

กรอบแนวคิดการเจรจาต่อรองแบบอิเล็กทรอนิกส์
สำหรับสนับสนุนสถาปัตยกรรมเชิงบริการ
An Electronic Negotiation Framework for Supporting
Service-Oriented Architecture

ชัยวัฒน์ ไพบูลย์พุฒิพงศ์
Chaiwat Phaiboonphuthiphong

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

มหาวิทยาลัยสงขลานครินทร์

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

2553

๗ ลิขสิทธิ์ของมหาวิทยาลัยสงขลานครินทร์

เลขหมู่	QA76.9.A73	๗64	๒๕๕๓ ค.๒
Dib Key	๑๒๔๕๒๙		
	๓๐.๒๕๕๓		

(1)

ชื่อวิทยานิพนธ์	กรอบแนวคิดการเจรจาต่อรองแบบอิเล็กทรอนิกส์สำหรับสนับสนุน
	สถาปัตยกรรมเชิงบริการ
ผู้เขียน	นายชัยวัฒน์ ไพบุลย์พุดพิงศ์
สาขาวิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2552

บทคัดย่อ

ตั้งแต่แนวคิดสถาปัตยกรรมเชิงบริการได้รับการแนะนำและเริ่มได้รับความสนใจจากนักพัฒนาซอฟต์แวร์ ส่งผลให้ผลิตภัณฑ์ซอฟต์แวร์ถูกสร้างอยู่ในรูปบริการ การเจรจาต่อรองเป็นประเด็นสำคัญสำหรับการพัฒนาซอฟต์แวร์เชิงบริการให้สอดคล้องกับความต้องการของทั้งผู้ให้บริการและผู้ใช้บริการ อย่างไรก็ตามการเจรจาต่อรองค่อนข้างมีความซับซ้อนและยุ่งยาก และกรอบแนวคิดการเจรจาต่อรองที่มีอยู่ยังขาดรายละเอียดที่เหมาะสมต่อการนำมาประยุกต์ใช้ได้จริง

วิทยานิพนธ์นี้จึงมุ่งเน้นในการศึกษาและเสนอกรอบแนวคิดโครงสร้างการทำงานของการเจรจาต่อรองสำหรับสนับสนุนแนวคิดสถาปัตยกรรมเชิงบริการ และดำเนินการพัฒนาระบบที่สอดคล้องกับกรอบแนวคิดที่เสนอ ในลักษณะตัวกลางเพื่อจัดการข้อมูลที่เกี่ยวข้อง ทั้งนี้ระบบที่ได้รับการพัฒนาและทำการทดสอบในห้องปฏิบัติการเท่านั้น โดยให้ผลเป็นที่น่าพอใจซึ่งคาดว่าสามารถนำไปปรับปรุงใช้งานจริงได้

Thesis Title	An Electronic Negotiation Framework for Supporting Service-Oriented Architecture
Author	Mr.Chaiwat Phaiboonphuthiphong
Major Program	Computer Science
Academic Year	2009

ABSTRACT

Since service-oriented architecture (SOA) was introduced and gained attentions from software developers, this new notion has affected software products to be built in the form of services. Negotiation becomes a key issue for service-oriented software development in order to satisfy the needs between consumers and providers. However, negotiation seems to be complicated and existing negotiation frameworks lack in details of implementation.

This research has proposed a negotiation framework for supporting service-oriented architecture and developed a software system corresponding to the proposed framework as a broker for data management. The system has been only tested in the laboratory environment. The results satisfy the objectives of this project and may lead to real adoption in future.

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สามารถสำเร็จลุล่วงได้ด้วยดีด้วยความช่วยเหลือจากบุคคลหลายท่านซึ่งผู้วิจัยรู้สึกซาบซึ้งและขอขอบคุณทุกท่านเป็นอย่างสูงมา ณ โอกาสนี้คือ

ผศ.ดร.อำนาจ เปาะทอง ที่ให้คำปรึกษา ข้อเสนอแนะสำหรับการวิจัย ดูแลการทำวิทยานิพนธ์มาโดยตลอด พร้อมทั้งตรวจทานและแก้ไขวิทยานิพนธ์ให้แก่ผู้วิจัย

ดร.สุภาภรณ์ กานต์สมเกียรติ อาจารย์ที่ปรึกษาร่วม ที่ช่วยให้คำแนะนำและข้อเสนอแนะในการทำวิทยานิพนธ์

อาจารย์ประจำภาควิชาวิทยาการคอมพิวเตอร์ทุกท่านที่ได้ประสิทธิ์ประสาทวิชาให้แก่ผู้วิจัยซึ่งเป็นพื้นฐานสำคัญในการทำวิทยานิพนธ์

คณะกรรมการสอบวิทยานิพนธ์ทุกท่าน ที่กรุณาให้คำแนะนำและตรวจทานรายงานวิทยานิพนธ์ให้มีความสมบูรณ์

มูลนิธิเพื่อการศึกษาคอมพิวเตอร์และการสื่อสารที่สนับสนุนให้ทุนการศึกษาประจำปีการศึกษา 2549 ซึ่งช่วยสนับสนุนให้สามารถดำเนินการวิจัยได้โดยสะดวก

เจ้าหน้าที่ภาควิชาวิทยาการคอมพิวเตอร์ และเจ้าหน้าที่บัณฑิตวิทยาลัยทุกท่านที่อำนวยความสะดวกต่อการจัดทำและดำเนินการเกี่ยวกับเอกสารต่างๆ ที่เกี่ยวข้องกับการทำวิทยานิพนธ์จนเป็นที่เรียบร้อย

เพื่อน ๆ พี่ ๆ และน้อง ๆ ภาควิชาวิทยาการคอมพิวเตอร์ทุกคน ที่ให้คำปรึกษาและเป็นกำลังใจมาโดยตลอด

คุณพ่อ คุณแม่ พี่ น้อง และครอบครัวที่คอยเคียงข้างและเป็นกำลังใจสนับสนุนทุกสิ่งทุกอย่างมาโดยตลอด จนกระทั่งผู้วิจัยสำเร็จการศึกษา

ผู้วิจัยรู้สึกซาบซึ้งและขอขอบพระคุณทุกท่านไว้ ณ ที่นี้ด้วย

ชัยวัฒน์ ไพบูลย์พุดพิงศ์

สารบัญ

	หน้า
สารบัญ.....	(6)
รายการตาราง.....	(10)
รายการภาพประกอบ.....	(12)
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญและที่มาของงานวิจัย.....	1
1.2 การตรวจเอกสารและงานวิจัยที่เกี่ยวข้อง.....	2
1.2.1 การตรวจเอกสาร.....	2
1.2.2 งานวิจัยที่เกี่ยวข้อง.....	3
1.3 วัตถุประสงค์ของงานวิจัย.....	4
1.4 ขอบเขตการดำเนินงานของการวิจัย.....	4
1.5 ขั้นตอนการดำเนินงาน.....	5
1.6 ระยะเวลาดำเนินงาน.....	5
1.7 สถานที่และเครื่องมือที่ใช้ในการดำเนินการวิจัย.....	6
1.7.1 สถานที่.....	6
1.7.2 เครื่องมือที่ใช้.....	6
1.8 ประโยชน์ที่ได้รับ.....	7
บทที่ 2 แนวคิดและทฤษฎีที่เกี่ยวข้อง.....	8
2.1 สถาปัตยกรรมเชิงบริการ.....	8
2.1.1 บริการและแนวคิดเชิงบริการ.....	8
2.1.1.1 บริการ.....	8
2.1.1.2 แนวคิดเชิงบริการ.....	11
2.1.2 สถาปัตยกรรมเชิงบริการและองค์ประกอบ.....	16
2.1.2.1 ความหมายของสถาปัตยกรรมเชิงบริการ.....	16
2.1.2.2 องค์ประกอบของสถาปัตยกรรมเชิงบริการ.....	16
2.1.2.3 คุณสมบัติของสถาปัตยกรรมเชิงบริการ.....	18
2.1.3 เทคโนโลยีสนับสนุนสถาปัตยกรรมเชิงบริการ.....	19

สารบัญ (ต่อ)

	หน้า
2.2 การเจรจาต่อรอง (Negotiation).....	27
2.2.1 แนวคิดการเจรจาต่อรองโดยทั่วไป	27
2.2.1.1 ความหมายของการเจรจาต่อรอง	27
2.2.1.2 การเจรจาต่อรองโดยทั่วไป	28
2.2.2 การเจรจาต่อรองแบบอิเล็กทรอนิกส์ (Electronic negotiation).....	29
2.2.2.1 ลักษณะและรูปแบบของ การเจรจาต่อรองแบบอิเล็กทรอนิกส์	29
2.2.2.2 ประเภทและองค์ประกอบของ การเจรจาต่อรองแบบอิเล็กทรอนิกส์	31
2.2.2.3 เทคโนโลยีที่ใช้พัฒนาการเจรจาต่อรองแบบอิเล็กทรอนิกส์.....	33
2.3 การเจรจาต่อรองสำหรับแนวคิดสถาปัตยกรรมเชิงบริการ.....	35
2.3.1 บทบาทการเจรจาต่อรองในสถาปัตยกรรมเชิงบริการ.....	35
2.3.2 ตัวอย่างกรอบแนวคิดการเจรจาต่อรอง สำหรับสถาปัตยกรรมเชิงบริการ.....	37
2.4 สรุปท้ายบท.....	39
บทที่ 3 กรอบแนวคิดการเจรจาต่อรองสำหรับสถาปัตยกรรมเชิงบริการ	40
3.1 ความครอบคลุมของมาตรฐานการพัฒนา ซอฟต์แวร์กับแนวคิดการเจรจาต่อรอง.....	40
3.1.1 แบบจำลองวุฒิภาวะความสามารถแบบบูรณาการ.....	41
3.1.2 มาตรฐาน IEEE 1062, ฉบับพิมพ์ปี 1998 ข้อเสนอแนะการปฏิบัติสำหรับการจัดหาซอฟต์แวร์.....	45
3.1.3 มาตรฐานสากล ISO/IEC 12207 มาตรฐานสำหรับ เทคโนโลยีสารสนเทศ – กระบวนการวัฏจักรชีวิตของซอฟต์แวร์	47
3.2 กรอบแนวคิดการเจรจาต่อรองสำหรับสถาปัตยกรรมเชิงบริการ.....	49
3.3 กระบวนการเจรจาต่อรองสำหรับสถาปัตยกรรมเชิงบริการ	51

สารบัญ (ต่อ)

	หน้า
3.4 สรุปท้ายบท.....	54
บทที่ 4 การออกแบบระบบผู้ประสานงานการเจรจาต่อรอง	55
4.1 ผู้ประสานงานการเจรจาต่อรอง (Negotiation Coordinator)	55
4.2 ความสัมพันธ์และกิจกรรมของระบบ.....	55
4.3 ขั้นตอนการดำเนินงานของระบบ.....	61
4.4 สถาปัตยกรรมของระบบ	65
4.5 การแทนความรู้ (Knowledge Representation).....	70
4.6 สรุปท้ายบท.....	73
บทที่ 5 การพัฒนาระบบและกรณีตัวอย่าง.....	74
5.1 การพัฒนาระบบผู้ประสานงานการเจรจาต่อรอง.....	74
5.1.1 เครื่องมือที่ใช้ในการพัฒนา.....	74
5.1.2 การพัฒนาส่วนดำเนินการ.....	75
5.1.3 การพัฒนาฐานข้อมูลของระบบ.....	84
5.1.4 การพัฒนาการแทนความรู้สำหรับระบบ.....	92
5.1.5 การพัฒนาส่วนติดต่อผู้ใช้.....	92
5.2 การพัฒนากรณีตัวอย่าง	107
5.2.1 การพัฒนาส่วนดำเนินการ.....	107
5.2.2 การพัฒนาหน่วยลงทะเบียนบริการ.....	110
5.3 สรุปท้ายบท.....	111
บทที่ 6 การทดสอบระบบ	112
6.1 เครื่องมือที่ใช้ในการทดสอบ	112
6.2 การทดสอบบริการกรณีตัวอย่าง.....	113
6.2.1 การทดสอบกรณีตัวอย่างที่หนึ่ง.....	113
6.2.2 การทดสอบกรณีตัวอย่างที่สอง.....	118
6.3 การทดสอบหน่วยลงทะเบียนบริการ.....	122

สารบัญ (ต่อ)

	หน้า
6.3.1 การทดสอบความพร้อมใช้งาน.....	122
6.3.2 การทดสอบการบันทึกข้อมูล	123
6.3.3 การทดสอบการค้นหาข้อมูล	125
6.4 การทดสอบระบบผู้ประสานงานการเจรจาต่อรอง	128
6.5 สรุปท้ายบท.....	130
บทที่ 7 บทสรุปและข้อเสนอแนะ.....	131
7.1 สรุปผลการวิจัย.....	131
7.2 ข้อจำกัดของการดำเนินงานวิจัย.....	132
7.3 ข้อเสนอแนะ	133
บรรณานุกรม.....	135
ภาคผนวก	140
ภาคผนวก ก การติดตั้งระบบผู้ประสานงานการเจรจาต่อรอง	141
ภาคผนวก ข เอกสาร XML Schema ที่ใช้แทนความรู้.....	144
ภาคผนวก ค ตัวอย่างเอกสารสัญญา	155
ผลงานตีพิมพ์เผยแพร่จากวิทยานิพนธ์.....	167
ประวัติผู้เขียน	178

รายการตาราง

ตาราง	หน้า
2.1 การสนับสนุนหลักการแนวคิดเชิงบริการของเทคโนโลยีเว็บเซอร์วิส	25
4.1 คำอธิบายกิจกรรม register client	56
4.2 คำอธิบายกิจกรรม register provider	57
4.3 คำอธิบายกิจกรรม define requirement	57
4.4 คำอธิบายกิจกรรม analyze feasibility	57
4.5 คำอธิบายกิจกรรม discover services	58
4.6 คำอธิบายกิจกรรม select service from result list	58
4.7 คำอธิบายกิจกรรม make allowance of public announcement	58
4.8 คำอธิบายกิจกรรม make contract	59
4.9 คำอธิบายกิจกรรม view document	59
4.10 คำอธิบายกิจกรรม define service	60
4.11 คำอธิบายกิจกรรม submit proposal	60
4.12 คำอธิบายกิจกรรม retrieve information	61
4.13 เอกสารที่เกี่ยวข้องกับการเจรจาต่อรองในระบบ	69
5.1 ตารางข้อมูลรายละเอียดผู้ใช้ (user)	86
5.2 ตารางข้อมูลรายละเอียดผู้เรียกใช้บริการ (client)	86
5.3 ตารางข้อมูลรายละเอียดผู้ให้บริการ (provider)	86
5.4 ตารางข้อมูลที่อยู่ของผู้ใช้ (address)	87
5.5 ตารางข้อมูลของบริการ (service)	87
5.6 ตารางข้อมูลของส่วนดำเนินการของบริการ (operation)	88
5.7 ตารางข้อมูลของแพลตฟอร์มที่บริการสนับสนุน (platform)	88
5.8 ตารางข้อมูลการลงทะเบียนเป็นสมาชิก (registration)	89
5.9 ตารางข้อมูลการระบุความต้องการบริการของผู้เรียกใช้บริการ (requirement)	89
5.10 ตารางข้อมูลรายการบริการจากการค้นหา (candidateshortlist)	89
5.11 ตารางข้อมูลผลลัพธ์บริการที่ได้จากการค้นหา (candidateservice)	90

รายการตาราง (ต่อ)

ตาราง	หน้า
5.12 ตารางข้อมูลประกาศความต้องการสู่สาธารณะ (publicannouncement)	90
5.13 ตารางข้อมูลเอกสารข้อเสนอบริการ (serviceproposal)	90
5.14 ตารางข้อมูลเอกสารข้อเสนอบริการแบบไม่ระบุตัวตน (anonymousproposal)	90
5.15 ตารางข้อมูลเอกสารข้อตกลงบริการฉบับร่าง (draftagreement)	91
5.16 ตารางข้อมูลเอกสารข้อตกลงบริการ (serviceagreement)	91
5.17 ตารางข้อมูลการส่งมอบบริการ (delivery)	92
5.18 ตารางข้อมูลคุณลักษณะบริการของผู้ให้บริการรายที่หนึ่ง	107
5.19 ตารางข้อมูลคุณลักษณะบริการของผู้ให้บริการรายที่สอง	109
6.1 กรณีทดสอบบริการกรณีตัวอย่างที่หนึ่ง (Company One)	113
6.2 กรณีทดสอบบริการกรณีตัวอย่างที่สอง (Company Two)	119
6.3 กรณีทดสอบการบันทึกข้อมูลลงหน่วยลงทะเบียนบริการ	124
6.4 กรณีทดสอบการค้นหาข้อมูลลงหน่วยลงทะเบียนบริการ	125
6.5 กรณีทดสอบการทำงานของระบบในบทบาทผู้เรียกใช้บริการ	128
6.6 กรณีทดสอบการทำงานของระบบในบทบาทผู้ให้บริการ	129

รายการภาพประกอบ

ภาพประกอบ	หน้า
2.1 ตัวอย่างขอบเขตของบริการ (ที่มา: Erl, 2005)	10
2.2 ความสัมพันธ์และการดำเนินงานระหว่างองค์ประกอบบริการ	11
2.3 คุณสมบัติภาวะอิสระที่สามารถดำเนินการได้ด้วยตนเอง ของบริการภายใต้การดำเนินการหนึ่ง (ที่มา: Erl, 2005)	13
2.4 การประกอบบริการแบบ Orchestration (ที่มา: ทวีतीय, 2006)	14
2.5 การประกอบบริการแบบ Choreography (ที่มา: ทวีतीय, 2006)	15
2.6 ลักษณะการไม่เก็บสถานะ (Statelessness) ของบริการ (ที่มา: Erl, 2005)	16
2.7 องค์ประกอบสำหรับแนวคิดสถาปัตยกรรมเชิงบริการ (ที่มา: McIntosh, 2004)	17
2.8 โครงสร้างลำดับชั้นการทำงานของสถาปัตยกรรมเว็บเซอร์วิส (ที่มา: W3C, 2004)	20
2.9 ตัวอย่างเอกสาร XML	22
2.10 โครงสร้างของเอกสาร SOAP (ที่มา: Java Passion, 2010)	22
2.11 ตัวอย่างเอกสาร SOAP สำหรับการร้องขอบริการ	23
2.12 ตัวอย่างเอกสาร SOAP สำหรับการตอบกลับผลลัพธ์	23
2.13 โครงสร้างเอกสาร WSDL (ที่มา: Cerami, 2002)	24
2.14 กระบวนการเรียกใช้งานบริการ (ที่มา: Bruijn et al., 2008)	36
2.15 กรอบแนวคิดการเจรจาต่อรองสำหรับซอฟต์แวร์เชิงบริการ (ที่มา: Elfataty and Layzell, 2004)	38
<hr/>	
3.1 โครงสร้างซีเอ็มเอ็มไอของการแทนรูปแบบขั้นบันได (ที่มา: CMMI Product Team, 2002)	42
3.2 กลุ่มกระบวนการจัดการโครงการขั้นพื้นฐาน (ที่มา: CMMI Product Team, 2002)	44
3.3 กลุ่มกระบวนการจัดการโครงการแบบก้าวหน้า (ที่มา: CMMI Product Team, 2002)	44
3.4 ขั้นตอนการดำเนินงานของการจัดหาซอฟต์แวร์ตามมาตรฐานไออีอีอี 1062	46
3.5 กระบวนการหลักของมาตรฐานสากลไอเอสโอ/ไออีซี 12207 (ที่มา: Singh, 1996)	47

รายการภาพประกอบ

ภาพประกอบ	หน้า
3.6 กรอบแนวคิดของสถาปัตยกรรมของระบบเจรจาต่อรอง สำหรับการพัฒนาซอฟต์แวร์ตามสถาปัตยกรรมเชิงบริการ	49
3.7 แผนภาพกิจกรรมของการเจรจาต่อรองสำหรับสถาปัตยกรรม เชิงบริการที่สอดคล้องกับมาตรฐานการพัฒนาซอฟต์แวร์ที่ได้กล่าวมา	52
4.1 กิจกรรม (Use case) และความสัมพันธ์ระหว่างกิจกรรม ในระบบผู้ประสานงานการเจรจาต่อรอง	56
4.2 แผนภาพลำดับกิจกรรมของการดำเนินงานโดยรวม	63
4.3 แผนภาพลำดับกิจกรรมของการลงทะเบียนเป็นสมาชิกของระบบ	64
4.4 แผนภาพลำดับกิจกรรมของการเสนอบริการของผู้ให้บริการที่เป็นสมาชิกกับระบบ	64
4.5 สถาปัตยกรรมของระบบผู้ประสานงานการเจรจาต่อรอง (Negotiation Coordinator)	66
4.6 ขั้นตอนการสกัดข้อมูลผู้เข้าใช้จากการลงทะเบียนสมัครเป็นสมาชิกกับระบบ	67
4.7 ขั้นตอนการสกัดข้อมูลบริการจากการระบุบริการของผู้ให้บริการ	68
4.8 แผนภาพแบบจำลองออนโทโลยีของผู้ดำเนินการที่มีเกี่ยวข้องกับระบบ	71
4.9 แผนภาพแบบจำลองออนโทโลยีของบริการ	71
4.10 แผนภาพแบบจำลองออนโทโลยีของคุณภาพบริการ	72
4.11 ตัวอย่างการแทนความรู้ของบริการธนาคารออนไลน์ ที่เป็นไปตามออนโทโลยีของบริการ	73
<hr/>	
5.1 ส่วนการดำเนินงานของระบบผู้ประสานงานการเจรจาต่อรองที่สอดคล้อง กับแบบแผนการออกแบบระบบซอฟต์แวร์ Model-View-Controller (MVC)	76
5.2 แผนภาพคลาสของแบบแผนการเข้าถึงส่วนข้อมูล Data Accessor	77
5.3 แผนภาพลำดับการทำงานของการทำงานของการอ่านข้อมูลจาก ระบบจัดการฐานข้อมูลผ่านทาง Data Accessor	77
5.4 แผนภาพคลาสของระบบผู้ประสานงานการเจรจาต่อรอง	79
5.5 ตัวอย่างแผนภาพคลาสแสดงองค์ประกอบเชิงโครงสร้างของ Factory Method	81

รายการภาพประกอบ

ภาพประกอบ	หน้า
5.6 แผนภาพลำดับการทำงานของการสร้างเอกสารผ่านทางคลาส DocumentCreator ที่เป็นไปตามแบบแผนการออกแบบ Factory Method	81
5.7 ตัวอย่างแผนภาพคลาสแสดงองค์ประกอบเชิงโครงสร้างของ Adapter	83
5.8 แผนภาพลำดับการทำงานของการใช้ความสามารถของคลาส Notification ผ่านทางคลาส Requirement ที่เป็นไปตามแบบแผนการออกแบบ Adapter	84
5.9 แผนภาพแสดงความสัมพันธ์ระหว่างข้อมูลในฐานข้อมูลของระบบผู้ประสานงานการเจรจาต่อรอง	85
5.10 แผนภาพโครงสร้างของ XML Schema ที่สอดคล้องกับออนโทโลยีของผู้ดำเนินการที่มีส่วนเกี่ยวข้องกับระบบ	93
5.11 แผนภาพโครงสร้างของ XML Schema ที่สอดคล้องกับออนโทโลยีบริการ	94
5.12 แผนภาพโครงสร้างของ XML Schema ที่สอดคล้องกับออนโทโลยีคุณภาพของบริการ	95
5.13 แผนผังโครงสร้างโปรแกรมประยุกต์ผ่านเว็บของระบบผู้ประสานงานการเจรจาต่อรอง	95
5.14 หน้าจอหลักที่แสดงการเข้าใช้งานระบบผู้ประสานงานการเจรจาต่อรอง	96
5.15 หน้าจอแสดงการลงทะเบียนสมัครเป็นสมาชิกกับระบบ	97
5.16 หน้าจอหลักแสดงส่วนการทำงานเมื่อผู้ให้บริการเข้าสู่ระบบ	97
5.17 หน้าจอแสดงการกำหนดบริการของผู้ให้บริการ	98
5.18 หน้าจอแสดงรายการการประกาศความต้องการสู่สาธารณะ	99
5.19 หน้าจอแสดงรายการสำหรับการสร้างข้อเสนอบริการ	100
5.20 หน้าจอแสดงรายการร่างข้อตกลงบริการเพื่อให้ผู้ให้บริการพิจารณายืนยัน	100
5.21 หน้าจอแสดงรายการข้อตกลงบริการโดยผู้ให้บริการสามารถกำหนดการส่งมอบได้เมื่อพร้อม	101
5.22 หน้าจอแสดงรายการของบริการที่ได้กำหนดการส่งมอบเรียบร้อยแล้ว	102
5.23 หน้าจอหลักแสดงส่วนการทำงานเมื่อผู้เรียกใช้บริการเข้าสู่ระบบ	102

รายการภาพประกอบ

ภาพประกอบ	หน้า
5.24 หน้าจอแสดงรายการของข้อกำหนดความต้องการบริการ	103
5.25 หน้าจอแสดงการกำหนดความต้องการบริการของผู้เรียกใช้บริการ	104
5.26 หน้าจอแสดงรายการข้อเสนอบริการเพื่อพิจารณาตัดเลือก	105
5.27 หน้าจอแสดงรายการร่างข้อตกลงบริการเพื่อพิจารณายืนยัน	105
5.28 หน้าจอแสดงรายการข้อตกลงบริการที่ผ่านการยอมรับจากทั้งสองฝ่าย	106
5.29 หน้าจอแสดงรายการของบริการที่พร้อมสำหรับการเรียกใช้งาน	106
6.1 การเรียกใช้ส่วนดำเนินการ <code>getAvailableRoom</code> ของบริการ <code>HotelService</code>	114
6.2 ผลลัพธ์จากการเรียกใช้ <code>getAvailableRoom</code> ของบริการ <code>HotelService</code>	115
6.3 การเรียกใช้ส่วนดำเนินการ <code>bookingRoom</code> ของบริการ <code>HotelService</code>	115
6.4 ผลลัพธ์จากการเรียกใช้ <code>bookingRoom</code> ของบริการ <code>HotelService</code>	116
6.5 การเรียกใช้ส่วนดำเนินการ <code>getAvailableSeat</code> ของบริการ <code>AirTicket</code>	116
6.6 ผลลัพธ์จากการเรียกใช้ <code>getAvailableSeat</code> ของบริการ <code>AirTicket</code>	117
6.7 การเรียกใช้ส่วนดำเนินการ <code>bookingSeat</code> ของบริการ <code>AirTicket</code>	117
6.8 ผลลัพธ์จากการเรียกใช้ <code>bookingSeat</code> ของบริการ <code>AirTicket</code>	118
6.9 การเรียกใช้ <code>getAvailableRoom</code> ของบริการ <code>BookingHotel</code> และผลลัพธ์	120
6.10 การเรียกใช้ <code>bookingRoom</code> ของบริการ <code>BookingHotel</code> และผลลัพธ์	120
6.11 การเรียกใช้ <code>getAvailableSeat</code> ของบริการ <code>FlightBooking</code> และผลลัพธ์	121
6.12 การเรียกใช้ <code>bookingSeat</code> ของบริการ <code>FlightBooking</code> และผลลัพธ์	121
6.13 ตัวอย่างหน้าเว็บที่แสดงรายละเอียดของระบบลงทะเบียนบริการ <code>JUDDI</code>	123
6.14 การเรียกใช้การบันทึกข้อมูลผู้ให้บริการและผลลัพธ์ที่ได้กลับมา	124
6.15 การเรียกใช้การบันทึกข้อมูลของบริการและผลลัพธ์ที่ได้กลับมา	125
6.16 การเรียกใช้การค้นหาข้อมูลของผู้ให้บริการและผลลัพธ์ที่ได้กลับมา	126
6.17 การเรียกใช้การค้นหาข้อมูลของบริการและผลลัพธ์ที่ได้กลับมา	127
6.18 การเรียกใช้การค้นหาข้อมูลการเรียกใช้บริการและผลลัพธ์ที่ได้กลับมา	127
ก.1 การแสดงผลคลิกเลือก <code>Browse</code> ในส่วน <code>Deploy</code> ของการจัดการระบบ	142

รายการภาพประกอบ

ภาพประกอบ	หน้า
ข.1 เอกสาร XML Schema ของผู้ดำเนินการที่เกี่ยวข้องกับระบบ	144
ข.2 เอกสาร XML Schema ของบริการ	148
ข.3 เอกสาร XML Schema ของคุณภาพของบริการ	153
ค.1 ตัวอย่างเอกสารข้อกำหนดความต้องการบริการ (Requirement specification)	155
ค.2 ตัวอย่างเอกสารข้อเสนอบริการ (Service proposal)	157
ค.3 ตัวอย่างเอกสารข้อเสนอบริการแบบไม่ระบุตัวตน (Anonymous proposal)	162
ค.4 ตัวอย่างเอกสารข้อตกลงบริการ (Service Agreement)	165

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของงานวิจัย

แนวโน้มใหม่ของการพัฒนาซอฟต์แวร์ในยุคปัจจุบันจำเป็นต้องตอบสนองต่อความต้องการและโอกาสทางธุรกิจของบริษัทหรือองค์กรต่างๆให้สามารถดำเนินการได้อย่างมีประสิทธิภาพด้วยความสะดวกและรวดเร็ว ดังนั้นระเบียบวิธีการพัฒนาซอฟต์แวร์ (Software methodology) จึงต้องมีความยืดหยุ่นและปรับเปลี่ยนได้ง่ายกว่าในอดีตที่ผ่านมา และเมื่อเทคโนโลยีอินเทอร์เน็ตได้รับความนิยมทำให้ส่งผลต่อแนวทางการพัฒนาซอฟต์แวร์ในแง่ของการจัดหาและจำหน่ายชิ้นส่วนซอฟต์แวร์ (Software component) ประกอบกับการเปลี่ยนแปลงของเทคโนโลยีของการประมวลผลแบบกระจาย (Distributed computing) ในปัจจุบัน ทำให้มีการนำเสนอแนวคิดการพัฒนาซอฟต์แวร์เชิงบริการบนพื้นฐานแนวคิดสถาปัตยกรรมเชิงบริการ (Service-Oriented Architecture, SOA) (Bennett et al., 2000; McIntosh, 2004) ซึ่งปรับแนวคิดในกระบวนการผลิตซอฟต์แวร์ให้เป็นการให้บริการ (Service) ร่วมกันทำงานในกระบวนการประมวลผล

จากความต้องการด้านซอฟต์แวร์ของลูกค้าที่มีหลากหลายและเปลี่ยนแปลงไปตามเทคโนโลยี ประกอบกับผู้ให้บริการที่มีมากขึ้นและมีเงื่อนไขในการให้บริการที่หลากหลาย เป็นปัจจัยสำคัญ ทำให้การพัฒนากระบวนการเจรจาต่อรอง (Negotiation system) ระหว่างผู้ให้บริการ (Service provider) และผู้ใช้บริการ (Service client) มีความสำคัญในการสนับสนุนให้ได้ผลลัพธ์ที่สอดคล้องกับความต้องการของทั้งสองฝ่าย

ในโลกความเป็นจริงกระบวนการการเจรจาต่อรองค่อนข้างมีความซับซ้อนและยุ่งยาก การนำแนวคิดการเจรจาต่อรองที่ปรากฏมาประยุกต์ใช้ให้ทำงานได้จริงบนพื้นฐานทางเทคโนโลยีนั้นเป็นสิ่งท้าทาย ดังนั้นงานวิจัยนี้จึงมุ่งเน้นการออกแบบกรอบแนวคิดและโครงสร้างการทำงานของการเจรจาต่อรองสำหรับสนับสนุนแนวคิดสถาปัตยกรรมเชิงบริการ และดำเนินการพัฒนาระบบที่สอดคล้อง

ในสภาพแวดล้อมการเจรจาต่อรองอย่างเต็มรูปแบบซึ่งผู้ให้บริการและผู้ใช้บริการไม่ต้องเผชิญหน้ากันนั้น ความรู้เกี่ยวกับบริการ ความรู้เกี่ยวกับผู้ให้บริการ และความรู้เกี่ยวกับผู้ให้บริการจะเป็นปัจจัยสำคัญของระบบ

1.2 การตรวจเอกสารและงานวิจัยที่เกี่ยวข้อง

1.2.1 การตรวจเอกสาร

สถาปัตยกรรมเชิงบริการ (Service-oriented architecture) เป็นสถาปัตยกรรมที่สะท้อนให้เห็นถึงกระบวนการทัศน์สำหรับการจัดระเบียบและการใช้ประโยชน์จากขีดความสามารถที่ถูกแจกจ่าย (Distributed capabilities) โดยอยู่ภายใต้การควบคุมดูแลของขอบเขตกรรมสิทธิ์ (Ownership domains) ต่างๆที่หลากหลาย (OASIS, 2006) แนวคิดนี้จะช่วยให้การปรับเปลี่ยนหรือการประกอบบริการเพื่อพัฒนาซอฟต์แวร์กระทำได้อย่างอิสระ ทำให้มีความสามารถในการทำงานร่วมกับระบบอื่น (Interoperability) นอกจากนี้ยังช่วยสนับสนุนการนำกลับมาใช้ใหม่ (Reuse) ของบริการได้ และด้วยความสามารถดังกล่าวนี้ทำให้สถาปัตยกรรมเชิงบริการถูกประยุกต์ใช้สนับสนุนการดำเนินธุรกิจอิเล็กทรอนิกส์แบบธุรกิจกับธุรกิจ (Business-to-Business, B2B) ความสัมพันธ์ทางธุรกิจแบบ B2B นี้ หัวใจหลักคือการเจรจาต่อรองโดยมีคำกล่าวไว้ว่า “80 % ของธุรกิจได้มาจากการซื้อขายแบบเจรจาต่อรอง” (Hurwitz, 2000) ดังนั้นการเจรจาต่อรองช่วยทำให้คู่ค้าทางธุรกิจสามารถดำเนินงานได้อย่างมีประสิทธิภาพและเป็นไปตามความต้องการของทั้งสองฝ่าย

การพาณิชย์อิเล็กทรอนิกส์ (Electronic commerce) เป็นกระบวนการดำเนินงานทางธุรกิจรูปแบบใหม่ที่สำคัญในปัจจุบัน และได้มีการศึกษาพิจารณาถึงประเด็นเรื่องการเจรจาต่อรองสำหรับกระบวนการทางธุรกิจนี้เช่นกัน การทำให้การเจรจาต่อรองสัมฤทธิ์ผลในการพาณิชย์อิเล็กทรอนิกส์นั้น การศึกษาและวิจัยจะมุ่งความสนใจไปยังการเจรจาต่อรองแบบอัตโนมัติ (Automated negotiation) Beam และ Segev (1997) ได้ทำการสำรวจและศึกษาการเจรจาต่อรองแบบอัตโนมัติในงานพาณิชย์อิเล็กทรอนิกส์พบว่าประเด็นทางเทคโนโลยีที่ต้องคำนึงถึงสำหรับงานพัฒนาระบบการเจรจาต่อรอง ได้แก่ ระบบสนับสนุนการเจรจาต่อรอง (Negotiation support system) เอเจนต์ที่เฉลียวฉลาด (Intelligent agent) และระบบการสร้างการตัดสินใจ (Decision making) นอกจากนี้ยังระบุถึงปัญหาสำหรับการพัฒนาการเจรจาต่อรอง

แบบอัตโนมัติสำหรับการพาณิชย์อิเล็กทรอนิกส์เกี่ยวกับประเด็นเรื่องออนโทโลยี (Ontology) ที่ใช้อธิบายระบบ และปัญหาของกลยุทธ์การเจรจาต่อรองที่นำมาใช้งาน ดังนั้นเทคโนโลยีและปัญหาที่พบเหล่านี้จะเป็นปัจจัยที่ต้องพิจารณาในการศึกษาและวิจัยนี้ด้วย

งานวิจัยของ Al-Sakran และ Sergueivskaia (2006) ได้เสนอกรอบแนวคิดสำหรับการพัฒนาระบบการเจรจาต่อรองแบบอิเล็กทรอนิกส์ (Electronic negotiation) ซึ่งใช้เทคนิคการเรียนรู้โดยคอมพิวเตอร์แบบการให้เหตุผลบนฐานของเนื้อหา (Content based reasoning, CBR) มาช่วยในการตัดสินใจ ผสานกับเพิ่มความสามารถในการท่องเที่ยวในโครงข่ายข้อมูลเพื่อหาข้อพิจารณาประกอบการตัดสินใจที่นอกเหนือทางเลือกที่มี โดยใช้เทคโนโลยีเอเจนต์แบบเคลื่อนที่ (Mobile agent) ส่วนงานวิจัยของ Bartolini, Preist และ Jennings (2005) ได้เสนอกรอบแนวคิดสำหรับการเจรจาต่อรองในอีกรูปแบบหนึ่งโดยสร้างอนุกรมวิธาน (Taxonomy) ของกฎที่ใช้จัดการกลไกการเจรจาต่อรองที่หลากหลายโปรโตคอล (Protocol) สำหรับการติดต่อปฏิสัมพันธ์ที่สนับสนุนกลไกดังกล่าว และกรอบแนวคิดของซอฟต์แวร์สำหรับการเจรจาต่อรองที่ให้เอเจนต์ดำเนินงานโดยใช้กฎและโปรโตคอลเหล่านี้ โดยภาษาที่ใช้สำหรับการเจรจาต่อรองตั้งอยู่บนพื้นฐานของภาษา Web ontology language ระดับ Lite (OWL-Lite) (Dean and Schreiber, 2004)

1.2.2 งานวิจัยที่เกี่ยวข้อง

Software as a Service: Negotiation Perspective (Elfatry and Layzell, 2002)

งานวิจัยนี้ได้เสนอแง่มุมในการเจรจาต่อรองของระบบซอฟต์แวร์เชิงบริการ (Software as a service, SAAS) ซึ่งเป็นการมองซอฟต์แวร์หนึ่งๆเป็นบริการที่หาได้บนเครือข่าย โดยพิจารณาถึงคุณลักษณะของตลาดของซอฟต์แวร์เชิงบริการ และความต้องการที่จำเป็นในการสร้างกรอบแนวคิดขึ้นมา รวมถึงพิจารณาถึงผลลัพธ์ที่จะได้หลังจากเจรจาต่อรองบรรลุผลแล้ว

Negotiating in Service-Oriented Environment (Elfatry and Layzell, 2004)

งานวิจัยนี้เป็นการต่อยอดจากแนวคิดงานวิจัยในข้างต้น โดยได้เสนอคุณลักษณะของตลาดสำหรับซอฟต์แวร์เชิงบริการ และคุณลักษณะของการเจรจาต่อรองใน

สิ่งแวดล้อมนั้นๆ เพื่อนำมาระบุถึงโครงสร้างการดำเนินงานสำหรับซอฟต์แวร์เชิงบริการในรุ่นถัดไปให้เหมาะสมขึ้น นอกจากนี้ได้แสดงขั้นตอนของการเจรจาต่อรองซอฟต์แวร์เชิงบริการในรูปแบบของระยะ (Phase) ในการดำเนินงานอีกด้วย

Modeling e-Negotiation Processes for a Service Oriented Architecture (Benyoucef and Rinderle, 2006)

งานวิจัยนี้ได้เสนอระบบการเจรจาต่อรองแบบอิเล็กทรอนิกส์สำหรับสถาปัตยกรรมเชิงบริการจากกรอบแนวคิดและการพัฒนาระบบโดยใช้แบบจำลอง (Model-driven approach) โดยการประกบ (mapping) โปรโตคอลของการเจรจาต่อรองซึ่งอธิบายรูปแบบแผนภาพสถานะ (Statechart diagram) เข้ากับแนวคิดการจัดประสานเว็บเซอร์วิสเพื่อทำงานประสานกัน (Web service orchestration)

A Framework and Ontology for Dynamic Web Services Selection (Maximilien, 2004)

งานวิจัยนี้ได้เสนอระบบคัดเลือกบริการแบบพลวัต (Dynamic) โดยใช้กรอบแนวคิดเทคโนโลยีเอเจนต์ร่วมกับกับออนโทโลยีของคุณภาพบริการ (Quality of service: QoS) ซึ่งจะช่วยให้การค้นหบริการให้ได้ผลลัพธ์ที่ใกล้เคียงที่สุด

1.3 วัตถุประสงค์ของงานวิจัย

1. เพื่อออกแบบกรอบแนวคิดและโครงสร้างการทำงานของการเจรจาต่อรองแบบอิเล็กทรอนิกส์สำหรับสนับสนุนสถาปัตยกรรมเชิงบริการ
2. เพื่อพัฒนาระบบงานประยุกต์ตัวอย่างที่ใช้แนวคิดการเจรจาต่อรองแบบอิเล็กทรอนิกส์สำหรับสนับสนุนสถาปัตยกรรมเชิงบริการ

1.4 ขอบเขตการดำเนินงานของการวิจัย

ขอบเขตของการดำเนินงานการวิจัยมีรายละเอียดดังนี้

1. ออกแบบกรอบแนวคิดและโครงสร้างการทำงานโดยประยุกต์ใช้แนวคิดการเจรจาต่อรองแบบอิเล็กทรอนิกส์ที่สนับสนุนสถาปัตยกรรมเชิงบริการ โดยมีรายละเอียดดังนี้

- 1.1 ออกแบบกรอบแนวคิด (Framework) และลำดับขั้นตอนการทำงาน (Algorithm) ของการดำเนินการเจรจาต่อรอง
- 1.2 ออกแบบสถาปัตยกรรมระบบที่สนับสนุนกระบวนการเจรจาต่อรอง
- 1.3 ออกแบบส่วนประกอบสนับสนุนการทำงานของการทำงานของการเจรจาต่อรองตามที่จำเป็น
- 1.4 ออกแบบส่วนติดต่อกับผู้ใช้ (User Interface)
2. พัฒนาระบบงานประยุกต์ด้วยกรณีตัวอย่างตามต้นแบบโครงสร้างการทำงานที่ได้ออกแบบไว้

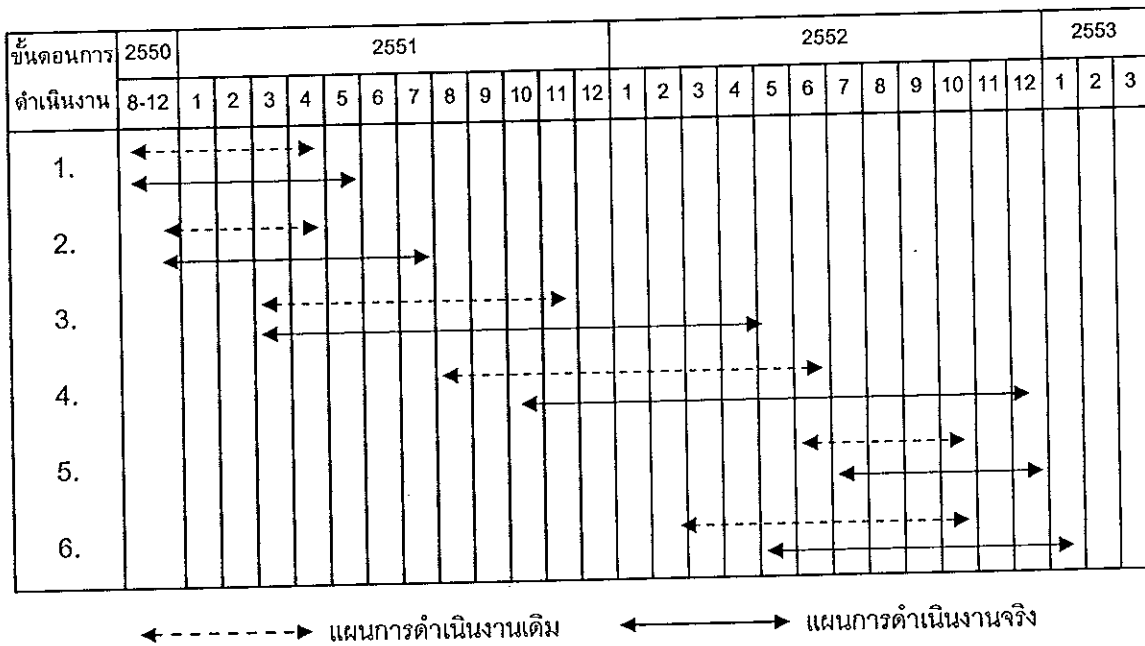
1.5 ขั้นตอนการดำเนินงาน

1. ศึกษาวิจัยและเอกสารที่เกี่ยวข้อง ได้แก่ แนวคิดสถาปัตยกรรมเชิงบริการ หลักการการเจรจาต่อรอง รวมถึงแนวคิดการเจรจาต่อรองสำหรับสถาปัตยกรรมเชิงบริการ และเทคโนโลยีอื่นๆที่เกี่ยวข้อง
2. ศึกษาและเลือกเทคโนโลยี เครื่องมือที่สนับสนุนในการสร้างและพัฒนาระบบงานประยุกต์ ตัวอย่างเช่น การโปรแกรมด้วยภาษาจาวา การโปรแกรมเพื่อพัฒนาเว็บเซอร์วิส การพัฒนาโปรแกรมประยุกต์บนเว็บ (Web application) เป็นต้น
3. วิเคราะห์และออกแบบกรอบแนวคิดสำหรับการเจรจาต่อรองที่สนับสนุนสถาปัตยกรรมเชิงบริการ รวมถึงลำดับขั้นตอนการทำงานของการทำงานการเจรจาต่อรองที่เกิดขึ้น
4. พัฒนาสถาปัตยกรรมระบบและส่วนประกอบต่างๆของระบบตามที่ได้วิเคราะห์และออกแบบไว้
5. ติดตั้งใช้งาน พร้อมทดสอบการทำงานของตัวกลางและส่วนต่างๆของระบบ
6. จัดทำเอกสารประกอบการวิจัยและวิทยานิพนธ์

1.6 ระยะเวลาดำเนินงาน

กันยายน 2550 - มกราคม 2553

รวมระยะเวลาทั้งสิ้น 2 ปี 6 เดือน การวางแผนการดำเนินงานในเบื้องต้นเทียบกับการดำเนินงานจริงดังแสดงในตารางที่ 1.1



1.7 สถานที่และเครื่องมือที่ใช้ในการดำเนินการวิจัย

1.7.1 สถานที่

ห้องวิจัยกลุ่มวิศวกรรมซอฟต์แวร์และงานประยุกต์ ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์

1.7.2 เครื่องมือที่ใช้

1. ด้านฮาร์ดแวร์ (Hardware)

- 1.1 เครื่องคอมพิวเตอร์ส่วนบุคคล หน่วยประมวลผล Intel Pentium IV 3.2 GHz หน่วยความจำ 1024 MB ฮาร์ดดิสก์ความจุ 160 GB จำนวน 2 เครื่องสำหรับการพัฒนาระบบตัวกลางที่สนับสนุนการเจรจาต่อรองและการจำลองผู้ให้บริการตามกรณีตัวอย่างที่กำหนด

2. ด้านซอฟต์แวร์ (Software)

- 2.1 ระบบปฏิบัติการ Microsoft Windows XP Professional
- 2.2 ระบบปฏิบัติการ Microsoft Windows Server 2003
- 2.3 Java Development Kit Standard Edition 1.6 เป็นภาษาในการพัฒนาระบบ
- 2.4 ระบบจัดการฐานข้อมูล MySQL 5.1

- 2.5 เครื่องมือสิ่งแวดล้อมสำหรับการพัฒนาแบบเบ็ดเสร็จสำหรับจาวา Netbeans 6.5
- 2.6 เครื่องมือจัดการและบริการระบบงานประยุกต์ผ่านเว็บ Glassfish 2 และ Apache Tomcat 5.5
- 2.7 Java API Library สำหรับรองรับการทำงานต่างๆ เช่น Jena API สำหรับการพัฒนางานออนโทโลยี, JAX-WS สำหรับการพัฒนาเว็บเซอร์วิส, JAXP สำหรับการประมวลผลภาษา XML เป็นต้น
- 2.8 โปรแกรมเว็บเบราว์เซอร์ Mozilla Firefox และ Google Chrome

1.8 ประโยชน์ที่ได้รับ

1. ได้กรอบแนวคิดและลำดับขั้นตอนการทำงานที่ประยุกต์ใช้แนวคิดการเจรจาต่อรองสำหรับสนับสนุนสถาปัตยกรรมเชิงบริการ
 2. ได้ระบบงานประยุกต์ที่พัฒนาตามกรอบแนวคิดและลำดับขั้นตอนการทำงานตามที่วิเคราะห์และออกแบบมา ซึ่งสามารถนำไปใช้เป็นต้นแบบสำหรับการพัฒนาซอฟต์แวร์ระบบงานประยุกต์
 3. ได้ตัวอย่างระบบงานประยุกต์ที่พัฒนาตามแนวทางสถาปัตยกรรมเชิงบริการที่สามารถนำไปสาธิตประกอบการจัดการเรียนการสอนทางวิศวกรรมซอฟต์แวร์ที่เกี่ยวกับหลักคิดการพัฒนาซอฟต์แวร์ตามแนวทางนี้
-

บทที่ 2

แนวคิดและทฤษฎีที่เกี่ยวข้อง

เนื้อหาในบทนี้กล่าวถึงแนวคิดและหลักการต่างๆที่เกี่ยวข้องกับสถาปัตยกรรมเชิงบริการ ทั้งในส่วนของความหมาย องค์ประกอบและคุณสมบัติของสถาปัตยกรรมเชิงบริการ รวมไปถึงบริการและหลักแนวคิดเชิงบริการ และในส่วนของสองเป็นการเสนอเนื้อหาและทฤษฎีต่างๆของการเจรจาต่อรองบนระบบอิเล็กทรอนิกส์

2.1 สถาปัตยกรรมเชิงบริการ

สถาปัตยกรรมเชิงบริการ (Service-Oriented Architecture) เป็นแนวคิดทางระบบสารสนเทศที่ได้รับความนิยมกว้างขวางในปัจจุบัน ซึ่งมีรายละเอียดพื้นฐานที่สำคัญรวมทั้งเนื้อหาต่างๆ ที่เกี่ยวข้องมากมาย ดังสรุปได้ดังต่อไปนี้

2.1.1 บริการและแนวคิดเชิงบริการ

2.1.1.1 บริการ

บริการ (Service) นับว่าเป็นส่วนประกอบพื้นฐานที่สำคัญที่สุดสำหรับแนวคิดสถาปัตยกรรมเชิงบริการ อย่างไรก็ตามบริการในบริบทของแนวคิดสถาปัตยกรรมเชิงบริการมีลักษณะและรายละเอียดแตกต่างจากบริการในบริบทโดยทั่วไป สามารถสรุปใจความสำคัญของบริการในแนวคิดสถาปัตยกรรมเชิงบริการได้ดังนี้

1. ความหมายของบริการ

ตามความหมายในพจนานุกรมภาษาอังกฤษ (Longman, 2009) บริการ หมายถึงงานที่บุคคลหนึ่งทำให้กับบุคคลหรือองค์กร โดยเฉพาะในช่วงเวลายาว ("The work that someone does for a person or organization, especially over a long period") แต่สำหรับใน

บริบทของระบบซอฟต์แวร์นั้น บริการมีความหมายในแง่ของส่วนโปรแกรมหนึ่งๆที่ประกอบด้วย ลำดับและขั้นตอนการทำงาน ที่สามารถให้ผลลัพธ์ที่ตอบสนองกับผู้ต้องการเรียกใช้งาน โดยผู้เรียกใช้งานเพียงแต่ส่งคำร้องขอไปให้ดำเนินงาน เมื่อประมวลผลเสร็จสิ้นก็จะส่งผลลัพธ์กลับมา (ทวิช, 2551)

Organization for Advancement of Structured Information Standards (OASIS) (2006) ซึ่งเป็นองค์กรสากลที่ขับเคลื่อนการพัฒนามาตรฐานทางการพาณิชย์ อิเล็กทรอนิกส์และเว็บเซอร์วิส ได้เสนอความหมายของบริการในบริบทของแนวคิด สถาปัตยกรรมเชิงบริการหมายถึงกลไกที่ทำให้เข้าถึงการดำเนินงานต่างๆผ่านทางส่วนติดต่อที่ ได้แนะนำไว้ และใช้งานตามแนวทางและข้อกำหนดที่ได้อธิบายอย่างสอดคล้อง

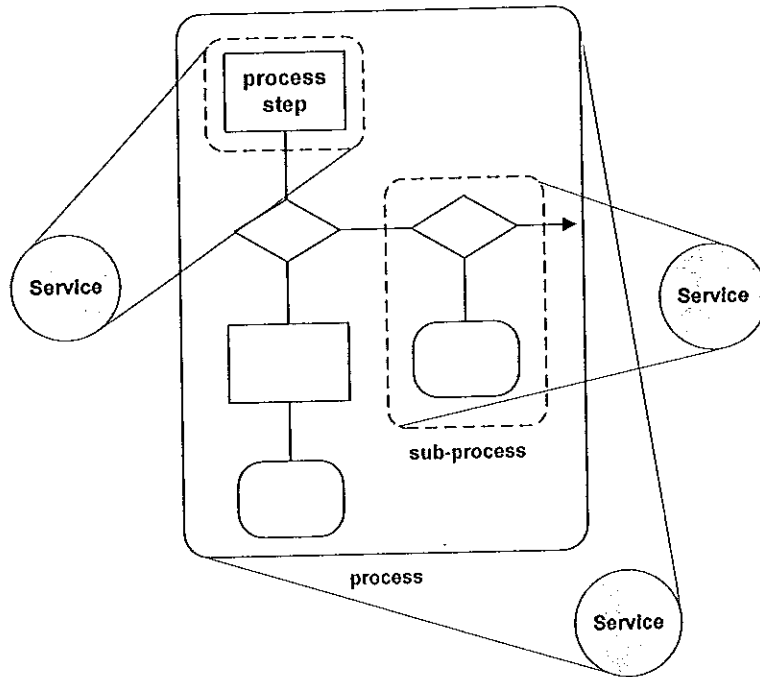
2. ขอบเขตการดำเนินการของบริการ

จากกระบวนการดำเนินงานของระบบซอฟต์แวร์ บริการสามารถเป็นได้ตั้งแต่ หน่วยการดำเนินงานขนาดเล็ก ไปจนถึงหน่วยการทำงานประมวลผลระดับที่ซับซ้อนขึ้น หรือแม้กระทั่งซอฟต์แวร์โดยองค์รวมก็สามารถเป็นบริการได้ด้วยเช่นกัน นอกจากนี้ขอบเขตของ บริการอาจจะรวมไปถึงขอบเขตของบริการตัวอื่นๆ ซึ่งในกรณีนี้บริการจะถูกประกอบเข้าไว้ ด้วยกันกลายเป็นบริการใหม่ ดังแสดงตัวอย่างในภาพประกอบที่ 2.1

3. รายละเอียดของบริการ (Service description)

ในแนวคิดสถาปัตยกรรมเชิงบริการนั้น ผู้เรียกใช้บริการอาจเป็นได้ทั้งบุคคล หรือองค์กร, โปรแกรมคอมพิวเตอร์ หรือบริการอื่นๆ ดังนั้นการปฏิสัมพันธ์กับบริการจำเป็นต้อง เข้าใจในวิธีการและรายละเอียดของการติดต่อสื่อสารกับบริการนั้นๆ สิ่งเหล่านี้จะได้มาจาก รายละเอียดของบริการ (Service description)

รายละเอียดของบริการในรูปแบบที่ง่ายที่สุดประกอบด้วย ชื่อของบริการ ข้อมูล นำเข้าและข้อมูลผลลัพธ์ที่ได้จากบริการ และอาจผนวกเพิ่มรายละเอียดต่างๆตามความจำเป็น โดยที่ยังต้องคงไว้ซึ่งคุณสมบัติการปกปิดส่วนที่ไม่จำเป็นเอาไว้



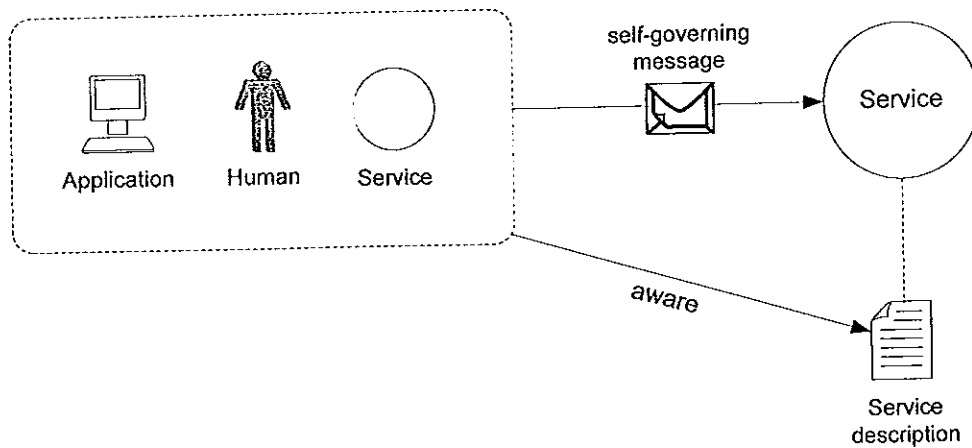
ภาพประกอบที่ 2.1 ตัวอย่างขอบเขตของบริการ (ที่มา: Erl, 2005)

4. ข่าวสารของบริการ (Message)

เนื่องจากบริการมีคุณสมบัติไม่เก็บค่าสถานะของการดำเนินการต่างๆเอาไว้ ดังนั้นบริการจะสูญเสียการรับรู้ใดๆจากข่าวสารต่างๆเมื่อส่งออกไปแล้ว การปฏิสัมพันธ์กับบริการจึงจำเป็นต้องมีรูปแบบกลางในการติดต่อสื่อสารที่เป็นมาตรฐานเพื่อให้ผู้มีส่วนได้ส่วนเสียทั้งหมดสามารถติดต่อกันได้อย่างสมบูรณ์

โดยสรุปแล้วบริการประกอบด้วยรายละเอียดบริการที่อธิบายวิธีการและรูปแบบ

ในการเข้าถึงและใช้งาน และติดต่อสื่อสารกันโดยใช้รูปแบบของข่าวสารที่เป็นมาตรฐานกลาง ดังแสดงในภาพประกอบที่ 2.2 ซึ่งจะเห็นได้ว่าลักษณะของสถาปัตยกรรมแบบนี้มีความใกล้เคียงกับสถาปัตยกรรมแบบกระจายในอดีต แต่ที่แตกต่างกันคือองค์ประกอบหลักทั้งสามคือตัวบริการ รายละเอียดบริการ และรูปแบบของข่าวสารที่ใช้จะต้องเป็นไปตามแนวคิดเชิงบริการนั่นเอง



ภาพประกอบที่ 2.2 ความสัมพันธ์และการดำเนินงานระหว่างองค์ประกอบบริการ

2.1.1.2 แนวคิดเชิงบริการ

โดยแท้จริงแล้วแนวคิดเชิงบริการมีพื้นฐานมาจากหลักการทางวิศวกรรมซอฟต์แวร์ที่เรียกว่า การแยกส่วนที่เกี่ยวข้องกัน (Separate of concerns) (Greer, 2008) ซึ่งหลักการนี้คือการแยกส่วนการทำงานของโปรแกรมคอมพิวเตอร์ไปสู่หน่วยย่อย โดยที่การทำงานของมันมีการซ้อนทับหรือคาบเกี่ยวระหว่างกันให้น้อยที่สุดเท่าที่เป็นไปได้ หลักการนี้ช่วยแยกย่อยวิธีการแก้ปัญหาที่เราสนใจให้เล็กลง โดยที่แต่ละส่วนการทำงานจะแก้ปัญหาเฉพาะเรื่องใดเรื่องหนึ่งเท่านั้น

การแยกส่วนที่เกี่ยวข้องกันถูกนำไปประยุกต์ใช้ในหลากหลายแนวคิด ทั้งแนวคิดการพัฒนาโปรแกรมเชิงวัตถุ (Object-oriented development) แนวคิดเชิงชิ้นส่วน (Component-based development) และรวมทั้งแนวคิดเชิงบริการด้วยเช่นกัน

แนวคิดเชิงบริการ (Service orientation) เป็นแนวคิดที่มองการดำเนินการต่างๆ ในระบบซอฟต์แวร์ให้แยกออกจากกันตามหลักการ Separate of concerns ซึ่งแต่ละส่วนการดำเนินการย่อยนั้นคือ บริการ (Service) โดยบริการจะต้องสนับสนุนคุณลักษณะต่างๆ ดังต่อไปนี้ (Erl, 2005)

- บริการสามารถนำกลับมาใช้ใหม่ได้ (Reusability)

แนวคิดเชิงบริการสนับสนุนให้บริการเกิดความสามารถในการนำกลับมาใช้ใหม่ตามคำร้องขอของผู้เรียกใช้ ดังนั้นเมื่อมีการเปลี่ยนแปลงความต้องการระบบซอฟต์แวร์ในอนาคตจะทำให้ใช้เวลาในการพัฒนาน้อยลงและสะดวกมากยิ่งขึ้น

- บริการเสนอกฎเกณฑ์ข้อตกลงในการใช้งาน (Service contract)

ข้อตกลงในการใช้งานจะประกอบด้วย ตำแหน่งปลายทางที่บริการอยู่ (Service endpoint), การดำเนินการต่างๆที่บริการมี, รูปแบบของการนำเข้าข้อมูล (Input) และการออกผลลัพธ์ (Output) และกฎเกณฑ์คุณลักษณะของบริการนั้นๆ ทั้งนี้อาจรวมไปถึงสัญญาการดำเนินการระหว่างผู้ให้บริการและผู้ใช้บริการก็เป็นได้ ข้อตกลงการใช้งานที่ดีนั้นสมควรจะมีการอธิบายรายละเอียดความหมายของสิ่งต่างๆประกอบควบคู่ไปด้วย

- บริการมีรูปแบบการผูกติดแบบหลวม (Loose coupling)

การผูกติดแบบหลวมหรือ Loose coupling เป็นสถานะเงื่อนไขที่หน่วยการทำงานหนึ่งมีความเกี่ยวข้องกับอีกหน่วยการทำงานหนึ่งน้อยที่สุด หรือการดำเนินงานไม่พึ่งพาต่อกัน ถ้าบริการสามารถทำงานร่วมกับบริการตัวอื่นๆได้โดยที่ตัวมันเองยังคงสามารถดำเนินงานได้คงเดิม นั่นคือบริการมีการผูกติดแบบหลวมระหว่างกัน

การจะทำให้เกิดการผูกติดแบบหลวมระหว่างหน่วยการทำงานนั้นๆ จำเป็นต้องมีส่วนติดต่อกกลางในการช่วยติดต่อสื่อสารระหว่างกันซึ่งก็คือ Service contract ดังที่กล่าวมา

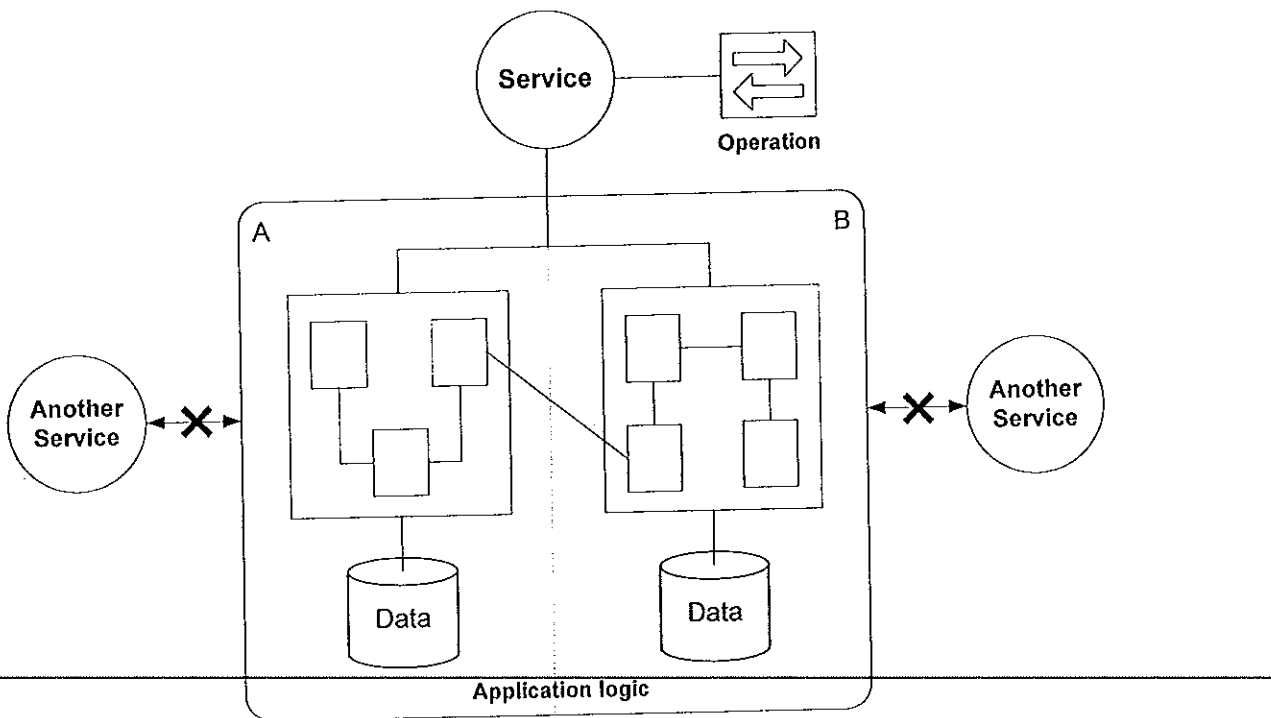
- บริการซ่อนตรรกะที่ไม่เกี่ยวข้องกับการดำเนินการเอาไว้ (Abstraction)

Abstraction เป็นทฤษฎีที่ทำให้บริการเปรียบเสมือนกับกล่องดำ คือทำการซ่อนหรือปกปิดรายละเอียดส่วนอื่นๆที่นอกเหนือจากรายละเอียดในส่วนที่เกี่ยวข้องกับการดำเนินการเอาไว้ เนื่องจากไม่มีข้อกำหนดใดๆที่จะบอกว่าจะขอบเขตของการดำเนินแคไหนที่บริการสมควรปกปิด ดังนั้นการพิจารณาว่าส่วนใดต้องปกปิดนั้นจะขึ้นอยู่กับแนวทางการออกแบบการดำเนินการของบริการ และเป้าหมายของบริการนั้นๆเป็นสำคัญ

- บริการที่สามารถดำเนินการได้สมบูรณ์ด้วยตนเอง (Autonomy)

ภาวะอิสระที่สามารถดำเนินการได้ด้วยตนเอง (Autonomy) เป็นปัจจัยสำคัญในการพิจารณาว่าการดำเนินการส่วนใดบ้างสมควรจะจัดเป็นบริการหรือจะจัดกลุ่มการดำเนินการแคไหนที่จะทำให้เป็นบริการ การเรียกบริการใดๆว่ามีภาวะอิสระ แสดงว่าบริการนั้นจะต้องอิสระจากปัจจัยที่เกี่ยวข้องจากภายนอก โดยสามารถควบคุมการดำเนินการได้ด้วย

ตนเองโดยไม่มีข้องเกี่ยวจากบริการหรือสิ่งแวดล้อมอื่นๆภายนอก ดังนั้นระดับภาวะอิสระของบริการแสดงให้เห็นถึงระดับความเป็นอิสระในการจัดการและควบคุมการดำเนินการด้วยตนเองของบริการนั้น ซึ่งสามารถอธิบายลักษณะของภาวะอิสระได้ดังภาพประกอบที่ 2.3 บริการหนึ่งประกอบด้วยการทำงานภายในสองส่วนคือ A และ B ที่ทำงานเชื่อมโยงกัน ซึ่งจะเห็นได้ว่าบริการนี้สามารถทำงานได้สมบูรณ์ได้ด้วยตนเอง ไม่เกี่ยวข้องกับบริการอื่นๆภายนอก แต่ถ้าเราพิจารณาให้ส่วนดำเนินการ A เป็นบริการหนึ่งและส่วนดำเนินการ B เป็นอีกบริการหนึ่งจะสังเกตได้ว่าบริการทั้งสองนั้นไม่สามารถทำงานให้สมบูรณ์ได้ในตัวเอง จำเป็นต้องพึ่งพาอาศัยซึ่งกันและกัน บริการในลักษณะนี้ถือว่าขาดคุณสมบัติภาวะอิสระ



ภาพประกอบที่ 2.3 คุณสมบัติภาวะอิสระที่สามารถดำเนินการได้ด้วยตนเอง
ของบริการภายใต้การดำเนินการหนึ่ง (ที่มา: Erl, 2005)

ในบางกรณีบริการอาจไม่จำเป็นต้องเป็นภาวะอิสระโดยสมบูรณ์ เพียงแต่ในช่วงการดำเนินงานของบริการ บริการสามารถควบคุมตรรกะการดำเนินงานได้ก็เพียงพอ ซึ่งจากลักษณะของภาวะอิสระนี้เราสามารถแบ่งได้ออกเป็น 2 ชนิดคือ

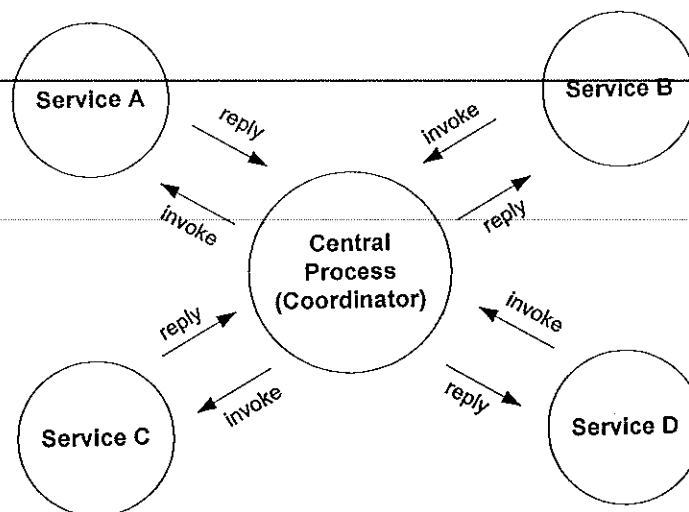
(1) **Service-level autonomy** การดำเนินการของบริการแยกออกจากกัน อย่างชัดเจน แต่ยังสามารถใช้งานทรัพยากรร่วมกันได้ ตัวอย่างเช่น ในระบบซอฟต์แวร์ที่มีอยู่เดิม (Legacy system) เมื่อถูกปรับเปลี่ยนให้อยู่ในแนวคิดเชิงบริการแล้ว การดำเนินการต่างๆ จะสามารถทำงานได้อย่างอิสระ แต่ทรัพยากรของระบบที่ใช้ อาจจำเป็นต้องใช้งานร่วมกันกับระบบอื่นๆ ที่มีอยู่

(2) **Pure autonomy** การดำเนินการของบริการและทรัพยากรต่างๆ ที่ใช้ แยกออกจากกันโดยสมบูรณ์ ลักษณะนี้จะเกิดกับระบบที่สร้างตามแนวคิดเชิงบริการจากตั้งต้น ไม่ได้ปรับเปลี่ยนมาจากระบบที่มีอยู่เดิม

- บริการสามารถนำมาประกอบกันได้ (Composibility)

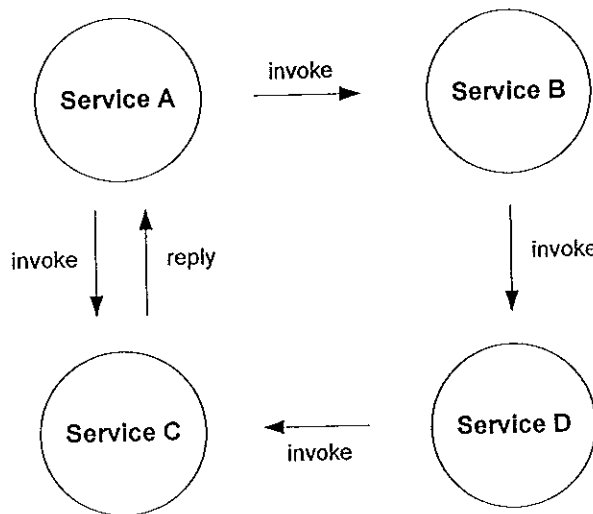
ในบางสถานการณ์บริการหนึ่งๆ อาจจะไม่สามารถแก้ปัญหาตามความต้องการที่เกิดขึ้นได้ แต่การประกอบบริการต่างๆ เข้าไว้ด้วยกันเพื่อให้ได้ผลลัพธ์ที่ตรงตามความต้องการอาจเป็นหนทางที่ดีกว่า ดังนั้นการออกแบบบริการจำเป็นต้องคำนึงถึงความสามารถในการประกอบกันของบริการไว้ด้วย แนวคิดที่ขยายมาจากแนวคิดการประกอบกันของบริการมี ดังนี้ (ทวิตีย์, 2006)

(1) **Orchestration** เป็นแนวคิดการประกอบกันของบริการโดยที่มี กระบวนการกลางเป็นตัวคอยควบคุมลำดับการดำเนินการส่งข้อมูลนำเข้าหรือรับข้อมูลออกของ เหล่าบริการที่นำมาประกอบกัน ดังแสดงในภาพประกอบที่ 2.4 กระบวนการกลางนี้จะปฏิบัติตาม กระบวนการทางธุรกิจ (Business process) ที่เกิดขึ้นและจะกลายเป็นบริการใหม่อีกบริการ หนึ่ง



ภาพประกอบที่ 2.4 การประกอบบริการแบบ Orchestration (ที่มา: ทวิตีย์, 2006)

(2) **Choreography** คือแนวคิดการประกอบกันของบริการโดยที่ไม่มีกระบวนการกลางคอยควบคุมอยู่ บริการจะทราบเองว่าเมื่อใดจะต้องดำเนินการและเมื่อใดจะต้องส่งผลลัพธ์ไปให้อีกบริการหนึ่งดำเนินงาน ดังแสดงในภาพประกอบที่ 2.5 ดังนั้นบริการทุกตัวจะต้องรู้ถึงกระบวนการทางธุรกิจที่ดำเนินการอยู่ซึ่งต่างจาก Orchestration ที่บริการในกลุ่มไม่จำเป็นต้องรู้ถึงกระบวนการทางธุรกิจ แต่จะเป็นหน้าที่ของกระบวนการกลางแทน



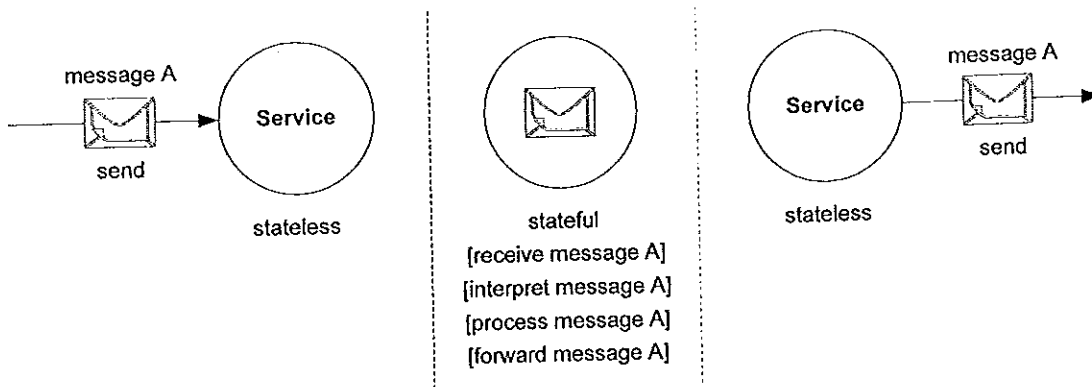
ภาพประกอบที่ 2.5 การประกอบบริการแบบ Choreography (ที่มา: ทวีติย์, 2006)

- บริการมีลักษณะไม่เก็บสถานะ (Statelessness)

เมื่อบริการได้รับข้อมูลนำเข้าแล้วดำเนินการ บริการควรจะไม่มีการเก็บค่าสถานะของข้อมูลนั้นเอาไว้ กล่าวคือดำเนินการเสร็จเรียบร้อยแล้วก็จบกันไป ไม่สามารถจำค่าข้อมูลจากการดำเนินการนั้นๆ เพื่อนำไปใช้กับคำสั่งใหม่ต่อไปได้ อย่างไรก็ตามในขณะที่บริการกำลังดำเนินงานอยู่นั้น บริการมีลักษณะเก็บสถานะชั่วคราว เพื่อขัดขวางมิให้ผู้เรียกใช้บริการรายอื่นเข้ามาดำเนินงาน เป็นการป้องกันไม่ให้เกิดการทำงานที่ซ้ำซ้อนในทางหนึ่ง ดังแสดงในภาพประกอบที่ 2.6

- บริการสนับสนุนต่อกลไกในการค้นหา (Discoverability)

การค้นหาบริการช่วยหลีกเลี่ยงการสร้างบริการที่มีการทำงานซ้ำซ้อน ดังนั้นบริการจำเป็นต้องได้รับการออกแบบที่สนับสนุนต่อกลไกในการค้นหาบริการ นั่นคือมีรายละเอียดของบริการและการดำเนินงานที่เพียงพอ



ภาพประกอบที่ 2.6 ลักษณะการไม่เก็บสถานะ (Statelessness) ของบริการ (ที่มา: Erl, 2005)

2.1.2 สถาปัตยกรรมเชิงบริการและองค์ประกอบ

2.1.2.1 ความหมายของสถาปัตยกรรมเชิงบริการ

สถาปัตยกรรมเชิงบริการ เป็นแนวคิดสำหรับระบบซอฟต์แวร์ที่สามารถดำเนินงานได้จากการติดต่อสื่อสารระหว่างกันของส่วนการทำงานย่อยต่างๆ ที่เรียกว่าบริการ (Service) ซึ่งถูกจัดวางไว้ให้ทำหน้าที่อย่างใดอย่างหนึ่งอย่างชัดเจนบนระบบเครือข่าย โดยใช้เทคโนโลยีมาตรฐานเปิด (Open standard) ในการพัฒนาทั้งหมด อันที่จริงแล้วสถาปัตยกรรมเชิงบริการถือว่าเป็นกลไกในการประมวลผลแบบกระจาย (Distributed computing) ลักษณะหนึ่ง และการพัฒนาระบบซอฟต์แวร์ให้เป็นไปตามสถาปัตยกรรมเชิงบริการนั้น จะต้องมีแบบแผนการดำเนินการที่ตั้งอยู่บนแนวคิดเชิงบริการ (Service orientation) ด้วย

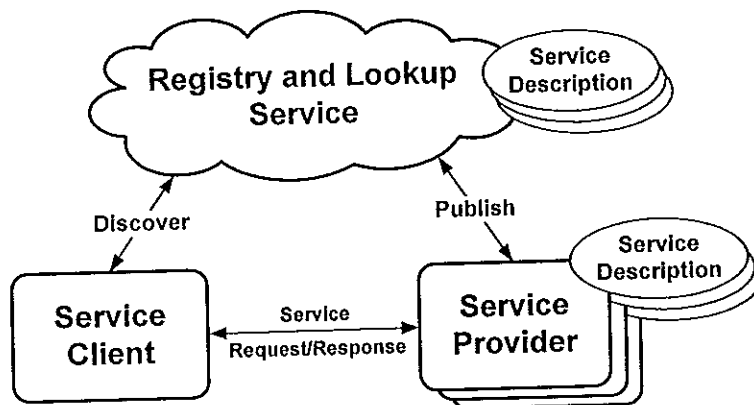
2.1.2.2 องค์ประกอบของสถาปัตยกรรมเชิงบริการ

การประยุกต์ใช้แนวคิดสถาปัตยกรรมเชิงบริการให้เป็นผลนั้น จำเป็นต้องปรับกรอบแนวคิดให้เหมาะสมต่อการนำไปประยุกต์ใช้งาน โดย (McIntosh, 2004) ได้ให้ข้อคิดเกี่ยวกับส่วนการทำงานหลักที่จำเป็นสำหรับการประยุกต์ใช้สถาปัตยกรรมเชิงบริการดังนี้

- ข้อตกลงในการทำการสื่อสาร การกำหนดความหมายของการสื่อสาร และวิธีการในการสื่อสาร หรือที่เรียกว่าโปรโตคอล (Protocol) สำหรับการติดต่อสื่อสารระหว่างกัน
- การอธิบายข้อกำหนดและการเชื่อมต่อบริการ ซึ่งแสดงให้เห็นว่าบริการดังกล่าวสามารถทำงานอะไรได้และจะติดต่อกับบริการได้อย่างไร

- กระบวนการการประกาศรายละเอียดของบริการนั้น ๆ เพื่อให้ผู้ใช้บริการได้รับทราบ และกระบวนการการค้นหาบริการตามที่ใช้บริการต้องการ
- ส่วนกลางที่คอยรับแจ้งการประกาศบริการจากผู้ให้บริการ และผู้ใช้บริการสามารถร้องขอบริการตามที่กำหนด

สามารถแสดงแบบจำลององค์ประกอบของสถาปัตยกรรมเชิงบริการ ดังภาพประกอบที่ 2.7



ภาพประกอบที่ 2.7 องค์ประกอบสำหรับแนวคิดสถาปัตยกรรมเชิงบริการ
(ที่มา: McIntosh, 2004)

จากภาพประกอบที่ 2.7 จะเห็นได้ว่าแนวคิดสถาปัตยกรรมเชิงบริการประกอบด้วย 3 ส่วนหลักดังนี้ (Chu, 2005)

1. ผู้ให้บริการ (Service provider) อาจเป็นได้ทั้งองค์กร บริษัทหรือหน่วยงานทางธุรกิจที่มีบริการและรายละเอียดบริการพร้อมสำหรับการใช้งาน
2. ผู้เรียกใช้บริการ (Service client หรือ Service consumer) เป็นบุคคลใดๆที่ต้องการเรียกใช้บริการ
3. หน่วยลงทะเบียนบริการ (Registry and lookup service) เป็นส่วนกลางที่คอยเก็บรวบรวมและจัดกลุ่มรายละเอียดของบริการจากผู้ให้บริการ และเสนอกลไกในการค้นหาบริการตามที่ใช้บริการต้องการ

ทั้งนี้สามารถอธิบายการดำเนินงานโดยสรุประหว่างส่วนต่างๆในสถาปัตยกรรมเชิงบริการได้คือ ผู้ให้บริการจะระบุและกำหนดรายละเอียดบริการของตนและประกาศรายละเอียดบริการนั้นไว้ที่หน่วยลงทะเบียนบริการ แล้วเมื่อผู้เรียกใช้บริการต้องการใช้บริการก็จะสามารถค้นหาได้ในหน่วยลงทะเบียนบริการ เมื่อพบรายละเอียดบริการตามที่ต้องการแล้ว ผู้เรียกใช้บริการก็สามารถเรียกใช้บริการจากผู้ให้บริการได้โดยตรง

2.1.2.3 คุณสมบัติของสถาปัตยกรรมเชิงบริการ

แนวคิดสถาปัตยกรรมเชิงบริการมีประโยชน์ในหลากหลายรูปแบบคุณลักษณะขึ้นอยู่กับจุดประสงค์และวิธีการนำไปใช้งาน สามารถอธิบายคุณลักษณะต่างๆบางส่วนได้ดังนี้ (Erl, 2005)

1. สนับสนุนให้ระบบทำงานร่วมกันได้อย่างแท้จริง

สถาปัตยกรรมเชิงบริการทำให้สามารถสร้างระบบที่เกิดจากการนำบริการมาประกอบกัน โดยบริการเหล่านี้ไม่ยึดติดกับโครงสร้างของการสื่อสารแบบใด แต่จะใช้โครงสร้างการสื่อสารกลางในการดำเนินการ ดังนั้นเพียงแต่ระบบใด ๆ เข้าใจกับรูปแบบการสื่อสารกลางนี้ได้ก็สามารถดำเนินการได้ทันที สถาปัตยกรรมเชิงบริการจึงสนับสนุนให้เกิดการทำงานระหว่างระบบ (Interoperability) ไม่ว่าจะระบบนั้นจะถูกพัฒนาขึ้นมาในแพลตฟอร์มใดก็ตาม

2. สนับสนุนให้ระบบเพิ่มความสามารถใหม่ได้อย่างรวดเร็ว

จากแนวคิดเชิงบริการในประเด็นความสามารถในการประกอบบริการทำให้ระบบตามแนวคิดสถาปัตยกรรมเชิงบริการสามารถเพิ่มความสามารถใหม่เข้าไปได้โดยง่าย เพียงแต่สร้างบริการที่เหมาะสมสำหรับงานหนึ่ง ๆ แล้วประกอบเข้ากับกลุ่มบริการที่ดำเนินงานที่

มีอยู่

3. ช่วยเพิ่มความคล่องตัวให้กับระบบสารสนเทศขององค์กรหรือบริษัท

องค์กรหรือบริษัทใด ๆ ก็ตามที่ประยุกต์ใช้แนวคิดสถาปัตยกรรมเชิงบริการเข้ากับระบบสารสนเทศของตน เมื่อเกิดการปรับปรุงกระบวนการดำเนินงานในองค์กร เช่น การนำเทคโนโลยีใหม่เข้ามาใช้งาน หรือการปรับปรุงสมรรถนะของระบบสารสนเทศ องค์กรนั้น ๆ เพียงแต่เปลี่ยนแปลงในส่วนการทำงานของบริการ โดยไม่จำเป็นต้องเข้าไปเปลี่ยนแปลงในส่วนการดำเนินหลักใดๆเลย ส่งผลให้ช่วยลดต้นทุนค่าใช้จ่ายและระยะเวลาที่เกิดจากการเปลี่ยนแปลงนี้ ลักษณะดังกล่าวเป็นการเพิ่มความคล่องตัวให้กับระบบขององค์กรได้เป็นอย่างดี

4. สนับสนุนให้เกิดการรวมกันของโปรแกรมประยุกต์

เมื่อองค์กรหรือบริษัทใด ๆ ต้องการนำแนวคิดสถาปัตยกรรมเชิงบริการมาใช้งาน องค์กรนั้น ๆ อาจจะไม่ต้องนำแนวคิดนี้เข้ามาปรับเปลี่ยนและแทนที่ระบบเดิมทั้งหมด เนื่องจากความสามารถในการดำเนินการระหว่างระบบดั้งที่ได้กล่าวมาในข้างต้น ทำให้ส่วนการดำเนินงานต่าง ๆ หรือระบบโปรแกรมประยุกต์ดั้งเดิมที่มีอยู่สามารถดำเนินการควบคู่ไปกับระบบงานใหม่ได้ทันที เพียงแค่รวบรวมการดำเนินการจากระบบดั้งเดิมให้เป็นบริการที่เหมาะสม และใช้รูปแบบการสื่อสารมาตรฐานกลางในการติดต่อสื่อสาร

5. ช่วยเพิ่มคุณภาพของการดำเนินงานให้แก่ระบบ

การนำแนวคิดสถาปัตยกรรมเชิงบริการเข้ามาประยุกต์ใช้ในระบบสารสนเทศใด ๆ ก็ตามทำให้เกิดความปลอดภัยและความน่าเชื่อถือมากกว่าการประยุกต์ใช้งานเทคโนโลยีการประมวลผลแบบกระจายแบบอื่น ๆ เนื่องจากเหตุผลดังนี้

- บริการและข่าวสารมีคุณสมบัติ Autonomy ตามข้างต้น ช่วยทำให้ป้องกันการเข้าถึงการดำเนินงานที่บริการปกป้องอยู่ได้
- บริการมีลักษณะไม่เก็บค่าสถานะของการดำเนินการเอาไว้ เมื่อข้อมูลข่าวสารนำเข้ามาไม่ครบถ้วน บริการนั้น ๆ ก็ไม่สามารถดำเนินการได้ ซึ่งถ้าเพิ่มความสามารถในการตรวจสอบความสมบูรณ์ของข่าวสารก็เท่ากับว่าเป็นการเพิ่มความน่าเชื่อถือให้กับการส่งข่าวสารไปในตัว
- และความสามารถในการยกเลิกการดำเนินงานเมื่อข่าวสารไม่สมบูรณ์ นี้จะเป็นคุณลักษณะที่ช่วยสนับสนุนให้เกิดความน่าเชื่อถือในงานทางด้านธุรกรรม (Transaction) ได้เป็นอย่างดี แต่อย่างไรก็ตามอาจจำเป็นต้องขยายความสามารถของบริการและข่าวสารให้สนับสนุนในอีกทางหนึ่งด้วย

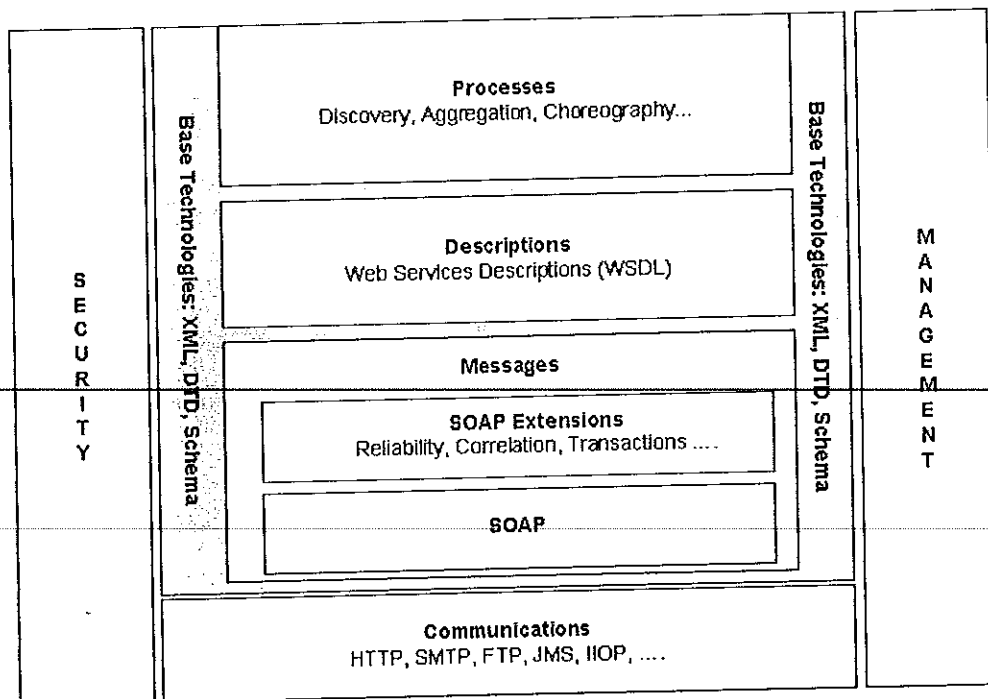
2.1.3 เทคโนโลยีสนับสนุนสถาปัตยกรรมเชิงบริการ

ในยุคปัจจุบันมีการพัฒนาเทคโนโลยีและมาตรฐานต่าง ๆ ขึ้นมาเพื่อตอบสนองการทำงานที่สอดคล้องกับแนวคิดสถาปัตยกรรมเชิงบริการ เช่น เทคโนโลยี CORBA (Common Object Request Broker Architecture) (CORBA, 2010) สนับสนุนให้โปรแกรมสามารถทำงานติดต่อระหว่างกันโดยไม่ขึ้นกับแพลตฟอร์มโดยใช้โปรโตคอลเฉพาะของตน เทคโนโลยี DCOM (Distributed Component Object Model) (DCOM, 2010) ของบริษัทไมโครซอฟต์ เน้นสนับสนุนการติดต่อระหว่างส่วนการทำงานผ่านทางระบบเครือข่ายที่ใช้ผลิตภัณฑ์ของบริษัท

เท่านั้น และเทคโนโลยี RMI (Java Remote Method Invocation) (RMI, 2010) สนับสนุนการทำงานระหว่างระบบที่ใช้เทคโนโลยีจาวาเหมือนกัน อย่างไรก็ตามเทคโนโลยีดังกล่าวมายังมีข้อบกพร่องบางส่วนที่ทำให้ไม่สามารถนำมาประยุกต์ใช้พัฒนาระบบตามแนวคิดสถาปัตยกรรมเชิงบริการได้สมบูรณ์ ด้วยเหตุนี้จึงมีการสร้างมาตรฐานเว็บเซอร์วิสซึ่งประยุกต์ใช้ได้ดีสำหรับการพัฒนาระบบภายใต้แนวคิดสถาปัตยกรรมเชิงบริการ

1. สถาปัตยกรรมเว็บเซอร์วิส

โครงสร้างการทำงานของสถาปัตยกรรมเว็บเซอร์วิสมีลักษณะการทำงานตามแนวคิดสถาปัตยกรรมเชิงบริการ แต่กำหนดเทคโนโลยีในการใช้งานไว้เฉพาะตามมาตรฐานที่องค์กร W3C (World Wide Web Consortium) กำหนดไว้กล่าวคือสถาปัตยกรรมเว็บเซอร์วิสยังคงมีส่วนดำเนินการประกาศ การค้นหา และการเรียกใช้บริการเช่นเดียวกับแนวคิดสถาปัตยกรรมเชิงบริการ แต่กำหนดมาตรฐานการดำเนินการส่วนต่างๆไว้โดยเฉพาะ ซึ่งสามารถแสดงเป็นโครงสร้างสถาปัตยกรรมเป็นชั้นของการทำงานได้ดังภาพประกอบที่ 2.8



ภาพประกอบที่ 2.8 โครงสร้างลำดับชั้นการทำงานของสถาปัตยกรรมเว็บเซอร์วิส (ที่มา: W3C, 2004)

สถาปัตยกรรมแบ่งออกเป็น 4 ส่วนหลักคือ การสื่อสาร (Communications) ข่าวสาร รายละเอียดและกระบวนการ (Messages, Descriptions and Process) ความปลอดภัย (Security) และการจัดการ (Management) ซึ่งมีเพียงสองส่วนแรกเท่านั้นที่ W3C ได้กำหนดมาตรฐานเจาะจงไว้ อีกสองส่วนที่เหลือนั้นเป็นหน้าที่ของผู้พัฒนาที่จะเลือกใช้เทคโนโลยีหรือวิธีการใดๆ มาประยุกต์ใช้

ส่วนการสื่อสารได้กำหนดให้ใช้โปรโตคอลพื้นฐานบนอินเทอร์เน็ต โดยเฉพาะอย่างยิ่งการใช้โปรโตคอล HTTP (Hypertext Transfer Protocol) ซึ่งจะช่วยให้ผู้ใช้เรียกใช้บริการและผู้ให้บริการสามารถติดต่อระหว่างกันได้โดยง่ายเนื่องจากเป็นโปรโตคอลที่ใช้งานกันโดยทั่วไป

สำหรับส่วนข่าวสาร รายละเอียดและกระบวนการนั้นมีมาตรฐานสำคัญสำหรับการทำงานหรือพัฒนา ได้แก่ SOAP (Simple Object Access Protocol) (SOAP, 2007), WSDL (Web Services Description Language) (WSDL, 2001) และ UDDI (Universal Description, Discovery and Integration) (UDDI, 2002) ซึ่งจะกล่าวถึงรายละเอียดในหัวข้อถัดไป

2. มาตรฐานต่างๆที่เกี่ยวข้องกับเทคโนโลยีเว็บเซอร์วิส

- XML (Extensible Markup Language) (XML, 2008)

XML เป็นภาษาสำหรับการอธิบายหรือนิยามความหมายของข้อมูล ปัจจุบันได้รับความนิยมใช้เป็นมาตรฐานในการกำหนดรูปแบบเอกสารที่ใช้บนระบบคอมพิวเตอร์และเครือข่าย ภาษา XML มีรูปแบบลักษณะการอธิบายข้อมูลด้วยการใช้แท็ก (Tag) เช่น <title>Service-Oriented Architecture</title> ซึ่งทำให้มนุษย์สามารถอ่านเข้าใจได้ อีกทั้งยังสนับสนุนให้เครื่องคอมพิวเตอร์นำไปประมวลผลได้โดยสะดวก ดังนั้นสามารถนำมาพัฒนาประยุกต์ใช้กับโปรแกรมคอมพิวเตอร์ให้สามารถประมวลผลเอกสารโดยไม่ขึ้นกับระบบหรือแพลตฟอร์มใดๆ การนำ XML มาใช้เพื่อการติดต่อสื่อสารจึงถือว่าเป็นเทคนิควิธีการที่ได้รับการยอมรับและอำนวยความสะดวกต่อการดำเนินงาน ซึ่งสถาปัตยกรรมเว็บเซอร์วิสได้กำหนดให้ XML เป็นพื้นฐานหลักสำหรับข้อมูลที่ใช้แลกเปลี่ยนระหว่างกัน ตัวอย่างของเอกสาร XML สามารถแสดงได้ดังภาพประกอบที่ 2.9

```

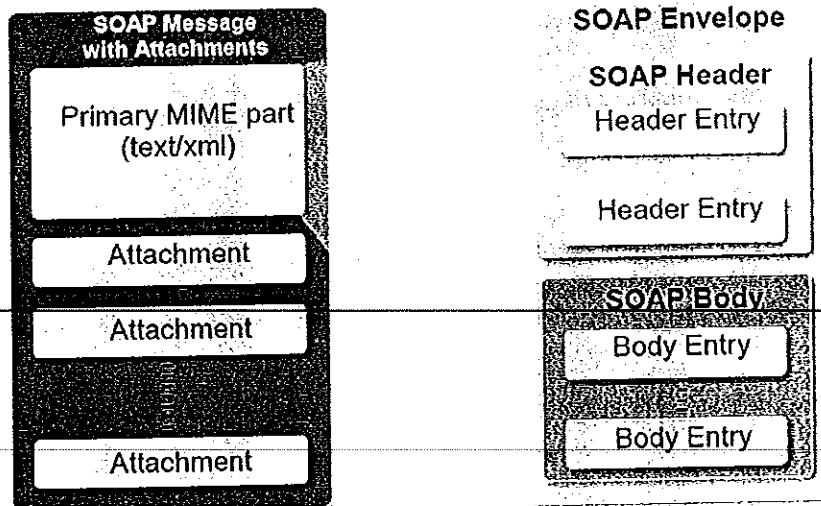
<books>
  <book isbn="12345688">
    <title>Web service technology</title>
    <author>
      <firstname>John</firstname>
      <lastname>Smith</lastname>
    </author>
  </book>
</books>

```

ภาพประกอบที่ 2.9 ตัวอย่างเอกสาร XML

- SOAP (Simple Object Access Protocol) (SOAP, 2007)

SOAP เป็นโปรโตคอลที่ใช้สำหรับการสื่อสารของเว็บเซอร์วิส ลักษณะของ SOAP เป็นเหมือนเอกสารจดหมายซึ่งเรียกว่า SOAP Message โดยจะถูกแนบไปกับโปรโตคอลสื่อสารอินเทอร์เน็ตที่ใช้งาน สามารถแสดงโครงสร้างของเอกสาร SOAP ดังภาพประกอบที่ 2.10



ภาพประกอบที่ 2.10 โครงสร้างของเอกสาร SOAP (ที่มา: Java Passion, 2010)

SOAP Message จะถูกจัดส่งด้วยโปรโตคอล HTTP และสามารถแนบเพิ่มข้อมูลชนิดอื่นพร้อมกับเอกสารได้โดยบรรจุอยู่ในส่วน Attachment เนื้อหาข้อมูลภายในจะอยู่ในรูปแบบ XML ซึ่งอยู่ในส่วน SOAP Envelope แบ่งออกได้เป็น 2 ส่วนย่อย คือ SOAP Header เป็นส่วนเลือกเพิ่มเติมอาจประกอบด้วยเนื้อหาคำสั่งต่างๆที่ต้องการให้ประมวลผลกับข้อมูลหรืออาจเป็นข้อมูลอื่นๆ ที่เกี่ยวข้องกับการติดต่อเว็บเซอร์วิสนั้นๆ

ส่วนที่สองคือ SOAP Body เป็นส่วนข้อมูลเนื้อหาหลักสำหรับการประมวลผล โดยผู้เรียกใช้บริการจะกำหนดข้อมูลร้องขอการทำงาน และผู้ให้บริการจะตอบกลับผลลัพธ์จากการทำงานไว้ในส่วนนี้ ตัวอย่างเอกสาร SOAP ของการร้องขอและผลตอบกลับสามารถแสดงได้ในภาพประกอบที่ 2.11 และ 2.12 ตามลำดับ

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schema.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schema.xmlsoap.org/soap/encoding">
  <SOAP-ENV:Body>
    <m:GetLastCurrencyRate xmlns:m="Some-URI">
      <symbol>BATH</symbol>
    </m:GetLastCurrencyRate>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

ภาพประกอบที่ 2.11 ตัวอย่างเอกสาร SOAP สำหรับการร้องขอบริการ

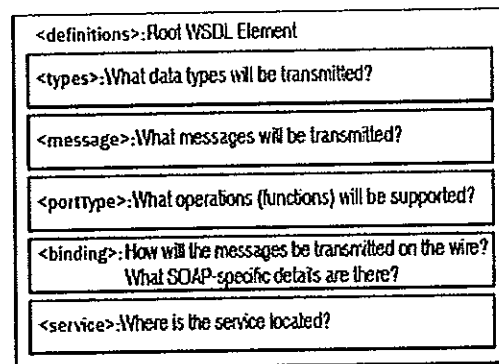
```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schema.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schema.xmlsoap.org/soap/encoding">
  <SOAP-ENV:Body>
    <m:GetLastCurrencyRateResponse xmlns:m="Some-URI">
      <Price>34.5</Price>
    </m:GetLastCurrencyRateResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

ภาพประกอบที่ 2.12 ตัวอย่างเอกสาร SOAP สำหรับการตอบกลับผลลัพธ์

- **WSDL (Web Service Description Language) (WSDL, 2001)**

WSDL เป็นภาษาในรูปแบบ XML ที่ใช้สำหรับอธิบายเว็บเซอร์วิสเพื่อเป็นการบอกให้ทราบว่าเว็บเซอร์วิสนั้นมีจุดเชื่อมต่อเพื่อใช้งานที่ตำแหน่งไหน ข้อความที่ใช้ในการสื่อสารมีลักษณะเป็นอย่างไร หรือมีคำสั่งอะไรบ้างที่สามารถเรียกใช้งานได้ WSDL จะช่วยให้สามารถทำการสื่อสารหรือแลกเปลี่ยนข้อมูลเอกสารกันได้อย่างถูกต้อง

เอกสาร WSDL ประกอบด้วยเนื้อหาต่างๆแบ่งออกได้เป็น 6 ส่วนสำคัญ ได้แก่ <definitions> เป็นส่วนแรกของเอกสารซึ่งเป็นรากของเนื้อหาทั้งหมด <types> เป็นส่วนสำหรับการกำหนดชนิดข้อมูลต่างๆ ที่จะนำไปใช้งาน <message> เป็นส่วนกำหนดข้อความที่ใช้เป็นข้อมูลคำสั่งหรือข้อมูลตอบกลับของเว็บเซอร์วิส <portType> เป็นส่วนกำหนดชุดคำสั่งต่างๆ ที่จะเกิดขึ้นในการติดต่อเว็บเซอร์วิส <binding> เป็นส่วนกำหนดรูปแบบชนิดของโปรโตคอลที่เว็บเซอร์วิสให้บริการ และรูปแบบข้อมูลที่ส่งผ่านโปรโตคอลนั้นๆ ในแต่ละคำสั่งการทำงาน และ<service> เป็นส่วนสุดท้ายสำหรับกำหนดชื่อของบริการและที่อยู่ของเว็บเซอร์วิส ดังแสดงได้ในภาพประกอบที่ 2.13



ภาพประกอบที่ 2.13 โครงสร้างเอกสาร WSDL (ที่มา: Cerami, 2002)

- **UDDI (Universal Description, Discovery and Integration) (UDDI, 2002)**

UDDI เป็นมาตรฐานที่เกิดขึ้นเพื่อสนับสนุนการจัดการทะเบียนและการสืบค้นเว็บเซอร์วิส ซึ่งจะช่วยให้ผู้ให้บริการสามารถเผยแพร่บริการของตนเองให้เป็นที่รู้จักแก่สาธารณะและ ในด้านผู้ต้องการเรียกใช้บริการก็สามารถค้นหาบริการตามความต้องการที่เหมาะสมได้

ลักษณะโครงสร้างการจัดเก็บข้อมูลของ UDDI จะใช้การอธิบายเว็บเซอร์วิส ออกเป็น 3 ส่วนคือ White page เป็นข้อมูลขององค์กรหรือบริษัทที่รับผิดชอบเว็บเซอร์วิสนั้นๆ Yellow page เป็นข้อมูลอธิบายรายละเอียดของเว็บเซอร์วิส พร้อมทั้งจัดเป็นหมวดหมู่ตาม โครงสร้างการจัดกลุ่มที่กำหนด และ Green page เป็นข้อมูลรายละเอียดที่เกี่ยวกับการนำเว็บ เซอร์วิสไปใช้งาน เช่น ที่อยู่ของเว็บเซอร์วิส และเอกสาร WSDL เป็นต้น

3. การสนับสนุนแนวคิดเชิงบริการของเทคโนโลยีเว็บเซอร์วิส

จากการพิจารณาเทคโนโลยีเว็บเซอร์วิส จะเห็นได้ว่าสามารถสนับสนุน หลักการของแนวคิดเชิงบริการได้ในบางส่วน ซึ่งมีรายละเอียดดังแสดงในตารางที่ 2.1 (Erl, 2005)

ตารางที่ 2.1 การสนับสนุนหลักการแนวคิดเชิงบริการของเทคโนโลยีเว็บเซอร์วิส

หลักการแนวคิดเชิงบริการ	การสนับสนุนของเทคโนโลยีเว็บเซอร์วิส
1. การนำกลับมาใช้ใหม่ (Reusability)	เว็บเซอร์วิสไม่ได้ถูกกำหนดให้มีความสามารถ สนับสนุนคุณสมบัติการนำกลับมาใช้ใหม่ แต่การ ทำให้เว็บเซอร์วิสมีคุณสมบัตินี้ขึ้นอยู่กับสภาพการ ทำงานของเว็บเซอร์วิสเอง
2. กฎเกณฑ์ข้อตกลงในการใช้งาน (Service contract)	เว็บเซอร์วิสกำหนดรายละเอียดของบริการโดยใช้ มาตรฐาน WSDL ดังนั้นเว็บเซอร์วิสสนับสนุน คุณสมบัตินี้
3. รูปแบบการผูกติดแบบหลวม (Loose coupling)	การสั่งดำเนินการเว็บเซอร์วิสทำได้โดยส่งข้อมูล นำเข้าแล้วเว็บเซอร์วิสจะประมวลผลจนได้ผลลัพธ์ ส่งกลับให้ผู้เรียกใช้ ซึ่งจะเห็นได้ว่าเว็บเซอร์วิส สามารถดำเนินการได้จนสำเร็จโดยไม่เกี่ยวข้องกับ เว็บเซอร์วิสอื่นๆ ดังนั้นเว็บเซอร์วิสจึงมีคุณสมบัติ การผูกติดแบบหลวม
4. การซ่อนตรรกะที่ไม่เกี่ยวข้องกับการดำเนินการ (Abstraction)	การติดต่อสื่อสารกับเว็บเซอร์วิสมีลักษณะการ ดำเนินการแบบกล่องดำ (Black box) คือไม่เห็น การดำเนินการภายใน เพราะข้อมูลการสื่อสารมี เพียงพารามิเตอร์ข้อมูลนำเข้าและข้อมูลผลลัพธ์ที่ เกี่ยวข้องเท่านั้น ดังนั้นเว็บเซอร์วิสสนับสนุน คุณสมบัตินี้

ตารางที่ 2.1 (ต่อ)

หลักการแนวคิดเชิงบริการ	การสนับสนุนของเทคโนโลยีเว็บเซอร์วิส
5. ความสามารถดำเนินการได้ด้วยตนเอง (Autonomy)	เว็บเซอร์วิสไม่ได้ถูกกำหนดให้มีความสามารถสนับสนุนคุณสมบัติการดำเนินการได้ด้วยตนเอง แต่การทำให้เว็บเซอร์วิสมีคุณสมบัตินี้ขึ้นอยู่กับออกแบบการทำงานของระบบที่เรียกใช้เว็บเซอร์วิส
6. ความสามารถในการประกอบกัน (Composibility)	ผลลัพธ์ของเว็บเซอร์วิสหนึ่งสามารถกำหนดเป็นข้อมูลนำเข้าของอีกเว็บเซอร์วิสหนึ่งได้ ดังนั้นเว็บเซอร์วิสสนับสนุนคุณสมบัตินี้
7. ลักษณะไม่เก็บสถานะ (Statelessness)	เว็บเซอร์วิสไม่ได้ถูกกำหนดให้มีความสามารถสนับสนุนคุณสมบัติการไม่เก็บค่าสถานะ แต่การทำให้เว็บเซอร์วิสมีคุณสมบัตินี้ขึ้นอยู่กับวิธีการสร้างการทำงานของระบบเบื้องหลังของเว็บเซอร์วิส
8. กลไกสำหรับการค้นหา (Discoverability)	เว็บเซอร์วิสไม่ได้ถูกกำหนดให้มีความสามารถรองรับกลไกสำหรับการค้นหา แต่มีมาตรฐานอื่นที่เว็บเซอร์วิสกำหนดเพื่อช่วยสนับสนุนการลงทะเบียนและค้นหา ซึ่งคือ UDDI ดังนั้นเว็บเซอร์วิสสนับสนุนคุณสมบัตินี้ผ่านทางการประยุกต์ใช้งานมาตรฐานการค้นหาอื่นๆ

อย่างไรก็ตามหลักการแนวคิดเชิงบริการที่เทคโนโลยีเว็บเซอร์วิสไม่รองรับ

นั้น สามารถพัฒนาเพื่อเสริมการดำเนินการได้ในภายหลัง และปัจจุบันได้มีมาตรฐานทางเทคโนโลยีแบบใหม่ที่ได้รับการพัฒนาเพื่อให้สนับสนุนหลักการแนวคิดเชิงบริการเพิ่มเติมจากที่เทคโนโลยีเว็บเซอร์วิสมีอยู่ ตัวอย่างเช่น Web Service Extension ต่างๆ (IBM, 2010) ได้แก่ WS-AtomicTransaction หรือ WS-I Basic Profile เป็นต้น

2.2 การเจรจาต่อรอง (Negotiation)

การเจรจาต่อรองเป็นกลไกสำคัญที่ช่วยผู้มีส่วนได้ส่วนเสียที่เกี่ยวข้องได้รับผลลัพธ์ที่ตรงตามความต้องการของตนเพื่อประโยชน์ทั้งในทางการแข่งขันและการร่วมมือซึ่งกันและกัน การเจรจาต่อรองถูกนำไปใช้หลากหลายบริบท อาทิ วิทยาศาสตร์ สังคมศาสตร์ เศรษฐศาสตร์ เป็นต้น โดยเฉพาะอย่างยิ่งบทบาทในบริบทการพาณิชย์อิเล็กทรอนิกส์ (Electronic commerce) ซึ่งเป็นประเด็นสำคัญที่งานวิจัยนี้ได้สนใจทำการศึกษา โดยมีรายละเอียดดังต่อไปนี้.

2.2.1 แนวคิดการเจรจาต่อรองโดยทั่วไป

2.2.1.1 ความหมายของการเจรจาต่อรอง

การเจรจาต่อรอง เป็นวิธีการที่เกิดขึ้นควบคู่กับสังคมของมนุษย์มาช้านาน ซึ่งเป็นหนทางที่ทำให้บุคคลสามารถจัดการกับความแตกต่างระหว่างกันไม่ว่าความแตกต่างนั้นจะเกิดขึ้นในสถานการณ์หรือเหตุการณ์ใดก็ตาม

คำว่า Negotiation ในภาษาอังกฤษนั้นมีรากศัพท์มาจากคำในภาษาละตินคือ Negotiatus ซึ่งแปลว่าการดำเนินธุรกิจ (Harper, 2009) และเมื่อค้นหาความหมายการเจรจาต่อรองในภาษาอังกฤษพบว่าหมายถึง การพิจารณาโต้เถียงและการจัดการร่วมกันในขอบข่ายของการติดต่อทางธุรกิจหรือการค้นหาข้อตกลง (Random House, 2009)

สำหรับความหมายการเจรจาต่อรองในเชิงวิชาการนั้น ได้มีงานศึกษาและวิจัยจำนวนหนึ่งให้คำจำกัดความของการเจรจาต่อรองเอาไว้ดังนี้

(1) D.G.Pruitt (1986) ได้เสนอความหมายของการเจรจาต่อรองโดยทั่วไปคือ กระบวนการที่ทำให้เกิดการตัดสินใจร่วมกันของผู้มีส่วนได้ส่วนเสียตั้งแต่สองฝ่ายขึ้นไป โดยการตัดสินใจนั้นเริ่มมาจากทุกฝ่ายเสนอความต้องการของตนซึ่งไม่ตรงกัน แล้วเคลื่อนไปสู่การตกลงร่วมกันโดยการรอมชอมหรือเสนอทางเลือกใหม่ขึ้นมา

(2) Benyoucef และ Rinderle (2006) ได้กล่าวว่า การเจรจาต่อรองเป็นกลไกที่ทำให้เกิดปฏิสัมพันธ์ที่มีลักษณะการดำเนินงานแบบทำซ้ำขั้นตอนเดิม (Recursive) จนได้ผลลัพธ์ตามที่ต้องการระหว่างผู้ตั้งต้นกับผู้ตอบสนองในสถานการณ์หนึ่งๆ

(3) Bichler และคณะ (2003) เสนอว่าการเจรจาต่อรองคือกระบวนการติดต่อดสื่อสารและการตัดสินใจที่ดำเนินการซ้ำระหว่างผู้มีส่วนเกี่ยวข้องหรือตัวแทน ซึ่งยังไม่บรรลุเป้าหมายของตน, ติดต่อกันแลกเปลี่ยนข้อมูลระหว่างกันและค้นหาการตกลงยอมรับระหว่างกันและกัน

กล่าวโดยสรุปแล้วการเจรจาต่อรองจะเป็นวิธีการหรือกระบวนการที่ทำให้ผู้มีส่วนได้ส่วนเสียที่เกี่ยวข้องทั้งหมดสามารถอภิปรายระหว่างกัน เพื่อให้ได้รับผลลัพธ์ที่ต่างฝ่ายต่างยอมรับได้

2.2.1.2 การเจรจาต่อรองโดยทั่วไป

การทำความเข้าใจถึงการเจรจาต่อรองนั้นจำเป็นต้องเข้าใจถึงองค์ประกอบพื้นฐานที่ประกอบกันขึ้น ซึ่งโดยทั่วไปแล้วการเจรจาต่อรองสามารถแบ่งองค์ประกอบเป็น 3 ส่วนหลัก (Rosa, 2009) ได้แก่ กระบวนการ (Process) พฤติกรรม (Behavior) และเนื้อหาใจความ (Substance) โดยกระบวนการจะเกี่ยวข้องกับสิ่งที่ทำให้ผู้มีส่วนร่วมสามารถเจรจาต่อรองได้ เช่น บริบทของการเจรจาต่อรอง ผู้มีส่วนร่วม ยุทธวิธีที่ผู้มีส่วนร่วมนำมาใช้ และลำดับหรือขั้นตอนการดำเนินการที่ทำให้ผู้มีส่วนร่วมกระทำ ส่วนพฤติกรรมจะบ่งบอกถึงความสัมพันธ์ระหว่างผู้มีส่วนร่วม การติดต่อดสื่อสาร และรูปแบบลีลาที่ผู้มีส่วนร่วมนำมาใช้ สุดท้ายเนื้อหาใจความจะเกี่ยวกับสิ่งที่ผู้มีส่วนร่วมนำมาพิจารณาเจรจาต่อรองกัน เช่น ข้อกำหนดการเจรจาต่อรอง ประเด็นในการเจรจาต่อรองนั้นๆ เป้าหมายที่จะเจรจาต่อรองไปให้ถึง เป็นต้น

ลักษณะของการเจรจาต่อรองตามรูปแบบของเป้าหมายที่ได้รับสามารถแบ่งออกได้เป็น 2 ประเภท (นิตยสารภาพข่าวทักษิณ, 2009) ประเภทแรกคือการเจรจาต่อรองแบบแบ่งปันปันส่วน (Distributive negotiation) เป็นการเจรจาต่อรองเกี่ยวกับการกระจายผลประโยชน์ที่มีปริมาณคงที่และจำกัด โดยประเด็นสำคัญที่คำนึงถึงคือ "ใครจะเป็นผู้ได้รับผลประโยชน์สูงสุด" ดังนั้นในกระบวนการเมื่อถึงที่สุดจะต้องมีฝ่ายหนึ่งแพ้และอีกฝ่ายหนึ่งได้ผลประโยชน์นั้นไปหรือที่เรียกว่ามีการแพ้ชนะ (Win-lose) สำหรับบางคนอาจเรียกการเจรจาต่อรองแบบนี้ว่า การเจรจาต่อรองแบบผลรวมเป็นศูนย์ (Zero-sum) หรือผลรวมคงที่ (Constant-sum) ซึ่งมูลค่าสำหรับคนที่ได้เพิ่มและคนที่สูญเสียมีจำนวนเท่ากัน

ประเภทที่สองคือการเจรจาต่อรองแบบบูรณาการ (Integrative negotiation) หมายถึงการเจรจาต่อรองที่แต่ละฝ่ายให้ความร่วมมือกันเพื่อให้เกิดประโยชน์สูงสุด โดยการบูร

ผลการผลประโยชน์ของตนให้กลายเป็นข้อตกลง ทุกฝ่ายที่ร่วมเจรจาต่อรองจะร่วมมือกันเพื่อสร้างผลประโยชน์โดยรวมให้มากที่สุด และในขณะเดียวกันก็มีการแข่งขันเพื่อจัดสรรผลประโยชน์รวมดังกล่าวให้เหมาะสมสำหรับแต่ละฝ่ายด้วย การทำข้อตกลงในลักษณะนี้เป็นการสร้างสรรค์ และเรียกร่องผลประโยชน์ อาจเรียกการเจรจาต่อรองในลักษณะนี้ว่ามีการชนะทั้งคู่ (Win-win)

2.2.2 การเจรจาต่อรองแบบอิเล็กทรอนิกส์ (Electronic negotiation)

การเจรจาต่อรองแบบอิเล็กทรอนิกส์เป็นการนำแนวคิดการเจรจาต่อรองที่เกิดขึ้นในโลกความเป็นจริงมาประยุกต์ใช้ในระบบคอมพิวเตอร์และเครือข่าย ซึ่งจะช่วยสนับสนุนให้การพาณิชย์อิเล็กทรอนิกส์มีความสมบูรณ์ ทั้งในแง่การสร้างปฏิสัมพันธ์ทางธุรกิจแบบผู้ประกอบการต่อผู้ประกอบการ (Business-to-Business, B2B) หรือในการช่วยปฏิสัมพันธ์ทางธุรกิจแบบผู้บริโภคกับผู้บริโภค (Consumer-to-Consumer, C2C) ให้มีความสะดวกและง่ายดายขึ้น การศึกษาและวิจัยแนวคิดการเจรจาต่อรองแบบอิเล็กทรอนิกส์เป็นการผสมผสานหลักการของหลากหลายสาขาวิชา เช่น เศรษฐศาสตร์ สังคมศาสตร์ เทคโนโลยีสารสนเทศ เทคโนโลยีการสื่อสาร และวิทยาการคอมพิวเตอร์ เป็นต้น เราสามารถสรุปเนื้อหาและใจความสำคัญได้ดังนี้

2.2.2.1 ลักษณะและรูปแบบของการเจรจาต่อรองแบบอิเล็กทรอนิกส์

การเจรจาต่อรองแบบอิเล็กทรอนิกส์สามารถแบ่งประเภทตามลักษณะความเป็นอัตโนมัติการเจรจาต่อรองได้เป็น 3 รูปแบบ (Benyoucef and Rinderle, 2006) ได้แก่

1. การเจรจาต่อรองแบบอิเล็กทรอนิกส์แบบไม่อัตโนมัติ (Manual e-negotiation) เป็นการเจรจาต่อรองแบบอิเล็กทรอนิกส์ที่ผู้มีส่วนเกี่ยวข้องทั้งหมดเป็นคน ซึ่งดำเนินการเจรจาต่อรองผ่านทางระบบที่จัดสรรให้ไว้
2. การเจรจาต่อรองแบบอิเล็กทรอนิกส์แบบกึ่งอัตโนมัติ (Semi-automated e-negotiation) เป็นการเจรจาต่อรองแบบอิเล็กทรอนิกส์ที่ผู้มีส่วนเกี่ยวข้องฝ่ายหนึ่งเป็นคนและอีกฝ่ายเป็นตัวแทนที่เป็นโปรแกรมคอมพิวเตอร์ ซึ่งส่วนที่เป็นโปรแกรมคอมพิวเตอร์นี้จะต้องมีความสามารถที่วิเคราะห์หรือพิจารณาแทนคนได้
3. การเจรจาต่อรองแบบอิเล็กทรอนิกส์แบบอัตโนมัติ (Fully automated e-negotiation) เป็นการเจรจาต่อรองแบบอิเล็กทรอนิกส์ที่มีส่วนเกี่ยวข้อง

ทั้งหมดเป็นตัวแทนที่เป็นโปรแกรมคอมพิวเตอร์ ซึ่งสามารถดำเนินการเจรจาต่อรองแทนบุคคลต่างๆได้อย่างอัตโนมัติ

แต่อย่างไรก็ตามในปัจจุบันได้มีการศึกษาและวิจัยการเจรจาต่อรองแบบอิเล็กทรอนิกส์แบบอัตโนมัติกันอย่างแพร่หลาย โดยสาเหตุหลักอันเนื่องมาจาก

- การเจรจาต่อรองแบบอิเล็กทรอนิกส์แบบไม่อัตโนมัติใช้เวลามากเพราะบุคคลที่เกี่ยวข้องจะต้องติดต่อสื่อสารกันไปจนกว่าทุกฝ่ายจะพึงพอใจ และมีต้นทุนที่สูงเพราะต้องสร้างระบบที่คอยสนับสนุนให้การเจรจาต่อรองทั้งกระบวนการสำเร็จลุล่วง
- การเจรจาต่อรองแบบอิเล็กทรอนิกส์ทั้งแบบไม่อัตโนมัติและแบบกึ่งอัตโนมัติเป็นการติดต่อระหว่างผู้มีส่วนได้ส่วนเสียที่อย่างน้อยหนึ่งฝ่ายจะเป็นมนุษย์ ดังนั้นปัจจัยความรู้สึกรของมนุษย์จะเข้าเป็นส่วนหนึ่งในการพิจารณาด้วย ซึ่งอาจก่อให้เกิดปัญหาที่ตามมาได้มากมาย เพื่อการหลีกเลี่ยงปัญหาเหล่านี้การเจรจาต่อรองแบบอิเล็กทรอนิกส์ต้องเป็นระบบที่ทุกฝ่ายเป็นตัวแทนที่เป็นโปรแกรมคอมพิวเตอร์ นั่นก็คือการเจรจาต่อรองแบบอิเล็กทรอนิกส์แบบอัตโนมัติ นั่นเอง
- การเจรจาต่อรองแบบอิเล็กทรอนิกส์แบบอัตโนมัติมีผู้มีส่วนเกี่ยวข้องทั้งหมดคือตัวแทนที่เป็นโปรแกรมคอมพิวเตอร์ ดังนั้นถ้ามีฝ่ายหนึ่งเพิ่มขึ้นหรืออีกฝ่ายหนึ่งถอนออกไปก็สามารถดำเนินการได้อย่างอิสระ หรือแม้กระทั่งสามารถเพิ่มหรือลดประเด็นสำหรับเจรจาต่อรองก็กระทำได้ง่ายเช่นเดียวกัน

รูปแบบพื้นฐานที่สุดของการเจรจาต่อรองแบบอิเล็กทรอนิกส์คือการกำหนดราคาขายแบบคงที่ (Fixed price) กล่าวคือผู้ขายเสนอราคาของสินค้าหรือบริการที่ราคาขายเจาะจงเอาไว้บนสื่ออิเล็กทรอนิกส์ ส่วนผู้ซื้อสามารถตัดสินใจได้เพียงแต่ซื้อหรือไม่ซื้อตามราคาที่แสดงไว้เท่านั้น

รูปแบบที่สองคือการประมูล (Auction) ซึ่งมีกระบวนการที่ซับซ้อนขึ้นมาเล็กน้อยและเป็นรูปแบบที่พบเห็นได้มากสำหรับงานทางด้านการเจรจาต่อรองแบบอิเล็กทรอนิกส์ โดยที่ผู้ต้องการซื้อทั้งหมดเสนอราคาที่ตนเองต้องการผ่านสื่ออิเล็กทรอนิกส์ จากนั้นก็ประมูล แล้วผู้ขายจะขายสินค้าหรือบริการนั้นๆให้กับผู้ที่เสนอราคามากที่สุด ซึ่งในความเป็นจริงแล้วสามารถแบ่งกลุ่มรูปแบบการประมูลได้หลายรูปแบบ (จามิกร, 2550)

ตัวอย่างเช่น (1) English auction เป็นการประมูลที่เสนอราคาขั้นต่ำเอาไว้ แล้วประมูลจนกว่ามีผู้เสนอราคาสูงสุดและไม่มีผู้ขายเสนอราคาสูงกว่าจึงทำการขาย (2) Dutch auction เป็นการประมูลที่เสนอราคาสูงสุดแล้วให้ผู้เข้าร่วมประมูลเสนอราคาลดต่ำลงมาเรื่อยๆ จนมีผู้พอใจราคา นั้นจึงขาย เป็นต้น

รูปแบบที่สามคือการเจรจาต่อรองซึ่งมีความสลับซับซ้อนค่อนข้างสูง การเจรจาต่อรองนี้เกิดจากการสร้างข้อเสนอ (Proposal) และข้อเสนอดตอบกลับ (Counterproposal) เป็นจำนวนมากในระหว่างการดำเนินการจนบรรลุข้อตกลง (Sandholm, 1999) โดยอาจจะมีการเกี่ยวข้องกับระบบเพียงสองฝ่ายหรือมากกว่านั้น

2.2.2.2 ประเภทและองค์ประกอบของการเจรจาต่อรองแบบอิเล็กทรอนิกส์

การเจรจาต่อรองแบบอิเล็กทรอนิกส์จะไม่สามารถดำเนินการได้ ถ้าปราศจากเทคโนโลยีที่คอยสนับสนุนให้ผู้เกี่ยวข้องดำเนินการเจรจาต่อรอง ดังนั้นสามารถแบ่งการเจรจาต่อรองแบบอิเล็กทรอนิกส์ตามลักษณะของระบบสารสนเทศที่นำมาใช้ในการพัฒนาได้ 3 ประเภทดังนี้ (Bichler et al., 2003)

1. ระบบสนับสนุนการตัดสินใจสำหรับการเจรจาต่อรอง (Negotiation decision support system)

การเจรจาต่อรองแบบอิเล็กทรอนิกส์ลักษณะนี้เป็นตัวช่วยสนับสนุนให้ผู้ดำเนินการเจรจาต่อรองสามารถตัดสินใจได้ดีขึ้น โดยทั่วไปแล้วจะประกอบด้วยสองส่วนสำคัญ (Lo, 1999) คือส่วนประกอบสนับสนุนการตัดสินใจและส่วนประกอบสนับสนุนการติดต่อสื่อสาร ส่วนประกอบสนับสนุนการตัดสินใจช่วยเพิ่มประสิทธิภาพความสามารถในการพิจารณาข้อมูลต่างที่จำเป็นสำหรับการเจรจาต่อรองของผู้ดำเนินการเจรจาต่อรอง หรือเป็นระบบสนับสนุนการตัดสินใจ (Decision support system) นั้นเอง ส่วนประกอบสนับสนุนการติดต่อสื่อสารช่วยทำให้ระบบเกิดการแลกเปลี่ยนข้อมูลข้อเสนอและข้อโต้แย้งได้สะดวกขึ้น ส่งผลทำให้ลดระยะเวลาในการดำเนินการและเพิ่มความพึงพอใจในผลลัพธ์ที่ได้มากขึ้น

เมื่อพิจารณาถึงระดับของการสนับสนุนกระบวนการเจรจาต่อรองของระบบสนับสนุนดังกล่าว สามารถแบ่งย่อยได้อีก 3 รูปแบบ (Al-Sakran and Serguievskaja, 2006) ได้แก่ ระดับที่ใช้ระบบช่วยสนับสนุนกระบวนการให้สามารถดำเนินการได้โดยง่าย, ระดับที่ใช้ระบบสร้างทางเลือกต่างๆที่อาจเป็นไปได้มาเสนอแก่ผู้ดำเนินการเจรจาต่อรอง และระดับที่

พยายามใช้ตัวแทนที่เป็นโปรแกรมคอมพิวเตอร์ในการดำเนินการทั้งกระบวนการเจรจาต่อรองที่เกิดขึ้น

2. เอเจนต์การเจรจาต่อรอง (Negotiation software agent)

การเจรจาต่อรองแบบอิเล็กทรอนิกส์ลักษณะนี้เป็นการนำเทคโนโลยีเอเจนต์ซึ่งเป็นโปรแกรมคอมพิวเตอร์มาทำหน้าที่ในการเจรจาต่อรองแทนมนุษย์ ทั้งในด้านการตัดสินใจ การติดต่อสื่อสาร และการดำเนินกิจกรรมที่เกิดขึ้น

3. สื่อกลางสำหรับการเจรจาต่อรองแบบอิเล็กทรอนิกส์ (Electronic negotiation media)

การเจรจาต่อรองแบบอิเล็กทรอนิกส์ลักษณะนี้เป็นระบบสารสนเทศที่พัฒนาช่องทางในการสื่อสารซึ่งช่วยดำเนินการและขนส่งข้อมูลระหว่างผู้มีส่วนร่วมในการเจรจาต่อรอง หรือเป็นแพลตฟอร์มที่ทำให้เอเจนต์สามารถดำเนินการร่วมกันได้ ระบบนี้จะต้องพัฒนาตามข้อกำหนดหรือกฎในการติดต่อสื่อสารของการเจรจาต่อรอง

สำหรับองค์ประกอบที่จำเป็นสำหรับการดำเนินการเจรจาต่อรองได้แก่ (Dinitto et al., 2007)

- เป้าหมายของการเจรจาต่อรอง (Negotiation objective) เป็นกลุ่มของปัจจัยต่างๆที่ต้องการให้ได้ข้อตกลงร่วมกัน สิ่งเหล่านี้อาจจะเป็นได้ทั้งราคา คุณภาพ ระยะเวลา หรือเงื่อนไขเป็นต้น
- ขั้นตอนการทำงานของการทำงานของการเจรจาต่อรอง (Negotiation workflow) เป็นกลุ่มของขั้นตอนสำหรับการดำเนินการเจรจาต่อรอง ซึ่งจะมีลักษณะเป็นอย่างไรจะขึ้นอยู่กับรูปแบบการเจรจาต่อรองดังที่ได้กล่าวมาในข้างต้น
- โพรโตคอลสำหรับการเจรจาต่อรอง (Negotiation protocol) เป็นกลุ่มของเงื่อนไขในการตรวจสอบความถูกต้องของข้อมูลที่ปรากฏทั้งหมด ทั้งจากผู้มีส่วนร่วมและในกระบวนการเจรจาต่อรอง เช่น ในระบบการประมูลแบบอังกฤษ (English auction) โพรโตคอลสำหรับการเจรจาต่อรองจะต้องมีการกำหนดให้รับแต่ข้อเสนอที่เกี่ยวกับการเพิ่มค่าของเป้าหมายของการเจรจาต่อรองขึ้นเท่านั้น
- แบบจำลองการตัดสินใจสำหรับเอเจนต์ (Agent decision model) เป็นเครื่องมือที่ช่วยในการตัดสินใจของผู้มีส่วนร่วม โดยเป็นไปตามโพรโตคอลสำหรับการเจรจาต่อรองที่กำหนด เช่น ฟังก์ชันสำหรับการประเมินข้อเสนอ เป้าหมายที่จะไปให้ถึง ช่วงค่าของเป้าหมายของการ

เจรจาต่อรองที่ยอมรับได้ อัลกอริทึมที่ใช้ในการตัดสินใจว่าจะเสนอต่อไปหรือไม่ เป็นต้น

2.2.2.3 เทคโนโลยีที่ใช้ในการพัฒนาการเจรจาต่อรองแบบอิเล็กทรอนิกส์

(1) เทคโนโลยีเอเจนต์ (Agent Technology)

เอเจนต์ (จามิกร, 2550) เป็นโปรแกรมคอมพิวเตอร์ซึ่งทำงานอัตโนมัติ แทนคนหรือองค์กร โดยแต่ละเอเจนต์มีส่วนร่วมการประมวลผลที่สามารถเริ่มทำงานด้วยตนเอง และสามารถทำงานร่วมกับเอเจนต์อื่นๆได้ ซึ่งมีคุณสมบัติพื้นฐานดังนี้ (Murugesan, 2000)

- การทำงานด้วยตนเอง (Autonomy)

เอเจนต์สามารถดำเนินการได้อย่างอัตโนมัติ โดยไม่จำเป็นต้องมีมนุษย์เป็นผู้สั่งดำเนินการ มีกลไกบางอย่างในการควบคุมการกระทำและสถานะของเอเจนต์

- การทำงานในเชิงรุก (Proactive)

เอเจนต์มีความสามารถในการริเริ่มทำงานได้เองมากกว่าที่จะคอยทำงานตอบสนองการกระตุ้นจากสิ่งแวดล้อม

- การทำงานร่วมกัน (Cooperation)

เอเจนต์มีความสามารถในการดำเนินการบางอย่างร่วมกับเอเจนต์อื่นๆ

- การเรียนรู้ (Learning)

สำหรับระบบเอเจนต์ที่มีความฉลาด จะหมายความว่าเอเจนต์นั้นๆมีความสามารถในการเรียนรู้จากการมีปฏิสัมพันธ์กับสิ่งแวดล้อมหรือจากการติดต่อสื่อสารกับมนุษย์

ทั้งนี้สามารถแบ่งประเภทของระบบเอเจนต์ได้ 2 ลักษณะ (จามิกร, 2550)

ได้แก่ (1) เอเจนต์แบบอยู่กับที่ (Stationary agent) คือเอเจนต์ที่สามารถทำการประมวลผลได้เฉพาะบนระบบที่สร้างเอเจนต์นั้น (2) เอเจนต์แบบเคลื่อนที่ (Mobile agent) คือเอเจนต์ที่ไม่ผูกติดกับระบบที่เอเจนต์เริ่มดำเนินการประมวลผล แต่สามารถเคลื่อนย้ายตัวเองไปยังระบบอื่นๆภายในเครือข่ายได้

เมื่อเอเจนต์ถูกพัฒนาและนำไปประยุกต์ใช้งานในอุตสาหกรรม มีความจำเป็นต้องควบคุมให้เทคโนโลยีเอเจนต์เป็นไปในทางเดียวกัน ดังนั้นจึงมีหน่วยงานขึ้นมากำกับดูแล มาตรฐาน ได้แก่ Foundation for Intelligent Physical Agent (FIPA) และกลุ่ม Agent Working Group ขององค์กร Object Management Group (OMG) โดย FIPA ได้ตั้ง

มาตรฐานขึ้นมามุ่งเน้นไปยังระดับเอเจนต์ ส่วน OMG สร้างมาตรฐานเพื่อเน้นไปยังระดับการสั่งการและการสื่อสาร ทั้งนี้ทั้งสองมาตรฐานได้กำหนดรูปแบบกรอบแนวคิด รูปแบบการสื่อสาร และวิธีการดำเนินการของเอเจนต์ที่แตกต่างกันออกไป ขึ้นอยู่กับว่าผู้นำไปประยุกต์ใช้จะเลือกให้เหมาะสมกับการดำเนินงานของตน

สำหรับในระบบการเจรจาต่อรองแบบอิเล็กทรอนิกส์นั้น เอเจนต์จะถูกนำมาใช้พัฒนาในการเจรจาต่อรองแบบอัตโนมัติ ซึ่งเอเจนต์จะเป็นตัวแทนของแต่ละฝ่ายในสถานการณ์การเจรจาต่อรองนั้นๆ

(2) เทคโนโลยีออนโทโลยี (Ontology Technology)

ออนโทโลยีเป็นแนวคิดที่ปรากฏอยู่ในหลากหลายสาขาวิชา ซึ่งในแต่ละสาขาวิชา也将มีความหมายแตกต่างกันออกไป ในสาขาคอมพิวเตอร์เองก็มีการให้คำจำกัดความไว้หลากหลาย แต่คำจำกัดความหนึ่งที่เป็นที่ยอมรับกันแพร่หลายคือ ออนโทโลยีเป็นแบบแผนหรือข้อกำหนดเกี่ยวกับแนวคิดที่ใช้ร่วมกัน ("Ontology is a formal, explicit specification of a shared conceptualization") (Studer, 1998) กล่าวคือออนโทโลยีเป็นการพยายามที่จะบรรยายแนวคิดของขอบเขตความสนใจหรือโดเมนให้อยู่ในรูปสิ่งต่างๆและความสัมพันธ์ระหว่างสิ่งเหล่านั้น ซึ่งสามารถแสดงออกมาอยู่ในรูปของระบบสัญลักษณ์ อาจมีรายละเอียดดังนี้

คลาส (Class) ใช้แทนแนวคิดของกลุ่มที่สนใจ เช่น ในโดเมนการท่องเที่ยว คลาสคือที่พัก สถานที่และวิธีการเดินทาง และคลาสนี้ยังมีคุณสมบัติของการสืบทอด (Inheritance) เช่น เมืองและหมู่บ้านสืบทอดคุณสมบัติมาจากคลาสนสถานที่

อินสแตนซ์ (Instance) คือค่าของสิ่งต่างๆในคลาสของออนโทโลยี เช่น

กรุงเทพฯ หาดใหญ่ หรือภูเก็ตเป็นอินสแตนซ์ของเมือง

ความสัมพันธ์ (Relationship) ใช้แทนการเชื่อมโยงกันระหว่างแนวคิดในขอบเขตความสนใจนั้นๆ ซึ่งอยู่ในรูปความสัมพันธ์จากโดเมนไปยังเรนจ์ เช่น จากประโยค "รถยนต์ทะเบียน ข2552 เดินทางมาถึงหาดใหญ่" มีความสัมพันธ์คือ "เดินทางมาถึง" โดยมีค่าในโดเมนคือ "รถยนต์ทะเบียน ข2552" และค่าในเรนจ์คือ "หาดใหญ่"

ทั้งนี้ระบบสัญลักษณ์จะช่วยให้นักุษย์สามารถเข้าใจและนำไปใช้ประโยชน์ได้เหมาะสม โดยเฉพาะอย่างยิ่งการแทนและการค้นหาสิ่งที่ต้องการตามความหมายของสิ่งนั้น โดยการสร้างแนวคิดออนโทโลยีในการอธิบายขอบข่ายที่สนใจ แล้วกำหนดอินสแตนซ์ของแต่ละ

แนวคิด เมื่อเราต้องการค้นหาสิ่งต่างๆ เราก็สามารถค้นหาได้จากโดเมน เรนจ์และความสัมพันธ์ของมัน ซึ่งทำให้ผลลัพธ์ที่ได้เป็นไปตามความหมายที่ควรจะเป็น

2.3 การเจรจาต่อรองสำหรับแนวคิดสถาปัตยกรรมเชิงบริการ

2.3.1 บทบาทการเจรจาต่อรองในสถาปัตยกรรมเชิงบริการ

เมื่อนำแนวคิดสถาปัตยกรรมเชิงบริการมาประยุกต์ใช้ในโลกทางธุรกิจ การคำนึงถึงมุมมองในแง่การดำเนินการเรียกใช้บริการตามองค์ประกอบของสถาปัตยกรรมเชิงบริการดังที่ได้กล่าวมาคงไม่เพียงพอ อาจจำเป็นต้องขยายมุมมองลงไปรายละเอียดปลีกย่อยเพื่อความสอดคล้องกับกระบวนการที่เกิดขึ้นในความเป็นจริง

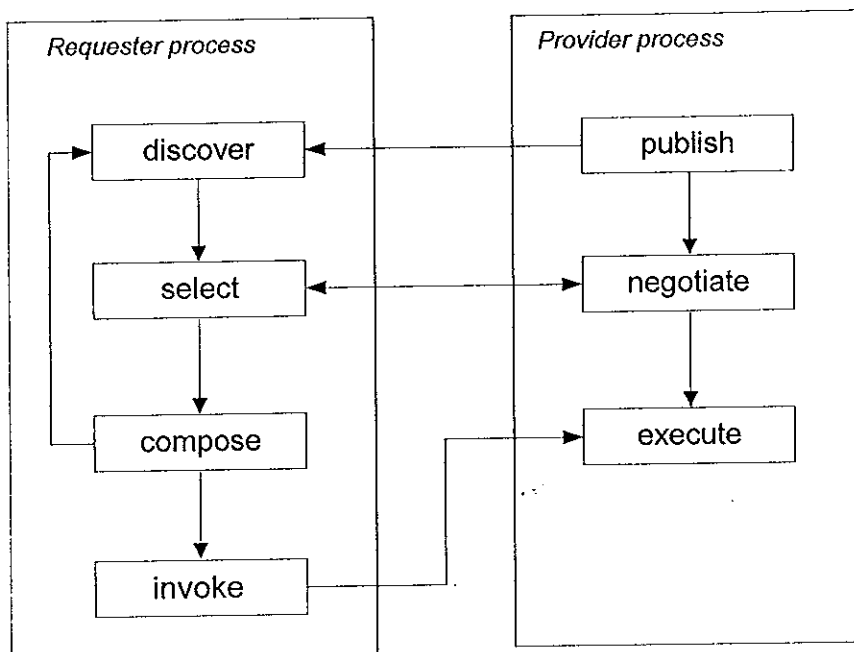
Jos de Bruijn และคณะ (2008) ได้พิจารณาถึงกระบวนการที่ควรเกิดขึ้นก่อนการเรียกใช้บริการ ซึ่งมีรายละเอียดในบทบาทของผู้เรียกใช้บริการดังนี้

1. ค้นหาบริการที่มีการทำงานตามความต้องการของตน
2. คัดเลือกบริการที่ดีที่สุดจากกลุ่มผลลัพธ์ที่ได้จากการค้นหาตามความชื่นชอบ และเจรจาต่อรองการใช้งานบริการกับผู้ให้บริการรายนั้นๆ
3. ถ้าจำเป็นต้องเลือกบริการหลายตัวมาใช้งานประกอบกัน จะต้องพิจารณาถึงลำดับการดำเนินงานด้วย

ส่วนบทบาทของผู้ให้บริการเริ่มต้นจากดำเนินการประกาศรายละเอียดของบริการไว้ เพื่อให้ลูกค้าเข้ามาคัดเลือกได้ จากนั้นรับการเจรจาต่อรองการใช้งานบริการจากผู้เรียกใช้บริการ เมื่อเรียบบร้อยก็ดำเนินการบริการเมื่อมีการเรียกใช้เข้ามา การดำเนินการทั้งหมดสามารถแสดงได้ในภาพประกอบที่ 2.14 และมีรายละเอียดประกอบโดยสรุปดังนี้

(1) การประกาศบริการ (Publication)

ผู้ให้บริการจำเป็นต้องประกาศรายละเอียดของบริการของตน เพื่อให้ผู้เรียกใช้บริการสามารถเข้ามาค้นหาและคัดเลือกได้อย่างเหมาะสม สถานที่ที่สำหรับประกาศบริการได้แก่ คลังเก็บบริการ (Service repository) เช่น Universal Description Discovery Integration (UDDI) (OASIS, 2009) ซึ่งจะต้องมีหน่วยบริการการค้นหาประกอบอยู่ด้วย รายละเอียดบริการที่ประกาศนั้นจะต้องอยู่ในรูปแบบที่ผู้เรียกใช้บริการสามารถทำความเข้าใจได้



ภาพประกอบที่ 2.14 กระบวนการเรียกใช้งานบริการ (ที่มา: Bruijn et al., 2008)

(2) การค้นหา (Discovery)

ผู้เรียกใช้บริการจะกำหนดความต้องการบริการของตน และสอบถามไปยังคลังเก็บบริการเพื่อให้ค้นหาบริการที่มีอยู่และสอดคล้องกับความต้องการ ดังนั้นคำสอบถามดังกล่าวจะประกอบด้วยรายละเอียดความต้องการของผู้เรียกใช้ ซึ่งอยู่ในรูปแบบที่คลังเก็บบริการสามารถจับคู่ความต้องการกับรายละเอียดบริการที่มีอยู่ได้

ทั้งนี้ถ้าต้องการให้เกิดการค้นหาอย่างอัตโนมัติ ทั้งรายละเอียดบริการและความต้องการของผู้เรียกใช้บริการจะต้องถูกปรับเปลี่ยนให้อยู่ในรูปแบบภาษาที่เหมือนหรือมีความสัมพันธ์กันและเครื่องสามารถนำไปดำเนินงานได้อย่างราบรื่น

(3) การคัดเลือกและการเจรจาต่อรอง (Selection and Negotiation)

ในขั้นตอนการค้นหาบริการตามความต้องการของผู้เรียกใช้นั้น ผลลัพธ์ที่ได้ อาจมีจำนวนมากมาย ดังนั้นจึงจำเป็นที่ผู้เรียกใช้จะต้องเลือกบริการจากรายการผลลัพธ์นั้นเพียงหนึ่งเดียว และดำเนินการเจรจาต่อรองการใช้งานของบริการที่เลือกกับผู้ให้บริการของบริการนั้น

การคัดเลือกบริการจากกลุ่มผลลัพธ์นี้จะคัดเลือกจากการจับคู่ความชอบของผู้เรียกใช้บริการกับรายละเอียดของบริการที่ไม่เกี่ยวข้องกับการดำเนินการ (Nonfunctional description) เช่น คุณภาพของบริการ (Quality of service, QoS) ราคา เป็นต้น ดังนั้นผู้เรียกใช้บริการจะต้องระบุความขึ้นชอบต้องการของตนและผู้ให้บริการต้องประกาศรายละเอียดของบริการที่ไม่เกี่ยวข้องกับการดำเนินการไว้ด้วย

(4) การประกอบบริการ (Composition)

ในแต่ละบริการอาจจะมีการดำเนินการเป็นเพียงแค่บางส่วนของความต้องการในระบบงานประยุกต์ของผู้เรียกใช้ ดังนั้นในกรณีนี้จะต้องนำบริการต่างๆมาประกอบกันเพื่อให้ได้ผลลัพธ์ตามจุดมุ่งหมายตามต้องการของระบบงานประยุกต์นั้นๆ อย่างไรก็ตามในการประกอบบริการนี้จะต้องคำนึงถึงประเด็นการจัดลำดับการทำงานของแต่ละบริการให้เหมาะสมไม่เกิดความขัดแย้งของการทำงานในระบบ เพื่อหลีกเลี่ยงผลลัพธ์จากการทำงานที่ผิดพลาด ในขั้นตอนนี้อาจจำเป็นต้องมีการค้นหา คัดเลือก และเจรจาต่อรองบริการใหม่ๆเพิ่มเติมจนกว่าได้ผลลัพธ์ตามที่คาดหวัง

(5) การเรียกใช้และการดำเนินการ (Invocation and Execution)

หลังจากการค้นหา การคัดเลือก การเจรจาต่อรองและการประกอบบริการเป็นที่เรียบร้อยแล้ว ผู้ให้บริการก็ดำเนินการบริการนั้นๆ และพร้อมสำหรับให้ผู้เรียกใช้บริการส่งข่าวสารมาติดต่อเพื่อขอรับบริการ ซึ่งเป็นไปตามข้อกำหนดและข้อตกลงที่ได้ดำเนินการเจรจาต่อรองไว้ในข้างต้นนั่นเอง

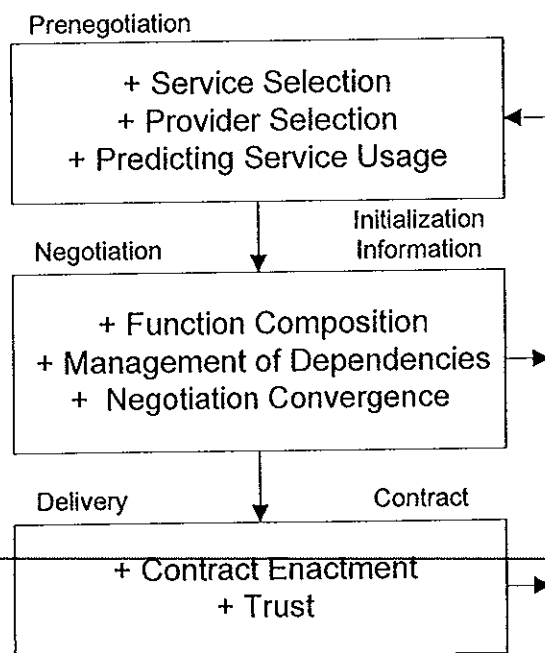
ในปัจจุบันมีงานวิจัยจำนวนหนึ่งที่ยุ่งประเด็นความสนใจไปยังการศึกษาแนวคิดการเจรจาต่อรองสำหรับสถาปัตยกรรมเชิงบริการ ซึ่งจะได้อธิบายในลำดับถัดไป

2.3.2 ตัวอย่างกรอบแนวคิดการเจรจาต่อรองสำหรับสถาปัตยกรรมเชิงบริการ

งานวิจัยของ Elfataty และ Layzell (2004) ได้เสนอกรอบแนวคิดการเจรจาต่อรองของระบบซอฟต์แวร์เชิงบริการ โดยพิจารณาถึงคุณลักษณะของตลาดของซอฟต์แวร์เชิงบริการ และคุณลักษณะของการเจรจาต่อรองในสิ่งแวดล้อมนั้นๆ เพื่อนำมาพิจารณาถึงกรอบแนวคิดหลักที่เหมาะสม ซึ่งสามารถแบ่งออกได้เป็น 3 ระยะดังในภาพประกอบที่ 2.15

(1) ระยะเวลาการเจรจาต่อรอง (Prenegotiation)

เป็นระยะเตรียมข้อมูลต่างๆที่จำเป็นให้พร้อมก่อนดำเนินการเจรจาต่อรอง แบ่งย่อยได้ 3 ขั้นตอน ได้แก่ การคัดเลือกบริการ (Service selection) เนื่องด้วยในตลาดซอฟต์แวร์เชิงบริการมีบริการที่ถูกนำเสนออยู่หลากหลายจึงจำเป็นต้องมีวิธีการคัดเลือกบริการให้เหมาะสมกับความต้องการของลูกค้าตามคุณสมบัติของบริการที่ได้ระบุไว้ การคัดเลือกผู้ให้บริการ (Provider selection) ต้องมีการคัดเลือกให้เหมาะสมเช่นเดียวกับการคัดเลือกบริการ โดยอาจจะค้นหาโดยวัดจากข้อมูลตอบกลับ (Feedback) ที่ได้จากระยะถัดไป และการทำนายการใช้บริการ (Predicting service usage) ในการส่งข่าวสารของแต่ละฝ่าย แต่ครั้งอาจมีการส่งข่าวสารติดต่อกันในปริมาณมากกว่าจะบรรลุข้อตกลง ซึ่งจะใช้ทรัพยากรของระบบสูงขึ้นตามไปด้วย ดังนั้นจำเป็นต้องทำนายแนวทางการใช้บริการเพื่อควบคุมการดำเนินการ



ภาพประกอบที่ 2.15 กรอบแนวคิดการเจรจาต่อรองสำหรับซอฟต์แวร์เชิงบริการ (ที่มา: Elfatary and Layzell, 2004)

(2) ระยะเวลาการเจรจาต่อรอง (Negotiation)

เป็นระยะดำเนินการเจรจาต่อรองโดยใช้ข้อมูลที่เตรียมเอาไว้จนบรรลุข้อตกลง สามารถแบ่งย่อยได้ 3 ขั้นตอน ได้แก่ การประกอบส่วนการทำงาน (Functional

composition) ทำการประกอบบริการที่ค้นหาได้ เพื่อให้ได้ผลการดำเนินการที่เหมาะสม สอดคล้องกับความต้องการที่กำหนด การจัดการภาวะพึ่งพิง (Management of dependencies) จำเป็นต้องคำนึงถึงความพึ่งพิงกันระหว่างบริการ เช่น บริการท่องเที่ยวประกอบด้วยบริการที่ต้องพึ่งพิงกัน ได้แก่ บริการจองตั๋วเครื่องบินและบริการจองที่พักซึ่งต้องมีลำดับการทำงานที่เหมาะสม และขั้นตอนสุดท้ายเป็นการบรรลุเจรจาต่อรอง (Negotiation convergence) ซึ่งต้องดำเนินการให้แล้วเสร็จโดยเร็ว โดยมีกลไกอยู่สองส่วนที่สามารถช่วยให้บรรลุผลได้คือ โพรโตคอลการเจรจาต่อรอง (Negotiation protocol) จะช่วยกำหนดพฤติกรรมของทุกฝ่ายในการเจรจาต่อรองให้อยู่ในเส้นทางที่ถูกต้อง และกลยุทธ์ในการเจรจาต่อรอง (Negotiation strategy) ช่วยให้แต่ละฝ่ายสามารถดำเนินการเจรจาต่อรองได้อยู่บนแนวทางที่เหมาะสมสำหรับตนเอง

(3) ระยะเวลาส่งมอบบริการ (Service Delivery)

เป็นระยะส่งมอบบริการที่เสร็จสมบูรณ์และพร้อมใช้งาน อีกทั้งจัดทำสัญญาการเรียกใช้งานระหว่างผู้ให้บริการและผู้เรียกใช้บริการ แบ่งย่อยได้ 2 ขั้นตอนได้แก่ การร่างสัญญา (Contract enactment) สัญญาอาจจะอธิบายการดำเนินงานของบริการ เงื่อนไข คุณสมบัติ เป็นต้น และความไว้วางใจ (Trust) เป็นสิ่งที่ผู้ให้บริการต้องทำให้เกิดขึ้นเพราะว่า ผู้บริโภคมีโอกาสที่จะเลือกบริการ

2.4 สรุปท้ายบท

สถาปัตยกรรมเชิงบริการเป็นแนวคิดที่สำคัญสำหรับการพัฒนาซอฟต์แวร์ในยุคปัจจุบัน เนื่องจากความยืดหยุ่นในการปรับเปลี่ยนซอฟต์แวร์ให้สามารถตอบสนองต่อโอกาสทางธุรกิจและความต้องการที่เปลี่ยนแปลงไปอย่างรวดเร็ว อย่างไรก็ตามในสภาวะที่มีการแข่งขันกันสูง ประเด็นการเจรจาต่อรองจะเป็นส่วนช่วยสนับสนุนให้ได้ผลลัพธ์ที่สอดคล้องกับความต้องการของทั้งสองฝ่าย ดังนั้นการพัฒนาการเจรจาต่อรองสำหรับสถาปัตยกรรมเชิงบริการจึงเป็นสิ่งสำคัญ โดยเฉพาะอย่างยิ่งในบริบทการเจรจาต่อรองแบบอิเล็กทรอนิกส์ที่จะเป็นแนวทางในการพัฒนาและออกแบบสำหรับงานวิจัยนี้

บทที่ 3

กรอบแนวคิดการเจรจาต่อรองสำหรับสถาปัตยกรรมเชิงบริการ

เนื้อหาในบทนี้กล่าวถึงความครอบคลุมถึงประเด็นการเจรจาต่อรองสำหรับสถาปัตยกรรมเชิงบริการของกลุ่มมาตรฐานการพัฒนาซอฟต์แวร์ที่มีอยู่ในปัจจุบัน และการวิเคราะห์และออกแบบกรอบแนวคิดการเจรจาต่อรองสำหรับสถาปัตยกรรมเชิงบริการ รวมไปถึงกระบวนการเจรจาต่อรองที่สอดคล้องกับกรอบแนวคิดการเจรจาต่อรองดังกล่าว เพื่อนำไปสู่การพัฒนาระบบงานในลำดับถัดไป

3.1 ความครอบคลุมของมาตรฐานการพัฒนาซอฟต์แวร์กับแนวคิดการเจรจาต่อรอง

แนวทางการพัฒนาซอฟต์แวร์สำหรับองค์กรหรือบริษัทโดยทั่วไป มักจะเป็นการจัดหาผลิตภัณฑ์ซอฟต์แวร์ที่สอดคล้องกับความต้องการตามที่กำหนด ซึ่งกระบวนการทั้งหมดเกิดจากการค้นหาและคัดเลือกผู้พัฒนาหรือผู้จัดจำหน่ายซอฟต์แวร์ที่เหมาะสมและทำความตกลงระหว่างกันจนได้ผลิตภัณฑ์ที่ตรงตามความต้องการ ดังนั้นการเจรจาต่อรองจึงเป็นกระบวนการที่สำคัญที่จะช่วยให้การจัดหาซอฟต์แวร์สามารถบรรลุผลสำเร็จ

มาตรฐานสำหรับการพัฒนาซอฟต์แวร์ในปัจจุบันจำนวนหนึ่งก็ได้มีการกล่าวถึงการจัดหา (Acquisition) ผลิตภัณฑ์ซอฟต์แวร์และการได้มา (Supply) ซึ่งผลิตภัณฑ์ซอฟต์แวร์เอาไว้ ดังนั้นการศึกษามาตรฐานเหล่านี้จะสามารถเป็นแนวทางในการวิเคราะห์กระบวนการเจรจาต่อรองสำหรับสถาปัตยกรรมเชิงบริการได้ในระดับหนึ่ง

ผู้ดำเนินการวิจัยได้เลือกศึกษามาตรฐานการพัฒนาซอฟต์แวร์ได้แก่แบบจำลองวุฒิภาวะความสามารถแบบบูรณาการหรือซีเอ็มเอ็มไอ (Capability Maturity Model Integration, CMMI) (CMMI Product Team, 2002; Chrissis et al., 2003) มาตรฐาน IEEE 1062, ฉบับพิมพ์ปี 1998 ข้อเสนอแนะการปฏิบัติสำหรับการจัดหาซอฟต์แวร์ (IEEE Standard 1062, 1998 Edition - Recommend Practice for Software Acquisition) (IEEE, 1998) และ

มาตรฐานสากล ISO/IEC 12207 มาตรฐานสำหรับเทคโนโลยีสารสนเทศ – กระบวนการวัฏจักรชีวิตของซอฟต์แวร์ (International Standard ISO/IEC 12207: Standard for Information Technology - Software life cycle processes) (Singh, 1996) มาดำเนินการศึกษาและวิเคราะห์เพื่อสร้างกระบวนการเจรจาต่อรองสำหรับสถาปัตยกรรมเชิงบริการที่สอดคล้อง โดยมีรายละเอียดดังต่อไปนี้

3.1.1 แบบจำลองวุฒิภาวะความสามารถแบบบูรณาการ (Capability Maturity Model® Integration, CMMI) (CMMI Product Team, 2002; Chrissis et al., 2003)

วุฒิภาวะความสามารถแบบบูรณาการเป็นต้นแบบที่ถูกพัฒนาขึ้นจากสถาบัน Software Engineering Institute (SEI) ของมหาวิทยาลัย Carnegie Mellon ประเทศสหรัฐอเมริกา เพื่อเสนอวิธีปฏิบัติที่ดีที่สุด (Best practice) ที่จะช่วยให้องค์กรสามารถปรับปรุงกระบวนการดำเนินงานให้มีประสิทธิภาพ โดยจะครอบคลุมตั้งแต่กระบวนการออกแบบจนถึงการส่งมอบและการบำรุงรักษาผลิตภัณฑ์

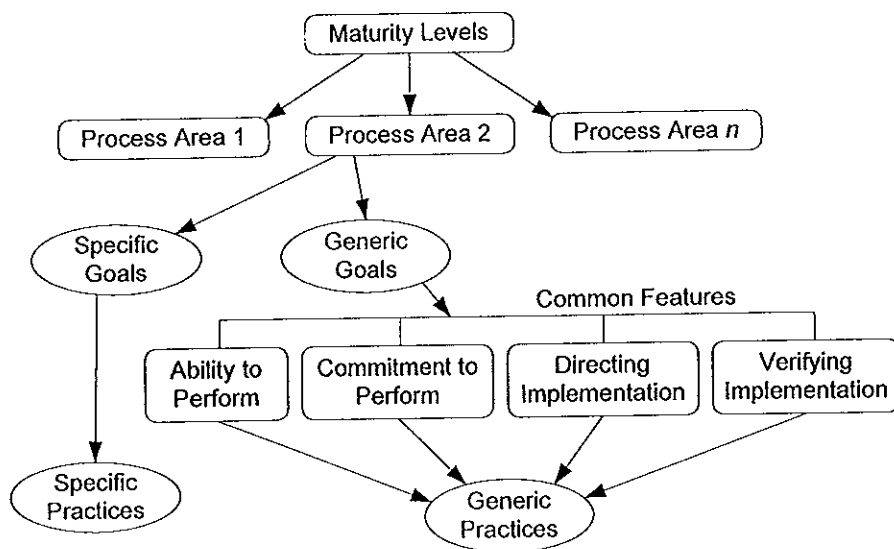
ซีเอ็มเอ็มไอถูกพัฒนาจากการรวมสามแบบจำลองที่มีอยู่เดิมได้แก่

- 1) The Capability Maturity Model for Software (SW-CMM) v2.0 draft C
- 2) The Systems Engineering Capability Model (SECM)
- 3) The Integrated Product Development Capability Maturity Model (IPD-CMM) v0.98

ซีเอ็มเอ็มไอมีองค์ความรู้ครอบคลุมใน 4 ส่วนได้แก่ วิศวกรรมระบบ (System engineering) วิศวกรรมซอฟต์แวร์ (Software engineering) การพัฒนากระบวนการและผลิตภัณฑ์แบบบูรณาการ (Integrated product and process development) และการสรรหาแหล่งตัวแทนจำหน่าย (Supplier sourcing) ซึ่งสามารถแสดงองค์ประกอบของแบบจำลองซีเอ็มเอ็มไอดังแสดงดังภาพประกอบที่ 3.1

แนวปฏิบัติสำหรับการปรับปรุงกระบวนการของซีเอ็มเอ็มไอจะเสนออยู่ในรูปแบบของกลุ่มกระบวนการ (Process area) กล่าวคือกลุ่มของแนวปฏิบัติที่สัมพันธ์กันในด้านใดด้านหนึ่ง ซึ่งเมื่อนำไปปฏิบัติอย่างต่อเนื่องจะสามารถตอบสนองต่อเป้าหมายของการปรับปรุงกระบวนการในด้านนั้นๆ ได้ โดยประกอบด้วยองค์ประกอบสำคัญ 4 ส่วนได้แก่ เป้าหมายที่ใช้งานได้ทั่วไป (Generic goal) จะอธิบายลักษณะที่ต้องปรากฏในการปฏิรูปกระบวนการ แนว

ปฏิบัติที่ใช้งานได้ทั่วไป (Generic practice) บรรยายส่วนสำคัญที่จะทำให้เป้าหมายทั่วไปที่เกี่ยวข้องนั้นสำเร็จ เป้าหมายเฉพาะ (Specific goal) อธิบายคุณลักษณะเฉพาะที่ต้องปรากฏเพื่อสนองกลุ่มกระบวนการนั้น และแนวปฏิบัติเฉพาะ (Specific practice) บรรยายกิจกรรมที่คาดหวังเพื่อให้เกิดผลสำเร็จตามเป้าหมายเฉพาะของกลุ่มกระบวนการนั้น ซีเอ็มเอ็มไอถูกนำเสนอโดยการแทนอยู่ใน 2 รูปแบบคือ แบบต่อเนื่อง (Continuous representation) และแบบขั้นบันได (Staged representation)



ภาพประกอบที่ 3.1 โครงสร้างซีเอ็มเอ็มไอของการแทนรูปแบบขั้นบันได

(ที่มา: CMMI Product Team, 2002)

การแทนแบบต่อเนื่องแสดงให้เห็นถึงลำดับในการปรับปรุงประสิทธิภาพของกระบวนการตามที่ต้องการ เพื่อให้สอดคล้องใกล้เคียงกับวัตถุประสงค์ขององค์กร หรือจุดกระบวนการที่มีความเสี่ยงหรืออาจเกิดปัญหาที่องค์กรสามารถประสพได้ ซึ่งจะประเมินอยู่ในระดับความสามารถ (Capability levels) ของกลุ่มกระบวนการมี 6 ระดับตั้งแต่ 0-5 ได้แก่ Incomplete, Performed, Managed, Defined, Quantitatively Managed และ Optimizing ตามลำดับ

การแทนแบบขั้นบันไดแสดงให้เห็นถึงลำดับในการปรับปรุงการทำงาน ซึ่งองค์กรที่ต้องการปรับปรุงกระบวนการจะต้องดำเนินการจากลำดับขั้นพื้นฐาน และพัฒนาก้าวหน้าขึ้นเรื่อยๆจนถึงระดับที่ต้องการ โดยประเมินอยู่ในระดับวุฒิภาวะ (Maturity Levels) มี 5 ระดับตั้งแต่ 1-5 อันได้แก่ Initial, Managed, Defined, Quantitatively Managed และ

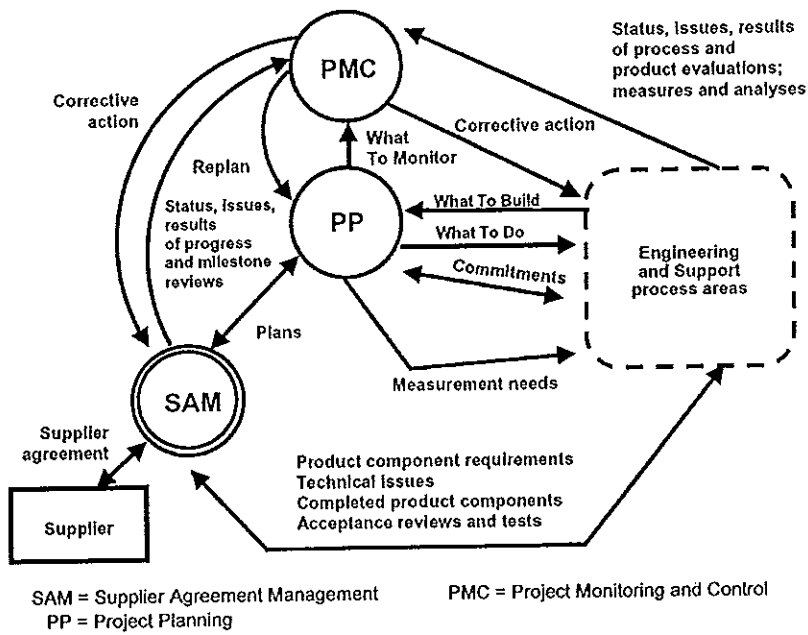
Optimizing ตามลำดับ และเมื่อวิเคราะห์รายละเอียดความรู้ของขอบเขตกระบวนการสำหรับวิศวกรรมระบบและวิศวกรรมซอฟต์แวร์ ซีเอ็มเอ็มไอจะเน้นในประเด็นต่อไปนี้

- Casual Analysis and Resolution
- Configuration Management
- Decision Analysis and Resolution
- Integrated Project Management
- Integrated Supplier Management
- Integrated Teaming
- Measurement and Analysis
- Organizational Environment for Integration
- Organizational Innovation and Deployment
- Organizational Process Definition
- Organizational Process Focus
- Organizational Process Performance
- Organizational Training
- Product Integration
- Project Monitoring and Control
- Project Planning
- Process and Product Quality Assurance
- Quantitative Project Management
- Requirements Development
- Requirements Management
- Risk Management
- Supplier Agreement Management
- Technical Solution
- Validation
- Verification

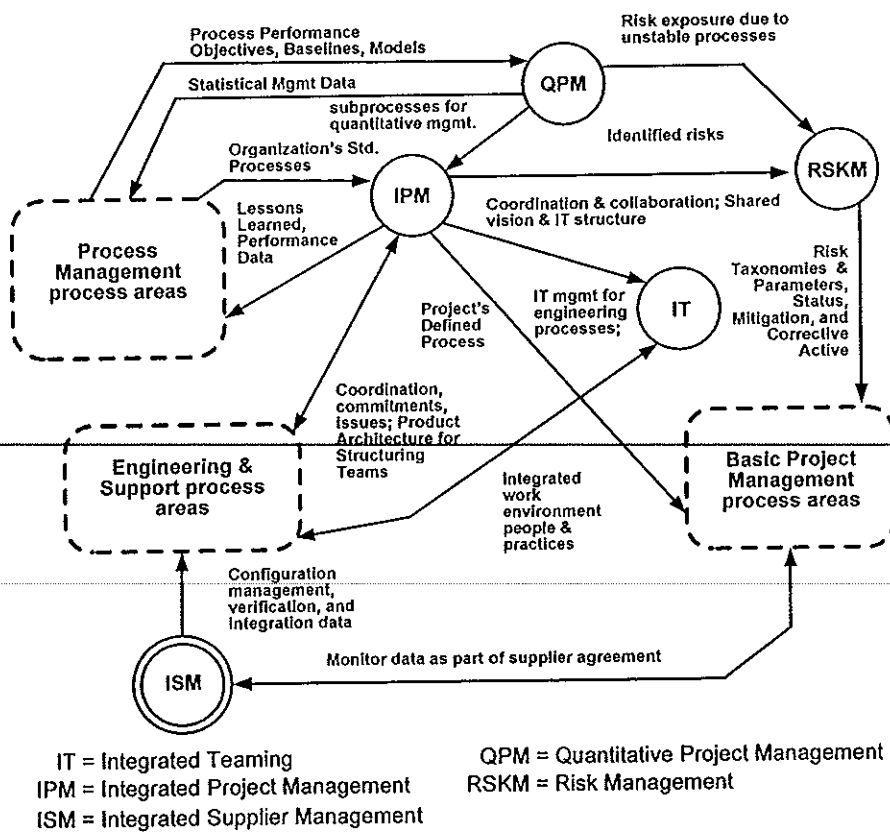
สำหรับขอบเขตกระบวนการสำหรับการพัฒนาผลิตภัณฑ์และกระบวนการแบบบูรณาการได้เพิ่มเติมในประเด็น Integrated Teaming และ Organizational Environment for Integration และขอบเขตกระบวนการสำหรับการสรรหาแหล่งตัวแทนจำหน่ายเพิ่มเติมในประเด็น Integrated Supplier Management

จากรายละเอียดประเด็นต่างๆในข้างต้น เมื่อพิจารณาถึงการเจรจาต่อรองสำหรับสถาปัตยกรรมเชิงบริการจะพบว่าส่วนการจัดการตัวแทนจำหน่ายแบบบูรณาการ (Integrated Supplier Management, ISM) ซึ่งอยู่ในหมวดการจัดการโครงการ (Project Management) ดังแสดงภาพประกอบที่ 3.2 และ 3.3 จะเกี่ยวข้องกับกิจกรรมต่อไปนี้

- 1) การระบุ การวิเคราะห์ และการคัดเลือกแหล่งของผลิตภัณฑ์ที่เป็นไปได้
- 2) การประเมินและการกำหนดแหล่งที่จะใช้ในการจัดหาผลิตภัณฑ์
- 3) การดูแลและการวิเคราะห์กระบวนการของตัวแทนจำหน่ายที่ถูกเลือก
- 4) การประเมินผลิตภัณฑ์งานของตัวแทนจำหน่ายที่ถูกเลือก
- 5) การทบทวนข้อตกลงกับตัวแทนจำหน่ายหรือทบทวนความสัมพันธ์ตามความเหมาะสม



ภาพประกอบที่ 3.2 กลุ่มกระบวนการจัดการโครงการขั้นพื้นฐาน
 (ที่มา: CMMI Product Team, 2002)



ภาพประกอบที่ 3.3 กลุ่มกระบวนการจัดการโครงการแบบก้าวหน้า
 (ที่มา: CMMI Product Team, 2002)

จากการวิเคราะห์ปรากฏว่ายังขาดประเด็นที่เกี่ยวข้องกับการเจรจาต่อรองสำหรับสถาปัตยกรรมเชิงบริการ โดยส่วนที่อ้างอิงถึงตัวแทนจำหน่ายจะต้องไปคำนึงถึงผู้ให้บริการ (Service provider) และในส่วนของผลิตภัณฑ์จะต้องคำนึงถึงตัวบริการ (Service) แทน

ในส่วนของการจัดการข้อตกลงกับตัวแทนจำหน่าย (Supplier agreement management) ซึ่งอยู่ในหมวดการจัดการโครงการ จะเกี่ยวข้องกับกิจกรรมดังต่อไปนี้

- 1) การกำหนดชนิดของการจัดหาที่ใช้สำหรับจัดหาผลิตภัณฑ์
- 2) การคัดเลือกตัวแทนจำหน่าย
- 3) การสร้างและจัดการข้อตกลงกับตัวแทนจำหน่าย
- 4) การปฏิบัติตามข้อตกลงกับตัวแทนจำหน่าย
- 5) การตรวจรับการส่งมอบผลิตภัณฑ์ที่จัดหามา
- 6) การนำผลิตภัณฑ์ที่ได้รับมาใช้กับโครงการต่อไป

จากการวิเคราะห์แล้วก็ไม่ปรากฏความครอบคลุมถึงการเจรจาต่อรองสำหรับสถาปัตยกรรมเชิงบริการ โดยส่วนที่อ้างอิงถึงตัวแทนจำหน่ายจะต้องไปคำนึงถึงผู้ให้บริการ และในส่วนของผลิตภัณฑ์จะต้องคำนึงเป็นตัวบริการด้วยเช่นเดียวกัน

อย่างไรก็ตามเมื่อพิจารณาหมวดทางวิศวกรรมซอฟต์แวร์และระบบแล้ว จะพบว่าสามารถเพิ่มการปฏิบัติในขอบเขตกระบวนการแก้ไขปัญหาเชิงเทคนิค (Technical Solution) ในเทคนิคการออกแบบและการโปรแกรมซอฟต์แวร์โดยอาศัยสถาปัตยกรรมเชิงบริการที่ยังไม่ได้กล่าวถึงได้

3.1.2 มาตรฐาน IEEE 1062, ฉบับพิมพ์ปี 1998 ข้อเสนอแนะการปฏิบัติสำหรับการจัดหาซอฟต์แวร์ (IEEE Standard 1062, 1998 Edition - Recommend Practice for Software Acquisition) (IEEE, 1998)

มาตรฐาน IEEE 1062 เป็นมาตรฐานที่แนะนำแนวปฏิบัติสำหรับการจัดหาผลิตภัณฑ์ซอฟต์แวร์ โดยการอธิบายแนวปฏิบัติที่มีคุณภาพ ที่สามารถเลือกหรือประยุกต์ใช้งานในกระบวนการจัดหาซอฟต์แวร์ โดยมีได้คำนึงถึงขนาด หรือความซับซ้อนของซอฟต์แวร์

การจัดหาซอฟต์แวร์จะอธิบายไปตามช่วงเวลาตั้งแต่เริ่มต้นการตัดสินใจจัดหาผลิตภัณฑ์ซอฟต์แวร์และสิ้นสุดลงเมื่อไม่ใช้งานผลิตภัณฑ์ซอฟต์แวร์นั้นอีกต่อไป โดยสามารถแบ่งออกได้เป็น 5 ระยะย่อย ได้แก่

1) ระยะเวลาวางแผน (Planning phase) ระยะเวลาจะเริ่มต้นจากแนวคิดหรือความจำเป็นในการจัดหาผลิตภัณฑ์ซอฟต์แวร์ สร้างกระบวนการจัดหาผลิตภัณฑ์ซอฟต์แวร์ที่เหมาะสม ระบุข้อกำหนดความต้องการผลิตภัณฑ์ซอฟต์แวร์ จนกระทั่งได้เอกสารแสดงข้อเสนอความต้องการ (Request for proposal)

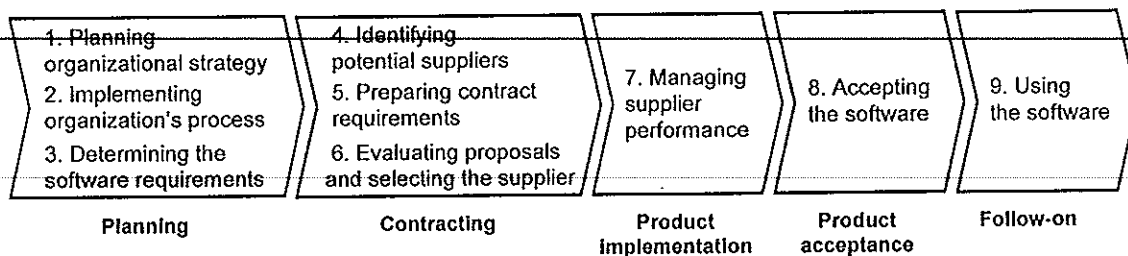
2) ระยะร่างสัญญา (Contracting phase) ประกอบด้วยกิจกรรมที่จำเป็นสำหรับการรับรองว่าผลิตภัณฑ์ซอฟต์แวร์ที่ได้มานั้นตรงกับข้อกำหนดความต้องการที่กำหนดก่อนการดำเนินการเซ็นสัญญา

3) ระยะเวลาพัฒนาผลิตภัณฑ์ (Product implementation phase) เป็นระยะในการพัฒนาผลิตภัณฑ์ซอฟต์แวร์ตามที่ได้รับมอบหมายพร้อมทั้งติดตามเฝ้าดูการดำเนินการพัฒนาของผู้รับจ้างเพื่อให้แน่ใจว่าผู้รับจ้างดำเนินการพัฒนาผลิตภัณฑ์ซอฟต์แวร์ได้เป็นไปตามกำหนดการที่วางเอาไว้

4) ระยะเวลาตรวจรับผลิตภัณฑ์ซอฟต์แวร์ (Product acceptance phase) ระยะเวลาจะเกี่ยวข้องกับกิจกรรมที่จำเป็นสำหรับการประเมิน ทดสอบ และยอมรับผลิตภัณฑ์ซอฟต์แวร์

5) ระยะติดตามผล (Follow-on phase) เป็นระยะที่เกี่ยวข้องกับการนำผลิตภัณฑ์ซอฟต์แวร์ที่ได้มาประยุกต์ใช้งาน และดำเนินการประเมินความพึงพอใจของผู้ใช้งานต่อซอฟต์แวร์ เอกสารกำกับและการสนับสนุนจากผู้รับจ้าง

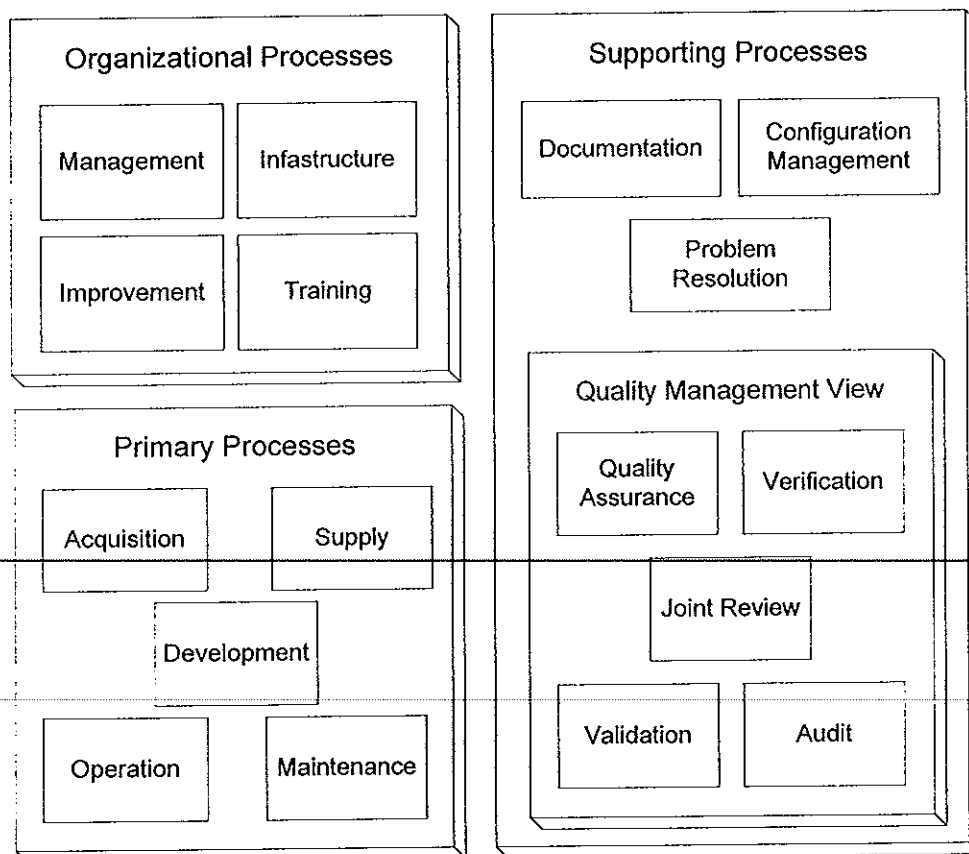
ในแต่ละระยะสามารถแบ่งกระบวนการได้ออกเป็น 9 ขั้นตอนดังแสดงในภาพประกอบที่ 3.4 และจากการตรวจสอบรายละเอียดยังขาดประเด็นที่ครอบคลุมถึงการเจรจาต่อรองสำหรับสถาปัตยกรรมเชิงบริการ



ภาพประกอบที่ 3.4 ขั้นตอนการดำเนินงานของการจัดหาซอฟต์แวร์ตามมาตรฐาน IEEE 1062

3.1.3 มาตรฐานสากล ISO/IEC 12207 มาตรฐานสำหรับเทคโนโลยีสารสนเทศ - กระบวนการวัฏจักรชีวิตของซอฟต์แวร์ (International Standard ISO/IEC 12207: Standard for Information Technology - Software life cycle processes) (Singh, 1996)

มาตรฐาน ISO/IEC เป็นการรวบรวมโครงสร้างแนวทางปฏิบัติโดยทั่วไปสำหรับแต่ละกระบวนการในวัฏจักรชีวิตซอฟต์แวร์ (Software life cycle process) ที่ไม่ได้แสดงรายละเอียดว่าจะต้องทำการปฏิบัติอย่างไรหรือเอกสารจะต้องมีรูปแบบเป็นอย่างไร ในแต่ละกระบวนการประกอบด้วยกลุ่มของกิจกรรม (Activity) ต่างๆ และในแต่ละกิจกรรมจะประกอบด้วยกลุ่มของงาน (Task) ที่ต้องดำเนินการเพื่อให้บรรลุวัตถุประสงค์ โดยจะประกอบด้วยกระบวนการหลักทั้งหมด 3 กลุ่ม ดังแสดงในภาพประกอบที่ 3.5 และมีรายละเอียดโดยสรุปของแต่ละกลุ่มดังนี้



ภาพประกอบที่ 3.5 กระบวนการหลักของมาตรฐานสากล ISO/IEC 12207

(ที่มา: Singh, 1996)

1) กลุ่มกระบวนการขั้นพื้นฐาน (Primary process) เป็นกลุ่มกระบวนการหลักที่ใช้ในการพัฒนาซอฟต์แวร์ให้เกิดขึ้นได้ โดยทั่วไปจะประกอบด้วยกระบวนการย่อยที่ใช้ในการสรรหา จัดหา พัฒนา ปฏิบัติ และบำรุงรักษา

2) กลุ่มกระบวนการวัฏจักรชีวิตด้านการสนับสนุน (Supporting life cycle process) เป็นกลุ่มกระบวนการที่ใช้ในการสนับสนุนกระบวนการต่างๆ เพื่อให้โครงการมีคุณภาพที่ดีและประสบผลสำเร็จ

3) กลุ่มกระบวนการวัฏจักรชีวิตด้านองค์กร (Organizational life cycle process) เป็นกลุ่มกระบวนการที่เกี่ยวข้องกับภาพรวมในระดับองค์กร อาทิเช่น การจัดการโครงการ การพัฒนาบุคคล เป็นต้น

สำหรับในงานวิจัยนี้ได้ให้ความสนใจในกลุ่มกระบวนการขั้นพื้นฐานเป็นสำคัญ ซึ่งจะประกอบด้วยกระบวนการที่สำคัญ 5 กระบวนการคือ การจัดหา (Acquisition) การได้มา (Supply) การพัฒนา (Development) การดำเนินงาน (Operation) และการบำรุงรักษา (Maintenance) ซึ่งในส่วนกระบวนการจัดหามีกิจกรรมและรายละเอียดงานดังนี้

1) การเริ่มต้น (Initiation) เป็นขั้นตอนระบุความต้องการ และความจำเป็นต่างๆของระบบที่จะทำการสรรหา กำหนดความต้องการของทั้งระบบและซอฟต์แวร์ เตรียมแผนการจัดหาและกำหนดกลยุทธ์การตรวจรับ

2) การร้องขอข้อเสนอ (Request for proposal) เป็นขั้นตอนจัดเตรียมข้อมูลต่างๆที่จำเป็นให้แก่ตัวแทนจำหน่าย เช่น เอกสารความต้องการการจัดหา แผนกำหนดเวลาในการเซ็นสัญญา เป็นต้น

3) การจัดเตรียมสัญญา (Contract preparation) เริ่มต้นจากการสร้างวิธีประเมินผู้เสนอโครงการและหลักเกณฑ์ต่างๆที่ต้องใช้ในการประเมิน แล้วประเมินคัดเลือกตัวแทนจำหน่ายโดยใช้วิธีการและหลักเกณฑ์ต่างๆที่ได้กำหนดขึ้นมา สุดท้ายดำเนินการเขียนสัญญา

4) การติดตามดูตัวแทนจำหน่าย (Supplier monitoring) เป็นขั้นตอนการติดตามการดำเนินงานของตัวแทนจำหน่ายทั้งการตรวจดูและตรวจสอบความถูกต้องต่างๆ

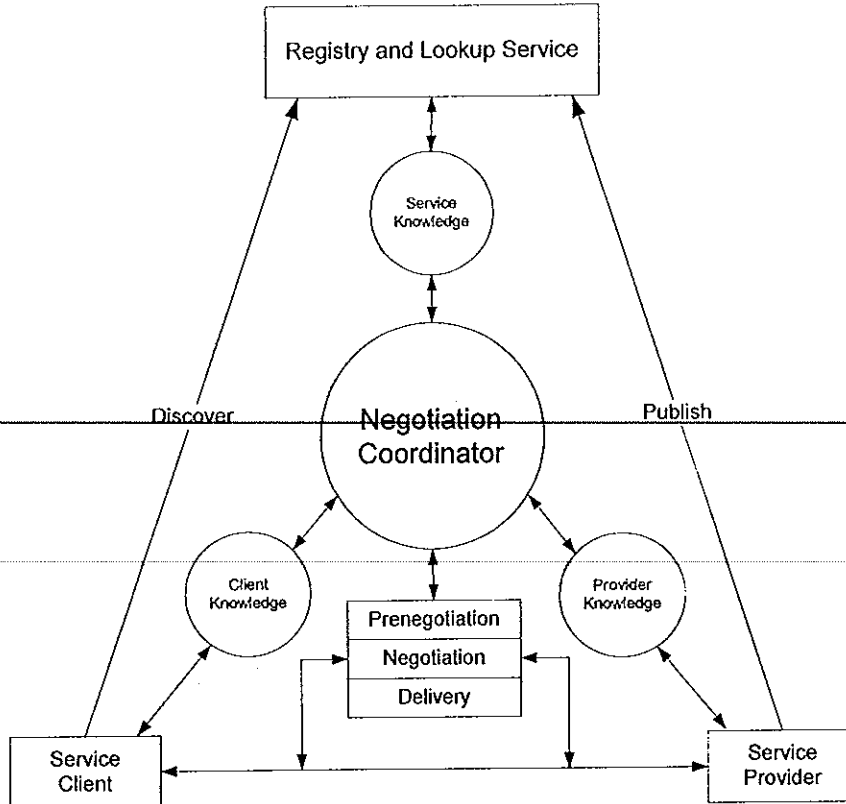
5) การตรวจรับและทำให้สมบูรณ์ (Acceptance and completion) เป็นขั้นตอนสุดท้ายของกระบวนการสรรหาซึ่งเป็นการตรวจรับระบบโดยตรวจรับตามเกณฑ์ต่างๆที่วางไว้ตั้งแต่ขั้นต้น ทบทวนและทดสอบ และปรับแต่งจัดการผลิตภัณฑ์ซอฟต์แวร์ที่ส่งมอบมา

และเมื่อตรวจสอบรายละเอียดของกระบวนการดังกล่าวแล้ว จะพบว่ายังขาดประเด็นที่เกี่ยวข้องกับการเจรจาต่อรองสำหรับซอฟต์แวร์เชิงบริการด้วยเช่นกัน

3.2 กรอบแนวคิดการเจรจาต่อรองสำหรับสถาปัตยกรรมเชิงบริการ

จากการวิเคราะห์ความครอบคลุมของมาตรฐานการพัฒนาซอฟต์แวร์ที่ได้นำเสนอในข้างต้น ผู้วิจัยจึงเสนอกรอบแนวคิดการเจรจาต่อรองที่สนับสนุนการพัฒนาซอฟต์แวร์ตามสถาปัตยกรรมเชิงบริการ โดยอำนวยความสะดวกให้แก่ทั้งผู้ให้บริการและผู้เรียกใช้บริการสามารถเจรจาต่อรองระหว่างกันได้ผ่านเครือข่ายอินเทอร์เน็ต ดังแสดงในภาพประกอบที่ 3.6

ความรู้ความเข้าใจเกี่ยวกับบริการ ผู้ให้บริการและผู้เรียกใช้บริการจะเป็นส่วนสำคัญที่จะทำให้การเจรจาต่อรองสามารถสำเร็จบรรลุผล แต่ในสภาพแวดล้อมเชิงบริการนั้นระบบสามารถเปลี่ยนแปลงได้ตลอดเวลาส่งผลให้ผู้ให้บริการและผู้เรียกใช้บริการมีความรู้ระหว่างกันไม่สมบูรณ์แบบ กล่าวคือในส่วนผู้ให้บริการไม่มีความรู้ที่เพียงพอเกี่ยวกับผู้เรียกใช้บริการ และในทางกลับกันผู้เรียกใช้บริการก็ไม่มีความรู้ที่เพียงพอเกี่ยวกับผู้ให้บริการ นอกจากนี้ทั้งสองก็ยังไม่มีความรู้ความเข้าใจในการให้บริการที่เหมาะสมสอดคล้องกับความต้องการของทั้งสองฝ่ายด้วยเช่นเดียวกัน



ภาพประกอบที่ 3.6 กรอบแนวคิดของสถาปัตยกรรมของระบบเจรจาต่อรองสำหรับการพัฒนาซอฟต์แวร์ตามสถาปัตยกรรมเชิงบริการ

ดังนั้นผู้ประสานงานการเจรจาต่อรอง (Negotiation coordinator) จะเป็นตัวกลางที่ช่วยควบคุมและจัดการความรู้ของผู้ให้บริการ ผู้ใช้บริการและตัวบริการ ตลอดจนความรู้ที่เกี่ยวข้องกับกระบวนการเจรจาต่อรองที่พึงมี ช่วยทำให้เกิดการแลกเปลี่ยนความรู้ระหว่างกัน และยังช่วยเพิ่มศักยภาพให้กระบวนการการเจรจาต่อรองสามารถดำเนินการได้อย่างสมบูรณ์ โดยหน้าที่หลักของผู้ประสานงานการเจรจาต่อรองในเบื้องต้นมีดังนี้

- 1) การจัดการความรู้ซึ่งประกอบด้วยความรู้เกี่ยวกับบริการ ความรู้เกี่ยวกับผู้ให้บริการและความรู้เกี่ยวกับผู้เรียกใช้บริการ
- 2) ช่วยสนับสนุนการเจรจาต่อรองซึ่งครอบคลุมตั้งแต่ช่วงก่อนการเจรจาต่อรอง ช่วงเจรจาต่อรอง และช่วงส่งมอบบริการเพื่อใช้งาน ตลอดถึงการลงนามรับรองการจับคู่ธุรกิจเชิงบริการและพร้อมทั้งเป็นพยานในการทำความเข้าใจ

รูปแบบการดำเนินงานโดยทั่วไปของผู้ประสานงานการเจรจาต่อรองมีลักษณะการสื่อสารไปยังผู้ที่เกี่ยวข้องแบบสองทางดังนี้

- 1) ผู้ประสานงานการเจรจาต่อรองสามารถสืบค้นบริการที่เกี่ยวข้อง จัดเก็บความรู้เกี่ยวกับบริการนั้นๆ และส่งข่าวสารที่จำเป็นไปยังผู้ให้บริการและผู้เรียกใช้บริการที่เป็นสมาชิกของระบบ ตัวอย่างข้อมูลที่เกี่ยวข้องกับตัวบริการ ได้แก่ ชื่อบริการ คำอธิบายโดยย่อ เจ้าของหรือตัวแทนจำหน่าย มูลค่าของตัวบริการ ระยะเวลาที่ส่งมอบ บริการหลังการส่งมอบ เป็นต้น
- 2) ผู้ให้บริการสามารถสมัครเป็นสมาชิกของระบบโดยให้ข้อมูลต่างๆที่เป็นประโยชน์ต่อการจับคู่กับความต้องการบริการของผู้เรียกใช้บริการ ตัวอย่างข้อมูลที่เป็นประโยชน์และสามารถจัดเก็บเป็นความรู้ในการเจรจาต่อรองได้ เช่น ขอบข่ายประเภทธุรกิจของบริการ (Business domain of service) ประวัติความสามารถในการให้บริการหรือรางวัลที่เคยได้รับ เขตพื้นที่ที่ให้บริการหรือมีตัวแทนจำหน่าย ทุนจดทะเบียนประกันความมั่นคงของผู้ให้บริการ ทั้งนี้ผู้ให้บริการสามารถพัฒนาตัวบริการและประกาศเผยแพร่สู่สาธารณะได้อย่างอิสระ
- 3) ผู้เรียกใช้บริการหากต้องการให้ผู้ประสานงานการเจรจาต่อรองเป็นสื่อกลางจับคู่ธุรกิจสามารถลงทะเบียนเป็นสมาชิกของระบบและให้ข้อมูลที่เป็นประโยชน์ต่อการเจรจาต่อรอง เช่น ประเภทธุรกิจ ขนาดเงินทุนของธุรกิจ หลักประกันความมั่นคงของบริษัทหรือประกันความน่าเชื่อถือ
- 4) นอกจากนี้ผู้ประสานงานสามารถเผยแพร่ข้อมูลระดับความน่าเชื่อถือของผู้ให้บริการและผู้ให้บริการ ตลอดถึงคุณภาพของบริการ (Quality of Service) ต่อสมาชิกที่

ลงทะเบียนกับระบบให้รับทราบ เพื่อประโยชน์ต่อทุกฝ่ายในการพัฒนาศักยภาพของธุรกิจเชิงบริการและความมั่นคงในการดำเนินธุรกิจในอนาคต

3.3 กระบวนการเจรจาต่อรองสำหรับสถาปัตยกรรมเชิงบริการ

ผู้วิจัยได้ออกแบบกระบวนการเจรจาต่อรองสำหรับสถาปัตยกรรมเชิงบริการบนพื้นฐานของมาตรฐานสำหรับการพัฒนาซอฟต์แวร์ดังกล่าวมาข้างต้น ดังแสดงด้วยแผนภาพกิจกรรมได้ในภาพประกอบที่ 3.7

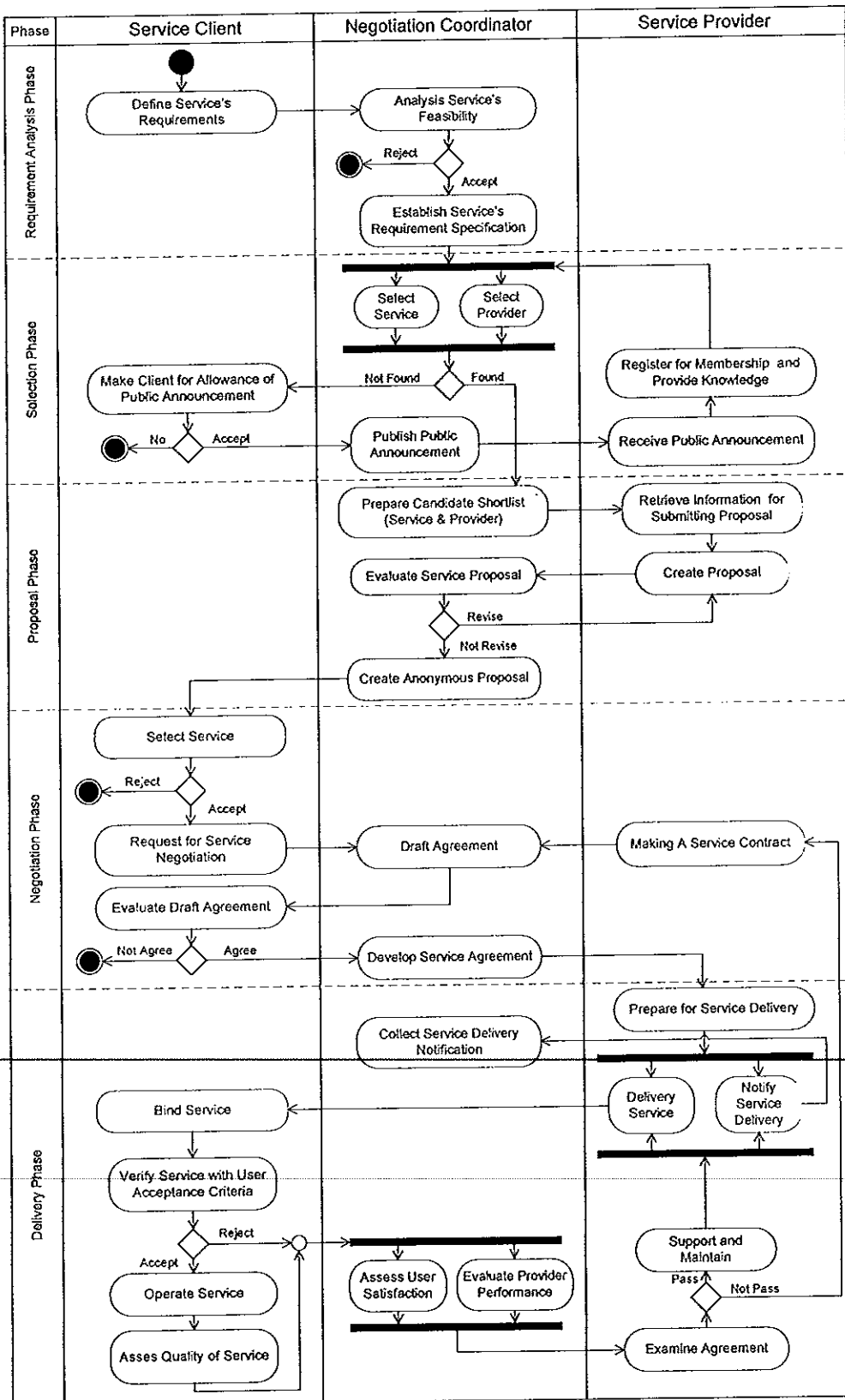
กระบวนการเจรจาต่อรองสามารถแบ่งส่วนการดำเนินงานออกเป็น 5 ระยะคือ ระยะเวลาวิเคราะห์ความต้องการ (Requirement analysis phase) ระยะเวลาคัดเลือก (Selection phase) ระยะเวลาเสนอ (Proposal Phase) ระยะเวลาเจรจาต่อรอง (Negotiation phase) และ ระยะเวลาส่งมอบ (Delivery phase) ซึ่งมีรายละเอียดดังนี้

1) ระยะเวลาวิเคราะห์ความต้องการ (Requirement Analysis Phase)

เริ่มต้นจากที่ผู้เรียกใช้บริการกำหนดความต้องการของตนเองและแจ้งต่อผู้ประสานงานการเจรจาต่อรองให้ทราบ จากนั้นผู้ประสานงานการเจรจาต่อรองจึงประเมินความเป็นไปได้ของความต้องการบริการที่ผู้เรียกใช้บริการระบุมาจากข้อมูลจากฐานความรู้ที่มีอยู่ และจะส่งข่าวสารแจ้งผลการประเมินกลับสู่ผู้เรียกใช้บริการให้รับทราบ ถ้าความต้องการบริการมีความเป็นไปได้ ผู้ประสานงานการเจรจาต่อรองจะจัดทำข้อกำหนดความต้องการ (Requirement specification)

2) ระยะเวลาคัดเลือก (Selection Phase)

ผู้ประสานงานการเจรจาต่อรองดำเนินการคัดเลือกบริการและผู้ให้บริการตามข้อกำหนดความต้องการ ถ้าไม่พบผู้ให้บริการและบริการที่เหมาะสม ผู้ประสานงานการเจรจาต่อรองจะแจ้งเกี่ยวกับการประกาศข้อกำหนดความต้องการแก่สาธารณะ (Public announcement) ให้ผู้เรียกใช้บริการพิจารณาอนุญาต ถ้าผู้เรียกใช้บริการอนุญาต ผู้ประสานงานการเจรจาต่อรองจะดำเนินการประกาศข้อกำหนดความต้องการแก่สาธารณะ ผู้ให้บริการก็จะรับข้อกำหนดความต้องการนี้ไปพิจารณา หากผู้ให้บริการรายใดๆรับข้อเสนอก็จะลงทะเบียนกับผู้ประสานงานการเจรจาต่อรองพร้อมกับให้ความรู้เพื่อผู้ประสานงานการเจรจาต่อรองนำไปใช้เป็นข้อมูลความรู้ในอนาคต



ภาพประกอบที่ 3.7 แผนภาพกิจกรรมของการเจรจาต่อรองสำหรับสถาปัตยกรรมเชิงบริการที่สอดคล้องกับมาตรฐานการพัฒนาซอฟต์แวร์ที่ได้กล่าวมา

3) ระยะเวลาเสนอ (Proposal Phase)

เมื่อคัดเลือกบริการและผู้ให้บริการเรียบร้อยแล้ว ผู้ประสานงานการเจรจาต่อรองจะจัดเตรียมบัญชีสรุปรายการของผู้เข้าแข่งขัน (Candidate shortlist) ซึ่งเป็นรายการของทั้งผู้ให้บริการและบริการที่ใกล้เคียงหรือเหมาะสมสอดคล้องกับความต้องการที่ระบุ และแจ้งให้ผู้ให้บริการเหล่านี้ยื่นข้อเสนอบริการ (Service proposal) เข้ามา

เมื่อผู้ประสานงานการเจรจาต่อรองได้รับข้อเสนอบริการจากผู้ให้บริการแล้วจึงประเมินข้อเสนอว่าต้องดำเนินการทบทวนแก้ไขหรือไม่ ถ้าจำเป็นต้องแก้ไขก็ส่งข่าวสารให้ผู้ให้บริการแก้ไขข้อเสนอและนำกลับมาเสนอใหม่อีกครั้ง จากนั้นผู้ประสานงานการเจรจาต่อรองจัดทำข้อเสนอบริการที่ไม่มีการระบุตัวผู้ให้บริการ เพื่อส่งให้ผู้เรียกใช้พิจารณาคัดเลือกบริการจากข้อเสนอบริการด้วยความเป็นธรรม

4) ระยะเวลาเจรจาต่อรอง (Negotiation Phase)

เมื่อผู้เรียกใช้บริการคัดเลือกบริการจากข้อเสนอบริการที่ไม่ระบุตัวผู้ให้บริการแล้ว ผู้เรียกใช้บริการจะร้องขอการเจรจาต่อรองเพื่อใช้บริการนั้น และหลังจากเจรจาต่อรองบรรลุผล ผู้ให้บริการก็จะจัดทำสัญญาการบริการของตน จากนั้นผู้ประสานงานการเจรจาต่อรองจึงร่างข้อตกลงบริการ (Service agreement) เพื่อให้ผู้เรียกใช้บริการได้ประเมินข้อตกลงบริการก่อนจะยินยอมทำข้อตกลงบริการที่แท้จริง ถ้าผ่านการประเมิน ทั้งผู้ให้บริการและผู้เรียกใช้บริการลงนามในข้อตกลงการใช้บริการระหว่างกัน โดยมีผู้ประสานงานการเจรจาต่อรองลงนามรับรองการจับคู่ธุรกิจและเป็นพยานในสัญญา ถ้าผู้เรียกใช้บริการไม่ยอมรับร่างข้อตกลงบริการ ผู้ประสานงานการเจรจาต่อรองก็ต้องเก็บผลตอบรับไว้ด้วย

5) ระยะเวลาส่งมอบ (Delivery Phase)

หลังจากทำข้อตกลงบริการเรียบร้อยแล้ว ผู้ให้บริการจะเตรียมความพร้อมที่จะส่งมอบบริการ หลังจากนั้นก็จะมีการส่งมอบบริการพร้อมกับแจ้งการส่งมอบให้ผู้ประสานงานการเจรจาต่อรองจัดเก็บรวบรวมไว้เป็นหลักฐานด้วย

จากนั้นผู้เรียกใช้บริการก็จะได้รับบริการ แล้วจึงตรวจสอบความถูกต้องของบริการ ถ้าตรวจสอบไม่ผ่านจะแจ้งผลให้ผู้ประสานงานการเจรจาต่อรองทราบ เพื่อส่งผลการประเมินไปยังผู้ให้บริการต่อไป ถ้าตรวจสอบผ่านถือว่าสามารถเรียกใช้บริการนั้นได้สมบูรณ์ และในระหว่างการใช้งานนั้นผู้เรียกใช้บริการก็จะประเมินคุณภาพของบริการ (Quality of service) ไปด้วย โดยส่งผลการประเมินให้ผู้ประสานงานการเจรจาต่อรองประเมินความพึงพอใจของผู้เรียกใช้บริการและประสิทธิภาพของผู้ให้บริการเพื่อเก็บไว้เป็นข้อมูลความรู้ในระบบ

ทั้งนี้เมื่อผู้ให้บริการได้รับผลการประเมินจะดำเนินการตรวจสอบเงื่อนไขในข้อตกลงด้วยว่าสอดคล้องกันหรือผิดเงื่อนไขหรือไม่ เช่น ช่วงระยะเวลาในสัญญาการให้บริการ เป็นต้น ถ้าเป็นไปตามเงื่อนไขข้อตกลง ผู้ให้บริการก็จะดำเนินการสนับสนุนและบำรุงรักษาบริการ แต่ถ้าผิดเงื่อนไขข้อตกลง ผู้ให้บริการจะดำเนินการจัดทำสัญญาเพื่อเจรจาข้อตกลงเกี่ยวกับบริการอีกครั้งหนึ่ง

3.4 สรุปท้ายบท

การศึกษาความสอดคล้องครอบคลุมของแนวคิดการเจรจาต่อรองสำหรับสถาปัตยกรรมเชิงบริการกับมาตรฐานการพัฒนาซอฟต์แวร์ที่มีอยู่ในปัจจุบัน โดยเฉพาะอย่างยิ่งแนวทางปรับปรุงกระบวนการตามแบบจำลองวุฒิภาวะความสามารถแบบบูรณาการ ผลจากการศึกษากลุ่มมาตรฐานดังกล่าวยังขาดการกล่าวถึงการพิจารณาในประเด็นที่เกี่ยวข้องกับสถาปัตยกรรมเชิงบริการ โดยเฉพาะในประเด็นการเจรจาต่อรอง ซึ่งจำเป็นต้องมีการปรับเปลี่ยนเพิ่มเติมในอนาคตเพื่อให้สามารถสนับสนุนการพัฒนาซอฟต์แวร์เชิงบริการ

นอกจากนี้ผู้วิจัยได้เสนอกรอบแนวคิดของการเจรจาต่อรองสำหรับสถาปัตยกรรมเชิงบริการที่สอดคล้องกับกระบวนการเจรจาต่อรอง เพื่อนำไปสู่การออกแบบและพัฒนาระบบในบทถัดไป

บทที่ 4

การออกแบบระบบผู้ประสานงานการเจรจาต่อรอง

เนื้อหาในบทนี้กล่าวถึงการออกแบบรูปแบบและโครงสร้างของระบบผู้ประสานงานการเจรจาต่อรอง ซึ่งเป็นระบบตัวกลางที่คอยสนับสนุนการเจรจาต่อรองสำหรับแนวคิดสถาปัตยกรรมเชิงบริการ และกล่าวถึงลำดับขั้นตอนการทำงาน วิธีการดำเนินงาน และข้อมูลต่างๆที่ประยุกต์ใช้งาน

4.1 ผู้ประสานงานการเจรจาต่อรอง (Negotiation Coordinator)

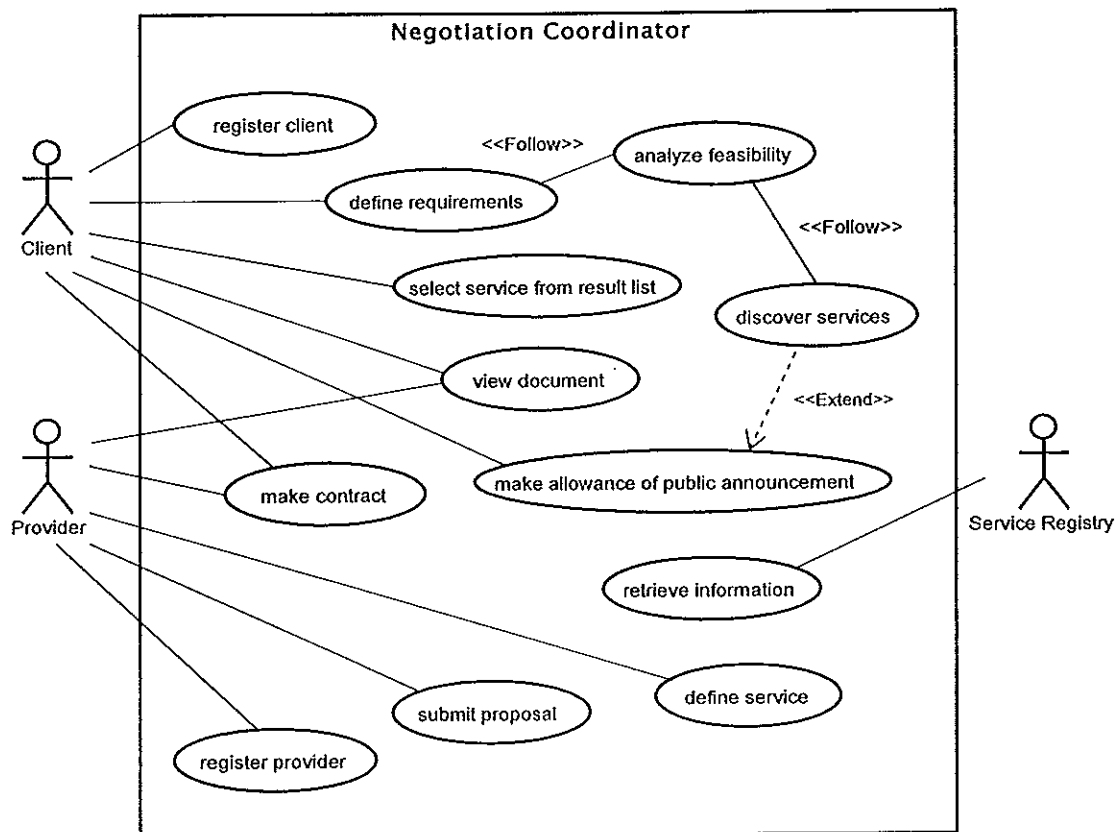
ผู้ประสานงานการเจรจาต่อรอง (Negotiation Coordinator) เป็นส่วนดำเนินการที่ใช้เทคนิคแนวคิดตัวกลาง (Mediator) มาประยุกต์ใช้ โดยมีบทบาทและหน้าที่ในการช่วยสนับสนุนและจัดการดำเนินการของการเจรจาต่อรองให้สามารถประสบความสำเร็จทั้งสองฝ่าย ทั้งนี้ยังช่วยให้ลดความเสี่ยงการเผชิญหน้าระหว่างกันในขั้นตอนการเจรจาต่อรองอีกด้วย

ผู้ประสานงานการเจรจาต่อรองจะช่วยอำนวยความสะดวกให้แก่ผู้เรียกใช้บริการสามารถค้นหาบริการได้เหมาะสมกับความต้องการบริการ และสนับสนุนให้ผู้ให้บริการสามารถนำบริการที่ตนมีมาเสนอเพื่อให้ผู้เรียกใช้บริการสามารถดำเนินการค้นหาได้ นอกจากนี้ยังสามารถช่วยดำเนินการทำสัญญาการใช้งานระหว่างกันของผู้เรียกใช้บริการและผู้ให้บริการ ซึ่งผู้ประสานงานการเจรจาต่อรองจะเป็นพยานในการจับคู่ทางธุรกิจนี้

4.2 ความสัมพันธ์และกิจกรรมของระบบ

ระบบผู้ประสานงานการเจรจาต่อรองมีส่วนที่เกี่ยวข้องทั้งหมด 3 ฝ่ายได้แก่ ผู้เรียกใช้บริการ ผู้ให้บริการ และหน่วยลงทะเบียนบริการ ซึ่งมีกิจกรรมและความสัมพันธ์ต่างๆ

ระหว่างกัน ดังแสดงภาพประกอบที่ 4.1 และมีรายละเอียดของแต่ละกิจกรรม (Use case) ดังคำอธิบายในตารางที่ 4.1 – ตารางที่ 4.12



ภาพประกอบที่ 4.1 กิจกรรม (Use case) และความสัมพันธ์ระหว่างกิจกรรมในระบบผู้ประสานงานการเจรจาต่อรอง

ตารางที่ 4.1 คำอธิบายกิจกรรม register client

ชื่อ	register client
เป้าหมาย	เพื่อให้ผู้เรียกใช้บริการลงทะเบียนการเป็นสมาชิกกับระบบ
ผู้ดำเนินการ	ผู้เรียกใช้บริการ (Client)
การดำเนินการ	ผู้เรียกใช้บริการจะต้องลงทะเบียนเพื่อสมัครเป็นสมาชิก พร้อมทั้งให้ข้อมูลรายละเอียดที่จำเป็นต่อการใช้งานกับระบบด้วย
เงื่อนไขหรือข้อยกเว้น	-
ผลลัพธ์	ข้อมูลสมาชิกของผู้เรียกใช้บริการรายนั้นๆ

ตารางที่ 4.2 คำอธิบายกิจกรรม register provider

ชื่อ	register provider
เป้าหมาย	เพื่อให้ผู้ให้บริการลงทะเบียนเป็นสมาชิกกับระบบ
ผู้ดำเนินการ	ผู้ให้บริการ (Provider)
การดำเนินการ	ผู้ให้บริการจะต้องลงทะเบียนเพื่อสมัครเป็นสมาชิก พร้อมทั้งให้ข้อมูลรายละเอียดที่จำเป็นต่อการใช้งานกับระบบด้วย
เงื่อนไขหรือข้อยกเว้น	-
ผลลัพธ์	ข้อมูลสมาชิกของผู้ให้บริการรายนั้นๆ

ตารางที่ 4.3 คำอธิบายกิจกรรม define requirement

ชื่อ	define requirement
เป้าหมาย	เพื่อให้ผู้เรียกใช้บริการสามารถกำหนดคุณลักษณะของบริการตามที่ตนต้องการ
ผู้ดำเนินการ	ผู้เรียกใช้บริการ (Client)
การดำเนินการ	ผู้เรียกใช้บริการดำเนินการระบุข้อมูลรายละเอียดคุณลักษณะของบริการที่ตนต้องการ ซึ่งเป็นไปตามรูปแบบที่ระบบได้กำหนดไว้
เงื่อนไขหรือข้อยกเว้น	ผู้เรียกใช้บริการจะต้องเข้าสู่ระบบ
ผลลัพธ์	ข้อกำหนดคุณลักษณะบริการ

ตารางที่ 4.4 คำอธิบายกิจกรรม analyze feasibility

ชื่อ	analyze feasibility
เป้าหมาย	เพื่อให้ระบบดำเนินการตรวจสอบความเป็นไปได้จากการกำหนดความต้องการบริการของผู้เรียกใช้บริการในเบื้องต้น ก่อนการดำเนินการค้นหาบริการในลำดับถัดไป
ผู้ดำเนินการ	ระบบผู้ประสานงานการเจรจาต่อรอง (Negotiation Coordinator)
การดำเนินการ	เมื่อผู้เรียกใช้บริการกำหนดคุณลักษณะของบริการตามความต้องการแล้ว ระบบจะดำเนินการตรวจสอบความเป็นไปได้ของข้อกำหนดนั้นๆ ก่อนที่จะนำคุณลักษณะดังกล่าวไปดำเนินการค้นหาบริการที่เหมาะสมที่ปรากฏอยู่ในระบบต่อไป
เงื่อนไขหรือข้อยกเว้น	ผู้เรียกใช้บริการจะต้องกำหนดคุณลักษณะบริการตามที่ต้องการ
ผลลัพธ์	สถานะการผ่านการพิจารณาความเป็นไปได้

ตารางที่ 4.5 คำอธิบายกิจกรรม discover services

ชื่อ	discover services
เป้าหมาย	เพื่อดำเนินการค้นหาบริการที่เหมาะสมจากคุณลักษณะบริการตามที่ผู้เรียกใช้บริการกำหนด
ผู้ดำเนินการ	ระบบผู้ประสานงานการเจรจาต่อรอง (Negotiation Coordinator)
การดำเนินการ	เมื่อพิจารณาความเป็นไปได้ของคุณลักษณะบริการเป็นที่เรียบร้อยแล้วระบบจะนำคุณลักษณะดังกล่าวไปค้นหาบริการที่สอดคล้องเหมาะสมโดยประยุกต์ใช้ฐานข้อมูลความรู้เข้ามาสนับสนุน
เงื่อนไขหรือข้อยกเว้น	ข้อกำหนดคุณลักษณะบริการของผู้เรียกใช้บริการจะต้องมีความเป็นไปได้
ผลลัพธ์	รายการของกลุ่มบริการที่เป็นผลลัพธ์ของการค้นหา

ตารางที่ 4.6 คำอธิบายกิจกรรม select service from result list

ชื่อ	select service from result list
เป้าหมาย	เพื่อให้ผู้เรียกใช้บริการสามารถคัดเลือกบริการจากกลุ่มผลลัพธ์บริการตามที่ตนพึงพอใจ
ผู้ดำเนินการ	ผู้เรียกใช้บริการ (Client)
การดำเนินการ	ระบบจะเสนอรายการของกลุ่มผลลัพธ์บริการที่ได้จากการค้นหาให้ผู้เรียกใช้บริการสามารถคัดเลือกบริการตามที่ตนต้องการ
เงื่อนไขหรือข้อยกเว้น	ค้นพบกลุ่มผลลัพธ์บริการจากการค้นหาในระบบ
ผลลัพธ์	บริการที่ผู้เรียกใช้บริการคัดเลือกเรียบร้อยแล้ว

ตารางที่ 4.7 คำอธิบายกิจกรรม make allowance of public announcement

ชื่อ	make allowance of public announcement
เป้าหมาย	สำหรับประกาศข้อกำหนดคุณลักษณะของบริการที่ระบบค้นหาแล้วไม่พบ เพื่อให้ผู้ให้บริการที่มีบริการตรงกับข้อกำหนดคุณลักษณะบริการตามที่ประกาศได้เสนอบริการของตนแก่ระบบ
ผู้ดำเนินการ	ผู้เรียกใช้บริการ (Client)
การดำเนินการ	ถ้าดำเนินการค้นหาบริการที่มีในระบบตามข้อกำหนดคุณลักษณะบริการไม่พบ ระบบจะดำเนินการสร้างประกาศข้อกำหนดความต้องการสู่สาธารณะ (Public announcement) เพื่อให้ผู้ให้บริการราย

ตารางที่ 4.7 (ต่อ)

การดำเนินการ	อื่นๆสามารถนำบริการที่ตัวเองมีอยู่และสอดคล้องไปกับข้อกำหนดความต้องการที่ประกาศไว้มาเสนอแก่ระบบ ทั้งนี้การประกาศดังกล่าวจะต้องได้รับความเห็นชอบจากผู้เรียกใช้บริการรายนั้นๆ ด้วย
เงื่อนไขหรือข้อยกเว้น	ไม่ปรากฏกลุ่มผลลัพธ์บริการจากการค้นหาในระบบ
ผลลัพธ์	รายการประกาศข้อกำหนดความต้องการสู่สาธารณะ

ตารางที่ 4.8 คำอธิบายกิจกรรม make contract

ชื่อ	make contract
เป้าหมาย	เพื่อกระทำการสร้างสัญญาสำหรับการเรียกใช้งานบริการระหว่างผู้เรียกใช้บริการและผู้ให้บริการ โดยมีผู้ประสานงานการเจรจาต่อรองเป็นผู้คอยสนับสนุนและเป็นพยานในการจับคู่ทางธุรกิจนี้
ผู้ดำเนินการ	ผู้เรียกใช้บริการ (Client), ผู้ให้บริการ (Provider) และระบบผู้ประสานงานการเจรจาต่อรอง (Negotiation Coordinator)
การดำเนินการ	ระบบผู้ประสานงานการเจรจาต่อรองจะดำเนินการสร้างเอกสารสัญญาฉบับร่าง (Draft agreement) ซึ่งจะเสนอให้ผู้เรียกใช้บริการตรวจสอบและยอมรับ และจากนั้นจึงสร้างเอกสารสัญญาฉบับจริงที่ทั้งผู้เรียกใช้บริการ และผู้ให้บริการยอมรับ โดยผู้ให้บริการอาจจะแนบสัญญาการใช้งานอื่นๆประกอบมาได้
เงื่อนไขหรือข้อยกเว้น	ไม่ปรากฏการคัดเลือกบริการตามที่ต้องการ
ผลลัพธ์	เอกสารสัญญาฉบับร่าง และเอกสารสัญญาฉบับจริง

ตารางที่ 4.9 คำอธิบายกิจกรรม view document

ชื่อ	view document
เป้าหมาย	เพื่อให้ทั้งผู้เรียกใช้บริการและผู้ให้บริการที่เป็นสมาชิกอยู่ในระบบสามารถพิจารณาเอกสารต่างที่เกิดขึ้นจากขั้นตอนการดำเนินการต่างๆ
ผู้ดำเนินการ	ผู้เรียกใช้บริการ (Client) และผู้ให้บริการ (Provider)
การดำเนินการ	ระบบผู้ประสานงานการเจรจาต่อรองจะแสดงเอกสารต่างๆที่สอดคล้องกับการดำเนินการ เช่น เอกสารข้อกำหนดความต้องการ

ตารางที่ 4.9 (ต่อ)

การดำเนินการ	บริการ (Requirement specification) หรือเอกสารสัญญาฉบับร่าง (Draft agreement) เป็นต้น
เงื่อนไขหรือข้อยกเว้น	-
ผลลัพธ์	รายการแสดงเอกสารต่างที่เกิดขึ้น

ตารางที่ 4.10 คำอธิบายกิจกรรม define service

ชื่อ	define service
เป้าหมาย	เพื่อให้ผู้ให้บริการที่ลงทะเบียนกับระบบ สามารถกำหนดบริการที่ตนมีอยู่แก่ระบบ โดยระบุถึงคุณลักษณะต่างๆที่จำเป็นของบริการ
ผู้ดำเนินการ	ผู้ให้บริการ (Provider)
การดำเนินการ	ผู้ให้บริการจะกำหนดข้อมูลรายละเอียดของคุณลักษณะของบริการที่ตนมีอยู่ ซึ่งระบบจะกำหนดขอบเขตของคุณลักษณะเหล่านั้นไว้ ระบบจะนำข้อมูลที่ได้ไปประกอบการค้นหาในขั้นตอนถัดไป
เงื่อนไขหรือข้อยกเว้น	ผู้ให้บริการต้องลงทะเบียนเป็นสมาชิกกับระบบ
ผลลัพธ์	ข้อมูลคุณลักษณะของบริการของผู้ให้บริการรายนั้นๆ

ตารางที่ 4.11 คำอธิบายกิจกรรม submit proposal

ชื่อ	submit proposal
เป้าหมาย	เพื่อให้ผู้ให้บริการสามารถนำเสนอบริการที่ตนมีอยู่โดยการยื่นข้อเสนอบริการ (Service proposal) ให้แก่ผู้เรียกใช้บริการ
ผู้ดำเนินการ	ผู้ให้บริการ (Provider)
การดำเนินการ	เมื่อระบบดำเนินการค้นหาบริการตามข้อกำหนดความต้องการบริการของผู้เรียกใช้บริการแล้ว ผู้ให้บริการที่เป็นเจ้าของผลลัพธ์บริการรายนั้นๆจะต้องยื่นข้อเสนอบริการเข้ามา เพื่อให้ผู้เรียกใช้บริการคัดเลือกในลำดับถัดไป
เงื่อนไขหรือข้อยกเว้น	-
ผลลัพธ์	เอกสารข้อเสนอบริการ (Service proposal) จากผู้ให้บริการ

ตารางที่ 4.12 คำอธิบายกิจกรรม retrieve information

ชื่อ	retrieve information
เป้าหมาย	เพื่อค้นคืนข้อมูลที่เกี่ยวข้องกับบริการจากหน่วยลงทะเบียนบริการ (Service registry)
ผู้ดำเนินการ	ระบบผู้ประสานงานการเจรจาต่อรอง (Negotiation Coordinator)
การดำเนินการ	ระบบผู้ประสานงานการเจรจาต่อรองจะสอบถามข้อมูลคุณลักษณะของบริการที่มีอยู่ในหน่วยลงทะเบียนบริการ เพื่อมาผนวกเข้ากับข้อมูลที่ปรากฏอยู่ในระบบ เพื่อส่งไปใช้เป็นฐานความรู้ในลำดับถัดไป
เงื่อนไขหรือข้อยกเว้น	ข้อมูลคุณลักษณะบริการที่ได้จะขึ้นอยู่กับลักษณะของหน่วยลงทะเบียนบริการที่นำมาประยุกต์ใช้
ผลลัพธ์	ข้อมูลคุณลักษณะบริการอื่นๆ

4.3 ขั้นตอนการดำเนินงานของระบบ

จากการพิจารณากิจกรรมและความสัมพันธ์ระหว่างกิจกรรมในภาพประกอบที่ 4.1 สามารถอธิบายกระบวนการดำเนินงานหลักระหว่างผู้มีส่วนร่วมในระบบ ได้แก่ ผู้เรียกใช้บริการ ผู้ให้บริการ และผู้ประสานงานการเจรจาต่อรอง ซึ่งอธิบายด้วยแผนภาพลำดับกิจกรรม (Sequence diagram) ได้ดังภาพประกอบที่ 4.2 ส่วนภาพประกอบที่ 4.3 แสดงขั้นตอนลำดับกิจกรรมย่อยของการลงทะเบียนเป็นสมาชิกของระบบ และภาพประกอบที่ 4.4 เป็นขั้นตอนลำดับกิจกรรมย่อยสำหรับการประกาศบริการของผู้ให้บริการโดยประกอบด้วยผู้มีส่วนร่วมเพิ่มเติมคือหน่วยลงทะเบียนบริการ

รายละเอียดของลำดับขั้นตอนการดำเนินงานของระบบมีดังนี้

1) ผู้เรียกใช้บริการ (Client) หรือผู้ให้บริการ (Provider) ดำเนินการลงทะเบียนสมัครสมาชิกกับระบบ โดยให้ข้อมูลที่เกี่ยวข้องกับรายละเอียดส่วนต่างๆตามข้อกำหนด

2) หลังจากลงทะเบียนและเข้าสู่ระบบเรียบร้อยแล้ว ผู้ให้บริการสามารถระบุข้อมูลคุณลักษณะต่างๆของบริการที่ต้องการจะเสนอแก่ระบบ จากนั้นระบบจะนำคุณลักษณะของบริการ ไปค้นหาในหน่วยลงทะเบียนบริการ (Service registry) เพื่อดึงเอาคุณลักษณะเสริมอื่นๆของบริการนั้นไปผนวกเข้ากับคุณลักษณะของบริการตามที่ผู้ให้บริการได้กำหนดมา เพื่อให้ข้อมูลบริการที่ได้มีความสมบูรณ์ยิ่งขึ้นก่อนการจัดเก็บข้อมูล

3) ในส่วนผู้เรียกใช้บริการ หลังจากลงทะเบียนและเข้าสู่ระบบเรียบร้อยแล้ว สามารถกำหนดความต้องการบริการ ซึ่งก็คือการระบุข้อมูลคุณลักษณะของบริการที่ตนเองต้องการ

4) ระบบผู้ประสานงานการต่อรองจะนำข้อมูลรายละเอียดของการกำหนดความต้องการดังกล่าวนี้ไปดำเนินการพิจารณาความเป็นไปได้ (Analyze feasibility) ถ้าพิจารณาแล้วไม่ผ่านก็จะแจ้งให้ผู้เรียกใช้บริการทราบ เพื่อให้ผู้เรียกใช้บริการระบุข้อมูลความต้องการที่เหมาะสมต่อไป

5) หลังจากการทดสอบความเป็นไปได้ของความต้องการแล้ว ระบบจะนำข้อมูลเหล่านี้มาสร้างข้อกำหนดความต้องการบริการ (Service requirements specification) ซึ่งผู้เรียกใช้บริการนั้นๆสามารถเรียกแสดงได้ และระบบจะนำข้อมูลเหล่านี้ไปดำเนินการค้นหาบริการในลำดับถัดไป

6) ถ้าค้นหาบริการแล้วไม่ปรากฏบริการที่ตรงกับความต้องการอยู่ในระบบจะดำเนินการสร้างประกาศข้อกำหนดความต้องการบริการสู่สาธารณะ (Service public announcement) ซึ่งเป็นการประกาศข้อกำหนดความต้องการบริการที่ดำเนินการค้นหาบริการที่มีอยู่ในระบบแล้วไม่พบออกสู่สาธารณะ เพื่อให้ผู้ให้บริการรายอื่นๆที่มีบริการสอดคล้องกับข้อกำหนดความต้องการบริการนี้มาเสนอบริการให้แก่ระบบ อย่างไรก็ตามผู้เรียกใช้บริการที่กำหนดความต้องการดังกล่าวจะต้องเห็นชอบต่อการประกาศนั้นๆเสียก่อนถึงจะนำไปประกาศข้อกำหนดความต้องการบริการออกสู่สาธารณะได้

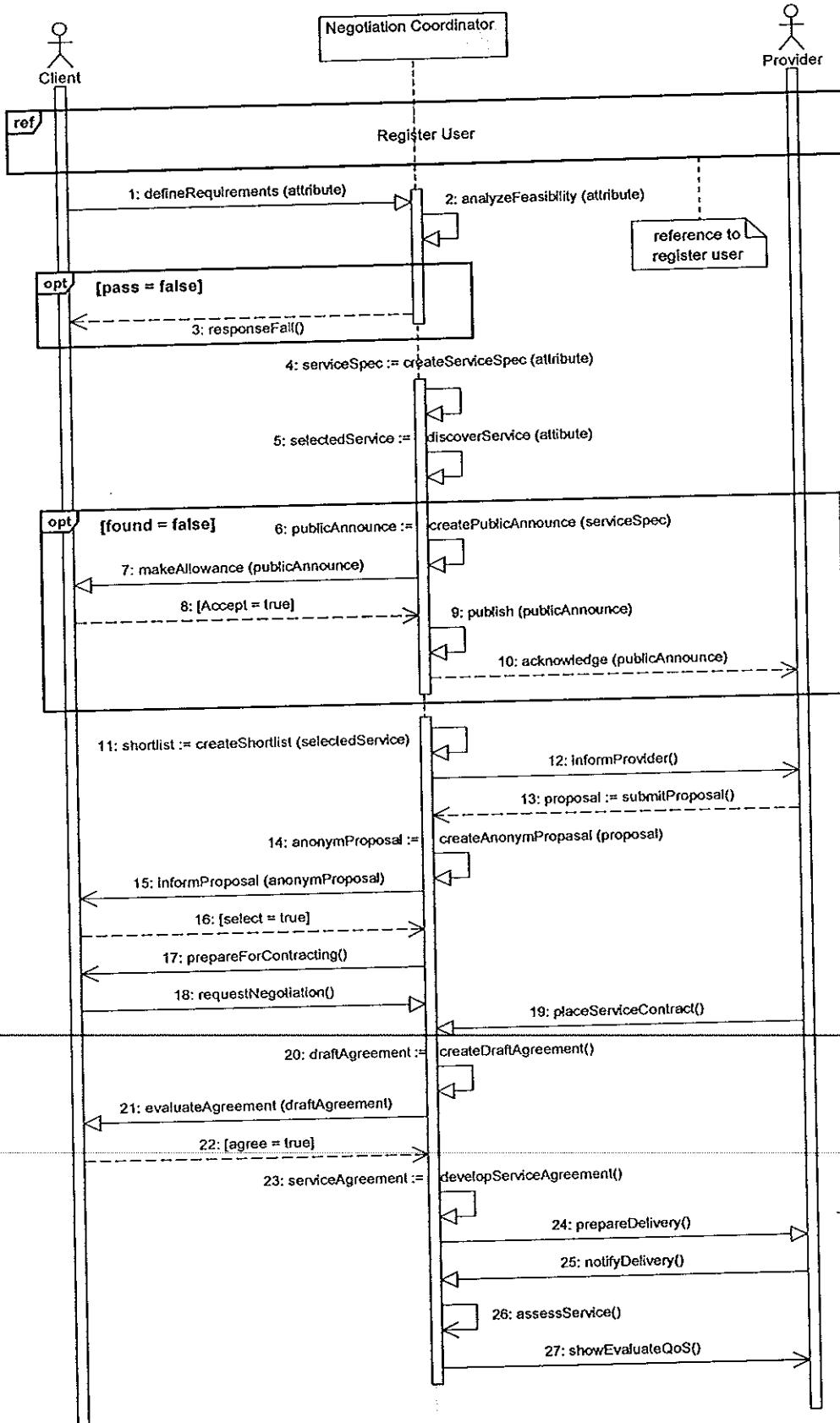
7) ถ้าค้นหาพบบริการตามผู้เรียกใช้ต้องการแล้ว ระบบจะดำเนินการสร้างบัญชีสรุปรายการผู้เข้าแข่งขัน (Candidate shortlist) ซึ่งก็คือรายการของกลุ่มบริการที่หาได้ พร้อมทั้งผู้ให้บริการรายนั้นๆด้วย และแจ้งข่าวสารให้ผู้ให้บริการที่ปรากฏอยู่ในบัญชีให้รับทราบผ่านทางสื่อต่างๆ เช่น ส่วนติดต่อผู้ใช้ อีเมล หรือเทคโนโลยีการส่งข่าวสารอื่นๆ เป็นต้น

8) เมื่อได้รับข่าวสารจากระบบเรียบร้อยแล้ว ผู้ให้บริการจะยื่นข้อเสนอบริการ (Service proposal) ให้แก่ระบบ

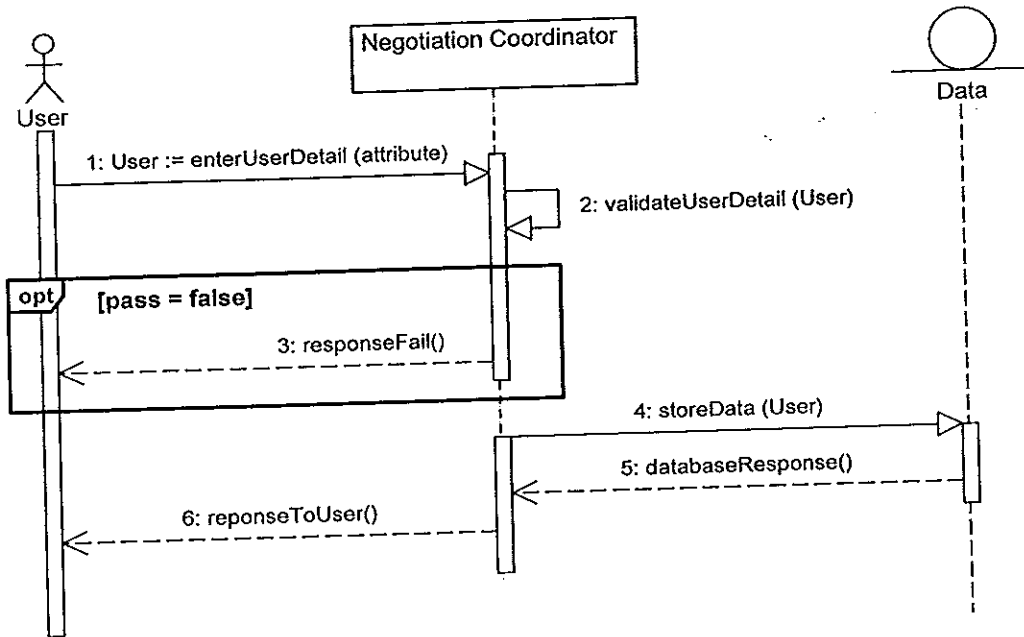
9) จากนั้นระบบจะนำแบบข้อเสนอเหล่านั้นไปดำเนินการตัดชื่อและข้อมูลของผู้ให้บริการออกไปให้เหลือเพียงแต่ข้อมูลที่เกี่ยวข้องกับบริการเท่านั้นหรือที่เรียกว่าข้อเสนอแบบปกปิดชื่อ (Anonymous proposal) แล้วนำเสนอให้ผู้เรียกใช้บริการนั้นรับทราบ

10) เมื่อผู้เรียกใช้บริการรับทราบ ก็จะคัดเลือกบริการเพียงหนึ่งเดียวตามที่ตนเองพึงพอใจโดยพิจารณาจากข้อมูลคุณลักษณะของบริการตามที่ปรากฏอยู่

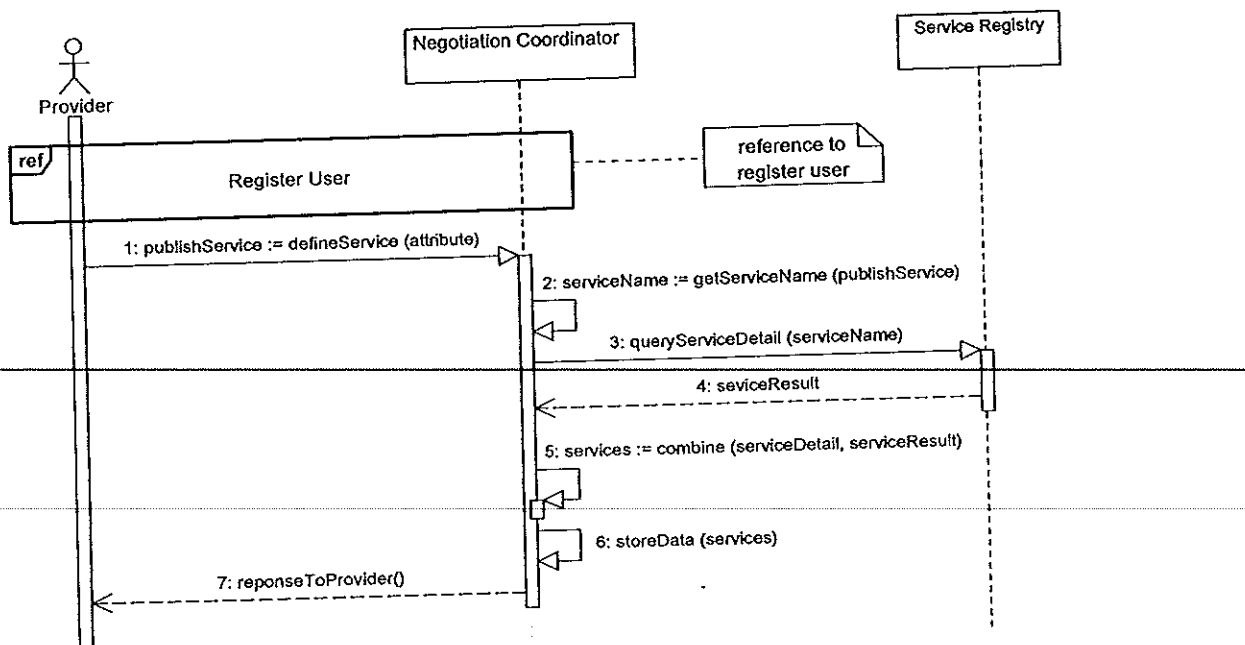
11) หลังจากผู้เรียกใช้เลือกบริการเรียบร้อยแล้ว ระบบจะกระทำการแจ้งให้ผู้เรียกใช้บริการและผู้ให้บริการรับทราบถึงการเข้าสู่กระบวนการทำสัญญาระหว่างผู้ให้บริการกับผู้เรียกใช้บริการ โดยมีผู้ประสานงานการต่อรองเป็นตัวกลางคอยสนับสนุน พร้อมทั้งให้ผู้เรียกใช้รอกอยการรับข่าวสารที่แสดงถึงการพร้อมสำหรับการทำสัญญาด้วย



ภาพประกอบที่ 4.2 แผนภาพลำดับกิจกรรมของการดำเนินงานโดยรวม



ภาพประกอบที่ 4.3 แผนภาพลำดับกิจกรรมของการลงทะเบียนเป็นสมาชิกของระบบ



ภาพประกอบที่ 4.4 แผนภาพลำดับกิจกรรมของการเสนอบริการ
ของผู้ให้บริการที่เป็นสมาชิกกับระบบ

12) ระบบจะแจ้งให้ผู้ให้บริการยื่นสัญญาบริการของตนแก่ระบบ

13) เมื่อผู้ให้บริการยื่นสัญญาเรียบร้อยแล้ว ระบบจะดำเนินการร่างสัญญาฉบับร่าง (Draft service agreement) และพร้อมทั้งแจ้งให้ผู้เรียกใช้บริการมาตรวจสอบสัญญาฉบับร่างและ/หรือสัญญาของผู้ให้บริการว่าพึงพอใจหรือไม่ ถ้าไม่ก็ดำเนินการยกเลิกสัญญาฉบับร่าง และระบบจะแจ้งการยกเลิกสัญญาให้กับผู้ให้บริการรายนั้นๆ ด้วย

14) ถ้าผู้เรียกใช้บริการพึงพอใจกับสัญญาฉบับร่างนี้ ระบบจะดำเนินการสร้างสัญญาบริการ (Service agreement) ขึ้นมาโดยอ้างอิงสัญญาฉบับร่างเป็นหลัก และเสนอให้ผู้เรียกใช้บริการและผู้ให้บริการรับทราบ พร้อมทั้งแจ้งให้ผู้ให้บริการเตรียมตัวสำหรับการส่งมอบบริการ

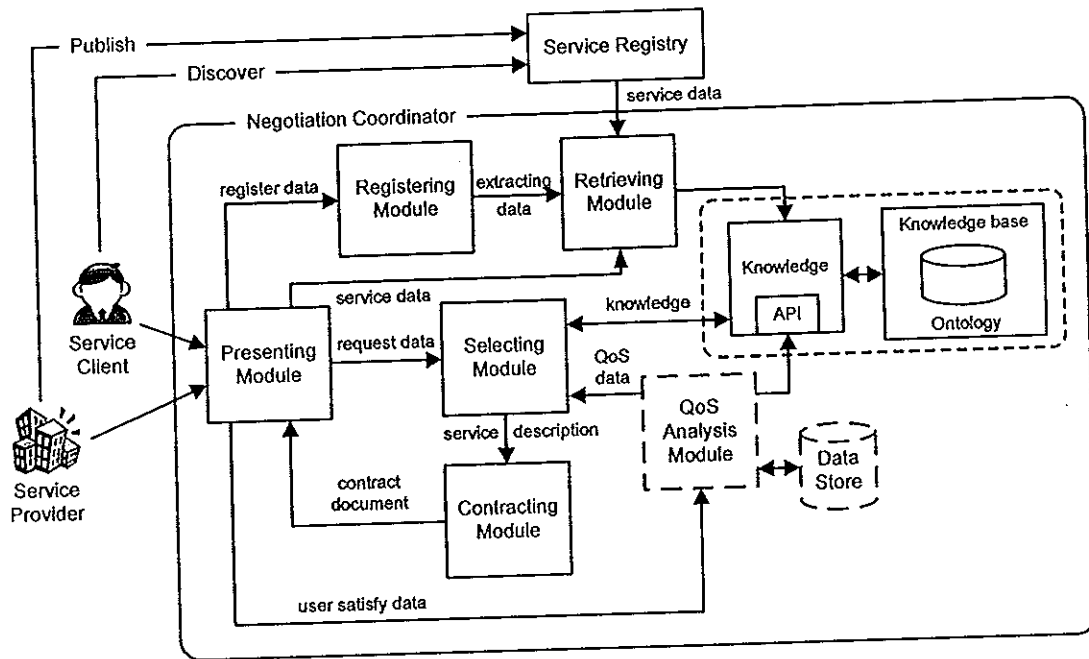
15) เมื่อผู้ให้บริการพร้อมสำหรับการส่งมอบบริการแล้ว ผู้ให้บริการจะแจ้งให้ระบบรับทราบ จากนั้นระบบจะแจ้งข่าวสารการส่งมอบบริการและส่งข้อมูลการเรียกใช้บริการนั้นๆ ไปยังผู้เรียกใช้บริการ ซึ่งหลังจากขั้นตอนนี้ผู้เรียกใช้บริการสามารถเรียกใช้บริการได้อย่างสมบูรณ์

16) ในระหว่างที่ผู้เรียกใช้บริการดำเนินการใช้บริการอยู่นั้น ระบบจะทดสอบและประเมินบริการ ซึ่งในที่นี้คือการทดสอบคุณภาพของบริการ (Quality of Service) เพื่อใช้เป็นข้อมูลเสริมในการคัดเลือกบริการในลำดับถัดไป

17) เมื่อช่วงระยะเวลาในสัญญาการให้บริการสิ้นสุดลง ระบบจะยกเลิกสัญญาบริการและสัญญาที่เกี่ยวข้องนั้นๆ แล้วดำเนินการแจ้งให้ทั้งผู้ให้บริการและผู้เรียกใช้บริการรับทราบต่อไป

4.4 สถาปัตยกรรมของระบบ

ระบบผู้ประสานงานการเจรจาต่อรองมีองค์ประกอบการทำงานเพื่อรองรับการลงทะเบียนของผู้เรียกใช้บริการและผู้ให้บริการ การกำหนดความต้องการบริการ การสืบค้นบริการที่สอดคล้องกับความต้องการ รวมถึงการสนับสนุนการทำสัญญาและการประเมินคุณภาพของบริการด้วย ซึ่งสามารถแสดงโครงสร้างของระบบผู้ประสานงานการเจรจาต่อรองได้ดังภาพประกอบที่ 4.5



ภาพประกอบที่ 4.5 สถาปัตยกรรมของระบบผู้ประสานงาน

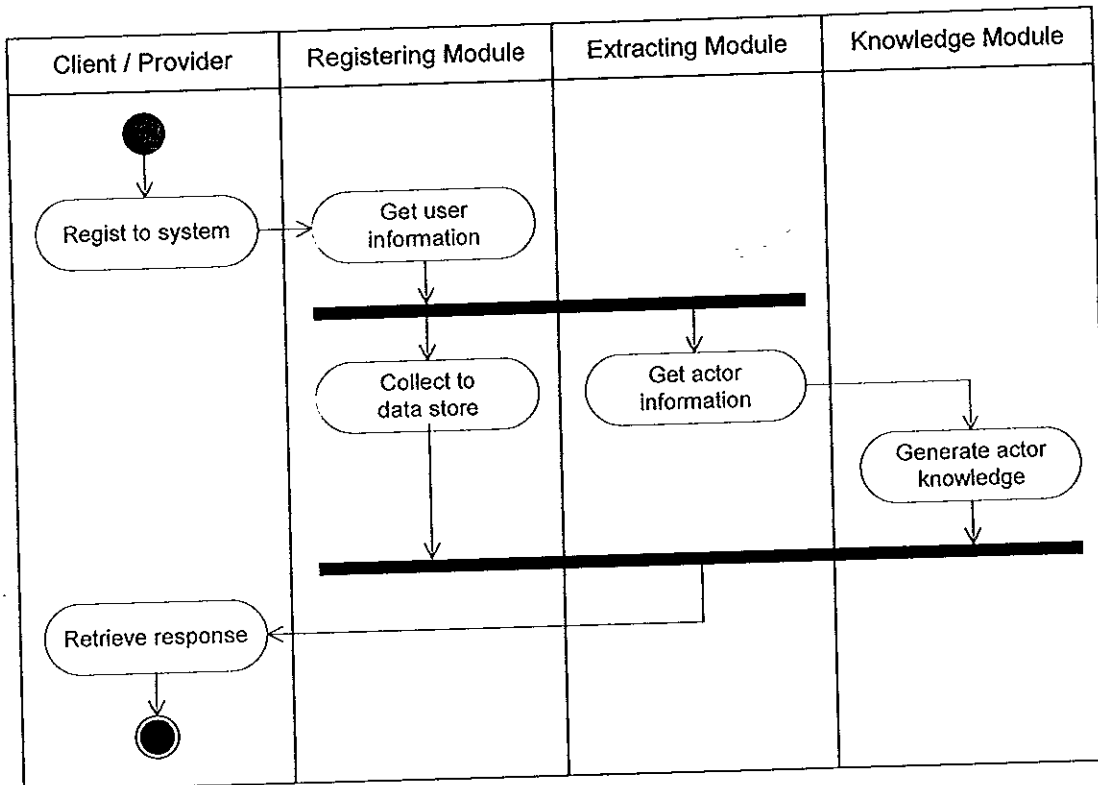
การเจรจาต่อรอง (Negotiation Coordinator)

ระบบผู้ประสานงานการต่อรองจะประกอบด้วยองค์ประกอบหลักทั้งหมด 7 ส่วน ซึ่งมีรายละเอียดของการดำเนินการดังนี้

1) ส่วนนำเสนอ (Presenting module) เป็นส่วนที่เกี่ยวข้องกับการติดต่อกับผู้เรียกใช้บริการและผู้ให้บริการในระบบ เช่น การแสดงการลงทะเบียนในระบบ การแสดงร่างสัญญาต่างๆ และการแสดงส่วนกำหนดความต้องการบริการ เป็นต้น รวมถึงการส่งข่าวสารต่างๆที่เกิดขึ้นในระบบแก่ผู้ใช้ ตัวอย่างส่วนติดต่อผู้ใช้ที่สามารถนำมาประยุกต์ใช้งานได้ เช่น หน้าเว็บ หรือส่วนติดต่อกับผู้ใช้แบบกราฟิก (Graphic user interface)

2) ส่วนลงทะเบียนสมาชิก (Registering module) ทำหน้าที่ควบคุมและจัดการการลงทะเบียนเป็นสมาชิกกับระบบของผู้ให้บริการและผู้เรียกใช้บริการ ซึ่งในส่วนนี้จะส่งข้อมูลที่จำเป็นไปยังส่วนอื่นๆของระบบด้วย

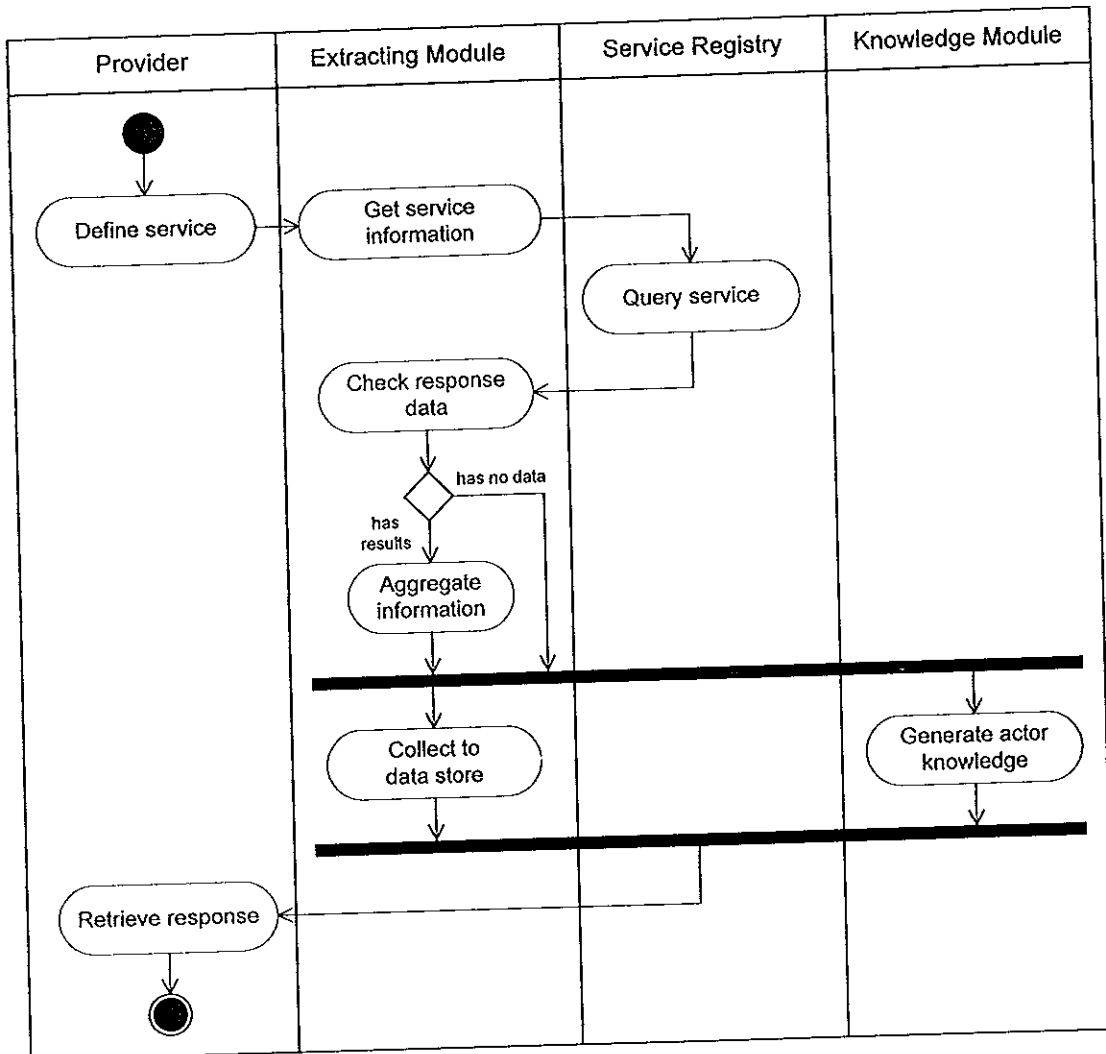
3) ส่วนดึงข้อมูล (Retrieving module) จะดำเนินการดึงข้อมูลคุณลักษณะของบริการจากหน่วยลงทะเบียนบริการ (Service registry) และข้อมูลรายละเอียดต่างๆของทั้งผู้ให้บริการ ผู้เรียกใช้บริการ และตัวบริการจากที่ได้ลงทะเบียนและกำหนดเอาไว้ โดยสามารถแสดงการดำเนินงานได้ดังภาพประกอบที่ 4.6 และ 4.7



ภาพประกอบที่ 4.6 ขั้นตอนการดึงข้อมูลผู้ใช้จาก
การลงทะเบียนสมัครเป็นสมาชิกกับระบบ

การดำเนินงานของการดึงข้อมูลจะปรากฏใน 2 ขั้นตอนคือ ในขั้นตอนการลงทะเบียนสมัครเป็นสมาชิกกับระบบทั้งผู้ให้บริการและผู้เรียกใช้บริการ และในขั้นตอนการระบุบริการที่ต้องการจะเสนอของผู้ให้บริการ

สำหรับขั้นตอนการลงทะเบียนเป็นสมาชิกกับระบบนั้นจะมีการดำเนินงานเกี่ยวข้องทั้งหมด 4 ส่วนได้แก่ ส่วนผู้เรียกใช้บริการหรือผู้ให้บริการ ส่วนลงทะเบียนสมาชิก ส่วนดึงข้อมูล และส่วนความรู้ โดยเริ่มจากผู้เรียกใช้บริการหรือผู้ให้บริการกระทำการลงทะเบียนเป็นสมาชิกกับระบบ โดยจะต้องระบุข้อมูลคุณลักษณะต่างๆตามความเหมาะสมด้วย ส่วนดึงข้อมูลจะดึงเอาข้อมูลเหล่านี้ไปบันทึกลงฐานข้อมูล และส่งข้อมูลดังกล่าวไปให้ส่วนความรู้ ซึ่งจะนำไปสร้างความรู้ของผู้ใช้ที่สอดคล้องกับข้อมูลที่ใช้ระบุมา เพื่อพร้อมนำไปเป็นฐานความรู้สำหรับการค้นหาในส่วนถัดไป



ภาพประกอบที่ 4.7 ขั้นตอนการดึงข้อมูลบริการจากการระบุบริการของผู้ให้บริการ

สำหรับในขั้นตอนที่ผู้ให้บริการดำเนินการระบุบริการนั้น ประกอบด้วยส่วน
 เกี่ยวข้อง 4 ส่วนคือผู้ให้บริการ ส่วนดึงข้อมูล ส่วนความรู้ และหน่วยลงทะเบียนบริการ โดย
 เริ่มต้นจากผู้ให้บริการดำเนินการระบุบริการที่ตนมีอยู่แก่ระบบ ส่วนดึงข้อมูลจะนำข้อมูลหลัก
 จากกลุ่มข้อมูลที่ผู้ให้บริการระบุมา ไปค้นหาในหน่วยลงทะเบียนบริการเพื่อให้ได้ข้อมูลอื่นๆที่
 นอกเหนือจากที่มีอยู่สำหรับเป็นข้อมูลเสริม หลังจากได้รับผลตอบรับจากหน่วยลงทะเบียน
 บริการแล้ว ส่วนดึงข้อมูลจะทำการตรวจสอบว่ามีข้อมูลที่เป็นผลลัพธ์หรือไม่ ซึ่งถ้ามีก็จะนำไป
 ผสมเข้ากับข้อมูลของบริการเดิมที่มีอยู่ จากนั้นจึงนำไปบันทึกลงฐานข้อมูล และส่งกลุ่มข้อมูล
 ดังกล่าวไปยังส่วนความรู้เพื่อนำไปสร้างเป็นความรู้ของบริการที่พร้อมนำไปเป็นฐานความรู้
 สำหรับการค้นหาเช่นเดียวกัน

4) ส่วนความรู้ (Knowledge module) เป็นส่วนที่ควบคุมและจัดการความรู้ที่เกี่ยวข้องทั้งหมดในระบบ ซึ่งมีส่วนดำเนินการย่อยได้แก่การแทนความรู้ การจัดเก็บความรู้ และการค้นคืนความรู้

5) ส่วนคัดเลือกบริการ (Selecting module) ดำเนินการคัดเลือกบริการที่ตรงกับความต้องการของผู้ใช้ตามที่กำหนดมาโดยใช้ความรู้ที่ได้จากส่วนความรู้ในการค้นหาบริการ ซึ่งจะช่วยให้สามารถค้นหาบริการที่ตรงกับความต้องการได้ และยังรวมไปถึงการค้นหาบริการในเบื้องต้นเพื่อเป็นการตรวจสอบความเป็นไปได้ของความต้องการบริการนั้นๆ

6) ส่วนร่างสัญญา (Contracting module) เป็นส่วนที่ดำเนินการทางด้านกรร่างเอกสารสัญญาที่เกิดขึ้นระหว่างผู้เรียกใช้บริการและผู้ให้บริการที่อยู่ในระบบ ครอบคลุมทั้งในส่วนการร่างสัญญาฉบับร่างและการร่างสัญญาฉบับสมบูรณ์ และรวมถึงเอกสารอื่นๆที่เกี่ยวข้อง ซึ่งสามารถแสดงเอกสารทั้งหมดได้ดังตารางที่ 4.13

7) ส่วนวิเคราะห์คุณภาพของบริการ (QoS analysis module) เป็นส่วนเสริมสำหรับระบบ ซึ่งจะช่วยให้สามารถค้นหาบริการให้ตรงกับความต้องการมากยิ่งขึ้น โดยนำข้อมูลคุณภาพของบริการที่ได้รับมาจากทั้งผู้เรียกใช้บริการที่นำบริการไปใช้งานแล้วส่งผลตอบรับกลับมา และระบบผู้ประสานงานการต่อเองที่ดำเนินการทดสอบคุณภาพของบริการที่อยู่ในระบบมาจัดเก็บเป็นข้อมูลความรู้ควบคู่กับความรู้หลักที่มีอยู่ในระบบ ทำให้เพิ่มประสิทธิภาพในการค้นหาบริการ

ตารางที่ 4.13 เอกสารที่เกี่ยวข้องกับการเจรจาต่อรองในระบบ

เอกสาร	คำอธิบาย
ข้อกำหนดความต้องการบริการ (Requirement specification)	เอกสารระบุความต้องการบริการของผู้เรียกใช้บริการ
ข้อเสนอบริการ (Service proposal)	เอกสารเสนอรายละเอียดคุณลักษณะของบริการที่ผู้ให้บริการต้องการนำเสนอแก่ผู้เรียกใช้บริการ
ข้อเสนอบริการที่ตัดชื่อผู้ให้บริการออก (Anonymous proposal)	เอกสารเสนอรายละเอียดคุณลักษณะของบริการที่ระบบดำเนินการตัดชื่อและรายละเอียดของผู้ให้บริการนั้นออกไป
สัญญาฉบับร่าง (Draft Agreement)	เอกสารการทำสัญญาระหว่างผู้เรียกใช้บริการและผู้ให้บริการฉบับร่าง
สัญญาบริการ (Service Agreement)	เอกสารสัญญาระหว่างผู้เรียกใช้บริการและผู้ให้บริการที่ผ่านการยอมรับจากทั้งสองฝ่ายเรียบร้อยแล้ว

4.5 การแทนความรู้ (Knowledge Representation)

สำหรับสถาปัตยกรรมของระบบผู้ประสานงานการเจรจาต่อรองที่กล่าวมาในข้างต้น จะเห็นได้ว่าส่วนความรู้เป็นส่วนสำคัญที่ช่วยให้ระบบสามารถดำเนินการค้นหาบริการตามความต้องการได้อย่างมีประสิทธิภาพและเหมาะสม แต่อย่างไรก็ตามการสร้างความรู้ภายใต้บริบทเฉพาะหนึ่ง ๆ จะเป็นสิ่งที่ต้องพิจารณาถึงการนำไปประยุกต์ใช้งานในบริบทนั้นเป็นอย่างยิ่ง

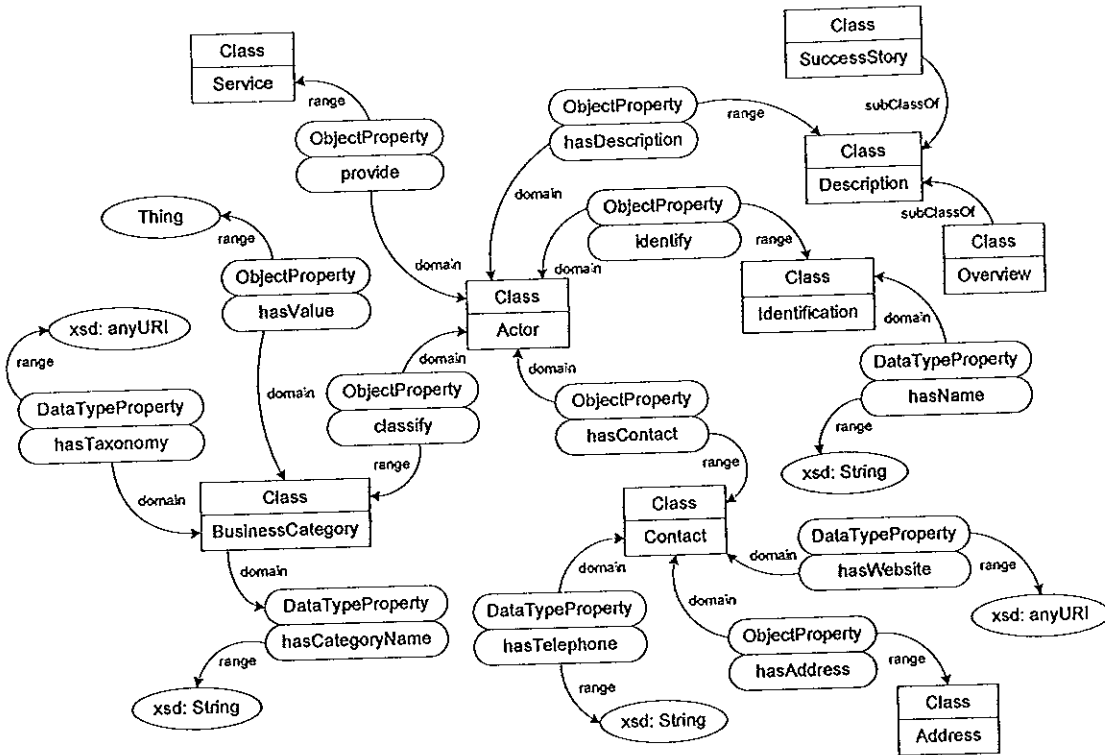
ความรู้ คือความเข้าใจในเรื่องบางเรื่องหรือสิ่งบางสิ่ง ซึ่งอาจจะรวมไปถึงความสามารถในการนำสิ่งนั้นไปใช้เพื่อเป้าหมายบางประการ ตัวอย่างเช่น ความรู้ของผู้เรียกใช้บริการอาจเป็น ชื่อของผู้เรียกใช้บริการ ที่อยู่ของผู้เรียกใช้บริการ หรือสามารถติดต่อผู้เรียกใช้บริการได้อย่างไร เป็นต้น ดังนั้นข้อมูลต่างๆที่เหมาะสมสามารถนำมาประยุกต์ใช้เป็นความรู้ได้

มนุษย์โดยทั่วไปอธิบายความรู้ของตนโดยใช้ภาษาเพราะภาษาเป็นสื่อกลางที่ใช้ในการติดต่อสื่อสารระหว่างกันของมนุษย์ เครื่องจักรก็เช่นกันจำเป็นต้องมีภาษาที่ทำให้สามารถนำความรู้ต่าง ๆ นั้นไปดำเนินงานได้ ดังนั้นการแทนความรู้ (Knowledge representation) ที่มีอยู่ให้อยู่ในรูปแบบที่เครื่องจักรสามารถนำไปใช้งานได้นั้นจะเป็นส่วนสำคัญ

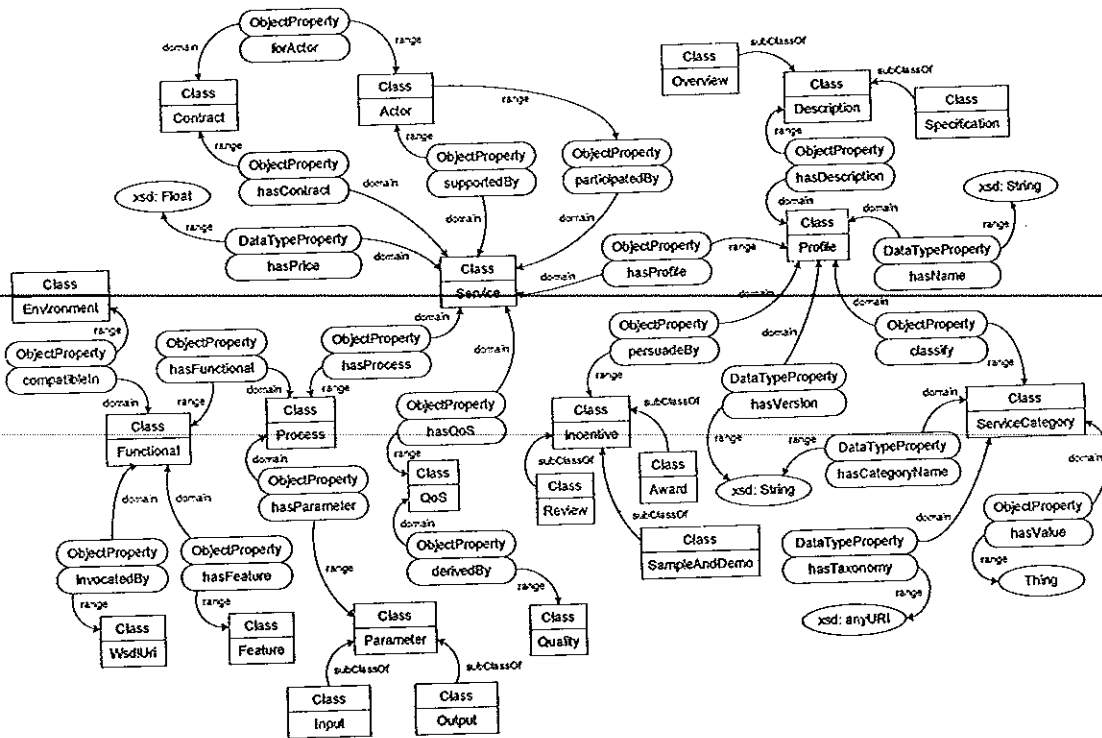
ในปัจจุบันนักวิจัยในสาขาปัญญาประดิษฐ์ (Artificial intelligent) ได้ค้นคว้าและวิจัยถึงภาษาที่จะนำมาแทนความรู้อย่างหลากหลาย และหนึ่งในนั้นที่เป็นที่นิยมใช้คือ การใช้ออนโทโลยี (Ontology) สำหรับการแทนความรู้ในรูปแบบสัญลักษณ์ที่แสดงถึงแนวคิดสำหรับบริบทเฉพาะหนึ่ง ๆ ซึ่งช่วยให้ทั้งมนุษย์และเครื่องจักรสามารถเข้าใจความรู้ในขอบเขตนั้นๆ ได้ตรงกัน งานวิจัยนี้จึงแทนความรู้ที่ใช้ในระบบผู้ประสานงานการเจรจาต่อรองด้วยออนโทโลยี

ออนโทโลยีความรู้ที่ออกแบบสำหรับใช้ในระบบนั้นประกอบด้วย 2 แบบหลักคือ

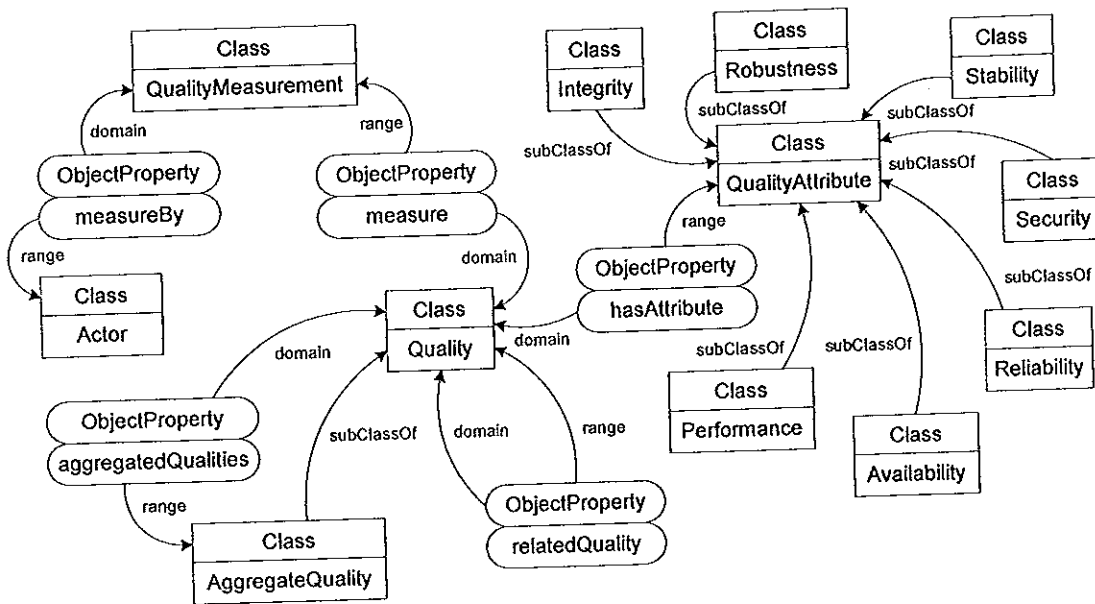
ออนโทโลยีของผู้กระทำ (Actor ontology) ที่เกี่ยวข้องกับระบบ โดยผู้เรียกใช้บริการและผู้ให้บริการ และออนโทโลยีบริการ (Service ontology) เป็นออนโทโลยีที่อธิบายแนวคิดความรู้ของบริการที่อยู่ในระบบ ส่วนออนโทโลยีที่ช่วยเสริมให้ออนโทโลยีบริการมีความสมบูรณ์แบบมากขึ้นคือออนโทโลยีคุณภาพบริการ (QoS ontology) โดยสามารถแสดงแผนภาพออนโทโลยีผู้ดำเนินการ ออนโทโลยีบริการ และออนโทโลยีคุณภาพบริการได้ดังภาพประกอบที่ 4.8, 4.9 และ 4.10 ตามลำดับ (Maximilien 2004)



ภาพประกอบที่ 4.8 แผนภาพแบบจำลองออนโทโลยีของผู้ดำเนินการที่มีส่วนเกี่ยวข้องในระบบ (ที่มา: Maximilien, 1996)



ภาพประกอบที่ 4.9 แผนภาพแบบจำลองออนโทโลยีของบริการ (ที่มา: Maximilien, 1996)



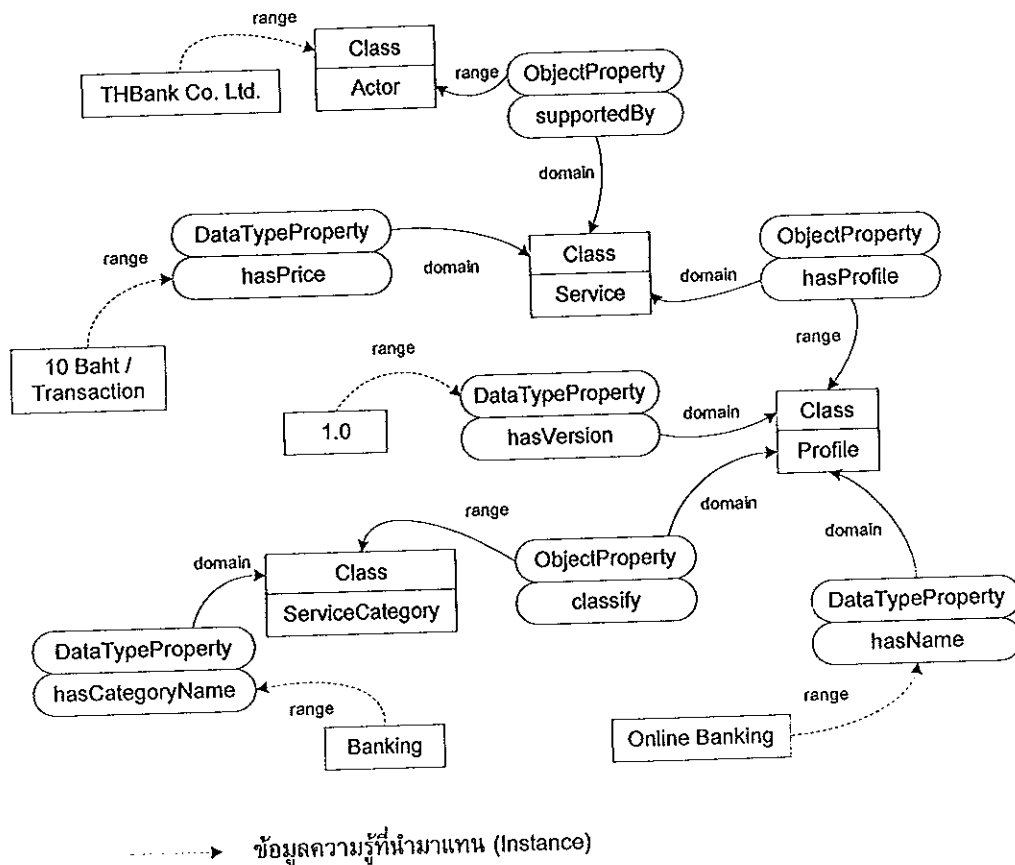
ภาพประกอบที่ 4.10 แผนภาพแบบจำลองออนโทโลยีของคุณภาพบริการ
(ที่มา: Maximilien, 1996)

ผู้วิจัยได้ออกแบบแผนภาพออนโทโลยีในข้างต้นโดยอ้างอิงตามภาษา OWL (Ontology web language) ซึ่งจะประกอบด้วยสัญลักษณ์ที่สำคัญเพิ่มเติมได้แก่

- โดเมน (Domain) เป็นการอธิบายขอบเขตของแนวคิดที่สนใจ เช่น โดเมนของบริการหมายถึงแนวคิดที่สนใจอยู่ในขอบเขตของบริการเท่านั้น
- เรนจ์ (Range) เป็นการอธิบายระยะของเขตที่โดเมนจะครอบคลุมถึง เช่น โดเมนของบริการมีเรนจ์คือราคา แสดงว่าแนวคิดที่สนใจอยู่ในขอบเขตของบริการและสนใจไปยังเรื่องราคาของบริการ
- ObjectProperty คือการบอกความสัมพันธ์ระหว่างสองแนวคิด
- DatatypeProperty คือการบอกความสัมพันธ์ระหว่างแนวคิดและชนิดของข้อมูลที่แนวคิดนั้นจะเป็นได้

ระบบผู้ประสานงานการเจรจาต่อรองสามารถประยุกต์ใช้งานออนโทโลยีดังกล่าวในการแทนความรู้จากกลุ่มข้อมูลที่มีอยู่ในระบบ ซึ่งสามารถแสดงตัวอย่างได้เช่น ข้อมูลบริการคือบริการธนาคารออนไลน์ (Online banking) รุ่นของบริการเป็น 1.0 อยู่ในกลุ่มบริการประเภทธุรกรรมทางธนาคาร (Banking) โดยมีบริษัท THBank เป็นผู้ให้บริการ และราคา 10 บาทต่อหนึ่งครั้งการเรียกใช้บริการ ดังนั้นสามารถแทนข้อมูลที่สอดคล้องกับออนโทโลยีความรู้

ของบริการได้ดังภาพประกอบที่ 4.11 และนอกจากนั้นถ้ามีข้อมูลที่เพิ่มเติมนอกเหนือจากที่กำหนดก็สามารถแทนข้อมูลเพิ่มเติมได้แต่จะต้องเป็นไปตามขอบเขตความรู้ที่ออนไลน์กำหนดไว้



ภาพประกอบที่ 4.11 ตัวอย่างการแทนความรู้ของบริการธนาคารออนไลน์

ที่เป็นไปตามออนไลน์ของบริการ

4.6 สรุปท้ายบท

ระบบผู้ประสานงานการเจรจาต่อรองได้รับการออกแบบให้มีการดำเนินงานของระบบใน 7 ส่วน คือ ส่วนนำเสนอ ส่วนลงทะเบียนสมาชิก ส่วนดึงข้อมูล ส่วนความรู้ ส่วนคัดเลือกบริการ ส่วนร่างสัญญา และส่วนวิเคราะห์คุณภาพของบริการ นอกจากนี้ยังได้ออกแบบวิธีการแทนความรู้จากข้อมูลที่มีอยู่ในระบบโดยใช้เทคโนโลยีออนไลน์ ดันแบบระบบนี้จะนำไปพัฒนาและทดสอบประเมินผลการทำงานในส่วนถัดไป

บทที่ 5

การพัฒนาระบบและกรณีตัวอย่าง

ในบทนี้กล่าวถึงการพัฒนาระบบผู้ประสานงานการเจรจาต่อรองซึ่งจะช่วยสนับสนุนการดำเนินการเจรจาต่อรองระหว่างกันของผู้เรียกใช้บริการและผู้ให้บริการผ่านตัวกลาง และการพัฒนาระบบตัวอย่างกรณีศึกษาสำหรับนำมาทดสอบความถูกต้องสมบูรณ์ในการดำเนินงานของระบบผู้ประสานงานการเจรจาต่อรองดังกล่าว

5.1 การพัฒนาระบบผู้ประสานงานการเจรจาต่อรอง

ระบบผู้ประสานงานการเจรจาต่อรองเป็นตัวกลางที่คอยสนับสนุนการเจรจาต่อรองระหว่างผู้เรียกใช้บริการและผู้ให้บริการ โดยมีส่วนติดต่อผู้ใช้เป็นโปรแกรมประยุกต์ผ่านเว็บ ซึ่งมีรายละเอียดดังนี้

5.1.1 เครื่องมือที่ใช้ในการพัฒนา

เครื่องมือต่าง ๆ ที่ใช้ในการพัฒนาระบบผู้ประสานงานการเจรจาต่อรองประกอบด้วย

1. Microsoft Windows XP สำหรับเป็นระบบปฏิบัติการ
2. NetBeans IDE สำหรับเป็นเครื่องมือช่วยพัฒนาโปรแกรมด้วยภาษาจาวา
3. Glassfish Application Server เพื่อให้บริการโปรแกรมประยุกต์ผ่านเว็บ
4. MySQL 5.1 สำหรับใช้เป็นฐานข้อมูลเชิงสัมพันธ์
5. Java API Library ต่างๆ ได้แก่
 - MySQL JDBC Connector API สำหรับการเชื่อมต่อระบบจัดการฐานข้อมูลด้วยภาษาจาวา รวมถึงการประยุกต์เพื่อจัดเก็บข้อมูล XML
 - JavaServer Page (JSP) และ Java Servlet API สำหรับการเขียนโปรแกรมประยุกต์ผ่านเว็บด้วยภาษาจาวา
 - Java API for XML สำหรับการสร้าง การลบและการสกัดข้อมูลจาก เอ็กซ์เอ็มแอล

- Javazoom Upload Bean API สำหรับการอัปโหลดไฟล์ขึ้นสู่เว็บ
 - Apache Commons เพื่อเป็นชุดคำสั่งที่ช่วยสนับสนุนทำงานต่าง ๆ เช่น การจัดการเกี่ยวกับข้อความ การจัดการเรื่องเครือข่าย เป็นต้น
6. jQuery เป็นกลุ่มชุดคำสั่งของภาษาจาวาสคริปต์ เพื่อช่วยในการพัฒนาเว็บ

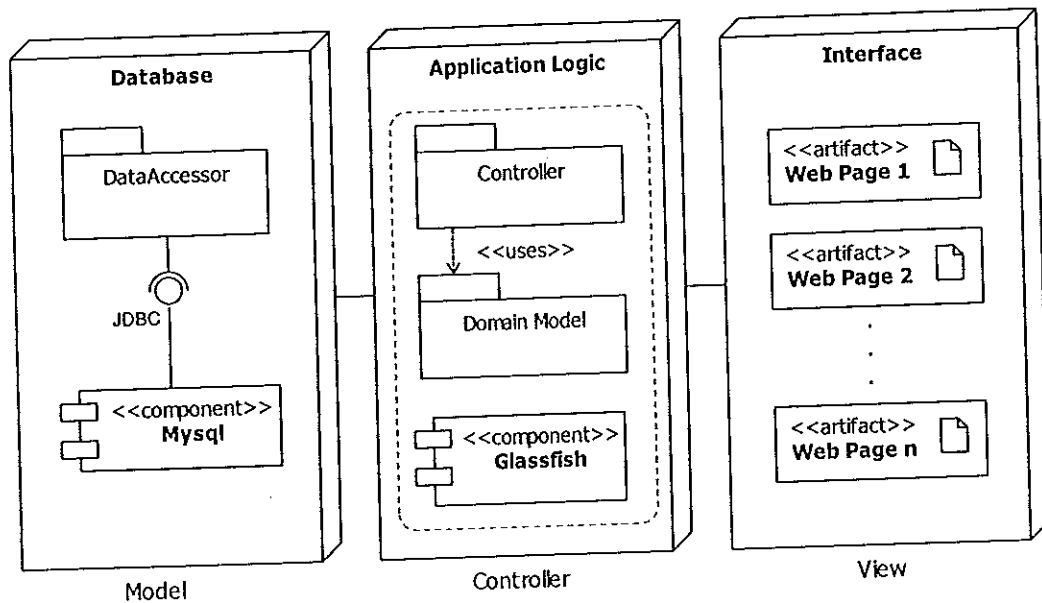
5.1.2 การพัฒนาส่วนดำเนินการ

การพัฒนาส่วนดำเนินการของระบบผู้ประสานงานการเจรจาต่อรองจะประกอบด้วย 3 ส่วนหลักได้แก่ ส่วนแสดงผล ส่วนการประมวลผล และส่วนข้อมูล ซึ่งเห็นได้ว่าสอดคล้องกับแบบแผนสำหรับการออกแบบระบบซอฟต์แวร์ (Design pattern) ที่มีชื่อว่า Model-View-Controller หรือ MVC (Reenskaug, 2003)

Model-View-Controller เป็นรูปแบบสำหรับการออกแบบที่ดำเนินการแยกส่วนดำเนินการของโปรแกรม ที่ประกอบด้วยส่วนจัดการการสร้าง สืบค้น แก้ไข และลบข้อมูลที่เกี่ยวข้อง และส่วนการทำงานของระบบออกมาจากส่วนแสดงผล ช่วยทำให้มีความอิสระในการพัฒนา การดูแลจัดการ และการทดสอบระบบที่พัฒนา ก่อให้เกิดการเกาะเกี่ยวกันระหว่างส่วนดำเนินงานที่น้อยลงหรือ loosely coupling

ระบบผู้ประสานงานการเจรจาต่อรองมีการแยกส่วนดำเนินงานได้ดังภาพประกอบที่ 5.1 และมีรายละเอียดดังนี้

1) ส่วนข้อมูล เป็นส่วนที่เกี่ยวข้องกับการดำเนินการกับข้อมูลไม่ว่าจะเป็นการจัดเก็บ การค้นคืน การแก้ไข หรือการลบข้อมูล โดยทั่วไปแล้วสำหรับการพัฒนาระบบซอฟต์แวร์ส่วนข้อมูลอาจจะเป็นได้ทั้งระบบจัดการฐานข้อมูลหรือระบบจัดการไฟล์ และจะมีส่วนหนึ่งของโปรแกรมคอยทำหน้าที่ช่วยสนับสนุนการติดต่อระหว่างการดำเนินการที่เหลือของระบบกับส่วนที่จัดการกับข้อมูล ตัวอย่างเช่น เมื่อต้องการพัฒนาโปรแกรมด้วยภาษาจาวาเพื่อติดต่อกับระบบจัดการฐานข้อมูลต่างๆ สามารถกระทำได้ผ่านทางชุดคำสั่งสำหรับการติดต่อระบบจัดการฐานข้อมูลที่เรียกว่า Java Database Connectivity (JDBC) (JDBC 2010) เป็นต้น

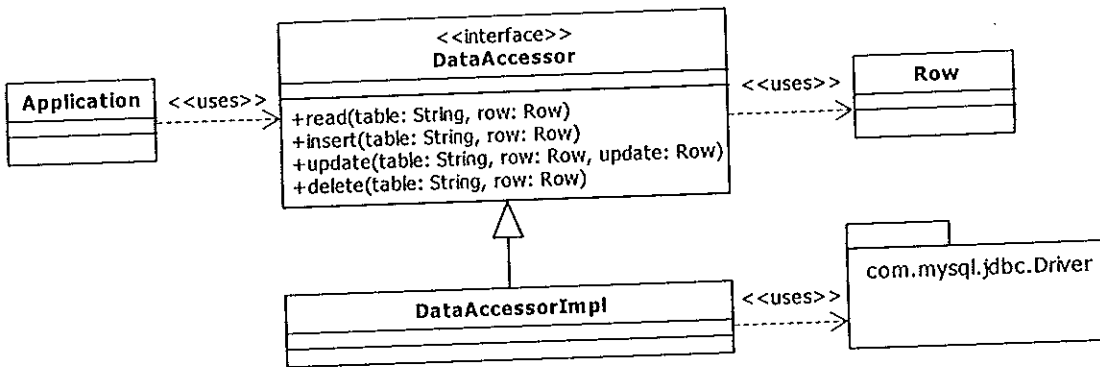


ภาพประกอบที่ 5.1 ส่วนการดำเนินงานของระบบผู้ประสานงานการเจรจาต่อรองที่สอดคล้องกับแบบแผนการออกแบบระบบซอฟต์แวร์ Model-View-Controller (MVC)

งานวิจัยนี้ได้เลือกระบบจัดการฐานข้อมูล MySQL เป็นส่วนข้อมูลสำหรับระบบ และได้ออกแบบส่วนการติดต่อกับระบบจัดการฐานข้อมูลนี้โดยใช้แบบแผนการเข้าถึงส่วนข้อมูล (Data Access Patterns) ที่ชื่อว่า Data Accessor (Nock, 2003)

Data Accessor ช่วยให้ระบบซอฟต์แวร์แยกส่วนการดำเนินการออกจากส่วนจัดการข้อมูลได้อย่างชัดเจน โดยปกปิดส่วนการทำงานที่เกี่ยวข้องกับการจัดการข้อมูลเอาไว้ภายในส่วนประกอบเดียวและเปิดเผยเฉพาะส่วนทำงานที่จำเป็น ส่งผลให้เกิดความยืดหยุ่นในการพัฒนาโปรแกรม เพราะสามารถปรับเปลี่ยนระบบจัดการข้อมูลเป็นแบบอื่น ๆ ได้โดยไม่ต้องแก้ไขหรือจัดการส่วนดำเนินงานหลักของระบบ

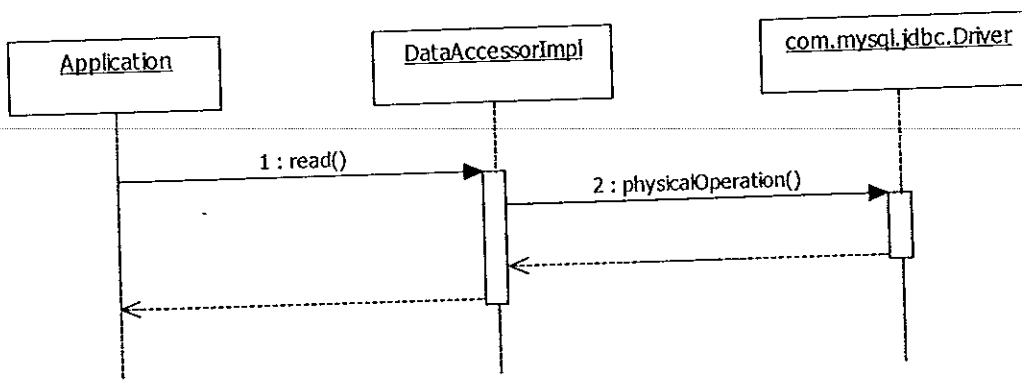
โครงสร้างของแบบแผนการเข้าถึงส่วนข้อมูล Data Accessor สำหรับระบบสามารถแสดงได้ดังภาพประกอบที่ 5.2



ภาพประกอบที่ 5.2 แผนภาพคลาสของแบบแผนการเข้าถึงส่วนข้อมูล Data Accessor

อินเทอร์เฟซ DataAccessor ทำหน้าที่ระบุการดำเนินการสำหรับการเข้าถึงระบบจัดการข้อมูล โดยประกอบด้วยกระบวนการพื้นฐานที่จำเป็น ได้แก่ การอ่านข้อมูล การเพิ่มข้อมูล การเปลี่ยนแปลงข้อมูล และการลบข้อมูล ส่วนคลาส DataAccessorImpl เป็นคลาสที่นำอินเทอร์เฟซดังกล่าวไปพัฒนา สำหรับในงานวิจัยนี้ได้พัฒนาให้ติดต่อกับระบบจัดการฐานข้อมูล MySQL ผ่านทางส่วนติดต่อ JDBC นอกจากนี้ยังสามารถเพิ่มการติดต่อกับระบบจัดการข้อมูลได้ตามแต่ต้องการเพียงแต่พัฒนาให้มีส่วนดำเนินการตามอินเทอร์เฟซ DataAccessor เท่านั้น ซึ่งจะช่วยให้ระบบสามารถสลับเปลี่ยนส่วนจัดการข้อมูลที่ใช้งานได้ทันที

การเรียกใช้งานจะกระทำผ่านอินเทอร์เฟซ DataAccessor ซึ่งจะไปเรียกส่วน DataAccessorImpl ที่ติดต่อกับฐานข้อมูล MySQL มากกระทำอีกทีหนึ่ง ทั้งนี้การทำงานต่างๆจะต้องระบุข้อมูลที่จำเป็นตามโครงสร้างที่กำหนดไว้ในคลาส Row ลำดับขั้นตอนการดำเนินงานดังกล่าวสามารถแสดงได้ในภาพประกอบที่ 5.3 ซึ่งเป็นตัวอย่างการเรียกใช้งานการอ่านข้อมูลจากระบบจัดการฐานข้อมูล MySQL ผ่านทาง Data Accessor



ภาพประกอบที่ 5.3 แผนภาพลำดับการทำงานของการทำงานของการอ่านข้อมูลจากระบบจัดการฐานข้อมูลผ่านทาง Data Accessor

2) ส่วนประมวลผล เป็นส่วนที่ดำเนินการประมวลผลสำหรับการดำเนินการต่างๆที่ระบบพึงมี กล่าวคือการทำงานหลักที่ระบบสามารถกระทำได้นั้นเอง ส่วนประมวลผลนี้จะรับข้อมูลเข้าจากผู้ใช้งานทางส่วนแสดงผล เริ่มกระบวนการด้วยข้อมูลเหล่านั้นจึงได้ผลลัพธ์และแสดงผลกลับสู่ส่วนแสดงผลเพื่อให้ผู้ใช้รับทราบต่อไป ในระหว่างกระบวนการอาจจะต้องมีการติดต่อกับส่วนข้อมูลเพื่อดำเนินการต่างๆ

การพัฒนาระบบด้วยแนวคิดเชิงวัตถุสามารถแยกองค์ประกอบที่สำคัญได้ 2

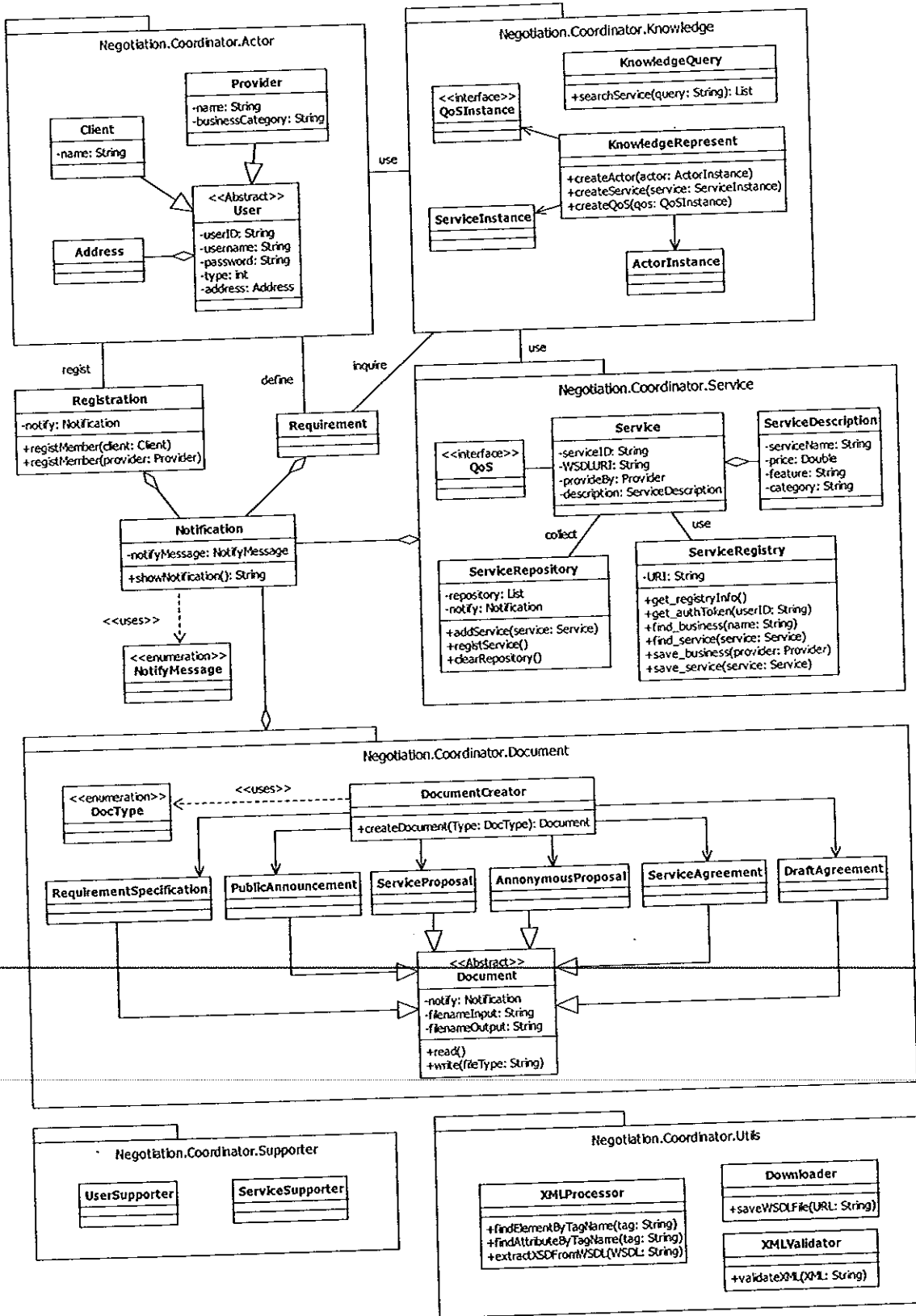
ส่วนหลักคือ

- Domain model หรือ Domain object model เป็นส่วนที่แสดงถึงโครงสร้างหรือองค์ประกอบที่พึงมีทั้งหมดในระบบ ซึ่งรวมถึงความสัมพันธ์ระหว่างกันด้วย โดยทั่วไปแล้วในระบบที่ใช้ระบบจัดการฐานข้อมูล Domain model ของมันมักจะสอดคล้องหรือเป็นไปตามฐานข้อมูลที่จัดเก็บ กล่าวคือมีคลาสต่างๆตรงกับตารางในฐานข้อมูล และมีคุณลักษณะของคลาสตรงกับฟิลด์ที่บรรจุในตารางนั้นในฐานข้อมูล รวมไปถึงความสัมพันธ์ระหว่างคลาสดรตรงกับความสัมพันธ์ระหว่างตารางในฐานข้อมูลด้วย
- ส่วนควบคุมการทำงาน เป็นส่วนที่ทำงานโดยนำ Domain model ที่ได้พัฒนาไปแล้วมาประกอบเพื่อทำงานร่วมกันจนได้ผลลัพธ์ตามที่ต้องการ นอกจากนี้อาจจะเพิ่มการดำเนินงานอื่นๆที่นอกเหนือที่ Domain model มีอยู่ได้

การพัฒนาส่วน Domain model ของระบบผู้ประสานงานการเจรจาต่อรองมีโครงสร้างต่างๆดังแสดงในภาพประกอบที่ 5.4 ซึ่งประกอบด้วย 5 ส่วนหลักโดยมีรายละเอียดของโครงสร้างในแต่ละส่วนดังนี้

(1) Negotiation.Coordinator.Service สำหรับจัดเก็บคลาสที่เกี่ยวข้องกับการดำเนินงานกับบริการ โดยมีคลาสต่างๆดังนี้

- Service เป็นคลาสของข้อมูลของบริการเชิงการทำงาน (Functional attribute) เช่น ตำแหน่งของไฟล์ WSDL เป็นต้น
- ServiceDescription เป็นคลาสของข้อมูลของบริการที่ไม่ใช่การทำงาน เช่น ชื่อบริการ หมวดหมู่ของบริการ ราคา เป็นต้น โดยจะสัมพันธ์กับคลาส Service
- ServiceRepository สำหรับการจัดเก็บข้อมูลบริการต่างๆลงฐานข้อมูล



ภาพประกอบที่ 5.4 แผนภาพคลาสของระบบผู้ประสานงานการเจรจาต่อรอง

- ServiceRegistry สำหรับจัดการหน่วยลงทะเบียนบริการ ได้แก่ การค้นคืนและจัดเก็บบริการ สามารถอธิบายรายละเอียดได้ในลำดับถัดไป
- QoS สำหรับการคำนวณคุณภาพบริการ

(2) Neogotiation.Coordinator.Knowledge สำหรับการจัดการความรู้ที่เกี่ยวข้องกับระบบนั้นก็คือ บริการ ผู้เรียกใช้บริการและผู้ให้บริการ โดยมีคลาสต่างๆดังนี้

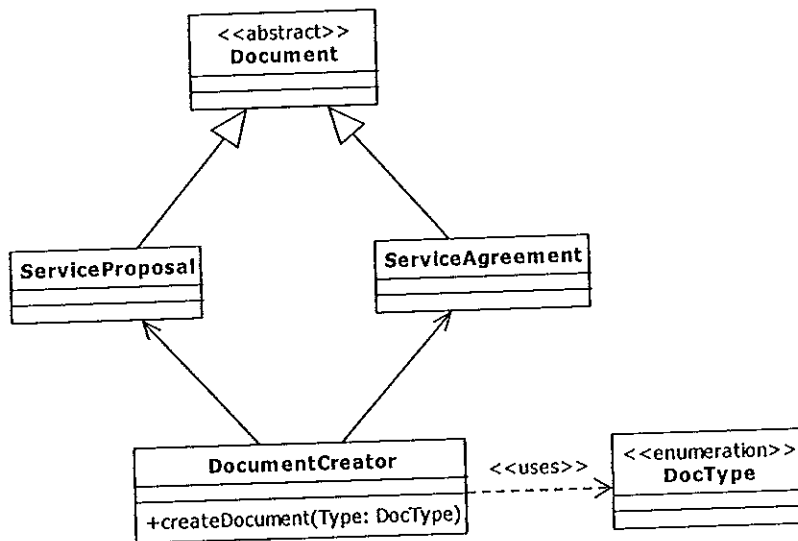
- KnowledgeRepresent สำหรับการแทนความรู้จากข้อมูลที่จัดเก็บอยู่ในระบบ ซึ่งสามารถอธิบายรายละเอียดของการพัฒนาการแทนความรู้ได้ในลำดับถัดไป
- ActorInstance, ServiceInstance และ QoSInstance เป็นคลาสของความรู้ในระบบ ได้แก่ ผู้มีส่วนเกี่ยวข้อง (คือผู้ให้บริการและผู้เรียกใช้บริการ) บริการ และคุณภาพของบริการตามลำดับ
- KnowledgeQuery สำหรับการค้นคืนความรู้จากฐานความรู้

(3) Negotiation.Coordinator.Document สำหรับจัดการการสร้างเอกสารต่างๆที่เกี่ยวข้องได้แก่ ข้อกำหนดความต้องการบริการ (Requirement specification) ประกาศข้อกำหนดความต้องการแก่สาธารณะ (Public announcement) ข้อเสนอบริการ (Service proposal) ข้อเสนอบริการที่ไม่มีการระบุตัวผู้ให้บริการ (Anonymous proposal) ร่างข้อตกลงบริการ (Draft agreement) และข้อตกลงบริการ (Service agreement)

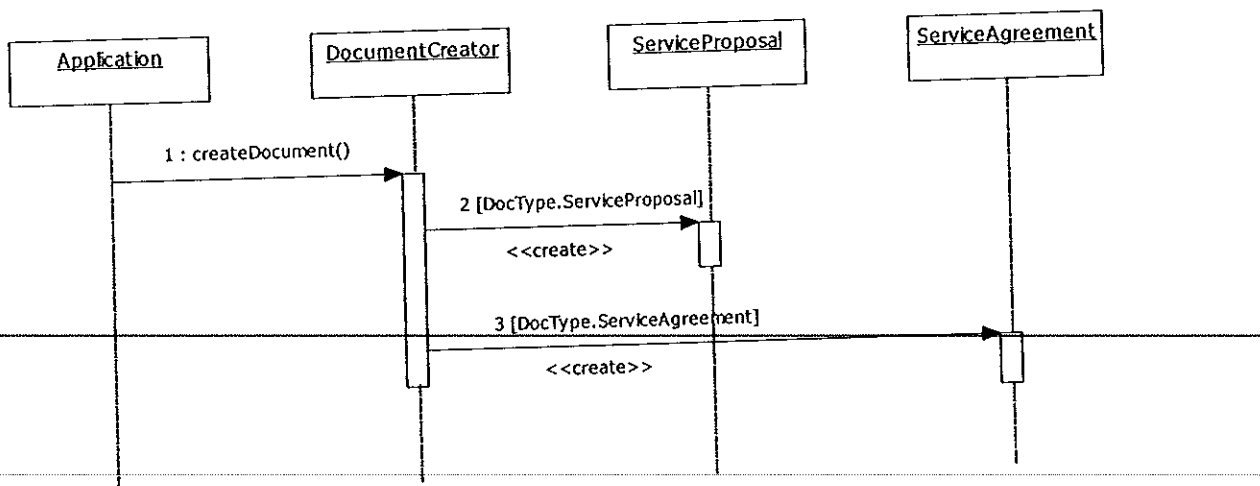
งานวิจัยนี้ได้เลือกแบบแผนการออกแบบ (Design pattern) ที่ชื่อว่า Factory Method (Gamma 1994) มาประยุกต์ใช้งานสำหรับการดำเนินงานในส่วนนี้ เนื่องจากช่วยเพิ่มความยืดหยุ่นต่อการปรับเปลี่ยนหรือเพิ่มเติมเอกสารต่างๆได้โดยง่าย

แนวคิดของ Factory Method คือการสร้างวัตถุพิเศษตัวหนึ่งทำหน้าที่เสมือนโรงงานสำหรับการสร้างวัตถุอื่นๆในกลุ่มขอบเขตเดียวกัน ดังนั้นการสร้างเอกสารต่างๆที่เกี่ยวข้องกับระบบดังที่กล่าวมาก็สามารถกระทำได้ผ่านวัตถุพิเศษเช่นกัน นั่นก็คือคลาส DocumentCreator และเอกสารทั้งหมดจะถูกกำหนดภายใต้ขอบเขตของคลาส Document ซึ่งประกอบด้วยวิธีการอ่านเอกสารในรูปแบบต่างๆ และการเขียนเอกสารจากข้อมูลตามที่กำหนด องค์ประกอบเชิงโครงสร้างของ Factory Method สามารถแสดงได้ดังภาพประกอบที่ 5.5

ภาพประกอบที่ 5.5 เป็นตัวอย่างการสร้างเอกสารข้อเสนอบริการและข้อตกลงบริการผ่านทางคลาส DocumentCreator ที่เพียงแต่เรียกใช้เมธอด createDocument และกำหนดชนิดของเอกสารที่ต้องการตามที่กำหนดไว้ใน DocType เป็นพารามิเตอร์ ลำดับขั้นตอนการดำเนินงานดังกล่าวสามารถแสดงได้ในภาพประกอบที่ 5.6



ภาพประกอบที่ 5.5 ตัวอย่างแผนภาพคลาสแสดงองค์ประกอบ
เชิงโครงสร้างของ Factory Method



ภาพประกอบที่ 5.6 แผนภาพลำดับการทำงานของการสร้างเอกสารผ่านทางคลาส
DocumentCreator ที่เป็นไปตามแบบแผนการออกแบบ Factory Method

(4) Negotiation.Coordinator.Supporter สำหรับจัดเก็บคลาสที่คอยสนับสนุนการดำเนินงานที่ถูกเรียกใช้งานบ่อยครั้ง เช่น

- UserSupporter ประกอบด้วยการทำงานที่สนับสนุนการจัดการผู้ใช้ในระบบ ตัวอย่างเช่น การค้นหารหัสผ่านจากชื่อผู้ใช้งานหรือการค้นหาประเภทของผู้ใช้จากชื่อผู้ใช้งาน เป็นต้น
- ServiceSupporter ประกอบด้วยการทำงานที่สนับสนุนการจัดการบริการในระบบ ตัวอย่างเช่น การค้นหาบริการจากรหัสของบริการนั้น เป็นต้น

(5) Negotiation.Coordinator.Utils สำหรับจัดเก็บคลาสที่สนับสนุนการทำงานทั่วไป เช่น

- XMLProcessor สำหรับการดำเนินงานกับไฟล์เอกสาร XML ซึ่งรวมไปถึงไฟล์เอกสาร WSDL
- XMLValidator สำหรับการตรวจสอบความถูกต้องของไวยากรณ์ของเอกสาร XML
- Downloader สำหรับการดาวน์โหลดไฟล์ต่างที่ปรากฏบนอินเทอร์เน็ตมาเก็บไว้ในระบบ

(6) Negotiation.Coordinator.Actor สำหรับจัดเก็บคลาสข้อมูลของผู้ใช้ในระบบ ซึ่งประกอบด้วยผู้ให้บริการและผู้เรียกใช้บริการ มีคลาสดังนี้

- User เป็นคลาสข้อมูลทั่วไปของผู้ใช้
- Provider เป็นคลาสข้อมูลของผู้ให้บริการ
- Client เป็นคลาสข้อมูลของผู้เรียกใช้บริการ
- Address เป็นคลาสข้อมูลของสถานที่ที่ติดต่อได้ของผู้ใช้ ซึ่งจะสัมพันธ์กับคลาส User

(7) Registration เป็นคลาสสำหรับการลงทะเบียนเป็นสมาชิกของผู้ใช้ โดยต้องกำหนดข้อมูลต่างๆของผู้ใช้ในรูปคลาสตามที่ปรากฏใน Negotiation.Coordinator.Actor ให้เหมาะสม

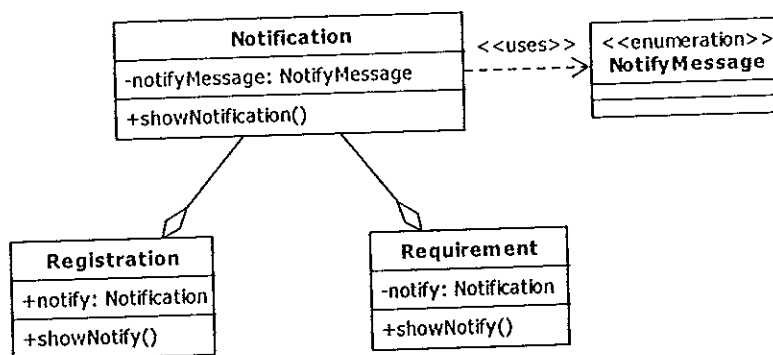
(8) Requirement เป็นคลาสสำหรับการกำหนดความต้องการบริการของผู้เรียกใช้บริการ

(9) Notification เป็นคลาสสำหรับการจัดการข่าวสารการแจ้งประกาศต่างๆที่เกิดขึ้นในระบบ ได้แก่ ข่าวสารการระบุบริการจากผู้ให้บริการ ข่าวสารการตอบรับการประกาศความต้องการสู่สาธารณะของผู้เรียกใช้บริการ ข่าวสารการแจ้งให้ผู้ให้บริการพร้อมสำหรับการเสนอบริการ ข่าวสารแจ้งให้ผู้ให้บริการและผู้เรียกใช้บริการพร้อมสำหรับการทำสัญญาระหว่าง

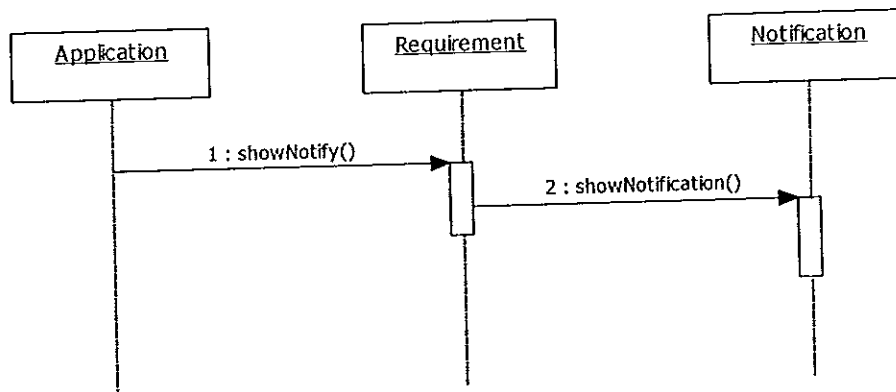
กัน ข่าวสารการแจ้งยกเลิกสัญญาฉบับร่างจากผู้เรียกใช้บริการ และข่าวสารการแจ้งให้พร้อม สำหรับการส่งมอบและตอบรับบริการจากผู้ให้บริการและผู้เรียกใช้บริการตามลำดับ

ในการพัฒนาส่วนนี้ได้เลือกแบบแผนการออกแบบที่ชื่อว่า Adapter (Gamma 1994) มาประยุกต์ใช้งาน เพราะช่วยสนับสนุนให้คลาสต่างๆมีขีดความสามารถในการแสดงหรือแจ้งข้อความข่าวสารโดยที่ไม่ต้องเพิ่มการดำเนินงานใดๆลงในคลาสเหล่านั้น

แนวคิดของ Adapter ใช้สำหรับการแปลงคลาสหนึ่งๆให้มีความสามารถ เหมือนกับคลาสอื่นโดยคลาสเหล่านี้ไม่ได้มีความเกี่ยวข้องสัมพันธ์กัน ทำให้สามารถติดต่อหรือ ทำงานร่วมกันได้ องค์ประกอบเชิงโครงสร้างของ Adapter สามารถแสดงได้ดังภาพประกอบที่ 5.7 ซึ่งเป็นตัวอย่างการแปลงความสามารถของคลาส Registration และคลาส Requirement ให้มีความสามารถในการแสดงข่าวสารตามที่คลาส Notification กระทำได้ ดังนั้นเมื่อมีการเรียกใช้งานคลาส Registration หรือ Requirement ก็สามารถเรียกการแสดงผลข่าวสารผ่านทางคุณลักษณะของคลาสนั้นๆได้เลย ลำดับขั้นตอนการดำเนินงานดังกล่าวสามารถแสดงได้ใน ภาพประกอบที่ 5.8



ภาพประกอบที่ 5.7 ตัวอย่างแผนภาพคลาสแสดงองค์ประกอบเชิงโครงสร้างของ Adapter



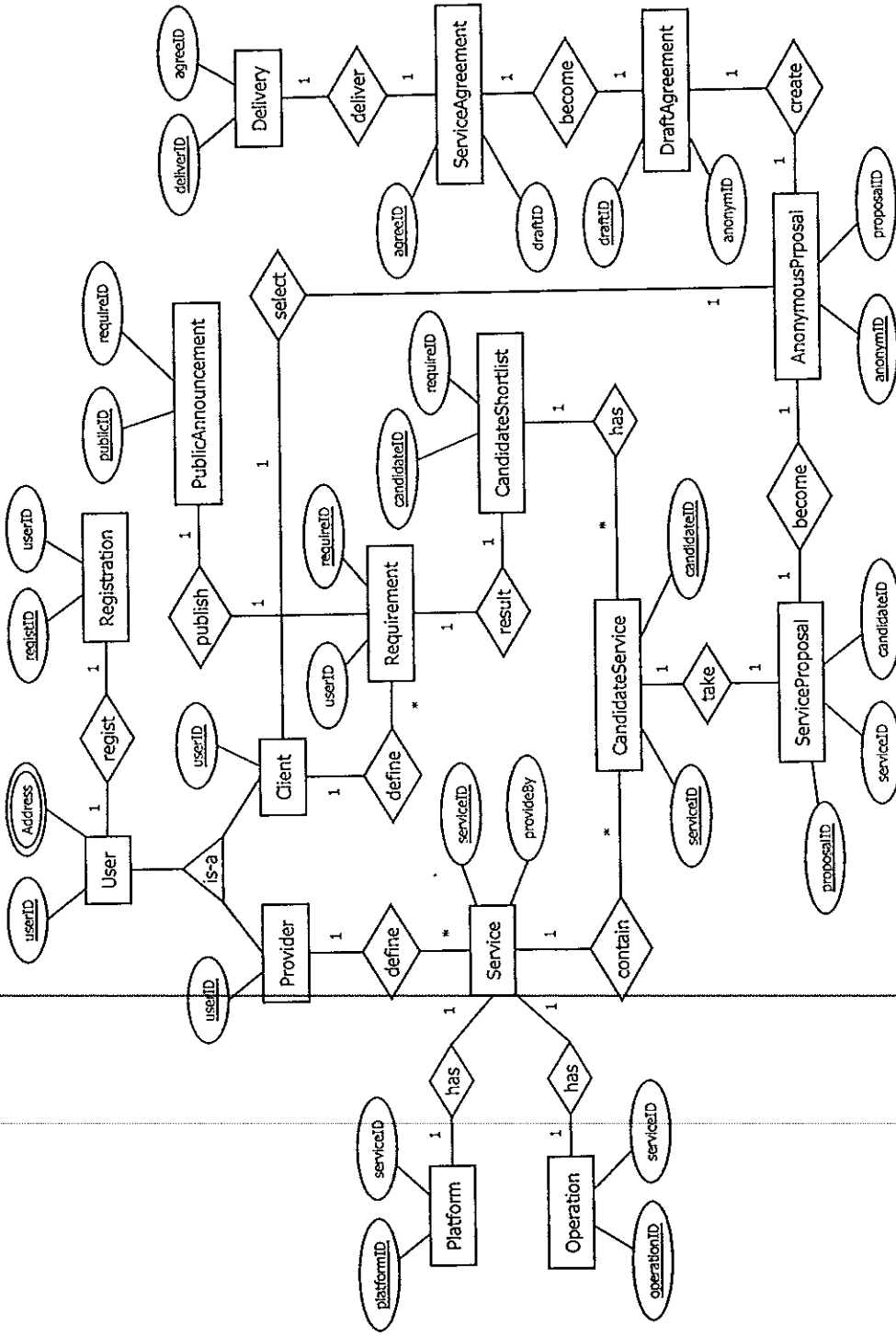
ภาพประกอบที่ 5.8 แผนภาพลำดับการทำงานของการทำงานของการเรียกใช้ความสามารถของคลาส Notification ผ่านทางคลาส Requirement ที่เป็นไปตามแบบแผนการออกแบบ Adapter

การพัฒนาส่วนควบคุมการทำงานของระบบด้วยภาษาจาวาจะเลือกใช้เทคโนโลยี Java Servlet ซึ่งได้รับความนิยม Servlet เป็นกลุ่มคลาสของภาษาจาวาที่ทำงานสัมพันธ์กับ Java Servlet API สนับสนุนการทำงานตอบสนองกับโปรโตคอล HTTP ทั้งการร้องขอและการตอบรับ ดังนั้น Servlet จึงเหมาะสำหรับการพัฒนาส่วนควบคุมที่จะต้องสนับสนุนให้เกิดทำงานร่วมกันระหว่างเว็บและเทคโนโลยีจาวา

3) ส่วนแสดงผล เป็นส่วนสำหรับติดต่อผู้ใช้โดยพัฒนาในรูปแบบโปรแกรมประยุกต์ผ่านเว็บที่ใช้ภาษา HTML JavaServer Page (JSP) และ Javascript ในการพัฒนาเพื่อให้ผู้ใช้สามารถเข้าถึงระบบผู้ประสานงานการเจรจาต่อรองได้โดยสะดวกและสามารถเรียกใช้งานต่างๆที่ระบบจัดให้ได้ทันที การพัฒนาเว็บได้จัดแบ่งตามการใช้งานสำหรับผู้ใช้ 2 กลุ่ม ได้แก่ ผู้เรียกใช้บริการและผู้ให้บริการ โดยทั้งสองกลุ่มจะต้องสมัครลงทะเบียนเข้าเป็นสมาชิกของระบบเสียก่อนจึงสามารถดำเนินงานอื่นๆต่อไปได้ โครงสร้างเว็บ รูปแบบ และการใช้งานจะกล่าวในหัวข้อถัดไป

5.1.3 การพัฒนาฐานข้อมูลของระบบ

โครงสร้างฐานข้อมูลที่ใช้ในระบบผู้ประสานงานการเจรจาต่อรองสามารถอธิบายด้วยแผนภาพอี-อาร์ (Entity-Relationship (E-R) diagram) ดังภาพประกอบที่ 5.9 และมีรายละเอียดระดับพจนานุกรมข้อมูลทั้งหมด 16 ตาราง ดังตารางที่ 5.1 – 5.17



ภาพประกอบที่ 5.9 แผนภาพแสดงความสัมพันธ์ทางข้อมูลในฐานะข้อมูลของระบบผู้ประสานงานการเจรจาต่อรอง

ตารางที่ 5.1 ตารางข้อมูลรายละเอียดผู้ใช้ (user)

เขตข้อมูล	คำอธิบาย	ประเภทข้อมูล	หมายเหตุ
userID	รหัสผู้ใช้	int(5)	Primary key
username	ชื่อเข้าใช้งาน	varchar(50)	
password	รหัสผ่าน	varchar(80)	
type	ประเภทของผู้ใช้	int(1)	1 คือประเภทผู้เรียกใช้บริการ 2 คือประเภทผู้ให้บริการ
pubDate	วันที่สมัครสมาชิก	date	
pubTime	เวลาที่สมัครสมาชิก	time	

ตารางที่ 5.2 ตารางข้อมูลรายละเอียดผู้เรียกใช้บริการ (client)

เขตข้อมูล	คำอธิบาย	ประเภทข้อมูล	หมายเหตุ
userID	รหัสผู้ใช้	int(5)	Primary key
name	ชื่อผู้เรียกใช้บริการ	varchar(80)	
email	อีเมลผู้เรียกใช้บริการ	varchar(50)	
description	รายละเอียดส่วนตัวอื่นๆ	varchar(200)	

ตารางที่ 5.3 ตารางข้อมูลรายละเอียดผู้ให้บริการ (provider)

เขตข้อมูล	คำอธิบาย	ประเภทข้อมูล	หมายเหตุ
userID	รหัสผู้ใช้	int(5)	Primary key
name	ชื่อผู้ให้บริการ	varchar(80)	
email	อีเมลผู้ให้บริการ	varchar(50)	
description	รายละเอียดส่วนตัวอื่นๆ	varchar(200)	
overview	คำอธิบายของธุรกิจโดยคร่าว	varchar(200)	
supportEmail	อีเมลฝ่ายสนับสนุน	varchar(50)	
supportPhone	หมายเลขโทรศัพท์ฝ่ายสนับสนุน	varchar(30)	
supportWebsite	เว็บไซต์ฝ่ายสนับสนุน	varchar(80)	
reseller	ชื่อตัวแทนจำหน่าย	varchar(150)	
referenceCustomer	ชื่อลูกค้าที่อ้างอิงได้	varchar(100)	จัดเก็บแต่ละคำค้นด้วยเครื่องหมายอัฒภาค (;)

ตารางที่ 5.3 (ต่อ)

เขตข้อมูล	คำอธิบาย	ประเภทข้อมูล	หมายเหตุ
businessCategory	ประเภทของธุรกิจ	varchar(50)	
successStory	กรณีตัวอย่างที่ใช้งานผลิตภัณฑ์จนประสบความสำเร็จ	varchar(200)	

ตารางที่ 5.4 ตารางข้อมูลที่อยู่ของผู้ใช้ (address)

เขตข้อมูล	คำอธิบาย	ประเภทข้อมูล	หมายเหตุ
userID	รหัสผู้ใช้	int(5)	Primary key
addressNumber	หมายเลขที่อยู่	varchar(15)	
street	ชื่อถนน	varchar(50)	
city	ชื่อเมือง	varchar(50)	
province	ชื่อจังหวัด	varchar(50)	
country	ชื่อประเทศ	varchar(50)	
postalcode	หมายเลขไปรษณีย์	varchar(10)	
phoneNumber	หมายเลขโทรศัพท์	varchar(15)	
mobileNumber	หมายเลขโทรศัพท์เคลื่อนที่	varchar(15)	
faxNumber	หมายเลขโทรสาร	varchar(15)	

ตารางที่ 5.5 ตารางข้อมูลของบริการ (service)

เขตข้อมูล	คำอธิบาย	ประเภทข้อมูล	หมายเหตุ
serviceID	รหัสของบริการ	int(5)	Primary key
name	ชื่อบริการ	varchar(80)	
version	รุ่นของบริการ	varchar(20)	
category	ประเภทของบริการ	varchar(50)	
taxonomy	ข้อมูลอ้างอิงถึงอนุกรมวิธานของบริการ	varchar(100)	
review	ข้อมูลวิจารณ์การใช้งานบริการ	varchar(200)	
overview	คำอธิบายของบริการโดยคร่าว	varchar(150)	
specification	ข้อกำหนดการใช้งานบริการ	varchar(150)	

ตารางที่ 5.5 (ต่อ)

เขตข้อมูล	คำอธิบาย	ประเภทข้อมูล	หมายเหตุ
feature	จุดเด่นของบริการ	varchar(150)	
sample	ตัวอย่างการใช้งานบริการ	varchar(150)	
demo	สาธิตการใช้งานบริการ	varchar(150)	
price	ราคา	double	
WSDLURL	ตำแหน่งที่ตั้งของไฟล์ WSDL	varchar(80)	
pubDate	วันที่สร้างข้อมูล	date	
pubTime	เวลาที่สร้างข้อมูล	time	
provideBy	รหัสผู้ใช้ (ผู้ให้บริการ)	int(5)	Foreign key (provider)

ตารางที่ 5.6 ตารางข้อมูลของส่วนดำเนินการของบริการ (operation)

เขตข้อมูล	คำอธิบาย	ประเภทข้อมูล	หมายเหตุ
operationID	รหัสส่วนดำเนินการ	int(5)	Primary key
name	ชื่อส่วนดำเนินการ	varchar(100)	
type	ชนิดของส่วนดำเนินการ	varchar(10)	input หรือ output
schama	ข้อมูลรูปแบบของชนิดของข้อมูลที่ใช้	varchar(50)	ตำแหน่งที่จัดเก็บไฟล์ XML Schema
description	รายละเอียดส่วนดำเนินการอื่นๆ	varchar(100)	
serviceID	รหัสของบริการ	int(5)	Foreign key (service)

ตารางที่ 5.7 ตารางข้อมูลของแพลตฟอร์มที่บริการสนับสนุน (platform)

เขตข้อมูล	คำอธิบาย	ประเภทข้อมูล	หมายเหตุ
platformID	รหัสแพลตฟอร์ม	int(5)	Primary key
name	ชื่อแพลตฟอร์ม	varchar(100)	
description	รายละเอียดต่างๆ	varchar(200)	
serviceID	รหัสของบริการ	int(5)	Foreign key (service)

ตารางที่ 5.8 ตารางข้อมูลการลงทะเบียนเป็นสมาชิก (registration)

เขตข้อมูล	คำอธิบาย	ประเภทข้อมูล	หมายเหตุ
registID	รหัสการลงทะเบียน	int(5)	Primary key
registDate	วันที่ลงทะเบียน	date	
registTime	เวลาลงทะเบียน	time	
userID	รหัสผู้ใช้	int(5)	Foreign key (user)

ตารางที่ 5.9 ตารางข้อมูลการระบุความต้องการบริการของผู้เรียกใช้บริการ (requirement)

เขตข้อมูล	คำอธิบาย	ประเภทข้อมูล	หมายเหตุ
requireID	รหัสความต้องการ	int(5)	Primary key
pubDate	วันที่สร้างความต้องการ	date	
pubTime	เวลาที่สร้างความต้องการ	time	
serviceName	ชื่อบริการที่ต้องการ	varchar(100)	
description	รายละเอียดของบริการตามที่ต้องการ	varchar(200)	
categoryName	ประเภทของบริการที่ต้องการ	varchar(100)	
operationName	การดำเนินการของบริการที่ต้องการ	varchar(100)	
price	ราคาของบริการที่ต้องการ	double	
filename	ชื่อไฟล์เอกสารข้อกำหนดความต้องการบริการ	varchar(50)	
userID	รหัสผู้ใช้	int(5)	Foreign key (user)

ตารางที่ 5.10 ตารางข้อมูลรายการบริการจากการค้นหา (candidateshortlist)

เขตข้อมูล	คำอธิบาย	ประเภทข้อมูล	หมายเหตุ
candidateID	รหัสรายการผลลัพธ์	int(5)	Primary key
number	จำนวนบริการที่หาได้	int(4)	
requireID	รหัสความต้องการ	int(5)	Foreign key (requirement)

ตารางที่ 5.11 ตารางข้อมูลผลลัพธ์บริการที่ได้จากการค้นหา (candidateservice)

เขตข้อมูล	คำอธิบาย	ประเภทข้อมูล	หมายเหตุ
candidateID	รหัสรายการผลลัพธ์	int(5)	Primary key
serviceID	รหัสของบริการ	int(5)	Primary key

ตารางที่ 5.12 ตารางข้อมูลประกาศความต้องการสู่สาธารณะ (publicannouncement)

เขตข้อมูล	คำอธิบาย	ประเภทข้อมูล	หมายเหตุ
publicID	รหัสการประกาศความต้องการ	int(5)	Primary key
requireID	รหัสความต้องการ	int(5)	Foreign key (requirement)

ตารางที่ 5.13 ตารางข้อมูลเอกสารข้อเสนอบริการ (serviceproposal)

เขตข้อมูล	คำอธิบาย	ประเภทข้อมูล	หมายเหตุ
proposalID	รหัสข้อเสนอบริการ	int(5)	Primary key
pubDate	วันที่ส่งข้อเสนอบริการ	date	
pubTime	เวลาที่ส่งข้อเสนอบริการ	time	
filename	ชื่อไฟล์เอกสารข้อเสนอบริการ	varchar(50)	
candidateID	รหัสรายการผลลัพธ์	int(5)	Foreign key (candidateservice)

ตารางที่ 5.14 ตารางข้อมูลเอกสารข้อเสนอบริการแบบไม่ระบุตัวตน (anonymousproposal)

เขตข้อมูล	คำอธิบาย	ประเภทข้อมูล	หมายเหตุ
anonymID	รหัสข้อเสนอบริการแบบไม่ระบุตัวตน	int(5)	Primary key
pubDate	วันที่สร้างข้อเสนอบริการแบบไม่ระบุตัวตน	date	
pubTime	เวลาที่สร้างข้อเสนอบริการ	time	
filename	ชื่อไฟล์เอกสารข้อเสนอบริการ	varchar(50)	
select	สถานะการเลือกข้อเสนอบริการของผู้เรียกใช้บริการรายนั้นๆ	int(1)	0 คือไม่ถูกเลือก 1 คือถูกเลือก
userID	รหัสผู้ใช้	int(5)	Foreign key (client)

ตารางที่ 5.14 (ต่อ)

เขตข้อมูล	คำอธิบาย	ประเภทข้อมูล	หมายเหตุ
proposallID	รหัสข้อเสนอบริการ	int(5)	Foreign key (serviceproposal)

ตารางที่ 5.15 ตารางข้อมูลเอกสารข้อตกลงบริการฉบับร่าง (draftagreement)

เขตข้อมูล	คำอธิบาย	ประเภทข้อมูล	หมายเหตุ
draftID	รหัสข้อตกลงบริการฉบับร่าง	int(5)	Primary key
filename	ชื่อไฟล์ข้อตกลงบริการฉบับร่าง	varchar(50)	
extend	ชื่อไฟล์เอกสารเพิ่มเติม	varchar(50)	
clientStatus	สถานะผู้เรียกใช้บริการยอมรับ	int(1)	0 คือไม่ยอมรับ 1 คือยอมรับ
provider Status	สถานะผู้ให้บริการยอมรับ	int(1)	0 คือไม่ยอมรับ 1 คือยอมรับ
pubDate	วันที่สร้างข้อตกลงบริการฉบับร่าง	date	
pubTime	เวลาที่สร้างข้อตกลงบริการฉบับร่าง	time	
anonymID	รหัสข้อเสนอบริการแบบไม่ระบุ ตัวตน	int(5)	Foreign key (anonymousproposal)

ตารางที่ 5.16 ตารางข้อมูลเอกสารข้อตกลงบริการ (serviceagreement)

เขตข้อมูล	คำอธิบาย	ประเภทข้อมูล	หมายเหตุ
agreeID	รหัสข้อตกลงบริการ	int(5)	Primary key
filename	ชื่อไฟล์ข้อตกลงบริการ	varchar(50)	
extend	ชื่อไฟล์เอกสารเพิ่มเติม	varchar(50)	
pubDate	วันที่สร้างข้อตกลงบริการ	date	
pubTime	เวลาที่สร้างข้อตกลงบริการ	time	
draftID	รหัสข้อตกลงบริการฉบับร่าง	int(5)	Foreign key (draftagreement)

ตารางที่ 5.17 ตารางข้อมูลการส่งมอบบริการ (delivery)

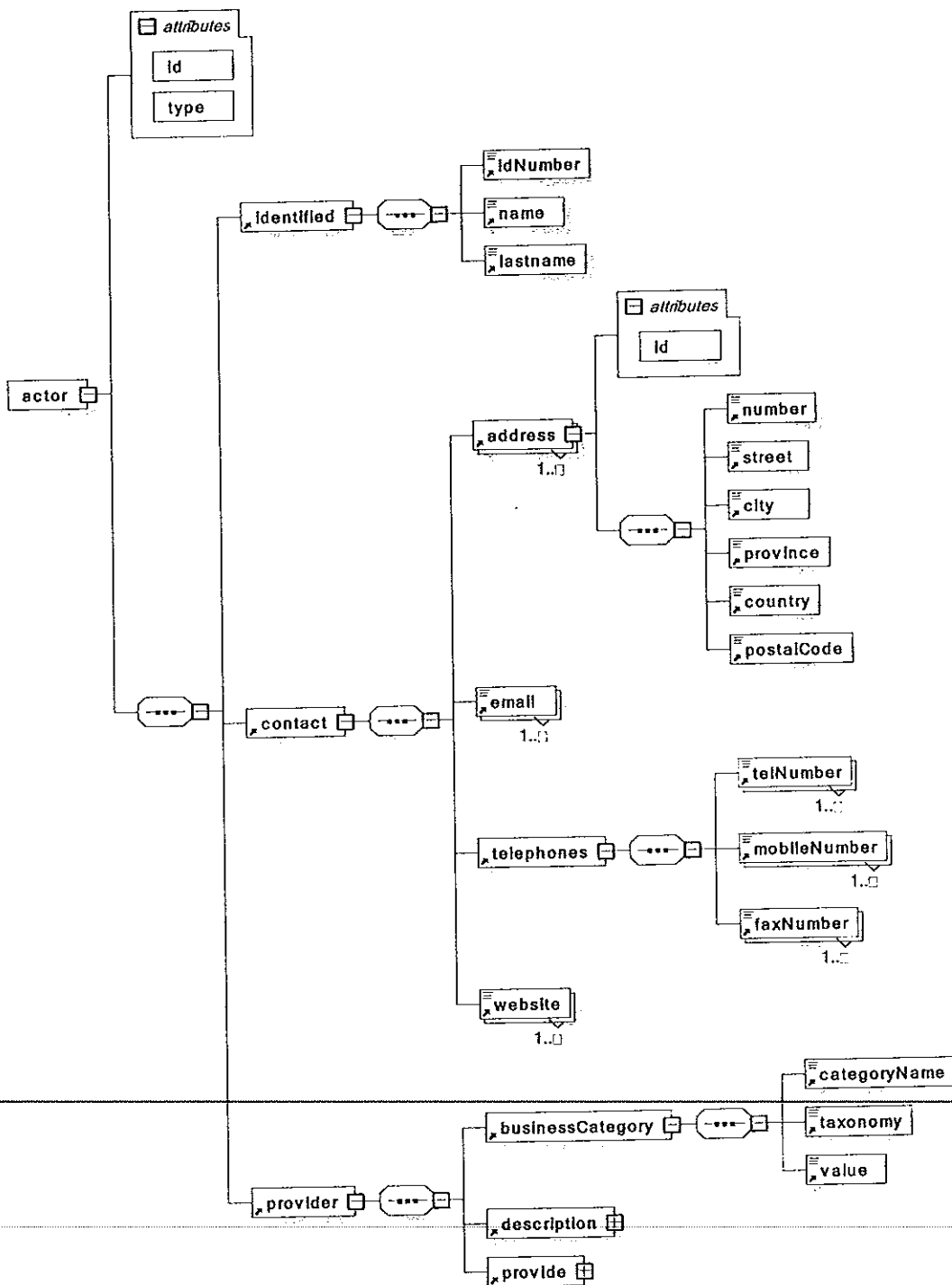
เขตข้อมูล	คำอธิบาย	ประเภทข้อมูล	หมายเหตุ
deliverID	รหัสการส่งมอบบริการ	int(5)	Primary key
agreelD	รหัสข้อตกลงบริการ	int(5)	
pubDate	วันที่สร้างข้อตกลงบริการ	date	
pubTime	เวลาที่สร้างข้อตกลงบริการ	time	

5.1.4 การพัฒนาการแทนความรู้สำหรับระบบ

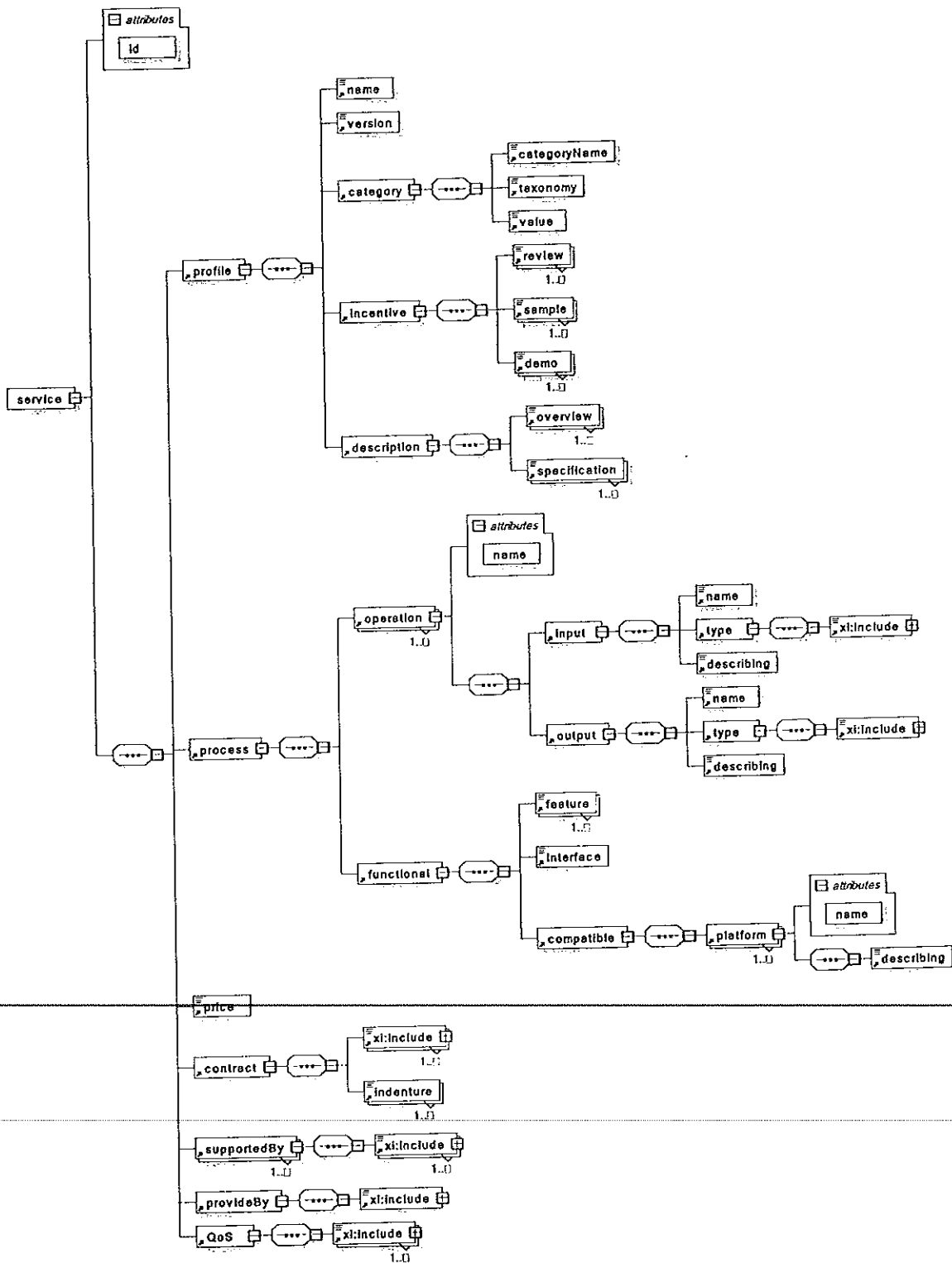
การพัฒนาส่วนการแทนความรู้ที่สอดคล้องกับออนโทโลยีที่ได้ออกแบบในบทที่ 4 หัวข้อที่ 4.5 นั้น งานวิจัยนี้ได้พัฒนาโดยใช้เทคโนโลยี XML โดยเลือก XML Schema สำหรับประยุกต์ใช้แทนออนโทโลยีตามที่ออกแบบไว้ และตัวความรู้ได้ประยุกต์ใช้ XML ที่สอดคล้องกับ XML Schema ดังกล่าว ดังแสดงในภาพประกอบที่ 5.10 – 5.12 สำหรับรายละเอียดเอกสารจริงสามารถดูได้ในภาคผนวก ข.

5.1.5 การพัฒนาส่วนติดต่อผู้ใช้

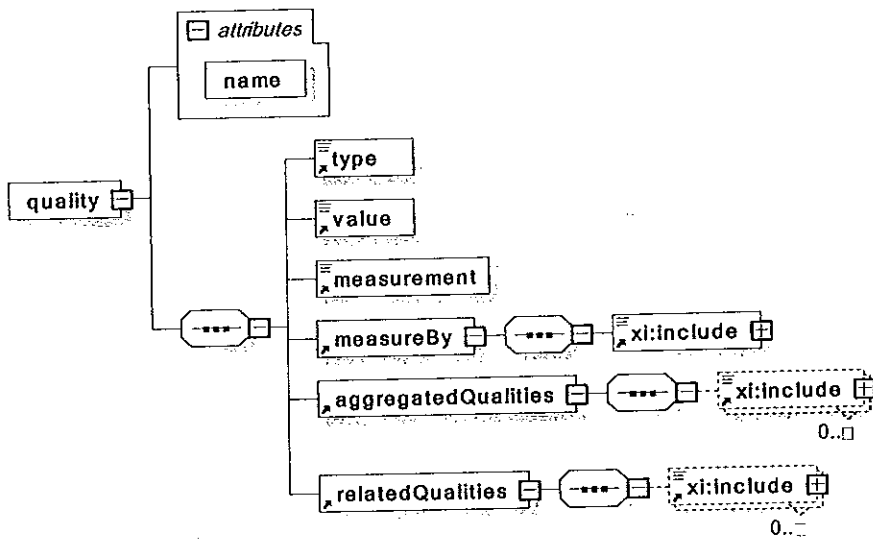
การดำเนินงานและการแสดงผลต่างๆของระบบผู้ประสานงานการเจรจาต่อรอง จะกระทำผ่านโปรแกรมประยุกต์ที่ให้บริการผ่านเว็บ ซึ่งได้รับการพัฒนาให้มีฟังก์ชันรองรับการใช้งานตามที่ได้ออกแบบไว้ในข้างต้น แผนผังโครงสร้างเว็บไซต์ถูกแบ่งออกเป็น 2 แบบสำหรับสิทธิ์ในการเข้าใช้งานสองบทบาทคือผู้ให้บริการและผู้เรียกใช้บริการดังแสดงในภาพประกอบที่ 5.13 และมีรายละเอียดของส่วนต่างๆดังต่อไปนี้



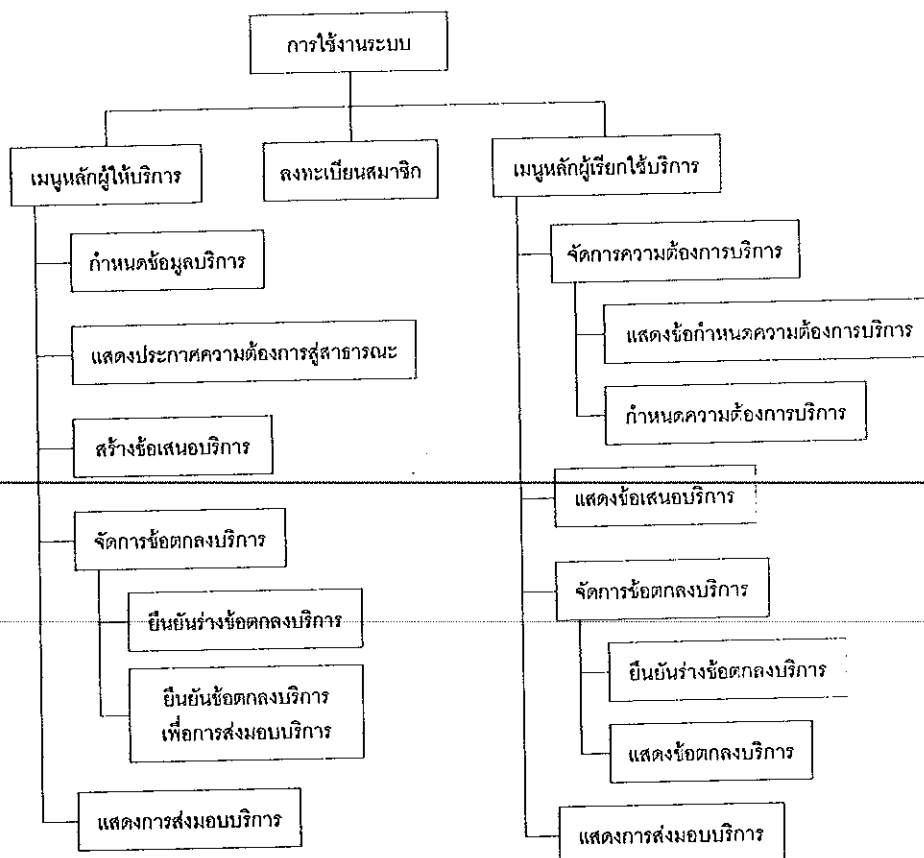
ภาพประกอบที่ 5.10 แผนภาพโครงสร้างของ XML Schema ที่สอดคล้องกับ
 ontology ของผู้ดำเนินการที่มีส่วนเกี่ยวข้องกับระบบ



ภาพประกอบที่ 5.11 แผนภาพโครงสร้างของ XML Schema ที่สอดคล้องกับ ontology บริการ



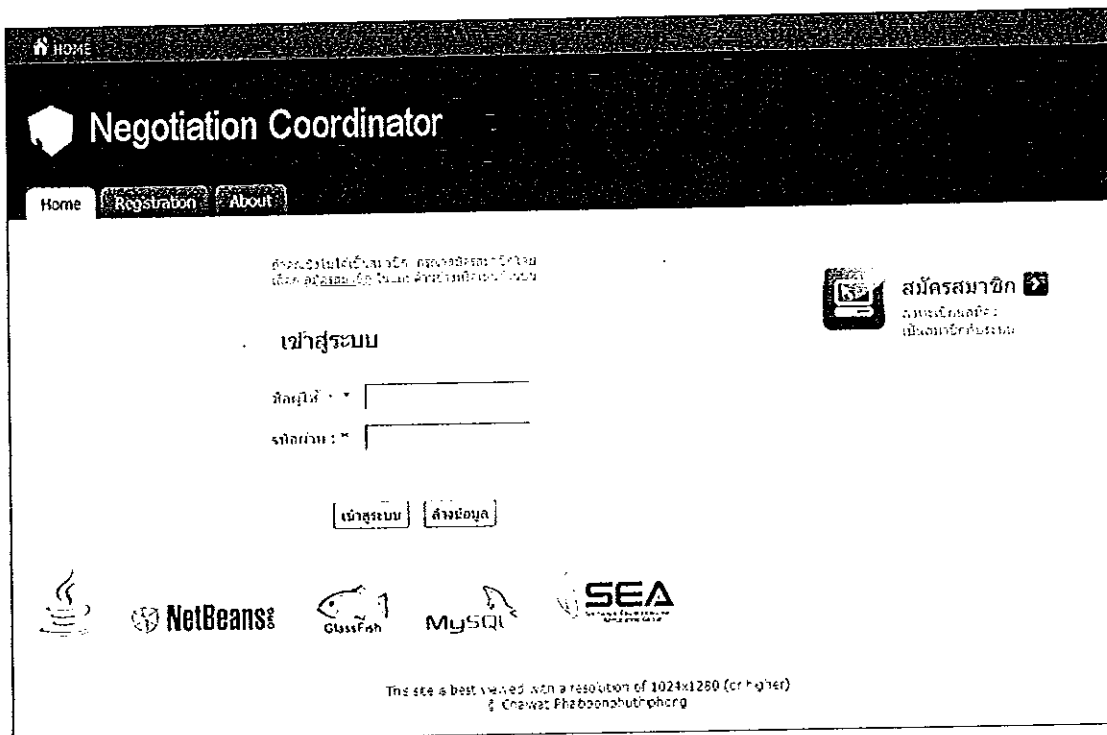
ภาพประกอบที่ 5.12 แผนภาพโครงสร้างของ XML Schema ที่สอดคล้องกับ
 ออนโทโลยีคุณภาพของบริการ



ภาพประกอบที่ 5.13 แผนผังโครงสร้างโปรแกรมประยุกต์ผ่านเว็บของ
 ระบบผู้ประสานงานการเจรจาต่อรอง

1. การเข้าใช้งานระบบ

หน้าจอหลักของระบบคือการเข้าใช้งานระบบ ผู้ใช้จะต้องมีบัญชีผู้ใช้นั้นคือชื่อผู้ใช้ (Username) และรหัสผ่าน (Password) เพื่อใช้ในการตรวจสอบสิทธิ์ในการเข้าใช้งานระบบ ซึ่งจะกำหนดถึงฟังก์ชันการทำงานที่ผู้ใช้นั้นๆสามารถดำเนินการได้ แสดงหน้าจอการเข้าใช้งานระบบดังภาพประกอบที่ 5.14



ภาพประกอบที่ 5.14 หน้าจอหลักที่แสดงการเข้าใช้งานระบบผู้ประสานงานการเจรจาต่อรอง

2. การลงทะเบียนสมาชิก

ถ้าหากมีผู้ใช้รายใดต้องการเข้าใช้งานระบบ แต่ยังไม่ได้เป็นสมาชิกกับระบบ ผู้ใช้สามารถเลือกลงทะเบียนสมัครเป็นสมาชิกกับระบบได้โดยการเลือกสมัครสมาชิกและกรอกข้อมูลต่างๆให้สมบูรณ์ตามลำดับขั้นตอน ดังแสดงในภาพประกอบที่ 5.15

3. หน้าจอสำหรับผู้ให้บริการ

เมื่อผู้ใช้ที่มีสิทธิ์การเข้าใช้งานเป็นผู้ให้บริการได้เข้าสู่ระบบ ระบบจะแสดงแถบรายการการทำงานที่สามารถเลือกใช้ได้ดังภาพประกอบที่ 5.16 แถบรายการการใช้งานสามารถอธิบายได้ดังนี้

HOME | REGISTRATION | STEP 1

Negotiation Coordinator

Home Registration About

ขั้นตอนที่ 1

ขั้นตอนที่ 1
กำหนดข้อมูลสมาชิก

ข้อมูลสมาชิก

ชื่อผู้เช่า *

รหัสผ่าน *

ประเภทสมาชิก * Client / Provider

This site is best viewed with a resolution of 1024x1280 (or higher)
Chaivat Phatbenphatphong

ภาพประกอบที่ 5.15 หน้าจอแสดงการลงทะเบียนสมัครเป็นสมาชิกกับระบบ

HOME

Negotiation Coordinator

Home Define Service Public Announcement Service Proposal Agreement Delivery Logout

Provider

ไฉฉวี ไชวถนุญ
หมายเลข 0602531394

ขอต้อนรับสู่ระบบผู้ประสานงานการเจรจาต่อรอง

ขอแสดงความยินดีแก่ผู้ประสานงานในการได้งานจากผู้ขาย หรือผู้บริการจากทางระบบโดย **คลิกที่นี่**

NetBeans GlassFish MySQL SEA

This site is best viewed with a resolution of 1024x1280 (or higher)
Chaivat Phatbenphatphong

ภาพประกอบที่ 5.16 หน้าจอหลักแสดงส่วนการทำงานเมื่อผู้ให้บริการเข้าสู่ระบบ

(1) การกำหนดข้อมูลบริการ


ผู้ให้บริการสามารถกำหนดข้อมูลบริการที่ตนต้องการเสนอให้แก่ระบบได้โดยการกรอกข้อมูลเกี่ยวกับบริการในช่องกรอกข้อมูลตามความเหมาะสม ดังแสดงในภาพประกอบที่ 5.17

HOME | DEFINE SERVICE

Negotiation Coordinator

Home Define Service Public Announcement Service Proposal Agreement Delivery Logout

การกำหนดข้อมูลบริการ

Provider  **Provider**
 56918 chainathay
 2553-07-15 15:02:53 13241

ข้อมูลของบริการ

ชื่อของบริการ *	Hotelservice	ราคา	1.0
คำอธิบายของบริการ	Travel agencies and service		
หมวดหมู่ประเภท ?	Hotel reservation	taxonomy ?	http://uddi.org/taxonomies
ราคา	5		
ข้อมูลองค์กรที่ให้บริการ	Mike Company One		

ข้อมูลรายละเอียดของบริการ

ตัวอย่างบริการ

ภาพประกอบที่ 5.17 หน้าจอแสดงการกำหนดบริการของผู้ให้บริการ

(2) การแสดงประกาศความต้องการสู่สาธารณะ


ผู้ให้บริการสามารถเรียกดูเอกสารความต้องการบริการจากผู้เรียกใช้บริการต่างๆได้เพื่อช่วยในการตัดสินใจสำหรับการกำหนดบริการที่ตนมีอยู่หรือพัฒนาบริการเพิ่มเติมให้สอดคล้องกับความต้องการของลูกค้า ซึ่งสามารถแสดงในภาพประกอบที่ 5.18

HOME | PUBLIC ANNOUNCEMENT

Negotiation Coordinator

Home Define Service Public Announcement Service Proposal Agreement Delivery Logout

รวมการประกาศความต้องการสู่สาธารณะ

Provider  **เอกชัย chatwathuy**
หมายเลขระบบ 160253 13:41

№	วันที่	เวลา	เอกสารข้อกำหนดความต้องการบริการ	
7	2552-11-19	2:56 PM	ข้อกำหนดความต้องการบริการที่ 1	๒
6	2552-11-19	6:41 PM	ข้อกำหนดความต้องการบริการที่ 3	๒
5	2552-11-19	6:13 PM	ข้อกำหนดความต้องการบริการที่ 4	๒
4	2552-11-19	6:45 PM	ข้อกำหนดความต้องการบริการที่ 5	๒
3	2552-11-19	6:46 PM	ข้อกำหนดความต้องการบริการที่ 6	๒
2	2552-11-22	5:39 PM	ข้อกำหนดความต้องการบริการที่ 9	๒
1	2552-11-11	1:51 PM	ข้อกำหนดความต้องการบริการที่ 11	๒

Show per page: 12 15 20 ALL

Showing 1 - 7 of 7

ภาพประกอบที่ 5.18 หน้าจอแสดงรายการการประกาศความต้องการสู่สาธารณะ

(3) การสร้างข้อเสนอบริการ

ผู้ให้บริการสามารถสร้างข้อเสนอบริการเพื่อเสนอให้ผู้เรียกใช้บริการรายต่างๆ ได้พิจารณาตัดสินใจสำหรับการเลือกใช้บริการ ดังแสดงได้ในภาพประกอบที่ 5.19

(4) การยืนยันร่างข้อตกลง

ผู้ให้บริการสามารถสร้างข้อตกลงบริการฉบับร่างหรือพร้อมทั้งแนบเอกสารข้อตกลงอื่นๆ ที่ตนมีเพื่อให้ผู้เรียกใช้บริการพิจารณายอมรับในลำดับถัดไป ดังแสดงได้ในภาพประกอบที่ 5.20

HOME SERVICE PROPOSAL

Negotiation Coordinator


Home Define Service Public Announcement Service Proposal Agreement Delivery Logout

กำหนดข้อเสนอบริการ

เลข	วันที่	เวลา	ชื่อผู้เรียกใช้บริการ	สร้างข้อเสนอบริการ
1	2553-02-09	7:29 PM	123	<input type="button" value="เลือก"/>
2	2553-02-09	7:37 PM	123	<input type="button" value="เลือก"/>
1	2553-02-09	7:46 PM	123	<input type="button" value="เลือก"/>

Showing 1 - 3 of 3

Show per page: 10 15 20 All

Provider  **chaiwathuy**
 16-02-53 13:41

ภาพประกอบที่ 5.19 หน้าจอแสดงรายการสำหรับการสร้างข้อเสนอบริการ

HOME DRAFT AGREEMENT LIST

Negotiation Coordinator

Home Define Service Public Announcement Service Proposal Agreement Delivery Logout


รายการร่างข้อตกลงบริการ

เลข	วันที่	เวลา	เอกสารร่างข้อตกลงบริการ	จำนวน	เลือก
5	2553-02-10	3:29 PM	ร่างข้อตกลงบริการที่ 5	๖	<input type="button" value="เลือก"/>
6	2553-02-10	3:29 PM	ร่างข้อตกลงบริการที่ 6	๖	<input type="button" value="เลือก"/>
7	2553-02-10	3:29 PM	ร่างข้อตกลงบริการที่ 7	๖	<input type="button" value="เลือก"/>
8	2553-02-10	3:29 PM	ร่างข้อตกลงบริการที่ 8	๖	<input type="button" value="เลือก"/>
9	2553-02-10	3:29 PM	ร่างข้อตกลงบริการที่ 9	๖	<input type="button" value="เลือก"/>

Showing 1 - 5 of 5

Show per page: 10 15 20 All

Service Agreement

Provider  **chaiwathuy**
 16-02-53 13:41

ภาพประกอบที่ 5.20 หน้าจอแสดงรายการร่างข้อตกลงบริการ
 เพื่อให้ผู้ให้บริการพิจารณายืนยัน

(5) การยืนยันข้อตกลงบริการเพื่อการส่งมอบ

เมื่อผู้เรียกใช้บริการยอมรับร่างข้อตกลงแล้ว ผู้ให้บริการสามารถเรียกดูรายละเอียดของข้อตกลงบริการ และสามารถกำหนดการส่งมอบบริการเมื่อพร้อมให้ผู้เรียกใช้บริการเรียกใช้งานได้ ดังแสดงได้ดังภาพประกอบที่ 5.21

The screenshot shows the 'Negotiation Coordinator' interface. At the top, there are navigation tabs: Home, Define Service, Public Announcement, Service Proposal, Agreement, Delivery, and Logout. The 'Agreement' tab is active, showing a 'Draft Agreement' and 'Service Agreement' sub-tab. Below the navigation, there is a user profile for 'Provider' with the name 'Seq'U' chaiwatkey and a phone number '16-07-53 1341'. The main content area is titled 'รายการข้อตกลงบริการ' (Service Agreement List) and contains a table with the following data:

เลข	วันที่	เวลา	เอกสารข้อตกลงบริการ	เอกสารแนบเดิม	
4	2023-02-16	3:40 PM	☑️ เอกสารบริการที่ 5	☑️ 5trname1.td	🗑️
3	2023-02-16	3:05 PM	☑️ เอกสารบริการที่ 4	☑️ 4trname1.td	🗑️
2	2023-02-16	3:01 PM	☑️ เอกสารบริการที่ 3	☑️ 3trname1.td	🗑️ <input type="button" value="อนุมัติ"/>
1	2022-10-04	5:24 PM	☑️ เอกสารบริการที่ 2	☑️ 2trname1.td	🗑️

At the bottom of the table, there is a pagination control: 'Showing 1 - 4 of 4' and 'Show per page: 10 15 20 All'.

ภาพประกอบที่ 5.21 หน้าจอแสดงรายการข้อตกลงบริการ

โดยผู้ให้บริการสามารถกำหนดการส่งมอบได้เมื่อพร้อม

(6) การแสดงการส่งมอบบริการ

ผู้ให้บริการสามารถเรียกดูรายการของบริการที่ได้กำหนดการส่งมอบบริการให้แก่ผู้เรียกใช้บริการเรียกใช้งาน ซึ่งจะช่วยในการพิจารณาถึงเงื่อนไขการเรียกใช้งาน ดังแสดงได้ดังภาพประกอบที่ 5.22

4. หน้าจอสำหรับผู้เรียกใช้บริการ

สำหรับผู้ใช้ที่มีสิทธิ์การเข้าใช้งานเป็นผู้เรียกใช้บริการได้เข้าสู่ระบบ ระบบจะแสดงแถบรายการการทำงานที่สามารถเลือกใช้ได้ดังภาพประกอบที่ 5.23 แถบรายการการทำงานต่างๆสามารถอธิบายได้ดังนี้

HOME | DELIVERY

Negotiation Coordinator


Home | Define Service | Public Announcement | Service Proposal | Agreement | Delivery | Logout

รายการส่งมอบบริการ

เลข	วันที่	เวลา	ชื่อบริการ	เอกสารข้อตกลงบริการ	เอกสารเพิ่มเติม	
3	2553-11-31	6:55 PM	HotelEmail	ข้อตกลงบริการที่ 2	2Helsa.bt	๒
2	2553-02-10	3:27 PM	HotelEmail	ข้อตกลงบริการที่ 4	4HName1.bt	๒
1	2553-02-10	4:05 PM	HotelEmail	ข้อตกลงบริการที่ 5	5HName1.bt	๒

Showing 1 - 3 of 3

Show per page: 10 15 20 ALL

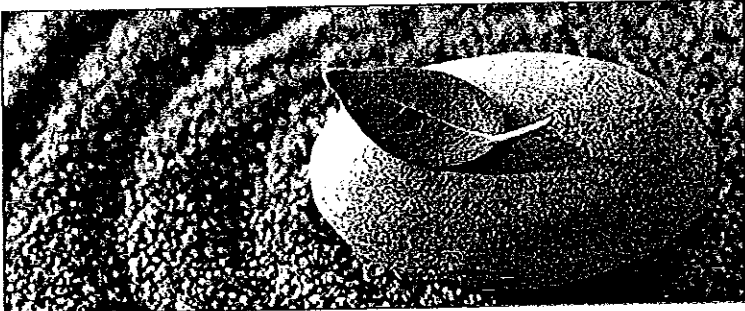
Provider  **Provider**
 25.000 | chawathuy
 เวลาที่เข้าชม: 16/02/53 13:41


ภาพประกอบที่ 5.22 หน้าจอแสดงรายการของบริการที่ได้กำหนดการส่งมอบเรียบร้อยแล้ว

HOME






Negotiation Coordinator

Home | Requirement | Service Proposal | Agreement | Delivery | Logout



Client  **Client**
 Login: 123
 เวลาที่เข้าชม: 16/02/53 13:41

ขอต้อนรับสู่ระบบสู่ประสานงานการเจรจาต่อรอง
 ขอแนะนำ... [คลิกที่นี่](#)

This site is best viewed with a resolution of 1024x1280 (or higher).
 © Chawat Pha bbenputti@phong

ภาพประกอบที่ 5.23 หน้าจอหลักแสดงส่วนการทำงานเมื่อผู้เรียกใช้บริการเข้าสู่ระบบ

(1) การแสดงข้อกำหนดความต้องการบริการ

ผู้เรียกใช้บริการสามารถเรียกดูรายการของข้อกำหนดความต้องการบริการทั้งหมดที่ตนเองเคยกำหนด ซึ่งระบบจะแสดงเอกสารความต้องการบริการควบคู่ไปกับสถานะแสดงการพบผลลัพธ์จากการค้นหาบริการตามความต้องการนั้นๆ ดังแสดงได้ในภาพประกอบที่

5.24

The screenshot displays a web application titled "Negotiation Coordinator". The main content area is a table listing service requirements. The table has the following columns: ID, Name, Time, Description, and Status. The status for all items is "ไม่พบ" (Not Found). The interface also includes a navigation menu at the top with options: Home, Requirement, Service Proposal, Agreement, Drawings, and Logout. A client profile is shown in the top right corner with the name "Client" and a profile picture.

ID	วันที่	เวลา	เอกสารความต้องการบริการ	ผลการค้นหา
71	2553-02-11	2:53 PM	ข้อกำหนดความต้องการบริการที่ 76	ไม่พบ
70	2553-02-11	2:52 PM	ข้อกำหนดความต้องการบริการที่ 75	ไม่พบ
69	2553-02-11	2:52 PM	ข้อกำหนดความต้องการบริการที่ 74	ไม่พบ
68	2553-02-11	2:52 PM	ข้อกำหนดความต้องการบริการที่ 73	ไม่พบ
67	2553-02-11	2:51 PM	ข้อกำหนดความต้องการบริการที่ 72	ไม่พบ
66	2553-02-11	2:51 PM	ข้อกำหนดความต้องการบริการที่ 71	ไม่พบ
65	2553-02-11	2:51 PM	ข้อกำหนดความต้องการบริการที่ 70	ไม่พบ
64	2553-02-11	2:51 PM	ข้อกำหนดความต้องการบริการที่ 69	ไม่พบ
63	2553-02-11	2:50 PM	ข้อกำหนดความต้องการบริการที่ 68	ไม่พบ
62	2553-02-11	2:50 PM	ข้อกำหนดความต้องการบริการที่ 67	ไม่พบ

ภาพประกอบที่ 5.24 หน้าจอแสดงรายการของข้อกำหนดความต้องการบริการ

(2) การกำหนดความต้องการบริการ

ผู้เรียกใช้บริการสามารถกำหนดความต้องการบริการผ่านทางช่องกรอกข้อมูลตามที่กำหนดไว้ และเมื่อกำหนดความต้องการเรียบร้อยแล้ว ระบบจะสร้างเอกสารข้อกำหนดความต้องการบริการพร้อมทั้งประมวลผลค้นหาบริการที่เหมาะสมในลำดับถัดไป ซึ่งสามารถแสดงได้ดังภาพประกอบที่ 5.25

The screenshot shows a web application interface for a 'Negotiation Coordinator'. At the top, there are navigation tabs: 'Home', 'Requirement', 'Service Proposal', 'Agreement', 'Delivery', and 'Logout'. The main heading is 'Negotiation Coordinator'. Below this, there is a user profile section for 'Client' with a profile picture, ID '123', and a phone number '15 9263 14 09'. The main content area is titled 'รายการข้อกำหนดความต้องการบริการ' (Service Requirement List). It contains a form with the following fields:

- บริการ (Service): HotelService
- ประเภทผลิตภัณฑ์ (Product Category): Hotel Reservation
- ประเภทของบริการ (Service Type): Travel agencies and service
- การดำเนินงาน (Operation):
- ราคา (Price): 5

At the bottom of the form, there are two buttons: 'ตกลง' (OK) and 'ใส่ข้อมูล' (Enter Data).

ภาพประกอบที่ 5.25 หน้าจอแสดงการกำหนดความต้องการบริการของผู้เรียกใช้บริการ

(3) การแสดงข้อเสนอบริการ

ผู้เรียกใช้บริการสามารถเรียกดูรายการของข้อเสนอบริการซึ่งสอดคล้องกับผลลัพธ์ที่ระบบได้ค้นหาตามความต้องการบริการที่กำหนดไว้ และคัดเลือกข้อเสนอบริการจากผู้ให้บริการรายต่างๆ ในรายการตามที่ตนพึงพอใจ ดังแสดงได้ในภาพประกอบที่ 5.26

(4) การยืนยันร่างข้อตกลง

ผู้เรียกใช้บริการสามารถพิจารณาเพื่อยืนยันการยอมรับร่างข้อตกลงบริการที่ผู้ให้บริการรายที่ถูกคัดเลือกจัดทำขึ้น ซึ่งร่างเอกสารข้อตกลงบริการนี้จะแปรเปลี่ยนเป็นข้อตกลงบริการฉบับจริง ดังแสดงได้ในภาพประกอบที่ 5.27

(5) การแสดงข้อตกลงบริการ

ผู้เรียกใช้บริการสามารถเรียกดูรายการของข้อตกลงบริการที่ได้มาจากการยอมรับร่างข้อตกลงบริการร่วมกัน ซึ่งระบบจะแสดงทั้งเอกสารข้อตกลงบริการและเอกสารสัญญาอื่นๆ ที่ผู้ให้บริการแนบมา ดังแสดงได้ในภาพประกอบที่ 5.28

HOME | SERVICE PROPOSAL


Negotiation Coordinator

Home Requirement Service Proposal Agreement Delivery Logout

เลือกขุดข้อเสนอบริการ

NO	วันที่	เวลา	เอกสารความต้องการบริการ	
1	2552-10-11	5:45 PM	ได้ เอกสารความต้องการบริการที่ 64	เลือก
4	2552-11-27	5:49 PM	ได้ เอกสารความต้องการบริการที่ 65	เลือก
1	2553-02-11	5:29 PM	ได้ เอกสารความต้องการบริการที่ 70	เลือก
2	2553-02-13	7:37 PM	ได้ เอกสารความต้องการบริการที่ 71	เลือก
1	2553-02-14	7:46 PM	ได้ เอกสารความต้องการบริการที่ 72	เลือก

Showing 1 - 5 of 5 Show per page: 10 15 20 All

Client  **Client**
 56914 123
 15/02/53 14:09

ภาพประกอบที่ 5.26 หน้าจอแสดงรายการข้อเสนอบริการเพื่อพิจารณาคัดเลือก

HOME | DRAFT AGREEMENT

Negotiation Coordinator


Home Requirement Service Proposal Agreement Delivery Logout

Draft Agreement
Service Agreement

รายการร่างข้อตกลงบริการ

NO	วันที่	เวลา	เอกสารร่างข้อตกลงบริการ	
1	2553-02-11	5:09 PM	ได้ ร่างข้อตกลงบริการที่ 5	รับ
1	2553-02-11	5:09 PM	ได้ ร่างข้อตกลงบริการที่ 6	รับ
1	2553-02-11	5:12 PM	ได้ ร่างข้อตกลงบริการที่ 7	รับ
1	2553-02-11	5:14 PM	ได้ ร่างข้อตกลงบริการที่ 8	รับ
1	2553-02-11	5:23 PM	ได้ ร่างข้อตกลงบริการที่ 9	รับ

Showing 1 - 5 of 5 Show per page: 10 15 20 All

Client  **Client**
 56914 123
 15/02/53 14:09

ภาพประกอบที่ 5.27 หน้าจอแสดงรายการร่างข้อตกลงบริการเพื่อพิจารณายืนยัน

HOME | SERVICE AGREEMENT

Negotiation Coordinator

Home Requirement Service Proposal Agreement Delivery Logout


Draft Agreement
Service Proposal

รายการข้อตกลงบริการ

№	วันที่	เวลา	เอกสารข้อตกลงบริการ	เอกสารแนบเดิม	ผู้เสนอบริการ	
4	2023-10-17	3:41 PM	☑️ ข้อตกลงบริการที่ 5	☑️ S10name1.txt	ชัชวาลย์	👍
3	2023-10-17	3:01 PM	☑️ ข้อตกลงบริการที่ 4	☑️ 410name1.txt	ชัชวาลย์	👍
2	2023-10-17	3:01 PM	☑️ ข้อตกลงบริการที่ 3	☑️ 310name1.txt	ชัชวาลย์	👍
1	2023-10-26	5:24 PM	☑️ ข้อตกลงบริการที่ 2	☑️ 210name1.txt	ชัชวาลย์	👍

Showing 1 - 4 of 4

Show per page: 10 15 20 ALL

Client  5 คนใช้ 123
เวลาเริ่มต้น: 15-02-53 14:03

ภาพประกอบที่ 5.28 หน้าจอแสดงรายการข้อตกลงบริการที่ผ่านการยอมรับจากทั้งสองฝ่าย

(6) การแสดงการส่งมอบบริการ

ผู้เรียกใช้บริการสามารถเรียกดูรายการของบริการที่พร้อมสำหรับการเรียกใช้งานโดยมีเอกสารข้อตกลงบริการและเอกสารสัญญาอื่น ๆ ก่ากับการเรียกใช้งานไว้ ดังแสดงได้ดังภาพประกอบที่ 5.29

HOME | DELIVERY

Negotiation Coordinator


Home Requirement Service Proposal Agreement Delivery Logout

รายการส่งมอบบริการ

№	วันที่	เวลา	ชื่อบริการ	เอกสารข้อตกลงบริการ	เอกสารแนบเดิม	
3	2023-10-10	4:15 PM	☑️ HeteEmail	☑️ ข้อตกลงบริการที่ 2	☑️ 210name1.txt	👍
2	2023-10-10	3:27 PM	☑️ HeteEmail	☑️ ข้อตกลงบริการที่ 4	☑️ 410name1.txt	👍
1	2023-10-20	4:05 PM	☑️ HeteEmail	☑️ ข้อตกลงบริการที่ 5	☑️ 510name1.txt	👍

Showing 1 - 3 of 3

Show per page: 10 15 20 ALL

Client  5 คนใช้ 123
เวลาเริ่มต้น: 15-02-53 14:03

ภาพประกอบที่ 5.29 หน้าจอแสดงรายการของบริการที่พร้อมสำหรับการเรียกใช้งาน

5.2 การพัฒนากรณีตัวอย่าง

งานวิจัยนี้ได้เลือกบริการการท่องเที่ยวเป็นกรณีตัวอย่างสำหรับการทดสอบระบบผู้ประสานงานการเจรจาต่อรอง โดยทั่วไปแล้วบริการการท่องเที่ยวมักจะประกอบด้วย การดำเนินงาน 2 ส่วนหลักได้แก่ การจองที่พัก เช่น การจองเข้าพักรวมโรงแรม และการจองพาหนะสำหรับการเดินทาง เช่น การจองตั๋วเครื่องบินหรือการจองตั๋วรถยนต์ มาประกอบกันเป็นชุดการท่องเที่ยวเพื่อนำเสนอแก่ลูกค้า ดังนั้นจึงเลือกพัฒนาการบริการจองโรงแรมและบริการจองสายการบินเป็นบริการที่จะสนับสนุนการทดสอบระบบ โดยสมมติผู้ให้บริการ 2 ราย แต่ละรายจะมีบริการจองโรงแรมและบริการจองสายการบินที่ดำเนินงานเหมือนกัน สามารถแสดงรายละเอียดได้ดังนี้

5.2.1 การพัฒนาส่วนดำเนินการ

การบริการการท่องเที่ยวของผู้ให้บริการทั้งสองรายถูกพัฒนาโดยประยุกต์ใช้ Metro (Metro 2010) ซึ่งเป็นกรอบการทำงานสำหรับการพัฒนาเว็บเซอร์วิสด้วยภาษาจาวา ประกอบด้วยบริการจองโรงแรมและบริการจองตั๋วเครื่องบิน โดยบริการจองโรงแรมจะมีการดำเนินการสองส่วนคือการร้องขอข้อมูลห้องพักที่ยังว่างอยู่และการจองห้องพัก ส่วนบริการจองสายการบินมีการดำเนินการสองส่วนเช่นกันคือการร้องขอข้อมูลที่นั่งที่ยังว่างและการจองที่นั่ง รายละเอียดคุณลักษณะต่างๆของทั้งสองผู้ให้บริการสามารถแสดงได้ดังตารางที่ 5.16 และ 5.17

ตารางที่ 5.18 ตารางข้อมูลคุณลักษณะบริการของผู้ให้บริการรายที่หนึ่ง

Service Information		
Service name	HotelService	AirTicket
Version	1.0	1.0
Category	Travel agencies and service.	Travel agencies and service.
Overview	Find the appropriate quarterage for rooms and booking it.	AirTicket make you easy to search list of available seats and booking.
Specification	http://company_one/hSpec.html	http://company_one/aSpec.html
Review	-	-

ตารางที่ 5.18 (ต่อ)

Service Information		
Sample	http://company_one/hSamp.html	http://company_one/aSamp.html
Demo	http://company_one/hDemo.html	http://company_one/aDemo.html
Feature	Searching available room at any price and type, and booking it.	Finding list of available seats and booking seat.
Development environment	Java technology: Metro framework with GlassFish application server.	Java technology: Metro framework with GlassFish application server.
Access protocol	SOAP with link to WSDL file.	SOAP with link to WSDL file.
Operation	getAvailableRoom, bookingRoom	getAvailableSeat, bookingSeat
Compatibility	Web service	Web service
Interface	http://company_one/ws/HotelService?wsdl	http://company_one/ws/AirTicket?wsdl
Price	5 baht per transaction.	5 baht per transaction.
Delivery	Transaction report at the end of month.	Transaction report at the end of month.
Business Information		
Name	Company One Co. Ltd,	
Address	15 Karnjanavanit, Hat Yai, Songkhla 90112 Thailand	
E-mail	Request@company_one.do	
Tel.	+66 (074) 288572	
Fax.	+66 (074) 288572	
Mobile.	+66 (081) 1234567	
Web site	http://company_one/index.php	
Category	Travel agencies and service.	
Description	The travel agency company in the south of Thailand.	
Overview	Provide online travel reservation.	
Reseller	Company One Derived Business (CODeB)	
Reference customer	Company Three Travel.	
Support tel.	+66 (074) 288572	

ตารางที่ 5.18 (ต่อ)

Business Information	
Support fax.	+66 (074) 288572
Support mobile.	+66 (081) 1234567
Support E-mail	Support@company_one.do
Support web site	http://company_one/support/index.php

ตารางที่ 5.19 ตารางข้อมูลคุณลักษณะบริการของผู้ให้บริการรายที่สอง

Service Information		
Service name	BookingHotel	FlightBooking
Version	1.0	1.0
Category	Travel agencies and service.	Travel agencies and service.
Overview	You can booking room at the lowest price with your preference.	Choose the best one of preferred seat and booking it with us.
Specification	http://company_two/bSpec.html	http://company_two/fSpec.html
Review	-	-
Sample	http://company_two/bSamp.html	http://company_two/fSamp.html
Demo	http://company_two/bDemo.html	http://company_two/fDemo.html
Feature	Finding list of available room, and booking preference room.	Finding list of available seats and booking preference seat.
Development environment	Java technology: Metro framework with Tomcat web container.	Java technology: Metro framework with Tomcat web container.
Access protocol	SOAP with link to WSDL file.	SOAP with link to WSDL file.
Operation	getAvailableRoom, bookingRoom	getAvailableSeat, bookingSeat
Compatibility	Web service	Web service
Interface	http://company_two/ws/BookingHotel?wsdl	http://company_two/ws/FlightBooking?wsdl
Price	5 baht per transaction.	5 baht per transaction.
Delivery	Transaction report in biweekly.	Transaction report in biweekly.

ตารางที่ 5.19 (ต่อ)

Business Information	
Name	Company Two Co. Ltd,
Address	75 Niphat-u-thit 3, Hat Yai, Songkhla 90110 Thailand
E-mail	Request@company_two.do
Tel.	+66 (074) 221246
Fax.	+66 (074) 221246
Mobile.	+66 (081) 7654321
Web site	http://company_one/index.php
Category	Travel agencies and service.
Description	The newest and the best travel agency in the southern.
Overview	Provide online hotel and airline reservation.
Reseller	Company Two Directed Sale.
Reference customer	Company Four for Fun.
Support tel.	+66 (074) 221246
Support fax.	+66 (074) 221246
Support mobile.	+66 (081) 7654321
Support E-mail	Support@company_two.do
Support web site	http://company_one/support/index.php

ผู้วิจัยจะนำข้อมูลของบริการและผู้ให้บริการดังกล่าวไปทดสอบกับระบบผู้ประสานงานการเจรจาต่อรองที่ได้นำเสนอมา เพื่อแสดงให้เห็นถึงการดำเนินงานของระบบที่ถูกต้องและสอดคล้องกับความต้องการที่กำหนด ซึ่งสามารถเสนอรายละเอียดได้แนบถัดไป

5.2.2 การพัฒนาหน่วยลงทะเบียนบริการ

ผู้วิจัยได้พัฒนาหน่วยลงทะเบียนบริการตามมาตรฐานของเว็บเซอร์วิสคือ Universal Description, Discovery and Integration หรือ UDDI (UDDI 2010) โดยเลือกระบบ UDDI ที่อยู่บนพื้นฐานของภาษา Java นั่นคือ Apache jUDDI (jUDDI 2010)

Apache jUDDI เป็นซอฟต์แวร์แบบรหัสเปิด (Open source) สำหรับการพัฒนาระบบการลงทะเบียนบริการตามข้อกำหนดมาตรฐานของเว็บเซอร์วิสโดยใช้ภาษาจาวา

ซึ่งสามารถทำงานได้ดีกับระบบให้บริการงานประยุกต์สำหรับเทคโนโลยีจาวาที่หลากหลาย เช่น Apache Tomcat, IBM Websphere เป็นต้น สำหรับงานวิจัยนี้ได้เลือก Apache Tomcat มาเป็นระบบให้บริการสำหรับงานประยุกต์ที่จะทำงานร่วมกับ Apache jUDDI การทดสอบการทำงานของหน่วยลงทะเบียนบริการที่พัฒนานั้นสามารถแสดงในรายละเอียดได้ในบทถัดไปเช่นเดียวกัน

5.3 สรุปท้ายบท

จากการพัฒนาระบบผู้ประสานงานการเจรจาต่อรองและกรณีตัวอย่างบริการการท่องเที่ยวแสดงให้เห็นได้ว่าสามารถนำแนวคิดและเทคโนโลยีต่างๆที่มีอยู่ในปัจจุบัน โดยเฉพาะอย่างยิ่งเทคโนโลยีจาวามาประยุกต์ใช้ เพื่อสร้างระบบงานที่สอดคล้องกับส่วนการออกแบบที่ได้เสนอในขั้นต้น สำหรับบทถัดไปจะนำกรณีตัวอย่างนี้ไปทดสอบการดำเนินการต่างๆกับระบบผู้ประสานงานการเจรจาต่อรองและเสนอผลการทดสอบ

บทที่ 6

การทดสอบระบบ

เนื้อหาในบทนี้กล่าวถึงการทดสอบการดำเนินงานของกรณีตัวอย่างว่าสามารถดำเนินการได้จริง และการทดสอบประสิทธิภาพของระบบผู้ประสานงานการเจรจาต่อรองที่พัฒนาขึ้นตามโครงสร้างการทำงานที่ออกแบบไว้

6.1 เครื่องมือที่ใช้ในการทดสอบ

1. เครื่องไมโครคอมพิวเตอร์สำหรับทำหน้าที่เป็นเครื่องแม่ข่ายให้บริการโปรแกรมประยุกต์ผ่านเว็บของระบบผู้ประสานงานการเจรจาต่อรอง กรณีตัวอย่างคือเว็บเซอร์วิสบริการการท่องเที่ยว และหน่วยลงทะเบียนบริการที่พัฒนาตามมาตรฐานยูดีดีไอ จำนวน 1 เครื่อง มีหน่วยประมวลผลกลาง Intel Pentium IV 3.20 GHz หน่วยความจำ 2 GB ระบบปฏิบัติการ Microsoft Windows XP Professional สำหรับเป็นระบบปฏิบัติการ
2. เครื่องไมโครคอมพิวเตอร์สำหรับทำหน้าที่เป็นเครื่องลูกข่ายเรียกใช้บริการต่างๆ จำนวน 1 เครื่อง มีหน่วยประมวลผลกลาง Intel Pentium IV 3.20 GHz หน่วยความจำ 1 GB ระบบปฏิบัติการ Microsoft Windows XP Professional สำหรับเป็นระบบปฏิบัติการ
3. Web-browser ต่างๆ เช่น Internet Explorer หรือ Mozilla Firefox เป็นต้น
4. Apache TCPMon (TCPMon, 2006) สำหรับการดักจับข้อมูลที่รับ-ส่งผ่านโปรโตคอล TCP และสามารถแสดงผลให้อยู่ในรูปแบบ XML ได้
5. เครือข่ายภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์

6.2 การทดสอบบริการกรณีตัวอย่าง

การทดสอบกรณีตัวอย่างบริการการท่องเที่ยวมีวัตถุประสงค์เพื่อทดสอบว่ากรณีตัวอย่างดังกล่าวนี้สามารถดำเนินการได้อย่างถูกต้อง กรณีตัวอย่างถูกพัฒนาในรูปแบบเว็บเซอร์วิส ซึ่งประกอบด้วย 2 ตัวอย่างและแต่ละตัวอย่างจะประกอบด้วย 2 บริการคือ บริการการจองโรงแรมและการจองเที่ยวบินตามที่ได้กล่าวไปแล้วในข้างต้น ดังนั้นการทดสอบการดำเนินงาน จะทำการส่งข่าวสารร้องขอ (SOAP Request) ไปยังเว็บเซอร์วิสแล้วพิจารณาผลตอบกลับ (SOAP Response) ว่าเป็นไปตามที่กำหนดในเอกสารอธิบายบริการ (WSDL) หรือไม่ โดยมีรายละเอียดต่างๆดังนี้

6.2.1 การทดสอบกรณีตัวอย่างที่หนึ่ง

กรณีตัวอย่างนี้ได้รับการพัฒนาให้ทำงานบน Glassfish Application Server ซึ่งตัว Glassfish นี้มีฟังก์ชันการทำงานที่บรรจุเอาไว้อย่างหนึ่งคือการทดสอบเว็บเซอร์วิส Glassfish จะดำเนินการสร้างหน้าเว็บที่สอดคล้องกับเว็บเซอร์วิสที่เปิดให้บริการอยู่ ส่งผลให้ผู้เรียกใช้สามารถใช้งานเว็บเซอร์วิสได้อย่างสะดวก และตัวทดสอบเว็บเซอร์วิสนี้ยังสามารถแสดงข้อมูลอื่นๆที่จำเป็นอีกด้วย ดังนั้นผู้วิจัยจึงเลือกตัวทดสอบเว็บเซอร์วิสของ Glassfish สำหรับการทดสอบกรณีตัวอย่างที่หนึ่ง

เมื่อพิจารณาถึงเอกสารรายละเอียดบริการของบริการรายที่หนึ่งแล้ว สามารถสร้างกรณีทดสอบและผลที่คาดว่าจะได้รับดังตารางที่ 6.1 และการแสดงการทดสอบตามกรณีต่างๆในภาพประกอบที่ 6.1 – 6.8

ตารางที่ 6.1 กรณีทดสอบบริการกรณีตัวอย่างที่หนึ่ง (Company One)

กรณีทดสอบ	ผลที่คาดว่าจะได้รับ
การเรียกใช้บริการ HotelService	
1. ส่ง SOAP Request ที่ไม่มีตัวแปรประกอบเพื่อเรียกใช้การดำเนินงาน getAvailableRoom	กลุ่มรายการของแต่ละห้องที่ยังว่าง โดยแต่ละรายการประกอบด้วย หมายเลขห้อง ชนิดของห้อง และราคาเช่าห้องต่อคืน
2. ส่ง SOAP Request เพื่อเรียกใช้การดำเนินงาน bookingRoom ที่ประกอบด้วย	ข้อมูลหมายเลขการจอง วันหมดเขตของการจองนี้ และราคาเช่าต่อหนึ่งคืน

ตารางที่ 6.1 (ต่อ)

กรณีทดสอบ	ผลที่คาดว่าจะได้รับ
เลขที่บัตรประชาชน ชื่อ ที่อยู่ เบอร์โทรศัพท์ หมายเลขห้อง และวันที่ต้องการเข้าพัก	
การเรียกใช้บริการ AirTicket	
3. ส่ง SOAP Request เพื่อเรียกใช้การ ดำเนินงาน getAvailableSeat ที่ประกอบด้วย รหัสสถานที่ต้นทาง รหัสสถานที่ปลายทาง วันที่เริ่มต้น และวันที่สิ้นสุด	กลุ่มรายการของแต่ละเที่ยวบิน โดยแต่ละ รายการประกอบด้วย หมายเลขเที่ยวบิน หมายเลขเครื่องบิน วันที่ออกเดินทาง วันที่ เดินทางถึง และราคาค่าโดยสาร
4. ส่ง SOAP Request เพื่อเรียกใช้การ ดำเนินงาน bookingSeat ที่ประกอบด้วย เลขที่บัตรประชาชน ชื่อ ที่อยู่ เบอร์โทรศัพท์ และหมายเลขเที่ยวบิน	ข้อมูลหมายเลขการจอง วันหมดเขตของการ จองนี้ และราคาค่าโดยสาร

HotelServiceService Web Service Tester

This form will allow you to test your web service implementation ([WSDL File](#))

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

Methods :

```
public abstract th.ac.psu.sea.serviceprovider.AvailableRoom th.ac.psu.sea.serviceprovider.HotelService.getAvailableRoom()
 
```

ภาพประกอบที่ 6.1 การเรียกใช้ส่วนดำเนินการ getAvailableRoom ของบริการ HotelService

getAvailableRoom Method invocation

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getAvailableRoomResponse xmlns:ns2="http://serviceprovider.sea.psu.ac.th/">
      <return>
        <listCount>4</listCount>
        <room>
          <price>1020.0</price>
          <roomId>600</roomId>
          <roomType>Single Bed</roomType>
        </room>
        <room>
          <price>1020.0</price>
          <roomId>602</roomId>
          <roomType>Single Bed</roomType>
        </room>
        <room>
          <price>1020.0</price>
          <roomId>603</roomId>
          <roomType>Single Bed</roomType>
        </room>
        <room>
          <price>1520.0</price>
          <roomId>705</roomId>
          <roomType>Twin Bed</roomType>
        </room>
        <room>
          <price>1520.0</price>
          <roomId>709</roomId>
          <roomType>Double Bed</roomType>
        </room>
      </return>
    </ns2:getAvailableRoomResponse>
  </S:Body>
</S:Envelope>
```

ภาพประกอบที่ 6.2 ผลลัพธ์จากการเรียกใช้ getAvailableRoom ของบริการ HotelService

HotelServiceService Web Service Tester

This form will allow you to test your web service implementation (WSDL File)

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

Methods :

```
public abstract th.ac.psu.sea.serviceprovider.ReserveRoom th.ac.psu.sea.serviceprovider.HotelService bookingRoom
(java.lang.String java.lang.String java.lang.String java.lang.String java.lang.String java.lang.String)
 (  ,  ,  ,  , 
18-02-2010 )
```

ภาพประกอบที่ 6.3 การเรียกใช้ส่วนดำเนินการ bookingRoom ของบริการ HotelService

bookingRoom Method invocation

Method parameter(s)

Type	Value
java.lang.String	3909800663459
java.lang.String	Chaiwat
java.lang.String	Hatyai
java.lang.String	074288572
java.lang.String	309
java.lang.String	18/02/2010

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:bookingRoomResponse xmlns:ns2="http://serviceprovider.sea.psu.ac.th/">
      <return>
        <bookingID>3909800663459309</bookingID>
        <expireBookingDate>2010-02-28</expireBookingDate>
        <price>1025.0</price>
      </return>
    </ns2:bookingRoomResponse>
  </S:Body>
</S:Envelope>
```

ภาพประกอบที่ 6.4 ผลลัพธ์จากการเรียกใช้ bookingRoom ของบริการ HotelService

AirTicketService Web Service Tester

This form will allow you to test your web service implementation ([WSDL File](#))

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

Methods :

public abstract th.ac.psu.sea.serviceprovider.AvailableSeat th.ac.psu.sea.serviceprovider.AirTicket.getAvailableSeat
(java.lang.String,java.lang.String,java.lang.String,java.lang.String)

getAvailableSeat (HDY , BKK , 13/07/2009 , 15/07/2009)

ภาพประกอบที่ 6.5 การเรียกใช้ส่วนดำเนินการ getAvailableSeat ของบริการ AirTicket

getAvailableSeat Method invocation

Method parameter(s)

Type	Value
java.lang.String	HDY
java.lang.String	BKK
java.lang.String	13 07 2009
java.lang.String	15 07 2009

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getAvailableSeatResponse xmlns:ns2="http://serviceprovider.sea.psu.ac.th/">
      <return>
        <flight>
          <airplaneID>F8232</airplaneID>
          <departureDate>2009-07-15</departureDate>
          <departureTime>05:15:00</departureTime>
          <fare>1576.00</fare>
          <flightID>05HDYF8232BKK</flightID>
        </flight>
        <flight>
          <airplaneID>F8234</airplaneID>
          <departureDate>2009-07-15</departureDate>
          <departureTime>11:50:00</departureTime>
          <fare>1576.00</fare>
          <flightID>06HDYF8234BKK</flightID>
        </flight>
        <flight>
          <airplaneID>F8238</airplaneID>
          <departureDate>2009-07-15</departureDate>
          <departureTime>20:20:00</departureTime>
          <fare>1576.00</fare>
          <flightID>07HDYF8238BKK</flightID>
        </flight>
      </return>
    </ns2:getAvailableSeatResponse>
  </S:Body>
</S:Envelope>
```

ภาพประกอบที่ 6.6 ผลลัพธ์จากการเรียกใช้ getAvailableSeat ของบริการ AirTicket

AirTicketService Web Service Tester

This form will allow you to test your web service implementation ([WSDL File](#))

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name

Methods :

public abstract th.ac.psu.sea.serviceprovider.ReserveSeat th.ac.psu.sea.serviceprovider.AirTicket.bookingSeat
(java.lang.String,java.lang.String,java.lang.String,java.lang.String)

bookingSeat (3909800663459 , Chaisat , Hatyai , 074288572 , 07HDYFG238BKK)

ภาพประกอบที่ 6.7 การเรียกใช้ส่วนดำเนินการ bookingSeat ของบริการ AirTicket

bookingSeat Method invocation

Method parameter(s)

Type	Value
java.lang.String	3909800663459
java.lang.String	Chaiwat
java.lang.String	Hatyai
java.lang.String	074288572
java.lang.String	07HDYPG238BKK

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:BookingSeatResponse xmlns:ns2="http://serviceprovider.sea.pw.ac.th/">
      <return>
        <bookingID>390980066345907HDYPG238BKK</bookingID>
        <expireBookingDate>2555-02-26</expireBookingDate>
        <fare>1576.1</fare>
      </return>
    </ns2:BookingSeatResponse>
  </S:Body>
</S:Envelope>
```

ภาพประกอบที่ 6.8 ผลลัพธ์จากการเรียกใช้ bookingSeat ของบริการ AirTicket

จากการทดสอบจะพบว่าการทำงานของบริการกรณีตัวอย่างที่หนึ่งสามารถทำงานได้ถูกต้องตามที่ออกแบบไว้และเป็นไปตามผลที่คาดว่าจะได้รับในทุกกรณี

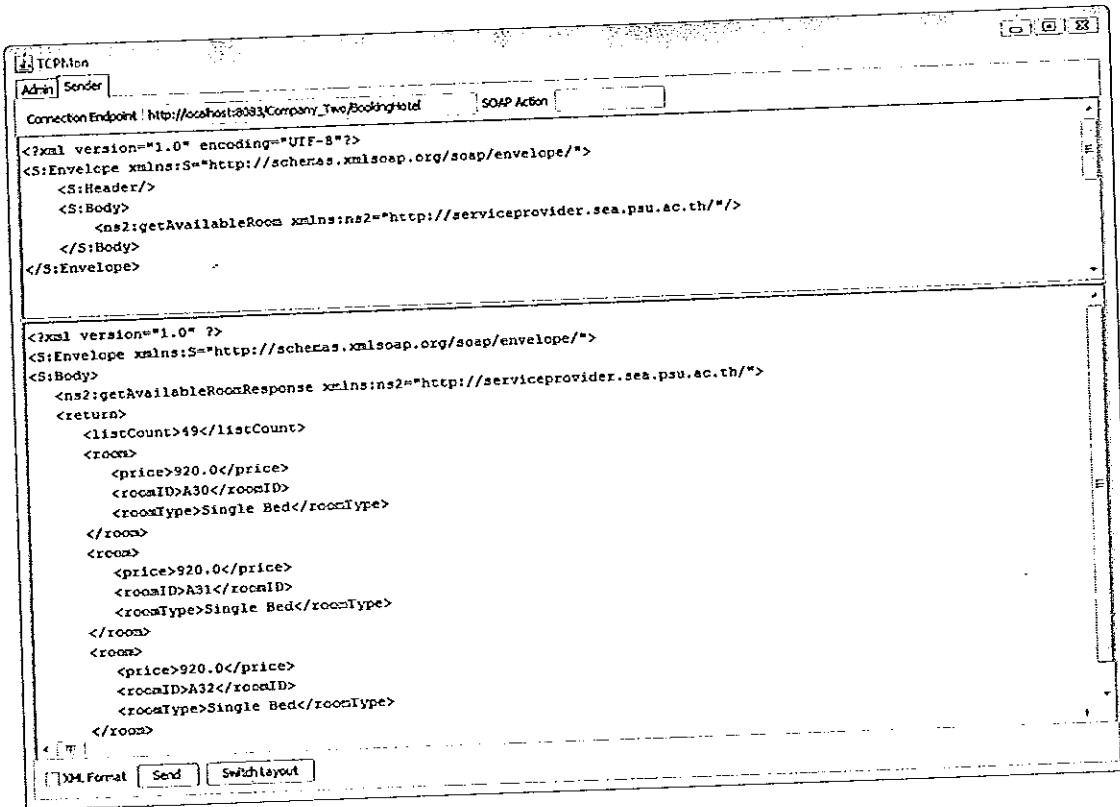
6.2.2 การทดสอบกรณีตัวอย่างที่สอง

กรณีตัวอย่างที่สองนี้ได้รับการพัฒนาให้ทำงานบน Apache Tomcat ซึ่งเป็นเพียงตัวช่วยให้เว็บสามารถทำงานร่วมกับเทคโนโลยีเว็บของจาวาได้เท่านั้น ไม่ได้มีส่วนฟังก์ชันการทดสอบเว็บเซอร์วิสเหมือนการพัฒนาบน Glassfish Application Server ดังนั้นผู้วิจัยจึงเลือกใช้เครื่องมือ Apache TCPMon ในการดักจับการรับ-ส่งข้อมูลบนโปรโตคอล TCP เพื่อให้สามารถแสดงผลข่าวสารร้องขอและผลลัพธ์ที่ตอบกลับจากเว็บเซอร์วิสได้

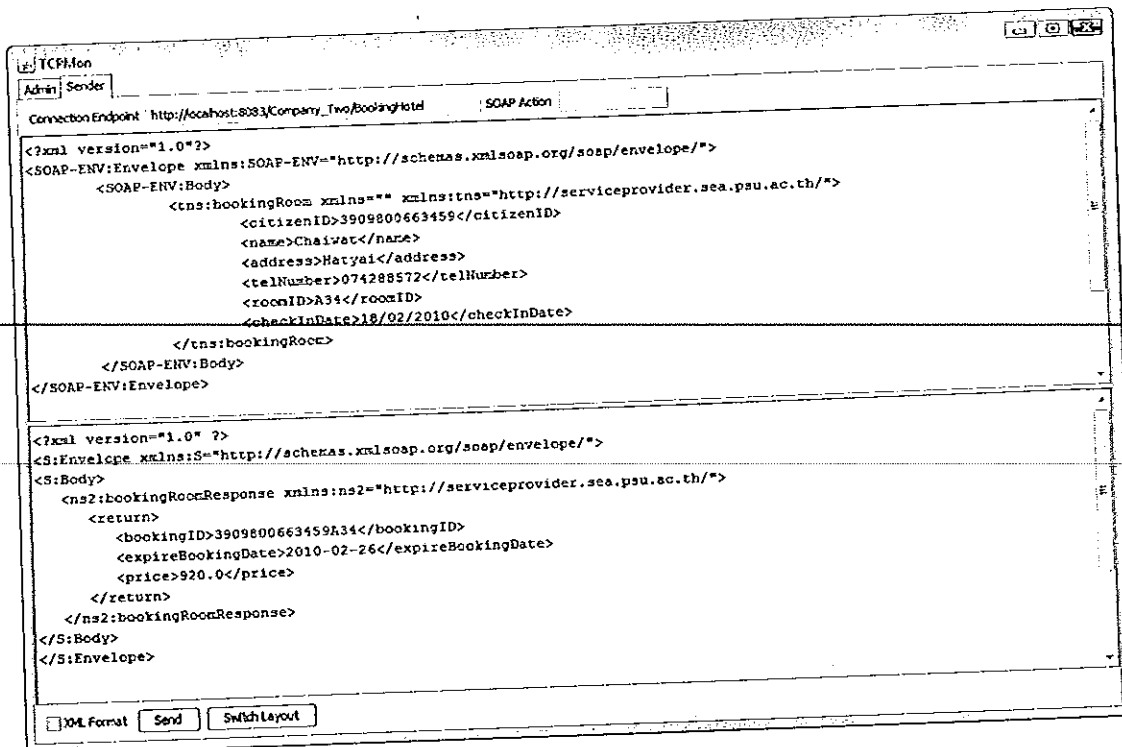
จากการพิจารณาถึงเอกสารรายละเอียดบริการของบริการรายที่สองแล้ว สามารถสร้างกรณีทดสอบและผลที่คาดว่าจะได้รับดังตารางที่ 6.2 และการแสดงการทดสอบตามกรณีต่างๆในภาพประกอบที่ 6.9 – 6.12 ภาพประกอบจะแสดงการดักจับข้อมูลโดยแสดงส่วนข่าวสารร้องขอในด้านบนและผลตอบกลับในด้านล่าง

ตารางที่ 6.2 กรณีทดสอบบริการกรณีตัวอย่างที่สอง (Company Two)

กรณีทดสอบ	ผลที่คาดว่าจะได้รับ
การเรียกใช้บริการ BookingHotel	
1. ส่ง SOAP Request ที่ไม่มีตัวแปรประกอบ เพื่อเรียกใช้การดำเนินงาน getAvailableRoom	กลุ่มรายการของแต่ละห้องที่ยังว่าง โดยแต่ละรายการประกอบด้วย หมายเลขห้อง ชนิดของห้อง และราคาเช่าห้องต่อคืน
2. ส่ง SOAP Request เพื่อเรียกใช้การดำเนินงาน bookingRoom ที่ประกอบด้วย เลขที่บัตรประชาชน ชื่อ ที่อยู่ เบอร์โทรศัพท์ หมายเลขห้อง และวันที่ต้องการเข้าพัก	ข้อมูลหมายเลขการจอง วันหมดเขตของการจองนี้ และราคาเช่าต่อหนึ่งคืน
การเรียกใช้บริการ FlightBooking	
3. ส่ง SOAP Request เพื่อเรียกใช้การดำเนินงาน getAvailableSeat ที่ประกอบด้วย รหัสสถานที่ต้นทาง รหัสสถานที่ปลายทาง วันที่เริ่มต้น และวันที่สิ้นสุด	กลุ่มรายการของแต่ละเที่ยวบิน โดยแต่ละรายการประกอบด้วย หมายเลขเที่ยวบิน หมายเลขเครื่องบิน วันที่ออกเดินทาง วันที่เดินทางถึง และราคาค่าโดยสาร
4. ส่ง SOAP Request เพื่อเรียกใช้การดำเนินงาน bookingSeat ที่ประกอบด้วย เลขที่บัตรประชาชน ชื่อ ที่อยู่ เบอร์โทรศัพท์ และหมายเที่ยวบิน	ข้อมูลหมายเลขการจอง วันหมดเขตของการจองนี้ และราคาค่าโดยสาร



ภาพประกอบที่ 6.9 การเรียกใช้ getAvailableRoom ของบริการ BookingHotel และผลลัพธ์



ภาพประกอบที่ 6.10 การเรียกใช้ bookingRoom ของบริการ BookingHotel และผลลัพธ์

The screenshot shows a SOAP request and response in TCPMon. The request is a `tns:getAvailableSeat` call with parameters for departure location (HDY), arrival location (BKK), start date (13/07/2009), and end date (15/07/2009). The response contains two flight options with details like airplane ID, departure date, time, and fare.

```

<?xml version="1.0"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <tns:getAvailableSeat xmlns="" xmlns:tns="http://serviceprovider.sea.psu.ac.th/">
      <departureLocation>HDY</departureLocation>
      <arrivalLocation>BKK</arrivalLocation>
      <startDate>13/07/2009</startDate>
      <endDate>15/07/2009</endDate>
    </tns:getAvailableSeat>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

<?xml version="1.0" ?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
  <ns2:getAvailableSeatResponse xmlns:ns2="http://serviceprovider.sea.psu.ac.th/">
    <return>
      <flight>
        <airplaneID>FD232</airplaneID>
        <departureDate>2009-07-13</departureDate>
        <departureTime>09:45:00</departureTime>
        <fare>1800.00</fare>
        <flightID>05HDYFD232BKK</flightID>
      </flight>
      <flight>
        <airplaneID>FD234</airplaneID>
        <departureDate>2009-07-13</departureDate>
        <departureTime>14:00:00</departureTime>
        <fare>1800.00</fare>
        <flightID>06HDYFD234BKK</flightID>
      </flight>
    </return>
  </ns2:getAvailableSeatResponse>
</S:Body>
</S:Envelope>

```

ภาพประกอบที่ 6.11 การเรียกใช้ `getAvailableSeat` ของบริการ FlightBooking และผลลัพธ์

The screenshot shows a SOAP request and response in TCPMon. The request is a `tns:BookingSeat` call with user details (citizen ID, name, address, tel number) and flight ID. The response returns a `BookingSeatResponse` with booking ID, expire date, and fare.

```

<?xml version="1.0"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <tns:BookingSeat xmlns="" xmlns:tns="http://serviceprovider.sea.psu.ac.th/">
      <citizenID>3909800663459</citizenID>
      <name>Chaiwat</name>
      <address>Hatyai</address>
      <telNumber>074288572</telNumber>
      <flightID>05HDYFD232BKK</flightID>
    </tns:BookingSeat>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

<?xml version="1.0" ?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
  <ns2:BookingSeatResponse xmlns:ns2="http://serviceprovider.sea.psu.ac.th/">
    <return>
      <bookingID>390980066345906HDYFD234BKK</bookingID>
      <expireBookingDate>2010-02-26</expireBookingDate>
      <fare>1800.0</fare>
    </return>
  </ns2:BookingSeatResponse>
</S:Body>
</S:Envelope>

```

ภาพประกอบที่ 6.12 การเรียกใช้ `bookingSeat` ของบริการ FlightBooking และผลลัพธ์

จากการทดสอบจะพบว่าการทำงานของบริการกรณีตัวอย่างที่สองสามารถทำงานได้ถูกต้องตามที่ออกแบบไว้และเป็นไปตามผลที่คาดว่าจะได้รับในทุกกรณี

6.3 การทดสอบหน่วยลงทะเบียบริการ

หน่วยลงทะเบียบริการที่ได้เลือกใช้ในงานวิจัยนี้คือ Apache jUDDI ซึ่งรายละเอียดต่างๆได้กล่าวมาแล้วในข้างต้น jUDDI จะบรรจุหน้าเว็บสำหรับแสดงข้อมูลรายละเอียดต่างๆและหน้าเว็บสำหรับการเรียกใช้งานมาให้ในระดับหนึ่ง โดยเมื่อพิจารณาในการทดสอบหน่วยลงทะเบียบริการนี้ สามารถแบ่งออกได้เป็น 3 การทดสอบได้แก่ การทดสอบการพร้อมใช้งานโดยใช้หน้าเว็บแสดงรายละเอียดของระบบ การทดสอบการบันทึกข้อมูลโดยใช้โปรแกรมดักจับข้อมูล TCPMon และการทดสอบการค้นหาข้อมูลซึ่งจะเรียกใช้โปรแกรม TCPMon ด้วยเช่นกัน

6.3.1 การทดสอบความพร้อมใช้งาน

jUDDI มีฟังก์ชันที่ช่วยให้ผู้ใช้งานสามารถเรียกตรวจสอบความพร้อมในการทำงานของระบบ ได้โดยเลือกเข้าหน้าเว็บที่ jUDDI เปิดให้บริการอยู่ ดังนั้นผู้วิจัยจึงได้เลือกเพื่อตรวจสอบความพร้อมใช้งานของระบบดังแสดงในภาพประกอบที่ 6.13

จากภาพแสดงให้เห็นตัวอย่างส่วนหนึ่งของข้อมูลรายละเอียดแสดงความพร้อมใช้งาน โดยเป็นข้อมูลการค้นหาส่วนทำงานต่างๆที่ jUDDI ต้องการเรียกใช้ในการทำงานว่าพบเจอหรือไม่ ซึ่งผลจากการทดสอบปรากฏว่าพบเจอส่วนต่างๆครบสมบูรณ์ ดังนั้นระบบลงทะเบียบริการนี้พร้อมสำหรับการใช้งาน

jUDDI

Happy jUDDI!

jUDDI Version Information

jUDDI Version: 2.0rc6
 UDDI Version: 2.0

jUDDI Dependencies: Class Files & Libraries

```
Looking for: org.apache.juddi.IRegistry
+Found in: C:\Tomcat_5_5_23\webapps\juddi\WEB-INF\lib\juddi-2.0rc6.jar
Looking for: org.apache.axis.transport.http.AxisServlet
+Found in: C:\Tomcat_5_5_23\common\lib\axis-1.4.jar
Looking for: org.apache.commons.discovery.Resource
+Found in: C:\Tomcat_5_5_23\common\lib\commons-discovery-0.2.jar
Looking for: org.apache.commons.logging.Log
+Found in: C:\Tomcat_5_5_23\bin\commons-logging-api.jar
Looking for: org.apache.log4j.Layout
+Found in: C:\Tomcat_5_5_23\webapps\juddi\WEB-INF\lib\log4j-1.2.13.jar
Looking for: javax.xml.scap.SOAPMessage
+Found in an unknown location
Looking for: javax.xml.soap.Service
+Found in: C:\Tomcat_5_5_23\common\lib\axis-jaxrpc-1.4.jar
Looking for: com.ibm.wsdl.factory.WSDLFactoryImpl
+Found in: C:\Tomcat_5_5_23\webapps\juddi\WEB-INF\lib\wsdl4j-1.6.2.jar
Looking for: javax.xml.parsers.SAXParserFactory
+Found in an unknown location
```

ภาพประกอบที่ 6.13 ตัวอย่างหน้าเว็บที่แสดงรายละเอียดของระบบลงทะเบียนบริการ jUDDI

6.3.2 การทดสอบการบันทึกข้อมูล

การเรียกใช้งานหน่วยลงทะเบียนบริการ jUDDI จะเป็นไปตามมาตรฐาน UDDI ที่กำหนดขึ้นกล่าวคือมีข่าวสารการร้องขอเรียกใช้งานอยู่ในรูปแบบมาตรฐาน SOAP และผลลัพธ์ที่ได้ก็อยู่ในรูปแบบมาตรฐาน SOAP เช่นเดียวกัน ดังนั้นจากการพิจารณา UDDI สามารถสร้างกรณีทดสอบและผลที่คาดว่าจะได้รับดังตารางที่ 6.3 และการแสดงการทดสอบตามกรณีต่างๆในภาพประกอบที่ 6.14 – 6.15 ซึ่งใช้โปรแกรม TCPMon สำหรับดักจับข้อมูลมาแสดง ข้อมูลที่กำหนดในการทดสอบคือข้อมูลของกรณีตัวอย่างนั่นเอง

จากภาพประกอบแสดงให้เห็นถึงข่าวสารการเรียกใช้งานและผลตอบกลับในรูปแบบของมาตรฐาน SOAP ซึ่งทำให้สรุปได้ว่าหน่วยลงทะเบียนบริการสามารถทำการบันทึกข้อมูลได้ถูกต้องและเป็นไปตามผลที่คาดว่าจะได้รับในทุกกรณี

ตารางที่ 6.3 กรณีทดสอบการบันทึกข้อมูลลงหน่วยลงทะเบียนบริการ

กรณีทดสอบ	ผลที่คาดว่าจะได้รับ
1. ส่ง SOAP Request เพื่อเรียกใช้การบันทึกข้อมูลของผู้ให้บริการ ที่ประกอบด้วย ชื่อ รายละเอียด ชื่อบุคคลที่ติดต่อได้ เบอร์โทรศัพท์ และอีเมล	ข้อมูลแสดงรหัสของผู้ให้บริการ URL ในการเข้าถึง และข้อมูลรายละเอียดที่กำหนดทวนซ้ำอีกครั้ง
2. ส่ง SOAP Request เพื่อเรียกใช้การบันทึกข้อมูลของบริการ ที่ประกอบด้วย ชื่อ รายละเอียด และตำแหน่งที่ตั้งไฟล์เอกสารอธิบายรายละเอียดบริการ (WSDL)	ข้อมูลแสดงรหัสบริการ และข้อมูลรายละเอียดที่กำหนดทวนซ้ำอีกครั้ง

```

TCPMon
Admin Sender
Connection Endpoint http://localhost:8080/juddi/admin SOAP Action
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <save_business generic="2.0" xmlns="urn:uddi-org:api_v2">
      <authInfo>authToken:7132ECB0-1BA9-11DF-B296-96666B9B6B7D</authInfo>
      <businessEntity businessKey="">
        <name>Company One</name>
        <description>The travel agency company in the south of Thailand.</description>
        <contacts>
          <contact useType="sales contact">
            <personName>Mike One</personName>
            <phone>+66 (074) 288572</phone>
            <email>Support@company_one.do</email>
          </contact>
        </contacts>
      </businessEntity>
    </save_business>
  </soapenv:Body>
</soapenv:Envelope>

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body xmlns="urn:uddi-org:api_v2">
    <businessDetail generic="2.0" <operator="jUDDI.org">
      <businessEntity authorizedName="juddi_usr" businessKey="BD430300-1BAA-11DF-B296-B479526C9DC4" operator="jUDDI.org">
        <discoveryURLs>
          <discoveryURL useType="businessEntity">http://localhost:8080/juddi/uddiget.jsp?businesskey=BD430300-1BAA-11DF-
        </discoveryURLs>
        <name>Company One</name>
        <description>The travel agency company in the south of Thailand.</description>
        <contacts>
          <contact useType="sales contact">
            <personName>Mike One</personName>
            <phone>+66 (074) 288572</phone>
            <email>Support@company_one.do</email>
          </contact>
        </contacts>
      </businessEntity>
    </businessDetail>
  </soapenv:Body>
</soapenv:Envelope>
XML Format Send Switch Layout
  
```

ภาพประกอบที่ 6.14 การเรียกใช้การบันทึกข้อมูลผู้ให้บริการและผลลัพธ์ที่ได้กลับมา

```

TCPMon
Admin | Sender | SOAP Action |
Connection Endpoint: http://localhost:8082/juddi/svrIn
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <save_service generic="2.0" xmlns="urn:uddi-org:api_v2">
      <authInfo>authToken:7132ECB0-1BA9-11DF-B296-96666B9B6B7D</authInfo>
      <businessService businessKey="BD430300-1BAA-11DF-B296-B479526C9DC4" serviceKey="">
        <name>HotelService</name>
        <description>Find the appropriate quarterage for rooms and booking it.</description>
        <bindingTemplates>
          <bindingTemplate bindingKey="">
            <accessPoint URLType="http">http://localhost:8082/Company_One/HotelServiceService?wsdl</accessPoint>
          </bindingTemplate>
        </bindingTemplates>
      </businessService>
    </save_service>
  </soapenv:Body>
</soapenv:Envelope>

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body xmlns="urn:uddi-org:api_v2">
    <serviceDetail generic="2.0" operator="jUDDI.org">
      <businessService businessKey="BD430300-1BAA-11DF-B296-B479526C9DC4" serviceKey="3A23A170-1BAD-11DF-B296-BC743EBFC">
        <name>HotelService</name>
        <description>Find the appropriate quarterage for rooms and booking it.</description>
        <bindingTemplates>
          <bindingTemplate bindingKey="3A2ACD60-1BAD-11DF-B296-D2C3BA12E73D" serviceKey="3A23A170-1BAD-11DF-B296-BC743EBFC">
            <accessPoint URLType="http">http://localhost:8082/Company_One/HotelServiceService?wsdl</accessPoint>
          </bindingTemplate>
        </bindingTemplates>
      </businessService>
    </serviceDetail>
  </soapenv:Body>
</soapenv:Envelope>
  
```

ภาพประกอบที่ 6.15 การเรียกใช้การบันทึกข้อมูลของบริการและผลลัพธ์ที่ได้กลับมา

6.3.3 การทดสอบการค้นหาข้อมูล

และจากการพิจารณามาตรฐาน UDDI เหมือนดังข้างต้นสามารถสร้างกรณีทดสอบและผลที่คาดว่าจะได้รับดังตารางที่ 6.4 และการแสดงการทดสอบตามกรณีต่างๆในภาพประกอบที่ 6.16 – 6.18 ซึ่งใช้โปรแกรม TCPMon สำหรับดักจับข้อมูลมาแสดงด้วยเช่นกัน

ตารางที่ 6.4 กรณีทดสอบการค้นหาข้อมูลลงหน่วยลงทะเบียนบริการ

กรณีทดสอบ	ผลที่คาดว่าจะได้รับ
1. ส่ง SOAP Request เพื่อเรียกใช้การค้นหาข้อมูลของผู้ให้บริการ ที่ประกอบด้วย ชื่อของผู้ให้บริการ พร้อมทั้งลักษณะของการค้นหา	กลุ่มข้อมูลรายการที่ประกอบด้วยรหัสของผู้ให้บริการ ชื่อของผู้ให้บริการ และรายการบริการที่ผู้ให้บริการรายนั้นกำหนด ซึ่งก็ประกอบด้วยรหัสบริการและชื่อบริการ

ตารางที่ 6.4 (ต่อ)

กรณีทดสอบ	ผลที่คาดว่าจะได้รับ
2. ส่ง SOAP Request เพื่อเรียกใช้การค้นหาข้อมูลของบริการ ที่ประกอบด้วย รหัสของผู้ให้บริการ และชื่อบริการ พร้อมทั้งลักษณะของการค้นหา	กลุ่มข้อมูลรายการที่ประกอบด้วยรหัสของบริการ และชื่อของบริการ
3. ส่ง SOAP Request เพื่อเรียกใช้การค้นหาข้อมูลการเรียกใช้บริการ ที่ประกอบด้วย รหัสของบริการ พร้อมทั้งลักษณะของการค้นหา	ข้อมูลตำแหน่งที่ตั้งไฟล์เอกสารอธิบายรายละเอียดบริการ (WSDL)

The screenshot shows a SOAP client interface with two panels. The top panel displays a SOAP request XML, and the bottom panel displays the corresponding SOAP response XML.

```

Connection Endpoint: http://localhost:8084/uddi/req.rwy
SOAP Action

<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <find_business maxRows="100" generic="2.0" xmlns="urn:uddi-org:api_v2">
      <findQualifiers>
        <findQualifier>exactNameMatch</findQualifier>
      </findQualifiers>
      <name>Company One</name>
    </find_business>
  </soapenv:Body>
</soapenv:Envelope>

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body xmlns="urn:uddi-org:api_v2">
    <businessList generic="2.0" operator="jUBDI.org">
      <businessInfos>
        <businessInfo businessKey="BD430300-1BAA-11DF-B296-B479526C9DC4">
          <name>Company One</name>
          <description>The travel agency company in the south of Thailand.</description>
          <serviceInfos>
            <serviceInfo businessKey="BD430300-1BAA-11DF-B296-B479526C9DC4" serviceKey="3A23A170-1BAD-11DF-B296-BC743EBFC">
              <name>KoralService</name>
            </serviceInfo>
          </serviceInfos>
        </businessInfo>
      </businessInfos>
    </businessList>
  </soapenv:Body>
</soapenv:Envelope>
  
```

At the bottom of the interface, there are buttons for "XML Format", "Send", and "Switch Layout".

ภาพประกอบที่ 6.16 การเรียกใช้การค้นหาข้อมูลของผู้ให้บริการและผลลัพธ์ที่ได้กลับมา

The screenshot shows a SOAP client window with the following XML content:

```

<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <find_service businessKey="BD430300-1BAA-11DF-B296-B479526C9DC4" generic="2.0" xmlns="urn:uddi-org:api_v2">
      <findQualifiers>
        <findQualifier>exactNameMatch</findQualifier>
      </findQualifiers>
      <name>HotelService</name>
    </find_service>
  </soapenv:Body>
</soapenv:Envelope>

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body xmlns="urn:uddi-org:api_v2">
    <serviceList generic="2.0" operator="jUDDI.org">
      <serviceInfo>
        <serviceInfo businessKey="BD430300-1BAA-11DF-B296-B479526C9DC4" serviceKey="3A23A170-1BAD-11DF-B296-BC743EBFC55F">
          <name>HotelService</name>
        </serviceInfo>
      </serviceInfo>
    </serviceList>
  </soapenv:Body>
</soapenv:Envelope>

```

ภาพประกอบที่ 6.17 การเรียกใช้การค้นหาข้อมูลของบริการและผลลัพธ์ที่ได้กลับมา

The screenshot shows a SOAP client window with the following XML content:

```

<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <find_binding serviceKey="3A23A170-1BAD-11DF-B296-BC743EBFC55F" maxRows="100" generic="2.0" xmlns="urn:uddi-org:api_v2">
      <findQualifiers>
        <findQualifier>exactNameMatch</findQualifier>
      </findQualifiers>
    </find_binding>
  </soapenv:Body>
</soapenv:Envelope>

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body xmlns="urn:uddi-org:api_v2">
    <bindingDetail generic="2.0" operator="jUDDI.org">
      <bindingTemplate bindingKey="3A2ACB60-1BAD-11DF-B296-D2C38A12E73D" serviceKey="3A23A170-1BAD-11DF-B296-BC743EBFC55F">
        <accessPoint URLType="http">http://localhost:8082/Company_One/HotelServiceServiceWsdlt/accessPoint</accessPoint>
      </bindingTemplate>
    </bindingDetail>
  </soapenv:Body>
</soapenv:Envelope>

```

ภาพประกอบที่ 6.18 การเรียกใช้การค้นหาข้อมูลการเรียกใช้บริการและผลลัพธ์ที่ได้กลับมา

จากภาพประกอบทำให้สรุปได้ว่าหน่วยลงทะเบียนบริการสามารถทำการค้นหาข้อมูลต่างๆได้ถูกต้องและเป็นไปตามผลที่คาดว่าจะได้รับในทุกกรณี

6.4 การทดสอบระบบผู้ประสานงานการเจรจาต่อรอง

ระบบผู้ประสานงานการเจรจาต่อรองถูกพัฒนาในรูปแบบโปรแกรมประยุกต์ผ่านเว็บ โดยประกอบด้วยสองบทบาทที่สามารถเรียกใช้งานได้คือผู้ให้บริการและผู้เรียกใช้บริการ ซึ่งในแต่ละบทบาทก็จะมีฟังก์ชันการทำงานที่ต่างกันขึ้นอยู่กับข้อกำหนดสำหรับการดำเนินการที่ได้กำหนด ดังนั้นในการทดสอบการดำเนินการจะเป็นการทดสอบการเรียกใช้งานที่ต้องสอดคล้องกับความต้องการของระบบตามที่ได้ออกแบบเอาไว้

จากการพิจารณาความต้องการของระบบซึ่งก็คือรายละเอียดกิจกรรมของระบบ ในบทที่ 4 หัวข้อที่ 4.2 ผู้วิจัยสามารถสร้างกรณีทดสอบและผลที่คาดว่าจะได้รับดังตารางที่ 6.5 และ 6.6 ซึ่งเป็นกรณีทดสอบสำหรับบทบาทผู้เรียกใช้บริการและผู้ให้บริการตามลำดับ

ตารางที่ 6.5 กรณีทดสอบการทำงานของระบบในบทบาทผู้เรียกใช้บริการ

กรณีทดสอบ	ผลที่คาดว่าจะได้รับ
1. ผู้เรียกใช้บริการลงทะเบียนสมัครเป็นสมาชิกกับระบบโดยการกรอกข้อมูลที่เหมาะสม	การบันทึกข้อมูลรายละเอียดของผู้เรียกใช้บริการลงในระบบ และแสดงข่าวสารแจ้งผู้เรียกใช้บริการให้รับทราบ
2. ผู้เรียกใช้บริการเข้าสู่ระบบโดยระบุชื่อผู้เข้าใช้และรหัสผ่าน	หน้าจอหลักที่แสดงรายการสำหรับการเลือกใช้งานต่างๆ
3. ผู้เรียกใช้บริการกำหนดความต้องการบริการผ่านทางช่องกรอกข้อมูลที่ระบบกำหนดให้	ระบบสร้างเอกสารข้อกำหนดความต้องการบริการขึ้นมา และทำการค้นหาบริการต่างๆในระบบ ถ้าไม่พบระบบจะแสดงหน้าจอสำหรับกำหนดการอนุญาตให้ประกาศข้อกำหนดความต้องการบริการนี้ออกสู่สาธารณะ ท้ายสุดระบบจะแจ้งข่าวสารให้ผู้เรียกใช้บริการรับทราบ

ตารางที่ 6.5 (ต่อ)

กรณีทดสอบ	ผลที่คาดว่าจะได้รับ
4. ผู้เรียกใช้บริการเรียกดูเอกสารข้อกำหนดความต้องการที่ตนเองได้กำหนด	ระบบจะแสดงรายการความต้องการบริการ พร้อมทั้งแสดงเอกสารข้อกำหนดความต้องการบริการ
5. ผู้เรียกใช้บริการคัดเลือกข้อเสนอบริการจากผู้ให้บริการรายต่างๆตามความต้องการที่กำหนด	ระบบจะแสดงข่าวสารแจ้งให้ผู้เรียกใช้บริการรับทราบ
6. ผู้เรียกใช้บริการเลือกยอมรับร่างข้อตกลงบริการและ/หรือเอกสารข้อตกลงอื่นๆที่ผู้ให้บริการสร้างขึ้น	ระบบจะแสดงข่าวสารแจ้งให้ผู้เรียกใช้บริการรับทราบ
7. ผู้เรียกใช้บริการเรียกดูข้อตกลงบริการที่ตนตกลงไว้กับผู้ให้บริการรายต่างๆ	ระบบจะแสดงรายการข้อตกลงบริการ พร้อมทั้งแสดงเอกสารข้อตกลงบริการ
8. ผู้เรียกใช้บริการเรียกดูบริการที่ผู้ให้บริการแจ้งความพร้อมสำหรับเรียกใช้งาน	ระบบจะแสดงรายการของบริการที่ผู้ให้บริการแจ้งการส่งมอบ

ตารางที่ 6.6 กรณีทดสอบการทำงานของระบบในบทบาทผู้ให้บริการ

กรณีทดสอบ	ผลที่คาดว่าจะได้รับ
1. ผู้ให้บริการลงทะเบียนสมัครเป็นสมาชิกกับระบบโดยการกรอกข้อมูลที่เหมาะสม	การบันทึกข้อมูลรายละเอียดของผู้ให้บริการลงในระบบ และแสดงข่าวสารแจ้งผู้ให้บริการรับทราบ
2. ผู้ให้บริการเข้าสู่ระบบโดยระบุชื่อผู้เข้าใช้และรหัสผ่าน	หน้าจอหลักที่แสดงรายการสำหรับการเลือกใช้งานต่างๆ
3. ผู้ให้บริการกำหนดข้อมูลบริการที่ตนมีอยู่ในช่องกรอกข้อมูลที่ระบบกำหนดให้	การบันทึกข้อมูลรายละเอียดของบริการลงในระบบ และแสดงข่าวสารแจ้งผู้ให้บริการรับทราบ
4. ผู้ให้บริการเรียกข้อกำหนดความต้องการบริการที่ผู้เรียกใช้บริการอนุญาตให้ประกาศสู่สาธารณะ	ระบบจะแสดงรายการของข้อกำหนดความต้องการบริการ พร้อมทั้งแสดงเอกสารข้อกำหนดความต้องการบริการ
5. ผู้ให้บริการสร้างข้อเสนอบริการ	ระบบสร้างเอกสารข้อเสนอบริการ และแสดงข่าวสารแจ้งให้ผู้ให้บริการรับทราบ

ตารางที่ 6.6 (ต่อ)

กรณีทดสอบ	ผลที่คาดว่าจะได้รับ
6. ผู้ให้บริการสร้างร่างข้อตกลงบริการพร้อมทั้งแนบเอกสารข้อตกลงอื่นๆ	ระบบจะแสดงข่าวสารแจ้งให้ผู้ให้บริการรับทราบ
7. ผู้ให้บริการเรียกดูข้อตกลงบริการที่ตนตกลงไว้กับผู้เรียกใช้บริการรายต่างๆ	ระบบจะแสดงรายการข้อตกลงบริการ พร้อมทั้งแสดงเอกสารข้อตกลงบริการ
8. ผู้ให้บริการกำหนดการส่งมอบบริการให้แก่ผู้เรียกใช้บริการ	ระบบจะแสดงข่าวสารแจ้งให้ผู้ให้บริการรับทราบ
9. ผู้ให้บริการเรียกดูบริการที่ได้กำหนดการส่งมอบเรียบร้อยแล้ว	ระบบจะแสดงรายการของบริการที่ผู้ให้บริการแจ้งการส่งมอบเรียบร้อยแล้ว

จากการทดสอบการดำเนินการต่างๆโดยพิจารณาจากผลตอบรับที่แสดงและการทำงานของระบบ พบว่าระบบสามารถทำงานได้ถูกต้องตามที่ได้ออกแบบไว้และเป็นไปตามผลที่คาดว่าจะได้รับในทุกกรณีไม่ว่าจะเป็นบทบาทผู้เรียกใช้บริการหรือบทบาทผู้ให้บริการ ซึ่งสรุปได้ว่าระบบสามารถใช้งานได้จริง

6.5 สรุปท้ายบท

จากการทดสอบระบบผู้ประสานงานการเจรจาต่อรองและกรณีตัวอย่างบริการการท่องเที่ยวที่ถูกพัฒนาขึ้นตามโครงสร้างการทำงานที่ออกแบบไว้แสดงให้เห็นได้ว่า สามารถนำโครงสร้างการทำงานดังกล่าวไปประยุกต์ใช้ในการพัฒนาระบบตัวกลางที่คอยดำเนินการสนับสนุนการเจรจาต่อรองได้ โดยจะกล่าวถึงบทสรุปของงานวิจัยนี้ในบทถัดไป

บทที่ 7

บทสรุปและข้อเสนอแนะ

จากเนื้อหางานวิจัยทั้งหมดที่นำเสนอในรายงานนี้ ได้กล่าวถึงความเป็นมาของงานวิจัยและศึกษาแนวคิดที่เกี่ยวข้อง นำแนวคิดและหลักการที่ได้มาวิเคราะห์และออกแบบระบบงาน จนถึงการพัฒนาระบบพร้อมทั้งการทดสอบใช้งาน ดังนั้นสำหรับในบทนี้จะกล่าวถึงผลสรุปในการทำวิจัย ปัญหาและอุปสรรคในการวิจัย รวมถึงข้อเสนอแนะสำหรับผู้สนใจในการพัฒนาระบบผู้ประสานงานการเจรจาต่อรองในอนาคต

7.1 สรุปผลการวิจัย

แนวคิดสถาปัตยกรรมเชิงบริการนับว่าเป็นแนวคิดหนึ่งที่มีความนิยมและรู้จักกันอย่างแพร่หลายสำหรับการพัฒนาระบบสารสนเทศต่างๆ ในยุคปัจจุบัน และการเจรจาต่อรองก็เป็นปัจจัยหนึ่งที่ช่วยให้การขับเคลื่อนธุรกิจภายใต้บริบทของแนวคิดสถาปัตยกรรมเชิงบริการสามารถดำเนินการได้อย่างสอดคล้องจากผู้มีส่วนได้ส่วนเสียที่เกี่ยวข้อง ดังนั้นงานวิจัยนี้ได้เสนอกรอบแนวคิดและโครงสร้างการทำงานของ การเจรจาต่อรองสำหรับสถาปัตยกรรมเชิงบริการ พร้อมทั้งพัฒนาระบบสนับสนุนการเจรจาต่อรองที่สอดคล้องกับกรอบแนวคิดดังกล่าว

การวิเคราะห์กรอบแนวคิดการทำงานเริ่มต้นจากการศึกษามาตรฐานที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์ที่มีอยู่จำนวนหนึ่ง ซึ่งพบว่ามาตรฐานเหล่านี้ไม่ได้มีการกล่าวถึงขั้นตอนการเจรจาต่อรองสำหรับสถาปัตยกรรมเชิงบริการ แต่ปรากฏแนวทางที่เกี่ยวข้องกับการจัดหาและคัดเลือกผลิตภัณฑ์ซอฟต์แวร์มาใช้งานซึ่งสามารถที่จะขยายให้ครอบคลุมแนวทางการพัฒนาซอฟต์แวร์เชิงบริการได้ในอนาคต และงานวิจัยนี้ได้นำความรู้พื้นฐานนี้มาช่วยเป็นแนวทางในการออกแบบโครงสร้างและขั้นตอนการทำงานของระบบด้วย

กรอบแนวคิดการเจรจาต่อรองที่สนับสนุนการพัฒนาซอฟต์แวร์ตามสถาปัตยกรรมเชิงบริการที่ได้ออกแบบจะช่วยอำนวยความสะดวกให้แก่ทั้งผู้ให้บริการและผู้เรียกใช้บริการสามารถเจรจาต่อรองระหว่างกันได้ผ่านเครือข่ายอินเทอร์เน็ต โดยการ

ดำเนินการทั้งหมดกระทำผ่านระบบตัวกลางซึ่งเรียกว่าระบบผู้ประสานงานการเจรจาต่อรอง และกระบวนการเจรจาต่อรองนั้นจะมีความทำงานแบ่งออกเป็น 5 ระยะคือ ระยะการวิเคราะห์ความต้องการ ระยะการคัดเลือก ระยะการเสนอ ระยะการเจรจาต่อรอง และระยะการส่งมอบ ซึ่งมีรายละเอียดแตกต่างกันตามที่ได้กล่าวมาแล้ว

ระบบผู้ประสานงานการเจรจาต่อรองประกอบด้วยส่วนการทำงานหลักทั้งหมด

7 ส่วนได้แก่

- (1) ส่วนนำเสนอที่ครอบคลุมการแสดงผลให้แก่ผู้ใช้
- (2) ส่วนลงทะเบียนเป็นสมาชิกกับระบบ
- (3) ส่วนสกัดข้อมูลที่จำเป็นเพื่อให้พร้อมใช้งานในส่วนอื่นๆ
- (4) ส่วนความรู้ที่ช่วยสร้างความรู้ที่สอดคล้องกับข้อมูลในระบบ
- (5) ส่วนคัดเลือกบริการที่ทำการค้นหาบริการที่เหมาะสมกับความต้องการของผู้เรียกใช้บริการ
- (6) ส่วนร่างสัญญาเป็นส่วนที่สร้างเอกสารข้อตกลงต่างๆระหว่างผู้เรียกใช้บริการและผู้ให้บริการ
- (7) ส่วนวิเคราะห์คุณภาพบริการที่มีอยู่ในระบบ

สำหรับการพัฒนาระบบผู้ประสานงานการเจรจาต่อรองนั้นได้ใช้เทคโนโลยีจาวา ซึ่งเป็นซอฟต์แวร์แบบรหัสเปิดมาประยุกต์ใช้งานโดยเฉพาะเทคโนโลยีจาวาสำหรับเว็บ รวมถึงซอฟต์แวร์อื่นๆที่สนับสนุนเทคโนโลยีจาวาไม่ว่าจะเป็นระบบให้บริการเว็บ Glassfish หรือระบบจัดการฐานข้อมูล MySQL เป็นต้น บริการการท่องเที่ยวถูกเลือกมาเป็นกรณีตัวอย่างสำหรับการทดสอบระบบ โดยผลการทดสอบระบบผู้ประสานงานการเจรจาต่อรองสามารถทำงานได้ผล บรรลุตามวัตถุประสงค์ของงานวิจัย

7.2 ข้อจำกัดของการดำเนินงานวิจัย

ข้อจำกัดที่เกิดขึ้นของการดำเนินการวิจัยมีดังนี้

1. งานวิจัยนี้ประกอบด้วยทฤษฎี แนวคิดและเทคโนโลยีที่เกี่ยวข้องที่มีความหลากหลาย ส่งผลให้การดำเนินการวิจัยใช้เวลาล่าช้ากว่าที่กำหนด เช่น การศึกษามาตรฐานการพัฒนาซอฟต์แวร์ที่จะต้องมีการกล่าวถึงประเด็นการเจรจาต่อรอง การศึกษาเทคโนโลยีออนไลน์เพื่อนำมาใช้ในการ

- อธิบายความรู้ การออกแบบระบบตามแนวทางเชิงวัตถุ หรือการพัฒนาโปรแกรมประยุกต์ผ่านเว็บโดยใช้เทคโนโลยีจาวา เป็นต้น
2. เนื่องจากผู้วิจัยจบการศึกษาในระดับปริญญาตรีสาขาเคมี จึงขาดประสบการณ์ในงานทางด้านคอมพิวเตอร์ ทำให้จะต้องเริ่มต้นศึกษาเรียนรู้ใหม่ด้วยตนเองตั้งแต่เริ่มต้น

7.3 ข้อเสนอแนะ

- ข้อเสนอแนะสำหรับการนำงานวิจัยนี้ไปใช้หรือพัฒนาต่อในอนาคตมีดังนี้
1. กระบวนการเจรจาต่อรองสำหรับสถาปัตยกรรมเชิงบริการที่ได้เสนอมานั้นสามารถนำไปประยุกต์ใช้ในงานด้านอื่นได้โดยไม่จำกัดเฉพาะสำหรับแนวคิดสถาปัตยกรรมเชิงบริการ เช่น การเจรจาต่อรองสำหรับการให้บริการทั่วไป หรือการเจรจาต่อรองสำหรับการจัดหาซอฟต์แวร์มาใช้งาน เป็นต้น
 2. ระบบผู้ประสานงานการเจรจาต่อรองเป็นเพียงระบบที่คอยสนับสนุนกระบวนการเจรจาต่อรองให้สัมฤทธิ์ผลโดยที่ผู้มีส่วนได้ส่วนเสียติดต่อกันให้น้อยที่สุด ในความคาดหวังว่าระบบควรจะสามารถช่วยการเจรจาต่อรองได้อย่างอัตโนมัติโดยปราศจากการติดต่อสื่อสารระหว่างกันของผู้เจรจาต่อรอง ซึ่งการจะทำให้ระบบสามารถดำเนินการเช่นนี้ได้ ระบบจะต้องมีแนวคิดการดำเนินการเพิ่มความฉลาดให้แก่ระบบโดยใช้ทฤษฎีที่เกี่ยวข้องกับระบบปัญญาประดิษฐ์ และการใช้เทคโนโลยีเอเจนต์มาใช้งานเป็นตัวแทนของแต่ละฝ่ายในการเจรจาต่อรอง อย่างไรก็ตามสิ่งนี้สามารถเป็นประเด็นในการศึกษาและวิจัยในอนาคตได้
 3. ระบบผู้ประสานงานการเจรจาต่อรองที่พัฒนาขึ้นนี้ มีการพิจารณาในมุมมองความปลอดภัยในการดำเนินการใช้งานเพียงบางส่วน ซึ่งในทางปฏิบัตินั้นความปลอดภัยสำหรับระบบเป็นสิ่งที่จำเป็น โดยเฉพาะการทำงานที่เกี่ยวข้องกับเอกสารข้อตกลงต่าง ๆ ซึ่งอาจจะเป็นผลที่เกี่ยวข้องในทางกฎหมายได้ ดังนั้นการพัฒนาต่อในอนาคตเพื่อนำไปประยุกต์ใช้งานได้อย่างแท้จริงควรให้ลำดับความสำคัญในเรื่องนี้เป็นอันดับต้น
 4. แม้ว่าการออกแบบระบบผู้ประสานงานการเจรจาต่อรองมีส่วนการวิเคราะห์คุณภาพบริการประกอบอยู่ด้วย แต่การดำเนินการเรื่องนี้อยู่นอกเหนือขอบเขตการวิจัย ซึ่งสามารถดำเนินการวิจัยเพิ่มเติมได้ในอนาคต

5. การพัฒนาส่วนความรู้ที่ใช้ในระบบได้เลือกประยุกต์ใช้เทคโนโลยีออนไลน์แต่เพียงบางส่วน ยังขาดการศึกษาและพัฒนาโดยเฉพาะอย่างยิ่งการพัฒนาการค้นหาวีการเชิงความหมายโดยใช้เทคโนโลยีออนไลน์ ซึ่งการดำเนินการเรื่องนี้อยู่นอกเหนือขอบเขตการวิจัยเช่นกัน ดังนั้นสามารถนำไปวิจัยต่อยอดได้ในอนาคต
6. เนื่องจากสภาวะแวดล้อมในการพัฒนาและทดสอบระบบผู้ประสานงานการเจรจาต่อรองนั้นจัดทำขึ้นภายในห้องปฏิบัติการของกลุ่มวิจัยวิศวกรรมซอฟต์แวร์และการประยุกต์ ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ซึ่งหากมีการนำระบบดังกล่าวไปประยุกต์ใช้งานในสภาวะแวดล้อมเชิงธุรกิจจริง ควรจะต้องมีการทดสอบการทำงานก่อนนำไปใช้ รวมถึงการปรับปรุงระบบให้มีสภาพการทำงานที่เหมาะสมต่อการรองรับปัญหาต่างๆ ที่อาจจะเกิดขึ้นโดยไม่สามารถคาดเดาได้

บรรณานุกรม

- จามิกร หิรัญรัตน์. 2550. การประยุกต์เอเจนต์แบบเคลื่อนที่สำหรับการประมวลผลอิเล็กทรอนิกส์: กรณีศึกษาการประมวลผลยาง ณ ตลาดกลางยางพาราแห่งประเทศไทย. วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ สงขลา
- ทวีติย์ เสนีวงศ์ ณ อยุธยา และ ยรรยง เต็งอำนาจ. 2006. ปรากฏการณ์เอสโอเอ [ออนไลน์] เข้าถึงได้จาก: http://se.cp.eng.chula.ac.th/soa/files/20061226SOA_Phenomenon.pdf. (วันที่สืบค้น 29 กรกฎาคม 2552)
- ทวิช ว่องทวี. 2551. การแลกเปลี่ยนความรู้สำหรับชุมชนปฏิบัติผ่านตัวกลางความรู้โดยอาศัยเทคโนโลยีเว็บเซอร์วิส. วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ สงขลา
- นิตยสารภาพข่าวทักษิณ. การเจรจาต่อรอง [ออนไลน์] เข้าถึงได้จาก: <http://www.depthfirst.co.th/isab/taksinbook05/2.pdf>. (วันที่สืบค้น 6 กันยายน 2552)
- Al-Sakran, H. and Serguievskiaia, I. 2006. A Framework for Developing Experience Based e-Negotiation System. *Journal of Computer Science*. 2006, 2(2): 180-184.
- Bartolini, C., Priest, C. and Jennings, N. R. 2005. A Software Framework for Automated Negotiation. In: *Software Engineering for Multi-Agent Systems III: Research Issues and Practical Applications*. Springer-Verlag Berlin Heidelberg, Germany., pp. 213-235.
- Beam, C. and Segev, A. 1997. Automated Negotiations: A Survey of the State of the Art. *CMIT Working Paper 97-WP-1022*.
- Bennett, K., Layzell, P., Budgen, D., Brereton, P., Macaulay, L., and Munro, M. 2000. *Service-Based Software: The Future for Flexible Software*. *Proceeding of the 9th Asia-Pacific Software Engineering Conference (APSEC 2000)*. December 5-8, 2000. pp. 214-222.
- Benyoucef, M. and Rinderle, S. 2006. Modeling e-Negotiation Processes for a Service Oriented Architecture. *Group Decision and Negotiation*. 15: 449-467.
- Bichler, M., Kersten, G. and Strecker, S. 2003. Towards a Structured Design of Electronic Negotiations. *Group Decision and Negotiation*. 12: 311-335.
- Business-in-a-Box. *Business-in-a-Box - The Business Document Software*. <http://www.biztree.com/> (accessed: April 30, 2010).

- Bruijn, J. d., Fensel, D., Kerrigan, M., Keller, U., Lausen, H., and Scicluna, J. 2008. *Modeling Semantic Web Services: The Web Service Modeling Language*. Springer: Springer-Verlag Berlin Heidelberg.
- Cerami, E. 2002. *Web services essentials: Distributed applications with XML-RPC, SOAP, UDDI & WSDL*. 1st edition. O'Reilly.
- Chrissis, M.B., Konrad, M., and Shrum, S. 2003. *CMMI® Guidelines for Process Integration and Product Improvement*. Addison-Wesley Professional.
- Chu, S.C. 2005. *From Component-based to Service Oriented Software Architecture for Healthcare*. Proceedings of 7th International Workshop on Enterprise networking and Computing in Healthcare Industry 2005 (HEALTHCOM 2005). Busan, Korea, June 23-25, 2005. pp. 96-100.
- CORBA. Common Object Request Broker Architecture (CORBA/IIOP). http://www.omg.org/technology/documents/corba_spec_catalog.htm (accessed: February 24, 2010).
- CMMI Product Team. 2002. *CMMI for Development, Version 1.1*. Technical Report. Software Engineering Institution (SEI). (accessed: February 24, 2010).
- DCOM. Distributed Component Object Model (DCOM). [http://msdn.microsoft.com/th-th/library/cc201989\(en-us,PROT.10\).aspx](http://msdn.microsoft.com/th-th/library/cc201989(en-us,PROT.10).aspx). (accessed: February 24, 2010).
- Dean, M. and Schreiber, G. 2004. *OWL Web Ontology Language Reference W3C Recommendation*. <http://www.w3.org/TR/owl-ref/> (accessed December 07, 2009).
- Di Nitto, E., Di Penta, M., Gambi, A., Ripa, G., and Villani, M. L. 2007. *Negotiation of Service Level Agreements: An Architecture and a Search-Based Approach*. The 5th International Conference on Service-Oriented Computing 2007 (ICSOC 2007). Vienna, Austria, September 17-20, 2007. pp. 295-306.
- Elfatraty, A. and Layzell, P. 2002. *Software As A Service : A Negotiation Perspective*. Proceedings of the 26th Annual International Computer Software and Applications Conference (COMPSAC'02). Oxford, August 26-29, 2002. pp. 501-506.
- Elfatraty, A. and Layzell, P. 2004. *Negotiating in Service-Oriented Environments*. Communications of the ACM. 47(8): 103-108.
- Erl, T. 2005. *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall PTR: Indiana.

- Gamma, E., Helm, R., Johnson, R., and Vlissides, J.M. 1994. Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Professional: United State of America.
- Greer, D. 2008. The Art of Separation of Concerns. <http://www.ctrl-shift-b.com/2008/01/art-of-separation-of-concerns.html>. (accessed July 08, 2009).
- Harper, D. negotiation. Dictionary.com. Online Etymology Dictionary. <http://dictionary.reference.com/browse/negotiation> (accessed: September 06, 2009).
- Hurwitz Report. 2000. Negotiated Trade: the Next Frontier for B2B e-commerce. The Hurwitz Group. Technical Report.
- IEEE (Institute of Electrical and Electronics Engineers). 1998. IEEE Recommended Practice for Software Acquisition. IEEE Standard 1062. 1998 Edition. IEEE Computer Society.
- IBM. Standards and Web services. <http://www.ibm.com/developerworks/webservices/standards/> (accessed: February 24, 2010).
- Java Passion. SOAP 1.2 (Simple Object Access Protocol). <http://www.javapassion.com/webservices/SOAPBasics.pdf> (accessed: February 24, 2010).
- JDBC (Java Database Connectivity). JDBC Overview. <http://java.sun.com/j2se/1.4.2/docs/guide/jdbc/> (accessed: January 16, 2010).
- jUDDI. Apache jUDDI. <http://ws.apache.org/juddi/> (accessed: January 16, 2010)
- Lo, G., and Kersten, G.E. 1999. Negotiation in Electronic Commerce Integrating Negotiation Support and Software Agent Technologies. 5th Annual Atlantic Canadian Operational Research Society Conference. Halifax, Nova Scotia.
-
- Maximilien, E.M. and Singh, N.P. 2004. A Framework and Ontology for Dynamic Web Services Selection. IEEE Internet Computing. 8(5): 84-93.
-
- McIntosh, R.L. 2004. Open-Source Tools for Distributed Device Control within A Service-Oriented Architecture. Journal of the Association for Laboratory Automation. 2004, 9(6): 404-410.
- Metro. Discover Metro. <https://metro.dev.java.net/discover/> (accessed: January 16, 2010).
- Murugesan, S. 2000. Negotiation by Software Agents in Electronic Marketplace. Proceedings of the TENCON 2000, Volume 2. September 24-27, 2000. pp. 286-290.

- Nock, C. 2003. *Data Access Patterns: Database Interactions in Object-Oriented Applications*. Addison Wesley: United State of America.
- OASIS (Organization for the Advancement of Structured Information Standards). 2006. Reference Model for Service Oriented Architecture 1.0. OASIS Standard.
- OASIS (Organization for the Advancement of Structured Information Standards). UDDI Specifications. <http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm> (accessed September 6, 2009).
- Pruit, D.G. 1982. *Negotiation Behavior*. Academic Press: New York.
- Random House, Inc. negotiation. Dictionary.com. Dictionary.com Unabridged (v 1.1). <http://dictionary.reference.com/browse/negotiation> (accessed: September 06, 2009).
- Reenskaug, T. 2003. The Model-View-Controller (MVC) Its Past and Present. http://heim.ifi.uio.no/~trygver/2003/javazone-jaoo/MVC_pattern.pdf (accessed: January 16, 2010).
- RMI. Remote Method Invocation (RMI). <http://java.sun.com/javase/technologies/core/basic/rmi/index.jsp> (accessed: February 24, 2010).
- Rosa, J.L., Hormazábal, N., Aciar, S., Lopardo, G., Trias, A. and Montaner, M. 2009. A Negotiation Style Recommender Based on Computational Ecology in Open Negotiation Environments. *IEEE Transactions on Industrial Electronics* (December): 1-13.
- Sandholm, T. 1999. An Algorithm for Optimal Winner Determination in Combinatorial Auctions. *Proceedings of the International Joint Conference on Artificial Intelligence*. Stockholm, Sweden, July 31 - August 6, 1999. pp. 542-547.
-
- Santini, S. and Gibson, J.D. 2001. *Exploratory Image Databases: Content-Based Retrieval*. Academic Press: United State of America.
-
- Singh, R. 1996. *International Standard ISO/IEC 12207 Software Life Cycle Processes*. Federal Aviation Administration. Washington DC. USA.
- SOAP (Simple Object Access Protocol). 2007. SOAP Version 1.2. <http://www.w3.org/TR/soap12> (accessed: February 24, 2010).
- Studer, R., Benjamins, V.R., and Fensel, D. 1998. Knowledge engineering: Principles and methods. *IEEE Transactions on Data and Knowledge Engineering*. 25(1-2): 161-197.
-

- TCPMon. 2006. Apache TCPMon. <http://ws.apache.org/commons/tcpmon/> (accessed: February 18, 2010).
- Longman. 2009. The Longman Dictionary of Contemporary English Online - Result for service - Longman English Dictionary Online. http://www.ldoceonline.com/dictionary/service_1. (accessed: July 8, 2009).
- UDDI (Universal Description, Discovery and Integration). 2002. UDDI Version 2.04 API Specification. <http://uddi.org/pubs/ProgrammersAPI-V2.04-Published-20020719> (accessed: January 16, 2010)
- W3C (The World Wide Web Consortium). 2004. Web Services Architecture. <http://www.w3.org/TR/ws-arch/> (accessed: February 24, 2010).
- WSDL (Web Services Description Language). 2001. Web Services Description Language (WSDL) 1.1. <http://www.w3.org/TR/wsdl> (accessed: February 24, 2010).
- XML (Extensible Markup Language). 2008. Extensible Markup Language (XML) 1.0 (Fifth Edition). <http://www.w3.org/TR/REC-xml/> (accessed: April 6, 2010).
-
-

ภาคผนวก

ภาคผนวก ก

การติดตั้งระบบผู้ประสานงานการเจรจาต่อรอง

ระบบผู้ประสานงานการเจรจาต่อรองได้รับการพัฒนาด้วยภาษาจาวาและดำเนินการในรูปแบบโปรแกรมประยุกต์ผ่านเว็บ ซึ่งใช้ Java Servlet Technology และ Java Server Page (JSP) ในการพัฒนา โดยมีระบบให้บริการเว็บเป็น Glassfish Application Server และระบบจัดการฐานข้อมูลเป็น MySQL ดังนั้นสามารถแสดงรายละเอียดการติดตั้งระบบผู้ประสานงานการเจรจาต่อรองโดยคร่าวได้ดังนี้

ก.1 สิ่งที่ต้องดำเนินการก่อน

ในการติดตั้งระบบจะกระทำบนระบบปฏิบัติการวินโดวส์ XP ซึ่งจะต้องทำการลงเครื่องมือเหล่านี้ให้เสร็จเรียบร้อยก่อนการดำเนินการในขั้นถัดไป ได้แก่

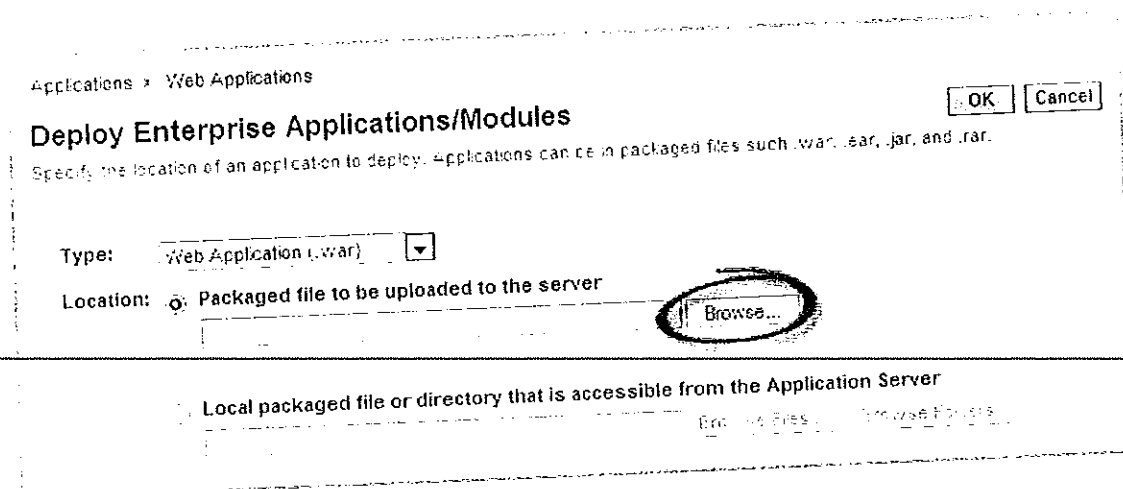
- (1) Java Software Development Kit (JDK) ซึ่งใช้ได้ตั้งแต่รุ่น 6.0 ขึ้นไป โดยแนะนำให้กำหนดไดเรกทอรีที่ติดตั้งเป็น C:/Java
- (2) Glassfish Application Server ซึ่งเลือกใช้รุ่น 6.5 เป็นหลักในงานวิจัยนี้ โดยแนะนำให้กำหนดไดเรกทอรีที่ติดตั้งเป็น C:/Glassfish
- (3) MySQL ซึ่งใช้ได้ตั้งแต่รุ่น 5.0 ขึ้นไป โดยแนะนำให้กำหนดไดเรกทอรีที่ติดตั้งเป็น C:/MySQL

ก.2 การติดตั้งระบบ

ขั้นตอนการติดตั้งระบบผู้ประสานงานการเจรจาต่อรองแบบคร่าวๆมีดังนี้

1. ทำการคลายไฟล์ Negotiation-Coordinator.zip ที่ปรากฏอยู่ในแผ่นซีดีประกอบเอกสาร ลงในเครื่องที่ต้องการติดตั้ง

2. ทำการคัดลอกไฟล์ Negotiation-Coordinator.war ซึ่งอยู่ภายในไดเรกทอรี WEB ที่ได้จากขั้นตอนที่หนึ่ง ไปไว้ในที่ที่ต้องการ
3. ดำเนินการเริ่มต้นระบบ Glassfish Application Server โดยเข้าไปที่ path Glassfish/bin แล้วกระทำคำสั่ง "asadmin.bat start-domain domain1" จากนั้นรอจนกระทั่งกระบวนการเสร็จสิ้น
4. เปิดเบราว์เซอร์และเข้าไปที่ http://localhost:4848 ซึ่ง 4848 เป็นพอร์ตสำหรับการเข้าถึงส่วนแสดงผลของการจัดการระบบ โดยสามารถกำหนดหรือเปลี่ยนแปลงได้ในขั้นตอนการติดตั้ง จากนั้นกรอกข้อมูลรหัสผู้ใช้งานและรหัสผ่านตามที่ได้กำหนดไว้ในตอนติดตั้ง เช่นเดียวกัน
5. เมื่อเข้าสู่หน้าการจัดการระบบเรียบร้อยแล้ว เลือก Application/Web Applications จากเมนูทางด้านขวา แล้วคลิกเลือกปุ่ม Deploy ในส่วนเนื้อหาทางด้านซ้าย
6. จากนั้นเลือกคลิก Browse ในส่วน Location: เลือก Packaged file to be uploaded to the server และเลือกไฟล์ Negotiation-Coordinator.war ที่ได้คัดเลือกไว้ในขั้นตอนเมื่อเลือกไฟล์เรียบร้อยแล้วคลิกเลือก OK



ภาพประกอบที่ ก.1 การแสดงผลคลิกเลือก Browse ในส่วน Deploy ของการจัดการระบบ

7. ทำการคัดลอกไฟล์ Negotiation-Coordinator.sql ซึ่งอยู่ภายในไดเรกทอรี SQL ที่ได้จากขั้นตอนที่หนึ่ง ไปไว้ในที่ที่ต้องการ
8. ไปที่ path ที่ติดตั้ง MySQL แล้วกระทำคำสั่ง "mysql -h localhost -u [username] -p [password] < Negotiation-Coordinator.sql" โดยกำหนดรหัสผู้ใช้งานและรหัสผ่าน

ที่ได้จากการกำหนดในการติดตั้ง MySQL แทนในส่วน [username] และ [password] ตามลำดับ ส่วนไฟล์ Negotiation-Coordinator.sql กำหนด path ที่อยู่กำกับให้ชัดเจน เช่น C:/MySQL/Negotiation-Coordinator.sql เป็นต้น

9. เมื่อดำเนินการตามขั้นตอนข้างต้นเรียบร้อยแล้ว สามารถเข้าใช้งานระบบผู้ประสานงานการเจรจาต่อรองได้ผ่านเว็บเบราว์เซอร์ โดยไปที่ <http://localhost:8080/Thesis/index.jsp> (8080 เป็นหมายเลขพอร์ตที่กำหนดการเข้าถึงเว็บของ Glassfish Application Server ซึ่งสามารถกำหนดหรือเปลี่ยนแปลงได้ในขั้นตอนการติดตั้ง)

ภาคผนวก ข

เอกสาร XML Schema ที่ใช้แทนความรู้

เอกสาร XML Schema ที่ใช้สำหรับอธิบายความรู้มีทั้งหมด 3 รูปแบบคือ เอกสาร XML Schema ที่สอดคล้องกับออนโทโลยีของผู้ดำเนินการ (Actor) ที่เกี่ยวข้องกับระบบ, เอกสาร XML Schema ที่สอดคล้องกับออนโทโลยีบริการ และเอกสาร XML Schema ที่สอดคล้องกับออนโทโลยีคุณภาพของบริการ โดยสามารถแสดงรายละเอียดทั้งหมดของเอกสารดังกล่าวได้ดังภาพประกอบที่ ข.1 – ข.3

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xi="http://www.w3.org/2001/XInclude" elementFormDefault="qualified">
  <xs:import namespace="http://www.w3.org/2001/XInclude"
    schemaLocation="XInclude.xsd"/>
  <xs:import namespace="http://www.w3.org/2001/XMLSchema-instance"
    schemaLocation="XMLSchema-instance.xsd"/>
  <xs:element name="actor">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="identified"/>
        <xs:element ref="contact"/>
        <xs:element ref="provider"/>
      </xs:sequence>
      <xs:attribute name="id" type="xs:integer" use="required"/>
      <xs:attribute name="type" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="identified">
    <xs:complexType>
      <xs:sequence>
```

ภาพประกอบที่ ข.1 เอกสาร XML Schema ของผู้ดำเนินการที่เกี่ยวข้องกับระบบ


```

        <xs:element ref="idNumber"/>
        <xs:element ref="name"/>
        <xs:element ref="lastname"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="idNumber" type="xs:integer"/>
<xs:element name="name" type="xs:string"/>
<xs:element name="lastname" type="xs:string"/>
<xs:element name="contact">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="address" maxOccurs="unbounded"/>
            <xs:element ref="email" maxOccurs="unbounded"/>
            <xs:element ref="telephones"/>
            <xs:element ref="website" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="address">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="number"/>
            <xs:element ref="street"/>
            <xs:element ref="city"/>
            <xs:element ref="province"/>
            <xs:element ref="country"/>
            <xs:element ref="postalCode"/>
        </xs:sequence>
        <xs:attribute name="id" type="xs:integer" use="required"/>
    </xs:complexType>
</xs:element>
<xs:element name="number" type="xs:string"/>
<xs:element name="street" type="xs:string"/>
<xs:element name="city" type="xs:string"/>
<xs:element name="province" type="xs:string"/>

```

ภาพประกอบที่ ข.1 (ต่อ)

```

<xs:element name="country" type="xs:string"/>
<xs:element name="postalCode" type="xs:string"/>
<xs:element name="email" type="xs:string"/>
<xs:element name="telephones">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="telNumber" ↗
        maxOccurs="unbounded"/>
      <xs:element ref="mobileNumber" ↗
        maxOccurs="unbounded"/>
      <xs:element ref="faxNumber" ↗
        maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="telNumber" type="xs:string"/>
<xs:element name="mobileNumber" type="xs:string"/>
<xs:element name="faxNumber" type="xs:string"/>
<xs:element name="website" type="xs:anyURI"/>
<xs:element name="provider">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="businessCategory"/>
      <xs:element ref="description"/>
      <xs:element ref="provide"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="categoryName" type="xs:string"/>
<xs:element name="taxonomy" type="xs:string"/>
<xs:element name="value" type="xs:string"/>
<xs:element name="description">
  <xs:complexType>

```

ภาพประกอบที่ ข.1 (ต่อ)

```

        <xs:sequence>
            <xs:element ref="overview"/>
            <xs:element ref="successStory"/>
            <xs:element ref="reseller"/>
            <xs:element ref="referenceCustomer"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="overview" type="xs:string"/>
<xs:element name="successStory" type="xs:string"/>
<xs:element name="reseller" type="xs:string"/>
<xs:element name="referenceCustomer" type="xs:string"/>
<xs:element name="provide">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="xi:include"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="include">
    <xs:complexType mixed="true">
        <xs:attribute name="parse" type="xs:string" use="required"/>
    </xs:complexType>
</xs:element>
</xs:schema>

```

ภาพประกอบที่ ข.1 (ต่อ)

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xi="http://www.w3.org/2001/XInclude" elementFormDefault="qualified">
  <xs:import namespace="http://www.w3.org/2001/XInclude"
    schemaLocation="XInclude.xsd"/>

  <xs:element name="service">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="profile"/>
        <xs:element ref="process"/>
        <xs:element ref="price"/>
        <xs:element ref="contract"/>
        <xs:element ref="supportedBy"
          maxOccurs="unbounded"/>
        <xs:element ref="provideBy"/>
        <xs:element ref="QoS"/>
      </xs:sequence>
      <xs:attribute name="id" use="required"/>
    </xs:complexType>
  </xs:element>

  <xs:element name="profile">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="name"/>
        <xs:element ref="version"/>
        <xs:element ref="category"/>
        <xs:element ref="incentive"/>
        <xs:element ref="description"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="version" type="xs:decimal"/>
  <xs:element name="category">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="categoryName"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

```

        <xs:element ref="taxonomy" />
        <xs:element ref="value" />
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="categoryName" type="xs:string"/>
<xs:element name="taxonomy" type="xs:string"/>
<xs:element name="value" type="xs:string"/>
<xs:element name="incentive">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="review" maxOccurs="unbounded"/>
            <xs:element ref="sample" maxOccurs="unbounded"/>
            <xs:element ref="demo" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="review" type="xs:string"/>
<xs:element name="sample" type="StringandAnyURI"/>
<xs:element name="demo" type="StringandAnyURI"/>
<xs:element name="description">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="overview" maxOccurs="unbounded"/>
            <xs:element ref="specification"
                maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="overview" type="xs:string"/>
<xs:element name="specification" type="StringandAnyURI"/>
<xs:element name="process">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="operation" maxOccurs="unbounded"/>
            <xs:element ref="functional"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```

        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="operation">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="input"/>
            <xs:element ref="output"/>
        </xs:sequence>
        <xs:attribute name="name" use="required"/>
    </xs:complexType>
</xs:element>
<xs:element name="input">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="name"/>
            <xs:element ref="type"/>
            <xs:element ref="describing"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="output">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="name"/>
            <xs:element ref="type"/>
            <xs:element ref="describing"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="functional">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="feature" maxOccurs="unbounded"/>
            <xs:element ref="interface"/>
            <xs:element ref="compatible"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```

        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="feature" type="xs:string"/>
<xs:element name="interface" type="xs:anyURI"/>
<xs:element name="compatible">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="platform" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="platform">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="describing"/>
        </xs:sequence>
        <xs:attribute name="name" use="required"/>
    </xs:complexType>
</xs:element>
<xs:element name="price" type="xs:decimal"/>
<xs:element name="contract">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="xi:include" maxOccurs="unbounded"/>
            <xs:element ref="indenture" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="indenture" type="xs:anyURI"/>
<xs:element name="supportedBy">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="xi:include" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>

```

```

</xs:element>
<xs:element name="provideBy">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="xi:include"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="QoS">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="xi:include" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="name" type="xs:string"/>
<xs:element name="describing" type="xs:string"/>
<xs:element name="type">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="xi:include"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="include">
  <xs:complexType mixed="true">
    <xs:attribute name="parse" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>
<xs:simpleType name="StringandAnyURI">
  <xs:union memberTypes="xs:string xs:anyURI"/>
</xs:simpleType>
</xs:schema>

```

ภาพประกอบที่ ข.2 (ต่อ)


```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xi="http://www.w3.org/2001/XInclude" elementFormDefault="qualified"
  <xs:import namespace="http://www.w3.org/2001/XInclude"
    schemaLocation="XInclude.xsd"/>

  <xs:element name="quality">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="type"/>
        <xs:element ref="value"/>
        <xs:element ref="measurement"/>
        <xs:element ref="measureBy"/>
        <xs:element ref="aggregatedQualities"/>
        <xs:element ref="relatedQualities"/>
      </xs:sequence>
      <xs:attribute name="name" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="type" type="xs:string"/>
  <xs:element name="value" type="xs:integer"/>
  <xs:element name="measurement" type="xs:string"/>
  <xs:element name="measureBy">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="xi:include" minOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="aggregatedQualities">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="xi:include"
          maxOccurs="unbounded" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

ภาพประกอบที่ ข.3 เอกสาร XML Schema ของคุณภาพของบริการ

```
<xs:element name="relatedQualities">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="xi:include" ↗
        minOccurs="0" maxOccurs="unbounded" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="include">
  <xs:complexType mixed="true">
    <xs:attribute name="parse" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>
</xs:schema>
```

ภาพประกอบที่ ข.3 (ต่อ)

ภาคผนวก ค

ตัวอย่างเอกสารสัญญา

เอกสารสัญญาต่างๆที่พัฒนาขึ้นมาใช้ในระบบประกอบด้วย 4 รูปแบบคือ เอกสารข้อกำหนดความต้องการบริการ (Requirement specification) เอกสารข้อเสนอบริการ (Service proposal) เอกสารข้อเสนอบริการที่ไม่มีการระบุตัวผู้ให้บริการ (Anonymous proposal) และเอกสารข้อตกลงบริการ (Service agreement) โดยมีประยุกต์มาจากเอกสารข้อตกลงต่างๆทางธุรกิจจริง (Business-in-a-Box, 2010) ซึ่งสามารถแสดงได้ดังภาพประกอบที่ ค.1 – ค.4 และตัวอย่างข้อมูลที่น่ามาใช้คือข้อมูลของกรณีตัวอย่างรายแรก (Company One)

Negotiation Coordinator	Service Requirement Specification
Service Requirements Specification	
Number : 001	
Date : 2010/02/24	
Time : 09:00	
Prepared By Chaiwat	
001SVCREq.html	1/2

ภาพประกอบที่ ค.1 ตัวอย่างเอกสารข้อกำหนดความต้องการบริการ (Requirement specification)

Negotiation Coordinator	Service Requirement Specification																														
<p>Purpose: The purpose of the Service Requirements Specification is to document requirements for the service. The Service Requirements Specification should be used in conjunction with the business requirements documented.</p>																															
<h3>Requirements</h3>																															
<p>The Requirements section provides information on the requirements that the service must provide.</p>																															
<table border="1"> <tr> <td colspan="2">Requirement Description</td> </tr> <tr> <td colspan="2">To search service that has attributes corresponding user defined attributes.</td> </tr> <tr> <td colspan="2">Other Comments</td> </tr> <tr> <td colspan="2"><i>Pro Search, Non-Service, etc. to provide</i></td> </tr> <tr> <td colspan="2">Requirement List</td> </tr> <tr> <td colspan="2">I desire a service which named like</td> </tr> <tr> <td>Service Name</td> <td>HotelService</td> </tr> <tr> <td colspan="2">_WORDLINK1_ has feature or overview like</td> </tr> <tr> <td>Description</td> <td>Find available room and booking room</td> </tr> <tr> <td colspan="2">_WORDLINK2_ has category name like</td> </tr> <tr> <td>Category Name</td> <td>Travel agencies and service</td> </tr> <tr> <td colspan="2">_WORDLINK3_ has operation name like</td> </tr> <tr> <td>Operation Name</td> <td>getAvailableRoom</td> </tr> <tr> <td colspan="2">_WORDLINK4_ has price around</td> </tr> <tr> <td>Price</td> <td>5</td> </tr> </table>		Requirement Description		To search service that has attributes corresponding user defined attributes.		Other Comments		<i>Pro Search, Non-Service, etc. to provide</i>		Requirement List		I desire a service which named like		Service Name	HotelService	_WORDLINK1_ has feature or overview like		Description	Find available room and booking room	_WORDLINK2_ has category name like		Category Name	Travel agencies and service	_WORDLINK3_ has operation name like		Operation Name	getAvailableRoom	_WORDLINK4_ has price around		Price	5
Requirement Description																															
To search service that has attributes corresponding user defined attributes.																															
Other Comments																															
<i>Pro Search, Non-Service, etc. to provide</i>																															
Requirement List																															
I desire a service which named like																															
Service Name	HotelService																														
WORDLINK1 has feature or overview like																															
Description	Find available room and booking room																														
WORDLINK2 has category name like																															
Category Name	Travel agencies and service																														
WORDLINK3 has operation name like																															
Operation Name	getAvailableRoom																														
WORDLINK4 has price around																															
Price	5																														
001SVCREq.html	2/2																														

Negotiation Coordinator	Service Proposal
<p data-bbox="523 712 895 763" style="text-align: center;">Service Proposal</p> <p data-bbox="539 824 887 969" style="text-align: center;">Number : 001 Date : 2010/02/24 Time : 09:00</p> <p data-bbox="596 1070 839 1189" style="text-align: center;">Prepared for Chaiwat</p> <p data-bbox="584 1339 863 1458" style="text-align: center;">Prepared By Company One</p>	
<p data-bbox="169 1827 440 1861">001SVCPproposal.html</p>	<p data-bbox="1246 1805 1286 1839">1/5</p>

ภาพประกอบที่ ค.2 ตัวอย่างเอกสารข้อเสนอบริการ (Service proposal)

Negotiation Coordinator	Service Proposal																
<p>1. COMPANY SUMMARY</p> <p>Company One is pleased to present Chaiwat with this proposal for your requirement. Having duly examined your requirement, we are confident that our proposed service effectively addresses your needs. With our unique ability and our successful track record makes us an enviable partner of yours. We look forward to forming a mutually rewarding relationship with Chaiwat</p>																	
<p>2. COMPANY BACKGROUND</p>																	
<p>2.1 Mission Statement</p> <p>The company's mission to meet the customer needs.</p>																	
<p>2.2 Location</p> <table border="1"> <tr> <td>Address</td> <td>15 Karnjanavanit</td> </tr> <tr> <td>City</td> <td>Hatyai</td> </tr> <tr> <td>Country</td> <td>Songkhla</td> </tr> <tr> <td>Postal Code</td> <td>90110</td> </tr> <tr> <td>Tel.</td> <td>074-288-572</td> </tr> <tr> <td>Mobile</td> <td>081-123-4567</td> </tr> <tr> <td>Fax</td> <td>074-288-572</td> </tr> <tr> <td>Email</td> <td>Request@company.do</td> </tr> </table>		Address	15 Karnjanavanit	City	Hatyai	Country	Songkhla	Postal Code	90110	Tel.	074-288-572	Mobile	081-123-4567	Fax	074-288-572	Email	Request@company.do
Address	15 Karnjanavanit																
City	Hatyai																
Country	Songkhla																
Postal Code	90110																
Tel.	074-288-572																
Mobile	081-123-4567																
Fax	074-288-572																
Email	Request@company.do																
<p>Customer Service / Supporter [optional]</p> <table border="1"> <tr> <td>Tel.</td> <td>074-288-572</td> </tr> <tr> <td>Mobile</td> <td>081-123-4567</td> </tr> <tr> <td>Fax</td> <td>074-288-572</td> </tr> <tr> <td>Email</td> <td>Support@company.do</td> </tr> <tr> <td>Web Site</td> <td>http://company_one/support/index.php</td> </tr> </table>		Tel.	074-288-572	Mobile	081-123-4567	Fax	074-288-572	Email	Support@company.do	Web Site	http://company_one/support/index.php						
Tel.	074-288-572																
Mobile	081-123-4567																
Fax	074-288-572																
Email	Support@company.do																
Web Site	http://company_one/support/index.php																
<p>2.3 Company Description</p> <p>The travel agency company in the south of Thailand.</p>																	
001SVCPproposal.html	2/5																

Negotiation Coordinator	Service Proposal
Overview	
Provide online travel reservation	
Category	
Travel agency	
Reseller	
Company One Derived Business (CODeB)	
Reference Customer	
Company Three Travel	
Success Story	
-	
3. PROPOSED SERVICE	
1.1 General Description	
Name of Service	HotelService
Service Category	Travel agencies and service
Brief Description of Service	
-	
Dedicated Web / Portals	
http://company-one/ws/HotelService	
1.2 Scope	
Service offered to	
Directly to end users	
Service offered	
Searching available room at any price and type, and booking it	
Service designed and offered for	
Travel agencies and service	
001SVCPproposal.html	3/5

Negotiation Coordinator	Service Proposal
1.3 Development of Service	
Version	1.0
Technology used	
Java	
Compatibility	
Platform Name	Metro
Platform Description	Metro framework with Glassfish application server
1.4 Marketing Aspects	
Price	5 baht
Pricing Scheme	
Transaction report at the end of month	
Review	
http://company_one/hReview.html	
Sample	
http://company_one/hSample.html	
Demonstration	
http://company_one/hDemo.html	
Specification	
http://company_one/hSpec.html	
Support	
Company One Co. Ltd,	
1.5 Supplement [optional]	
-	
001SVCPproposal.html	4/5

Negotiation Coordinator	Service Proposal
<p>4. CONCLUSION</p> <p>We sincerely hope that Chaiwat will consider us as a long-term partner and allow us to enter into a mutually beneficial relationship. We are available to answer any question you may have and look forward to discussing this opportunity further.</p> <p>After reviewing this document, the following steps should be performed in order to come to a final agreement</p> <ul style="list-style-type: none">- Submission of questions / suggestions- Approval by Chaiwat- Negotiation of fees, terms, clauses and conditions <p>Thank you for your interest,</p> <p>Sincerely,</p> <p>Company One</p> <p><i>001SVCProposal.html</i> 5/5</p>	

ภาพประกอบที่ ค.2 (ต่อ)

Negotiation Coordinator	Service Proposal
<p>Service Proposal</p> <p>Number : 001 Date : 2010/02/24 Time : 09:00</p> <p>Prepared for Chaiwat</p> <p>Prepared By Company One</p>	
001ANONYMProposal.html	1/3

Negotiation Coordinator	Service Proposal
1. PROPOSED SERVICE	
1.1 General Description	
Name of Service	HotelService
Service Category	Travel agencies and service
Brief Description of Service	
-	
Dedicated Web / Portals	
http://company-one/ws/HotelService	
1.2 Scope	
Service offered to	
Directly to end users	
Service offered	
Searching available room at any price and type, and booking it	
Service designed and offered for	
Travel agencies and service	
1.3 Development of Service	
Version	1.0
Technology used	
Java	
Compatibility	
Platform Name	Metro
Platform Description	Metro framework with Glassfish application server
001ANONYMProposal.html	
2/3	

Negotiation Coordinator	Service Proposal
1.4 Marketing Aspects	
Price	5 baht
Pricing Scheme	
Transaction report at the end of month	
Review	
http://company_one/hReview.html	
Sample	
http://company_one/hSample.html	
Demonstration	
http://company_one/hDemo.html	
Specification	
http://company_one/hSpec.html	
Support	
Company One Co. Ltd,	
1.5 Supplement [optional]	
-	
2. CONCLUSION	
<p>We sincerely hope that Chaiwat will consider us as a long-term partner and allow us to enter into a mutually beneficial relationship. We are available to answer any question you may have and look forward to discussing this opportunity further.</p> <p>After reviewing this document, the following steps should be performed in order to come to a final agreement</p> <ul style="list-style-type: none"> - Submission of questions / suggestions - Approval by Chaiwat - Negotiation of fees, terms, clauses and conditions <p>Thank you for your interest,</p> <p>Sincerely,</p> <p>Company One</p>	
001ANONYMProposal.html	3/3

SERVICES AGREEMENT

This Service Agreement (the "Agreement") is made and effective 2010/02/24,

BETWEEN: **Company One** (the "Service Provider"), a corporation organized and existing under the laws of Thailand, with its address located at: 15 Karnjanavanit, Hatyai, Songkhla 90112 Thailand

AND: **Chaiwat** (the "Service Requestor"), a corporation or individual organized and existing under the laws of Thailand, with its address located at: 75 Niphat-u-thit 3 rd., Hatyai, Songkhla 90100 Thailand

WHEREAS The Service Provider is in the business of development, supply and operation of products and services relating to provide online travel reservation; and

WHEREAS this Agreement contains the Service Provider's terms of engagement;

NOW, THEREFORE, in consideration of the mutual covenants and agreements herein contained, the parties hereto, intending, to be legally bound, agree as follows:

1. SERVICES PROVIDED

Service Provider is prepared to provide the following services to Service Requestor.

2. SERVICE LEVEL AGREEMENT

This description is related to supplement; http://company_one/doc/001SLA.pdf

3. CALCULATION OF FEES AND OTHER CHARGES

5 baht per transaction.

4. BILLING ARRANGEMENTS

At the end of month, transaction report will be sent.

5. NON PAYMENT

-

6. REQUEST FOR FEE DETAILS & REMAINING WORK ESTIMATE

Service Provider will provide upon request the itemized details of the work, fees and costs incurred so far and estimate of the work, fees and costs payable to complete the matter.

7. TERMINATION BY Chaiwat

Service Requestor may terminate this agreement by giving Service Provider a written notice at any time and will be liable for all work fees incurred up to that time. If Service Requestor does not provide such a notice, it will be obliged to pay all fees for work done and for other charges incurred.

8. TERMINATION BY Company One

Service Provider may terminate this agreement and stop acting for Service Requestor if:

- i. Service Requestor does not comply with this Agreement
- ii. Service Provider forms the opinion, on reasonable grounds, that mutual confidence and trust do not exist between both parties; or
- iii. Service Provider believes on reasonable grounds that, by continuing to act for Service Requestor, it may breach the professional conduct rules which are binding upon professionals in the industry.

IN WITNESS WHEREOF, each party to this agreement has caused it to be executed on date indicated above

Company One

Chaiwat

Authorized Signature

Authorized Signature

Print Name and Title

Print Name and Title

ผลงานตีพิมพ์เผยแพร่จากวิทยานิพนธ์

กรอบแนวคิดการเจรจาต่อรองสำหรับการพัฒนาซอฟต์แวร์เชิงบริการโดยการเพิ่มสมรรถนะของ
แบบจำลองวุฒิภาวะความสามารถแบบบูรณาการ
A NEGOTIATION FRAMEWORK FOR SERVICE-ORIENTED ARCHITECTURE
BY ENHANCING CMMI

ชัยวัฒน์ ไพบูลย์พิสุทธิพงศ์ และ อำนวย เปาะทอง

กลุ่มวิจัยทางวิศวกรรมซอฟต์แวร์และการประยุกต์ ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
มหาวิทยาลัยสงขลานครินทร์ อำเภอหาดใหญ่ จังหวัดสงขลา 90112 โทรศัพท์ +66 7428-8578
E-mail: chaiwat_@hotmail.com, amnart.p@psu.ac.th

บทคัดย่อ

ตั้งแต่แนวคิดสถาปัตยกรรมเชิงบริการได้รับการแนะนำและเริ่มได้รับความสนใจจากนักพัฒนาซอฟต์แวร์ ส่งผลให้ผลิตภัณฑ์ซอฟต์แวร์ถูกสร้างอยู่ในรูปบริการ และการเจรจาต่อรองเป็นประเด็นสำคัญสำหรับการพัฒนาซอฟต์แวร์เชิงบริการให้สอดคล้องกับความต้องการของทั้งผู้ให้บริการและผู้ใช้บริการ อย่างไรก็ตามมาตรฐานที่สนับสนุนการพัฒนาผลิตภัณฑ์ซอฟต์แวร์ในปัจจุบัน โดยเฉพาะแบบจำลองวุฒิภาวะความสามารถแบบบูรณาการยังขาดความชัดเจนและเพียงพอสำหรับการพัฒนาซอฟต์แวร์เชิงบริการในประเด็นเกี่ยวกับการเจรจาต่อรอง งานวิจัยนี้ได้ทำการศึกษาและเสนอกรอบแนวคิดการเจรจาต่อรองสำหรับการพัฒนาซอฟต์แวร์เชิงบริการ โดยการเพิ่มสมรรถนะของแบบจำลองวุฒิภาวะความสามารถแบบบูรณาการ และสามารถนำไปพัฒนาระบบการเจรจาต่อรองแบบอัตโนมัติทางเทคโนโลยีอินเทอร์เน็ตได้

คำสำคัญ— สถาปัตยกรรมเชิงบริการ, แบบจำลองวุฒิภาวะความสามารถแบบบูรณาการ, การเจรจาต่อรอง

ABSTRACT

Since service-oriented architecture was introduced and gained attentions from software developers, this new notion has affected

software products to be built in the form of services. Negotiation becomes a key issue for service-oriented software development in order to satisfy the needs between consumers and providers. However, the existing standards for software development especially capability maturity model integration (CMMI) lacks clear and enough details for service-oriented development in the term of service negotiation. Therefore, the research presented in this paper aims to study and propose a negotiation framework for service-oriented architecture by enhancing CMMI. The proposed framework can lead to the development of an automated negotiation system via the Internet.

Index Terms— Service-Oriented Architecture, Capability Maturity Model® Integration, Negotiation

1. บทนำ

แนวทางการพัฒนาซอฟต์แวร์ในอดีตโดยทั่วไป ผู้พัฒนาซอฟต์แวร์จะต้องมีข้อกำหนดคุณลักษณะความต้องการ (Requirements specification) ซึ่งมักจะมาจากผู้ใช้และนำไปออกแบบระบบ (System design) แล้วเขียนโปรแกรมขึ้นใช้งาน (Implementation) หลังจากนั้นมีการทดสอบระบบ (System testing) ในที่สุดมีการส่งมอบผลิตภัณฑ์ (Product delivery) และการตรวจรับ (Acceptance testing) หากไม่มีปัญหาใดๆ ก็ดำเนินการติดตั้งและใช้งานระบบ (System installation and operation) และกำหนดแนวทางการบำรุงรักษาระบบ (System maintenance) เมื่อเทคโนโลยีอินเทอร์เน็ตได้รับความนิยมทำให้ส่งผลต่อแนวทางการพัฒนาซอฟต์แวร์ในแง่ของการจัดหาและจำหน่ายชิ้นส่วน

ซอฟต์แวร์ (Software component) และลักษณะการทำงานที่มีเทคนิคการประมวลผลแบบกระจาย (Distributed computing) เนื่องจากปัจจัยการกระจายของข้อมูล การกระจายของการประมวลผล และการกระจายของผู้ใช้ อาทิเช่น เทคโนโลยีเอเจนต์ (Agent technology) และจากการเปลี่ยนแปลงของเทคโนโลยีในปัจจุบันทำให้มีการนำเสนอแนวคิดการพัฒนาซอฟต์แวร์เชิงบริการบนพื้นฐานแนวคิดสถาปัตยกรรมเชิงบริการ (Service-Oriented Architecture, SOA) [1,2] ซึ่งปรับแนวคิดในกระบวนการผลิตซอฟต์แวร์เป็นการใช้บริการ (Service) มาร่วมกันทำงานในกระบวนการประมวลผล

แต่อย่างไรก็ตาม ความต้องการด้านซอฟต์แวร์ของลูกค้าที่มีหลากหลายและเปลี่ยนแปลงไปตามเทคโนโลยี ประกอบกับผู้ใช้บริการมีมากขึ้นและมีเงื่อนไขในการให้บริการที่หลากหลาย การพัฒนาระบบการเจรจาต่อรอง (Negotiation system) ระหว่างผู้ให้บริการ (Service provider) และผู้ใช้บริการ (Service client /Service consumer) จึงเป็นกลไกที่สำคัญสำหรับการพัฒนาซอฟต์แวร์เชิงบริการ

งานวิจัยนี้ในระยะแรกได้เริ่มต้นจากการศึกษาความครอบคลุมของมาตรฐานที่สนับสนุนการพัฒนาผลิตภัณฑ์ซอฟต์แวร์กับแนวคิดการเจรจาต่อรองสำหรับการพัฒนาซอฟต์แวร์เชิงบริการ โดยเฉพาะแบบจำลองวุฒิภาวะความสามารถแบบบูรณาการหรือซีเอ็มเอ็มไอ (Capability Maturity Model@ Integration, CMMI) ที่ได้รับความนิยมค่อนข้างมากในกลุ่มของอุตสาหกรรมผู้ผลิตซอฟต์แวร์ ซึ่งจะนำไปสู่การเสนอกรอบแนวคิดการเจรจาต่อรองสำหรับการพัฒนาซอฟต์แวร์เชิงบริการ โดยการเพิ่มสมรรถนะของแบบจำลองวุฒิภาวะความสามารถแบบบูรณาการ และสามารถนำไปพัฒนาระบบการเจรจาต่อรองในลักษณะการทำงานแบบอัตโนมัติโดยอาศัยเทคโนโลยีทางอินเทอร์เน็ตในงานวิจัยระยะถัดไป

บทความนี้แบ่งออกเป็น 6 ส่วนโดยจะกล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้องในหัวข้อที่ 2 แนวคิดการเจรจาต่อรองของซอฟต์แวร์เชิงบริการในหัวข้อที่ 3 สำหรับหัวข้อที่ 4 เป็นการวิเคราะห์ความสอดคล้องครอบคลุมของซีเอ็มเอ็มไอกับแนวคิดการพัฒนาซอฟต์แวร์เชิงบริการและการเจรจาต่อรองที่กล่าวมา

หลังจากนั้นในหัวข้อที่ 5 จะเสนอกรอบแนวคิดการเจรจาต่อรองอัตโนมัติสำหรับสถาปัตยกรรมเชิงบริการเพื่อเพิ่มสมรรถนะของแบบจำลองซีเอ็มเอ็มไอ และส่วนสุดท้ายในหัวข้อที่ 6 จะสรุปผลและกล่าวถึงงานวิจัยที่ดำเนินการในอนาคต

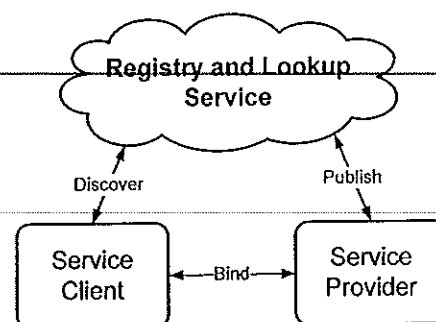
2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 สถาปัตยกรรมเชิงบริการ (Service-Oriented Architecture)

สถาปัตยกรรมเชิงบริการ (Service-Oriented Architecture, SOA) เกิดขึ้นเพื่อสนับสนุนการสื่อสารระหว่างโปรแกรมหรือชิ้นส่วนซอฟต์แวร์บนเทคโนโลยีอินเทอร์เน็ต อันที่จริงแล้วถือว่าเป็นกลไกในการประมวลผลแบบกระจาย (Distributed computing) ลักษณะหนึ่ง McIntosh [2] ได้ให้ข้อคิดเกี่ยวกับฟังก์ชันการทำงานหลักสำหรับการประยุกต์ใช้สถาปัตยกรรมเชิงบริการดังนี้

- โพรโตคอลสำหรับการประยุกต์ที่อนุญาตให้มีการสื่อสารระหว่างกันแบบกระจาย (Protocol implementation for distributed communication)
- การอธิบายการเชื่อมต่อบริการ (Service interface description)
- การเผยแพร่และค้นหาค้นหาบริการ (Service publishing and service discovery)
- การลงทะเบียนบริการ (Service registry)

โครงสร้างการทำงานหลักของสถาปัตยกรรมเชิงบริการแสดงในรูปที่ 1



รูปที่ 1 สถาปัตยกรรมซอฟต์แวร์เชิงบริการ [2]

ผู้ให้บริการ (Service Provider) จัดเตรียมและประกาศเผยแพร่ รายละเอียดบริการไว้ที่ส่วนการลงทะเบียนและสืบค้นบริการเพื่อพร้อมสำหรับการเรียกใช้งาน

ผู้ใช้บริการ (Service Client) ดำเนินการสอบถามข้อมูลจากส่วนการลงทะเบียนและสืบค้นบริการ และเรียกใช้บริการโดยตรงจากผู้ให้บริการ

ส่วนการลงทะเบียนและสืบค้นบริการ (Registry and Lookup Service) เป็นศูนย์กลางที่รวบรวมบริการต่างๆซึ่งช่วยในการสืบค้นบริการต่างๆตามต้องการของผู้ใช้

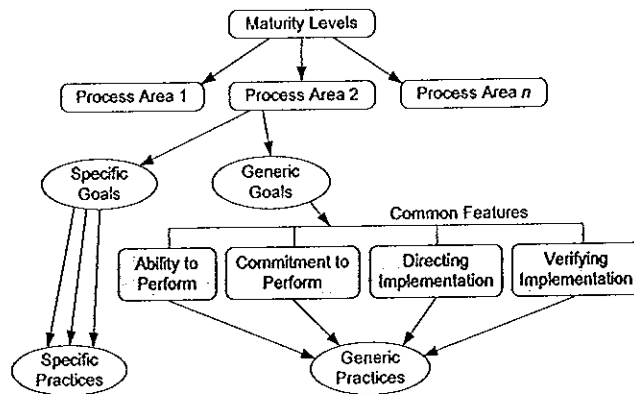
2.2 แบบจำลองวุฒิภาวะความสามารถแบบบูรณาการ (Capability Maturity Model® Integration— CMMI) [3,4]

วุฒิภาวะความสามารถแบบบูรณาการเป็นตัวแทนที่ถูกพัฒนาขึ้นเพื่อเสนอวิธีปฏิบัติที่ดีที่สุด (Best practice) ซึ่งช่วยองค์กรในการปรับปรุงกระบวนการให้มีประสิทธิภาพครอบคลุมตั้งแต่กระบวนการออกแบบจนถึงการส่งมอบและการบำรุงรักษาผลิตภัณฑ์ โดยซีเอ็มเอ็มไอถูกพัฒนาเพื่อรวม 3 แบบจำลองคือ

- 1) The Capability Maturity Model for Software (SW-CMM) v2.0 draft C
- 2) The Systems Engineering Capability Model (SECM)
- 3) The Integrated Product Development Capability Maturity Model (IPD-CMM) v0.98

ซึ่งมีองค์ความรู้ครอบคลุมในส่วนวิศวกรรมระบบ (System engineering) วิศวกรรมซอฟต์แวร์ (Software engineering) การพัฒนาผลิตภัณฑ์และกระบวนการแบบบูรณาการ (Integrated product and process development) และการจัดหาแหล่งตัวแทนจำหน่าย (Supplier sourcing) โดยองค์ประกอบของแบบจำลองซีเอ็มเอ็มไอแสดงในรูปที่ 2

ซีเอ็มเอ็มไอได้เสนอแนวปฏิบัติที่ปรับปรุงในรูปแบบของกลุ่มกระบวนการ (Process area) กล่าวคือกลุ่มของแนวปฏิบัติที่สัมพันธ์กันในด้านใดด้านหนึ่ง ซึ่งเมื่อนำไปปฏิบัติอย่างต่อเนื่องสามารถตอบสนองต่อเป้าหมายของการปรับปรุงในด้านนั้นได้ ซึ่งประกอบด้วยองค์ประกอบสำคัญ 4 ส่วนคือเป้าหมายที่ใช้งานได้

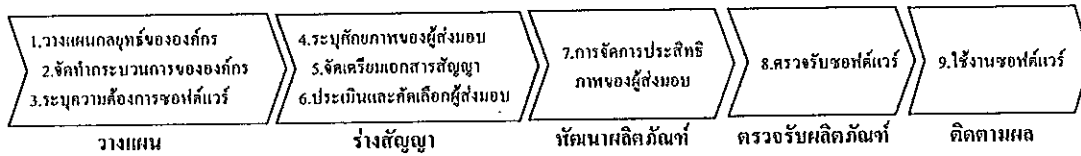


รูปที่ 2 โครงสร้างซีเอ็มเอ็มไอของการแทนรูปแบบขั้นบันได [4]

ทั่วไป (Generic goal) จะอธิบายลักษณะที่ต้องปรากฏในการปฏิบัติกระบวนการ แนวปฏิบัติที่ใช้งานได้ทั่วไป (Generic practice) บรรยายส่วนสำคัญที่จะทำให้เป้าหมายทั่วไปที่เกี่ยวข้องนั้นสำเร็จ เป้าหมายเฉพาะ (Specific goal) อธิบายคุณลักษณะเฉพาะที่ต้องปรากฏเพื่อสนองกลุ่มกระบวนการนั้น และแนวปฏิบัติเฉพาะ (Specific practice) บรรยายกิจกรรมที่คาดหวังเพื่อให้เกิดผลสำเร็จตามเป้าหมายเฉพาะของกลุ่มกระบวนการนั้น ซีเอ็มเอ็มไอถูกนำเสนอโดยการแทนอยู่ใน 2 รูปแบบคือ แบบต่อเนื่อง (Continuous representation) และแบบขั้นบันได (Staged representation)

แบบต่อเนื่องจะอนุญาตให้เลือกลำดับในการปรับปรุงประสิทธิภาพของกระบวนการตามที่ต้องการ เพื่อให้สอดคล้องใกล้เคียงกับวัตถุประสงค์ขององค์กร หรือจุดกระบวนการที่มีความเสี่ยงหรืออาจเกิดปัญหาที่องค์กรสามารถประสบได้ ซึ่งจะประเมินอยู่ในระดับความสามารถ (Capability levels) ของกลุ่มกระบวนการ (Process area) มี 6 ระดับจาก 0-5 อันได้แก่ Incomplete, Performed, Managed, Defined, Quantitatively Managed และ Optimizing ตามลำดับ

แบบขั้นบันไดจะมีแนวกำหนดลำดับในการปรับปรุงไว้เรียบร้อยแล้ว ซึ่งองค์กรที่ต้องการปรับปรุงกระบวนการจะดำเนินการจากลำดับพื้นฐาน และพัฒนาก้าวหน้าขึ้นเรื่อยๆจนถึงระดับที่ต้องการ โดยประเมินอยู่ในระดับวุฒิภาวะ (Maturity Levels) มี 5 ระดับจาก 1-5 อันได้แก่ Initial, Managed, Defined, Quantitatively Managed และ Optimizing ตามลำดับ



รูปที่ 3 ขั้นตอนการดำเนินงานของการจัดหาซอฟต์แวร์ตามมาตรฐานไออีอีอี 1062

2.3 ประเด็นการเจรจาต่อรองสำหรับการพัฒนาซอฟต์แวร์เชิงบริการในมาตรฐานอื่น

2.3.1 มาตรฐานไออีอีอี 1062, ฉบับพิมพ์ปี 1998 ข้อเสนอแนะการปฏิบัติสำหรับการจัดหาซอฟต์แวร์ (IEEE Standard 1062, 1998 Edition – Recommend Practice for Software Acquisition) [5]

มาตรฐานไออีอีอี 1062 เป็นมาตรฐานของการแนะนำแนวปฏิบัติสำหรับการจัดหาผลิตภัณฑ์ซอฟต์แวร์โดยอิงตามการปฏิบัติที่มีคุณภาพ ซึ่งสามารถเลือกหรือประยุกต์ใช้งานในกระบวนการจัดหาซอฟต์แวร์ได้

การจัดหาซอฟต์แวร์จะอธิบายไปตามช่วงเวลาตั้งแต่เริ่มต้นตัดสินใจจัดหาผลิตภัณฑ์ซอฟต์แวร์และสิ้นสุดลงเมื่อไม่ใช้งานผลิตภัณฑ์นั้นอีกต่อไป สามารถแบ่งออกได้เป็น 5 ระยะย่อย ได้แก่

1) ระยะเวลาวางแผน (Planning phase) เริ่มต้นจากแนวคิดหรือความจำเป็นในการจัดหาซอฟต์แวร์จนกระทั่งได้เอกสารแสดงข้อเสนอที่ต้องการ

2) ระยะเวลาสัญญา (Contracting phase) ร่างสัญญาซึ่งรวมถึงกิจกรรมที่ทำให้มั่นใจว่าผลิตภัณฑ์และบริการเป็นไปตามความต้องการของผู้จัดหา

3) ระยะเวลาพัฒนาผลิตภัณฑ์ (Product implementation phase) พัฒนาผลิตภัณฑ์ตามที่มอบหมายพร้อมทั้งติดตามผลการดำเนินงานของผู้รับจ้าง

4) ระยะเวลาตรวจรับผลิตภัณฑ์ (Product acceptance phase) ประเมิน ทดสอบ และยอมรับผลิตภัณฑ์

5) ระยะติดตามผล (Follow-on phase) ประยุกต์ใช้ผลิตภัณฑ์และประเมินความพึงพอใจของผู้ใช้คือซอฟต์แวร์ เอกสารกำกับ และการสนับสนุนจากผู้รับจ้าง

ในแต่ละระยะสามารถแบ่งกระบวนการจัดหาซอฟต์แวร์ออกเป็น 9 ขั้นตอนดังแสดงในรูปที่ 3 และจากการตรวจสอบรายละเอียดยังขาดประเด็นเกี่ยวกับการเจรจาต่อรองสำหรับซอฟต์แวร์เชิงบริการ

2.3.2 มาตรฐานสากลไอเอสไอ/ไออีซี 12207 มาตรฐานสำหรับเทคโนโลยีสารสนเทศ – กระบวนการวัฏจักรชีวิตของซอฟต์แวร์ (International Standard ISO/IEC 12207: Standard for Information Technology – Software life cycle processes) [6]

มาตรฐานไอเอสไอ/ไออีซี 12207 เป็นการรวบรวมโครงสร้างแนวทางในการปฏิบัติโดยทั่วไปสำหรับแต่ละกระบวนการในวัฏจักรชีวิตซอฟต์แวร์ (Software life cycle) ซึ่งไม่ได้แสดงรายละเอียดว่าจะต้องทำการปฏิบัติอย่างไรหรือเอกสารจากกระบวนการต้องมีรูปแบบเป็นอย่างไร ในแต่ละกระบวนการประกอบด้วยกลุ่มของกิจกรรม (Activity) ต่างๆและแต่ละกิจกรรมประกอบด้วยกลุ่มของงาน (Task) ที่ต้องทำให้บรรลุวัตถุประสงค์ โดยจะประกอบด้วยกลุ่มกระบวนการหลักทั้งหมด 3 กลุ่มได้แก่ กลุ่มกระบวนการขั้นพื้นฐาน (Primary Processes) เป็นกลุ่มกระบวนการหลักที่ใช้ในการพัฒนาระบบซอฟต์แวร์ให้เกิดขึ้นได้ กลุ่มกระบวนการวัฏจักรชีวิตด้านการสนับสนุน (Supporting life cycle processes) จะสนับสนุนกระบวนการต่างๆ เพื่อให้โครงการเกิดคุณภาพและประสบความสำเร็จ กลุ่มกระบวนการวัฏจักรชีวิตด้านองค์กร (Organizational life cycle processes) จะเกี่ยวข้องกับภาพรวมในระดับองค์กร อาทิเช่น การจัดการโครงการ การพัฒนาบุคคล เป็นต้น

กลุ่มกระบวนการขั้นพื้นฐานจะประกอบด้วยกระบวนการที่สำคัญ 5 กระบวนการคือ การจัดหา (Acquisition) การได้มา (Supply) การพัฒนา (Development) การดำเนินการ (Operation)

และการบำรุงรักษา (Maintenance) ซึ่งในส่วนกระบวนการจัดการจะมีกิจกรรมและรายละเอียดงานดังนี้

1) การเริ่มต้น (Initiation) เป็นขั้นตอนระบุความต้องการ กำหนดความต้องการของทั้งระบบและซอฟต์แวร์ เตรียมแผนการจัดการและกำหนดกลยุทธ์การตรวจรับ

2) การร้องขอข้อเสนอ (Request for proposal) เป็นขั้นตอนจัดเตรียมข้อมูลต่างๆที่จำเป็นให้แก่ตัวแทนจำหน่าย เช่น เอกสารความต้องการการจัดการ แผนกำหนดเวลาในการเซ็นสัญญา เป็นต้น

3) การจัดเตรียมสัญญา (Contract preparation) เริ่มต้นจากการสร้างวิธีประเมินผู้เสนอโครงการเข้ามา แล้วประเมินคัดเลือกตัวแทนจำหน่าย สุดท้ายกระทำการเขียนสัญญา

4) การติดตามดูตัวแทนจำหน่าย (Supplier monitoring) เป็นการติดตามการดำเนินงานของตัวแทนจำหน่ายทั้งการตรวจสอบและตรวจสอบความถูกต้องต่างๆ

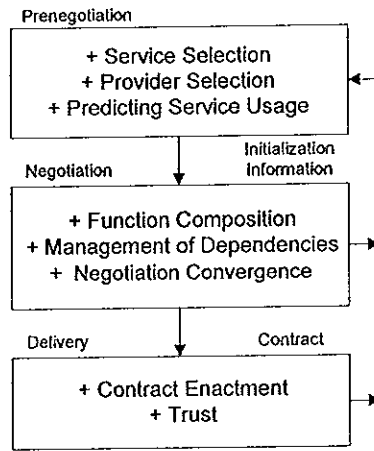
5) การตรวจรับและทำให้สมบูรณ์ (Acceptance and completion) เป็นขั้นตอนสุดท้ายของกระบวนการสรรหาคือ การตรวจรับระบบโดยตรวจรับตามเกณฑ์ต่างๆ ทบทวนและทดสอบและปรับแต่งจัดการซอฟต์แวร์ที่ส่งมอบมา

เมื่อตรวจสอบรายละเอียดของกระบวนการนี้แล้วยังขาดประเด็นที่เกี่ยวข้องกับการเจรจาต่อรองสำหรับซอฟต์แวร์เชิงบริการด้วยเช่นกัน

3. แนวคิดการเจรจาต่อรองของซอฟต์แวร์เชิงบริการ

Sierra และคณะ [7] ได้ให้ความหมายการเจรจาต่อรอง (Negotiation) ว่าเป็นกระบวนการที่สร้างการตัดสินใจร่วมกันระหว่างสองฝ่ายขึ้นไป อันดับแรกฝ่ายหนึ่งจะเสนอสิ่งที่แย้งกับความต้องการของตน และจากนั้นพยายามเคลื่อนไปสู่การยินยอมตกลง ซึ่งผู้มีส่วนได้ส่วนเสียที่เกี่ยวข้องได้ผลประโยชน์ตามความต้องการของตน

การเจรจาต่อรองถูกนำไปใช้หลากหลายบริบท อาทิเช่น วิทยาศาสตร์ สังคมศาสตร์ เศรษฐศาสตร์ เป็นต้น Elafatry และ Layzell [8,9] ได้เสนอกรอบแนวคิดการทำงานของเจรจา



รูปที่ 4 กรอบแนวคิดการเจรจาต่อรองของซอฟต์แวร์เชิงบริการ [9]

ต่อรองที่มองในบริบทซอฟต์แวร์เชิงบริการ โดยพิจารณาตามคุณลักษณะของตลาดซึ่งสามารถแบ่งออกได้เป็น 3 ระยะดังรูปที่ 4

1) ระยะก่อนการเจรจาต่อรอง (Prenegotiation) เป็นระยะเตรียมข้อมูลต่างๆที่จำเป็นให้พร้อมก่อนดำเนินการเจรจาต่อรอง แบ่งย่อยได้ 3 ขั้นตอน ได้แก่ การคัดเลือกบริการ (Service selection) เนื่องจากในตลาดมีบริการที่นำเสนออยู่หลากหลาย จึงจำเป็นต้องมีวิธีการเลือกให้เหมาะสม การคัดเลือกผู้ให้บริการ (Provider selection) ต้องมีการคัดเลือกให้เหมาะสมเช่นเดียวกับการคัดเลือกบริการ โดยอาจจะค้นหาโดยวัดจากข้อมูลตอบกลับ (Feedback) ที่ได้จากระยะถัดไป และการทำนายการใช้บริการ (Predicting service usage) ในการส่งข่าวสารของแต่ละฝ่าย แต่ละครั้งอาจมีการส่งข่าวสารติดต่อกันในปริมาณมากกว่าจะบรรลุข้อตกลง ซึ่งจะใช้ทรัพยากรของระบบสูงขึ้นตามไปด้วย ดังนั้นจำเป็นต้องทำนายแนวทางการใช้บริการเพื่อควบคุมการดำเนินการ

2) ระยะเจรจาต่อรอง (Negotiation) เป็นระยะดำเนินการเจรจาต่อรอง โดยใช้ข้อมูลที่เตรียมเอาไว้จนบรรลุข้อตกลง แบ่งย่อยได้ 3 ขั้นตอน ได้แก่ การประกอบส่วนการทำงาน (Functional composition) ทำการประกอบบริการเพื่อให้ผลการดำเนินการที่เหมาะสม การจัดการภาระที่พึ่งพิง (Management of dependencies) จำเป็นต้องคำนึงถึงความพึ่งพิงกันระหว่างบริการ เช่น บริการห้องเที่ยวประกอบด้วยบริการที่ต้องพึ่งพิงกัน ได้แก่ บริการจองตั๋วเครื่องบินและบริการจองที่พักซึ่งต้องมีลำดับการ

ทำงานที่เหมาะสม และขั้นตอนสุดท้ายเป็นการบรรลุเจรจาต่อรอง (Negotiation convergence) ซึ่งต้องดำเนินการให้แล้วเสร็จโดยเร็ว

3) ระยะเวลาส่งมอบบริการ (Service Delivery) เป็นระยะส่งมอบบริการที่เสร็จสมบูรณ์และพร้อมใช้งานแก่ผู้บริโภคนำแบ่งย่อยได้ 2 ขั้นตอนได้แก่ การร่างสัญญา (Contract enactment) สัญญาอาจจะอธิบายการดำเนินงานของบริการ เงื่อนไข คุณสมบัติ เป็นต้น และความไว้วางใจ (Trust) เป็นสิ่งที่ผู้ให้บริการต้องทำให้เกิดขึ้นเพราะว่าผู้บริโภคมักมีโอกาสที่จะเลือกบริการ

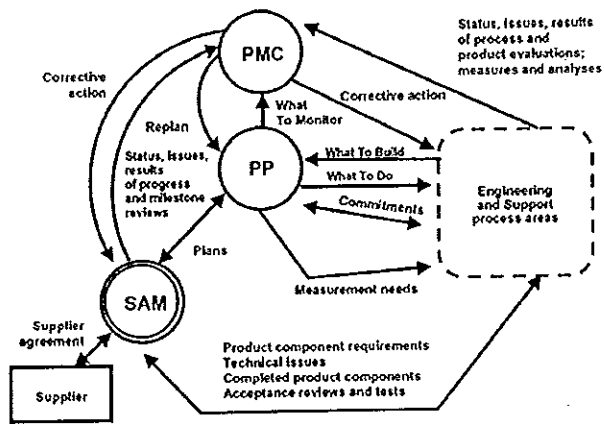
4. การวิเคราะห์ความครอบคลุมของแบบจำลองซีเอ็มเอ็มไอโอ ในประเด็นการพัฒนาซอฟต์แวร์เชิงบริการและการเจรจาต่อรอง

จากการวิเคราะห์รายละเอียดความรู้ของขอบเขตกระบวนการสำหรับวิศวกรรมระบบและวิศวกรรมซอฟต์แวร์ จะเน้นในประเด็นต่อไปนี้

- Casual Analysis and Resolution
- Configuration Management
- Decision Analysis and Resolution
- Integrated Project Management
- Integrated Supplier Management
- Integrated Teaming
- Measurement and Analysis
- Organizational Environment for Integration
- Organizational Innovation and Deployment
- Organizational Process Definition
- Organizational Process Focus
- Organizational Process Performance
- Organizational Training
- Product Integration
- Project Monitoring and Control
- Project Planning
- Process and Product Quality Assurance
- Quantitative Project Management
- Requirements Development
- Requirements Management
- Risk Management
- Supplier Agreement Management
- Technical Solution
- Validation
- Verification

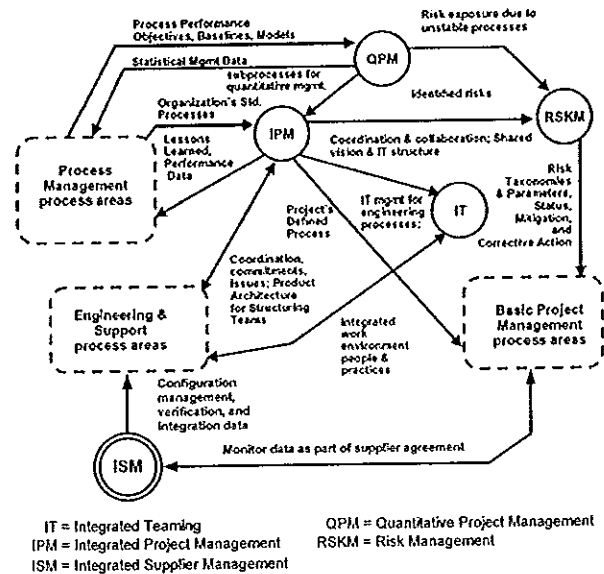
ส่วนขอบเขตกระบวนการสำหรับการพัฒนาผลิตภัณฑ์และกระบวนการแบบบูรณาการได้เพิ่มในประเด็น Integrated Teaming และ Organizational Environment for Integration และขอบเขตกระบวนการสำหรับการสรรหาแหล่งตัวแทนจำหน่ายเพิ่มเติมในประเด็น Integrated Supplier Management

ดังนั้นจากรายละเอียดประเด็นข้างต้น เมื่อพิจารณาการเจรจาต่อรองสำหรับการพัฒนาซอฟต์แวร์เชิงบริการ ควรมีการเพิ่มเติม



SAM = Supplier Agreement Management
PP = Project Planning
PMC = Project Monitoring and Control

รูปที่ 5 กลุ่มกระบวนการจัดการ โครงการขั้นพื้นฐาน [4]



IT = Integrated Teaming
IPM = Integrated Project Management
ISM = Integrated Supplier Management
QPM = Quantitative Project Management
RSKM = Risk Management

รูปที่ 6 กลุ่มกระบวนการจัดการ โครงการแบบก้าวหน้า [4]

ความครอบคลุมในส่วนต่อไปนี้คือ Supplier Agreement Management (SAM) และ Integrated Supplier Management (ISM) ซึ่งก็คือการจัดการข้อตกลงกับตัวแทนจำหน่ายและการจัดการตัวแทนจำหน่ายแบบบูรณาการตามลำดับ ซึ่งอยู่ในหมวดการจัดการโครงการ (Project Management) ตามรูปที่ 5 และ 6 ทั้งนี้ในส่วนของการจัดการตัวแทนจำหน่ายแบบบูรณาการ (Integrated Supplier Management) จะเกี่ยวข้องกับกิจกรรมต่อไปนี้

- 1) การระบุ การวิเคราะห์ และการเลือกแหล่งของผลิตภัณฑ์ที่เป็นไปได้
- 2) การประเมินและการกำหนดแหล่งที่จะใช้ในการจัดหาผลิตภัณฑ์
- 3) การดูแลและการวิเคราะห์กระบวนการของตัวแทนจำหน่ายที่ถูกเลือก
- 4) การประเมินผลิตภัณฑ์งานของตัวแทนจำหน่ายที่ถูกเลือก
- 5) การทบทวนข้อตกลงตัวแทนจำหน่ายหรือความสัมพันธ์ ความเหมาะสม

เมื่อพิจารณาเป้าหมายเฉพาะและแนวปฏิบัติเฉพาะในส่วนนี้ที่อ้างอิงถึงตัวแทนจำหน่ายจะต้องไปคำนึงถึงผู้ให้บริการ (Service provider) และในส่วนของผลิตภัณฑ์จะต้องคำนึงเป็นค่าบริการ (Service) แทน

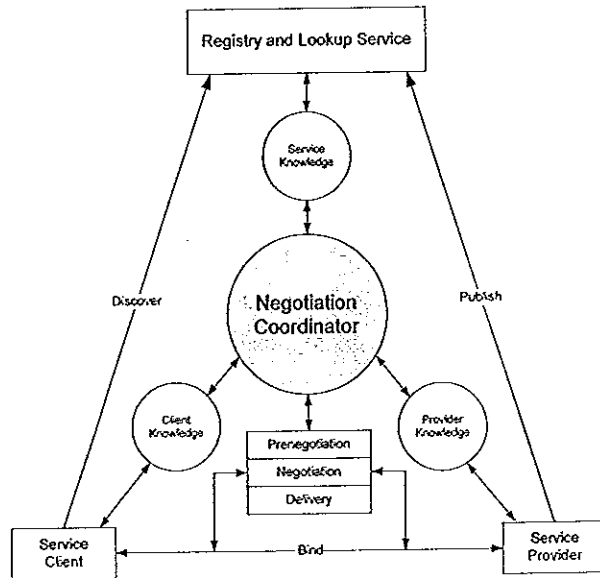
ในทำนองเดียวกันในกรณีของการจัดการข้อตกลงตัวแทนจำหน่าย (Supplier Agreement Management) ซึ่งเกี่ยวข้องกับกิจกรรมดังต่อไปนี้

- 1) การกำหนดชนิดของการจัดหาที่ใช้สำหรับจัดหาผลิตภัณฑ์
- 2) การคัดเลือกตัวแทนจำหน่าย
- 3) การสร้างและจัดการข้อตกลงกับตัวแทนจำหน่าย
- 4) การปฏิบัติตามข้อตกลงกับตัวแทนจำหน่าย
- 5) การตรวจรับการส่งมอบผลิตภัณฑ์ที่จัดหามา
- 6) การนำผลิตภัณฑ์ที่ได้รับมาใช้กับโครงการต่อไป

เมื่อพิจารณาเป้าหมายเฉพาะและแนวปฏิบัติเฉพาะในส่วนนี้ที่อ้างอิงถึงตัวแทนจำหน่ายจะต้องไปคำนึงถึงผู้ให้บริการและในส่วนของผลิตภัณฑ์จะต้องคำนึงเป็นค่าบริการเช่นเดียวกัน นอกจากนี้ควรมีการเพิ่มเกี่ยวกับประเด็นการเจรจาต่อรองไว้ในหมวดการจัดการโครงการด้วย

5. กรอบแนวคิดการเจรจาต่อรองอัตโนมัติสำหรับสถาปัตยกรรมเชิงบริการเพื่อเพิ่มสมรรถนะของแบบจำลองซีเอ็มเอ็มไอ

5.1 ผู้ประสานงานการต่อรอง (Negotiation Coordinator)



รูปที่ 7 กรอบแนวคิดของสถาปัตยกรรมของระบบเจรจาต่อรองสำหรับการพัฒนาซอฟต์แวร์เชิงบริการ

ทั้งนี้จากการวิเคราะห์ความครอบคลุมของซีเอ็มเอ็มไอกับการเจรจาต่อรองของซอฟต์แวร์เชิงบริการ เมื่อพิจารณาหมวดทางวิศวกรรมซอฟต์แวร์และระบบจะพบว่าสามารถเพิ่มการปฏิบัติในขอบเขตกระบวนการแก้ไขปัญหาเชิงเทคนิค (Technical Solution) ในเทคนิคการออกแบบและการโปรแกรมซอฟต์แวร์โดยอาศัยสถาปัตยกรรมเชิงบริการซึ่งยังไม่ได้กล่าวถึง

ดังนั้นงานวิจัยนี้จึงเสนอกรอบแนวคิดการเจรจาต่อรองแบบอัตโนมัติเพื่อสนับสนุนซอฟต์แวร์เชิงบริการ โดยอำนวยความสะดวกให้แก่ทั้งผู้ให้บริการและผู้ใช้บริการเจรจาต่อรองผ่านเครือข่ายอินเทอร์เน็ต ดังแสดงในรูปที่ 7

ความรู้ความเข้าใจเกี่ยวกับบริการ ผู้ให้บริการและผู้ใช้บริการจะเป็นส่วนสำคัญที่จะทำให้การเจรจาต่อรองสำเร็จบรรลุผล แต่ในสภาพแวดล้อมเชิงบริการนั้น ระบบสามารถเปลี่ยนแปลงได้ตลอดเวลาส่งผลให้ผู้ให้บริการและผู้ใช้บริการมีความรู้ระหว่างกันไม่สมบูรณ์แบบ [9] กล่าวคือในส่วนผู้ให้บริการไม่มีความรู้ที่เพียงพอเกี่ยวกับผู้ให้บริการ และในทางกลับกันผู้ให้บริการก็ไม่มีความรู้ที่เพียงพอเกี่ยวกับผู้ให้บริการ นอกจากนั้นทั้งสองก็ยังไม่มีความรู้ความเข้าใจในบริการที่เพียงพอด้วยเช่นเดียวกัน

ดังนั้นผู้ประสานงานการต่อรอง (Negotiation coordinator) จะเป็นตัวกลางที่ช่วยควบคุมและจัดการความรู้ของผู้ให้บริการ ผู้ใช้บริการและตัวบริการ ตลอดจนความรู้ที่เกี่ยวข้องกับ กระบวนการเจรจาต่อรองที่พึงมี ช่วยให้เกิดการแลกเปลี่ยนความรู้ ระหว่างกัน และยังเพิ่มศักยภาพให้กระบวนการการเจรจาต่อรอง สามารถดำเนินการ ได้อย่างสมบูรณ์ โดยหน้าที่หลักของผู้ ประสานงานการต่อรองในเบื้องต้นมีดังนี้

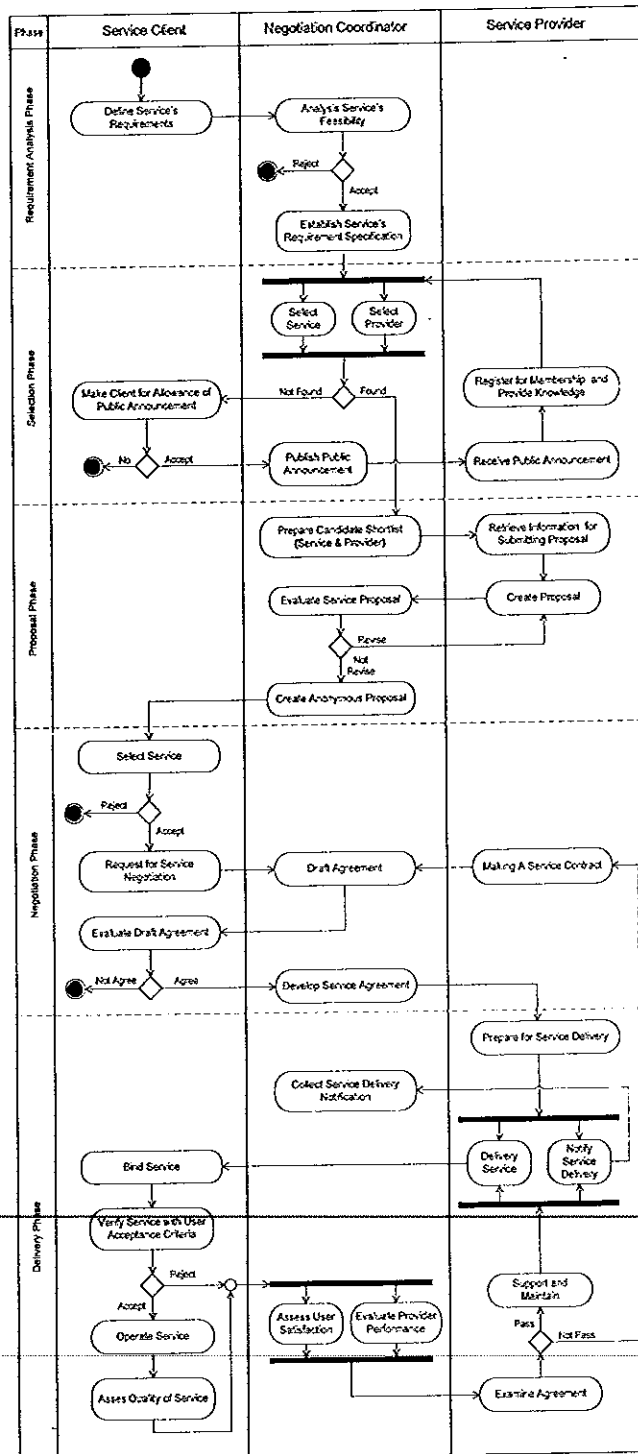
- 1) การจัดการความรู้ซึ่งประกอบด้วยความรู้เกี่ยวกับบริการ ความรู้เกี่ยวกับผู้ให้บริการและความรู้เกี่ยวกับผู้ใช้บริการ
- 2) สนับสนุนการเจรจาต่อรองซึ่งครอบคลุมตั้งแต่ช่วงก่อนการ เจรจาต่อรอง ช่วงเจรจาต่อรอง และช่วงส่งมอบบริการเพื่อใช้งาน ตลอดจนการลงนามรับรองการจับคู่ธุรกิจเชิงบริการและเป็นพยาน

โดยรูปแบบการดำเนินงานโดยทั่วไปของผู้ประสานงานการ ต่อรองมีลักษณะการสื่อสารไปยังผู้ที่เกี่ยวข้องแบบ 2 ทางดังนี้

- 1) ผู้ประสานงานการต่อรองสามารถสืบค้นบริการที่น่าสนใจ และจัดเก็บความรู้เกี่ยวกับบริการนั้นๆ อีกทั้งส่งข่าวไปยังผู้ ให้บริการเพื่อชักชวนให้เป็นสมาชิกของระบบ สำหรับข้อมูล เกี่ยวกับตัวบริการ เช่น ชื่อบริการ คำอธิบายโดยย่อ เจ้าของหรือ ตัวแทนจำหน่าย มูลค่าของตัวบริการ ระยะเวลาที่ส่งมอบ บริการ หลังการส่งมอบ เป็นต้น

- 2) ผู้ให้บริการสามารถสมัครเป็นสมาชิกของระบบโดยให้ ข้อมูลต่างๆที่เป็นประโยชน์ต่อการจับคู่กับผู้ใช้บริการ ข้อมูลที่เป็น ประโยชน์และสามารถจัดเก็บเป็นความรู้ในการต่อรอง เช่น ขอบข่ายประเภทธุรกิจของบริการ (Business domain of service) ประวัติความสามารถในการให้บริการหรือรางวัลที่เคยได้รับ เขต พื้นที่ที่ให้บริการ หรือมีตัวแทนจำหน่าย ทุนจดทะเบียนประกัน ความมั่นคงของผู้ให้บริการ ทั้งนี้ผู้ให้บริการสามารถพัฒนาตัว บริการและประกาศเผยแพร่สู่สาธารณะได้อย่างอิสระ

- 3) ผู้ใช้บริการหากต้องการให้ผู้ประสานงานการต่อรองเป็น สื่อกกลางจับคู่ธุรกิจสามารถลงทะเบียนเป็นสมาชิกของระบบและ ให้อินพุตที่เป็นประโยชน์ต่อการเจรจาต่อรอง เช่น ประเภทธุรกิจ ขนาดเงินทุนของธุรกิจ หลักประกันความมั่นคงของบริษัทหรือ ประกันความน่าเชื่อถือ



รูปที่ 8 แผนภาพกิจกรรมของการเจรจาต่อรองสำหรับ สถาปัตยกรรมเชิงบริการที่สอดคล้องกับซีเอ็มเอ็มไอ

นอกจากนี้ผู้ประสานงานสามารถเผยแพร่ข้อมูลระดับความน่าเชื่อถือของผู้ให้บริการและผู้ให้บริการ ตลอดจนถึงคุณภาพของบริการ (Quality of Service) ต่อสมาชิกที่ลงทะเบียนกับระบบให้รับทราบ เพื่อประโยชน์ต่อทุกฝ่ายในการพัฒนาศักยภาพของธุรกิจเชิงบริการและความมั่นคงในการดำเนินธุรกิจในอนาคต

5.2 กระบวนการเจรจาต่อรอง

จากการวิเคราะห์แบบจำลองวุฒิภาวะความสามารถแบบบูรณาการถึงความครอบคลุมประเด็นเกี่ยวกับการพัฒนาซอฟต์แวร์เชิงบริการและการเจรจาต่อรองดังที่กล่าวมาในหัวข้อที่ 4 ผู้วิจัยได้ออกแบบกระบวนการการเจรจาต่อรองของซอฟต์แวร์เชิงบริการบนพื้นฐานการจัดการซอฟต์แวร์ตามมาตรฐาน ไออีอีอี 1062, กลุ่มกระบวนการจัดการข้อตกลงตัวแทนจำหน่ายของซีเอ็มเอ็มไอ และอ้างอิงตามวิจัยที่เกี่ยวข้อง [10] สามารถแบ่งกระบวนการออกเป็น 5 ระยะได้แก่ ระยะการวิเคราะห์ความต้องการ (Requirement analysis phase) ระยะการคัดเลือก (Selection phase) ระยะการเสนอ (Proposal Phase) ระยะการเจรจาต่อรอง (Negotiation phase) และระยะการส่งมอบ (Delivery phase) สามารถแสดงได้ด้วยแผนภาพกิจกรรมในรูปที่ 8

ระยะการวิเคราะห์ความต้องการ เริ่มต้นจากที่ผู้เรียกใช้บริการกำหนดความต้องการของตนเองและแจ้งต่อผู้ประสานงานการต่อรองให้ทราบ จากนั้นผู้ประสานงานการต่อรองจึงประเมินความเป็นไปได้ของความต้องการบริการที่ผู้เรียกใช้บริการระบุมา โดยใช้ข้อมูลจากฐานความรู้ที่มีอยู่ และจะส่งข่าวสารแจ้งผลการประเมินกลับสู่ผู้เรียกใช้บริการให้รับทราบ ถ้าความต้องการบริการมีความเป็นไปได้ ผู้ประสานงานการต่อรองจะจัดทำข้อกำหนดความต้องการ (Requirement specification)

ระยะการคัดเลือก ผู้ประสานงานการต่อรองดำเนินการคัดเลือกบริการและผู้ให้บริการตามข้อกำหนดความต้องการ ถ้าไม่พบผู้ให้บริการและบริการที่เหมาะสม ผู้ประสานงานการต่อรองจะแจ้งเกี่ยวกับการประกาศข้อกำหนดความต้องการแก่สาธารณะให้ผู้เรียกใช้บริการพิจารณาอนุญาต ถ้าผู้เรียกใช้บริการอนุญาต ผู้ประสานงานการต่อรองจะดำเนินการ ประกาศข้อกำหนดความ

ต้องการแก่สาธารณะ ผู้ให้บริการที่จะรับข้อกำหนดความต้องการนี้ไปพิจารณา หากรับข้อเสนอที่จะลงทะเบียนกับผู้ประสานงานการต่อรองพร้อมกับให้ความรู้เพื่อผู้ประสานงานการต่อรองนำไปใช้เป็นข้อมูลความรู้ในอนาคต

ระยะการเสนอ เมื่อคัดเลือกบริการและผู้ให้บริการเรียบร้อยแล้ว ผู้ประสานงานการต่อรองจะจัดเตรียมบัญชีสรุปรายการของผู้เข้าแข่งขัน (Candidate shortlist) ซึ่งเป็นรายการของทั้งผู้ให้บริการและบริการที่ใกล้เคียงหรือเหมาะสมสอดคล้องกับความต้องการที่ระบุ และแจ้งให้ผู้ให้บริการเหล่านี้ยื่นข้อเสนอบริการ (Service proposal) เข้ามา

เมื่อผู้ประสานงานการต่อรองได้รับข้อเสนอบริการจากผู้ให้บริการแล้วจึงประเมินข้อเสนอเห็นว่าต้องดำเนินการทบทวนแก้ไขหรือไม่ ถ้าจำเป็นต้องแก้ไขก็ส่งข่าวสารให้ผู้ให้บริการแก้ไขข้อเสนอและนำกลับมาเสนอใหม่อีกครั้ง จากนั้นผู้ประสานงานการต่อรองจัดทำข้อเสนอบริการที่ไม่มีภาระผูกพันผู้ให้บริการ เพื่อส่งให้ผู้เรียกใช้พิจารณาคัดเลือกบริการจากข้อเสนอบริการด้วยความเป็นธรรม

ระยะการเจรจาต่อรอง เมื่อผู้เรียกใช้บริการคัดเลือกบริการจากข้อเสนอบริการที่ไม่ระบุตัวผู้ให้บริการแล้ว ผู้เรียกใช้บริการจะร้องขอการเจรจาต่อรองเพื่อใช้บริการนั้น และหลังจากเจรจาต่อรองบรรลุผล ผู้ให้บริการก็จะจัดทำสัญญาการบริการของตน จากนั้นผู้ประสานงานการต่อรองจึงร่างข้อตกลงบริการ (Service agreement) เพื่อให้ผู้เรียกใช้บริการได้ประเมินข้อตกลงบริการก่อนจะยินยอมทำข้อตกลงบริการที่แท้จริง ถ้าผ่านการประเมิน ทั้งผู้ให้บริการและผู้เรียกใช้บริการลงนามในข้อตกลงการให้บริการระหว่างกัน และมีผู้ประสานงานการต่อรองลงนามรับรองการจับคู่ธุรกิจและเป็นพยานในสัญญา ถ้าผู้เรียกใช้บริการไม่ยอมรับร่างข้อตกลงบริการ ผู้ประสานงานการต่อรองก็ต้องเก็บผลตอบแทนไว้ด้วย

ระยะการส่งมอบ หลังจากทำข้อตกลงบริการเรียบร้อยแล้ว ผู้ให้บริการจะเตรียมความพร้อมที่จะส่งมอบบริการ หลังจากนั้นก็จะมีการส่งมอบบริการพร้อมกับแจ้งการส่งมอบให้ผู้ประสานงานการต่อรองจัดเก็บรวบรวมไว้เป็นหลักฐานด้วย

จากนั้นผู้เรียกใช้บริการก็จะได้รับบริการ แล้วจึงตรวจสอบความถูกต้องของบริการ ถ้าตรวจสอบไม่ผ่านจะแจ้งผลให้ผู้ประสานงานการต่อรองทราบ เพื่อส่งผลการประเมินไปยังผู้ให้บริการต่อไป ถ้าตรวจสอบผ่านถือว่าสามารถเรียกใช้บริการนั้นได้สมบูรณ์ และในระหว่างการใช้งานนั้นผู้เรียกใช้บริการก็จะประเมินคุณภาพของบริการ (Quality of service) ไปด้วย โดยส่งผลการประเมินให้ผู้ประสานงานการต่อรองประเมินความพึงพอใจของผู้เรียกใช้บริการและประสิทธิภาพของผู้ให้บริการเพื่อเก็บไว้เป็นข้อมูลความรู้ในระบบ

ทั้งนี้เมื่อผู้ให้บริการได้รับผลการประเมินจะดำเนินการตรวจสอบเงื่อนไขในข้อตกลงด้วยว่าสอดคล้องกันหรือผิดเงื่อนไขหรือไม่ เช่น ช่วงระยะเวลาในสัญญาการให้บริการ เป็นต้น ถ้าเป็นไปได้ตามเงื่อนไขข้อตกลง ผู้ให้บริการก็จะดำเนินการสนับสนุนและบำรุงรักษาบริการ แต่ถ้าผิดเงื่อนไขข้อตกลง ผู้ให้บริการจะดำเนินการจัดทำสัญญาเพื่อเจรจาข้อตกลงเกี่ยวกับบริการอีกครั้งหนึ่ง

6. บทสรุปและงานวิจัยในอนาคต

งานวิจัยนี้ได้ศึกษาความสอดคล้องครอบคลุมของแนวคิดการเจรจาต่อรองสำหรับสถาปัตยกรรมเชิงบริการกับแนวทางปรับปรุงกระบวนการตามแบบจำลองวุฒิภาวะความสามารถแบบบูรณาการและมาตรฐานอื่นที่เกี่ยวข้อง โดยผลจากการศึกษากลุ่มมาตรฐานดังกล่าวยังขาดการกล่าวถึงการพิจารณาในประเด็นที่เกี่ยวข้องกับสถาปัตยกรรมเชิงบริการ โดยเฉพาะในประเด็นการเจรจาต่อรอง ซึ่งจำเป็นต้องมีการปรับเปลี่ยนเพิ่มเติมในลำดับถัดไป

นอกจากนี้ผู้วิจัยยังได้เสนอกรอบแนวคิดและขั้นตอนวิธีที่จะเป็นแนวทางในการพัฒนาการเจรจาต่อรองของซอฟต์แวร์เชิงบริการแบบอัตโนมัติ ซึ่งมีผู้ประสานงานการต่อรองเป็นตัวกลางช่วยสนับสนุนให้การดำเนินการเจรจาต่อรองบรรลุผลสำเร็จ อย่างไรก็ตามการปรับเปลี่ยนสภาพแวดล้อมผู้เจรจาต่อรองแบบอัตโนมัติอย่างเต็มรูปแบบซึ่งผู้ให้บริการและผู้ใช้บริการไม่ต้องเผชิญหน้ากันนั้น ความรู้เกี่ยวกับบริการ ความรู้เกี่ยวกับผู้ให้บริการ และความรู้เกี่ยวกับผู้ให้บริการจะเป็นปัจจัยสำคัญของ

ระบบ ซึ่งการประยุกต์ใช้เทคโนโลยีเอเจนต์เพื่อการทำงานอัตโนมัติ และแนวคิดออนโทโลยีเพื่ออธิบายความรู้ที่เกี่ยวข้องให้สนับสนุนการทำงานของระบบเจรจาต่อรองอัตโนมัติได้จะเป็นงานวิจัยในอนาคต

7. เอกสารอ้างอิง

- [1] K. Bennett, P. Layzell, D. Budgen, P. Brereton, L. Macaulay, and M. Munro, "Service-Based Software: The Future for Flexible Software," *Proceeding of the 9th Asia-Pacific Software Engineering Conference (APSEC'02)*, IEEE CS Press, pp. 214-222, 5-8 Dec 2000
- [2] R.L. McIntosh, "Open-Source Tools for Distributed Device Control within A Service-Oriented Architecture," *Journal of the Association for Laboratory Automation*, pp. 404-410, Dec 2004
- [3] M.B. Chrissis, M. Konrad, and S. Shrum, *CMMI[®] Guidelines for Process Integration and Product Improvement*, Addison-Wesley Professional, 6 Mar 2003
- [4] CMMI Product Team, "CMMI for Development, Version 1.1," *Technical Report*, Software Engineering Institution (SEI), August 2002
- [5] Institute of Electrical and Electronics Engineers, "IEEE Recommended Practice for Software Acquisition," *IEEE Standard 1062, 1998 Edition*, IEEE Computer Society, Dec 1998
- [6] R. Singh, "International Standard ISO/IEC 12207 Software Life Cycle Processes," *Federal Aviation Administration*, Washington DC, USA, 23 Aug 1996
- [7] C. Sierra, P. Faratin, and N.R. Jennings, "A Service-Oriented Negotiation Model between Autonomous Agents," *Human and Artificial Societies*, Springer-Verlag Berlin Heidelberg, pp. 103-108, 1999
- [8] A. Elfatraty, and P. Layzell, "Software As A Service: Negotiation Perspective," *Proceeding of 26th Annual International Computer Software and Applications Conference (COMPSAC'02)*, IEEE Society Press, pp. 501-506, 26-29 Aug 2002
- [9] A. Elfatraty, and P. Layzell, "Negotiating in Service-Oriented Environments," *Communications of the ACM*, vol.47, no.8, pp. 103-108, August 2004
- [10] J. Saemu, and N. Prompoon, "Tool and guidelines Support for Capability Maturity Model's Software Subcontract Management," *Proceeding of the 11th Asia-Pacific Software Engineering Conference (APSEC'04)*, IEEE CS Press, pp. 158-165, 30 Nov-3 Dec 2004

ประวัติผู้เขียน

ชื่อ สกุล	นายชัยวัฒน์ ไพบูลย์พุดพิงศ์		
รหัสประจำตัวนักศึกษา	4910220116		
วุฒิการศึกษา			
วุฒิ	ชื่อสถาบัน	ปีที่สำเร็จการศึกษา	
วิทยาศาสตร์บัณฑิต (เคมี)	มหาวิทยาลัยสงขลานครินทร์	2546	

ทุนการศึกษา (ที่ได้รับในระหว่างการศึกษา)

ทุนมูลนิธิเพื่อการศึกษาคอมพิวเตอร์และการสื่อสารประจำปีการศึกษา 2549

การตีพิมพ์เผยแพร่ผลงาน

ชัยวัฒน์ ไพบูลย์พุดพิงศ์ และ อำนาจ เปาะทอง. 2551. กรอบแนวคิดการเจรจาต่อรองสำหรับการพัฒนาซอฟต์แวร์เชิงบริการโดยการเพิ่มสมรรถนะของแบบจำลองวุฒิภาวะความสามารถแบบบูรณาการ. จากประชุมวิชาการ The 2nd National Conference on Information Technology 2008 (NCIT 2008). โรงแรมแกรนด์เมอร์เคียว ฟอรัจูน. กรุงเทพฯ ประเทศไทย. 6-7 พฤศจิกายน 2551. หน้า 490-499.