บทที่ 4

ต้นแบบส่วนขยายยูดีดีไอแบบล่วงรู้บริบท

บทนี้จะกล่าวถึงการพัฒนาส่วนขยายยูดีดีไอแบบล่วงรู้บริบทหรือซีทีเอ็กซ์ยูดีดีไอ รวมถึงการใช้คุณลักษณะบริบทที่เพิ่มเติมมาเป็นเงื่อนไขในการค้นหา ตามแนวทางการออกแบบสถาปัตยกรรมการค้นหาบริการในยูดีดีไอแบบล่วงรู้บริบทดังที่กล่าวมาในบทที่ 3 ต้นแบบส่วนขยายยูดีดีไอแบบล่วงรู้บริบทในวิทยานิพนธ์นี้ได้รับการพัฒนาขึ้นภายใต้ข้อกำหนดของสถาปัตยกรรมยูดีดีไอรุ่น 2.0 โดยขยายจากยูดีดีไอแอพพลิเคชันของบริษัทอาปาเช่ (Apache) ที่ชื่อ "เจยูดีดีไอ" (jUDDI) [21] รุ่น 0.9 ซึ่งเป็นซอฟต์แวร์แบบโอเพนซอร์ซ (Open Source Software) ที่ถูกพัฒนาโดยภาษาจาวาทำให้สามารถทำงานบนระบบปฏิบัติการใด ๆ อีกทั้งสามารถใช้ได้กับฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ทุกตัวที่สนับสนุนมาตรฐานเอเอ็นเอสไอ เอสคิวเอล (ANSI SQL) อีกทั้งยังสามารถประกาศเจยูดีดีไอได้บนเซิร์ฟเวอร์ฝั่งจาวา การเข้าถึงบริการของเจยูดีดีไอแอพพลิเคชันจะทำโดยการเรียกใช้งานผ่านไลบรารียูดีดีไอโฟร์เจ (UDDI4J) [22] รุ่น 2.0.4 ผ่านทางโพรโทคอลโซพ ลำดับชั้นการทำงานแสดงในรูปที่ 4.1

รูปที่ 4.1 ลำดับชั้นการทำงานของซีทีเอ็กซ์ยูดีดีไอ

ในบทนี้จะแบ่งหัวข้อหลักออกเป็น 5 ส่วนคือ ส่วนเอพีไอของเจยูดีดีไอซึ่งสนับสนุนการลงทะเบียนตามมาตรฐาน ส่วนเอพีไอของซีทีเอ็กซ์ยูดีดีไอซึ่งสนับสนุนการลงทะเบียนแบบล่วงรู้บริบท ส่วนฐานข้อมูลซีทีเอ็กซ์ยูดีดีไอที่เพิ่มเติมจากฐานข้อมูลมาตรฐานยูดีดีไอ การร้องขอข้อมูลบริบทแบบพลวัตของผู้ให้บริการ และการตีความข้อมูลบริบท

4.1 เอพีไอในการลงทะเบียนของเจยูดีดีไอ (jUDDI Registry APIs)

เอพีไอในการลงทะเบียนของเจยูดีดีไอสำหรับให้ผู้พัฒนาสามารถเรียกใช้ในการลงทะเบียนประกอบด้วยฟังก์ชันการทำงานดังนี้

4.1.1 เจยูดีดีไอเอพีไอสำหรับการประกาศ (jUDDI Publish API)

เอพีไอการประกาศเพื่อใช้สำหรับการเก็บและแก้ไขข้อมูลในการลงทะเบียนให้บริการซึ่งประกอบด้วยฟังก์ชันการทำงานดังนี้

- save_business บันทึกข้อมูลส่วนบิสิเนสเอนทิตีลงฐานข้อมูลยูดีดีไอ
- save_service บันทึกข้อมูลส่วนบิสิเนสเซอร์วิสลงฐานข้อมูลยูดีดีไอ
- save_binding บันทึกข้อมูลส่วนไบน์ดิงลงฐานข้อมูลยูดีดีไอ
- save_tModel บันทึกข้อมูลส่วนทีโมเดลลงฐานข้อมูลยูดีดีไอ
- add_publisherAssertions เพิ่มข้อมูลส่วนคีย์อ้างอิงที่บอกความสัมพันธ์ระหว่างสองบิสิเนสเอนทิตี เช่น เป็นหน่วยงานที่ขึ้นต่อกัน (Parent-Child) หรือพันธมิตรกัน (Peer2Peer) ฯลฯ
- delete_business ลบข้อมูลส่วนบิสิเนสเอนทิตีจากฐานข้อมูลยูดีดีไอ
- delete_service ลบข้อมูลส่วนบิสิเนสเซอร์วิสจากฐานข้อมูลยูดีดีไอ
- delete_binding ลบข้อมูลส่วนไบน์ดิงจากฐานข้อมูลยูดีดีไอ
- delete_publisherAssertions ลบข้อมูลส่วนคีย์อ้างอิงที่บอกความสัมพันธ์ระหว่างสองบิสิเนสเอนทิตีจากฐานข้อมูลยูดีดีไอ
- delete_tModel ลบข้อมูลส่วนทีโมเดลจากฐานข้อมูลยูดีดีไอ
- get_authToken ขอโทเคน (Token) ในการระบุสิทธิ์เข้าใช้งานยูดีดีไอ
- discard_authToken ยกเลิกโทเคนในการระบุสิทธิ์เข้าใช้งานยูดีดีไอ
- get_registeredInfo เรียกดูรายการข้อมูลทั้งหมดในส่วนบิสิเนสเอนทิตีและทีโมเดลของผู้ให้บริการรายนั้น ๆ
- get_assertionStatusReport เรียกดูข้อมูลแสดงสถานะปัจจุบันของการอ้างอิงถึงผู้ให้บริการ (publisherAssertions) ที่เกี่ยวข้องกับบิสิเนสเอนทิตีใด ๆ ที่มีการลงทะเบียนของผู้ลงทะเบียนรายนั้น ๆ

- get_publisherAssertions เรียกดูข้อมูลส่วนคีย์อ้างอิงที่บอกความสัมพันธ์ระหว่างสองบิสิเนสเอนทิตี
- set_publisherAssertions ตั้งค่าข้อมูลส่วนคีย์อ้างอิงที่บิสิเนสเอนทิตีมีความสัมพันธ์เกี่ยวข้องทั้งหมด

4.1.2 **เจยูดีดีไอเอพีไอสำหรับการสอบถาม** (jUDDI Inquiry API)

เอพีไอการสอบถามเพื่อใช้สำหรับสอบถามและอ่านข้อมูลที่ได้มีการลงทะเบียนซึ่งประกอบด้วยฟังก์ชันการทำงานดังนี้

- find_business ค้นหาข้อมูลส่วนบิสิเนสเอนทิตีจากฐานข้อมูลยูดีดีไอ
- find_service ค้นหาข้อมูลส่วนบิสิเนสเซอร์วิสจากฐานข้อมูลยูดีดีไอ
- find_binding ค้นหาข้อมูลส่วนไบน์ดิงจากฐานข้อมูลยูดีดีไอ
- find_tModel ค้นหาข้อมูลส่วนทีโมเดลจากฐานข้อมูลยูดีดีไอ
- find_relatedBusinesses ค้นหาบิสิเนสเอนทิตีที่มีความเกี่ยวข้องสัมพันธ์กัน
- get_businessDetail เรียกดูข้อมูลส่วนบิสิเนสเอนทิตีจากการระบุบิสิเนสคีย์
- get_serviceDetail เรียกดูข้อมูลส่วนบิสิเนสเซอร์วิสจากการระบุเซอร์วิสคีย์
- get_bindingDetail เรียกดูข้อมูลส่วนไบน์ดิงจากการระบุไบน์ดิงคีย์
- get_tModelDetail เรียกดูข้อมูลส่วนบิสิเนสเอนทิตีจากการระบุทีโมเดลคีย์
- get_businessDetailExt เรียกดูข้อมูลส่วนบิสิเนสเอนทิตีที่มีเพิ่มคุณลักษณะพิเศษจากการระบุบิสิเนสคีย์

4.2 **เอพีไอในการลงทะเบียนของซีทีเอ็กซ์ยูดีดีไอ** (Ctx-UDDI Registry APIs)

ซีทีเอ็กซ์ยูดีดีไอได้เพิ่มเติมฟังก์ชันอื่น ๆ เพื่อพัฒนาให้ยูดีดีไอมาตรฐานสามารถรองรับการทำงานแบบล่วงรู้บริบทได้ โดยมีฟังก์ชันเพิ่มขึ้นดังนี้

4.2.1 **ซีทีเอ็กซ์เอพีไอสำหรับการประกาศ** (Ctx-UDDI Publish API)

เอพีไอการประกาศเพื่อใช้สำหรับการเก็บและแก้ไขข้อมูลในการลงทะเบียนบริการซึ่งยูดีดีไอมาตรฐานไม่รองรับ ประกอบด้วยฟังก์ชันการทำงานดังนี้

- save_contextType บันทึกข้อมูลประเภทบริบทที่มีการประกาศลงฐานข้อมูลซีทีเอ็กซ์ยูดีดีไอ
- delete_contextType ลบข้อมูลประเภทบริบทที่มีการประกาศออกจากฐานข้อมูลซีทีเอ็กซ์ยูดีดีไอ
- save_publisherContext_businessEntity บันทึกข้อมูลบริบทของผู้ให้บริการในส่วนบิสิเนสเอนทิตีลงฐานข้อมูลซีทีเอ็กซ์ยูดีดีไอ
- delete_publisherContext_businessEntity ลบข้อมูลบริบทของผู้ให้บริการในส่วนบิสิเนสเอนทิตีออกจากฐานข้อมูลซีทีเอ็กซ์ยูดีดีไอ
- save_publisherContext_businessService บันทึกข้อมูลบริบทของผู้ให้บริการในส่วนบิสิเนสเซอร์วิสลงฐานข้อมูลซีทีเอ็กซ์ยูดีดีไอ
- delete_publisherContext_businessService ลบข้อมูลบริบทของผู้ให้บริการในส่วนบิสิเนสเซอร์วิสออกจากฐานข้อมูลซีทีเอ็กซ์ยูดีดีไอ

4.2.2 **ซีทีเอ็กซ์เอพีไอสำหรับการสอบถาม** (Ctx-UDDI Inquiry API)

เอพีไอการสอบถามเพื่อใช้สำหรับสอบถามและอ่านข้อมูลที่ได้มีการลงทะเบียนไว้ในส่วนที่ยูดีดีไอมาตรฐานไม่รองรับ ประกอบด้วยฟังก์ชันการทำงานดังนี้

- find_ctxBusiness ค้นหาบิสิเนสแบบล่วงรู้บริบท
- find_ctxService ค้นหาเซอร์วิสแบบล่วงรู้บริบท
- get_providerDynamicCtx เรียกดูข้อมูลบริบทแบบพลวัตของผู้ให้บริการ
- interpret_context ตีความหมายบริบทโดยใช้ออนโทโลจี

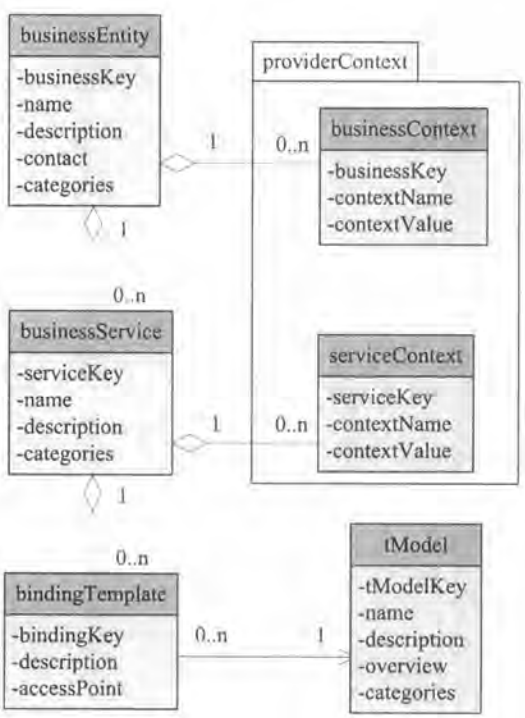
4.3 ฐานข้อมูลซีทีเอ็กซ์ยูดีดีไอ (Ctx-UDDI Database)

การพัฒนาฐานข้อมูลเพื่อสนับสนุนส่วนขยายของยูดีดีไอแบบล่วงรู้บริบทนั้น ไม่ได้ทำการแก้ไขในส่วนของฐานข้อมูลยูดีดีไอรีจิสตรีเดิม แต่ได้ทำการเพิ่มตารางที่ใช้สำหรับเก็บข้อมูลส่วนขยายยูดีดีไอ โดยซีทีเอ็กซ์ยูดีดีไอได้ออกแบบการเก็บข้อมูลบริบทออกเป็น 2 ส่วนคือ

1) **ข้อมูลบริบทของผู้ให้บริการในส่วนบิสิเนสเอนทิตี** จะมีการเชื่อมโยงไปยังข้อมูลบิสิเนสเอนทิตีผ่านบิสิเนสคีย์

2) **ข้อมูลบริบทของผู้ให้บริการในส่วนบิสิเนสเซอร์วิส** จะมีการเชื่อมโยงไปยังข้อมูลบิสิเนสเซอร์วิสผ่านเซอร์วิสคีย์

จากการออกแบบดังกล่าวสามารถแสดงการออกแบบฐานข้อมูลให้มีความสัมพันธ์กับข้อมูลบริบทที่ได้กำหนดไว้ดังรูปที่ 4.2



รูปที่ 4.2 แผนภาพความสัมพันธ์ของข้อมูลบริบทในฐานข้อมูล

4.4 การร้องขอข้อมูลบริบทแบบพลวัตของผู้ให้บริการ (Dynamic Context Invocation)

ข้อมูลบริบทแบบพลวัตของผู้ให้บริการจะไม่ได้ถูกเก็บอยู่ในฐานข้อมูลของซีทีเอ็กซ์ยูดีดีไอดังเช่นข้อมูลบริบทแบบสถิต แต่จะมีการเก็บค่าเอ็นด์พอยต์ (Endpoint) เช่น http://localhost:8080/axis/PrintServiceOne/PrintServiceOneLoad.jws ลงในฐานข้อมูลของซีทีเอ็กซ์ยูดีดีไอ

แทน เพื่อให้ซีทีเอ็กซ์ยูดีดีไอสามารถใช้ในการร้องขอค่าบริบทแบบพลวัตเมื่อต้องการได้ ผู้ให้บริการสามารถประกาศการล่วงรู้บริบทโดยค่าบริบทเป็นแบบพลวัตได้มากตราบเท่าที่ต้องการ แต่ผู้ให้บริการต้องระบุเอ็นด์พอยต์สำหรับค่าของแต่ละประเภทบริบทให้แตกต่างกัน โดยที่เอ็นพอยต์แต่ละตำแหน่งจะระบุไปยังเมท็อดเพื่อเรียกดูค่าบริบทแบบพลวัตของผู้ให้บริการ เมท็อดที่มีการประกาศไว้จะมีชนิดข้อมูลคืนกลับที่แตกต่างกันไปตามประเภทของบริบทที่ได้กำหนดไว้ล่วงหน้า เช่น ถ้าประเภทบริบทแบนด์วิดท์กำหนดชนิดข้อมูลไว้เป็นจำนวนเต็ม ผู้ให้บริการก็ต้องประกาศเมท็อดโดยข้อมูลคืนกลับเป็นข้อมูลแบบจำนวนเต็ม ถ้าประเภทบริบทเมืองถูกกำหนดชนิดข้อมูลไว้เป็นสตริง ผู้ให้บริการก็ต้องประกาศเมท็อดโดยข้อมูลคืนกลับเป็นข้อมูลแบบสตริง เป็นต้น ซิกเนเชอร์ของเมท็อดที่ได้กำหนดไว้เบื้องต้นมีดังนี้

- public String getDynamicContext(){...}
- public int getDynamicContext(){...}
- public double getDynamicContext(){...}
- public boolean getDynamicContext(){...}

เมื่อทุก ๆ เว็บเซอร์วิสกำหนดเมท็อดที่ใช้ในการร้องขอข้อมูลเหมือนกันหมด ซีทีเอ็กซ์ยูดีดีไอจะสามารถใช้ข้อความโซพในรูปแบบเดียวกันหมดเพื่อร้องขอข้อมูลบริบทแบบพลวัตไปยังเว็บเซอร์วิสนั้น โดยทำเพียงแค่เปลี่ยนยูอาร์เอลที่เป็นเอ็นด์พอยต์ในการร้องขอค่าเท่านั้น ในงานวิจัยนี้ได้ใช้ไลบรารีที่ช่วยในการรับและส่งข้อความโซพที่ชื่อ เอสเอเอเจ (SAAJ) ซึ่งเขียนโดยภาษาจาวา ข้อความโซพที่ส่งไปร้องขอข้อมูลบริบทแบบพลวัตเป็นดังรูปที่ 4.3

```
<soapenv:Envelope
 xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

 <soapenv:Body>

  <ns1:getProviderDynamicContext
   xmlns:ns1="http://localhost:8080/ctxUDDI"/>

 </soapenv:Body>

</soapenv:Envelope>
```

รูปที่ 4.3 ข้อความโซพที่ส่งไปร้องขอข้อมูลบริบทแบบพลวัต

ตัวอย่างข้อความโซพที่ได้หลังจากร้องขอข้อมูลบริบทปริมาณงานซึ่งเป็นบริบทแบบพลวัตของเว็บเซอร์วิสให้บริการพิมพ์เป็นดังรูปที่ 4.4

```
<?xml version="1.0" encoding="UTF-8"?>

 <soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <soapenv:Body>

   <ns1:getProviderDynamicContextResponse
   soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/enco
   ding/" xmlns:ns1="http://localhost:8080/ctxUDDI">

    <getProviderDynamicContextReturn
    xsi:type="xsd:int">9</getProviderDynamicContextReturn>

   </ns1:getProviderDynamicContextResponse>

  </soapenv:Body>

</soapenv:Envelope>
```

รูปที่ 4.4 ข้อความโซพตอบกลับจากการร้องขอข้อมูลบริบทแบบพลวัตของเว็บเซอร์วิสให้บริการพิมพ์

4.5 การตีความข้อมูลบริบท (Context Interpretation)

งานวิจัยนี้ได้ใช้จีนา (Jena) [23] รุ่น 2.4 เป็นเอพีไอที่ช่วยในการจัดการกับเอกสารออนโทโลจี ส่วนตีความข้อมูลบริบทของซีทีเอ็กซ์ยูดีดีไอจะรับค่าข้อมูลบริบทของผู้ให้บริการและผู้ใช้บริการมาเปรียบเทียบกันว่าเป็นคลาสแม่ลูกกัน คลาสสมมูลในระดับเดียวกัน หรือเป็นอินสแตนซ์ของคลาสในออนโทโลจีที่กำหนดไว้หรือไม่ ไม่ว่าจะพบความสัมพันธ์ของข้อมูลเป็นลักษณะใดก็จะถือว่าข้อมูลบริบททั้งสองนั้นสอดคล้องกันทั้งสิ้น เช่น ออนโทโลจีอาหารได้กำหนดให้พิซซ่าเป็นคลาสลูกของคลาสอาหารอิตาเลียน ดังแสดงในรูปที่ 4.5 เมื่อซีทีเอ็กซ์ทำงานมาจนถึงส่วนตีความบริบทก็จะส่งข้อมูลบริบทผู้ใช้บริการคือ "Pizza" ไปเปรียบเทียบกับข้อมูลผู้ให้บริการคือ "Italian Food" ระบบก็จะตีความได้ว่าเป็นบริบทที่สอดคล้องกัน เป็นต้น

รูปที่ 4.5 ส่วนหนึ่งของออนโทโลจีชนิดอาหาร

ซีทีเอ็กซ์ยูดีดีไอมีการเรียกใช้เอพีไอของจีนาเพื่อทำงานดังต่อไปนี้

➢ การอ่านและเขียนเอกสารออนโทโลจี ดังรูปที่ 4.6

```
OntModel model = ModelFactory.createOntologyModel();
InputStream in = FileManager.get().open(inputFileName );
if (in == null) {
    throw new IllegalArgumentException( "File: " +
    inputFileName + " not found");
}
// read the RDF/XML file
model.read(in, "");
// write it to standard out
model.write(System.out);
```

รูปที่ 4.6 การอ่านและเขียนเอกสารออนโทโลจี

การจัดการกับแบบจำลองเริ่มจากการใช้ ModelFactory สร้างแบบจำลองเปล่าขึ้นมา จากนั้นใช้เมท็อด read เพื่อการอ่านเอกสารออนโทโลจี และเมท็อด write เพื่อเขียนแบบจำลองออนโทโลจีออกมาในรูปแบบของตัวอักษร

➢ การเรียกดึงทรัพยากร (Resource) และทำงานกับออนโทโลจีคลาส ดังรูปที่ 4.7

```
Resource res = model.getResource(resourceURI);
OntClass c = (OntClass)res.canAs(OntClass.class);
```

รูปที่ 4.7 การเรียกดึงทรัพยากรและทำงานกับออนโทโลจีคลาส

resourceURI เป็นที่อยู่ของทรัพยากรของแบบจำลอง และถ้าทรัพยากรสามารถแปลงเป็นคลาสได้ ก็สามารถแปลงทรัพยากรได้โดยใช้ canAs ให้ได้ออกมาเป็นคลาสอีกต่อหนึ่ง

- การหาคลาสลูก คลาสแม่ คลาสสมมูลในระดับเดียวกัน และอินสแตนซ์จากแบบจำลองออนโทโลจีหนึ่ง ๆ ดังรูปที่ 4.8

```
private List listSuperClassOfInfModel(InfModel m, Resource
s,  Resource o, int level) {

. . .

}

private List listSubClassOfInfModel(InfModel m, Resource
o, int level) {

. . .

}

private List inferenceEquivalentClass(InfModel m, Resource
s,  Resource o, int level) {

. . .

}

private List inferenceInstance(InfModel m, Resource o, int
level) {

. . .

}
```

รูปที่ 4.8 การหาคลาสลูก คลาสแม่ คลาสสมมูลในระดับเดียวกัน และอินสแตนซ์จากแบบจำลองออนโทโลจี

จากคลาสที่ได้จากทรัพยากรจะสามารถหาคลาสลูกและคลาสแม่ โดยใช้เมท็อด listSubClassesOfInfModel และ listSuperClassesOfInfModel โดยที่สามารถกำหนดลำดับชั้นของการอนุมานได้ นอกจากนี้สามารถหาคลาสสมมูลในระดับเดียวกันหรืออินสแตนซ์ของคลาสได้โดยใช้ inferenceEquivalentClass และ inferenceInstance ตามลำดับ

จากการออกแบบในบทที่ 3 ซีทีเอ็กซ์ยูดีดีไอจะใช้เอกสารออนโทโลจีจากภายนอกซึ่งประกาศโดยผู้เชี่ยวชาญในโดเมนต่าง ๆ ทั้งที่มีประกาศไว้ตามเว็บไซต์ขององค์กรที่สนับสนุนเว็บเชิงความหมายหรือที่สามารถค้นหาได้จากสวูเกิล [24] ซึ่งเป็นโปรแกรมค้นหาเอกสารออนโทโลจีที่มีอยู่แล้วในปัจจุบัน แต่ก็ยังคงมีอุปสรรคในเรื่องการขาดมาตรฐานเป็นสิ่งสำคัญ กล่าวคือเอกสารที่ได้รับการเผยแพร่ออกสู่สาธารณะในปัจจุบันยังมิได้มีการกำหนดมาตรฐานไว้อย่างแน่ชัด ทำให้เอกสารส่วนใหญ่ที่มีอยู่เป็นเอกสารที่มีความถูกต้องในเรื่องของภาษาที่ใช้ในการเขียนเท่านั้น โดยยังไม่สามารถนำมาใช้งานได้อย่างสอดคล้องกับประเภทบริบทที่มีการประกาศไว้กับซีทีเอ็กซ์ยูดีดีไอ เพราะขาดความสามารถในการอธิบายทรัพยากรต่าง ๆ ได้อย่างเหมาะสมและลึกซึ้งเพียง

พอที่จะนำมาทำการวิเคราะห์ได้ทั้งหมด อีกทั้งเอกสารส่วนใหญ่ยังถูกเขียนขึ้นโดยผู้ที่เริ่มสนใจและเริ่มต้นพัฒนา จึงยังไม่สามารถใช้ความสามารถที่แท้จริงเพื่อนำมาตีความหมายบริบทของผู้ให้บริการและผู้ใช้บริการให้สอดคล้องกับเว็บเซอร์วิสที่นำมาเป็นตัวอย่างได้ ซึ่งถือว่าเป็นข้อจำกัดของงานวิจัยในขณะนี้ ดังนั้นผู้วิจัยจึงทำการออกแบบออนโทโลจีขึ้นเองเพื่อให้สอดคล้องกับการตีความของเว็บเซอร์วิสที่นำมาเป็นตัวอย่างสำหรับใช้ในการทดสอบการทำงาน อีกทั้งในส่วนของการค้นหาและเลือกใช้เอกสารออนโทโลจีในงานวิจัยนี้จะปรับใช้ในลักษณะของฐานข้อมูล ผู้วิจัยจะนำออนโทโลจีมาจัดเก็บลงสู่ฐานข้อมูลและจัดทำดัชนีด้วย โดยจะจัดเก็บแบบจำลองออนโทโลจีให้อยู่ในรูปแบบจำลองที่สามารถอธิบายความสัมพันธ์ทั้งหมดได้ และเก็บคำที่เป็นทรัพยากรของเอกสารออนโทโลจีไว้เป็นดัชนีควบคู่กับการเก็บแบบจำลองออนโทโลจีในฐานข้อมูล

4.5.1 **ฐานข้อมูลออนโทโลจี** (Ontology Database)

เป็นส่วนของระบบฐานข้อมูลที่ใช้เก็บข้อมูลเกี่ยวกับคำที่เป็นทรัพยากรของเอกสารออนโทโลจีทั้งหมดที่มีในระบบ รวมทั้งเก็บแบบจำลองออนโทโลจีทั้งหมดของระบบเพื่อไว้ใช้งานอีกด้วย ในการนำแบบจำลองออนโทโลจีเข้าไปเก็บไว้ในฐานข้อมูลจะทำโดยการรับเอกสารออนโทโลจีจากภายนอกที่เลือกสรรแล้วไปทำการวิเคราะห์ และสร้างเป็นแบบจำลองออนโทโลจีที่ใช้ในการเก็บความสัมพันธ์ต่าง ๆ โดยใช้จีน่าช่วยในการทำงานในส่วนนี้ แล้วเก็บแบบจำลองออนโทโลจีที่ได้ลงในฐานข้อมูลและจดจำข้อมูลตำแหน่งที่เป็นที่เก็บโมเดลไว้ (Path) เรียกชื่อเป็นโมเดลไอดี (ModelID) ควบคู่ไปกับการจัดดัชนีของคำ (Indexing) เพื่อใช้ในการเรียกแบบจำลองออนโทโลจีที่เกี่ยวข้องกับคำต่าง ๆ ออกมาใช้งาน

ซีทีเอ็กซ์ยูดีดีไอสามารถรับเอกสารออนโทโลจีเพื่อสร้างเป็นแบบจำลองออนโทโลจีที่เก็บความสัมพันธ์ระหว่างคำดังรูปที่ 4.9

```
String [] filename = {"C:\\\\Jena-
      2.4\\\\testing\\\\owl_test\\\\Software.owl"};
String [] filepath = {"C:\\Jena-
      2.4\\testing\\owl_test\\Software.owl"};
Indexing id = new Indexing();
id.setUser("root");
id.setPassword("root");
id.setUrl("jdbc:mysql://localhost/ontologydb");
id.setPersistOnt();
OntologyAnalysisLayer ontAnaLayer = new
      OntologyAnalysisLayer();
for(int i = 0 ;i<filename.length;i++){
      ontAnaLayer.setInputFileName(filepath[i]);
      ontAnaLayer.setModel(false);
      List l = ontAnaLayer.listAllResource();
      //Get ready to set up Table
      id.setFilePath(filename[i]);
      id.setFilePath2(filepath[i]);
      id.setResource(l);
      //Set up Table
      d.setUpTable();
}
```

รูปที่ 4.9 การรับเอกสารออนโทโลจีเพื่อสร้างเป็นแบบจำลองออนโทโลจีเก็บความสัมพันธ์ระหว่างคำ

สาเหตุที่ใช้การจัดเก็บแบบจำลองออนโทโลจีลงในฐานข้อมูล แทนที่จะเก็บเอกสารออนโทโลจีลงในฐานข้อมูลก็เพื่อความสะดวกรวดเร็วในการทำงานของระบบ นอกจากการดึงข้อมูลที่เป็นแบบจำลองจากฐานข้อมูลจะทำได้ง่ายแล้ว ยังช่วยในเรื่องความเร็วในการทำงานของส่วนวิเคราะห์ออนโทโลจีด้วย ทำให้ไม่ต้องทำการอนุมาน (Inference) ทุกครั้งที่ติดต่อกับฐานข้อมูล แต่

จะสามารถดึงแบบจำลองที่พร้อมใช้งานออกมาได้เลย ซึ่งจะช่วยเพิ่มประสิทธิภาพในการทำงานให้กับระบบได้เป็นอย่างมาก

โดยในการสร้างและอัพเดทฐานข้อมูลนี้จำเป็นต้องอาศัยการทำงานด้วยกำลังคน โดยผู้ให้บริการต้องทำการสร้างฐานข้อมูลเริ่มต้นเอง และทำการอัพเดทฐานข้อมูลทุกครั้งที่ต้องการนำแบบจำลองออนโทโลจีใหม่เข้าไปเก็บไว้ในฐานข้อมูล เมื่อได้รับเอกสารออนโทโลจีใหม่มาหรือเมื่อมีเอกสารออนโทโลจีใหม่ ๆ เผยแพร่ออกมาสู่สาธารณะ

4.5.2 **ส่วนการจัดทำดัชนีของคำและการเก็บแบบจำลองออนโทโลจี** (Indexing Module)

เป็นการทำงานควบคู่ไปกับส่วนฐานข้อมูลออนโทโลจี ทั้งในการจัดเก็บแบบจำลองออนโทโลจีลงสู่ฐานข้อมูลเพื่อไว้ใช้งานต่อไป และการดึงแบบจำลองออนโทโลจีออกมาจากฐานข้อมูลเพื่อใช้งานในขั้นตอนของการวิเคราะห์คำในส่วนของชั้นวิเคราะห์ออนโทโลจี โดยจะมีการจัดทำดัชนีในขั้นตอนของการจัดเก็บแบบจำลองออนโทโลจีลงสู่ฐานข้อมูลด้วย เพื่อใช้สำหรับการเข้าถึงข้อมูลในฐานข้อมูลต่อไป

สำหรับการทำงานในส่วนนี้ตามที่ได้กล่าวไว้นั้น จะสามารถแบ่งออกเป็น 2 ส่วนหลัก คือ

1) **การจัดเก็บแบบจำลองออนโทโลจีลงในฐานข้อมูล**

การจัดเก็บแบบจำลองออนโทโลจีลงในฐานข้อมูลจะมีการจัดทำดัชนีเพื่อจับคู่ (Mapping) ระหว่างคำที่ได้มาทั้งหมดจากแบบจำลองออนโทโลจีแต่ละตัวไว้กับโมเดลไอดี ซึ่งเป็นข้อมูลตำแหน่งที่เก็บตัวแบบจำลองออนโทโลจีที่เกี่ยวข้องกับคำเหล่านั้นไว้ เพื่อใช้ในการทำงานของชั้นวิเคราะห์ออนโทโลจีในการเลือกดึงแบบจำลองออนโทโลจีที่เกี่ยวข้องกับบริบทใด ๆ ออกมาได้อย่างถูกต้อง โดยมีการทำงานดังนี้

(1) จัดทำดัชนีโดยการจับคู่ทรัพยากรที่มีทั้งหมดของเอกสารออนโทโลจี ซึ่งก็คือคำศัพท์ทั้งหมดที่มีกับแบบจำลองออนโทโลจีเอง โดยเก็บเป็นข้อมูลตำแหน่งที่เก็บแบบจำลองออนโทโลจีนั้นไว้ในฐานข้อมูล

(2) นำแบบจำลองออนโทโลจีที่อนุมานได้จากเอกสารออนโทโลจีเก็บลงฐานข้อมูล

2) **การดึงแบบจำลองออนโทโลจีที่เก็บไว้ออกจากฐานข้อมูลออนโทโลจีเพื่อนำมาใช้งาน**

เริ่มจากการพิจารณาค่าบริบทของผู้ใช้บริการที่ส่งเข้ามา แล้วทำการค้นหาแบบจำลองออนโทโลจีที่เกี่ยวข้องกับค่าบริบทดังกล่าว ตามที่ได้เก็บข้อมูลคำทั้งหมดไว้แล้ว โดยการไล่หาจากดัชนีคำทุกคำที่จัดทำไว้ เพื่อหาคำที่ตรงกับค่าบริบทที่ส่งเข้ามา และทำการดึงแบบจำลองดังกล่าวขึ้นมาเพื่อส่งไปยังส่วนวิเคราะห์ออนโทโลจีต่อไป

4.5.3 **ชั้นวิเคราะห์ออนโทโลจี** (Ontology Analysis Module)

การทำงานของชั้นวิเคราะห์ออนโทโลจีจะเป็นการนำเอาค่าบริบทของผู้ใช้บริการมาทำการอนุมานเพื่อให้ได้ค่าบริบทที่มีความหมายเพิ่มเติมส่งกลับไปยังส่วนตีความบริบทของซีทีเอ็กซ์ยูดีดีไอ โดยมีรูปแบบการอนุมานหลัก ๆ 4 แบบคือ

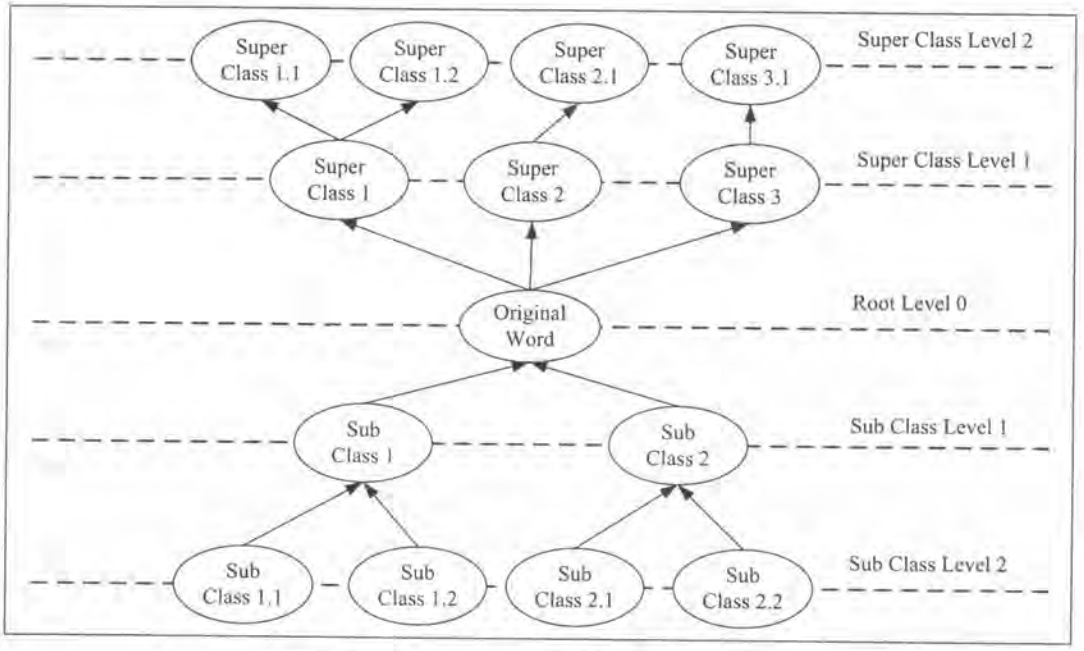
1) อนุมานคลาสลูก (Subclass)

2) อนุมานคลาสแม่ (Superclass)

3) อนุมานคลาสสมมูลในระดับเดียวกัน (Equivalent Class)

4) อนุมานอินสแตนซ์ (Instance)

การอนุมานจะทำเป็นลำดับขั้นต่อกันไปจากโหนดตั้งต้นดังแสดงในรูปที่ 4.10



รูปที่ 4.10 ตัวอย่างลำดับขั้นในการอนุมาน

จากรูปที่ 4.10 แสดงการอนุมาน 2 รูปแบบคืออนุมานคลาสแม่และอนุมานคลาสลูก ในเบื้องต้นงานวิจัยนี้จะทำการอนุมานคลาสแม่ขึ้นไป 2 ลำดับชั้นจากคำหลักตั้งต้นของผู้ใช้ และทำการอนุมานคลาสลูกลงมา 2 ลำดับชั้นจากคำหลักตั้งต้นของผู้ใช้ เมื่อได้คำที่ได้จากการอนุมานค่าบริบทของผู้ใช้บริการแล้วก็จะส่งไปเปรียบเทียบกับค่าบริบทของผู้ให้บริการต่อไป