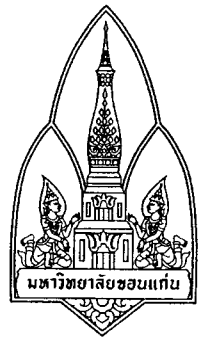


ธุรกิจการท่องเที่ยวแบบครบวงจรโดยวิธีแบบจำลององค์ประกอบตัวบริการเว็บ

ONE STOP SERVICE FOR TOURISM BUSINESS USING WEB SERVICE COMPOSITION MODEL

นายเอกชัย แน่นอุดร

วิทยานิพนธ์ปริญญาวิทยาศาสตรมหาบัณฑิต
มหาวิทยาลัยขอนแก่น
พ.ศ. 2549
ISBN 974-626-846-5

ธุรกิจการท่องเที่ยวแบบครบวงจรโดยวิธีแบบจำลององค์ประกอบตัวบริการเว็บ

นายเอกชัย แน่นอุดร

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ
บัณฑิตวิทยาลัย มหาวิทยาลัยขอนแก่น
พ.ศ. 2549

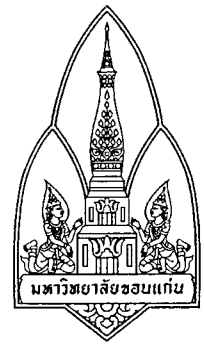
ISBN 974-626-846-5

ONE STOP SERVICE FOR TOURISM BUSINESS USING WEB SERVICE COMPOSITION MODEL

MR. EKKACHAI NAENUDORN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF SCIENCE
IN INFORMATION TECHNOLOGY
GRADUATE SCHOOL KHON KAEN UNIVERSITY
2006

ISBN 974-626-846-5



ใบรับรองวิทยานิพนธ์
มหาวิทยาลัยขอนแก่น
หลักสูตร
วิทยาศาสตรมหาบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ

ชื่อวิทยานิพนธ์: ธุรกิจการท่องเที่ยวแบบครบวงจรโดยวิธีแบบจำลององค์ประกอบตัวบริการเว็บ

ชื่อผู้ทำวิทยานิพนธ์: นายเอกชัย แน่นอุดร

คณะกรรมการสอบวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร.สมจิตร อาจอินทร์ ประธานกรรมการ
ผู้ช่วยศาสตราจารย์ ดร.งามนิจ อาจอินทร์ กรรมการ
รองศาสตราจารย์ วิรัตน์ พงษ์ศิริ กรรมการ

อาจารย์ที่ปรึกษาวิทยานิพนธ์:

งามนิจ อาจอินทร์ อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์ ดร. งามนิจ อาจอินทร์)

(รองศาสตราจารย์ ดร. ลำปาง แม่นมาตย์)
คณบดีบัณฑิตวิทยาลัย

(ศาสตราจารย์ ดร. ละออศรี เสนาะเมือง)
คณบดีคณะวิทยาศาสตร์

ลิขสิทธิ์ของมหาวิทยาลัยขอนแก่น

เอกชัย แน่นอุดร. 2549. **ระบบการท่องเที่ยวแบบครบวงจรโดยวิธีแบบจำลององค์ประกอบตัวบริการเว็บ.** วิทยานิพนธ์ปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย มหาวิทยาลัยขอนแก่น. [ISBN 974-626-846-5]
อาจารย์ที่ปรึกษาวิทยานิพนธ์ : ผู้ช่วยศาสตราจารย์ ดร.งามนิจ อาจอินทร์

บทคัดย่อ

ปัจจุบันแต่ละองค์กรมีการพัฒนาระบบงานด้วยโปรแกรมประยุกต์ที่แตกต่างกัน จึงทำให้การแลกเปลี่ยนข้อมูลกันทำได้ยาก ตัวอย่างเช่น องค์กรที่ให้บริการจองตั๋วเครื่องบินและองค์กรที่ให้บริการโรงแรมต่างก็มีการพัฒนาโปรแกรมประยุกต์ที่เป็นของตนเอง โดยใช้ภาษาในการพัฒนาที่แตกต่างกัน จึงทำให้เกิดปัญหาในการทำงานร่วมกันและการแลกเปลี่ยนข้อมูลระหว่างกัน ทำให้การทำงานประสานกันระหว่างระบบขององค์กรไม่สามารถเกิดขึ้นได้อย่างอัตโนมัติ

ในปัจจุบันเทคโนโลยีตัวบริการเว็บ (Web Services) ได้เข้ามามีบทบาทสำคัญในการช่วยแก้ปัญหาที่เกิดขึ้นได้ สามารถทำให้โปรแกรมประยุกต์ที่แตกต่างกัน สามารถทำงานร่วมกันได้และแลกเปลี่ยนข้อมูลร่วมกันได้ เนื่องจากเทคโนโลยีตัวบริการเว็บมีคุณลักษณะที่มีการใช้ภาษา XML เป็นภาษามาตรฐานในการแลกเปลี่ยนข้อมูลระหว่างกัน การทำธุรกิจระหว่างองค์กรจึงเป็นไปได้ง่ายขึ้น และในการนำตัวบริการเว็บหลายตัวมาประกอบร่วมกันหรือเทคโนโลยีองค์ประกอบตัวบริการเว็บ (Web Services Composition) ทำให้ผู้ให้บริการสามารถทำธุรกิจร่วมกันและแลกเปลี่ยนข้อมูลร่วมกันได้ ทำให้เกิดการสร้างพันธมิตรทางการค้า เป็นการเพิ่มช่องทางการขายทำให้มีโอกาสทางการตลาดที่มากขึ้นและเพิ่มกำไรให้กับผู้ร่วมทำธุรกิจ ทางด้านผู้ใช้บริการได้รับบริการอย่างครบวงจรและรวดเร็วเพราะมีบริการแบบเสร็จสรรพในจุดเดียว

งานวิจัยนี้ได้นำเสนอแบบจำลององค์ประกอบตัวบริการเว็บที่เกี่ยวข้องกับธุรกิจด้านการท่องเที่ยว ซึ่งประกอบด้วยกลุ่มธุรกิจจองตั๋วเครื่องบิน ธุรกิจโรงแรม ธุรกิจรถเช่า และธุรกิจด้านการเงิน โดยที่ตัวบริการเว็บของผู้ให้บริการแต่ละราย สามารถติดต่อสื่อสารรับส่งข้อมูล และทำงานร่วมกันได้อย่างอัตโนมัติ โดยมีระบบสารสนเทศการท่องเที่ยว (Tourism Information Systems : TIS) ทำหน้าที่เป็นตัวกลางในการประสานงานการทำงานร่วมกัน ซึ่งพัฒนาขึ้นจากภาษาที่ใช้สร้างกระบวนการทางธุรกิจคือภาษา BPEL4WS (Business Process Execution Language for Web Services) ทำให้การสื่อสารแลกเปลี่ยนข้อมูลระหว่างตัวบริการเว็บเป็นไปแบบอัตโนมัติ ในการพัฒนาระบบ TIS โดยใช้วิธีองค์ประกอบตัวบริการเว็บใช้เครื่องมือในการพัฒนาคือ Oracle JDeveloper BPEL Designer และสร้าง User Interface เพื่อทดสอบระบบ TIS โดยใช้ภาษา JSP

โดยสรุปผลการวิจัยในครั้งนี้พบว่าแบบจำลององค์ประกอบตัวบริการเว็บนี้ สามารถแก้ปัญหาข้อจำกัดในการทำงานร่วมกัน และแลกเปลี่ยนข้อมูล ระหว่างตัวบริการเว็บ ทำให้การทำธุรกิจระหว่างองค์กรเป็นไปได้ง่ายขึ้น อีกทั้งแบบจำลององค์ประกอบตัวบริการเว็บนี้ยังมีความยืดหยุ่นสามารถรองรับการเพิ่มขึ้นของตัวบริการเว็บของกลุ่มธุรกิจบริการ (Web Services Brokers) หรือการเพิ่มขึ้นของตัวบริการเว็บของผู้ให้บริการ (Web Services Providers) ที่จะเกิดขึ้นในอนาคตได้

Ekkachai Naenudorn. 2006. **One Stop Service for Tourism Business using Web Service Composition Model**. Master of Science Thesis in Information Technology, Graduate School, Khon Kaen University. [ISBN 974-626-846-5]

Thesis Advisor : Asst.Prof. Dr.Ngamnij Arch-int

ABSTRACT

At present, each organization develops its own work system by using different applied programs. Thus it causes difficulties in exchanging data with one another. For example, organizations which offer airfare ticket reservation service and organizations which offer hotel services each develops its own applied programs by using different languages for developing its system. This can cause problems of working together and of data exchange among themselves, and it can cause coordination among systems of these organizations to be unable to emerge automatically.

Currently web service technology has been playing its important role in helping solve problems that occur. Such technology can make different applied programs able to work together and exchange data with one another. Because the web services have a characteristic that users XML as a standard language for exchange information with one another, business performance among organizations is more easily possible. Also, various web services that are composed together or web services composition technology enables service providers to perform their businesses together and exchange information with one another. It can originate creation of trade allies. Channels of sales, marketing opportunities, and profits of business cooperators can increase. For service users, they can receive a complete cycle of services in rapidity because there are total and one-stop services.

This research presented to web service composition model involving tourism business consisting of an airfare ticket reservation business group, a hotel business group, a car rental business group and a financial business group. The web services of each service providers can communicate in receiving and sending information and can work together automatically. The tourism information systems performed their functions as a medium of coordination in working together. These systems were developed from the language used to construct the business process : BPEL4WS (Business Process Execution Language for Web Services). The systems caused the communication and exchanges of information among web services to go on automatically. In developing tourism information systems (TIS) by using the method of web service composition, the instrument used was Oracle JDeveloper BPEL Designer; and User Interface was constructed for testing TIS by using the JSP language.

In conclusion, the results of this research were as follows. This web service composition model could solve problems of limitations of working together and of exchanges of information among web services. This caused business performance among organizations to go on more easily. Also, this web service composition model was flexibility and can underlie increment of web services of web service brokers or increment of web services of web service providers that will occur in the future.

งานวิทยานิพนธ์นี้มอบส่วนดีให้บุพการีและคณาจารย์

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จสมบูรณ์ได้ ด้วยความกรุณาและช่วยเหลืออย่างสูงยิ่งจาก ผู้ช่วยศาสตราจารย์ ดร.งามนิจ อาจอินทร์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร.สมจิตร อาจอินทร์ และรองศาสตราจารย์วิรัตน์ พงษ์ศิริ ที่กรุณาให้คำแนะนำและข้อคิดเห็น ตลอดจนให้คำปรึกษาและช่วยเหลือเป็นอย่างดียิ่ง ผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูง

ขอกราบขอบพระคุณคณาจารย์ในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ ผู้ประสิทธิประสาทวิชาความรู้ ตลอดทั้งเจ้าของตำรา ผลงานและเอกสารวิชาการทุกท่าน ที่ผู้วิจัยได้นำมาศึกษาก่อให้เกิดแนวคิดอันมีคุณค่าต่อวิทยานิพนธ์ฉบับนี้

ขอกราบขอบพระคุณผู้ช่วยศาสตราจารย์ ดร.ปพฤกษ์ อุตสาหะวาณิชกิจ รองคณบดีฝ่ายวิชาการ คณะการบัญชีและการจัดการ มหาวิทยาลัยมหาสารคาม ที่ช่วยผลักดันและสนับสนุนทำให้ได้รับทุนการศึกษาจากมหาวิทยาลัยมหาสารคาม และเป็นผู้ให้โอกาสต่าง ๆ มากมายมาโดยตลอด ขอขอบคุณบัณฑิตวิทยาลัย มหาวิทยาลัยขอนแก่น ที่สนับสนุนทุนอุดหนุนค้นคว้าและวิจัยในการทำวิทยานิพนธ์ในครั้งนี้ และขอกราบขอบพระคุณท่านอาจารย์วีรวัฒน์ มะเสนา ที่เป็นแรงบันดาลใจในการศึกษาที่ดียิ่ง

ขอกราบขอบพระคุณผู้ให้กำลังใจที่สำคัญยิ่งต่อการศึกษาทำให้สำเร็จลุล่วงไปด้วยดี ขอกราบขอบพระคุณ คุณพ่อทองอินทร์ แน่นอุดร คุณแม่เรียม มุลเมืองแสน และน้อง ๆ ทุกคน ที่คอยสนับสนุนในทุก ๆ ด้าน ขอขอบคุณเพื่อน ๆ นักศึกษาเทคโนโลยีสารสนเทศ รุ่นที่ 1 ทุกคน ที่ช่วยเหลือในการศึกษามาด้วยดีตลอด และสุดท้ายขอบคุณนางสาวปิยวรรณ สุวรรณเกษม ที่คอยเป็นกำลังใจในการศึกษาจนประสบผลสำเร็จ

เอกชัย แน่นอุดร

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
คำอุทิศ	ค
กิตติกรรมประกาศ	ง
สารบัญตาราง	ช
สารบัญภาพ	ซ
บทที่ 1 บทนำ	1
1. ความเป็นมาและความสำคัญของปัญหา	1
2. วัตถุประสงค์ของการวิจัย	1
3. ขอบเขตของการวิจัย	2
4. ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	3
1. ทฤษฎีพื้นฐาน	3
2. งานวิจัยที่เกี่ยวข้อง	13
บทที่ 3 วิธีการดำเนินการวิจัย	23
1. ออกแบบสถาปัตยกรรมของธุรกิจการท่องเที่ยวแบบครบวงจรโดยใช้องค์ประกอบตัวบริการเว็บ	23
2. ออกแบบลำดับขั้นตอนการดำเนินงานทางธุรกิจของธุรกิจบริการท่องเที่ยวแบบครบวงจร	24
3. จำลองตัวบริการเว็บที่เกี่ยวข้องกับธุรกิจการท่องเที่ยว	32
4. สร้างแบบจำลององค์ประกอบตัวบริการเว็บตามมาตรฐานของ BPEL4WS	33
5. พัฒนาโปรแกรมประยุกต์เพื่อทดสอบแบบจำลององค์ประกอบตัวบริการเว็บ	38
6. เครื่องมือที่ใช้ในการพัฒนาระบบ	39
บทที่ 4 ผลการวิจัย	40
1. ทดสอบระบบธุรกิจการท่องเที่ยวแบบครบวงจรโดยวิธีแบบจำลององค์ประกอบตัวบริการเว็บผ่านทาง User Interface	40
2. ผลการทดสอบแบบจำลององค์ประกอบตัวบริการเว็บ	45
3. การเปรียบเทียบระหว่างงานวิจัยนี้กับระบบ Thai Tourism C-Commerce	46
4. การเปรียบเทียบแบบจำลององค์ประกอบตัวบริการเว็บกรณีที่ใช้ Web Service Broker และไม่ใช้ Web Service Broker	49
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	51
1. สรุปผลการศึกษาวิจัย	51
2. ข้อเสนอแนะ	51
เอกสารอ้างอิง	52

สารบัญ (ต่อ)

หน้า

ภาคผนวก 54
ภาคผนวก ก ตัวอย่างไฟล์ BPEL ของระบบ TIS 55
ภาคผนวก ข ตัวอย่างไฟล์ JSP สำหรับส่วนติดต่อกับผู้ใช้ 70
ภาคผนวก ค การเผยแพร่ผลงานวิทยานิพนธ์ 78
ประวัติผู้เขียน 89

สารบัญตาราง

		หน้า
ตารางที่ 1	Input / Output of Airline Broker Web Services	27
ตารางที่ 2	Input / Output of Hotel Broker Web Services	28
ตารางที่ 3	Input / Output of Car Rental Web Services	29
ตารางที่ 4	Input / Output of Corporate Expense Services	30
ตารางที่ 5	การเปรียบเทียบประสิทธิภาพระหว่าง WSCM และ TCC	49

สารบัญภาพ

		หน้า
ภาพที่ 1	ตัวอย่าง XML Purchase Order	5
ภาพที่ 2	โครงสร้างของเอกสาร SOAP	6
ภาพที่ 3	ภาพจำลองการทำงานของตัวบริการเว็บที่เกิดจากการทำงานร่วมกันของ SOAP, UDDI และ WSDL	8
ภาพที่ 4	Standards Building Blocks of BPEL	9
ภาพที่ 5	BPEL Depends on WSDL and WSDL Extensions	9
ภาพที่ 6	Declare the Web Services and roles used by the process	10
ภาพที่ 7	Messages sent and received from partners	11
ภาพที่ 8	ตัวอย่างของไฟล์ BPEL	12
ภาพที่ 9	องค์ประกอบที่เกี่ยวข้องกับผู้ขาย	14
ภาพที่ 10	กระบวนการทางธุรกรรมอิเล็กทรอนิกส์สำหรับการจัดซื้อจัดหา	14
ภาพที่ 11	กระบวนการทางธุรกรรมอิเล็กทรอนิกส์สำหรับการขาย	15
ภาพที่ 12	Model of trading alliance, including e-business processes	16
ภาพที่ 13	โครงสร้างพื้นฐานขององค์ประกอบสำหรับ e-marketplace	17
ภาพที่ 14	เค้าเรื่องตัวอย่างแสดงการทำงานของ Travel Agency	18
ภาพที่ 15	สถาปัตยกรรมของระบบการท่องเที่ยวแบบครบวงจร โดยวิธีแบบจำลององค์ประกอบตัวบริการเว็บ	23
ภาพที่ 16	แสดงลำดับขั้นตอนการทำงานของธุรกิจบริการท่องเที่ยวต่าง ๆ	24
ภาพที่ 17	แผนภาพกระบวนการทางธุรกิจของระบบการท่องเที่ยวแบบครบวงจรในระดับบน : Level 0	26
ภาพที่ 18	แผนภาพการทำงานของตัวแทนการบริการจองตั๋วเครื่องบินในระดับล่าง : Level 1	26
ภาพที่ 19	แผนภาพการทำงานของตัวแทนการบริการโรงแรมในระดับล่าง : Level 1	27
ภาพที่ 20	แผนภาพการทำงานของตัวแทนการบริการรถเช่าในระดับล่าง : Level 1	28
ภาพที่ 21	แผนภาพการทำงานของตัวบริการจัดการค่าใช้จ่าย : Level 1	29
ภาพที่ 22	แผนภาพการทำงานของ ThaiAirway Web Service	31
ภาพที่ 23	แผนภาพการทำงานของ Sofitel Hotel Web Service	31
ภาพที่ 24	แผนภาพการทำงานของ Khon Kaen Car Rental web service	32
ภาพที่ 25	สถาปัตยกรรมของเว็บเซอร์วิส	33
ภาพที่ 26	The WSDL for the BPEL process	34
ภาพที่ 27	Basic structure of BPEL process definition document	34
ภาพที่ 28	แผนภาพการทำงานของ User Interface	38
ภาพที่ 29	หน้าจอของโปรแกรม Oracle JDeveloper BPEL Designer	40
ภาพที่ 30	Select Travel Services : User Interface	41

สารบัญภาพ (ต่อ)

		หน้า
ภาพที่ 31	Airline Ticket Reservation : User Interface	41
ภาพที่ 32	Select Flight : User Interface	42
ภาพที่ 33	Result of Airline Ticket Reservation : User Interface	42
ภาพที่ 34	Select Hotel : User Interface	43
ภาพที่ 35	Result of Hotel Reservation : User Interface	43
ภาพที่ 36	Select Car Rental : User Interface	44
ภาพที่ 37	Result of Car Rental Reservation : User Interface	44
ภาพที่ 38	แสดงรายการทั้งหมดเพื่อให้ลูกค้ายืนยันการใช้บริการ	45
ภาพที่ 39	ภาพแสดงกรอบของความร่วมมือตามหลักการของ C-Commerce	46
ภาพที่ 40	ภาพแสดงการเชื่อมต่อของธุรกิจท่องเที่ยวต่าง ๆ เข้าเครือข่าย C-Commerce	47
ภาพที่ 41	ภาพแสดงแบบจำลององค์ประกอบตัวบริการเว็บกรณีที่ใช้ Web Service Broker	50
ภาพที่ 42	ภาพแสดงแบบจำลององค์ประกอบตัวบริการเว็บกรณีที่ไม่ใช้ Web Service Broker	50

บทที่ 1
บทนำ

1. ความเป็นมาและความสำคัญของปัญหา

เนื่องจากโลกของธุรกิจในปัจจุบันมีการแข่งขันกันสูงมาก แต่ละองค์กรต้องหากลยุทธ์ที่จะทำให้ธุรกิจของตนเองสามารถแข่งขันกับคู่แข่งได้ การค้าขายในระบบดั้งเดิมเพียงอย่างเดียวนั้น อาจทำให้ธุรกิจก้าวไปไม่ทันคู่แข่ง ดังนั้นองค์กรธุรกิจควรหาแนวทางใหม่ๆ ในการดำเนินธุรกิจ เช่น หาตลาดใหม่ให้กับสินค้า หาผู้ร่วมทำธุรกิจ (Partner) หรือปรับปรุงแนวทางการดำเนินธุรกิจแบบเดิมให้มีประสิทธิภาพเป็นที่ยอมรับของผู้ร่วมทำธุรกิจทั่วไป เทคโนโลยีต่างๆ จะเป็นปัจจัยสำคัญที่ทำให้เกิดความแตกต่างระหว่างธุรกิจ หากธุรกิจใดนำเอาเทคโนโลยีที่มีอยู่มาใช้ได้อย่างเหมาะสมก็จะทำให้ธุรกิจได้เปรียบคู่แข่งขัน

การพาณิชย์อิเล็กทรอนิกส์ (E-Commerce) เป็นแนวทางหนึ่งที่องค์กรธุรกิจส่วนใหญ่หันมาใช้เพื่อเพิ่มกำไรให้กับธุรกิจ บางธุรกิจตั้งขึ้นมาใหม่เพื่อทำการค้าอิเล็กทรอนิกส์เพียงอย่างเดียว การทำการค้าพาณิชย์อิเล็กทรอนิกส์ที่เห็นกันในปัจจุบัน มีลักษณะทั้งแบบ B2B (Business to Business) คือการทำธุรกิจการค้าระหว่างองค์กรธุรกิจด้วยกัน และแบบ B2C (Business to Customer) คือการขายสินค้าจากผู้ขายที่เป็นองค์กรธุรกิจไปยังผู้ซื้อซึ่งเป็นบุคคลทั่วไป

ปัจจุบันแต่ละองค์กรมีการพัฒนาระบบงานโดยสร้างเป็นโปรแกรมประยุกต์ที่แตกต่างกัน ดังนั้นจึงทำให้การแลกเปลี่ยนข้อมูลกันทำได้ยาก ตัวอย่างเช่น องค์กรที่ให้บริการจองตั๋วเครื่องบินที่มีการพัฒนาโปรแกรมประยุกต์ที่เป็นของตนเอง ในขณะที่องค์กรที่ให้บริการโรงแรมก็มีการพัฒนาโปรแกรมประยุกต์ที่เป็นของตนเอง และอาจมีการใช้ภาษาในการพัฒนาที่แตกต่างกัน ซึ่งจะทำให้เกิดปัญหาในการทำงานร่วมกันและการแลกเปลี่ยนข้อมูลระหว่างกัน ทำให้การทำงานประสานกันระหว่างระบบขององค์กรไม่สามารถเกิดขึ้นได้อย่างอัตโนมัติ

ในปัจจุบันเทคโนโลยีตัวบริการเว็บหรือ Web Services [1] ได้เข้ามามีบทบาทและช่วยแก้ปัญหาที่เกิดขึ้นได้ โดยสามารถทำให้โปรแกรมประยุกต์ที่แต่ละองค์กรพัฒนาขึ้น สามารถทำงานร่วมกันได้ (Interoperability) และใช้งานข้อมูลร่วมกันได้ (Sharebility) เนื่องจากคุณลักษณะของตัวบริการเว็บที่มีการใช้ภาษา XML [2] เป็นภาษามาตรฐานในการแลกเปลี่ยนข้อมูลระหว่างกัน ทำให้การทำธุรกิจระหว่างองค์กรเป็นไปได้ง่ายขึ้น

การสร้างองค์ประกอบตัวบริการเว็บ (Web Services Composition) จะทำให้ผู้ให้บริการสามารถทำธุรกิจร่วมกันและแลกเปลี่ยนข้อมูลร่วมกันได้ ทำให้เกิดการสร้างพันธมิตรทางการค้า เป็นการเพิ่มช่องทางการขายทำให้มีโอกาสทางการตลาดที่มากขึ้นและเพิ่มกำไรให้กับผู้ร่วมทำธุรกิจ (Partner) ทางด้านลูกค้าทำให้ประหยัดเวลาเพราะมีบริการอย่างครบวงจรและเสร็จสรรพในจุดเดียว (One Stop Service)

2. วัตถุประสงค์ของการวิจัย

2.1 ออกแบบและพัฒนาสถาปัตยกรรมของระบบธุรกิจการท่องเที่ยวแบบครบวงจร โดยใช้องค์ประกอบตัวบริการเว็บ เพื่อทำให้ธุรกิจต่างๆ สามารถติดต่อสื่อสารและทำงานร่วมกันได้อย่างอัตโนมัติ และเป็นการอำนวยความสะดวกให้กับลูกค้าที่ใช้บริการ

2.2 ออกแบบและพัฒนาแบบจำลององค์ประกอบตัวบริการเว็บ เพื่อทำให้ตัวบริการเว็บของบริการธุรกิจท่องเที่ยวต่าง ๆ สามารถติดต่อสื่อสารกันและแลกเปลี่ยนข้อมูลร่วมกันได้

2.3 ออกแบบและพัฒนาโปรแกรมประยุกต์ เพื่อสร้างระบบธุรกิจการท่องเที่ยวแบบครบวงจรที่สอดคล้องกับสถาปัตยกรรมของระบบที่ออกแบบไว้ เพื่อเป็นการทดสอบการนำไปใช้งานได้จริงในทางปฏิบัติ

3. ขอบเขตของการวิจัย

3.1 ศึกษาการสร้างกระบวนการทางธุรกิจ (Business Process) และการทำงานร่วมกันระหว่างตัวบริการเว็บ

3.2 ศึกษาและเปรียบเทียบภาษาที่ใช้สร้างองค์ประกอบตัวบริการเว็บระหว่างภาษา BPEL และภาษาอื่น ๆ

3.3 จำลองตัวเว็บบริการเว็บที่เกี่ยวข้องกับธุรกิจการท่องเที่ยวเพื่อประกอบการสร้างแบบจำลององค์ประกอบตัวบริการเว็บในธุรกิจที่ให้บริการการท่องเที่ยว

3.4 สร้างกระบวนการทางธุรกิจโดยใช้องค์ประกอบตัวบริการเว็บที่ทำให้ตัวบริการเว็บของธุรกิจการท่องเที่ยวต่าง ๆ ทำงานร่วมกันได้อย่างอัตโนมัติ และบริการลูกค้าแบบ One Stop Service

3.5 ทดสอบการทำงานของแบบจำลององค์ประกอบตัวบริการเว็บ

3.6 ศึกษาปัญหาที่เกิดขึ้นในระบบเพื่อวิเคราะห์และเสนอแนะแนวทางการแก้ไขปัญหาในการทำงานร่วมกันระหว่างตัวบริการเว็บในระบบธุรกิจการท่องเที่ยวแบบครบวงจร

4. ประโยชน์ที่คาดว่าจะได้รับ

4.1 ได้โปรแกรมประยุกต์ที่พัฒนาขึ้นเพื่อเป็นตัวกลางในการขอใช้บริการต่าง ๆ จากตัวบริการเว็บของแต่ละธุรกิจการท่องเที่ยวที่เกี่ยวข้อง

4.2 ทำให้เกิดการสร้างพันธมิตรทางการค้า เป็นการเพิ่มช่องทางการขายทำให้มีโอกาสทางการตลาดที่มากขึ้นและเพิ่มกำไรให้กับผู้ร่วมทำธุรกิจ (Partner)

4.3 ลูกค้าหรือผู้ใช้บริการ สะดวกและประหยัดเวลาในการใช้บริการ เพราะมีบริการเสร็จสรรพในจุดเดียว (One Stop Service)

4.4 ไม่เสียเวลาพัฒนาโปรแกรมขึ้นมาใหม่ โดยเรียกใช้ตัวบริการเว็บที่มีอยู่แล้ว

4.5 ลดเวลาและค่าใช้จ่าย ทั้งในส่วนของลูกค้าและผู้ให้บริการ

4.6 สามารถนำไปประยุกต์ใช้งานกับระบบงานอื่น ๆ ที่ต่างแพลตฟอร์มกันให้สามารถติดต่อสื่อสารกันและแลกเปลี่ยนข้อมูลร่วมกันได้

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

1. ทฤษฎีพื้นฐาน

1.1 Web Service [1]

1.1.1 ตัวบริการเว็บ (Web Service) เป็นเทคโนโลยีที่ทำให้โปรแกรมประยุกต์ต่าง ๆ ที่พัฒนาขึ้นโดยมีพื้นฐานทางด้านภาษา และสถาปัตยกรรมที่แตกต่างกัน สามารถสื่อสารแลกเปลี่ยนข้อมูลกันได้โดยใช้ XML [2] เป็นภาษากลางในการสื่อสารแลกเปลี่ยนข้อมูล และทำงานร่วมกันอยู่บนมาตรฐานอื่นอีกเช่น SOAP [3], WSDL [4] และ UDDI [5] ซึ่งจะกล่าวถึงในหัวข้อถัดไป และได้สรุปความหมายของตัวบริการเว็บเป็นข้อ ๆ ดังนี้

(1) เป็นเทคโนโลยีที่สามารถใช้งานร่วมกันได้ทุกแพลตฟอร์ม

(2) โปรแกรมประยุกต์ต่าง ๆสามารถติดต่อสื่อสารกันโดยใช้ XML เป็นสื่อกลาง

(3) ถูกเรียกใช้งานจาก Application อื่น ๆในรูปแบบ RPC (Remote Procedure Call)

(4) การให้บริการจะมีเอกสารที่อธิบายคุณสมบัติของบริการกำกับไว้ (WSDL : Web Services Description Language)

(5) เป็นมาตรฐานเปิดที่ทำให้เราสามารถ "ประกาศ" (Publish) ว่ามี "บริการ" ใดบ้าง และ "Web Application" อื่น สามารถเข้าไปสืบค้น และ เรียกใช้บริการจาก services นั้นได้ (UDDI : Universal Description, Discovery and Integration)

(6) ใช้มาตรฐาน SOAP ในการสื่อสาร ซึ่งสามารถทำงานร่วมกับโพรโทคอลอื่น เช่น HTTP, SMTP

1.1.2 **ประโยชน์ของตัวบริการเว็บ**

(1) การดำเนินธุรกิจการค้าและบริการระหว่างองค์กรสามารถเป็นไปได้ แบบอัตโนมัติ โดยการแลกเปลี่ยนข้อมูลผ่านทางตัวบริการเว็บ

(2) พันธมิตรทางการค้าสามารถเกิดขึ้นได้ตลอดเวลา โดยการค้นหาจาก UDDI

(3) ตัวบริการเว็บใช้ XML สำหรับอธิบายข้อมูลโดย XML เป็นข้อมูลในรูปของ text file ที่คนอ่านกันเข้าใจและไม่ขึ้นกับ platform หรือระบบปฏิบัติการ และด้วยความข้อมูลที่เป็น text file ธรรมดาทำให้มีความยืดหยุ่นในการใช้งาน รวมทั้งการเขียนโปรแกรมที่เกี่ยวข้องก็จะทำได้ง่าย

(4) ตัวบริการเว็บช่วยให้การ Integrate หรืออย่างน้อยก็การแลกเปลี่ยนข้อมูลทำได้ง่ายขึ้น เนื่องจากการเขียนโปรแกรมเพื่อทำงานกับตัวบริการเว็บ สามารถเขียนด้วยภาษาอะไรก็ได้ และไม่ขึ้นกับ Hardware และระบบปฏิบัติการ

(5) ตัวบริการเว็บสามารถใช้เป็นส่วนหนึ่งในการดำเนินธุรกิจตามเฟรมเวิร์คของ ebXML

(6) ง่ายต่อการนำไปใช้งานเนื่องจากในปัจจุบันมีเครื่องมือมากมายที่ใช้ช่วยเหลือในการพัฒนาตัวบริการเว็บ

(7) ลดต้นทุนในการพัฒนาระบบบางอย่างที่ไม่จำเป็นโดยขอบริการจากตัวบริการเว็บของพันธมิตรทางการค้า

1.1.3 เครื่องมือที่ใช้ในการพัฒนาตัวบริการเว็บ

ตัวบริการเว็บเป็นมาตรฐานที่ง่ายต่อการพัฒนาเนื่องจากอยู่ในรูปแบบของข้อมูลตัวอักษร มีชุดเครื่องมือที่สนับสนุนการสร้าง Web Services หลายตัว เช่น

(1) ชุดเครื่องมือที่สนับสนุนโดย Microsoft ตามแพลตฟอร์มของ Microsoft .NET Framework เช่น Microsoft SOAP Toolkit 3.0 [6]

(2) ชุดเครื่องมือที่สนับสนุนโดย Sun Microsystem ตามแพลตฟอร์มของ Sun ONE (Sun Open Net Environment) [7]

(3) ชุดเครื่องมือที่สนับสนุนโดย IBM เช่น Web Services Toolkit [8], WebSphere Studio Application Developer [9]

(4) เครื่องมืออื่น ๆที่สนับสนุน SOAP, XML ทั้งที่เป็น Commercial Product และ Open Source

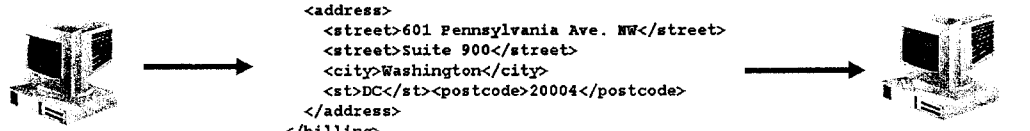
1.2 มาตรฐานที่เกี่ยวข้องกับตัวบริการเว็บ

ตัวบริการเว็บ (Web Service) เป็นเทคโนโลยีที่สามารถทำให้ซอฟต์แวร์แลกเปลี่ยนข้อมูลร่วมกันได้ ทำให้เกิดผลลัพธ์ที่สามารถทำงานร่วมกันระหว่างแพลตฟอร์มที่ต่างกัน เพื่อเป็นประโยชน์ต่อผู้ใช้ มาตรฐานเทคโนโลยีหลักที่เกี่ยวข้องกับตัวบริการเว็บ คือ XML SOAP WSDL และ UDDI ซึ่งแต่ละมาตรฐาน สามารถอธิบายได้ดังนี้

1.2.1 XML (Extensible Markup Language)

เป็นภาษามาตรฐานที่ถูกออกแบบมาเพื่อให้เป็นภาษากลางสำหรับแลกเปลี่ยนข้อมูลระหว่างกัน มีลักษณะข้อมูลเป็น text-based โดยเป็นภาษาที่ไม่ติดอยู่กับแพลตฟอร์มหรือโปรแกรมเฉพาะใด ๆ (Platform and Application Independent) ดังนั้นระบบที่อยู่ต่างแพลตฟอร์มกัน ใช้ฐานข้อมูลต่างกัน หรือมีสถาปัตยกรรมที่ต่างกันก็สามารถแลกเปลี่ยนข้อมูลกันได้ โดยผู้ใช้เองสามารถกำหนดแท็ก (Tag) ที่ใช้สื่อความหมายของเนื้อหา กำหนดโครงสร้างของเอกสาร และยังสามารถนำ xml นั้นไปประมวลผลต่อในแอพพลิเคชั่นอื่นได้

XML จะอาศัยโพรโทคอลที่ชื่อว่า SOAP (Simple Object Access Protocol) ซึ่งเปรียบเสมือนกติกา หรือข้อตกลงในการสื่อสารบนเครือข่ายอินเทอร์เน็ต




```
<?xml version="1.0" ?>
<?xml:stylesheet?>
<purchase-order>
  <header>
    <po-number>1234</po-number>
    <date>1999-02-08</date><time>14:05</time>
  </header>
  <billing>
    <company>XMLSolutions</company>
    <address>
      <street>601 Pennsylvania Ave. NW</street>
      <street>Suite 900</street>
      <city>Washington</city>
      <st>DC</st><postcode>20004</postcode>
    </address>
  </billing>
  <order items="1">
    <item>
      <reference>097251</reference>
      <description>Widgets</description>
      <quantity>4</quantity>
      <unit-price>11.99</unit-price>
      <price>47.46</price>
    </item>
    <tax type="sales">
      <tax-unit>VA</tax-unit>
      <calculation>0.045</calculation>
      <amount>2.16</amount>
    </tax>
    ...
  </order>
</purchase-order>
```

ภาพที่ 1 ตัวอย่าง XML Purchase Order

1.2.2 SOAP (Simple Object Access Protocol)

เนื่องจากจุดประสงค์หลักของการใช้งานตัวบริการเว็บคือ ต้องการให้โปรแกรมประยุกต์ติดต่อสื่อสารกันได้ผ่านทางระบบเครือข่าย ซึ่งเทคโนโลยีที่มีอยู่ในปัจจุบันที่ใช้มีการสื่อสารระหว่างออปเจ็กต์ในระยะไกล (Remote Procedure Calls : RPC) เช่น DCOM, EJB หรือ CORBA ไม่ได้ถูกออกแบบมาใช้สำหรับโพรโทคอล HTTP

เทคนิค RPC ของเทคโนโลยีที่กล่าวข้างต้นนั้นมีปัญหาในด้านการนำมาใช้งานในแง่ของความเข้ากันได้ของการเรียกใช้งานข้ามเทคโนโลยี เนื่องจากเป็นเทคโนโลยีเฉพาะของแต่ละค่าย (ยกเว้น CORBA) ทำให้ผู้พัฒนาระบบต้องพัฒนาโปรแกรมที่มีความซับซ้อน และยังมีปัญหาในส่วนของ Firewalls และ Proxy เซิร์ฟเวอร์ เนื่องจากโดยปกติเซิร์ฟเวอร์จะปิดการสื่อสารที่ไม่ใช่โพรโทคอล HTTP เพื่อความปลอดภัยของระบบที่มีการติดต่อสื่อสารกับภายนอก

ดังนั้นทางเลือกของการสื่อสารที่จะนำมาใช้ในการทำตัวบริการเว็บ คือทำงานอยู่บนโพรโทคอล HTTP และเป็นมาตรฐานเปิดที่ทำให้สามารถติดต่อสื่อสารกับเครื่องคอมพิวเตอร์ที่มีความแตกต่างกันทั้งระบบปฏิบัติการ เทคโนโลยี รวมไปถึงภาษาที่ใช้ในการพัฒนาด้วย

SOAP เป็นกลไกการแลกเปลี่ยนข้อมูลระหว่างกันแบบกระจายจากศูนย์กลาง โดยใช้โครงสร้างของ XML เป็นพื้นฐาน สามารถนำไปใช้กับระบบได้หลากหลายตั้งแต่ระบบข้อความ (Messaging System) จนถึง RPC

SOAP ประกอบด้วย 3 ส่วน คือ

(1) SOAP Envelop สำหรับใช้ในการระบุสิ่งที่อยู่ในเอกสาร การจัดการเอกสาร และบอกความจำเป็นในการใช้งาน

(2) SOAP Encoding Rule สำหรับกำหนดกลไกที่ใช้ในการแลกเปลี่ยนข้อมูล

(3) SOAP RPC Representation สำหรับนิยามรูปแบบ RPC และ response

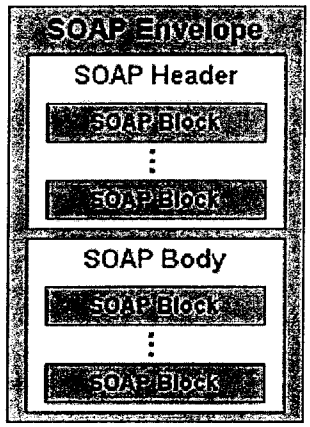
โครงสร้างของ SOAP

เอกสาร SOAP นั้นมีโครงสร้างในรูปแบบ XML ซึ่งเราสามารถแบ่งเป็นส่วนของเอกสารได้เป็น 3 ส่วนหลักดังนี้คือ

(1) SOAP envelop เนื้อหาสาระ (Content) ของเอกสารทั้งหมด

(2) SOAP header ส่วนเพิ่มเติมของเอกสาร SOAP ซึ่งจะมีก็ได้ หรือไม่มีก็ได้

(3) SOAP body ส่วนที่ใช้ในการเรียกใช้งานเซอร์วิส และผลลัพธ์ที่ได้



ภาพที่ 2 โครงสร้างของเอกสาร SOAP

1.2.3 WSDL (Web Services Description Language)

คิดค้นโดยบริษัท IBM และ Microsoft เป็นภาษาที่ใช้อธิบายคุณลักษณะการให้บริการของตัวบริการเว็บ และวิธีการติดต่อขอรับบริการจากตัวบริการเว็บ เช่น ชื่อตัวบริการเว็บ ชื่อเมทธอดของ COM Component ที่เปิดให้บริการ พารามิเตอร์ที่ส่งไปยังเมทธอดชนิดข้อมูลของพารามิเตอร์ เป็นต้น โดยรายละเอียดเหล่านี้จะเป็นไปตามไวยากรณ์ของภาษา XML (Extensible Markup Language)

SOAP UDDI และ WSDL เป็นแกนหลักสำคัญของตัวบริการเว็บ สิ่งสำคัญในการเรียกใช้งานตัวบริการเว็บ คือต้องรู้ที่อยู่จาก UDDI รู้คุณลักษณะของตัวบริการเว็บนั้น ๆจากเมทธอดของตัวบริการเว็บและพารามิเตอร์ต่าง ๆที่จำเป็นจากเอกสาร WSDL และร้องขอบริการ รอผลลัพธ์จากผู้ให้บริการด้วยโพรโทคอล SOAP

ความสัมพันธ์ระหว่าง 3 ส่วนนี้ สามารถแสดงได้ดังนี้

(1) SOAP เป็นโพรโทคอลสื่อสารที่อาศัยไวยากรณ์ของภาษา XML และทำงานกับโพรโทคอลอื่น ๆได้หลายชนิด เนื่องจากใช้ไวยากรณ์ XML จึงทำให้ทำงานได้ทุกแพลตฟอร์ม ดังนั้นจึงสามารถเรียกใช้คอมโพเนนต์ข้ามแพลตฟอร์มได้

(2) WSDL เป็นเอกสารที่ใช้อธิบายคุณลักษณะของตัวบริการเว็บ โดยใช้ไวยากรณ์ของภาษา XML

(3) UDDI เปรียบเสมือนฐานข้อมูลที่เก็บรายละเอียดของตัวบริการเว็บไว้ เพื่อให้ผู้ใช้บริการค้นหาบริการ ในกรณีของผู้ให้บริการต้องนำข้อมูลเกี่ยวกับตัวบริการเว็บของตนไปเก็บไว้ใน UDDI

โครงสร้างของ WDSL

เอกสาร WSDL นั้นแบ่งออกเป็น 2 ส่วนใหญ่ ๆ คือ

(1) Abstract Definitions Group

ทำหน้าที่กำหนด message SOAP ที่ไม่ขึ้นกับแพลตฟอร์มหรือรูปแบบภาษา กล่าวคือ ไม่บรรจุส่วนประกอบที่มีเฉพาะในเครื่องบางเครื่อง หรือภาษาบางภาษา โดยที่ส่วนประกอบของ abstract นั้น จะประกอบไปด้วย type message และ port type

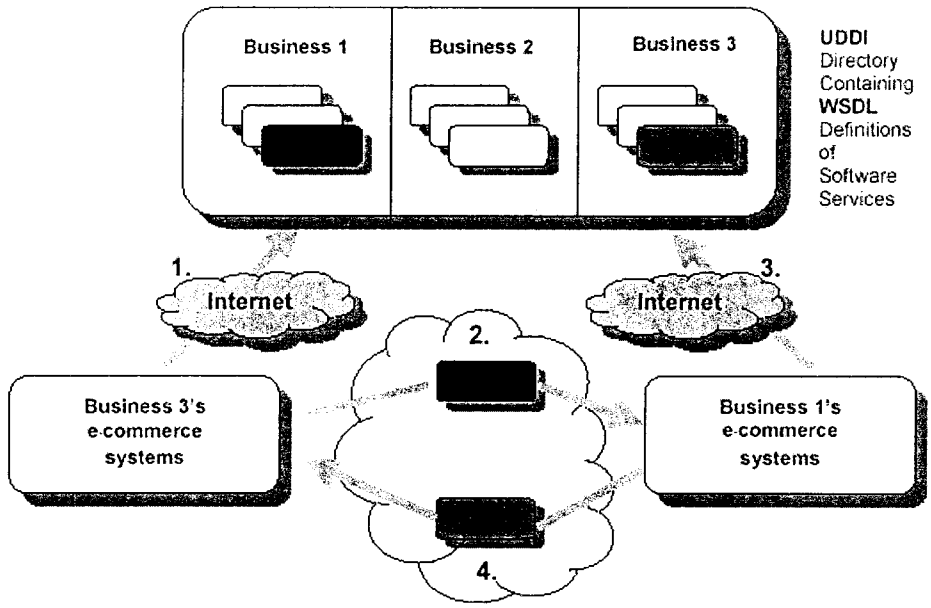
(2) Concrete Descripions Group

ทำหน้าที่ตรงกันข้ามกับส่วนของ abstract Definitions คือ เป็นข้อมูลที่ขึ้นอยู่กับเว็บไซต์ เครื่องและภาษา โดยประกอบด้วย binding และ service

1.2.4 UDDI (Universal Description, Discovery and Integration)

เป็นมาตรฐานในการค้นหาบริการต่าง ๆของ Web Services ทำหน้าที่เป็นตัวกลางเพื่อให้ Provider มาลงทะเบียนไว้ โดยจะมีไฟล์ WSDL สำหรับบอกรายละเอียดของการบริการ เพื่อให้ Requestor สามารถค้นหาและเรียกใช้การบริการของ Web service นั้น ๆได้ โดยดูรายละเอียดของการบริการและวิธีการเรียกใช้จากไฟล์ WSDL ซึ่งเปรียบได้กับสมุดโทรศัพท์นั่นเอง

UDDI เป็นมาตรฐานที่จัดตั้งขึ้นโดย IBM, Microsoft และ Ariba ปัจจุบันมีบริษัทที่ร่วมกันกำหนดมาตรฐานของ UDDI มากกว่า 70 บริษัท ซึ่งมาตรฐานของ UDDI ถูกกำหนดให้เป็นมาตรฐานสำหรับ B2B interoperability



ภาพที่ 3 ภาพจำลองการทำงานของตัวบริการเว็บ ที่เกิดจากการทำงานร่วมกันของ SOAP, UDDI และ WSDL [1]

ภาพจำลองการทำงานร่วมกันของ SOAP, UDDI และ WSDL จะแสดงได้ดังภาพที่ 3 ดังรายละเอียดต่อไปนี้

(1) Business 3 ค้นหาการให้บริการของ Business 1 ใน UDDI ซึ่งรายละเอียดจะอยู่ใน WSDL ผ่านอินเทอร์เฟสของ Business 1

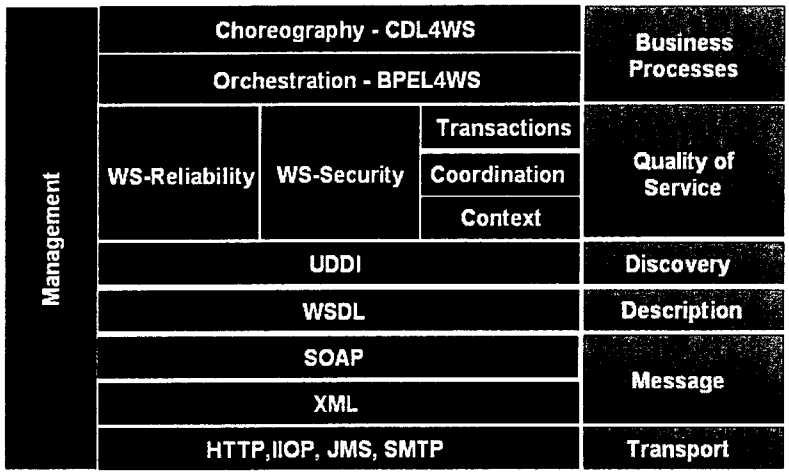
(2) ติดต่อผ่านอินเทอร์เฟสของแอพพลิเคชั่นที่ให้บริการ (ด้วย SOAP message) ผ่านทางอินเทอร์เน็ต

(3) และ 4. Business 1 รวมหรือ Integration กับ Business 3 ด้วยวิธีเดียวกัน

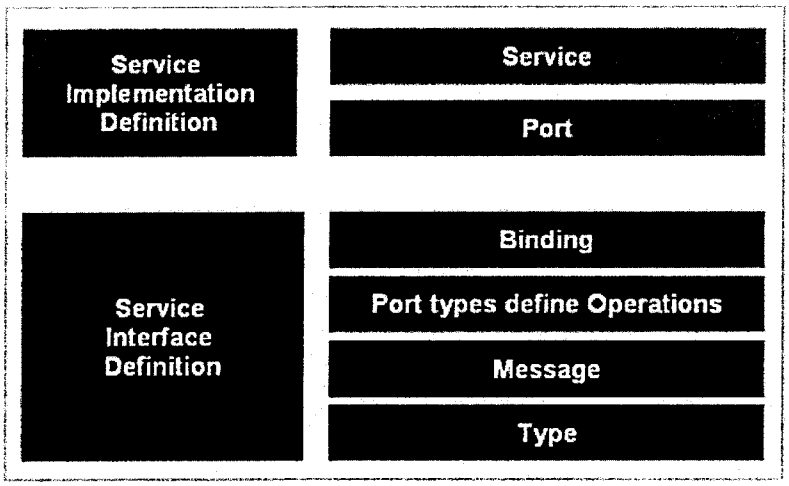
1.2.5 BPEL4WS (Business Process Execution Language for Web Services) [10]

BPEL4WS คือมาตรฐานที่ใช้กำหนดถึงการทำงานร่วมกันของตัวบริการเว็บ โดยจะกำหนดถึงวิธีการ และลำดับขั้นตอนของการดำเนินการร่วมกัน เกิดขึ้นจากความพยายามที่จะสร้างมาตรฐานร่วมกันระหว่าง มาตรฐาน WSFL [11] ของ IBM และมาตรฐาน XLANG [12] ของ Microsoft เพื่อให้เกิดมาตรฐานที่ดีที่สุดโดยหยิบยกเอาข้อดีของทั้งสองมาตรฐานมารวมกัน

BPEL4WS เป็น Top-level ของ WSDL ตามระดับชั้นมาตรฐานดังภาพที่ 4 ใช้รูปแบบไวยากรณ์ของภาษา XML ในการที่จะกำหนดถึงลำดับการทำงานในแต่ละ process ของตัวบริการเว็บ จะดูได้จากไฟล์ WSDL ของแต่ละตัวบริการเว็บ ที่ได้ประกาศไว้บน UDDI ว่าบริการนี้จะต้องส่งค่าอะไรไปบ้าง มาสร้างเป็นกระบวนการทางธุรกิจ (Business Process) ซึ่งจะได้เป็นนามสกุล .bpel (ดังภาพที่ 5) ส่วนเครื่องมือที่ใช้ในการพัฒนาอย่างเช่น WebSphere Studio Application Developer ของ IBM หรือ Oracle JDeveloper ของ Oracle



ภาพที่ 4 Standards Building Blocks of BPEL [10]



ภาพที่ 5 BPEL Depends on WSDL and WSDL Extensions [10]

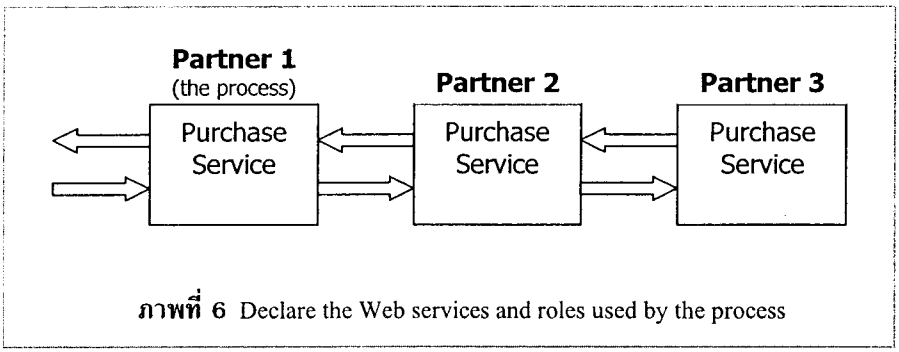
(1) ตัวอย่างโครงสร้างของไฟล์ BPEL (BPEL Scenario Structure)

```
<process>
  <!– Definition and roles of process participants -->
        <partnerLinks> ... </partnerLinks>
  <!- Data/state used within the process -->
  <variables> ... </variables>
  <!- Properties that enable conversations -->
  <correlationSets> ... </correlationSets>
  <!- Exception handling -->
  <faultHandlers> ... </faultHandlers>
        <!- Error recovery – undoing actions  -->
  <compensationHandlers> ... </compensationHandlers>
  <!- Concurrent events with process itself  -->
  <eventHandlers> ... </eventHandlers>
  <!- Business process flow  -->
  (activities)*
</process>
```

(2) กลุ่ม Element Activities ของไฟล์ BPEL แบ่งออกเป็น 2 กลุ่ม คือ

Basic Activities	**Structured Activities**
• <invoke>	• <sequence>
• <receive>	• <switch>
• <assign>	• <pick>
• <reply>	• <flow>
• <throw>	• <link>
• <terminate>	• <while>
• <wait>	• <scope>

(3) Partners Element จะระบุถึง service ของผู้ร่วมทำธุรกิจทั้งหมดว่ามี service ใดบ้าง และมี Attributes กำหนดถึง Service Link Type ว่าคืออะไรเพื่อโยงไปถึง WSDL และบอกถึงบทบาทหน้าที่ว่าทำอะไร ดังตัวอย่างนี้



ภาพที่ 6 Declare the Web services and roles used by the process

- ตัวอย่างไฟล์ BPEL :

```
<partnerLinks>
 <partnerLink name="customer" serviceLinkType="lns:purchasePLT"
   myRole="purchaseService"/>
 <partnerLink name="inventoryChecker" serviceLinkType="lns:inventoryPLT"
   myRole="inventoryRequestor" partnerRole="inventoryService"/>
 <partnerLink name="creditChecker" serviceLinkType="lns:creditPLT"
 myRole="creditRequestor" partnerRole="creditService"/>
</partnerLinks>
```

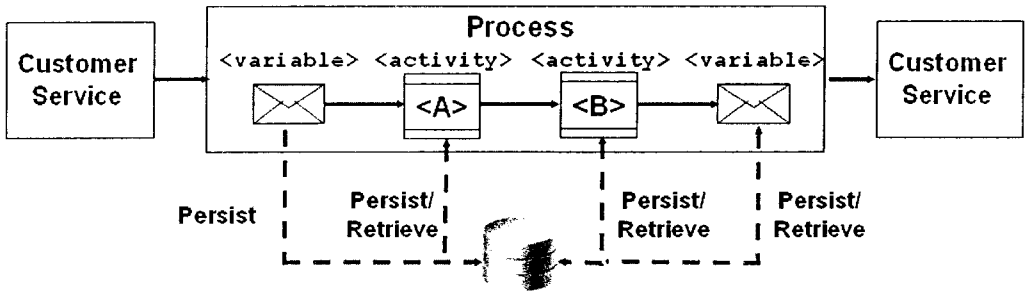
- Purchase Process WSDL:

```
<plt:partnerLinkType name="purchasePLT">
 <plt:role name="purchaseService">
  <plt:portType name="tns:purchasePT"/>
 </plt:role>
</plt:partnerLinkType>
```

- Purchase Process PortType:

```
<portType name="purchasePT">
 <operation name="sendPurchase">
 </operation>
</portType>
```

(4) Variables Element จะกำหนดถึง messages ที่ใช้รับส่งถึงกันระหว่างตัวบริการเว็บ



ภาพที่ 7 Messages sent and received from partners

- ตัวอย่างไฟล์ BPEL :

```
<variables>
 <variable name="PO" messageType="lns:POMessage"/>
 <variable name="Invoice" messageType="lns:InvMessage"/>
 <variable name="POFault" messageType="lns:orderFaultType"/>
</variables>
```

- Purchase Process WSDL:

```
<message name="POMessage">
  <part name="customerInfo" type="sns:customerInfo"/>
  <part name="purchaseOrder" type="sns:purchaseOrder"/>
</message>
<message name="InvMessage">
  <part name="IVC" type="sns:Invoice"/>
</message>
<message name="orderFaultType">
 <part name="problemInfo" type="xsd:string"/>
</message>
```

(5) การจัดการโยกย้ายข้อมูลระหว่าง variables สามารถใช้ <assign> และ <copy> ซึ่ง element <copy> สนับสนุนการสืบค้นข้อมูลแบบ XPath ด้วย ตัวอย่างเช่น

```
<assign>
 <copy>
  <from variable="PO" part="customerInfo"/>
  <to variable="creditRequest" part="customerInfo"/>
 </copy>
</assign>
```

(6) Activities

Basic Activities เช่น

- invoke ส่ง message เพื่อเรียกใช้บริการจาก partner ใด แบ่งได้เป็น 2 ประเภท คือ Synchronous และ Asynchronous
- Receive รอรับ messages จาก partner
- Reply ตอบกลับ message
- Assign จัดการ copy ข้อมูลจากที่หนึ่งไปอีกที่หนึ่ง
- throw บอกถึงเมื่อมีข้อผิดพลาดเกิดขึ้น

- Wait หยุดรอการประมวลผล
- Empty กำหนดเป็นค่าว่าง
- Terminate สิ้นสุดการประมวลผลของ activities นั้น ๆ

Structured Activities เช่น

- Sequence จัดการกับ Activities ต่าง ๆ ให้ประมวลผลตามลำดับ
- Flow จัดการกับ Activities ต่าง ๆ ให้ประมวลผลไปพร้อม ๆกัน
- While กำหนดการวนซ้ำ
- Switch การควบคุม logic ของ sequence ย่อย

(7) ตัวอย่างของ BPEL Code ต่อไปนี้ ประกอบด้วย 3 บริการ โดย Process แรกคือ ServiceA เรียกใช้แบบ Synchronous ส่วน ServiceB และ ServiceC เรียกใช้แบบ Asynchronous ประมวลผลแบบขนาน (flow) โดยรับ Output ของ ServiceA มาเป็น input แล้วทำการประมวลผล สุดท้ายจึงส่งผลลัพธ์กลับมาที่ Client

```
<process name="test">
    <partnerLinks>
        <partnerLink name="client"/>
        <partnerLink name="serviceA"/>
        <partnerLink name="serviceB"/>
        <partnerLink name="serviceC"/>
    </partnerLinks>
    <variables>
        <variable name="processInput"/>
        <variable name="AInput"/>
        <variable name="AOutput"/>
        <variable name="BCInput"/>
        <variable name="BOutput"/>
        <variable name="COutput"/>
        <variable name="processOutput"/>
        <variable name="AError"/>
    </variables>
    <sequence>
        <receive name="receiveInput" variable="input"/>
        <assign>
            <copy>
                <from variable="processInput"/>
                <to variable="AInput"/>
            </copy>
        </assign>
        <scope>
            <faultHandlers>
                <catch faultName="faultA" faultVariable="AError"/>
            </faultHandlers>
            <sequence>
                <invoke name="invokeA" partnerLink="serviceA" inputVariable="AInput" outputVariable="AOutput"/>
            </sequence>
        </scope>
        <assign>
            <copy>
                <from variable="AOutput"/>
                <to variable="BCInput"/>
            </copy>
        </assign>
        <flow>
            <sequence>
                <invoke name="invokeB" partnerLink="serviceB" inputVariable="BCInput"/>
                <receive name="receive_invokeB" partnerLink="serviceB" variable="BOutput"/>
            </sequence>
            <sequence>
                <invoke name="invokeC" partnerLink="serviceC" inputVariable="BCInput"/>
                <receive name="receive_invokeC" partnerLink="serviceC" variable="COutput"/>
            </sequence>
        </flow>
        <switch>
            <case>
                <!-- assign value to processOutput -->
            </case>
        </switch>
        <invoke name="reply" partnerLink="client" inputVariable="processOutput"/>
    </sequence>
</process>
```

ภาพที่ 8 ตัวอย่างของไฟล์ BPEL

2. งานวิจัยที่เกี่ยวข้อง

2.1 Web-Services Modeling for E-Marketplace [13]

2.1.1 บทนำ

ในปัจจุบันการทำธุรกิจร่วมกันระหว่างผู้ซื้อและผู้ขาย หรือระหว่างผู้ขายด้วยกัน ไม่ว่าจะเป็นด้านการผลิต การจัดหา การขาย การนำส่ง และการจัดซื้อในสายการผลิต ต้องเสียเวลาในการกำหนดข้อตกลงซึ่งต้องทำร่วมกันเพื่อให้เป็นแนวทางหรือแบบแผนเดียวกัน

เว็บเซอร์วิสทำให้เกิด e-marketplace ซึ่งเป็นพื้นฐานริเริ่มของการซื้อขายสินค้าหรือบริการระหว่างผู้ซื้อและผู้ขาย หรือระหว่างผู้ขายด้วยกันเอง และในปัจจุบันมีแนวโน้มที่เพิ่มมากขึ้นเรื่อย ๆที่จะนำเทคโนโลยีของเว็บเซอร์วิสมาใช้ในทำธุรกิจการค้าอิเล็กทรอนิกส์ (E-Commerce) แต่ก็ยังมีข้อบกพร่องในการจัดการ e-marketplace ที่ดี ซึ่งเป็นปัญหาที่ต้องเปลี่ยนแปลงและพัฒนาให้ดีขึ้น ซึ่งในงานวิจัยนี้ได้สร้างแบบจำลองเว็บเซอร์วิส เพื่อเป็นแนวทางในการดำเนินการอย่างเป็นระบบในการซื้อขายสินค้า และการเข้าร่วมทำธุรกิจระหว่างผู้ขายเอง และในแบบจำลองนี้ยังแสดงให้เห็นขั้นตอนที่ลูกค้าซื้อบริการท่องเที่ยวผ่านทางตัวแทนท่องเที่ยว (Travel Agency) และกิจกรรมในการทำธุรกิจระหว่างผู้ขายเอง ตั้งแต่การจองโรงแรม การจองเที่ยวบิน การจองรถเช่า และการชำระเงินกับธนาคาร ซึ่งทำให้เกิดการเชื่อมโยงเครือข่ายธุรกิจกัน

แบบจำลองของเว็บเซอร์วิสที่ถูกคิดค้นออกแบบขึ้นมานี้ อยู่บนพื้นฐานของความเป็นจริงของการดำเนินการทางธุรกิจ ทำงานโดยอาศัยมาตรฐานเทคโนโลยีของเว็บเซอร์วิส ดังเช่น XML, SOAP, WSDL และ UDDI

งานวิจัยนี้ได้อธิบายถึงกิจกรรมของสิ่งที่เกี่ยวข้องกับระบบ พร้อมแสดงตัวอย่างประกอบ และได้นำเสนอถึงขั้นตอนการจัดการของเว็บเซอร์วิสทั้งภายในองค์กรและระหว่างองค์กร การแนะนำถึงการบริการแบบกระจายที่เป็นโครงสร้างพื้นฐานที่เหมาะสมสำหรับสนับสนุนการดำเนินการ และการอภิปรายถึงงานที่เกี่ยวข้องและสรุปผล

2.1.2 Service oriented computing context

งานวิจัยนี้มีลักษณะการทำงานแบบกระจายและใช้กับ e-business เป็นแบบจำลองที่มีเทคโนโลยีสนับสนุน มีระเบียบแบบแผน มีหลักการและทฤษฎีทางวิศวกรรม และมีสิ่งที่เกี่ยวข้องกับการจัดการองค์ความรู้ทั้งภายในและระหว่างองค์กร

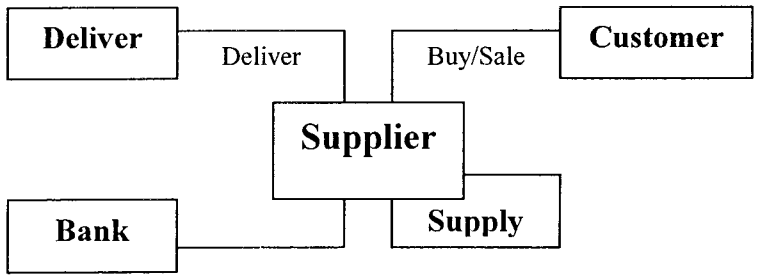
ในการจัดการกับปัญหาในงานนี้มีแบบจำลองที่เป็นระเบียบแบบแผน และโดยเฉพาะอย่างยิ่งใช้เป็นแบบแผนในการดำเนินการของเว็บเซอร์วิสทั้งภายในและระหว่างองค์กรก่อให้เกิดความสัมพันธ์ในธุรกิจระหว่างผู้ขายด้วยกันเอง ความแตกต่างของหน้าที่ของแต่ละส่วนซึ่งสามารถอธิบายได้ดังนี้ (ดูภาพที่ 9 ประกอบ)

Customer (ลูกค้าหรือผู้ซื้อ) – ทำการซื้อสินค้าหรือบริการจากบ้าน โดยผ่านทางอินเทอร์เน็ต และใช้บัตรเครดิตในการชำระค่าสินค้าหรือบริการ

Supplier (ผู้ขาย) – ทำการจัดเตรียมสินค้าหรือบริการแก่ลูกค้า รวมถึงการเชื่อมต่อกับธนาคารเพื่อชำระเงิน และติดต่อทำธุรกิจกับผู้ขายหรือผู้ผลิตรายอื่น

Bank (ธนาคาร) – รับชำระเงินโดยมีระบบ SET (Secure Electronic Transaction) สำหรับป้องกันความปลอดภัยในการจ่ายเงินของลูกค้าที่ชำระเงินผ่านทางบัตรเครดิต

Delivery (การนำส่ง) – เมื่อมีสินค้าที่ต้องการนำส่ง ทำการส่งสินค้าหรือผลิตภัณฑ์



ภาพที่ 9 องค์ประกอบที่เกี่ยวข้องกับผู้ขาย [13]

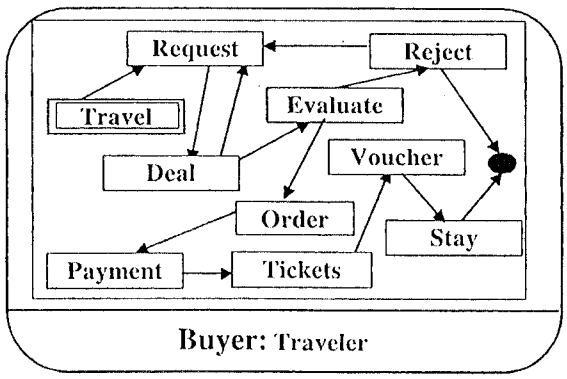
กระบวนการทำงานของแบบจำลองเว็บเซอร์วิส โดยที่ตัวแทนท่องเที่ยว (Travel Agency) จะทำหน้าที่เป็นสื่อกลางในการให้บริการลูกค้าในการท่องเที่ยว โดยลูกค้าสามารถเข้าใช้บริการได้โดยผ่านทางเว็บไซต์ โดยมีองค์ประกอบดังนี้คือ (1) ลูกค้าสั่งซื้อเป็นลักษณะของการท่องเที่ยวแบบเหมาจ่าย (Package Tour) ซึ่งในแต่ละ package ประกอบด้วย 3 ส่วนคือ ตั๋วเดินทาง สิ่งอำนวยความสะดวก และรถเช่า (2) บริษัทที่ทำธุรกิจเกี่ยวกับเที่ยวบิน (3) ธุรกิจรถเช่าสำหรับใช้ในการเดินทางเมื่อมาถึง (4) ธุรกิจโรงแรมที่พัก (5) ธนาคาร (6) ธุรกิจการขนส่ง ในกรณีที่นำส่งตั๋วเดินทาง (7) ตัวแทนการท่องเที่ยว หรือ Travel Agency เพื่อเป็นตัวกลางในการติดต่อกับลูกค้า และประสานงานทางธุรกิจกับบริษัทสายการบิน โรงแรม และบริษัทที่ให้บริการรถเช่า

การพาณิชย์อิเล็กทรอนิกส์ (E-Commerce) เป็นแนวทางหนึ่งที่องค์กรธุรกิจส่วนใหญ่หันมาใช้เพื่อเพิ่มกำไรให้กับธุรกิจ บางธุรกิจตั้งขึ้นใหม่เพื่อทำการค้าอิเล็กทรอนิกส์เพียงอย่างเดียว การทำการพาณิชย์อิเล็กทรอนิกส์ที่เห็นกันในปัจจุบัน มีลักษณะเป็น B2C (Business to Customer) คือการขายสินค้าจาก ผู้ขายที่เป็นองค์กรธุรกิจไปยังผู้ซื้อซึ่งเป็นบุคคลทั่วไป และ B2B (Business to Business) ซึ่งเป็นการติดต่อทำธุรกิจกันระหว่างทางธุรกิจ ผ่านโลกอิเล็กทรอนิกส์ ทำให้การดำเนินธุรกิจเป็นไปได้อย่างมีประสิทธิภาพมากขึ้น ซึ่งประโยชน์ อย่างแรกคือเป็นการดำเนินการทางธุรกิจระหว่างองค์กรที่เป็นไปอย่างอัตโนมัติ และอย่างที่สองลูกค้าสามารถทำการซื้อสินค้าหรือบริการได้โดยง่ายผ่านทางอินเทอร์เน็ต

2.1.3 การจัดการเกี่ยวกับการดำเนินงานทั้งภายในและระหว่างองค์กร

ในการจัดการกระบวนการทำงานมีสาระสำคัญอยู่ที่กิจกรรมของ e-marketplace ทั้งภายในและระหว่างองค์กรเอง แบบจำลองของพันธมิตรทางการค้านี้ได้แสดงถึงกิจกรรมความสัมพันธ์ที่เกิดขึ้น มีด้วยกัน 2 ส่วน ดังนี้

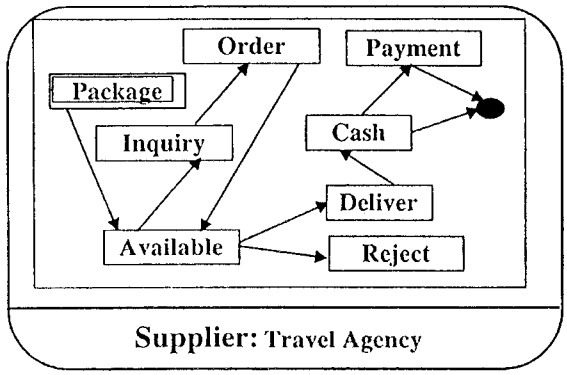
(1) กระบวนการทางธุรกรรมอิเล็กทรอนิกส์ภายในองค์กร



ภาพที่ 10 กระบวนการทางธุรกรรมอิเล็กทรอนิกส์สำหรับการจัดซื้อจัดหา [13]

กระบวนการทำงานของลูกค้าหรือผู้ซื้อจะเป็นไปตามขั้นตอนดังภาพที่ 10 นี้ โดยเริ่มจากลูกค้าต้องการที่จะท่องเที่ยวสอบถามรายการการท่องเที่ยวกับตัวแทนของบริษัททท่องเที่ยว จากนั้นตัวแทนของบริษัททท่องเที่ยวก็จะนำเสนอรายการท่องเที่ยวให้กับลูกค้า เมื่อลูกค้าสนใจก็จะตอบกลับมา จากนั้นตัวแทนของบริษัททท่องเที่ยวจะสอบถามไปยังบริษัทผู้ให้บริการต่าง ๆเช่น การจองโรงแรมที่พัก การจองเที่ยวบิน การจองรถเช่า พร้อมกับสอบถามถึงราคา แล้วจึงประเมินราคาตอบกลับลูกค้าซึ่งมีลักษณะเป็นการท่องเที่ยวแบบเหมาจ่าย (Package Tour) เมื่อลูกค้ายอมรับข้อเสนอแล้วก็ทำการชำระเงิน จากนั้นบริษัทตัวแทนก็จะส่งตั๋วเดินทางและใบเสร็จรับเงินกลับมาให้

ทางด้านของบริษัทตัวแทนท่องเที่ยวเมื่อได้รับการซื้อบริการจากลูกค้าหรือผู้ซื้อแล้ว ก็จะไปดำเนินการจัดเตรียมหาบริการให้กับลูกค้าที่ซื้อบริการกับบริษัทต่าง ๆ เช่น โรงแรม บริษัทการบิน และบริษัทขนส่ง เป็นลักษณะการทำธุรกิจ B2B



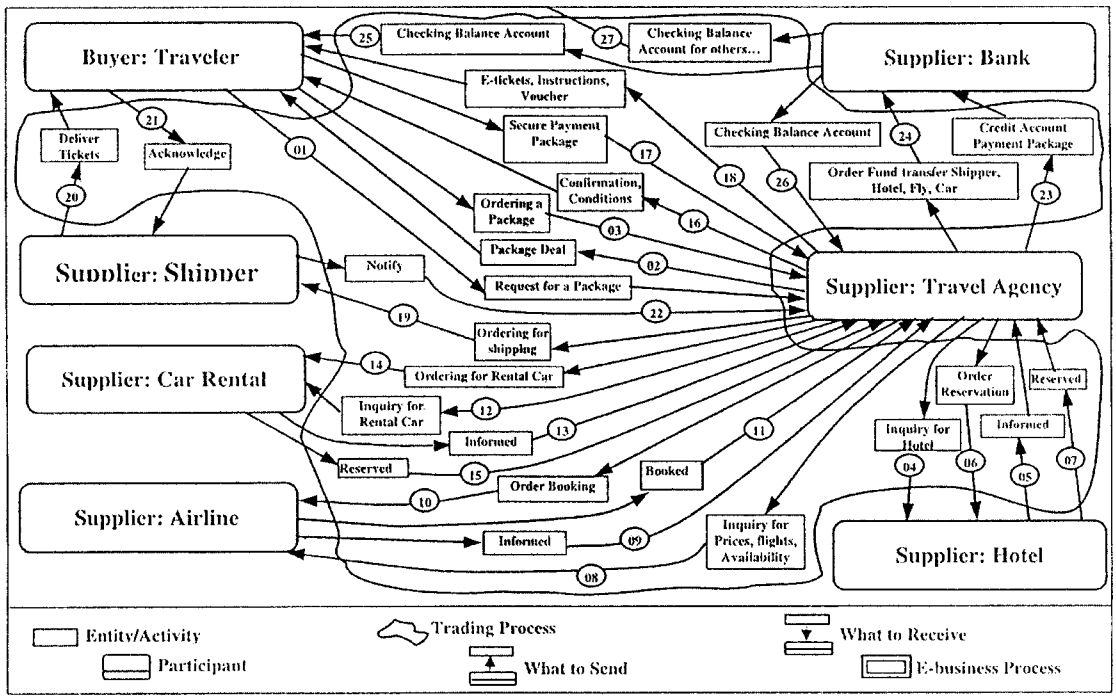
ภาพที่ 11 กระบวนการทางธุรกรรมอิเล็กทรอนิกส์สำหรับการขาย [13]

ขั้นตอนของการขาย package ท่องเที่ยวของผู้ขายหรือบริษัทตัวแทนท่องเที่ยว ซึ่งถูกกำหนดให้เป็นไปตามภาพที่ 11 คือเมื่อได้รับการร้องขอบริการจากลูกค้าแล้ว ทางบริษัทตัวแทนท่องเที่ยวต้องดำเนินการตามที่สั่งให้สมบูรณ์ ไม่เช่นนั้นลูกค้าอาจจะยกเลิกโปรแกรมท่องเที่ยวนี้ได้ และนี่คือส่วนที่สำคัญ ถ้าดำเนินการสมบูรณ์ ก็จะนำส่งสิ่งต่าง ๆ ซึ่งประกอบไปด้วยรายละเอียด เอกสารที่แสดงค่าใช้จ่าย และคำแนะนำที่จำเป็นทุก ๆอย่างสำหรับการเดินทาง ไปทางสื่ออิเล็กทรอนิกส์หรือทางจดหมาย และเมื่อลูกค้าชำระเงินแล้วบริษัทตัวแทนท่องเที่ยวก็จะจ่ายเงินไปยังบริษัทต่าง ๆ เพื่อจ่ายค่าที่พัก ค่าเที่ยวบิน และค่าเช่ารถ

(2) กระบวนการระหว่างองค์กรพันธมิตรทางการค้า

แบบจำลองของพันธมิตรทางการค้าเป็นการกำหนดและยอมรับข้อตกลงในการทำงานร่วมกันระหว่างองค์กร เป็นรูปแบบของกระบวนการทางธุรกิจที่มีความซับซ้อน ในแต่ละองค์กรได้มีความสัมพันธ์ซึ่งจะขึ้นอยู่กับหน้าที่และฟังก์ชั่นของแต่ละองค์กรในการทำงานร่วมกันบนตลาดอิเล็กทรอนิกส์ (e-marketplace) โดยอาศัยเว็บเซอร์วิสที่องค์กรได้จัดหามาทำให้เกิดการทำงานที่มีประสิทธิภาพ

จากตัวอย่างตามภาพที่ 12 เป็นลักษณะการดำเนินธุรกิจแบบ B2B โดยมีบริษัทตัวแทนท่องเที่ยวทำหน้าที่ให้บริการลูกค้าที่ต้องการท่องเที่ยว มีบริษัทการบินสำหรับจองเที่ยวบิน มีบริษัทที่ทำธุรกิจโรงแรมที่พัก มีบริษัทที่ให้บริการรถเช่า มีบริษัทขนส่งสินค้า และธนาคาร



ภาพที่ 12 Model of trading alliance, including e-business processes [13]

จากภาพที่ 12 สามารถอธิบายถึงขั้นตอนตามหมายเลขที่กำกับในแบบจำลองได้ดังนี้

① ผู้ซื้อหรือลูกค้าขอรายการท่องเที่ยวจาก Travel Agency

② Travel Agency แจ้งให้ทราบถึงรายการท่องเที่ยว

③ ส่ง order ของ package ที่ต้องการไปยัง TA

④ - ⑮ Travel Agency สอบถามถึงค่าใช้จ่ายและจองบริการกับ Supplier อื่น ๆ ทั้งโรงแรมที่พัก เที่ยวบิน และบริษัทรถเช่า

⑯ สอบถามการยืนยันจากลูกค้า โดยส่งรายละเอียดเงื่อนไขไปให้

⑰ ลูกค้าชำระเงินค่า Package

⑱ Travel Agency ออกใบรับรองการจ่ายเงินโดยใช้ระบบ E-tickets

⑲ จากนั้นส่ง order ไปยังบริษัทขนส่ง

⑳ บริษัทขนส่ง นำส่งตั๋วเดินทาง ใบเสร็จรับเงิน และเอกสารคำแนะนำที่จำเป็นทุก ๆ อย่างสำหรับการเดินทาง

㉑ ลูกค้าแจ้งว่าได้รับแล้ว

㉒ บริษัทขนส่งแจ้งกับ Travel Agency ว่าได้ส่งสินค้าให้ลูกค้าเรียบร้อยแล้ว

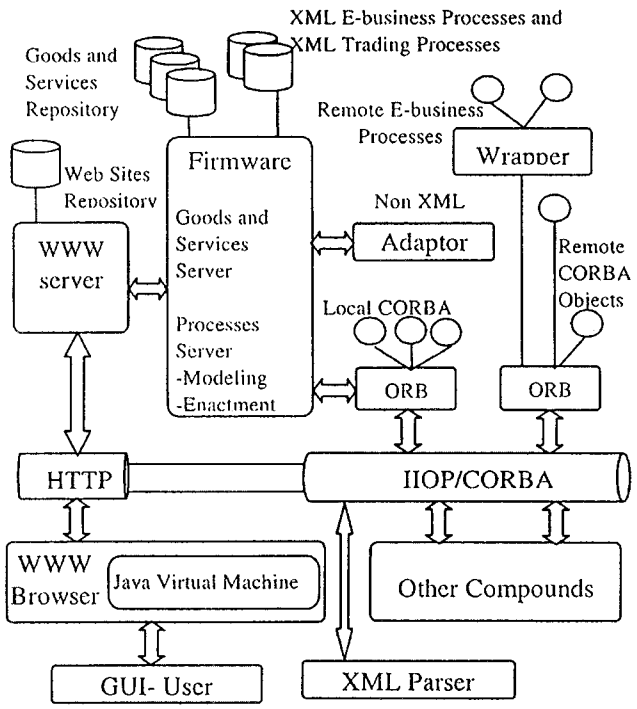
㉓ - ㉔ Travel Agency นำเงินเข้าบัญชีธนาคาร เพื่อโอนเงินให้กับ Supplier อื่น ๆ

㉕ - ㉗ ธนาคารเช็คยอดเงินของลูกค้าที่ทำธุรกรรม

2.1.4 โครงสร้างพื้นฐานและสถาปัตยกรรมที่ใช้ในการสื่อสารที่ส่งผ่านบริการต่าง ๆของระบบ

ระบบงานข้างต้นนี้จะมีความพยายามที่จะกระจายการทำงานของโปรแกรมประยุกต์บนอินเทอร์เน็ตซึ่งยังทำได้ไม่มีประสิทธิภาพมากนัก เนื่องจากแต่ละเทคโนโลยีที่ใช้ยังยึดติดกับมาตรฐานของตนเองมากเกินไป ทำให้ระบบที่ต่างกันไม่สามารถติดต่อกันได้ จึงได้มีการพัฒนามาตรฐาน Web Services ขึ้นมาเพื่อแก้ปัญหาโดยมีแนวความคิดหลักก็คือ การแลกเปลี่ยนข้อมูลซึ่งกันและกันระหว่างโปรแกรมประยุกต์ที่แตกต่างกัน

หนึ่งในคุณลักษณะของเว็บเซอร์วิสก็คือการใช้ภาษา XML ซึ่งถูกออกแบบมาเพื่อรองรับการแลกเปลี่ยนข้อมูลระหว่างกันท่ามกลางมาตรฐานที่แตกต่าง เช่น CORBA หรือ DCOM ทำให้โปรแกรมประยุกต์ทำงานร่วมกันได้อย่างมีประสิทธิภาพ ดังภาพที่ 13



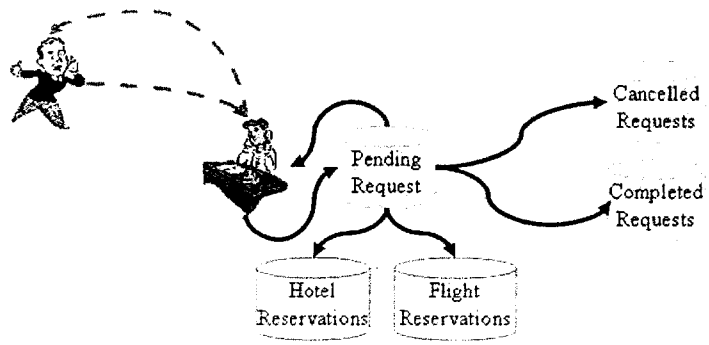
ภาพที่ 13 โครงสร้างพื้นฐานขององค์ประกอบสำหรับ e-marketplace

2.1.5 สรุปผล

มีงานวิจัยที่เกี่ยวข้องการงานวิจัยนี้หลายงาน เช่น A conceptual Architecture for a Combined Negotiation Support System [14] และ Business to Business E-Commerce [15] เพราะว่าเป็นการทำงานในลักษณะของการประมวลผลแบบกระจาย แต่ก็มีปัญหาที่ว่าไม่สนับสนุนกับงานในอนาคตที่ไม่สามารถคาดเดาสถานการณ์ได้ และไม่สามารถทำงานร่วมกันกับพันธมิตรทางการค้าอื่นผ่านทางเว็บได้ดีนัก ในขณะที่งานวิจัยนี้มีแบบจำลองแสดงถึงขั้นตอนการทำงานร่วมกันระหว่าง Suppliers อย่างชัดเจน อย่างเช่นแบบจำลองของการจัดซื้อ จัดหาสินค้า ทำให้เกิดการประสานงานกันอย่างเป็นระบบ แต่อาจยังไม่ยืดหยุ่นพอ

2.2 Web Service Composition–Current Solutions and Open Problems [16]

ทีมนักวิจัย 2 ท่านจากห้องปฏิบัติงานวิจัยของ IBM ได้ทำการศึกษาวิจัยนี้เพื่อเป็นการเปรียบเทียบและอภิบายถึงวิธีการแก้ปัญหาของการสร้างองค์ประกอบของเว็บเซอร์วิสระหว่างภาษาที่ใช้ควบคุมการไหลเวียนของข้อมูล คือ BPEL4WS [10] และ DAML-S [17] โดยได้สร้างเค้าเรื่องตัวอย่างเพื่อเป็นการประกอบการวิจัยเกี่ยวกับเรื่องจองบริการท่องเที่ยว (Package Tour) ผ่านทางตัวแทนท่องเที่ยว (Travel Agency) ดังภาพที่ 14



ภาพที่ 14 เค้าเรื่องตัวอย่างแสดงการทำงานของ Travel Agency

จากภาพที่ 14 อธิบายได้ว่าเมื่อลูกค้าต้องการที่จะเดินทางไปท่องเที่ยว จะทำการติดต่อผ่านทางตัวแทนท่องเที่ยว จากนั้นตัวแทนท่องเที่ยวซึ่งทำหน้าที่รับคำร้องขอทั้งหมดเพื่อทำการติดต่อไปยังผู้ให้บริการจองตั๋วเครื่องบิน และจองโรงแรม ตามที่ลูกค้าต้องการและจะต้องแจ้งให้ลูกค้าทราบเพื่อทำการยืนยัน ถ้าการจองล้มเหลวหรือลูกค้าต้องการยกเลิก ตัวแทนท่องเที่ยวจะติดต่อทุกผู้ให้บริการเพื่อยกเลิกทั้งหมด ลองจินตนาการดูว่าถ้าต้องติดต่อกับผู้ให้บริการอื่น ๆ จำนวนมาก ทั้งในส่วนของการจองตั๋วเครื่องบิน และการบริการที่พัก แน่นอนว่าโปรแกรมประยุกต์ของผู้ให้บริการแต่ละรายนั้นทำงานแยกกัน หรือมีระบบเป็นของตัวเอง นั้นก็ทำให้เห็นภาพที่ชัดเจนขึ้นว่า การใช้ระบบเดิม หรือระบบปิดไม่สามารถดำเนินการเพื่อสนองความต้องการที่แท้จริงของลูกค้าต่อไปได้

ดังนั้นการให้บริการรูปแบบใหม่จะต้องมีการผสานความร่วมมือระหว่างกัน โดยแต่ละส่วนต้องมีการสนองตอบต่อกันในแนวทางที่ถูกต้องหรือแนวทางเดียวกัน ในสถานการณ์ใหม่นี้ ตัวแทนการท่องเที่ยวยังคงต้องสร้างหรือวางแผนการท่องเที่ยว ซึ่งในขณะนี้มีการทำงานที่อิสระต่อกัน ซึ่งผลการติดต่อจะถูกส่งกลับมายังตัวแทนการท่องเที่ยว เพื่อตอบผลการดำเนินการว่าสำเร็จหรือล้มเหลว เมื่อการขั้นตอนทั้งหมดเสร็จสมบูรณ์ การยืนยันจะเป็นการนำเอาเอกสารทั้งหมดออกมาเพื่อให้เกิดการยืนยันทุกบริการพร้อมกันเกี่ยวกับแผนการเดินทางที่เสร็จสิ้นสมบูรณ์ แต่ถ้าบางบริการล้มเหลว ขั้นตอนการยกเลิกบริการอื่น ๆจะเกิดขึ้นเพื่อทบทวนทัวร์ดังกล่าวอีกครั้ง และการที่จะเห็นสิ่งที่เกิดขึ้นในความเป็นจริงที่การวางแผนดังกล่าวไม่ใช่เรื่องง่าย

งานวิจัยนี้ได้แสดงถึงวิธีการนำ 2 วิธีการที่ใช้ในการแก้ปัญหาในทำงานร่วมกันอย่างเป็นลำดับขั้นตอนของเว็บเซอร์วิสของแต่ละผู้ให้บริการ ดังนี้

(1) Industry Solution : WSDL + BPELWS

ส่วนที่ 1 เป็นการสร้างไฟล์ WSDL (Web Services Description Language) โดย WSDL เป็นภาษา XML ประเภทหนึ่งที่มาอธิบายวิธีการเรียกใช้เว็บเซอร์วิสซึ่งเปรียบเสมือนการอ่านคู่มือการใช้งานโปรแกรม แต่

มีข้อแตกต่างกันตรงที่ไม่เฉพาะมนุษย์เท่านั้นที่สามารถเข้าใจคู่มือ โปรแกรมที่สามารถอ่านเอกสารภาษา XML เข้าใจ ก็สามารถที่จะเข้าใจเอกสาร WSDL ได้เช่นกัน ซึ่งจากคุณสมบัตินี้ช่วยทำให้การเรียกใช้เว็บเซอร์วิสเป็นไปได้อย่างอัตโนมัติ

ในงานวิจัยนี้จะแสดงให้เห็นถึงโค้ดของ WSDL ที่ใช้ในเรียกใช้บริการเว็บเซอร์วิสจากผู้ให้บริการต่าง ๆ ผ่านทางตัวแทนท่องเที่ยว ดังนี้

```
<definitions targetNamespace="http://..." xmlns="http://schemas.xmlsoap.org/wsdl/">
    <message name = "OrderEvent"></message>
    <message name = "TripRquest"></message>
    <message name = "FlightRequest"></message>
    <message name = "HotelRequest"></message>
    <message name = "BookingFailure"></message>

    <portType name ="pt1">
        <operation name ="CToCI">
        <input message ="TripRequest"/>
        </operation>
    </portType>
    <portType name ="pt2">
        <operation name ="CIToHS">
        <output message ="HotelRequest"/>
        </operation>
    </portType>
    <portType name ="pt3">
        <operation name ="CIToFS">
        <output message ="FlightRequest"/>
        </operation>
    </portType>
    ...
    <portType name ="pt9">
        <operation name ="RIToFS">
        <output message ="BookingFailure"/>
        </operation>
    </portType>
</definitions>
```

ส่วนที่ 2 เป็นการสร้างไฟล์ BPEL เพื่อสร้าง Business Process ในธุรกิจการท่องเที่ยว

```
<process name="TripHandling">
    <partners>
      <partner name="Customer" myRole="TripHandlingAgent"
      serviceLinkType="ExternalServiceLink"
        partnerRole="CustomerAgent"/>
      <partner name="FlightService" myRole="TripHandlingAgent"
      serviceLinkType="InternalServiceLink"
        partnerRole="FlightServiceAgent"/>
      <partner name="HotelService" myRole="tripHandlingAgent"
      serviceLinkType="InternalServiceLink"
        partnerRole="HotelServiceAgent"/>
    </partners>
    <containers>
      <container name="OrderEvent" messageType="OrderEventType"/>
      <container name="TripRequest" messageType="TripRequestType"/>
      <container name="FlightRequest" messageType="FlightRequestType"/>
      <container name="HotelRequest" messageType="HotelRequestType"/>
      <container name="BookingFailure" messageType="BookingFailureType"/>
    </containers>
    <sequence>
      <receive partner="Customer" portType="pt1" operation="CToCI" container="OrderEvent"/>
      <flow>
        <invoke partner="HotelService" portType="pt2" operation="CIToHS"
            inputContainer="HotelRequest"/>
        <invoke partner="FlightService" portType="pt3" operation="CIToFS"
```

```
            inputContainer="FlightRequest"/>
        </flow>
      </sequence>
      <flow>
        <receive partner="HotelService" portType="pt4" operation="HSToEVAL1"
          container="HotelRequest"/>
        <receive partner="FlightService" portType="pt5" operation="FSToEVAL1"
          container="FlightRequest"/>
      </flow>
      <switch>
        <case condition="condition1">
          <invoke partner="Customer" portType="pt6" operation="ConIToC"
            inputContainer="TripRequest"/>
        </case>
        <otherwise>
          <flow>
            <invoke partner="Customer" portType="pt7" operation="RIToC"
              inputContainer="BookingFailure"/>
            <switch>
              <case condition="condition2">
                <invoke partner="HotelService" portType="pt8" operation="EVAL2ToHS"
                  inputContainer="BookingFailure"/>
              </case>
              <otherwise>
                <invoke partner="FlightService" portType="pt9" operation="EVAL2ToFS"
                  inputContainer="BookingFailure"/>
              </otherwise>
            </switch>
          </flow>
        </otherwise>
      </switch>
</process>
```

(2) Semantic Web Solution : RDF/DAML-s + Golog/Planning

ส่วนที่ 1 เป็นการเว็บเซอร์วิสในรูปแบบของ Semantic Web ซึ่งวัตถุประสงค์หลักของ Semantic Web คือการเพิ่มส่วนที่เรียกว่าเมตะดาต้า (Metadata) เข้าไปเป็นส่วนหนึ่งของข้อมูลออนไลน์ เพื่อเครื่องคอมพิวเตอร์จะสามารถอ่านข้อมูลเหล่านั้นและเข้าใจได้ว่าข้อมูลนั้นคืออะไรหรือเป็นส่วนไหนของชุดข้อมูล โดยแสดงออกเป็น Resource Description Format (RDF) และสามารถนำข้อมูลเหล่านั้นไปประมวลผลได้อย่างอัตโนมัติ

ในตัวอย่างนี้ จะเพิ่มในส่วนของ RDF เข้าไปตามรูปแบบมาตรฐาน DAML-S ซึ่งตัวอย่างนี้จะคล้ายกับวิธีจองตั๋วเครื่องบินกับบริษัท Bravo Air ที่ผู้ใช้คุ้นเคย ดังนี้

```
<profile:output>
    <profile:ParameterDescription rdf:ID="FlightItinerary">
    <profile:parameterName>
            FlightItinerary
    </profile:parameterName>
    <profile:restrictedTo rdf:resource=".../concepts.daml#FlightItinerary"/>
    <profile:refersTo rdf:resource="#roundTrip_In"/>
    </profile:ParameterDescription>
</profile:output>

<grounding:wsdlOutputMessageParts rdf:parseType="daml:collection">
    <grounding:WsdlMessageMap>
        <grounding:damlsParameter rdf:resource=".../FlightItinerary"/>
    <grounding:wsdlMessagePart>
            <xsd:uriReference rdf:value=".../availFlightItinerary"/>
        </grounding:wsdlMessagePart>
    </grounding:WsdlMessageMap>
</grounding:wsdlOutputMessageParts>
```

ส่วนที่ 2 เป็นโค้ดตัวอย่างของการประมวลผลซึ่งเป็นไปตามมาตรฐาน DAML-S

```
<daml:Class rdf:ID="BravoAir_Process">
 <daml:subClassOf rdf:resource=".../Process.daml#CompositeProcess"/>
  <daml:subClassOf>
   <daml:Restriction>
    <daml:onProperty rdf:resource=".../Process.daml#composedOf"/>
     <daml:toClass>
      <daml:Class>
       <daml:intersectionOf rdf:parseType="daml:collection">
        <daml:Class rdf:about="process:Sequence"/>
         <daml:Restriction>
          <daml:onProperty rdf:resource=".../Process.daml#components"/>
           <daml:toClass>
            <daml:Class>
  <process:listOfInstancesOf rdf:parseType="daml:collection">
    <daml:Class rdf:about="#GetDesiredFlightDetails"/>
    <daml:Class rdf:about="#SelectAvailableFlight"/>
    <daml:Class rdf:about="#BookFlight"/>
  </process:listOfInstancesOf>
                </daml:Class>
</daml:Class>
```

สรุปผลและการศึกษาวิจัยต่อ

จากการศึกษาเปรียบเทียบใน 2 วิธีที่ใช้แก้ปัญหาคือ the industrial approach และ the Semantic Web approach มีความสามารถในการทำ Business Process เทียบเท่ากัน แต่ใน Semantic Web นั้นมีโครงสร้างภาษาที่ซับซ้อนกว่า นอกจากนี้ผู้วิจัยพบว่าทั้ง 2 วิธีนี้ยังมีประเด็นปัญหาย่อยอื่น ๆ ที่เกี่ยวข้องและสัมพันธ์กับเรื่องของการวางแผนและนำเอาเทคโนโลยีทางด้าน AI มาใช้ ดังนั้นในการที่จะทำการศึกษาต่อ ควรที่จะมีการนำเอาเทคโนโลยีของ AI เข้ามาใช้ในการออกแบบการศึกษาวิจัยด้วย เพราะเป็นส่วนที่มีความสัมพันธ์กับลักษณะของปัญหาที่ศึกษาในครั้งนี้

2.3 A Multi-Agent Based Tourism Kiosk on Internet [18]

Chris Yeung, Pang-Fei Tung และ Jerome Yen ได้ศึกษาวิจัยระบบของบริษัทบริการการท่องเที่ยวที่อาศัยตัวแทนการท่องเที่ยวหลายตัว (Multi-Agent) ในประเทศฮ่องกง ซึ่งระบบนี้ทำให้ผู้ใช้บริการได้ดึงเอาสารสนเทศที่เป็นปัจจุบันที่สุดเกี่ยวกับฮ่องกงออกมาใช้ผ่านทางเว็บบราวเซอร์ที่สนับสนุนภาษาจาวา ระบบประกอบด้วยชุดตัวแทนซอฟต์แวร์ซึ่งจัดการกับสารสนเทศประเภทต่าง ๆ เช่น โรงแรม ศูนย์การค้า โรงภาพยนตร์ ฯลฯ โดยมีภาษา KQML เป็นภาษาที่ใช้ในการติดต่อสื่อสารและแลกเปลี่ยนสารสนเทศระหว่างตัวแทนบริการ ผู้ใช้ระบบสามารถส่งคำค้นไปยังระบบได้ แล้วระบบจะส่งต่อไปยังตัวแทนบริการท่องเที่ยวต่อไปเรื่อย ๆ จนได้ข้อมูลผลลัพธ์และดึงสารสนเทศมาให้ผู้ใช้ต่อไป

ในการติดต่อสื่อสารและแลกเปลี่ยนสารสนเทศระหว่างตัวแทนบริการท่องเที่ยวในแต่ละแห่งนั้น พบว่ายังมีข้อจำกัดอยู่เช่น ในกรณีที่ตัวแทนบริการในแต่ละแห่งนั้นมีความแตกต่างกันทางด้านเทคโนโลยีหรือภาษาในการพัฒนาตัวแทนบริการ จะทำให้เกิดปัญหาในการแลกเปลี่ยนสารสนเทศระหว่างตัวแทนบริการเองได้ ซึ่งต่างจากงานวิทยานิพนธ์นี้ ซึ่งมีการนำเอาเทคโนโลยีตัวบริการเว็บ ทำให้ลดข้อจำกัดในเรื่องการแลกเปลี่ยนข้อมูลระหว่างองค์กร ถึงแม้ว่าตัวแทนบริการจะพัฒนาจากเทคโนโลยีหรือภาษาใดก็ตาม ก็ไม่มีปัญหาในเรื่องของการติดต่อสื่อสารและและเปลี่ยนข้อมูลร่วมกันระหว่างตัวแทนบริการได้

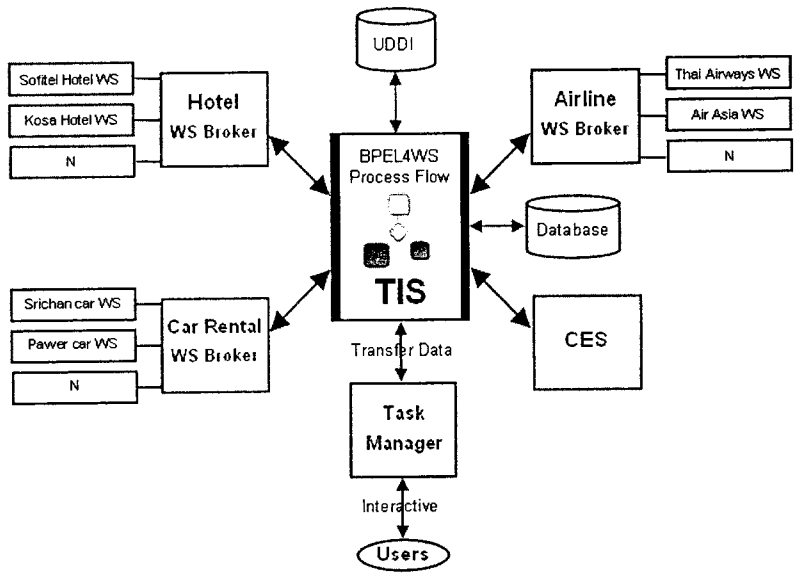
2.4 Web Service Composition Languages: Old Wine in New Bottles? [19]

Van der Aalst, Marlon Dumas และ Arthur ter Hofstede ได้ทำการศึกษาวิจัยภาษาต่าง ๆ ที่ใช้สร้างองค์ประกอบตัวบริการเว็บ ทั้งนี้เมื่อเร็ว ๆ นี้ เกิดมีภาษาหลายภาษาสำหรับใช้สร้างองค์ประกอบตัวบริการเว็บ อย่างเช่น ภาษา BPEL4WS (Business Execution Language for Web Services) และภาษา WSCI (Web Service Choreography Interface) โดยเป้าหมายของภาษาเหล่านี้คือ เพื่อเชื่อมตัวบริการเว็บต่าง ๆ เข้าด้วยกัน ภาษาต่าง ๆ เหล่านี้ได้ยืมแนวคิดจากระบบการจัดการการไหลเวียนของงานแล้วนำแนวคิดเหล่านี้เข้าไว้ในสิ่งที่เรียกว่า Web Services Stack จนถึงปัจจุบันนี้ได้ใช้ความพยายามน้อยมากหรือไม่ได้ใช้เลยที่จะประเมินความสามารถและข้อจำกัดต่าง ๆ ของภาษาเหล่านี้อย่างเป็นระบบ เช่น ภาษา BPEL4WS กล่าวกันว่าจะรวมมาตรฐานอื่น ๆ ที่ดีที่สุดสำหรับการสร้างองค์ประกอบตัวบริการเว็บ โดยดึงเอาข้อดีจากภาษา WSFL (Web Services Flow Language) ของบริษัทไอบีเอ็ม และภาษา XLANG ของบริษัทไมโครซอฟต์ ทำให้ได้การผสมผสานกันเป็นรูปแบบกระบวนการที่เป็นโครงสร้างชั้นของแบบจำลอง อย่างไรก็ตาม ด้านต่าง ๆ เช่น ความลึกซึ้งเชิงความหมาย ความเพียงพอ มุมมองในแง่คิดของคนทั่วไป และการบอกลักษณะอย่างเป็นทางการของ BPEL4WS เช่นความสามารถการเข้าถึง ยังไม่ได้ทำการศึกษาอย่างเป็นระบบ ถึงแม้ว่า BPEL4WS จะไม่ใช่ข้อเสนอที่เลวอะไรนัก แต่ก็น่าที่งที่ได้รับความสนใจอย่างมากในปัจจุบัน ในขณะที่ประเด็นปัญหาที่เป็นพื้นฐานมากกว่าหลายประเด็น เช่น ความลึกซึ้งเชิงความหมาย ไม่ได้รับความสนใจเท่าที่ควร ดังนั้นงานวิจัยนี้จึงทำการวิเคราะห์และประเมินภาษาที่เรียกว่าเป็นมาตรฐานสำหรับการสร้างองค์ประกอบตัวบริการเว็บ ซึ่งก็พบว่าภาษา BPEL4WS มีฟังก์ชั่นการทำงานที่มากกว่าภาษาที่ใช้ในการสร้างองค์ประกอบตัวบริการเว็บอื่น ๆ และเป็นภาษาที่ดีที่สุดสำหรับใช้ในการสร้างองค์ประกอบตัวบริการเว็บในปัจจุบัน

บทที่ 3
วิธีการดำเนินการวิจัย

1. ออกแบบสถาปัตยกรรมของธุรกิจการท่องเที่ยวแบบครบวงจรโดยใช้องค์ประกอบตัวบริการเว็บ

ออกแบบและพัฒนาสถาปัตยกรรมของระบบธุรกิจการท่องเที่ยวแบบครบวงจร โดยใช้องค์ประกอบตัวบริการเว็บ เพื่อทำให้ตัวบริการเว็บของธุรกิจบริการการท่องเที่ยวต่าง ๆ สามารถติดต่อสื่อสารกันและทำงานร่วมกันได้อย่างอัตโนมัติ ซึ่งเป็นการอำนวยความสะดวกให้กับลูกค้าที่ใช้บริการโดยสามารถได้รับบริการข้อมูลการท่องเที่ยวและเลือกใช้บริการต่าง ๆ ได้โดยผ่านระบบสารสนเทศการท่องเที่ยว (Tourism Information System) หรือระบบ TIS



ภาพที่ 15 สถาปัตยกรรมของระบบการท่องเที่ยวแบบครบวงจร
โดยวิธีแบบจำลององค์ประกอบตัวบริการเว็บ

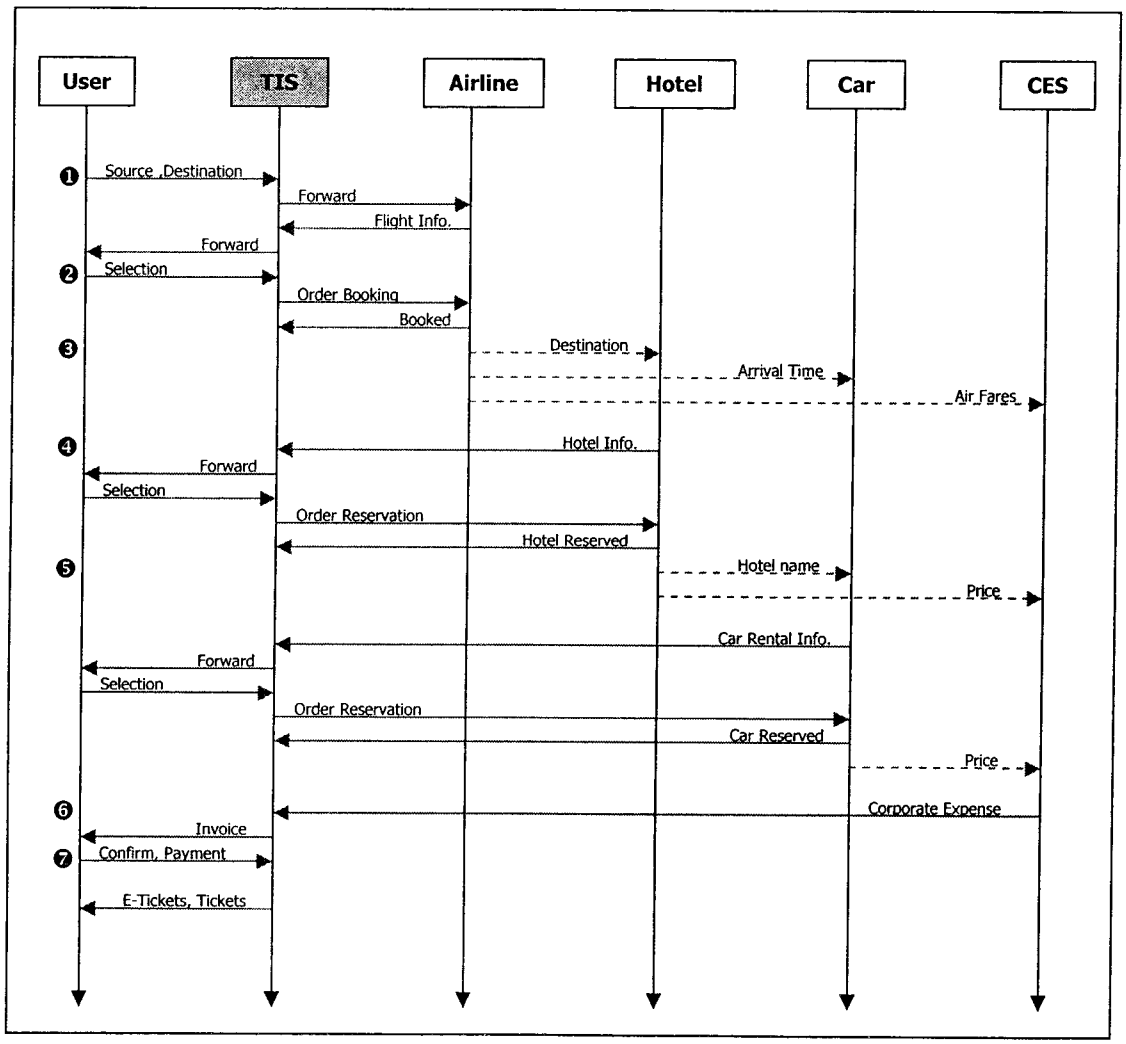
ในแต่ละองค์ประกอบสามารถอธิบายหน้าที่ได้ดังต่อไปนี้

- ระบบสารสนเทศการท่องเที่ยว หรือ ระบบ TIS ทำหน้าที่เป็นตัวกลางในการประสานงานการทำงานร่วมกันของตัวบริการเว็บต่าง ๆ ของธุรกิจบริการการท่องเที่ยว
- ตัวแทนบริการท่องเที่ยวต่าง ๆ (Web Service Broker) ทำหน้าที่เป็นตัวแทนในการเรียกไปยังตัวบริการเว็บในกลุ่มบริการของตน เพื่อรวบรวมผลลัพธ์ส่งไปยังระบบ TIS เพื่อส่งข้อมูลผ่านไปยังตัวบริหารงานให้ลูกค้าเลือกรายการ
- ตัวบริหารงาน (Task Manager Service) ทำหน้าที่ประสานงานระหว่างตัว BPEL Process กับลูกค้าหรือผู้ใช้
- ตัวบริการจัดการค่าใช้จ่าย (Corporate Expense Service หรือ CES) ทำหน้าที่สรุปค่าใช้จ่ายที่เกิดขึ้นในระบบทั้งหมด และส่งกลับไปยัง TIS เพื่อประมวลผลต่อไป

- ลูกค้าหรือผู้ใช้งานทั่วไป (Users) ทำหน้าที่เป็นผู้เรียกใช้บริการ (Service requester) จากระบบ TIS โดยลูกค้าสามารถเลือกได้ว่าจะใช้บริการใดบ้าง
- Database ทำหน้าที่เก็บ Transaction ที่เกิดขึ้นจากการทำงานของระบบ TIS
- UDDI เก็บรวบรวมการให้บริการของตัวบริการเว็บต่าง ๆ เพื่อให้ระบบ TIS ค้นหาไฟล์ WSDL เพื่อประกอบการสร้างองค์ประกอบตัวบริการเว็บ

2. ออกแบบลำดับขั้นตอนการดำเนินงานทางธุรกิจของธุรกิจบริการท่องเที่ยวแบบครบวงจร

การออกแบบกระบวนการทางธุรกิจของระบบธุรกิจบริการท่องเที่ยวแบบครบวงจร จะประกอบไปด้วยธุรกิจต่าง ๆ ที่ลูกค้าสามารถเลือกที่จะใช้บริการแบบครบวงจรได้ และเป็นธุรกิจที่ต้องทำงานร่วมกัน เช่น Airline, Hotel, Car Rental และ Corporate expense service ซึ่งทุกธุรกิจถูกออกแบบเป็นตัวบริการเว็บ ที่มีตัวกลางในการเรียกใช้บริการต่าง ๆ จากตัวบริการเว็บ ซึ่งได้แก่ ระบบ TIS โดยมีลำดับขั้นตอนการทำงานดังแสดงในภาพที่ 16



ภาพที่ 16 แสดงลำดับขั้นตอนการทำงานของธุรกิจบริการท่องเที่ยวต่าง ๆ

จากภาพที่ 16 อธิบายได้ว่าเป็นระบบการท่องเที่ยวที่จะประกอบไปด้วยธุรกิจต่าง ๆ ที่ต้องทำงานร่วมกัน ได้แก่ Airline, Hotel, Car Rental และ Corporate expense service ซึ่งทุกธุรกิจถูกออกแบบเป็นตัวบริการเว็บ (Web Service) โดยมีตัวกลางในการเรียกใช้บริการต่าง ๆจากตัวบริการเว็บนั้น ๆ หรือที่เรียกว่าระบบ TIS และสามารถอธิบายเป็นขั้นตอนได้ดังนี้

(1) ลูกค้าร้องขอบริการท่องเที่ยวจากตัวแทน หรือ TIS โดยเริ่มจากค้นหาเที่ยวบิน ซึ่งลูกค้าจะต้องส่งข้อมูลเบื้องต้น เช่น ต้นทาง ปลายทาง และวันเวลาที่ต้องการเดินทาง จากนั้น TIS จะส่งข้อมูลไปยัง Airline Web Service ทำการค้นหาและแสดงรายการข้อมูลเที่ยวบินกลับมาให้ลูกค้าทราบ

(2) ลูกค้าเลือกว่าต้องการไปกับสายการบินใด กี่ที่นั่ง เลือกชั้นแบบธรรมดาหรือแบบธุรกิจ จากนั้น TIS ก็จะส่งข้อมูลไปยัง Airline Web Service เพื่อทำการสำรองที่นั่ง

(3) ข้อมูลปลายทาง เวลาที่มาถึง และค่าโดยสารเครื่องบิน จะถูกส่งอัตโนมัติไปที่ Web Service ของ Hotel, Car Rental, Corporate Expense Service (CES) ตามลำดับ

(4) เมื่อ Hotel Web Service ได้รับข้อมูลปลายทาง ก็จะทำการค้นหาโรงแรมในบริเวณที่ลูกค้าต้องการไปออกมา พร้อมกับประเภทโรงแรมและราคามาให้ลูกค้าเลือกด้วย เช่น ปลายทางคือจังหวัดขอนแก่น ก็จะทำการค้นหาโรงแรมที่อยู่ภายในจังหวัดขอนแก่น

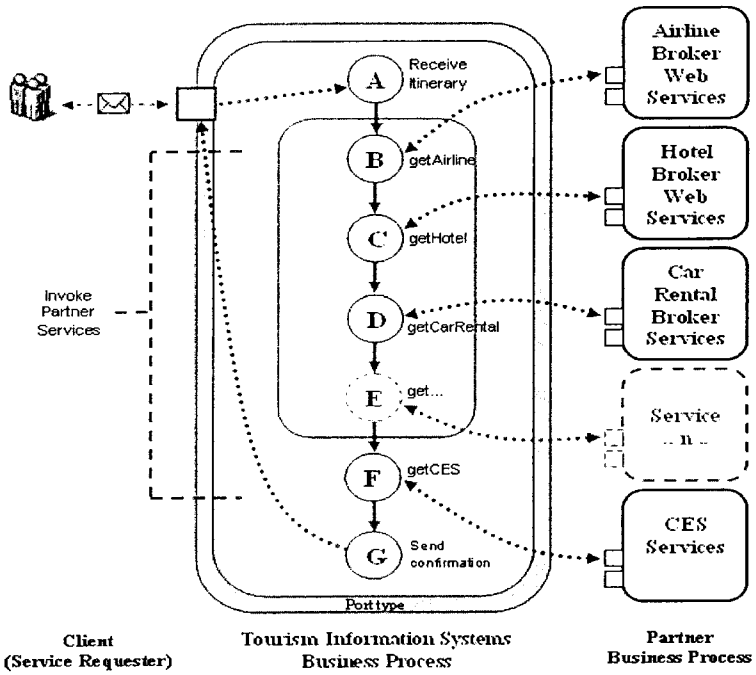
(5) เมื่อเลือกโรงแรมได้แล้ว TIS ก็จะส่งราคาค่าเช่าห้องพักโรงแรมไปยัง CES และส่งชื่อโรงแรมไปยัง Car Rental Web Service เพื่อแสดงประเภทของรถมาให้ลูกค้าเลือก โดยที่ Car Rental Web Service จะทราบอัตโนมัติแล้วว่าจะต้องไปรับลูกค้าที่ Airport ใด เวลาใด และจะต้องไปส่งลูกค้าที่โรงแรมใด เพราะได้รับข้อมูลมาแล้วจาก TIS

(6) Corporate Expense Service (CES) ทำการสรุปค่าใช้จ่ายทั้งหมด ซึ่งลูกค้าจะต้องเป็นสมาชิกอยู่ในบริการจ่ายเงินนี้ด้วย เพื่อสรุปส่งเป็น Invoice และ TIS ก็จะแสดงรายการค่าใช้จ่ายนั้นให้ลูกค้าทราบด้วย ลูกค้าทำการยืนยันรายการที่เลือกไป และชำระเงิน จากนั้น TIS ก็จะนำส่ง Tickets หรือจะสั่งพิมพ์ตั๋วผ่านระบบ E-Tickets ทาง Web Browser ได้อีกด้วย

2.1 การออกแบบกระบวนการทางธุรกิจของระบบในระดับบน : Level 0

การออกแบบกระบวนการทางธุรกิจในระดับบนหรือ Level 0 เป็นการแสดงการทำงานและการประกอบร่วมกันของตัวบริการเว็บของกลุ่มธุรกิจต่าง ๆ ได้แก่ กลุ่มธุรกิจจองตั๋วเครื่องบิน กลุ่มธุรกิจโรงแรม และกลุ่มธุรกิจรถเช่า ซึ่งสถาปัตยกรรมของระบบสามารถรองรับกับกลุ่มธุรกิจที่จะเกิดขึ้นในอนาคตได้ เช่น กลุ่มธุรกิจร้านอาหาร หรือธุรกิจสปา เป็นต้น โดยมีตัวกลางในการประสานงานระหว่างกลุ่มธุรกิจต่าง ๆ เรียกว่าระบบ TIS ซึ่งเป็นตัวจัดการกระบวนการทางธุรกิจ (Business Process) ที่จะเกิดขึ้น ทำให้การติดต่อสื่อสารและการรับส่งค่าระหว่างกลุ่มธุรกิจเป็นไปอย่างอัตโนมัติ

กระบวนการทางธุรกิจจะเริ่มตั้งแต่รับข้อมูลความต้องการเบื้องต้นจากลูกค้า (Activity A) จากนั้นทำการส่งข้อมูลต่อไปยัง Airline Broker Web Services (Activity B) ซึ่งจะมีการประมวลผลจาก Activity A ถึง Activity G ตามลำดับ ดังภาพที่ 17 โดยเริ่มแรกลูกค้าสามารถเลือกได้ว่าจะใช้บริการใดบ้าง และจะไม่ใช้บริการใดบ้าง เช่น ลูกค้าเลือกบริการจองตั๋วเครื่องบินและจองบริการจองโรงแรม แต่อาจไม่ได้เลือกบริการจองรถเช่าเป็นต้น



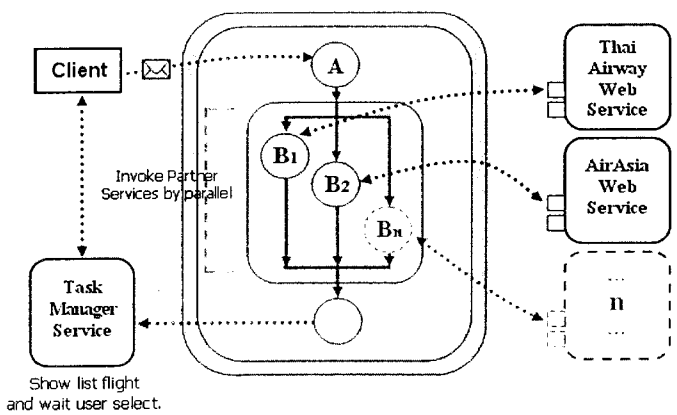
ภาพที่ 17 แผนภาพกระบวนการทางธุรกิจของระบบการท่องเที่ยวแบบครบวงจรในระดับบน : Level 0

2.2 การออกแบบกระบวนการทางธุรกิจของตัวแทนการบริการในระดับล่าง : Level 1

การออกแบบกระบวนการทางธุรกิจในระดับล่างหรือ Level 1 แบ่งตามธุรกิจบริการต่าง ๆ ออกเป็น 4 กลุ่ม ได้แก่ Airline Broker Web Services, Hotel Broker Web Services, Car Rental Broker Web Services และ Corperate Expense Service สามารถอธิบายในแต่ละกลุ่มได้ดังนี้

2.2.1 ตัวแทนบริการจองตั๋วเครื่องบิน (Airline Broker Web Services)

เป็นการออกแบบกระบวนการทางธุรกิจของตัวแทนบริการจองตั๋วเครื่องบิน ซึ่งจะเรียกไปยัง Partner อื่น ๆ ประกอบไปด้วยตัวบริการเว็บของธุรกิจการจองตั๋วเครื่องบิน ได้แก่ ThaiAirway web service และ AirAsia web service และพร้อมสนับสนุนสำหรับการเกิดขึ้นของธุรกิจจองตั๋วเครื่องบินของบริษัทอื่น ๆในอนาคต ดังภาพที่ 18



ภาพที่ 18 แผนภาพการทำงานของตัวแทนการบริการจองตั๋วเครื่องบินในระดับล่าง : Level 1

การประมวลผลจะเริ่มตั้งแต่รับค่าชื่อลูกค้า (Customer Name) สถานีต้นทาง (Source) สถานีปลายทาง (Destination) วันที่เดินทาง (Departure Date) และจำนวนผู้โดยสาร (Passenger) จาก Client (Activity A) หลังจากนั้นระบบจะประมวลผลโดยเรียกไปยังตัวบริการเว็บจองตั๋วเครื่องบินต่าง ๆ (Activity B) และส่งค่ารายการเที่ยวบินที่ได้ไปยังตัวบริหารงานหรือ Task Manager Service เพื่อแสดงรายการให้ลูกค้าได้เลือกผ่านทางเว็บบราวเซอร์ โดยค่าผลลัพธ์ที่ได้จากบริการนี้ก็จะถูกส่งต่อไปยังธุรกิจบริการอื่น ๆ ต่อไป เช่น ตัวแทนบริการโรงแรม เป็นต้น ค่าผลลัพธ์ที่ได้จากบริการนี้ ดังแสดงในตารางที่ 1

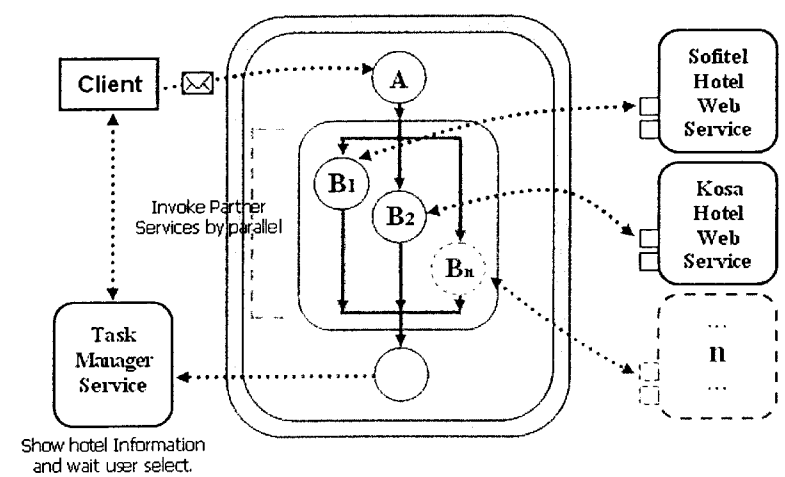
ตารางที่ 1 Input / Output of Airline Broker Web Services

Input	Output
1. Customer Name	1. Customer Name
2. Source	2. Source
3. Destination	3. Destination
4. Departure Date	4. Departure Date
5. Passenger	5. Passenger
	6. Company
	7. Flight Number
	8. Cabin Class
	9. Departing Time
	10. Arriving Time
	11. Air Fares

หมายเหตุ ค่าที่ขีดเส้นใต้คือค่าที่ถูกส่งไปประมวลผลยังตัวบริการเว็บอื่น

2.2.2 ตัวแทนบริการโรงแรม (Hotel Broker Web Services)

เป็นการออกแบบกระบวนการทางธุรกิจของตัวแทนบริการจองโรงแรมและห้องพัก ซึ่งจะเรียกไปยัง Partner อื่น ๆ ประกอบไปด้วยตัวบริการเว็บของธุรกิจโรงแรม ได้แก่ Sofitel Hotel web service และ Kosa Hotel web service และพร้อมสนับสนุนสำหรับการเกิดขึ้นของโรงแรมของบริษัทอื่น ๆ ในอนาคต ดังภาพที่ 19



ภาพที่ 19 แผนภาพการทำงานของตัวแทนการบริการโรงแรมในระดับล่าง : Level 1

การประมวลผลจะเริ่มตั้งแต่รับค่าชื่อลูกค้า (Customer Name) และสถานีปลายทาง (Destination) จาก Client หรือจากตัวแทนบริการจองตั๋วเครื่องบิน (Activity A) หลังจากนั้นระบบจะประมวลผลโดยเรียกไปยังตัวบริการเว็บของโรงแรมต่าง ๆ (Activity B) และส่งค่ารายชื่อโรงแรมและรายการห้องพักที่ได้ไปยังตัวบริหารงานหรือ Task Manager Service เพื่อแสดงรายการให้ลูกค้าได้เลือกผ่านทางเว็บบราวเซอร์ โดยค่าผลลัพธ์ที่ได้จากบริการนี้ ดังแสดงในตารางที่ 2

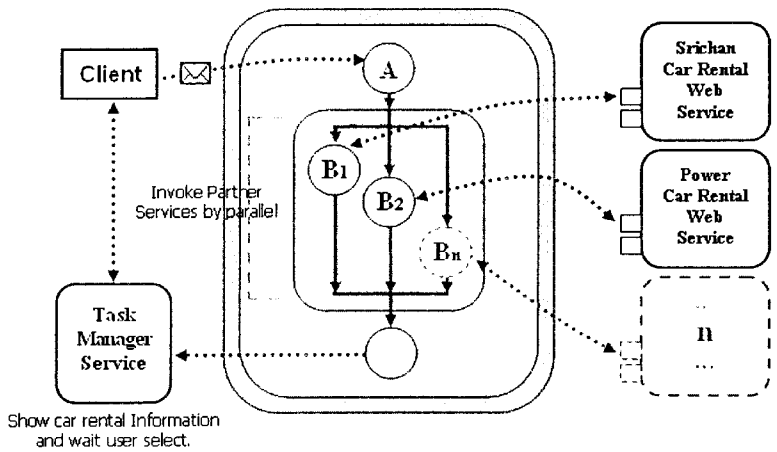
ตารางที่ 2 Input / Output of Hotel Broker Web Services

Input	Output
1. Customer Name 2. Destination	1. Customer Name 2. Destination 3. Hotel Name 4. Rating 5. Class Room 6. Check-in Date 7. Check-out Date 8. No. of Night 9. No. of Room 10. Price hotel

หมายเหตุ ค่าที่ขีดเส้นใต้คือค่าที่ถูกส่งไปประมวลผลยังตัวบริการเว็บอื่น

2.2.3 ตัวแทนบริการรถเช่า (Car Rental Broker Web Service)

เป็นการออกแบบกระบวนการทางธุรกิจของตัวแทนบริการจองรถเช่า ซึ่งจะเรียกไปยัง Partner อื่น ๆ ประกอบไปด้วยตัวบริการเว็บของธุรกิจรถเช่า ได้แก่ Srichan Car Rental web service และ Power Car Rental web service และพร้อมสนับสนุนสำหรับการเกิดขึ้นของธุรกิจรถเช่าของบริษัทอื่น ๆในอนาคต ดังสถาปัตยกรรมในภาพที่ 20



ภาพที่ 20 แผนภาพการทำงานของตัวแทนการบริการรถเช่าในระดับล่าง : Level 1

การประมวลผลจะเริ่มตั้งแต่รับค่าชื่อลูกค้า (Customer Name) สถานีปลายทาง (Destination) เวลาที่มาถึง (Arriving Time) และชื่อโรงแรม (Hotel Name) จาก Client หรือจากตัวแทนบริการจองตั๋วเครื่องบิน และตัวแทนบริการโรงแรม (Activity A) หลังจากนั้นระบบจะประมวลผลโดยเรียกไปยังตัวบริการเว็บของบริษัทรถเช่าต่าง ๆ (Activity B) และส่งค่ารายการรถเช่าที่ได้ไปยังตัวบริหารงานหรือ Task Manager Service เพื่อแสดงรายการให้ลูกค้าได้เลือกผ่านทางเว็บบราวเซอร์ โดยค่าผลลัพธ์ที่ได้จากบริการนี้ ดังแสดงในตารางที่ 3

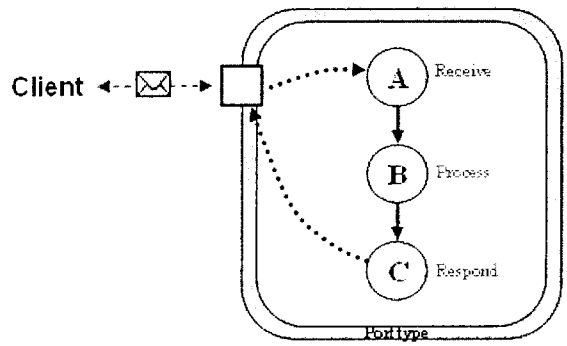
ตารางที่ 3 Input / Output of Car Rental Web Services

Input	Output
1. Customer Name	1. Customer Name
2. Destination	2. Destination
3. Arriving Time	3. Arriving Time
4. Hotel Name	4. Hotel Name
	5. Company
	6. Car Type
	7. Start Date
	8. End Date
	9. No. of Day
	10. No. of Car
	11. Price car rental

หมายเหตุ ค่าที่ขีดเส้นใต้คือค่าที่ถูกส่งไปประมวลผลยังตัวบริการเว็บอื่น

2.2.4 Corporate Expense Service

เป็นบริการสำหรับคำนวณและสรุปค่าใช้จ่ายต่าง ๆ ที่เกิดขึ้นในระบบธุรกิจการท่องเที่ยวแบบครบวงจรทั้งหมด โดยรับค่าของค่าใช้จ่ายในบริการต่าง ๆที่ลูกค้าได้ใช้บริการ ทำการประมวลผลและส่งกลับผลลัพธ์ ดังกระบวนการทางธุรกิจภาพที่ 21



ภาพที่ 21 แผนภาพการทำงานของตัวบริการจัดการค่าใช้จ่าย : Level 1

การประมวลโดยจะรับค่า ค่าโดยสารเครื่องบิน (Air Fares) ค่าห้องพักโรงแรม (Price Hotel) และค่าเช่ารถ (Price Car Rental) จาก Client หรือจากตัวบริการเว็บต่าง ๆ (Activity A) หลังจากนั้นระบบจะประมวลผล (Activity B) และส่งกลับผลลัพธ์ (Activity C) โดยค่าผลลัพธ์ที่ได้จากบริการนี้ดังแสดงในตารางที่ 4

ตารางที่ 4 Input / Output of Corporate Expense Services

Input	Output
1. Air Fares	1. Air Fares
2. Price hotel	2. Price hotel
3. Price car rental	3. Price car rental
	4. Total

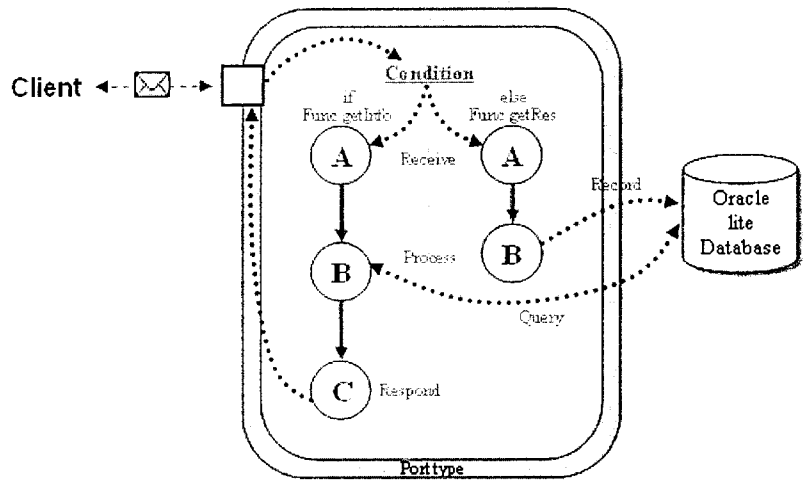
2.3 การออกแบบกระบวนการทางธุรกิจของระบบในระดับ Level 2

ออกแบบกระบวนการทางธุรกิจใน Level 2 เป็นการออกแบบการให้บริการของตัวบริการเว็บของผู้ให้บริการในกลุ่มต่าง ๆ เช่น ตัวบริการเว็บในกลุ่มที่ให้บริการจองตั๋วเครื่องบิน ได้แก่ ThaiAirway Web Service และ AirAsia Web Service ตัวบริการเว็บในกลุ่มที่ให้บริการจองโรงแรม ได้แก่ Sofitel Hotel web service และ Kosa Hotel web service และตัวบริการเว็บในกลุ่มที่ให้บริการจองรถเช่า ได้แก่ Srichan Car Rental web service และ Power Car Rental web service

การออกแบบกระบวนการทางธุรกิจในระดับ Level 2 ตัวบริการเว็บในแต่ละกลุ่มมีการออกแบบที่คล้ายกัน จึงขออภิบายกลุ่มบริการละ 1 ตัวบริการเว็บ ดังนี้

2.3.1 ThaiAirway Web Service

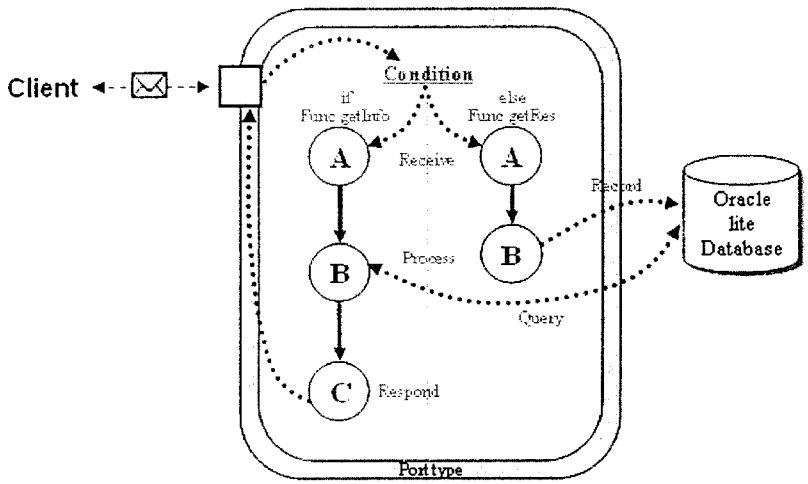
เป็นการออกแบบสถาปัตยกรรมของตัวบริการเว็บที่ให้บริการในการจองตั๋วเครื่องบิน ซึ่งตัวบริการเว็บนี้ทำงานใน 2 ฟังก์ชั่น คือ getInformation สำหรับแสดงรายการของเที่ยวบิน โดยจะรับค่าต้นทาง (Source) และปลายทาง (Destination) มาประมวลผลและส่งกลับรายละเอียดเกี่ยวกับรายการเที่ยวบิน และ getReservation สำหรับรับรายการจอง โดยรับค่าชื่อลูกค้า (Customer Name) ต้นทาง (Source) ปลายทาง (Destination) หมายเลขเที่ยวบิน (Flight Number) วันที่เดินทาง (Departure Date) ค่าโดยสาร (Air Fares) และรายละเอียดเกี่ยวกับเที่ยวบินที่ลูกค้าได้ทำการเลือก โดยใช้ภาษา BPEL ในการพัฒนาและใช้ Oracle lite Database เป็นฐานข้อมูล (รายละเอียดการจำลองตัวบริการเว็บนี้ อยู่ในภาคผนวก ข) ดังสถาปัตยกรรมในภาพที่ 22



ภาพที่ 22 แผนภาพการทำงานของ ThaiAirway Web Service

2.3.2 Sofitel Hotel Web Service

เป็นการออกแบบสถาปัตยกรรมของตัวบริการเว็บที่ให้บริการในการจองโรงแรมและห้องพัก ซึ่งตัวบริการเว็บนี้ทำงานใน 2 ฟังก์ชั่น คือ getInformation สำหรับแสดงรายการของโรงแรมและประเภทห้องพัก โดยจะรับค่าและปลายทาง (Destination) มาประมวลผลและส่งกลับรายละเอียดเกี่ยวกับรายการโรงแรมและประเภทห้องพัก และ getReservation สำหรับรับรายการจอง โดยรับค่าชื่อลูกค้า ประเภทห้องพัก จำนวนห้องพัก จำนวนวันที่เข้าพัก และราคาห้องพัก โดยใช้ภาษา BPEL ในการพัฒนาและใช้ Oracle lite Database เป็นฐานข้อมูล (รายละเอียดการจำลองตัวบริการเว็บนี้ อยู่ในภาคผนวก ข) ดังสถาปัตยกรรมในภาพที่ 23

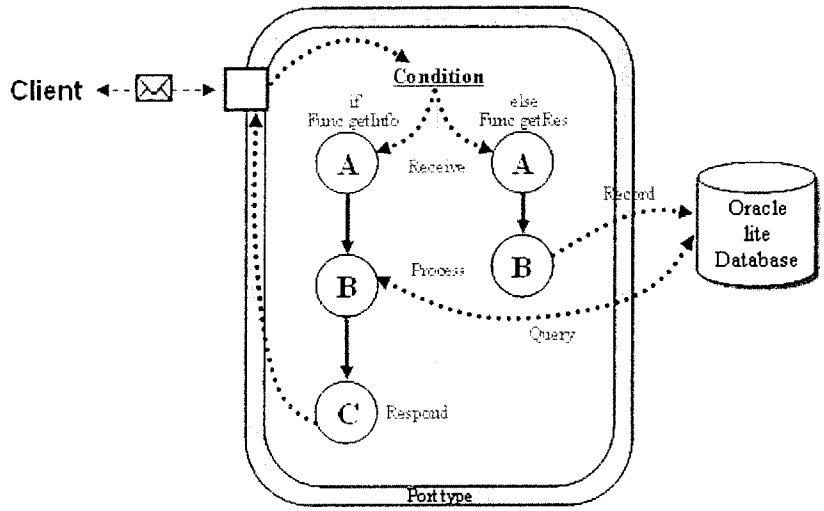


ภาพที่ 23 แผนภาพการทำงานของ Sofitel Hotel Web Service

2.3.3 Khon Kaen Car Rental web service

เป็นการออกแบบสถาปัตยกรรมของตัวบริการเว็บที่ให้บริการในการจองรถเช่า ซึ่งตัวบริการเว็บนี้ทำงานใน 2 ฟังก์ชั่น คือ getInformation สำหรับแสดงรายการรถเช่า โดยจะรับค่าและปลายทางมา

ประมวลผลและส่งกลับรายละเอียดเกี่ยวกับรายการรถเช่าและประเภทของรถเช่า และ getReservation สำหรับรับรายการจอง โดยรับค่าชื่อลูกค้า ประเภทรถเช่า จำนวนวันที่เช่ารถ และค่าเช่ารถ โดยใช้ภาษา BPEL ในการพัฒนาและใช้ Oracle lite Database เป็นฐานข้อมูล ดังสถาปัตยกรรมในภาพที่ 24



ภาพที่ 24 แผนภาพการทำงานของ Khon Kaen Car Rental web service

3. จำลองตัวบริการเว็บที่เกี่ยวข้องกับธุรกิจการท่องเที่ยว

ในการสร้างองค์ประกอบตัวบริการเว็บ (Web Service Composition) ในงานวิจัยครั้งนี้ได้ทำการจำลองตัวบริการเว็บขึ้นมาเพื่อประกอบการทำกระบวนการทางธุรกิจ มีขั้นตอนดังนี้

3.1 จำลองตัวบริการเว็บของผู้ให้บริการ (Web Service Providers) ใน Level 2

จำลองตัวบริการเว็บตามสถาปัตยกรรมที่ได้ทำการออกแบบไว้ ประกอบด้วยผู้ให้บริการในกลุ่มต่าง ๆ ได้แก่ กลุ่มที่ให้บริการจองตั๋วเครื่องบิน เช่น ThaiAirway web service และ AirAsia web service กลุ่มที่ให้บริการจองโรงแรมที่พัก ได้แก่ Sofitel hotel web service และ Kosa hotel web service กลุ่มที่ให้บริการจองรถเช่า ได้แก่ Srichan car rental web service และ Power car rental web service

บริการของแต่ละ Web Service Providers ประกอบด้วย 2 Operations ได้แก่ getInformation สำหรับให้ข้อมูลเกี่ยวกับบริการของตนเอง เช่น ThaiAirways web service เมื่อร้องขอบริการไปจะได้ข้อมูลเกี่ยวกับรายการเที่ยวบินกลับมา และ getReservation สำหรับการจองรายการที่ได้เลือกไป สำหรับตัวบริการเว็บที่ได้จำลองขึ้นมามีดังนี้

(1) จำลอง ThaiAirway web service

(2) จำลอง AirAsia web service

(3) จำลอง Sofitel hotel web service

(4) จำลอง Kosa hotel web service

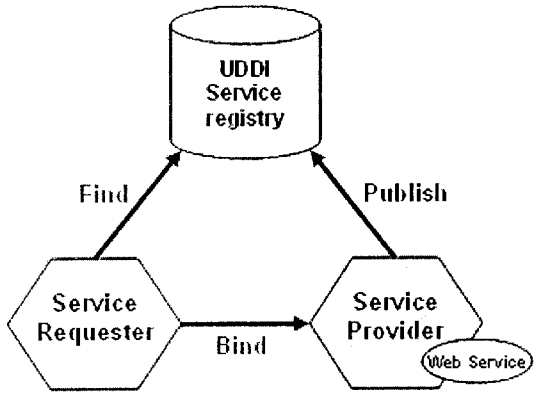
(5) จำลอง Power car rental web service

(6) จำลอง Srichan car rental web service

(ตัวอย่าง Source Code อยู่ในภาคผนวก ข)

3.2 ลงทะเบียนการบริการบน UDDI

เมื่อสร้าง Web Service Providers แต่ละบริการเสร็จเรียบร้อยแล้ว ทำการลงทะเบียนบริการของตนไว้ใน UDDI Service registry เพื่อให้ Service Requester ค้นหาบริการที่ต้องการเรียกใช้ ในงานวิจัยนี้ในเบื้องต้นได้ทำการประกาศการให้บริการของตัวบริการเว็บไว้บน UDDI ของ localhost ซึ่งตัวโปรแกรม Oracle Jdeveloper ซึ่งเป็นโปรแกรมที่ใช้ในการสร้างกระบวนการทางธุรกิจจะทำการค้นหาบริการนั้นแล้วนำมาประกอบการสร้างกระบวนการทางธุรกิจ



ภาพที่ 25 สถาปัตยกรรมของเว็บเซอร์วิส

จากภาพที่ 25 สามารถอธิบายได้ดังนี้

Requestor เป็นใครก็ตามที่ต้องการเรียกใช้บริการจาก Provider ซึ่งสามารถค้นหาบริการที่ต้องการได้จาก UDDI registry หรือ Service Registry หรือติดต่อจาก Provider โดยตรง

Registry ทำหน้าที่เป็นตัวกลางให้ Provider มาลงทะเบียนไว้ โดยใช้ WSDL ไฟล์ บอกรายละเอียดของบริษัทและบริการที่มีให้ ซึ่งอาจจะใช้หรือไม่ใช้ก็ได้

Provider เป็นผู้ให้บริการ มีหน้าที่ในการเปิดบริการเพื่อรองรับการขอใช้บริการจาก Requestor ที่เรียกเข้ามาขอใช้

4. สร้างแบบจำลององค์ประกอบตัวบริการเว็บตามมาตรฐานของ BPEL4WS

สร้างแบบจำลององค์ประกอบตัวบริการเว็บของธุรกิจการท่องเที่ยวแบบครบวงจรโดยใช้ภาษา BPEL พัฒนาขึ้นจากโปรแกรม Oracle Jdeveloper โดยจะ Composite ใน 2 ระดับ คือ สถาปัตยกรรมของ Level 0 จะ Composite ตัว Web Service Brokers (ดังภาพที่ 15) และ สถาปัตยกรรมของ Level 1 จะ Composite ตัว Web Service Providers (ดังภาพที่ 16 ภาพที่ 17 และภาพที่ 18)

4.1 ค้นหา WSDL ของบริการที่ต้องการเรียกใช้งานจาก UDDI

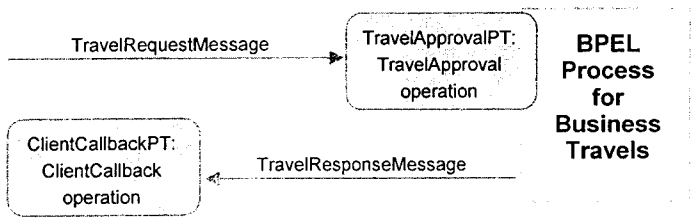
WSDL (Web Service Description Language) เป็นภาษาที่ใช้อธิบายคุณลักษณะการใช้บริการของ Web Services และวิธีการติดต่อกับ Web Services ฉะนั้นก่อนที่จะทำการสร้าง Web Service Composition จะต้อง

ทราบที่อยู่ของ WSDL ของบริการที่ต้องการจะเรียกใช้บริการ โดยสามารถค้นหาได้จาก UDDI หรือในกรณีที่ทราบที่อยู่ของ WSDL อยู่แล้วก็อาจจะไม่ต้องค้นหาจาก UDDI ก็ได้

4.2 สร้างกระบวนการทางธุรกิจตามสถาปัตยกรรมในระดับ Level 0

4.2.1 Define WSDL for the BPEL Process

กำหนดค่าของ WSDL สำหรับเตรียมรับ message จาก Clients และตอบกลับค่าผลลัพธ์ โดยได้กำหนดไว้ในรูปแบบของ XML Schema แล้วจึงทำการ import ไฟล์ xsd นั้นเข้ามาใน WSDL โดยติดต่อผ่านทาง Port Type กำหนด Partner Link Types สำหรับติดต่อกับ Web Services อื่นๆ และเป็นลักษณะการประมวลผลแบบไม่ประสานเวลา (Asynchronously) ดังภาพที่ 26 ซึ่งตัวอย่างไฟล์ TravelAgent.wsdl และ TravelAgent.xsd อยู่ในภาคผนวก



ภาพที่ 26 The WSDL for the BPEL process

4.2.2 Create the Business Process

การสร้างกระบวนการทางธุรกิจ (Business Process) ตามมาตรฐานของ BPEL4WS โครงสร้างพื้นฐานของไฟล์ BPEL ประกอบด้วย 3 ส่วนหลัก ได้แก่ <partnerLinks> <variables> และ <sequence> ดังภาพที่ 27

```
<process name="BusinessTravelProcess" ... >
  <partnerLinks>
    <!-- The declaration of partner links -->
  </partnerLinks>
  <variables>
    <!-- The declaration of variables -->
  </variables>
  <sequence>
    <!-- The definition of the BPEL business process main body -->
  </sequence>
</process>
```

ภาพที่ 27 Basic structure of BPEL process definition document

การสร้างกระบวนการทางธุรกิจตามสถาปัตยกรรมในระดับ Level 0 สามารถอธิบายถึงการสร้างกระบวนการทางธุรกิจได้เป็นขั้นตอนดังนี้

(1) กำหนดชื่อของ Process และ namespaces

```
<process name="TravelAgent"
    targetNamespace="http://xmlns.oracle.com/TravelAgent"
    xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
    xmlns:ns1="http://xmlns.oracle.com/AirlineBrokerWS"
    xmlns:ns2="http://xmlns.oracle.com/HotelBrokerWS"
```

```
    xmlns:ns3="http://xmlns.oracle.com/CarRentalBrokerWS"
    xmlns:ns4="http://xmlns.oracle.com/CESservice" >
  ...
```

(2) กำหนด Partner Links

```
<partnerLinks>
  <partnerLink name="client"
        partnerLinkType="client:TravelAgent"
        myRole="TravelAgentProvider"
        partnerRole="TravelAgentRequester"/>
  <partnerLink name="AirlineBrokerWS"
        partnerLinkType="ns1:AirlineBrokerWS"
  myRole="AirlineBrokerWSRequester"
        partnerRole="AirlineBrokerWSProvider"/>
  <partnerLink name="HotelBrokerWS"
        partnerLinkType="ns2:HotelBrokerWS"
        myRole="HotelBrokerWSRequester"
        partnerRole="HotelBrokerWSProvider"/>
  <partnerLink name="CarRentalBrokerWS"
        partnerLinkType="ns3:CarRentalBrokerWS"
        myRole="CarRentalBrokerWSRequester"
        partnerRole="CarRentalBrokerWSProvider"/>
  <partnerLink name="CESservice"
        partnerLinkType="ns4:CESservice"
        partnerRole="CESserviceProvider"/>
</partnerLinks>
```

(3) กำหนด Variable

```
<variable name="inputVariable" messageType="client:TravelAgentRequestMessage"/>
  <variable name="outputVariable"
      messageType="client:TravelAgentResponseMessage"/>
  <variable name="invoke_AB"
      messageType="ns1:AirlineBrokerWSRequestMessage"/>
  <variable name="receive_AB"
      messageType="ns1:AirlineBrokerWSResponseMessage"/>
  <variable name="invoke_HB" messageType="ns2:HotelBrokerWSRequestMessage"/>
  <variable name="receive_HB"
      messageType="ns2:HotelBrokerWSResponseMessage"/>
  <variable name="invoke_CB"
      messageType="ns3:CarRentalBrokerWSRequestMessage"/>
```

```
<variable name="receive_CB"
    messageType="ns3:CarRentalBrokerWSResponseMessage"/>
<variable name="invoke_CES" messageType="ns4:CESserviceRequestMessage"/>
<variable name="receive_CES" messageType="ns4:CESserviceResponseMessage"/>
</variables>
```

(4) กำหนด Flow ของระบบ

```
<sequence name="main">
<receive name="receiveInput" partnerLink="client" portType="client:TravelAgent"
    operation="initiate" variable="inputVariable" createInstance="yes"/>
<scope name="getAirline">
 <switch name="Switch_1">
  <case
    condition="bpws:getVariableData('inputVariable','payload','/ns9:TravelAgent/ns9:
    Services/ns9:airline') = 'yes'">
   <sequence name="Sequence_1">
    <assign name="Assign_AB1">
     <copy>
      <from variable="inputVariable" part="payload"
    query="/ns9:TravelAgent/ns9:AirlineService/ns9:source"/>
      <to variable="invoke_AB" part="payload"
    query="/ns1:AirlineBrokerWSProcessRequest/ns1:source"/>
     </copy>
     <copy>
      <from variable="inputVariable" part="payload"
    query="/ns9:TravelAgent/ns9:AirlineService/ns9:destination"/>
      <to variable="invoke_AB" part="payload"
    query="/ns1:AirlineBrokerWSProcessRequest/ns1:destination"/>
     </copy>
    </assign>
    <invoke name="Invoke_AB" partnerLink="AirlineBrokerWS"
    portType="ns1:AirlineBrokerWS" operation="initiate"
    inputVariable="invoke_AB"/>
    <receive name="Receive_AB" createInstance="no"
    partnerLink="AirlineBrokerWS" portType="ns1:AirlineBrokerWSCallback"
    operation="onResult" variable="receive_AB"/>
    <assign name="Assign_AB2">
     <copy>
      <from variable="receive_AB" part="payload"
    query="/ns1:AirlineBrokerWSProcessResponse/ns1:airlineName"/>
```

```
        <to variable="outputVariable" part="payload"
       query="/ns9:TravelAgent/ns9:AirlineService/ns9:airlineName"/>
        </copy>
       </assign>
      </sequence>
     </case>
     <otherwise>
      <assign name="Assign_AB3">
       <copy>
        <from expression="0"/>
        <to variable="invoke_CES" part="payload"
       query="/ns4:CESserviceProcessRequest/ns4:priceAirline"/>
       </copy>
      </assign>
     </otherwise>
    </switch>
   </scope>
   ...
   </sequence>
```

4.2.3 Deploy to Web Server

หลังจากการสร้างกระบวนการทางธุรกิจแล้ว จะต้องทำการ Deploy ไปยัง Web Server เพื่อเป็นการนำเอาแบบจำลององค์ประกอบตัวบริการเว็บที่สร้าง ขึ้นสู่สาธารณะบน Server จริงพร้อมทำงาน โดยมี User Interface เป็นตัวกลางในการประสานงานรับส่งค่าระหว่างผู้ใช้และแบบจำลององค์ประกอบตัวบริการเว็บ และก่อนจะทำการ Deploy ต้องกำหนดตัว Descriptor โดยมีการระบุถึงที่อยู่ของ WSDL ของแต่ละตัวบริการเว็บดังนี้

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<BPELSuitcase>
  <BPELProcess id="TravelAgent" src="TravelAgent.bpel">
    <partnerLinkBindings>
      <partnerLinkBinding name="client">
        <property name="wsdlLocation">TravelAgent.wsdl</property>
      </partnerLinkBinding>
      <partnerLinkBinding name="AirlineBrokerWS">
        <property
name="wsdlLocation">http://ekkachai:9700/orabpel/AirlineBrokerWS/AirlineBrokerWS?wsdl</property>
      </partnerLinkBinding>
      <partnerLinkBinding name="HotelBrokerWS">
        <property
name="wsdlLocation">http://ekkachai:9700/orabpel/HotelBrokerWS/HotelBrokerWS?wsdl</property>
      </partnerLinkBinding>
```

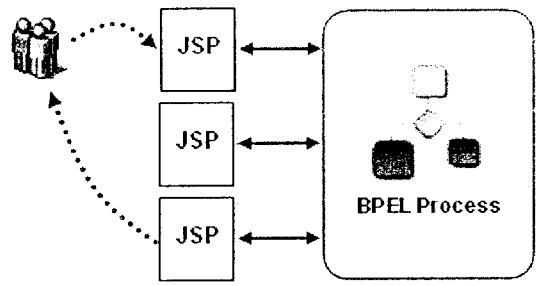
```
                    <partnerLinkBinding name="CarRentalBrokerWS">
                      <property
name="wsdlLocation">http://ekkachai:9700/orabpel/CarRentalBrokerWS/CarRentalBrokerWS?wsdl</property>
                    </partnerLinkBinding>
                    <partnerLinkBinding name="CESservice">
                      <property
name="wsdlLocation">http://ekkachai:9700/orabpel/default/CESservice/CESservice?wsdl</property>
                    </partnerLinkBinding>
                  </partnerLinkBindings>
                </BPELProcess>
              </BPELSuitcase>
```

5. พัฒนาโปรแกรมประยุกต์เพื่อทดสอบแบบจำลององค์ประกอบตัวบริการเว็บ

หลังจากทำการพัฒนาแบบจำลององค์ประกอบตัวบริการเว็บเสร็จเรียบร้อยแล้ว เพื่อเป็นการทดสอบแบบจำลองดังกล่าว จึงได้ทำการพัฒนาโปรแกรมประยุกต์เพื่อใช้เป็น User Interface ใช้ภาษา JSP ในการพัฒนา โดยมีขั้นตอนการพัฒนาดังนี้

5.1 สร้าง User Interface

สร้าง User Interface เพื่อเป็นตัวกลางในการประสานงานรับค่าและแสดงรายการข้อมูลระหว่างผู้ใช้และ BPEL Process ดังแสดงในภาพที่ 28



ภาพที่ 28 แผนภาพการทำงานของ User Interface

ในการสร้าง User Interface ประกอบไปด้วยไฟล์ต่าง ๆ ดังนี้

(1) TravelAgentUI\index.jsp เป็นหน้าเริ่มต้นสำหรับแสดงรายการบริการต่าง ๆ ลูกค้าสามารถเลือกได้ว่าจะใช้บริการใดบ้าง

(2) AirlineBrokerWSUI\bookFlight.jsp สำหรับให้ลูกค้าเลือกต้นทาง ปลายทาง วันที่เดินทาง และจำนวนผู้โดยสาร สำหรับการจองเที่ยวบิน

(3) AirlineBrokerWSUI\listFlight.jsp สำหรับแสดงรายละเอียดเกี่ยวกับรายการเที่ยวบิน และชั้นโดยสารให้ลูกค้าเลือก

(4) AirlineBrokerWSUI\completeFlight.jsp สำหรับดำเนินการจองตั๋วเครื่องบิน

(5) HotelBrokerWSUI\listHotel.jsp สำหรับแสดงรายละเอียดรายชื่อโรงแรม ประเภทห้องพัก และวันที่ Check-in/out ให้ลูกค้าทำการเลือก

(6) HotelBrokerWSUI\completeHotel.jsp สำหรับดำเนินการจองโรงแรมและห้องพัก

(7) CarRentalWSUI\listCarRental.jsp สำหรับแสดงรายชื่อบริษัทที่ให้บริการรถเช่า และประเภทรถเช่า ในจังหวัดปลายทางที่ลูกค้าเดินทางไป

(8) CarRentalWSUI\completeCarRental.jsp สำหรับดำเนินการจองรถเช่า

(9) CESserviceUI\index.jsp สำหรับแสดงรายละเอียดของค่าใช้จ่ายทั้งหมด เพื่อให้ลูกค้ายืนยันการใช้บริการ

หมายเหตุ : Source Code แสดงอยู่ในภาคผนวก

5.2 Compile และ Deploy ไปยัง Web Server

หลังจากสร้าง User Interface เสร็จต้อง Compile และ Deploy ไปยัง Web Server มีขั้นตอนดังนี้

(1) เปิดโปรแกรม Command Prompt

(2) เข้าไปยังไดเร็กทรอรี่ที่เก็บไฟล์ ใช้คำสั่ง cd D:\OraBPELPM_1\integration\TravelAgent

(3) ใช้คำสั่ง obant

6. เครื่องมือที่ใช้ในการพัฒนาระบบ

6.1 ฮาร์ดแวร์

6.1.1 เครื่องคอมพิวเตอร์เซิร์ฟเวอร์ CPU Xeon 3.0 GHz 1 เครื่อง

6.1.2 เครื่องคอมพิวเตอร์ CPU P4^{HT} 2.8 GHz / RAM 512 MB 1 เครื่อง

6.2 ซอฟต์แวร์

6.2.1 Oracle BPEL Process Manager 10.1.2

6.2.2 Oracle Application Server Containers for J2EE 10g

6.2.3 Oracle Jdeveloper BPEL Designer 10g

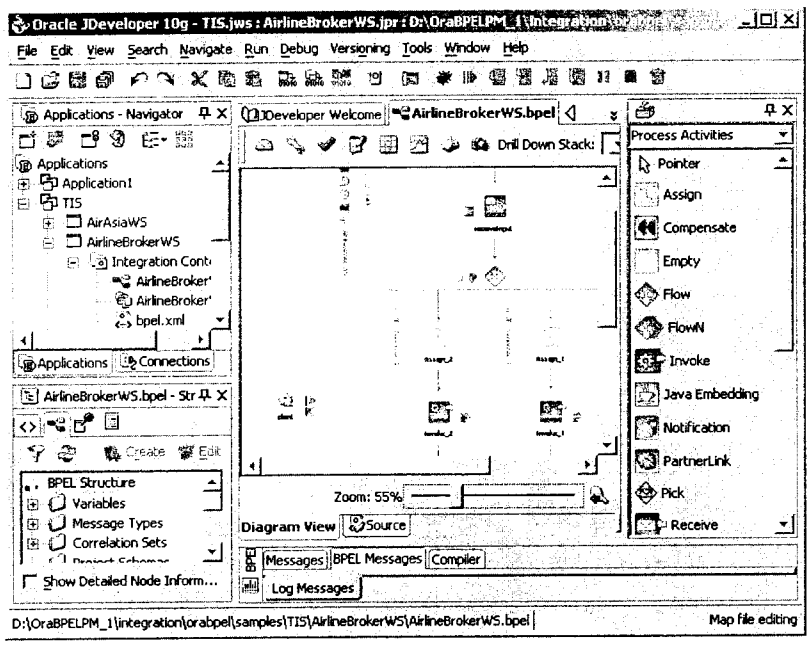
6.2.4 Oracle Lite Database

6.2.5 nusoap V0.6.7

6.2.6 Macromedia DreamWeaver MX 2004

บทที่ 4
ผลการวิจัย

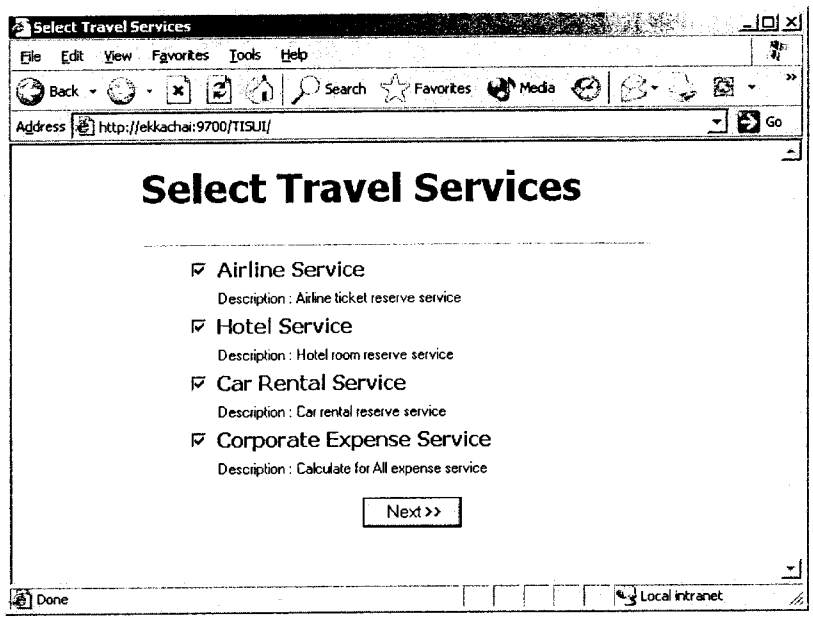
ผลการวิจัยในครั้งนี้ ในขั้นตอนแรกได้ทำการจำลองตัวบริการเว็บของบริษัทผู้ให้บริการจองตั๋วเครื่องบิน บริการโรงแรม และบริการรถเช่า เพื่อประกอบการสร้างแบบจำลององค์ประกอบตัวบริการเว็บ (Web Service Composition Model) โดยทำให้ตัวบริการเว็บของแต่ละบริการสามารถสื่อสารข้อมูลและทำงานร่วมกันแบบอัตโนมัติได้ พัฒนาขึ้นจากภาษา BPEL4WS ใช้โปรแกรม Oracle JDeveloper BPEL Designer [20] เป็นเครื่องมือในการพัฒนาระบบ ดังภาพที่ 29 และใช้ภาษา JSP เพื่อสร้างเป็น User Interface สำหรับทดสอบระบบ โดยมีรายละเอียดของผลการทดสอบดังต่อไปนี้



ภาพที่ 29 หน้าจอของโปรแกรม Oracle JDeveloper BPEL Designer

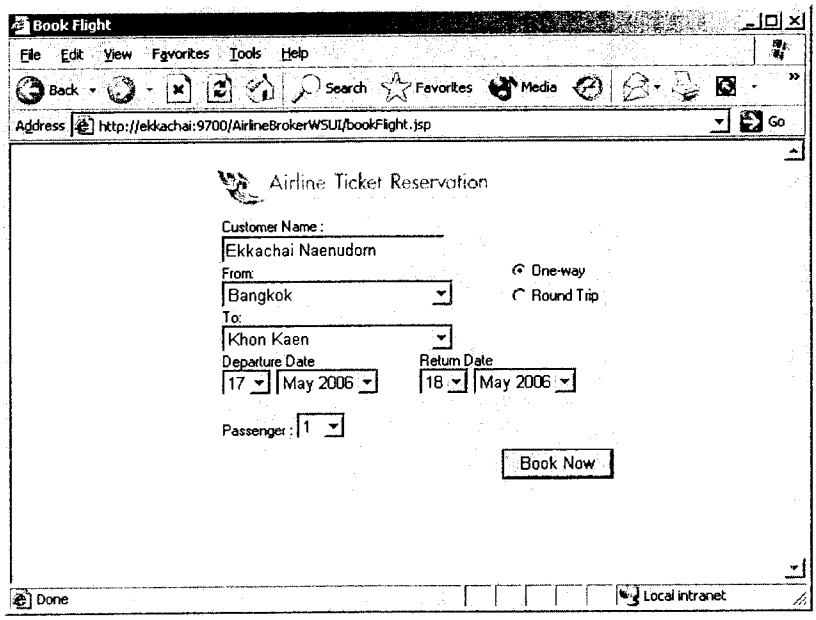
1. ทดสอบระบบธุรกิจการท่องเที่ยวแบบครบวงจรโดยวิธีแบบจำลององค์ประกอบตัวบริการเว็บผ่านทาง User Interface

1.1 เมื่อเข้าสู่ขั้นตอนแรกระบบจะให้ลูกค้าเลือก Services ที่ต้องการใช้บริการ ซึ่งลูกค้าสามารถที่จะไม่เลือกในบางบริการได้ จากตัวอย่างนี้ได้ทำการเลือกใช้บริการในทุกบริการ ดังภาพที่ 30



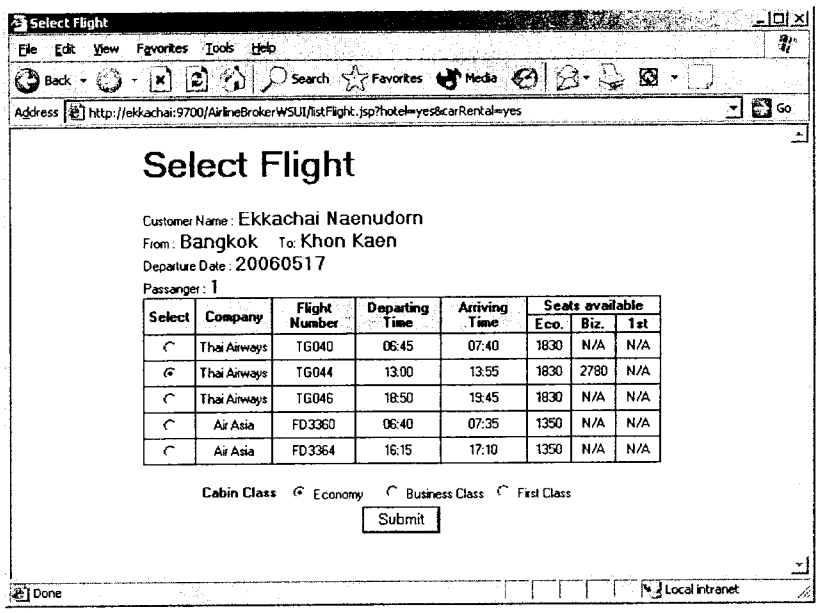
ภาพที่ 30 Select Travel Services : User Interface

1.2 เข้าสู่บริการการจองตั๋วเครื่องบิน ลูกค้าจะต้องกรอกชื่อสกุล เลือกสถานีต้นทางและสถานีปลายทางที่จะเดินทางไป เลือกวันที่เดินทาง และจำนวนที่นั่ง จากตัวอย่างนี้ชื่อสกุลคือ Ekkachai Naenudorn สถานีต้นทางคือ Bangkok สถานีปลายทางคือ Khon Kaen วันที่เดินทางคือ 17 May 2006 และจำนวนที่นั่งคือ 1 ที่นั่ง ดังภาพที่ 31



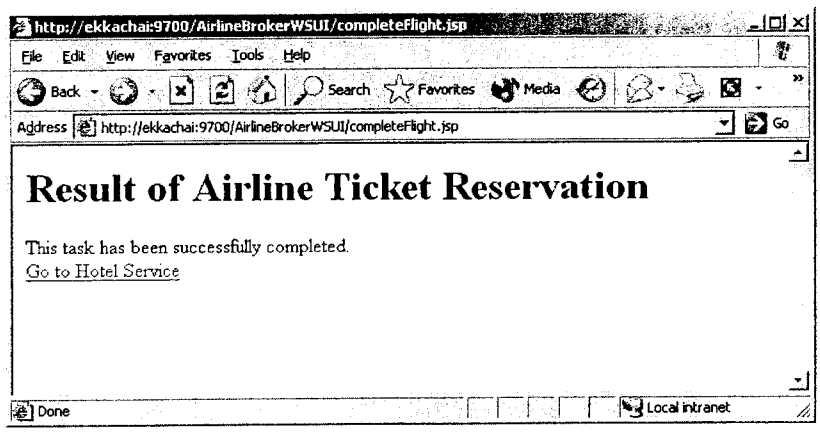
ภาพที่ 31 Airline Ticket Reservation : User Interface

1.3 ระบบจะแสดงรายการเที่ยวบินทั้งหมดให้ลูกค้าเลือกเที่ยวบิน ประเภทของชั้นโดยสาร และราคาของแต่ละชั้นโดยสาร จากตัวอย่างนี้แสดงข้อมูลเที่ยวบินจำนวน 5 รายการ ซึ่งเป็นของสายการบิน Thai Airways 3 รายการ และของสายการบิน Air Asia 2 รายการ ดังภาพที่ 32



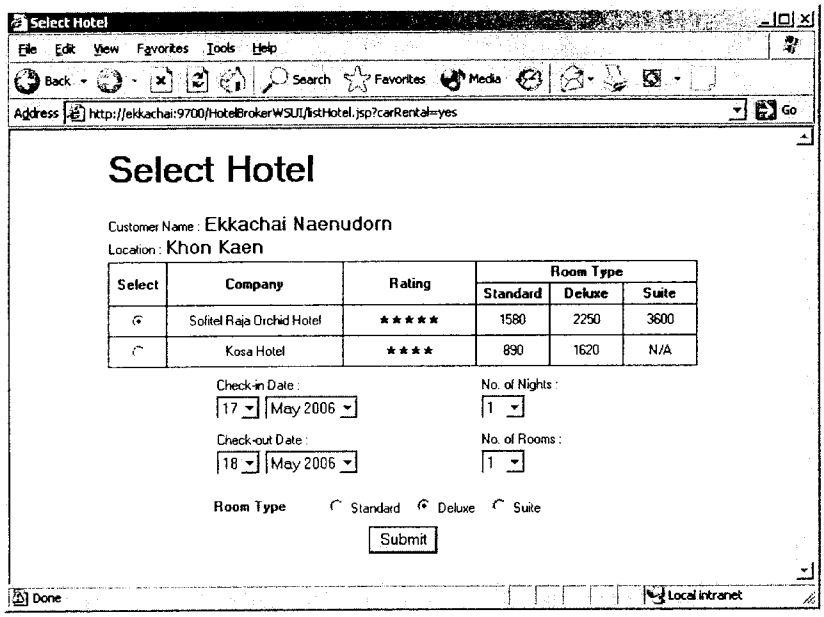
ภาพที่ 32 Select Flight : User Interface

1.4 ระบบทำการจองตั๋วเครื่องบินเรียบร้อยแล้ว คลิ๊กที่ Go to Hotel Service เพื่อเข้าสู่บริการถัดไปคือบริการโรงแรม ดังภาพที่ 33



ภาพที่ 33 Result of Airline Ticket Reservation : User Interface

1.5 เข้าสู่บริการการจองโรงแรมและห้องพัก มาถึงบริการนี้ระบบสามารถนำเสนอรายการของโรงแรมตามสถานีปลายทางที่ลูกค้าจะเดินทางไปได้ทันที เพราะค่าข้อมูลสถานีปลายทางถูกส่งจากบริการแรกมายังบริการจองโรงแรมโดยอัตโนมัติแล้ว ซึ่งเป็นกระบวนการธุรกิจ (Business Process) ที่เกิดขึ้นภายในระบบ จากตัวอย่างนี้ระบบทำการค้นหาโรงแรมที่อยู่ในจังหวัดขอนแก่นแล้วนำมาแสดงโดยพบข้อมูลทั้งหมดจำนวน 2 รายการ คือ Sofitel Raja Orchid Hotel และ Kasa Hotel จากนั้นลูกค้าต้องทำการเลือกโรงแรมที่จะเข้าพัก จำนวนวันที่เข้าพัก จำนวนห้อง และเลือกประเภทห้องพัก ดังภาพที่ 34

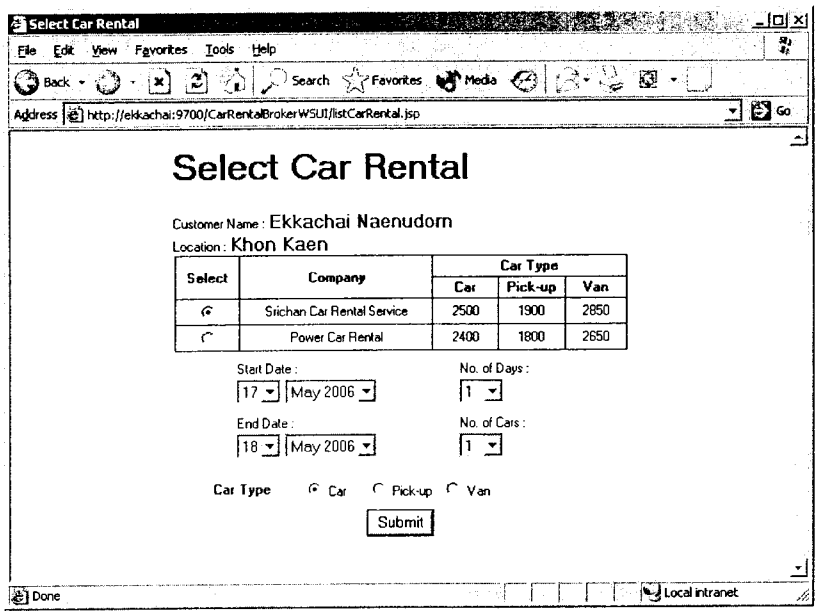


ภาพที่ 34 Select Hotel : User Interface

1.6 ระบบทำการจองโรงแรมและห้องพักเสร็จเรียบร้อย คลิ๊กที่ Go to Car Rental Service เพื่อเข้าสู่บริการจองรถเช่าถัดไป ดังภาพที่ 35

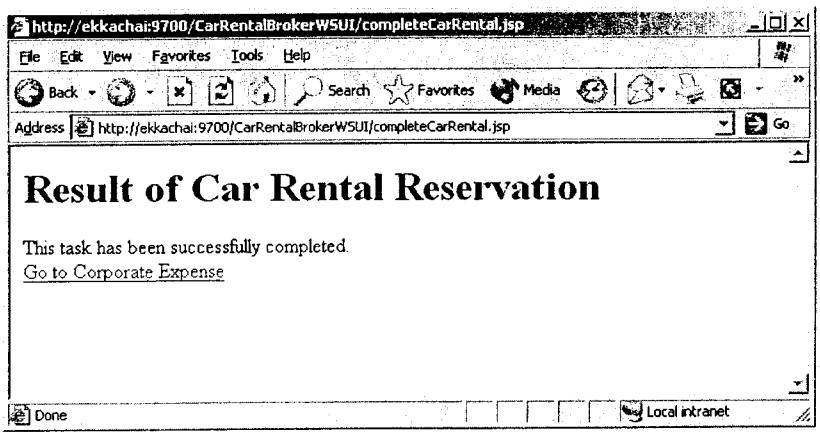
ภาพที่ 35 Result of Hotel Reservation : User Interface

1.7 เข้าสู่บริการจองรถเช่า ระบบจะแสดงรายการรถเช่าของแต่ละบริษัทที่อยู่ในจังหวัดปลายทางที่ลูกค้าเดินทางมาให้ลูกค้าเลือก โดยได้รับข้อมูลปลายทาง เวลาที่มาถึง และชื่อโรงแรมที่เข้าพัก มาจากบริการก่อนหน้านี้โดยอัตโนมัติแล้ว จากตัวอย่างนี้ระบบทำการแสดงข้อมูลรายการรถเช่าจำนวน 2 รายการ คือ บริษัท Srichan Car Rental Service และบริษัท Power Car Rental จากนั้นลูกค้าต้องทำการเลือกบริษัทรถเช่า จำนวนวันที่เช่า จำนวนคัน และประเภทของรถเช่า ดังภาพที่ 36



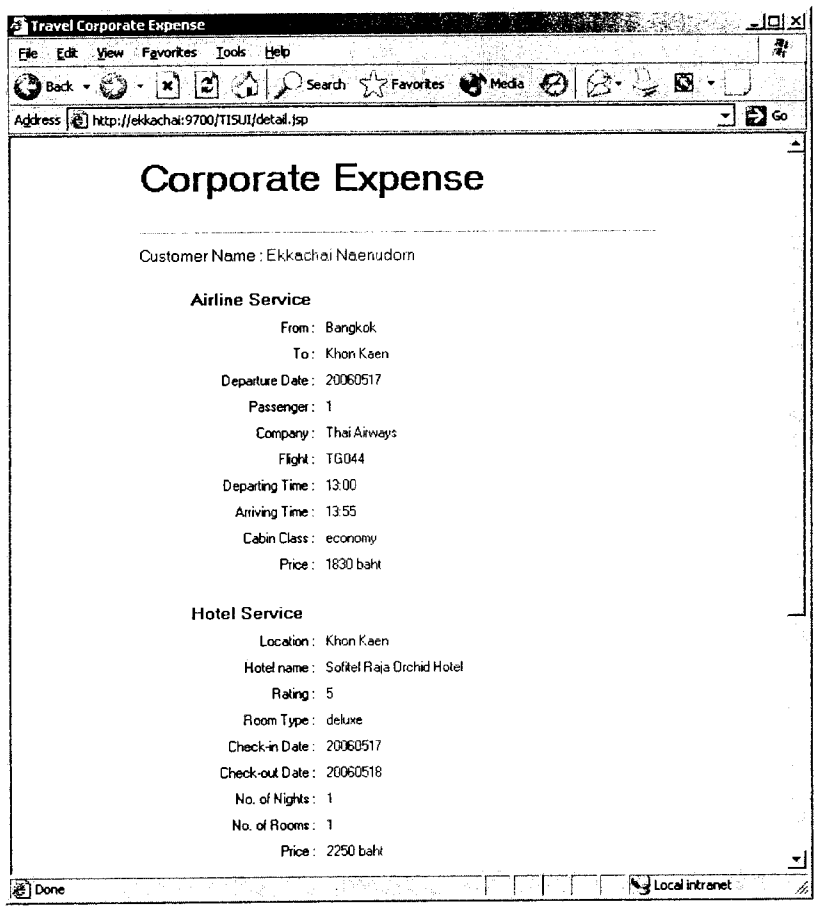
ภาพที่ 36 Select Car Rental : User Interface

1.8 ระบบทำการจองรถเช่าเรียบร้อยแล้ว คลิ๊กที่ Go to Coporate Expense เพื่อเข้าสู่บริการถัดไปคือบริการจัดการค่าใช้จ่าย ดังภาพที่ 37



ภาพที่ 37 Result of Car Rental Reservation : User Interface

1.9 ขั้นตอนสุดท้าย Corporate Expense Service ทำการสรุปค่าใช้จ่ายที่เกิดขึ้นทั้งหมดในระบบ และแสดงรายละเอียดของรายการที่ลูกค้าได้เลือกไว้ เพื่อให้ลูกค้ายืนยันการใช้บริการ ดังภาพที่ 38



ภาพที่ 38 แสดงรายการทั้งหมดเพื่อให้ลูกค้ายืนยันการใช้บริการ

2. ผลการทดสอบแบบจำลององค์ประกอบตัวบริการเว็บ

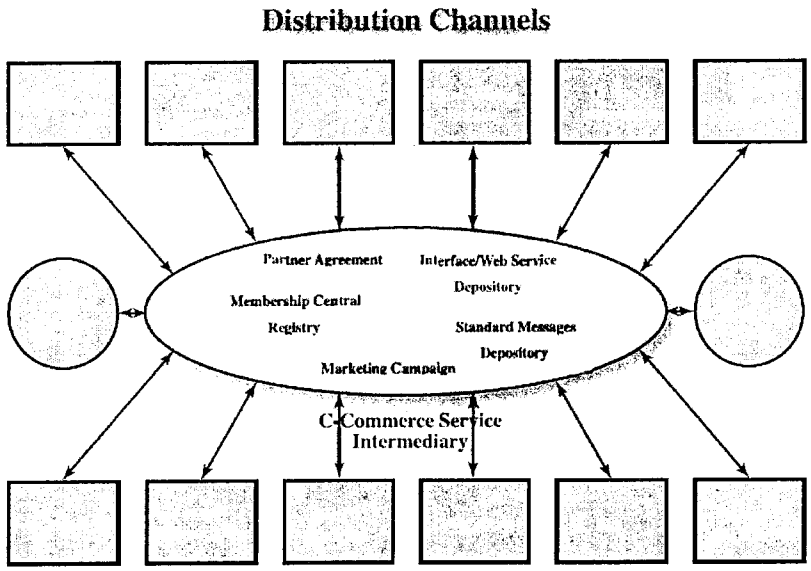
จากผลการทดสอบแบบจำลององค์ประกอบตัวบริการเว็บของธุรกิจด้านการท่องเที่ยวผ่านทาง User Interface นั้น ตัวบริการเว็บของแต่ละธุรกิจบริการสามารถทำงานร่วมกันแบบอัตโนมัติ สามารถรับส่งค่าระหว่างตัวบริการเว็บได้อย่างสมบูรณ์ ไม่ว่าจะเป็นการทำงานระหว่าง Web Service Providers ใน Level 2 และระหว่าง Web Service Brokers ใน Level 1

จากผลการทดสอบดังกล่าวทำให้เห็นได้ว่า แบบจำลององค์ประกอบตัวบริการเว็บนี้สามารถแก้ปัญหาข้อจำกัดในการทำงานร่วมกัน (Interoperability) และแลกเปลี่ยนข้อมูล (Sharebility) ระหว่างตัวบริการเว็บ ทำให้การทำธุรกิจระหว่างองค์กรเป็นไปได้ง่ายขึ้น อีกทั้งแบบจำลององค์ประกอบตัวบริการเว็บนี้ยังมีความยืดหยุ่น (Flexibility) สามารถรองรับการเพิ่มขึ้นของตัวบริการเว็บของกลุ่มธุรกิจบริการ (Web Service Broker) หรือเพิ่มขึ้นของตัวบริการเว็บของผู้ให้บริการ (Web Service Provider) ที่จะเกิดขึ้นในอนาคตได้

3. การเปรียบเทียบระหว่างงานวิจัยนี้กับระบบ Thai Tourism C-Commerce [21]

3.1 ข้อมูลพื้นฐานของระบบ Thai Tourism C-Commerce

ระบบ Thai Tourism C-Commerce ย่อมาจาก Thai Tourism Collaborative Commerce เป็นการร่วมมือทางการตลาดของกลุ่มผู้ประกอบการธุรกิจการท่องเที่ยว โดยใช้ประโยชน์จากเทคโนโลยีสารสนเทศและการสื่อสาร แลกเปลี่ยนข้อมูลซึ่งกันและกัน (Network Marketing) เพื่อเพิ่มมูลค่าให้เกิดขึ้นกับกลุ่มธุรกิจและเป็นแนวทางในการสร้างความร่วมมือทางธุรกิจ พัฒนาขึ้นโดยสำนักงานส่งเสริมอุตสาหกรรมซอฟต์แวร์แห่งชาติ หรือ SIPA (Software Industry Promotion Agency) โดยจะทำหน้าที่ในการพัฒนาซอฟต์แวร์กลางหรือ "Engine" ที่ทำให้ธุรกิจการท่องเที่ยวต่าง ๆ สามารถเชื่อมโยงและส่งข้อมูลถึงกันได้ ตามภาพแสดงกรอบของความร่วมมือตามหลักการของ C-Commerce ดังภาพที่ 39



ภาพที่ 39 ภาพแสดงกรอบของความร่วมมือตามหลักการของ C-Commerce [21]

โครงการ Thai Tourism C-Commerce ที่ SIPA ได้จัดสร้างขึ้นมานี้ ถือเป็นรูปแบบหนึ่งของ DMS (Destination Management System) โดยมีคุณสมบัติหลัก ดังนี้

- Integration

การเชื่อมต่อกับคู่ค้า เพื่อให้เกิดการรวมตัวของกลุ่ม Destination Representatives เช่น กลุ่มสมาคมฯ กลุ่มผู้ประกอบการ โดยเฉพาะกลุ่ม SMEs เพื่อทำธุรกิจในรูปแบบ B2C และ B2B

- Distribution

การช่วยให้ผู้ขายหรือผู้ให้บริการ รวมทั้ง Agents สามารถเข้าถึง Supply Chain ของสินค้าท่องเที่ยว และบริการในลักษณะ Personalize ได้ รวมทั้งเป็นเวทีในการทำ Destination Management และ Marketing ได้

- Promotion

การช่วยให้ผู้บริโภคได้ข้อมูลที่สมบูรณ์และถูกต้อง เพื่อจะได้เตรียมโปรแกรมท่องเที่ยวได้อย่างเหมาะสม โดยผู้ให้บริการสามารถร่วมมือกันหาลูกค้า หรือแนะนำธุรกิจที่เกี่ยวกับการท่องเที่ยวให้กันและกัน

3.2 รูปแบบของ Thai Tourism C-Commerce

สำนักงานส่งเสริมอุตสาหกรรมซอฟต์แวร์แห่งชาติ (องค์การมหาชน) หรือ SIPA ได้ริเริ่มโครงการนี้ด้วยความร่วมจากบริษัทซอฟต์แวร์ โดยมีวัตถุประสงค์ในการส่งเสริมให้กลุ่ม SMEs ได้ใช้ ICT เพื่อสร้างศักยภาพการแข่งขัน อีกทั้งขยายตลาดและสร้างรายได้ สำหรับโครงการ Thai Tourism C-Commerce นี้ได้นำเทคโนโลยี ICT ที่ทันสมัยสองชุดมาประยุกต์ ชุดแรกเป็นการใช้เทคโนโลยี IPTV มาเป็นสื่อการเผยแพร่ข้อมูลเกี่ยวกับสินค้าและบริการด้านท่องเที่ยวผ่านระบบสื่อสารดาวเทียม และเทคโนโลยีชุดที่สองเป็นการนำเทคโนโลยีเว็บเซอร์วิส มาเป็นตัวเชื่อมโยงทำให้ผู้ประกอบการสามารถร่วมมือกันทำรายการในลักษณะ C-Commerce ผ่านเครือข่ายอินเทอร์เน็ตได้

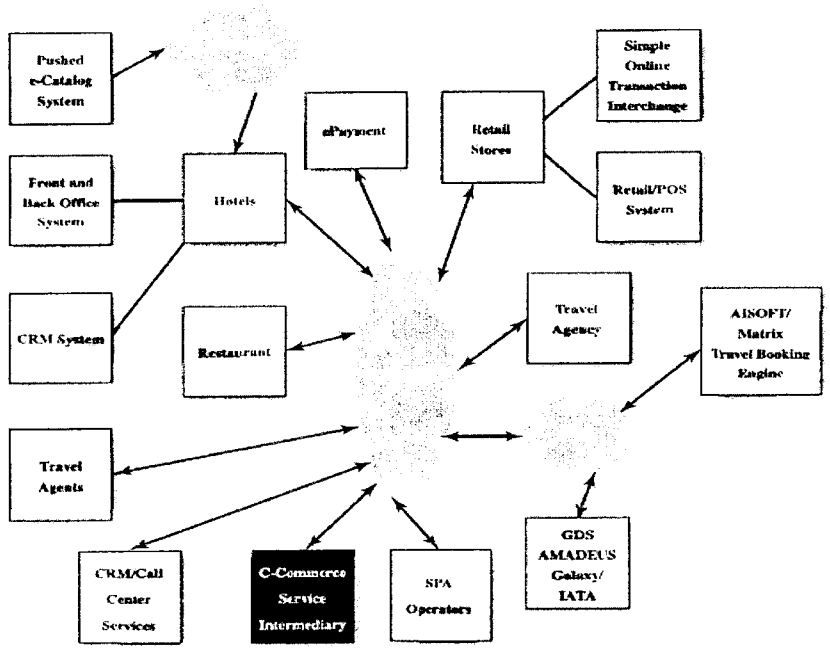
3.2.1 การเผยแพร่ Tourism Product Information ผ่านระบบ IPTV

เป็นการนำเทคนิคการแพร่ข้อมูลข่าวสารผ่านดาวเทียม อาศัยมาตรฐานการสื่อสารข้อมูลแบบ Internet Protocol (IP) ผู้ประกอบการจะเตรียมข้อมูลประชาสัมพันธ์สินค้าและบริการของตนในรูปแบบ e-Catalog ประกอบด้วยตัวหนังสือ ภาพนิ่ง และหรือภาพเคลื่อนไหว ส่งไปเก็บบันทึกในฐานข้อมูลของศูนย์บริการเครื่องคอมพิวเตอร์ของศูนย์บริการจะนำส่งข้อมูลดังกล่าวผ่านดาวเทียมไปบันทึกไว้ในคอมพิวเตอร์ของผู้รับปลายทาง ซึ่งส่วนใหญ่จะเป็นโรงแรมหรือสถานบริการที่พัก คอมพิวเตอร์ปลายทางจะทำหน้าที่แพร่ข้อมูลผ่านระบบแลน (Local Area Network : LAN) ซึ่งจะนำส่งไปยังกลุ่มเครื่องโทรทัศน์ตามห้องพัก ห้องโถง ห้องจัดเลี้ยง ห้องอาหาร และอื่น ๆ ภายใต้การควบคุมของเครื่อง Set top box ทำให้รับภาพที่เป็นข่าวสารและสาระอื่น ๆ ได้หลายช่อง

3.2.2 ผู้ให้บริการ (Service Provider)

(1) Booking Engine Service Provider ได้แก่ ธุรกิจให้บริการด้าน Booking Engine

(2) Information Aggregator and Distribution Service Provider ได้แก่ ธุรกิจที่ให้บริการรวบรวมและกระจายข่าวสารด้านท่องเที่ยวไปสู่ระบบ IPTV ผ่านระบบดาวเทียม



ภาพที่ 40 ภาพแสดงการเชื่อมต่อของธุรกิจท่องเที่ยวต่าง ๆ เข้าเครือข่าย C-Commerce [21]

(3) e-Catalog Design and Development Service Provider ได้แก่ ธุรกิจที่รับออกแบบและสร้าง e-Catalog เพื่อเผยแพร่ผ่านระบบ IPTV

(4) CRM/Call Center Service Provider ได้แก่ ธุรกิจที่ให้บริการด้าน CRM และ Call Center

(5) Application Service Provider (ASP) ได้แก่ ศูนย์บริการให้เช่าใช้ซอฟต์แวร์ในรูปบริการใช้ Hosting และ ASP ซอฟต์แวร์ที่ให้เช่าใช้มีตั้งแต่ระบบบริหารโรงแรม บริหารธุรกิจ Spa บริการร้านอาหารและอื่นๆ

(6) C-Commerce Service Intermediary ได้แก่ ศูนย์กลางบริการจัดการเบื้องต้นของ C-Commerce เช่น ศูนย์ทะเบียนกลางของสมาชิกที่เข้าร่วมเครือข่าย

3.2.3 เทคโนโลยีของระบบ Thai Tourism C-Commerce

SIPA จะเป็นตัวกลางในการประสานงานระหว่างบริษัทผู้พัฒนาซอฟแวร์ และผู้ประกอบการธุรกิจท่องเที่ยว ให้ร่วมกันสร้างเครือข่าย กำหนดขั้นตอนและรูปแบบการทำธุรกิจร่วมกัน ด้วยการร่วมมือของกลุ่มพันธมิตร ดำเนินการพัฒนาระบบซอฟต์แวร์เพื่อเป็นศูนย์กลางในการบริการเชื่อมต่อเครือข่ายเพื่อการร่วมค้าในธุรกิจท่องเที่ยว (Thai Tourism C-Commerce Service Intermediary) ในชื่อ www.thaitourismccommerce.com ด้วย Java Technology และ Mircosoft Technology โดยระบบจะถูกออกแบบด้วยการใช้รูปแบบข้อมูลมาตรฐาน (Universal Data Model) และอาศัยมาตรฐานของ XML ร่วมกับการใช้เทคโนโลยีของ WSDL และเว็บเซอร์วิส

3.2.4 Business Model

ผู้ที่เกี่ยวข้องกับการสนับสนุนโครงการ Tourism C-Commerce มีหลากหลายกลุ่มอันเป็นแนวคิดที่นำเอาทรัพยากรของแต่ละกลุ่มมาร่วมมือกัน ทำให้เกิดการทำธุรกรรมแบบ C-Commerce เพื่อร่วมกันพัฒนาอุตสาหกรรมท่องเที่ยว และอุตสาหกรรมซอฟต์แวร์ของประเทศ รูปแบบการทำธุรกิจของเครือข่าย C-Commerce จึงไม่มีรูปแบบคงที่ ไม่มีหน่วยงานใดเป็นเจ้าของหรือหุ้นส่วนใหญ่ ด้วยเหตุนี้รูปแบบการทำธุรกิจร่วมกันหรือ Business Model ของระบบ Tourism C-Commerce จึงสรุปย่อๆ ได้ดังนี้

(1) Self support, share revenue, share cost and expenses

(2) Service providers and commercial fee based entities

(3) Price and fee are set at market prices

3.3 การเปรียบเทียบประสิทธิภาพระหว่าง WSCM และ TCC

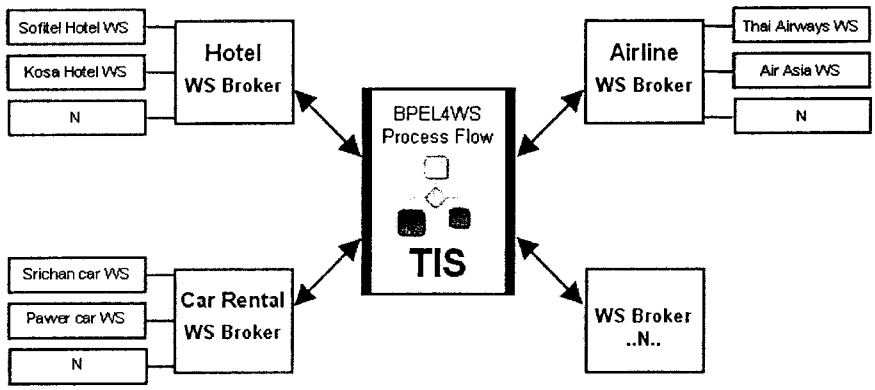
การเปรียบเทียบประสิทธิภาพของธุรกิจการท่องเที่ยวระหว่างการใช้เทคโนโลยี Web Service Composition Model (WSCM) และเทคโนโลยีของ Thai Tourism C-Commerce (TCC) ทั้งสองเทคโนโลยีนี้มีวัตถุประสงค์ที่คล้ายกัน คือต้องการทำให้กลุ่มธุรกิจด้านการท่องเที่ยวสามารถเชื่อมโยงข้อมูลและแลกเปลี่ยนข้อมูลร่วมกันได้ แต่ก็มีรายละเอียดบางส่วนที่แตกต่างกันโดยสรุปเปรียบเทียบข้อดีและข้อเสียได้เป็นประเด็นดังตารางที่ 5 ดังนี้

ตารางที่ 5 การเปรียบเทียบประสิทธิภาพระหว่าง WSCM และ TCC

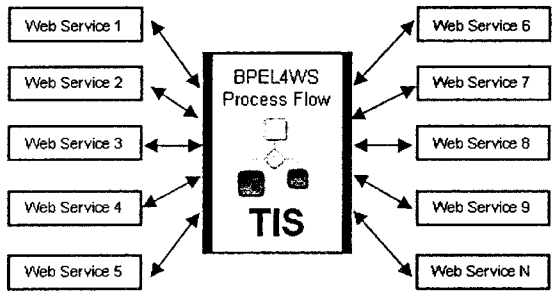
เทคโนโลยี WSCM	เทคโนโลยี TCC
1. มีการออกแบบสถาปัตยกรรมของกระบวนการทางธุรกิจ และบอกถึงลำดับการทำงานในแต่ละส่วนอย่างชัดเจน	1. การออกแบบกระบวนการทางธุรกิจยังไม่ครอบคลุม แต่มีการกำหนดหน้าที่ในแต่ละองค์ประกอบได้อย่างชัดเจน
2. ใช้ภาษาที่มีประสิทธิภาพในการสร้างกระบวนการทางธุรกิจอย่างภาษา BEPL4WS ซึ่งเป็นการรวมเอาข้อดีของภาษา WSFL ของ IBM และภาษา XLang ของ Microsoft เข้าไว้ด้วยกัน	2. ใช้เทคนิค Java Technology และ Microsoft Technology เพื่อพัฒนาเป็นระบบซอฟต์แวร์กลาง โดย SIPA จะเป็นผู้เริ่มต้นพัฒนาในระยะแรก
3. ในขั้นตอนแรก ลูกค้าสามารถเลือกใช้บริการธุรกิจการท่องเที่ยวในบางบริการได้ ทำให้ลูกค้าสะดวกและประหยัดเวลาในการใช้บริการ	3. ลูกค้าต้องเสียเวลาในการค้นหาบริการและในการเลือกใช้บริการด้านการท่องเที่ยว
4. ทำงานอยู่ในรูปแบบสถาปัตยกรรม SOA ร้อยเปอร์เซ็นต์ (Pure-SOA)	4. มีบางส่วนของระบบที่เป็น Web Application เช่น ในการลงทะเบียนผู้ประกอบการ
5. WSCM ทำให้ลดเวลาและค่าใช้จ่าย ทางด้านผู้ให้บริการเป็นการเพิ่มช่องทางการขายและการตลาด ทางด้านผู้ใช้บริการได้รับบริการอย่างครบวงจรและรวดเร็วเพราะมีบริการแบบเสร็จสรรพในจุดเดียว	5. มีค่าใช้จ่ายสูง เช่น ในการทำระบบ IPTV เพื่อเผยแพร่ Tourism Product Information ผ่านดาวเทียม

4. การเปรียบเทียบแบบจำลององค์ประกอบตัวบริการเว็บกรณีที่ใช้ Web Service Broker และไม่ใช้ Web Service Broker

ตัวแทนตัวบริการเว็บหรือ Web Service Broker ทำหน้าที่เป็นตัวแทนในการเรียกใช้บริการระหว่างระบบ TIS และตัวบริการเว็บของผู้ให้บริการต่าง ๆ (Web Service Provider) หลักการทำงานของตัวแทนตัวบริการเว็บเมื่อได้รับการร้องขอจากระบบ TIS ตัวแทนตัวบริการเว็บก็จะทำการเรียกไปยังตัวบริการเว็บในกลุ่มของตนเอง ตัวอย่างเช่น เมื่อตัวแทนบริการโรงแรม (Hotel Broker Web Service) ได้รับการร้องของจากระบบ TIS โดยรับค่าข้อมูลสถานีปลายทาง (Destination) เช่น ขอนแก่น จากนั้นตัวแทนบริการโรงแรมก็จะส่งข้อมูลต่อไปยัง Sofitel hotel web service และ Kosa hotel Web Service เพื่อทำการค้นหาบริการโรงแรมที่อยู่ในจังหวัดขอนแก่น จากนั้นตัวแทนบริการโรงแรมก็จะทำการรวบรวมรายการโรงแรมและประเภทห้องพักมานำเสนอให้ลูกค้าเลือกต่อไป



ภาพที่ 41 ภาพแสดงแบบจำลององค์ประกอบตัวบริการเว็บกรณีที่ใช้ Web Service Broker



ภาพที่ 42 ภาพแสดงแบบจำลององค์ประกอบตัวบริการเว็บกรณีที่ไม่ใช้ Web Service Broker

จากการเปรียบเทียบ (ดังภาพที่ 41 และ 42) จะเห็นได้ว่า ในกรณีที่สร้างระบบธุรกิจการท่องเที่ยวแบบครบวงจรโดยใช้องค์ประกอบตัวบริการเว็บในกรณีที่ไม่ใช้ตัวแทนตัวบริการเว็บหรือ Web Service Broker จะทำให้ตัวบริการเว็บของผู้ให้บริการอยู่กันอย่างกระจัดกระจายเพราะไม่มีการจัดกลุ่มบริการ แบบจำลองไม่มีความยืดหยุ่นทำให้เกิดปัญหาในการเพิ่มขึ้นของตัวบริการเว็บและเกิดปัญหาในขั้นตอนการสร้างองค์ประกอบตัวบริการเว็บ และในการเรียกใช้บริการในแต่ละบริการมีความล่าช้าเพราะตัวบริการเว็บของผู้ให้บริการมีจำนวนมากและติดต่อกับระบบ TIS เองโดยตรง ซึ่งแตกต่างกับแบบจำลององค์ประกอบตัวบริการเว็บที่มี Web Service Broker ซึ่งมีการจัดกลุ่มบริการไว้อย่างชัดเจน และมีความยืดหยุ่นสามารถรองรับกับการเพิ่มขึ้นของตัวบริการเว็บของกลุ่มธุรกิจบริการ (Web Service Broker) และการเพิ่มขึ้นของตัวบริการเว็บของผู้ให้บริการ (Web Service Provider) ที่จะเกิดขึ้นในอนาคตได้

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

งานวิจัยนี้ได้นำเสนอแบบจำลองของการทำงานร่วมกันของแต่ละตัวบริการเว็บ ซึ่งประกอบด้วยกลุ่มธุรกิจการจองตั๋วเครื่องบิน ธุรกิจโรงแรม ธุรกิจรถเช่า และธุรกิจด้านการเงิน โดยมีตัวกลางหรือ Tourism Information System: TIS ทำหน้าที่ประสานงานในการทำงานร่วมกันโดยใช้ภาษาที่ใช้สร้างกระบวนการทางธุรกิจคือภาษา BPEL (Business Process Execution Language) สามารถสรุปประเด็นและข้อเสนอแนะได้ดังนี้

1. สรุปผลการศึกษาวิจัย

จากผลการสร้างแบบจำลองของการทำงานร่วมกันของแต่ละเว็บเซอร์วิสธุรกิจท่องเที่ยว ด้วยชุดโปรแกรมของ Oracle สามารถสนับสนุนการสร้างกระบวนการทางธุรกิจ โดยใช้ภาษา BPEL ได้อย่างสมบูรณ์ ผลจากการทดลองสรุปเป็นประเด็นที่สำคัญ ดังนี้

1.1 เป็นการสร้างกระบวนการทางธุรกิจ เพื่อทำให้เว็บเซอร์วิสของธุรกิจท่องเที่ยวต่าง ๆ สามารถติดต่อสื่อสารและแลกเปลี่ยนข้อมูลร่วมกันได้แบบอัตโนมัติ

1.2 ได้แบบจำลององค์ประกอบตัวบริการเว็บของธุรกิจท่องเที่ยวแบบครบวงจร เพื่อนำไปพัฒนาและปรับปรุงแอพพลิเคชั่นรองรับกับธุรกิจที่จะเกิดเพิ่มขึ้นในอนาคต เพื่อให้ได้แบบจำลององค์ประกอบตัวบริการเว็บที่สมบูรณ์ต่อไป

1.3 สามารถนำแบบจำลององค์ประกอบตัวบริการเว็บไปประยุกต์ใช้กับการบริการการท่องเที่ยวจริงในปัจจุบันได้

1.4 แบบจำลององค์ประกอบตัวบริการเว็บนี้สามารถรองรับกับกลุ่มธุรกิจหรือตัวบริการเว็บอื่น ๆ ที่อาจเพิ่มขึ้นในอนาคตได้

2. ข้อเสนอแนะ

จากงานวิจัยนี้ทำให้ได้แบบจำลององค์ประกอบตัวบริการเว็บของธุรกิจท่องเที่ยว แต่ยังพบว่ายังขาดองค์ประกอบอีกบางส่วน เช่น การให้บริการทางด้านการชำระเงิน อาจจะต้องเพิ่มระบบตัดเงินผ่านบัตรเครดิตอัตโนมัติ สนับสนุนการเพิ่มขึ้นของผู้ร่วมทำธุรกิจในอนาคตได้แบบอัตโนมัติ เพิ่มในส่วนของระบบ E-Tickets เพิ่มระบบ Web Services Security เพื่อรักษาความปลอดภัยของข้อมูล และเพิ่มระบบ AI ในการวิเคราะห์และเลือกทางเลือกที่ดีที่สุดมานำเสนอต่อลูกค้า เพื่อให้การบริการการท่องเที่ยวแบบครบวงจรตอบสนองความต้องการของลูกค้าได้สมบูรณ์มากยิ่งขึ้น

เอกสารอ้างอิง

[1] W3C-WS. World Wide Web Consortium: Web Service. [online] 2002 [cited 2005 Jun]. Available from: http://www.w3.org/2002/ws/

[2] W3C-XML. World Wide Web Consortium: Extensible Markup Language (XML) version 1.0 (Second Edition). [online] 2000 Oct 6 [cited 2005 May]. Available from: http://www.w3.org/XML/

[3] W3C-SOAP. World Wide Web Consortium: Simple Object Access Protocol version 1.2 Part 0. [online] 2003 Jun 24 [cited 2005 Jun]. Available from: http://www.w3.org/TR/soap12-part0/

[4] E. Christensen, F. Curbera, G. Meredith, S. Weerawarana. Web Services Description Language (WSDL) version 1.1. [online] 2001 Mar 15 [cited 2005 Jun]. Available from: http://www.w3.org/TR/wsdl

[5] UDDI. Universal Description Discovery and Integration. [online] 2002 Jul 30 [cited 2005 Jul]. Available from: http://www.uddi.org/

[6] Microsoft SOAP Toolkit. [online] 2002 [cited 2005 Jul]. Available from: http://msdn.microsoft.com/webservices/building/soaptk/

[7] Sun Open Net Environment. [online] 2002 [cited 2005 Jul]. Available from: http://www.sun.com/software/sunone/

[8] Web Services Toolkit. [online] 2003 [cited 2005 Jul]. Available from: http://www.alphaworks.ibm.com/tech/webservicestoolkit/

[9] WebSphere Studio Application Developer. [online] 2002 [cited 2005 Aug]. Available from: http://www-306.ibm.com/software/awdtools/studioappdev/

[10] BPEL4WS (Business Process Execution Language for Web Services). BPEL4WS version 1.1 [online] 2003 May 5 [cited 2005 Aug]. Available from: http://www-128.ibm.com/developerworks/webservices/library/ws-bpel/

[11] IBM-WSFL, Web Services Flow Language (WSFL). IBM, [online] 2001 [cited 2005 Jul]. Available from: http://www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf

[12] MS-XLANG, S.Thatte: Web Services for Business Process Design Microsoft Corporation. [online] 2001 [cited 2005 Jul]. Available from: http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/

[13] Pierre F. Tiako. **Web-Services Modeling for E-Marketplace**: Proceedings of Symposium on Applications and the Internet Workshops (SAINT 2003), 2003 January 27-31; Orlando, FL, USA. p. 111-115.

[14] M. Benyoucef, R.K.Keller. **A Conceptual Architecture for a Combined Negotiation Support System.** Proceedings of the 11th International Workshop on Database and Expert Systems Applications, 2000 September; Los Alamitos, CA; 2000. p. 1015-1019.

[15] G. Alonso, U. Fiedler, C. Hagen, A. Lazcano, H. Schuldt, N. Weiler. **Business to Business E-Commerce.** Proceeding of the 9th International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises (RIDE-VE' 99), 1999 March 23-24; Sydney, Australia. P. 132.

[16] B. Srivastava, J. Koehler. **Web Service Composition – Current Solutions and Open Problems.** Proceeding of International Conference on Automated Planning and Scheduling (ICAPS2003), 2003 June 9-13; Trento, Italy.

[17] Ankolendar, et al. **DAML Service**. [online] 2002 [cited 2005 Jul]. Available from: http://www.daml.org/services/

[18] Yeung C., Pang-Fei T., Yen J. **A Multi-Agent Based Tourism Kiosk on Internet.** Proceedings of the Thirty-First Annual Hawaii International Conference on System Sciences-Volume 4, 1998 January 6-9; Kohala Coast, Hawaii, USA. p. 452.

[19] Wil M. P. van der Aalst , Marlon Dumas, Arthur H. M. ter Hofstede. **Web Service Composition Languages: Old Wine in New Bottles?.** Proceedings of the 29th Conference on EUROMICRO, 2003 September 01-06; Antalya, Turkey. p. 298.

[20] Oracle BPEL Process Manager. [online] 2003 [cited 2005 Jul]. Available from: http://otn.oracle.com/bpel/

[21] Thai Tourism C-Commerce. [online] 2004 [cited 2005 Jul]. Available from: http://www.thaitourismccommerce.com

ภาคผนวก

ภาคผนวก ก

ตัวอย่างไฟล์ BPEL ของระบบ TIS

1. TIS.bpel

```
<process name="TIS" targetNamespace="http://xmlns.oracle.com/TIS"
xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
xmlns:xp20="http://www.oracle.com/XSL/Transform/java/oracle.tip.pc.services.functions.Xpath20"
xmlns:bpws="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
xmlns:ns4="http://xmlns.oracle.com/CESservice"
xmlns:ns7="http://www.CarRentalBrokerWS.com/ns/sales"
xmlns:ldap="http://schemas.oracle.com/xpath/extension/ldap"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ns5="http://www.AirlineBrokerWS.com/ns/sales" xmlns:client="http://xmlns.oracle.com/TIS"
xmlns:ns6="http://www.HotelBrokerWS.com/ns/sales"
xmlns:ora="http://schemas.oracle.com/xpath/extension" xmlns:ns9="http://www.tis.com/ns/sales"
xmlns:ns1="http://xmlns.oracle.com/AirlineBrokerWS"
xmlns:ns3="http://xmlns.oracle.com/CarRentalBrokerWS"
xmlns:ns2="http://xmlns.oracle.com/HotelBrokerWS"
xmlns:bpelx="http://schemas.oracle.com/bpel/extension"
xmlns:orcl="http://www.oracle.com/XSL/Transform/java/oracle.tip.pc.services.functions.ExtFunc">

 <partnerLinks>
  <partnerLink name="client" partnerLinkType="client:TIS" myRole="TISProvider"
partnerRole="TISRequester"/>
  <partnerLink myRole="AirlineBrokerWSRequester" name="AirlineBrokerWS"
partnerRole="AirlineBrokerWSProvider" partnerLinkType="ns1:AirlineBrokerWS"/>
  <partnerLink myRole="HotelBrokerWSRequester" name="HotelBrokerWS"
partnerRole="HotelBrokerWSProvider" partnerLinkType="ns2:HotelBrokerWS"/>
  <partnerLink myRole="CarRentalBrokerWSRequester" name="CarRentalBrokerWS"
partnerRole="CarRentalBrokerWSProvider" partnerLinkType="ns3:CarRentalBrokerWS"/>
  <partnerLink name="CESservice" partnerRole="CESserviceProvider"
partnerLinkType="ns4:CESservice"/>
 </partnerLinks>

 <variables>
  <variable name="inputVariable" messageType="client:TISRequestMessage"/>
  <variable name="outputVariable" messageType="client:TISResponseMessage"/>
  <variable name="invoke_AB" messageType="ns1:AirlineBrokerWSRequestMessage"/>
  <variable name="receive_AB" messageType="ns1:AirlineBrokerWSResponseMessage"/>
  <variable name="invoke_HB" messageType="ns2:HotelBrokerWSRequestMessage"/>
  <variable name="receive_HB" messageType="ns2:HotelBrokerWSResponseMessage"/>
  <variable name="invoke_CB" messageType="ns3:CarRentalBrokerWSRequestMessage"/>
  <variable name="receive_CB" messageType="ns3:CarRentalBrokerWSResponseMessage"/>
  <variable name="invoke_CES" messageType="ns4:CESserviceRequestMessage"/>
  <variable name="receive_CES" messageType="ns4:CESserviceResponseMessage"/>
 </variables>

 <sequence name="main">
  <receive name="receiveInput" partnerLink="client" portType="client:TIS" operation="initiate"
variable="inputVariable" createInstance="yes"/>
  <scope name="getAirline">
   <switch name="Switch_1">
    <case
condition="bpws:getVariableData('inputVariable','payload','/ns9:TIS/ns9:Services/ns9:airline') =
'yes'">
     <sequence name="Sequence_1">
      <assign name="Assign_AB1">
       <copy>
```

```
          <from variable="inputVariable" part="payload"
query="/ns9:TIS/ns9:AirlineService/ns9:customerName"/>
          <to variable="invoke_AB" part="payload" query="/ns5:ABWS/ns5:customerName"/>
         </copy>
         <copy>
          <from variable="inputVariable" part="payload"
query="/ns9:TIS/ns9:AirlineService/ns9:source"/>
          <to variable="invoke_AB" part="payload" query="/ns5:ABWS/ns5:source"/>
         </copy>
         <copy>
          <from variable="inputVariable" part="payload"
query="/ns9:TIS/ns9:AirlineService/ns9:destination"/>
          <to variable="invoke_AB" part="payload" query="/ns5:ABWS/ns5:destination"/>
         </copy>
         <copy>
          <from variable="inputVariable" part="payload"
query="/ns9:TIS/ns9:AirlineService/ns9:departureDate"/>
          <to variable="invoke_AB" part="payload" query="/ns5:ABWS/ns5:departureDate"/>
         </copy>
         <copy>
          <from variable="inputVariable" part="payload"
query="/ns9:TIS/ns9:AirlineService/ns9:passenger"/>
          <to variable="invoke_AB" part="payload" query="/ns5:ABWS/ns5:passenger"/>
         </copy>
        </assign>
        <invoke name="Invoke_AB" partnerLink="AirlineBrokerWS"
portType="ns1:AirlineBrokerWS" operation="initiate" inputVariable="invoke_AB"/>
        <receive name="Receive_AB" createInstance="no" partnerLink="AirlineBrokerWS"
portType="ns1:AirlineBrokerWSCallback" operation="onResult" variable="receive_AB"/>
        <assign name="Assign_AB2">
         <copy>
          <from variable="receive_AB" part="payload" query="/ns5:ABWS"/>
          <to variable="outputVariable" part="payload" query="/ns9:TIS/ns9:AirlineService"/>
         </copy>
        </assign>
       </sequence>
      </case>
      <otherwise>
       <assign name="Assign_AB3">
        <copy>
         <from expression="0"/>
         <to variable="invoke_CES" part="payload"
query="/ns4:CESserviceProcessRequest/ns4:priceAirline"/>
        </copy>
       </assign>
      </otherwise>
     </switch>
    </scope>
    <scope name="getHotel">
     <switch name="Switch_2">
      <case
condition="bpws:getVariableData('inputVariable','payload','/ns9:TIS/ns9:Services/ns9:hotel')='yes'">
       <sequence name="Sequence_2">
        <assign name="Assign_HB1">
         <copy>
          <from variable="receive_AB" part="payload" query="/ns5:ABWS/ns5:customerName"/>
          <to variable="invoke_HB" part="payload" query="/ns6:HBWS/ns6:customerName"/>
```

```
          </copy>
          <copy>
            <from variable="receive_AB" part="payload" query="/ns5:ABWS/ns5:destination"/>
            <to variable="invoke_HB" part="payload" query="/ns6:HBWS/ns6:destination"/>
          </copy>
        </assign>
        <invoke name="Invoke_HB" partnerLink="HotelBrokerWS" portType="ns2:HotelBrokerWS"
operation="initiate" inputVariable="invoke_HB"/>
        <receive name="Receive_HB" createInstance="no" partnerLink="HotelBrokerWS"
portType="ns2:HotelBrokerWSCallback" operation="onResult" variable="receive_HB"/>
        <assign name="Assign_HB2">
          <copy>
            <from variable="receive_HB" part="payload" query="/ns6:HBWS"/>
            <to variable="outputVariable" part="payload" query="/ns9:TIS/ns9:HotelService"/>
          </copy>
        </assign>
      </sequence>
    </case>
    <otherwise>
      <assign name="Assign_HB3">
        <copy>
          <from variable="receive_AB" part="payload" query="/ns5:ABWS/ns5:customerName"/>
          <to variable="receive_HB" part="payload" query="/ns6:HBWS/ns6:customerName"/>
        </copy>
        <copy>
          <from variable="receive_AB" part="payload" query="/ns5:ABWS/ns5:destination"/>
          <to variable="receive_HB" part="payload" query="/ns6:HBWS/ns6:destination"/>
        </copy>
        <copy>
          <from expression="string( 'N/A' )"/>
          <to variable="receive_HB" part="payload"
query="/ns6:HBWS/ns6:listHotels/ns6:Item/ns6:hotelName"/>
        </copy>
        <copy>
          <from expression="0"/>
          <to variable="receive_HB" part="payload"
query="/ns6:HBWS/ns6:listHotels/ns6:Item/ns6:priceHotel"/>
        </copy>
      </assign>
    </otherwise>
  </switch>
 </scope>
 <scope name="getCarRental">
  <switch name="Switch_3">
   <case
condition="bpws:getVariableData('inputVariable','payload','/ns9:TIS/ns9:Services/ns9:carRental')='yes'
">
     <sequence name="Sequence_3">
      <assign name="Assign_CB1">
        <copy>
          <from variable="receive_HB" part="payload" query="/ns6:HBWS/ns6:customerName"/>
          <to variable="invoke_CB" part="payload" query="/ns7:CRBWS/ns7:customerName"/>
        </copy>
        <copy>
          <from variable="receive_HB" part="payload" query="/ns6:HBWS/ns6:destination"/>
          <to variable="invoke_CB" part="payload" query="/ns7:CRBWS/ns7:destination"/>
        </copy>
```

```
          <copy>
            <from variable="receive_AB" part="payload"
query="/ns5:ABWS/ns5:listFlights/ns5:Item/ns5:arrTime"/>
            <to variable="invoke_CB" part="payload" query="/ns7:CRBWS/ns7:arrTime"/>
          </copy>
          <copy>
            <from variable="receive_HB" part="payload"
query="/ns6:HBWS/ns6:listHotels/ns6:Item/ns6:hotelName"/>
            <to variable="invoke_CB" part="payload" query="/ns7:CRBWS/ns7:hotelName"/>
          </copy>
        </assign>
        <invoke name="Invoke_CB" partnerLink="CarRentalBrokerWS"
portType="ns3:CarRentalBrokerWS" operation="initiate" inputVariable="invoke_CB"/>
        <receive name="Receive_CB" partnerLink="CarRentalBrokerWS"
portType="ns3:CarRentalBrokerWSCallback" operation="onResult" variable="receive_CB"
createInstance="no"/>
        <assign name="Assign_CB2">
          <copy>
            <from variable="receive_CB" part="payload" query="/ns7:CRBWS"/>
            <to variable="outputVariable" part="payload" query="/ns9:TIS/ns9:CarRentalService"/>
          </copy>
        </assign>
      </sequence>
    </case>
    <otherwise>
      <assign name="Assign_CB3">
        <copy>
          <from expression="0"/>
          <to variable="receive_CB" part="payload"
query="/ns7:CRBWS/ns7:listCars/ns7:Item/ns7:priceCar"/>
        </copy>
      </assign>
    </otherwise>
  </switch>
 </scope>
 <scope name="getCES">
  <sequence name="Sequence_4">
    <assign name="Assign_1">
      <copy>
        <from variable="receive_AB" part="payload"
query="/ns5:ABWS/ns5:listFlights/ns5:Item/ns5:priceAirline"/>
        <to variable="invoke_CES" part="payload"
query="/ns4:CESserviceProcessRequest/ns4:priceAirline"/>
      </copy>
      <copy>
        <from variable="receive_HB" part="payload"
query="/ns6:HBWS/ns6:listHotels/ns6:Item/ns6:priceHotel"/>
        <to variable="invoke_CES" part="payload"
query="/ns4:CESserviceProcessRequest/ns4:priceHotel"/>
      </copy>
      <copy>
        <from variable="receive_CB" part="payload"
query="/ns7:CRBWS/ns7:listCars/ns7:Item/ns7:priceCar"/>
        <to variable="invoke_CES" part="payload"
query="/ns4:CESserviceProcessRequest/ns4:priceCar"/>
      </copy>
    </assign>
```

```
    <invoke name="Invoke_CES" partnerLink="CESservice" portType="ns4:CESservice"
operation="process" inputVariable="invoke_CES" outputVariable="receive_CES"/>
    <assign name="Assign_2">
     <copy>
      <from variable="receive_CES" part="payload"
query="/ns4:CESserviceProcessResponse/ns4:total"/>
      <to variable="outputVariable" part="payload" query="/ns9:TIS/ns9:CESService/ns9:total"/>
     </copy>
    </assign>
   </sequence>
  </scope>
  <invoke name="callbackClient" partnerLink="client" portType="client:TISCallback"
operation="onResult" inputVariable="outputVariable"/>
 </sequence>
</process>
```

2. AirlineBrokerWS.bpel

```
<process name="AirlineBrokerWS" targetNamespace="http://xmlns.oracle.com/AirlineBrokerWS"
xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
xmlns:xp20="http://www.oracle.com/XSL/Transform/java/oracle.tip.pc.services.functions.Xpath20"
xmlns:bpws="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
xmlns:ns4="http://www.AirlineBrokerWS.com/ns/sales"
xmlns:ns1="http://xmlns.oracle.com/ThaiAirwayWS"
xmlns:ldap="http://schemas.oracle.com/xpath/extension/ldap"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:ns3="http://services.oracle.com/bpel/task"
xmlns:ns2="http://xmlns.oracle.com/AirAsiaWS"
xmlns:client="http://xmlns.oracle.com/AirlineBrokerWS"
xmlns:bpelx="http://schemas.oracle.com/bpel/extension"
xmlns:ora="http://schemas.oracle.com/xpath/extension"
xmlns:orcl="http://www.oracle.com/XSL/Transform/java/oracle.tip.pc.services.functions.ExtFunc">

<partnerLinks>
  <partnerLink name="client" partnerLinkType="client:AirlineBrokerWS"
myRole="AirlineBrokerWSProvider" partnerRole="AirlineBrokerWSRequester"/>
  <partnerLink name="ThaiAirwayWS" partnerRole="ThaiAirwayWSProvider"
partnerLinkType="ns1:ThaiAirwayWS"/>
  <partnerLink name="AirAsiaWS" partnerRole="AirAsiaWSProvider"
partnerLinkType="ns2:AirAsiaWS"/>
  <partnerLink myRole="TaskManagerRequester" name="TaskManager"
partnerRole="TaskManager" partnerLinkType="ns3:TaskManager"/>
</partnerLinks>

<variables>
  <variable name="inputVariable" messageType="client:AirlineBrokerWSRequestMessage"/>
  <variable name="outputVariable" messageType="client:AirlineBrokerWSResponseMessage"/>
  <variable name="invoke_TH" messageType="ns1:ThaiAirwayWSRequestMessage"/>
  <variable name="receive_TH" messageType="ns1:ThaiAirwayWSResponseMessage"/>
  <variable name="invoke_AS" messageType="ns2:AirAsiaWSRequestMessage"/>
  <variable name="receive_AS" messageType="ns2:AirAsiaWSResponseMessage"/>
 </variables>

 <sequence name="main">
  <receive name="receiveInput" partnerLink="client" portType="client:AirlineBrokerWS"
operation="initiate" variable="inputVariable" createInstance="yes"/>
  <flow name="Flow_1">
```

```
    <sequence name="Sequence_2">
     <assign name="Assign_2">
      <copy>
       <from variable="inputVariable" part="payload" query="/ns4:ABWS/ns4:source"/>
       <to variable="invoke_AS" part="payload"
query="/ns2:AirAsiaWSProcessRequest/ns2:source"/>
      </copy>
      <copy>
       <from variable="inputVariable" part="payload" query="/ns4:ABWS/ns4:destination"/>
       <to variable="invoke_AS" part="payload"
query="/ns2:AirAsiaWSProcessRequest/ns2:destination"/>
      </copy>
     </assign>
     <invoke name="Invoke_2" partnerLink="AirAsiaWS" portType="ns2:AirAsiaWS"
operation="process" inputVariable="invoke_AS" outputVariable="receive_AS"/>
    </sequence>
    <sequence name="Sequence_1">
     <assign name="Assign_1">
      <copy>
       <from variable="inputVariable" part="payload" query="/ns4:ABWS/ns4:source"/>
       <to variable="invoke_TH" part="payload"
query="/ns1:ThaiAirwayWSProcessRequest/ns1:source"/>
      </copy>
      <copy>
       <from variable="inputVariable" part="payload" query="/ns4:ABWS/ns4:destination"/>
       <to variable="invoke_TH" part="payload"
query="/ns1:ThaiAirwayWSProcessRequest/ns1:destination"/>
      </copy>
     </assign>
     <invoke name="Invoke_1" partnerLink="ThaiAirwayWS" portType="ns1:ThaiAirwayWS"
operation="process" inputVariable="invoke_TH" outputVariable="receive_TH"/>
    </sequence>
   </flow>
   <scope name="task">
    <variables>
     <variable name="reviewTask" element="ns3:task"/>
    </variables>
    <sequence name="Sequence_3">
     <assign name="configureTask">
      <copy>
       <from variable="inputVariable" part="payload" query="/ns4:ABWS/ns4:customerName"/>
       <to variable="reviewTask" query="/ns3:task/ns3:title"/>
      </copy>
      <copy>
       <from expression="string( 'AirlineBrokerWS' )"/>
       <to variable="reviewTask" query="/ns3:task/ns3:creator"/>
      </copy>
      <copy>
       <from variable="inputVariable" part="payload" query="/ns4:ABWS/ns4:source"/>
       <to variable="reviewTask" query="/ns3:task/ns3:modifier"/>
      </copy>
      <copy>
       <from variable="inputVariable" part="payload" query="/ns4:ABWS/ns4:destination"/>
       <to variable="reviewTask" query="/ns3:task/ns3:assignee"/>
      </copy>
      <copy>
       <from variable="inputVariable" part="payload" query="/ns4:ABWS/ns4:departureDate"/>
```

```
        <to variable="reviewTask" query="/ns3:task/ns3:customKey"/>
       </copy>
       <copy>
        <from variable="inputVariable" part="payload" query="/ns4:ABWS/ns4:passenger"/>
        <to variable="reviewTask" query="/ns3:task/ns3:priority"/>
       </copy>
       <copy>
        <from
expression="ora:mergeChildNodes(bpws:getVariableData('receive_TH','payload','/ns1:ThaiAirwayWS
ProcessResponse/ns1:result'),bpws:getVariableData('receive_AS','payload','/ns2:AirAsiaWSProcessRes
ponse/ns2:result'))"/>
        <to variable="reviewTask" query="/ns3:task/ns3:attachment"/>
       </copy>
      </assign>
      <scope name="taskUserInteraction">
       <variables>
        <variable name="taskRequest" messageType="ns3:taskMessage"/>
        <variable name="taskResponse" messageType="ns3:taskMessage"/>
       </variables>
       <sequence name="Sequence_4">
        <assign name="setPayload">
         <copy>
          <from variable="reviewTask"/>
          <to variable="taskRequest" part="payload"/>
         </copy>
        </assign>
        <invoke name="initiateTask" partnerLink="TaskManager" portType="ns3:TaskManager"
operation="initiateTask" inputVariable="taskRequest"/>
        <receive name="receiveTaskResult" createInstance="no" partnerLink="TaskManager"
portType="ns3:TaskManagerCallback" operation="onTaskResult" variable="taskResponse"/>
        <assign name="readPayload">
         <copy>
          <from variable="taskResponse" part="payload"/>
          <to variable="reviewTask"/>
         </copy>
        </assign>
       </sequence>
      </scope>
      <assign name="copyReviewSheet">
       <copy>
        <from variable="reviewTask" query="/ns3:task/ns3:attachment"/>
        <to variable="outputVariable" part="payload" query="/ns4:ABWS/ns4:listFlights/ns4:Item"/>
       </copy>
       <copy>
        <from variable="inputVariable" part="payload" query="/ns4:ABWS/ns4:customerName"/>
        <to variable="outputVariable" part="payload" query="/ns4:ABWS/ns4:customerName"/>
       </copy>
       <copy>
        <from variable="inputVariable" part="payload" query="/ns4:ABWS/ns4:source"/>
        <to variable="outputVariable" part="payload" query="/ns4:ABWS/ns4:source"/>
       </copy>
       <copy>
        <from variable="inputVariable" part="payload" query="/ns4:ABWS/ns4:destination"/>
        <to variable="outputVariable" part="payload" query="/ns4:ABWS/ns4:destination"/>
       </copy>
       <copy>
        <from variable="inputVariable" part="payload" query="/ns4:ABWS/ns4:departureDate"/>
```

```
        <to variable="outputVariable" part="payload" query="/ns4:ABWS/ns4:departureDate"/>
       </copy>
       <copy>
        <from variable="inputVariable" part="payload" query="/ns4:ABWS/ns4:passenger"/>
        <to variable="outputVariable" part="payload" query="/ns4:ABWS/ns4:passenger"/>
       </copy>
      </assign>
     </sequence>
   </scope>
   <invoke name="callbackClient" partnerLink="client" portType="client:AirlineBrokerWSCallback"
operation="onResult" inputVariable="outputVariable"/>
  </sequence>
</process>
```

3. HotelBrokerWS.bpel

```
<process name="HotelBrokerWS" targetNamespace="http://xmlns.oracle.com/HotelBrokerWS"
xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
xmlns:xp20="http://www.oracle.com/XSL/Transform/java/oracle.tip.pc.services.functions.Xpath20"
xmlns:bpws="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
xmlns:ns1="http://xmlns.oracle.com/SofitelHotelWS"
xmlns:ldap="http://schemas.oracle.com/xpath/extension/ldap"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:ns3="http://services.oracle.com/bpel/task"
xmlns:ns2="http://xmlns.oracle.com/KosaHotelWS"
xmlns:ns5="http://www.HotelBrokerWS.com/ns/sales"
xmlns:client="http://xmlns.oracle.com/HotelBrokerWS"
xmlns:bpelx="http://schemas.oracle.com/bpel/extension"
xmlns:ora="http://schemas.oracle.com/xpath/extension"
xmlns:orcl="http://www.oracle.com/XSL/Transform/java/oracle.tip.pc.services.functions.ExtFunc">

 <partnerLinks>
   <partnerLink name="client" partnerLinkType="client:HotelBrokerWS"
myRole="HotelBrokerWSProvider" partnerRole="HotelBrokerWSRequester"/>
   <partnerLink name="SofitelHotelWS" partnerRole="SofitelHotelWSProvider"
partnerLinkType="ns1:SofitelHotelWS"/>
   <partnerLink name="KosaHotelWS" partnerRole="KosaHotelWSProvider"
partnerLinkType="ns2:KosaHotelWS"/>
   <partnerLink myRole="TaskManagerRequester" name="TaskManager"
partnerRole="TaskManager" partnerLinkType="ns3:TaskManager"/>
 </partnerLinks>

 <variables>
   <variable name="inputVariable" messageType="client:HotelBrokerWSRequestMessage"/>
   <variable name="outputVariable" messageType="client:HotelBrokerWSResponseMessage"/>
   <variable name="invoke_ST" messageType="ns1:SofitelHotelWSRequestMessage"/>
   <variable name="receive_ST" messageType="ns1:SofitelHotelWSResponseMessage"/>
   <variable name="invoke_KS" messageType="ns2:KosaHotelWSRequestMessage"/>
   <variable name="receive_KS" messageType="ns2:KosaHotelWSResponseMessage"/>
 </variables>

 <sequence name="main">
   <receive name="receiveInput" partnerLink="client" portType="client:HotelBrokerWS"
operation="initiate" variable="inputVariable" createInstance="yes"/>
   <flow name="Flow_1">
    <sequence name="Sequence_2">
     <assign name="Assign_2">
```

```
        <copy>
          <from variable="inputVariable" part="payload" query="/ns5:HBWS/ns5:customerName"/>
          <to variable="invoke_KS" part="payload"
query="/ns2:KosaHotelWSProcessRequest/ns2:customerName"/>
        </copy>
        <copy>
          <from variable="inputVariable" part="payload" query="/ns5:HBWS/ns5:destination"/>
          <to variable="invoke_KS" part="payload"
query="/ns2:KosaHotelWSProcessRequest/ns2:destination"/>
        </copy>
      </assign>
      <invoke name="Invoke_2" partnerLink="KosaHotelWS" portType="ns2:KosaHotelWS"
operation="process" inputVariable="invoke_KS" outputVariable="receive_KS"/>
    </sequence>
    <sequence name="Sequence_1">
      <assign name="Assign_1">
        <copy>
          <from variable="inputVariable" part="payload" query="/ns5:HBWS/ns5:customerName"/>
          <to variable="invoke_ST" part="payload"
query="/ns1:SofitelHotelWSProcessRequest/ns1:customerName"/>
        </copy>
        <copy>
          <from variable="inputVariable" part="payload" query="/ns5:HBWS/ns5:destination"/>
          <to variable="invoke_ST" part="payload"
query="/ns1:SofitelHotelWSProcessRequest/ns1:destination"/>
        </copy>
      </assign>
      <invoke name="Invoke_1" partnerLink="SofitelHotelWS" portType="ns1:SofitelHotelWS"
operation="process" inputVariable="invoke_ST" outputVariable="receive_ST"/>
    </sequence>
  </flow>
  <scope name="task">
    <variables>
      <variable name="reviewTask" element="ns3:task"/>
    </variables>
    <sequence name="Sequence_3">
      <assign name="configureTask">
        <copy>
          <from variable="inputVariable" part="payload" query="/ns5:HBWS/ns5:customerName"/>
          <to variable="reviewTask" query="/ns3:task/ns3:title"/>
        </copy>
        <copy>
          <from expression="string( 'HotelBrokerWS' )"/>
          <to variable="reviewTask" query="/ns3:task/ns3:creator"/>
        </copy>
        <copy>
          <from variable="inputVariable" part="payload" query="/ns5:HBWS/ns5:destination"/>
          <to variable="reviewTask" query="/ns3:task/ns3:assignee"/>
        </copy>
        <copy>
          <from
expression="ora:mergeChildNodes(bpws:getVariableData('receive_ST','payload','/ns1:SofitelHotelWS
ProcessResponse/ns1:result'),bpws:getVariableData('receive_KS','payload','/ns2:KosaHotelWSProcess
Response/ns2:result'))"/>
          <to variable="reviewTask" query="/ns3:task/ns3:attachment"/>
        </copy>
      </assign>
```

```
    <scope name="taskUserInteraction">
     <variables>
      <variable name="taskRequest" messageType="ns3:taskMessage"/>
      <variable name="taskResponse" messageType="ns3:taskMessage"/>
     </variables>
     <sequence name="Sequence_4">
      <assign name="setPayload">
       <copy>
        <from variable="reviewTask"/>
        <to variable="taskRequest" part="payload"/>
       </copy>
      </assign>
      <invoke name="initiateTask" partnerLink="TaskManager" portType="ns3:TaskManager"
operation="initiateTask" inputVariable="taskRequest"/>
      <receive name="receiveTaskResult" createInstance="no" partnerLink="TaskManager"
portType="ns3:TaskManagerCallback" operation="onTaskResult" variable="taskResponse"/>
      <assign name="readPayload">
       <copy>
        <from variable="taskResponse" part="payload"/>
        <to variable="reviewTask"/>
       </copy>
      </assign>
     </sequence>
    </scope>
    <assign name="copyReviewSheet">
     <copy>
      <from variable="reviewTask" query="/ns3:task/ns3:attachment"/>
      <to variable="outputVariable" part="payload" query="/ns5:HBWS/ns5:listHotels/ns5:Item"/>
     </copy>
     <copy>
      <from variable="inputVariable" part="payload" query="/ns5:HBWS/ns5:customerName"/>
      <to variable="outputVariable" part="payload" query="/ns5:HBWS/ns5:customerName"/>
     </copy>
     <copy>
      <from variable="inputVariable" part="payload" query="/ns5:HBWS/ns5:destination"/>
      <to variable="outputVariable" part="payload" query="/ns5:HBWS/ns5:destination"/>
     </copy>
    </assign>
   </sequence>
  </scope>
  <invoke name="callbackClient" partnerLink="client" portType="client:HotelBrokerWSCallback"
operation="onResult" inputVariable="outputVariable"/>
 </sequence>
</process>
```

4. CarRentalBrokerWS.bpel

```
<process name="CarRentalBrokerWS"
targetNamespace="http://xmlns.oracle.com/CarRentalBrokerWS"
xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
xmlns:bpws="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
xmlns:xp20="http://www.oracle.com/XSL/Transform/java/oracle.tip.pc.services.functions.Xpath20"
xmlns:ns1="http://xmlns.oracle.com/SrichanCarRentalWS"
xmlns:ldap="http://schemas.oracle.com/xpath/extension/ldap"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:ns3="http://services.oracle.com/bpel/task"
xmlns:ns2="http://xmlns.oracle.com/PowerCarRentalWS"
```

```
xmlns:bpelx="http://schemas.oracle.com/bpel/extension"
xmlns:client="http://xmlns.oracle.com/CarRentalBrokerWS"
xmlns:ns6="http://www.CarRentalBrokerWS.com/ns/sales"
xmlns:ora="http://schemas.oracle.com/xpath/extension"
xmlns:orcl="http://www.oracle.com/XSL/Transform/java/oracle.tip.pc.services.functions.ExtFunc">

 <partnerLinks>
  <partnerLink name="client" partnerLinkType="client:CarRentalBrokerWS"
myRole="CarRentalBrokerWSProvider" partnerRole="CarRentalBrokerWSRequester"/>
  <partnerLink name="SrichanCarRentalWS" partnerRole="SrichanCarRentalWSProvider"
partnerLinkType="ns1:SrichanCarRentalWS"/>
  <partnerLink name="PowerCarRentalWS" partnerRole="PowerCarRentalWSProvider"
partnerLinkType="ns2:PowerCarRentalWS"/>
  <partnerLink myRole="TaskManagerRequester" name="TaskManager"
partnerRole="TaskManager" partnerLinkType="ns3:TaskManager"/>
 </partnerLinks>

 <variables>
  <variable name="inputVariable" messageType="client:CarRentalBrokerWSRequestMessage"/>
  <variable name="outputVariable" messageType="client:CarRentalBrokerWSResponseMessage"/>
  <variable name="invoke_SC" messageType="ns1:SrichanCarRentalWSRequestMessage"/>
  <variable name="receive_SC" messageType="ns1:SrichanCarRentalWSResponseMessage"/>
  <variable name="invoke_PW" messageType="ns2:PowerCarRentalWSRequestMessage"/>
  <variable name="receive_PW" messageType="ns2:PowerCarRentalWSResponseMessage"/>
 </variables>

 <sequence name="main">
  <receive name="receiveInput" partnerLink="client" portType="client:CarRentalBrokerWS"
operation="initiate" variable="inputVariable" createInstance="yes"/>
  <flow name="Flow_1">
   <sequence name="Sequence_2">
    <assign name="Assign_2">
     <copy>
      <from variable="inputVariable" part="payload" query="/ns6:CRBWS/ns6:customerName"/>
      <to variable="invoke_PW" part="payload"
query="/ns2:PowerCarRentalWSProcessRequest/ns2:customerName"/>
     </copy>
     <copy>
      <from variable="inputVariable" part="payload" query="/ns6:CRBWS/ns6:destination"/>
      <to variable="invoke_PW" part="payload"
query="/ns2:PowerCarRentalWSProcessRequest/ns2:destination"/>
     </copy>
    </assign>
    <invoke name="Invoke_2" partnerLink="PowerCarRentalWS"
portType="ns2:PowerCarRentalWS" operation="process" inputVariable="invoke_PW"
outputVariable="receive_PW"/>
   </sequence>
   <sequence name="Sequence_1">
    <assign name="Assign_1">
     <copy>
      <from variable="inputVariable" part="payload" query="/ns6:CRBWS/ns6:customerName"/>
      <to variable="invoke_SC" part="payload"
query="/ns1:SrichanCarRentalWSProcessRequest/ns1:customerName"/>
     </copy>
     <copy>
      <from variable="inputVariable" part="payload" query="/ns6:CRBWS/ns6:destination"/>
```

```
        <to variable="invoke_SC" part="payload"
query="/ns1:SrichanCarRentalWSProcessRequest/ns1:destination"/>
       </copy>
      </assign>
      <invoke name="Invoke_1" partnerLink="SrichanCarRentalWS"
portType="ns1:SrichanCarRentalWS" operation="process" inputVariable="invoke_SC"
outputVariable="receive_SC"/>
    </sequence>
   </flow>
   <scope name="task">
    <variables>
     <variable name="reviewTask" element="ns3:task"/>
    </variables>
    <sequence name="Sequence_3">
     <assign name="configureTask">
      <copy>
       <from variable="inputVariable" part="payload" query="/ns6:CRBWS/ns6:customerName"/>
       <to variable="reviewTask" query="/ns3:task/ns3:title"/>
      </copy>
      <copy>
       <from expression="string( 'CarRentalBrokerWS' )"/>
       <to variable="reviewTask" query="/ns3:task/ns3:creator"/>
      </copy>
      <copy>
       <from variable="inputVariable" part="payload" query="/ns6:CRBWS/ns6:destination"/>
       <to variable="reviewTask" query="/ns3:task/ns3:assignee"/>
      </copy>
      <copy>
       <from
expression="ora:mergeChildNodes(bpws:getVariableData('receive_SC','payload','/ns1:SrichanCarRent
alWSProcessResponse/ns1:result'),bpws:getVariableData('receive_PW','payload','/ns2:PowerCarRental
WSProcessResponse/ns2:result'))"/>
       <to variable="reviewTask" query="/ns3:task/ns3:attachment"/>
      </copy>
     </assign>
     <scope name="taskUserInteraction">
      <variables>
       <variable name="taskRequest" messageType="ns3:taskMessage"/>
       <variable name="taskResponse" messageType="ns3:taskMessage"/>
      </variables>
      <sequence name="Sequence_4">
       <assign name="setPayload">
        <copy>
         <from variable="reviewTask"/>
         <to variable="taskRequest" part="payload"/>
        </copy>
       </assign>
       <invoke name="initiateTask" partnerLink="TaskManager" portType="ns3:TaskManager"
operation="initiateTask" inputVariable="taskRequest"/>
       <receive name="receiveTaskResult" createInstance="no" partnerLink="TaskManager"
portType="ns3:TaskManagerCallback" operation="onTaskResult" variable="taskResponse"/>
       <assign name="readPayload">
        <copy>
         <from variable="taskResponse" part="payload"/>
         <to variable="reviewTask"/>
        </copy>
       </assign>
```

```
      </sequence>
     </scope>
     <assign name="copyReviewSheet">
      <copy>
       <from variable="reviewTask" query="/ns3:task/ns3:attachment"/>
       <to variable="outputVariable" part="payload" query="/ns6:CRBWS/ns6:listCars/ns6:Item"/>
      </copy>
      <copy>
       <from variable="inputVariable" part="payload" query="/ns6:CRBWS/ns6:customerName"/>
       <to variable="outputVariable" part="payload" query="/ns6:CRBWS/ns6:customerName"/>
      </copy>
      <copy>
       <from variable="inputVariable" part="payload" query="/ns6:CRBWS/ns6:destination"/>
       <to variable="outputVariable" part="payload" query="/ns6:CRBWS/ns6:destination"/>
      </copy>
      <copy>
       <from variable="inputVariable" part="payload" query="/ns6:CRBWS/ns6:arrTime"/>
       <to variable="outputVariable" part="payload" query="/ns6:CRBWS/ns6:arrTime"/>
      </copy>
      <copy>
       <from variable="inputVariable" part="payload" query="/ns6:CRBWS/ns6:hotelName"/>
       <to variable="outputVariable" part="payload" query="/ns6:CRBWS/ns6:hotelName"/>
      </copy>
     </assign>
    </sequence>
   </scope>
   <invoke name="callbackClient" partnerLink="client"
portType="client:CarRentalBrokerWSCallback" operation="onResult"
inputVariable="outputVariable"/>
  </sequence>
</process>
```

5. ThaiAirwayWS.bpel

```
<process name="ThaiAirwayWS" targetNamespace="http://xmlns.oracle.com/ThaiAirwayWS"
xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
xmlns:xp20="http://www.oracle.com/XSL/Transform/java/oracle.tip.pc.services.functions.Xpath20"
xmlns:bpws="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
xmlns:ldap="http://schemas.oracle.com/xpath/extension/ldap"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:client="http://xmlns.oracle.com/ThaiAirwayWS"
xmlns:bpelx="http://schemas.oracle.com/bpel/extension"
xmlns:ora="http://schemas.oracle.com/xpath/extension"
xmlns:orcl="http://www.oracle.com/XSL/Transform/java/oracle.tip.pc.services.functions.ExtFunc"
xmlns:ns10="http://www.AirlineBrokerWS.com/ns/sales">

 <partnerLinks>
  <partnerLink name="client" partnerLinkType="client:ThaiAirwayWS"
myRole="ThaiAirwayWSProvider"/>
 </partnerLinks>

 <variables>
  <variable name="inputVariable" messageType="client:ThaiAirwayWSRequestMessage"/>
  <variable name="outputVariable" messageType="client:ThaiAirwayWSResponseMessage"/>
 </variables>
```

```
 <sequence name="main">
  <receive name="receiveInput" partnerLink="client" portType="client:ThaiAirwayWS"
operation="process" variable="inputVariable" createInstance="yes"/>
  <switch name="Switch_1">
   <case
condition="bpws:getVariableData('inputVariable','payload','/client:ThaiAirwayWSProcessRequest/clie
nt:destination')='Khon Kaen'">
    <assign name="Assign_1">
     <copy>
      <from expression="ora:doc('data.xml','/result')"/>
      <to variable="outputVariable" part="payload"
query="/client:ThaiAirwayWSProcessResponse/client:result"/>
     </copy>
    </assign>
   </case>
   <otherwise/>
  </switch>
  <reply name="Reply_1" partnerLink="client" portType="client:ThaiAirwayWS"
operation="process" variable="outputVariable"/>
 </sequence>
</process>
```

ภาคผนวก ข

ตัวอย่างไฟล์ JSP สำหรับส่วนติดต่อกับผู้ใช้

1. index.jsp

```
<%@ page contentType="text/html; charset=windows-874" language="java" import="java.sql.*"
errorPage="" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Select Travel Services</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-874">
<style type="text/css">
<!--
.ms10 {
        font-family: "MS Sans Serif", "Microsoft Sans Serif";
        font-size: 10px;
}
-->
</style>
</head>

<body>
<table width="406" border="0" align="center" cellpadding="0" cellspacing="0">
 <tr>
  <td><h1>Select Travel Services</h1></td>
 </tr>
 <tr>
  <td><hr size="1"></td>
 </tr>
 <tr>
  <td><form name="form1" method="post" action="../../AirlineBrokerWSUI/bookFlight.jsp">
   <table width="340" border="0" align="center" cellpadding="1" cellspacing="1">
    <tr>
     <td width="22"><input name="airline" type="checkbox" id="airline" value="yes"
checked></td>
     <td width="311"><b>Airline Service</b></td>
    </tr>
    <tr>
     <td> </td>
     <td  class="ms10">Description : Airline ticket reserve service </td>
    </tr>
    <tr>
     <td><input name="hotel" type="checkbox" id="hotel" value="yes"></td>
     <td><b>Hotel Service </b></td>
    </tr>
    <tr>
     <td> </td>
     <td  class="ms10">Description : Hotel room reserve service </td>
    </tr>
    <tr>
     <td><input name="carRental" type="checkbox" id="carRental" value="yes"></td>
     <td><b>Car Rental Service </b></td>
    </tr>
    <tr>
     <td> </td>
     <td  class="ms10">Description : Car rental reserve service </td>
    </tr>
    <tr>
```

```
      <td><input name="CES" type="checkbox" id="CES" value="yes" checked></td>
      <td><b>Corporate Expense Service </b></td>
     </tr>
     <tr>
      <td> </td>
      <td  class="ms10">Description : Calculate for All expense service </td>
     </tr>
     <tr>
      <td height="10" colspan="2"></td>
      </tr>
     <tr>
      <td> </td>
      <td align="center"  class="ms10"><input type="submit" name="Submit" value="  Next >>
"></td>
     </tr>
    </table>
   </form></td>
 </tr>
</table>
</body>
</html>
```

2. listFlight.jsp

```
<%@page contentType="text/html; charset=windows-874" %>
<%@taglib uri="http://xmlns.oracle.com/j2ee/jsp/tld/ojsp/orabpel.tld" prefix="orabpel" %>
<%@ page import="java.util.*" %>
<%@ page import="java.sql.*" %>
<%@ page import="java.util.Date" %>
<%@ page import="java.text.SimpleDateFormat" %>
<%@ page import="org.w3c.dom.Element" %>
<%@ page import="com.oracle.bpel.client.Locator" %>
<%@ page import="com.oracle.services.bpel.task.IWorklistService" %>
<%@ page import="com.oracle.services.bpel.task.ITask" %>
<%@ page import="com.AirlineBrokerWS.www.ns.sales.ListFlightsType" %>
<%@ page import="com.AirlineBrokerWS.www.ns.sales.ListFlightsTypeFactory" %>
<%@ page import="com.AirlineBrokerWS.www.ns.sales.IListFlights" %>
<%@ page import="com.AirlineBrokerWS.www.ns.sales.IListFlightsType" %>

<html>
<head>
<title>Select Flight</title>
<meta http-equiv="PRAGMA" content="NO-CACHE" />
<meta http-equiv="EXPIRES" content="-1" />
<meta http-equiv="Content-Type" content="text/html; charset=windows-874">
<style type="text/css">
<!--
.ms10 {
        font-family: "MS Sans Serif", "Microsoft Sans Serif";
        font-size: 10px;
}
-->
</style>
</head>

<body>
```

```
<%
// connect to oracle lite database
Connection conn = null;
try
{

// Date and Time
String dateTime = new String();
SimpleDateFormat formatter = new SimpleDateFormat("2006-MM-dd HH");
dateTime = formatter.format(new Date());

Class.forName("oracle.lite.poljdbc.POLJDBCDriver");
conn = DriverManager.getConnection("jdbc:polite4@localhost:100:orabpel","system","manager");

Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("SELECT * FROM task WHERE creator='AirlineBrokerWS'
ORDER BY creation_date DESC");

if(rs.next())
{
                String taskId = rs.getString("CONVERSATION_ID");

                Locator locator = new Locator( "default", "bpel" );

    IWorklistService worklist =
      (IWorklistService)locator.lookupService( IWorklistService.SERVICE_NAME );

                ITask task  = worklist.lookupTask( taskId );
                Element rsElement = (Element) task.getAttachment();

    ListFlightsType srs = ListFlightsTypeFactory.createFacade(rsElement);

                int orderCount = srs.getItemCount();
    String airlineName, flightNumber, depTime, arrTime, economy, business, first;
    int i, rowNo;

                String customerName = task.getTitle();
                String source = task.getModifier();
                String destination = task.getAssignee();
                String departureDate = task.getCustomKey();
                int passenger = task.getPriority();
%>

<form action="completeFlight.jsp" method="POST">
<table width="460" border="0" align="center" cellpadding="0" cellspacing="0" class="ms10">
 <tr>
  <td height="30"><h1>Select Flight</h1>
        Customer Name : <font size=3><b><%=customerName%></b></font><br>
   From : <font size=3><b><%=source%></b></font>     
         To: <font size=3><b><%=destination%></b></font><br>
   Departure Date : <font size=3><b><%=departureDate%></b></font><br>
   Passanger : <font size=3><b><%=passenger%></b></font><input type="hidden"
name="passenger" value="<%=passenger%>"></td>
 </tr>
 <tr>
```

```
  <td align="center"><table width="460" border="0" align="center" cellpadding="1" cellspacing="1"
bgcolor="#000000" class="ms10">
    <tr bgcolor="#D8D1E1">
     <td width="43" rowspan="2" align="center"><b>Select</b></td>
     <td width="65" rowspan="2" align="center"><b>Company</b></td>
     <td width="71" rowspan="2" align="center"><b>Flight Number</b></td>
     <td width="74" rowspan="2" align="center"><b>Departing Time </b></td>
     <td width="74" rowspan="2" align="center"><b>Arriving<br>
    Time </b></td>
     <td colspan="3" align="center"><b>Seats available</b></td>
    </tr>
    <tr align="center">
     <td width="36" bgcolor="#D8D1E1"><b>Eco.</b></td>
     <td width="36" bgcolor="#D8D1E1"><b>Biz.</b></td>
     <td width="36" bgcolor="#D8D1E1"><b>1st</b></td>
    </tr>
        <%
            for (i=0; i<orderCount;i++) {
      IListFlights item = srs.getItem(i);
            rowNo = i+1;

            airlineName = item.getAirlineName();
            flightNumber = item.getFlightNumber();
                  depTime = item.getDepTime();
                  arrTime = item.getArrTime();
            economy = item.getEconomy();
                  business = item.getBusiness();
                  first = item.getFirst();
        %>
    <tr bgcolor="#FFFFFF">
     <td align="center">
      <input name="flight_id" type="radio" value="<%=i%>">
     </td>
     <td align="center"><%=airlineName%><input type="hidden" name="airlineName"
value="<%=airlineName%>"></td>
     <td align="center"><%=flightNumber%><input type="hidden" name="flightNumber"
value="<%=flightNumber%>"></td>
     <td align="center"><%=depTime%><input type="hidden" name="depTime"
value="<%=depTime%>"></td>
     <td align="center"><%=arrTime%><input type="hidden" name="arrTime"
value="<%=arrTime%>"></td>
     <td align="center"><%=economy%></td>
     <td align="center"><%=business%></td>
     <td align="center"><%=first%></td>
    </tr>
        <% } %>
   </table>
    <br>
    <table width="360" border="0" align="center" cellpadding="1" cellspacing="1" class="ms10">
     <tr>
      <td width="73"><b>Cabin Class </b></td>
      <td width="80"><input name="cabinClass" type="radio" value="economy" checked>
    Economy </td>
      <td width="97"><input name="cabinClass" type="radio" value="business">
    Business Class </td>
      <td width="97"><input name="cabinClass" type="radio" value="first">
    First Class </td>
```

```
      </tr>
      <tr align="center">
        <td colspan="4">
                    <input type="hidden" name="taskId" value="<%=taskId%>" />
                    <input type="hidden" name="hotel" value="<%=request.getParameter("hotel")%>">
                    <input type="hidden" name="carRental"
value="<%=request.getParameter("carRental")%>">
                    <input type="submit" name="Submit" value=" Submit "></td>
      </tr>
    </table></td>
  </tr>
</table>
</form>

<%
}

}

catch(SQLException e)
{
out.println("SQLException: " + e.getMessage() + "<BR>");
while((e = e.getNextException()) != null)
out.println(e.getMessage() + "<BR>");
}
catch(ClassNotFoundException e)

{
out.println("ClassNotFoundException: " + e.getMessage() + "<BR>");
}
finally
{

//Clean up resources, close the connection.
if(conn != null)
{
try
{
conn.close();
}
catch (Exception ignored) {}
}
}
%>

</body>
</html>
```

3. completeFlight.jsp

```
<%@ page import="java.util.*" %>
<%@ page import="org.w3c.dom.Element" %>
<%@ page import="com.oracle.bpel.client.Locator" %>
<%@ page import="com.oracle.services.bpel.task.ITask" %>
<%@ page import="com.oracle.services.bpel.task.IWorklistService" %>
<%@ page import="com.AirlineBrokerWS.www.ns.sales.ListFlights" %>
```

```
<%@ page import="com.AirlineBrokerWS.www.ns.sales.ListFlightsFactory" %>
<%@ page import="com.AirlineBrokerWS.www.ns.sales.ListFlightsType" %>
<%@ page import="com.AirlineBrokerWS.www.ns.sales.ListFlightsTypeFactory" %>
<%@ page import="com.AirlineBrokerWS.www.ns.sales.IListFlights" %>
<%@ page import="com.AirlineBrokerWS.www.ns.sales.IListFlightsType" %>
<%@ page import="com.AirlineBrokerWS.www.ns.sales.ABWSType" %>
<%@ page import="com.AirlineBrokerWS.www.ns.sales.ABWSTypeFactory" %>
<%@ page import="com.collaxa.common.util.MathUtils" %>
<%@page import="javax.xml.namespace.QName"%>
<%@ page import="java.sql.*" %>

<html>
<head>
  <meta http-equiv="PRAGMA" content="NO-CACHE" />
  <meta http-equiv="EXPIRES" content="-1" />
</head>
<body>

<h1>Result of Airline Ticket Reservation</h1>

<%
                // Create an XML element using the raw DOM API can be painful.
                // We can instead use an XML facade to create and empty element
                // and populate it using a typed interface
                ListFlights srs = ListFlightsFactory.createFacade(new
QName("http://xmlns.oracle.com/AirlineBrokerWS", "ListFlights"));

                        ABWSType srs3 = ABWSTypeFactory.createFacade(new
QName("http://xmlns.oracle.com/AirlineBrokerWS", "ABWSType"));

    // Obtain a reference to the service locator for the "default" BPEL domain.
    // The initial password of the 'default' domain is 'bpel'.
    Locator locator = new Locator( "default","bpel" );

      // Lookup the worklist service.
    IWorklistService worklist =
      (IWorklistService)locator.lookupService( IWorklistService.SERVICE_NAME );

                String taskId = request.getParameter("taskId");
                // Lookup the task that we want to update and complete
    ITask task = worklist.lookupTask( taskId );

                Element rsElement = (Element) task.getAttachment();
                ListFlightsType srs2 = ListFlightsTypeFactory.createFacade(rsElement);
    String airlineName, flightNumber, depTime, arrTime, economy, business, first;

                 String i = request.getParameter("flight_id");
     IListFlights item = srs2.getItem(  Integer.parseInt( i ) );

          airlineName = item.getAirlineName();
          flightNumber = item.getFlightNumber();
                depTime = item.getDepTime();
                arrTime = item.getArrTime();
          economy = item.getEconomy();
                business = item.getBusiness();
                first = item.getFirst();
```

```
    srs.setAirlineName( airlineName );
    srs.setFlightNumber( flightNumber );
                srs.setDepTime( depTime );
    srs.setArrTime( arrTime );

                String strCabinClass = request.getParameter( "cabinClass" );
    srs.setCabinClass( strCabinClass );

                String hotel = request.getParameter("hotel");
    srs.setHotel( hotel );

                String carRental = request.getParameter("carRental");
    srs.setCarRental( carRental );

                String countPassenger = request.getParameter("passenger");
                if ( strCabinClass.equals("economy")) {
                        srs.setPriceAirline (  Integer.parseInt( countPassenger ) *  Integer.parseInt(
economy ));
                } else if ( strCabinClass.equals("business")) {
                        srs.setPriceAirline (  Integer.parseInt( countPassenger ) *  Integer.parseInt(
business ));
                } else if ( strCabinClass.equals("first")) {
                        srs.setPriceAirline (  Integer.parseInt( countPassenger ) *  Integer.parseInt(
first ));
                }

        // Update the attachment so that it reflects the changes submitted by the user.
    task.setAttachment( srs.getRootElement( ) );

    worklist.completeTask( task );

                out.println("This task has been successfully completed.");
%>

<%
                if (hotel.equals("yes")){
                        out.println("<div style='width:100%'>");
                        out.println("<a
href='../../HotelBrokerWSUI/listHotel.jsp?carRental="+carRental+"'>Go to Hotel Service</a>");
                        out.println("</div>");
                } else if (hotel.equals("null") && carRental.equals("yes")) {
                        out.println("<div style='width:100%'>");
                        out.println("<a href='../../CarRentalBrokerWSUI/listCarRental.jsp'>Go to
Car Rental Service</a>");
                        out.println("</div>");
                } else if (hotel.equals("null") && carRental.equals("null")) {
                        out.println("<div style='width:100%'>");
                        out.println("<a href='../../TISUI/detail.jsp'>Go to Corporate Expense</a>");
                        out.println("</div>");
                }
%>

</body>
</html>
```

ภาคผนวก ค

การเผยแพร่ผลงานวิทยานิพนธ์

การเผยแพร่ผลงานวิทยานิพนธ์

1. เอกชัย แน่นอุดร, งามนิจ อาจอินทร์ และ สมจิตร อาจอินทร์ (2549, 29-30 มิถุนายน). ระบบบูรณาการบริการสำหรับข้อมูลสารสนเทศการท่องเที่ยวโดยใช้องค์ประกอบร่วมเว็บเซอร์วิส. ใน: การประชุมวิชาการวิทยาการคอมพิวเตอร์และวิศวกรรมซอฟต์แวร์ ครั้งที่ 3. (หน้า 219-226). กรุงเทพฯ: คณะวิทยาศาสตร์ประยุกต์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ.

ระบบบูรณาการบริการสำหรับข้อมูลสารสนเทศการท่องเที่ยว โดยใช้องค์ประกอบร่วมเว็บเซอร์วิส

Services Integration System for Tourism Information using Web Services Composition

เอกชัย แน่นอุดร[1], งามนิจ อาจอินทร์[2] และ สมจิตร อาจอินทร์[3]

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น 44002

E-mail: [1]nkcombat@hotmail.com, [2]ngamnij@kku.ac.th, [3]somjit@kku.ac.th

บทคัดย่อ

งานวิจัยนี้มีจุดประสงค์เพื่อออกแบบและพัฒนาองค์ประกอบร่วมของเว็บเซอร์วิสที่เกี่ยวข้องกับธุรกิจด้านการท่องเที่ยว ซึ่งประกอบด้วยกลุ่มธุรกิจจองตั๋วเครื่องบิน จองโรงแรมและห้องพัก และจองรถเช่า โดยที่เว็บเซอร์วิสของผู้ให้บริการแต่ละราย สามารถติดต่อสื่อสารรับส่งข้อมูล และทำงานร่วมกันได้อย่างอัตโนมัติ โดยมีระบบสารสนเทศการท่องเที่ยว (Tourism Information Systems : TIS) เป็นตัวกลางในการประสานงาน ซึ่งพัฒนาขึ้นจากภาษาที่ใช้สร้างกระบวนการทางธุรกิจคือ BPEL4WS ทำให้การสื่อสารแลกเปลี่ยนข้อมูลระหว่างเว็บเซอร์วิสเป็นไปแบบอัตโนมัติ ระหว่างผู้ให้บริการทำให้เกิดการสร้างพันธมิตรทางการค้า เป็นการเพิ่มช่องทางการขายทำให้มีโอกาสทางการตลาดที่มากขึ้นและเพิ่มกำไรให้กับผู้ร่วมทำธุรกิจ ทางด้านผู้ใช้บริการได้รับบริการอย่างครบวงจรและรวดเร็วเพราะมีบริการแบบเสร็จสรรพในจุดเดียว (One Stop Service)

คำสำคัญ: เว็บเซอร์วิส, องค์ประกอบร่วมเว็บเซอร์วิส, กระบวนการทางธุรกิจ, BPEL4WS

Abstract

This research aims to design and develop the Web Service Composition for tourism business particularly air ticket reservation, hotel reservation and car rental. Web Service can communicate, share information and work with each others automatically which control by the coordinated Tourism Information Systems (TIS). The language develops from the Business Process Execution Language for Web Services (BPEL4WS) therefore information exchange could be done without human intervention. Networking and alternate market could be the remarkable benefit to business. On the other hand, the customers could meet their satisfactory on One Stop Service.

Keyword: Web Services, Web Services Composition, Business Process, BPEL4WS

1. บทนำ

ปัจจุบันแต่ละองค์กรมีการพัฒนาระบบงานด้วยโปรแกรมประยุกต์ที่แตกต่างกัน จึงทำให้การแลกเปลี่ยนข้อมูลกันทำได้ยาก ตัวอย่างเช่น องค์กรที่ให้บริการจองตั๋วเครื่องบินและองค์กรที่ให้บริการโรงแรมต่างก็มีการพัฒนาโปรแกรมประยุกต์ที่เป็นของตนเอง โดยใช้ภาษาในการพัฒนาที่แตกต่างกัน จึงทำให้เกิดปัญหาในการทำงานร่วมกัน

(Interoperability) และการแลกเปลี่ยนข้อมูลระหว่างกัน (Exchangeability) ทำให้การทำงานประสานกันระหว่างระบบขององค์กรไม่สามารถเกิดขึ้นได้อย่างอัตโนมัติ

ในปัจจุบันเทคโนโลยีเว็บเซอร์วิส (Web Services) [1] ได้เข้ามามีบทบาทและช่วยแก้ปัญหาที่เกิดขึ้นได้ โดยสามารถทำให้โปรแกรมประยุกต์ที่แตกต่างกัน สามารถทำงานร่วมกันได้และแลกเปลี่ยนข้อมูลร่วมกันได้ เนื่องจากคุณลักษณะของเว็บเซอร์วิสที่มีการใช้ภาษา XML [2] เป็นภาษามาตรฐานในการแลกเปลี่ยนข้อมูลระหว่างกัน ทำให้การทำธุรกิจระหว่างองค์กรเป็นไปได้ง่ายขึ้น การนำเว็บเซอร์วิสมาประกอบร่วมกัน (Web Services Composition) ทำให้ผู้ให้บริการสามารถทำธุรกิจร่วมกันและแลกเปลี่ยนข้อมูลร่วมกันได้ ทำให้เกิดการสร้างพันธมิตรทางการค้า เป็นการเพิ่มช่องทางการขายทำให้มีโอกาสทางการตลาดที่มากขึ้นและเพิ่มกำไรให้กับผู้ร่วมทำธุรกิจ (Partner) ทางด้านลูกค้าทำให้ประหยัดเวลาเพราะมีบริการอย่างครบวงจรและเสร็จสรรพในจุดเดียว (One Stop Service)

บทความนี้จะกล่าวถึงการออกแบบและสร้างองค์ประกอบร่วมของเว็บเซอร์วิสที่ให้บริการด้านการท่องเที่ยว โดยเนื้อหาของบทความจะมีลำดับการนำเสนอดังต่อไปนี้ ทฤษฎีที่เกี่ยวข้อง งานวิจัยที่เกี่ยวข้อง การบูรณาการบริการระบบสารสนเทศการท่องเที่ยว การสร้างองค์ประกอบร่วมของเว็บเซอร์วิส ผลการวิจัย และบทสรุปและข้อเสนอแนะ

2. ทฤษฎีที่เกี่ยวข้อง

2.1 เว็บเซอร์วิส (Web Services) คือ โปรแกรมหรือแอพพลิเคชั่น (application) ที่ถูกติดตั้งอยู่บนเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นผู้ให้บริการ (service provider) ที่ทำหน้าที่ให้บริการส่วนของการทำงานอย่างใดอย่างหนึ่ง โดยจะมีแอพพลิเคชั่นอื่นที่อาจถูกสร้างด้วยภาษาโปรแกรมที่แตกต่างกันบนเครื่องที่มีระบบปฏิบัติการที่ต่างกัน ที่ทำหน้าที่เป็นผู้เรียกใช้บริการ (service requester) มาเรียกใช้งานหรือขอใช้บริการผ่านเครือข่ายอินเทอร์เน็ตบนโพรโทคอล HTTP เทคโนโลยีที่เกี่ยวกับข้องกับเว็บเซอร์วิสมีดังนี้

(1) SOAP (Simple Object Access Protocol) [3] เป็นโพรโทคอลที่มีโครงสร้างพื้นฐานภาษา XML ถูกใช้เป็นสื่อกลางในการแลกเปลี่ยนข้อมูลระหว่างเครื่องผู้ให้บริการและผู้เรียกใช้บริการ โดยทำงานร่วมกับโพรโทคอล HTTP

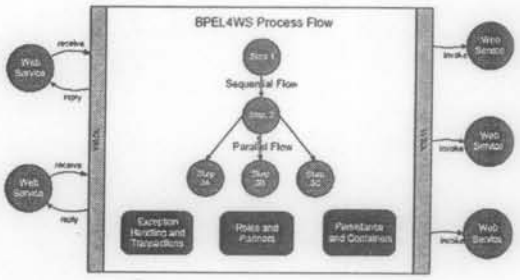
(2) WSDL (Web Services Description Language) [4] เป็นภาษาที่ใช้เพื่อกำหนดรูปแบบของการอธิบายการเรียกใช้เว็บเซอร์วิส เช่น ชื่อของบริการ เมธอด (method) พารามิเตอร์ที่เป็นข้อมูลเข้าและค่าผลลัพธ์ที่ส่งกลับมา เป็นต้น เพื่อให้ผู้เรียกใช้บริการสามารถเขียนโปรแกรมเรียกใช้บริการเหล่านั้นได้

(3) UDDI (Universal Description, Discovery and Integration) [5] เป็นมาตรฐานที่ใช้สำหรับค้นหาบริการต่างๆ ของเว็บเซอร์วิสที่ผู้ให้บริการมาลงทะเบียนไว้ และบอกตำแหน่ง URL ของไฟล์ WSDL ที่ใช้อธิบายรายละเอียดการให้บริการให้แก่ผู้ที่มาค้นหาบริการ

2.2 องค์ประกอบร่วมของเว็บเซอร์วิส (Web Services Composition) คือ วิธีการประกอบร่วมของเว็บเซอร์วิส เพื่อทำให้เว็บเซอร์วิสต่างๆ นั้นสามารถทำงานร่วมกันได้อย่างอัตโนมัติ ตัวอย่างภาษาที่ใช้สร้างองค์ประกอบร่วมของเว็บเซอร์วิส เช่น BPEL4WS [6], WSFL [7] และ XLANG [8] เครื่องมือที่ใช้ในการพัฒนา เช่น WebSphere Studio Application Developer [9] และ Oracle JDeveloper [10] โดยแต่ละภาษามีจุดมุ่งหมายเดียวกันคือสร้างกระบวนการทางธุรกิจ (Business Process) เพื่อกำหนดถึงการทำงานร่วมกันของเว็บเซอร์วิส โดยจะ

กำหนดถึงวิธีการ และลำดับขั้นตอนของการดำเนินการร่วมกัน

BPEL4WS เป็นอีกภาษาหนึ่งที่นิยมใช้ในการสร้างองค์ประกอบร่วมของเว็บเซอร์วิสในปัจจุบัน เป็นภาษาที่ใช้กำหนดถึงการทำงานร่วมกันของเว็บเซอร์วิส โดยจะกำหนดถึงวิธีการ และลำดับขั้นตอนของการดำเนินการร่วมกัน เกิดขึ้นจากความพยายามที่จะสร้างมาตรฐานร่วมกันระหว่าง มาตรฐาน WSFL ของ IBM และมาตรฐาน XLANG ของไมโครซอฟต์ เพื่อให้เกิดมาตรฐานที่ดีที่สุดโดยหยิบยกเอาข้อดีของทั้งสองมาตรฐานมารวมกัน



ภาพที่ 1 BPEL4WS Process Flow [11]

BPEL4WS จะกำหนดถึงลำดับการทำงานในแต่ละ process ของเว็บเซอร์วิส จะดูได้จากไฟล์ WSDL ของแต่ละเว็บเซอร์วิส ที่ได้ประกาศไว้บน UDDI ว่าบริการนี้จะต้องรับส่งค่าพารามิเตอร์อะไรไปบ้าง มาสร้างเป็นกระบวนการทางธุรกิจ โครงสร้างพื้นฐานของไฟล์ BPEL ประกอบด้วย 3 ส่วนหลัก ได้แก่ <partnerLinks> เพื่อกำหนดถึงเว็บเซอร์วิสที่เกี่ยวข้องทั้งหมดที่อยู่ในระบบ <variables> เพื่อกำหนดถึงค่าที่จะใช้รับส่งระหว่างเว็บเซอร์วิส และ <sequence> เพื่อกำหนดลำดับของการทำงาน ดังภาพที่ 2

```
<process name="BusinessTravelProcess" ... >
  <partnerLinks>
    <!-- The declaration of partner links -->
  </partnerLinks>
  <variables>
    <!-- The declaration of variables -->
  </variables>
  <sequence>
    <!-- The definition of the BPEL business process main
body -->
  </sequence>
```

ภาพที่ 2 โครงสร้างพื้นฐานของเอกสาร BPEL

3. งานวิจัยที่เกี่ยวข้อง

หลายงานวิจัยมีการนำเสนอถึงวิธีการสร้างระบบข้อมูลสารสนเทศการท่องเที่ยวเพื่อความสะดวกและตอบสนองความต้องการของผู้ใช้ ตัวอย่างเช่น งานวิจัยของ [12] ได้แสดงถึงการสร้างระบบข้อมูลสารสนเทศเกี่ยวกับธุรกิจการท่องเที่ยวโดยการสร้าง Multi-Agent ซึ่งในแต่ละ Agent เก็บข้อมูลของกลุ่มบริการท่องเที่ยวที่แตกต่างกันไป เช่น Agent A เก็บข้อมูลของโรงแรม A และ Agent B เก็บข้อมูลของโรงแรม B ผู้ใช้สามารถติดต่อได้ผ่านทาง Client Agent (CA) และมี Agent Name Server ซึ่งเป็นตัวกลางที่จะค้นข้อมูลที่อยู่ใน Agent ในกลุ่มต่างๆ และส่งกลับผลลัพธ์ไปยังผู้ใช้ และงานวิจัย [13] ได้แสดงถึงการเปรียบเทียบถึงข้อดีข้อเสียและความสามารถของภาษาที่ใช้ในการสร้างองค์ประกอบร่วมเว็บเซอร์วิสระหว่างภาษา BPEL กับภาษาอื่นๆ และการจัดการกระบวนการทางธุรกิจไว้อย่างชัดเจน

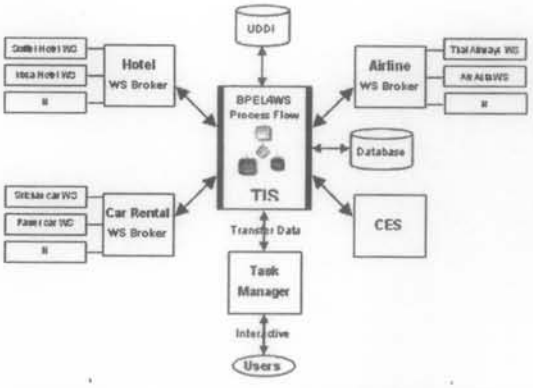
ถึงแม้งานวิจัยส่วนใหญ่จะมุ่งเน้นไปที่การสร้างระบบข้อมูลสารสนเทศ แต่ยังไม่มีงานวิจัยใดที่มุ่งเน้นการแก้ปัญหาการ บูรณาการบริการระบบสารสนเทศการท่องเที่ยวของผู้ให้บริการที่มีความหลากหลาย ซึ่งเป็นปัญหาสำคัญที่เกิดขึ้นในปัจจุบัน

4. การบูรณาการบริการระบบสารสนเทศการท่องเที่ยว

งานวิจัยนี้ได้ทำการออกแบบและพัฒนาสถาปัตยกรรมการบูรณาการบริการระบบข้อมูลสารสนเทศ (Tourism Information System) หรือระบบ TIS โดยใช้องค์ประกอบร่วมของเว็บเซอร์วิส เพื่อทำให้ธุรกิจบริการการท่องเที่ยวต่างๆ สามารถติดต่อสื่อสารกันและทำงานร่วมกันได้อย่างอัตโนมัติ ซึ่งเป็นการอำนวยความสะดวกให้กับลูกค้าที่ใช้บริการโดยสามารถได้รับบริการข้อมูลการท่องเที่ยวและเลือกใช้บริการต่าง ๆ ได้โดยผ่านระบบสารสนเทศการท่องเที่ยว

4.1 สถาปัตยกรรมระบบสารสนเทศการท่องเที่ยว

ในหัวข้อนี้จะนำเสนอถึงสถาปัตยกรรมของระบบสารสนเทศการท่องเที่ยว โดยมีระบบ TIS ทำหน้าที่เป็นตัวกลางในการประสานงานการทำงานร่วมกันของเว็บเซอร์วิสต่างๆของธุรกิจบริการการท่องเที่ยว ซึ่งประกอบไปด้วยกลุ่มธุรกิจจองตั๋วเครื่องบิน ธุรกิจโรงแรม และธุรกิจรถเช่า ดังภาพที่ 3



ภาพที่ 3 สถาปัตยกรรมของระบบสารสนเทศการท่องเที่ยว โดยใช้องค์ประกอบร่วมเว็บเซอร์วิส

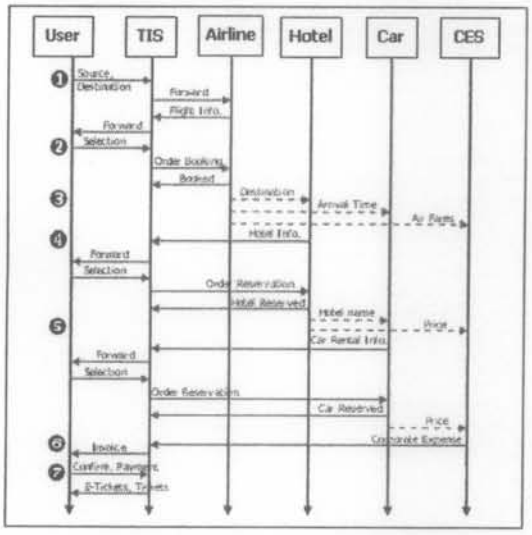
ในแต่ละองค์ประกอบสามารถอธิบายหน้าที่ได้ดังต่อไปนี้

- ระบบสารสนเทศการท่องเที่ยว หรือ ระบบ TIS ทำหน้าที่เป็นตัวกลางในการประสานงานการทำงานร่วมกันของเว็บเซอร์วิสต่างๆ ของธุรกิจบริการการท่องเที่ยว
- ตัวแทนบริการท่องเที่ยวต่างๆ (Web Service Broker) ทำหน้าที่เป็นตัวแทนในการเรียกไปยังเว็บเซอร์วิสในกลุ่มบริการของตน เพื่อรวบรวมผลลัพธ์ส่งไปยังระบบ TIS เพื่อส่งข้อมูลผ่านไปยังตัวบริหารงานให้ลูกค้าเลือกรายการ
- ตัวบริหารงาน (Task Manager Service) ทำหน้าที่ประสานงานระหว่างตัว BPEL Process กับลูกค้าหรือผู้ใช้
- ตัวบริการจัดการค่าใช้จ่าย (Corporate Expense Service หรือ CES) ทำหน้าที่สรุปค่าใช้จ่ายทั้งหมดส่งกลับไปยัง TIS
- ลูกค้าหรือผู้ใช้งานทั่วไป (Users) ทำหน้าที่เป็นผู้เรียกใช้บริการ (Service requester) จากระบบ TIS โดยลูกค้าสามารถเลือกได้ว่าจะใช้บริการใดบ้าง
- Database ทำหน้าที่เก็บ Transaction ที่เกิดขึ้นจากการทำงานของระบบ TIS
- UDDI เก็บรวบรวมการให้บริการของเว็บเซอร์วิสต่างๆ เพื่อให้ระบบ TIS ค้นหาไฟล์ WSDL เพื่อประกอบการสร้างองค์ประกอบร่วมเว็บเซอร์วิส

4.2 แผนภาพกระบวนการทางธุรกิจขององค์ประกอบร่วมเว็บเซอร์วิส

การออกแบบกระบวนการทางธุรกิจของระบบสารสนเทศการท่องเที่ยว จะประกอบไปด้วยธุรกิจต่างๆที่ลูกค้าสามารถเลือกที่จะใช้บริการแบบครบวงจรได้ และเป็นธุรกิจที่ต้องทำงานร่วมกัน เช่น Airline, Hotel, Car Rental และ Corporate expense service ซึ่งทุกธุรกิจถูกออกแบบเป็นเว็บเซอร์วิส ที่มีตัวกลางในการเรียกใช้บริการต่างๆจากเว็บเซอร์วิสซึ่งได้แก่ ระบบ TIS โดยมีลำดับขั้นตอนการทำงานที่แสดงในภาพที่ 4 โดยเริ่มจากลูกค้าร้องขอบริการท่องเที่ยวผ่านทางระบบ TIS โดยลูกค้าจะส่งข้อมูลเบื้องต้น เช่น สถานที่ต้นทาง ปลายทาง และวันเวลาที่ต้องการเดินทาง จากนั้นระบบ TIS จะส่งข้อมูลไปยัง Airline Web Service เพื่อค้นหาและแสดงรายการข้อมูลเที่ยวบินกลับมาให้ลูกค้าทำการเลือกต่อว่าต้องการไปกับสายการบินใด กี่ที่นั่ง เลือกชั้นแบบประหยัดหรือแบบธุรกิจ จากนั้น ระบบ TIS ก็จะส่งข้อมูลต่อไปยัง Airline Web Service เพื่อทำการสำรองที่นั่ง โดยข้อมูลปลายทาง เวลาที่มาถึง และค่าโดยสารเครื่องบิน จะถูกส่งต่อไปที่เว็บเซอร์วิสของ Hotel, Car rental และ CES ตามลำดับ ซึ่ง Hotel Web Service รับข้อมูลปลายทางเข้ามา เพื่อค้นหาโรงแรมในบริเวณที่ลูกค้าต้องการไปออกมา พร้อมกับประเภทโรงแรมและราคามาให้ลูกค้าเลือกด้วย เมื่อเลือกโรงแรมได้แล้วระบบ TIS ก็จะส่งราคาค่าเช่า

ห้องพักโรงแรมไปยังบริการ CES และส่งชื่อโรงแรมไปยัง Car Rental Web Service เพื่อแสดงประเภทของรถมาให้ลูกค้าเลือก โดยที่ Car Rental Web Service จะทราบอัตโนมัติว่าจะต้องไปรับลูกค้าที่ Airport ใด เวลาใด และจะต้องไปส่งลูกค้าที่โรงแรมใด และสำหรับตัวบริการจัดการค่าใช้จ่าย หรือ CES ก็ทำการสรุปค่าใช้จ่ายทั้งหมด ซึ่งลูกค้าจะต้องเป็นสมาชิกอยู่ในบริการจ่ายเงินนี้ด้วย เพื่อสรุปส่งเป็น Invoice และระบบ TIS ก็จะแสดงรายการค่าใช้จ่ายทั้งหมดให้ลูกค้าทราบด้วย ลูกค้าทำการยืนยันรายการที่เลือกไปและชำระเงิน จากนั้นระบบ TIS ก็จะนำส่ง Tickets หรือจะสั่งพิมพ์ตั๋วผ่านระบบ E-Tickets ทาง Web Browser ได้อีกด้วย

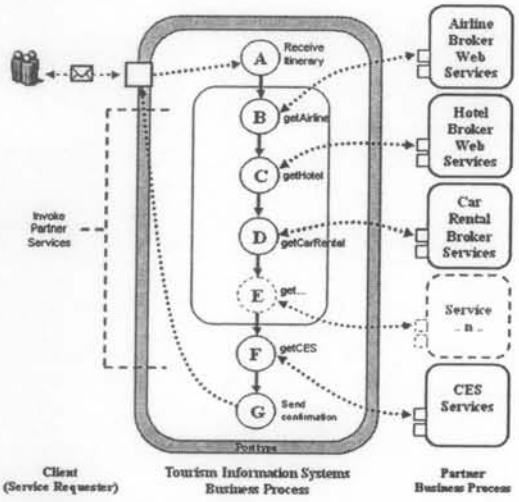


ภาพที่ 4 ลำดับขั้นตอนการทำงานของธุรกิจบริการท่องเที่ยวต่าง ๆ

4.2.1 แผนภาพกระบวนการทางธุรกิจของระบบในระดับบน : Level 0

การออกแบบกระบวนการทางธุรกิจในระดับบน หรือ Level 0 เป็นการแสดงการทำงานร่วมกันของกลุ่มธุรกิจต่างๆ ได้แก่ กลุ่มธุรกิจจองตั๋วเครื่องบิน กลุ่มธุรกิจโรงแรม และกลุ่มธุรกิจรถเช่า ซึ่งสถาปัตยกรรมของระบบจะรองรับกลุ่มธุรกิจที่จะเกิดขึ้นในอนาคตด้วย เช่น กลุ่มธุรกิจร้านอาหาร หรือธุรกิจสปา เป็นต้น โดยมีตัวกลางในการประสานงานระหว่างกลุ่มธุรกิจต่างๆ ซึ่งเป็นระบบ TIS เป็นตัวจัดการกระบวนการทางธุรกิจที่จะเกิดขึ้น ทำให้การติดต่อสื่อสารและการรับส่งค่าระหว่างกลุ่มธุรกิจเป็นไปอย่างอัตโนมัติ

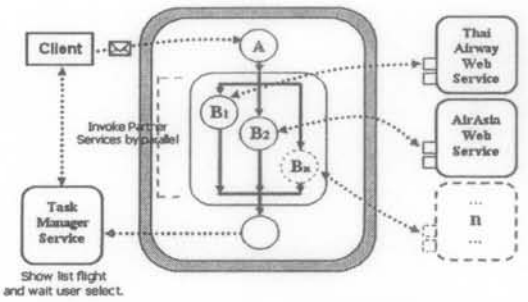
กระบวนการทางธุรกิจจะเริ่มตั้งแต่รับข้อมูลความต้องการเบื้องต้นจากลูกค้า (Activity A) จากนั้นทำการส่งข้อมูลต่อไปยัง Airline Broker Web Services (Activity B) ซึ่งจะมีการประมวลผลจาก Activity A ถึง Activity G ตามลำดับ ดังภาพที่ 5 โดยเริ่มแรกลูกค้าสามารถเลือกได้ว่าจะใช้บริการใดบ้าง และจะไม่ใช้บริการใดบ้าง เช่น ลูกค้าเลือกบริการจองตั๋วเครื่องบิน และจองโรงแรม แต่อาจจะไม่ได้เลือกบริการจองรถเช่า เป็นต้น



ภาพที่ 5 กระบวนการทางธุรกิจของระบบสารสนเทศการท่องเที่ยวในระดับบน : Level 0

4.2.2 แผนภาพกระบวนการทางธุรกิจของตัวแทนการบริการในระดับล่าง : Level 1

การออกแบบกระบวนการทางธุรกิจในระดับล่าง หรือ Level 1 แบ่งตามธุรกิจบริการต่างๆ ออกเป็น 4 กลุ่ม ได้แก่ Airline Broker Web Services, Hotel Broker Web Services, Car Rental Broker Web Services และ Corperate Expense Service ซึ่งในแต่ละกลุ่มมีลำดับขั้นตอนการทำงานที่คล้ายกันดังตัวอย่างในภาพที่ 6



ภาพที่ 6 แผนภาพการทำงานของตัวแทนการบริการจองตั๋วเครื่องบินในระดับล่าง : Level 1

การประมวลผลจะเริ่มตั้งแต่รับค่าชื่อลูกค้า ต้นทาง ปลายทาง วันที่เดินทาง และจำนวนผู้โดยสาร จาก Client (Activity A) หลังจากนั้นระบบจะประมวลผลโดยเรียกไปยังเว็บเซอร์วิสจองตั๋วเครื่องบินต่างๆ (Activity B) และส่งค่ารายการเที่ยวบินที่ได้ไปยังตัวบริหารงานหรือ Task Manager Service เพื่อแสดงรายการให้ลูกค้าได้เลือกผ่านทางเว็บบราวเซอร์ โดยค่าผลลัพธ์ที่ได้จากบริการนี้ก็จะถูกส่งต่อไปยังธุรกิจบริการอื่นๆต่อ เช่น ตัวแทนบริการโรงแรม เป็นต้น

5. การสร้างองค์ประกอบร่วมของเว็บเซอร์วิส

หัวข้อนี้จะแสดงกระบวนการสร้างองค์ประกอบร่วมเว็บเซอร์วิสของระบบสารสนเทศการท่องเที่ยวในระดับบนสุดโดยใช้ภาษา BPEL ซึ่งพัฒนาขึ้นจากโปรแกรม Oracle JDeveloper BPEL Designer [10] ดังรายละเอียดต่อไปนี้

5.1 การค้นหาบริการที่ต้องการ

กระบวนการแรกสุดของการสร้างองค์ประกอบร่วมเว็บเซอร์วิส ได้แก่ การค้นหาบริการจากไดเร็ก-เทอรีกลาง UDDI เพื่อให้ทราบถึงบริการ ถึงเมธอด พารามิเตอร์ข้อมูลเข้าออก และตำแหน่งที่อยู่ของเอกสาร WSDL ซึ่งจะถูกนำไปสร้างเป็นองค์ประกอบร่วมเว็บเซอร์วิสต่อไป

5.2 การสร้างกระบวนการทางธุรกิจของระบบสารสนเทศการท่องเที่ยว

5.2.1 กำหนดค่าเอกสาร WSDL

กำหนดค่าของ WSDL สำหรับเตรียมรับ message จาก Clients และตอบกลับค่าผลลัพธ์ โดยได้กำหนดไว้ในรูปแบบของ XML Schema แล้วจึงทำการ import ไฟล์ xsd นั้นเข้ามาใน WSDL โดยติดต่อผ่านทาง Port Type และกำหนด Partner Link Types สำหรับติดต่อกับเว็บเซอร์วิสอื่นๆ

5.2.2 สร้างกระบวนการทางธุรกิจด้วยภาษา BPEL4WS

เป็นขั้นตอนการสร้างกระบวนการทางธุรกิจ ตามมาตรฐานของ BPEL4WS ซึ่งโครงสร้างพื้นฐานของไฟล์ BPEL ประกอบด้วย 3 ส่วนหลัก ได้แก่ <partnerLinks> เพื่อกำหนดถึงเว็บเซอร์วิสที่เกี่ยวข้องทั้งหมดที่อยู่ในระบบ TIS <variables> เพื่อกำหนดถึงค่าที่จะใช้รับส่งระหว่างเว็บเซอร์วิส และ <sequence> เพื่อกำหนดลำดับของการทำงาน

การสร้างกระบวนการทางธุรกิจของระบบ TIS ในระดับบน จะประกอบด้วยขั้นตอนดังนี้

(1) กำหนดชื่อของ Process และ namespaces

```
<process name="TIS"
  targetNamespace="http://xmlns.oracle.com/
  TIS"
  xmlns:ns1="http://xmlns.oracle.com/
  AirlineBrokerWS"
  xmlns:ns2="http://xmlns.oracle.com/
  HotelBrokerWS"
  xmlns:client="http://xmlns.oracle.com/TIS"
  .../>
```

กำหนดชื่อของ Process เป็น TIS และกำหนด namespaces ของแต่ละกลุ่มบริการเพื่อป้องกันการอ้างอิงค่าที่ซ้ำซ้อน

(2) กำหนด Partner Links

```
<partnerLinks>
  <partnerLink name="Customer"
    partnerLinkType="client:TIS"
    myRole="TISProvider"
    partnerRole="TISRequester"/>
  <partnerLink name="AirlineBrokerWS"
    partnerLinkType="ns1:AirlineBrokerWS"
     myRole="AirlineBrokerWSRequester"
    partnerRole="AirlineBrokerWSProvider"/
    >
```

```
<partnerLink name="HotelBrokerWS"
  partnerLinkType="ns2:HotelBrokerWS"
  myRole="HotelBrokerWSRequester"
  partnerRole="HotelBrokerWSProvider"/>
....
</partnerLinks>
```

กำหนด partner หรือ เว็บเซอร์วิสของตัวแทนกลุ่มบริการต่างๆ ที่อยู่ในระบบ TIS ทั้งหมด ได้แก่ AirlineBrokerWS, HotelBrokerWS และ CarRentalBrokerWS

(3) กำหนด Variables

```
<variables>
  <variable name="inputVariable"
    messageType="client:TISRequestMessag
    e"/>
  <variable name="outputVariable"
    messageType="client:TISResponseMessa
    ge"/>
  <variable name="invoke_AB"
    messageType="ns1:AirlineBrokerWSReq
    uestMessage"/>
  <variable name="receive_AB"
    messageType="ns1:AirlineBrokerWSRes
    ponseMessage"/>
  ....
</variables>
```

กำหนดตัวแปรสำหรับรับส่งค่าที่เกิดขึ้นในระบบระหว่าง Partner หรือเว็บเซอร์วิสต่างๆ

(4) กำหนด Sequence ของระบบ

```
<sequence name="main">
  <receive name="receiveInput"
    partnerLink="customer"
    portType="client:TIS"
    variable="inputVariable" />
  <invoke name="Invoke_AB"
    partnerLink="AirlineBrokerWS"
    portType="ns1:AirlineBrokerWS"
    inputVariable="invoke_AB"/>
  <invoke name="Invoke_HB"
    partnerLink="HotelBrokerWS"
    portType="ns2:HotelBrokerWS"
    inputVariable="invoke_HB"/>
  ....
  <reply name="replyOutput"
    partnerLink="customer"
    portType="client:TIS"
    inputVariable="outputVariable"/>
</sequence>
```

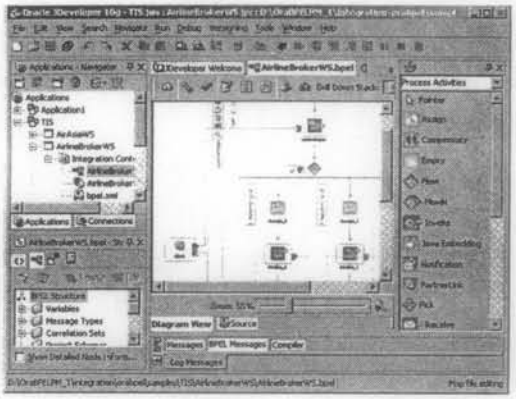
ขั้นตอนสุดท้ายกำหนดถึงลำดับขั้นตอนการทำงาน โดยรับค่าเบื้องต้นจากลูกค้า (receiveInput) แล้วเรียกไปยังตัวแทนบริการจองตั๋วเครื่องบิน (Invoke_AB) ตัวแทนบริการโรงแรม (Invoke_HB) และตัวแทนบริการรถเช่า(Invoke_CR) ตามลำดับ และตอบกลับค่าผลลัพธ์กลับไปให้ลูกค้า (replyOutput)

5.2.3 Deploy to Web Server

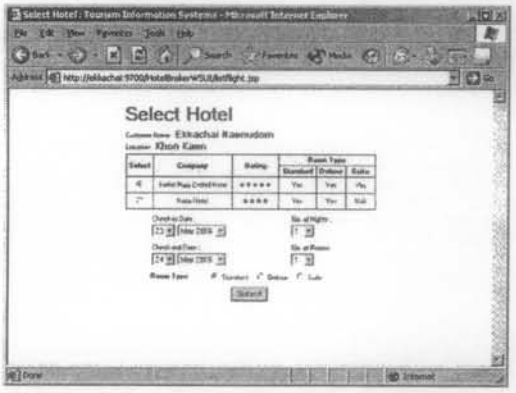
หลังจากการสร้างกระบวนการทางธุรกิจแล้ว จะต้องทำการ Deploy ไปยัง Web Server เพื่อเป็นการนำเอาแบบจำลองการประกอบร่วมของเว็บเซอร์วิสนี้ประกาศขึ้นสู่สาธารณะบน Server จริงพร้อมทำงาน โดยมี User Interface เป็นตัวกลางในการประสานงานรับส่งค่าระหว่างผู้ใช้และองค์ประกอบร่วมของเว็บเซอร์วิสผ่านทางตัวบริหารงาน

6. ผลการวิจัย

ในการพัฒนาระบบ TIS โดยใช้วิธีองค์ประกอบร่วมเว็บเซอร์วิสใช้เครื่องมือในการพัฒนาคือ Oracle JDeveloper BPEL Designer ดังภาพที่ 7 และสร้าง User Interface เพื่อทดสอบระบบ TIS โดยใช้ภาษา JSP ดังตัวอย่างในภาพที่ 8 ซึ่งจะแสดงผลลัพธ์ของรายการโรงแรมและประเภทห้องพัก เมื่อระบบ TIS เรียกไปยังตัวแทนบริการโรงแรม โดยส่งค่าปลายทางคือ ขอนแก่น ที่ได้รับมาจากตัวแทนบริการจองตั๋วเครื่องบินและส่งไปยังตัวบริหารงานเพื่อให้ลูกค้าเลือกผ่านทางเว็บบราวเซอร์ จากผลลัพธ์นี้แสดงให้เห็นว่ามีการรับส่งค่ากันระหว่างตัวแทนบริการจองตั๋วเครื่องบินและตัวแทนบริการโรงแรมผ่านทางระบบ TIS



ภาพที่ 7 หน้าจอของโปรแกรม Oracle JDeveloper



ภาพที่ 8 หน้าจอแสดงผลลัพธ์ของรายการโรงแรมและประเภทห้องพัก

จากผลการทดสอบแบบจำลองการประกอบร่วมของเว็บเซอร์วิสของธุรกิจด้านการท่องเที่ยวผ่านทาง User Interface นั้น เว็บเซอร์วิสของแต่ละธุรกิจบริการสามารถทำงานร่วมกันแบบอัตโนมัติ สามารถรับส่งค่าระหว่างเว็บเซอร์วิสได้อย่างสมบูรณ์ ไม่ว่าจะเป็นการทำงานระหว่าง Web Service Providers ใน Level 2 และระหว่าง Web Service Brokers ใน Level 1

7. บทสรุปและข้อเสนอแนะ

งานวิจัยนี้ได้นำเสนอแบบจำลองของการทำงานร่วมกันของแต่ละเว็บเซอร์วิส ซึ่งประกอบด้วยกลุ่มธุรกิจการจองตั๋วเครื่องบิน ธุรกิจโรงแรม ธุรกิจรถเช่า และธุรกิจด้านการเงิน โดยมีระบบบริการข้อมูลสารสนเทศการท่องเที่ยวหรือระบบ TIS ทำหน้าที่ประสานงานการทำงานร่วมกัน โดยใช้ภาษาที่ใช้สร้างกระบวนการทางธุรกิจ คือภาษา BPEL (Business Process Execution Language) และจากผลการทดสอบเห็นได้ว่า แบบจำลองการประกอบร่วมของเว็บเซอร์วิสนี้สามารถแก้ปัญหาข้อจำกัดในการทำงานร่วมกัน และแลกเปลี่ยนข้อมูล ระหว่างเว็บเซอร์วิส ทำให้การทำธุรกิจระหว่างองค์กรเป็นไปได้ง่ายขึ้น อีกทั้งแบบจำลองการประกอบร่วมของเว็บเซอร์วิสนี้ยังมีความยืดหยุ่นสามารถรองรับการเพิ่มขึ้นของเว็บเซอร์วิสของกลุ่มธุรกิจบริการ (Web Service Broker) หรือการเพิ่มขึ้นของเว็บเซอร์วิสของผู้ให้บริการ (Web Service Provider) ที่จะเกิดขึ้นในอนาคตได้

จากงานวิจัย แม้จะสามารถสร้างแบบจำลองการประกอบร่วมของเว็บเซอร์วิสของธุรกิจท่องเที่ยวและเรียกใช้บริการต่างๆทางธุรกิจแบบครบวงจรได้ แต่ก็ยังขาดการพัฒนาองค์ประกอบอีกบางส่วน เช่น การให้บริการทางด้านการชำระเงิน ซึ่งจะต้องเพิ่มระบบตัดเงินผ่านบัตรเครดิตอัตโนมัติ การสนับสนุนการเพิ่มขึ้นของผู้ร่วมทำธุรกิจในอนาคตได้แบบอัตโนมัติ การสนับสนุนในส่วนของระบบ E-Tickets และการเพิ่มระบบ AI ในการวิเคราะห์และเลือกทางเลือกที่ดีที่สุดมานำเสนอต่อลูกค้า เพื่อให้การบริการการท่องเที่ยวแบบครบวงจรตอบสนองความต้องการของลูกค้าได้สมบูรณ์มากยิ่งขึ้น

8. เอกสารอ้างอิง

[1] W3C-WS, "Web Service Architecture," W3C Working Group,2004 Feb 11,Available from: http://www.w3.org/TR/ws-arch/.

[2] W3C-XML, "Extensible Markup Language (XML) version 1.0 (Second Edition)", World Wide Web Consortium, 2000 Oct 6, Available from: http://www.w3.org/XML/.

[3] W3C-SOAP, "SOAP version 1.2 Part 0: Primer," World Wide Web Consortium, 2003 Jun 24, Available from: http://www.w3.org/TR/soap12-part0/.

[4] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, Web Services Description Language (WSDL) version 1.1, 2001 Mar 15, Available from: http://www.w3.org/TR/wsdl.

[5] UDDI, Universal Description Discovery and Integration, The UDDI Web Site, 2002 Jul 30, Available from: http://www.uddi.org.

[6] BPEL4WS, Business Process Execution Language for Web Services, 2003 May 5, Available from: http://www-128.ibm.com/developerworks/webservices/library/ws-bpel/.

[7] IBM-WSFL, Web Services Flow Language (WSFL), IBM, 2001, Available from: http://www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf.

[8] MS-XLANG, S.Thatte: Web Services for Business Process Design, Microsoft Corporation, 2001, Available from: http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/.

[9] WebSphere Studio Application Developer, Available from: http://www-306.ibm.com/software/awdtools/studioappdev/.

[10] Oracal JDeveloper, Available from: http://otn.oracle.com/bpel/.

[11] Chris Paletz, "Web Services Orchestration. A review of emerging technologies, tools and standards," Hewllett Packard White Paper, January 2003.

[12] Yeung C., Pang-Fei T., and Yen J., "A Multi-Agent Based Tourism Kiosk on Internet", Proceedings of the Thirty-First Annual Hawaii International Conference on System Sciences-Volume 4, Kohala Coast, Hawaii, USA.,1998 January 06-09, pp.452.

[13] Wil M. P. van der Aalst , Marlon Dumas , and Arthur H.M. ter Hofstede, "Web Service Composition Languages: Old Wine in New Bottles?", Proceedings of the 29th Conference on EUROMICRO, Antalya, Turkey, 2003 September 01-06, pp.298.

ประวัติผู้เขียน

นายเอกชัย แน่นอุดร เกิดเมื่อวันที่ 19 ตุลาคม พ.ศ.2524 ณ อำเภอเมือง จังหวัดร้อยเอ็ด

ประวัติการศึกษา

พ.ศ.2546	ปริญญาตรี บริหารธุรกิจบัณฑิต สาขาวิชาคอมพิวเตอร์ธุรกิจ มหาวิทยาลัยมหาสารคาม
พ.ศ.2549	ปริญญาโท วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ มหาวิทยาลัยขอนแก่น ได้รับทุนอุดหนุนการศึกษาจากมหาวิทยาลัยมหาสารคาม และได้รับทุนอุดหนุน ค้นคว้าและวิจัยในการทำวิทยานิพนธ์ จากบัณฑิตวิทยาลัย มหาวิทยาลัยขอนแก่น

ประวัติการทำงาน

พ.ศ.2546 – 2547	นักวิชาการคอมพิวเตอร์ คณะการบัญชีและการจัดการ มหาวิทยาลัยมหาสารคาม
พ.ศ.2548 – 2549	พนักงานวิทยาศาสตร์และเทคโนโลยี คณะการบัญชีและการจัดการ มหาวิทยาลัยมหาสารคาม
พ.ศ.2549 – ปัจจุบัน	อาจารย์ประจำสาขาวิชาคอมพิวเตอร์ธุรกิจ คณะการบัญชีและการจัดการ มหาวิทยาลัยมหาสารคาม