

ความหมายของพฤติกรรมอิงออนไลน์จี้สำหรับดื่บเบ็ลยูเอสดีแอส



นาย ทิวากร โภมูท

สถาบันวิทยบริการ

จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

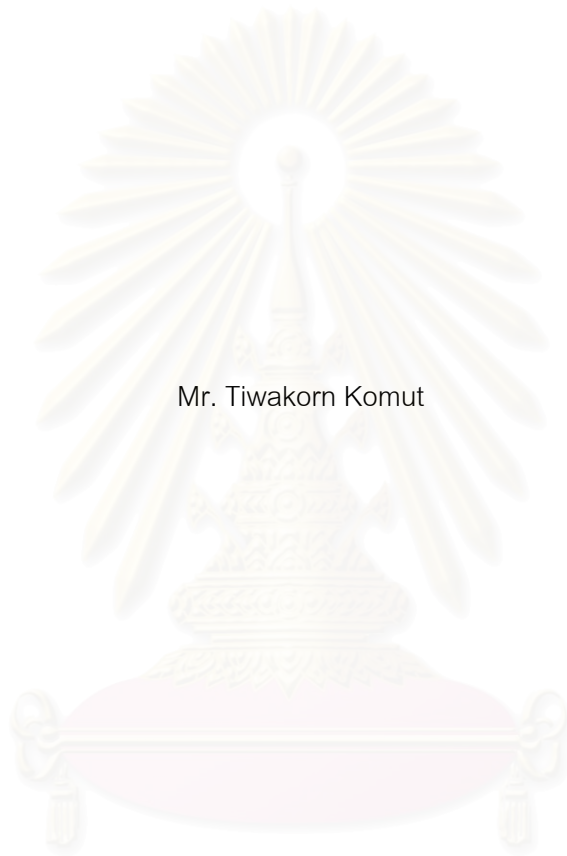
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2547

ISBN 974-17-6289-5

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

AN ONTOLOGY – BASED BEHAVIORAL SEMANTICS FOR WSDL



Mr. Tiwakorn Komut

สถาบันวิทยบริการ

A Thesis Submitted in Partial Fulfillment of the Requirements  
for the Degree of Master of Science in Computer Science

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2004

ISBN 974-17-6289-5

หัวข้อวิทยานิพนธ์      ความหมายของพฤติกรรมเชิงออนไลน์สำหรับดัดแปลงเบบิยูเอสดีแอล  
โดย                              นาย ทิวากร โกมุต  
สาขาวิชา                      วิทยาศาสตร์คอมพิวเตอร์  
อาจารย์ที่ปรึกษา              ผู้ช่วยศาสตราจารย์ ดร.ทวิติย์ เสนีวงศ์ ณ อยุธยา

---

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้รับวิทยานิพนธ์ฉบับนี้  
เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์  
(ศาสตราจารย์ ดร.ดิเรก ลาวัณย์ศิริ)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ  
(อาจารย์ ดร.ยรรยง เต็งอำนวย)

..... อาจารย์ที่ปรึกษา  
(ผู้ช่วยศาสตราจารย์ ดร.ทวิติย์ เสนีวงศ์ ณ อยุธยา)

..... กรรมการ  
(อาจารย์ จารุมาตร ปิ่นทอง)

..... กรรมการ  
(อาจารย์ ดร.วิษณุ โคตรจรัส)

สถาบันนวัตกรรมการ  
จุฬาลงกรณ์มหาวิทยาลัย

ทิวากร โกมุต : ความหมายของพฤติกรรมอิงออนโทโลยีสำหรับดับเบิลยูเอสดีแอล. (AN ONTOLOGY – BASED BEHAVIORAL SEMANTICS FOR WSDL) อ. ที่ปรึกษา: ผู้ช่วยศาสตราจารย์ ดร.ทวิติย์ เสนีวงศ์ ณ อยุธยา, 76 หน้า. ISBN 974-17-6289-5.

ในปัจจุบันการค้นหาเว็บเซอร์วิสจะกระทำผ่านบริการไคเรกทอรีเช่นยูดีดีไอ ผู้รับบริการสามารถค้นหาบริการได้โดยกำหนดคำสำคัญซึ่งจะถูกนำไปพิจารณาเปรียบเทียบกับแอททริบิวต์ที่อธิบายรายละเอียดของบริการ ได้แก่ บิสิเนสเอนทิตี บิสิเนสเซอร์วิส ไลน์ดิงเทมเพลต และ ที่โมเดล แต่ในกรณีที่ผู้รับบริการต้องการค้นหาเว็บเซอร์วิสจากพฤติกรรมของบริการ โครงสร้างคำอธิบายบริการในยูดีดีไอที่อยู่ในรูปแบบแอททริบิวต์กลับไม่เอื้อต่อการค้นหา ผู้รับบริการมีอีกทางเลือกหนึ่งโดยพิจารณาพฤติกรรมของบริการได้จากเอกสารดับเบิลยูเอสดีแอล แต่วิธีนี้ก็ไม้อาจทราบได้แน่ชัดว่าเป็นบริการที่ตรงกับความต้องการหรือไม่ ทั้งนี้เพราะความหมายของส่วนต่างๆในเอกสารดับเบิลยูเอสดีแอลที่ผู้ให้บริการกำหนดอาจแตกต่างจากที่ผู้รับบริการเข้าใจ

ในปัจจุบันเริ่มมีการนำออนโทโลยีซึ่งใช้ในการอธิบายความหมายของสิ่งต่างๆมาใช้อธิบายความหมายให้กับเว็บเพจ เพื่อให้สามารถค้นพบเว็บเพจที่ตรงกับความต้องการของผู้ใช้ได้ดีกว่าการค้นหาด้วยคำสำคัญ เนื่องด้วยเว็บเซอร์วิสก็เป็นทรัพยากรรูปแบบหนึ่งบนเว็บ จึงได้เริ่มมีงานวิจัยที่พยายามอธิบายเว็บเซอร์วิสด้วยออนโทโลยี

ในงานวิจัยนี้ผู้วิจัยมีแนวคิดที่จะออกแบบและพัฒนาเฟรมเวิร์คสำหรับอธิบายและค้นหาเว็บเซอร์วิส จากเอกสารดับเบิลยูเอสดีแอลที่ถูกแทรกข้อสนเทศเชิงพฤติกรรมซึ่งกำหนดไว้ด้วยออนโทโลยี เพื่อให้ผู้รับบริการสามารถค้นหาบริการซึ่งมีความหมายและพฤติกรรมตรงกับที่ต้องการได้โดยอัตโนมัติ โดยแบบจำลองพฤติกรรมที่ใช้อธิบายพฤติกรรมของเว็บเซอร์วิสจะอยู่ในรูปของ การดำเนินการ ข้อมูลเข้า ข้อมูลออก ผลกระทบ และเงื่อนไขต่างๆ ได้แก่ เงื่อนไขก่อนการทำงาน เงื่อนไขสำหรับข้อมูลออก และ เงื่อนไขสำหรับผลกระทบ

ภาควิชา.....วิศวกรรมคอมพิวเตอร์..... ลายมือชื่อนิสิต.....  
 สาขาวิชา.....วิทยาศาสตร์คอมพิวเตอร์..... ลายมือชื่ออาจารย์ที่ปรึกษา.....  
 ปีการศึกษา.....2547.....

# # 4570336021 : MAJOR COMPUTER SCIENCE

KEY WORD: WEB SERVICE / BEHAVIORAL INFORMATION / ONTOLOGY / WSDL

TIWAKORN KOMUT : AN ONTOLOGY – BASED BEHAVIORAL SEMANTICS FOR WSDL. THESIS ADVISOR : ASSISTANT PROFESSOR TWITTIE SENIVONGSE, Ph.D., 76 pp. ISBN 974-17-6289-5.

Today, a discovery of Web Services is achieved through a directory service such as UDDI. Service requesters can find services by matching keywords against service attributes (i.e. Business Entity, Business Service, Binding Template and tModel). In the case that a service requester wants to search for Web Services by service behavior, the attribute-based information model of UDDI cannot serve the purpose well. The service requester has an alternative to consider a service behavior description from WSDL document, but this cannot guarantee the result because the meaning of each component in the WSDL document that is defined by the service provider may be understood by the service requester in a different way.

Now there are approaches that use ontology to describe semantics for the content of webpages. Searching webpages can be performed efficiently than using keywords. Because Web Services are also resources on the Web, number of research has been developed to describe Web Services using ontology.

This thesis proposes a framework for advertising and discovering Web Services from ontology-based behavioral information that is annotated within WSDL documents. Thus a service requester can discover Web Services automatically by defining behavioral constraints. The behavioral model that is used to describe the behavior of Web Services is in terms of operation, input, output, effect, and conditions such as precondition, condition on output, and condition on effect.

Department.....Computer Engineering..... Student's signature signature.....

Field of study.....Computer Science..... Advisor's signature .....

Academic year.... 2004.....

## กิตติกรรมประกาศ

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.ทวีติย์ เสนีวงศ์ ณ อยุธยา สำหรับคำปรึกษาแนะนำอันเป็นประโยชน์ ทำให้วิทยานิพนธ์นี้สำเร็จลุล่วงไปด้วยดี อีกทั้ง อาจารย์ ดร.ยรรยง เต็งอำนาจ อาจารย์ จารูมาตร ปิ่นทอง และ อาจารย์ ดร.วิษณุ โคตรจรัส ที่ช่วยแนะแนวทางการทำวิจัยขณะสอบโครงร่างวิทยานิพนธ์

สุดท้ายนี้ขอกราบขอบพระคุณ พ่อ และ แม่ ที่ให้โอกาสและสนับสนุนด้านการศึกษาตลอดมา



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ญ
สารบัญภาพ.....	ฎ
บทที่	
1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการวิจัย .....	3
1.3 ขอบเขตของการวิจัย .....	4
1.4 ประโยชน์ที่คาดว่าจะได้รับ .....	4
1.5 วิธีดำเนินการวิจัย .....	4
1.6 โครงสร้างวิทยานิพนธ์.....	4
2 เอกสารและงานวิจัยที่เกี่ยวข้อง .....	6
2.1 ทฤษฎีที่เกี่ยวข้อง.....	6
2.1.1 ดับเบิลยูเอสดีแอล.....	6
2.1.1.1 คำอธิบายเชิงนามธรรม (Abstract Description) .....	7
2.1.1.2 คำอธิบายเชิงรูปธรรม (Concrete Description) .....	8
2.1.2 ออนโทโลยี.....	8
2.1.3 อาร์ดีเอฟ.....	9
2.1.4 อาร์ดีเอฟเอส .....	11
2.1.5 อาวล์ .....	13
2.1.6 โพรเทเจ .....	14
2.1.7 จีนา .....	15
2.1.8 อาร์ดีคิวแอล .....	15
2.1.9 ดับเบิลยูเอสดีแอลเอดิเตอร์.....	16

## สารบัญ (ต่อ)

	หน้า
2.2 งานวิจัยที่เกี่ยวข้อง .....	17
2.2.1 งานวิจัย “Adding Semantics to Web Services Standards” .....	17
2.2.2 งานวิจัย “Supercharging WSDL with RDF – Managing Structured Web Service Metadata” .....	18
2.2.3 งานวิจัย “Discovering Web Services Using Behavioral Constraints and Ontology” .....	19
3 การอธิบายเว็บเซอร์วิสด้วยข้อสนเทศเชิงพฤติกรรมของออนโทโลยี .....	21
3.1 การออกแบบอ็อบเจกต์ออนโทโลยีเพื่อใช้อธิบายพฤติกรรมของเว็บเซอร์วิส .....	21
3.1.1.1 คลาสในอ็อบเจกต์ออนโทโลยี .....	23
3.1.1.2 คุณสมบัติในอ็อบเจกต์ออนโทโลยี .....	24
3.1.1.3 นิพจน์ภาษาอวาล์ในอ็อบเจกต์ออนโทโลยี .....	27
3.1.1.4 การตัดสินค่าความจริงของเงื่อนไข .....	27
3.2 การแทรกข้อสนเทศเชิงพฤติกรรมให้กับเอกสารดับเบิลยูเอสดีแอล .....	31
3.2.1 แนวคิดในการแทรกข้อสนเทศเชิงพฤติกรรม .....	31
3.2.2 ข้อกำหนดในการแทรกข้อสนเทศเชิงพฤติกรรม .....	32
3.3 ตัวอย่างการแทรกข้อสนเทศเชิงพฤติกรรมให้กับเอกสารดับเบิลยูเอสดีแอล .....	34
4 เฟรมเวิร์คสำหรับการโฆษณาและการค้นหาบริการเว็บเซอร์วิส .....	49
4.1 การออกแบบเฟรมเวิร์คสำหรับการโฆษณาและการค้นหาบริการเว็บเซอร์วิส .....	49
4.1.1 การออกแบบส่วนต่อประสานสำหรับโฆษณาบริการ .....	50
4.1.2 การออกแบบแผนแบบการค้นหา .....	52
4.1.2.1 การจับคู่คลาสที่ต้องการกับอินสแตนซ์ของผู้ให้บริการ .....	56
4.1.2.2 การจับคู่กลุ่มของคลาสที่ต้องการกับกลุ่มของอินสแตนซ์ของผู้ให้บริการ .....	57
4.2 ต้นแบบโปรแกรมประยุกต์สำหรับการโฆษณาและการค้นหาบริการเว็บเซอร์วิส .....	61
4.2.1 ต้นแบบโปรแกรมประยุกต์ PublishInterfaceServer .....	61
4.2.2 ต้นแบบโปรแกรมประยุกต์ PublishInterfaceClient .....	62
4.2.3 ต้นแบบโปรแกรมประยุกต์ DiscoveryTemplate .....	63
4.2.3.1 ส่วนจัดการกลุ่มบริการ .....	64
4.2.3.2 ส่วนจัดการทรีของคำสามัญ .....	64



## สารบัญ (ต่อ)

	หน้า
4.2.3.3 ส่วนกำหนดข้อบังคับเชิงพฤติกรรม .....	65
4.2.3.4 ส่วนแสดงข้อสนเทศเชิงพฤติกรรม .....	66
4.2.3.5 ส่วนพิจารณาเงื่อนไข .....	67
5 สรุปผลการวิจัย .....	70
5.1 สรุปผลการวิจัย .....	70
5.2 ปัญหาและข้อจำกัดที่พบจากการวิจัย .....	70
5.3 แนวทางการทำวิจัยในอนาคต .....	72
รายการอ้างอิง.....	74
ประวัติผู้เขียนวิทยานิพนธ์.....	76

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 3.1 สัญลักษณ์ทางตรรกะที่ใช้แสดงนิพจน์ภาษาอาวล์.....	23



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## สารบัญภาพ

รูปที่	หน้า
รูปที่ 1.1 โครงสร้างเอสไอเอ	1
รูปที่ 1.2 การเรียกใช้เว็บเซอร์วิส	2
รูปที่ 1.3 การค้นหาเว็บเซอร์วิสผ่านแผ่นแบบการค้นหา	3
รูปที่ 2.1 ตัวอย่างเอกสารดับเบิลยูเอสดีแอล	6
รูปที่ 2.2 อาร์ดีเอฟทริปเปิล	9
รูปที่ 2.3 ข้อมูลตัวอย่าง	10
รูปที่ 2.4 อาร์ดีเอฟโมเดลในรูปแบบกราฟ	10
รูปที่ 2.5 อาร์ดีเอฟโมเดลในรูปแบบเอ็นทริปเปิล	10
รูปที่ 2.6 อาร์ดีเอฟโมเดลในรูปแบบโนเทชัน 3	11
รูปที่ 2.7 อาร์ดีเอฟโมเดลในรูปแบบอาร์ดีเอฟเอ็กซ์เอ็มแอล	11
รูปที่ 2.8 ตัวอย่างการให้ประโยชน์จากอาร์ดีเอฟเอส	12
รูปที่ 2.9 แสดงของภาษาออนโทโลจี	13
รูปที่ 2.10 ภาษาย่อยของอวาล์	14
รูปที่ 2.11 ส่วนต่อประสานกับผู้ใช้ของตัวเสริมอวาล์	14
รูปที่ 2.12 ตัวอย่างอาร์ดีเอฟโมเดลที่ใช้ในการสอบถาม	16
รูปที่ 2.13 ส่วนต่อประสานกับผู้ใช้ของดับเบิลยูเอสดีแอลเอดิเตอร์	17
รูปที่ 3.1 โครงสร้างออนโทโลจีที่ใช้อธิบายข้อเสนอเทศกาลเชิงพฤติกรรมของบริการ	21
รูปที่ 3.2 คลาสและคุณสมบัติในอ็อบเจกต์ออนโทโลจี	23
รูปที่ 3.3 ตัวอย่างการพิจารณาค่าความจริงจากคลาสแบบแจนนับ	28
รูปที่ 3.4 ตัวอย่างเงื่อนไขที่พิจารณาค่าความจริงจากชนิดข้อมูลมากกว่า 1 ชนิด	29
รูปที่ 3.5 ขั้นตอนการแทรกข้อเสนอเทศกาลเชิงพฤติกรรม	31
รูปที่ 3.6 ข้อกำหนดในการเพิ่มอีลีเมนต์ของเอกสารดับเบิลยูเอสดีแอล รุ่นที่ 1.1	32
รูปที่ 3.7 ตำแหน่งในการแทรกข้อเสนอเทศกาลเชิงพฤติกรรม	34
รูปที่ 3.8 แชร์ดออนโทโลจีของกลุ่มบริการ TravelAgent	35
รูปที่ 3.9 พฤติกรรมของการดำเนินการ BookTicket	37
รูปที่ 3.10 พฤติกรรมของการดำเนินการ BookTicket ที่ปรับมาจากแชร์ดออนโทโลจี	37
รูปที่ 3.11 โลคัลออนโทโลจีของบริษัท AmexTravel	38
รูปที่ 3.12 ซีแมนติคอินสแตนซ์ของบริการ BookFlight	39

## สารบัญภาพ (ต่อ)

รูปที่	หน้า
รูปที่ 3.13 ซีแมนติคอินสแตนซ์ของบริการ BookFlight ที่ผ่านการอนุมาน.....	40
รูปที่ 3.14 ตัวอย่างข้อสนเทศเชิงพฤติกรรมในรูปแบบอาร์ดีเอฟเอ็กซ์เอ็มแอลสำหรับบริการ BookFlight.....	41
รูปที่ 3.15 ตัวอย่างเอกสารดับเบิลยูเอสดีแอลสำหรับบริการ BookFlight .....	44
รูปที่ 3.16 ตัวอย่างเอกสารดับเบิลยูเอสดีแอลสำหรับบริการ BookFlight ที่ถูกแทรกข้อสนเทศเชิงพฤติกรรม.....	45
รูปที่ 4.1 เฟรมเวิร์คสำหรับการโฆษณาและการค้นหาบริการเว็บเซอร์วิส .....	49
รูปที่ 4.2 การทำงานของส่วนต่อประสานสำหรับโฆษณาบริการ.....	51
รูปที่ 4.3 แผนภาพซีเควนซ์สำหรับการโฆษณาบริการ .....	52
รูปที่ 4.4 โครงสร้างการทำงานของแผนแบบการค้นหา.....	52
รูปที่ 4.5 แผนภาพซีเควนซ์สำหรับการค้นหาบริการ .....	55
รูปที่ 4.6 ตัวอย่างออนโทโลยีสำหรับการจับคู่.....	56
รูปที่ 4.7 ขั้นตอนวิธีการจับคู่คลาสกับอินสแตนซ์ .....	57
รูปที่ 4.8 ตัวอย่างการจับคู่แบบสุ่ม.....	60
รูปที่ 4.9 ตัวอย่างการจับคู่แบบต่ำสู่สูง.....	61
รูปที่ 4.10 ส่วนต่อประสานกับผู้ใช้ของ PublishInterfaceServer .....	62
รูปที่ 4.11 ส่วนต่อประสานกับผู้ใช้ของ PublishInterfaceClient .....	63
รูปที่ 4.12 ส่วนต่อประสานกับผู้ใช้ของส่วนจัดการกลุ่มบริการ .....	64
รูปที่ 4.13 ส่วนต่อประสานกับผู้ใช้ของส่วนจัดการทรีของคำสามัญ.....	65
รูปที่ 4.14 ส่วนต่อประสานกับผู้ใช้ของแผนแบบการค้นหา .....	66
รูปที่ 4.15 ส่วนต่อประสานกับผู้ใช้ของส่วนแสดงข้อสนเทศเชิงพฤติกรรม .....	67
รูปที่ 4.16 ตัวอย่างการกำหนดค่าให้กับข้อมูลเข้า .....	68
รูปที่ 4.17 ตัวอย่างผลลัพธ์ของการดำเนินการ .....	68
รูปที่ 5.1 ตัวอย่างการอนุมานด้วยจิ้นาเทียบกับเรซเซอร์ .....	71
รูปที่ 5.2 การประกาศคลาสแบบแฉงนับ CreditCard ตามปกติ.....	72
รูปที่ 5.3 การประกาศคลาสแบบแฉงนับ CreditCard โดยโพเทเจ .....	72

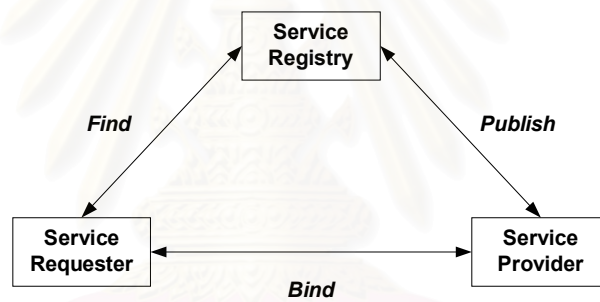
# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

เว็บเซอร์วิส (Web Service) [1] เป็นสถาปัตยกรรมของการให้บริการบนเครือข่ายอินเทอร์เน็ต โดยบริการจะถูกพัฒนาในรูปแบบของโปรแกรมที่สามารถเรียกใช้ผ่านโพรโทคอลมาตรฐาน เช่น เอชทีทีพี (HTTP) เอ็ฟทีพี (FTP) และ เอสเอ็มทีพี (SMTP) โครงสร้างของข้อความ (Message) ที่ใช้ในการติดต่อกับบริการจะอยู่ในรูปของเอกสารเอ็กซ์เอ็มแอล (XML: Extensible Markup Language) ซึ่งเป็นไปตามข้อกำหนดของโพรโทคอลโซป (SOAP: Simple Object Access Protocol) ทำให้สามารถเรียกใช้บริการได้โดยไม่ต้องขึ้นกับสถาปัตยกรรมของระบบคอมพิวเตอร์

โครงสร้างของเว็บเซอร์วิสอ้างอิงมาจากสถาปัตยกรรมเอสโอเอ (SOA: Service-Oriented Architecture) ซึ่งประกอบด้วย 3 เอนทิตี (Entity) และ 3 การดำเนินการ (Operation) ดังรูปที่ 1.1

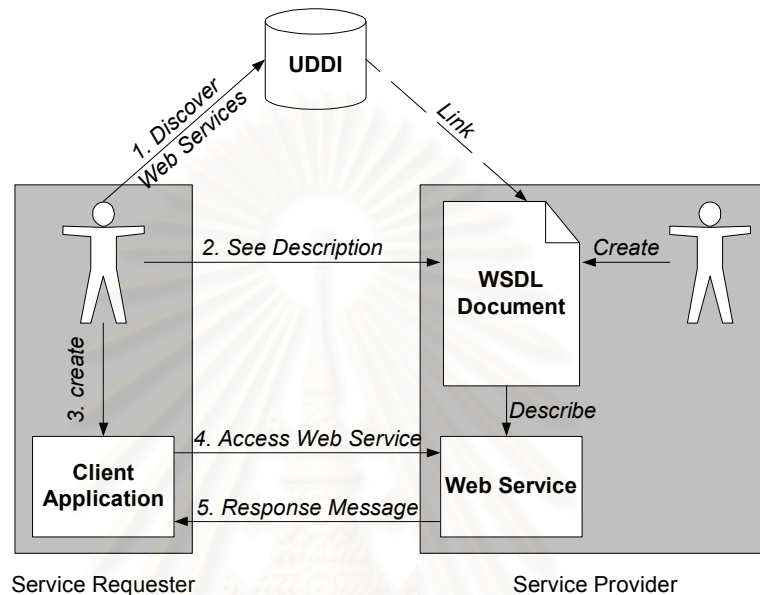


รูปที่ 1.1 โครงสร้างเอสโอเอ

1. คลังเก็บข้อมูลโฆษณาบริการ (Service Registry) ทำหน้าที่เก็บข้อมูลของบริการทั้งรายละเอียดของการจัดแบ่งหมวดหมู่ และรายละเอียดของการเชื่อมต่อกับบริการ
2. ผู้รับบริการ (Service Requester) สามารถค้นหา (Find) บริการที่ต้องการได้จากคลังเก็บข้อมูลโฆษณาบริการ
3. ผู้ให้บริการ (Service Provider) สามารถโฆษณา (Publish) บริการของตนได้ที่คลังเก็บข้อมูลโฆษณาบริการ

องค์กรด้านธุรกิจและคอมพิวเตอร์หลายองค์กรได้ทำข้อตกลงร่วมกันเพื่อกำหนดลักษณะของคลังเก็บข้อมูลโฆษณาบริการเรียกว่ายูดีดีไอ (UDDI: Universal Description, Discovery and Integration) [2] เมื่อผู้รับบริการค้นหาบริการที่ต้องการได้จากยูดีดีไอแล้ว จะใช้ข้อมูลของบริการนั้นเพื่อเข้าถึงเอกสารดับเบิลยูเอสดีแอล (WSDL: Web Service Description Language) [3] ซึ่ง

อธิบายข้อมูลเกี่ยวกับพฤติกรรมของบริการในระดับของการดำเนินการ ข้อความนำเข้า (Input Message) และ ข้อความส่งออก (Output Message) ที่บริการรองรับ รวมถึงข้อมูลของโพรโทคอลสำหรับผูกมัด (Bind) เพื่อเรียกใช้บริการ ดังแสดงในรูปที่ 1.2



รูปที่ 1.2 การเรียกใช้เว็บเซอร์วิส

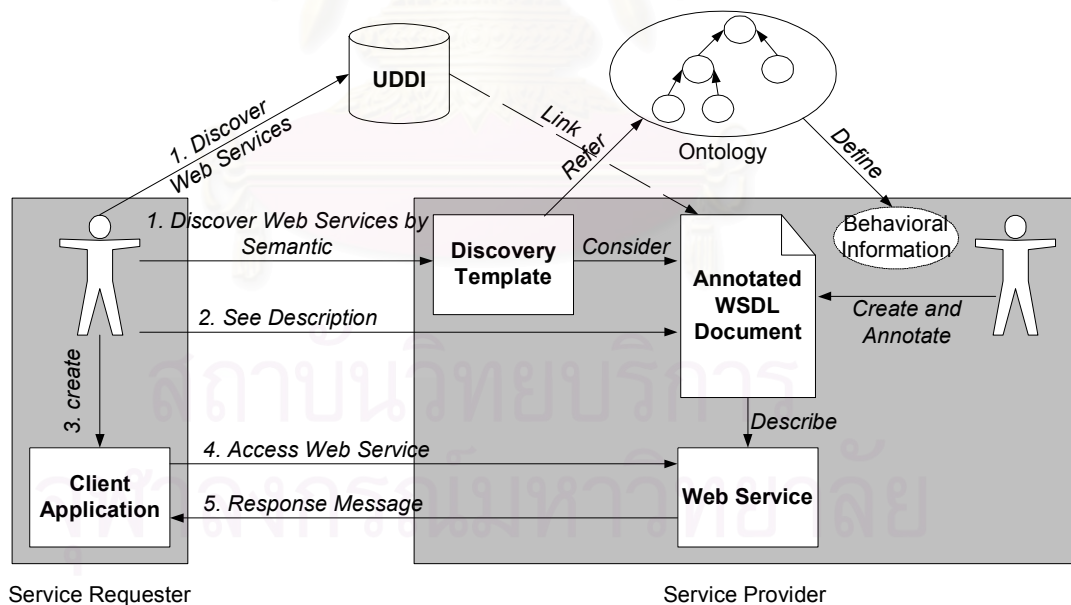
จากการเรียกใช้เว็บเซอร์วิสในรูปที่ 1.2 ปัญหาที่เกิดขึ้นคือ โครงสร้างคำอธิบายบริการในยูดีดีไอให้ข้อมูลเกี่ยวกับบริการในรูปแบบแอททริบิวต์ (Attribute) ซึ่งประกอบด้วย 4 ส่วนหลัก ได้แก่ บิสิเนสเอนทิตี (Business Entity) อธิบายรายละเอียดเกี่ยวกับองค์กรธุรกิจของผู้ให้บริการ บิสิเนสเซอร์วิส (Business Service) อธิบายรายละเอียดของบริการ ไบนดิงเทมเพลต (Binding Template) เป็นแบบจำลองสำหรับการผูกมัดบริการ และ ทีโมเดล (tModel) เป็นคำอธิบายที่ใช้ในการกำหนดชนิดของบริการ ผู้รับบริการสามารถใช้แอททริบิวต์เหล่านี้เพื่อค้นหาเว็บเซอร์วิสที่สนใจได้ แต่ในกรณีที่ผู้รับบริการต้องการค้นหาเว็บเซอร์วิสโดยพิจารณาจากพฤติกรรมของบริการ (ยกตัวอย่างเช่นต้องการค้นหาบริการจองตั๋วเครื่องบินที่จัดส่งตั๋วให้หลังจากชำระเงิน) จะทำได้ยากเพราะโครงสร้างคำอธิบายบริการในยูดีดีไอที่อยู่ในรูปแบบแอททริบิวต์นั้นไม่เอื้อต่อการอธิบายพฤติกรรมของบริการ

แม้ว่าผู้รับบริการสามารถนำข้อมูลเกี่ยวกับพฤติกรรมของบริการจากเอกสารดับเบิลยูเอสดีแอลมาพิจารณาเพื่อค้นหาบริการได้ แต่ก็ไม่อาจทราบได้ว่าบริการที่ค้นหาได้นั้นตรงกับความต้องการจริงๆหรือไม่ ทั้งนี้เพราะผู้รับบริการไม่อาจทราบความหมาย (Semantics) ที่แท้จริงจากชื่อ

ที่กำหนดให้กับส่วนต่างๆของเอกสารดับเบิลยูเอสดีแอลได้ ยกตัวอย่างเช่น การดำเนินการ "createTable" คำว่า "Table" ผู้ให้บริการอาจหมายความว่า "โต๊ะ" หรือ "ตาราง" ก็ได้

ในปัจจุบันเริ่มมีการนำออนโทโลยี (Ontology) [4] ซึ่งใช้ในการอธิบายความหมายของสิ่งต่างๆที่เราสนใจ (Domain of Interest) มาใช้อธิบายความหมายให้กับเว็บเพจ เพื่อให้การค้นหาเว็บเพจเป็นไปอย่างมีประสิทธิภาพ และสามารถค้นพบเว็บเพจที่ตรงกับความต้องการของผู้ใช้จริงๆได้ดีกว่าการค้นหาด้วยคำสำคัญ เนื่องด้วยเว็บเซอร์วิสก็เป็นทรัพยากรรูปแบบหนึ่งบนเว็บเช่นเดียวกันกับเว็บเพจ จึงได้เริ่มมีงานวิจัยที่พยายามอธิบายเว็บเซอร์วิสด้วยออนโทโลยี และทำการค้นหาเว็บเซอร์วิสตามความหมาย สำหรับการอธิบายออนโทโลยีทำได้โดยใช้ภาษาออนโทโลยีซึ่งมีพื้นฐานอยู่บนภาษาเอ็กซ์เอ็มแอล

ในงานวิจัยนี้ผู้วิจัยมีแนวคิดที่จะออกแบบและพัฒนาเฟรมเวิร์ค (Framework) สำหรับอธิบายและค้นหาเว็บเซอร์วิสจากเอกสารดับเบิลยูเอสดีแอลที่ถูกแทรกข้อสนเทศเชิงพฤติกรรม (Behavioral Information) ซึ่งกำหนดไว้ด้วยออนโทโลยี เพื่อให้ผู้รับบริการสามารถค้นหาบริการซึ่งมีความหมายและพฤติกรรมตรงกับที่ต้องการได้โดยอัตโนมัติผ่านแผ่นแบบการค้นหา (Discovery Template) ดังรูปที่ 1.3 ทั้งนี้การค้นหาเว็บเซอร์วิสด้วยแอสทริบิวต์ผ่านยูดีดีไอจะยังคงทำได้เช่นเดิม



รูปที่ 1.3 การค้นหาเว็บเซอร์วิสผ่านแผ่นแบบการค้นหา

## 1.2 วัตถุประสงค์ของการวิจัย

เพื่อออกแบบและพัฒนาเฟรมเวิร์คเพื่อสนับสนุนการโฆษณาและการค้นหาเว็บเซอร์วิสจากเอกสารดับเบิลยูเอสดีแอลที่ถูกแทรกข้อสนเทศเชิงพฤติกรรมอิงออนโทโลยี

### 1.3 ขอบเขตของการวิจัย

1. ต้นแบบเฟรมเวิร์คที่พัฒนาสามารถใช้งานได้ในสภาพแวดล้อมของระบบปฏิบัติการ ไมโครซอฟท์วินโดวส์เอ็กซ์พี เป็นอย่างน้อย
2. อับเปอร์ออนโทโลยี เซิร์ตออนโทโลยี และ โดคัลออนโทโลยี ซึ่งจะใช้ร่วมกันในการอธิบาย พฤติกรรมของเว็บเซอร์วิสจะได้รับการอธิบายในรูปแบบภาษาอาวล์
3. รูปแบบคำอธิบายบริการเป็นไปตามข้อกำหนดของอาร์ดีเอฟ รุ่นที่ 1.0 (W3C RDF Model and Syntax 1.0 Recommendation) เป็นอย่างน้อย
4. ต้นแบบเฟรมเวิร์คถูกพัฒนาโดยใช้จีนาเอพีไอ รุ่นที่ 2.0 เป็นอย่างน้อย
5. การค้นหาบริการกระทำผ่านแผนแบบการค้นหาซึ่งอ้างอิงจากออนโทโลยีที่ได้กำหนดไว้

### 1.4 ประโยชน์ที่คาดว่าจะได้รับ

ได้ต้นแบบเฟรมเวิร์คเพื่อสนับสนุนการโฆษณาและการค้นหาเว็บเซอร์วิสจากเอกสารดับเบิลยูเอสดีแอลที่ถูกแทรกข้อสนเทศเชิงพฤติกรรมของออนโทโลยี

### 1.5 วิธีดำเนินการวิจัย

1. ศึกษาทฤษฎีของเว็บเซอร์วิส
2. ศึกษาทฤษฎีของเอกสารดับเบิลยูเอสดีแอล
3. ศึกษาทฤษฎีของอาร์ดีเอฟ อาร์ดีเอฟเอส และ อาวล์
4. ออกแบบเฟรมเวิร์คสำหรับการโฆษณาและการค้นหาบริการเว็บเซอร์วิส
5. พัฒนาด้านแบบเฟรมเวิร์คสำหรับการโฆษณาและการค้นหาบริการเว็บเซอร์วิส
6. ทดสอบและปรับปรุงแก้ไขต้นแบบ
7. วิเคราะห์และสรุปผล พร้อมข้อเสนอแนะ
8. จัดทำรายงานวิทยานิพนธ์

### 1.6 โครงสร้างวิทยานิพนธ์

ในวิทยานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 4 ส่วน คือ บทที่ 2 จะกล่าวถึงเอกสารและงานวิจัยที่เกี่ยวข้อง รวมถึงทฤษฎีและเครื่องมือซอฟต์แวร์ที่ใช้ในการวิจัย บทที่ 3 จะกล่าวถึงการอธิบายเว็บเซอร์วิสด้วยข้อสนเทศเชิงพฤติกรรมของออนโทโลยี ซึ่งประกอบด้วย การออกแบบออนโทโลยีเพื่อให้อธิบายพฤติกรรมของเว็บเซอร์วิส ข้อกำหนดในการแทรกข้อสนเทศเชิงพฤติกรรมให้กับเอกสารดับเบิลยูเอสดีแอล รวมถึง ตัวอย่างข้อสนเทศและตัวอย่างการแทรกข้อสนเทศให้กับ



เอกสารฉบับเบ็ดเสร็จแอสดีแอล บทที่ 4 จะกล่าวถึงเฟรมเวิร์คสำหรับการโฆษณาและการค้นหาบริการเว็บเซอริวิส ซึ่งประกอบด้วย การออกแบบเฟรมเวิร์คสำหรับการโฆษณาและการค้นหาบริการเว็บเซอริวิส และ การพัฒนาต้นแบบโปรแกรมประยุกต์สำหรับการโฆษณาและการค้นหาบริการเว็บเซอริวิส ส่วนบทที่ 5 จะกล่าวถึงการสรุปผลการวิจัย ปัญหาและข้อจำกัดที่พบจากการวิจัย รวมถึงแนวทางการทำวิจัยในอนาคต



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## บทที่ 2

### เอกสารและงานวิจัยที่เกี่ยวข้อง

#### 2.1 ทฤษฎีที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้องในงานวิจัยนี้ ได้แก่ ดัชนีเบิลยูเอสดีแอล ออนโทโลยี อาร์ดีเอฟ (RDF: Resource Description Framework) อาร์ดีเอฟเอส (RDFS: Resource Description Framework Schema) อวาล์ (OWL: Web Ontology Language) โปรเทเจ (Protégé) เจนา (Jena) อาร์ดีคิวแอล (RDQL) และ ดัชนีเบิลยูเอสดีแอลเอดิเตอร์ (WSDL Editor) ซึ่งมีรายละเอียดดังต่อไปนี้

##### 2.1.1 ดัชนีเบิลยูเอสดีแอล [3]

ดัชนีเบิลยูเอสดีแอลเป็นเอกสารที่ใช้โครงสร้างพื้นฐานของภาษาเอ็กซ์เอ็มแอล เพื่อบรรยายเว็บเซอร์วิสในด้านรายละเอียดเกี่ยวกับการติดต่อผ่านเครือข่ายอินเทอร์เน็ต การดำเนินการที่บริการรองรับ รวมถึง ข้อความที่เกี่ยวข้องกับการเรียกใช้บริการซึ่งสามารถส่งผ่านไพธอนคอลลเซพระหว่างผู้รับบริการกับผู้ให้บริการได้ โครงสร้างของเอกสารดัชนีเบิลยูเอสดีแอลถูกแบ่งออกเป็น 2 กลุ่ม ดังตัวอย่างในรูปที่ 2.1

	<pre>&lt;?xml version="1.0"?&gt; &lt;definitions name="StockQuote"  targetNamespace="http://example.com/stockquote.wsdl" xmlns:tns="http://example.com/stockquote.wsdl" xmlns:xsd1="http://example.com/stockquote.xsd" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns="http://schemas.xmlsoap.org/wsdl/"&gt;</pre>
(ก)	<pre>&lt;types&gt; &lt;schema targetNamespace="http://example.com/stockquote.xsd" xmlns="http://www.w3.org/2000/10/XMLSchema"&gt; &lt;element name="TradePriceRequest"&gt; &lt;complexType&gt; &lt;all&gt; &lt;element name="tickerSymbol" type="string"/&gt; &lt;/all&gt; &lt;/complexType&gt; &lt;/element&gt; &lt;element name="TradePrice"&gt; &lt;complexType&gt; &lt;all&gt; &lt;element name="price" type="float"/&gt; &lt;/all&gt; &lt;/complexType&gt; &lt;/element&gt; &lt;/schema&gt; &lt;/types&gt;</pre>

รูปที่ 2.1 ตัวอย่างเอกสารดัชนีเบิลยูเอสดีแอล

(ข)	<pre>&lt;message name="GetLastTradePriceInput"&gt;   &lt;part name="body" element="xsd1:TradePriceRequest"/&gt; &lt;/message&gt;  &lt;message name="GetLastTradePriceOutput"&gt;   &lt;part name="body" element="xsd1:TradePrice"/&gt; &lt;/message&gt;</pre>
(ค)	<pre>&lt;portType name="StockQuotePortType"&gt;   &lt;operation name="GetLastTradePrice"&gt;     &lt;input message="tns:GetLastTradePriceInput"/&gt;     &lt;output message="tns:GetLastTradePriceOutput"/&gt;   &lt;/operation&gt; &lt;/portType&gt;</pre>
(ง)	<pre>&lt;binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType"&gt;   &lt;soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/&gt;   &lt;operation name="GetLastTradePrice"&gt;     &lt;soap:operation soapAction="http://example.com/GetLastTradePrice"/&gt;     &lt;input&gt;       &lt;soap:body use="literal"/&gt;     &lt;/input&gt;     &lt;output&gt;       &lt;soap:body use="literal"/&gt;     &lt;/output&gt;   &lt;/operation&gt; &lt;/binding&gt;</pre>
(จ)	<pre>&lt;service name="StockQuoteService"&gt;   &lt;documentation&gt;My first service&lt;/documentation&gt;   &lt;port name="StockQuotePort" binding="tns:StockQuoteBinding"&gt;     &lt;soap:address location="http://example.com/stockquote"/&gt;   &lt;/port&gt; &lt;/service&gt;  &lt;/definitions&gt;</pre>

รูปที่ 2.1 (ต่อ) ตัวอย่างเอกสารดับเบิลยูเอสดีแอล

### 2.1.1.1 คำอธิบายเชิงนามธรรม (Abstract Description)

คำอธิบายเชิงนามธรรมประกอบด้วยอีลีเมนต์ (Element) ที่บรรยายถึงความสามารถของเว็บเซอร์วิส ดังนี้

1. <wsdl:types> (รูปที่ 2.1 (ก)) ได้แก่ ชนิดข้อมูลที่ผู้ใช้กำหนด (User-Defined Data Type) ซึ่งอาจประกอบด้วยชนิดข้อมูลพื้นฐาน (Primitive Data Type) หลายชนิดที่อ้างอิงมาจากเอกซ์เอ็มแอลสคีมา (XML Schema) จากข้อกำหนดของโพรโทคอลSOAP อนุญาตให้มีข้อมูลเข้า (Input) เพียง 1 ชนิด และข้อมูลออก (Output) เพียง 1 ชนิด ในข้อความหนึ่งที่ถูกส่งผ่านเครือข่ายอินเทอร์เน็ต

2. <wsdl:message> (รูปที่ 2.1 (ข)) ได้แก่ ข้อความที่ถูกส่งระหว่างผู้ให้บริการและผู้ให้บริการ ซึ่งภายใน 1 การดำเนินการอนุญาตให้มีได้เพียง 3 ข้อความ คือ
  - Input Message แสดงข้อความเข้า กำหนดโดยอิลีเมนต์ <wsdl:input>
  - Output Message แสดงข้อความออก กำหนดโดยอิลีเมนต์ <wsdl:output>
  - Fault Message แสดงข้อความแสดงข้อผิดพลาด กำหนดโดยอิลีเมนต์ <wsdl:fault>
3. <wsdl:operation> (รูปที่ 2.1 (ค)) ได้แก่ การดำเนินการ
4. <wsdl:portType> (รูปที่ 2.1 (ค)) ได้แก่ กลุ่มการดำเนินการ

#### 2.1.1.2 คำอธิบายเชิงรูปธรรม (Concrete Description)

คำอธิบายเชิงรูปธรรมประกอบด้วยอิลีเมนต์ที่บรรยายถึงการผูกมัดผู้ให้บริการเข้ากับบริการ ดังนี้

1. <wsdl:binding> (รูปที่ 2.1 (ง)) ได้แก่ ส่วนเชื่อมโยงระหว่างคำอธิบายเชิงนามธรรมกับคำอธิบายเชิงรูปธรรม รวมถึงเป็นที่เก็บข้อมูลของโพรโทคอลที่ใช้ในการติดต่อกับบริการ
2. <wsdl:port> (รูปที่ 2.1 (จ)) ได้แก่ ส่วนที่ใช้กำหนดยูอาร์แอล (URL) ของส่วน binding
3. <wsdl:service> (รูปที่ 2.1 (จ)) ได้แก่ อิลีเมนต์ที่บรรยายถึงพอร์ททั้งหมดภายในเว็บไซต์

#### 2.1.2 ออนโทโลยี [4]

ความหมายของออนโทโลยีมีผู้ให้คำจำกัดความไว้หลากหลายทั้ง สาขาปรัชญา และสาขาเทคโนโลยีสารสนเทศ (Information Technology) ทางสาขาปรัชญานั้นออนโทโลยีหมายถึง “สาขาหนึ่งของอภิปรัชญาที่ศึกษาเกี่ยวกับธรรมชาติและความสัมพันธ์ของสิ่งต่างๆในโลก” (“A branch of metaphysics concerned with the nature and relations of being”) และ “ทฤษฎีที่เกี่ยวกับธรรมชาติของสิ่งต่างๆในโลกหรือประเภทของสิ่งที่มีอยู่จริง” (“A particular theory about the nature of being or the kinds of existents”) ส่วนสาขาเทคโนโลยีสารสนเทศคำจำกัดความของออนโทโลยีที่ได้รับการยอมรับกันอย่างแพร่หลายคือ “ข้อกำหนดเกี่ยวกับแนวคิด” (“The specification of a conceptualization” [8])

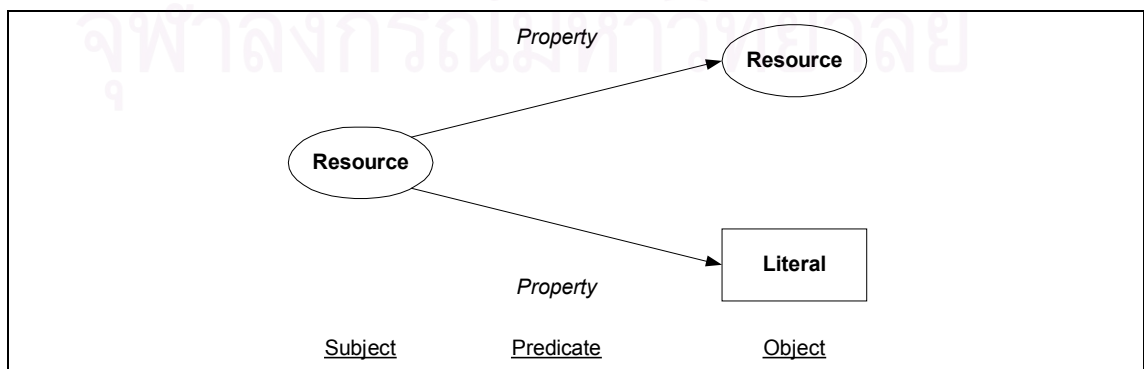
ออนโทโลยีถูกใช้เพื่อบรรยายแนวคิดของโดเมนหรือขอบเขตความสนใจใดๆ ในรูปของสิ่งต่างๆ ที่อยู่ภายในโดเมนและความสัมพันธ์ระหว่างสิ่งเหล่านั้น โดยแสดงออกมาในรูปแบบของคำสามัญ (Common Words) เพื่อให้สามารถเข้าใจและนำไปใช้ร่วมกันได้ ในภาษาธรรมชาติ (Natural Language) สามารถแสดงโดยใช้ คำศัพท์ (Vocabulary) และ ประโยค (Sentence) ซึ่งเกิดจากการรวมคำศัพท์ต่างๆ เข้าไว้ด้วยกันเพื่อแสดงความสัมพันธ์ระหว่างคำศัพท์เหล่านั้น ส่วนในสาขาเทคโนโลยีสารสนเทศจะแสดงในรูปแบบของระบบสัญลักษณ์ (Notation) ยกตัวอย่างเช่น คลาส (Class) อินสแตนซ์ (Instance) ความสัมพันธ์ (Relationship) คุณสมบัติ (Property) และ กฎ (Rule) โดยใช้ภาษาสำหรับแสดงความรู้ (Knowledge Representation Language) ซึ่งมีความชัดเจนและเที่ยงตรงมากกว่าคำศัพท์และประโยคในภาษาธรรมชาติ ทั้งนี้เพื่อให้ซอฟต์แวร์และเครื่องมือสามารถนำไปประมวลผลได้

### 2.1.3 อาร์ดีเอฟ [9]

อาร์ดีเอฟเป็นภาษามาตรฐาน (W3C Recommendation) สำหรับบรรยายทรัพยากร (Resource) ซึ่งอยู่ในรูปแบบอิเล็กทรอนิกส์ เช่น แฟ้มข้อมูล (File) หรือแนวคิดเกี่ยวกับสิ่งต่างๆ เช่น “คน” “สัตว์” “สิ่งของ” ฯลฯ

อาร์ดีเอฟโมเดล (RDF Model) หรือ สเตตเมนต์ (Statement) จะอธิบายทรัพยากรหนึ่งในรูปของทริปเปิล (Triple) เนื่องจากประกอบด้วยองค์ประกอบ 3 ส่วน ดังแสดงในรูปที่ 2.2

1. ซับเจกต์ (Subject) ได้แก่ ทรัพยากรซึ่งถูกบรรยายด้วยเพรดิเคทและออบเจกต์
2. เพรดิเคท (Predicate) ได้แก่ ส่วนที่แสดงคุณสมบัติของทรัพยากรซึ่งแสดงความสัมพันธ์ระหว่างซับเจกต์ และ ออบเจกต์
3. ออบเจกต์ (Object) ได้แก่ ส่วนที่ถูกอ้างถึงซึ่งมีความสัมพันธ์กับซับเจกต์ตามเพรดิเคท ออบเจกต์สามารถเป็นได้อย่างใดอย่างหนึ่งระหว่าง ทรัพยากร และ ค่าสัญลักษณ์ (Literal)



รูปที่ 2.2 อาร์ดีเอฟทริปเปิล

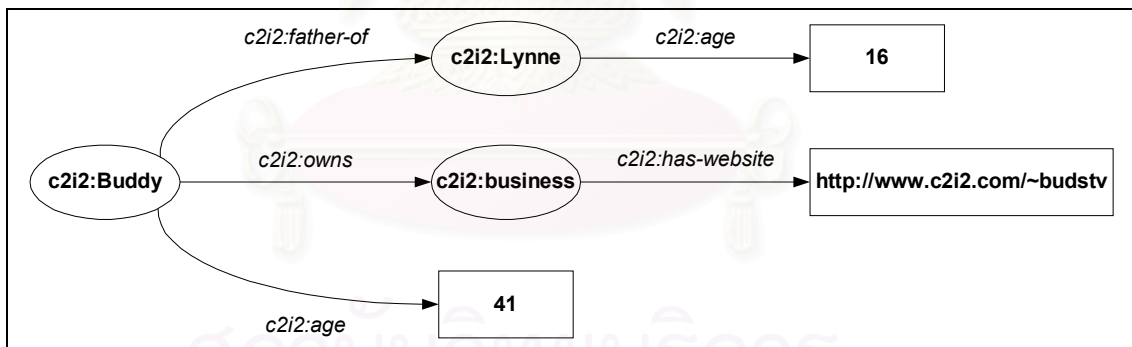
ทั้งซัพเจกต์ เพรดิเคท และ ออบเจกต์ (ยกเว้นออบเจกต์ที่อยู่ในรูปแบบสัญพจน์) ต้องถูกระบุด้วยยูอาร์ไอ (URI: Uniform Resource Identifier) ซึ่งเป็นวากยสัมพันธ์ (Syntax) มาตรฐานสำหรับอักขระ (String) ที่ใช้ระบุทรัพยากรในเวิร์ลไวด์เว็บ โดยยูอาร์ไอประกอบด้วย 2 ส่วนคือ เนมสเปซ (Namespace) และโลคัลเนม (Local Name)

รูปที่ 2.3 แสดงข้อมูลตัวอย่างซึ่งสามารถบรรยายด้วยอาร์ดีเอฟได้หลายรูปแบบ ดังต่อไปนี้ (กำหนดให้เนมสเปซ c2i2 มีค่าเป็น http://www.c2i2.com/rdf#)

Buddy Belden owns a business.  
 Buddy is 41 years old.  
 The business has a web site accessible at http://www.c2i2.com/~budstv.  
 Buddy is the father of Lynne.  
 Lynne is 16 years old.

รูปที่ 2.3 ข้อมูลตัวอย่าง [10]

1. อธิบายในรูปแบบอาร์ดีเอฟกราฟ (รูปที่ 2.4)



รูปที่ 2.4 อาร์ดีเอฟโมเดลในรูปแบบกราฟ [10]

2. อธิบายในรูปแบบเอ็นทริปเปิล (N-Triple) (รูปที่ 2.5)

```

    <c2i2:Buddy> <c2i2:father-of> <c2i2:Lynne> .
    <c2i2:Lynne> <c2i2:age> "16" .
    <c2i2:Buddy> <c2i2:owns> <c2i2:business> .
    <c2i2:business> <c2i2:has-website> "http://www.c2i2.com/~budstv" .
    <c2i2:Buddy> <c2i2:age> "41" .
    
```

รูปที่ 2.5 อาร์ดีเอฟโมเดลในรูปแบบเอ็นทริปเปิล [10]

### 3. อธิบายในรูปแบบโนเทชัน 3 (Notation 3) (รูปที่ 2.6)

```
# Base: http://www.c2i2.com/rdf
@prefix : <#> .

:Lynne
  :age "16" .

:Buddy
  :age "41" ;
  :father-of :Lynne ;
  :owns :business .

:business
  :has-website "http://www.c2i2.com/~budstv" .
```

รูปที่ 2.6 อาร์ดีเอฟโมเดลในรูปแบบโนเทชัน 3 [10]

### 4. อธิบายในรูปแบบอาร์ดีเอฟเอ็กซ์เอ็มแอล (RDF/XML) ซึ่งเป็นภาษาที่ใช้โครงสร้างพื้นฐานของภาษาเอ็กซ์เอ็มแอล (รูปที่ 2.7)

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:c2i2="http://www.c2i2.com/rdf#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xml:base="http://www.c2i2.com/rdf">
  <rdf:Description rdf:about="http://www.c2i2.com/rdf#Lynne">
    <c2i2:age> 16 </c2i2:age>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.c2i2.com/rdf#Buddy">
    <c2i2:age> 41 </c2i2:age>
    <c2i2:owns>
      <rdf:Description rdf:about="http://www.c2i2.com/rdf#business">
        <c2i2:has-website> http://www.c2i2.com/~budstv </c2i2:has-website>
      </rdf:Description>
    </c2i2:owns>
    <c2i2:father-of rdf:resource="http://www.c2i2.com/rdf#Lynne"/>
  </rdf:Description>
</rdf:RDF>
```

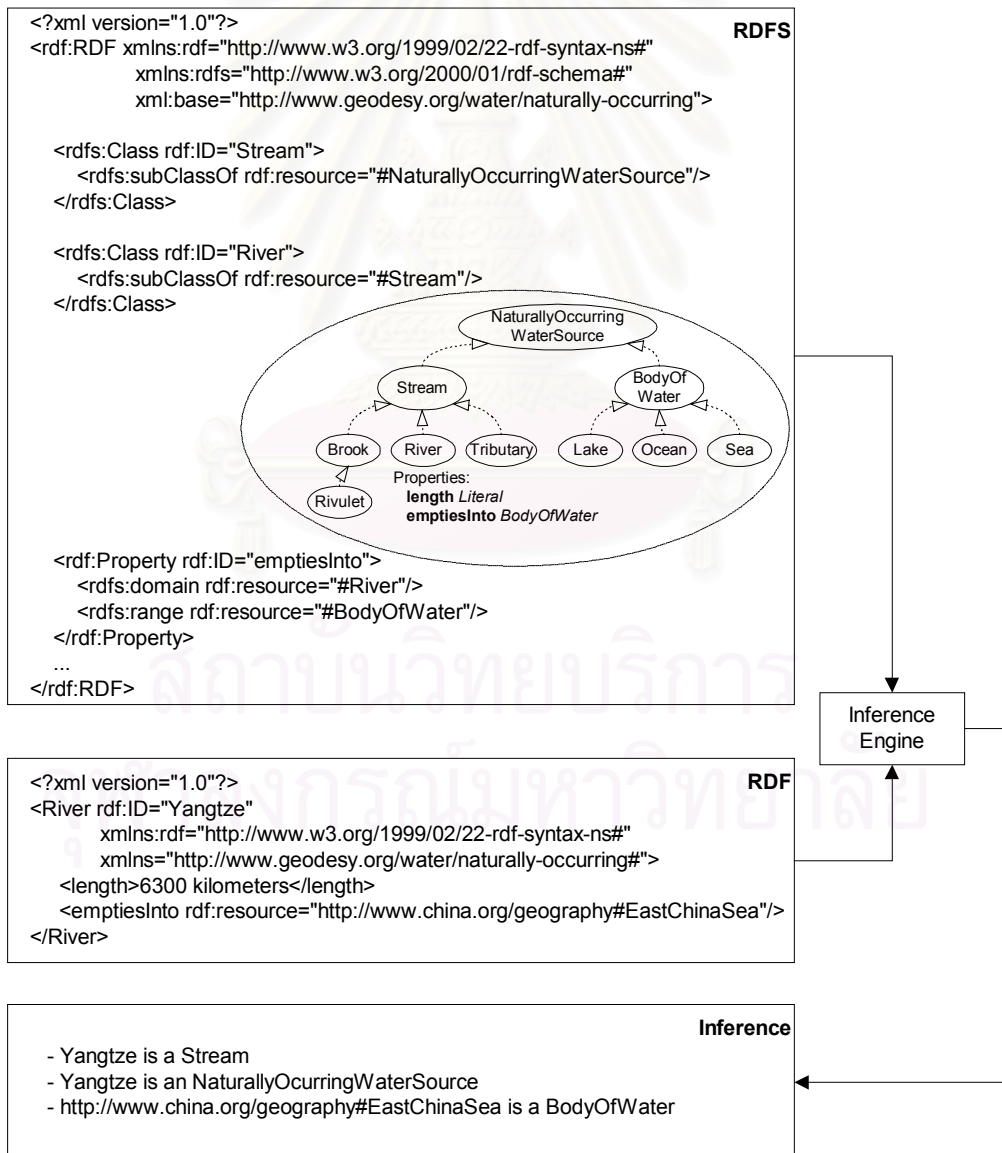
รูปที่ 2.7 อาร์ดีเอฟโมเดลในรูปแบบอาร์ดีเอฟเอ็กซ์เอ็มแอล [10]

#### 2.1.4 อาร์ดีเอฟเอส [11]

อาร์ดีเอฟเอสเป็นภาษามาตรฐาน (W3C Recommendation) ซึ่งใช้โครงสร้างพื้นฐานของภาษาอาร์ดีเอฟเพื่อบรรยายลำดับชั้นของแนวคิดและคุณสมบัติ (Hierarchy of Concepts and Properties) เพื่อสร้างออนโทโลยีอย่างง่าย ทำให้สามารถอนุมาน (Inference) หาข้อเท็จจริง

(Fact) นอกเหนือจากที่ประกาศให้กับทรัพยากรได้ โดยอาร์ดีเอฟเอสประกอบด้วยอีลีเมนต์หลักดังต่อไปนี้

1. <rdfs:Class> ได้แก่ อีลีเมนต์สำหรับอธิบายคลาส
2. <rdfs:subClassOf> ได้แก่ อีลีเมนต์สำหรับอธิบายคุณสมบัติที่แสดงความสัมพันธ์แบบชั้นคลาส (Subclass) ระหว่างคลาส
3. <rdfs:Property> ได้แก่ อีลีเมนต์สำหรับอธิบายคุณสมบัติของคลาส ประกอบด้วยอีลีเมนต์ย่อย คือ
  - <rdfs:domain> ได้แก่ อีลีเมนต์สำหรับกำหนดคลาสของซับเจกต์
  - <rdfs:range> ได้แก่ อีลีเมนต์สำหรับกำหนดคลาสของออบเจกต์



รูปที่ 2.8 ตัวอย่างการใช้ประโยชน์จากอาร์ดีเอฟเอส [12]



รูปที่ 2.8 แสดงออนโทโลยีของ “NaturallyOccurringWaterSource” ซึ่งแสดงคำสามัญหรือแนวคิดในโดเมนของแหล่งน้ำธรรมชาติ รวมถึงความสัมพันธ์ระหว่างแนวคิดในโดเมน ในส่วน RDF ทรัพยากร “Yangtze” สามารถอ้างถึงออนโทโลยีเพื่ออธิบายความหมายของตัวเองได้ นอกจากนี้สามารถอนุมานเพื่อหาข้อเท็จจริงเพิ่มเติมได้ ดังแสดงในส่วน Inference

### 2.1.5 อวาล์ [13]

อวาล์เป็นภาษาออนโทโลยีมาตรฐาน (W3C Recommendation) ซึ่งใช้โครงสร้างพื้นฐานของภาษาอาร์ดีเอฟเอส และเป็นส่วนหนึ่งของกลุ่มภาษาที่ดับเบิลยูทีซี (W3C: World Wide Web Consortium) กำหนดให้ใช้กับเว็บเชิงความหมาย ซึ่งสามารถแสดงได้ดังรูปที่ 2.9

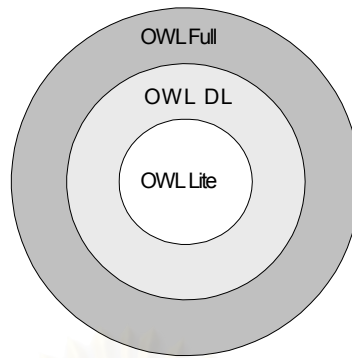
Higher Semantics	DAML+OIL, OWL
Semantics	RDFS
Simple Semantics	RDF
Structure	xml schema
Syntax: Data	xml

รูปที่ 2.9 แสดงของภาษาออนโทโลยี

จากรูปที่ 2.9 ในส่วน Syntax: Data และในส่วน Structure ทั้งเอ็กซ์เอ็มแอลและเอ็กซ์เอ็มแอลสคีมามาไม่ได้อธิบายความหมายใดๆให้กับเอกสาร เนื่องจากเอ็กซ์เอ็มแอลเป็นเพียงวากยสัมพันธ์ที่กำหนดให้กับเอกสารเพื่อให้สามารถใช้เอกสารร่วมกันได้บนเครือข่ายอินเทอร์เน็ต ส่วนเอ็กซ์เอ็มแอลสคีมามาทำหน้าที่กำหนดโครงสร้างของเอกสารเอ็กซ์เอ็มแอล รวมถึงกำหนดชนิดข้อมูลนอกเหนือจากชนิดข้อมูลพื้นฐาน

จากรูปที่ 2.9 ในส่วน Simple Semantics และในส่วน Semantics สามารถอธิบายความหมายของเอกสารได้เนื่องจาก อาร์ดีเอฟเป็นแบบจำลองข้อมูล (Datamodel) ที่ใช้อธิบายทรัพยากรและความสัมพันธ์ระหว่างทรัพยากร ส่วนอาร์ดีเอฟเอสเป็นภาษาที่สนับสนุนคำศัพท์ที่ใช้อธิบายคลาสและคุณสมบัติของทรัพยากรที่อยู่ในรูปอาร์ดีเอฟ

จากรูปที่ 2.9 ในส่วน Higher Semantics สามารถอธิบายความหมายของเอกสารได้มากขึ้น โดยอวาล์เป็นภาษาที่พัฒนามาจากภาษาดีเอเอ็มแอลพลัสออยล์ (DAML + OIL: DARPA Agent Markup Language + Ontology Inference Layer) สนับสนุนคำศัพท์ที่ใช้อธิบายคลาสและคุณสมบัติเพิ่มเติมจากอาร์ดีเอฟเอส ภาษาอวาล์สามารถแบ่งออกได้เป็น 3 ภาษาย่อย (Sublanguage) ดังรูปที่ 2.10

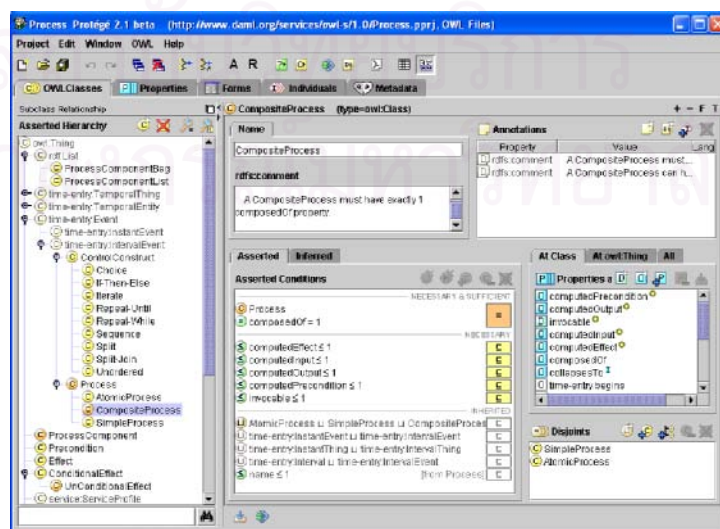


รูปที่ 2.10 ภาษาย่อยของอวาล์

1. อวาล์ไลท์ (OWL Lite) เป็นภาษาย่อยขั้นต้นของอวาล์ สนับสนุนการจัดแบ่งลำดับชั้นของคลาส และการกำหนดข้อบังคับของภาษาอวาล์ (Restriction) เบื้องต้น
2. อวาล์ดีแอล (OWL DL) เป็นภาษาย่อยขั้นต่อมาของอวาล์ สามารถอธิบายออนโทโลยีได้ละเอียดกว่าอวาล์ไลท์ และสามารถกำหนดเวลาที่แน่นอนในการประมวลผลได้
3. อวาล์ฟูลล์ (OWL Full) เป็นภาษาย่อยขั้นสูงของอวาล์ สามารถอธิบายออนโทโลยีได้ละเอียดที่สุด แต่ไม่สามารถกำหนดเวลาที่แน่นอนในการประมวลผลได้ ในปัจจุบันจึงยังไม่มีเครื่องมือซอฟต์แวร์ใดที่สนับสนุนความสามารถทั้งหมดของอวาล์ฟูลล์

### 2.1.6 โพรเทจ [14]

โพรเทจเป็นเครื่องมือซอฟต์แวร์สำหรับสร้างออนโทโลยีและระบบฐานความรู้ (Knowledge-Based Systems) สนับสนุนภาษาอวาล์ผ่านตัวเสริมอวาล์ (OWL Plug-in) ซึ่งมีส่วนต่อประสานกับผู้ใช้ดังรูปที่ 2.11



รูปที่ 2.11 ส่วนต่อประสานกับผู้ใช้ของตัวเสริมอวาล์

ตัวเสริมอวาล์มีคุณสมบัติหลักๆดังนี้

1. บรรจุ (Load) และ บันทึก (Save) คอนโทโลจีในรูปแบบอวาล์และอาร์ดีเอฟ
2. ตรวจแก้ (Edit) และแสดงผลคลาสรวมถึงคุณสมบัติของภาษาอวาล์ในรูปแบบกราฟิก
3. สามารถกำหนดลักษณะเชิงตรรกะของคลาส (Logical Class Characteristic) ได้ในรูปแบบนิพจน์ (Expression)
4. สามารถประมวลผลร่วมกับเครื่องมืออนุมาน (Inference Engine) อื่นๆได้
5. สามารถตรวจแก้อินสแตนซ์ของภาษาอวาล์เพื่อใช้อธิบายเว็บเชิงความหมายได้

### 2.1.7 จีนา [15]

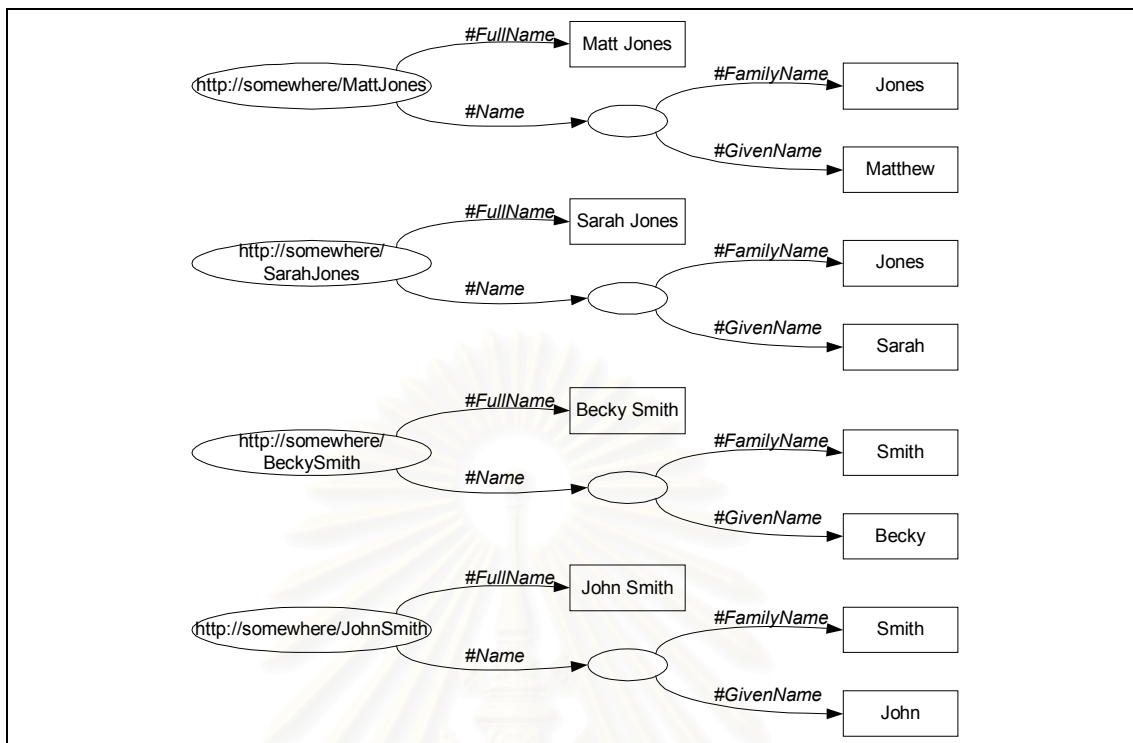
จีนาเป็นเอพีไอภาษาจาวา (Java API) ที่ได้รับความนิยมจากดัดเบิลยูทริซี ใช้สำหรับสร้างโปรแกรมประยุกต์ทางด้านเว็บเชิงความหมาย สนับสนุนสภาพแวดล้อมในการพัฒนาโปรแกรมสำหรับจัดการเอกสาร อาร์ดีเอฟ อาร์ดีเอฟเอส อวาล์ รวมถึงการอนุมานโดยใช้กฎ (Rule-Based Inference Engine)

ส่วนประกอบของจีนามีดังนี้

1. อาร์ดีเอฟเอพีไอ (RDF API)
2. นำเข้าและส่งออกอาร์ดีเอฟโมเดลในรูปแบบ อาร์ดีเอฟเอ็กซ์เอ็มแอล โนเทชัน 3 และ เอ็นทีริปเปิล
3. อวาล์เอพีไอ (OWL API)
4. สนับสนุนการจัดการอาร์ดีเอฟโมเดลทั้งในหน่วยความจำและจากฐานข้อมูล
5. สนับสนุนภาษาอาร์ดีคิวแอล (RDQL – Query Language) สำหรับสอบถามอาร์ดีเอฟโมเดล

### 2.1.8 อาร์ดีคิวแอล [16]

อาร์ดีคิวแอลเป็นภาษาสำหรับสอบถาม (Query Language) สำหรับอาร์ดีเอฟโมเดลที่อยู่ในสภาพแวดล้อมของจีนา โดยมองอาร์ดีเอฟโมเดลเป็นข้อมูลในรูปแบบตรีปเปิล ทำให้รูปแบบการสอบถามต้องถูกกำหนดตามแบบแผนของกราฟ ดังตัวอย่างในรูปที่ 2.12



รูปที่ 2.12 ตัวอย่างอาร์ติเฟโมเดลที่ใช้ในการสอบถาม [16]

รูปแบบการสอบถามมีลักษณะดังต่อไปนี้

```
SELECT ?x, ?fname
WHERE (?x <#FullName> ?fname)
```

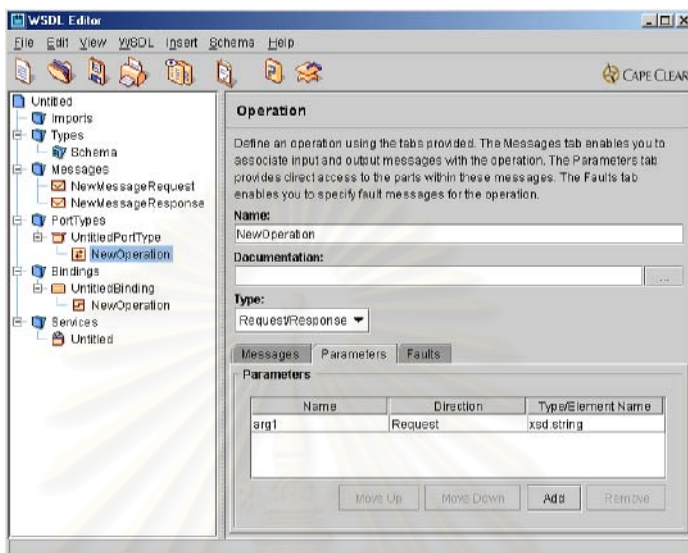
ซึ่งหมายถึงการค้นหา x และ fname ใดๆที่คุณสมบัติ FullName ของ x มีค่าเป็น fname ผลลัพธ์ที่ได้มีลักษณะดังต่อไปนี้

x	fname
http://somewhere/JohnSmith/	"John Smith"
<http://somewhere/RebeccaSmith/>	"Becky Smith"
<http://somewhere/SarahJones/>	"Sarah Jones"
http://somewhere/MattJones/	"Matt Jones"

### 2.1.9 ดับเบิลยูเอสดีแอลเอดิเตอร์ [17]

ดับเบิลยูเอสดีแอลเอดิเตอร์เป็นเครื่องมือซอฟต์แวร์เชิงกราฟิก (Graphical Tool) ของบริษัทเคปเคลียร์ซอฟต์แวร์ (Cape Clear Software) สำหรับอำนวยความสะดวกในการจัดการเอกสารดับเบิลยูเอสดีแอล เช่น การสร้าง การแก้ไข และการตรวจสอบซึ่งจะกระทำเทียบกับดับ

เบิลยูเอสดีแอลสคีมา (WSDL Schema) สำหรับส่วนต่อประสานกับผู้ใช้ของดับเบิลยูเอสดีแอลเอดิเตอร์แสดงดังรูปที่ 2.13



รูปที่ 2.13 ส่วนต่อประสานกับผู้ใช้ของดับเบิลยูเอสดีแอลเอดิเตอร์

## 2.2 งานวิจัยที่เกี่ยวข้อง

จากการศึกษาเบื้องต้นพบว่ามีงานวิจัยที่เกี่ยวข้องจำนวน 3 งานวิจัย โดยแบ่งออกเป็นงานวิจัยที่เกี่ยวข้องกับการเพิ่มความหมายให้เอกสารดับเบิลยูเอสดีแอลเพื่ออธิบายข้อเสนอเทศการดำเนินการ (Operational Information) โดยใช้ออนโทโลยี งานวิจัยที่เกี่ยวข้องกับการแสดงเอกสารดับเบิลยูเอสดีแอลให้อยู่ในรูปแบบอาร์ดีเอฟ และงานวิจัยที่เกี่ยวข้องกับเฟรมเวิร์คสำหรับการโฆษณา (Advertising) และการสอบถาม (Querying) เว็บเซอร์วิสโดยอาศัยข้อบังคับเชิงพฤติกรรมและกฎ ได้แก่ “Adding Semantics to Web Services Standards”: *Karthik Sivashanmugam* “Supercharging WSDL with RDF – Managing Structured Web Service Metadata”: *Uche Ogbuji* และ “Discovering Web Services Using Behavioral Constraints and Ontology”: *Natenapa Sriharee* ตามลำดับ ซึ่งมีรายละเอียดดังต่อไปนี้

### 2.2.1 งานวิจัย “Adding Semantics to Web Services Standards” [5]

งานวิจัยนี้นำเสนอแนวคิดในการเพิ่มความหมายให้เอกสารดับเบิลยูเอสดีแอล เพื่ออธิบายข้อเสนอเทศการดำเนินการ ข้อมูลเข้า และ ข้อมูลออก ของเว็บเซอร์วิส โดยใช้ความสามารถในการเพิ่มอ็ลลิเมนต์และแอททริบิวต์ตามข้อกำหนดของเอกสารดับเบิลยูเอสดีแอล รุ่นที่ 1.2 (WSDL Specification version 1.2) เพื่อเชื่อมโยงส่วนต่างๆในโครงสร้างของเอกสารดับเบิลยูเอสดีแอลเข้า

กับออนโทโลยี ทำให้สามารถแสดงคำอธิบายเว็บเซอร์วิสในรูปแบบที่คอมพิวเตอร์สามารถแปลความหมายได้ พร้อมทั้ง

1. นำเสนอการเพิ่มอ็ลลีเมนต์สำหรับอธิบายเงื่อนไขก่อนการทำงาน (Precondition) และผลกระทบ (Effect) โดยที่การดำเนินการหนึ่งๆอาจมีเงื่อนไขก่อนการทำงานและผลกระทบหลายอย่าง ซึ่งเงื่อนไขก่อนการทำงานอาจเป็นเงื่อนไขเชิงตรรกะ (Logical Condition) ที่ต้องเป็นจริงเพื่อให้สามารถประมวลผลการดำเนินการได้ ส่วนผลกระทบจะเป็นการเปลี่ยนแปลงหลังจากการประมวลผลการดำเนินการ สำหรับการเพิ่มอ็ลลีเมนต์สำหรับอธิบายเงื่อนไขก่อนการทำงานและผลกระทบจะอยู่ในรูปอ็ลลีเมนต์ลูกของอ็ลลีเมนต์การดำเนินการในเอกสารดับเบิลยูเอสดีแอล

งานวิจัยนี้ได้ให้เหตุผลสนับสนุนการเพิ่มอ็ลลีเมนต์สำหรับอธิบายเงื่อนไขก่อนการทำงานและผลกระทบ โดยยกกรณีการคัดเลือกบริการโดยอัตโนมัติที่พิจารณาแต่เพียงการดำเนินการ ข้อมูลเข้า และ ข้อมูลออก เท่านั้น หลังจากที่ได้คัดเลือกบริการอย่างอัตโนมัติได้แล้วจำนวนหนึ่งแล้ว ผู้รับบริการสามารถพิจารณาเงื่อนไขก่อนการทำงานและผลกระทบของบริการ เพื่อคัดเลือกบริการที่เหมาะสมอีกต่อหนึ่ง

2. นำเสนอขั้นตอนวิธี (Algorithm) เพื่อการคัดเลือกบริการโดยกำหนดให้ผู้รับบริการป้อนความต้องการ (Requirement) ของเว็บเซอร์วิสผ่านแม่แบบ (Template) ที่อ้างอิงออนโทโลยี

ผู้วิจัยจะใช้แนวทางการเพิ่มความหมายให้กับเอกสารดับเบิลยูเอสดีแอลเช่นเดียวกับงานวิจัยนี้เพื่อสร้างเฟรมเวิร์ค แต่การค้นหาเว็บเซอร์วิสโดยอัตโนมัติจะพิจารณาทั้งจาก การดำเนินการ ข้อมูลเข้า ข้อมูลออก เงื่อนไขก่อนการทำงาน และ ผลกระทบ ของเว็บเซอร์วิส

## 2.2.2 งานวิจัย “Supercharging WSDL with RDF – Managing Structured Web Service Metadata” [6]

บทความนี้ได้นำเสนอการแสดงเอกสารดับเบิลยูเอสดีแอลให้อยู่ในรูปแบบอาร์ดีเอฟ โดยเปลี่ยนอ็ลลีเมนต์ `<wsdl:message>` `<wsdl:operation>` `<wsdl:portType>` `<wsdl:binding>` `<wsdl:port>` และ `<wsdl:service>` ของเอกสารดับเบิลยูเอสดีแอลให้อยู่ในรูปแบบอ็ลลีเมนต์ของอาร์ดีเอฟ และครอบอ็ลลีเมนต์ทั้งหมดนี้ด้วยอ็ลลีเมนต์ `<rdf:RDF>` ตามข้อกำหนดของอาร์ดีเอฟ รุ่นที่ 1.0 (RDF Model and Syntax 1.0 Recommendation) การแปลงเอกสารดับเบิลยูเอสดีแอลให้อยู่ในรูปแบบอาร์ดีเอฟนี้ อาจทำได้อัตโนมัติโดยเครื่องมือซอฟต์แวร์ หรือเอ็กซ์เอสแอลที (XSLT: Extensible Stylesheet Language Transformation)

ประโยชน์ที่ได้รับจากการเขียนเอกสารดับเบิลยูเอสดีแอลในรูปแบบอาร์ดีเอฟ คือการแสดงผลโมเดลของเอกสารดับเบิลยูเอสดีแอลในรูปแบบกราฟด้วยเครื่องมือแบบเห็นภาพสำหรับอาร์

ดีเอฟ (RDF Visualization Tool) แต่ว่าการเขียนเอกสารดับเบิลยูเอสดีแอลในรูปแบบอาร์ดีเอฟนี้ ไม่ถือว่าถูกต้องสมบูรณ์ตามข้อกำหนดของเอกสารดับเบิลยูเอสดีแอล ทำให้ต้องเก็บทั้งเอกสารดับเบิลยูเอสดีแอลปกติและเอกสารดับเบิลยูเอสดีแอลในรูปแบบอาร์ดีเอฟไว้อย่างละหนึ่งชุด อีกทั้งยังต้องรักษาความสอดคล้องระหว่างเอกสารทั้งสองชุดนี้ด้วย

จากแนวคิดในบทความนี้ ทำให้ผู้วิจัยได้แนวคิดในการอธิบายพฤติกรรมของบริการในรูปแบบอาร์ดีเอฟ แล้วจึงนำไปใส่เป็นคำอธิบายประกอบให้กับเอกสารดับเบิลยูเอสดีแอลของบริการนั้น แทนการเปลี่ยนแปลงอีลีเมนต์ที่มีอยู่ เพื่อให้เอกสารดับเบิลยูเอสดีแอลยังคงถูกต้องสมบูรณ์ตามข้อกำหนด และสามารถอธิบายพฤติกรรมของบริการได้

### 2.2.3 งานวิจัย “Discovering Web Services Using Behavioral Constraints and Ontology” [7]

ในงานวิจัยนี้นำเสนอวิธีการยกระดับคำอธิบายเว็บเซอร์วิสให้กับเอกสารดับเบิลยูเอสดีแอลโดยใช้ข้อสนเทศเชิงพฤติกรรมที่อ้างอิงออนโทโลยีซึ่งแบ่งออกเป็น 3 ระดับ ดังนี้

1. อัปเปอร์ออนโทโลยี (Upper Ontology) ใช้อธิบายความสามารถ (Capability) ในเชิงพฤติกรรมของเว็บเซอร์วิสโดยอาศัย
  - การดำเนินการ (Operation)
  - ข้อมูลเข้า (Input)
  - ข้อมูลออกแบบมีเงื่อนไข (Conditional Output)
  - ข้อมูลออกแบบไม่มีเงื่อนไข (Unconditional Output)
  - เงื่อนไขก่อนการทำงาน (Precondition)
  - ผลกระทบแบบมีเงื่อนไข (Conditional Effect)
  - ผลกระทบแบบไม่มีเงื่อนไข (Unconditional Effect)
2. แชร์ดออนโทโลยี (Shared Ontology) เป็นออนโทโลยีสำหรับเซอร์วิสโดเมน (Service Domain) ซึ่งผู้ให้บริการสามารถใช้อ้างอิงเพื่อโฆษณาบริการของตน
3. โลคัลออนโทโลยี (Local Ontology) เป็นออนโทโลยีที่ปรับ (Derive) จากแชร์ดออนโทโลยี เพื่อให้ผู้ให้บริการสามารถเพิ่มรายละเอียดที่ต้องการได้ ในกรณีที่แชร์ดออนโทโลยีไม่ครอบคลุมรายละเอียดทั้งหมดที่ต้องการโฆษณา

นอกจากนี้ยังสามารถนำเสนอเฟรมเวิร์คสำหรับการโฆษณาและการสอบถามเว็บไซต์โดยอาศัยข้อบังคับเชิงพฤติกรรม (Behavioral Constraint) และ กฎ (Rule-Based Reasoning) เพื่อตัดสินใจพฤติกรรมของบริการเมื่อมีเงื่อนไขบนข้อมูลออกหรือผลกระทบ

ผู้วิจัยจะใช้หลักการของแอปเปอร็อนโทโลจี แชร็ดออนโทโลจี และ โลกัลดออนโทโลจี ของงานนี้ในการพัฒนาเฟรมเวิร์ค



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



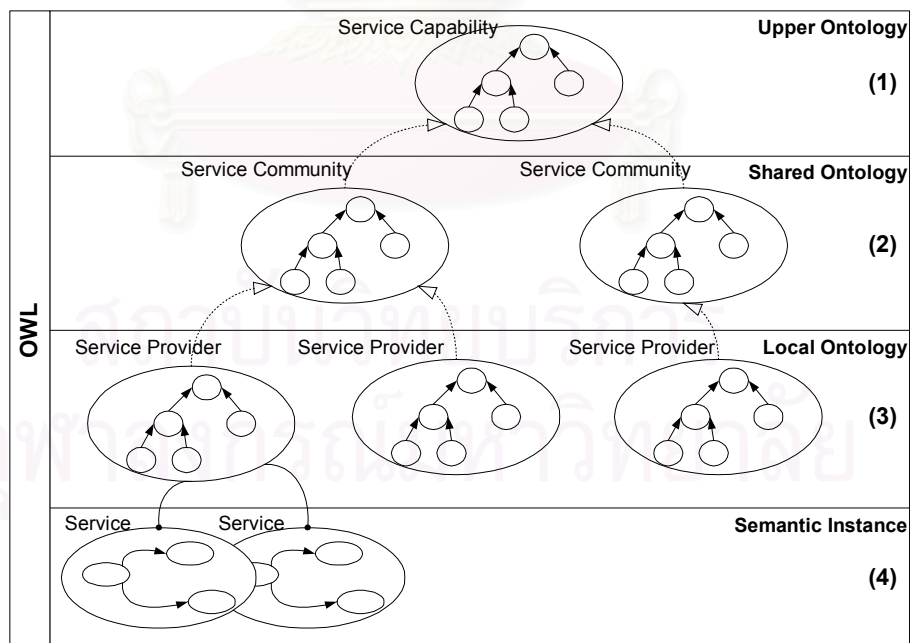
### บทที่ 3

## การอธิบายเว็บเซอร์วิสด้วยข้อสนเทศเชิงพฤติกรรมของออนโทโลยี

เนื่องจากเฟรมเวิร์คที่งานวิจัยนี้ได้ออกแบบและพัฒนาจะทำการโฆษณาและค้นหาบริการเว็บเซอร์วิสจากเอกสารดับเบิลยูเอสดีแอลที่ถูกแทรกข้อสนเทศเชิงพฤติกรรมซึ่งกำหนดไว้ด้วยออนโทโลยี ดังนั้นการออกแบบออนโทโลยีเพื่อใช้อธิบายพฤติกรรมของเว็บเซอร์วิส และข้อกำหนดในการแทรกข้อสนเทศเชิงพฤติกรรมให้กับเอกสารดับเบิลยูเอสดีแอล จึงถือเป็นส่วนหนึ่งของงานวิจัยนี้ สำหรับการออกแบบออนโทโลยีเพื่อใช้อธิบายพฤติกรรมของเว็บเซอร์วิส จะอธิบายในหัวข้อที่ 3.1 การแทรกข้อสนเทศเชิงพฤติกรรมให้กับเอกสารดับเบิลยูเอสดีแอล จะอธิบายในหัวข้อที่ 3.2 และ ตัวอย่างข้อสนเทศเชิงพฤติกรรมและตัวอย่างการแทรกข้อสนเทศเชิงพฤติกรรม จะแสดงในหัวข้อที่ 3.3 ส่วนการออกแบบและพัฒนาเฟรมเวิร์ค จะอธิบายในบทที่ 4

### 3.1 การออกแบบอับเปอร์ออนโทโลยีเพื่อใช้อธิบายพฤติกรรมของเว็บเซอร์วิส

งานวิจัยนี้ได้นำแนวความคิดในการสร้างข้อสนเทศเชิงพฤติกรรมสำหรับอธิบายบริการจากงานวิจัย [7] มาปรับปรุงเพื่อสร้างข้อสนเทศเชิงพฤติกรรมโดยอ้างอิงออนโทโลยีซึ่งมีการแบ่งออกเป็น 3 ระดับ ได้แก่ อับเปอร์ออนโทโลยี แชร์ออนโทโลยี และ โลคัลออนโทโลยี ดังรูปที่ 3.1



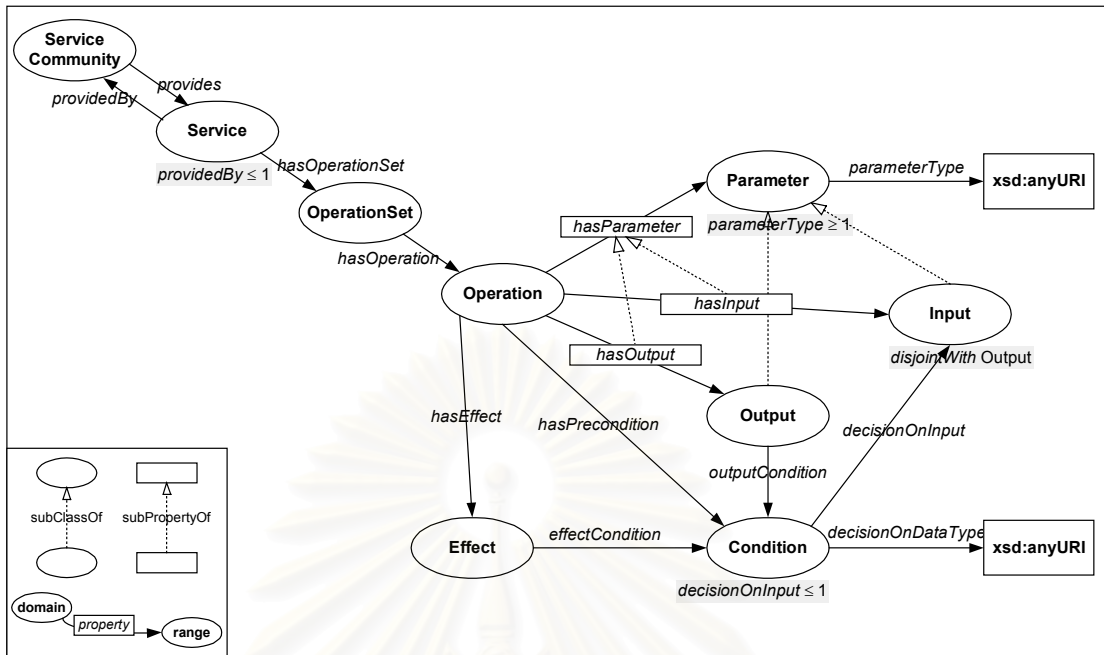
รูปที่ 3.1 โครงสร้างออนโทโลยีที่ใช้อธิบายข้อสนเทศเชิงพฤติกรรมของบริการ

จากรูปที่ 3.1 (1) อับเปอร์ออนโทโลยีเป็นออนโทโลยีในระดับบนสุดใช้อธิบายความสามารถในเชิงพฤติกรรมที่ทุกๆเว็บเซอร์วิสมีส่วนร่วม เปรียบได้กับสคีมาที่ใช้กำหนดว่าพฤติกรรมของเว็บเซอร์วิสหนึ่งๆประกอบด้วยส่วนใดบ้าง และส่วนเหล่านั้นมีความสัมพันธ์กันอย่างไร จากรูปที่ 3.1 (2) เซร์คออนโทโลยีเป็นออนโทโลยีที่อ้างอิงมาจากอับเปอร์ออนโทโลยีใช้อธิบายพฤติกรรมของบริการที่ผู้ให้บริการภายในกลุ่มบริการหนึ่งๆตกลงใช้ร่วมกัน จากรูปที่ 3.1 (3) โลกัลออนโทโลยีเป็นออนโทโลยีในระดับของผู้ให้บริการแต่ละราย ซึ่งปรับมาจากเซร์คออนโทโลยีหรืออับเปอร์ออนโทโลยีเพื่ออธิบายพฤติกรรมของบริการเพิ่มเติมจากเซร์คออนโทโลยี ในกรณีที่เซร์คออนโทโลยีไม่ครอบคลุมรายละเอียดทั้งหมดที่ผู้ให้บริการแต่ละรายต้องการ จากรูปที่ 3.1 (4) ซีแมนติกอินสแตนซ์ (Semantic Instance) เป็นข้อสนเทศเชิงพฤติกรรมของบริการหนึ่งๆ ซึ่งผู้ให้บริการนำเสนอและอยู่ในรูปอินสแตนซ์ของคลาสในโลคัลออนโทโลยีและเซร์คออนโทโลยี

สำหรับการออกแบบอับเปอร์ออนโทโลยีเพื่อใช้อธิบายพฤติกรรมของเว็บเซอร์วิส เนื่องจากเอกสารดับเบิลยูเอสดีแอลมีข้อมูลเกี่ยวกับพฤติกรรมของบริการอยู่แล้วคือ การดำเนินการ (อ็ดีเมนต์ <wsdl:operation>) ข้อมูลเข้า (อ็ดีเมนต์ <wsdl:input>) และ ข้อมูลออก (อ็ดีเมนต์ <wsdl:output>) แต่ข้อมูลเหล่านี้อาจไม่เพียงพอสำหรับใช้ในการคัดเลือกบริการ งานวิจัยนี้ได้นำแนวความคิดจากงานวิจัย [7] มาปรับปรุงเพื่อทำการกำหนดข้อสนเทศเชิงพฤติกรรมของบริการเพิ่มเติมจากเอกสารดับเบิลยูเอสดีแอล รวมเป็น การดำเนินการ ข้อมูลเข้า ข้อมูลออกแบบมีเงื่อนไข ข้อมูลออกแบบไม่มีเงื่อนไข เงื่อนไขก่อนการทำงาน ผลกระทบแบบมีเงื่อนไข และ ผลกระทบแบบไม่มีเงื่อนไข ทั้งนี้เพื่อช่วยให้สามารถคัดเลือกบริการได้ละเอียดขึ้น

สำหรับคลาสและคุณสมบัติทั้งหมดในอับเปอร์ออนโทโลยีได้แสดงในรูปที่ 3.2 โดยใช้สัญลักษณ์สำหรับ คลาส พรอพเพอร์ตี้ (Property) ซับคลาส และ ซับพรอพเพอร์ตี้ (Subproperty) ที่อ้างอิงมาจากอวล์เอส (OWL-S) [19] ส่วนการแสดงความบังคับของภาษาอวล์และลักษณะเชิงตรรกะของคลาสในวิทยานิพนธ์นี้ จะใช้สัญลักษณ์ทางตรรกะ (นิพจน์ที่มีพื้นหลังเป็นสีเทา) ซึ่งอ้างอิงมาจากเครื่องมือซอฟต์แวร์โพรเทจ [14] ดังตารางที่ 3.1

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.2 คลาสและคุณสมบัติในอ็ปเปอร์ออนโทโลยี

ตารางที่ 3.1 สัญลักษณ์ทางตรรกะที่ใช้แสดงนิพจน์ภาษาอาวล์

OWL Element	Symbol	Key	Example	Meaning of example
allValuesFrom	$\forall$	*	$\forall$ children Male	All children must be of type Male
someValuesFrom	$\exists$	?	$\exists$ children Lawyer	At least one child must be of type Lawyer
hasValue	$\exists$	\$	rich $\exists$ true	The rich property must have the value true
cardinality	=	=	children = 3	There must be exactly 3 children
minCardinality	$\geq$	>	children $\geq$ 3	There must be at least 3 children
maxCardinality	$\leq$	<	children $\leq$ 3	There must be at most 3 children
complementOf	$\neg$	!	$\neg$ Parent	Anything that is not of type Parent
intersectionOf	$\cap$	&	Human $\cap$ Male	All Humans that are Male
unionOf	$\cup$		Doctor $\cup$ Lawyer	Anything that is either Doctor or Lawyer
enumeration	{...}	{ }	{male female}	The individuals male or female

รายละเอียดของคลาสและคุณสมบัติทั้งหมดในอ็ปเปอร์ออนโทโลยี จะอธิบายในหัวข้อที่ 3.1.1.1 และ หัวข้อที่ 3.1.1.2 ตามลำดับ ส่วนรายละเอียดของนิพจน์ภาษาอาวล์ในอ็ปเปอร์ออนโทโลยี จะอธิบายในหัวข้อที่ 3.1.1.3 สำหรับหัวข้อที่ 3.1.1.4 จะอธิบายเกี่ยวกับการตัดสินใจค่าความจริงของเงื่อนไข

### 3.1.1.1 คลาสในอ็ปเปอร์ออนโทโลยี

จากรูปที่ 3.2 อ็ปเปอร์ออนโทโลยีประกอบด้วยคลาสดังต่อไปนี้

1. ServiceCommunity อธิบายกลุ่มบริการที่เว็บเซอร์วิสสังกัด

2. **Service** อธิบายบริการเว็บเซอร์วิส เปรียบได้กับอิลีเมนต์ <wsdl:service> ของเอกสารดับเบิลยูเอสดีแอล
3. **OperationSet** อธิบายกลุ่มการดำเนินการ เปรียบได้กับอิลีเมนต์ <wsdl:portType> และ อิลีเมนต์ <wsdl:binding> ของเอกสารดับเบิลยูเอสดีแอล
4. **Operation** อธิบายการดำเนินการ เปรียบได้กับอิลีเมนต์ <wsdl:operation> ของเอกสารดับเบิลยูเอสดีแอล
5. **Parameter** อธิบายพารามิเตอร์ของการดำเนินการ
6. **Input** เป็นชั้นคลาสของคลาส Parameter ใช้อธิบายข้อมูลเข้าของการดำเนินการ เปรียบได้กับอิลีเมนต์ <wsdl:input> ของเอกสารดับเบิลยูเอสดีแอล
7. **Output** เป็นชั้นคลาสของคลาส Parameter ใช้อธิบายข้อมูลออกของการดำเนินการ เปรียบได้กับอิลีเมนต์ <wsdl:output> ของเอกสารดับเบิลยูเอสดีแอล
8. **Effect** อธิบายผลกระทบ
9. **Condition** อธิบายเงื่อนไขโดยใช้กำหนดเงื่อนไขก่อนการทำงานของการทำงานของการดำเนินการ (Precondition) เงื่อนไขที่ทำให้เกิดข้อมูลออกหนึ่งๆ หรือเงื่อนไขที่ทำให้เกิดผลกระทบหนึ่งๆ สำหรับข้อมูลออกที่ได้ทุกครั้งของการเรียกใช้การดำเนินการจะเรียกว่าเป็นข้อมูลแบบไม่มีเงื่อนไข (Unconditional Output) แต่หากการได้ข้อมูลออกนั้นขึ้นกับเงื่อนไขบางอย่างก็จะเรียกข้อมูลออกนั้นว่าข้อมูลแบบมีเงื่อนไข (Conditional Output) สำหรับผลกระทบก็เป็นไปในลักษณะเดียวกัน หากผลกระทบใดเกิดขึ้นทุกครั้งของการเรียกใช้การดำเนินการจะเรียกว่าเป็นผลกระทบแบบไม่มีเงื่อนไข (Unconditional Effect) แต่หากจะเกิดขึ้นหรือไม่ขึ้นกับเงื่อนไขบางอย่างก็จะเรียกว่าเป็นผลกระทบแบบมีเงื่อนไข (Conditional Effect)
10. **DataType** เป็นคลาสแม่ (Parent Class) ใช้อธิบายชนิดข้อมูลที่ผู้ใช้กำหนด (ไม่ถูกแสดงในรูปที่ 3.2 เนื่องจากไม่มีความสัมพันธ์กับคลาสอื่นๆ)

### 3.1.1.2 คุณสมบัติในอ็อบเจกต์ออนโทโลยี

ในภาษาอาร์แวงคุณสมบัติออกเป็น 2 ชนิด คือ คุณสมบัติเชิงวัตถุ (ObjectProperty) และ คุณสมบัติเชิงชนิดข้อมูล (DatatypeProperty) สำหรับคุณสมบัติเชิงวัตถุ ใช้แสดงความสัมพันธ์ระหว่างอินสแตนซ์ของคลาส ส่วนคุณสมบัติเชิงชนิดข้อมูล ใช้แสดงความสัมพันธ์ระหว่างอินสแตนซ์ของคลาสกับค่าสัจพจน์ หรือระหว่างอินสแตนซ์ของคลาสกับชนิดข้อมูลของเอ็กซ์เอ็มแอลสคีมา

จากรูปที่ 3.2 คุณสมบัติเชิงวัตถุในอ็อบเจกต์โอโนโทโลยีมีดังต่อไปนี้

1. **provides** แสดงความสัมพันธ์ระหว่างอินสแตนซ์ของคลาส ServiceCommunity กับอินสแตนซ์ของคลาส Service ใช้อธิบายว่ากลุ่มบริการสนับสนุนบริการใดบ้าง
2. **providedBy** แสดงความสัมพันธ์ระหว่างอินสแตนซ์ของคลาส Service กับอินสแตนซ์ของคลาส ServiceCommunity ใช้อธิบายว่าบริการเป็นสมาชิกของกลุ่มบริการใด
3. **hasOperationSet** แสดงความสัมพันธ์ระหว่างอินสแตนซ์ของคลาส Service กับอินสแตนซ์ของคลาส OperationSet ใช้อธิบายว่าบริการประกอบด้วยกลุ่มการดำเนินการใดบ้าง
4. **hasOperation** แสดงความสัมพันธ์ระหว่างอินสแตนซ์ของคลาส OperationSet กับอินสแตนซ์ของคลาส Operation ใช้อธิบายว่ากลุ่มการดำเนินการประกอบด้วยการดำเนินการใดบ้าง
5. **hasParameter** แสดงความสัมพันธ์ระหว่างอินสแตนซ์ของคลาส Operation กับอินสแตนซ์ของคลาส Parameter ใช้อธิบายว่าการดำเนินการประกอบด้วยพารามิเตอร์ใดบ้าง
6. **hasInput** เป็นซับพรอพเพอร์ตี้ของคุณสมบัติ hasParameter แสดงความสัมพันธ์ระหว่างอินสแตนซ์ของคลาส Operation กับอินสแตนซ์ของคลาส Input ใช้อธิบายว่าการดำเนินการประกอบด้วยข้อมูลเข้าใดบ้าง
7. **hasOutput** เป็นซับพรอพเพอร์ตี้ของคุณสมบัติ hasParameter แสดงความสัมพันธ์ระหว่างอินสแตนซ์ของคลาส Operation กับอินสแตนซ์ของคลาส Output ใช้อธิบายว่าการดำเนินการประกอบด้วยข้อมูลออกใดบ้าง
8. **hasPrecondition** แสดงความสัมพันธ์ระหว่างอินสแตนซ์ของคลาส Operation กับอินสแตนซ์ของคลาส Condition ใช้อธิบายเงื่อนไขที่ต้องเป็นจริงก่อนที่การดำเนินการจะทำงานได้
9. **hasEffect** แสดงความสัมพันธ์ระหว่างอินสแตนซ์ของคลาส Operation กับอินสแตนซ์ของคลาส Effect ใช้อธิบายว่าการดำเนินการประกอบด้วยผลกระทบใดบ้าง
10. **outputCondition** แสดงความสัมพันธ์ระหว่างอินสแตนซ์ของคลาส Output กับอินสแตนซ์ของคลาส Condition ใช้อธิบายเงื่อนไขที่เมื่อเป็นจริงจะทำให้ได้ข้อมูลออกนั้น

11. **effectCondition** แสดงความสัมพันธ์ระหว่างอินสแตนซ์ของคลาส Effect กับอินสแตนซ์ของคลาส Condition ใช้อธิบายเงื่อนไขที่เมื่อเป็นจริงจะทำให้ได้ผลกระทบนั้น
12. **decisionOnInput** แสดงความสัมพันธ์ระหว่างอินสแตนซ์ของคลาส Condition กับอินสแตนซ์ของคลาส Input ใช้อธิบายว่าค่าความจริงของเงื่อนไขจะพิจารณาจากข้อมูลเข้าตัวใด

สำหรับคุณสมบัติ **outputCondition** และ **effectCondition** ซึ่งเป็นคุณสมบัติที่แสดงว่าข้อมูลออกหรือผลกระทบบางอย่างของการดำเนินการจะเกิดขึ้นเฉพาะเมื่อเงื่อนไขที่กำหนดเป็นจริงเท่านั้น โดยเงื่อนไขนั้นจะเป็นจริงหรือไม่จะขึ้นอยู่กับค่าข้อมูลเข้าบางอย่าง (คุณสมบัติ **decisionOnInput**) ดังนั้นในเชิงความหมาย การดำเนินการหนึ่งๆจะมีพฤติกรรมได้หลายแบบ กล่าวคือสามารถให้ข้อมูลออกหรือผลกระทบได้หลายรูปแบบ ขึ้นอยู่กับเงื่อนไขบนข้อมูลเข้าที่กำหนด ตัวอย่างเช่น การดำเนินการซื้อ BookTicket ของกลุ่มบริการ TravelAgent ซึ่งทำการจองตั๋วเครื่องบินให้ลูกค้า อาจมีพฤติกรรมคือ ให้ข้อมูลออกเป็นตั๋วเครื่องบินที่จองให้ลูกค้าหากลูกค้าให้ข้อมูลเข้าเป็นบัตรเครดิต Amex และให้ผลกระทบเป็นตัวนั้นได้รับการจัดส่งไปยังลูกค้าด้วย หรืออาจมีพฤติกรรมรูปแบบอื่น คือ หากลูกค้าให้ข้อมูลเข้าเป็นบัตรเครดิตอื่นที่ไม่ใช่ Amex ก็จะทำให้ข้อมูลออกเป็นตั๋วเครื่องบินที่จองให้ลูกค้าพร้อมทั้งค่าธรรมเนียมที่ลูกค้าต้องเสียเพิ่ม ซึ่งอันที่จริงแล้วการดำเนินการหนึ่งๆจะสามารถมีพฤติกรรมหลายแบบหรือไม่นั้นเป็นเรื่องของการออกแบบเว็บเซอร์วิส กล่าวคือเราอาจจะออกแบบให้การดำเนินการหนึ่งๆมีพฤติกรรมที่ตายตัว (UnConditional) ไปเลยก็ได้ เช่นจากตัวอย่างข้างต้นเราอาจแยกการดำเนินการของเว็บเซอร์วิสนั้นเป็น การดำเนินการ BookTicketWithAmex กับการดำเนินการ BookTicketWithOtherCards เลยก็ได้ โดยแต่ละการดำเนินการก็จะมีข้อมูลออกและผลกระทบที่ตายตัว อย่างไรก็ตามการออกแบบเว็บเซอร์วิสให้มีการดำเนินการอย่างไรบ้างนั้นเป็นประเด็นนอกเหนืองานวิจัยนี้ ดังนั้นเพื่อความยืดหยุ่น ผู้วิจัยเลือกที่จะอธิบายการดำเนินการแบบกลางๆโดยให้สามารถกำหนดเงื่อนไขกำกับข้อมูลออกและผลกระทบได้ เพื่อให้สามารถแสดงพฤติกรรมที่หลากหลายของการดำเนินการได้

จากรูปที่ 3.2 คุณสมบัติเชิงชนิดข้อมูลในอับเปอร์ออนโทโลยีมีดังต่อไปนี้

1. **parameterType** แสดงความสัมพันธ์ระหว่างอินสแตนซ์ของคลาส Parameter กับข้อมูลชนิด xsd:anyURI ของเอ็กซ์เอ็มแอลสคีมา ใช้อธิบายชนิดพารามิเตอร์ ซึ่งสามารถกำหนดให้เป็นยูอาร์ไอใดๆก็ได้
2. **decisionOnDataType** แสดงความสัมพันธ์ระหว่างอินสแตนซ์ของคลาส Condition กับข้อมูลชนิด xsd:anyURI ของเอ็กซ์เอ็มแอลสคีมา ใช้อธิบายชนิดข้อมูลที่เงื่อนไขใช้พิจารณาค่าความจริง ซึ่งชนิดข้อมูลนี้สามารถกำหนดให้เป็นยูอาร์ไอใดๆก็ได้ โดยทั่วไปจะกำหนดให้เป็นชนิดข้อมูลจากเอ็กซ์เอ็มแอลสคีมา หรือชนิดข้อมูลที่ผู้กำหนดซึ่งเป็นซับคลาสของคลาส DataType

### 3.1.1.3 นิพจน์ภาษาอาวลีในอับเปอร์ออนโทโลยี

จากรูปที่ 3.2 ในอับเปอร์ออนโทโลยีมีการกำหนดนิพจน์ภาษาอาวลีให้กับ คลาส และคุณสมบัติ ดังต่อไปนี้

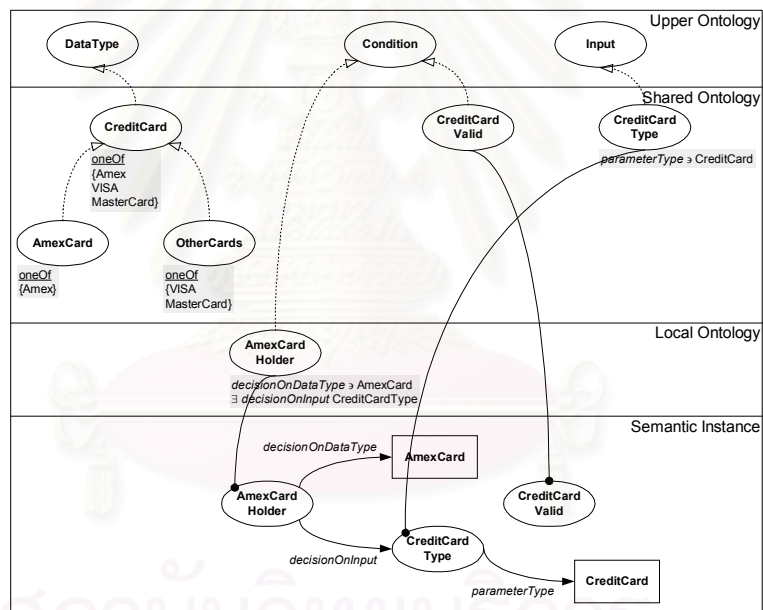
1. กำหนดให้คลาส Input และคลาส Output เป็นคลาสไร้สมาชิกร่วม (Disjoint Classes)
2. กำหนดให้คุณสมบัติ provides และคุณสมบัติ providedBy เป็นคุณสมบัติที่ผกผันกัน (Inverse) และกำหนดให้อินสแตนซ์ของคลาส Service ประกอบด้วยคุณสมบัติ providedBy ได้  $\leq 1$  คุณสมบัติ ใช้อธิบายว่ากลุ่มบริการหนึ่งสามารถสนับสนุนบริการได้หลายบริการ แต่บริการหนึ่งสามารถสังกัดกลุ่มบริการได้มากที่สุด 1 กลุ่มบริการ
3. กำหนดให้อินสแตนซ์ของคลาส Parameter ประกอบด้วยคุณสมบัติ parameterType ได้  $\geq 1$  คุณสมบัติ ใช้อธิบายว่าพารามิเตอร์แต่ละตัวต้องถูกระบุชนิดพารามิเตอร์อย่างน้อย 1 ชนิด
4. กำหนดให้อินสแตนซ์ของคลาส Condition ประกอบด้วยคุณสมบัติ decisionOnInput ได้  $\leq 1$  คุณสมบัติ ใช้อธิบายว่าเงื่อนไขสามารถพิจารณาค่าความจริงจากข้อมูลเข้าได้มากที่สุด 1 ข้อมูล

### 3.1.1.4 การตัดสินค่าความจริงของเงื่อนไข

ในงานวิจัยนี้ได้แบ่งเงื่อนไขที่กำหนดให้กับ เงื่อนไขก่อนการทำงาน ข้อมูลออกแบบมีเงื่อนไข และผลกระทบแบบมีเงื่อนไข ออกเป็น 2 ประเภท คือ

1. เงื่อนไขที่สามารถตัดสินค่าความจริงได้จากการพิจารณาข้อสนเทศเชิงพฤติกรรมของบริการในขณะค้นหาบริการ
2. เงื่อนไขที่ไม่สามารถตัดสินค่าความจริงได้จากการพิจารณาข้อสนเทศเชิงพฤติกรรมของบริการในขณะค้นหาบริการ เนื่องจากเงื่อนไขประเภทนี้ต้องใช้ข้อมูลนอกเหนือจากข้อสนเทศเชิงพฤติกรรมของบริการ ยกตัวอย่างเช่น ข้อมูลที่อยู่ในฐานข้อมูล

สำหรับเงื่อนไขที่สามารถตัดสินค่าความจริงได้ ผู้วิจัยได้เสนอการพิจารณาค่าความจริงจากคลาสแบบแจกแจง (Enumerated Class) ที่กำหนดไว้ในออนโทโลยี โดยการตัดสินค่าความจริงจะพิจารณาจากค่าของข้อมูลเข้าที่กำหนดให้กับเงื่อนไขผ่านคุณสมบัติ `decisionOnInput` ซึ่งค่าของข้อมูลเข้าจะทำให้เงื่อนไขเป็นจริงหรือไม่ ขึ้นอยู่กับคลาสแบบแจกแจงที่กำหนดให้กับเงื่อนไขผ่านคุณสมบัติ `decisionOnDataType` ดังตัวอย่างในรูปที่ 3.3



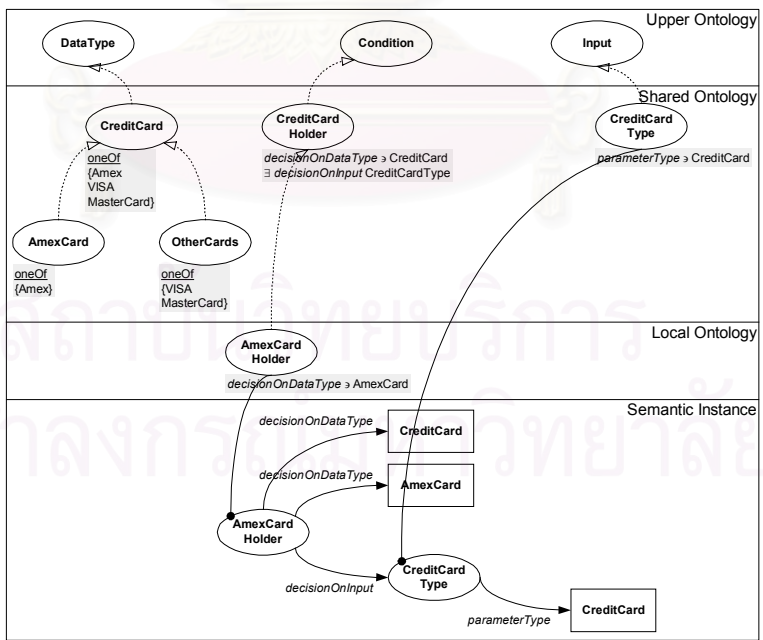
รูปที่ 3.3 ตัวอย่างการพิจารณาค่าความจริงจากคลาสแบบแจกแจง

จากตัวอย่างในรูปที่ 3.3 ซึ่งแสดงเงื่อนไขที่เกี่ยวข้องกับการดำเนินการของเว็บเซอร์วิสตัวหนึ่งซึ่งเกี่ยวกับบัตรเครดิต ได้แก่เงื่อนไข `CreditCardValid` และเงื่อนไข `AmexCardHolder` สำหรับเงื่อนไข `CreditCardValid` นั้นมีความหมายว่าการดำเนินการจะสามารถทำงานได้หากบัตรเครดิตของผู้รับบริการมีความถูกต้องใช้งานได้เช่นยังมีวงเงินเหลือพอจ่ายค่าบริการ ดังนั้นอินสแตนซ์ `CreditCardValid` ในส่วน `Semantic Instance` ของรูปที่ 3.3 จึงเป็นเงื่อนไขที่ไม่สามารถตัดสินค่าความจริงได้ขณะค้นหาบริการ (Query Time) เนื่องจากการตัดสินค่าความจริงว่าบัตรเครดิตของผู้รับบริการใช้งานได้หรือไม่ จะต้องไปสอบถามจากบริษัทบัตรเครดิตว่าผู้รับ



บริการรายนี้ยังมีวงเงินเหลือพอหรือไม่ ซึ่งการสอบถามนี้จะเป็นสิ่งที่เกิดขึ้นในขณะที่เว็บเซอร์วิส ถูกเรียกให้ทำงานจริงๆแล้ว (Run Time) สำหรับเงื่อนไข AmexCardHolder ซึ่งในส่วน Semantic Instance ของรูปที่ 3.3 ระบุว่าเงื่อนไขที่ขึ้นกับข้อมูลเข้า CreditCardType โดยหาก CreditCardType มีค่าเป็นชนิด AmexCard แล้ว เงื่อนไข AmexCardHolder ก็จะเป็นจริง แต่หากข้อมูลเข้า CreditCardType มีค่าเป็นชนิด OtherCards ซึ่งหมายถึงบัตรเครดิตอื่นๆได้แก่ VISA และ MasterCard เงื่อนไข AmexCardHolder จะเป็นเท็จ ดังนั้นเงื่อนไข AmexCardHolder ในที่นี้จึงเป็นเงื่อนไขที่สามารถตัดสินค่าความจริงได้ในขณะค้นหาบริการ (โดยพิจารณาจากข้อมูลเข้าจากผู้รับบริการ) ซึ่งในกรณีนี้คือ CreditCardType ที่มีชนิดข้อมูลเป็น CreditCard ซึ่งมีค่าเป็นแบบแฉงนับนั่นเอง

เนื่องจากคลาส Condition ในอ็ปเปอร์ออนโทโลยีไม่ได้ถูกกำหนดไว้ด้วยนิพจน์ภาษา อวาล์ว่าสามารถประกอบด้วยคุณสมบัติ decisionOnDataType ได้มากที่สุดเท่าใด อินสแตนซ์ ของคลาส Condition จึงสามารถมีคุณสมบัติ decisionOnDataType ได้หลายๆคุณสมบัติ ซึ่งในที่นี้หมายถึงการใช้ชนิดข้อมูลหลายชนิดในการพิจารณาค่าความจริง สำหรับงานวิจัยนี้ได้กำหนดให้ ชนิดข้อมูลทั้งหมดที่ถูกกำหนดให้กับเงื่อนไขหนึ่งๆต้องมีความสัมพันธ์กันแบบซึบคลาส ดังตัวอย่างในรูปที่ 3.4



รูปที่ 3.4 ตัวอย่างเงื่อนไขที่พิจารณาค่าความจริงจากชนิดข้อมูลมากกว่า 1 ชนิด

จากตัวอย่างในรูปที่ 3.4 ในส่วน Shared Ontology แสดงเงื่อนไขที่เกี่ยวข้องกับการตรวจสอบชนิดบัตรเครดิต ได้แก่เงื่อนไข CreditCardHolder ซึ่งมีความหมายว่าผู้รับบริการเป็นผู้ถือบัตรเครดิตที่ได้รับการยอมรับภายในกลุ่มบริการ โดยเงื่อนไข CreditCardHolder นี้ขึ้นกับข้อมูลเข้า CreditCardType โดยหาก CreditCardType มีค่าเป็นชนิด CreditCard ซึ่งหมายถึงบัตรเครดิต Amex, VISA และ MasterCard แล้ว เงื่อนไข CreditCardHolder ก็จะเป็นจริง

ในกรณีที่ผู้ให้บริการต้องการสร้างเงื่อนไข AmexCardHolder ซึ่งเป็นเงื่อนไขที่เกี่ยวข้องกับการตรวจสอบชนิดบัตรเครดิตเช่นเดียวกับเงื่อนไข CreditCardHolder แต่จะพิจารณาบัตรเครดิตชนิด AmexCard ซึ่งหมายถึงบัตรเครดิต Amex แทนชนิด CreditCard ผู้ให้บริการสามารถประกาศเงื่อนไข AmexCardHolder ในส่วน Local Ontology ได้ 2 วิธีคือ

1. ประกาศเงื่อนไข AmexCardHolder ขึ้นใหม่โดยกำหนดให้เป็น subclass ของคลาส Condition เช่นเดียวกับตัวอย่างในรูปที่ 3.3
2. ผู้ให้บริการอาจมองว่าเงื่อนไข AmexCardHolder ถือเป็นเงื่อนไข CreditCardHolder ชนิดหนึ่ง เพียงแต่จะพิจารณาเฉพาะบัตรเครดิต Amex ซึ่งเป็นสมาชิกตัวหนึ่งของชนิด CreditCard ดังนั้นจึงประกาศเงื่อนไข AmexCardHolder โดยกำหนดให้เป็น subclass ของคลาส CreditCardHolder และกำหนดให้เงื่อนไข AmexCardHolder พิจารณาชนิดข้อมูล AmexCard

จากรูปที่ 3.4 ในส่วน Semantic Instance อินสแตนซ์ของเงื่อนไข AmexCardHolder ที่ถูกประกาศโดยวิธีที่ 2 มีความสัมพันธ์กับชนิดข้อมูล 2 ชนิดคือ CreditCard และ AmexCard ซึ่งถูกกำหนดโดยคลาส CreditCardHolder และคลาส AmexCardHolder ตามลำดับ สำหรับชนิดข้อมูลที่จะใช้ในการพิจารณาค่าความจริงคือชนิด AmexCard เนื่องจากถูกกำหนดที่หลังและเป็น subclass ของชนิด CreditCard

สำหรับการกำหนดเงื่อนไขให้กับ การดำเนินการ ข้อมูลออก และผลกระทบ ผ่านคุณสมบัติ hasPrecondition, outputCondition และ effectCondition ตามลำดับ สามารถกำหนดให้มีจำนวนเท่าใดก็ได้ ในกรณีที่มากกว่า 1 เงื่อนไข ผู้วิจัยกำหนดให้เงื่อนไขทั้งหมดต้องเป็นจริง เพื่อให้สามารถประมวลผลการดำเนินการ ให้ข้อมูลออกแบบมีเงื่อนไข และ ให้ผลกระทบแบบมีเงื่อนไข

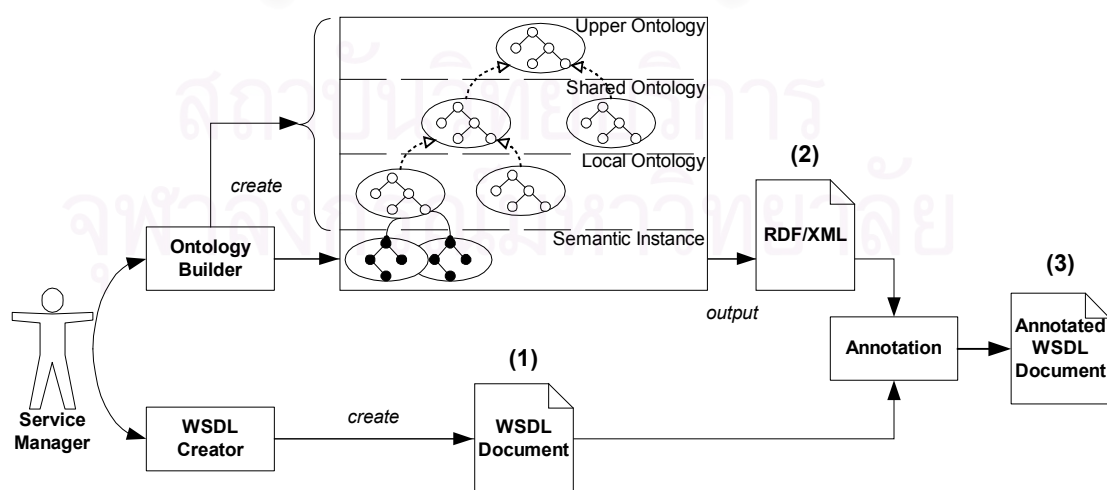
### 3.2 การแทรกข้อสนเทศเชิงพฤติกรรมให้กับเอกสารดับเบิลยูเอสดีแอล

ในการแทรกข้อสนเทศเชิงพฤติกรรมให้กับเอกสารดับเบิลยูเอสดีแอลเพื่ออธิบายเว็บเซอร์วิสประกอบด้วย แนวคิดในการแทรกข้อสนเทศเชิงพฤติกรรม และ ข้อกำหนดในการแทรกข้อสนเทศเชิงพฤติกรรม ซึ่งจะอธิบายในหัวข้อที่ 3.2.1 และ หัวข้อที่ 3.2.2 ตามลำดับ

#### 3.2.1 แนวคิดในการแทรกข้อสนเทศเชิงพฤติกรรม

จากงานวิจัย [5] ได้นำเสนอแนวคิดในการแทรกข้อสนเทศเชิงพฤติกรรมที่อยู่ในรูป อีลีเมนต์ และ แอททริบิวต์ ของภาษาเอ็กซ์เอ็มแอลให้กับเอกสารดับเบิลยูเอสดีแอล แต่วิธีการนี้ทำให้ไม่สะดวกทั้งในขั้นตอนการใส่คำอธิบายประกอบเพื่อโฆษณาบริการ และการดึง (Extraction) คำอธิบายประกอบ เพื่อใช้ในการค้นหาบริการ ส่วนบทความ [6] ได้นำเสนอการเปลี่ยนแปลงอีลีเมนต์ของเอกสารดับเบิลยูเอสดีแอลให้อยู่ในรูปแบบแบบอาร์ตีเฟค เพื่อให้สามารถแสดงโมเดลของเอกสารดับเบิลยูเอสดีแอลได้ในรูปแบบกราฟ โดยใช้เครื่องมือแบบเห็นภาพสำหรับอาร์ตีเฟค แต่ทว่าการแปลงเอกสารดับเบิลยูเอสดีแอลให้อยู่ในรูปแบบแบบอาร์ตีเฟคไม่ถือว่าถูกต้องสมบูรณ์ตามข้อกำหนด ทำให้ไม่สามารถนำเอกสารดับเบิลยูเอสดีแอลนี้ไปใช้งานได้ตามปกติ

งานวิจัยนี้จึงได้เสนอการผสานแนวความคิดจากงานวิจัย [5] และบทความ [6] โดยแทรกข้อสนเทศเชิงพฤติกรรมในรูปแบบแบบอาร์ตีเฟคเอ็กซ์เอ็มแอลให้กับเอกสารดับเบิลยูเอสดีแอล ซึ่งเอื้อต่อการเพิ่มอีลีเมนต์ที่เป็นไปตามข้อกำหนดของภาษาเอ็กซ์เอ็มแอล [3] ทั้งนี้เพื่อให้ง่ายต่อการแทรกและดึงข้อสนเทศเชิงพฤติกรรม อีกทั้งยังสามารถใช้ประโยชน์จากเครื่องมือแบบเห็นภาพสำหรับอาร์ตีเฟค ในการสร้างข้อสนเทศเชิงพฤติกรรมให้กับบริการอีกด้วย



รูปที่ 3.5 ขั้นตอนการแทรกข้อสนเทศเชิงพฤติกรรม

จากรูปที่ 3.5 ขั้นตอนการแทรกข้อสนเทศเชิงพฤติกรรมเริ่มต้นจาก เซอร์วิสเมเนเจอร์ (Service Manager) ใช้เครื่องมือซอฟต์แวร์ ยกตัวอย่างเช่นดับเบิลยูเอสดีแอลเอดิเตอร์ เพื่อสร้างเอกสารดับเบิลยูเอสดีแอลให้กับเว็บเซอร์วิส (ใน ส่วน (1)) และใช้ออนโทโลยีบิลเดอร์ (Ontology Builder) ยกตัวอย่างเช่นโพรเทเจ เพื่อสร้างข้อสนเทศเชิงพฤติกรรมที่อยู่ในรูปแบบอาร์ดีเอฟเอ็กซ์เอ็มแอล (ใน ส่วน (2)) หลังจากนั้นจึงนำข้อสนเทศเชิงพฤติกรรมไปแทรกให้เอกสารดับเบิลยูเอสดีแอล (ใน ส่วน (3))

### 3.2.2 ข้อกำหนดในการแทรกข้อสนเทศเชิงพฤติกรรม

ตามข้อกำหนดของเอกสารดับเบิลยูเอสดีแอล รุ่นที่ 1.1 (WSDL Specification version 1.1) [3] สนับสนุนการเพิ่มอิลีเมนต์ที่เป็นไปตามข้อกำหนดของภาษาเอ็กซ์เอ็มแอล ดังนี้

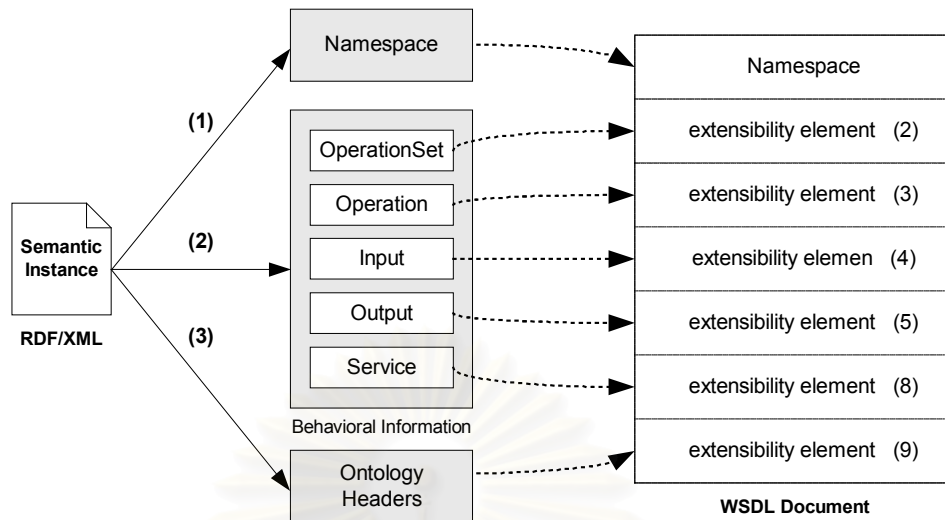
	<pre>&lt;wsdl:definitions name="nmtoken"? targetNamespace="uri"?&gt;   &lt;import namespace="uri" location="uri"/&gt;*   &lt;wsdl:documentation .... /&gt; ?</pre>	
(ก)	<pre>&lt;wsdl:types&gt; ?   &lt;wsdl:documentation .... /&gt;?   &lt;xsd:schema .... /&gt;*   &lt;-- extensibility element --&gt; * &lt;/wsdl:types&gt;</pre>	(1)
(ข)	<pre>&lt;wsdl:message name="nmtoken"&gt; *   &lt;wsdl:documentation .... /&gt;?   &lt;part name="nmtoken" element="qname"? type="qname"?/&gt; * &lt;/wsdl:message&gt;</pre>	
(ค)	<pre>&lt;wsdl:portType name="nmtoken"&gt;*   &lt;wsdl:documentation .... /&gt;?   &lt;wsdl:operation name="nmtoken"&gt;*   &lt;wsdl:documentation .... /&gt; ?   &lt;wsdl:input name="nmtoken"? message="qname"?&gt;?   &lt;wsdl:documentation .... /&gt; ?   &lt;/wsdl:input&gt;   &lt;wsdl:output name="nmtoken"? message="qname"?&gt;?   &lt;wsdl:documentation .... /&gt; ?   &lt;/wsdl:output&gt;   &lt;wsdl:fault name="nmtoken" message="qname"&gt; *   &lt;wsdl:documentation .... /&gt; ?   &lt;/wsdl:fault&gt;   &lt;/wsdl:operation&gt; &lt;/wsdl:portType&gt;</pre>	

รูปที่ 3.6 ข้อกำหนดในการเพิ่มอิลีเมนต์ของเอกสารดับเบิลยูเอสดีแอล รุ่นที่ 1.1

(ง)	<pre> &lt;wsdl:binding name="nmtoken" type="qname"&gt;*   &lt;wsdl:documentation .... /&gt;?   &lt;!-- extensibility element --&gt; *   &lt;wsdl:operation name="nmtoken"&gt;*     &lt;wsdl:documentation .... /&gt; ?     &lt;!-- extensibility element --&gt; *     &lt;wsdl:input&gt; ?       &lt;wsdl:documentation .... /&gt; ?       &lt;!-- extensibility element --&gt;     &lt;/wsdl:input&gt;     &lt;wsdl:output&gt; ?       &lt;wsdl:documentation .... /&gt; ?       &lt;!-- extensibility element --&gt; *     &lt;/wsdl:output&gt;     &lt;wsdl:fault name="nmtoken"&gt; *       &lt;wsdl:documentation .... /&gt; ?       &lt;!-- extensibility element --&gt; *     &lt;/wsdl:fault&gt;   &lt;/wsdl:operation&gt; &lt;/wsdl:binding&gt; </pre>	(2)
		(3)
		(4)
		(5)
		(6)
(จ)	<pre> &lt;wsdl:service name="nmtoken"&gt; *   &lt;wsdl:documentation .... /&gt;?   &lt;wsdl:port name="nmtoken" binding="qname"&gt; *     &lt;wsdl:documentation .... /&gt; ?     &lt;!-- extensibility element --&gt;   &lt;/wsdl:port&gt;   &lt;!-- extensibility element --&gt; &lt;/wsdl:service&gt;  &lt;!-- extensibility element --&gt; * </pre>	(7)
		(8)
		(9)
	</wsdl:definitions>	

รูปที่ 3.6 (ต่อ) ข้อกำหนดในการเพิ่มอีลีเมนต์ของเอกสารดับเบิลยูเอสดีแอล รุ่นที่ 1.1

จากข้อกำหนดในรูปที่ 3.6 ตำแหน่งที่สนับสนุนการเพิ่มอีลีเมนต์ถูกแสดงด้วยอีลีเมนต์ “<- extensibility element -->” ส่วนสัญลักษณ์ “?” หมายถึง 0 หรือ 1 สัญลักษณ์ “\*” หมายถึง 0 หรือมากกว่า 0 และ สัญลักษณ์ “+” หมายถึง 1 หรือมากกว่า 1 ในงานวิจัยนี้ได้กำหนดให้แทรกข้อสนเทศเชิงพฤติกรรมในตำแหน่งต่างๆของเอกสารดับเบิลยูเอสดีแอล ดังรูปที่ 3.7

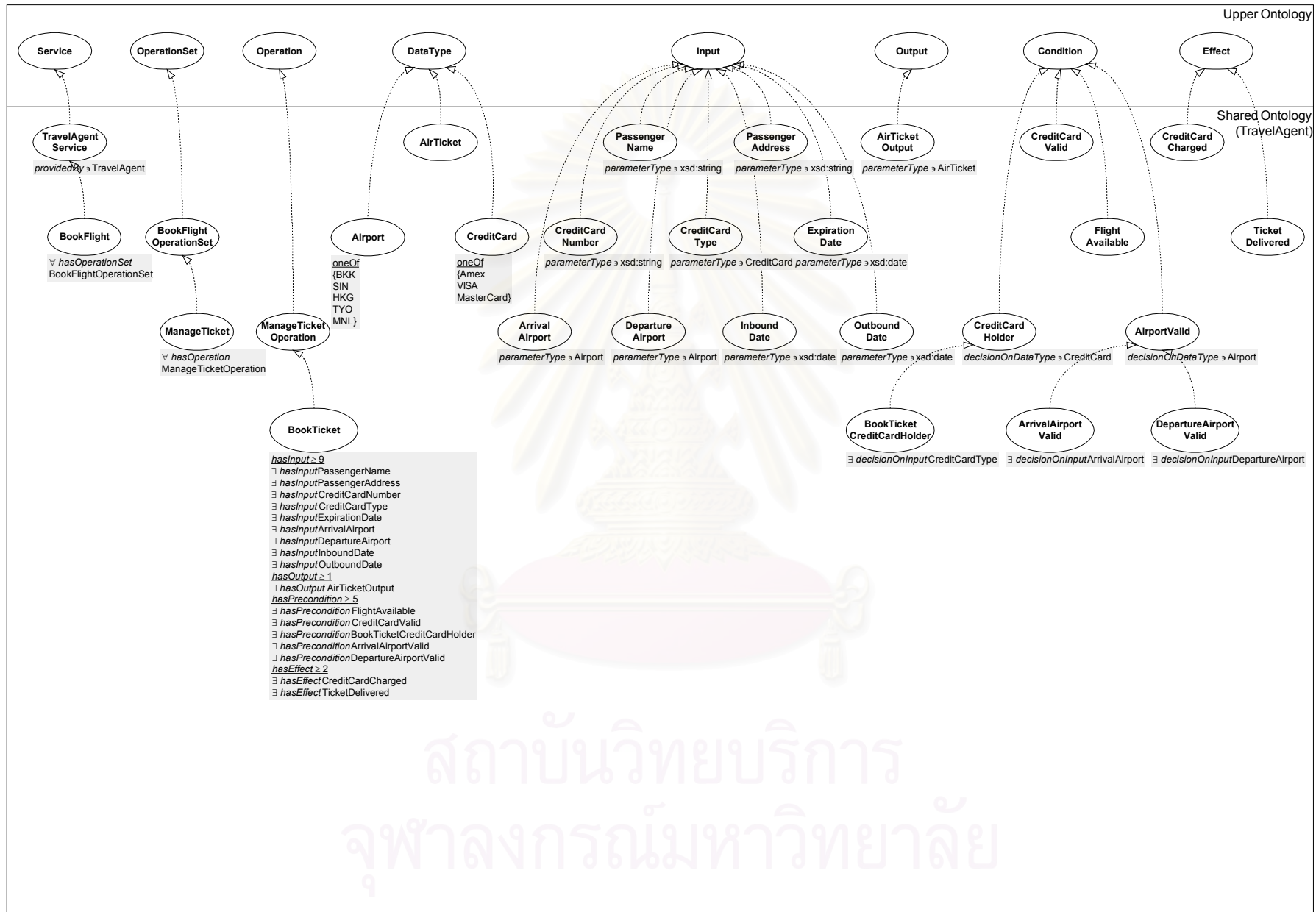


รูปที่ 3.7 ตำแหน่งในการแทรกข้อสนเทศเชิงพฤติกรรม

จากรูปที่ 3.7 กำหนดให้แบ่งเอกสารอาร์ดีเอฟเอ็กซ์เอ็มแอลซึ่งระบุพฤติกรรมของบริการ ออกเป็น 3 ส่วนคือ ส่วนกำหนดเนมสเปซ ส่วนข้อสนเทศเชิงพฤติกรรมของบริการ และส่วนออนโทโลยีเฮดเดอร์ (Ontology Headers) เนื่องจากข้อสนเทศเชิงพฤติกรรมที่สร้างจากออนโทโลยีบิลเดอร์ (ดังตัวอย่างในรูปที่ 3.14) จะรวมกันเป็นกลุ่มภายใต้ไอดีลิเมนต์สำหรับอธิบายอินสแตนซ์ของบริการ ทำให้ต้องแบ่งข้อสนเทศเชิงพฤติกรรมออกเป็นกลุ่มๆตามบริการ กลุ่มการดำเนินการ การดำเนินการ ข้อมูลเข้า และข้อมูลออก หลังจากนั้นจึงนำไปแทรกภายใต้ไอดีลิเมนต์ต่างๆของเอกสารดับเบิลยูเอสดีแอล เพื่อเชื่อมโยงไอดีลิเมนต์เหล่านั้นเข้ากับข้อสนเทศเชิงพฤติกรรมที่สัมพันธ์กัน

### 3.3 ตัวอย่างการแทรกข้อสนเทศเชิงพฤติกรรมให้กับเอกสารดับเบิลยูเอสดีแอล

ในวิทยานิพนธ์ฉบับนี้ ผู้วิจัยได้เสนอตัวอย่างเซิร์ฟเวอร์ออนโทโลยีของกลุ่มบริการตัวแทนสำนักงานท่องเที่ยว TravelAgent ซึ่งประกอบด้วยบริการจองตั๋วเครื่องบิน BookFlight กลุ่มการดำเนินการ ManageTicket และการดำเนินการ BookTicket ซึ่งข้อสนเทศเชิงพฤติกรรมของบริการ BookFlight นี้จะใช้เป็นตัวอย่างในการแทรกข้อสนเทศเชิงพฤติกรรมให้กับเอกสารดับเบิลยูเอสดีแอล สำหรับเซิร์ฟเวอร์ออนโทโลยีของกลุ่มบริการ TravelAgent จะแสดงในรูปที่ 3.8



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

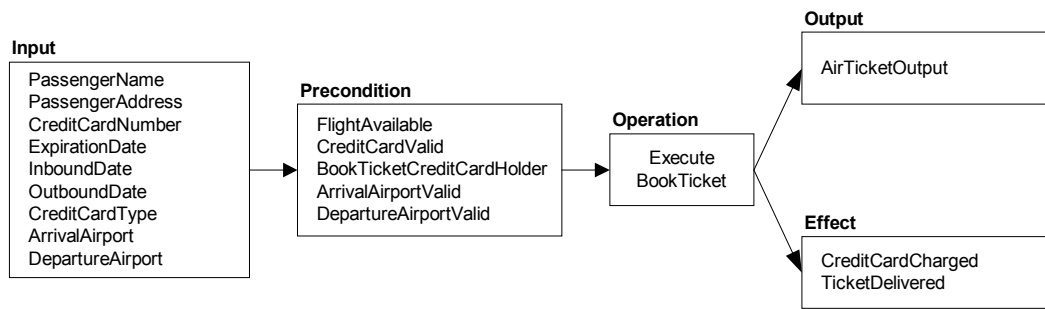
รูปที่ 3.8 แร็ดออนโทโลยีของกลุ่มบริการ TravelAgent

เนื่องจากแอปพลิเคชันออนไลน์กำหนดเพียงความสามารถที่ทุกๆเว็บไซต์มีส่วนร่วมกันโดยไม่ได้กำหนดพฤติกรรมเฉพาะเจาะจงสำหรับบริการในแต่ละกลุ่มบริการ ผู้จัดการออนไลน์ของแต่ละกลุ่มบริการสามารถกำหนดพฤติกรรมของบริการภายในกลุ่มได้ด้วยนิพจน์ภาษาอวาล์ผ่าน แคร็ดออนไลน์ จากตัวอย่างแคร็ดออนไลน์ของกลุ่มบริการ TravelAgent ในรูปที่ 3.8 ได้กำหนดพฤติกรรมของบริการจองตั๋วเครื่องบิน BookFlight โดยแบ่งออกได้เป็น 3 ส่วน คือ

1. ส่วนกำหนดโครงสร้างบริการซึ่งประกอบด้วยบริการ BookFlight กลุ่มการดำเนินการ ManageTicket และการดำเนินการ BookTicket คลาส TravelAgentService เป็นคลาสแม่ของทุกบริการภายในแคร็ดออนไลน์ กำหนดให้คุณสมบัติ providedBy ต้องมีค่าเป็น TravelAgent ส่วนคลาส BookFlightOperationSet และ คลาส ManageTicketOperation กำหนดให้เป็นคลาสแม่ของกลุ่มการดำเนินการ และการดำเนินการ ตามลำดับ
2. ส่วนกำหนดคลาสของชนิดข้อมูล ข้อมูลเข้า ข้อมูลออก เงื่อนไข และผลกระทบ โดยสร้างชั้นคลาสของ DataType, Input, Output, Condition และ Effect ตามลำดับ
3. ส่วนอธิบายพฤติกรรมของการดำเนินการที่กำหนดไว้ในส่วนที่ 1 โดยใช้คลาสที่กำหนดไว้ในส่วนที่ 2 ร่วมกับนิพจน์ภาษาอวาล์ ยกตัวอย่างเช่น อธิบายว่าการดำเนินการ BookTicket ต้องประกอบด้วยข้อมูลเข้า ข้อมูลออก เงื่อนไขก่อนการทำงาน และ ผลกระทบ ไต่บ้าง

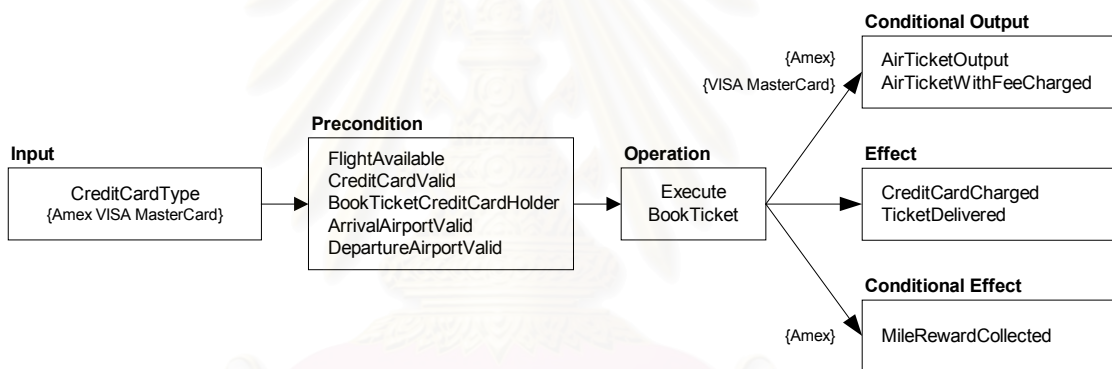
จากรูปที่ 3.8 แคร็ดออนไลน์กำหนดให้การดำเนินการ BookTicket ต้องประกอบด้วยข้อมูลเข้าอย่างน้อย 9 ข้อมูล ค่าของข้อมูลเข้าต้องเป็นจริงตามเงื่อนไขก่อนการทำงานทั้ง 5 เงื่อนไข เพื่อให้สามารถประมวลผลการดำเนินการได้ หลังจากประมวลผลแล้ว การดำเนินการ BookTicket ต้องให้ข้อมูลออกอย่างน้อย 1 ข้อมูลคือ AirTicketOutput และต้องให้ผลกระทบอย่างน้อย 2 ผลกระทบคือ CreditCardCharged และ TicketDelivered ซึ่งสามารถแสดงด้วยแผนภาพได้ดังรูปที่ 3.9





รูปที่ 3.9 พฤติกรรมของการดำเนินการ BookTicket

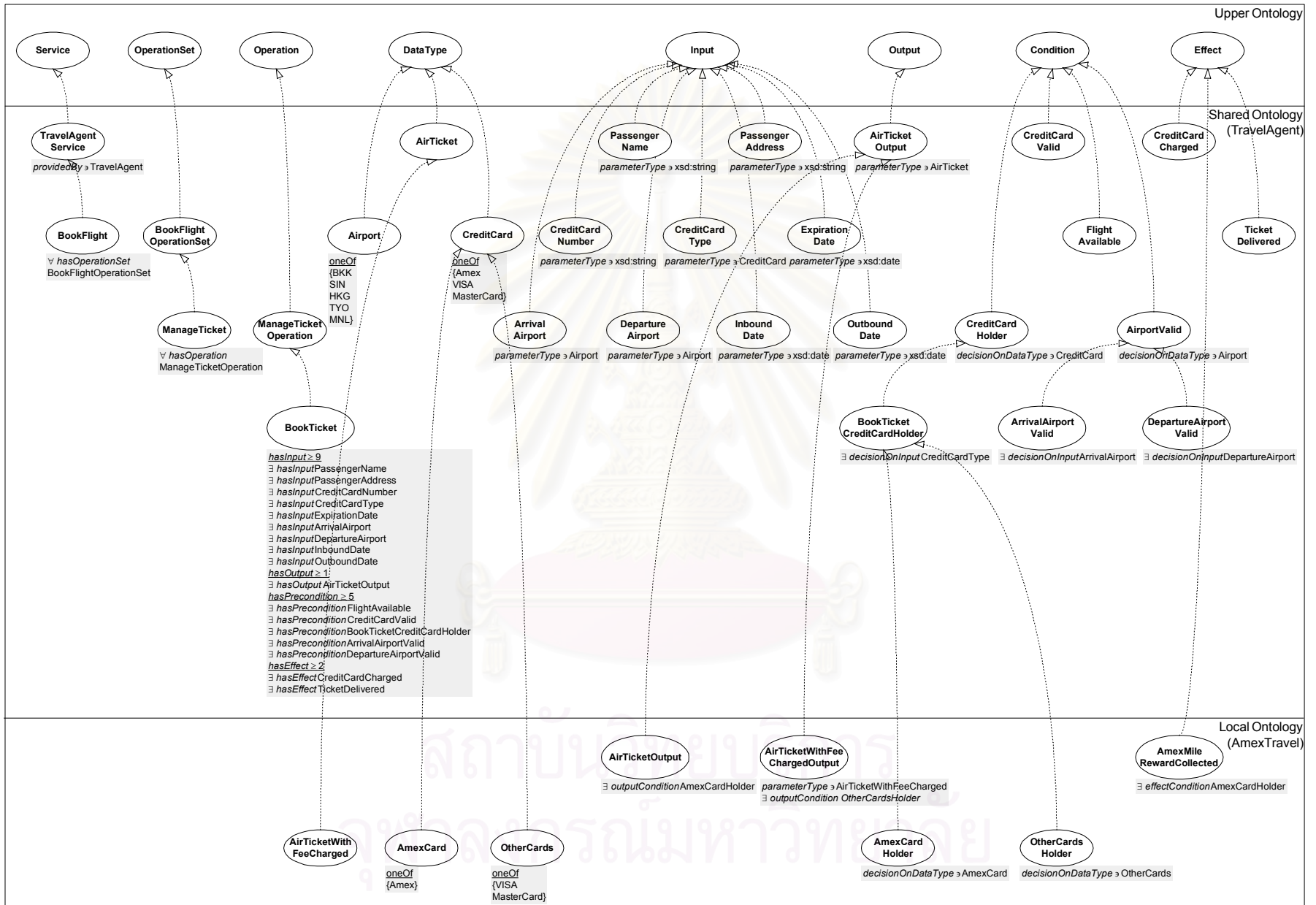
บริษัท AmexTravel เป็นผู้ให้บริการรายหนึ่งซึ่งสังกัดกลุ่มบริการ TravelAgent สนับสนุนบริการ BookFlight ที่ปรับมาจากแอร์ดอทคอมโทโลจี โดยกำหนดข้อมูลออกแบบมีเงื่อนไข และผลกระทบแบบมีเงื่อนไข ให้กับการดำเนินการ BookTicket ดังแผนภาพในรูปที่ 3.10



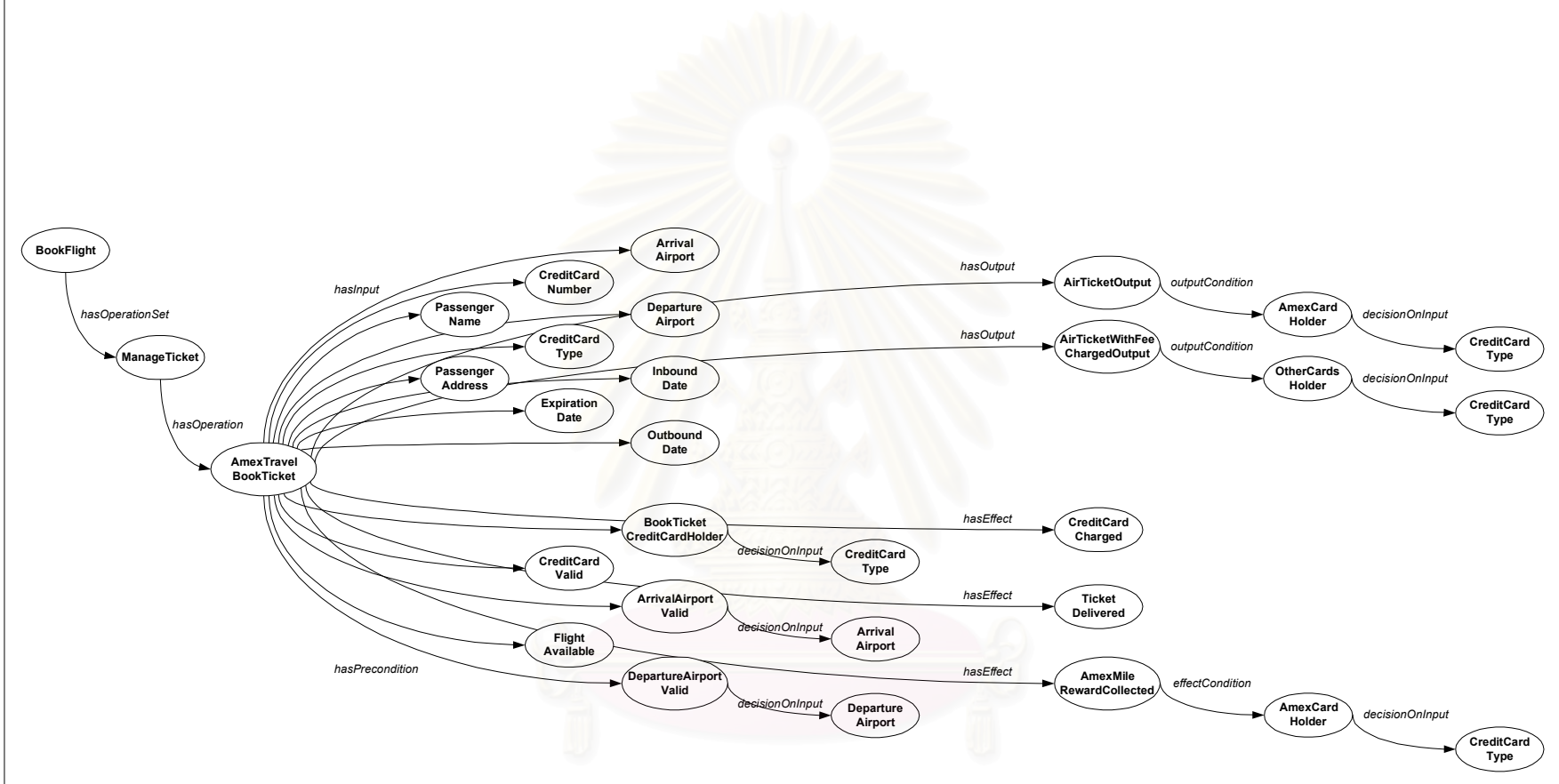
รูปที่ 3.10 พฤติกรรมของการดำเนินการ BookTicket ที่ปรับมาจากแอร์ดอทคอมโทโลจี

จากรูปที่ 3.10 การดำเนินการ BookTicket จะให้ข้อมูลออก AirTicketOutput ผลกระทบ MileRewardCollected เมื่อข้อมูลเข้า CreditCardType มีค่าเป็น Amex และจะให้ข้อมูลออก AirTicketWithFeeCharged เมื่อข้อมูลเข้า CreditCardType มีค่าเป็น VISA หรือ MasterCard

โลคัลดอทคอมโทโลจีของบริษัท AmexTravel จะแสดงในรูปที่ 3.11 ส่วนซีแมนติคอินสแตนซ์ และซีแมนติคอินสแตนซ์ที่ผ่านการอนุมาน ของบริการ BookFlight จะแสดงในรูปที่ 3.12 และ รูปที่ 3.13 ตามลำดับ

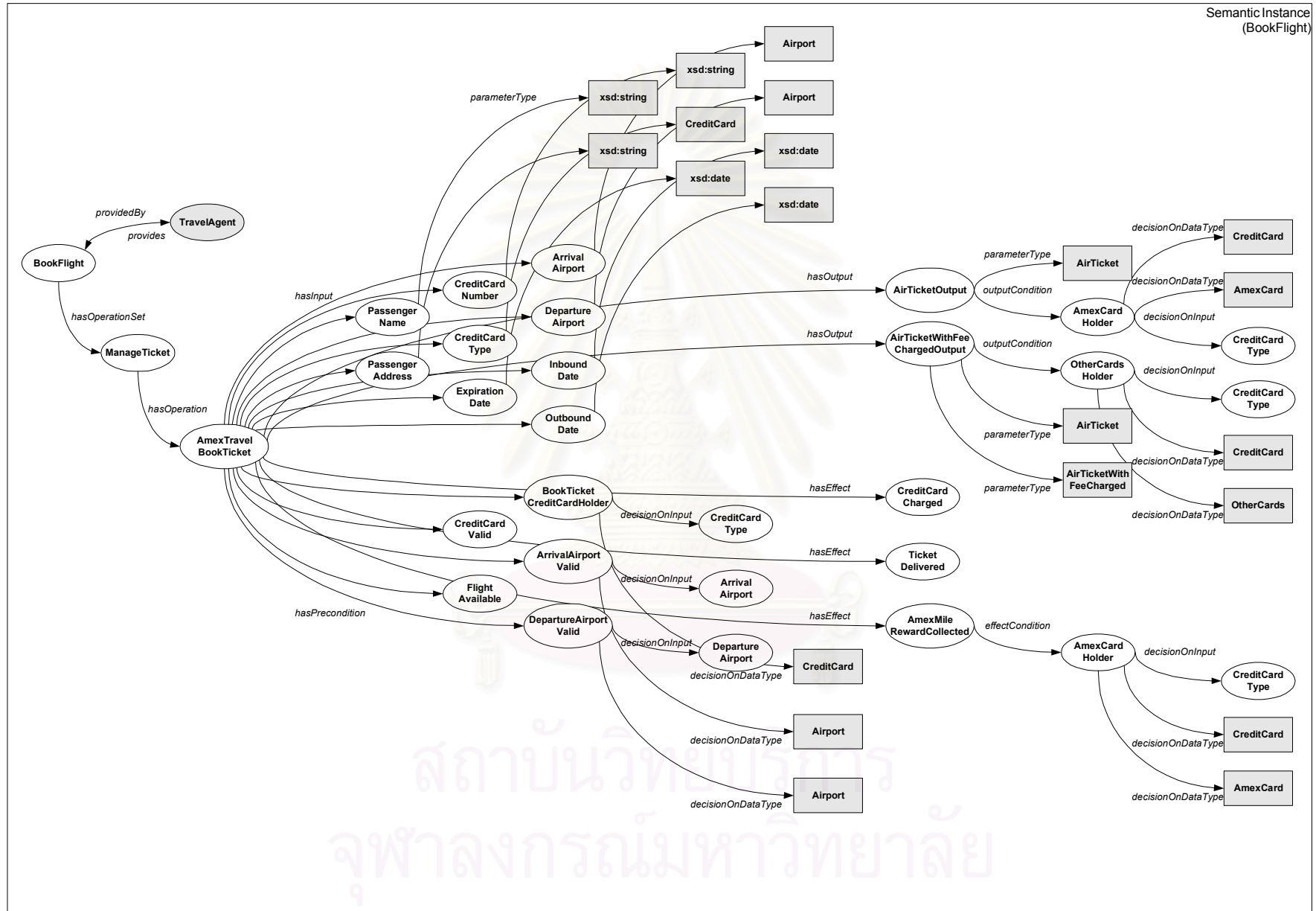


รูปที่ 3.11 โคล์ลอนโทโลยีของบริษัท AmexTravel



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

รูปที่ 3.12 ซีแมนติคอินสแตนซ์ของบริการ BookFlight



รูปที่ 3.13 ซีแมนติคอินสแตนซ์ของบริการ BookFlight ที่ผ่านการอนุมาน

ตัวอย่างข้อสนเทศเชิงพหุติกรรมสำหรับบริการ BookFlight ของบริษัท AmexTravel ในรูปที่ 3.12 สามารถเขียนให้อยู่ในรูปแบบอาร์ดีเอฟเอ็กซ์เอ็มแอลได้ดังรูปที่ 3.14 และตัวอย่างเอกสารดับเบิลยูเอสดีแอลก่อนแทรกข้อสนเทศเชิงพหุติกรรมได้แสดงในรูปที่ 3.15 ส่วนตัวอย่างเอกสารดับเบิลยูเอสดีแอลที่ผูกแทรกข้อสนเทศเชิงพหุติกรรมแล้วจะแสดงในรูปที่ 3.16

```

<?xml version="1.0" encoding="ISO-8859-1" ?>

<!-- //////////////////////////////////AmexTravelBookTicket Semantic Instance
-->
<rdf:RDF
  xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:up="http://www.Ontology.org/UpperOnt/UpperOnt.owl#"
  xmlns:sh="http://www.TravelAgent.com/SharedOnt/TravelAgent.owl#"
  xmlns:lo="http://www.AmexTravel.com/LocalOnt/AmexTravel.owl#"
  xmlns="http://www.AmexTravel.com/SemanticInst/AmexTravelBookTicket.owl#"
  xml:base="http://www.AmexTravel.com/SemanticInst/AmexTravelBookTicket.owl">

  <!-- //////////////////////////////////owl:Ontology
  -->
  <owl:Ontology rdf:about="">
    <owl:versionInfo>27/6/2547, 21:06:57</owl:versionInfo>
    <rdfs:comment>AmexTravelBookTicket Semantic Instance.</rdfs:comment>
    <owl:imports>
      <owl:Ontology rdf:about="http://www.Ontology.org/UpperOnt/UpperOnt.owl" />
    </owl:imports>
    <owl:imports>
      <owl:Ontology rdf:about="http://www.TravelAgent.com/SharedOnt/TravelAgent.owl" />
    </owl:imports>
    <owl:imports>
      <owl:Ontology rdf:about="http://www.AmexTravel.com/LocalOnt/AmexTravel.owl" />
    </owl:imports>
  </owl:Ontology>

  <!-- //////////////////////////////////Semantic Instance
  -->

  <!-- //////////////////////////////////Service
  -->
  <sh:BookFlight rdf:ID="BookFlight">
    <rdfs:comment>BookFlight Service.</rdfs:comment>
    <up:hasOperationSet>
      <!-- //////////////////////////////////OperationSet
      -->
      <sh:ManageTicket rdf:ID="ManageTicket">
        <rdfs:comment>ManageTicket OperationSet.</rdfs:comment>

```

รูปที่ 3.14 ตัวอย่างข้อสนเทศเชิงพหุติกรรมในรูปแบบอาร์ดีเอฟเอ็กซ์เอ็มแอลสำหรับบริการ

BookFlight

```

<up:hasOperation>
  <!-- ////////////////Operation
  -->
  <sh:BookTicket rdf:ID="AmexTravelBookTicket">
    <rdfs:comment>AmexTravelBookTicket Operation.</rdfs:comment>
    <!-- ////////////////hasInput
    -->
    <up:hasInput>
      <sh:PassengerName rdf:ID="PassengerName" />
    </up:hasInput>
    <up:hasInput>
      <sh:PassengerAddress rdf:ID="PassengerAddress" />
    </up:hasInput>
    <up:hasInput>
      <sh:CreditCardNumber rdf:ID="CreditCardNumber" />
    </up:hasInput>
    <up:hasInput>
      <sh:CreditCardType rdf:ID="CreditCardType" />
    </up:hasInput>
    <up:hasInput>
      <sh:ExpirationDate rdf:ID="ExpirationDate" />
    </up:hasInput>
    <up:hasInput>
      <sh:ArrivalAirport rdf:ID="ArrivalAirport" />
    </up:hasInput>
    <up:hasInput>
      <sh:DepartureAirport rdf:ID="DepartureAirport" />
    </up:hasInput>
    <up:hasInput>
      <sh:InboundDate rdf:ID="InboundDate" />
    </up:hasInput>
    <up:hasInput>
      <sh:OutboundDate rdf:ID="OutboundDate" />
    </up:hasInput>

    <!-- ////////////////hasPrecondition
    -->
    <up:hasPrecondition>
      <sh:CreditCardValid rdf:ID="CreditCardValid" />
    </up:hasPrecondition>
    <up:hasPrecondition>
      <sh:FlightAvailable rdf:ID="FlightAvailable" />
    </up:hasPrecondition>
    <up:hasPrecondition>
      <sh:BookTicketCreditCardHolder rdf:ID="BookTicketCreditCardHolder">
        <up:decisionOnInput rdf:resource="#CreditCardType" />
      </sh:BookTicketCreditCardHolder>
    </up:hasPrecondition>
    <up:hasPrecondition>
      <sh:ArrivalAirportValid rdf:ID="ArrivalAirportValid">
        <up:decisionOnInput rdf:resource="#ArrivalAirport" />
      </sh:ArrivalAirportValid>

```

รูปที่ 3.14 (ต่อ) ตัวอย่างข้อสนเทศเชิงพฤติกรรมในรูปแบบอาร์ดีเอฟเอ็กซ์เอ็มแอลสำหรับบริการ

BookFlight

```

</up:hasPrecondition>
<up:hasPrecondition>
  <sh:DepartureAirportValid rdf:ID="DepartureAirportValid">
    <up:decisionOnInput rdf:resource="#DepartureAirport" />
  </sh:DepartureAirportValid>
</up:hasPrecondition>

<!-- ////////////////hasOutput
-->
<up:hasOutput>
  <lo:AirTicketOutput rdf:ID="AirTicketOutput">
    <up:outputCondition>
      <lo:AmexCardHolder rdf:ID="AmexCardHolder">
        <up:decisionOnInput rdf:resource="#CreditCardType" />
      </lo:AmexCardHolder>
    </up:outputCondition>
  </lo:AirTicketOutput>
</up:hasOutput>
<up:hasOutput>
  <lo:AirTicketWithFeeChargedOutput rdf:ID="AirTicketWithFeeChargedOutput">
    <up:outputCondition>
      <lo:OtherCardsHolder rdf:ID="OtherCardsHolder">
        <up:decisionOnInput rdf:resource="#CreditCardType" />
      </lo:OtherCardsHolder>
    </up:outputCondition>
  </lo:AirTicketWithFeeChargedOutput>
</up:hasOutput>

<!-- ////////////////hasEffect
-->
<up:hasEffect>
  <sh:CreditCardCharged rdf:ID="CreditCardCharged" />
</up:hasEffect>
<up:hasEffect>
  <sh:TicketDelivered rdf:ID="TicketDelivered" />
</up:hasEffect>
<up:hasEffect>
  <lo:AmexMileRewardCollected rdf:ID="AmexMileRewardCollected">
    <up:effectCondition>
      <lo:AmexCardHolder rdf:ID="AmexCardHolder">
        <up:decisionOnInput rdf:resource="#CreditCardType" />
      </lo:AmexCardHolder>
    </up:effectCondition>
  </lo:AmexMileRewardCollected>
</up:hasEffect>
</sh:BookTicket>
</up:hasOperation>
</sh:ManageTicket>
</up:hasOperationSet>
</sh:BookFlight>

</rdf:RDF>

```

รูปที่ 3.14 (ต่อ) ตัวอย่างข้อสนเทศเชิงพฤติกรรมในรูปแบบอาร์ดีเอฟเอ็กซ์เอ็มแอลสำหรับบริการ

	<pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;definitions   name="BookFlight"   targetNamespace="http://www.AmexTravel.com/wsd/BookFlight.wsd/"   xmlns="http://schemas.xmlsoap.org/wsd/"   xmlns:soap="http://schemas.xmlsoap.org/wsd/soap/"   xmlns:tns="http://www.AmexTravel.com/wsd/BookFlight.wsd/"   xmlns:xsd="http://www.w3.org/2001/XMLSchema"&gt;   &lt;!--Add Namespace--&gt; </pre>
(ก)	
(ข)	<pre> &lt;message name="AmexTravelBookTicketOutput"&gt; &lt;/message&gt; &lt;message name="AmexTravelBookTicketInput"&gt; &lt;/message&gt; </pre>
(ค)	<pre> &lt;portType name="ManageTicketPortType"&gt;   &lt;operation name="AmexTravelBookTicket"&gt;     &lt;input message="tns:AmexTravelBookTicketInput"/&gt;     &lt;output message="tns:AmexTravelBookTicketOutput"/&gt;   &lt;/operation&gt; &lt;/portType&gt; </pre>
(ง)	<pre> &lt;binding name="ManageTicketBinding" type="tns:ManageTicketPortType"&gt;   &lt;soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/&gt;   &lt;!--Annotate OperationSet--&gt;   &lt;operation name="AmexTravelBookTicket"&gt;     &lt;soap:operation       soapAction="capeconnect::ManageTicketPortType#AmexTravelBookTicket"/&gt;     &lt;!--Annotate Operation--&gt;     &lt;input&gt;       &lt;soap:body use="literal"/&gt;     &lt;!--Annotate Input--&gt;     &lt;/input&gt;     &lt;output&gt;       &lt;soap:body use="literal"/&gt;     &lt;!--Annotate Output--&gt;     &lt;/output&gt;   &lt;/operation&gt; &lt;/binding&gt; </pre>
(จ)	<pre> &lt;service name="BookFlightService"&gt;   &lt;port binding="tns:ManageTicketBinding" name="ManageTicketPort"&gt;     &lt;soap:address location="http://www.AmexTravel.com/Service/BookFlightService"/&gt;   &lt;/port&gt;   &lt;!--Annotate Service--&gt; &lt;/service&gt; &lt;!--Annotate Ontology Headers--&gt; &lt;/definitions&gt; </pre>

รูปที่ 3.15 ตัวอย่างเอกสารดับเบิลยูเอสดีแอลสำหรับบริการ BookFlight



```

<?xml version="1.0" encoding="UTF-8"?>
<definitions
  name="BookFlight"
  targetNamespace="http://www.AmexTravel.com/wsdl/BookFlightAnnotated.wsdl"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://www.AmexTravel.com/wsdl/BookFlightAnnotated.wsdl"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:up="http://www.Ontology.org/UpperOnt/UpperOnt.owl#"
  xmlns:sh="http://www.TravelAgent.com/SharedOnt/TravelAgent.owl#"
  xmlns:lo="http://www.AmexTravel.com/LocalOnt/AmexTravel.owl#"
  xml:base="http://www.AmexTravel.com/wsdl/BookFlightAnnotated.wsdl">
  <message name="AmexTravelBookTicketOutput"> </message>
  <message name="AmexTravelBookTicketInput"> </message>
  <portType name="ManageTicketPortType">
    <operation name="AmexTravelBookTicket">
      <input message="tns:AmexTravelBookTicketInput"/>
      <output message="tns:AmexTravelBookTicketOutput"/>
    </operation>
  </portType>
  <binding name="ManageTicketBinding" type="tns:ManageTicketPortType">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <!-- extensibility element -->
    <rdf:RDF>
      <!-- ////////////////OperationSet
      -->
      <sh:ManageTicket rdf:ID="ManageTicket">
        <rdfs:comment>ManageTicket OperationSet.</rdfs:comment>
        <up:hasOperation>
          <sh:BookTicket rdf:ID="AmexTravelBookTicket" />
        </up:hasOperation>
      </sh:ManageTicket>
    </rdf:RDF>
    <!-- end extensibility element -->
    <operation name="AmexTravelBookTicket">
      <soap:operation
        soapAction="capeconnect::ManageTicketPortType#AmexTravelBookTicket"/>
      <!-- extensibility element -->
      <rdf:RDF>
        <!-- ////////////////Operation
        -->
        <sh:BookTicket rdf:ID="AmexTravelBookTicket">
          <rdfs:comment>AmexTravelBookTicket Operation.</rdfs:comment>
          <!-- ////////////////hasInput
          -->
          <up:hasInput>
            <sh:PassengerName rdf:ID="PassengerName" />
          </up:hasInput>
          <up:hasInput>

```

รูปที่ 3.16 ตัวอย่างเอกสารดับเบิลยูเอชดีแอลสำหรับบริการ BookFlight ที่ถูกแทรกข้อเสนอพิเศษเชิงพาณิชย์

กรรม

```

    <sh:PassengerAddress rdf:ID="PassengerAddress" />
  </up:hasInput>
  <up:hasInput>
    <sh:CreditCardNumber rdf:ID="CreditCardNumber" />
  </up:hasInput>
  <up:hasInput>
    <sh:CreditCardType rdf:ID="CreditCardType" />
  </up:hasInput>
  <up:hasInput>
    <sh:ExpirationDate rdf:ID="ExpirationDate" />
  </up:hasInput>
  <up:hasInput>
    <sh:ArrivalAirport rdf:ID="ArrivalAirport" />
  </up:hasInput>
  <up:hasInput>
    <sh:DepartureAirport rdf:ID="DepartureAirport" />
  </up:hasInput>
  <up:hasInput>
    <sh:InboundDate rdf:ID="InboundDate" />
  </up:hasInput>
  <up:hasInput>
    <sh:OutboundDate rdf:ID="OutboundDate" />
  </up:hasInput>

  <!-- ////////////////hasPrecondition
  -->
  <up:hasPrecondition>
    <sh:CreditCardValid rdf:ID="CreditCardValid" />
  </up:hasPrecondition>
  <up:hasPrecondition>
    <sh:FlightAvailable rdf:ID="FlightAvailable" />
  </up:hasPrecondition>
  <up:hasPrecondition>
    <sh:BookTicketCreditCardHolder rdf:ID="BookTicketCreditCardHolder">
      <up:decisionOnInput rdf:resource="#CreditCardType" />
    </sh:BookTicketCreditCardHolder>
  </up:hasPrecondition>
  <up:hasPrecondition>
    <sh:ArrivalAirportValid rdf:ID="ArrivalAirportValid">
      <up:decisionOnInput rdf:resource="#ArrivalAirport" />
    </sh:ArrivalAirportValid>
  </up:hasPrecondition>
  <up:hasPrecondition>
    <sh:DepartureAirportValid rdf:ID="DepartureAirportValid">
      <up:decisionOnInput rdf:resource="#DepartureAirport" />
    </sh:DepartureAirportValid>
  </up:hasPrecondition>

  <!-- ////////////////hasOutput
  -->
  <up:hasOutput>
    <lo:AirTicketOutput rdf:ID="AirTicketOutput">

```

รูปที่ 3.16 (ต่อ) ตัวอย่างเอกสารดับเบิลยูเอสดีแอลสำหรับบริการ BookFlight ที่ถูกแทรกข้อสนเทศเชิง  
พฤติกรรม

```

    <up:outputCondition>
      <lo:AmexCardHolder rdf:ID="AmexCardHolder">
        <up:decisionOnInput rdf:resource="#CreditCardType" />
      </lo:AmexCardHolder>
    </up:outputCondition>
  </lo:AirTicketOutput>
</up:hasOutput>
<up:hasOutput>
  <lo:AirTicketWithFeeChargedOutput rdf:ID="AirTicketWithFeeChargedOutput">
    <up:outputCondition>
      <lo:OtherCardsHolder rdf:ID="OtherCardsHolder">
        <up:decisionOnInput rdf:resource="#CreditCardType" />
      </lo:OtherCardsHolder>
    </up:outputCondition>
  </lo:AirTicketWithFeeChargedOutput>
</up:hasOutput>

<!-- ////////////////hasEffect
-->
<up:hasEffect>
  <sh:CreditCardCharged rdf:ID="CreditCardCharged" />
</up:hasEffect>
<up:hasEffect>
  <sh:TicketDelivered rdf:ID="TicketDelivered" />
</up:hasEffect>
<up:hasEffect>
  <lo:AmexMileRewardCollected rdf:ID="AmexMileRewardCollected">
    <up:effectCondition>
      <lo:AmexCardHolder rdf:ID="AmexCardHolder">
        <up:decisionOnInput rdf:resource="#CreditCardType" />
      </lo:AmexCardHolder>
    </up:effectCondition>
  </lo:AmexMileRewardCollected>
</up:hasEffect>
</sh:BookTicket>
</rdf:RDF>
<!-- end extensibility element -->
<input>
  <soap:body use="literal"/>
</input>
<output>
  <soap:body use="literal"/>
</output>
</operation>
</binding>
<service name="BookFlightService">
  <port binding="tns:ManageTicketBinding" name="ManageTicketPort">
    <soap:address location="http://www.AmexTravel.com/Service/BookFlightService"/>
  </port>
  <!-- extensibility element -->
</service>
</rdf:RDF>
<!-- ////////////////Service

```

รูปที่ 3.16 (ต่อ) ตัวอย่างเอกสารดับเบิลยูเอชดีแอลสำหรับบริการ BookFlight ที่ถูกแทรกข้อสนเทศเชิงพฤติกรรม

```

-->
<sh:BookFlight rdf:ID="BookFlight">
  <rdfs:comment>BookFlight Service.</rdfs:comment>
  <up:hasOperationSet>
    <sh:ManageTicket rdf:ID="ManageTicket" />
  </up:hasOperationSet>
</sh:BookFlight>
</rdf:RDF>
<!-- end extensibility element -->
</service>

<!-- extensibility element -->
<rdf:RDF>
<!-- //owl:Ontology
-->
<owl:Ontology rdf:about="">
  <owl:versionInfo>27/6/2547, 21:06:57</owl:versionInfo>
  <rdfs:comment>AmexTravelBookTicket Semantic Instance.</rdfs:comment>
  <owl:imports>
    <owl:Ontology rdf:about="http://www.Ontology.org/UpperOnt/UpperOnt.owl" />
  </owl:imports>
  <owl:imports>
    <owl:Ontology rdf:about="http://www.TravelAgent.com/SharedOnt/TravelAgent.owl" />
  </owl:imports>
  <owl:imports>
    <owl:Ontology rdf:about="http://www.AmexTravel.com/LocalOnt/AmexTravel.owl" />
  </owl:imports>
</owl:Ontology>
</rdf:RDF>
<!-- end extensibility element -->
</definitions>

```

### Ontology Headers

รูปที่ 3.16 (ต่อ) ตัวอย่างเอกสารดับเบิลยูเอชดีแอลสำหรับบริการ BookFlight ที่ถูกแทรกข้อสนเทศเชิง  
พฤติกรรม

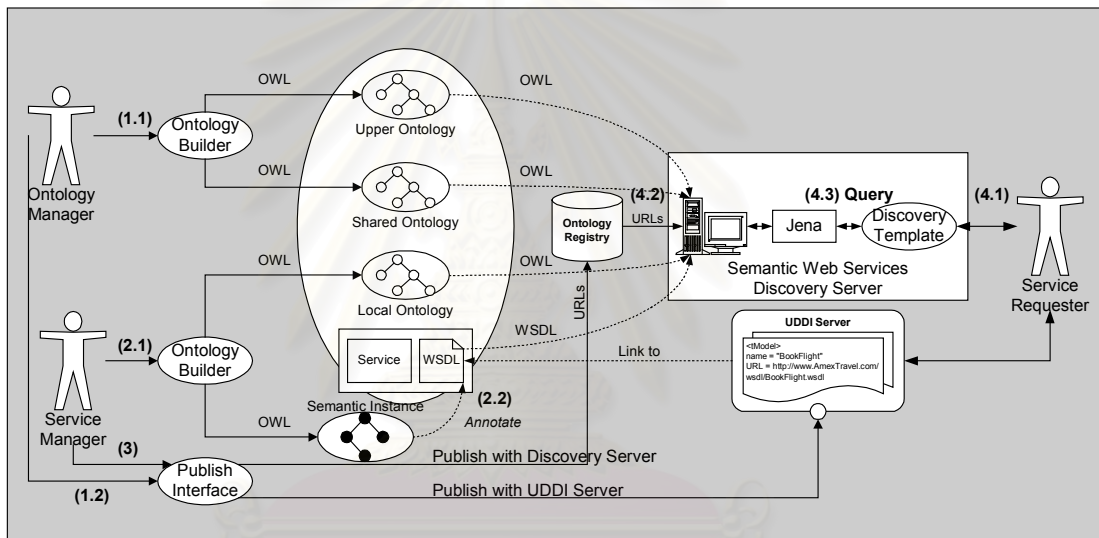
สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## บทที่ 4

### เฟรมเวิร์คสำหรับการโฆษณาและการค้นหาบริการเว็บเซอร์วิส

ในบทนี้จะกล่าวถึงเฟรมเวิร์คสำหรับการโฆษณาและการค้นหาบริการเว็บเซอร์วิสจากเอกสารดับเบิลยูเอสดีแอลที่ได้ผ่านการแทรกข้อสนเทศเชิงพฤติกรรม โดยการออกแบบเฟรมเวิร์คจะแสดงในหัวข้อที่ 4.1 ส่วนต้นแบบโปรแกรมประยุกต์สำหรับการโฆษณาและการค้นหาบริการเว็บเซอร์วิสจะแสดงในหัวข้อที่ 4.2

#### 4.1 การออกแบบเฟรมเวิร์คสำหรับการโฆษณาและการค้นหาบริการเว็บเซอร์วิส



รูปที่ 4.1 เฟรมเวิร์คสำหรับการโฆษณาและการค้นหาบริการเว็บเซอร์วิส

จากรูปที่ 4.1 การทำงานภายในเฟรมเวิร์คมีรายละเอียดดังนี้

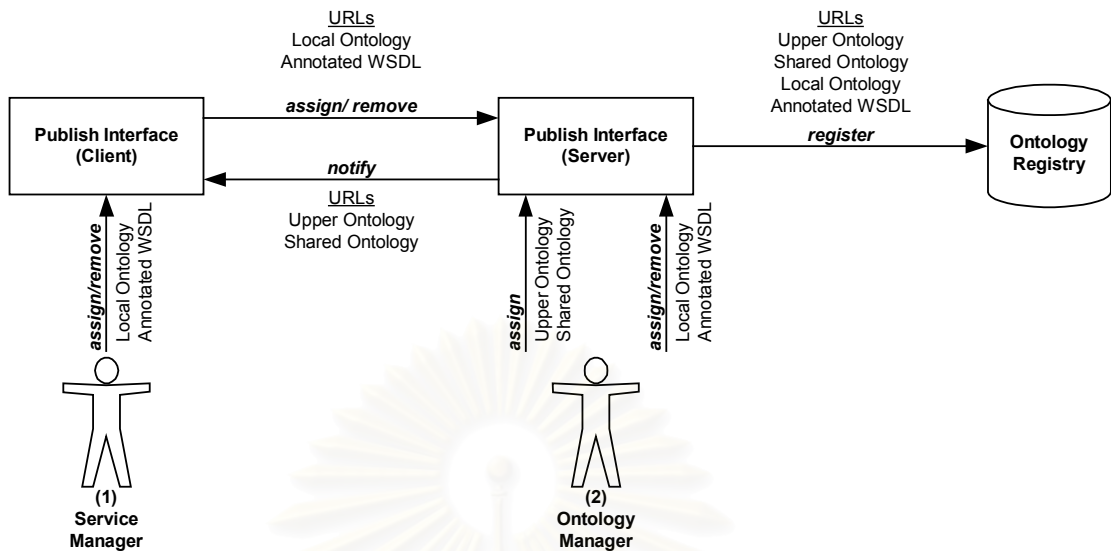
- (1) ออนโทโลยีเมเนเจอร์ (Ontology Manager) ใช้ออนโทโลยีบิลเดอร์เช่นพรอทเจเพื่อสร้างอับเปอร์ออนโทโลยี และแชร์ดออนโทโลยี ในรูปแบบภาษาอวาล์ (1.1) หลังจากนั้นจึงลงทะเบียนยูอาร์แอลของอับเปอร์ออนโทโลยี และยูอาร์แอลของแชร์ดออนโทโลยี ไปยังส่วนลงทะเบียนออนโทโลยี (Ontology Registry) ผ่านส่วนต่อประสานสำหรับโฆษณาบริการ (Publish Interface) (1.2)
- (2) เซอร์วิสเมเนเจอร์ของผู้ให้บริการแต่ละรายใช้ออนโทโลยีบิลเดอร์เพื่อสร้างโลคัลออนโทโลยี และข้อสนเทศเชิงพฤติกรรม ในรูปแบบภาษาอวาล์ (2.1) หลังจากนั้นจึงนำข้อสนเทศเชิงพฤติกรรมไปแทรกให้กับเอกสารดับเบิลยูเอสดีแอล (2.2) เพื่อโฆษณาเว็บเซอร์วิสที่ผู้ให้บริการสนับสนุน

- (3) เซอร์วิสเมเนเจอร์ใช้ส่วนต่อประสานสำหรับโฆษณาบริการเพื่อโฆษณาเว็บเซอร์วิส โดยลง ทะเบียนยูอาร์แอลของโลกออนไลน์โทโลจี และยูอาร์แอลของเอกสารดับเบิลยูเอสดีแอลที่ ผ่านการแทรกข้อสนเทศเชิงพฤติกรรม ไปยังส่วนลงทะเบียนออนไลน์โทโลจี นอกจากนี้แล้ว เซอร์วิสเมเนเจอร์อาจลงทะเบียนข้อมูลอื่นๆของเว็บเซอร์วิส ยกตัวอย่างเช่นคีย์อ้างอิง (Reference Key) ของบิสิเนสเซอร์วิส และบิสิเนสเอนทิตี ในยูดีดีไอเซิร์ฟเวอร์ (UDDI Server) เพื่อให้ผู้รับบริการสามารถเข้าถึงข้อมูลที่ถูกลงทะเบียนไว้ในยูดีดีไอเซิร์ฟเวอร์ได้
- (4) ผู้รับบริการ (Service Requester) ค้นหาบริการผ่านเซิร์ฟเวอร์ค้นหาเว็บเซอร์วิสเชิงความหมาย (Semantic Web Services Discovery Server) โดยกำหนดข้อบังคับเชิงพฤติกรรม ผ่านแผ่นแบบการค้นหา (Discovery Template) (4.1) ซึ่งจะคัดเลือกบริการจากทุกๆ บริการที่ได้ลงทะเบียนไว้ในส่วนลงทะเบียนออนไลน์โทโลจี (4.2) โดยใช้จ็วนาเอพีไอเพื่อนำ เอกสารอาร์วล์ของแอปเปอรชั่นโทโลจี แคร็ดออนไลน์โทโลจี และโลกออนไลน์โทโลจี รวมถึงข้อ เสนอแนะเชิงพฤติกรรมในเอกสารดับเบิลยูเอสดีแอลของผู้ให้บริการรายต่างๆ มาใช้ร่วมกัน เพื่ออนุมานหาข้อเท็จจริง หลังจากนั้นจึงสอบถามข้อสนเทศเชิงพฤติกรรมที่ผ่านการ อนุมาน ด้วยข้อคำถามที่สอดคล้องกับข้อบังคับเชิงพฤติกรรมที่ผู้รับบริการได้กำหนด (4.3) แล้วจึงรายงานบริการที่คัดเลือกได้กลับไปยังผู้รับบริการ

#### 4.1.1 การออกแบบส่วนต่อประสานสำหรับโฆษณาบริการ

สำหรับการโฆษณาบริการ เนื่องจากสิทธิในการลงทะเบียนออนไลน์โทโลจีของออนไลน์โทโลจีเมเนเจอร์และเซอร์วิสเมเนเจอร์มีความแตกต่างกัน (จากรูปที่ 4.1 ในส่วน (1) และในส่วน (3)) ผู้วิจัยจึง ได้แบ่งการทำงานของส่วนต่อประสานสำหรับโฆษณาบริการออกเป็น 2 ส่วนคือ สำหรับออนไลน์ โทโลจีเมเนเจอร์ และสำหรับเซอร์วิสเมเนเจอร์ ดังรูปที่ 4.2

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

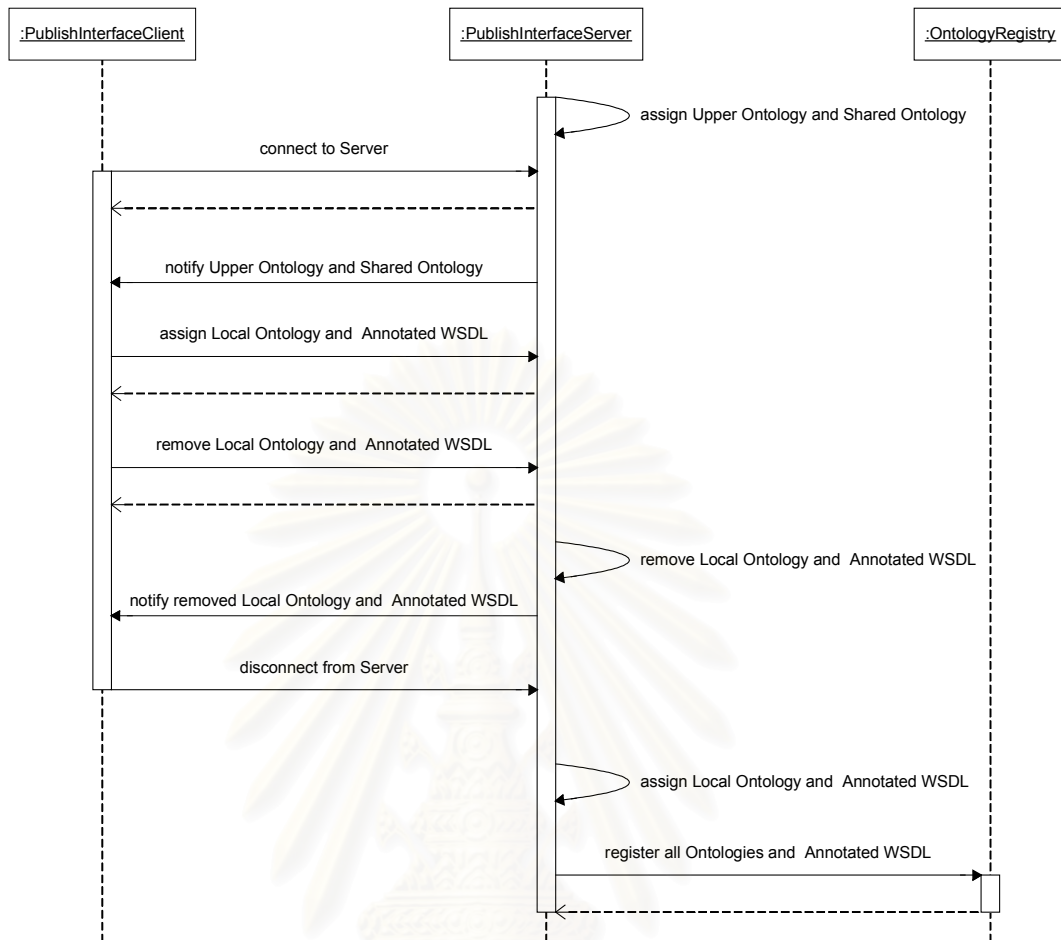


รูปที่ 4.2 การทำงานของส่วนต่อประสานสำหรับโฆษณาบริการ

จากรูปที่ 4.2 การทำงานของส่วนต่อประสานสำหรับโฆษณาบริการแบ่งออกเป็น 2 ส่วน คือ

- (1) เซอร์วิสเมเนเจอร์ของผู้ให้บริการแต่ละรายใช้ Publish Interface (Client) เพื่อกำหนดหรือลบยูอาร์แอลของโลคัลออนโทโลจีและยูอาร์แอลของเอกสารดับเบิลยูเอสดีแอล ซึ่งยูอาร์แอลที่ได้ถูกกำหนดหรือลบนี้จะถูกแจ้งไปยัง Publish Interface (Server)
- (2) ออนโทโลจีเมเนเจอร์ใช้ Publish Interface (Server) เพื่อกำหนดอัปเปอร์ออนโทโลจีและแชร์ดออนโทโลจีของกลุ่มบริการ หลังจากนั้น Publish Interface (Server) จะรายงานยูอาร์แอลของอัปเปอร์ออนโทโลจีและแชร์ดออนโทโลจีที่ได้ถูกกำหนดไปยังเซอร์วิสเมเนเจอร์ผ่าน Publish Interface (Client) นอกจากนี้ออนโทโลจีเมเนเจอร์ยังสามารถเพิ่มหรือลบยูอาร์แอลของโลคัลออนโทโลจีและยูอาร์แอลของเอกสารดับเบิลยูเอสดีแอลได้ รวมถึงสามารถลงทะเบียนยูอาร์แอลทั้งหมดไปยังส่วนลงทะเบียนออนโทโลจีเพื่อให้แผนแบบการค้นหาใช้ในการคัดเลือกบริการต่อไป

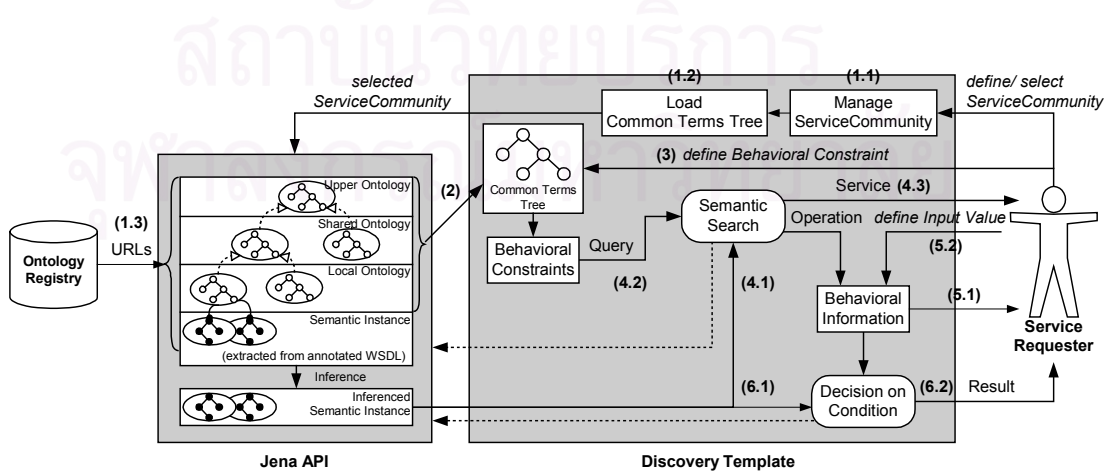
การทำงานระหว่าง Publish Interface (Server), Publish Interface (Client) และ Ontology Registry สามารถแสดงโดยแผนภาพซีควเอนซ์ (Sequence Diagram) ได้ดังรูปที่ 4.3



รูปที่ 4.3 แผนภาพซีควเอนซ์สำหรับการโฆษณาบริการ

#### 4.1.2 การออกแบบแผ่นแบบการค้นหา

สำหรับการค้นหาบริการเว็บเซอร์วิส ผู้รับบริการจะกระทำผ่านแผ่นแบบการค้นหาซึ่งมีโครงสร้างการทำงานดังรูปที่ 4.4 และสามารถแสดงการทำงานโดยแผนภาพซีควเอนซ์ได้ดังรูปที่ 4.5



รูปที่ 4.4 โครงสร้างการทำงานของแผ่นแบบการค้นหา

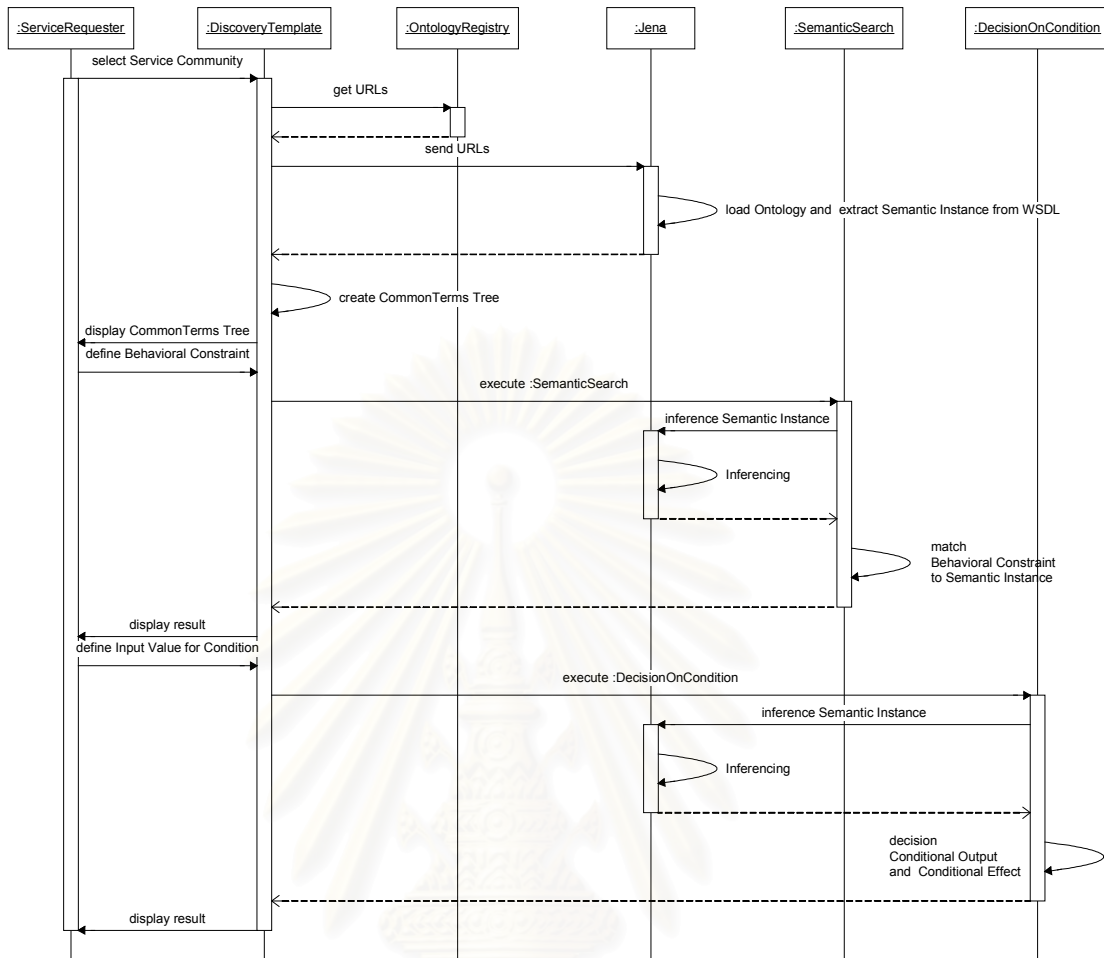


จากรูปที่ 4.4 การทำงานภายในแผนแบบการค้นหาที่มีรายละเอียดดังนี้

- (1) ผู้วิจัยได้ออกแบบให้แผนแบบการค้นหาสามารถคัดเลือกบริการได้จากหลายๆกลุ่มบริการ โดยผู้รับบริการสามารถกำหนดกลุ่มบริการที่ต้องการค้นหาได้ผ่านส่วนจัดการกลุ่มบริการ (1.1) และเลือกกลุ่มบริการที่ต้องการค้นหาได้ผ่านส่วนนำเข้าทรีของคำสามัญ (1.2) เมื่อเลือกกลุ่มบริการที่ต้องการค้นหาแล้วจึงนำเอาเข้าเอกสารอวาล์ของอัปเปอร์ออนโทโลยี แชร์ดออนโทโลยี และโลคัลออนโทโลยี รวมถึงเอกสารดับเบิลยูเอสดีแอสแอล จากยูอาร์แอลที่ได้ลงทะเบียนไว้ในส่วนลงทะเบียนออนโทโลยีของกลุ่มบริการนั้น (1.3) ยกตัวอย่างเช่นผู้รับบริการกำหนดให้ค้นหาบริการจากกลุ่มบริการ TravelAgent
- (2) อัปเปอร์ออนโทโลยี แชร์ดออนโทโลยี และโลคัลออนโทโลยี ของกลุ่มบริการที่จินาเอพีไอ นำเข้าจากขั้นตอนที่ (1.2) จะถูกใช้ร่วมกันเพื่อสร้างทรีของคำสามัญ (Common Terms Tree) โดยโครงสร้างของทรีนี้จะขึ้นไปตามความสัมพันธ์ของคลาสตามลำดับชั้นในออนโทโลยี ซึ่งแต่ละโหนด (Node) ถูกแทนด้วยคำสามัญหรือคลาสที่อยู่ในออนโทโลยี ยกตัวอย่างเช่น จากขั้นตอนที่ (1) อัปเปอร์ออนโทโลยี แชร์ดออนโทโลยีของกลุ่มบริการ TravelAgent และโลคัลออนโทโลยีของบริษัท AmexTravel (จากตัวอย่างออนโทโลยีในรูปที่ 3.11) จะถูกใช้ร่วมกันเพื่อสร้างทรีของคำสามัญ
- (3) ผู้รับบริการเลือกคลาสจากทรีของคำสามัญเพื่อใช้กำหนดข้อบังคับเชิงพฤติกรรมสำหรับค้นหาบริการ ซึ่งข้อบังคับเชิงพฤติกรรมนี้ประกอบด้วย ServiceCommunity, Service, OperationSet, Operation, Input, Output, Precondition และ Effect ยกตัวอย่างเช่น ผู้รับบริการกำหนดให้ค้นหาบริการจองตั๋วเครื่องบิน BookFlight ซึ่งประกอบด้วยการดำเนินการ BookTicket ซึ่งมีข้อมูลเข้าเป็น CreditCardType
- (4) ส่วนค้นหาเชิงความหมาย (Semantic Search) ดึงข้อเสนอเทศเชิงพฤติกรรมที่ถูกแทรกอยู่ในเอกสารดับเบิลยูเอสดีแอสแอลของทุกบริการที่ได้ลงทะเบียนไว้มาอนุมานร่วมกับ อัปเปอร์ออนโทโลยี แชร์ดออนโทโลยี และโลคัลออนโทโลยี โดยใช้เครื่องมืออนุมานของจินาเอพีไอ (4.1) และแปลงข้อบังคับเชิงพฤติกรรมที่ผู้รับบริการกำหนดให้เป็นข้อคำถามสำหรับสอบถามอาร์ดีเอฟโมเดล (4.2) เพื่อคัดเลือกบริการโดยการจับคู่คลาสในข้อบังคับเชิงพฤติกรรมกับอินสแตนซ์ในข้อเสนอเทศเชิงพฤติกรรม หลังจากนั้นจึงแสดงบริการที่ตรงกับข้อบังคับเชิงพฤติกรรมเป็นผลลัพธ์กลับไปยังผู้รับบริการ (4.3) สำหรับข้อคำถามสำหรับสอบถามอาร์ดีเอฟโมเดลนั้นในงานวิจัยนี้ได้เลือกใช้ฟังก์ชันระดับต่ำของจินาเอพีไอแทนภาษาอาร์ดีคิวแอล (จากหัวข้อที่ 4.1.2.2) ส่วนตัวอย่างการค้นหาบริการจากขั้น

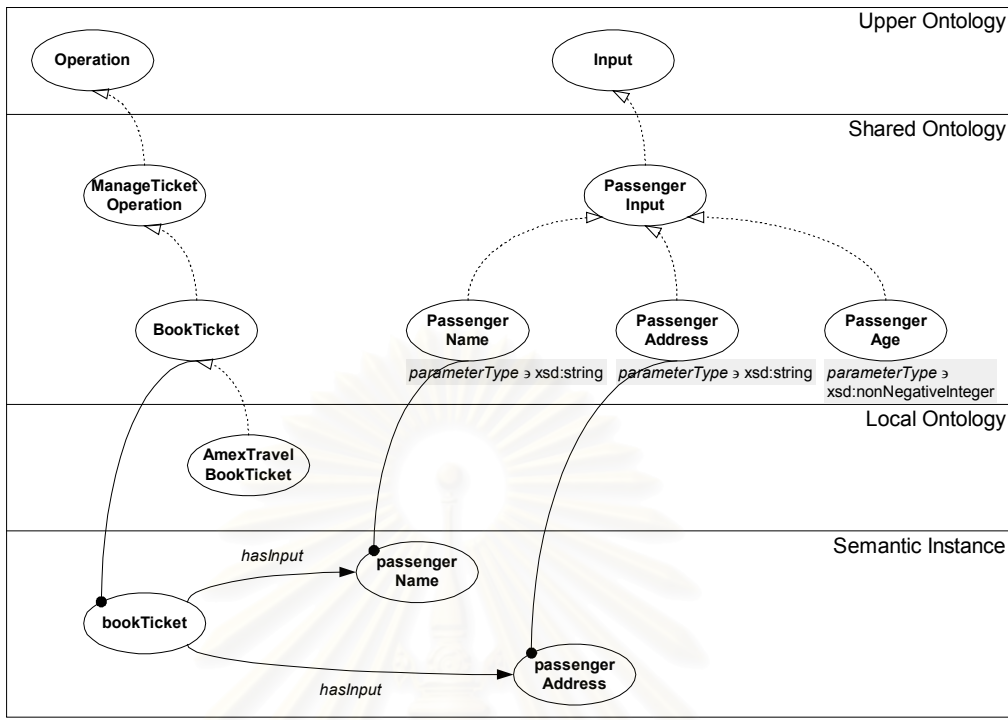
ตอนที่ (3) จะให้ผลลัพธ์เป็นบริการ BookFlight ของบริษัท AmexTravel (จากตัวอย่างข้อสนเทศเชิงพฤติกรรมในรูปที่ 3.12)

- (5) จากบริการทั้งหมดที่ค้นหาได้ในขั้นตอนที่ (4) การดำเนินการซึ่งอยู่ภายในบริการแต่ละบริการจะถูกแสดงให้กับผู้รับบริการ เมื่อผู้รับบริการเลือกการดำเนินการที่สนใจได้แล้ว ข้อสนเทศเชิงพฤติกรรมของการดำเนินการนั้นซึ่งประกอบด้วย Input, Output, Precondition และ Effect จะถูกแสดงให้กับผู้รับบริการผ่านส่วนแสดงข้อสนเทศเชิงพฤติกรรม (Behavioral Information) (5.1) ซึ่งผู้รับบริการสามารถหาผลลัพธ์ของการดำเนินการได้โดยป้อนค่าให้กับข้อมูลเข้าของการดำเนินการ (5.2) สาเหตุที่ต้องแยกการกำหนดค่าให้กับข้อมูลเข้าของการดำเนินการออกจากส่วนค้นหาเชิงความหมายเนื่องจาก ก่อนทำการค้นหาบริการผู้รับบริการไม่อาจทราบได้ว่าการดำเนินการของบริการที่ค้นหาได้ประกอบด้วยข้อมูลเข้าใดบ้างที่จำเป็นต้องใช้ในการพิจารณาเงื่อนไขก่อนการทำงาน ยกตัวอย่างเช่น บริการ BookFlight ของบริษัท AmexTravel ที่ค้นหาได้จากขั้นตอนที่ (4) ประกอบด้วยการดำเนินการ AmexTravelBookTicket ซึ่งมีข้อมูลเข้า 9 ข้อมูล (จากตัวอย่างข้อสนเทศเชิงพฤติกรรมในรูปที่ 3.12) ส่วนข้อมูลเข้าที่จำเป็นต้องใช้ในการพิจารณาเงื่อนไขก่อนการทำงานมี 3 ข้อมูลคือ CreditCardType, ArrivalAirport และ DepartureAirport ซึ่งหากผู้รับบริการกำหนดแต่เพียงค่าของข้อมูลเข้า CreditCardType จะทำให้ไม่สามารถหาผลลัพธ์ของการดำเนินการ AmexTravelBookTicket ได้ เนื่องจากมีข้อมูลไม่เพียงพอในการตัดสินใจก่อนการทำงาน ในงานวิจัยนี้จึงได้กำหนดให้ผู้รับบริการทำการค้นหาบริการก่อนโดยส่วนค้นหาเชิงความหมาย เพื่อให้ทราบได้ว่าการดำเนินการที่ค้นหาได้ประกอบด้วยข้อมูลเข้าใดบ้าง แล้วจึงกำหนดค่าให้กับข้อมูลเข้าเหล่านั้นเพื่อหาผลลัพธ์ของการดำเนินการในขั้นตอนต่อมา
- (6) ส่วนพิจารณาเงื่อนไข (Decision on Condition) นำค่าของข้อมูลเข้าที่ผู้รับบริการป้อนให้กับการดำเนินการในขั้นตอนที่ (5.2) มาพิจารณาร่วมกับข้อสนเทศเชิงพฤติกรรมที่ได้จากการอนุมาน (6.1) เพื่อตรวจสอบเงื่อนไขก่อนการทำงานของการดำเนินการ ถ้าเงื่อนไขก่อนการทำงานเป็นจริง ส่วนพิจารณาเงื่อนไขจะพิจารณาว่าการดำเนินการจะให้ ข้อมูลออก ผลกระทบ ข้อมูลออกแบบมีเงื่อนไข และผลกระทบแบบมีเงื่อนไขใดบ้าง แล้วจึงรายงานผลลัพธ์ที่ได้กลับไปยังผู้รับบริการ (6.2)



รูปที่ 4.5 แผนภาพซีควเอนซ์สำหรับการค้นหาบริการ

เนื่องจากต้นแบบโปรแกรมประยุกต์สำหรับค้นหาบริการเว็บเซอร์วิสในงานวิจัยนี้ (หัวข้อที่ 4.2.3) อนุญาตให้ผู้รับบริการกำหนดคลาสของการดำเนินการภายในบริการที่ต้องการค้นหาได้ครั้งละ 1 คลาส จึงสามารถแบ่งขั้นตอนในการคัดเลือกบริการในส่วนค้นหาเชิงความหมายได้เป็น 2 ส่วน คือ ขั้นตอนจับคู่คลาสที่ต้องการกับอินสแตนซ์ของผู้ให้บริการ (ส่วน ServiceCommunity, Service, OperationSet และ Operation ในข้อบังคับเชิงพฤติกรรม) และขั้นตอนจับคู่กลุ่มของคลาสที่ต้องการกับกลุ่มของอินสแตนซ์ของผู้ให้บริการ (ส่วน Input, Output, Precondition และ Effect ของการดำเนินการในข้อบังคับเชิงพฤติกรรม) ซึ่งจะอธิบายโดยอ้างอิงจากตัวอย่างออนโทโลยีที่ดัดแปลงมาจากออนโทโลยีของบริษัท AmexTravel (รูปที่ 3.11) ดังแสดงในรูปที่ 4.6

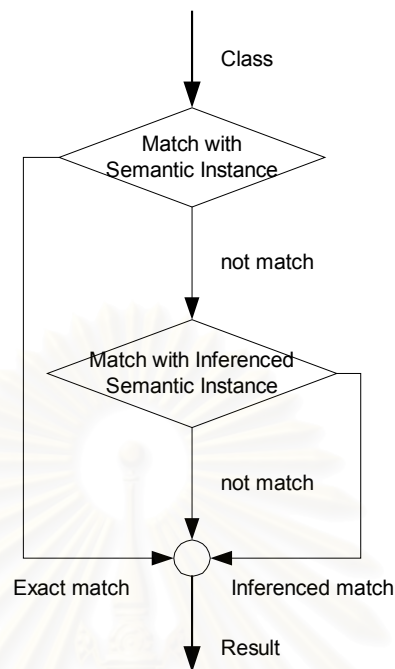


รูปที่ 4.6 ตัวอย่างอนโทโลยีสำหรับการจับคู่

4.1.2.1 การจับคู่คลาสที่ต้องการกับอินสแตนซ์ของผู้ให้บริการ

ในงานวิจัยนี้อ้างอิงแนวคิดในการจับคู่คลาสที่ต้องการกับอินสแตนซ์ของผู้ให้บริการจาก ภาษาเชิงวัตถุ ซึ่งนอกจากการจับคู่คลาสกับอินสแตนซ์ที่เป็นชนิดเดียวกันแล้ว คลาสแม่ยังสามารถจับคู่กับอินสแตนซ์ของคลาสลูก (Child Class) ได้ ยกตัวอย่างเช่น จากรูปที่ 4.6 คลาส ManageTicketOperation ที่ผู้รับบริการต้องการสามารถจับคู่กับอินสแตนซ์ของคลาส BookTicket ของผู้ให้บริการได้ เนื่องจากคลาส ManageTicketOperation เป็นคลาสแม่ของคลาส BookTicket แต่หากคลาสที่ต้องการเป็น AmexTravelBookTicket จะไม่สามารถจับคู่กับอินสแตนซ์ของคลาส BookTicket ได้ เนื่องจากเป็นคลาสลูกของคลาส BookTicket สำหรับขั้นตอนวิธีการจับคู่คลาสกับอินสแตนซ์ได้แสดงดังรูปที่ 4.7

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.7 ขั้นตอนวิธีการจับคู่คลาสกับอินสแตนซ์

การจับคู่คลาสที่ต้องการกับอินสแตนซ์ของผู้ให้บริการเริ่มต้นจากการพิจารณาอินสแตนซ์ที่อยู่ในข้อสนเทศเชิงพฤติกรรม เพื่อจับคู่คลาสกับอินสแตนซ์ที่เป็นชนิดเดียวกันซึ่งจะให้ผลลัพธ์เป็น Exact match ถ้าไม่สามารถจับคู่ได้จะพิจารณาต่อโดยใช้ข้อสนเทศเชิงพฤติกรรมที่ถูกอนุมานเพื่อจับคู่คลาสกับอินสแตนซ์ของคลาสลูกซึ่งหากจับคู่ได้จะให้ผลลัพธ์เป็น Inferred match สำหรับการจับคู่นี้สามารถทำได้โดยภาษาอาร์ดีคิวแอลหรือฟังก์ชันระดับต่ำของจิงนาเอพีไอ

#### 4.1.2.2 การจับคู่กลุ่มของคลาสที่ต้องการกับกลุ่มของอินสแตนซ์ของผู้ให้บริการ

แม้ว่าจิงนาเอพีไอจะสนับสนุนภาษาอาร์ดีคิวแอลซึ่งทำให้การสอบถามอาร์ดีเอฟโมเดลทำได้สะดวกขึ้น โดยสามารถตั้งข้อความซึ่งประกอบด้วยเงื่อนไขที่อยู่ในรูปอาร์ดีเอฟทริปเปิลได้หลายๆเงื่อนไขพร้อมกัน แต่การจับคู่กลุ่มของคลาสกับกลุ่มของอินสแตนซ์ในบางกรณีไม่สามารถทำได้โดยตรงด้วยภาษาอาร์ดีคิวแอล ยกตัวอย่างจากออนโทโลยีในรูปที่ 4.6 ต้องการค้นหาการดำเนินการ BookTicket ซึ่งประกอบด้วยข้อมูลเข้าจำนวน 3 ข้อมูล กำหนดให้ข้อมูลแรกเป็นอินสแตนซ์ของคลาส PassengerName ข้อมูลที่สองเป็นอินสแตนซ์ของคลาส PassengerAddress และข้อมูลที่สามไม่เฉพาะเจาะจงเพียงแต่ต้องเป็นอินสแตนซ์ของคลาส PassengerInput สามารถเขียนเป็นข้อความแบบอาร์ดีคิวแอลได้ดังนี้

```

SELECT ?operation
WHERE (?operation, <rdf:type>, <BookTicket>),
      (?operation, <hasInput>, ?input1),
      (?operation, <hasInput>, ?input2),
      (?operation, <hasInput>, ?input3),
      (?input1, <rdf:type>, <PassengerName>),
      (?input2, <rdf:type>, <PassengerAddress>),
      (?input3, <rdf:type>, <PassengerInput>)
USING rdf FOR <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

```

ผลลัพธ์ที่ได้มีลักษณะดังต่อไปนี้

```

?operation
=====
<bookTicket>

```

จากผลลัพธ์ที่ได้ อินสแตนซ์ bookTicket เป็นผลลัพธ์ที่ไม่ตรงกับความต้องการเพราะประกอบด้วยข้อมูลเข้าเพียง 2 ข้อมูล แต่เป็นผลลัพธ์ที่ถูกต้องตามข้อคำถามเนื่องจากคลาส PassengerName และคลาส PassengerAddress เป็นคลาสลูกของคลาส PassengerInput ด้วย ทำให้ข้อมูลเข้าเพียง 2 ข้อมูลนี้ตรงกับข้อมูลเข้า 3 ข้อมูลที่กำหนดในข้อคำถาม ดังนั้นเพื่อให้ผลลัพธ์ที่ได้จากข้อคำถามสอดคล้องกับความต้องการจริงๆ การค้นหาต้องแบ่งออกเป็น 3 ขั้นตอน คือ

1. ค้นหาอินสแตนซ์ของคลาส BookTicket จากข้อสนเทศเชิงพฤติกรรมซึ่งสามารถเขียนเป็นข้อคำถามแบบอาร์ดีคิวแอลได้ดังนี้

```

SELECT ?operation
WHERE (?operation, <rdf:type>, <BookTicket>),
USING rdf FOR <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

```

ผลลัพธ์ที่ได้คือ

```

?operation
=====
<bookTicket>

```

2. ค้นหาข้อมูลเข้าของอินสแตนซ์ bookTicket จากข้อสนเทศเชิงพฤติกรรมซึ่งสามารถเขียนเป็นข้อคำถามแบบอาร์ดีคิวแอลได้ดังนี้

```
SELECT ?input
WHERE (<bookTicket>, <hasInput>, ?input)
```

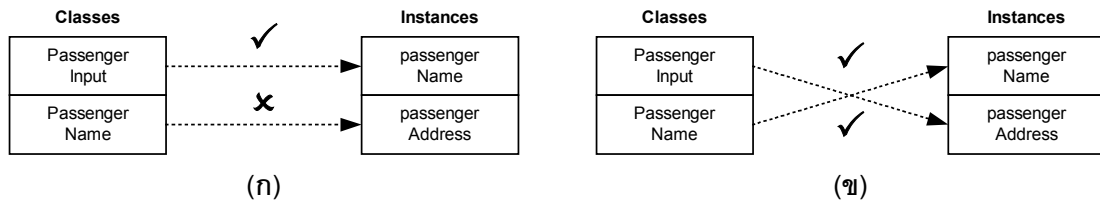
ผลลัพธ์ที่ได้คือ

```
?input
=====
<passengerName>
<passengerAddress>
```

- 3. จับคู่กลุ่มของคลาสที่ผู้รับบริการกำหนด คือ PassengerName, PassengerAddress และ PassengerInput เข้ากับกลุ่มของอินสแตนซ์ที่ค้นหาได้ คือ passengerName และ passengerAddress พบว่าไม่สามารถจับคู่ได้เนื่องจากอินสแตนซ์ bookTicket ประกอบด้วยข้อมูลเข้าเพียง 2 ข้อมูล

จากปัญหาข้างต้นทำให้การคัดเลือกบริการด้วยภาษาอาร์ดีคิวแอลต้องแบ่งออกเป็นหลายขั้นตอน ในแต่ละขั้นตอนสามารถตั้งข้อความคำถามซึ่งประกอบด้วยเงื่อนไขที่อยู่ในรูปอาร์ดีคิวแอลเปิดได้เพียง 1 เงื่อนไข การใช้ภาษาอาร์ดีคิวแอลเพื่อคัดเลือกบริการจึงไม่แตกต่างจากการใช้ฟังก์ชันระดับต่ำของจิงนาเอพีไอซึ่งอนุญาตให้ตั้งข้อความคำถามได้ครั้งละ 1 เงื่อนไข ในงานวิจัยนี้ผู้วิจัยจึงได้เลือกใช้ฟังก์ชันระดับต่ำของจิงนาเอพีไอแทนเพราะสามารถทำงานได้รวดเร็วกว่าภาษาอาร์ดีคิวแอล

สำหรับขั้นตอนวิธีการจับคู่กลุ่มของคลาสกับกลุ่มของอินสแตนซ์ เนื่องจากกลุ่มของอินสแตนซ์ที่ค้นหาได้จากจิงนาเอพีไอไม่มีลำดับที่แน่นอน การจับคู่แบบสุ่มอาจทำให้การค้นหาไม่ประสบความสำเร็จ ยกตัวอย่างจากออนโทโลยีในรูปที่ 4.6 ถ้าต้องการค้นหาการดำเนินการ BookTicket ซึ่งประกอบด้วยข้อมูลเข้า 2 ข้อมูล ข้อมูลแรกเป็นอินสแตนซ์ของคลาส PassengerName ข้อมูลที่สองไม่เฉพาะเจาะจงเพียงแต่ต้องเป็นอินสแตนซ์ของคลาส PassengerInput จากข้อบังคับนี้อินสแตนซ์ bookTicket ถือเป็นผลลัพธ์ที่ตรงกับความต้องการเพราะประกอบด้วยข้อมูลเข้า passengerName ซึ่งเป็นอินสแตนซ์ของคลาส PassengerName และประกอบด้วยข้อมูลเข้า passengerAddress ซึ่งถือว่าเป็นอินสแตนซ์ของคลาส PassengerInput (Inferenced match) ตัวอย่างการจับคู่แบบสุ่มสำหรับกลุ่มของคลาสกับกลุ่มของอินสแตนซ์นี้แสดงได้ดังรูปที่ 4.8



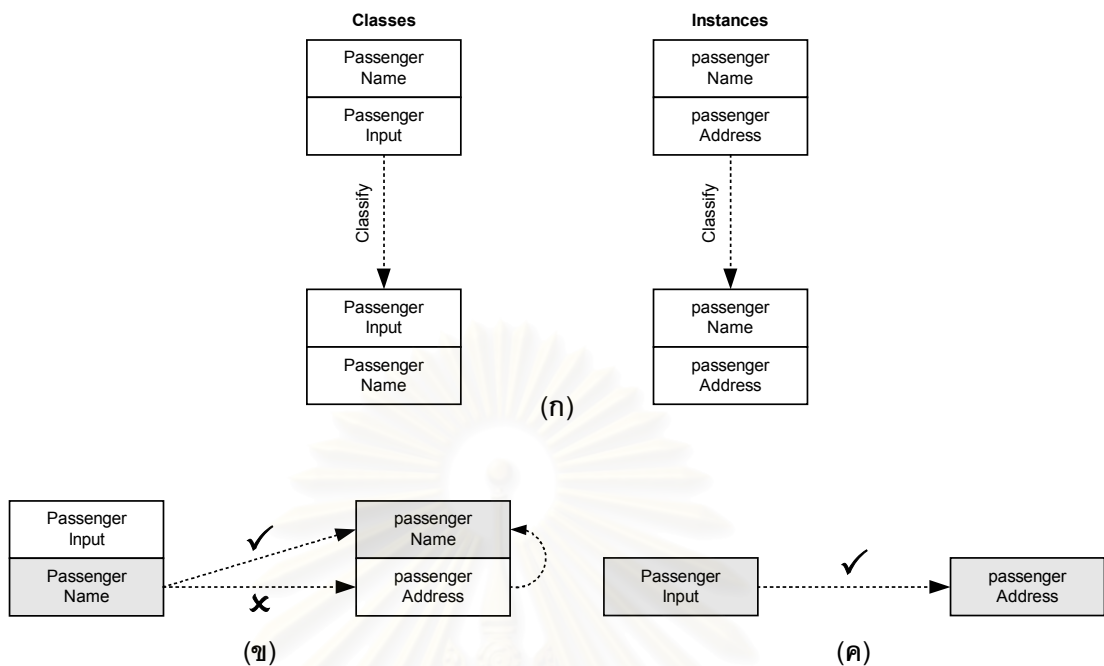
รูปที่ 4.8 ตัวอย่างการจับคู่แบบสุ่ม

รูปที่ 4.8 (ก) แสดงการจับคู่แบบสุ่มที่ไม่ประสบความสำเร็จ ส่วนรูปที่ 4.8 (ข) แสดงการจับคู่แบบสุ่มที่ประสบความสำเร็จ สาเหตุที่การจับคู่กลุ่มของคลาสกับกลุ่มของอินสแตนซ์ในรูปที่ 4.8 (ก) ไม่ประสบความสำเร็จเนื่องจากคลาส PassengerInput ไม่ได้อยู่ล่างสุดของลำดับชั้น ดังนั้นจึงสามารถจับคู่ได้ทั้งกับอินสแตนซ์ที่เป็นชนิดเดียวกันและอินสแตนซ์ของคลาสลูกได้หลายคลาส แต่กลับถูกนำมาพิจารณาก่อนทำให้อาจไปจับคู่กับอินสแตนซ์ที่ไม่เหมาะสมและเป็นการตัดโอกาสคลาสลูกเช่น PassengerName ในการจับคู่ ดังนั้นในการจับคู่จึงควรพิจารณาคลาสในลำดับชั้นต่ำกว่าให้จับคู่ไปให้ได้ก่อน ก่อนที่จะพิจารณาคลาสในลำดับชั้นสูงกว่าซึ่งมีโอกาสจับคู่ได้มากกว่า

ในงานวิจัยนี้จึงได้เสนอ วิธีการจับคู่กลุ่มของคลาสกับกลุ่มของอินสแตนซ์แบบต่ำสู่สูง (Bottom-Up) โดยพยายามจับคู่คลาสกับอินสแตนซ์ที่อยู่ล่างสุดของลำดับชั้นก่อน แล้วจึงจับคู่คลาสกับอินสแตนซ์ที่อยู่ถัดขึ้นไปในลำดับชั้น ดังนั้นก่อนทำการจับคู่จึงต้องนำกลุ่มของคลาสและกลุ่มของอินสแตนซ์มาจัดเป็นลำดับ (Classify) ตามลำดับชั้นในออนโทโลจีก่อน ดังตัวอย่างในรูปที่ 4.9

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย





รูปที่ 4.9 ตัวอย่างการจับคู่แบบตัวคู่สูง

จากรูปที่ 4.9 (ก) แสดงการจัดลำดับกลุ่มของคลาสและกลุ่มของอินสแตนซ์ตามลำดับชั้นในออนโทโลยี (ตามรูปที่ 4.6) รูปที่ 4.9 (ข) แสดงการจับคู่คลาส PassengerName โดยเริ่มพิจารณาจากอินสแตนซ์ passengerAddress ซึ่งอยู่ล่างสุดในลำดับชั้น พบว่าไม่สามารถจับคู่ได้ จึงพิจารณากับอินสแตนซ์ลำดับถัดไปคือ passengerName พบว่าสามารถจับคู่ได้ รูปที่ 4.9 (ค) แสดงการจับคู่คลาส PassengerInput กับอินสแตนซ์ passengerAddress พบว่าสามารถจับคู่ได้ การจับคู่คลาสกับอินสแตนซ์ทั้งหมดจึงประสบความสำเร็จ

สำหรับวิธีการจับคู่แบบตัวคู่สูงนี้ผู้วิจัยได้ทดสอบโดยใช้ตัวอย่างข้อสนเทศเชิงพฤติกรรมที่ได้สร้างขึ้นแล้วพบว่าสามารถจับคู่กลุ่มของคลาสกับกลุ่มของอินสแตนซ์ได้อย่างถูกต้อง

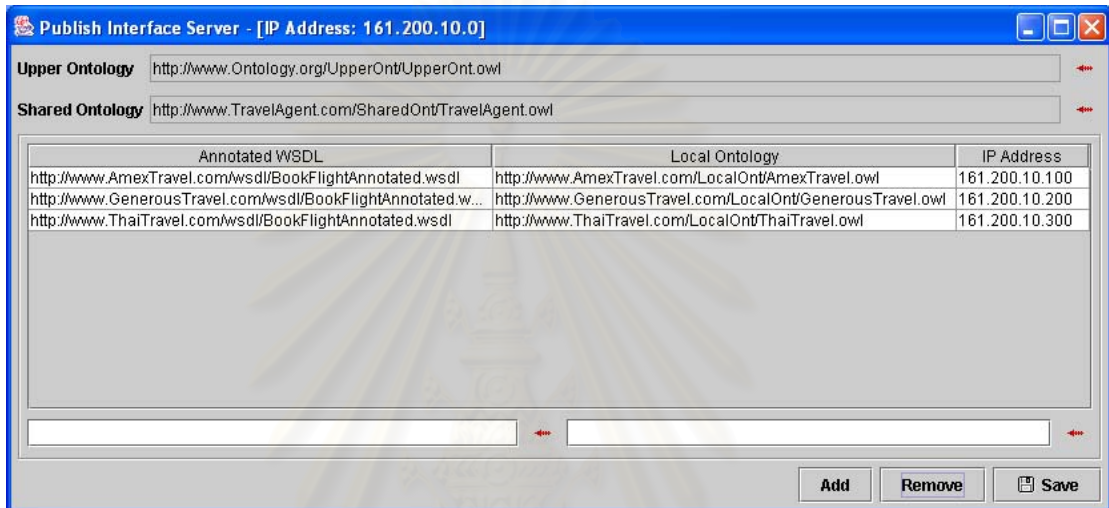
#### 4.2 ต้นแบบโปรแกรมประยุกต์สำหรับการโฆษณาและการค้นหาบริการเว็บเซอร์วิส

ต้นแบบโปรแกรมประยุกต์สำหรับการโฆษณาและการค้นหาบริการเว็บเซอร์วิส ประกอบด้วย PublishInterfaceServer, PublishInterfaceClient และ DiscoveryTemplate ซึ่งจะอธิบายในหัวข้อที่ 4.2.1 หัวข้อที่ 4.2.2 และ หัวข้อที่ 4.2.3 ตามลำดับ

##### 4.2.1 ต้นแบบโปรแกรมประยุกต์ PublishInterfaceServer

PublishInterfaceServer อนุญาตให้ออนโทโลยีเมเนเจอร์กำหนดอัปเดตออนโทโลยี และแชร์ออนโทโลยีของกลุ่มบริการ รวมถึงสามารถลบและบันทึกบริการที่ถูกลงทะเบียนไว้ได้ นอก

จากนี้ออนโทโลยีเมเนเจอร์ยังสามารถลงทะเบียนบริการเพิ่มเติมจากที่เซอร์วิสเมเนเจอร์ได้ลงทะเบียนไว้ผ่านส่วน Annotated WSDL และส่วน Local Ontology สำหรับการบันทึก PublishInterfaceServer จะบันทึกยูอาร์แอลทั้งหมดของแอปพลิเคชันออนโทโลยี เซอร์วิสออนโทโลยี ไลค์ออนโทโลยี และเอกสารดับเบิลยูเอสดีแอล ไปยังส่วนลงทะเบียนออนโทโลยีที่อยู่ในรูปแบบแฟ้มข้อความ (Text File)



รูปที่ 4.10 ส่วนต่อประสานกับผู้ใช้ของ PublishInterfaceServer

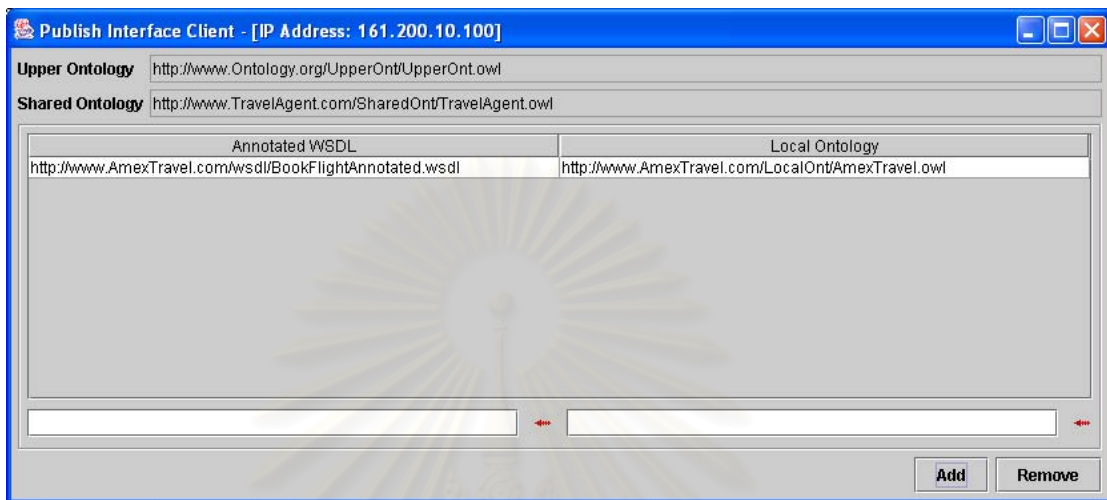
จากรูปที่ 4.10 แสดงออนโทโลยีของกลุ่มบริการตัวแทนสำนักงานท่องเที่ยว TravelAgent ซึ่งประกอบด้วยแอปพลิเคชันออนโทโลยี <http://www.Ontology.org/UpperOnt/UpperOnt.owl> เซอร์วิสออนโทโลยี <http://www.TravelAgent.com/SharedOnt/TravelAgent.owl> และบริการที่ได้ถูกลงทะเบียนไว้ 3 บริการ

เนื่องจากต้นแบบโปรแกรมประยุกต์ PublishInterfaceServer และ PublishInterfaceClient ถูกพัฒนาด้วยภาษาจาวาในรูปแบบไคลเอนต์เซิร์ฟเวอร์ซึ่งติดต่อกันผ่านซ็อกเก็ต (Socket) การเรียกใช้งาน PublishInterfaceServer จึงต้องระบุพอร์ตที่ใช้ในการติดต่อกับตัวอย่างเช่น `java PublishInterfaceServer 200` หมายถึงการเรียกใช้งาน PublishInterfaceServer ผ่านพอร์ตหมายเลข 200

#### 4.2.2 ต้นแบบโปรแกรมประยุกต์ PublishInterfaceClient

PublishInterfaceClient จะแสดงแอปพลิเคชันออนโทโลยี และเซอร์วิสออนโทโลยี ที่ถูกกำหนดโดยออนโทโลยีเมเนเจอร์ และอนุญาตให้เซอร์วิสเมเนเจอร์ลงทะเบียนบริการได้โดย กำหนดยูอาร์

แอดของเอกสารดับเบิลยูเอสดีแอด (ผ่านส่วน Annotated WSDL) และยูอาร์แอลของโลคัลออนโทโลยี (ผ่านส่วน Local Ontology) ดังรูปที่ 4.11



รูปที่ 4.11 ส่วนต่อประสานกับผู้ใช้ของ PublishInterfaceClient

จากรูปที่ 4.11 แสดงการลงทะเบียนบริการจองตั๋วเครื่องบิน BookFlight ของบริษัท AmexTravel ภายใต้โลคัลออนโทโลยี <http://www.AmexTravel.com/LocalOnt/AmexTravel.owl> ซึ่งอ้างอิงอัปเปอร์ออนโทโลยีและเซิร์ฟเวอร์ออนโทโลยีของกลุ่มบริการ TravelAgent สอดคล้องกับรูปที่ 3.11 และรูปที่ 3.12 ในบทที่ 3

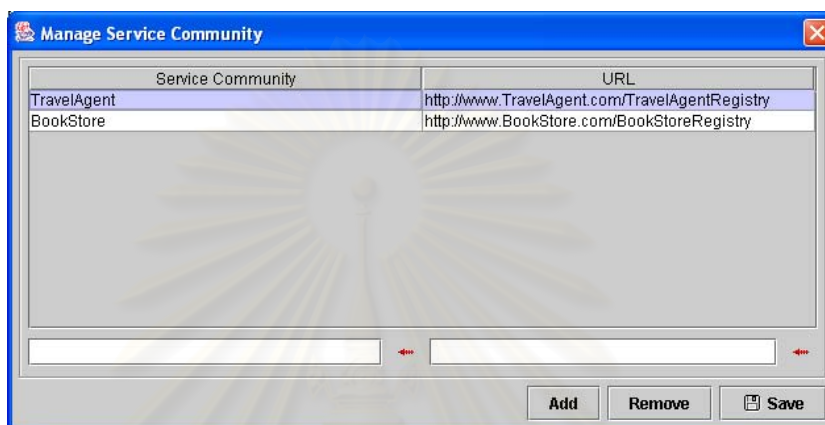
สำหรับการเรียกใช้งาน PublishInterfaceClient ต้องกำหนด IP Address ของเครื่องคอมพิวเตอร์ที่ Publish Interface (Server) ทำงานอยู่ รวมถึงหมายเลขพอร์ตที่ใช้ในการติดต่อด้วย ยกตัวอย่างเช่น `java PublishInterfaceClient 161.200.10.0 200`

#### 4.2.3 ต้นแบบโปรแกรมประยุกต์ DiscoveryTemplate

สำหรับส่วนต่อประสานกับผู้ใช้ขององค์ประกอบต่างๆภายในแผนแบบการค้นหา จะอธิบายประกอบกับตัวอย่างการค้นหาบริการจองตั๋วเครื่องบิน BookFlight ภายใต้กลุ่มบริการตัวแทนสำนักงานท่องเที่ยว TravelAgent ซึ่งรับชำระเงินด้วยบัตรเครดิตและจัดส่งตั๋วให้หลังจากชำระเงิน โดยมีรายละเอียดดังต่อไปนี้

#### 4.2.3.1 ส่วนจัดการกลุ่มบริการ

ส่วนจัดการกลุ่มบริการอนุญาตให้ผู้รับบริการกำหนดกลุ่มบริการที่ต้องการค้นหา โดยระบุชื่อกลุ่มบริการ (ผ่านส่วน Service Community) และกำหนดยูอาร์แอลของเพิ่มข้อความที่เก็บข้อมูลการลงทะเบียนออนไลน์ (ผ่านส่วน URL) ดังรูปที่ 4.12

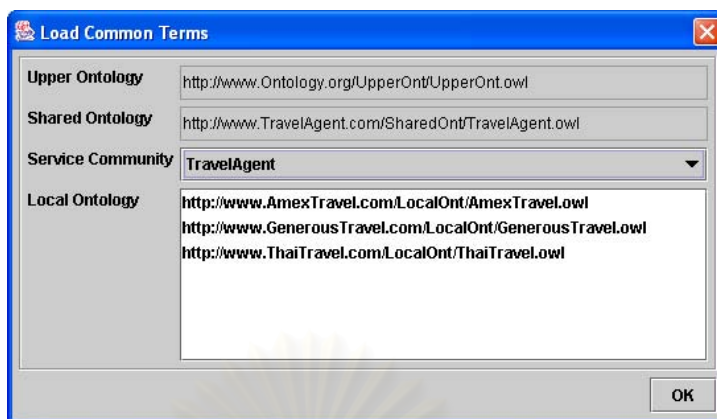


รูปที่ 4.12 ส่วนต่อประสานกับผู้ใช้ของส่วนจัดการกลุ่มบริการ

จากรูปที่ 4.12 ในระเบียนแรก ผู้รับบริการกำหนดให้ค้นหาบริการจากกลุ่มบริการ TravelAgent โดยกำหนดยูอาร์แอลของเพิ่มข้อความที่เก็บข้อมูลการลงทะเบียนออนไลน์ให้เป็น <http://www.TravelAgent.com/TravelAgentRegistry>

#### 4.2.3.2 ส่วนจัดการทรีของคำสามัญ

ส่วนจัดการทรีของคำสามัญแสดงกลุ่มบริการทั้งหมดที่ผู้รับบริการกำหนด (ในส่วน Service Community) เมื่อผู้รับบริการเลือกกลุ่มบริการที่ต้องการแล้ว เอกสารอาร์ชของอัปเปอร์ออนไลน์และเซิร์ดออนไลน์จะถูกใช้เพื่อสร้างทรีของคำสามัญ นอกจากนี้ผู้รับบริการยังสามารถเลือกเอกสารอาร์ชของโลคัลออนไลน์ เพื่อใช้ร่วมในการสร้างทรีของคำสามัญได้อีกด้วย ดังรูปที่ 4.13



รูปที่ 4.13 ส่วนต่อประสานกับผู้ใช้ของส่วนจัดการทรีของคำสามัญ

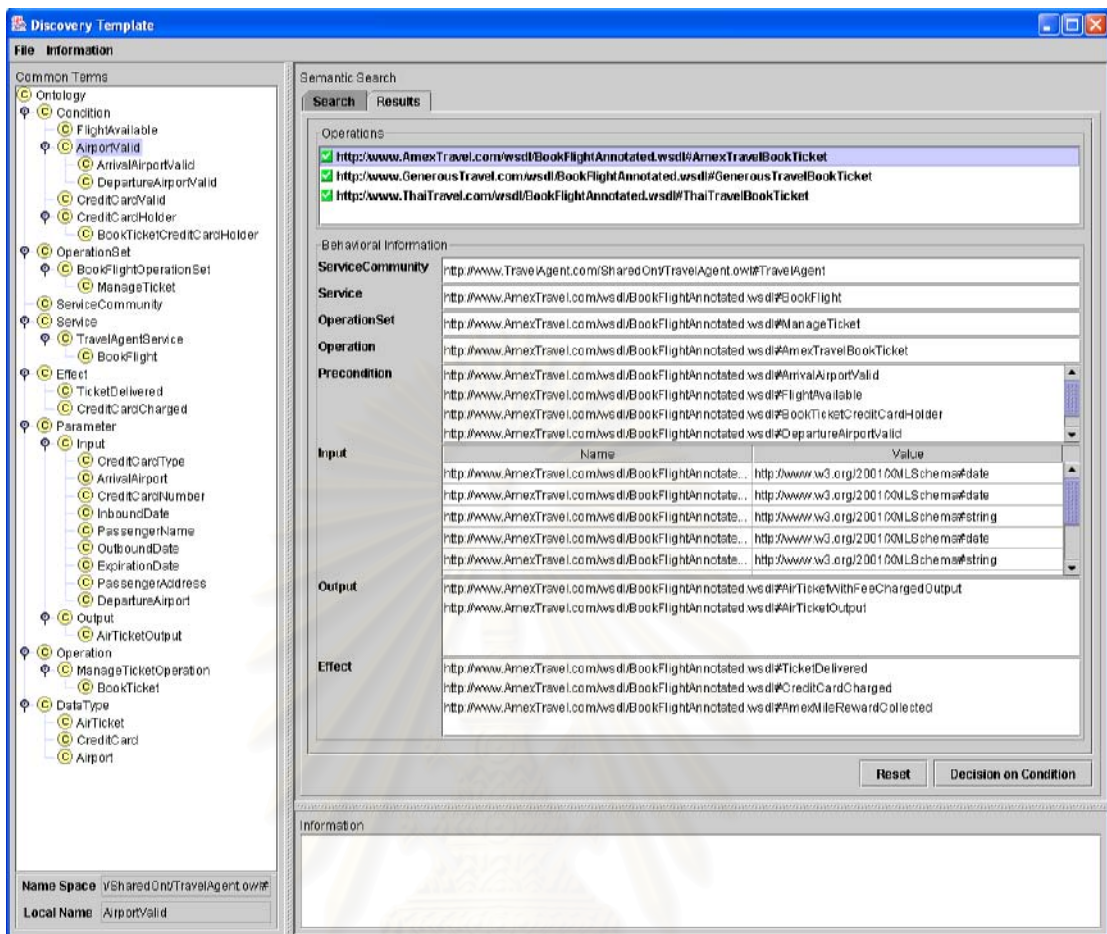
จากรูปที่ 4.13 ผู้รับบริการเลือกกลุ่มบริการ TravelAgent เพื่อสร้างทรีของคำสามัญจากเอกสาร <http://www.Ontology.org/UpperOnt/UpperOnt.owl> ของอัปเปอร์ออนโทโลยี และเอกสาร <http://www.TravelAgent.com/SharedOnt/TravelAgent.owl> ของแชร์ดออนโทโลยี

#### 4.2.3.3 ส่วนกำหนดข้อบังคับเชิงพฤติกรรม

รูปที่ 4.14 แสดงแผนแบบการค้นหาซึ่งอนุญาตให้ผู้รับบริการเลือกคำสามัญจากทรีของคำสามัญ (ในส่วน Common Terms) เพื่อกำหนดข้อบังคับเชิงพฤติกรรม (ในส่วน Behavioral Constraints) ซึ่งประกอบด้วย ServiceCommunity, Service, OperationSet, Operation, Precondition, Input, Output และ Effect สำหรับใช้ในการค้นหาบริการ โดยบริการที่จะถือว่าตรงกับความต้องการจะต้องมีพฤติกรรมตรงตามองค์ประกอบที่ระบุในข้อบังคับ สำหรับองค์ประกอบที่ไม่ถูกกำหนดด้วยคำสามัญกำหนดให้คัดเลือกบริการโดยไม่พิจารณาองค์ประกอบนั้น

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



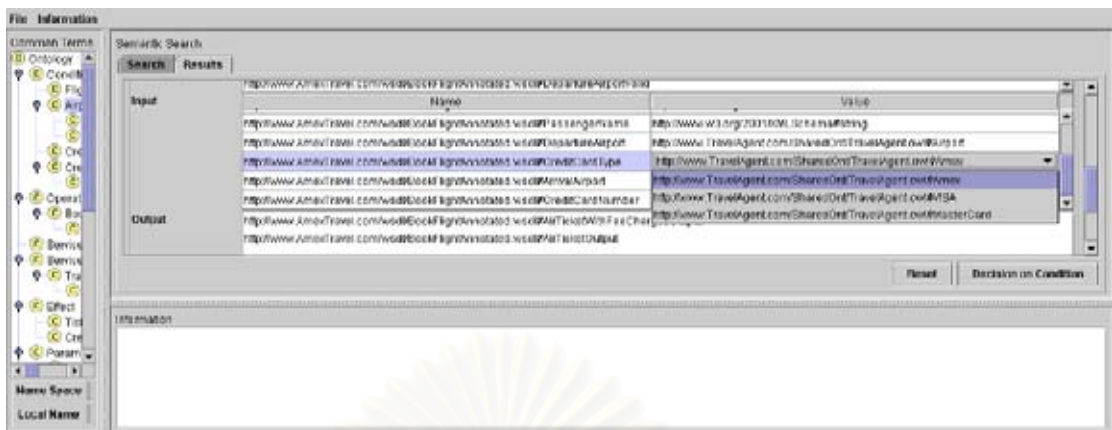


รูปที่ 4.15 ส่วนต่อประสานกับผู้ใช้ของส่วนแสดงข้อสนเทศเชิงพฤติกรรม

จากส่วน Operations ในรูปที่ 4.15 แสดงการดำเนินการทั้งหมดที่ได้จากการค้นหาบริการโดยใช้ข้อบังคับเชิงพฤติกรรมในรูปที่ 4.14 เพื่อค้นหาบริการจองตั๋วเครื่องบินซึ่งรับชำระเงินด้วยบัตรเครดิตและจัดส่งตั๋วให้หลังจากชำระเงิน สำหรับส่วน Behavioral Information ในรูปที่ 4.15 ได้แสดงข้อสนเทศเชิงพฤติกรรมของ AmexTravelBookTicket ซึ่งเป็นอินสแตนซ์ของการดำเนินการ BookTicket ภายใต้บริการจองตั๋วเครื่องบินของบริษัท AmexTravel

#### 4.2.3.5 ส่วนพิจารณาเงื่อนไข

ในส่วน Input ของส่วน Behavioral Information ผู้รับบริการสามารถกำหนดค่าให้กับข้อมูลเข้าที่มีชนิดพารามิเตอร์เป็นคลาสแบบแจกแจงได้ เพื่อนำไปใช้ตัดสินค่าความจริงของเงื่อนไขสำหรับ เงื่อนไขก่อนการทำงาน ข้อมูลออกแบบมีเงื่อนไข และผลกระทบแบบมีเงื่อนไข ดังรูปที่ 4.16



รูปที่ 4.16 ตัวอย่างการกำหนดค่าให้กับข้อมูลเข้า

จากรูปที่ 4.16 อินสแตนซ์ AmexTravelBookTicket ประกอบด้วยข้อมูลเข้า 9 ข้อมูล ข้อมูลเข้าที่มีชนิดพารามิเตอร์เป็นคลาสแบบแจกันบ คือ CreditCardType, ArrivalAirport และ DepartureAirport ในที่นี้กำหนดให้มีค่าเป็น Amex, BKK และ HKG ตามลำดับ เมื่อผ่านการพิจารณาเงื่อนไขแล้วผลลัพธ์ของการดำเนินการจะถูกแสดงผ่านส่วน Information ดังรูปที่ 4.17

```

////////////////////////////////////Decision on Condition
Upper Ontology: http://www.Ontology.org/UpperOnt/UpperOnt.owl
Shared Ontology: http://www.TravelAgent.com/SharedOnt/TravelAgent.owl
Local Ontology: http://www.AmexTravel.com/LocalOnt/AmexTravel.owl
Semantic Instance: http://www.AmexTravel.com/wsd/BookFlightAnnotated.wsd
////////////////////////////////////Search Result1
Operation: http://www.AmexTravel.com/wsd/BookFlightAnnotated.wsd#AmexTravelBookTicket .
////////////////////////////////////Checking Precondition...
Checking Precondition completed
////////////////////////////////////Computing Conditional Output...
http://www.AmexTravel.com/wsd/BookFlightAnnotated.wsd#AirTicketOutput .
////////////////////////////////////Computing Conditional Effect...
http://www.AmexTravel.com/wsd/BookFlightAnnotated.wsd#TicketDelivered .
http://www.AmexTravel.com/wsd/BookFlightAnnotated.wsd#CreditCardCharged .
http://www.AmexTravel.com/wsd/BookFlightAnnotated.wsd#AmexMileRewardCollected .
Decision on Condition finish .

```

รูปที่ 4.17 ตัวอย่างผลลัพธ์ของการดำเนินการ

จากตัวอย่างในรูปที่ 4.17 แสดงผลลัพธ์ของการพิจารณาเงื่อนไขสำหรับอินสแตนซ์ AmexTravelBookTicket ซึ่งประกอบด้วยเงื่อนไขก่อนการทำงาน 5 เงื่อนไข เงื่อนไขที่สามารถตัดสินค่าความจริงได้มี 3 เงื่อนไข คือ BookTicketCreditCardHolder, ArrivalAirportValid และ DepartureAirportValid ซึ่งเมื่อพิจารณาจากค่าของข้อมูลเข้าแล้วพบว่าเป็นจริงทั้ง 3 เงื่อนไข ทำให้สามารถประมวลผลการดำเนินการได้ และเนื่องจากข้อมูลเข้า CreditCardType มีค่าเป็น



Amex อินสแตนซ์ AmexTravelBookTicket ซึ่งมีพฤติกรรมตามรูปที่ 3.10 จึงให้ผลลัพธ์เป็น ข้อมูลออกแบบมีเงื่อนไขคือ AirTicketOutput ผลกระทบคือ TicketDelivered และ CreditCardCharged และผลกระทบแบบมีเงื่อนไขคือ AmexMileRewardCollected ทั้งนี้เป็นไปตามเงื่อนไขที่กำหนดเมื่อใช้บัตรเครดิต Amex



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## บทที่ 5

### สรุปผลการวิจัย

จากการศึกษาและวิจัยเพื่อออกแบบและพัฒนาเฟรมเวิร์คสำหรับสนับสนุนการโฆษณา และการค้นหาเว็บเซอร์วิสจากเอกสารดับเบิลยูเอสดีแอลที่ถูกแทรกข้อสนเทศเชิงพฤติกรรมอิงออนโทโลยี สามารถสรุปผลและเสนอแนะแนวทางเพื่อการทำวิจัยต่อในอนาคตได้ดังนี้

#### 5.1 สรุปผลการวิจัย

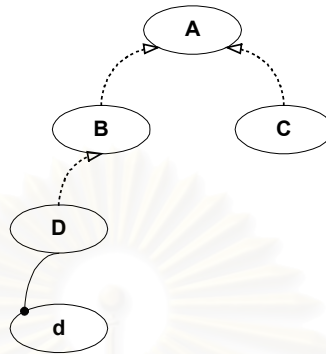
ผลที่ได้จากการวิจัยนี้คือต้นแบบเฟรมเวิร์คสำหรับการโฆษณาและการค้นหาบริการเว็บเซอร์วิส สำหรับการโฆษณาบริการเว็บเซอร์วิส ผู้วิจัยได้ออกแบบอับเปอร์ออนโทโลยีซึ่งใช้สำหรับการอธิบายข้อสนเทศเชิงพฤติกรรมของบริการและเชื่อมต่อการตัดสินใจค่าความจริงของเงื่อนไข สำหรับ เงื่อนไขก่อนการทำงาน ข้อมูลออกแบบมีเงื่อนไข และ ผลกระทบแบบมีเงื่อนไข จากการพิจารณาคลัสแบบแฉงนับที่กำหนดไว้ในออนโทโลยี รวมทั้งได้ออกแบบข้อกำหนดในการแทรกข้อสนเทศเชิงพฤติกรรมในรูปแบบอาร์ดีเอฟเอ็กซ์เอ็มแอลให้กับเอกสารดับเบิลยูเอสดีแอล นอกจากนี้ยังได้พัฒนาต้นแบบโปรแกรมประยุกต์ PublishInterfaceServer และ PublishInterfaceClient เพื่อใช้โฆษณาบริการ ส่วนการค้นหาบริการเว็บเซอร์วิสนั้นผู้วิจัยได้พัฒนาต้นแบบโปรแกรมประยุกต์ DiscoveryTemplate เพื่อใช้ค้นหาบริการซึ่งมีความหมายและพฤติกรรมตรงกับที่ผู้รับบริการต้องการ นอกจากนี้ผู้วิจัยยังได้สร้างตัวอย่างของ แทร็คออนโทโลยี โคลด์ออนโทโลยี ซีแมนติกอินสแตนซ์ รวมถึงเอกสารดับเบิลยูเอสดีแอล เพื่อใช้ทดสอบการทำงานของเฟรมเวิร์คอีกด้วย

#### 5.2 ปัญหาและข้อจำกัดที่พบจากการวิจัย

ปัญหาและข้อจำกัดที่พบจากการวิจัยมีดังนี้

1. งานวิจัยนี้ได้พัฒนาต้นแบบโปรแกรมประยุกต์ DiscoveryTemplate โดยใช้จิงนาเอพีไอ เพื่อ นำเข้า จัดการ รวมถึงอนุมานเอกสารอาร์วล์ของ อับเปอร์ออนโทโลยี แทร็คออนโทโลยี โคลด์ออนโทโลยี และ ซีแมนติกอินสแตนซ์ เนื่องจากจิงนาเอพีไอถูกพัฒนาด้วยภาษาจาวา ทำให้เกิดความล่าช้าและใช้หน่วยความจำจำนวนมากในขั้นตอนอนุมาน นอกเหนือจากเครื่องมืออนุมานภายในแล้วจิงนาเอพีไอยังสนับสนุนการเชื่อมต่อกับเครื่องมืออนุมานภายนอกด้วยโพรโทคอลเอชทีทีพีผ่าน localhost ในปัจจุบันเครื่องมืออนุมานที่สามารถทำงานร่วมกับจิงนาเอพีไอได้คือเรเซอร์ (Racer) [18] ซึ่งเป็นเครื่องมืออนุมานที่ได้รับการยอมรับว่ามีประสิทธิภาพและทำงานได้รวดเร็ว ผู้วิจัยได้ทดสอบการทำงานของเรเซอร์ร่วมกับ

แผนแบบการค้นหา พบว่าอาร์ดีเอฟทริปเปิลที่ผ่านการอนุมานด้วยเรซเซอร์แตกต่างจากจี้นาเอพีไอ ดังตัวอย่างในรูปที่ 5.1



Jena	Racer
<B><rdfs:subClassOf><A>	<B><rdfs:subClassOf><A>
<C>< rdfs:subClassOf ><A>	<C>< rdfs:subClassOf ><A>
<D>< rdfs:subClassOf ><B>	<D>< rdfs:subClassOf ><B>
<d><rdf:type><D>	<d><rdf:type><D>
<D>< rdfs:subClassOf ><A>	<D>< rdfs:subClassOf ><A>
<d><rdf:type><B>	
<d><rdf:type><A>	

รูปที่ 5.1 ตัวอย่างการอนุมานด้วยจี้นาเทียบกับเรซเซอร์

จากรูปที่ 5.1 อาร์ดีเอฟทริปเปิลที่ได้จากการอนุมานจะแสดงด้วยตัวอักษรสีเทา เมื่อพิจารณาเปรียบเทียบกันพบว่าเรซเซอร์จะไม่อนุมานชนิดของอินสแตนซ์จากลำดับชั้นของคลาส ด้วยเหตุนี้ทำให้ไม่สามารถเปลี่ยนเครื่องมืออนุมานที่แผนแบบการค้นหาใช้ได้โดยตรง เนื่องจากต้องเปลี่ยนขั้นตอนวิธีในการคัดเลือกบริการให้สอดคล้องกับอาร์ดีเอฟทริปเปิลที่ได้จากการอนุมานด้วย

2. ในขั้นตอนคัดเลือกบริการของส่วนค้นหาเชิงความหมาย การจับคู่กลุ่มของคลาสที่ต้องการกับกลุ่มของอินสแตนซ์ของผู้ให้บริการในบางกรณีไม่สามารถทำได้โดยตรงด้วยภาษาอาร์ดีคิวแอล ทำให้การคัดเลือกบริการด้วยภาษาอาร์ดีคิวแอลต้องแบ่งออกเป็นหลายขั้นตอน ในแต่ละขั้นตอนจะตั้งข้อความซึ่งประกอบด้วยเงื่อนไขที่อยู่ในรูปอาร์ดีเอฟทริปเปิลครั้งละ 1 เงื่อนไข การใช้ภาษาอาร์ดีคิวแอลเพื่อคัดเลือกบริการจึงไม่แตกต่างจากการใช้ฟังก์ชันระดับต่ำของจี้นาเอพีไอ
3. สำหรับการสร้างข้อเสนอเชิงพฤติกรรมมิงออนโทโลจีเพื่ออธิบายเว็บเซอร์วิส ในงานวิจัยนี้เสนอให้ใช้เครื่องมือซอฟต์แวร์โพรเทจเพื่อสร้างออนโทโลจีและข้อเสนอเชิงพฤติกรรมในรูปแบบภาษาอวล์ เนื่องจากโพรเทจไม่สามารถประกาศคลาสแบบแรงนับได้โดยตรง แต่ต้องประกาศโดยใช้คุณสมบัติ `equivalentClass` ทำให้ต้องพัฒนาแผนแบบการค้นหา

ให้รองรับการประกาศคลาสแบบแจกแจงโดยใช้คุณสมบัติ `equivalentClass` ด้วย สำหรับ การประกาศทั้งสองแบบจะยกตัวอย่างจากการประกาศคลาสแบบแจกแจง CreditCard ดังรูปที่ 5.2 และรูปที่ 5.3

```
<owl:Class rdf:ID="CreditCard">
  <rdfs:subClassOf rdf:resource="http://www.Ontology.org/UpperOnt/UpperOnt.owl#DataType" />
  <owl:oneOf rdf:parseType="Collection">
    <CreditCard rdf:ID="Amex" />
    <CreditCard rdf:ID="VISA" />
    <CreditCard rdf:ID="MasterCard" />
  </owl:oneOf>
</owl:Class>
```

รูปที่ 5.2 การประกาศคลาสแบบแจกแจง CreditCard ตามปกติ

```
<owl:Class rdf:ID="CreditCard">
  <rdfs:subClassOf rdf:resource="http://www.Ontology.org/UpperOnt/UpperOnt.owl#DataType" />
  <owl:equivalentClass>
    <owl:Class>
      <owl:oneOf rdf:parseType="Collection">
        <CreditCard rdf:ID="Amex" />
        <CreditCard rdf:ID="VISA" />
        <CreditCard rdf:ID="MasterCard" />
      </owl:oneOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>
```

รูปที่ 5.3 การประกาศคลาสแบบแจกแจง CreditCard โดยโพเทเจ

### 5.3 แนวทางการทำวิจัยในอนาคต

จากงานวิจัยนี้ ยังมีประเด็นที่สามารถนำมาศึกษาเพื่อทำวิจัยในอนาคตได้ ดังนี้

1. เพื่อให้โปรแกรมประยุกต์ DiscoveryTemplate สามารถคัดเลือกบริการได้รวดเร็วขึ้น ควรปรับเปลี่ยนขั้นตอนวิธีในการคัดเลือกบริการเพื่อรองรับกับเครื่องมืออณูมานอื่นๆ เช่น เรข เซอร์
2. ในกรณีที่การคัดเลือกบริการโดยส่วนค้นหาเชิงความหมายให้ผลลัพธ์มากกว่าหนึ่งบริการ ควรมีขั้นตอนวิธีในการจัดอันดับบริการเพื่อแสดงผลลัพธ์ที่ตรงกับความต้องการของผู้รับ บริการมากที่สุด
3. สำหรับการหาผลลัพธ์ของการดำเนินการที่มีเงื่อนไขขึ้นกับค่าข้อมูลเข้าบางอย่าง การตัดสินใจค่าความจริงของเงื่อนไขโดยพิจารณาจากค่าข้อมูลเข้าที่มีชนิดเป็นคลาสแบบแจกแจง เพียงอย่างเดียวอาจไม่เพียงพอ จึงควรเพิ่มวิธีตัดสินค่าความจริงของเงื่อนไขจากชนิดข้อ

มูลอื่นๆร่วมด้วย ยกตัวอย่างเช่น ชนิดข้อมูลของอิเล็กทรอนิกส์เ็มแอลสดีมา นอกจากนี้ควรจะพัฒนาให้เงื่อนไขมีค่าความเป็นจริงขึ้นอยู่กับข้อมูลเข้ามากกว่า 1 ตัวได้

4. เนื่องจากการแทรกข้อสนเทศเชิงพฤติกรรมให้กับเอกสารดับเบิลยูเอสดีแอลในงานวิจัยนี้ ยังต้องทำด้วยมือ จึงสามารถพัฒนาเครื่องมือซอฟต์แวร์เพื่อแทรกข้อสนเทศเชิงพฤติกรรมให้กับเอกสารดับเบิลยูเอสดีแอลโดยอัตโนมัติได้



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## รายการอ้างอิง

- [1] Web Service. Available from: <http://www.w3.org/2002/ws/>: [2004, Sep 27]
- [2] UDDI. Available from: <http://www.uddi.org>: [2004, Sep 27]
- [3] Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S. "Web Services Description Language (WSDL) 1.1". Available from: <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>: [March 15, 2001]
- [4] Ontology. Available from: <http://www.semanticweb.org>: [2004, Sep 27]
- [5] Sivashanmugam, K., et al. "Adding Semantics to Web Services Standards". Proceedings of the International Conference on Web Services, ICWS '03, June 23 - 26, 2003.
- [6] Ogbuji, U. "Supercharging WSDL with RDF Managing structured Web service metadata". Available from: <http://www-106.ibm.com/developerworks/library/ws-wsdlrdf/>: [2004, Sep 27]
- [7] Sriharee, N., Senivongse, T. "Discovering Web Services Using Behavioral Constraints and Ontology". DAIS conferences 2003.
- [8] Gruber, T. R. "A translation approach to portable ontologies. Knowledge Acquisition". Knowledge Acquisition, Vol. 5, No. 2, 1993, pp. 199-220.
- [9] Lassila, O., Swick, R., "Resource Description Framework (RDF) Model and Syntax Specification W3C Recommendation". Available from: <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222>: [1999, February 22]
- [10] Daconta, M. C., Obrst, L. J., Smith K. T., "The Semantic Web : A Guide to the Future of XML, Web Services, and Knowledge Management". John Wiley & Sons, June 2003.
- [11] Brickley, D., Guha, R.V. "RDF Vocabulary Description Language 1.0: RDF Schema W3C Working Draft". Available from: <http://www.w3.org/TR/2003/WD-rdf-schema-20031010/>: [2003, October 10]
- [12] Costello, R. L. "Inferring and Discovering Relationships using RDF Schemas". Available from: <http://www.xfront.com/rdf-schema/>: [2004, Sep 27]
- [13] OWL. Available from: <http://www.w3.org>: [2004, Sep 27]
- [14] Protégé. Available from: <http://protege.stanford.edu>: [2004, Sep 27]

- [15] An Introduction to RDF and the Jena RDF API. Available from:  
[http://jena.sourceforge.net/tutorial/RDF\\_API/index.html](http://jena.sourceforge.net/tutorial/RDF_API/index.html): [2004, Sep 27]
- [16] Seaborne, A. "A Programmer's Introduction to RDQL". Available from:  
<http://jena.sourceforge.net/tutorial/RDQL/index.html>: [2004, Sep 27]
- [17] WSDL Editor. Available from: <http://www.capeclear.com>: [2004, Sep 27]
- [18] Racer. Available from: <http://www.cs.concordia.ca/~haarslev/racer/>: [2004, Sep 27]
- [19] OWL-S. Available from: <http://www.daml.org/services/owl-s/>: [2004, Sep 27]



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## ประวัติผู้เขียนวิทยานิพนธ์

นาย ทิวากร โกมุต เกิดเมื่อวันที่ 7 มิถุนายน 2521 ที่กรุงเทพมหานคร สำเร็จการศึกษา  
วิศวกรรมศาสตรบัณฑิต จากภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบัน  
เทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปีการศึกษา 2542 และเข้าศึกษาต่อในหลัก  
สูตรวิทยาศาสตร์มหาบัณฑิต สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ที่ภาควิชาวิศวกรรม  
คอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ. 2545



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย