

การเข้ากันได้ของเว็บไซต์ตามแบบจำลองชนิดบริการเพื่อการทำงานแทนที่กัน



นายเรวัตร เพียรวัฒนผล

สถาบันวิทยบริการ

จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

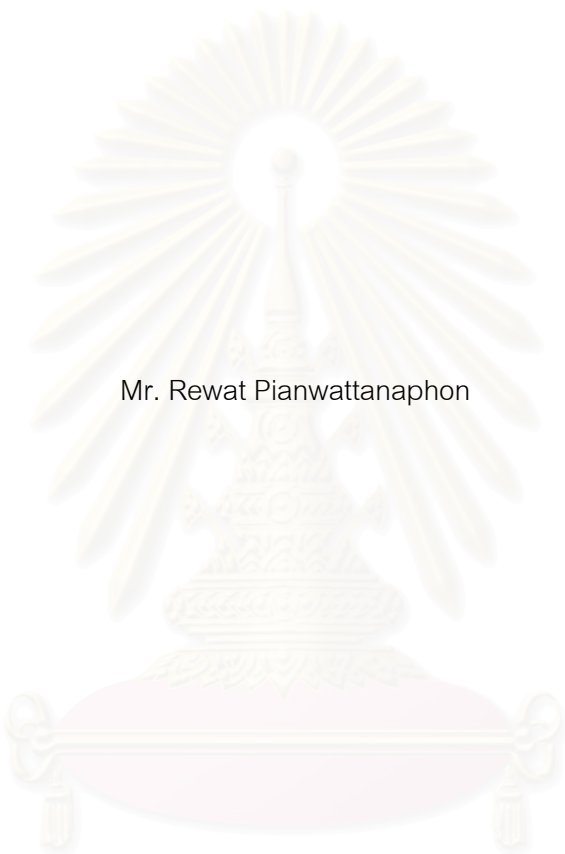
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2549

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

COMPATIBILITY OF WEB SERVICES BY SERVICE TYPE MODEL FOR SUBSTITUTION



Mr. Rewat Pianwattanaphon

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

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

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2006

Copyright of Chulalongkorn University

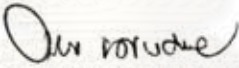
หัวข้อวิทยานิพนธ์	การเข้ากันได้ของเว็บเซอร์วิสตามแบบจำลองชนิดบริการเพื่อการทำงานแทนที่กัน
โดย	นาย เรวัตร์ เพียรวัฒนผล
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.ทวิติย์ เสนีวงศ์ ณ อยุธยา

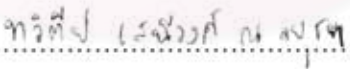
---

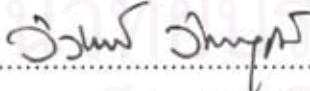
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้บัณฑิตวิทยาลัย จดทะเบียนวิทยานิพนธ์ฉบับนี้เป็นส่วน  
หนึ่งของการศึกษาตามหลักสูตรปริญญาโทมหาบัณฑิต


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

คณะกรรมการสอบวิทยานิพนธ์

  
..... ประธานกรรมการ  
(อาจารย์ ดร.ยรรยง เต็งอำนาจ)

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

  
..... กรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ)

  
..... กรรมการ  
(อาจารย์ ดร.เนตรนภา สีหารี)

เรวัตร์ เพียรวัฒนผล : การเข้ากันได้ของเว็บเซอร์วิสตามแบบจำลองชนิดบริการเพื่อการทำงานแทนที่กัน. (COMPATIBILITY OF WEB SERVICES BY SERVICE TYPE MODEL FOR SUBSTITUTION) อ. ที่ปรึกษา : ผศ. ดร.ทวิติย์ เสนิงวงศ์ ณ อยุธยา, 73 หน้า.

ในการให้บริการเว็บเซอร์วิสนั้น ผู้ให้บริการอาจจะประสบปัญหาระบบของผู้ให้บริการเกิดขัดข้องได้ ดังนั้นจึงมีความจำเป็นที่จะต้องค้นหาเว็บเซอร์วิสอื่นและทำการเรียกใช้บริการแทน แต่เนื่องจากการประกาศคำอธิบายบริการในโครงสร้างของยูดีดีไอนั้น แต่ละเว็บเซอร์วิสจะระบุประเภทของการให้บริการผ่านทางแอททริบิวต์แคทากอรี ซึ่งไม่ได้มีข้อกำหนดว่าเว็บเซอร์วิสที่มีแคทากอรีเดียวกัน จะมีรูปแบบของการให้บริการเหมือนกันหรือทำงานแทนกันได้ ดังนั้นการค้นหาเว็บเซอร์วิสที่มีความเข้ากันได้ จึงต้องทำการพิจารณาคำอธิบายบริการที่ประกาศไว้ที่ยูดีดีไออย่างละเอียดเพื่อให้มั่นใจว่าเว็บเซอร์วิสตัวหนึ่ง จะเข้ากันได้กับเว็บเซอร์วิสอีกตัวหนึ่งและสามารถทำงานแทนได้

แนวทางหนึ่งที่ช่วยในการค้นหาเว็บเซอร์วิสที่เข้ากันได้คือ การประยุกต์ใช้แนวคิดของแบบจำลองชนิดบริการซึ่งประกอบด้วยชนิดบริการและข้อเสนอบริการ ชนิดบริการจะกำหนดรูปแบบการให้บริการของเซอร์วิสประเภทหนึ่งๆ ส่วนข้อเสนอบริการจะหมายถึงคำอธิบายบริการของเว็บเซอร์วิสแต่ละตัวภายใต้ชนิดบริการ แนวคิดนี้มีใช้ในมาตรฐานระบบกระจาย เช่น บริการเทรเดอริโนอาร์เอ็ม-ไอดีพี และคอร์บา

วิทยานิพนธ์นี้ประยุกต์แนวคิดแบบจำลองชนิดบริการมาใช้กับเว็บเซอร์วิส เพื่อใช้ในการกำหนดรูปแบบการให้บริการของเว็บเซอร์วิสที่อยู่ในประเภทเดียวกัน โดยกำหนดในรูปของข้อมูลเชิงซิกเนเจอร์ ข้อมูลเชิงความหมายซึ่งแสดงถึงรูปแบบพฤติกรรมของการให้บริการ และแอททริบิวต์เชิงความหมาย อันจะทำให้การค้นหาเว็บเซอร์วิสที่มีสภาพเข้ากันได้และทำงานแทนกันได้นั้น สามารถทำได้สะดวกขึ้นโดยพิจารณาความสัมพันธ์ระหว่างเว็บเซอร์วิสผ่านชนิดบริการ ผู้วิจัยได้นำเสนอกฎการเข้ากันได้ของเว็บเซอร์วิส พร้อมกับนำเสนอเฟรมเวิร์คของการทำงานแทนที่กันโดยอัตโนมัติของเว็บเซอร์วิส เพื่อให้ผู้ใช้บริการยังคงรับบริการต่อได้จากเว็บเซอร์วิสตัวแทนโดยไม่ต้องเปลี่ยนแปลงคำร้องขอใช้บริการ

ภาควิชา.....วิศวกรรมคอมพิวเตอร์..... ลายมือชื่อนิสิต..... 151๓5 เพียรวัฒนผล.....  
สาขาวิชา.....วิศวกรรมคอมพิวเตอร์..... ลายมือชื่ออาจารย์ที่ปรึกษา..... ทวิติย์ เสนิงวงศ์ ณ อยุธยา.....  
ปีการศึกษา.....2549.....

# # 4870442021 : MAJOR COMPUTER ENGINEERING

KEY WORD: WEB SERVICES / COMPATIBILITY / SERVICE TYPE MODEL / SIGNATURE / SEMANTICS / SUBSTITUTION

REWAT PIANWATTANAPHON: COMPATIBILITY OF WEB SERVICES BY SERVICE TYPE MODEL FOR SUBSTITUTION. THESIS ADVISOR : ASST. PROF. TWITTIE SENIVONGSE, Ph.D., 73 pp.

At Web Services invocation, Web Services may fail and cannot provide the service to consumers. Therefore, it is necessary that other Web Services are discovered and invoked instead. Web Services are generally published with UDDI registry under particular service categories via the attribute "category". However, there is no guarantee that services published under the same service category will have the same capability. Finding compatible Web Services then requires a thorough examination on published service descriptions to assure service compatibility and substitutability.

One approach that facilitates finding compatible Web Services is by applying the concept of service type model. Service type model consists of service types and service offers. A service type defines capabilities of a service of a particular type. A service offer refers to a published description of a service instance under a particular service type. The concept is the basis of service models in other standards such as RM-ODP and CORBA.

This thesis applies to Web Services the concept of service type model that can be used in describing capabilities of Web Services of the same service category in terms of their signature, behaviour, and semantic attributes. Finding compatible substitute services will be more convenient by determining relationships between Web Services through their service types. Service type compatibility rules and a framework for automatic Web Services substitution are proposed so that service consumers can continue using the services from the substitutes without having to change service invocations.

Department.....Computer Engineering..... Student's Signature...*Rewat Pianwattanaphon*  
Field of study.....Computer Engineering..... Advisor's Signature...*Twittie Senivongse*  
Academic year.....2006.....

## กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงด้วยความกรุณาเป็นอย่างสูงของผู้ช่วยศาสตราจารย์ ดร.ทวีชัย เสนีวงศ์ ณ อยุธยา อาจารย์ที่ปรึกษาวิทยานิพนธ์ของข้าพเจ้า ที่สละเวลาให้การอบรมสั่งสอน อีกทั้งยังให้คำปรึกษาที่ดีในการตัดสินใจทำสิ่งต่างๆ มาโดยตลอด หากมีสิ่งใดที่ข้าพเจ้าทำผิดพลาดไปข้าพเจ้ากราบขออภัย และกราบขอบพระคุณอาจารย์เป็นอย่างสูงไว้ ณ ที่นี้ด้วย

ขอขอบพระคุณ อาจารย์ ดร.ยรรยง เต็งอำนวยการ ประธานกรรมการสอบวิทยานิพนธ์ ผศ.ดร.วิวัฒน์ วัฒนาวุฒิ และ อาจารย์ ดร.เนตรนภา สีหารี กรรมการสอบวิทยานิพนธ์ของข้าพเจ้าที่กรุณาให้คำแนะนำต่างๆ ทำให้วิทยานิพนธ์นี้มีความถูกต้องและสมบูรณ์มากขึ้น

ขอขอบคุณพี่ ๆ ที่ห้องปฏิบัติการวิศวกรรมระบบสารสนเทศ สำหรับคำปรึกษาที่ดีในทุกๆ ด้าน รวมทั้งกำลังใจและความช่วยเหลืออื่นๆ ที่มอบให้มาโดยตลอด

ขอขอบคุณนายอิศเรศ บุญทรง และนายสุรศิริ นพภัณฑ์ รุ่นพี่สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ที่คอยให้คำปรึกษาในระหว่างการทำงานเสมอมา

ขอขอบคุณเพื่อน ๆ ทุกคน ที่คอยช่วยเหลือกันมาโดยตลอด ทำให้ชีวิตการเรียนปริญญาโทของข้าพเจ้าไม่รู้สึกลำบากเลย

ท้ายที่สุดนี้ ขอกราบขอบพระคุณ คุณแม่ และคุณป้าทุกคน ที่คอยอบรมสั่งสอนให้กำลังใจ และสนับสนุนข้าพเจ้ามาโดยตลอด คุณความดีของวิทยานิพนธ์ฉบับนี้ขอมอบเป็นเครื่องบูชาไว้ ณ โอกาสนี้

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 .....	2
1.3 ขอบเขตของการวิจัย.....	3
1.4 ขั้นตอนการวิจัย .....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ .....	4
1.6 ผลงานตีพิมพ์ .....	4
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง .....	5
2.1 แนวคิดและทฤษฎี.....	5
2.1.1 ดับเบิลยูเอสดีแอล (WSDL: Web Services Description Language) .....	5
2.1.2 ออนโทโลยี (Ontology) .....	7
2.1.3 อวอลล์-เอส (OWL-Based Web Service Ontology).....	8
2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง .....	10
2.2.1 แบบจำลองชนิดบริการของโอดีพี .....	10
2.2.2 An Approach to Web Service Compatibility in Cooperative Processes .....	10
2.2.3 The Capability Matching of Web Services .....	11
2.2.4 Matchmaking and Ranking of Semantic Web Services Using Integrated Service Profile .....	12
บทที่ 3 แบบจำลองชนิดบริการและกฎการเข้ากันได้ของเว็บเซอร์วิส .....	14
3.1 การอธิบายบริการโดยใช้แบบจำลองชนิดบริการ.....	14
3.1.1 ชนิดบริการ.....	14

3.1.2 การสร้างข้อกำหนดชนิดบริการ .....	15
3.1.3 อับเปอร์ออนโทโลยีสำหรับกำหนดความหมายให้กับชนิดบริการ.....	18
3.1.4 ออนโทโลยีสำหรับชนิดบริการ .....	19
3.1.5 ข้อเสนอบริการ .....	20
3.1.6 การใช้งานแบบจำลองชนิดบริการ .....	20
3.2 กฎการเข้ากันได้ของเว็บเซอร์วิสตามแบบจำลองชนิดบริการ.....	24
3.2.1 การเข้าคู่เชิงซิกเนเจอร์ .....	25
3.2.2 การเข้าคู่เชิงความหมาย .....	27
3.2.3 การเข้าคู่เชิงข้อมูลเฉพาะตัว.....	29
3.3 ตัวอย่างการพิจารณาสภาพเข้ากันได้และการทำงานแทนที่กันของเว็บเซอร์วิส .....	30
บทที่ 4 เฟรมเวิร์คสำหรับการทำงานแทนที่กันโดยอัตโนมัติของเว็บเซอร์วิส .....	34
4.1 การออกแบบเฟรมเวิร์คสำหรับการทำงานแทนที่กันโดยอัตโนมัติของเว็บเซอร์วิส .....	34
4.1.1 ขั้นตอนการสร้างชนิดบริการ .....	34
4.1.2 ขั้นตอนการประกาศข้อเสนอบริการ .....	37
4.1.3 ขั้นตอนการร้องขอใช้บริการเว็บเซอร์วิสผ่านตัวกลาง .....	39
4.1.4 ขั้นตอนการส่งต่อการร้องขอใช้บริการเว็บเซอร์วิสโดยตัวกลาง.....	40
4.1.5 ขั้นตอนการส่งผลลัพธ์กลับไปให้ผู้ให้บริการ .....	41
4.2 การออกแบบส่วนเข้าคู่.....	41
4.2.1 ส่วนเข้าคู่เชิงซิกเนเจอร์.....	41
4.2.2 ส่วนเข้าคู่เชิงความหมาย .....	42
4.2.3 ส่วนเข้าคู่เชิงข้อมูลเฉพาะตัว.....	44
4.3 เครื่องมือที่ใช้ในการพัฒนา .....	45
บทที่ 5 การทดสอบการทำงานของเฟรมเวิร์ค .....	46
5.1 การทดสอบการทำงานของเฟรมเวิร์คด้วยกรณีศึกษาที่ 1 .....	46
5.2 การทดสอบการทำงานของเฟรมเวิร์คด้วยกรณีศึกษาที่ 2 .....	50
5.3 การทดสอบการทำงานของเฟรมเวิร์คด้วยกรณีศึกษาที่ 3 .....	54
5.4 สรุปผลการทดสอบ.....	59
บทที่ 6 สรุปผลการวิจัย .....	60



6.1 สรุปผลการวิจัย.....	60
6.2 ปัญหาและข้อจำกัดที่พบจากการวิจัย .....	60
6.3 ข้อเสนอแนะ .....	62
รายการอ้างอิง.....	63
ภาคผนวก.....	65
ภาคผนวก ก ผลงานตีพิมพ์ .....	66
ประวัติผู้เขียนวิทยานิพนธ์ .....	73



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## สารบัญตาราง

ญ

ตาราง	หน้า
ตารางที่ 3.1 การเปรียบเทียบเงื่อนไขก่อนการทำงานและผลกระทบของการเรียกใช้งานโอเพอเรชั่น bookPurchase .....	18
ตารางที่ 3.2 การเปรียบเทียบข้อมูลในส่วน Operation Specification ระหว่างชนิดบริการ BookSeller กับ BookStore .....	31
ตารางที่ 3.3 ผลการจับคู่ระหว่างชนิดบริการ BookSeller กับ BookStore .....	32
ตารางที่ 3.4 ตัวอย่างข้อมูลเฉพาะตัวของเว็บเซอวิส Y และเว็บเซอวิส X .....	33
ตารางที่ 5.1 ข้อมูลของเว็บเซอวิสที่นำมาทดสอบในกรณีศึกษาที่ 1 .....	46
ตารางที่ 5.2 ข้อมูลเฉพาะตัวของเว็บเซอวิส PacificBook ในกรณีศึกษาที่ 1 .....	46
ตารางที่ 5.3 ข้อมูลเฉพาะตัวของเว็บเซอวิส CUBook ในกรณีศึกษาที่ 1 .....	47
ตารางที่ 5.4 ข้อมูลเฉพาะตัวของเว็บเซอวิส SUTBook ในกรณีศึกษาที่ 1 .....	47
ตารางที่ 5.5 ข้อมูลของเว็บเซอวิสที่นำมาทดสอบในกรณีศึกษาที่ 2 .....	50
ตารางที่ 5.6 ข้อมูลเฉพาะตัวของเว็บเซอวิส PacificBook ในกรณีศึกษาที่ 2 .....	50
ตารางที่ 5.7 ข้อมูลเฉพาะตัวของเว็บเซอวิส CUBook ในกรณีศึกษาที่ 2 .....	50
ตารางที่ 5.8 ข้อมูลเฉพาะตัวของเว็บเซอวิส IndianBook ในกรณีศึกษาที่ 2 .....	51
ตารางที่ 5.9 ข้อมูลของเว็บเซอวิสที่นำมาทดสอบในกรณีศึกษาที่ 3 .....	54
ตารางที่ 5.10 ข้อมูลเฉพาะตัวของเว็บเซอวิส PacificBook ในกรณีศึกษาที่ 3 .....	54
ตารางที่ 5.11 ข้อมูลเฉพาะตัวของเว็บเซอวิส CUBook ในกรณีศึกษาที่ 3 .....	55
ตารางที่ 5.12 ข้อมูลเฉพาะตัวของเว็บเซอวิส IndianBook ในกรณีศึกษาที่ 3 .....	55
ตารางที่ 5.13 ข้อมูลเฉพาะตัวของเว็บเซอวิส AlaskaBook ในกรณีศึกษาที่ 3 .....	55

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

ภาพประกอบ	หน้า
รูปที่ 2.1 ตัวอย่างเอกสารฉบับเบ็ดเตล็ด ..... 5	5
รูปที่ 2.2 ความสัมพันธ์ของวัตถุภายในออนไลน์ Book ..... 8	8
รูปที่ 2.3 โครงสร้างของอาร์ค-เอส ..... 9	9
รูปที่ 3.1 ตัวอย่างข้อกำหนดชนิดบริการสำหรับผู้ให้บริการขายหนังสือ..... 16	16
รูปที่ 3.2 การสร้างข้อกำหนดชนิดบริการด้วยวิธีการขยาย ..... 17	17
รูปที่ 3.3 คลาสและคุณสมบัติในแอปเปอร้ออนโทโลยีสำหรับกำหนดความหมายให้กับชนิดบริการ ..... 18	18
รูปที่ 3.4 ออนไลน์สำหรับชนิดบริการขายหนังสือ ..... 21	21
รูปที่ 3.5 เอกสารฉบับเบ็ดเตล็ดที่ใช้ในการประกาศข้อเสนอบริการของเว็บเซอร์วิสชื่อ Pacific ภายใต้ชนิดบริการขายหนังสือ ..... 22	22
รูปที่ 3.6 เอกสารอาร์ค-เอสที่ใช้ในการประกาศข้อเสนอบริการของเว็บเซอร์วิสชื่อ Pacific ภายใต้ ชนิดบริการขายหนังสือ ..... 23	23
รูปที่ 3.7 ภาพรวมแสดงแนวคิดของแบบจำลองชนิดบริการสำหรับเว็บเซอร์วิสขายหนังสือ..... 24	24
รูปที่ 3.8 โครงสร้างของวัน-เวย์โอเปอเรชัน ..... 25	25
รูปที่ 3.9 โครงสร้างของรีควีสท์-เรสปอนซ์โอเปอเรชัน ..... 25	25
รูปที่ 3.10 หมวดหมู่ของหนังสือ (Taxonomy of Book) ..... 28	28
รูปที่ 3.11 ชนิดบริการ BookSeller ที่นำมาพิจารณาสภาพเข้ากันได้..... 30	30
รูปที่ 3.12 ชนิดบริการ BookStore ที่นำมาพิจารณาสภาพเข้ากันได้ ..... 31	31
รูปที่ 3.13 ออนไลน์สำหรับชนิดบริการที่ใช้อธิบายความสัมพันธ์ของสองชนิดบริการ ..... 32	32
รูปที่ 4.1 เฟรมเวิร์คการทำงานแทนที่กันของเว็บเซอร์วิสที่เข้ากันได้ตามแบบจำลองชนิดบริการ 34	34
รูปที่ 4.2 ส่วนของเฟรมเวิร์คในขั้นตอนการสร้างชนิดบริการ ..... 35	35
รูปที่ 4.3 ตัวอย่างการสร้างข้อกำหนดชนิดบริการ BookSeller..... 36	36
รูปที่ 4.4 หน้าต่างสอบถามการเพิ่มโอเปอเรชัน ..... 36	36
รูปที่ 4.5 การระบุชื่อชนิดบริการที่ต้องการนำมาขยายข้อกำหนด ..... 37	37
รูปที่ 4.6 ตัวอย่างการสร้างข้อกำหนดชนิดบริการด้วยการขยายจากข้อกำหนดที่มีอยู่เดิม ..... 37	37
รูปที่ 4.7 ส่วนของเฟรมเวิร์คในขั้นตอนการประกาศข้อเสนอบริการ ..... 38	38
รูปที่ 4.8 ตัวอย่างการประกาศข้อเสนอบริการ..... 39	39
รูปที่ 4.9 ตัวอย่างข้อความชีพ ..... 39	39

ภาพประกอบ	หน้า
รูปที่ 4.10 ส่วนของเฟรมเวิร์คในขั้นตอนการส่งต่อการร้องขอใช้บริการเว็บเซอร์วิส .....	40
รูปที่ 4.11 ตัวอย่างผลลัพธ์ .....	41
รูปที่ 4.12 ลำดับการสืบต่อของชนิดข้อมูล.....	42
รูปที่ 4.13 โครงสร้างการทำงานส่วนเข้าคู่เชิงความหมาย .....	43
รูปที่ 4.14 โครงสร้างการทำงานส่วนเข้าคู่เชิงข้อมูลเฉพาะตัว .....	44
รูปที่ 5.1 ข้อกำหนดชนิดบริการ BookSeller.....	47
รูปที่ 5.2 ผลลัพธ์ของการทดสอบฝั่งตัวกลางในกรณีศึกษาที่ 1 .....	48
รูปที่ 5.3 ผลลัพธ์ของการทดสอบฝั่งผู้ให้บริการในกรณีศึกษาที่ 1.....	48
รูปที่ 5.4 ชนิดบริการ BookAndMovieSeller.....	51
รูปที่ 5.5 ผลลัพธ์ของการทดสอบฝั่งตัวกลางในกรณีศึกษาที่ 2 .....	52
รูปที่ 5.6 ผลลัพธ์ของการทดสอบฝั่งผู้ให้บริการในกรณีศึกษาที่ 2.....	52
รูปที่ 5.7 ข้อกำหนดชนิดบริการ BookShop .....	55
รูปที่ 5.8 ข้อกำหนดชนิดบริการ BookStore .....	56
รูปที่ 5.9 ผลลัพธ์ของการทดสอบฝั่งตัวกลางในกรณีศึกษาที่ 3 .....	56
รูปที่ 5.10 ผลลัพธ์ของการทดสอบฝั่งผู้ให้บริการในกรณีศึกษาที่ 3.....	57

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

เว็บเซอร์วิส (Web Service) [1], [2] คือระบบซอฟต์แวร์ที่ถูกออกแบบเพื่อสนับสนุนการดำเนินงานร่วมกันระหว่างเครื่องคอมพิวเตอร์กับเครื่องคอมพิวเตอร์ โดยติดต่อกันผ่านเครือข่าย เช่น เชนที่ทีพี (HTTP) ซึ่งเว็บเซอร์วิสนั้นจะมีส่วนต่อประสาน (Interface) ที่อธิบายรูปแบบกระบวนการทำงานคือ เอกสารดับเบิลยูเอสดีแอล (WSDL: Web Service Description Language) [3] ระบบอื่นๆ จะติดต่อกับเว็บเซอร์วิสโดยใช้โพรโทคอลโซฟ (SOAP: Simple Object Access Protocol) [4] ซึ่งส่งข้อความอยู่ในรูปของเอกสารเอ็กซ์เอ็มแอล (XML: Extensible Markup Language) ทำให้เว็บเซอร์วิสไม่ขึ้นกับระบบปฏิบัติการหรือภาษาโปรแกรมใดๆ

เว็บเซอร์วิสอ้างอิงจากสถาปัตยกรรมเอสโอเอ ซึ่งผู้ให้บริการ (Service Provider) จะมาประกาศบริการไว้ที่ยูดีดีไอ (UDDI: Universal Description, Discovery and Integration) [5] และผู้ใช้บริการ (Service Consumer) ก็จะมาค้นหาบริการที่ต้องการผ่านยูดีดีไอและทำการเรียกใช้บริการ แต่ในการใช้บริการนั้น ผู้ใช้บริการอาจจะประสบปัญหาต่างๆ ขึ้นได้ เช่น

- ระบบของผู้ให้บริการเกิดขัดข้อง
- ผู้ให้บริการอาจจะเปลี่ยนแปลงรายละเอียดบริการของตน โดยที่ผู้ใช้บริการเดิมไม่ทราบ
- ผู้ให้บริการยกเลิกการให้บริการนั้น

จะเห็นได้ว่าเมื่อเกิดปัญหาขึ้นไม่ว่ากรณีใดจะทำให้เกิดผลเสียต่อผู้ใช้บริการที่เสียโอกาสที่จะได้ใช้บริการ และเสียเวลาในการค้นหาบริการอื่นมาใช้แทน ดังนั้นการค้นหาเว็บเซอร์วิสที่เข้ากันได้โดยอัตโนมัติเป็นคำตอบหนึ่งสำหรับปัญหานี้

แนวทางหนึ่งในการค้นหาเว็บเซอร์วิสที่เข้ากันได้คือ การใช้แบบจำลองชนิดบริการ (Service Type Model) แนวคิดของแบบจำลองชนิดบริการ หมายถึงการกำหนดรูปแบบการให้บริการสำหรับงานประเภทหนึ่งๆ โดยผู้ให้บริการแต่ละรายที่อยู่ในประเภทเดียวกันจะมีการประกาศรูปแบบการให้บริการ (หรือที่เรียกว่า ข้อเสนอบริการ (Service Offer)) ที่เหมือนกัน เช่น ผู้ให้บริการแต่ละรายในประเภทร้านขายหนังสือ ก็จะให้บริการตรวจสอบรายการหนังสือหรือสั่งซื้อหนังสือในทำนองเดียวกัน แนวคิดนี้มีใช้ในมาตรฐานระบบกระจาย เช่น บริการเทรเดอริโนอาร์เอ็ม-โอดีพี (RM-ODP) [6], [7] และ คอร์บา (CORBA) [8] รวมทั้งบริการลูคัฟในจินี (Jini) [9] เพื่อกำหนดความสัมพันธ์ระหว่างชนิดบริการที่ต่างชนิดกัน และระหว่างข้อเสนอบริการที่มีชนิดบริการเดียวกัน กล่าวคือชนิดบริการที่สืบทอด (Inherit) ลักษณะของชนิดบริการอื่น จะสามารถผู้ใช้

แทนชนิดบริการแม่ที่ถูกสืบทอดมาได้ และข้อเสนอบริการหนึ่งๆ จะสามารถใช้แทนข้อเสนอบริการที่มีชนิดบริการเดียวกัน หรือใช้แทนข้อเสนอบริการที่มีชนิดบริการเป็นบริการแม่ได้

ปัจจุบันเว็บเซอร์วิสไม่มีแนวคิดของแบบจำลองชนิดบริการ คือทุกเว็บเซอร์วิสเทียบได้กับข้อเสนอบริการ แต่ละเว็บเซอร์วิสจะระบุประเภทผ่านทางแคเทกอรี (Category) ในโครงสร้างของยูติลิตี้ แต่ไม่ได้มีข้อกำหนดว่าเว็บเซอร์วิสที่มีแคเทกอรีเดียวกัน จะมีรูปแบบการให้บริการเหมือนกันหรือทำงานแทนกันได้ ดังนั้นการค้นหาเว็บเซอร์วิสที่มีความเข้ากันได้เพื่อประโยชน์ในการทำงานแทนกัน จึงต้องพิจารณาคำอธิบายบริการที่ประกาศไว้ที่ยูติลิตี้อย่างละเอียด เพื่อให้มั่นใจว่าเว็บเซอร์วิสตัวหนึ่ง จะเข้ากันได้กับเว็บเซอร์วิสอีกตัวหนึ่งและสามารถทำงานแทนได้

งานวิจัยนี้มีแนวคิดที่จะประยุกต์แบบจำลองชนิดบริการมาใช้กับเว็บเซอร์วิส เพื่อใช้ในการกำหนดรูปแบบการให้บริการและคุณสมบัติของเว็บเซอร์วิสที่อยู่ในประเภทเดียวกัน ซึ่งจะทำให้การค้นหาเว็บเซอร์วิสที่มีสภาพเข้ากันได้และทำงานแทนกันได้นั้น สามารถทำได้สะดวกขึ้น โดยการพิจารณาความสัมพันธ์ระหว่างชนิดบริการของเว็บเซอร์วิสเหล่านั้น อย่างไรก็ตามแบบจำลองชนิดบริการในมาตรฐานระบบกระจายที่กล่าวถึงมาแล้วนั้น จะกำหนดรูปแบบการให้บริการในรูปของส่วนต่อประสาน (ซิกเนเจอร์ของโอเปอเรชันที่ให้บริการ) และแอททริบิวต์ ซึ่งแสดงลักษณะของบริการชนิดนั้นๆ แต่การพิจารณาส่วนต่อประสานและแอททริบิวต์นั้น ไม่สามารถแสดงถึงความหมายที่แท้จริงของการให้บริการได้ดีนัก เช่น ในโอเปอเรชัน "bookPurchase" คำว่า "book" อาจจะหมายถึงหนังสือประเภทนวนิยายอย่างเดียวกันนั้นก็ได้นั้น นอกเหนือจากรายละเอียดเชิงซิกเนเจอร์แล้ว ผู้วิจัยจึงมีแนวคิดที่จะเพิ่มรายละเอียดเชิงความหมาย ซึ่งแสดงรูปแบบพฤติกรรมของการให้บริการและแอททริบิวต์เชิงความหมายไว้ในแบบจำลองชนิดบริการด้วย นอกจากนี้ผู้วิจัยจะนำเสนอเฟรมเวิร์ค (Framework) ของการทำงานแทนที่กันของเว็บเซอร์วิส โดยอาศัยการพิจารณาสภาพเข้ากันได้ของเว็บเซอร์วิสจากแบบจำลองชนิดบริการของเว็บเซอร์วิสเหล่านั้น เพื่อให้ผู้ใช้บริการสามารถเรียกใช้เว็บเซอร์วิสตัวแทนได้อย่างอัตโนมัติหากเกิดปัญหาในการเรียกใช้เว็บเซอร์วิสที่ต้องการ

## 1.2 วัตถุประสงค์ของการวิจัย

- 1.2.1 เพื่อออกแบบแบบจำลองชนิดบริการและข้อเสนอบริการสำหรับเว็บเซอร์วิส รวมทั้งกฎการเข้ากันได้ตามแบบจำลองชนิดบริการ
- 1.2.2 เพื่อออกแบบและพัฒนาเฟรมเวิร์คที่รองรับการทำงานแทนกันโดยอัตโนมัติของเว็บเซอร์วิสที่เข้ากันได้ตามแบบจำลองชนิดบริการ

### 1.3 ขอบเขตของการวิจัย

- 1.3.1 ออกแบบแบบจำลองชนิดบริการและข้อเสนอบริการ ซึ่งประกอบด้วย 3 ส่วนคือ
- ส่วนข้อมูลเชิงซิกเนเจอร์ กำหนดด้วยเอกสารดับเบิลยูเอสดีแอล ซึ่งเป็นไปตามข้อกำหนดของดับเบิลยูเอสดีแอล รุ่น 1.1
  - ส่วนข้อมูลเชิงพฤติกรรม กำหนดด้วยเอกสารอาร์เอส ซึ่งเป็นไปตามข้อกำหนดของอาร์เอส รุ่น 1.1
  - ส่วนแอททริบิวต์เชิงความหมาย กำหนดด้วยเอกสารอาร์เอส ซึ่งเป็นไปตามข้อกำหนดของอาร์เอส รุ่น 1.1
- 1.3.2 กำหนดกฎการเข้ากันได้ของเว็บเซอร์วิสตามแบบจำลองชนิดบริการ โดยแบ่งออกเป็น 3 ส่วนคือ
- การเข้าคู่เชิงซิกเนเจอร์
  - การเข้าคู่เชิงความหมายของพฤติกรรมและแอททริบิวต์
  - การเข้าคู่เชิงข้อมูลเฉพาะตัว
- 1.3.3 การกำหนดเงื่อนไขก่อนการดำเนินการหรือผลกระทบในส่วนข้อมูลเชิงพฤติกรรมสามารถระบุได้หลายตัว โดยถือว่ามีผลเทียบเท่าการแอนด์ (And) กันและแต่ละตัวสามารถกำหนดรายละเอียดในระดับข้อเสนอบริการได้ตามรูปแบบนิพจน์เชิงเปรียบเทียบ (Relational Expression) ซึ่งมีตัวกระทำได้แก่  $>$ ,  $<$ ,  $=$ ,  $\leq$ ,  $\geq$
- 1.3.4 งานวิจัยนี้จะไม่พิจารณาการจัดลำดับ (Ranking) การเข้ากันได้ของเว็บเซอร์วิส
- 1.3.5 ออกแบบและพัฒนาเฟรมเวิร์คการทำงานแทนกันของเว็บเซอร์วิส โดยพิจารณาจากกฎการเข้ากันได้ของเว็บเซอร์วิส
- 1.3.6 เฟรมเวิร์คจะทำการค้นหาเว็บเซอร์วิสที่สามารถเข้ากันได้ พร้อมทั้งปรับการเรียกใช้ไปยังเว็บเซอร์วิสตัวแทนโดยอัตโนมัติ ในกรณีที่เว็บเซอร์วิสที่ต้องการใช้งานเกิดขัดข้อง
- 1.3.7 เฟรมเวิร์ครองรับผู้ให้บริการที่สร้างเว็บเซอร์วิสด้วยซอฟต์แวร์อาพาเซ่แอ็กซิส (Apache Axis) และผู้ใช้บริการที่เรียกใช้บริการด้วยภาษาจาวา

### 1.4 ขั้นตอนการวิจัย

- 1.4.1 ศึกษาเทคโนโลยีเว็บเซอร์วิส
- 1.4.2 สำรวจและศึกษาแนวคิดในการพิจารณาหาเว็บเซอร์วิสที่สามารถเข้ากันได้จากงานวิจัยอื่น

- 1.4.3 สำรองและศึกษาการเพิ่มข้อมูลเชิงพฤติกรรมให้กับเว็บเซอร์วิสรวมถึงกฎที่ใช้ในการพิจารณาสภาพเข้ากันได้จากงานวิจัยอื่น
- 1.4.4 ศึกษาข้อกำหนดของเอกสารดับเบิลยูเอสดีแอลและข้อกำหนดของอาร์เอส-เอส
- 1.4.5 ออกแบบรูปแบบของชนิดบริการ ข้อเสนอบริการ และกฎที่ใช้ในการพิจารณาสภาพเข้ากันได้ของเว็บเซอร์วิส
- 1.4.6 วิเคราะห์โครงสร้างและออกแบบเฟรมเวิร์คสำหรับค้นหาเว็บเซอร์วิสที่สามารถเข้ากันได้
- 1.4.7 พัฒนาเฟรมเวิร์คที่ออกแบบ
- 1.4.8 ทดสอบและปรับปรุงแก้ไขการทำงานของเฟรมเวิร์คที่พัฒนาขึ้น
- 1.4.9 สรุปผลการวิจัย และจัดทำรายงานวิทยานิพนธ์

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

จะได้แนวคิดในการพิจารณาหาเว็บเซอร์วิสที่สามารถเข้ากันได้ โดยใช้รูปแบบของชนิดบริการที่มีรายละเอียดครอบคลุมทั้งข้อมูลเชิงซิกเนเจอร์ ข้อมูลเชิงพฤติกรรม และแอททริบิวต์เชิงความหมาย รวมทั้งได้ต้นแบบเฟรมเวิร์คที่ช่วยในการค้นหาเว็บเซอร์วิสที่สามารถเข้ากันได้ พร้อมทั้งปรับการเรียกใช้ไปยังเว็บเซอร์วิสตัวแทนนั้นโดยอัตโนมัติ ในกรณีที่เว็บเซอร์วิสที่ต้องการใช้งานเกิดขัดข้อง ทำให้ผู้ใช้บริการยังคงรับบริการต่อได้โดยไม่ต้องเปลี่ยนแปลงคำร้องขอใช้บริการ

## 1.6 ผลงานตีพิมพ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้ตีพิมพ์และนำเสนอในการประชุมวิชาการดังนี้

1. The 9th International Conference on Advanced Communication Technology (ICAICT 2007), pp. 76-81. 12-14 February 2007, Phoenix Park, Gangwon-Do, Republic of Korea ในบทความเรื่อง Compatibility by Service Type Model for Automatic Web Services Substitution โดยผู้แต่งคือ Rewat Pianwattanaphon และ Twittie Senivongse



## บทที่ 2

### ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

#### 2.1 แนวคิดและทฤษฎี

##### 2.1.1 ดับเบิลยูเอสดีแอล (WSDL: Web Services Description Language)

ดับเบิลยูเอสดีแอล [3] เป็นเอกสารที่ใช้โครงสร้างพื้นฐานของภาษาเอ็กซ์เอ็มแอล เพื่ออธิบายรายละเอียดของเว็บเซอร์วิสที่เกี่ยวกับการติดต่อบริการผ่านเครือข่าย การดำเนินการที่บริการรองรับ รวมถึงข้อความที่เกี่ยวข้องกับการเรียกใช้บริการซึ่งสามารถส่งผ่านโพรโทคอลโซฟระหว่างผู้รับบริการกับผู้ให้บริการได้ โดยโครงสร้างของเอกสารดับเบิลยูเอสดีแอลนั้นจะประกอบด้วย 7 อิลิเมนต์ (Element) แบ่งออกเป็น 2 กลุ่ม ดังรูปที่ 2.1

```
<?xml version="1.0"?>
<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/">

  <types>
    <schema targetNamespace="http://example.com/stockquote.xsd"
      xmlns="http://www.w3.org/2000/10/XMLSchema">
      <element name="TradePriceRequest">
        <complexType>
          <all>
            <element name="tickerSymbol" type="string"/>
          </all>
        </complexType>
      </element>
      <element name="TradePrice">
        <complexType>
          <all>
            <element name="price" type="float"/>
          </all>
        </complexType>
      </element>
    </schema>
  </types>
```

รูปที่ 2.1 ตัวอย่างเอกสารดับเบิลยูเอสดีแอล [3]

<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>
<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>
<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>
<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 ตัวอย่างเอกสารดับเบิลยูเอสดีแอล [3] (ต่อ)

- คำอธิบายเชิงนามธรรม (Abstract Description) ประกอบด้วยส่วนย่อยที่อธิบายถึงความสามารถของเว็บเซอร์วิส ดังนี้
  - <wsdl:types> คือ ข้อกำหนดชนิดข้อมูล (Data Type) ทั้งหมดที่ใช้อธิบายข้อความที่แลกเปลี่ยนกันระหว่างผู้ให้บริการและผู้ใช้บริการ ซึ่งจะใช้นิยามข้อมูลพื้นฐาน (Primitive Data Type) ตามข้อกำหนดของเอ็กซ์เอ็มแอลสคีมา (XML Schema) [10] เป็นค่าโดยปริยาย (Default)

- <wsdl:message> คือ การกำหนดข้อมูลที่จะใช้ในการสื่อสารระหว่างผู้ให้บริการและผู้ให้บริการซึ่งภายใน 1 การดำเนินการนั้นอนุญาตให้มีได้เพียง 3 ข้อความ ดังนี้
    - Input Message คือ ข้อความเข้า แสดงผ่านส่วนย่อยชื่อ <wsdl:input>
    - Output Message คือ ข้อความออก แสดงผ่านส่วนย่อยชื่อ <wsdl:output>
    - Fault Message คือ ข้อความที่แสดงความผิดปกติ แสดงผ่านส่วนย่อยชื่อ <wsdl:fault>
  - <wsdl:operation> คือ การดำเนินการที่เว็บเซอร์วิสสนับสนุน
  - <wsdl:portType> คือ กลุ่มของการดำเนินการทั้งหมดที่เว็บเซอร์วิสสนับสนุน
2. คำอธิบายเชิงรูปธรรม (Concrete Description) ประกอบด้วยส่วนย่อยที่อธิบายถึงการผูกมัดผู้ให้บริการกับบริการ ดังนี้
- <wsdl:binding> คือ ส่วนเชื่อมต่อกันระหว่างคำอธิบายเชิงนามธรรมกับคำอธิบายเชิงรูปธรรม โดยระบุข้อมูลโพรโทคอล และอ้างอิงข้อกำหนดรูปแบบข้อมูลสำหรับพอร์ตไทป์โดยเฉพาะ
  - <wsdl:port> คือ ยูอาร์แอล (URL) ที่ใช้สำหรับเรียกใช้เว็บเซอร์วิส โดย 1 พอร์ต (port) หมายถึง 1 ช่องทางในการติดต่อเว็บเซอร์วิส
  - <wsdl:service> ได้แก่ กลุ่มของพอร์ตทั้งหมดที่แสดงอยู่ในเอกสารดับเบิลยูเอสดีแอล

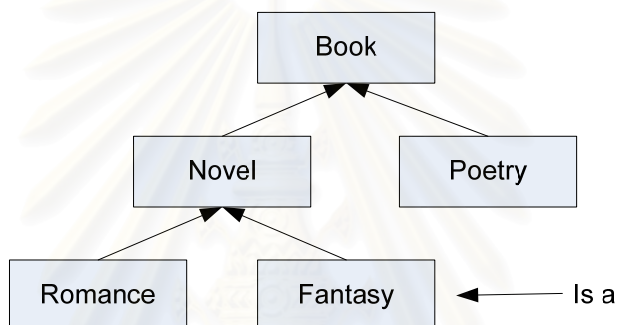
ในงานวิจัยนี้ จะใช้เอกสารดับเบิลยูเอสดีแอลแสดงรายละเอียดข้อมูลเชิงซิกเนเจอร์ของแบบจำลองชนิดบริการซึ่งอธิบายรายละเอียดเกี่ยวกับโอเปอเรชันของเว็บเซอร์วิส

### 2.1.2 ออนโทโลยี (Ontology)

ในสาขาวิทยาการคอมพิวเตอร์นั้น ออนโทโลยีคือแบบจำลองข้อมูลที่อธิบายแนวคิดของโดเมนในรูปของวัตถุภายในโดเมนนั้น และความสัมพันธ์ระหว่างสิ่งเหล่านั้น [11] โดยทั่วไปออนโทโลยีจะอธิบาย

- อินดิวิดูอล (Individuals) เป็นส่วนประกอบพื้นฐานของออนโทโลยี โดยอินดิวิดูอลจะแทนวัตถุภายในโดเมนที่เราสนใจ เช่น คน สัตว์ สิ่งของ
- คลาส (Classes) คือ กลุ่มของวัตถุที่ประกอบด้วยอินดิวิดูอล เช่น คลาส Book จะหมายถึง กลุ่มของหนังสือที่เราสนใจ

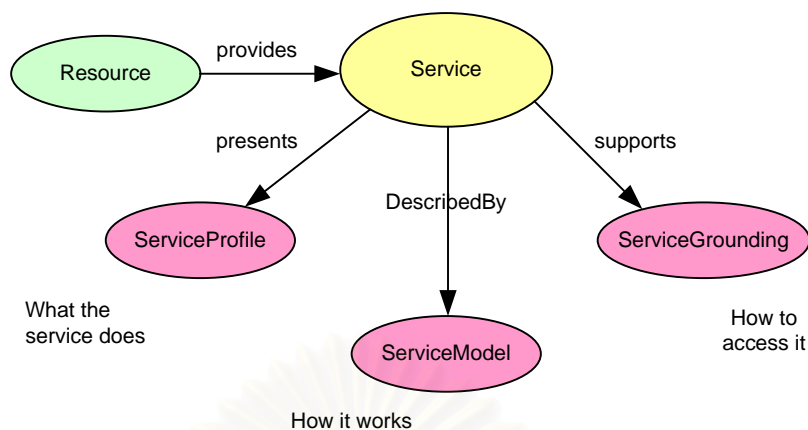
- แอททริบิวต์ (Attributes) คือ คุณลักษณะที่ใช้อธิบายวัตถุในออนโทโลยี ซึ่งอย่างน้อยจะประกอบด้วยชื่อและค่า ที่จะผูกไว้กับวัตถุเสมอเช่น วัตถุชื่อ Harry Potter จะมีคุณลักษณะดังนี้
  - Name: Harry Potter
  - Author: J. K. Rowling
- ความสัมพันธ์ (Relations) คือ การอธิบายความสัมพันธ์ระหว่างคลาสในออนโทโลยี ดังรูปที่ 2.2 คลาสชื่อ Romance จะเป็นซับคลาส (Subclass) ของคลาสชื่อ Novel



รูปที่ 2.2 ความสัมพันธ์ของวัตถุภายในออนโทโลยี Book

### 2.1.3 อวาล์-เอส (OWL-Based Web Service Ontology)

อวาล์-เอส [12] (เดิมคือดีเอเอ็มแอล-เอส: DAML-S) เป็นภาษาออนโทโลยีสำหรับเว็บเชิงความหมายที่พัฒนาต่อมาจากภาษาอวาล์ (OWL: the Ontology Web Language) [13] ซึ่งใช้สำหรับอธิบายคุณสมบัติและความสามารถของเว็บเซอร์วิส โดยจะแบ่งรายละเอียดออกเป็น 3 ส่วนคือ เซอร์วิสโพรไฟล์ (Service Profile) โพรเซสโมเดล (Process Model) และเซอร์วิสกราวนดิ้ง (Service Grounding) ดังรูปที่ 2.3



รูปที่ 2.3 โครงสร้างของอวาล์-เอส

1. เซอร์วิสโพรไฟล์ จะอธิบายรายละเอียดทั่วไปของเว็บเซอร์วิสที่ใช้เกี่ยวข้องกับการประกาศและการค้นหาบริการ โดยเซอร์วิสโพรไฟล์จะแบ่งคุณสมบัติออกเป็น 2 กลุ่มคือ
  - คุณสมบัติเชิงฟังก์ชัน (Functional Properties) เช่น ข้อมูลเข้า (Inputs) ข้อมูลออก (Outputs) เงื่อนไขก่อนการทำงาน (Preconditions) และผลกระทบ (Effect) ซึ่งข้อมูลส่วนนี้จะสืบต่อมาจาก (Derived) ส่วนโพรเซสโมเดล
  - คุณสมบัติที่ไม่เป็นฟังก์ชัน (Nonfunctional Properties) เช่น ชื่อบริการ (ServiceName) สารสนเทศเกี่ยวกับผู้ให้บริการ (Provider Information) และประเภทของบริการ (Service Category)
2. โพรเซสโมเดล จะอธิบายว่าบริการนี้จะทำงานอย่างไร โดยจะอธิบายลักษณะการทำงานของบริการในรูปของพฤติกรรมเชิงฟังก์ชัน (Functional Behaviour) กระบวนการภายในของบริการ (Internal Processes of The Service) และกระแสนงาน (Workflow)
3. เซอร์วิสกราวนด์อิง จะอธิบายถึงรายละเอียดของการติดต่อกับบริการ โดยจะมีการจับคู่ (Mapping) ระหว่างข้อกำหนดเชิงนามธรรมของอวาล์-เอสและข้อกำหนดเชิงรูปธรรมของเว็บเซอร์วิส เช่น ดับเบิลยูเอสดีแอล

โดยงานวิจัยนี้ จะนำอวาล์-เอสในส่วนของเซอร์วิสโพรไฟล์และโพรเซสโมเดลมาใช้ในการอธิบายข้อมูลเชิงพฤติกรรมกับแอททริบิวต์เชิงความหมายของแบบจำลองชนิดบริการ เพื่อให้ประโยชน์ในการพิจารณาสภาพเข้ากันได้ของเว็บเซอร์วิสต่อไป

## 2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง

### 2.2.1 แบบจำลองชนิดบริการของโอดีพี

โอดีพี (ODP: Open Distributed Processing) เป็นเฟรมเวิร์คสำหรับระบบกระจาย ที่มีเป้าหมายเพื่อการใช้บริการร่วมกันผ่านเครือข่ายที่ต่างกันหรือระบบปฏิบัติการที่ต่างกัน ซึ่งอาร์เอ็ม-โอดีพีและคอร์บาอ้างอิงนำไปใช้งาน โดยโอดีพีนั้นมีกลไกที่เรียกว่าเทรดเดอร์ (Trader) [6] คอยประสานระหว่างผู้ใช้บริการและผู้ให้บริการในระบบกระจาย โดยมีแนวคิดที่สำคัญ คือ แบบจำลองชนิดบริการและข้อเสนอบริการ ซึ่งมีรายละเอียด ดังนี้

- แบบจำลองชนิดบริการ คือการกำหนดรูปแบบการให้บริการ ซึ่งประกอบด้วย
    - ชนิดส่วนต่อประสาน ซึ่งจะแสดงซิกเนเจอร์ของส่วนต่อประสานของบริการ
    - ชนิดคุณสมบัติ (Property Types) จะแสดงถึงพฤติกรรม ข้อมูลที่ไม่เป็นฟังก์ชัน (Non-Functional) และข้อมูลที่ไม่เกี่ยวกับการคำนวณ (Non-Computational)
- ในแนวคิดของชนิดบริการนั้น ชนิดบริการที่ต่างชนิดกันสามารถที่จะสัมพันธ์กันได้คือ ชนิดบริการหนึ่งอาจจะสืบทอดชนิดส่วนต่อประสานและชนิดคุณสมบัติจากชนิดบริการอื่นมาได้ ซึ่งทำให้บริการที่อยู่ภายใต้ชนิดบริการที่สืบทอดมาสามารถใช้แทนบริการที่อยู่ภายใต้ชนิดบริการแม่ที่ถูกสืบทอดมาได้
- ข้อเสนอบริการ คือ คำอธิบายเกี่ยวกับบริการที่ผู้ให้บริการมาประกาศไว้ ซึ่งประกอบด้วย
    - ชื่อของชนิดบริการ
    - ส่วนต่อประสานที่อ้างอิง
    - ค่าคุณสมบัติสำหรับบริการ

โดยข้อเสนอบริการที่มีชนิดบริการเดียวกัน จะสามารถทำงานแทนกันได้เนื่องจากส่วนต่อประสานและคุณสมบัติมีรูปแบบเดียวกัน แต่ทั้งนี้ค่าคุณสมบัติก็ต้องสอดคล้องกันด้วย จากแนวคิดที่กล่าวมา ผู้วิจัยจึงนำแนวคิดนี้มาประยุกต์ใช้ในการออกแบบลักษณะของแบบจำลองชนิดบริการ สำหรับอธิบายพฤติกรรมการให้บริการของเว็บเซอร์วิส และนำไปใช้ในการพิจารณาสภาพเข้ากันได้ของเว็บเซอร์วิส

### 2.2.2 An Approach to Web Service Compatibility in Cooperative Processes

งานวิจัย [14] ได้นำเสนอแบบจำลองสำหรับวิเคราะห์สภาพเข้ากันได้เพื่อรองรับการแทนที่กันของเว็บเซอร์วิส โดยแนวคิดคือจะกำหนดให้เว็บเซอร์วิสที่ใช้อยู่เป็นบริการเชิงนามธรรม (Abstract Service) โดยรายละเอียดนั้นจะถูกอธิบายโดยนำข้อมูลจากเอกสารดับเบิลยูเอสดี

แอลมาสร้างเป็นแบบจำลองที่เรียกว่า ตัวบรรยายบริการ (Service Descriptor) ซึ่งอธิบายข้อมูลเข้าและข้อมูลออกของบริการ ส่วนเว็บเซอร์วิสที่ประกาศบริการไว้ในยูดีดีไอจะเรียกว่า บริการเชิงรูปธรรม (Concrete Service) ในกรณีที่เกิดปัญหาขึ้นกับเว็บเซอร์วิสที่เรียกใช้บริการอยู่ในขณะนั้น ก็จะทำการวิเคราะห์บริการเชิงรูปธรรมว่ามีฟังก์ชันการทำงานตรงกับบริการเชิงนามธรรมที่เกิดขัดข้องหรือไม่ โดยการวิเคราะห์จะแบ่งออกเป็น 2 ส่วน ดังนี้

- การวิเคราะห์ในระดับบน (High Level Analysis) ในส่วนนี้จะวิเคราะห์ชื่อโอเปอเรชัน ข้อมูลเข้าและข้อมูลออกในเชิงความหมาย โดยใช้ตัวบรรยายบริการ แล้วนำผลลัพธ์ที่ได้ซึ่งเป็นค่าตัวเลขที่แสดงถึงสภาพเข้ากันได้มาจัดเป็นรายการเรียกว่า คลาสสภาพเข้ากันได้ (Compatibility Class)
- การวิเคราะห์ในระดับโครงสร้าง (Structural Analysis) ส่วนนี้จะนำเว็บเซอร์วิสที่อยู่ในคลาสสภาพเข้ากันได้ มาทำการวิเคราะห์ข้อมูลเชิงโครงสร้าง ได้แก่ ชื่อโอเปอเรชัน ชื่อพารามิเตอร์และชนิดข้อมูล ถ้าเว็บเซอร์วิสใดผ่านการวิเคราะห์ในส่วนนี้ก็จะสร้างข้อมูลแมปปิง (Mapping Information) ที่จำเป็นสำหรับเรียกใช้เว็บเซอร์วิสใหม่

ผู้วิจัยพบว่าข้อจำกัดของงานวิจัยนี้คือ แบบจำลองของงานวิจัยนี้ใช้ข้อมูลเชิงซิกเนเจอร์ในการวิเคราะห์เพียงด้านเดียวเท่านั้น ผู้วิจัยจึงมีแนวคิดที่จะเพิ่มเติมข้อมูลเชิงพฤติกรรมเข้ามาใช้ในการวิเคราะห์ด้วย เพื่อให้ครอบคลุมถึงพฤติกรรมการทำงานของเว็บเซอร์วิส

### 2.2.3 The Capability Matching of Web Services

งานวิจัย [15] ได้เสนอภาษาชื่อ เอสซีดีแอล (SCDL) ขึ้นมาอธิบายความสามารถของเว็บเซอร์วิสเพื่อความสะดวกในการพิจารณาการเข้าคู่ความสามารถระหว่างสองเว็บเซอร์วิส โดยข้อกำหนดของภาษาเอสซีดีแอลนี้จะประกอบไปด้วยโครงสร้างส่วนต่างๆ ดังนี้คือ

- Name คือ ชื่อของความสามารถของเว็บเซอร์วิส
- OntoDescription คือ คำจำกัดความของเว็บเซอร์วิส และรายละเอียดเชิงความหมาย
- Type คือ ชนิดข้อมูล
- Input คือ การประกาศตัวแปรของข้อมูลเข้า
- Output คือ การประกาศตัวแปรของข้อมูลออก
- Pre-Condition and Pre-Constraints คือ เงื่อนไขที่ต้องเป็นจริงก่อนการทำงาน
- Post-Condition and Post-Constraints คือ เงื่อนไขที่เป็นจริงหลังการทำงาน

ส่วนการเข้าคู่ความสามารถของเว็บเซอร์วิสโดยใช้รายละเอียดที่อยู่ในรูปของภาษาเอสซีดีแอลนั้นจะมีขั้นตอนดังนี้

1. OntoDescription Matching การจับคู่นี้จะทำเป็นขั้นตอนแรก โดยมีจุดประสงค์คือเปรียบเทียบความสัมพันธ์เกี่ยวกับคำจำกัดความและรายละเอียดเชิงความหมายของสองเว็บเซอร์วิส
2. Syntactical Matching จะพิจารณาสyntaxของข้อมูลเข้าและข้อมูลออก
3. Semantic Matching เป็นขั้นตอนสุดท้ายของการพิจารณา โดยจะใช้เงื่อนไขที่ต้องเป็นจริงก่อนและหลังการทำงานมาพิจารณา

งานวิจัยนี้พิจารณาการเข้าคู่กันระหว่างเว็บเซอร์วิสโดยใช้ภาษาที่กำหนดขึ้นมาใหม่สำหรับอธิบายแต่ละเว็บเซอร์วิส ทำให้เกิดความไม่เป็นมาตรฐานต่อการนำไปใช้งานจริง อย่างไรก็ตามงานวิจัยนี้มีแนวคิดที่คล้ายกับงานของผู้วิจัยในการพิจารณาทั้งซิกเนเจอร์และความหมายของเว็บเซอร์วิสในการพิจารณาสภาพเข้ากันได้ แต่งานวิจัยนี้ทำการพิจารณาในระดับข้อเสนอบริการคือนำคำอธิบายบริการของสองเว็บเซอร์วิสในภาษาเอสดีดีแอลมาเปรียบเทียบกัน ดังนั้นการค้นหาเว็บเซอร์วิสที่เข้ากันได้จะยุ่งยาก ในขณะที่งานของผู้วิจัยจะนำแนวคิดเกี่ยวกับชนิดบริการมาประยุกต์ใช้ ซึ่งจะทำให้การค้นหาเว็บเซอร์วิสที่เข้ากันได้ทำได้สะดวกกว่า อีกทั้งการกำหนดชนิดบริการจะอ้างอิงกับมาตรฐานต่างๆ ที่เกี่ยวกับคำอธิบายบริการของเว็บเซอร์วิส

#### 2.2.4 Matchmaking and Ranking of Semantic Web Services Using Integrated Service Profile

งานวิจัย [16] นำเสนอแบบจำลองคำอธิบายเว็บเซอร์วิสเชิงความหมายเรียกว่า เซอร์วิสโพรไฟล์รวม (Integrated Service Profile) ซึ่งอธิบายความสามารถในการให้บริการของเซอร์วิสในรูปของโครงสร้างการให้บริการ พฤติกรรมของเซอร์วิสแบบมีกฎ รวมไปถึงคุณลักษณะอย่างง่ายและคุณลักษณะเชิงความหมายของเซอร์วิส และนำเสนออัลกอริทึมในการจับคู่บริการกับความต้องการของผู้ใช้บริการ โดยผลการค้นหาจับคู่นั้นนำมาจัดลำดับ (Ranking) ความใกล้เคียงกับความต้องการด้วย โดยเซอร์วิสโพรไฟล์รวมประกอบด้วย 2 ส่วนคือ

1. สารสนเทศเชิงแอททริบิวต์ (Attribute-Based Information) คือ กลุ่มของแอททริบิวต์ที่ใช้อธิบายเว็บเซอร์วิสที่สนใจ ซึ่งเป็นข้อมูลอย่างง่าย เช่น ข้อมูลผู้ให้บริการ
2. สารสนเทศเชิงความสามารถ (Capability-Based Information) คือ ลักษณะของเว็บเซอร์วิสที่ซับซ้อนขึ้น โดยแบ่งออกเป็น 3 แอททริบิวต์ ดังนี้
  - โครงสร้างของบริการ (Service Structure) แสดงโครงสร้างความรู้ (Knowledge Structure) ของเว็บเซอร์วิส ซึ่งผู้ให้บริการคาดว่าจะรู้ก่อนการตัดสินใจใช้บริการ เช่น ผลลัพธ์ของบริการรายละเอียดการขาย และการส่งของ



- พฤติกรรมของบริการ (Service Behaviour) แสดงสารสนเทศที่เกี่ยวกับพฤติกรรมของเว็บเซอร์วิส เช่น ต้องการข้อมูลเข้าบางอย่างเพื่อสั่งซื้อสินค้า
- เงื่อนไขของบริการ (Service Constraint) แสดงเงื่อนไขของบริการในรูปของกฎสำหรับอัลกอริทึมในการจับคู่ นั้น จะพิจารณาระหว่างเซอร์วิสโพรไฟล์พร้อมกับข้อความ (Query) ของผู้ใช้บริการ โดยจะแบ่งการจับคู่ออกเป็น 6 แบบคือ

- การจับคู่แนวคิดเชิงความหมาย (Matching Ontological Concepts)
- การจับคู่เงื่อนไขเชิงตัวเลข (Matching Numerical Constraints)
- การจับคู่กลุ่มของค่าเชิงความหมาย (Matching Sets of Ontological Values)
- การจับคู่เงื่อนไขของบริการ (Matching Service Constraints)
- การจับคู่โพรไฟล์พฤติกรรม (Matching Behavioural Profiles)
- การจับคู่แอตทริบิวต์อย่างง่าย (Matching Simple Attributes)

งานวิจัยนี้ยังได้นำเสนอการประเมินอัลกอริทึมที่ใช้ในการจับคู่และการจัดลำดับเซอร์วิส นอกจากนี้ยังนำเสนอเฟรมเวิร์คของกระบวนการค้นหาด้วยเซอร์วิสโพรไฟล์รวมอีกด้วย

ผู้วิจัยจะใช้อัลกอริทึมการจับคู่แนวคิดเชิงความหมายของงานวิจัยนี้ในการพิจารณาการจับคู่เชิงพฤติกรรมของแบบจำลองชนิดบริการ

## บทที่ 3

### แบบจำลองชนิดบริการและกฎการเข้ากันได้ของเว็บเซอร์วิส

แนวคิดในการออกแบบแบบจำลองชนิดบริการสำหรับเว็บเซอร์วิสนี้ได้ประยุกต์มาจากแบบจำลองชนิดบริการของระบบกระจายอื่นๆ เช่น บริการเทร็ดเดอริ์ในอาร์เอ็มไอดีพีและคอร์บา โดยมีการเพิ่มเติมลักษณะที่สำคัญของบริการในแง่มุมต่างๆ เพื่อให้ครอบคลุมและเป็นประโยชน์ต่อการพิจารณาค้นหาเว็บเซอร์วิสที่สามารถทำงานแทนกันได้ โดยรายละเอียดเกี่ยวกับแนวคิดของแบบจำลองชนิดบริการจะนำเสนอในหัวข้อที่ 3.1 ส่วนการพิจารณานั้นผู้วิจัยได้นำเสนอกฎการเข้ากันได้ของเว็บเซอร์วิสตามแบบจำลองชนิดบริการซึ่งจะอยู่ในหัวข้อที่ 3.2 และตัวอย่างการพิจารณาสภาพเข้ากันได้เพื่อการทำงานแทนที่กันของเว็บเซอร์วิสจะนำเสนอในหัวข้อที่ 3.3

#### 3.1 การอธิบายบริการโดยใช้แบบจำลองชนิดบริการ

แนวคิดของแบบจำลองชนิดบริการนี้จะใช้ประโยชน์ในการอธิบายความสัมพันธ์ของเว็บเซอร์วิส โดยจะประกอบด้วยส่วนสำคัญ 2 ส่วนคือ ชนิดบริการ และข้อเสนอบริการ ซึ่งมีรายละเอียดดังนี้

##### 3.1.1 ชนิดบริการ

ชนิดบริการคือ ข้อกำหนดพฤติกรรมการให้บริการของแต่ละเว็บเซอร์วิส โดยจะอธิบายพฤติกรรมการให้บริการของเว็บเซอร์วิสและแบ่งออกเป็น 3 ส่วนคือ

1. ส่วนข้อมูลเชิงซิกเนเจอร์ จะอธิบายถึงรายละเอียดของโอเปอเรชันทั้งหมดที่เว็บเซอร์วิสนั้นจะให้บริการ ได้แก่ โอเปอเรชัน (Operation) จำนวนพารามิเตอร์ (Number of Parameter) ชนิดของพารามิเตอร์ (Parameter Type) ชนิดของผลลัพธ์ที่จะคืนกลับเมื่อสิ้นสุดการทำงาน (Return Type) และความผิดพลาดที่อาจจะเกิดขึ้น (Fault)
2. ส่วนข้อมูลเชิงพฤติกรรม จะอธิบายถึงรายละเอียดของแต่ละโอเปอเรชันในเชิงของความหมาย ได้แก่ ข้อมูลเข้า (Input) ข้อมูลออก (Output) เงื่อนไขก่อนการดำเนินงาน (Precondition) ผลกระทบ (Effect) และความผิดพลาด (Fault) ซึ่งแบ่งการอธิบายออกเป็น 2 ระดับ คือ อับเปอร์ออนโทโลยี (Upper Ontology) (หัวข้อที่ 3.1.3) และออนโทโลยีสำหรับชนิดบริการ (Service Type Ontology) (หัวข้อที่ 3.1.4)
3. ส่วนแอททริบิวต์เชิงความหมาย (Semantic Attribute) ซึ่งจะอธิบายถึงคุณลักษณะเชิงความหมายซึ่งเกี่ยวกับการให้บริการของเว็บเซอร์วิสที่ไม่เกี่ยวข้องกับข้อมูลในส่วนที่ 1 และ 2 เช่น แบบจำลองชนิดบริการสายการบิน อาจจะมีแอททริบิวต์เชิง

ความหมายเป็น “ข้อสายการบิน” สำหรับให้เว็บเซอร์วิสแต่ละรายภายใต้ชนิดบริการนี้ใช้ในการระบุข้อสายการบินตามอินเทอร์เน็ตโลจี้สำหรับชนิดบริการสายการบิน

การมีข้อกำหนดชนิดบริการนั้น ทำให้รูปแบบของการประกาศเว็บเซอร์วิสเปลี่ยนไป ซึ่งแนวคิดเดิมของการประกาศเว็บเซอร์วิสนั้น ผู้ให้บริการแต่ละรายก็จะลงทะเบียนไว้ที่ผู้ดีไอทีที่เดียว และแนวคิดในเรื่องของชนิดบริการมีการแสดงข้อมูลผ่านทางแอททริบิวต์ชื่อ Category เท่านั้น แต่ข้อมูลนี้ไม่สามารถบอกถึงสภาพเข้ากันได้ของเว็บเซอร์วิสเลย การใช้เพียงแอททริบิวต์นี้ ผู้ใช้บริการจะรู้แต่เพียงว่าเว็บเซอร์วิสใดอยู่ในประเภทเดียวกันบ้าง แต่จะไม่สามารถรู้ได้เลยว่าพฤติกรรมในการให้บริการของแต่ละเว็บเซอร์วิสนั้นเป็นอย่างไร สามารถเข้ากันได้หรือไม่ แต่การมีข้อกำหนดชนิดบริการนั้น จะต้องมีการสร้างข้อกำหนดขึ้นมาเป็นแม่แบบก่อน ซึ่งผู้ที่สร้างนั้น อาจจะเป็นผู้ให้บริการหรือผู้เชี่ยวชาญในโดเมน (Domain Expert) จากนั้นผู้ให้บริการแต่ละรายก็ต้องสร้างเว็บเซอร์วิสขึ้นมาโดยมีรายละเอียดของบริการตามที่แสดงไว้ในข้อกำหนดชนิดบริการ แล้วผู้ให้บริการก็ต้องมาลงทะเบียนประกาศข้อเสนอบริการ (Service Offer) เพื่อบอกว่า เว็บเซอร์วิสของตนมีพฤติกรรมการทำงานที่สอดคล้องกับชนิดบริการใด ซึ่งข้อมูลเหล่านี้จะถูกนำไปใช้ประโยชน์ในการหาสภาพเข้ากันได้ของเว็บเซอร์วิสต่อไป

### 3.1.2 การสร้างข้อกำหนดชนิดบริการ

การสร้างข้อกำหนดชนิดบริการนั้นสามารถทำได้ 2 วิธีคือ

#### 1. การสร้างข้อกำหนดชนิดบริการขึ้นมาใหม่ทั้งหมด

ข้อกำหนดชนิดบริการนั้น ประกอบด้วยรายละเอียดของส่วนต่างๆ ดังนี้

- Service Type: คือส่วนที่แสดงชื่อของชนิดบริการ
- Operation Specification: คือส่วนที่แสดงข้อกำหนดเชิงซิกเนเจอร์ของโอเปอเรชันทั้งหมด ส่วนนี้เปรียบเสมือนการกำหนดแม่แบบของส่วนต่อประสานหรือพอร์ตโทพีในเอกสารดับเบิลยูเอสดีแอลสำหรับบริการชนิดหนึ่งๆ
- Behaviour Specification: คือส่วนที่แสดงข้อกำหนดเชิงพฤติกรรมของแต่ละโอเปอเรชัน โดยแต่ละโอเปอเรชันจะมีเพียง 1 Behaviour Specification เท่านั้น ข้อกำหนดนี้จะแสดงความหมายของส่วน Operation Specification โดยใช้ออนโทโลจี้
- Semantic Attribute Specification: คือส่วนที่แสดงแอททริบิวต์เชิงความหมายที่เกี่ยวข้องกับการให้บริการของเว็บเซอร์วิส โดยแต่ละข้อกำหนดชนิดบริการจะมีเพียง 1 Semantic Attribute Specification เท่านั้น แต่สามารถระบุแอททริบิวต์เชิงความหมายได้หลายตัว

## ตัวอย่างข้อกำหนดชนิดบริการแสดงดังรูปที่ 3.1

1.	Service Type: BookSeller
2.	Operation Specification:
3.	int bookPurchase(int isbn, String creditcardDetail);
4.	fault ExpCreditCard;
5.	Behaviour Specification of bookPurchase:
6.	Input: ISBN, CreditcardDetail;
7.	Output: PurchasedBook;
8.	Precondition: VisaCardHolder;
9.	Effect: ProductDelivered;
10.	Fault: ExpiredCreditCard;
11.	Semantic Attribute Specification: BookCategory;

รูปที่ 3.1 ตัวอย่างข้อกำหนดชนิดบริการสำหรับผู้ให้บริการขายหนังสือ

## 2. การขยายข้อกำหนดชนิดบริการที่มีอยู่เดิม

ถ้าชนิดบริการที่ต้องการสร้างขึ้นนั้นมีพฤติกรรมในการทำงานที่ใกล้เคียงกับข้อกำหนดที่มีอยู่เดิม ก็สามารถที่จะเลือกอ้างอิงข้อกำหนดนั้นมาเพื่อแก้ไขหรือเพิ่มเติมรายละเอียดใหม่เข้าไปได้ แต่ต้องเป็นไปตามกฎการขยาย (Extension Rules) ที่กำหนดไว้ ดังนี้

1. ส่วน Operation Specification จะไม่สามารถทำการแก้ไขข้อมูลของโอเปอเรชันที่ได้รับมาจากการสืบทอด (Derive) ได้เลย แต่สามารถเพิ่มโอเปอเรชันใหม่เข้าไปได้ โดยเมื่อเพิ่มโอเปอเรชันแล้วจะต้องทำการเพิ่มข้อมูลในส่วนของ Behaviour Specification ของโอเปอเรชันนี้ด้วย โดยลักษณะของการขยายข้อกำหนดในส่วนนี้จะสอดคล้องกับการสืบทอดส่วนต่อประสาน (Interface Inheritance) [17] ของดับเบิลยูเอสดีแอล รุ่น 2.0 ที่กำลังพัฒนาอยู่ ซึ่งมีแอททริบิวต์ชื่อ extends ซึ่งอนุญาตให้ส่วนต่อประสานสามารถขยายหรือสืบทอด (Inherit) จากส่วนต่อประสานอื่นๆ ได้
2. ส่วนของ Behaviour Specification จะสามารถแก้ไขข้อมูลที่สืบทอดมาได้เฉพาะในส่วนของการแก้ไขก่อนการทำงานกับผลกระทบเท่านั้น แต่ข้อมูลที่แก้ไขนั้นจะต้องสอดคล้องกับข้อมูลเดิมตามการเข้าคู่เชิงความหมาย (หัวข้อที่ 3.2.2) ด้วย และการจะเพิ่มเติมข้อมูลของ Behaviour Specification นั้น จะสามารถทำได้ในกรณีเดียวคือ มีการเพิ่มโอเปอเรชันขึ้นมาเท่านั้น
3. ส่วนของ Semantic Attribute Specification จะสามารถทำการแก้ไขข้อมูลที่สืบทอดมาได้ แต่ต้องเป็นไปตามกฎการเข้าคู่เชิงความหมายเช่นเดียวกัน นอกจากนี้ยังสามารถเพิ่มข้อมูลแอททริบิวต์เชิงความหมายตัวใหม่ได้

โดยการขยายข้อกำหนดชนิดบริการนี้สามารถทำการขยายจากหลายๆ ชนิดบริการได้ และในกรณีที่โอเปอเรชันในข้อกำหนดชนิดบริการนั้นๆ มีข้อมูลเชิงซิกเนเจอร์ที่ตรงกัน ก็จะทำให้การ

ยุบรวมเหลือเพียงโอเปอเรชันเดียว ซึ่งการขยายข้อกำหนดในลักษณะนี้จะสอดคล้องกับการสืบทอดส่วนต่อประสาน [17] ของดับเบิลยูเอสดีแอล รุ่น 2.0 ที่อนุญาตให้สามารถทำการสืบทอดจากหลายส่วนต่อประสานได้ (Multiple Interface)

ข้อกำหนดชนิดบริการที่ทำการขยายออกมานั้น จะได้รับการสืบทอดพฤติกรรมการให้บริการทั้งหมดจากข้อกำหนดชนิดบริการที่ได้อ้างอิงด้วย โดยจะมีคำว่า ExtensionOf เป็นคำสำคัญ (Keyword) ที่ต่อท้ายชื่อข้อกำหนดเสมอ ดังตัวอย่างในรูปที่ 3.2

1.	Service Type: BookAndMovieSeller ExtensionOf BookSeller
2.	Operation Specification:
3.	int bookPurchase(int isbn, String creditcardDetail);
4.	fault ExpCreditCard;
5.	int moviePurchase(String mName, String creditcardDetail);
6.	fault ExpCreditCard, MovieNotFound;
7.	boolean moviePreview(String mName);
8.	fault FileNotFound;
9.	Behaviour Specification of bookPurchase:
10.	Input: ISBN, CreditcardDetail;
11.	Output: PurchasedBook;
12.	Precondition: CreditCardHolder;
13.	Effect: ProductDeliveredByAir;
14.	Fault: ExpiredCreditCard;
15.	Behaviour Specification of moviePurchase:
16.	Input: MovieName, CreditcardDetail;
17.	Output: PurchasedMovie;
18.	Precondition: CreditCardHolder;
19.	Effect: ProductDeliveredByAir;
20.	Fault: ExpiredCreditCard, MovieNotFound;
21.	Behaviour Specification of moviePreview:
22.	Input: MovieName;
23.	Output: MovieClip;
24.	Precondition: MPEG-4Player;
25.	Effect: HandBillImages;
26.	Fault: FileNotFound;
27.	Semantic Attribute Specification: BookCategory, MovieCategory;

รูปที่ 3.2 การสร้างข้อกำหนดชนิดบริการด้วยวิธีการขยาย

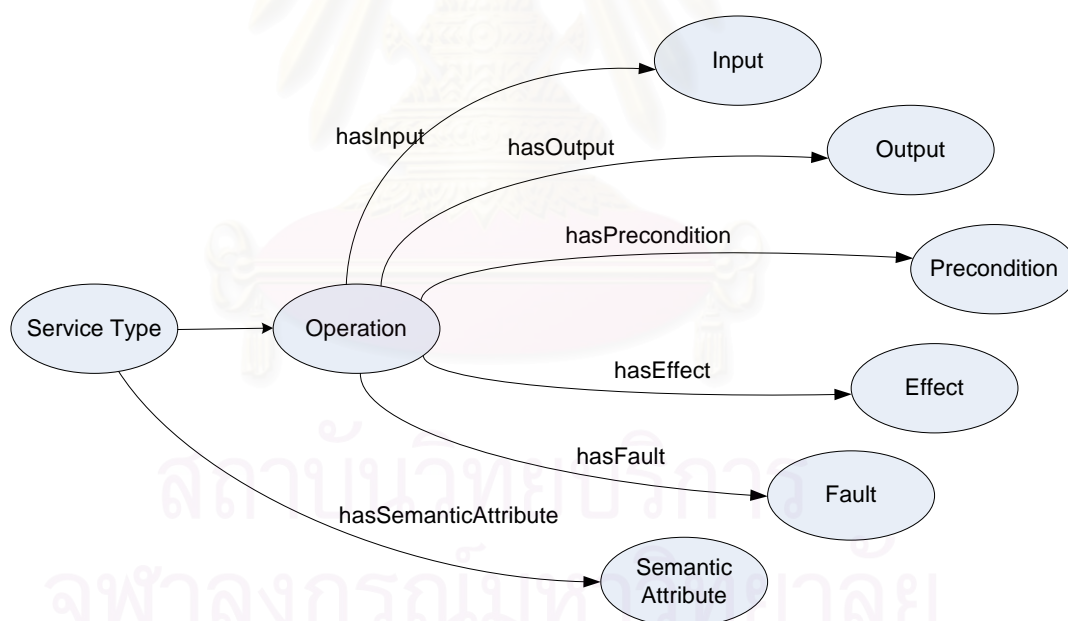
จากรูปที่ 3.2 BookAndMovieSeller เป็นชนิดบริการที่ถูกสร้างขึ้นโดยการขยายมาจากข้อกำหนดชื่อ BookSeller โดยมี moviePurchase และ moviePreview เป็นโอเปอเรชันใหม่ที่เพิ่มเข้ามา (บรรทัดที่ 5-8) ซึ่งจะจับคู่กับ Behaviour Specification (บรรทัดที่ 15-26) ในส่วน Behaviour Specification ของโอเปอเรชัน bookPurchase (บรรทัดที่ 9-14) ที่ได้รับสืบทอดมานั้น ได้ทำการแก้ไขข้อมูล 2 ที่ด้วยกันคือเงื่อนไขก่อนการดำเนินงานและผลกระทบดังสรุปในตารางที่ 3.1 ซึ่งหมายถึงว่าเงื่อนไขก่อนการดำเนินงานและผลกระทบของการเรียกใช้งานโอเปอเรชัน bookPurchase ของ BookAndMovieSeller จะยังมีความหมายที่สอดคล้องกับความหมายของโอเปอเรชัน bookPurchase ของ BookSeller และสามารถทำงานแทนได้

ตารางที่ 3.1 การเปรียบเทียบเงื่อนไขก่อนการทำงานและผลกระทบของการเรียกใช้งานโอเปอเรชัน  
bookPurchase

แบบจำลองชนิดบริการ	เงื่อนไขก่อนการทำงาน	ผลกระทบ
BookSeller	VisaCardHolder	ProductDelivered
BookAndMovieSeller	CreditCardHolder	ProductDeliveredByAir

### 3.1.3 อับเปอร์ออนโทโลยีสำหรับกำหนดความหมายให้กับชนิดบริการ

อับเปอร์ออนโทโลยีเป็นออนโทโลยีในระดับบนสุดใช้ในการอธิบายความสามารถเชิงพฤติกรรมและแอททริบิวต์เชิงความหมายที่แต่ละชนิดบริการมีส่วนร่วมกัน ซึ่งผู้วิจัยนำอับเปอร์ออนโทโลยีในส่วนของเซอริวิสโพรไฟล์และโพรเซสโมเดลของอาร์วล์-เอสมาใช้ โดยได้เพิ่มเติมคลาสและคุณสมบัติที่จำเป็นต่อการอธิบายข้อมูลในข้อกำหนดชนิดบริการเข้าไปด้วย ดังรูปที่ 3.3



รูปที่ 3.3 คลาสและคุณสมบัติในอับเปอร์ออนโทโลยีสำหรับกำหนดความหมายให้กับชนิดบริการ

คลาสในอับเปอร์ออนโทโลยี มีดังนี้

1. Input แสดงข้อมูลเข้า
2. Output แสดงข้อมูลออก

3. Precondition แสดงเงื่อนไขก่อนการดำเนินการ
4. Effect แสดงผลกระทบบ
5. Fault แสดงความผิดปกติ
6. SemanticAttribute แสดงลักษณะที่เกี่ยวข้องกับการให้บริการ

คุณสมบัติในอ็อบเจกต์ออนโทโลยี มีดังนี้

1. hasInput แสดงความสัมพันธ์ระหว่างอินสแตนซ์ของ Service Profile (การดำเนินการ) กับอินสแตนซ์ของคลาส Input ใช้กำหนดข้อมูลเข้าของการดำเนินการ
2. hasOutput แสดงความสัมพันธ์ระหว่างอินสแตนซ์ของ Service Profile กับอินสแตนซ์ของคลาส Output ใช้กำหนดข้อมูลออกของการดำเนินการ
3. hasPrecondition แสดงความสัมพันธ์ระหว่างอินสแตนซ์ของ Service Profile กับอินสแตนซ์ของคลาส Condition ใช้กำหนดเงื่อนไขที่ต้องเป็นจริงก่อนการดำเนินการ
4. hasEffect แสดงความสัมพันธ์ระหว่างอินสแตนซ์ของ Service Profile กับอินสแตนซ์ของคลาส Expression ใช้กำหนดผลกระทบหลังการดำเนินการ
5. hasFault แสดงความสัมพันธ์ระหว่างอินสแตนซ์ของ Service Profile กับอินสแตนซ์ของคลาส Fault ใช้กำหนดความผิดปกติของการดำเนินการ
6. hasSemanticAttribute แสดงความสัมพันธ์ระหว่างชนิดบริการกับอินสแตนซ์ของคลาส SemanticAttribute ใช้กำหนดคุณลักษณะเชิงความหมายที่เกี่ยวข้องกับการให้บริการของชนิดบริการ

### 3.1.4 ออนโทโลยีสำหรับชนิดบริการ

ออนโทโลยีสำหรับชนิดบริการ (Service Type Ontology) จะอธิบายพฤติกรรมและแอททริบิวต์เชิงความหมายของแต่ละชนิดบริการ ซึ่งเว็บเซอร์วิสภายใต้ชนิดบริการนี้จะใช้ร่วมกัน โดยจะอ้างอิงมาจากอ็อบเจกต์ออนโทโลยีในหัวข้อที่ 3.1.3 จึงทำให้ได้รับการสืบต่อคลาสและคุณสมบัติต่างๆ ด้วย โดยออนโทโลยีสำหรับชนิดบริการจะนำข้อมูลจากข้อกำหนดชนิดบริการในส่วนของ Behaviour Specification และ Semantic Attribute Specification ที่ใช้ในการอธิบายพฤติกรรมการทำงานของโอเปอเรชันทั้งหมดและแอททริบิวต์เชิงความหมายมาเขียนในรูปของเอกสารถาวร-เอส เพื่อใช้ประโยชน์ในการพิจารณาหาความสัมพันธ์ของข้อมูลในแต่ละส่วน ในกรณีนี้ข้อกำหนดนั้นไม่ได้ขยายกันมา โดยใช้เครื่องมือในการหาความสัมพันธ์เช่น เจส (Jess) [18] และบอสซาม (Bossam) [19] ร่วมกับกฎการเข้าคู่เชิงความหมาย (หัวข้อที่ 3.2.2) โดยรูปที่ 3.4 แสดงตัวอย่างออนโทโลยีสำหรับชนิดบริการขายหนังสือ

### 3.1.5 ข้อเสนอบริการ

ข้อเสนอบริการ หมายถึงคำอธิบายบริการที่ผู้ให้บริการมาลงทะเบียนเพื่อบอกว่าเว็บเซอวิสของตนมีพฤติกรรมการทำงานอยู่ภายใต้ชนิดบริการใด โดยผู้ให้บริการจะต้องมีเอกสารระดับเบสิคยูเอสดีแอลที่สอดคล้องกับรายละเอียดในส่วนของ Operation Specification และมีรายละเอียดของเอกสารอาร์เอส-เอสที่สอดคล้องกับรายละเอียดในส่วนของ Behaviour Specification กับ Semantic Attribute Specification แต่ผู้ให้บริการแต่ละรายนั้นอาจจะมีรายละเอียดในส่วนของไบนดิง นิพจน์เชิงเปรียบเทียบ (Relational Expression) ของเงื่อนไขก่อนการดำเนินงานและผลกระทบ รวมทั้งค่าคุณลักษณะเชิงความหมายที่แตกต่างกันได้ สำหรับการกำหนดนิพจน์นั้นสามารถมีตัวกระทำได้ดังนี้คือ  $>$ ,  $<$ ,  $=$ ,  $\leq$ ,  $\geq$  โดยแต่ละเว็บเซอวิสนั้นจะมีตัวแทนของบริการอยู่ภายใต้หนึ่งชนิดบริการเท่านั้น สำหรับตัวอย่างเอกสารระดับเบสิคยูเอสดีแอลและเอกสารอาร์เอส-เอสที่อยู่ภายใต้ชนิดบริการขายหนังสือ แสดงดังรูปที่ 3.5 และ 3.6 ตามลำดับ

### 3.1.6 การใช้งานแบบจำลองชนิดบริการ

- การค้นหาเว็บเซอวิสที่เข้ากันได้ : แนวคิดแบบจำลองชนิดบริการนั้น จะทำให้การสร้างเว็บเซอวิสมีพฤติกรรมการทำงานเป็นไปในทิศทางเดียวกัน ซึ่งทำให้การค้นหาเว็บเซอวิสที่เข้ากันได้นั้นสามารถทำได้รวดเร็วและมีประสิทธิภาพ เพราะว่าการกลไกของแบบจำลองชนิดบริการจะเป็นตัวบังคับตั้งแต่ต้นว่าเว็บเซอวิสใดที่ประกาศข้อเสนอบริการอยู่ภายใต้ชนิดบริการเดียวกัน จะมีพฤติกรรมการทำงานที่สอดคล้องกันทั้งในส่วนของข้อมูลเชิงซิกเนเจอร์และข้อมูลเชิงพฤติกรรม ดังนั้นการพิจารณาจึงพิจารณาเฉพาะค่าของข้อมูลในส่วนไบนดิง นิพจน์ของเงื่อนไขก่อนการดำเนินงานและผลกระทบ รวมทั้งแอททริบิวต์เชิงความหมายเท่านั้น
- การนำกลับมาใช้ซ้ำ : แนวคิดของการสร้างชนิดบริการนั้น สนับสนุนการนำกลับมาใช้ซ้ำ โดยสามารถสร้างชนิดบริการขึ้นมาใหม่โดยใช้วิธีการขยายจากชนิดบริการที่มีอยู่เดิมได้ ซึ่งวิธีการนี้จะเป็นตัวกำหนดให้ชนิดบริการที่สร้างขึ้นใหม่นั้นมีพฤติกรรมการทำงานที่สัมพันธ์กับชนิดบริการเดิมที่อ้างอิงมา ซึ่งทำให้แน่ใจได้ว่าข้อมูลเชิงซิกเนเจอร์และข้อมูลเชิงพฤติกรรมสามารถเข้ากันได้แน่นอน อีกทั้งช่วยให้การสร้างชนิดบริการใหม่ๆ เพื่อรองรับเว็บเซอวิสใหม่ๆ ทำได้เร็วขึ้นด้วย

ภาพรวมแนวคิดแบบจำลองชนิดบริการสำหรับเว็บเซอวิสขายหนังสือ แสดงดังรูปที่ 3.7



```

<?xml version="1.0"?>
<rdf:RDF
  xmlns="http://localhost:8080/ontology/servicetype/BookSeller/bookPurchase.owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:upper="http://localhost:8080/ontology/upperontology/upperonto.owl#"
  xml:base="http://localhost:8080/ontology/servicetype/BookSeller/bookPurchase.owl">
  <owl:Ontology rdf:about="">
    <owl:imports rdf:resource="http://localhost:8080/ontology/upperontology/upperonto.owl"/>
  </owl:Ontology>
  <owl:Class rdf:ID="BookNumber">
    <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/1.1/Process.owl#Input"/>
  </owl:Class>
  <owl:Class rdf:ID="ISBN">
    <rdfs:subClassOf rdf:resource="#BookNumber"/>
  </owl:Class>
  <owl:Class rdf:ID="CreditcardDetail">
    <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/1.1/Process.owl#Input"/>
  </owl:Class>
  <owl:Class rdf:ID="PurchasedBook">
    <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/1.1/Process.owl#Output"/>
  </owl:Class>
  <owl:Class rdf:ID="Payment">
    <rdfs:subClassOf
      rdf:resource="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#Condition"/>
  </owl:Class>
  <owl:Class rdf:ID="CreditCardHolder">
    <rdfs:subClassOf rdf:resource="#Payment"/>
  </owl:Class>
  <owl:Class rdf:ID="VisaCardHolder">
    <rdfs:subClassOf rdf:resource="#CreditCardHolder"/>
  </owl:Class>
  <owl:Class rdf:ID="ProductDelivered">
    <rdfs:subClassOf
      rdf:resource="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#Expression"/>
  </owl:Class>
  <owl:Class rdf:ID="ProductDeliveredByAir">
    <rdfs:subClassOf rdf:resource="#ProductDelivered"/>
  </owl:Class>
  <owl:Class rdf:ID="ExpiredCreditCard">
    <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/1.1/Process.owl#Fault"/>
  </owl:Class>
  <owl:Class rdf:ID="BookCategory">
    <rdfs:subClassOf
      rdf:resource="http://www.daml.org/services/owl-s/1.1/Profile.owl#SemanticAttribute"/>
  </owl:Class>
  <owl:Class rdf:ID="Novel">
    <rdfs:subClassOf rdf:resource="#BookCategory"/>
  </owl:Class>
  <owl:Class rdf:ID="Poetry">
    <rdfs:subClassOf rdf:resource="#BookCategory"/>
  </owl:Class>
</rdf:RDF>

```

รูปที่ 3.4 ออนโทโลยีสำหรับชนิดบริการขายหนังสือ

```

<?xml version="1.0" encoding="UTF-8" ?>
<wsdl:definitions targetNamespace="http://localhost:8080/axis/services/Pacific"
  xmlns:apachesoap="http://xml.apache.org/xml-soap"
  xmlns:impl="http://localhost:8080/axis/services/Pacific"
  xmlns:intf="http://localhost:8080/axis/services/Pacific"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns1="http://fault.com"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <wsdl:types>
    <schema targetNamespace="http://fault.com" xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
      <complexType name="ExpCreditCard">
        <sequence />
      </complexType>
    </schema>
  </wsdl:types>
  <wsdl:message name="ExpCreditCard">
    <wsdl:part name="fault" type="tns1:ExpCreditCard" />
  </wsdl:message>
  <wsdl:message name="bookPurchaseRequest">
    <wsdl:part name="in0" type="xsd:int" />
    <wsdl:part name="in1" type="soapenc:string" />
  </wsdl:message>
  <wsdl:message name="bookPurchaseResponse">
    <wsdl:part name="bookPurchaseReturn" type="xsd:int" />
  </wsdl:message>
  <wsdl:portType name="BookSeller">
    <wsdl:operation name="bookPurchase" parameterOrder="in0 in1">
      <wsdl:input message="impl:bookPurchaseRequest" name="bookPurchaseRequest" />
      <wsdl:output message="impl:bookPurchaseResponse" name="bookPurchaseResponse" />
      <wsdl:fault message="impl:ExpCreditCard" name="ExpCreditCard" />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="PacificSoapBinding" type="impl:BookSeller">
    <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="bookPurchase">
      <wsdlsoap:operation soapAction="" />
      <wsdl:input name="bookPurchaseRequest">
        <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="http://DefaultNamespace" use="encoded" />
      </wsdl:input>
      <wsdl:output name="bookPurchaseResponse">
        <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="http://localhost:8080/axis/services/Pacific" use="encoded" />
      </wsdl:output>
      <wsdl:fault name="ExpCreditCard">
        <wsdlsoap:fault encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          name="ExpCreditCard" namespace="http://localhost:8080/axis/services/Pacific"
          use="encoded" />
      </wsdl:fault>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="BookSellerService">
    <wsdl:port binding="impl:PacificSoapBinding" name="Pacific">
      <wsdlsoap:address location="http://localhost:8080/axis/services/Pacific" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>

```

รูปที่ 3.5 เอกสารดับเบิลยูเอสดีแอลที่ใช้ในการประกาศข้อเสนอบริการของเว็บไซต์ชื่อ Pacific  
ภายใต้ชนิดบริการขายหนังสือ

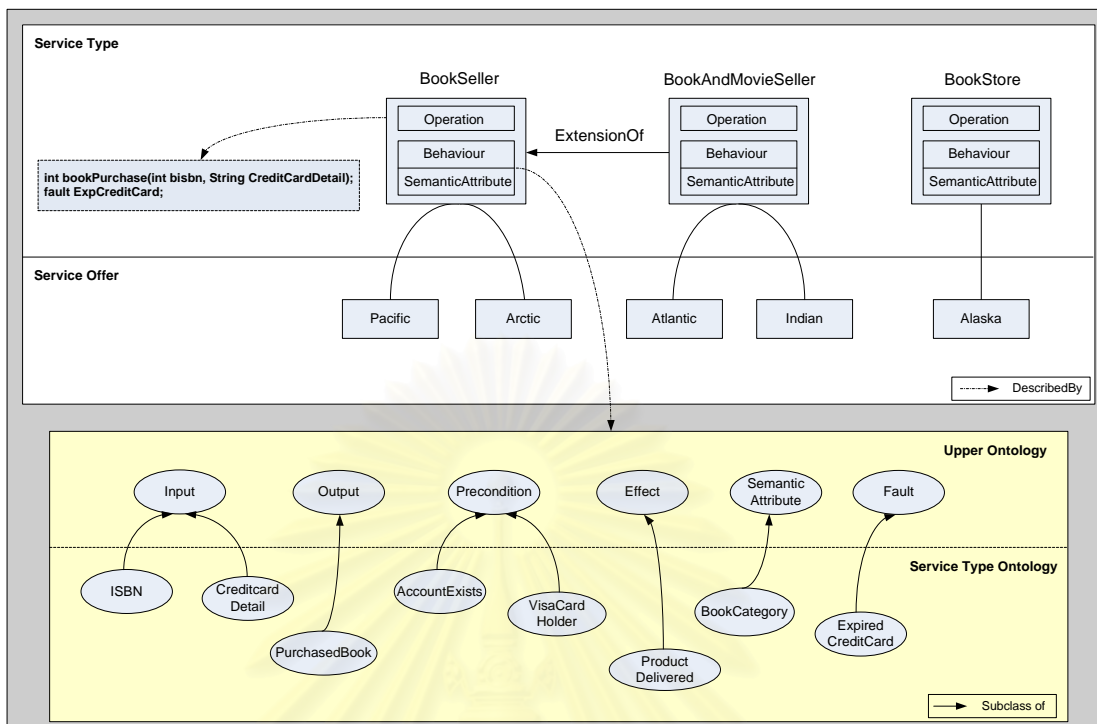
```

<?xml version="1.0"?>
<rdf:RDF
  xmlns="http://localhost:8080/ontology/serviceoffer/Pacific.owl#"
  xmlns:process="http://www.daml.org/services/owl-s/1.1/Process.owl#"
  xmlns:list="http://www.daml.org/services/owl-s/1.1/generic/ObjectList.owl#"
  xmlns:swrl="http://www.w3.org/2003/11/swrl#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:expr="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#"
  xmlns:bsbp="http://localhost:8080/ontology/servicetype/BookSeller/bookPurchase.owl#"
  xmlns:service="http://www.daml.org/services/owl-s/1.1/Service.owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:upper="http://localhost:8080/ontology/upperontology/upperonto.owl#"
  xmlns:profile="http://www.daml.org/services/owl-s/1.1/Profile.owl#"
  xml:base="http://localhost:8080/ontology/serviceoffer/Pacific.owl">
  <owl:Ontology rdf:about="">
    <owl:imports
      rdf:resource="http://localhost:8080/ontology/servicetype/BookSeller/bookPurchase.owl"/>
  </owl:Ontology>
  <bsbp:ISBN rdf:ID="PacificISBN"/>
  <bsbp:CredicardDetail rdf:ID="PacificCreditcardDetail"/>
  <bsbp:PurchasedBook rdf:ID="PacificPurchasedBook"/>
  <bsbp:VisaCardHolder rdf:ID="PacificVisaCardHolder"/>
  <bsbp:ExpiredCreditCard rdf:ID="PacificExpiredCreditCard"/>
  <process:ResultVar rdf:ID="Days"/>
  <profile:Profile rdf:ID="bookPurchase">
    <profile:hasInput rdf:resource="#PacificISBN"/>
    <profile:hasInput rdf:resource="#PacificCreditcardDetail"/>
    <profile:hasOutput rdf:resource="#PacificPurchasedBook"/>
    <profile:hasPrecondition rdf:resource="#PacificVisaCardHolder"/>
    <profile:hasResult>
      <process:Result rdf:ID="PacificPurchasedBookResult">
        <process:hasEffect>
          <bsbp:ProductDelivered rdf:ID="PacificProductDelivered">
            <expr:expressionBody
              rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">
              <swrl:AtomList>
                <rdf:first>
                  <swrl:BuiltinAtom>
                    <swrl:arguments>
                      <rdf:List>
                        <rdf:first rdf:resource="#Days"/>
                        <rdf:rest>
                          <rdf:List>
                            <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
                            <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#int">7</rdf:first>
                          </rdf:List>
                        </rdf:rest>
                      </rdf:List>
                    </swrl:arguments>
                    <swrl:builtin rdf:resource="http://www.w3.org/2003/11/swrlb#lessThanOrEqual"/>
                  </swrl:BuiltinAtom>
                </rdf:first>
                <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
              </swrl:AtomList>
            </expr:expressionBody>
          </bsbp:ProductDelivered>
        </process:hasEffect>
      </process:Result>
    </profile:hasResult>
    <profile:hasFault rdf:resource="#PacificExpiredCreditCard"/>
  </profile:Profile>
  <bsbp:Novel rdf:ID="PacificNovel">
    <upper:hasSemanticAttribute rdf:resource="#PacificNovel"/>
  </bsbp:Novel>
</rdf:RDF>

```

รูปที่ 3.6 เอกสารอวาล์-เอสที่ใช้ในการประกาศข้อเสนอบริการของเว็บเซอร์วิสชื่อ Pacific

ภายใต้ชนิดบริการขายหนังสือ



รูปที่ 3.7 ภาพรวมแสดงแนวคิดของแบบจำลองชนิดบริการสำหรับเว็บเซอร์วิสขายหนังสือ

### 3.2 กฎการเข้ากันได้ของเว็บเซอร์วิสตามแบบจำลองชนิดบริการ

ในการพิจารณาการเข้ากันได้ของเว็บเซอร์วิสนั้น ผู้วิจัยจะนำเสนอกฎที่ใช้ในการพิจารณา ซึ่งจะแบ่งการพิจารณาออกเป็น 3 ส่วนตามลำดับคือ การเข้าคู่เชิงซิกเนเจอร์ การเข้าคู่เชิงความหมาย และการเข้าคู่เชิงข้อมูลเฉพาะตัว โดยหากพิจารณาแล้วไม่เข้าคู่ในกรณีใดก็จะไม่พิจารณาการเข้าคู่แบบอื่นต่อ สำหรับเว็บเซอร์วิสที่ประกาศข้อเสนอบริการภายใต้ชนิดบริการหนึ่งๆ จะเข้าคู่เชิงซิกเนเจอร์และเชิงความหมายของพฤติกรรมกับเว็บเซอร์วิสอื่นที่ประกาศข้อเสนอภายใต้ชนิดบริการเดียวกันหรือภายใต้ชนิดบริการแม่อยู่แล้ว ทั้งนี้เป็นไปตามแนวคิดของแบบจำลองชนิดบริการและข้อเสนอบริการ ในหัวข้อที่ 3.1 ดังนั้นการจับคู่จึงเหลือเพียงการพิจารณาการเข้าคู่เชิงข้อมูลเฉพาะตัวว่าเข้าคู่กันหรือไม่ สำหรับเว็บเซอร์วิสใดๆ ที่ไม่ได้ประกาศข้อเสนอบริการไว้ภายใต้ชนิดบริการเดียวกัน หรือภายใต้ชนิดบริการที่เป็นแม่-ลูกกัน ก็อาจจะเข้ากันได้ แต่ต้องทำการพิจารณาการเข้าคู่ทั้งสามแบบอย่างละเอียด รายละเอียดการเข้าคู่เป็นดังนี้

### 3.2.1 การเข้าคู่เชิงซิงเกิลเนเซอร์

ตามข้อกำหนดของดับเบิลยูเอสดีแอลใน [3] นั้น ได้กำหนดรูปแบบของโอเปอเรชันซึ่งสามารถใช้งานได้จริงอยู่ 2 แบบ\* คือ

- วัน-เวย์ (One-way) หมายถึง โอเปอเรชันรับข้อความเข้าแต่ไม่มีการตอบกลับ ซึ่งมีโครงสร้างดังรูปที่ 3.8

```
<wsdl:definitions .... > <wsdl:portType .... > *
  <wsdl:operation name="nmtoken">
    <wsdl:input name="nmtoken"? message="qname" />
  </wsdl:operation>
</wsdl:portType >
</wsdl:definitions>
```

รูปที่ 3.8 โครงสร้างของวัน-เวย์โอเปอเรชัน [3]

- รีควีส-เรสปอนซ์ (Request-response) หมายถึง โอเปอเรชันรับข้อความเข้าและมีการตอบกลับ ซึ่งมีโครงสร้างดังรูปที่ 3.9

```
<wsdl:definitions .... >
  <wsdl:portType .... > *
    <wsdl:operation name="nmtoken" parameterOrder="nmtokens">
      <wsdl:input name="nmtoken"? message="qname" />
      <wsdl:output name="nmtoken"? message="qname" />
      <wsdl:fault name="nmtoken" message="qname" />*
    </wsdl:operation>
  </wsdl:portType >
</wsdl:definitions>
```

รูปที่ 3.9 โครงสร้างของรีควีส-เรสปอนซ์โอเปอเรชัน [3]

รูปแบบของโอเปอเรชันนั้นจะมีผลต่อการพิจารณาการเข้ากันได้เพื่อการทำงานแทนที่กันของเว็บเซอร์วิส โดยชนิดบริการที่จะเข้ากันได้นั้นจะต้องมีรูปแบบโอเปอเรชันเป็นแบบเดียวกันด้วย แต่เนื่องจากซิงเกิลเนเซอร์ของโอเปอเรชันได้อธิบายรูปแบบของโอเปอเรชันอยู่แล้ว การพิจารณาซิงเกิลเนเซอร์จึงเป็นทั้งการพิจารณารูปแบบการให้บริการและการพิจารณาข้อมูลที่มีการแลกเปลี่ยนไปด้วยในตัว ดังนั้นในการเข้าคู่เชิงซิงเกิลเนเซอร์จะนำข้อมูลชนิดบริการในส่วนของ Operation

\* สำหรับรูปแบบของโอเปอเรชันอีก 2 แบบที่กำหนดไว้ในข้อกำหนดของดับเบิลยูเอสดีแอล [3] ได้แก่ โซลิซิท-เรสปอนซ์ (Solicit-response) และโนติฟิเคชัน (Notification) นั้น ยังไม่สามารถใช้งานได้จริง เนื่องจากไม่ได้มีการกำหนดลักษณะการโบนดิงของทั้งสองแบบนี้ไว้ [3]

Specification ซึ่งประกอบด้วย โอปอเรชัน จำนวนพารามิเตอร์ ชนิดของพารามิเตอร์ ชนิดของผลลัพธ์ที่จะคืนกลับเมื่อสิ้นสุดการทำงาน และความผิดพลาดที่อาจเกิดขึ้นมาพิจารณา โดยแนวคิดที่ใช้ในการพิจารณานั้น อ้างอิงมาจาก Signature Subtyping Rules for Operation Interfaces จาก [20]

กำหนดให้ Operation Specification X จะเป็นซับไทป์เชิงซิกเนเจอร์ (Signature Subtyping) และเข้ากันได้กับ Operation Specification Y หากเป็นไปตามเงื่อนไขดังต่อไปนี้

- สำหรับทุกโอเปอเรชันที่อยู่ใน Y จะต้องมีโอเปอเรชันนั้นใน X ด้วย โดยต้องกำหนดชื่อโอเปอเรชันเหมือนกัน
- สำหรับทุกข้อมูลเข้าของโอเปอเรชันที่อยู่ใน Y จะต้องมีจำนวนพารามิเตอร์ตรงกับข้อมูลเข้าของโอเปอเรชันเดียวกันที่อยู่ใน X โดยชื่อพารามิเตอร์แตกต่างกันได้
- สำหรับทุกข้อมูลเข้าของโอเปอเรชันที่อยู่ใน Y ชนิดพารามิเตอร์จะต้องตรงกับชนิดพารามิเตอร์ในข้อมูลเข้าที่อยู่ตำแหน่งเดียวกันของโอเปอเรชันเดียวกันที่อยู่ใน X หรือเป็นชนิดพารามิเตอร์ที่สืบทอดมาจากชนิดพารามิเตอร์ในข้อมูลเข้าของโอเปอเรชันเดียวกันที่อยู่ใน X
- สำหรับทุกข้อมูลออกของโอเปอเรชันที่อยู่ใน Y จะต้องมีจำนวนพารามิเตอร์ผลลัพธ์ (Result Parameters) ตรงกับข้อมูลออกของโอเปอเรชันเดียวกันที่อยู่ใน X โดยชื่อพารามิเตอร์ผลลัพธ์แตกต่างกันได้
- สำหรับทุกข้อมูลออกของโอเปอเรชันที่อยู่ใน X ชนิดผลลัพธ์ (Result Type) จะต้องตรงกับชนิดผลลัพธ์ของข้อมูลออกที่อยู่ตำแหน่งเดียวกันของโอเปอเรชันเดียวกันที่อยู่ใน Y หรือเป็นชนิดผลลัพธ์ที่สืบทอดมาจากชนิดผลลัพธ์ของข้อมูลออกของโอเปอเรชันเดียวกันที่อยู่ใน Y
- สำหรับทุกความผิดพลาดของโอเปอเรชันที่อยู่ใน X จะต้องมีความผิดพลาดนั้นในโอเปอเรชันเดียวกันที่อยู่ใน Y ด้วย โดยต้องมีชื่อความผิดพลาดเหมือนกัน
- สำหรับทุกความผิดพลาดของโอเปอเรชันที่อยู่ใน X จะต้องมีจำนวนพารามิเตอร์ตรงกับความผิดพลาดของโอเปอเรชันเดียวกันที่อยู่ใน Y โดยชื่อพารามิเตอร์แตกต่างกันได้
- สำหรับทุกความผิดพลาดของโอเปอเรชันที่อยู่ใน X ชนิดพารามิเตอร์จะต้องตรงกับชนิดพารามิเตอร์ที่อยู่ตำแหน่งเดียวกันของความผิดพลาดของโอเปอเรชันเดียวกันที่อยู่ใน Y หรือเป็นชนิดพารามิเตอร์ที่สืบทอดมาจากชนิดพารามิเตอร์ของความผิดพลาดของโอเปอเรชันเดียวกันที่อยู่ใน Y

สำหรับการพิจารณาการสืบทอดของชนิดข้อมูลนั้น จะอ้างอิงจากโครงสร้างการสืบทอดชนิดข้อมูลตามเอ็กซ์เอ็มแอลสคีมา [10]

### 3.2.2 การเข้าคู่เชิงความหมาย

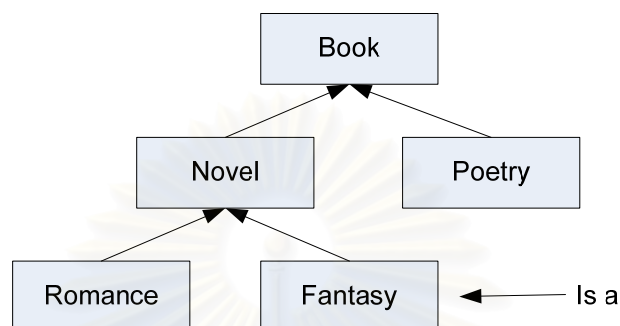
ในส่วนนี้จะอธิบายเกี่ยวกับการเข้าคู่เชิงความหมายที่อ้างอิงมาจาก [16]<sup>\*</sup>, [21] โดยการพิจารณานั้นจะอยู่บนพื้นฐานการเปรียบเทียบกันของออนโทโลยีเทอมในเอกสารอาร์เอส-เอส ซึ่งอธิบายข้อมูลในส่วน Behaviour Specification และ Semantic Attribute Specification ซึ่งประกอบด้วย ข้อมูลเข้า ข้อมูลออก เงื่อนไขก่อนการดำเนินงาน ผลกระทบ ความผิดพลาด และคุณลักษณะเชิงความหมาย

กำหนดให้  $C_x$  เป็นออนโทโลยีเทอม (Ontology Term) ที่ระบุอยู่ใน Behaviour Specification หรือ Semantic Attribute Specification ของชนิดบริการ X และ  $C_y$  เป็นออนโทโลยีเทอมที่ระบุอยู่ใน Behaviour Specification หรือ Semantic Attribute Specification ของชนิดบริการ Y การพิจารณาว่าความหมายของ X เข้ากันได้กับความหมายของ Y จะพิจารณาความเข้ากันได้ของแต่ละออนโทโลยีเทอม ดังนี้

1. ถ้า  $C_y \equiv C_x$  แล้ว  $C_x$  จะเป็นการเข้าคู่อย่างถูกต้อง (Exact Match) กับ  $C_y$  โดย  $\equiv$  หมายถึง มีค่าเท่าเทียมกับ ดังตัวอย่างรูปที่ 3.10 ชนิดบริการ X มีออนโทโลยีเทอมเป็นคำว่า Novel จะถือว่าเป็นการเข้าคู่อย่างถูกต้องกับชนิดบริการ Y ที่มีออนโทโลยีเทอมเป็นคำว่า Novel เหมือนกัน
2. ถ้า  $C_x \sqsubseteq C_y$  แล้ว  $C_x$  จะเป็นการเข้าคู่อย่างเจาะจง (Specialised Match) กับ  $C_y$  โดย  $\sqsubseteq$  หมายถึง ถูกครอบคลุมโดย (Subsumed By) ซึ่งในกรณีนี้คือ  $C_x$  เป็นคำที่มีความหมายเฉพาะเจาะจงกว่าของ  $C_y$  ดังตัวอย่างรูปที่ 3.10 ชนิดบริการ X มีออนโทโลยีเทอมเป็นคำว่า Romance จะถือว่าเป็นการเข้าคู่อย่างเจาะจงกับชนิดบริการ Y ที่มีออนโทโลยีเทอมเป็นคำว่า Novel
3. ถ้าไม่มีความสัมพันธ์ตรงกับนิพจน์ด้านบนแล้ว  $C_x$  จะไม่เข้าคู่ (Fail Match) กับ  $C_y$  การเข้าคู่ในแบบที่ 1 และ 2 นั้นจะถือว่ามีน้ำหนักเท่ากัน ในการพิจารณาความเข้ากันได้เชิงความหมายของเว็บเซอร์วิสแต่ละคู่ นั้นจะเริ่มตั้งแต่การพิจารณา ข้อมูลเข้า ข้อมูลออก ไปจนกระทั่งแอททริบิวต์เชิงความหมาย ซึ่งระบุอยู่ใน Behaviour Specification และ Semantic

<sup>\*</sup> เนื่องจากในงานวิจัย [16] จะเป็นการค้นหาเว็บเซอร์วิสที่ตรงกับความต้องการของผู้ใช้บริการ จึงมีการเข้าคู่อยู่หลายแบบ แต่ในงานวิจัยนี้เป็นการหาเว็บเซอร์วิสตัวแทนเพื่อมาทำงานแทนที่กันจึงได้เลือกใช้เฉพาะการเข้าคู่แบบเคร่งครัด (Strong) เท่านั้น และตัดการเข้าคู่แบบหย่อน (Weak) ออกไป ทั้งนี้เพื่อให้แน่ใจได้ว่าเว็บเซอร์วิสตัวแทนสามารถทำงานแทนได้จริง

Attribute Specification ตามลำดับ โดยแต่ละออนโทโลยีเทอมจะต้องเป็นไปตามกฎการเข้าสู่ของออนโทโลยีเทอมข้างต้นจึงจะถือว่ามีสภาพเข้ากันได้ ถ้าเกิดการไม่เข้าคู่เมื่อใดก็จะถือว่าไม่สามารถเข้ากันได้ทันที



รูปที่ 3.10 หมวดหมู่ของหนังสือ (Taxonomy of Book)

กำหนดให้ Behaviour Specification และ Semantic Attribute Specification ของชนิดบริการ X จะเป็นซับไทป์เชิงความหมาย (Semantic Subtyping) และเข้ากันได้กับ Behaviour Specification และ Semantic Attribute Specification ของชนิดบริการ Y หากเป็นไปตามเงื่อนไขดังต่อไปนี้

- สำหรับแต่ละคู่ของออนโทโลยีเทอมของข้อมูลเข้าใน X และ Y ที่อยู่ตำแหน่งเดียวกัน ออนโทโลยีเทอมของข้อมูลเข้าใน Y จะเข้าคู่อย่างถูกต้องหรือเข้าคู่อย่างเจาะจงกับออนโทโลยีเทอมของข้อมูลเข้าใน X
- สำหรับแต่ละคู่ของออนโทโลยีเทอมของข้อมูลออกใน X และ Y ที่อยู่ตำแหน่งเดียวกัน ออนโทโลยีเทอมของข้อมูลออกใน X จะเข้าคู่อย่างถูกต้องหรือเข้าคู่อย่างเจาะจงกับออนโทโลยีเทอมของข้อมูลออกใน Y
- สำหรับแต่ละออนโทโลยีเทอมของเงื่อนไขก่อนการดำเนินงานใน X จะมีออนโทโลยีเทอมของเงื่อนไขก่อนการดำเนินงานใน Y ที่เข้าคู่อย่างถูกต้องหรือเข้าคู่อย่างเจาะจงกับออนโทโลยีเทอมนั้นใน X
- สำหรับแต่ละออนโทโลยีเทอมของผลกระทบใน Y จะมีออนโทโลยีเทอมของผลกระทบใน X ที่เข้าคู่อย่างถูกต้องหรือเข้าคู่อย่างเจาะจงกับออนโทโลยีเทอมนั้นใน Y



- สำหรับแต่ละออนโทโลยีเทอมของความผิดพลาดใน  $X$  จะมีออนโทโลยีเทอมของความผิดพลาดใน  $Y$  โดยที่ออนโทโลยีเทอมใน  $X$  จะเข้าคู่อย่างถูกต้องหรือเข้าคู่อย่างเจาะจงกับออนโทโลยีเทอมใน  $Y$
- สำหรับแต่ละออนโทโลยีเทอมของคุณลักษณะเชิงความหมายใน  $Y$  จะมีออนโทโลยีเทอมของคุณลักษณะเชิงความหมายใน  $X$  ที่เข้าคู่อย่างถูกต้องหรือเข้าคู่อย่างเจาะจงกับออนโทโลยีเทอมใน  $Y$  อีกทั้งออนโทโลยีเทอมของค่าคุณลักษณะเชิงความหมายใน  $X$  ก็เข้าคู่อย่างถูกต้องหรือเข้าคู่อย่างเจาะจงกับออนโทโลยีเทอมของค่าคุณลักษณะเชิงความหมายใน  $Y$  ด้วย

### 3.2.3 การเข้าคู่เชิงข้อมูลเฉพาะตัว

หลังจากทำการพิจารณาการเข้าคู่เชิงซิกเนเจอร์และเชิงความหมายตามชนิดบริการของสองเว็บเซอร์วิสแล้ว จะต้องพิจารณาการเข้าคู่เชิงข้อมูลเฉพาะตัวที่อยู่ในส่วนข้อเสนอบริการของทั้งสองเว็บเซอร์วิสด้วย ดังนี้

1. ส่วนไบน์ดิงในเอกสารดับเบิลยูเอสดีแอล ในอิลิเมนต์ soap:binding ซึ่งประกอบด้วยแอททริบิวต์ style และ transport ของทั้งสองเว็บเซอร์วิสจะต้องมีค่าเหมือนกัน
2. นิพจน์ของเงื่อนไขก่อนการดำเนินงานและผลกระทบในเอกสารอาร์เอสของสองเว็บเซอร์วิสจะต้องเข้าคู่กันตามกฎการเข้าคู่เชิงพิสัยตัวเลข (Numerical Range) ซึ่งอ้างอิงมาจาก [16]<sup>\*</sup> ดังนี้

กำหนดให้  $N_x$  เป็นเซตไม่ว่าง (Nonempty Set) ของค่าพิสัยตัวเลขในนิพจน์ของเงื่อนไขก่อนการดำเนินงานหรือผลกระทบที่ระบุอยู่ในเว็บเซอร์วิส  $X$  ( $W_x$ ) และ  $N_y$  เป็นเซตไม่ว่างของค่าพิสัยตัวเลขในนิพจน์ของเงื่อนไขก่อนการดำเนินงานหรือผลกระทบที่ระบุอยู่ในเว็บเซอร์วิส  $Y$  ( $W_y$ )

1. ถ้า  $N_x \subseteq N_y$  แล้ว  $W_x$  จะเป็นการเข้าคู่อย่างถูกต้องกับ  $W_y$
2. ถ้าในกรณีอื่นจะถือว่า  $W_x$  ไม่เข้าคู่กับ  $W_y$
3. ค่าคุณลักษณะเชิงความหมายในเอกสารอาร์เอสของทั้งสองเว็บเซอร์วิสจะต้องเป็นการเข้าคู่กันในกรณีใดกรณีหนึ่งตามกฎในหัวข้อที่ 3.2.2

---

<sup>\*</sup> เนื่องจากในงานวิจัย [16] จะเป็นการค้นหาเว็บเซอร์วิสที่ตรงกับความต้องการของผู้ใช้บริการ จึงมีการเข้าคู่อยู่หลายแบบ แต่ในงานวิจัยนี้เป็นการหาเว็บเซอร์วิสตัวแทนเพื่อมาทำงานแทนที่กันจึงได้เลือกใช้เฉพาะการเข้าคู่แบบเคร่งครัด (Strong) เท่านั้น และตัดการเข้าคู่แบบหย่อน (Weak) ออกไป ทั้งนี้เพื่อให้แน่ใจได้ว่าเว็บเซอร์วิสตัวแทนสามารถทำงานแทนได้จริง

### 3.3 ตัวอย่างการพิจารณาสภาพเข้ากันได้และการทำงานแทนที่กันของเว็บเซอร์วิส

หัวข้อนี้จะนำแนวคิดในหัวข้อที่ 3.1 – 3.2 มาใช้ โดยสมมติให้เว็บเซอร์วิส Y ที่เกิดขัดข้องขึ้นประกาศข้อเสนอบริการอยู่ภายใต้ชนิดบริการ BookSeller การพิจารณาเลือกเว็บเซอร์วิสที่เข้ากันได้ นั้นจะแบ่งออกเป็น 3 วิธี คือ

1. พิจารณาเว็บเซอร์วิสที่ประกาศข้อเสนอบริการอยู่ภายใต้ชนิดบริการเดียวกันคือ BookSeller เป็นลำดับแรก โดยจะทำการเลือกแต่ละเว็บเซอร์วิสขึ้นมาพิจารณาการเข้าคู่เชิงข้อมูลเฉพาะตัวกับเว็บเซอร์วิส Y ตามที่นำเสนอในหัวข้อที่ 3.2.3 โดยข้อมูลเชิงซิกเนเจอร์และข้อมูลเชิงความหมายของพฤติกรรมนั้นไม่จำเป็นต้องทำการพิจารณาเนื่องจากเว็บเซอร์วิสที่อยู่ภายใต้ชนิดบริการเดียวกันจะมีข้อมูล 2 ส่วนนั้นสอดคล้องกันเสมอ
2. พิจารณาเว็บเซอร์วิสที่ประกาศข้อเสนอบริการอยู่ภายใต้ชนิดบริการที่ขยายมาจากชนิดบริการ BookSeller โดยวิธีนี้ก็จะมีขั้นตอนเหมือนกับวิธีที่ 1 แต่จะใช้วิธีนี้เมื่อพิจารณาด้วยวิธีที่ 1 แล้วไม่พบเว็บเซอร์วิสใดที่มีสภาพเข้ากันได้กับเว็บเซอร์วิส Y
3. พิจารณาชนิดบริการอื่นๆ ตามแนวคิดที่ได้นำเสนอในหัวข้อที่ 3.2.1 และ 3.2.2 ตามลำดับ ถ้าพบชนิดบริการใดเข้ากันได้กับชนิดบริการ BookSeller ก็จะใช้เว็บเซอร์วิสที่ประกาศข้อเสนอบริการอยู่ภายใต้ชนิดบริการนั้นมาพิจารณาการเข้าคู่เชิงข้อมูลเฉพาะตัวกับเว็บเซอร์วิส Y ดังตัวอย่างต่อไปนี้

จากรูปที่ 3.11 และ 3.12 สมมติเลือกชนิดบริการชื่อ BookStore ขึ้นมาพิจารณาว่าเข้าคู่กับชนิดบริการ BookSeller หรือไม่ ขั้นตอนการพิจารณามีดังนี้

```

Service Type: BookSeller
Operation Specification:
    int bookPurchase(int isbn, String creditcardDetail);
    fault ExpCreditCard;
Behaviour Specification of bookPurchase:
    Input: ISBN, CreditcardDetail;
    Output: PurchasedBook;
    Precondition: VisaCardHolder;
    Effect: ProductDelivered;
    Fault: ExpiredCreditCard;
Semantic Attribute Specification: BookCategory;
  
```

รูปที่ 3.11 ชนิดบริการ BookSeller ที่นำมาพิจารณาสภาพเข้ากันได้

<pre> Service Type: BookStore Operation Specification:     int bookPurchase(long bnumber, String creditcardDetail);     fault ExpCreditCard; Behaviour Specification of bookPurchase:     Input: BookNumber, CreditcardDetail;     Output: PurchasedBook;     Precondition: CreditCardHolder;     Effect: ProductDeliveredByAir;     Fault: ExpiredCreditCard; Semantic Attribute Specification: BookCategory; </pre>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

รูปที่ 3.12 ชนิดบริการ BookStore ที่นำมาพิจารณาสภาพเข้ากันได้

- พิจารณาข้อมูลในส่วน Operation Specification

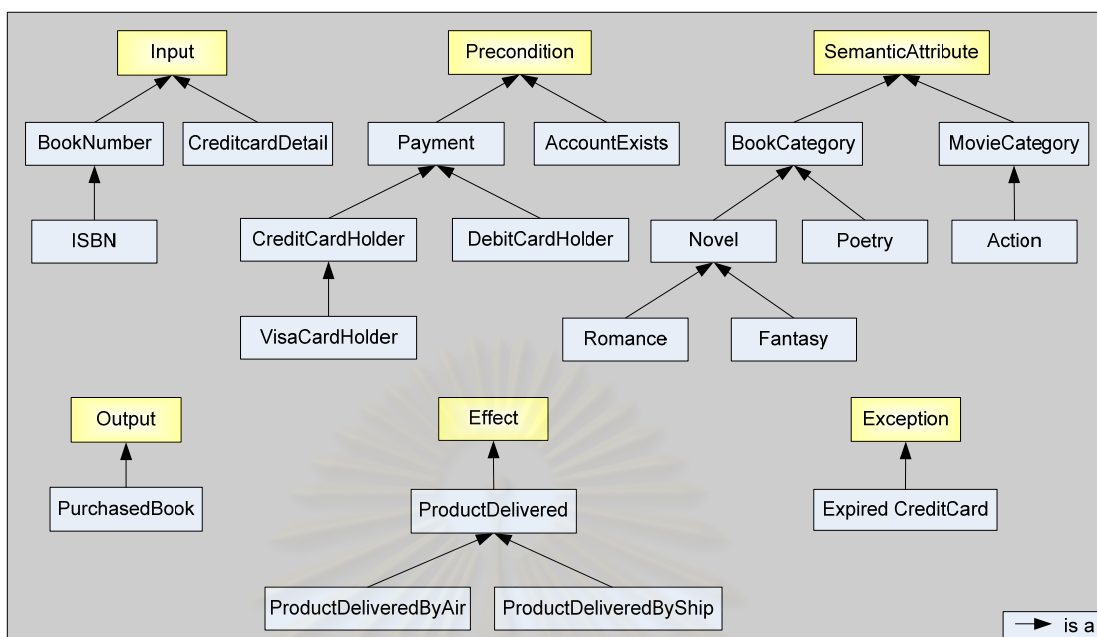
จากตารางที่ 3.2 จะเห็นว่าชนิดบริการทั้งสองมีซิกเนเจอร์ที่ต่างกัน โดยพารามิเตอร์ตัวแรกของโอเปอเรชัน bookPurchase ของชนิดบริการ BookSeller นั้นจะมีชนิดข้อมูลเป็น int แต่พารามิเตอร์ตัวแรกของโอเปอเรชัน bookPurchase ของชนิดบริการ BookStore นั้นมีชนิดข้อมูลเป็น long และทั้งสองชนิดบริการมีชื่อของพารามิเตอร์ตัวแรกต่างกัน แต่จากการเข้าคู่ซิกเนเจอร์นั้นสามารถสรุปได้ว่า ข้อมูลส่วนนี้ของทั้งสองชนิดบริการนั้นเข้าคู่กัน

ตารางที่ 3.2 การเปรียบเทียบข้อมูลในส่วน Operation Specification ระหว่างชนิดบริการ BookSeller กับ BookStore

ส่วนของข้อมูล	ชนิดบริการ BookSeller	ชนิดบริการ BookStore	ผลการเข้าคู่
ข้อมูลเข้า	int bisbn	long bnumber	เข้าคู่
	String creditcardDetail	String creditcardDetail	เข้าคู่
ข้อมูลออก	int	int	เข้าคู่
ความผิดพลาด	ExpCreditCard	ExpCreditCard	เข้าคู่

- พิจารณาข้อมูลในส่วนของ Behaviour Specification และ Semantic Attribute Specification

ในส่วนนี้จะใช้ออนโทโลยีสำหรับชนิดบริการในรูปที่ 3.13 ช่วยในการพิจารณาความสัมพันธ์ โดยรายละเอียดแต่ละส่วนแสดงดังตารางที่ 3.3



รูปที่ 3.13 ออนโทโลยีสำหรับชนิดบริการที่ใช้อธิบายความสัมพันธ์ของสองชนิดบริการ

ตารางที่ 3.3 ผลการจับคู่ระหว่างชนิดบริการ BookSeller กับ BookStore

ส่วนของข้อมูล	ชนิดบริการ BookSeller	ชนิดบริการ BookStore	ลักษณะการเข้าคู่
ข้อมูลเข้า	ISBN	BookNumber	เข้าคู่อย่างเจาะจง
	CreditcardDetail	CreditcardDetail	เข้าคู่อย่างถูกต้อง
ข้อมูลออก	PurchasedBook	PurchasedBook	เข้าคู่อย่างถูกต้อง
เงื่อนไขก่อนการทำงาน	VisaCardHolder	CreditCardHolder	เข้าคู่อย่างเจาะจง
ผลกระทบ	ProductDelivered	ProductDeliveredByAir	เข้าคู่อย่างเจาะจง
ความผิดพลาด	ExpiredCreditCard	ExpiredCreditCard	เข้าคู่อย่างถูกต้อง
คุณลักษณะเชิงความหมาย	BookCategory	BookCategory	เข้าคู่อย่างถูกต้อง

จากตารางที่ 3.3 จะเห็นว่า ทุกส่วนของข้อมูลของทั้งสองชนิดบริการนั้น สามารถเข้าคู่กันได้ทุกกรณีตามการเข้าคู่เชิงความหมาย ดังนั้นจึงสรุปได้ว่า ชนิดบริการ BookStore เข้ากันได้กับชนิดบริการ BookSeller ดังนั้นการพิจารณาเลือกเว็บไซต์ที่จะใช้แทนกันนั้น ก็จะพิจารณาเว็บไซต์ที่ประกาศข้อเสนอบริการอยู่ภายใต้แบบจำลอง BookStore ต่อไป

- พิจารณาข้อมูลเฉพาะตัวของเว็บเซอวิส

ในส่วนนี้จะทำการเลือกผู้เลือกเว็บเซอวิสที่อยู่ภายใต้ชนิดบริการ BookStore มาพิจารณาเปรียบเทียบกับเว็บเซอวิส Y ที่เกิดขัดข้อง โดยสมมติให้ X คือเว็บเซอวิสที่ผู้เลือกมา และให้ทั้งสองเว็บเซอวิสมีข้อมูลเฉพาะตัวดังตารางที่ 3.4

ตารางที่ 3.4 ตัวอย่างข้อมูลเฉพาะตัวของเว็บเซอวิส Y และเว็บเซอวิส X

ส่วนของข้อมูล	เว็บเซอวิส Y (ขัดข้อง)	เว็บเซอวิส X (ตัวแทน)
ไบนารีดิ้ง	style="rpc"; transport="http"	style="rpc"; transport="http"
นิพจน์ของเงื่อนไขก่อนการดำเนินการ	-	-
นิพจน์ของผลกระทบ	ProductDelivered.Days $\leq$ 7	ProductDeliveredByAir.Days $\leq$ 4
ค่าคุณลักษณะเชิงความหมาย	PacificNovel (อินสแตนซ์ของคลาส Novel)	AlaskaNovel (อินสแตนซ์ของคลาส Novel)

จากตารางที่ 3.4

- ในส่วนไบนารีดิ้ง จะเห็นว่าทั้งสองเว็บเซอวิสมีค่าเหมือนกัน
- ในส่วนนิพจน์ของผลกระทบ จะเห็นว่านิพจน์ของเว็บเซอวิส X เป็นเซตย่อยของนิพจน์ของเว็บเซอวิส Y ดังนั้นจึงเป็นการเข้าคู่อย่างถูกต้อง
- ในส่วนค่าคุณลักษณะเชิงความหมาย จะเห็นว่าออนโทโลยีชื่อ PacificNovel เป็นอินสแตนซ์ของคลาส Novel ซึ่งจะเป็นการเข้าคู่อย่างถูกต้องกับออนโทโลยีชื่อ AlaskaNovel ที่เป็นอินสแตนซ์ของคลาส Novel เช่นเดียวกัน

ดังนั้นจึงสรุปได้ว่าเว็บเซอวิส X ภายใต้ชนิดบริการ BookStore ที่เลือกมา สามารถทำงานแทนเว็บเซอวิส Y ที่อยู่ภายใต้ชนิดบริการ BookSeller ได้

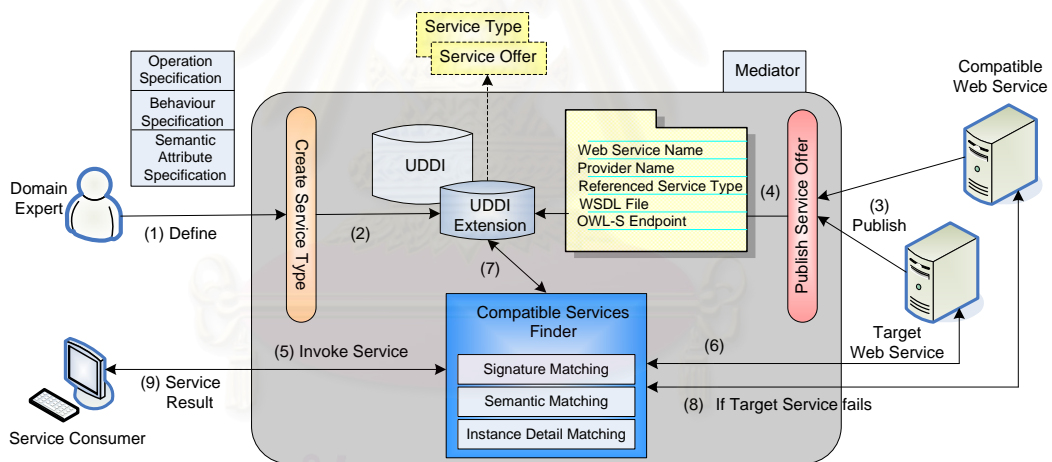
## บทที่ 4

### เฟรมเวิร์คสำหรับการทำงานแทนที่กันโดยอัตโนมัติของเว็บเซอร์วิส

ในบทนี้จะเป็นการกล่าวถึงการออกแบบเฟรมเวิร์คสำหรับการทำงานแทนที่กันโดยอัตโนมัติของเว็บเซอร์วิสตามแนวคิดของแบบจำลองชนิดบริการและกฎการเข้ากันได้ของเว็บเซอร์วิส โดยเฟรมเวิร์คนี้จะช่วยทำการค้นหาเว็บเซอร์วิสที่สามารถเข้ากันได้ พร้อมทั้งปรับการเรียกใช้ไปยังเว็บเซอร์วิสตัวแทนโดยอัตโนมัติ ในกรณีที่เว็บเซอร์วิสที่ต้องการใช้งานเกิดขัดข้อง

#### 4.1 การออกแบบเฟรมเวิร์คสำหรับการทำงานแทนที่กันโดยอัตโนมัติของเว็บเซอร์วิส

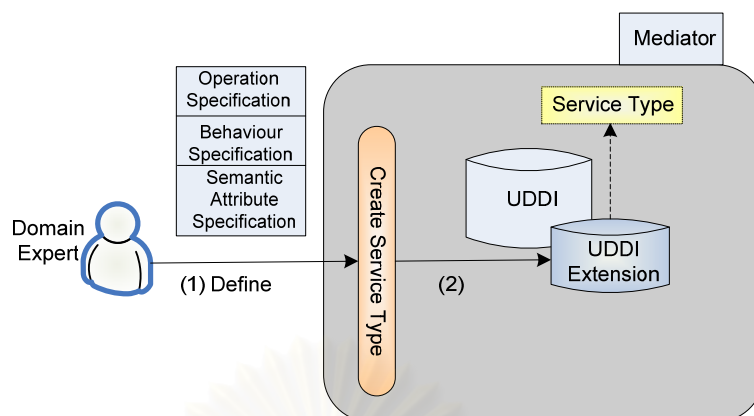
ผู้วิจัยได้ออกแบบให้เฟรมเวิร์คมีตัวกลาง (Mediator) ที่ทำหน้าที่เป็นตัวประมวลผลอยู่ระหว่างผู้ใช้บริการกับผู้ให้บริการ โดยโครงสร้างการทำงานภายในเฟรมเวิร์คเป็นดังรูปที่ 4.1 ซึ่งจะมีการทำงาน 5 ขั้นตอนดังนี้



รูปที่ 4.1 เฟรมเวิร์คการทำงานแทนที่กันของเว็บเซอร์วิสที่เข้ากันได้ตามแบบจำลองชนิดบริการ

##### 4.1.1 ขั้นตอนการสร้างชนิดบริการ

ในขั้นตอนนี้ ผู้เชี่ยวชาญในโดเมนต่างๆ หรือผู้ให้บริการจะต้องทำการกำหนดรายละเอียดของชนิดบริการประเภทต่างๆ ขึ้นมา รูปที่ 4.2 อ้างอิงมาจากรูปที่ 4.1 เฉพาะส่วนขั้นตอนการสร้างชนิดบริการ โดยมีรายละเอียดดังนี้



รูปที่ 4.2 ส่วนของเฟรมเวิร์คในขั้นตอนการสร้างชนิดบริการ

- (1) ผู้เชี่ยวชาญในโดเมนต่างๆ หรือผู้ให้บริการทำการสร้างข้อกำหนดชนิดบริการเพื่อให้ผู้ให้บริการสร้างเว็บเซอร์วิสตามข้อกำหนดชนิดบริการนั้นๆ โดยจะทำการสร้างชนิดบริการขึ้นมาใหม่ทั้งหมดเลยหรือเลือกที่จะขยายจากข้อกำหนดชนิดบริการที่มีอยู่เดิมแล้วก็ได้ ดังรูปที่ 4.3 และ 4.5 ตามลำดับ
- (2) ส่วนของตัวกลางที่ทำหน้าที่รับการประกาศชนิดบริการจะบันทึกข้อมูลชนิดบริการลงในส่วนขยายของยูดีดีไอ (ในส่วนเก็บชนิดบริการ)

รูปที่ 4.3 แสดงการสร้างข้อกำหนดชนิดบริการ BookSeller ผ่านเว็บเพจของตัวกลาง โดยข้อมูลที่ต้องป้อนลงไปนั้นจะประกอบด้วย ชื่อของชนิดบริการ ข้อกำหนดเชิงซิกเนเจอร์ของโอเปอเรชัน ข้อกำหนดเชิงพฤติกรรม และคุณลักษณะเชิงความหมาย ตามรายละเอียดที่ได้นำเสนอไว้ในหัวข้อที่ 3.1.2 หลังจากป้อนข้อมูลลงไปแล้ว เมื่อกดปุ่ม submit ข้อมูลทั้งหมดจะถูกบันทึกในส่วนขยายของยูดีดีไอ (ในส่วนเก็บชนิดบริการ) และจะมีหน้าต่างปรากฏขึ้นมาดังรูปที่ 4.4 เพื่อสอบถามว่าต้องการเพิ่มโอเปอเรชันอีกหรือไม่ ซึ่งในกรณีที่ชนิดบริการมีหลายโอเปอเรชันให้กดปุ่ม OK เว็บเพจก็จะเปลี่ยนไปที่หน้าป้อนข้อมูลดังรูปที่ 4.3 อีกครั้ง

Service Type Model Information	
Service Type	BookSeller
Operation Specification:	
Operation	int bookPurchase(int isbn,String creditcardDetail)
Fault	ExpCreditCard
Behaviour Specification:	
Input	ISBN,CreditcardDetail
Output	PurchasedBook
Precondition	VisaCardHolder
Effect	ProductDelivered
Fault	ExpiredCreditCard
Semantic Attribute Specification:	BookCategory
<input type="button" value="submit"/> <input type="button" value="reset"/>	

รูปที่ 4.3 ตัวอย่างการสร้างข้อกำหนดชนิดบริการ BookSeller

Service Type Model Information	
Service Type	BookSeller
Operation Specification:	
Operation	int bookPurchase(int isbn,String creditcardDetail)
Fault	ExpCreditCard
Behaviour Specification:	
Input	ISBN,CreditcardDetail
Output	PurchasedBook
Precondition	VisaCardHolder
Effect	ProductDelivered
Fault	ExpiredCreditCard
Semantic Attribute Specification:	BookCategory
<input type="button" value="submit"/> <input type="button" value="reset"/>	

http://localhost:8080

Would you like to add more operation?

รูปที่ 4.4 หน้าต่างสอบถามการเพิ่มโอเปอเรชัน

กรณีการสร้างข้อกำหนดชนิดบริการด้วยวิธีการขยายจากข้อกำหนดชนิดบริการเดิมนั้น จะต้องทำการระบุชื่อชนิดบริการที่ต้องการนำมาอ้างอิงก่อนดังรูปที่ 4.5 ซึ่งระบุว่าจะขยายข้อกำหนดจากชนิดบริการ BookSeller จากนั้นเมื่อกดปุ่ม Perform เว็บเพจจะแสดงข้อกำหนดของชนิดบริการ BookSeller และจะมีคำว่า Extension of BookSeller ต่อท้ายชื่อชนิดบริการที่สร้างขึ้นใหม่ ดังรูปที่ 4.6 ซึ่งวิธีนี้จะสามารถแก้ไขข้อมูลที่สืบต่อมาได้



เฉพาะในส่วนของเงื่อนไขก่อนการทำงาน ผลกระทบ และเพิ่มคุณลักษณะเชิงความหมายเท่านั้น แต่ถ้าต้องการเพิ่มโอเปอเรชันเข้าไปใหม่ก็สามารถทำได้โดยกดที่ปุ่ม Add more operations เว็บเพจก็จะเปลี่ยนไปที่หน้าป้อนข้อมูลเหมือนดังรูปที่ 4.3

Enter a service type name to get information

รูปที่ 4.5 การระบุชื่อชนิดบริการที่ต้องการนำมาขยายข้อกำหนด

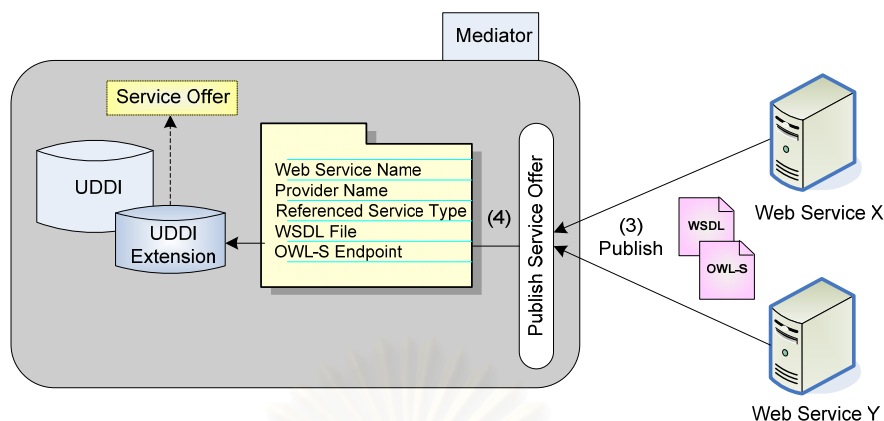
Service Type Model Information	
Service Type	BookAndMovieSeller Extension of BookSeller
Operation Specification:	
Operation	int bookPurchase(int isbn,String creditcardDetail)
Fault	ExpCreditCard
Behaviour Specification:	
Input	ISBN,CreditcardDetail
Output	PurchasedBook
Precondition	CreditCardHolder
Effect	ProductDeliveredByAir
Fault	ExpiredCreditCard
Semantic Attribute Specification:	BookCategory, MovieCategory

[1]

รูปที่ 4.6 ตัวอย่างการสร้างข้อกำหนดชนิดบริการด้วยการขยายจากข้อกำหนดที่มีอยู่เดิม

#### 4.1.2 ขั้นตอนการประกาศข้อเสนอบริการ

ในขั้นตอนนี้ ผู้ให้บริการจะต้องทำการเลือกชนิดบริการที่เหมาะสมกับการให้บริการของตน และทำการสร้างเว็บเซอวิสให้มีบริการเป็นไปตามข้อกำหนดชนิดบริการนั้นๆ รูปที่ 4.7 อ้างอิงมาจากรูปที่ 4.1 เฉพาะส่วนขั้นตอนการประกาศข้อเสนอบริการ โดยมีรายละเอียดดังนี้



รูปที่ 4.7 ส่วนของเฟรมเวิร์คในขั้นตอนการประกาศข้อเสนอบริการ

(3) ผู้ให้บริการทำการสร้างเว็บเซอร์วิสของตน (เช่น เว็บเซอร์วิส X และ Y) ตามชนิดบริการที่ต้องการ โดยต้องสร้างเอกสารดับเบิลยูเอสดีแอลซึ่งต้องมีรายละเอียดของโอเปอเรชันที่ให้บริการเป็นไปตามส่วนของ Operation Specification และต้องสร้างเอกสารอวล์-เอสซึ่งต้องมีรายละเอียดเป็นไปตามส่วนของ Behaviour Specification และ Semantic Attribute Specification โดยในขั้นตอนของการสร้างเอกสารอวล์-เอสนั้น ผู้ให้บริการจะต้องทำการอิมพอร์ตออนโทโลยีสำหรับชนิดบริการที่อยู่ในตัวกลางไปใช้ในการสร้างด้วย จากนั้นจึงทำการลงทะเบียนข้อเสนอบริการไว้กับตัวกลาง สำหรับข้อมูลที่ใช้ในการลงทะเบียนข้อเสนอบริการมีดังนี้

- Web Service Name คือ ชื่อของเว็บเซอร์วิสที่ให้บริการ
- Provider Name คือ ชื่อของผู้ให้บริการ
- Referenced Service Type คือ ชนิดบริการที่ใช้อ้างอิงในการสร้างเว็บเซอร์วิส
- WSDL File คือ แฟ้มเอกสารดับเบิลยูเอสดีแอล
- OWL-S Endpoint คือ เอนด์พอยท์ของเอกสารอวล์-เอส

โดยตัวอย่างการประกาศข้อเสนอบริการ PacificBook ซึ่งอยู่ภายใต้ชนิดบริการ BookSeller แสดงดังรูปที่ 4.8

(4) ส่วนของตัวกลางที่ทำหน้าที่รับการประกาศข้อเสนอบริการจากผู้ให้บริการจะบันทึกข้อมูลทั้งหมดลงในส่วนขยายของยูดีดีไอ (ในส่วนเก็บข้อเสนอบริการ)

Service Offer Information	
Web Service Name	<input type="text" value="PacificBook"/>
Provider Name	<input type="text" value="Pacific Publishing"/>
Referenced Service Type	<input type="text" value="BookSeller"/>
WSDL Endpoint	Select a file to upload : <input type="text" value="D:\Pacific.wsdl"/> <input type="button" value="Browse..."/>
OWL-S Endpoint	<input type="text" value="D:/OWL-S/Pacific.owl"/>
<input type="button" value="submit"/> <input type="button" value="reset"/>	

รูปที่ 4.8 ตัวอย่างการประกาศข้อเสนอบริการ

#### 4.1.3 ขั้นตอนการร้องขอใช้บริการเว็บเซอร์วิสผ่านตัวกลาง

ในขั้นตอน (5) ของรูปที่ 4.1 ผู้ใช้บริการทำการเรียกใช้บริการของเว็บเซอร์วิส โดยจะต้องทำการสร้างคำร้องขอใช้บริการนั้นๆ ในรูปของข้อความโซฟ (SOAP Message) พร้อมทั้งแนบเอนด์พอยท์ของเว็บเซอร์วิสที่ต้องการใช้บริการไว้ในส่วนของ AttachmentPart แล้วจึงส่งผ่านมายังตัวกลาง ซึ่งจะมีส่วนค้นหาเว็บเซอร์วิสที่เข้ากันได้ (Compatible Services Finder) เป็นตัวทำหน้าที่ในการรับ-ส่งข้อความโซฟระหว่างผู้ให้บริการกับผู้ให้บริการ และค้นหาเว็บเซอร์วิสที่เข้ากันได้ โดยตัวอย่างของข้อความโซฟ แสดงดังรูปที่ 4.9

```

1. -----_Part_0_5488661.1172163470074
2. Content-Type: text/xml; charset=utf-8
3.
4. <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
5.   <SOAP-ENV:Header/>
6.   <SOAP-ENV:Body>
7.     <bookPurchase>
8.       <isbn>4870442021</isbn>
9.       <creditcardDetail>49927358726</creditcardDetail>
10.    </bookPurchase>
11.  </SOAP-ENV:Body>
12. </SOAP-ENV:Envelope>
13. -----_Part_0_5488661.1172163470074
14. Content-Type: text/plain
15. Content-ID: WS_EndPoint
16.
17. http://localhost:8080/axis/services/Pacific
18. -----_Part_0_5488661.1172163470074--

```

รูปที่ 4.9 ตัวอย่างข้อความโซฟ

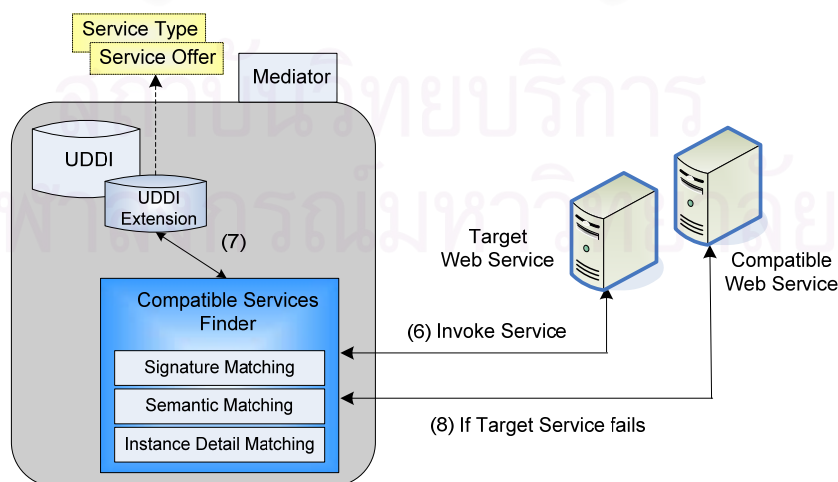
จากรูปที่ 4.9 bookPurchase คือโอเปอเรชันที่ต้องการใช้บริการ (บรรทัดที่ 7) โดยมี isbn เป็นพารามิเตอร์ตัวที่ 1 ซึ่งมีค่าเป็น 4870442021 (บรรทัดที่ 8) creditcardDetail เป็นพารามิเตอร์

ตัวที่ 2 ซึ่งมีค่าเป็น 49927358726 (บรรทัดที่ 9) และ <http://localhost:8080/axis/services/Pacific> คือเอนด์พอยท์ของเว็บเซอร์วิสชื่อ PacificBook ที่ต้องการใช้บริการ (บรรทัดที่ 17)

#### 4.1.4 ขั้นตอนการส่งต่อการร้องขอใช้บริการเว็บเซอร์วิสโดยตัวกลาง

ในขั้นตอนนี้ ตัวกลางจะทำการเรียกใช้บริการจากเว็บเซอร์วิสที่ผู้ใช้ต้องการ หรือในกรณีที่เว็บเซอร์วิสนั้นไม่สามารถให้บริการได้ ตัวกลางก็จะทำการค้นหาเว็บเซอร์วิสที่เข้ากันได้ให้อัตโนมัติ รูปที่ 4.10 อ้างอิงมาจากรูปที่ 4.1 เฉพาะส่วนขั้นตอนการส่งต่อการร้องขอ โดยมีรายละเอียดดังนี้

- (6) ตัวกลางจะทำการเรียกใช้บริการโดยส่งข้อความไชนไปยังเอนด์พอยท์ของเว็บเซอร์วิสที่ผู้ใช้ระบุ และรอรับผลลัพธ์ที่ส่งกลับมาจากเว็บเซอร์วิสนั้น
- (7) ตัวกลางทำการตรวจสอบหาอิลิเมนต์ <soapenv:Fault> ซึ่งแสดงถึงความผิดพลาดในข้อความไชนที่ได้รับ กรณีที่ไม่พบ ตัวกลางก็จะส่งคำตอบกลับไปยังผู้ใช้บริการ แต่ถ้ากรณีที่มีอิลิเมนต์นี้ก็จะพิจารณาอีกว่าเป็นความผิดพลาดแบบใด โดยจะตรวจสอบค่าของอิลิเมนต์ <faultcode> โดยถ้ามีค่าเป็น ns1:Server.NoService แสดงว่าเว็บเซอร์วิสที่ได้อ้างอิงมาในครั้งแรกนั้นไม่สามารถให้บริการได้ ดังนั้นตัวกลางจะทำการโหลดข้อมูลจากส่วนขยายของยูดีดีไอและนำมาพิจารณาว่ามีเว็บเซอร์วิสใดเข้ากันได้บ้าง โดยวิธีการเลือกนั้นจะเป็นไปตามที่ได้นำเสนอในหัวข้อที่ 3.3
- (8) เมื่อพบเว็บเซอร์วิสที่เข้ากันได้แล้ว ตัวกลางก็จะส่งคำร้องขอใช้บริการไปที่เว็บเซอร์วิสนั้นแทน และรอรับผลลัพธ์ที่ส่งกลับมาจากเว็บเซอร์วิสนั้น จากนั้นก็จะกลับไปทำงานตามขั้นตอนที่ (7)



รูปที่ 4.10 ส่วนของเฟรมเวิร์คในขั้นตอนการส่งต่อการร้องขอใช้บริการเว็บเซอร์วิส

#### 4.1.5 ขั้นตอนการส่งผลลัพธ์กลับไปให้ผู้ให้บริการ

ในขั้นตอน (9) ของรูปที่ 4.1 ตัวกลางจะทำการส่งผลลัพธ์ที่ได้กลับไปให้ผู้ให้บริการ โดยจะมีการแนบข้อมูลเกี่ยวกับการแทนที่ไว้ในส่วนของ AttachmentPart ของข้อความ SOAP ที่เป็นผลลัพธ์จากการเรียกใช้เว็บเซอร์วิสตัวแทนโดยข้อมูลนี้จะบอกผู้ให้บริการได้ว่าผลลัพธ์นี้มาจากการเรียกใช้บริการของเว็บเซอร์วิสตัวแทนใด ดังแสดงในรูปที่ 4.11

```

1. -----_Part_1_21845470.1172163470895
2. Content-Type: text/xml; charset=utf-8
3.
4. <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
5.   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
6.   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
7. <soapenv:Body>
8.   <bookPurchaseResponse
9.     soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
10.     <bookPurchaseReturn xsi:type="xsd:int">3146</bookPurchaseReturn>
11.   </bookPurchaseResponse>
12. </soapenv:Body></soapenv:Envelope>
13. -----_Part_1_21845470.1172163470895
14. Content-Type: text/plain
15. Content-ID: Substitute Web Service
16.
17. http://localhost:8080/axis/services/CU
18. -----_Part_1_21845470.1172163470895--

```

รูปที่ 4.11 ตัวอย่างผลลัพธ์

จากรูปที่ 4.11 ตัวเลข 3146 คือผลลัพธ์ที่ได้จากการเรียกใช้โอเปอเรชัน bookPurchase (บรรทัดที่ 10) และ http://localhost:8080/axis/services/CU คือเอนด์พอยน์ทของเว็บเซอร์วิสตัวแทนชื่อ CUBook ซึ่งตัวกลางไปเรียกใช้บริการแทนเว็บเซอร์วิสชื่อ PacificBook (บรรทัดที่ 17)

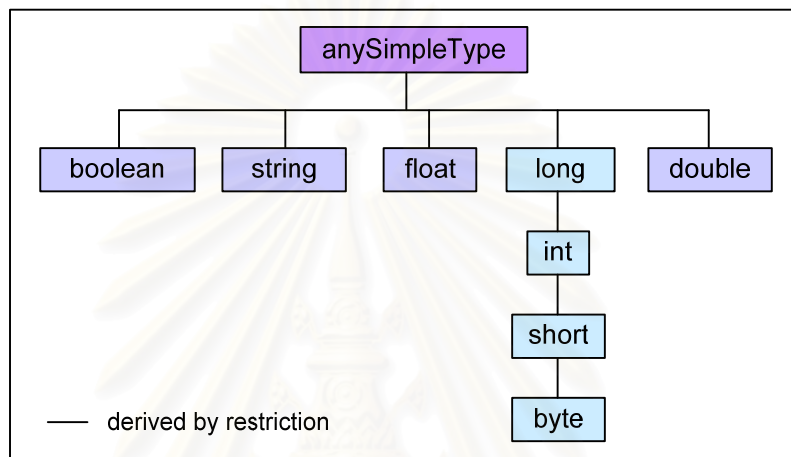
## 4.2 การออกแบบส่วนเข้าสู่

การค้นหาเว็บเซอร์วิสที่เข้ากันได้เพื่อมาเป็นเซอร์วิสตัวแทน จะพิจารณาการเข้าสู่เชิงซิกเนเจอร์ การเข้าสู่เชิงความหมาย และการเข้าสู่เชิงข้อมูลเฉพาะตัว โดยแต่ละส่วนมีรายละเอียดดังนี้

### 4.2.1 ส่วนเข้าสู่เชิงซิกเนเจอร์

ในส่วนการพิจารณาการเข้าสู่เชิงซิกเนเจอร์ของชนิดบริการนั้น จะเริ่มจากการโหลดข้อมูลเชิงซิกเนเจอร์ของชนิดบริการที่อยู่ในรูปของสายอักขระจากส่วนขยายของยูตีลิตี้ (ในส่วนเก็บชนิดบริการ) ที่ต้องการพิจารณาขึ้นมา จากนั้นจะทำการแยกออกเป็นโทเค็น เช่น โอเปอเรชัน bookPurchase มีซิกเนเจอร์เป็น int bookPurchase(int bisbn, String creditcardDetail); เมื่อทำการแยกออกเป็นโทเค็นจะได้เป็น 6 โทเค็นคือ int, bookPurchase, int, bisbn, String และ

creditcardDetail แล้วจึงนำข้อมูลดังกล่าวไปพิจารณาตามการเข้าคู่เชิงซิกเนเจอร์ที่ได้นำเสนอในหัวข้อที่ 3.2.1 สำหรับการพิจารณาการสืบต่อของชนิดข้อมูลนั้นจะอ้างอิงจากโครงสร้างการสืบต่อชนิดข้อมูลของเอ็กซ์เอ็มแอลสคีมา [10] แต่ในเบื้องต้นจะสามารถรองรับการสืบต่อชนิดข้อมูล ดังนี้คือ byte, short, int, long, float, double, boolean และ string โดยลำดับการสืบต่อของชนิดข้อมูลจะเป็นดังรูปที่ 4.12



รูปที่ 4.12 ลำดับการสืบต่อของชนิดข้อมูล [10]

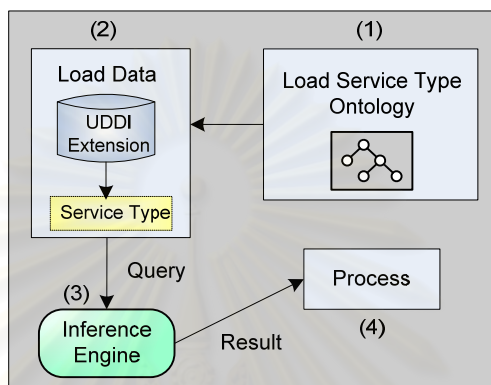
จากรูปที่ 4.12 ชนิดข้อมูล int จะถือว่าเข้ากันได้กับชนิดข้อมูล long เนื่องจากสืบต่อมาจากชนิดข้อมูล long

#### 4.2.2 ส่วนเข้าคู่เชิงความหมาย

ในส่วนการพิจารณาการเข้าคู่เชิงความหมายของข้อมูลเชิงพฤติกรรมและข้อมูลเฉพาะตัวเว็บเซอร์วิสในส่วนของคุณลักษณะเชิงความหมายนั้น จะมีความแตกต่างจากส่วนการเข้าคู่เชิงซิกเนเจอร์ เนื่องจากจะต้องใช้บอสซามซึ่งเป็นเครื่องอนุมาน (Inference Engine) ที่ใช้ในการค้นหาเหตุผลของออนโทโลยีเข้ามาช่วยในการพิจารณาความสัมพันธ์ระหว่างออนโทโลยีเทอม ดังนั้นผู้วิจัยจะอธิบายโครงสร้างการทำงานซึ่งมีรายละเอียดตามรูปที่ 4.13 ดังนี้

- (1) ทำการโหลดออนโทโลยีสำหรับชนิดบริการซึ่งอธิบายความสัมพันธ์ของออนโทโลยีเทอมต่างๆ ขึ้นมา (เช่นในรูปที่ 3.11)
- (2) ทำการโหลดข้อมูลเชิงพฤติกรรมและคุณลักษณะเชิงความหมายที่ต้องใช้ในการพิจารณาขึ้นมาจากส่วนขยายของยูดีดีไอ (ในส่วนเก็บชนิดบริการ)

- (3) ตัวกลางจะทำการตั้งชื่อคำถาม แล้วให้เครื่องอนุมานทำการหาความสัมพันธ์ระหว่าง  
 ออนโทโลยีเทอมตามออนโทโลยีสำหรับชนิดบริการ
- (4) ตัวกลางจะนำผลลัพธ์ที่ได้ไปประมวลผลตามการเข้าคู่เชิงความหมายที่ได้นำเสนอใน  
 หัวข้อที่ 3.2.2



รูปที่ 4.13 โครงสร้างการทำงานส่วนเข้าคู่เชิงความหมาย

สำหรับในการตั้งชื่อคำถามนั้นจะอยู่ในรูปของบุงชินแจ (Buchingae) ซึ่งเป็นภาษากฎ  
 สำหรับเว็บเชิงความหมายที่ใช้ตั้งชื่อคำถามในสภาพแวดล้อมของบอสซาม ยกตัวอย่างจากออน  
 โทโลยีในรูปที่ 3.11 เช่น หากต้องการค้นหาว่า ออนโทโลยีเทอมใดเป็นsubclassของออนโทโลยี  
 เทอม Novel จะสามารถเขียนเป็นชื่อคำถามแบบบุงชินแจได้ดังนี้

```

query q is
rdf:subClassOf(?x,http://www.servicetype.org/ontohierarchy.owl#Novel);
  
```

ผลลัพธ์ที่ได้คือ

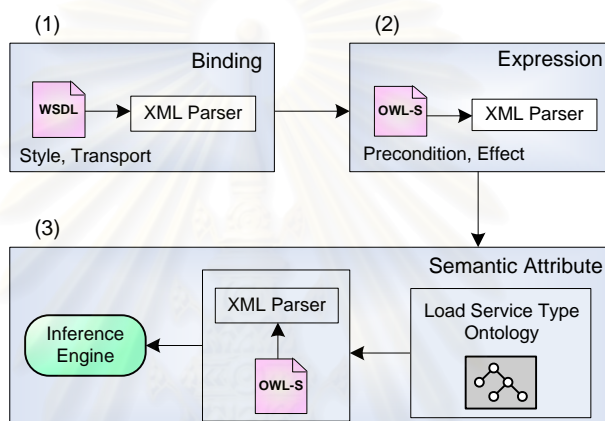
```

{[?x = http://www.servicetype.org/ontohierarchy.owl#Fantasy]}
{[?x = http://www.servicetype.org/ontohierarchy.owl#Romance]}
  
```

จากผลลัพธ์ที่ได้นั้น หมายความว่า อนโทโลจีเทอม Fantasy และอนโทโลจีเทอม Romance เป็นsubclassของอนโทโลจีเทอม Novel ซึ่งตัวกลางก็จะนำผลลัพธ์ที่ได้ไปประมวลผลต่อไป

#### 4.2.3 ส่วนเข้าคู่เชิงข้อมูลเฉพาะตัว

ในส่วนการพิจารณาการเข้าคู่เชิงข้อมูลเฉพาะตัวของสองเว็บเซอร์วิสนั้น จะพิจารณาข้อมูลทั้งหมด 3 ส่วน ซึ่งมีโครงสร้างการทำงานตามรูปที่ 4.14 ดังนี้



รูปที่ 4.14 โครงสร้างการทำงานส่วนเข้าคู่เชิงข้อมูลเฉพาะตัว

- (1) ตัวกลางจะเริ่มพิจารณาข้อมูลในส่วนโบนดิงเป็นอันดับแรก โดยจะมีตัวแฉงส่วนเอกซ์เอ็มแอล (XML Parser) ทำหน้าที่ในการอ่านข้อมูลอิลิเมนต์ Style และ Transport จากเอกสารดับเบิลยูเอสดีแอลที่ผู้ให้บริการอัพลอดมาตอนประกาศข้อเสนอบริการ แล้วพิจารณาว่ามีค่าตรงกันหรือไม่
- (2) พิจารณานิพจน์ของเงื่อนไขก่อนการดำเนินการและผลกระทบ โดยใช้ตัวแฉงส่วนเอกซ์เอ็มแอลอ่านข้อมูลจากเอกสารอวล์-เอสตามเอนด์พอยนทที่ผู้ให้บริการระบุไว้ตอนประกาศข้อเสนอบริการ จากนั้นก็ทำการพิจารณาตามกฎการเข้าคู่เชิงพิสัยตัวเลขที่ได้นำเสนอในหัวข้อที่ 3.2.3
- (3) พิจารณาค่าคุณลักษณะเชิงความหมาย โดยในส่วนนี้จะทำงานเหมือนส่วนการเข้าคู่เชิงความหมายเพียงแต่เปลี่ยนมาอ่านค่าข้อมูลคุณลักษณะเชิงความหมายจากเอกสารอวล์-เอสเท่านั้น



#### 4.3 เครื่องมือที่ใช้ในการพัฒนา

ในการพัฒนาเฟรมเวิร์คที่รองรับการทำงานแทนกันโดยอัตโนมัติของเว็บเซอร์วิสที่เข้ากันได้ตามแบบจำลองชนิดบริการนั้น ผู้วิจัยได้ใช้เครื่องมือในการพัฒนาดังนี้

1. จาวาเอสดีเค (Java SDK) รุ่น 1.5.0 เพื่อใช้เป็นตัวแปลโปรแกรม
2. อีคลิปส์ไอเอสดีเค (Eclipse SDK) รุ่น 3.1.1 สำหรับเป็นเครื่องมือและสภาพแวดล้อมในการพัฒนาเฟรมเวิร์ค
3. อาพาเซ่แอ็กซิส (Apache Axis) รุ่น 1.3 เพื่อใช้ในการสร้างเว็บเซอร์วิส
4. บอสซาม (Bossam) รุ่น 0.9b15 เป็นเครื่องอนุมานที่ใช้ในการค้นหาเหตุผลของออนโทโลยี
5. มายเอสคิวแอล (MySQL) รุ่น 5.0 สำหรับใช้ในการจัดเก็บฐานข้อมูล
6. อาพาเซ่ทอมแคท (Apache Tomcat) รุ่น 5.0 เพื่อใช้เป็นเว็บคอนเทนเนอร์
7. โพรเทเจ (Protégé) ใช้เป็นเครื่องมือสำหรับการพัฒนาออนโทโลยีและฐานความรู้ (Knowledge-based)

## บทที่ 5

### การทดสอบการทำงานของเฟรมเวิร์ค

ในบทนี้ผู้วิจัยจะทดสอบการทำงานของเฟรมเวิร์คสำหรับการทำงานแทนกันโดยอัตโนมัติของเว็บเซอร์วิสที่ได้ออกแบบและพัฒนาขึ้นตามแนวคิดของแบบจำลองชนิดบริการและกฎการเข้ากันได้ที่น่าเสนอในบทที่ 3 และ 4 โดยจะแบ่งออกเป็น 3 กรณีศึกษาซึ่งจะแสดงถึงการพิจารณาเลือกเว็บเซอร์วิสที่เข้ากันทั้ง 3 วิธีตามหัวข้อที่ 3.3

#### 5.1 การทดสอบการทำงานของเฟรมเวิร์คด้วยกรณีศึกษาที่ 1

กรณีศึกษาที่ 1 จะทดสอบการค้นหาเว็บเซอร์วิสตัวแทนที่ประกาศข้อเสนอบริการอยู่ภายใต้ชนิดบริการเดียวกันกับเว็บเซอร์วิสที่ขัดข้อง โดยกำหนดให้มีสภาพแวดล้อมของการทดสอบเป็นดังนี้คือ ข้อมูลของเว็บเซอร์วิสที่นำมาทดสอบเป็นดังตารางที่ 5.1 ข้อมูลเฉพาะตัวของแต่ละเว็บเซอร์วิสเป็นดังตารางที่ 5.2 - 5.4 และข้อกำหนดของชนิดบริการ BookSeller เป็นดังรูปที่ 5.1

ตารางที่ 5.1 ข้อมูลของเว็บเซอร์วิสที่นำมาทดสอบในกรณีศึกษาที่ 1

เว็บเซอร์วิส	ชนิดบริการ	สถานะ
PacificBook	BookSeller	ขัดข้อง
CUBook	BookSeller	ปกติ
SUTBook	BookSeller	ปกติ

ตารางที่ 5.2 ข้อมูลเฉพาะตัวของเว็บเซอร์วิส PacificBook ในกรณีศึกษาที่ 1

ส่วนของข้อมูล	เว็บเซอร์วิส PacificBook
ไบนารีดิ้ง	style="rpc"; transport="http"
นิพจน์ของเงื่อนไขก่อนการดำเนินการ	-
นิพจน์ของผลกระทบ	ProductDelivered.Days ≤ 10
ค่าคุณลักษณะเชิงความหมาย	PacificNovel

ตารางที่ 5.3 ข้อมูลเฉพาะตัวของเว็บเซอร์วิส CUBook ในกรณีศึกษาที่ 1

ส่วนของข้อมูล	เว็บเซอร์วิส CUBook
ไบนารีดิ้ง	style="rpc"; transport="http"
นิพจน์ของเงื่อนไขก่อนการดำเนินงาน	-
นิพจน์ของผลกระทบ	CUDelivered.Days $\leq$ 5
ค่าคุณลักษณะเชิงความหมาย	CUNovel

ตารางที่ 5.4 ข้อมูลเฉพาะตัวของเว็บเซอร์วิส SUTBook ในกรณีศึกษาที่ 1

ส่วนของข้อมูล	เว็บเซอร์วิส SUTBook
ไบนารีดิ้ง	style="rpc"; transport="http"
นิพจน์ของเงื่อนไขก่อนการดำเนินงาน	-
นิพจน์ของผลกระทบ	SUTDelivered.Days $\leq$ 7
ค่าคุณลักษณะเชิงความหมาย	SUTNovel

1. Service Type: BookSeller
2. Operation Specification:
3.     int bookPurchase(int isbn, String creditcardDetail);
4.     fault ExpCreditCard;
5. Behaviour Specification of bookPurchase:
6.     Input: ISBN, CreditcardDetail;
7.     Output: PurchasedBook;
8.     Precondition: VisaCardHolder;
9.     Effect: ProductDelivered;
10.    Fault: ExpiredCreditCard;
11. Semantic Attribute Specification: BookCategory;

รูปที่ 5.1 ข้อกำหนดชนิดบริการ BookSeller

จากข้อมูลทั้งหมด ผู้วิจัยเริ่มทดสอบโดยการร้องขอใช้บริการโอเปอเรชัน bookPurchase ของเว็บเซอร์วิส PacificBook ซึ่งมีสถานะขัดข้องอยู่ ซึ่งผลลัพธ์ของการทดสอบที่ฝั่งของตัวกลาง และผู้ใช้บริการจะเป็นดังรูปที่ 5.2 และ 5.3 ตามลำดับ

```

[The Mediator for service substitution]
- -> Receive message from a service consumer
- -> Send message to the target Web Service
- -> Receive message from the target Web Service
The target Web Service status : Failed!!
- - - - Finding a Compatible Web Service - - - -
Using criteria 1
Determining -> CU
  Instance Detail Matching
    Bindind details : passed
    Expression : passed
    Semantic Attribute value : passed
The new service endpoint = http://localhost:8080/axis/services/CU
- -> Send service result to the service consumer

```

รูปที่ 5.2 ผลลัพธ์ของการทดสอบฝั่งตัวกลางในกรณีศึกษาที่ 1

```

1.  REQUEST:
2.
3.  -----_Part_0_5488661.1172163470074
4.  Content-Type: text/xml; charset=utf-8
5.
6.  <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
7.  <SOAP-ENV:Header/><SOAP-ENV:Body>
8.    <bookPurchase>
9.      <isbn>2147483646</isbn><creditcardDetail>04870442021</creditcardDetail>
10.    </bookPurchase>
11.  </SOAP-ENV:Body></SOAP-ENV:Envelope>
12.  -----_Part_0_5488661.1172163470074
13.  Content-Type: text/plain
14.  Content-ID: WS_EndPoint
15.
16.  http://localhost:8080/axis/services/Pacific
17.  -----_Part_0_5488661.1172163470074--
18.
19.  RESPONSE:
20.
21.  -----_Part_1_21845470.1172163470895
22.  Content-Type: text/xml; charset=utf-8
23.
24.  <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
25.    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
26.    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
27.  <soapenv:Body>
28.    <bookPurchaseResponse
29.      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
30.      <bookPurchaseReturn xsi:type="xsd:int">3146</bookPurchaseReturn>
31.    </bookPurchaseResponse>
32.  </soapenv:Body></soapenv:Envelope>
33.  -----_Part_1_21845470.1172163470895
34.  Content-Type: text/plain
35.  Content-ID: Substitute Web Service
36.
37.  http://localhost:8080/axis/services/CU
38.  -----_Part_1_21845470.1172163470895--

```

รูปที่ 5.3 ผลลัพธ์ของการทดสอบฝั่งผู้ใช้บริการในกรณีศึกษาที่ 1

จากผลการทดสอบในรูปที่ 5.2 นั้นจะเห็นว่าตัวกลางได้ปรับการเรียกใช้บริการไปยังเว็บเซอริวิส CUBook แทน โดยพิจารณาเฉพาะการเข้าสู่เชิงข้อมูลเฉพาะตัวเท่านั้น เนื่องจากว่าตามแนวคิดของแบบจำลองชนิดบริการนั้น เว็บเซอริวิสที่ประกาศข้อเสนอบริการภายใต้ชนิดบริการเดียวกันจะเข้าสู่เชิงซิกเนเจอร์และเชิงความหมายของพฤติกรรมอยู่แล้ว ซึ่งทั้งเว็บเซอริวิส PacificBook และ CUBook นั้นประกาศข้อเสนอบริการภายใต้ชนิดบริการ BookSeller เหมือนกัน ดังนั้นจึงไม่จำเป็นต้องพิจารณาการเข้าสู่ในสองส่วนดังกล่าว

จากตารางที่ 5.2 และ 5.3 เมื่อพิจารณาแล้วจะเห็นได้ว่า ข้อมูลเฉพาะตัวแต่ละส่วนของเว็บเซอริวิส PacificBook และ CUBook สอดคล้องกันตามการเข้าสู่เชิงข้อมูลเฉพาะตัว เนื่องจาก

- ส่วนไบนารีมีค่าเหมือนกัน
- ส่วนนิพจน์ของผลกระทบ จะเห็นว่านิพจน์ของเว็บเซอริวิส CUBook เป็นเซตย่อยของนิพจน์ของเว็บเซอริวิส PacificBook ดังนั้นจึงเป็นการเข้าสู่อย่างถูกต้อง
- ส่วนค่าคุณลักษณะเชิงความหมาย จะเห็นว่าออนโทโลยีเทอมชื่อ PacificNovel เป็นอินสแตนซ์ของคลาส Novel ซึ่งจะเข้าสู่อย่างถูกต้องกับออนโทโลยีเทอมชื่อ CUNovel ที่เป็นอินสแตนซ์ของคลาส Novel เช่นเดียวกัน

ดังนั้นเว็บเซอริวิส CUBook จึงสามารถทำงานแทนเว็บเซอริวิส PacificBook ได้ สำหรับเว็บเซอริวิส SUTBook นั้นจะไม่ได้ถูกพิจารณาเนื่องจากว่า ตัวกลางจะเลือกเว็บเซอริวิสที่อยู่ภายใต้ชนิดบริการ BookSeller เรียงตามลำดับการประกาศข้อเสนอบริการ ซึ่งจากสภาพแวดล้อมที่กำหนดนี้ ตัวกลางจะทำการเลือกเว็บเซอริวิส CUBook ก่อน เมื่อพิจารณาแล้วเข้ากันได้กับเว็บเซอริวิส PacificBook จึงไม่ได้เลือกเว็บเซอริวิส SUTBook มาพิจารณา

สำหรับรูปที่ 5.3 นั้นบรรทัดที่ 1 - 18 แสดงข้อความโซฟที่เป็นคำร้องขอใช้บริการ โดย bookPurchase คือโอเปอเรชันที่ต้องการใช้บริการ (บรรทัดที่ 8) โดยมี isbn เป็นพารามิเตอร์ตัวที่ 1 ซึ่งมีค่าเป็น 2147483646 (บรรทัดที่ 9) creditcardDetail เป็นพารามิเตอร์ตัวที่ 2 ซึ่งมีค่าเป็น 04870442021 (บรรทัดที่ 9) และ <http://localhost:8080/axis/services/Pacific> คือเอนด์พอยท์ของเว็บเซอริวิสที่ต้องการใช้บริการ (บรรทัดที่ 16) ส่วนบรรทัดที่ 19 - 38 นั้นแสดงข้อความโซฟที่เป็นผลลัพธ์จากการใช้บริการ โดย 3146 คือค่าผลลัพธ์ที่ได้จากการเรียกใช้โอเปอเรชัน bookPurchase (บรรทัดที่ 30) และ <http://localhost:8080/axis/services/CU> คือเอนด์พอยท์ของเว็บเซอริวิส CUBook ซึ่งเป็นเว็บเซอริวิสที่ตัวกลางเรียกใช้บริการแทนเว็บเซอริวิส PacificBook ที่ขัดข้อง (บรรทัดที่ 37)

## 5.2 การทดสอบการทำงานของเฟรมเวิร์คด้วยกรณีศึกษาที่ 2

กรณีศึกษาที่ 2 จะทดสอบการค้นหาเว็บเซอร์วิสตัวแทนที่ประกาศข้อเสนอบริการอยู่ภายใต้ชนิดบริการที่ขยายมาจากชนิดบริการของเว็บเซอร์วิสที่เกิดขัดข้อง โดยกำหนดให้มีสภาพแวดล้อมของการทดสอบเป็นดังนี้คือ ข้อมูลของเว็บเซอร์วิสที่นำมาทดสอบเป็นดังตารางที่ 5.5 ข้อมูลเฉพาะตัวของแต่ละเว็บเซอร์วิสเป็นดังตารางที่ 5.6 – 5.8 ข้อกำหนดของชนิดบริการ BookSeller เป็นดังรูปที่ 5.1 และข้อกำหนดของชนิดบริการ BookAndMovieSeller เป็นดังรูปที่ 5.4

ตารางที่ 5.5 ข้อมูลของเว็บเซอร์วิสที่นำมาทดสอบในกรณีศึกษาที่ 2

เว็บเซอร์วิส	ชนิดบริการ	สถานะ
PacificBook	BookSeller	ขัดข้อง
CUBook	BookSeller	ขัดข้อง
IndianBook	BookAndMovieSeller	ปกติ

ตารางที่ 5.6 ข้อมูลเฉพาะตัวของเว็บเซอร์วิส PacificBook ในกรณีศึกษาที่ 2

ส่วนของข้อมูล	เว็บเซอร์วิส PacificBook
ไบนารี	style="rpc"; transport="http"
นิพจน์ของเงื่อนไขก่อนการดำเนินงาน	-
นิพจน์ของผลกระทบ	-
ค่าคุณลักษณะเชิงความหมาย	PacificNovel

ตารางที่ 5.7 ข้อมูลเฉพาะตัวของเว็บเซอร์วิส CUBook ในกรณีศึกษาที่ 2

ส่วนของข้อมูล	เว็บเซอร์วิส CUBook
ไบนารี	style="rpc"; transport="http"
นิพจน์ของเงื่อนไขก่อนการดำเนินงาน	-
นิพจน์ของผลกระทบ	-
ค่าคุณลักษณะเชิงความหมาย	CUNovel

ตารางที่ 5.8 ข้อมูลเฉพาะตัวของเว็บเซอร์วิส IndianBook ในกรณีศึกษาที่ 2

ส่วนของข้อมูล	เว็บเซอร์วิส IndianBook
ไบนารีดิ้ง	style="rpc"; transport="http"
นิพจน์ของเงื่อนไขก่อนการดำเนินงาน	-
นิพจน์ของผลกระทบ	-
ค่าคุณลักษณะเชิงความหมาย	IndianRomance (BookCategory) IndianFantasy (BookCategory) IndianAction (MovieCategory)

1. Service Type: BookAndMovieSeller ExtensionOf BookSeller
2. Operation Specification:
3. int bookPurchase(int isbn, String creditcardDetail);
4. fault ExpCreditCard;
5. int moviePurchase(String mName, String creditcardDetail);
6. fault ExpCreditCard, MovieNotFound;
7. boolean moviePreview(String mName);
8. fault FileNotFound;
9. Behaviour Specification of bookPurchase:
10. Input: ISBN, CreditcardDetail;
11. Output: PurchasedBook;
12. Precondition: CreditCardHolder;
13. Effect: ProductDeliveredByAir;
14. Fault: ExpiredCreditCard;
15. Behaviour Specification of moviePurchase:
16. Input: MovieName, CreditcardDetail;
17. Output: PurchasedMovie;
18. Precondition: CreditCardHolder;
19. Effect: ProductDeliveredByAir;
20. Fault: ExpiredCreditCard, MovieNotFound;
21. Behaviour Specification of moviePreview:
22. Input: MovieName;
23. Output: MovieClip;
24. Precondition: MPEG-4Player;
25. Effect: HandBillImages;
26. Fault: FileNotFound;
27. Semantic Attribute Specification: BookCategory, MovieCategory;

รูปที่ 5.4 ชนิดบริการ BookAndMovieSeller

จากข้อมูลทั้งหมด ผู้วิจัยเริ่มทดสอบโดยการร้องขอใช้บริการโอเปอเรชัน bookPurchase ของเว็บเซอร์วิส PacificBook ซึ่งมีสถานะขัดข้องอยู่ ซึ่งผลลัพธ์ของการทดสอบที่ฝั่งของตัวกลาง และผู้ใช้บริการจะเป็นดังรูปที่ 5.5 และ 5.6 ตามลำดับ

```

[The Mediator for service substitution]
- -> Receive message from a service consumer
- -> Send message to the target Web Service
- -> Receive message from the target Web Service
The target Web Service status : Failed!!
- - - - Finding a Compatible Web Service - - - -
Using criteria 1
Determining -> CU
  Instance Detail Matching
    Bindind details : passed
    Expression : passed
    Semantic Attribute value : passed
The new service endpoint = http://localhost:8080/axis/services/CU
The target Web Service status : Failed!!
- - - - Finding a Compatible Web Service - - - -
Using criteria 2
Determining -> Indian
  Instance Detail Matching
    Bindind details : passed
    Expression : passed
    Semantic Attribute value : passed
The new service endpoint = http://localhost:8080/axis/services/Indian
- -> Send service result to the service consumer

```

รูปที่ 5.5 ผลลัพธ์ของการทดสอบฝั่งตัวกลางในกรณีศึกษาที่ 2

```

1.  REQUEST:
2.
3.  -----_Part_0_5488661.1172164966946
4.  Content-Type: text/xml; charset=utf-8
5.
6.  <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
7.  <SOAP-ENV:Header/><SOAP-ENV:Body>
8.    <bookPurchase>
9.      <isbn>2147483646</isbn><creditcardDetail>04870442021</creditcardDetail>
10.    </bookPurchase>
11.  </SOAP-ENV:Body></SOAP-ENV:Envelope>
12.  -----_Part_0_5488661.1172164966946
13.  Content-Type: text/plain
14.  Content-ID: WS_EndPoint
15.
16.  http://localhost:8080/axis/services/Pacific
17.  -----_Part_0_5488661.1172164966946--
18.
19.  RESPONSE:
20.
21.  -----_Part_0_21388450.1172164971382
22.  Content-Type: text/xml; charset=utf-8
23.
24.  <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
25.    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
26.    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

```

รูปที่ 5.6 ผลลัพธ์ของการทดสอบฝั่งผู้ใช้บริการในกรณีศึกษาที่ 2



```

27. <soapenv:Body>
28.   <bookPurchaseResponse
29.     soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
30.     <bookPurchaseReturn xsi:type="xsd:int">3146</bookPurchaseReturn>
31.   </bookPurchaseResponse>
32. </soapenv:Body></soapenv:Envelope>
33. -----_Part_0_21388450.1172164971382
34. Content-Type: text/plain
35. Content-ID: Substitute Web Service
36.
37. http://localhost:8080/axis/services/Indian
38. -----_Part_0_21388450.1172164971382--

```

### รูปที่ 5.6 ผลลัพธ์ของการทดสอบฝั่งผู้ใช้บริการในกรณีศึกษาที่ 2 (ต่อ)

จากผลการทดสอบในรูปที่ 5.5 จะเห็นว่า เริ่มแรกตัวกลางได้เลือกเว็บเซอร์วิสตัวแทนที่อยู่ภายใต้ชนิดบริการ BookSeller มาพิจารณาก่อนซึ่งก็คือเว็บเซอร์วิส CUBook ซึ่งเมื่อทำการเข้าคู่เชิงข้อมูลเฉพาะตัวแล้วจะเห็นว่าเว็บเซอร์วิส CUBook เข้ากันได้กับเว็บเซอร์วิส PacificBook ที่ขัดข้อง แต่เนื่องจากสภาพแวดล้อมของการทดสอบนี้ได้กำหนดให้เว็บเซอร์วิส CUBook นั้นมีสถานะขัดข้องและไม่มีเว็บเซอร์วิสอื่นภายใต้ชนิดบริการ BookSeller ประกาศอยู่อีก ดังนั้นตัวกลางจึงได้เปลี่ยนไปใช้วิธีการเลือกเว็บเซอร์วิสที่เข้ากันได้แบบที่ 2 คือ เลือกเว็บเซอร์วิสตัวแทนที่ประกาศข้อเสนอบริการอยู่ภายใต้ชนิดบริการที่ขยายมาจากชนิดบริการ BookSeller ซึ่งก็คือเว็บเซอร์วิส IndianBook จากนั้นตัวกลางจะพิจารณาการเข้าคู่เชิงข้อมูลเฉพาะตัวเพียงอย่างเดียวเนื่องจากเว็บเซอร์วิส IndianBook ประกาศข้อเสนอบริการภายใต้ชนิดบริการ BookAndMovieSeller ซึ่งขยายมาจากชนิดบริการ BookSeller ตามข้อกำหนดที่แสดงในรูปที่ 5.4 ดังนั้นจะเข้าคู่เชิงซิกเนเจอร์และเชิงความหมายของพฤติกรรมอยู่แล้ว

จากตารางที่ 5.6 และ 5.8 เมื่อพิจารณาแล้วจะเห็นว่า ข้อมูลเฉพาะตัวแต่ละส่วนของเว็บเซอร์วิส PacificBook และ IndianBook สอดคล้องกันตามการเข้าคู่เชิงข้อมูลเฉพาะตัว เนื่องจาก

- ส่วนไบน์ดิงมีค่าเหมือนกัน
- ส่วนค่าคุณลักษณะเชิงความหมาย BookCategory จะเห็นว่าออนโทโลยีโทมชื่อ IndianRomance เป็นอินสแตนซ์ของคลาส Romance หรือออนโทโลยีโทมชื่อ IndianFantasy เป็นอินสแตนซ์ของคลาส Fantasy ซึ่งเป็นการเข้าคู่อย่างเจาะจงกับออนโทโลยีโทมชื่อ PacificNovel ที่เป็นอินสแตนซ์ของคลาส Novel

ดังนั้นเว็บเซอร์วิส IndianBook จึงสามารถทำงานแทนเว็บเซอร์วิส PacificBook ได้

สำหรับรูปที่ 5.6 นั้นบรรทัดที่ 1-18 แสดงข้อความชีพที่เป็นคำร้องขอใช้บริการ โดย bookPurchase คือโอเปอเรชันที่ต้องการใช้บริการ (บรรทัดที่ 8) โดยมี isbn เป็นพารามิเตอร์ตัวที่ 1 ซึ่งมีค่าเป็น 2147483646 (บรรทัดที่ 9) creditcardDetail เป็นพารามิเตอร์ตัวที่ 2 ซึ่งมีค่าเป็น

04870442021 (บรรทัดที่ 9) และ <http://localhost:8080/axis/services/Pacific> คือเอนด์พอยนท์ของเว็บเซอร์วิสที่ต้องการใช้บริการ (บรรทัดที่ 16) ส่วนบรรทัดที่ 19 - 38 นั้นแสดงข้อความที่เป็นผลลัพธ์จากการใช้บริการ โดย 3146 คือค่าผลลัพธ์ที่ได้จากการเรียกใช้โอเปอเรชัน bookPurchase (บรรทัดที่ 30) และ <http://localhost:8080/axis/services/Indian> คือเอนด์พอยนท์ของเว็บเซอร์วิส IndianBook ซึ่งเป็นเว็บเซอร์วิสที่ตัวกลางเรียกใช้บริการแทนเว็บเซอร์วิส PacificBook ที่ขัดข้อง (บรรทัดที่ 37)

### 5.3 การทดสอบการทำงานของเฟรมเวิร์คด้วยกรณีศึกษาที่ 3

กรณีศึกษาที่ 3 จะทดสอบการค้นหาเว็บเซอร์วิสตัวแทนที่ไม่มีมีความเกี่ยวข้องกับชนิดบริการของเว็บเซอร์วิสที่ขัดข้อง โดยกำหนดให้มีสภาพแวดล้อมของการทดสอบเป็นดังนี้คือ ข้อมูลของเว็บเซอร์วิสที่นำมาทดสอบเป็นดังตารางที่ 5.9 ข้อมูลเฉพาะตัวของแต่ละเว็บเซอร์วิสเป็นดังตารางที่ 5.10 – 5.13 ข้อกำหนดของชนิดบริการ BookSeller เป็นดังรูปที่ 5.1 ข้อกำหนดของชนิดบริการ BookAndMovieSeller เป็นดังรูปที่ 5.4 ข้อกำหนดของชนิดบริการ BookShop เป็นดังรูปที่ 5.7 และข้อกำหนดของชนิดบริการ BookStore เป็นดังรูปที่ 5.8

ตารางที่ 5.9 ข้อมูลของเว็บเซอร์วิสที่นำมาทดสอบในกรณีศึกษาที่ 3

เว็บเซอร์วิส	ชนิดบริการ	สถานะ
PacificBook	BookSeller	ขัดข้อง
CUBook	BookSeller	ปกติ
IndianBook	BookAndMovieSeller	ปกติ
AlaskaBook	BookStore	ปกติ

ตารางที่ 5.10 ข้อมูลเฉพาะตัวของเว็บเซอร์วิส PacificBook ในกรณีศึกษาที่ 3

ส่วนของข้อมูล	เว็บเซอร์วิส PacificBook
ไบนารีดิ้ง	style="rpc"; transport="http"
นิพจน์ของเงื่อนไขก่อนการดำเนินงาน	-
นิพจน์ของผลกระทบ	ProductDelivered.Days ≤ 7
ค่าคุณลักษณะเชิงความหมาย	PacificNovel

ตารางที่ 5.11 ข้อมูลเฉพาะตัวของเว็บเซอร์วิส CUBook ในกรณีศึกษาที่ 3

ส่วนของข้อมูล	เว็บเซอร์วิส CUBook
ไบนารี	style="rpc"; transport="http"
นิพจน์ของเงื่อนไขก่อนการดำเนินงาน	-
นิพจน์ของผลกระทบ	ProductDelivered.Days ≤ 10
ค่าคุณลักษณะเชิงความหมาย	CUNovel

ตารางที่ 5.12 ข้อมูลเฉพาะตัวของเว็บเซอร์วิส IndianBook ในกรณีศึกษาที่ 3

ส่วนของข้อมูล	เว็บเซอร์วิส IndianBook
ไบนารี	style="rpc"; transport="http"
นิพจน์ของเงื่อนไขก่อนการดำเนินงาน	-
นิพจน์ของผลกระทบ	ProductDelivered.Days ≤ 5
ค่าคุณลักษณะเชิงความหมาย	IndianPoetry (BookCategory) IndianAction (MovieCategory)

ตารางที่ 5.13 ข้อมูลเฉพาะตัวของเว็บเซอร์วิส AlaskaBook ในกรณีศึกษาที่ 3

ส่วนของข้อมูล	เว็บเซอร์วิส AlaskaBook
ไบนารี	style="rpc"; transport="http"
นิพจน์ของเงื่อนไขก่อนการดำเนินงาน	-
นิพจน์ของผลกระทบ	ProductDelivered.Days ≤ 7
ค่าคุณลักษณะเชิงความหมาย	AlaskaNovel

```

Service Type: BookShop
Operation Specification:
    int bookPurchase(long booknumber, String creditcardInfo);
    fault ExpCreditCard;
Behaviour Specification of bookPurchase:
    Input: BookNumber, CreditcardDetail;
    Output: PurchasedBook;
    Precondition: DebitCardHolder;
    Effect: ProductDelivered;
    Fault: ExpiredCreditCard;
Semantic Attribute Specification: BookCategory;

```

รูปที่ 5.7 ข้อกำหนดชนิดบริการ BookShop

```

Service Type: BookStore
Operation Specification:
    int bookPurchase(long bnumber, String creditcardDetail);
    fault ExpCreditCard;
Behaviour Specification of bookPurchase:
    Input: BookNumber, CreditcardDetail;
    Output: PurchasedBook;
    Precondition: CreditCardHolder;
    Effect: ProductDeliveredByAir;
    Fault: ExpiredCreditCard;
Semantic Attribute Specification: BookCategory;

```

### รูปที่ 5.8 ข้อกำหนดชนิดบริการ BookStore

จากข้อมูลทั้งหมด ผู้วิจัยเริ่มทดสอบโดยการร้องขอใช้บริการโอเปอเรชัน bookPurchase ของเว็บเซอร์วิส PacificBook ซึ่งมีสถานะขัดข้องอยู่ ซึ่งผลลัพธ์ของการทดสอบที่ฝั่งของตัวกลาง และผู้ใช้บริการจะเป็นดังรูปที่ 5.9 และ 5.10 ตามลำดับ

```

[The Mediator for service substitution]
- -> Receive message from a service consumer
- -> Send message to the target Web Service
- -> Receive message from the target Web Service
The target Web Service status : Failed!!
- - - - - Finding a Compatible Web Service - - - - -
Using criteria 1
Determining -> CU
  Instance Detail Matching
    Bindind details : passed
    Expression : failed!
- - - - - Finding a Compatible Web Service - - - - -
Using criteria 2
Determining -> Indian
  Instance Detail Matching
    Bindind details : passed
    Expression : passed
    Semantic Attribute value : failed!
- - - - - Finding a Compatible Web Service - - - - -
Using criteria 3
Determining -> BookShop
  Signature Matching : passed
  Behaviour Matching : failed!
-----

```

### รูปที่ 5.9 ผลลัพธ์ของการทดสอบฝั่งตัวกลางในกรณีศึกษาที่ 3

```

Determining -> BookStore
  Signature Matching : passed
  Behaviour Matching : passed
Determining -> Alaska
  Instance Detail Matching
    Bindind details : passed
    Expression : passed
    Semantic Attribute value : passed
-----
The new service endpoint = http://localhost:8080/axis/services/Alaska
- -> Send service result to the service consumer

```

รูปที่ 5.9 ผลลัพธ์ของการทดสอบฝั่งตัวกลางในกรณีศึกษาที่ 3 (ต่อ)

```

1.  REQUEST:
2.
3.  -----_Part_0_5488661.1172175608778
4.  Content-Type: text/xml; charset=utf-8
5.
6.  <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
7.  <SOAP-ENV:Header/><SOAP-ENV:Body>
8.    <bookPurchase>
9.      <isbn>2147483646</isbn><creditcardDetail>048704420210</creditcardDetail>
10.    </bookPurchase>
11.  </SOAP-ENV:Body></SOAP-ENV:Envelope>
12.  -----_Part_0_5488661.1172175608778
13.  Content-Type: text/plain
14.  Content-ID: WS_EndPoint
15.
16.  http://localhost:8080/axis/services/Pacific
17.  -----_Part_0_5488661.1172175608778--
18.
19.  RESPONSE:
20.
21.  -----_Part_0_12028181.1172175612864
22.  Content-Type: text/xml; charset=utf-8
23.
24.  <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
25.    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
26.    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
27.  <soapenv:Body>
28.    <bookPurchaseResponse
29.      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
30.      <bookPurchaseReturn xsi:type="xsd:int">3146</bookPurchaseReturn>
31.    </bookPurchaseResponse>
32.  </soapenv:Body></soapenv:Envelope>
33.  -----_Part_0_12028181.1172175612864
34.  Content-Type: text/plain
35.  Content-ID: Substitute Web Service
36.
37.  http://localhost:8080/axis/services/Alaska
38.  -----_Part_0_12028181.1172175612864--

```

รูปที่ 5.10 ผลลัพธ์ของการทดสอบฝั่งผู้ให้บริการในกรณีศึกษาที่ 3

จากผลการทดสอบในรูปที่ 5.9 จะเห็นว่า เริ่มแรกตัวกลางทำการเลือกเว็บเซอร์วิส CUBook ซึ่งอยู่ภายใต้ชนิดบริการ BookSeller มาพิจารณาข้อมูลเฉพาะตัว แต่เมื่อพิจารณาถึง ส่วนนิพจน์ของผลกระทบนั้น ตามตารางที่ 5.10 กับ 5.11 จะเห็นได้ว่า นิพจน์ของเว็บเซอร์วิส

CUBook นั้นไม่ได้เป็นเซตย่อยของเว็บเซอร์วิส PacificBook ดังนั้นจึงไม่ผ่านการพิจารณาและเมื่อไม่มีเว็บเซอร์วิสอื่นภายใต้ชนิดบริการ BookSeller ตัวกลางจึงทำการเปลี่ยนไปใช้วิธีการเลือกเว็บเซอร์วิสที่เข้ากันได้แบบที่ 2 คือ เลือกใช้เว็บเซอร์วิสของชนิดบริการส่วนขยาย

ตัวกลางเลือกเว็บเซอร์วิส IndianBook ที่อยู่ภายใต้ชนิดบริการ BookAndMovieSeller ซึ่งขยายมาจากชนิดบริการ BookSeller มาพิจารณา แต่เมื่อพิจารณาถึงส่วนค่าคุณลักษณะเชิงความหมายจะเห็นได้ว่า สำหรับเว็บเซอร์วิส PacificBook นั้น คุณลักษณะเชิงความหมาย BookCategory มีค่าเป็น PacificNovel ซึ่งเป็นอินสแตนซ์ของคลาส Novel ส่วนของเว็บเซอร์วิส IndianBook นั้นมีค่าเป็น IndianPoetry ซึ่งเป็นอินสแตนซ์ของคลาส Poetry แต่จากออนโทโลยีสำหรับชนิดบริการในรูปที่ 3.11 จะเห็นได้ว่าค่าทั้งหมดไม่มีความสัมพันธ์กันตามการเข้าคู่เชิงความหมาย ดังนั้นจึงไม่ผ่านการพิจารณา ตัวกลางจึงทำการเปลี่ยนไปใช้วิธีการเลือกเว็บเซอร์วิสที่เข้ากันได้แบบที่ 3 คือ เลือกชนิดบริการที่เก็บอยู่ในส่วนขยายของยูติไลตีทั้งหมดมาพิจารณาโดยเรียงตามลำดับการบันทึกในส่วนขยายของยูติไลตี

ตัวกลางเลือกชนิดบริการ BookShop ขึ้นมาพิจารณา แต่เมื่อพิจารณาถึงข้อมูลเชิงพฤติกรรมแล้ว จากรูปที่ 5.1 จะเห็นได้ว่าชนิดบริการ BookSeller นั้นมีค่าเงื่อนไขก่อนการดำเนินการเป็น CreditCardHolder แต่ชนิดบริการ BookShop นั้นมีค่าเป็น DebitCardHolder ตามรูปที่ 5.7 ซึ่งจากออนโทโลยีสำหรับชนิดบริการในรูปที่ 3.11 นั้นจะเห็นได้ว่าค่าทั้งสองไม่มีความสัมพันธ์กันตามการเข้าคู่เชิงความหมาย ดังนั้นจึงไม่ผ่านการพิจารณา ตัวกลางจึงเลือกชนิดบริการ BookStore มาพิจารณา ซึ่งชนิดบริการ BookStore สามารถเข้ากันได้กับชนิดบริการ BookSeller เนื่องจากส่วนข้อมูลเชิงซิกเนเชอร์นั้นเข้าคู่กันตามการเข้าคู่เชิงซิกเนเชอร์และสำหรับส่วนข้อมูลเชิงพฤติกรรมนั้น แต่ละออนโทโลยีเทอมสามารถเข้าคู่กันได้ทุกกรณีตามการเข้าคู่เชิงความหมาย (ดูที่หัวข้อที่ 3.3) ดังนั้นตัวกลางจะเลือกเว็บเซอร์วิสที่อยู่ภายใต้ชนิดบริการ BookStore มาพิจารณาข้อมูลเฉพาะตัวกับเว็บเซอร์วิส PacificBook ซึ่งจากสภาพแวดล้อมที่กำหนดก็คือเว็บเซอร์วิส AlaskaBook

จากตารางที่ 5.10 และ 5.13 เมื่อพิจารณาแล้วจะเห็นได้ว่า ข้อมูลเฉพาะตัวแต่ละส่วนของทั้งสองเว็บเซอร์วิสสอดคล้องกันตามการเข้าคู่เชิงข้อมูลเฉพาะตัว เนื่องจาก

- ส่วนไบน์ดิงมีค่าเหมือนกัน
- ส่วนนิพจน์ของผลกระทบ จะเห็นว่านิพจน์ของเว็บเซอร์วิส AlaskaBook มีค่าเท่ากับนิพจน์ของเว็บเซอร์วิส PacificBook ดังนั้นจึงเป็นการเข้าคู่อย่างถูกต้อง

- ส่วนค่าคุณลักษณะเชิงความหมาย จะเห็นว่าอนโทโลยีเทอมชื่อ PacificNovel เป็นอินสแตนซ์ของคลาส Novel ซึ่งจะเป็นการเข้าคู่อย่างถูกต้องกับอนโทโลยีเทอมชื่อ AlaskaNovel ที่เป็นอินสแตนซ์ของคลาส Novel เช่นเดียวกัน

ดังนั้นเว็บเซอร์วิส AlaskaBook จึงสามารถทำงานแทนเว็บเซอร์วิส PacificBook ได้

สำหรับรูปที่ 5.10 นั้นบรรทัดที่ 1-18 แสดงข้อความโซฟที่เป็นคำร้องขอใช้บริการ โดย bookPurchase คือโอเปอเรชันที่ต้องการใช้บริการ (บรรทัดที่ 9) โดยมี isbn เป็นพารามิเตอร์ตัวที่ 1 ซึ่งมีค่าเป็น 2147483646 (บรรทัดที่ 10) creditcardDetail เป็นพารามิเตอร์ตัวที่ 2 ซึ่งมีค่าเป็น 04870442021 (บรรทัดที่ 10) และ <http://localhost:8080/axis/services/Pacific> คือเอนด์พอยท์ของเว็บเซอร์วิสที่ต้องการใช้บริการ (บรรทัดที่ 16) ส่วนบรรทัดที่ 19 - 38 นั้นแสดงข้อความโซฟที่เป็นผลลัพธ์จากการใช้บริการ โดย 3146 คือค่าผลลัพธ์ที่ได้จากการเรียกใช้โอเปอเรชัน bookPurchase (บรรทัดที่ 30) และ <http://localhost:8080/axis/services/Alaska> คือเอนด์พอยท์ของเว็บเซอร์วิส AlaskaBook ซึ่งเป็นเว็บเซอร์วิสที่ตัวกลางเรียกใช้บริการแทนเว็บเซอร์วิส PacificBook ที่ขัดข้อง (บรรทัดที่ 37)

#### 5.4 สรุปผลการทดสอบ

จากผลการทดสอบ จะเห็นว่าตัวกลางจะทำการค้นหาเว็บเซอร์วิสที่เข้ากันได้โดยใช้วิธีที่ 1 ก่อนคือ ค้นหาเว็บเซอร์วิสตัวแทนที่ประกาศข้อเสนอบริการอยู่ภายใต้ชนิดบริการเดียวกันกับเว็บเซอร์วิสที่ขัดข้อง และจะเปลี่ยนไปใช้วิธีที่ 2 และ 3 ตามลำดับเมื่อใช้วิธีก่อนหน้าแล้วไม่พบเว็บเซอร์วิสที่เข้ากันได้ และจะเห็นว่าการค้นหาด้วยวิธีที่ 1 และ 2 นั้นจะสามารถทำได้สะดวกและรวดเร็วกว่าวิธีที่ 3 เพราะพิจารณาเพียงแค่การเข้าคู่เชิงข้อมูลเฉพาะตัวระหว่างเว็บเซอร์วิสเท่านั้น เนื่องจากตามแนวคิดของแบบจำลองชนิดบริการนั้น เว็บเซอร์วิสที่ประกาศข้อเสนอบริการภายใต้ชนิดบริการเดียวกันจะเข้าคู่เชิงซิกเนเจอร์และเชิงความหมายของพฤติกรรมอยู่แล้ว

จากการทดสอบการทำงานของเฟรมเวิร์คด้วยกรณีศึกษา 3 กรณี เฟรมเวิร์คสามารถทำงานได้ตรงตามข้อกำหนดขอบเขตของงานวิจัยในบทที่ 3 โดยการนำแนวคิดของแบบจำลองชนิดบริการ ข้อเสนอบริการ และกฎการเข้ากันได้ของเว็บเซอร์วิสมาใช้ในการค้นหาเว็บเซอร์วิสที่เข้ากันได้ พร้อมทั้งปรับการเรียกใช้ไปยังเว็บเซอร์วิสตัวแทนโดยอัตโนมัติได้

## บทที่ 6

### สรุปผลการวิจัย

#### 6.1 สรุปผลการวิจัย

งานวิจัยนี้ได้แสดงถึงปัญหาและข้อจำกัดของการค้นหาเว็บเซอริวิสที่เข้ากันได้ในปัจจุบัน อันมีสาเหตุมาจากการประกาศคำอธิบายบริการตามโครงสร้างของยูดีดีไอ ที่ผู้ให้บริการแต่ละรายระบุประเภทของเว็บเซอริวิสผ่านทางแอททริบิวต์แคเทกอรีนั้น ไม่ได้มีข้อกำหนดว่าเว็บเซอริวิสที่มีแคเทกอรี เดียวกัน จะมีรูปแบบการให้บริการเหมือนกันหรือทำงานแทนกันได้ ดังนั้นงานวิจัยนี้จึงได้ประยุกต์แนวคิดของการออกแบบแบบจำลองชนิดบริการและข้อเสนอบริการเข้ากับเว็บเซอริวิส ซึ่งจะใช้ในการกำหนดรูปแบบการให้บริการและคุณสมบัติของเว็บเซอริวิสที่อยู่ในประเภทเดียวกัน โดยข้อกำหนดชนิดบริการนั้นจะอธิบายพฤติกรรมการให้บริการของเว็บเซอริวิสครอบคลุมลักษณะที่แบ่งออกเป็นสามส่วน ได้แก่ ส่วนของข้อมูลเชิงซิกเนเจอร์ซึ่งใช้เอกสารดับเบิลยูเอสดีแอลในการอธิบายข้อกำหนด ส่วนของข้อมูลเชิงพฤติกรรม และส่วนของแอททริบิวต์เชิงความหมาย ซึ่งใช้เอกสารอาร์เอสในการอธิบายข้อกำหนด นอกจากนี้ในขั้นตอนการสร้างชนิดบริการนั้นยังสนับสนุนการนำกลับมาใช้ซ้ำด้วย

สำหรับการพิจารณาการเข้ากันได้ นั้น งานวิจัยนี้ได้นำเสนอกฎการเข้ากันได้ของเว็บเซอริวิสตามแบบจำลองชนิดบริการ โดยแบ่งการพิจารณาออกเป็นสามส่วนคือ การเข้าคู่เชิงซิกเนเจอร์ การเข้าคู่เชิงความหมาย และการเข้าคู่เชิงข้อมูลเฉพาะตัว ตามลำดับ

ผู้วิจัยได้พัฒนาเฟรมเวิร์คเพื่อช่วยในการค้นหาเว็บเซอริวิสที่สามารถเข้ากันได้ พร้อมทั้งปรับการเรียกใช้ไปยังเว็บเซอริวิสตัวแทนนั้นโดยอัตโนมัติ ในกรณีที่เว็บเซอริวิสที่ต้องการใช้งานเกิดขัดข้อง ทำให้ผู้ใช้บริการยังคงรับบริการต่อไปได้โดยไม่ต้องเปลี่ยนคำร้องขอใช้บริการ รวมถึงการแสดงตัวอย่างการใช้งานเพื่อทดสอบการทำงานของเฟรมเวิร์คอีกด้วย

#### 6.2 ปัญหาและข้อจำกัดที่พบจากการวิจัย

ปัญหาและข้อจำกัดที่พบจากการวิจัยมีดังนี้

1. การค้นหาเว็บเซอริวิสตัวแทนของเฟรมเวิร์คนั้น จะเป็นกรณีที่เว็บเซอริวิสที่ต้องการใช้งานเกิดขัดข้องตั้งแต่ก่อนการเรียกใช้บริการเท่านั้น แต่จะไม่ครอบคลุมถึงกรณีที่เกิดการขัดข้องขึ้นในขณะที่ทำงาน นอกจากนี้ลักษณะการให้บริการของเว็บเซอริวิสที่เฟรมเวิร์คจะรองรับการค้นหาเซอริวิสที่เข้ากันได้ นั้นจะเป็นแบบสเตตเลส (Stateless) เพียง



- อย่างเดี่ยว ซึ่งในลักษณะนี้เฟรมเวิร์คจะสามารถเลือกเว็บเซอริวิสใดๆ ที่เข้าคู่ตามแบบจำลองชนิดบริการมาทำงานแทนได้โดยไม่ต้องคำนึงถึงสเททของการทำงาน
2. ในการพิจารณานิพจน์ของเงื่อนไขก่อนการดำเนินงานและเงื่อนไขผลกระทบตามกฎหมาย การเข้าคู่เชิงพิสัยตัวเลขนั้นได้กำหนดไว้ว่า ค่าพิสัยตัวเลขของเว็บเซอริวิสตัวแทนจะต้องเป็นซับเซตของเว็บเซอริวิสที่ขัดข้อง ตัวอย่างเช่นในตารางที่ 3.4 เว็บเซอริวิสตัวแทนสามารถนำส่งสินค้าได้ภายใน 4 วัน ในขณะที่เว็บเซอริวิสที่ขัดข้องสามารถนำส่งสินค้าได้ภายใน 7 วัน การกำหนดกฎการเข้าคู่เชิงพิสัยตัวเลขในลักษณะซับเซตนี้เป็นการรับรองว่า เงื่อนไขของเว็บเซอริวิสตัวแทนจะทำให้เงื่อนไขของเว็บเซอริวิสที่ขัดข้องเป็นจริงด้วย [21] กล่าวคือลูกค้าจะยังคงได้สินค้าจากเว็บเซอริวิสตัวแทนภายใน 7 วันอยู่ อย่างไรก็ตามในบางกรณีค่าพิสัยตัวเลขแบบซูเปอร์เซตจะให้ผลลัพธ์ที่ดีกว่าแบบเป็นซับเซต เช่น กรณีของการกำหนดเงื่อนไขของผลกระทบเป็นการให้ส่วนลด ถ้าเว็บเซอริวิสที่ขัดข้องกำหนดให้ส่วนลด 10% สำหรับการซื้อสินค้าครั้งต่อไป แต่เว็บเซอริวิสหนึ่งให้ส่วนลด 15% ซึ่งถ้าพิจารณาจะเห็นได้ว่าไม่เป็นไปตามกฎซับเซตดังกล่าวและถือว่าเว็บเซอริวิสนั้นไม่เข้าคู่และจะไม่ถูกเลือกมาเป็นตัวแทน แต่ในความเป็นจริงนั้นผู้ใช้บริการย่อมต้องการได้รับส่วนลดที่มากกว่าและเว็บเซอริวิสนั้นควรจะสามารถเป็นตัวแทนได้ ซึ่งในลักษณะเช่นนี้จะถือว่าเป็นข้อจำกัดของการพิจารณา
  3. เนื่องจากเครื่องมือที่ใช้ในการสร้างเว็บเซอริวิสของงานวิจัยนี้และเครื่องมือที่นิยมใช้อื่นๆ เช่น ไมโครซอฟท์วิซวลสตูดิโอโดทเน็ต (Microsoft Visual Studio .NET) นั้น มีข้อจำกัดในเรื่องของการผูกกันระหว่างเว็บเซอริวิสเอนด์พอยน์ท์กับดับเบิลยูเอสดีแอลเอนด์พอยน์ท์ ซึ่งส่งผลให้ในกรณีที่เว็บเซอริวิสที่ต้องการใช้งานขัดข้องขึ้นเนื่องจากสาเหตุของการยกเลิกการให้บริการ จะทำให้ตัวกลางไม่สามารถพิจารณาการเข้าคู่เชิงข้อมูลเฉพาะตัวในส่วนไบนด์ซึ่งเป็นข้อมูลที่อยู่ในเอกสารดับเบิลยูเอสดีแอลได้ด้วยเหตุนี้ในการประกาศข้อเสนอบริการจึงจำเป็นที่จะต้องอัปโหลดไฟล์เอกสารดับเบิลยูเอสดีแอลมาเก็บไว้ที่ตัวกลางด้วย เพื่อให้สามารถทำการค้นหาเว็บเซอริวิสที่เข้ากันได้ได้อย่างถูกต้องและมีประสิทธิภาพ
  4. การประกาศข้อเสนอบริการในส่วนของเอกสารอวล์-เอสเอ็น ผู้ให้บริการแต่ละรายอาจจะทำการสร้างไฟล์โดยใช้เครื่องมือที่แตกต่างกันออกไป ซึ่งจะทำให้รูปแบบภายในเอกสารนั้นมีความแตกต่างกันได้ ดังนั้นในการพิจารณาการเข้าคู่เชิงข้อมูลเฉพาะตัวในส่วนนิพจน์ของเงื่อนไขก่อนการดำเนินงานและผลกระทบในเอกสาร

อาร์เอส-เอส นั้น ตัวกลางจะยึดรูปแบบของเอกสารอาร์เอสที่สร้างจากโพเทเจซึ่งเป็นเครื่องมือสำหรับพัฒนาอินเทอร์เน็ตและฐานความรู้เป็นค่าโดยปริยายในการประมวลผล

### 6.3 ข้อเสนอแนะ

ประเด็นที่งานวิจัยนี้ยังไม่ได้ศึกษา และสามารถทำการวิจัยเพิ่มเติมได้ในอนาคต มีดังนี้

1. เนื่องจากผลลัพธ์ของการค้นหาเว็บเซอร์วิสที่เข้ากันได้กับเว็บเซอร์วิสที่ขัดข้องตามกฎหมาย การพิจารณานั้นอาจจะได้ผลลัพธ์ที่มากกว่าหนึ่งเว็บเซอร์วิส ดังนั้นจึงน่าจะมีการศึกษาถึงการจัดลำดับของเว็บเซอร์วิสที่เข้ากันได้เพื่อให้สามารถเลือกเว็บเซอร์วิสตัวแทนที่มีความเหมาะสมมากที่สุด
2. ในงานวิจัยนี้ ชนิดบริการถูกออกแบบให้อธิบายพฤติกรรมการทำงานของเว็บเซอร์วิสสามส่วนคือ ส่วนซิกเนเจอร์ ส่วนพฤติกรรม และส่วนแอททริบิวต์เชิงความหมาย ดังนั้นการวิจัยในอนาคตอาจจะพิจารณาคุณลักษณะอื่นๆ ของเว็บเซอร์วิสเพิ่มเติม เช่น คุณภาพของการบริการ (QoS: Quality of Service) ความไว้วางใจ (Trust) ระหว่างผู้ให้บริการกับเว็บเซอร์วิสตัวแทน เพื่อให้การเลือกเว็บเซอร์วิสตัวแทนนั้นเหมาะสมกับพฤติกรรมในการเลือกใช้บริการของผู้ให้บริการ ทำให้ผู้ให้บริการได้รับผลประโยชน์สูงสุด
3. เพิ่มการระบุรูปแบบการประเมินเงื่อนไขก่อนการดำเนินงานและเงื่อนไขผลกระทบลงในอินเทอร์เน็ตเชิงชนิดบริการ เพื่อช่วยในการพิจารณานิพจน์เชิงพีสัยตัวเลขของเงื่อนไขดังกล่าว เช่น ในการส่งสินค้าซึ่งเกี่ยวข้องกับจำนวนวันนำส่ง ค่าจำนวนวันนำส่งหากยิ่งน้อยจะยิ่งดี แต่ในการให้ส่วนลดซึ่งเกี่ยวข้องกับเปอร์เซ็นต์การให้ส่วนลด ค่าเปอร์เซ็นต์หากยิ่งมากจะยิ่งดี การระบุความหมายเพิ่มขึ้นนี้จะช่วยในการพิจารณาค่าพีสัยตัวเลขได้อย่างเหมาะสมยิ่งขึ้น
4. แนวคิดเกี่ยวกับการเข้ากันได้ของสองเว็บเซอร์วิสตามแบบจำลองชนิดบริการนี้สามารถประยุกต์ใช้ในการหาเว็บเซอร์วิสตัวแทนภายใต้การทำงานแบบกระแสน้ำตามกระบวนการธุรกิจได้

## รายการอ้างอิง

- [1] W3C. Web Services [Online]. 2002. Available from: <http://www.w3.org/2002/ws/>
- [2] W3C. Web Services Architecture [Online]. 2004. Available from:  
<http://www.w3.org/TR/ws-arch/>
- [3] Christensen, E., Curbera, F., Meredith, G. and Weerawarana, S. Web Services Description Language (WSDL) 1.1 [Online]. 2001. Available from:  
<http://www.w3.org/TR/2001/NOTE-wsdl-20010315>
- [4] Box, D., et al. Simple Object Access Protocol (SOAP) 1.1 [Online]. 2000.  
Available from: <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- [5] OASIS. Universal Description, Discovery, and Integration of Web Services (UDDI) [Online]. Available from: <http://www.uddi.org/>
- [6] Bearman, M. ODP-Trader. Proceedings of the International Conference on Open Distributed Processing 93 (ICODP'93), Berlin, September, 1993.
- [7] ISO/IEC. Information technology -- Open Distributed Processing -- Trading function: Specification ISO/IEC 13235-1:1998. 1998.
- [8] OMG. Trading Object Service Specification Version 1.0. May 2000.
- [9] Edwards, W. K. and Rodden, T. Jini Example By Example. Prentice Hall PTR, 2001.
- [10] W3C. XML Schema Part 2: Datatypes Second Edition [Online]. 2004.  
Available from: <http://www.w3.org/TR/xmlschema-2/>
- [11] Ontology. 2004. Available from: <http://www.semanticweb.org>
- [12] OWL-S Coalition. OWL-S 1.1 Release [Online]. 2004. Available from:  
<http://www.daml.org/services/owl-s/1.1/>
- [13] McGuinness, D. L. and van Harmelen, F. Web Ontology Language (OWL) [Online]. 2004. Available from: <http://www.w3.org/TR/owl-features/>
- [14] Antonellis, V. D., Melchiori, M. and Plebani, P. An approach to Web Service compatibility in cooperative process. Proceedings of Workshop on Service Oriented Computing (SOC), Orlando, 2003.

- [15] Gao, X., Yang, J. and Papazoglou, M. P. The Capability Matching of Web Services. IEEE Fourth International Symposium on Multimedia Software Engineering (MSE'02), pp. 56-63. December, 2002.
- [16] Sriharee, N. and Senivongse, T. Matchmaking and Ranking of Semantic Web Services Using Integrated Service Profile. International Journal of Metadata, Semantics and Ontologies, 1, 2 (2006): 100-118.
- [17] W3C. Web Services Description Language (WSDL) Version 2.0 Part 0: Primer [Online]. 2006. Available from: <http://www.w3.org/TR/wsdl20-primer/#more-interfaces-inheritance>
- [18] Sandia National Laboratories. Jess the Rule Engine for the JAVATM Platform [Online]. 2006. Available from: <http://www.jessrules.com/>
- [19] Jang, M. Bossam Rule/Owl Reasoner 2006. [Online]. Available from: <http://mknows.etri.re.kr/bossam/>
- [20] ISO/IEC. Information Technology – Open Distributed Processing – Reference Model: Architecture. 1996.
- [21] Liskov, B. and Wing, J. M. A Behavioral Notion of Subtyping. ACM Transactions on Programming Languages and Systems, pp. 1811-1841. November, 1994.



ภาคผนวก

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## ภาคผนวก ก

### ผลงานตีพิมพ์

ผลงานตีพิมพ์ซึ่งเป็นส่วนหนึ่งของงานวิจัยมีดังนี้

1. The 9th International Conference on Advanced Communication Technology (ICTACT 2007), pp. 76-81. 12-14 February 2007, Phoenix Park, Gangwon-Do, Republic of Korea ในบทความเรื่อง Compatibility by Service Type Model for Automatic Web Services Substitution โดยผู้แต่งคือ Rewat Pianwattanaphon และ Twittie Senivongse



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

# Compatibility by Service Type Model for Automatic Web Services Substitution

Rewat Pianwattanaphon and Twittie Senivongse  
 Department of Computer Engineering, Chulalongkorn University  
 Phyathai Road, Pathumwan, Bangkok 10330 Thailand  
 rewat.pi@student.chula.ac.th, twittie.s@chula.ac.th

**Abstract** — Determining compatibility between Web Services is necessary for several purposes including finding a substitute for a failed service. This paper presents a service type model for describing capabilities of Web Services in terms of their signature, behaviour, and semantic attributes, and rules for service type compatibility are defined. This model will facilitate the substitute selection process. The paper also proposes a framework for invoking a substitute Web Service automatically in the case of invocation failure.

**Keywords** — Compatibility, service type model, signature, semantics, substitution, Web Services.

## 1. Introduction

Web Services technology realises the Service-Oriented Architecture (SOA) concept by which service providers can publish their Web Services on the Web, and service consumers can discover and invoke them [1]. However, service invocation may fail and it is necessary that another Web Service can be discovered and invoked instead. For this purpose, we need criteria for determining service compatibility as well as a mechanism to automatically substitute for the failed service.

An intuitive substitution criterion is that Web Services published with a UDDI registry under the same “category” attribute should be able to substitute for each other. However, there is no guarantee that such services will have the same capability; they can have different WSDL signatures and behavioural semantics even though they are published under the same service category. In other words, Web Services can be published without any constraint on the common model of the service categories to which they belong. Without such a common model, finding a compatible substitute is difficult as it requires thorough consideration on the description of each Web Service.

This paper applies to Web Services the concept of service type model, which is the basis for service models in other standards such as RM-ODP [2] and CORBA [3]. Service type model consists of service types and service offers. A service type defines an operational signature and attributes that are common to service instances under the same type (c.f. the same category). A service offer refers to a published description of a service instance under a particular service type. A service type can be inherited or derived into another service type; this establishes a substitution relation among service types and service offers of the same or related types. In

RM-ODP and CORBA terms, descriptions about Web Services (i.e. WSDLs and information entries in UDDI) are defined only at the service offer level. In other words, Web Services instances are disparate and no formal relation can be drawn between any two Web Services descriptions. Recently, WSDL 2.0 [4] adds the concept of interface inheritance to WSDL specification. This can help define relations between two Web Services that relate by interface extension, but it still cannot draw relations between two Web Services with identical or similar interfaces that are not related by interface inheritance. With the service type model, an abstract layer of service types is introduced. This service type layer effectively signifies the relations between different Web Service offers of the same type and of the related types. In our approach, we additionally enhance the service type model by incorporating behaviour specification and semantic attributes into the model so that a service type will reflect both invocation signature and behavioural semantics that are common among a number of Web Services under the same category.

Section 2 of this paper gives more detail about the service type model. Section 3 describes compatibility rules for Web Services which are based on matching of signatures and semantics of the service types. Section 4 gives an example of compatible Web Services. Compatibility rules are used in the service substitution framework in Section 5. Section 6 discusses our approach and Section 7 presents related work. The paper concludes in Section 8 with future research directions.

## 2. Service Type Model

The service type model consists of service types and service offers. Service types are metadata that describe common characteristics of a number of service instances; each service instance has specific details described by a service offer.

### 2.1. Service Type

A service type is metadata that describes an abstract capability of Web Services of a particular category. It comprises the following information:

- Operation Specification describes operational signatures for service invocation. This corresponds to the abstract interface part of WSDL.

- Behaviour Specification augments an operational signature with its semantics, i.e. input, output, precondition, effect, and fault semantics. It is described by two levels of ontology specifications, namely the upper ontology (Section 2.2) and the service type ontology (Section 2.3). These ontology specifications are based on OWL language [5].
- Semantic Attribute Specification describes non-computational attributes for the service type. These attributes will be associated with semantic values which are based on OWL language.

An example of a book seller service type is shown in Figure 1. A book seller provides an operation for book purchase. The customer specifies the ISBN of the book on purchase and the credit card detail. The operation can give a fault or an exception if the customer's credit card is expired. The operation has associated semantics, e.g. the service can accept only a visa credit card and the book will be delivered to the customer after purchase. A book category attribute is defined to characterise the category of books for sale.

```

Service Type: BookSeller
Operation Specification:
  int bookPurchase(int isbnbn, String creditcardDetail);
  fault ExpCreditCard;
Behaviour Specification of bookPurchase:
  Input: ISBN, CreditcardDetail;
  Output: PurchasedBook;
  Precondition: VisaCardHolder;
  Effect: ProductDelivered;
  Fault: ExpiredCreditCard;
Semantic Attribute Specification: BookCategory;

```

Figure 1. BookSeller service type

A service type can be an extension of another service type if the former has the capability and semantics similar to the latter but with some differences. The rules for service type extension are as follows:

1. The inherited Operation Specification cannot be changed but more operations can be added. Behaviour Specifications associated with those newly added operations will be added too. This signature extension corresponds to the concept of interface inheritance in WSDL 2.0 [4].
2. For the inherited Behaviour Specification, only the precondition and effect can be changed, but the change must preserve the original semantics according to semantic matching (Section 3.2). Moreover new Behaviour Specifications can be added only when new operations are added to the Operation Specification.
3. For the inherited Semantic Attribute Specification, semantic attributes can be changed, but the change must preserve the original semantics according to semantic matching (Section 3.2). Also, more semantic attributes can be added.

A service type can extend from one or more service types. If there is a clash between the operation signatures of the inherited service types, they are collapsed into a single

operation to resolve name ambiguity. This corresponds to WSDL 2.0 also.

In Figure 2, a service type called BookAndMovieSeller extends from the BookSeller service type in Figure 1 by using the keyword ExtensionOf. A new operation moviePurchase is added (line 5-6) with the associated Behaviour Specification (line 13-18). For the inherited Behaviour Specification of the operation bookPurchase (line 7-12), the precondition and the effect have been changed in this extension. These changes preserve the original semantics of the precondition VisaCardHolder and the effect ProductDelivered of the BookSeller (see Section 3.2 for details). A new semantic attribute MovieCategory is added also.

```

1. Service Type: BookAndMovieSeller ExtensionOf BookSeller
2. Operation Specification:
3.   int bookPurchase(int isbnbn, String creditcardDetail);
4.   fault ExpCreditCard;
5.   int moviePurchase(String mName, String creditcardDetail);
6.   fault ExpCreditCard, MovieNotFound;
7. Behaviour Specification of bookPurchase:
8.   Input: ISBN, CreditcardDetail;
9.   Output: PurchasedBook;
10.  Precondition: CreditCardHolder;
11.  Effect: ProductDeliveredByAir;
12.  Fault: ExpiredCreditCard;
13. Behaviour Specification of moviePurchase:
14.  Input: MovieName, CreditcardDetail;
15.  Output: PurchasedMovie;
16.  Precondition: CreditCardHolder;
17.  Effect: ProductDeliveredByAir;
18.  Fault: ExpiredCreditCard, MovieNotFound;
19. Semantic Attribute Specification: BookCategory, MovieCategory;

```

Figure 2. BookAndMovieSeller service type

## 2.2. Upper Ontology

The upper ontology in Figure 3 is the metamodel for describing behaviour and semantic attributes of service types. The service profile and process model of OWL-S [6] which are the OWL-based specifications for describing behaviour of Web Services are adopted here. A service type is described by the operations of which it is capable. The operation behaviour is represented by the inputs, outputs, preconditions, and effects of the operation. We extend OWL-S with fault and semantic attribute classes to accommodate our service type model.

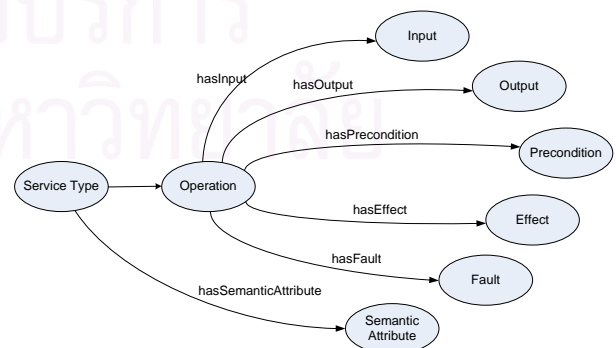


Figure 3. Upper ontology



### 2.3. Service Type Ontology

A service type ontology is derived from the upper ontology and defines the behaviour and semantics of a particular service type according to the Behaviour Specification and Semantic Attribute Specification. The BookSeller service type ontology is in Figure 4.

```

<rdf:RDF ...
  xmlns:process="http://www.daml.org/services/owl-s/1.1/Process.owl#"
  xmlns:profile="http://www.daml.org/services/owl-s/1.1/Profile.owl#"
  xmlns:exprofile="http://localhost:8080/onto/upperontology/
    upperonto.owl#"
  xmlns="http://localhost:8080/onto/servicetype/BookSeller/
    bookPurchase.owl#"
  ...
  <owl:Class rdf:ID="BookNumber">
    <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/
      1.1/Process.owl#Input"/>
  </owl:Class>
  <owl:Class rdf:ID="ISBN">
    <rdfs:subClassOf rdf:resource="#BookNumber"/>
  </owl:Class>
  <owl:Class rdf:ID="CreditcardDetail">
    <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/
      1.1/Process.owl#Input"/>
  </owl:Class>
  <owl:Class rdf:ID="PurchasedBook">
    <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/
      1.1/Process.owl#Output"/>
  </owl:Class>
  <owl:Class rdf:ID="Payment">
    <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/
      1.1/generic/Expression.owl#Condition"/>
  </owl:Class>
  <owl:Class rdf:ID="CreditCardHolder">
    <rdfs:subClassOf rdf:resource="#Payment"/>
  </owl:Class>
  <owl:Class rdf:ID="VisaCardHolder">
    <rdfs:subClassOf rdf:resource="#CreditCardHolder"/>
  </owl:Class>
  <owl:Class rdf:ID="ProductDelivered">
    <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/
      1.1/generic/Expression.owl#Expression"/>
  </owl:Class>
  <owl:Class rdf:ID="ExpiredCreditCard">
    <rdfs:subClassOf
      rdf:resource="http://localhost:8080/ontology/upperontology/
        upperonto.owl#Fault"/>
  </owl:Class>
  <owl:Class rdf:ID="BookCategory">
    <rdfs:subClassOf rdf:resource="http://localhost:8080/ontology/
      upperontology/upperonto.owl#SemanticAttribute"/>
  </owl:Class>
  <owl:Class rdf:ID="Novel">
    <rdfs:subClassOf rdf:resource="#BookCategory"/>
  </owl:Class>
  ...

```

Figure 4. Part of service type ontology for BookSeller

### 2.4. Service Offer

A service offer refers to the published descriptions of a Web Service instance under a particular service type. The descriptions comprise a WSDL and an OWL-S specification which conform to the service type of the Web Service. A WSDL such as the one of the Web Service named Pacific

(Figure 5) will be published with its interface part (i.e. portType tag for WSDL 1.1 or interface tag for WSDL 2.0) corresponding to the Operation Specification of the BookSeller service type.

```

...
<wsdl:portType name="BookSeller">
  <wsdl:operation name="bookPurchase"
    parameterOrder="in0 in1">
    <wsdl:input message="impl:bookPurchaseRequest"
      name="bookPurchaseRequest" />
    <wsdl:output message="impl:bookPurchaseResponse"
      name="bookPurchaseResponse" />
    <wsdl:fault message="impl:ExpCreditCard"
      name="ExpCreditCard" />
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="PacificSoapBinding" type="impl:BookSeller">
  <wsdlsoap:binding style="rpc"
    transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="bookPurchase">
    ...
  </wsdl:binding>

```

Figure 5. Part of WSDL of Pacific under BookSeller service type

The Web Service provider also publishes an OWL-S specification which corresponds to the Behaviour and Semantic Attribute Specifications. It refers to the service type ontology of the service category. Figure 6 shows part of the OWL-S specification of the Web Service named Pacific. The behaviour of its operation bookPurchase (line 16-46) refers to the ontological concepts in the BookSeller service type ontology (line 9-14, 44). Note that the OWL-S specification can specify details that are specific to a particular Web Service instance. In this example, the effect ProductDelivered of Pacific is further constrained such that the delivery will be within 7 days (line 23-39). The constraint is defined using SWRL language [7]. For the semantic attribute BookCategory, Pacific declares that it sells novels (line 15, 45).

```

1. <rdf:RDF ...
2.  xmlns:process="http://www.daml.org/services/owl-s/1.1/Process.owl#"
3.  xmlns:profile="http://www.daml.org/services/owl-s/1.1/Profile.owl#"
4.  xmlns:exprofile="http://localhost:8080/onto/upperontology/
5.    upperonto.owl#"
6.  xmlns:bsbp="http://localhost:8080/onto/servicetype/BookSeller/
7.    bookPurchase.owl#"
8.  ...
9.  <bsbp:ISBN rdf:ID="PacificISBN"/>
10. <bsbp:CreditcardDetail rdf:ID="PacificCreditcardDetail"/>
11. <bsbp:PurchasedBook rdf:ID="PacificPurchasedBook"/>
12. <bsbp:VisaCardHolder rdf:ID="PacificVisaCardHolder"/>
13. <bsbp:ProductDelivered rdf:ID="PacificProductDelivered"/>
14. <bsbp:ExpiredCreditCard rdf:ID="PacificExpiredCreditCard"/>
15. <bsbp:Novel rdf:ID="PacificNovel"/>
16. <profile:Profile rdf:ID="bookPurchase">
17.   <profile:hasInput rdf:resource="#PacificISBN"/>
18.   <profile:hasInput rdf:resource="#PacificCreditcardDetail"/>
19.   <profile:hasOutput rdf:resource="#PacificPurchasedBook"/>
20.   <profile:hasPrecondition rdf:resource="#PacificVisaCardHolder"/>
21.   <profile:hasResult>
22.     <process:Result rdf:ID="PacificPurchasedBookResult">
23.       <process:hasEffect rdf:resource="#PacificProductDelivered"/>
24.       <expr:expressionLanguage rdf:resource="http://www.daml.org/
25.         services/owl-s/1.1/generic/Expression.owl#SWRL"/>

```

Figure 6. Part of OWL-S specification of Pacific under BookSeller service type

```

26. <expr:expressionBody rdf:datatype="http://www.w3.org/1999/
27.   02/22-rdf-syntax-ns#XMLLiteral">
28.   <swrl:AtomList>
29.     <rdf:first>
30.       <swrl:BuiltinAtom>
31.         <swrl:builtin rdf:resource="http://www.w3.org/2003/11/
32.           swrlb#lessThanOrEqual"/>
33.     <swrl:arguments>
34.       <rdf:List>
35.         <rdf:first rdf:resource="#Days"/>
36.         <rdf:rest>
37.           <rdf:List>
38.             <rdf:first rdf:datatype="http://www.w3.org/2001/
39.               XMLSchema#int">7</rdf:first>
40.             <rdf:rest rdf:resource="http://www.w3.org/1999/02/
41.               22-rdf-syntax-ns#nil"/>
42.           </rdf:List>
43.         ...
44.     <exprofile:hasFault rdf:resource="#PacificExpiredCreditCard"/>
45.     <exprofile:hasSemanticAttribute rdf:resource="#PacificNovel"/>
46.   </profile:Profile>
47. ...

```

**Figure 6. Part of OWL-S specification of Pacific under BookSeller service type (continued)**

### 3. Compatibility Rules for Web Services

For a Web Service to be compatible and can substitute for another Web Service, we have to consider their signature matching and semantic matching. With the service type model, a particular service offer will be compatible with other offers of the same service type and other offers of the supertype because the Operation Specifications and Behaviour Specifications are compatible. However, we still have to determine particular details of the instances, i.e. the semantic values in the Semantic Attribute Specification, binding details, and constraints on service behaviour. For Web Service offers that are not related by their service types, full consideration on signature and semantics is needed.

#### 3.1. Signature Matching

Signature matching is adopted from signature subtyping rules for operation interfaces in [2]. An Operation Specification  $X$  is a signature subtyping and compatible with an Operation Specification  $Y$  if the conditions below are satisfied:

1. For each operation in  $Y$ , there is an operation in  $X$  with the same name;
2. For each operation in  $Y$ , the number of input parameters is the same as that of the corresponding operation in  $X$ ; parameter names can be different;
3. For each input parameter of the operation in  $Y$ , the type is the same or is derived from the type of the corresponding input parameter in  $X$ ;
4. For each operation in  $Y$ , the number of output parameters is the same as that of the corresponding operation in  $X$ ; parameter names can be different;
5. For each output parameter of the operation in  $X$ , the type is the same or is derived from the type of the corresponding output parameter in  $Y$ ;
6. For each fault in an operation in  $X$ , there is a fault with the same name in the corresponding operation in  $Y$ ;

7. For each fault in an operation in  $X$ , the number of fault parameters is the same as that of the corresponding fault in  $Y$ ; parameter names can be different;
8. For each fault parameter in an operation in  $X$ , the type is the same or is derived from the type of the corresponding fault parameter in  $Y$ .

Derivation of parameter types is according to XML schema data type specification [8].

#### 3.2. Semantic Matching

Since the Behaviour Specification and Semantic Attribute Specification are based on the service type ontology, compatibility of the behaviour and semantic attributes of Web Services relies on matching of ontological concepts specified in their OWL-S specifications [9]. Matching is determined from the concept hierarchy within the service type ontology.

Let  $C_X$  and  $C_Y$  be the ontological concepts in the Behaviour Specification or Semantic Attribute Specification of the service types  $X$  and  $Y$  respectively.  $C_X$  is an ontological match to  $C_Y$  if either of the following is satisfied:

1. If  $C_X \equiv C_Y$  then  $C_X$  is an exact match for  $C_Y$ , where  $\equiv$  means is equivalent to.
2. If  $C_X \sqsubseteq C_Y$  then  $C_X$  is a specialised match for  $C_Y$ , where  $\sqsubseteq$  means is subsumed by. For example the concept ISBN is a specialised match to the concept BookNumber in the BookSeller service type ontology.

Semantic matching follows the behavioural subtyping rules in [10]. The Behaviour Specification and Semantic Attribute Specification of a service type  $X$  is a semantic subtyping and compatible with the Behaviour Specification and Semantic Attribute Specification of a service type  $Y$  if the conditions below are satisfied:

1. For every pair of corresponding inputs in  $X$  and  $Y$ , the input in  $Y$  is an ontological match to an input in  $X$ ;
2. For every pair of corresponding outputs in  $X$  and  $Y$ , the output in  $X$  is an ontological match to the output in  $Y$ ;
3. For every precondition in  $X$ , there is a precondition in  $Y$  that is an ontological match to that precondition in  $X$ ;
4. For every effect in  $Y$ , there is an effect in  $X$  that is an ontological match to that effect in  $Y$ ;
5. For every fault in  $X$ , there is a fault in  $Y$  such that the fault in  $X$  is an ontological match to the fault in  $Y$ ;
6. For every semantic attribute in  $Y$ , there is a semantic attribute in  $X$  that is an ontological match to that semantic attribute in  $Y$ .

#### 3.3. Instance Detail Matching

Signature matching and semantic matching mentioned above consider the abstract capability and behaviour at the service type level. To select a Web Service  $X$  that is compatible enough to substitute for another Web Service  $Y$ ,

we have to also determine their specific instance details as follows.

1. Their binding details are the same, i.e the style and transport attributes of the soap:binding element in their WSDLs have the same value.
2. Their behaviour constraints are compatible, i.e. the constraints associated with the preconditions or the effects in their OWL-S specifications must match. In this paper, we only focus on numerical range constraints [9] as follows:  
Let  $N_X$  be a nonempty set of values within the numerical range that constrains the precondition or effect of a Web Service X ( $W_X$ ) and  $N_Y$  be a nonempty set of values within the numerical range that constrains the corresponding precondition or effect of a Web Service Y ( $W_Y$ ). If  $N_X \subseteq N_Y$  then  $W_X$  is a match for  $W_Y$ .
3. For every pair of corresponding semantic attributes in X and Y, the value of the semantic attribute in X is an ontological match to the value of the semantic attribute in Y.

#### 4. Example of Compatible Web Services

Suppose a Web Service Y (e.g. Pacific Web Service under the BookSeller service type) fails. A substitute Web Service instance is determined by the following criteria.

1. Select one of the Web Service instances under the same service type, where this service instance matches to Y by instance detail matching. Signature matching and semantic matching are not necessary because service instances under the same service type will have the same signature and behaviour.
2. Select one of the Web Service instances under the service types that extend from the service type of Y, where this service instance matches to Y by instance detail matching. Instances of the subtype have the signature and behaviour that are compatible with those of Y. This criterion is chosen when a substitute is not found by the criterion 1.
3. Find any service type that matches to the service type of Y by signature matching and semantic matching. Then select one of its instances which matches to Y by instance detail matching. This criterion is chosen when a substitute is not found by the criterion 2.

Figure 7 shows another service type BookStore. By signature matching and semantic matching, BookStore is compatible with BookSeller. Details of matching result are in Table 1.

Assume a Web Service X (e.g. Alaska Web Service) under the BookStore service type has the instance details as in Table 2. By instance detail matching, their binding details and semantic attribute values match exactly. The constraint on the effect ProductDeliveredByAir of X is a subset and therefore a match to the constraint on the effect ProductDelivered of Y.

Thus Web Service X is a compatible and a substitute service for Y.

Service Type: BookStore
Operation Specification: int bookPurchase(int isbn, String creditcardDetail); fault ExpCreditCard;
Behaviour Specification of bookPurchase: Input: BookNumber, CreditcardDetail; Output: PurchasedBook; Precondition: CreditCardHolder; Effect: ProductDeliveredByAir; Fault: ExpiredCreditCard;
Semantic Attribute Specification: BookCategory;

Figure 7. BookStore service type

Table 1. Matching result between BookSeller and BookStore service types

Part of description	BookSeller service type	BookStore service type	Match type
Input	ISBN CreditcardDetail	BookNumber CreditcardDetail	Specialised Exact
Output	PurchasedBook	PurchasedBook	Exact
Precondition	VisaCardHolder	CreditCardHolder	Specialised
Effect	ProductDelivered	ProductDeliveredByAir	Specialised
Fault	ExpiredCreditCard	ExpiredCredit Card	Exact
Semantic Attribute	BookCategory	BookCategory	Exact

Table 2. Instance details of Web Services Y and X

Part of description	Web Service Y (failed)	Web Service X
Binding	style="rpc" transport="http"	style="rpc" transport="http"
Precondition expression	-	-
Effect expression	ProductDelivered.Days ≤ 7	ProductDeliveredByAir.Days ≤ 4
Semantic attribute values	PacificNovel (Instance of Novel Class)	AlaskaNovel (Instance of Novel Class)

#### 5. Framework for Web Services Substitution

Figure 9 shows a mediator for service substitution. A domain expert or a service provider will first define a service type (1) which is stored in the service type repository within an extended UDDI. Service providers can create and publish Web Services (such as Web Services X and Y) which comply with the registered service type (3). The mediator will store Web Service name, service provider's name, the referenced service type, and the endpoints of WSDL and OWL-S specification in the service offer repository of the extended UDDI (4). With service type and service offer information, a list of compatible service types and offers is generated based on the relations between service types and compatibility rules. When a service consumer invokes Web Service Y through the mediator (5), the call will be passed to Y and the result will be returned to the consumer (6). But if Y fails, the mediator will intercept the call and select a compatible service, say X, from the compatibility list (7). The mediator will pass the call to X automatically and return the result to the consumer (9).

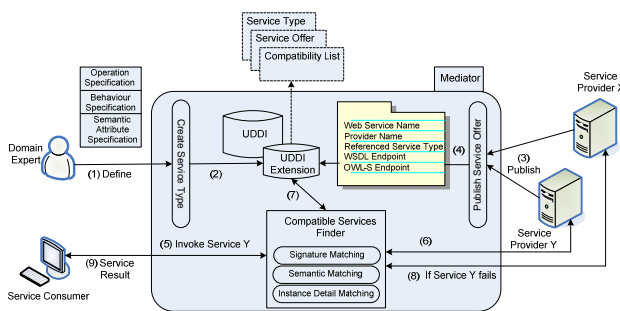


Figure 9. Substitution framework

## 6. Discussion

Our approach aims for dynamic selection of substitute Web Services and considers service compatibility when the need arises. The abstract layer of service types facilitates the selection because it helps reduce the number of Web Service instances to be considered. As opposed to the dynamic selection approach, a Web Service provider may define in advance a group of Web Service instances that can substitute for a particular Web Service. This static approach is convenient to a certain extent but the service provider may have to determine compatible service instances by himself and there is no guarantee that the predefined substitutes will always work properly.

For automatic service substitution to be effective, service descriptions must be rich and truly reflect the capability and semantics of Web Services. Our service type model accommodates both signature and semantic aspects of Web Services, therefore the compatibility rules based on this model should be able to determine good substitutes. Nevertheless, our service type model is by no means a complete service description model. The Behaviour Specification and associated constraints are still considered simple, and there are also other aspects that are not touched in this paper, e.g. quality of services and trust between service consumers and substitute services. A trade-off between richness of the service description model and its maintenance overheads is an important issue.

## 7. Related Work

Other research works have also addressed the service compatibility issue. In [11], compatibility is determined between an abstract service within a business process and target Web Service instances, but only the semantics of operations, inputs, and outputs are analysed. The work in [12] proposes a service capability description language (SCDL) to describe a Web Service in terms of service name, ontological description, data types, inputs, outputs, preconditions, and postconditions. This work is close to our approach in that both signature and semantics aspects of Web Services are considered. Unlike our approach, the work does not consider semantic attributes and the proposed description language does not comply with the standard or de-facto standard descriptions for Web Services. In [13], WSDL-S defines a

lightweight approach to describing semantics for a Web Service by annotating its WSDL with ontological concepts that represent semantics for the operation, input, output, precondition, effect, and service category. In our approach, such semantics are in a separate OWL-S specification because we believe that a semantic description should be rich in order for the automatic substitution to be effective, and it may not be appropriate for rich semantics to get cluttered in the WSDL. Service consumers may unnecessarily be exposed to semantic concepts that may not be relevant to them when studying how to invoke the Web Services.

## 8. Conclusion

We present a service type model with compatibility rules for determining compatible Web Services. The service type model considers both signature and semantics as well as Web Service instance details. The model facilitates the substitution framework in finding a substitute for a failed service. In the future work, the selection of a substitute service can be refined by ranking compatible services or considering other aspects such as QoS and trust in order to identify an optimal substitute. The performance of the framework will also be evaluated.

## ACKNOWLEDGMENT

This research is part of the Engineering New Paradigm Software for Enterprises with Service-Oriented Architecture Project, supported by Thailand's Software Industry Promotion Agency (Public Organisation).

## REFERENCES

- [1] D.K. Barry, *Web Services and Service-Oriented Architectures The Savvy Manager's Guide*, Morgan Kaufmann Publishers, San Francisco, 2003.
- [2] ISO/IEC, *Open Distributed Processing Reference Model Part 3: Architecture ISO/IEC 10746-3*, 1995.
- [3] Object Management Group, *Trading Object Service Specification, Version 1.0*, May 2000.
- [4] W3C, *Web Services Description Language (WSDL) Version 2.0 Part 0: Primer*, 27 March 2006. [Online] Available from <http://www.w3.org/TR/wsd120-primer/>
- [5] w3.org, *OWL Web Ontology Language Overview*. [Online] Available from <http://www.w3.org/TR/2004/REC-owl-features-20040210/>
- [6] OWL-S Coalition, *OWL-S 1.1 Release*. [Online] Available from <http://www.daml.org/services/owl-s/1.1/>
- [7] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosz, and M. Dean, *SWRL: A Semantic Web Rule Language combining OWL and RuleML*, 2003. [Online] Available from <http://daml.org/2003/11/swrl/>
- [8] w3.org, *XML Schema Part 2: Data Types Second Edition*, 28 October 2004. [Online] Available from <http://www.w3.org/TR/xmlschema-2/>
- [9] N. Sriharee and T. Senivongse, "Matchmaking and Ranking of Semantic Web Services Using Integrated Service Profile", *International Journal of Metadata, Semantics and Ontologies*, Vol. 1, No. 2, Inderscience Publishers, 2006, pp. 100-118.
- [10] B. Liskov and J.M. Wing, "A Behavioral Notion of Subtyping", *ACM Transactions on Programming Languages and Systems*, 16(6), November 1994, pp. 1811-1841.
- [11] V. De Antonellis, M. Melchiori, and P. Plebani, "An Approach to Web Service Compatibility in Cooperative Process", *Proceedings of Workshop on Service Oriented Computing (SOC 2003)*, Orlando, 2003, pp. 1-6.
- [12] X. Gao, J. Yang, and M. Papazoglou, "The Capability Matching of Web Services", *Proceedings of IEEE 4th International Symposium on Multimedia Software Engineering (MSE'02)*, 2002, pp. 56-63.
- [13] w3.org, *Web Service Semantics – WSDL-S*, 7 November 2005. [Online] Available from <http://www.w3.org/Submission/WSDL-S/>

## ประวัติผู้เขียนวิทยานิพนธ์

นายเรวัตร เพียรวัฒนผล เกิดเมื่อวันที่ 20 กุมภาพันธ์ พ.ศ. 2526 ที่จังหวัดจันทบุรี สำเร็จการศึกษาระดับปริญญาบัณฑิต หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ จาก มหาวิทยาลัยเทคโนโลยีสุรนารี เมื่อ พ.ศ. 2548 และเข้าศึกษาต่อในหลักสูตร วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ ณ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปี พ.ศ. 2548



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย