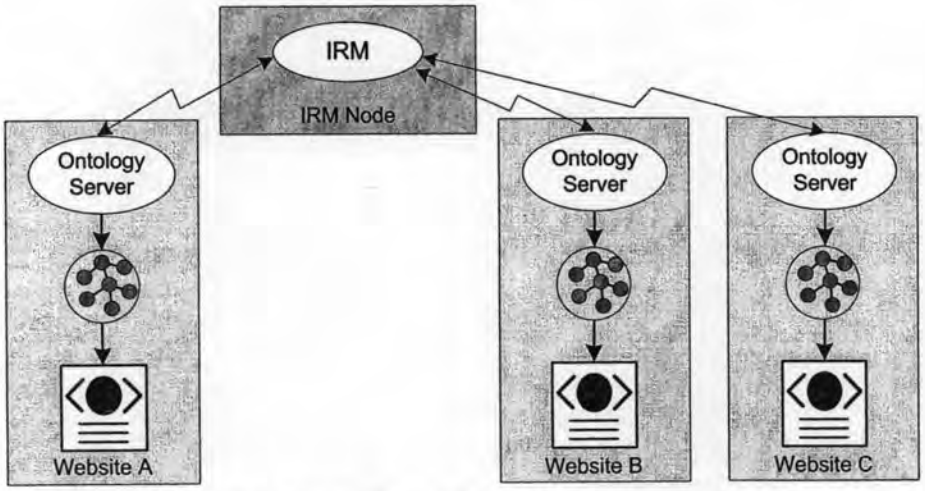
รูปที่ 2.8 ตัวอย่างการแมปจากลำดับชั้นข้อมูลไปยังโกลบอลออนโทโลจี [7]

2.2.2.2 OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies [8]

งานวิจัย OBSERVER (Ontology Based System Enhanced with Relationships for Vocabulary hEterogeneity Resolution) เป็นตัวอย่างการนำแนวคิดแบบหลายออนโทโลจีมาใช้ โดยลักษณะสำคัญของแนวคิดแบบหลายออนโทโลจีคือ การสร้างโลคัลออนโทโลจีหนึ่งชุดมาใช้อธิบายแบบจำลองข้อมูลแต่ละชุดแยกกัน แล้วสร้างแมปปิงระหว่างออนโทโลจีทั้งหมด ภายในงานวิจัยนี้นำเสนอวิธีการประมวลผลข้อคำถามในระบบสารสนเทศขนาดใหญ่ที่ข้อมูลถูกจัดเก็บในรูปของเว็บเพจ (Web Page) ซึ่งไม่มีการจัดการสารสนเทศแบบรวมศูนย์ (Centralized Information Management) ดังนั้นข้อมูลแต่ละชุดเป็นอิสระต่อกันอย่างสิ้นเชิง การใช้ออนโทโลจีช่วยให้สามารถสอบถามข้อมูลด้วยคำถามในเชิงความหมายได้เพิ่มจากการใช้คำหลัก (Keyword) เพื่อสอบถามตามแบบเดิม

ในส่วนแนวคิดแบบหลายออนโทโลจี งานวิจัยนี้เปิดให้แต่ละเว็บสร้างออนโทโลจีของตนเองอย่างอิสระโดยมีออนโทโลจีเซิร์ฟเวอร์ (Ontology Server) คอยควบคุม และในระบบจะมีตัวจัดการความสัมพันธ์ระหว่างออนโทโลจี (Interontology Relationship Manager (IRM)) ทำหน้าที่เป็นสารบัญความหมาย (Catalog of Semantics) จากทุกออนโทโลจี โดยผู้ใช้จะติดต่อผ่านออนโทโลจีที่เลือกใช้ (Selected Ontology) ชุดใดชุดหนึ่งจากออนโทโลจีทั้งหมด หลังจากนั้น เมื่อผู้ใช้สอบถาม

ไปยังออนโทโลจีที่เลือก ข้อคำถามจะถูกสร้างขึ้นโดยใช้คำศัพท์ของออนโทโลจีนั้น ก่อนจะตรวจสอบไปที่ตัวจัดการความสัมพันธ์ระหว่างออนโทโลจี เพื่อแปลเป็นข้อคำถามที่ใช้คำศัพท์ของออนโทโลจีปลายทางที่ต้องการ (Target Ontology) ตัวอย่างการทำงานของระบบ OBSERVER แสดงไว้ในรูปที่ 2.9



รูปที่ 2.9 องค์ประกอบของระบบตามงานวิจัย OBSERVER [8]

ในงานวิจัย OBSERVER นี้ ตัวจัดการความสัมพันธ์ระหว่างออนโทโลจี (ไอออาร์เอ็ม) ตามรูปที่ 2.9 เป็นส่วนประกอบสำคัญที่ใช้ในการรวมออนโทโลจีโดยสร้างเป็นการสอบถามข้ามออนโทโลจี เนื่องจากเป็นส่วนที่เก็บคุณสมบัติเชิงความหมายที่กำหนดระหว่างออนโทโลจีแต่ละชุด โดยทำหน้าที่จัดการกับปัญหาเรื่องการใช้คำศัพท์ที่แตกต่างกันในแต่ละออนโทโลจี ซึ่งภายในงานวิจัยนี้เลือกแก้ปัญหาความสัมพันธ์ระหว่างคำศัพท์ 6 รูปแบบ ได้แก่

1. ความสัมพันธ์แบบความหมายเหมือนกัน (Synonym Relationship)
2. ความสัมพันธ์แบบคำลูกกลุ่ม (Hyponym Relationship)
3. ความสัมพันธ์แบบคำแม่กลุ่ม (Hypernym Relationship)
4. ความสัมพันธ์แบบซ้อนเหลื่อมกัน (Overlap Relationship)
5. ความสัมพันธ์แบบแตกต่างกัน (Disjoint Relationship)
6. ความสัมพันธ์แบบครอบคลุม (Covering Relationship)

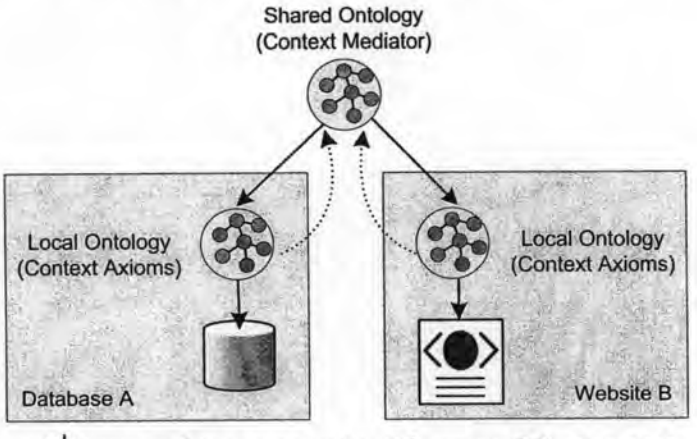
ผู้วิจัยเลือกนำเฉพาะแนวคิดการสร้างออนโทโลจีหนึ่งชุดสำหรับอธิบายข้อมูลแต่ละชุดจากงานวิจัยนี้ไปใช้ แต่มิได้นำขั้นตอนวิธีในการรวมออนโทโลจีโดยสร้างเป็นการสอบถามข้ามออนโทโลจี เนื่องจากเล็งเห็นว่าการสร้างสารบัญความหมายเพื่อเก็บความสัมพันธ์ระหว่างคำศัพท์ทั้งหมดจะทำให้ระบบยากต่อการพัฒนา และเกิดปัญหาในการใช้งานหากออนโทโลจีเปลี่ยนแปลง นอกจากนี้งานวิจัยนี้เลือกพิจารณาเฉพาะความสัมพันธ์ด้านความหมายของคำศัพท์เท่านั้น ดังนั้นหากข้อมูลที่

เก็บมีเงื่อนไขบังคับอื่น ตัวอย่างเช่น หน่วยการนับที่ต่างกัน เช่น แต่ละเว็บอาจแสดงหน่วยเงินเป็นดอลลาร์ เป็นปอนด์ หรือเป็นบาทไม่ตรงกัน ซึ่งอาจทำให้ผลการรวมคลาดเคลื่อนได้

2.2.2.3 Context Interchange: New Features and Formalisms for the Intelligent Integration of Information [9]

งานวิจัยนี้เป็นตัวอย่างการนำแนวคิดในการรวมข้อมูลโดยใช้ออนโทโลจีแบบผสมมาใช้ โดยลักษณะสำคัญของแนวคิดแบบผสมคือ การสร้างโลคัลออนโทโลจีหนึ่งชุดมาใช้อธิบายแบบจำลองข้อมูลแต่ละชุดแยกกัน แล้วสร้างแชร์ดออนโทโลจีขึ้นมาอีกชั้นหนึ่งเพื่อเก็บคำศัพท์ที่โลคัลออนโทโลจีแต่ละชุดใช้ร่วมกัน เฟรมเวิร์ค COIN (COntext INterchange) ทำงานในลักษณะของการสับเปลี่ยนบริบทโดยอาศัยแนวคิดเชิงตัวกลาง (Mediator-based) เพื่อจัดการด้านความหมายระหว่างข้อมูลต้นทางและปลายทางที่แตกต่างกัน โดยมุ่งเน้นการแก้ปัญหาความแตกต่างด้านความหมายของข้อมูลที่ไม่ได้รับการกำหนดไว้ล่วงหน้าโดยอาศัยการพิจารณาจากบริบท (Context) ของข้อมูล ดังนั้นเฟรมเวิร์คนี้ต้องมีความสามารถในการตรวจหาและปรับความแตกต่างที่พบในการใช้งานระหว่างข้อมูลแต่ละชุดไม่ว่าจะเป็นข้อมูลในแหล่งข้อมูลหรือข้อมูลจากเว็บก็ตาม

ภายในเฟรมเวิร์ค COIN (รูปที่ 2.10) จะมีตัวกลางด้านบริบท (Context Mediator) อยู่ภายในแชร์ดออนโทโลจีทำหน้าที่เปรียบเทียบสัจพจน์ของบริบท (Context Axiom) ซึ่งอยู่ในโลคัลออนโทโลจีแต่ละชุดที่ต้องการแลกเปลี่ยนข้อมูลกัน เมื่อมีการส่งการสอบถามเข้ามาในระบบตัวกลางด้านบริบทจะเข้ามารับข้อคำถามนั้นเพื่อแปลงให้อยู่ในรูปของการสอบถามของตัวกลาง (Mediator Query) แล้วจึงแปลเป็นแผนการสอบถาม (Query Plan) อีกต่อหนึ่ง ก่อนที่จะเรียกให้แผนการสอบถามทำงาน เพื่อส่งการสอบถามย่อยไปยังแต่ละระบบ เก็บผลลัพธ์ที่ได้ แปลงรูปแบบของผลลัพธ์ให้เหมาะสมและส่งคำตอบรวมที่ได้กลับไปยังผู้ใช้



รูปที่ 2.10 องค์ประกอบของเฟรมเวิร์คของงานวิจัย COIN [9]

ข้อแตกต่างสำคัญของงานวิจัยนี้กับงานของผู้วิจัยคือ วิธีที่ใช้ในการบอกความแตกต่างระหว่างโลคัลออนโทโลจี ซึ่งงานวิจัยนี้เลือกใช้วิธีการเปรียบเทียบบริบทของข้อมูลที่แชร์ดออนโทโลจีและจะเปรียบเทียบเพื่อแปลความหมายทุกครั้งที่ได้รับข้อคำถาม ในขณะที่งานของผู้วิจัยเลือก

เปรียบเทียบเพียงครั้งเดียว และนำผลที่ได้มาสร้างเป็นออนโทโลจีรวมเพื่อเก็บความแตกต่างไว้ล่วงหน้า เพื่อตัดขั้นตอนการเปรียบเทียบที่อาจใช้เวลานานออกไป

2.2.3 งานวิจัยเกี่ยวกับการรวมออนโทโลจี

งานวิจัยเกี่ยวกับการรวมออนโทโลจีได้รับความสนใจอย่างแพร่หลายในปัจจุบันตามความนิยมในการนำออนโทโลจีมาใช้แสดงแทนข้อมูลเกี่ยวกับโดเมนต่างๆ ที่สนใจ และความต้องการใช้ข้อมูลที่ได้รับการแสดงแทนด้วยออนโทโลจีร่วมกัน โดยการรวมออนโทโลจี [11] เป็นกระบวนการในการสร้างออนโทโลจีหนึ่งชุดเพื่อแสดงแทนเรื่องใดเรื่องหนึ่งขึ้นจากออนโทโลจีตั้งแต่หนึ่งชุดขึ้นไปที่แสดงแทนข้อมูลในต่างเรื่องกัน ทั้งนี้การรวมออนโทโลจีอาจประกอบด้วย [12] การแมปออนโทโลจี (Ontology Mapping) ซึ่งเป็นขั้นตอนการเชื่อมโยงคำศัพท์จากต่างออนโทโลจีที่ใช้งานในโดเมนเดียวกัน และการผสานออนโทโลจี (Ontology Merging) ซึ่งเป็นขั้นตอนการรวมหลายออนโทโลจีที่แสดงแทนข้อมูลเรื่องเดียวกันเข้าด้วยกันเป็นออนโทโลจีเดียว

ในงานวิจัย [13] ได้แบ่งแนวทางในการรวมออนโทโลจีออกเป็น 2 รูปแบบคือ แนวคิดแบบใช้การสอบถามข้ามออนโทโลจี เช่น งานวิจัย OBSERVER [9] ที่ได้กล่าวไปแล้วในหัวข้อที่ 2.2.2.2 กับแนวคิดแบบที่มีการสร้างออนโทโลจีรวมขึ้นจริง เช่น งานวิจัย PROMPT [16], Chimaera [17] และ FCA-MERGE [18] เนื่องจากผู้วิจัยเลือกใช้การรวมออนโทโลจีตามแนวคิดแบบที่มีการสร้างออนโทโลจีรวมขึ้นจริง ภายในหัวข้อนี้จึงนำเสนอเฉพาะตัวอย่างงานวิจัยที่แสดงการนำแนวคิดนี้ไปใช้เท่านั้น ได้แก่

2.2.3.1 PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment [16]

งานวิจัย PROMPT เป็นตัวอย่างหนึ่งของงานวิจัยเรื่องการรวมออนโทโลจีแบบสร้างเป็นออนโทโลจีรวม โดยมีจุดประสงค์ในการรวม (Merging) ออนโทโลจีของสารสนเทศทั้งหมดที่ต้องการให้เป็นออนโทโลจีเดียวหรือปรับแนว (Alignment) ให้ออนโทโลจีทั้งหมดอธิบายข้อมูลไปในทางเดียวกันหรือต่อเนื่องกันด้วยแนวทางแบบกึ่งอัตโนมัติ นั่นคืองานวิจัย PROMPT จะทำงานบางส่วนเองโดยอัตโนมัติ และให้คำแนะนำแก่ผู้รวมออนโทโลจีเพื่อจัดการงานส่วนที่เหลือซึ่งต้องการการตัดสินใจจากผู้รวม

การทำงานของ PROMPT เริ่มจากให้ผู้รวมออนโทโลจีป้อนออนโทโลจีคู่ที่ต้องการรวมเข้าสู่โปรแกรมเพื่อให้โปรแกรมสร้างรายการเริ่มต้นโดยการจับคู่จากชื่อคลาสที่ตรงกัน จากนั้นโปรแกรมจะให้ผู้รวมออนโทโลจีเลือกใช้การดำเนินการอย่างใดอย่างหนึ่งเพื่อรวมหรือปรับออนโทโลจีทั้งคู่เข้าด้วยกัน แล้วโปรแกรมจะดำเนินการตามที่ผู้ใช้เลือก หลังจากนั้นโปรแกรมจะตรวจสอบความขัดแย้ง (Conflict) ที่เกิดขึ้นจากการดำเนินการ และสร้างคำแนะนำใหม่เพิ่มให้กับผู้รวมออนโทโลจีโดยพิจารณาจากการดำเนินการล่าสุด เมื่อได้รับคำแนะนำจากโปรแกรม ผู้รวมออนโทโลจีจะวนกลับไปเลือกใช้การดำเนินการอย่างหนึ่งจากรายการคำแนะนำอีกครั้ง และโปรแกรมก็จะดำเนินการเช่นเดิม

ซ้ำเป็นรอบๆ จนกว่าผู้รวมออนโทโลจีจะเลือกหยุดการดำเนินการ ผลลัพธ์ที่ได้จากโปรแกรม PROMPT ก็คือคำแนะนำว่าผู้รวมออนโทโลจีควรสร้างออนโทโลจีรวมอย่างไร

ตัวอย่างการดำเนินการที่ให้ผู้รวมออนโทโลจีเลือกใช้ได้แก่ การรวมคลาส การรวมสลอต (Slot) หรือความสัมพันธ์ระหว่างคลาส การรวมแบบผูกมัด (Binding) ระหว่างสลอตและคลาส คัดลอกคลาสทั้งคลาสจากออนโทโลจีหนึ่งไปยังอีกออนโทโลจี คัดลอกบางส่วนของคลาส เป็นต้น ส่วนตัวอย่างความขัดแย้งที่โปรแกรมตรวจสอบได้แก่ ชื่อขัดแย้ง การอ้างอิงไปยังจุดที่ไม่มีจริง ความซ้ำซ้อนในลำดับชั้นของคลาส เป็นต้น

การทำงานของ PROMPT เป็นการทำงานเชิงโต้ตอบ (Interactive) ระหว่างโปรแกรมกับผู้รวมออนโทโลจี ถือเป็นแนวทางหนึ่งในการรวมออนโทโลจีขนาดใหญ่ อย่างไรก็ตามการทำงานของ PROMPT ต้องใช้ผู้รวมออนโทโลจีซึ่งมีความเชี่ยวชาญเรื่องออนโทโลจีที่จะรวมเป็นอย่างดี จึงจะตัดสินใจได้ว่าควรเลือกดำเนินการอะไรบ้างกับการรวม ถือเป็นข้อแตกต่างสำคัญจากงานของผู้วิจัยที่ต้องการให้การทำงานเป็นไปอย่างอัตโนมัติมากที่สุดเพื่อลดงานของผู้รวมออนโทโลจีลง

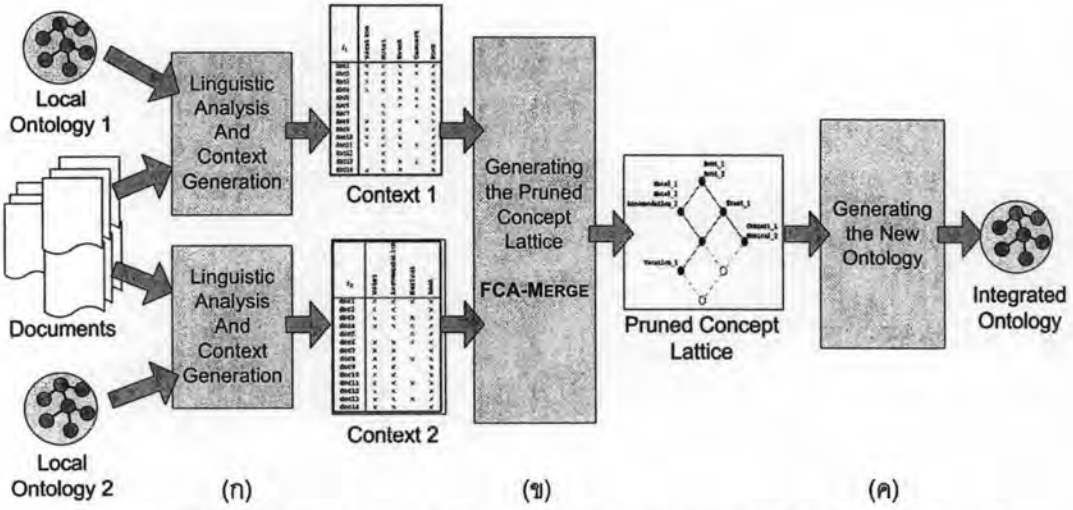
2.2.3.2 FCA-MERGE: Bottom-Up Merging of Ontologies [18]

งานวิจัย FCA-MERGE เป็นตัวอย่างหนึ่งของงานวิจัยเรื่องการรวมออนโทโลจีแบบสร้างเป็นออนโทโลจีรวม โดยขั้นตอนวิธีที่ใช้ในงานวิจัยนี้นำกลวิธีจากการประมวลผลภาษาธรรมชาติ (Natural Language Processing) และการวิเคราะห์แนวคิดเชิงรูปนัย (Formal Concept Analysis) มาใช้เปรียบเทียบออนโทโลจีที่ต้องการรวม แล้วแสดงข้อมูลทั้งหมดในรูปของโครงร่างของแนวคิด (Concept Lattice) เพื่อนำมาสร้างเป็นออนโทโลจีรวมต่อไป

ขั้นตอนการสร้างออนโทโลจีรวมในงานวิจัยนี้แบ่งการทำงานออกเป็น 3 ขั้นตอนแสดงไว้ในรูปที่ 2.11 ดังนี้

1. ขั้นตอนการวิเคราะห์ทางภาษาและการสร้างบริบท (รูปที่ 2.11(ก)) โดยนำเอกสาร (Document) ที่มีเนื้อหาเกี่ยวข้องกับออนโทโลจีทั้งสองชุดจำนวนหนึ่งกับออนโทโลจีแต่ละชุดมาเข้ากระบวนการทางภาษา เพื่อสร้างบริบท (Context) ขึ้นมาใช้กำกับออนโทโลจีแต่ละชุด เนื่องจากเอกสารที่นำมาใช้กับออนโทโลจีทั้งสองชุดเป็นเอกสารเดียวกัน ดังนั้นบริบทที่กำกับให้กับออนโทโลจีทั้งสองชุดจะมีความสัมพันธ์กัน
2. ขั้นตอนการสร้างโครงร่างของแนวคิด (รูปที่ 2.11(ข)) ถือเป็นขั้นตอนสำคัญในงานวิจัยนี้ เป็นการนำบริบทของออนโทโลจีที่ได้จากขั้นตอนแรก ซึ่งอธิบายเรื่องเดียวกันด้วยคำศัพท์จากสองออนโทโลจีมาสร้างขึ้นเป็นโครงร่างของแนวคิด เพื่อแสดงความสัมพันธ์ระหว่างออนโทโลจีทั้งสอง
3. ขั้นตอนการสร้างออนโทโลจีใหม่จากโครงร่างของแนวคิด (รูปที่ 2.11(ค)) โดยพิจารณาจากความสัมพันธ์ที่ปรากฏในโครงร่างดังกล่าว

ในการใช้งานจริง วิธีการที่ใช้ในงานวิจัย FCA-MERGE สามารถทำงานได้แบบอัตโนมัติ ผู้รวมออนโทโลจีมีหน้าที่กำหนดออนโทโลจีที่ต้องการรวม และเอกสารที่ใช้ในการสร้างบริบทให้โปรแกรมเท่านั้น แต่ความยุ่งยากของงานวิจัยนี้คือผู้รวมออนโทโลจีจำเป็นต้องเลือกใช้เอกสารที่เหมาะสมและเพียงพอต่อการกำหนดบริบทให้กับทั้งสองออนโทโลจี ออนโทโลจีรวมที่ได้จึงจะมีความถูกต้อง



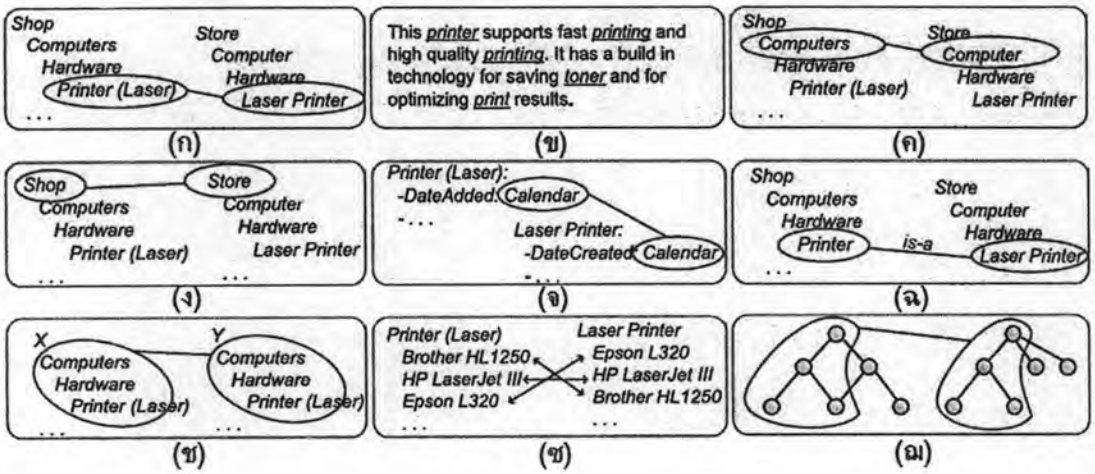
รูปที่ 2.11 ขั้นตอนการทำงานเพื่อสร้างออนโทโลจีรวมของงานวิจัย FCA- MERGE [18]

2.2.3.3 Identifying ontology integration methods and their applicability in the context of product classification and knowledge integration task [15]

งานวิจัยนี้ [15] ทำการสำรวจถึงวิธีการรวมออนโทโลจีจากงานวิจัยเกี่ยวกับแนวคิดในการรวมออนโทโลจีที่มีอยู่ในปัจจุบันกว่า 50 งานวิจัย และให้ข้อสรุปว่าสามารถจำแนกวิธีการพิจารณาในการรวมออนโทโลจีที่นิยมใช้งานภายในงานวิจัยต่างๆ ออกเป็น 9 วิธี (ตามรูปที่ 2.12) ดังต่อไปนี้

1. ความคล้ายคลึงของข้อความ (Text Similarity) เป็นวิธีที่ตรวจสอบจากความคล้ายคลึงของข้อความระหว่างสองออนโทโลจี โดยพิจารณาว่าชื่อหรือคำอธิบายในออนโทโลจีเหมือนกันหรือไม่ (รูปที่ 2.12(ก))
2. การสกัดคำหลัก (Keyword Extraction) เป็นการตรวจหาคำหลักจากคำอธิบายของคลาสเพื่อหาความหมายของคลาสและความสัมพันธ์กับคลาสอื่น (รูปที่ 2.12(ข))
3. วิธีการทางภาษา (Language-Based Methods) เช่น การพิจารณาจากข้อความที่อยู่ติดกัน การตัดคำทิ้งคำไม่สำคัญ (รูปที่ 2.12(ค))
4. การระบุความสัมพันธ์ระหว่างคำ (Identification of Word Relations) เป็นการนำทรัพยากรทางภาษาศาสตร์ เช่น คลังศัพท์ (Lexicon) มาตรวจหาคำพ้อง (รูปที่ 2.12(ง))
5. ความคล้ายคลึงของชนิดข้อมูล/ โดเมนและเรนจ์ (Type Similarity/ Domains and Ranges) เป็นการตรวจสอบว่าแอททริบิวต์ใดมีชนิดข้อมูลเหมือนกันบ้าง (รูปที่ 2.12(จ))

6. การวิเคราะห์การสืบทอดคลาส (Class Inheritance Analysis) เป็นการพิจารณาการสืบทอดระหว่างคลาส เพื่อหาความสัมพันธ์ชนิดอิสอะ (Is-A) (รูปที่ 2.12(ฉ))
7. การวิเคราะห์โครงสร้าง/ โครงสร้างของการแบ่งประเภท (Structure Analysis/ Taxonomic Structure) เป็นการระบุลักษณะร่วมกันของคลาสโดยพิจารณาจากแอททริบิวต์ของคลาสและการเชื่อมระหว่างคลาส (รูปที่ 2.12(ช))
8. การตีความข้อมูล/ คุณสมบัติหลัก (Data Interpretation/ Key Properties) ถ้าออนโทโลจีที่ต้องการรวมแสดงข้อมูลของอินสแตนซ์รวมอยู่ด้วย สามารถพิจารณาความคล้ายคลึงของคลาสได้หากอินสแตนซ์ของของสองคลาสเป็นตัวเดียวกัน (รูปที่ 2.12(ซ))
9. แมปปิงของกราฟ (Graph-Mapping) เป็นการใช้ลักษณะของส่วนที่เหมือนกันในกราฟของออนโทโลจีเพื่อระบุว่าโครงสร้างของออนโทโลจีสองชุดคล้ายคลึงกัน (รูปที่ 2.12(ฌ))



รูปที่ 2.12 วิธีการรวมออนโทโลจีที่ใช้งานในแนวคิดเกี่ยวกับการรวมออนโทโลจี [15]

สำหรับขั้นตอนวิธีในการรวมออนโทโลจีที่ผู้วิจัยพัฒนาขึ้นนั้น เมื่อนำมาวิเคราะห์ตามวิธีการรวมออนโทโลจีที่งานวิจัยนี้ได้จำแนกไว้ ผู้วิจัยพบว่าส่วนการเปรียบเทียบออนโทโลจีภายในงานของผู้วิจัยประกอบด้วยการพิจารณาโดยใช้วิธีตรวจสอบความคล้ายคลึงของข้อความ การระบุความสัมพันธ์ระหว่างคำ ความคล้ายคลึงของชนิดข้อมูล และการวิเคราะห์โครงสร้าง