

การพัฒนาเว็บเซอร์วิส
สำหรับ E-BUSINESS ด้วย APACHE AXIS
WEB SERVICE SUPPORT E-BUSINESS WITH APACHE AXIS

นาย นรศักดิ์ ภูผายาง รหัส 44362630

นาย อนันต์ กุลนนท์แดง รหัส 43360304

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ... 25 / พ.ค. 2553 /
เลขทะเบียน... 1501508X
เลขเรียกหนังสือ..... ๕๗.
มหาวิทยาลัยธรรมศาสตร์ ๒๕๑๓.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2546



ใบรับรองโครงการวิศวกรรม


หัวข้อโครงการ การพัฒนาเว็บเซอร์วิส สำหรับ E-Business ด้วย Apache Axis
ผู้ดำเนินโครงการ นาย นราศักดิ์ ภูผายาง รหัส 44362630
 นาย อนันต์ กุลนนท์แดง รหัส 43360304


อาจารย์ที่ปรึกษา อาจารย์พงศ์พันธ์ กิจสนาโยธิน
อาจารย์ที่ปรึกษาร่วม ผู้ช่วยศาสตราจารย์สุชาติ แยมเม่น
 อาจารย์ ดร.พนมขวัญ ริยะมงคล

สาขาวิชา วิศวกรรมคอมพิวเตอร์
ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา 2547

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ โครงการฉบับนี้เป็นส่วนหนึ่งของการศึกษา
ตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์
คณะกรรมการสอบโครงการวิศวกรรม


.....ประธานกรรมการ
(อาจารย์พงศ์พันธ์ กิจสนาโยธิน)


.....กรรมการ
(ผู้ช่วยศาสตราจารย์สุชาติ แยมเม่น)


.....กรรมการ
(อาจารย์ ดร.พนมขวัญ ริยะมงคล)

หัวข้อโครงการ : การพัฒนาเว็บเซอร์วิส สำหรับ E-Business ด้วย Apache Axis
ผู้ดำเนินโครงการ : นาย นราศักดิ์ ภูผายาง รหัส 44362630
นาย อนันต์ กุลนนท์แดง รหัส 43360304
อาจารย์ที่ปรึกษา : อาจารย์พงศ์พันธ์ กิจสนาโยธิน
สาขา : วิศวกรรมคอมพิวเตอร์
ภาควิชา : วิศวกรรมไฟฟ้าและคอมพิวเตอร์

บทคัดย่อ

โครงการฉบับนี้เป็นการศึกษาเกี่ยวกับเทคโนโลยีเว็บเซอร์วิส ซึ่งกำลังได้รับความนิยมในปัจจุบัน โดยใช้ Apache Axis เป็นเครื่องมือในการพัฒนา บริการที่ได้พัฒนาขึ้นคือการแลกเปลี่ยนไฟล์เมนูอาหารระหว่างภัตตาคาร 2 แห่ง เพื่อเป็นการอำนวยความสะดวกให้กับผู้จัดการร้านอาหารในการกำหนดเมนูที่จะให้บริการแก่ลูกค้า เว็บเซอร์วิสนี้จึงถือเป็นการพาณิชย์อิเล็กทรอนิกส์ธุรกิจด้วยกัน (B2B-Business to Business)

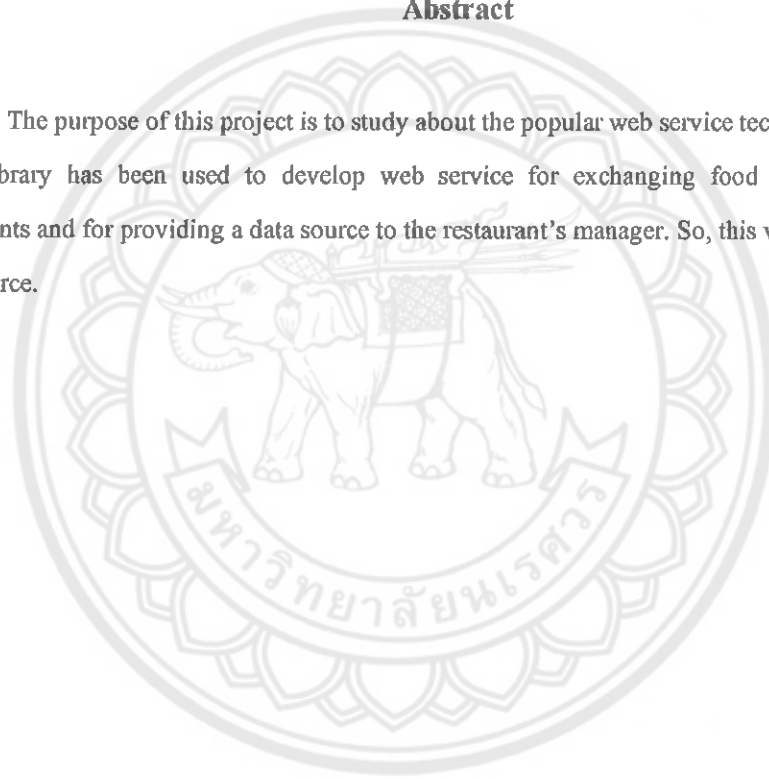


Project Title : Web Services Support E-Business with Apache Axis
Name : Mr.Narasak Poopayang ID 44362630
: Mr.Anan Kunnon dang ID 43360304
Project Adviser : Mr.Phongphun Kijsanayothin
Major : Computer Engineering
Department : Electrical and Computer Engineering
Academic year : 2004

.....

Abstract

The purpose of this project is to study about the popular web service technology. The Apache Axis library has been used to develop web service for exchanging food menus between two restaurants and for providing a data source to the restaurant's manager. So, this web service is B2B E-Commerce.



กิตติกรรมประกาศ

โครงการฉบับนี้คงไม่อาจสำเร็จลงได้ หากไม่ได้รับความช่วยเหลือ และการให้คำแนะนำ
ปรึกษาจาก ท่านอาจารย์พงศ์พันธ์ กิจสนาโยธิน ขอขอบคุณมากครับ

ขอขอบคุณอาจารย์ทุกท่านในภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ มหาวิทยาลัยนเรศวร
ที่ได้สั่งสอนและให้ความรู้แก่เรา ซึ่งสามารถนำไปประยุกต์ใช้ในการทำโครงการนี้ได้อย่างมากมาย
ขอขอบคุณ พี่ก๊อบ ที่ปรึกษาการฝึกงาน บริษัทจีเอเบิ้ล ที่ช่วยอธิบายให้ความกระจ่างในหลักการพัฒนา
เว็บเซอร์วิส

และที่สุดคือพวกเราเอง ต้องขอบคุณซึ่งกันและกันที่เป็นเพื่อนร่วมงานที่ดี จนทำให้โครงการนี้
สำเร็จลงได้

นาย นราศักดิ์ ภูผายาง

นาย อนันต์ กุลนนท์แดง



สารบัญ

หน้า

บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ช
สารบัญรูป	ซ

บทที่ 1 บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของโครงการ.....	2
1.3 ขอบเขตของโครงการ.....	2
1.4 วิธีการดำเนินงาน.....	2
1.5 ขั้นตอนการดำเนินงาน.....	3
1.6 ผลที่คาดว่าจะได้รับ.....	3
1.7 งบประมาณที่ใช้.....	4

บทที่ 2 ทฤษฎีและหลักการ

2.1 พาณิชยอิเล็กทรอนิกส์เบื้องต้น.....	5
2.1.1 พาณิชยอิเล็กทรอนิกส์คืออะไร.....	5
2.1.2 ประเภทของพาณิชยอิเล็กทรอนิกส์.....	6
2.2.3 รูปแบบการค้าพาณิชยอิเล็กทรอนิกส์.....	7
2.1.4 หลักการเลือกรูปแบบพาณิชยอิเล็กทรอนิกส์ที่เหมาะสม.....	8
2.1.5 การพาณิชยอิเล็กทรอนิกส์ในประเทศไทย.....	9
2.2 เว็บเซอร์วิส (Web Service).....	10
2.2.1 ความเป็นมาของเว็บเซอร์วิส.....	10
2.2.2 เปรียบเทียบระหว่าง Web Application และ Web Service.....	13
2.2.3 มาตรฐานที่เกี่ยวข้องกับเว็บเซอร์วิส.....	17
• XML.....	17
- โครงสร้างภาษา XML (Extensible Markup Language).....	18

สารบัญ (ต่อ)

	หน้า
- กฎกติกาเบื้องต้นสำหรับการสร้างข้อมูล XML.....	18
- การกำหนดความถูกต้องของ XML ด้วย DTD.....	20
- การกำหนดความถูกต้องด้วยการอ้างอิงชื่อไฟล์ *.dtd	22
• XSD.....	24
• XSL.....	25
- ทำความรู้จักกับ Style sheet เบื้องต้น.....	25
- การใช้ XSL เพื่อนำเสนอเอกสาร XML.....	25
- หน้าที่ของ XSL.....	25
- พื้นฐานการใช้ XSL Style Sheet.....	26
• SOAP.....	28
• WSDL.....	30
• UDDI.....	35
2.3 Apache Axis Tools.....	37
บทที่ 3 การใช้ Apache Axis พัฒนาเว็บเซอร์วิส	
3.1 Building the Web Services.....	39
3.1.1 ThaiPalaceMenuService.....	39
- menu.xsd.....	39
- menu.xml.....	42
3.1.2 Nutty Palace.....	45
- menu.xsd.....	45
- menu.xml.....	48
3.2 The Menu WSDL.....	57
3.3 The Menu Web Service Client.....	65
3.4 The SiamDelivery Server.....	77
3.5 The Servlet.....	78
3.6 The JavaServer Page.....	81
3.7 XTags library.....	84
3.8 การเรียกใช้งาน SiamDeliveryServer.....	90

สารบัญ (ต่อ)

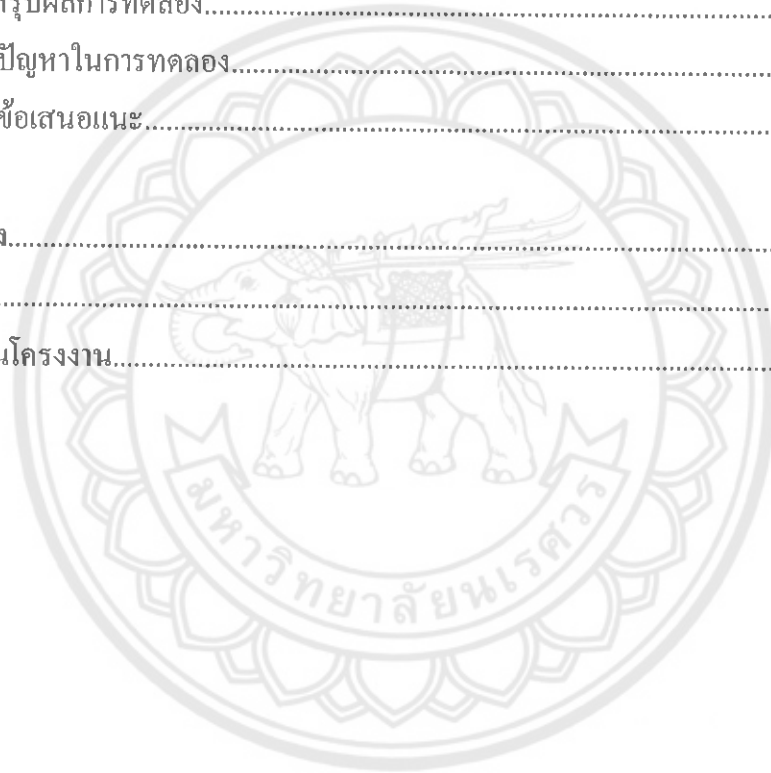
หน้า

บทที่ 4 ผลการทดลอง

4.1 สรุปขั้นตอนการทำงานของเซอร์วิสเมนูอาหาร.....	93
4.2 อธิบายแต่ละส่วนย่อย ๆ พอสังเขป.....	94
• Web Service Menu.....	94
• SiamDelivery.....	97


บทที่ 5 สรุปผลและข้อเสนอแนะ

5.1 สรุปผลการทดลอง.....	100
5.2 ปัญหาในการทดลอง.....	100
5.3 ข้อเสนอแนะ.....	100
เอกสารอ้างอิง.....	102
ภาคผนวก.....	103
ประวัติผู้เขียนโครงการ.....	105



สารบัญตาราง

ตารางที่	หน้า
1.1 แสดงขั้นตอนการดำเนินงาน.....	3
2.1 ประเด็นที่ควรพิจารณาต่อการเลือกรูปแบบพาณิชย์อิเล็กทรอนิกส์.....	8
2.2 อธิบาย Web Services Model.....	13
2.3 เปรียบเทียบคุณสมบัติของ web services และ web application.....	15



สารบัญรูป

รูปที่	หน้า
2.1 Distributed Component Technology.....	12
2.2 แสดง Web Services Model.....	13
2.3 แสดงคำศัพท์เปรียบเทียบระหว่างเว็บธรรมดากับเว็บเซอร์วิส.....	15
2.4 แสดงส่วนประกอบของเทคโนโลยีเว็บเซอร์วิส.....	17
2.5 แสดงหน้าที่การทำงานของ xsl.....	26
2.6 แสดงลักษณะการทำงานของ XSL.....	27
2.7 แสดง XML messaging using SOAP.....	29
2.8 โครงสร้างของเอกสาร SOAP.....	30
2.9 แสดงฐานข้อมูลเว็บเซอร์วิสของบริษัทผู้ค้าทางธุรกิจ (Registry Operation).....	36
2.10 แสดง Web Services with SOAP, UDDI และ WSDL.....	37
3.1 แสดงไฟล์ ThaiPalaceMenuService.wsdl ในรูปแบบ XML.....	54
3.2 แสดงไฟล์ NuttyMenuService.wsdl ในรูปแบบ XML.....	56
3.3 แสดงการ Generating โดยใช้ Axis.....	66
3.4 แสดงหน้า welcome.....	83
3.5 หน้า Welcome.....	90
3.6 หน้า menu ของ Thai Palace.....	91
3.7 หน้า menu ของ Natty Palace.....	92
4.1 สรุปขั้นตอนการทำงานของเว็บเซอร์วิส.....	93
4.2 Thai Palace.....	94
4.3 Nutty Palace.....	95
4.4 การทำงานของ Axis Tool.....	96
4.5 การเชื่อมต่อไฟล์ ThaiPalaceMenuService.wsdl ไปยัง Web Service Menu.....	97
4.6 การเชื่อมต่อไฟล์ NuttyMenuService.wsdl ไปยัง Web Service Menu.....	97
4.7 การแสดงไฟล์ html ทางบราวเซอร์.....	98

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันท่ามกลางสถานะเศรษฐกิจที่เปลี่ยนแปลง ประกอบกับการแข่งขันทางธุรกิจที่สูงขึ้นปัจจัยหลาย ๆ อย่างเข้ามามีอิทธิพลต่อรูปแบบการดำเนินธุรกิจ ทว่าแนวโน้มที่เห็นได้ชัดเจนสำหรับธุรกิจในยุคใหม่คือ การทำธุรกิจในลักษณะอิเล็กทรอนิกส์หรือที่เรียกว่า พาณิชย์อิเล็กทรอนิกส์ (E-Business) ซึ่งเป็นวิธีการทำธุรกิจที่น่าสนใจและกำลังแพร่หลายในหลาย ๆ กิจการ ซึ่งถือเป็นปัจจัยหนึ่งที่มีส่วนทำให้องค์กรทางธุรกิจนั้น ๆ ยืนหยัดอยู่ได้ องค์กรทางธุรกิจต่าง ๆ จึงเรียนรู้ที่จะนำข้อมูลและทรัพยากรด้านเทคโนโลยีสารสนเทศมาใช้เป็นเครื่องมือในการพัฒนาธุรกิจของตนมากยิ่งขึ้น เพื่อสร้างรากฐานที่มั่นคงมุ่งสู่การเติบโตทางเศรษฐกิจในอนาคต

องค์กรทางธุรกิจต่าง ๆ เริ่มมองเห็นวิธีที่ดีที่สุดสำหรับการพัฒนาประสิทธิภาพของการทำงาน ลดค่าใช้จ่าย และสานสัมพันธ์ที่มีกับลูกค้ารวมถึงซัพพลายเออร์รายสำคัญให้แน่นแฟ้นยิ่งขึ้น โดยสร้างวิธีการดำเนินธุรกิจแบบใหม่ที่ใช้เทคโนโลยีผสมผสานกับเครือข่ายอินเทอร์เน็ตที่แพร่หลายในปัจจุบัน ซึ่งเทคโนโลยีที่ได้รับความสนใจและยังเป็นเครื่องมือที่สนับสนุนความต้องการดังกล่าวในธุรกิจยุคใหม่ก็คือ Web Services

ผู้เชี่ยวชาญทางด้านเทคโนโลยีสารสนเทศหลายคนเชื่อว่า ในอนาคตอันใกล้ การพัฒนาเว็บไซต์จะมีลักษณะการให้บริการแบบแยกส่วนตามความชำนาญ และมีการแลกเปลี่ยนข้อมูลกันมากขึ้น บนเว็บไซต์ของหน่วยงานต่างๆ ทั้งภาครัฐ และเอกชน ทำให้การเข้าใช้งานบนหน้าจอเดียวอาจครอบคลุมในทุกบริการ เรียกได้ว่าสามารถทำได้ตั้งแต่เริ่มต้นไปจนจบ (one stop service) โดยบริษัทเอกชนจำนวนมาก เริ่มให้ความสนใจในเรื่องเว็บเซอร์วิสเพราะสามารถสร้างความได้เปรียบ สร้างรายได้ และช่วยลดต้นทุน ความได้เปรียบของเว็บเซอร์วิสที่สำคัญได้แก่ การทำงานข้ามระบบซึ่งมี Platform (ระบบการทำงาน) และภาษาที่ต่างกัน รวมทั้งการประสานงานระหว่างระบบแบบอัตโนมัติ

Web Services มีความสามารถในการแลกเปลี่ยนข้อมูลและประสานงานการทำงานบนระบบที่ต่างกันได้ โดยการติดต่อผู้ใช้ผ่านทาง Application ซึ่งการประยุกต์ใช้ในลักษณะนี้ผู้ใช้จะรู้เพียงว่าเมื่อเข้ามาที่เว็บไซต์ของเราก็จะได้รับบริการที่ครบวงจรเช่นบริการการท่องเที่ยว, บริการสถานที่พักผ่อน และสนทนาการ

Web Services ในประเทศไทย เริ่มเป็นที่รู้จักเมื่อปลายปี 2546 ที่ผ่านมา โดยกระทรวง เทคโนโลยีสารสนเทศและการสื่อสาร (ไอซีที) พยายามที่จะผลักดันให้เกิด โดยสำนักงานส่งเสริมอุตสาหกรรมซอฟต์แวร์แห่งชาติ (องค์การมหาชน) หรือซิป้าได้ตั้งงบประมาณไว้ 100 ล้านบาท ที่จะผลักดันเว็บเซอร์วิส เพื่อสร้างมูลค่าตลาดให้ถึง 3 หมื่นล้านบาท ในปี 2548 ได้คาดการณ์ไว้ว่าบริการดังกล่าวจะมาแรง และเห็นภาพชัดเจนขึ้น เหตุผลที่เว็บเซอร์วิสจะเข้ามา มีบทบาทเพิ่มมากขึ้น เพราะการเจริญเติบโตของอินเทอร์เน็ต เนื่องจากเว็บเซอร์วิสจะเป็นจุดรวมข้อมูล ที่รวมหลายบริการไว้ในที่ ๆ เดียวกัน และเกิดการเชื่อมโยงการทำงาน ซึ่งทำให้ผู้ใช้บริการอินเทอร์เน็ตที่ต้องการข้อมูล หรือสินค้า ใช้บริการได้สะดวกและรวดเร็วขึ้น ถึงแม้ว่าเว็บเซอร์วิสจะเป็นเรื่องใหม่ของไทย แต่ประเทศที่เจริญแล้วไม่ว่าจะเป็นสหรัฐอเมริกาหรือญี่ปุ่น ได้มีการให้บริการดังกล่าวมาแล้ว และค่อนข้างประสบความสำเร็จอย่างสูง ซึ่งในปี 2548 น่าจะเห็นภาพเว็บเซอร์วิสในประเทศไทยได้อย่างชัดเจนประมาณ 40-50% โดยบริษัทเอกชนที่เป็นบริษัทข้ามชาติจะเข้ามาให้บริการก่อน เพราะมีประสบการณ์ในการให้บริการอยู่แล้ว เช่น บริษัทไมโครซอฟท์ ไอบีเอ็ม และซันไมโครซิสเต็ม เป็นต้น

1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 ศึกษารูปแบบของพาณิชย์อิเล็กทรอนิกส์
- 1.2.2 ศึกษาองค์ประกอบของเว็บเซอร์วิส
- 1.2.3 ศึกษาเทคโนโลยีและเครื่องมือที่ช่วยในการพัฒนาเว็บเซอร์วิส
- 1.2.4 ศึกษาขั้นตอนการทำเว็บเซอร์วิสจากเครื่องมือที่ช่วยในการพัฒนา
- 1.2.5 จำลองตัวอย่างของเว็บเซอร์วิสในรูปแบบของพาณิชย์อิเล็กทรอนิกส์

1.3 ขอบเขตของโครงการ

- 1.3.1 นำเอารูปแบบของพาณิชย์อิเล็กทรอนิกส์ที่เหมาะสมมาใช้
- 1.3.2 เลือกเอาเครื่องมือที่เหมาะสมมาประยุกต์ใช้ในการพัฒนาเว็บเซอร์วิส
- 1.3.3 จำลองเว็บเซอร์วิสโดยใช้เครื่องมือที่เลือก เพื่อให้เกิดความเข้าใจในเว็บเซอร์วิสมากขึ้น

1.4 วิธีการดำเนินงาน

- 1.4.1 ศึกษาทฤษฎีพื้นฐานต่าง ๆ ที่เกี่ยวข้องกับการพาณิชย์อิเล็กทรอนิกส์
- 1.4.2 ศึกษาองค์ประกอบต่าง ๆ ที่เกี่ยวข้องกับการสร้างเว็บเซอร์วิส
- 1.4.3 ออกแบบโครงสร้างที่จะใช้ในการจำลองเว็บเซอร์วิส

1.4.4 ศึกษาเครื่องมือที่ใช้ในการพัฒนาเว็บเซอร์วิสเพื่อให้สร้างเว็บเซอร์วิสได้โดยง่าย ซึ่งโครงการนี้เลือกใช้ Apache Axis Tools เป็นเครื่องมือพัฒนา

1.4.5 จำลองเว็บเซอร์วิสขึ้นมาแล้วทดสอบการทำงาน พร้อมทั้งหาข้อผิดพลาดที่เกิดขึ้น

1.4.6 แก้ไขข้อผิดพลาด จัดทำคู่มือโครงการและสรุปผล

1.5 ขั้นตอนการดำเนินงาน

การดำเนินงานของโครงการพัฒนาเว็บเซอร์วิสสำหรับ E-Business ด้วย Apache Axis มีกำหนดการทำงาน ดังนี้

ตารางที่ 1.1 แสดงขั้นตอนการดำเนินงาน

กิจกรรม	พ.ศ. 2547					พ.ศ. 2548				
	มี.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
รวบรวมข้อมูลและกำหนดขอบเขตโครงการ	↔									
ศึกษาพาณิชย์อิเล็กทรอนิกส์เบื้องต้น		↔								
ศึกษา จาวาเว็บเซอร์วิส (Java Web Service)			↔							
ศึกษาการใช้งาน Apache Axis Tools				↔						
พัฒนาระบบ Web Services					↔	↔				
แก้ไขส่วนบกพร่องของระบบ								↔		
จัดทำคู่มือโครงการ									↔	
ส่งโครงการที่สมบูรณ์										↔

1.6 ผลที่คาดว่าจะได้รับ

1.6.1 ได้รู้ถึงรูปแบบพาณิชย์อิเล็กทรอนิกส์ สามารถที่จะนำมาประยุกต์ใช้ให้เหมาะสมกับการทำการค้าบนเครือข่ายอินเทอร์เน็ต

1.6.2 เข้าใจส่วนประกอบของเว็บเซอร์วิส และสามารถเลือกเครื่องมือที่เหมาะสมมาใช้ในการพัฒนาได้

1.6.3 สามารถนำเทคโนโลยีเว็บเซอร์วิสมาใช้ในการทำการค้าพาณิชย์อิเล็กทรอนิกส์

1.7 งบประมาณที่ใช้

1. ค่าถ่ายเอกสาร	500	บาท
2. ค่าจัดทำรายงาน	1,000	บาท
3. ค่าตำราและหนังสืออ้างอิง	500	บาท

รวมค่าใช้จ่ายเป็นเงิน 2,000 บาท

(สองพันบาทถ้วน)



บทที่ 2

ทฤษฎีและหลักการ

2.1 พาณิชย์อิเล็กทรอนิกส์เบื้องต้น

2.1.1 พาณิชย์อิเล็กทรอนิกส์คืออะไร

พาณิชย์อิเล็กทรอนิกส์ คือ การแลกเปลี่ยนข้อมูลธุรกิจผ่านสื่ออิเล็กทรอนิกส์ ในคำจำกัดความนี้มีคำ 2 คำที่เป็นหัวใจของพาณิชย์อิเล็กทรอนิกส์ คือ “แลกเปลี่ยนข้อมูลธุรกิจ” และ “ผ่านสื่ออิเล็กทรอนิกส์”

การแลกเปลี่ยนข้อมูลธุรกิจนั้น อาจอยู่ในรูปคำสั่งซื้อระหว่างร้านค้าและผู้ผลิตสินค้า บริษัทกับรัฐบาล หรือแม้กระทั่งการสื่อสารภายในองค์กรเองก็ได้ ขอเพียงให้มีการแลกเปลี่ยนทางธุรกิจก็จัดว่าเป็นพาณิชย์อิเล็กทรอนิกส์แล้ว

การแลกเปลี่ยนข้อมูลผ่านสื่ออิเล็กทรอนิกส์นั้นก็ได้จำกัดว่าต้องเป็นสื่อประเภทใด แม้จะใช้เพียงแผ่นซีดีรอม (CD-ROM) บันทึกข้อมูลเพื่อติดต่อกัน หรือการโอนเงินระหว่างธนาคาร การจองตั๋วภาพยนตร์ผ่านโทรศัพท์มือถือ ก็ล้วนเป็นพาณิชย์อิเล็กทรอนิกส์ทั้งสิ้น

แต่อย่างไรก็ตามพาณิชย์อิเล็กทรอนิกส์ที่ได้รับความนิยมมากที่สุดก็คือ พาณิชย์อิเล็กทรอนิกส์ผ่านเครือข่ายอินเทอร์เน็ต (Internet Commerce) ซึ่งก่อให้เกิดนวัตกรรมอยู่ 3 ประการ

ประการหนึ่ง เกิดตลาดใหม่ เป็นครั้งแรกที่ผู้ผลิตสินค้าสามารถจำหน่ายสินค้าของตนให้กับลูกค้าทั่วโลกได้โดยไม่ต้องผ่านคนกลาง อย่างไรก็ตามไม่ได้หมายความว่าอินเทอร์เน็ตจะเป็นสื่อที่เหมาะสมกับการทำพาณิชย์ได้ทุกประเภทสินค้า จึงไม่ควรหวังว่าจะมีคนเข้ามาสั่งซื้อสินค้าในเว็บไซต์ของเราเพียงอย่างเดียว แต่จะต้องทำการตลาดเชิงรุกด้วย

ประการสอง เกิดกระบวนการทำงานแบบใหม่ วิธีการประกอบธุรกิจที่ทำอยู่ดั้งเดิม เช่น การบริหารการขาย การลงบัญชีด้วยการใช้คอมพิวเตอร์ขนาดใหญ่เช่นในอดีต สามารถปรับรูปแบบมาเป็นการให้บริการผ่านเครื่องแม่ข่ายกลางในเครือข่ายอินเทอร์เน็ต (ASP - Application Service Provider) ได้ เช่น เว็บไซต์ salesforce.com, oraclesaleonline.com ซึ่งให้บริการบริหารงานขาย และ netledger.com ที่ให้บริการบริหารระบบบัญชี ซึ่งทำให้บริษัทไม่ต้องเสียเงินลงทุนทางด้านระบบคอมพิวเตอร์มากมาย ทั้งในด้านซอฟต์แวร์ฮาร์ดแวร์ รวมทั้งบุคลากร

ประการที่สาม เกิดผลิตภัณฑ์ใหม่ ได้แก่ ข้อมูลที่ส่งผ่านทางเครือข่ายอินเทอร์เน็ต ไม่ว่าจะเป็นเพลง ภาพยนตร์ ข่าวสาร ซึ่งข้อมูลในรูปแบบนี้มีชื่อคืออยู่ 3 ประการ คือ หนึ่ง ต้นทุนหน่วยสุดท้าย (Marginal

Cost) เท่ากับศูนย์ ผู้ผลิตได้ก่อนสามารถผูกขาดตลาดได้ สอง ได้รับการยกเว้นภาษีนำเข้าภายใต้กรอบขององค์การการค้าโลก (WTO) ตามข้อเรียกร้องของสหรัฐ เมื่อ พ.ศ. 2541 สาม ไม่มีต้นทุนในการจัดส่ง

2.1.2 ประเภทของพาณิชย์อิเล็กทรอนิกส์

คู่ค้าที่แลกเปลี่ยนข้อมูลทางธุรกิจกันสามารถจำแนกเป็น 3 กลุ่มใหญ่ คือ ธุรกิจ รัฐบาล และผู้บริโภค หรือประชาชน ซึ่งมีการจับคู่เพื่อทำการค้ากันอยู่ 6 ประเภทที่ไม่ซ้ำกัน ดังนี้

หนึ่ง พาณิชย์อิเล็กทรอนิกส์ประเภทธุรกิจกับธุรกิจ (B2B - Business to Business) คือ การซื้อขายระหว่างผู้ผลิตด้วยกัน เป็นการค้าขนาดใหญ่ ปริมาณมาก ๆ จำนวนเงินมาก ๆ การชำระเงินจะใช้ระบบเครดิต ในส่วนการพัฒนาาระบบเพื่อรองรับการซื้อขายแบบนี้จะมีความซับซ้อนมากกว่าการพาณิชย์อิเล็กทรอนิกส์ประเภทอื่น ๆ

สอง พาณิชย์อิเล็กทรอนิกส์ประเภทธุรกิจกับรัฐบาล (B2G-Business to Government) หมายถึง การสร้างระบบให้รัฐบาลสามารถติดต่อกับภาคธุรกิจผ่านเครือข่ายอินเทอร์เน็ต ตัวอย่างเช่น การประมูลออนไลน์ เพื่อให้ธุรกิจยื่นแบบประกวดราคาผ่านเครือข่ายอินเทอร์เน็ต ซึ่งทำให้ลดต้นทุนการทำธุรกิจในเรื่องการเดินทางติดต่อกับหน่วยงานราชการหลายเที่ยวได้

สาม พาณิชย์อิเล็กทรอนิกส์ประเภทธุรกิจกับผู้บริโภค (B2C - Business to Consumer) คือ การที่ธุรกิจขายสินค้าหรือบริการโดยตรงให้กับผู้บริโภคในลักษณะการค้าปลีก การชำระเงินส่วนใหญ่จะกระทำผ่านระบบบัตรเครดิต ซึ่งการพาณิชย์อิเล็กทรอนิกส์ประเภทนี้ถือเป็นจุดเริ่มต้นที่ทำให้เกิดการพาณิชย์อิเล็กทรอนิกส์ประเภทธุรกิจกับธุรกิจ ซึ่งทั้งสองแบบมีจุดประสงค์ที่ต่างกัน แต่หลาย ๆ องค์กรก็ทำทั้งสองอย่างควบคู่กันไป

สี่ พาณิชย์อิเล็กทรอนิกส์ประเภทรัฐบาลกับรัฐบาล (G2G- Government to Government) คือ การแลกเปลี่ยนข้อมูลธุรกิจภายในหน่วยงานภาครัฐหรือระหว่างรัฐต่อรัฐ เช่น การตรวจสอบสิทธิการรักษาโรคตามโครงการ 30 บาท ของสำนักงานประกันสุขภาพแห่งชาติ ที่มีการเชื่อมกับฐานข้อมูลทะเบียนราษฎร์แบบออนไลน์กับสำนักงานบริหารการทะเบียน (สนบท.) ในสังกัดกระทรวงมหาดไทย หรือ โครงการแลกเปลี่ยนข่าวกับการพัฒนารถไฟรางคู่ระหว่างรัฐบาลไทยกับอินเดีย เป็นต้น

ห้า พาณิชย์อิเล็กทรอนิกส์ประเภทรัฐบาลกับประชาชน (G2C-Government to Citizen) คือ การที่หน่วยงานรัฐบาลให้บริการประชาชนผ่านเว็บไซต์ เช่น เว็บไซต์ตรวจสอบผลสอบเข้ามหาวิทยาลัย การยื่นแบบภาษีเงินได้บุคคลธรรมดาผ่านเว็บไซต์ของกรมสรรพากร

หก พาณิชย์อิเล็กทรอนิกส์ประเภทผู้บริโภคกับผู้บริโภค (C2C - Consumer to Consumer) คือ การซื้อขายกันโดยตรงระหว่างผู้บริโภค เช่น การขายรถหรือบ้านมือสอง รูปแบบที่นิยมคือ การประมูลออนไลน์

ผ่านเว็บไซต์ เช่น eBay.com อย่างไรก็ตาม ปัญหาของพาณิชย์อิเล็กทรอนิกส์ประเภทนี้คือความเชื่อใจระหว่างผู้ซื้อและผู้ขาย เพราะส่วนใหญ่ไม่รู้จักกันมาก่อน ดังนั้นการแลกเปลี่ยนสินค้าจริงจะกระทำการนัดหมายกันและชำระเงินสดพร้อมกับส่งมอบสินค้า ในแง่ของการพัฒนาระบบเพื่อรองรับการซื้อขายแบบนี้ นั้น ถือว่าทำได้ง่ายกว่าการพาณิชย์อิเล็กทรอนิกส์ประเภทอื่น

2.1.3 รูปแบบการค้าพาณิชย์อิเล็กทรอนิกส์

หนึ่ง รายการสินค้าออนไลน์ (Online Catalogue) ได้แก่ ร้านค้าบนอินเทอร์เน็ตที่แสดงรายการสินค้าเพียงอย่างเดียวไม่มีการซื้อขายกัน มีวัตถุประสงค์เพื่อสร้างภาพพจน์บริษัท และให้ข้อมูลสินค้าเบื้องต้น แก่กลุ่มเป้าหมายคือผู้ค้าส่ง การสั่งซื้อมักกระทำผ่านไปรษณีย์อิเล็กทรอนิกส์ (E-mail) หรือโทรสาร (Fax) และชำระเงินโดยการโอนผ่านทางธนาคาร

สอง ร้านค้าปลีก (E-Tailer) การค้ารูปแบบนี้ร้านค้าจะเสนอขายสินค้าหรือบริการให้แก่ผู้บริโภคโดยตรงทางเว็บไซต์ ซึ่งจะระบุราคาสินค้าและค่าขนส่งอย่างชัดเจน การรับคำสั่งซื้อกระทำโดยระบบอัตโนมัติผ่านเว็บไซต์ และนิยมรับชำระเงินผ่านบัตรเครดิต เพื่อเพิ่มความสะดวกให้กับผู้ซื้อ

สาม การประมูลสินค้า (Auction) การค้ารูปแบบนี้เจ้าของสินค้าจะเข้ามาใช้บริการเว็บไซต์ประมูลสินค้า ซึ่งก็เป็นไปได้อยู่สองทาง คือ

- ประเภทแรก ผู้ขายเสนอขายก่อน แล้วให้ผู้ซื้อแข่งขันกันเสนอราคาซื้อภายในระยะเวลาที่กำหนด ซึ่งผู้เสนอราคาซื้อสูงที่สุดจะเป็นผู้ได้รับสิทธิ์ในการซื้อสินค้าเช่นเดียวกับการประมูลปกติเว็บไซต์ประมูลที่ให้บริการลักษณะนี้ได้แก่ eBay.com, Amazon.com
- ประเภทที่สอง ผู้ซื้อจะเป็นผู้เสนอซื้อก่อน แล้วให้ผู้ขายแข่งขันกันเสนอราคาขายภายในระยะเวลาที่กำหนด ผู้เสนอราคาขายต่ำสุดจะได้รับสิทธิ์ขายสินค้านั้นกับผู้ซื้อ การประมูลแบบนี้จึงเรียกว่าการประมูลย้อนกลับ ตัวอย่างเช่น เว็บไซต์ Priceline.com นอกจากนี้กลุ่มผู้ซื้อที่เป็นกลุ่มธุรกิจขนาดใหญ่หรือมีปริมาณการสั่งซื้อมาก เช่น บริษัทเจนเนอรัลมอเตอร์ (Generalmotor) หรือ จีอี อาจสร้างเว็บไซต์เพื่อทำการจัดซื้อโดยให้ผู้ขายแข่งขันเสนอราคาซึ่งอาจเรียกรูปแบบนี้ว่าระบบจัดซื้ออิเล็กทรอนิกส์ รูปแบบการขายแบบนี้ในประเทศไทยก็มีที่เว็บไซต์ Pantavani.com ซึ่งเป็นการรวมการจัดซื้อของกลุ่มธุรกิจขนาดใหญ่ เช่น เครื่องเจริญโภคภัณฑ์ ธนาคารกรุงเทพ ฯลฯ เป็นต้น

สี่ การประกาศขายสินค้าทางกระดานแสดงความคิดเห็น (Web Board) เช่น เว็บไซต์ Pantip.com, Sanook.com ซึ่งมีกระดานแสดงความคิดเห็นเฉพาะ สำหรับซื้อขายสินค้าโดยไม่คิดค่าใช้จ่าย คล้ายกับโฆษณาย่อในหนังสือพิมพ์ ซึ่งส่วนมากผู้ที่ต้องการขายสินค้าจะลงประกาศข้อความ และให้เบอร์โทรศัพท์หรือ

ไปรษณีย์อิเล็กทรอนิกส์ เพื่อให้ผู้ซื้อติดต่อมา ทั้งนี้เนื่องจากการค้าส่วนใหญ่อยู่ภายในประเทศ ซึ่งเมื่อตกลงราคากันเรียบร้อยแล้วผู้ขายจะนัดส่งมอบสินค้าให้กับผู้ซื้อพร้อมรับชำระเงินทันที

ท่า ตลาดกลางอิเล็กทรอนิกส์ (E-Marketplace) คือ เว็บไซต์ที่ให้บริการซื้อขายสินค้าเฉพาะบางชนิดเท่านั้น เช่นเดียวกับตลาดไทที่เป็นตลาดกลางสำหรับสินค้าเกษตร ตัวอย่างเว็บไซต์ในประเทศไทยก็เช่น - foodmarketexchange.com ซึ่งการสร้างตลาดกลางขึ้นมานี้ก็เพื่อลดค่าใช้จ่ายทางการตลาดสำหรับโฆษณาเว็บไซต์ของบริษัท หรือการเข้าไปประกาศขายสินค้าทางเว็บไซต์ชุมชน ซึ่งโอกาสที่จะขายสินค้าได้มีไม่มาก เพราะถึงแม้จะมีผู้เข้าชมเว็บไซต์ชุมชนเป็นจำนวนมาก แต่ก็มีความต้องการที่หลากหลาย จึงอาจไม่พบสินค้าของเรา การซื้อขายทางตลาดกลางอิเล็กทรอนิกส์ ผู้ขายต้องสมัครเป็นสมาชิกของเว็บไซต์เหล่านี้ก่อนให้บริการ

2.1.4 หลักการเลือกรูปแบบพาณิชย์อิเล็กทรอนิกส์ที่เหมาะสม

การทำพาณิชย์อิเล็กทรอนิกส์จะต้องเลือกรูปแบบให้เหมาะกับธุรกิจของตนเองให้มากที่สุดซึ่งจะต้องคำนึงถึงองค์ประกอบต่าง ๆ ดังตารางที่ 2.1

ตารางที่ 2.1 ประเด็นที่ควรพิจารณาต่อการเลือกรูปแบบพาณิชย์อิเล็กทรอนิกส์

รูปแบบการค้า	ค่าใช้จ่าย	โอกาสทางการตลาด	ความเป็นอิสระ	ความเสี่ยง
E-Tailer	ต่ำ-ปานกลาง	สูง	สูง	สูง
Auction	ปานกลาง	สูง	ปานกลาง-สูง	ปานกลาง
Web Board	ไม่มี	ต่ำ	ต่ำ	ต่ำ
Marketplace	สูง	สูง	ต่ำ	ต่ำ

หนึ่ง ค่าใช้จ่าย จะสูงในกรณีที่ผู้ประกอบการขนาดกลางหรือขนาดย่อมสร้างเว็บไซต์ขายสินค้าขึ้นเอง ซึ่งจะมีค่าใช้จ่ายในระยะยาว เช่น ค่าจ้างผู้ดูแลเว็บไซต์และระบบคอมพิวเตอร์ จึงควรใช้รูปแบบพาณิชย์อิเล็กทรอนิกส์จากผู้ให้บริการที่มีอยู่แล้ว โดยอาจเริ่มจากการประกาศขายสินค้าในเว็บบอร์ดต่างๆ ที่ไม่คิดค่าใช้จ่าย การสร้างเว็บไซต์โดยใช้ระบบสำเร็จรูป (Template) ซึ่งทำให้ผู้ประกอบการมีเว็บไซต์ของตัวเองอย่างรวดเร็ว โดยมีค่าใช้จ่ายค่อนข้างต่ำ แต่หากผู้ประกอบการมีเงินทุนและลักษณะสินค้าเป็นตลาดเฉพาะ การเลือกใช้บริการตลาดกลางอิเล็กทรอนิกส์ ก็อาจมีความเหมาะสม เพราะเป็นแหล่งพบปะทางการค้าที่ลูกค้าต้องการสินค้าเฉพาะอยู่แล้ว

สอง โอกาสทางการตลาด สำหรับการสร้างเว็บไซต์ที่พร้อมรับคำสั่งซื้อนั้นจะเป็นโอกาสให้ผู้ประกอบการขยายตลาดได้ง่าย อย่างไรก็ตามการประชาสัมพันธ์เว็บไซต์ของผู้ประกอบการก็มีค่าใช้จ่ายสูง ดังนั้นการร่วมกับเว็บไซต์ที่มีชื่อเสียงอยู่แล้ว และทำธุรกิจในรูปแบบการประมูลหรือซื้อขายในตลาดกลางอิเล็กทรอนิกส์ ก็จะมีโอกาสทางการตลาดสูงกว่าเมื่อเทียบกับรูปแบบอื่น ๆ โดยเฉพาะการประมูลกับหน่วยงานภาครัฐนั้นกำหนดให้ใช้บริการกับผู้ที่ยื่นทะเบียนไว้แล้ว เท่านั้น

สาม ความเป็นอิสระ การใช้บริการในเว็บไซต์ประมูลและตลาดกลางอิเล็กทรอนิกส์ นั้นทำให้ผู้ประกอบการขาดโอกาสที่จะสร้างชื่อของร้านตัวเอง ดังนั้นจำเป็นต้องผูกกับผู้ให้บริการค่อนข้างมาก ส่วนการสร้างเว็บไซต์และจดทะเบียนชื่อเว็บไซต์ของตนเองนั้นทำให้ผู้ประกอบการมีความเป็นอิสระสูง หากต้องการขยายกิจการในอนาคตโดยเปลี่ยนผู้ให้บริการก็สามารถทำได้อย่างเป็นอิสระ ซึ่งต้องตรวจสอบให้แน่นอนก่อนว่า ตนเองเป็นผู้ครอบครองสิทธิ์ในเว็บไซต์หรือชื่อโดเมน (Domain Name) นั้น เมื่อว่าจ้างผู้ให้บริการจดทะเบียนให้

สี่ ความเสี่ยงจากการทำธุรกรรม การทำธุรกรรมพาณิชย์อิเล็กทรอนิกส์โดยรับชำระเงินด้วยบัตรเครดิตนั้นเป็นที่นิยมในรูปแบบร้านค้าปลีก (E-Tailer) ซึ่งก็ยังมีความเสี่ยงสูงอยู่ ดังนั้นหากผู้ประกอบการได้รับคำสั่งซื้อในปริมาณที่สูง ก็ควรขอให้ธนาคารช่วยตรวจสอบก่อน หรือขอให้ลูกค้าใช้รูปแบบการชำระเงินที่มีความปลอดภัยขึ้น ส่วนการทำพาณิชย์รูปแบบอื่นนั้น มักจะชำระเงินด้วยวิธีดั้งเดิม เช่น โอนเงินเข้าธนาคาร หรือเปิด Letter of Credit ซึ่งเป็นวิธีที่ธนาคารเข้ามาค้ำประกันทำให้ความเสี่ยงจากการทำธุรกรรมต่ำ

2.1.5 การพาณิชย์อิเล็กทรอนิกส์ในประเทศไทย

การพาณิชย์อิเล็กทรอนิกส์ในประเทศไทยเกิดขึ้นพร้อม ๆ กับการกำเนิดของอินเทอร์เน็ตเชิงพาณิชย์ ซึ่งพอจำแนกวิวัฒนาการออกเป็น 3 ยุค ดังนี้

ยุคที่ 1 พ.ศ. 2537-2540

เริ่มแรกเป็นช่วงที่นิยมสร้างเว็บไซต์ขึ้นมาเพื่อแนะนำบริษัทของตนซึ่งยังไม่มีมารับคำสั่งซื้อ ต่อมาจึงพัฒนาเว็บไซต์พาณิชย์อิเล็กทรอนิกส์ประเภทธุรกิจกับผู้บริโภคขึ้น แต่ก็ยังมีปัญหาที่ระบบการชำระเงิน เพราะธนาคารมองว่าการชำระเงินด้วยบัตรเครดิตทางอินเทอร์เน็ตยังมีความเสี่ยงสูงอยู่ จึงใช้วิธีโอนเงินเข้าบัญชีแทน ซึ่งถึงแม้จะปลอดภัยแต่ก็ไม่สะดวกสำหรับลูกค้า

ยุคที่ 2 พ.ศ. 2541-2543

ได้มีการสร้างสภาพแวดล้อมให้เอื้อต่อการทำพาณิชย์อิเล็กทรอนิกส์มากขึ้น เช่น พัฒนาระบบการชำระเงิน อีกอย่างคือในช่วงนี้กระแสการซื้อขายเว็บไซต์ระบวมมาถึงประเทศไทย ทำให้เกิดภาพมิดเบือนในการทำธุรกิจ กล่าวคือ ไม่ได้ทำเว็บไซต์มาขายสินค้า แต่ทำมาเพื่อขายเว็บไซต์เองให้กับบริษัทร่วมทุน

(Venture Capital) แต่สุดท้ายเว็บไซต์เหล่านี้ก็ได้ล่มสลายไปเป็นจำนวนมากเมื่อตลาดหุ้นสหรัฐประสบวิกฤตในเดือนเมษายน 2543

ยุคที่ 3 พ.ศ. 2544-ปัจจุบัน

ได้มีการสร้างแบบจำลองทางธุรกิจให้เหมาะกับประเทศไทยมากขึ้น โดยเปลี่ยนทัศนคติที่ว่า เมื่อสร้างเว็บไซต์แล้วจะเกิดลูกค้ากลุ่มใหม่เข้ามาสร้างยอดขายเพิ่ม นอกจากนี้เว็บไซต์ที่มีกลุ่มบริษัทใหญ่สนับสนุน ได้รับการส่งเสริมจากรัฐบาลในเรื่องการจัดซื้อผ่านทางอิเล็กทรอนิกส์ (e-Procurement) ทำให้เกิดพาณิชย์อิเล็กทรอนิกส์รูปแบบธุรกิจกับธุรกิจ และธุรกิจกับรัฐบาลขึ้น

2.2 เว็บเซอร์วิส (Web Service)

2.2.1 ความเป็นมาของเว็บเซอร์วิส

- การพัฒนาของเทคโนโลยีเว็บ สามารถแบ่งออกได้เป็น 3 ยุค ดังนี้

ยุคที่ 1 สเตติกเว็บเพจ (Static webpage) เป็นยุคเริ่มต้นของการใช้งานเว็บ มีการวางข้อมูลเป็นไฟล์ตามมาตรฐาน HTML เก็บไว้บนเซิร์ฟเวอร์ ผู้ใช้งานสามารถใช้บราวเซอร์เรียกข้อมูลบนเว็บเซิร์ฟเวอร์ด้วยโปรโตคอล HTTP โดยลักษณะข้อมูลยังเป็นแบบสเตติก คือ ผลลัพธ์ที่ได้จะเป็นไฟล์ที่ถูกเก็บไว้บนเซิร์ฟเวอร์ก่อนแล้วเท่านั้น ยังไม่สามารถคำนวณ ประมวลผลหรือเชื่อมโยงกับฐานข้อมูลอื่นได้

ในยุคสเตติกเว็บเพจได้ทำการเก็บข้อมูลบนเว็บเซิร์ฟเวอร์ ให้ผู้ใช้ สามารถเรียกใช้งานได้ง่ายๆ การพัฒนาระบบบริการข้อมูลเว็บเพจจึงได้รับความสนใจอย่างกว้างขวาง

ยุคที่ 2 ไดนามิกเว็บเพจ (Dinamic Webpage) มีการพัฒนาโปรแกรมทางฝั่งเซิร์ฟเวอร์ เพื่อให้เซิร์ฟเวอร์ทำงานตามคำเรียกขอจากฝั่งไคลเอนต์ ผ่านทางโปรโตคอล HTTP และมีการเชื่อมโยงกับโปรแกรมเฉพาะ ตัวอย่างเช่น เมื่อเกิดการเรียกใช้ บราวเซอร์จะร้องขอมาที่เซิร์ฟเวอร์ เซิร์ฟเวอร์จะทำการตอบสนองโดยรันโปรแกรมหรือเรียกโปรแกรมอื่นที่ทางฝั่งเซิร์ฟเวอร์เตรียมไว้

ในยุคไดนามิกเว็บเพจ มีการสร้างระบบการเรียกเข้าหาโปรแกรม ที่เรียกว่า CGI (Common Gateway Interface) การเขียนโปรแกรมมีเทคนิควิธีการพิเศษที่ทำให้สามารถเข้าถึงฐานข้อมูล มีการปรับปรุงเครื่องมือการพัฒนา และภาษาสำหรับการพัฒนา เพื่อให้สามารถเรียกข้อมูลจากฐานข้อมูลหรือเพิ่มข้อมูลส่งให้ผู้เรียกใช้ได้

ภาษาที่นิยมใช้ในการพัฒนา ได้แก่ ภาษาซี PERL และเครื่องมือพิเศษที่ใช้เชื่อมโยงกับฐานข้อมูล
ที่ผู้ผลิตซอฟต์แวร์ดำเนินการให้ ตลอดจนมีการใช้โปรแกรมทางฝั่งไคลเอนต์ที่ซับซ้อนขึ้น

เหตุผลที่เรียกว่าไดนามิกเว็บเพจ เพราะข้อมูลจะแปรเปลี่ยนตามการเรียกขอ การปรับเปลี่ยนข้อมูล
กระทำตามโปรแกรมที่สร้างขึ้น ซึ่งสามารถคำนวณหรือประมวลผลข้อมูลได้ เช่น การเรียกดูข้อมูลสรุปผล
หรือสถิติต่าง ๆ ในแต่ละวัน ดังนั้นข้อมูลที่ดึงถึงทันทีโดยระบบเซิร์ฟเวอร์จะเชื่อมโยงกับฐานข้อมูล
หรือระบบงานอื่น เมื่อส่งข้อมูลไปยังผู้ใช้ที่เรียกขอเข้ามา จะแปลงข้อมูลให้อยู่ในรูปแบบของ HTML ก่อน
เพื่อให้เบราว์เซอร์แสดงผลได้

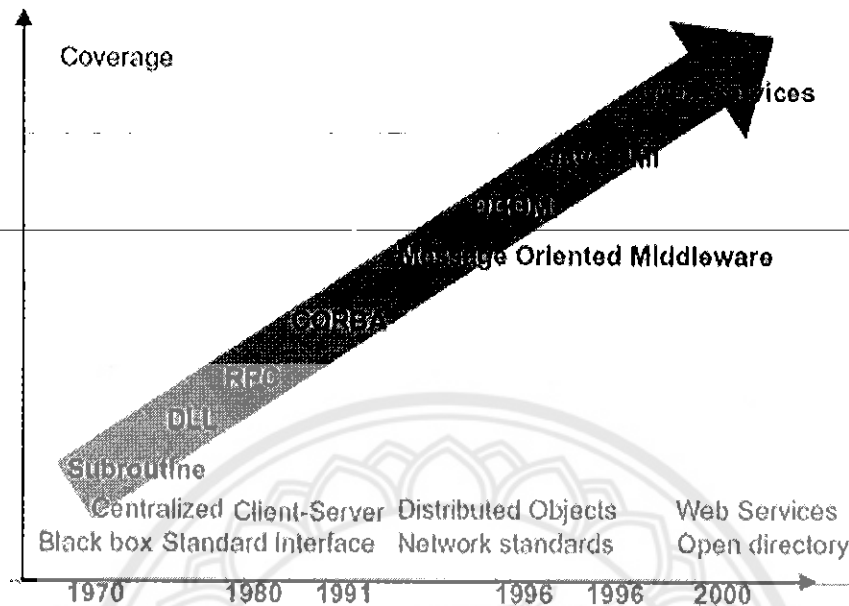
- ปัญหาในยุคไดนามิกเว็บเพจ การพัฒนาเว็บแอปพลิเคชันในยุคไดนามิกเว็บเพจมีความ

สลับซับซ้อนมากขึ้น ทั้งนี้เนื่องจากองค์ประกอบที่ทำงานอยู่ทางฝั่งเซิร์ฟเวอร์มีความซับซ้อนขึ้น เพราะไม่ได้
มีเพียงแต่เว็บเซิร์ฟเวอร์เท่านั้น แต่ยังรวมไปถึง คาด้านเบสเซอร์เวอร์ เมล์เซิร์ฟเวอร์ ตลอดจนแหล่งข้อมูลที่มีแต่
เดิมอยู่แล้วในองค์กร (Legacy System) ด้วยเหตุนี้จึงมีศัพท์สากลที่เรียกแอปพลิเคชันในลักษณะนี้ว่า “3-Tier”
หรือ “Distributed application” ประกอบไปด้วย ฝั่งไคลเอนต์ (Client Tier หรือ Presentation Tier) ฝั่งเว็บ
เซิร์ฟเวอร์ (Business Logic Tier) และฝั่งข้อมูล (Data Tier)

ดังนั้นการพัฒนาเว็บแอปพลิเคชันที่มีระบบซับซ้อน จึงจำเป็นต้องมีการพัฒนาแพลตฟอร์ม และ
โปรโตคอลที่มีความพร้อมมารองรับ ในยุคนี้ผู้พัฒนาค่ายต่าง ๆ ได้ทำการพัฒนาโปรโตคอลเพื่อการเรียกใช้
งานคอมพิวเตอร์ข้ามเครื่องและได้กำหนดมาตรฐานของตัวเองออกมามากมาย แต่เมื่อใช้งานจริง การ
สื่อสารระหว่างระบบที่ใช้มาตรฐานที่ต่างกัน ไม่สามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพ เพราะแต่ละ
เทคโนโลยีนั้น ยึดติดกับมาตรฐานของตนเองมากเกินไป ดังนั้นจึงทำให้เกิดเทคโนโลยีเว็บเซอร์วิสขึ้นมาเพื่อ
แก้ปัญหาการไม่สามารถทำงานร่วมกันได้ ของเทคโนโลยีต่าง ๆ

ยุคที่ 3 เว็บเซอร์วิส (Web Service) ปัจจุบันได้มีความพยายามที่จะเปลี่ยนรูปแบบการทำงานของ
แอปพลิเคชันบนอินเทอร์เน็ต ให้สามารถกระจายการทำงานไปยังระบบที่ต่างกันได้ ซึ่งในอดีตยังทำได้ไม่
มีประสิทธิภาพ เนื่องจากแต่ละเทคโนโลยีที่ใช้นั้น ยังยึดติดกับมาตรฐานของตนเอง ทำให้เกิดระบบที่
แตกต่างกัน ไม่สามารถติดต่อกันได้ ดังนั้นจึงได้มีการพัฒนามาตรฐาน XML เว็บเซอร์วิสขึ้น ซึ่งเป็นคอม
โพเนนต์ที่สามารถรันโปรแกรมได้บนอินเทอร์เน็ต และสามารถเข้าถึงได้จากหลาย ๆ ระบบที่แตกต่างกัน
ทำให้การทำงานร่วมกันของแอปพลิเคชันเป็นไปอย่างมีประสิทธิภาพ และง่ายกว่าในอดีต

เทคโนโลยีก่อน Web Services



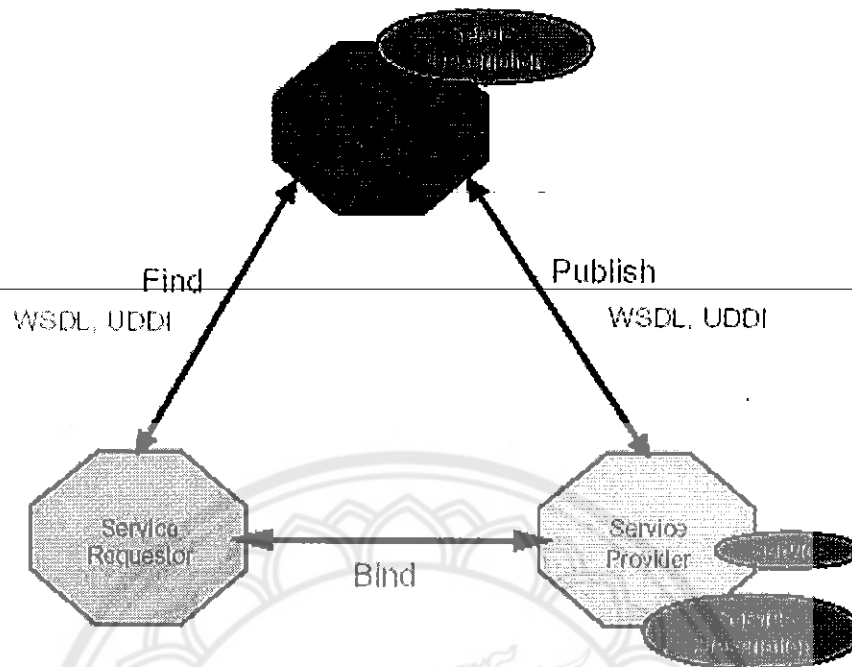
รูปที่ 2.1 Distributed Component Technology

- เว็บเซอร์วิส

เว็บเซอร์วิส คือ แอปพลิเคชันหรือโปรแกรม ซึ่งทำงานอย่างใดอย่างหนึ่งในลักษณะให้บริการ โดยจะถูกเรียกใช้งานจากแอปพลิเคชันอื่น ๆ ผ่านเว็บ การให้บริการของเว็บเซอร์วิสจะมีเอกสารที่บรรยายคุณสมบัติของบริการกำกับไว้ และมีการนำเสนอให้สาธารณชนรับทราบ ผู้ใช้บริการจึงสามารถค้นหาเว็บเซอร์วิสได้โดยไม่ต้องรู้ที่อยู่จริงของแอปพลิเคชันหรือโปรแกรมนั้น

แนวคิดของเว็บเซอร์วิส คือ ระบบอินเทอร์เน็ตเป็นระบบปฏิบัติการ และเว็บไซต์ต่าง ๆ ที่สร้างด้วยเทคโนโลยีเว็บเซอร์วิสเป็นซอฟต์แวร์ที่รันอยู่ภายใต้ระบบปฏิบัติการอินเทอร์เน็ต และการเรียกใช้เว็บไซต์ก็คือ เซอร์วิส (Service) หรือ เว็บเซอร์วิส (Web Service) สิ่งที่ย่อยให้บริการต่าง ๆ ในเว็บจะถูกเรียกว่า โซลูชัน (Solutions) หากเปรียบเทียบในยุคปัจจุบัน กล่าวได้ว่าโฮมเพจที่สร้างขึ้นมา มีโซลูชันหลากหลาย เช่น E-mail, Free Counter, Free GuestBook เป็นต้น

Web Services Model



รูปที่ 2.2 แสดง Web Services Model

ตารางที่ 2.2 อธิบาย Web Services Model

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

2.2.2 เปรียบเทียบระหว่าง Web Application และ Web Service

▪ Web Application คืออะไร

Web application คือ โปรแกรมที่อยู่ใน web server ที่คอยให้บริการสิ่งที่ร้องขอ (request) จากทาง client ผ่านโปรโตคอล HTTP ซึ่งจะแสดงผลที่ร้องขอในรูปแบบของเอกสาร HTML ผ่านทางเบราว์เซอร์ซึ่งก็คือเว็บไซต์ต่างๆ ที่เราใช้บริการอยู่นั่นเอง

Web Application สามารถตอบสนองความคิด Distributed Processing ได้ในระดับหนึ่งซึ่งก็คือ การแบ่งการประมวลผลไว้ที่ฝั่ง Client และฝั่ง Server และมักจะมีการใช้ฐานข้อมูล (database) ควบคู่กับการ ทำ Web Application ไปด้วยตามความต้องการในการทำ E-Business แต่ปัญหาที่ตามมาก็คือ เรื่องของการ จ่ายเงินที่เรียกว่า E-payment หรือ Payment-Gateway ซึ่ง Web Application ที่ทำ E-Commerce ต้องใช้ บริการจากธนาคารออนไลน์ ในการจัดเก็บเงินจากลูกค้า ซึ่งการใช้บริการเก็บเงินจากธนาคารออนไลน์นี้ จำเป็นที่ผู้ค้าจะต้องไปทำการตกลงกับธนาคารและเขียนโปรแกรมให้ตรงตามมาตรฐานที่ธนาคารออนไลน์ กำหนดไว้ซึ่งมีขั้นตอนยุ่งยากซับซ้อน

■ Web Services คืออะไร

Web Services คือ application หรือ โปรแกรมที่ทำงานอย่างใดอย่างหนึ่ง ในลักษณะให้บริการ โดยจะถูกเรียกใช้งานจาก application อื่นๆ ซึ่งการให้บริการจะมีเอกสารที่อธิบายคุณสมบัติของบริการ กำกับไว้ โดยภาษาที่ถูกใช้เป็นการแลกเปลี่ยนคือ XML ทำให้สามารถเรียกใช้ component ใด ๆ ก็ได้ ในระบบ หรือ platform ใด ๆ ก็ได้บนโปรโตคอล HTTP ซึ่งเป็นโปรโตคอลสำหรับ World Wide Web หรือ อินเทอร์เน็ต อันเป็นช่องทางที่ได้รับการยอมรับทั่วโลกในการติดต่อสื่อสารกันระหว่าง application กับ application ในปัจจุบัน

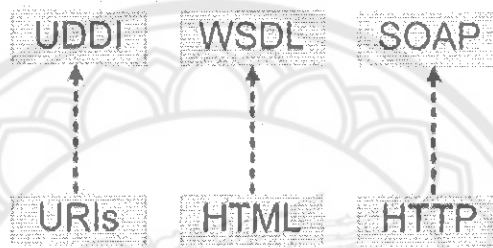
เมื่อเราเข้าใจความหมายและการทำงานของ Web Application และ Web Services แล้ว จะเห็นว่า เครื่องมือทั้งสองต่างใช้โปรโตคอล HTTP หรือ อินเทอร์เน็ต เป็นช่องทางในการสื่อสารเหมือนกัน แต่มี วัตถุประสงค์ต่างกัน โดย Web Application ใช้เพื่อการแลกเปลี่ยนไฟล์ HTML ระหว่างเว็บเซิร์ฟเวอร์ แต่ Web Services เป็นการแลกเปลี่ยน Software Components ระหว่างระบบสารสนเทศผ่านเว็บเซิร์ฟเวอร์

ในเรื่องของความสามารถโดยส่วนใหญ่จะใช้ Web Application ในการติดต่อกับผู้ใช้ผ่านทาง อินเทอร์เน็ตเบราว์เซอร์เพื่อนำเสนอข้อมูลและการทำธุรกรรมต่าง ๆ ส่วน Web Services จะทำหน้าที่ในการ ติดต่อกับเว็บเซิร์ฟเวอร์เพื่อแลกเปลี่ยนข้อมูลและการทำงานหรือ ใช้บริการข้ามระบบกันโดยใช้ Web Application หรือ Application Interface ในการติดต่อกับผู้ใช้ นอกจากนี้ Web Services ยังสามารถทำงานกับ ระบบต่าง ๆ ได้มากกว่า 1 ระบบ ในขณะที่ Web Application ไม่สามารถทำได้โดยตรง ซึ่งสามารถสรุปการ เปรียบเทียบได้ดังตาราง

ตารางที่ 2.3 เปรียบเทียบคุณสมบัติของ web services และ web application

หัวข้อ	Web Service	Web Application
การเชื่อมต่อ	Program-Program	Human-Program
ภาษาที่ใช้	XML	HTML
รายชื่อการให้บริการ	ค้นหาผ่านUDDI	ค้นหาผ่าน Search engine
ขอบเขตการใช้งาน	Business-to-Business(B2B)	Business-to-Customer(B2C)
โปรโตคอล(Protocol)	SOAP+HTTP	HTTP

ซึ่งแสดงคำศัพท์เปรียบเทียบได้ดังรูปต่อไปนี้



รูปที่ 2.3 แสดงคำศัพท์เปรียบเทียบระหว่างเว็บธรรมดากับเว็บเซอร์วิส

ในเรื่องของความปลอดภัย เนื่องจาก Web Service ทำงานอยู่บน Internet ซึ่งปัจจุบันมีเทคโนโลยีรักษาความปลอดภัยมากมายรองรับอยู่แล้ว และ Web Services ก็สามารถข้ามผ่านระบบรักษาความปลอดภัย (Firewall) เหล่านั้นได้ เนื่องจาก SOAP ถูกส่งโดยผ่านด้วยโปรโตคอล HTTP

ส่วนเครื่องมือที่ใช้ในการพัฒนา Web Services ก็มีอยู่มากมาย เนื่องจาก Web Services เป็นมาตรฐานที่พัฒนาได้ง่าย เพราะอยู่ในแบบของข้อมูลตัวอักษรหรือ ภาษา XML ทำให้มีชุดเครื่องมือที่สนับสนุนการพัฒนา Web Services มากมาย ซึ่งสามารถแบ่งออกเป็น 4 กลุ่มใหญ่ได้ดังนี้

1. ชุดเครื่องมือที่สนับสนุน โดย Microsoft ตาม Platform ของ Microsoft .NET Framework
2. ชุดเครื่องมือที่สนับสนุน โดย Sun Microsystems ตาม Platform ของ Sun ONE (Sun Open

Net-Environment)

3. ชุดเครื่องมือที่สนับสนุน โดย IBM เช่น Web Services Toolkit
4. ชุดเครื่องมืออื่นๆที่สนับสนุน SOAP, XML ทั้งที่เป็น Commercial Product และ Open

Source

ดังนั้น องค์กรธุรกิจต่างๆ จึงมั่นใจได้ว่าระบบขององค์กรจะสามารถดำเนินธุรกิจโดยใช้ Web

Services เป็นเครื่องมือได้อย่างแน่นอน และที่สำคัญคือ การนำเครื่องมือเหล่านี้ไปใช้แต่ละองค์กรไม่จำเป็นต้องเปลี่ยนระบบการทำงานที่มีอยู่เดิม เพียงนำเครื่องมือที่เหมาะสมกับ Platform ของระบบที่ใช้เข้ามาเสริมเข้าไปในระบบเดิม โดยที่ระบบเดิมสามารถประสานการทำงานกับเครื่องมืออื่นๆ เท่านั้นองค์กรก็สามารถดำเนินธุรกิจโดยใช้ Web Services เป็นเครื่องมือทันที

ประโยชน์ของการทำงานร่วมกันระหว่าง Web Application กับ Web service

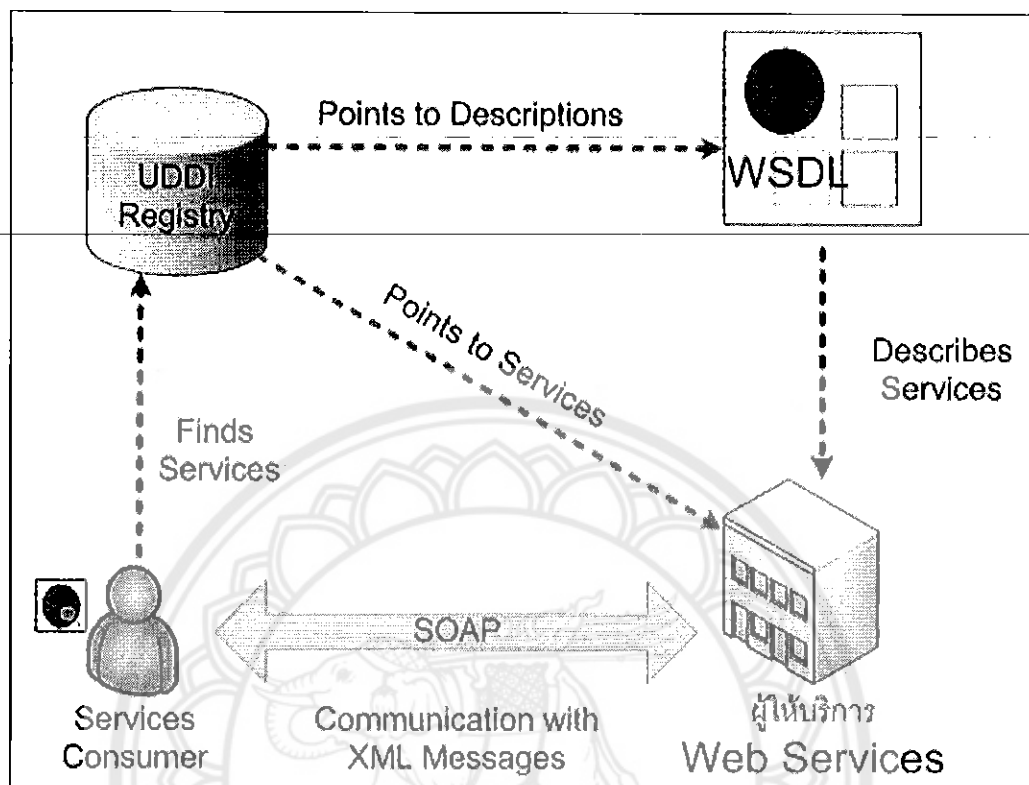
Web Service ช่วยให้การเข้าถึงข้อมูลสารสนเทศจากแอปพลิเคชันที่ต่างกันเป็นไปได้โดยง่าย โดยแอปพลิเคชันนั้น ๆ สามารถเขียนด้วย Java และรันอยู่บน Sun Solaris Application Server หรืออาจจะเขียนด้วย C++ และรันอยู่บน Windows NT หรืออาจจะเขียนด้วย Perl และรันอยู่บนเครื่อง Linux ซึ่งมาตรฐานของ Web Service ทำให้อินเทอร์เน็ตของแอปพลิเคชันเหล่านี้ ถูกอธิบายโดย WSDL และทำให้อยู่ในมาตรฐานของ UDDI หลังจากนั้น จึงสามารถติดต่อสื่อสารถึงกัน โดย XML ผ่าน SOAP อินเทอร์เน็ต

Web Service สามารถถูกเรียกใช้ภายในองค์กรเองหรือจากภายนอกองค์กร โดยผ่าน fire wall ดังนั้นจึงมีองค์กรใหญ่ ๆ มากมาย กำลังพัฒนาระบบที่มีอยู่ของตน ให้เข้ากับ Web Service ซึ่งนับเป็นการลงทุนที่คุ้มค่า เนื่องจาก Web Service สามารถเพิ่มศักยภาพในการทำงานขององค์กร อีกทั้งลดค่าใช้จ่ายในการจัดการทรัพยากรขององค์กรได้อีกทางหนึ่ง

นอกจากนั้น Web Service ยังสามารถใช้ร่วมกับ Web Application โดยส่งผ่านข้อมูลทางอินเทอร์เน็ต ได้อีกด้วยซึ่งนับเป็นวิธีที่มีประสิทธิภาพในการติดต่อสื่อสารกับลูกค้าหรือหุ้นส่วน ถึงแม้จะต้องคำนึงถึงระบบรักษาความปลอดภัย และการจัดการรายการของข้อมูลอยู่ก็ตาม แต่ Web Service ได้ใช้มาตรฐานทั่วไปของ internet เรื่องดังกล่าวจึงนับเป็นเรื่องธรรมดาของการสื่อสารผ่านระบบอิเล็กทรอนิกส์

2.2.3 มาตรฐานที่เกี่ยวข้องกับเว็บเซอร์วิส

Web Service = SOAP+WSDL+UDDI



รูปที่ 2.4 แสดงส่วนประกอบของเทคโนโลยีเว็บเซอร์วิส

รายละเอียดของส่วนประกอบเว็บเซอร์วิสมีดังนี้

- XML (Extensible Markup Language)

Markup Language คือ ภาษาที่ใช้อธิบายความหมายของเอกสารหรือข้อมูลที่อยู่ในรูปของแท็ก Markup Language มีข้อดีคือ สามารถใช้ได้กับทุกแพลตฟอร์ม วัฒนาการของ Markup Language เริ่มต้นจาก GML (Generalized Markup Language) ซึ่งมีแนวคิดที่ ผู้ใช้สามารถติดกลุ่มของแท็กขึ้นมาเองเพื่อใช้อธิบายเอกสาร ต่อมามีการพัฒนา GML ให้มีความเป็นมาตรฐานขึ้นจนกลายเป็น SGML (Standard Generalized Markup Language) ซึ่งมี DTD (Document Type Definition) เป็นตัวนิยามโครงสร้างของเอกสาร ต่อมาในปี ค.ศ. 1991 Tim Berners-Lee แห่งสถาบันวิจัย CERN ประเทศสวิตเซอร์แลนด์ ได้นำพื้นฐานของภาษา SGML ผสมเข้ากับความสามารถของ Hypertext และใช้ URL เพื่อการเชื่อมโยงระหว่างคอมพิวเตอร์หลาย ๆ เครื่อง จนกลายมาเป็น HTML (Hypertext Markup Language) โดยมีโครงสร้างกลุ่มของแท็กหรืออิลิเมนต์ขึ้น เพื่อใช้ในการแสดงผลบนเว็บผ่านบราวเซอร์ ซึ่งเป็นมาตรฐานที่มีประสิทธิภาพในการใช้ออกแบบเว็บเพจ

XML เป็นภาษามาตรฐานที่ทุกระบบสนับสนุน ทำให้ข้อมูลที่มีโครงสร้างของภาษา XML จะถูกนำไปประมวลผลต่ออย่างอัตโนมัติได้อย่างง่ายดาย ภาษา XML จึงถูกนำมาใช้เป็นภาษามาตรฐานในการแลกเปลี่ยนข้อมูลของ Web Services

- โครงสร้างภาษา XML (Extensible Markup Language)

ภาษา XML จะมีโครงสร้างประกอบไปด้วยแท็ก (Tag) ต่าง ๆ เช่นเดียวกับภาษา HTML กล่าวคือประกอบไปด้วยแท็กเปิดและแท็กปิด เพียงแต่ว่าแท็ก XML จะมีชื่อแปลก ๆ ที่ไม่เคยเห็นและคุ้นเคย ตัวอย่างโค้ด XML

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE greeting [<!ELEMENT greeting (#PCDATA)>]>
<greeting>Hello, world!</greeting>
```

ปัญหาอย่างหนึ่งของภาษา HTML ที่เป็นข้อจำกัดที่สำคัญคือ จะสามารถใช้แท็กต่างๆ ได้ก็ต่อเมื่อองค์กรร่วมวางมาตรฐาน W3C (World Wide Web Consortium) ประกาศรับรอง และต้องให้เบราว์เซอร์อย่าง Internet Explorer หรือ Netscape สนับสนุนแท็กดังกล่าวก่อน

สำหรับภาษา XML แท็กที่เราสร้างขึ้นมาก็เป็นมาตรฐานในตัวเอง และเป็นมาตรฐานของผู้สร้างแล้วภาษาอื่นก็ต้องยอมรับด้วย พร้อมทั้งจะเรียกใช้แท็กนี้ได้ทันที ไม่ว่าภาษาใดก็ตาม หน้าที่ของมันจะระบุขอบเขตของข้อมูลว่า ข้อมูลดังกล่าวคืออะไร สามารถอธิบายได้ด้วยตัวเอง ซึ่งเรียกว่าการกระทำ Markup Data ส่วนเรื่องการแสดงผลจะใช้ภาษาอื่นทำหน้าที่แทน

- กฎกติกาเบื้องต้นสำหรับการสร้างข้อมูล XML

การสร้างเอกสาร XML ขึ้นมาจะมีระเบียบข้อบังคับขั้นต้นบางอย่างเพื่อให้มาตรฐานเป็นที่ยอมรับของภาษาอื่นๆ โดยใช้ตัวอย่างต่อไปนี้

```
<?xml version="1.0" encoding="windows-874" ?>
<persondata>
  <id>001</id>
  <first-name>น.ส.ชลธิชา</first-name>
```

```

<last-name>ศรีดา</last-name>
<address>กรุงเทพ</address>
<e-mail>chon@chaiyo.com</e-mail>
</persondata>

```

เริ่มต้นเป็นการประกาศส่วนที่เรียก Prolog หรือเรียกอีกอย่างหนึ่งว่า XML Declaration หมายถึง ส่วนหัวของเอกสาร XML ที่ต้องการใช้งาน ในกรณีนี้ต้องการใช้งาน XML เวอร์ชัน 1.0 ซึ่งเป็นเวอร์ชัน ปัจจุบันและต้องการให้มีการเข้ารหัสภาษาไทยซึ่งตรงกับ windows-847

ต่อมาจะเข้าสู่พื้นที่แสดงแหล่งข้อมูลของเอกสาร XML ที่เรียกว่า Document Elements เอกสาร ประกอบด้วยแท็กเปิดและแท็กปิด แท็กปิดจะใช้เครื่องหมาย / นำหน้าชื่อแท็กเสมอ ซึ่งแท็กต่างๆ ที่ปรากฏ อยู่ในแหล่งข้อมูล XML คือ แท็กที่เราสร้างขึ้นเอง เพื่อระบุว่าข้อมูลที่อยู่ระหว่างแท็กดังกล่าวคือข้อมูล อะไร

แต่เนื่องจากแท็กต่าง ๆ ที่สร้างขึ้นต้องประกอบด้วยแท็กราก (Root Tag) เพื่อทำหน้าที่ปัจจุบันแท็ก อื่นๆ ที่ซ่อนอยู่ในแท็กราก ซึ่งเรียกว่าแท็กลูก (Childs Tag) ซ้อนกันเป็นวงลำดับชั้น โดยภาษา XML จะมอง โครงสร้างดังกล่าวเป็นเหมือนกับ โครงสร้างต้นไม้ (Tree View) ดังตัวอย่าง

```

<?xml version="1.0" encoding="windows-874"?>
<persondata> -----แท็กรากเปิด
<id>001</id> -----แท็กลูก
<first-name>น.ส.ชลธิชา</first-name>
<last-name>ศรีดา</last-name> -----แท็กลูก
<address>กรุงเทพ</address> -----แท็กลูก
<e-mail>chon@chaiyo.com</e-mail> -----แท็กลูก
</persondata> -----แท็กรากปิด

```

- การกำหนดความถูกต้องของ XML ด้วย DTD

DTD มาจากคำว่า Document Type Definition เป็นภาษาที่กำหนดกฎกติกาให้กับอีลิเมนต์ (Element), แอตทริบิวต์ (Attribute) และเอนทิตี (Entity) ที่มีในแหล่งข้อมูล XML ทำได้ 2 ลักษณะดังนี้

แบบฝังส่วนของภาษา DTD ไว้ในแหล่งข้อมูล XML

```
<?xml version="1.0" encoding="windows-874" ?>
<!DOCTYPE book >
<book>
  <bookdata>
    <isbn>974-8237-73-7</isbn>
    <bookname>เริ่มต้นอย่างมืออาชีพด้วย Delphi 5.0ฉบับสมบูรณ์</bookname>
    <author>สัจจะ จรัสรุ่งรวิวรร จรักพงศ์ สุขประเสริฐ</author>
    <publisher>Infopress</publisher>
    <price>395</price>
    <homepage>http://www.infopress2000.com/</homepage>
  </bookdata>
  <bookdata>
    <isbn>974-87679-5-7</isbn>
    <bookname>Database Programming ด้วย Visual Basicฉบับมืออาชีพ</bookname>
    <author>ศุภชัย สมพานิช บรรณาธิการ สัจจะ จรัสรุ่งรวิวรร</author>
    <publisher>Infopress</publisher>
    <price>345</price>
    <homepage>http://www.vbth.cjb.net/</homepage>
  </bookdata>
  <bookdata>
    <isbn>974-87161-3-9</isbn>
    <bookname>Internet Programming ด้วย VB 6.0 และ ASP</bookname>
    <author>สัจจะ จรัสรุ่งรวิวรร</author>
```

```

<publisher>Infopress</publisher>

<price>269</price>

<homepage>http://www.infopress2000.com/</homepage>

</bookdata>

</book>

```

อธิบายการทำงานของโค้ด

1. ตัวอย่างข้างต้นเป็นพื้นฐานการใช้งานภาษา DTD แบบฝังแหล่งข้อมูล XML เพื่อที่จะใช้เป็นกฎกติกา ในการตรวจสอบความถูกต้องของข้อมูลที่จะเก็บในแหล่งข้อมูล XML ซึ่งจะเห็นได้ว่า DT D ไม้มีความคล้ายกับภาษา XML แต่อย่างใด

2. เริ่มต้นในส่วน Prolog จะระบุส่วนหัวของข้อมูลตามปกติ
3. จากนั้นจะใช้คำสั่ง DOCTYPE เพื่อระบุชื่อของกฎ กติกาที่ถูกรวบรวมขึ้นมาในที่นี้ตั้งชื่อว่า book
4. เมื่อประกาศโครงสร้างให้กับแหล่งข้อมูล XML แล้ว ต่อไปจะเป็นการกำหนดวิธีการประมวลผลของข้อมูลในแหล่งข้อมูล XML ชุดนี้ให้กับตัวแปลภาษา XML (XML Parser) การประมวลผลของ XML Parser แบ่งเป็น 3 กรณีคือ

กรณีที่ 1 แสดงข้อความหรือข้อมูลที่อยู่ระหว่างอีลิเมนต์ที่สร้างขึ้นมาเรียกว่า PCDATA (Parsed Character Data) เป็นข้อความที่ต้องแปลด้วย XML Parser เพื่อระบุจุดเริ่มต้นและจุดสิ้นสุดของข้อมูล ดังนั้นจึงต้องใช้แท็กเปิดและแท็กปิดเสมอเพื่อ Markup ข้อมูลดังกล่าว เช่น

```
< publisher>Infopress</publisher>
```

กรณีที่ 2 ข้อความหรือข้อมูลที่ไม้ได้ตีความไม่ต้องแสดงผลออกมา เรียกว่า CDATA (Character Data) เป็นการกำหนดค่าให้กับแอดทริบิวต์

กรณีที่ 3 ไม้ใช่ข้อมูล ไม้ใช่ข้อความ แต่เป็นเอนทิตีแบบไม้ต้องตีความ (Unparsed-Entity) ที่เกี่ยวกับไฟล์รูปภาพ เช่น

```
<img src= "logo.gif" >
```

กรณีนี้แหล่งข้อมูล XML_With_DTD_Embedding.xml มีแบบเดียวคือ อีลิเมนต์ที่ต้องตีความ
ข้อความก่อนเสมอ ดังนั้นทั้ง 6 ตัว จึงกำหนดให้เป็นแบบ CDATA โดยใช้คำสั่งอีลิเมนต์ของภาษา DTD

<pre><!ELEMENT book (bookdata)> <!ELEMENT bookdata (isbn,bookname, author, publisher,price,homepage)> <!ELEMENT isbn (#PCDATA)></pre>
<pre><!ELEMENT bookname (#PCDATA)> <!ELEMENT author (#PCDATA)> <!ELEMENT publisher (#PCDATA)> <!ELEMENT price (#PCDATA)> <!ELEMENT homepage (#PCDATA)></pre>

- การกำหนดความถูกต้องด้วยการอ้างอิงชื่อไฟล์ *.dtd

<pre><?xml version="1.0" encoding="windows-874" ?> <!DOCTYPE myfriend SYSTEM "DTD_For_XML.dtd"> <myfriend> <people> <id>001</id> <name>น.ส.ชลธิดา ศรีดา</name> <address>กรุงเทพ</address> <telephone>02914-6758</telephone> <email>chon@hotmail.com</email> <url>http://www.chon.com/</url></pre>
<pre></people> <people> <id>002</id> <name>นายสุภชัย สมพานิช</name> <address>กรุงเทพ</address></pre>


```

<telephone>02524-1234</telephone>
<email>thaivb@chaiyo.com</email>
<url>http://www.vbth.cjb.net/</url>
</people>
<people>
  <id>003</id>

```

```

<name>น.ส.ชิตีมา ยูราวรรณ</name>
<address>นนทบุรี</address>
<telephone>02254-5476</telephone>
<email>aum@thaimail.com</email>
<url>http://www.aum.com/</url>
</people>
</myfriend>

```

อธิบายการทำงานของโค้ด

1. ในแหล่งข้อมูล XML With DTD Include.xml ส่วนของ Prolog ที่เพิ่มเติมขึ้นมาเรียกว่า DCTYPE Declaration เป็นการระบุว่าในแหล่งข้อมูล XML ต่อไปนี้ ให้มีการใช้ภาษา DTD เพื่อทำหน้าที่ตรวจสอบความถูกต้องอีลิเมนต์ แอตทริบิวต์ หรือเอนทิตี โดยใช้กฎ กติกาที่เก็บอยู่ในไฟล์ DTD For XML.dtd (แอตทริบิวต์ SYSTEM "DTD For XML.dtd") ตั้งชื่อกฎ กติกานี้ว่า myfriend (คำสั่ง DOCTYPE myfriend)

```
<!DOCTYPE myfriend SYSTEM "DTD_For_XML.dtd">
```

2. ในส่วนของกฎ กติกาที่อยู่ในไฟล์ DTD_For_XML.dtd ก็คือ การแยกส่วนของภาษา DTD ออกมาเป็นไฟล์ต่างหาก โดยการระบุโครงสร้างของแหล่งข้อมูล XML และวิธีการประมวลผลของข้อมูลอีลิเมนต์แต่ละตัว

```

<?xml version="1.0" encoding="windows-874"?>
<!ELEMENT myfriend (people,id,name,address,telephone,email,url)>
<!ELEMENT people (#PCDATA)>
<!ELEMENT id (#PCDATA)>

```

```

<!ELEMENT name (#PCDATA)>
<!ELEMENT address (#PCDATA)>
<!ELEMENT telephone (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT url (#PCDATA)>

```

- XSD

XSD ย่อมาจาก XML Schema Definition พัฒนาขึ้นในปี 2001 ซึ่งตั้งใจทำออกมาเพื่อใช้แทน DTD แต่ตัว XSD ก็อยู่ในรูปของเอกสาร XML เหมือนกัน หน้าทีของ XSD นั้น จะเป็นภาษาที่ใช้กำหนด ไวยากรณ์ของเอกสาร markup ในรูปของ XML ข้อกำหนดนั้นจึงถูกเรียกว่า schema หรือชุดไฟล์ข้อมูล XSD ซึ่งมันสามารถนิยามถึง ตัวอย่าง การเรียงลำดับของ elements ทั้งอิลิเมนต์ลูกและอิลิเมนต์หลัก XSD มีความสำคัญต่อเอกสาร XML ในการจับยึด schema หลัก เพราะ schema คือส่วนที่จัดการรวมข้อมูลให้เป็นหน่วยเดียวกัน นี่คือนส่วนที่สำคัญที่สุด พิจารณาตัวอย่าง XSD schema อย่างไม่้งาย ดังนี้

```

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <xs:element name="addressBook">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="card">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="name" type="xs:string"/>
              <xs:element name="email" type="xs:string"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

</xs:element>

</xs:schema>

956

ข. 241ก

254c

● **XSL (The Extensible Style Language)**

เป็นมาตรฐานใหม่ที่เกิดขึ้นมาเพื่อรองรับกับข้อมูลที่เป็น xml กล่าวคือ xsl จะเป็นส่วนที่บอกข้อมูลหรือเอกสารของเราที่จัดเก็บเป็นไฟล์ xml นั้นจะมีการแสดงผลเป็นรูปร่างหน้าตาเช่นไร แต่ xsl สามารถทำได้มากกว่าการจัดหน้าตาข้อมูลเท่านั้น คือ ยังใช้ xsl ในการแปลงรูปแบบ (format) ของข้อมูลได้ด้วย เช่น แปลงเอกสาร xml ไปเป็นไฟล์นามสกุลอื่นๆ เช่น .html หรือ .pdf เป็นต้น

- **ทำความเข้าใจกับ Style sheet เบื้องต้น**

ในการจัดทำ website นั้นจะมีนำเทคนิคสไตลชีตมาใช้ในการจัดสไตลหรือรูปแบบสารสนเทศให้อยู่ในรูปแบบเดียวกันตลอดเอกสารส่วนใหญ่ที่ใช้กันมากบนเว็บคือ CSS (Cascading Style Sheet) โดยนำมาใช้ในการสร้างคลาสของสไตลที่ต้องการ หลังจากนั้นจึงนำมาประยุกต์ใช้ตลอดเอกสาร HTML ปัจจุบันได้มีการพัฒนาเป็นสไตลชีตที่มีความสามารถมากขึ้นไปอีก นั่นคือ XSL ซึ่งสามารถทำการสร้างได้จากเครื่องมือที่ใช้สร้างสไตลชีตต่างๆ ไป

- **การใช้ XSL เพื่อนำเสนอเอกสาร XML**

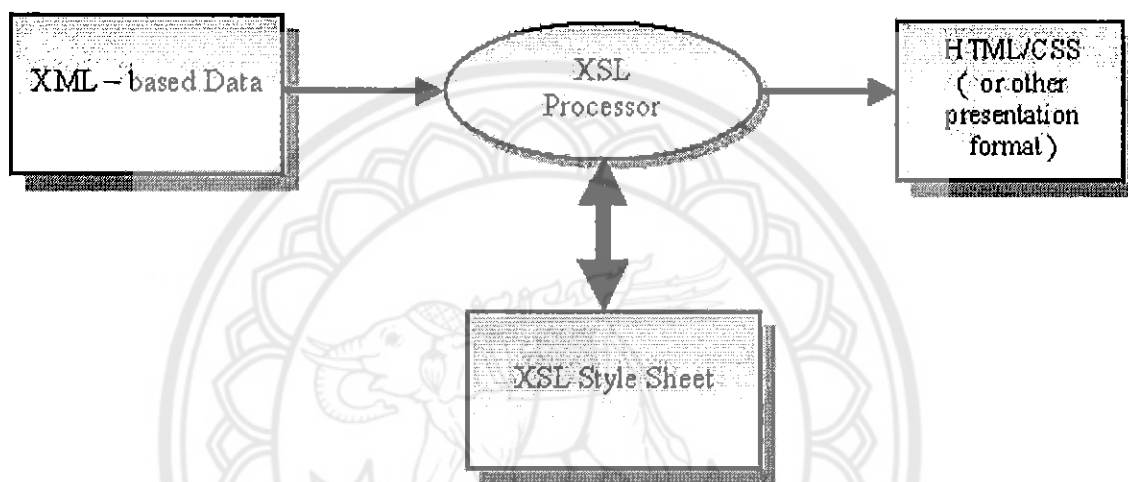
เอกสาร XSL คล้ายกับเอกสาร CSS ในกรณีที่ทั้งสองเอกสารมีการกำหนดสไตลที่ประยุกต์ไปยังอ็อบเจกต์ที่แน่นอน แต่มีสิ่งแตกต่างกันเล็กน้อยที่ CSS จะกำหนดสไตลการพิมพ์เฉพาะ HTML Element เท่านั้น ในทางตรงข้าม สไตลที่กำหนดในเอกสาร XSL จะประยุกต์ไปยังเอกสาร XML อื่นทั้งในการแสดงผล XML โดยใช้ XSL จะมีประสิทธิภาพและยืดหยุ่นมากกว่า CSS แม้ว่า CSS จะสามารถกำหนดการจัดรูปแบบของแต่ละ Element แต่ XSL สามารถที่จะควบคุมข้อมูลที่ส่งออกมาได้อย่างสมบูรณ์กว่า อีกทั้งข้อมูลที่จะนำเสนอ นั้นจะมีอิสระต่อการปรับปรุงหรือเพิ่มข้อมูล XSL ให้สามารถจัดเรียงและกรองข้อมูล XML ได้ง่ายกว่าด้วย

- **หน้าที่ของ XSL**

นอกจากการเป็นสไตลชีตแล้ว XSL ยังเป็นหรือทำได้มากกว่านั้น กล่าวคือ XSL จะประกอบด้วย 2 ส่วน ได้แก่ ส่วนแรกคือส่วนที่เอาไว้แปลงรูปแบบ (transforming) เอกสาร XML และส่วนที่สองเป็นส่วนการจัดรูปแบบเอกสาร(formatting)

ในส่วนแรกคือ "Transforming" เป็นการที่ XSL จะแปลงเอกสาร XML ให้อยู่ในรูปแบบที่เบราว์เซอร์ สามารถเข้าใจได้ ตัวอย่างที่เห็นได้ชัดก็คือการที่ XSL สามารถแปลงเอกสาร XML ให้อยู่ในรูปแบบของ HTML ได้

ส่วนที่สองคือ "Formatting" เป็นการที่ XSL สามารถทำฟิลเตอร์หรือเรียงลำดับข้อมูลได้เช่น อาจจะสามารถเลือกได้ว่าในเอกสารจะแสดงผลข้อมูลส่วนใดบ้างโดยกำหนดเงื่อนไขการแสดงผลเอาไว้หรืออาจจะนำข้อมูลนั้นมาเรียงลำดับค่าน้อยไปหามาก เป็นต้น



รูปที่ 2.5 แสดงหน้าที่การทำงานของ xsl

จากภาพจะมีไฟล์เอกสารอยู่ 2 ส่วน คือ ส่วนของเอกสาร XML กับเอกสาร XSL แล้วทั้ง 2 ส่วนเมื่อมาผ่านกระบวนการ XSL Processor ก็จะเปลี่ยนจากเอกสาร XML เป็น HTML

- พื้นฐานการใช้ XSL Style Sheet

มี 2 ขั้นตอนพื้นฐานสำหรับการใช้ XSL เพื่อแสดงผลเอกสาร XML

1. การสร้างไฟล์ XSL Style Sheet

เนื่องจากว่า XSL เป็น โปรแกรมประยุกต์ของ xml นั่นคือ XSL เป็นเอกสารที่มีรูปแบบถูกต้องตามกฎของ XSL เช่นเดียวกับเอกสาร XML โดย XSL Style Sheet จะประกอบด้วยข้อความที่สามารถสร้างได้ด้วย Text Editor

2. การลิงค์ XSL Style Sheet

ทั้งเอกสาร XML และ XSL Style Sheet จะถูกเก็บเป็นไฟล์ที่แยกจากกัน นั่นคือ ไฟล์เอกสาร XML

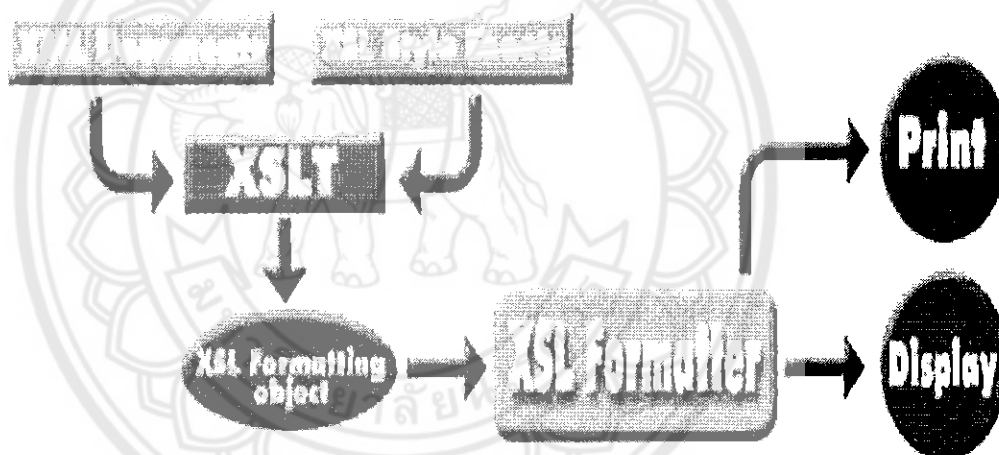
(นามสกุล .xml) และไฟล์ XSL Style Sheet (นามสกุล . XSL) สามารถทำ การรวมเอกสารทั้ง 2 ไว้ในส่วน xml-styleSheet มีรูปแบบดังนี้

```
<?xml – style sheet type = "text/xsl" href = XSLFilePath?>
```

ในที่นี้ XSLFilePath คือ ตำแหน่งที่สัมพันธ์กับตำแหน่งของเอกสาร XML

ถ้าเชื่อม XSL Style Sheet ไปยังเอกสาร XML สามารถเปิดเอกสารนั้นได้โดยตรงใน IE5 ขึ้นไป และบราวเซอร์จะแสดงผลเอกสาร XML โดยใช้คำสั่งการเปลี่ยนรูปใน Style Sheet ซึ่งไม่เหมือนกับ CSS ที่ ถ้าหากเชื่อม XSL Style Sheet มากกว่าหนึ่งไปยังเอกสาร XML โปรแกรมบราวเซอร์ จะใช้เฉพาะตัวแรก เท่านั้นและไม่สนใจตัวที่เหลือ แต่ถ้าหากเชื่อมทั้ง CSS และ XSL Style Sheet ไปยังเอกสาร XML โปรแกรมบราวเซอร์จะใช้เฉพาะ XSL Style Sheet เท่านั้น

XSL - PROCESSING



รูปที่ 2.6 แสดงลักษณะการทำงานของ XSL

จากภาพเมื่อมีเอกสาร XML Document และมีส่วนของ XSL Style Sheet ก็จะสามารถจัดรูปแบบเอกสาร xml ให้แสดงบนบราวเซอร์ได้

หมายเหตุ XSLT : เป็นภาษาสำหรับการแปลงรูปแบบเอกสาร XML ไปเป็นเอกสารประเภทข้อความอื่น ๆ ดังเช่น XHTML สามารถใช้ XSLT เพื่อกำหนดแม่แบบแก่ผลลัพธ์ทำให้ส่งข้อมูล XML ไปแสดงผลได้ อย่างไรก็ตาม XSL Style Sheets ใน Internet Explorer 5 ได้รับการยอมรับจากข้อกำหนดล่าสุดของ W3C อีกทั้ง XSLT นั้นจะมีการแนะนำคุณลักษณะต่าง ๆ มากมายให้ทราบ

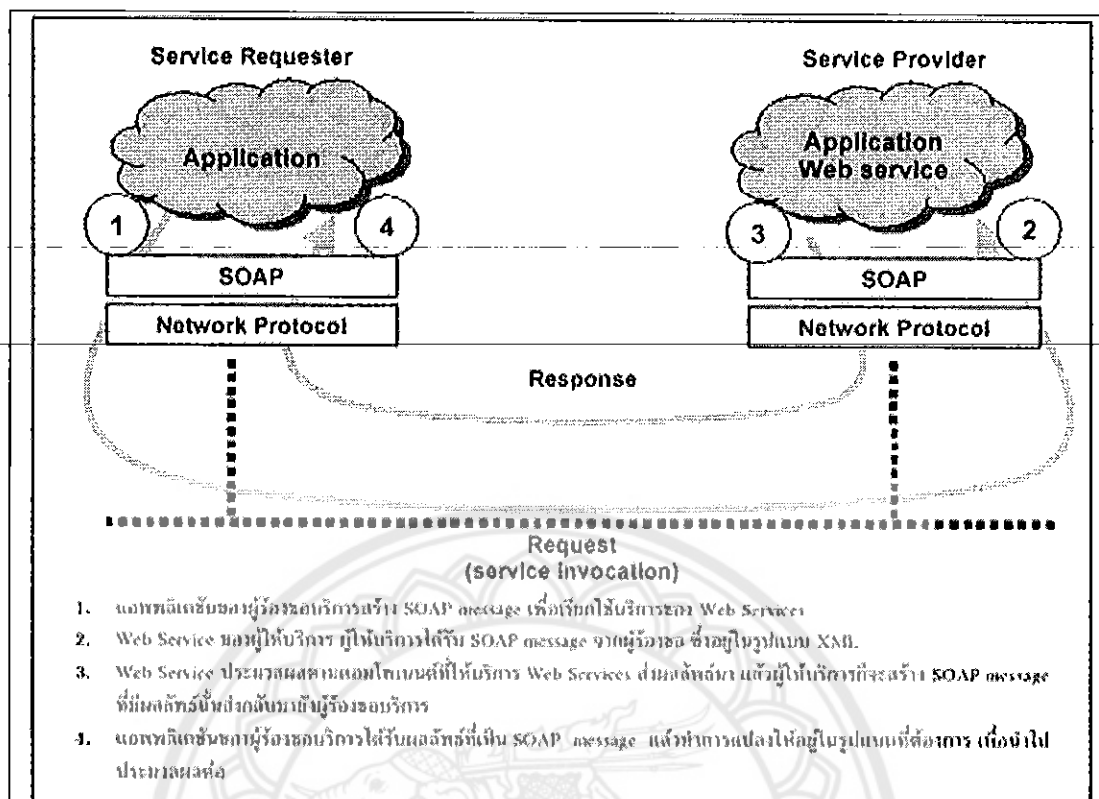
ดังนั้นจึงต้องมีการอัปเดต Style Sheet ไปเป็นโครงร่าง XSLT เพื่อให้ใช้ประโยชน์ลักษณะใหม่ๆ เหล่านี้ได้

- SOAP (Simple Object Access Protocol)

เนื่องจากจุดประสงค์หลักของการใช้งานเว็บเซอร์วิสคือ ต้องการให้แอปพลิเคชันมีการทำงานกับแอปพลิเคชันที่ทำงานอยู่ในเครื่องอื่นผ่านทางเครือข่าย ซึ่งเทคโนโลยีที่มีอยู่ในปัจจุบันมีการสื่อสารระหว่างออบเจกต์ในระยะไกล (Remote Procedure Calls :-RPC) เช่น DCOM, EJB หรือ CORBA ไม่ได้ถูกออกแบบมาใช้สำหรับโปรโตคอล

เทคนิค RPC ของเทคโนโลยีที่กล่าวข้างต้นนั้น มีปัญหาในด้านการนำมาใช้งานในแง่ของความเข้ากันได้ของการเรียกใช้งานข้ามเทคโนโลยี เนื่องจากเป็นเทคโนโลยีเฉพาะของแต่ละค่าย (ยกเว้น CORBA) ทำให้ผู้พัฒนาระบบต้องพัฒนาโปรแกรมที่มีความซับซ้อน และยังมีปัญหาในส่วนของ Firewalls และ Proxy เซิร์ฟเวอร์ เนื่องจากโดยปกติเซิร์ฟเวอร์จะปิดการสื่อสารที่ไม่ใช่โปรโตคอล HTTP เพื่อความปลอดภัยของระบบที่มีการติดต่อสื่อสารกับภายนอก ดังนั้น ทางเลือกของการสื่อสารที่จะนำมาใช้ในการให้บริการเว็บเซอร์วิส คือ ต้องสามารถทำงานอยู่บนโปรโตคอล HTTP และเป็นมาตรฐานเปิดที่ทำให้สามารถติดต่อสื่อสารกับเครื่องคอมพิวเตอร์ที่มีความแตกต่างกันทั้งระบบปฏิบัติการ เทคโนโลยี รวมไปถึงภาษาที่ใช้ในการพัฒนาด้วย

SOAP อ่านว่า โซพ เป็นมาตรฐานของเทคโนโลยี Distributed Objects แบบหนึ่ง โดยทำหน้าที่ส่งข้อมูลผ่านอินเทอร์เน็ต ในรูปแบบของ XML ทำให้เรียกใช้งานโปรแกรมข้ามระบบผ่านทางอินเทอร์เน็ตได้



รูปที่ 2.7 แสดง XML messaging using SOAP

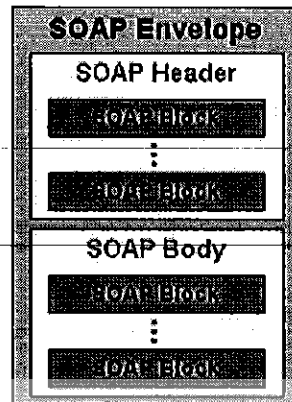
SOAP ประกอบด้วย 3 ส่วน คือ

• SOAP Envelop สำหรับใช้ในการระบุสิ่งที่อยู่ในเอกสาร การจัดการเอกสาร และบอกความจำเป็นในการใช้

• SOAP Encoding Rule สำหรับกำหนดกลไกที่ใช้ในการแลกเปลี่ยนข้อมูล

• SOAP RPC Representation สำหรับนิยามรูปแบบ Remote procedure call and response

โครงสร้างของ SOAP Message



รูปที่ 2.8 โครงสร้างของเอกสาร SOAP

เอกสาร SOAP นั้นมีโครงสร้างในรูปแบบ XML ซึ่งเราสามารถแบ่งเป็นส่วนของเอกสารได้เป็น 3 ส่วนหลัก ดังนี้

1. SOAP envelop เป็นเนื้อหาสาระ (Content) ของเอกสาร XML ที่ใช้แทน Message
 2. SOAP header เป็นส่วนเพิ่มเติมของเอกสาร SOAP
 3. SOAP body เป็นส่วนที่ใช้เก็บข้อมูลสำหรับส่งไปยังผู้รับปลายทาง
- WSDL (Web Services Description Language)

WSDL (Web Services Description Language) คิดค้นโดยบริษัท IBM และ Microsoft เป็นภาษาที่ใช้อธิบายคุณลักษณะการให้บริการของเว็บเซอร์วิส และวิธีการติดต่อขอรับบริการจาก เว็บเซอร์วิส เช่น ชื่อเว็บเซอร์วิส ชื่อเมธอดของ COM Component ที่เปิดให้บริการ พารามิเตอร์ที่ส่งไปยังเมธอด ชนิดข้อมูลของพารามิเตอร์ เป็นต้น โดยรายละเอียดเหล่านี้จะเป็นไปตามไวยากรณ์ของภาษา XML (Extensible Markup Language)

WSDL เป็นภาษามาตรฐานที่ใช้สำหรับอธิบายการใช้งาน โปรแกรมที่เปิดให้บริการ ซึ่งเขียนขึ้นตามแบบมาตรฐาน XML ดังนั้น WSDL จึงเป็นเสมือนคู่มือการใช้งานระบบ หรือแนะนำวิธีการเรียกใช้งาน Web Services

โครงสร้างของ WSDL

เอกสาร WSDL นั้นแบ่งออกเป็น 2 ส่วนใหญ่ ๆ คือ

1.) Abstract Definitions Group

ทำหน้าที่กำหนด message SOAP ที่ไม่ขึ้นกับแพลตฟอร์มหรือรูปแบบภาษา กล่าวคือ ไม่บรรจุ ส่วนประกอบที่มีเฉพาะในเครื่องบางเครื่อง หรือภาษาบางภาษา โดยที่ส่วนประกอบของ abstract นั้น จะ ประกอบไปด้วย type message และ port type

2.) Concrete Descriptions Group

ทำหน้าที่ตรงกันข้ามกับส่วนของ abstract Definitions คือ เป็นข้อมูลที่ขึ้นอยู่กับเว็บไซต์ เครื่องและ ภาษา โดยประกอบด้วย binding และ service

ตัวอย่างเอกสาร WSDL

```
<?xml version="1.0" encoding="UTF-8" ?>
<definitions targetNamespace="http://localhost:8081/axis/tpmenu1/
  ThaiPalaceMenuService.jws"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:serviceNS="http://localhost:8081/axis/tpmenu1/
    ThaiPalaceMenu Service.jws" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <message name="getMenuStylesheetRequest" />
  <message name="getMenuStylesheetResponse">
    <part name="getMenuStylesheetResult" type="xsd:string" />
  </message>
  <message name="getMenuResponse">
    <part name="getMenuResult" type="xsd:string" />
  </message>
  <message name="getMenuDocumentRequest" />
```

<pre> <message name="getMenuSchemaRequest" /> <message name="getMenuSchemaResponse"> <part name="getMenuSchemaResult" type="xsd:string" /> </message> <message name="fileToStringResponse"> <part name="fileToStringResult" type="xsd:string" /> </pre>	
<pre> </message> <message name="getMenuDocumentResponse"> <part name="getMenuDocumentResult" type="xsd:string" /> </message> <message name="getMenuRequest" /> <message name="fileToStringRequest"> <part name="arg0" type="xsd:string" /> </message> <portType name="ThaiPalaceMenuServicePortType"> <operation name="fileToString"> <input message="serviceNS:fileToStringRequest" /> <output message="serviceNS:fileToStringResponse" /> </operation> <operation name="getMenu"> <input message="serviceNS:getMenuRequest" /> <output message="serviceNS:getMenuResponse" /> </operation> </pre>	
<pre> <operation name="getMenuDocument"> <input message="serviceNS:getMenuDocumentRequest" /> <output message="serviceNS:getMenuDocumentResponse" /> </operation> <operation name="getMenuSchema"> <input message="serviceNS:getMenuSchemaRequest" /> </pre>	

```

        <output message="serviceNS:getMenuSchemaResponse" />
    </operation>
    <operation name="getMenuStylesheet">
        <input message="serviceNS:getMenuStylesheetRequest" />
        <output message="serviceNS:getMenuStylesheetResponse" />
    </operation>

```

```

</portType>
<binding name="ThaiPalaceMenuServiceSoapBinding"
    type="serviceNS:ThaiPalaceMenuServicePortType">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="fileToString">
        <soap:operation soapAction="" style="rpc" />
        <input>
            <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                namespace="" use="encoded" />
        </input>
        <output>
            <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                namespace="" use="encoded" />
        </output>
    </operation>
    <operation name="getMenu">
        <soap:operation soapAction="" style="rpc" />
        <input>

```

```

            <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                namespace="" use="encoded" />
        </input>
        <output>

```

```

            <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                namespace="" use="encoded" />
        </output>
    </operation>

```

```

            <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                namespace="" use="encoded" />
        </input>
        <output>

```

```

        namespace="" use="encoded" />
    </output>
</operation>
<operation name="getMenuDocument">
    <soap:operation soapAction="" style="rpc" />
    <input>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            namespace="" use="encoded" />
    </input>
</output>
<operation name="getMenuSchema">
    <soap:operation soapAction="" style="rpc" />
    <input>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            namespace="" use="encoded" />
    </input>
</output>
</output>
</operation>
<operation name="getMenuStylesheet">
    <soap:operation soapAction="" style="rpc" />
    <input>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"

```

<pre> namespace="" use="encoded" /> </input> <output> <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="" use="encoded" /> </output> </pre>
<pre> </operation> </binding> <service name="ThaiPalaceMenuService"> <port binding="serviceNS:ThaiPalaceMenuServiceSoapBinding" name="ThaiPalaceMenuServicePort"> <soap:address location=http://localhost:8081/axis/tpmenu1/ThaiPalaceMenuService.jws/> </port> </service> </definitions> </pre>

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

หลังจากสร้างเว็บเซอร์วิสขึ้นมาแล้ว หากเราไม่ได้ประกาศให้ผู้อื่นรับรู้หรือทราบเกี่ยวกับรายละเอียดและการให้บริการของเว็บเซอร์วิสที่เราสร้างขึ้นมาก็คงไม่มีความหมาย หรืออาจจะทราบกันเฉพาะในกลุ่มแคบ ๆ เท่านั้น ซึ่ง UDDI นี้จะเป็นตัวแทนที่คอยประกาศให้ผู้อื่นได้ทราบถึงเว็บเซอร์วิสของเรา โดยจะต้องไปลงทะเบียนกับผู้ให้บริการ UDDI เหมือนกับระบบ Search Engine

UDDI เป็นที่เก็บรวบรวม Web Services ต่างๆ ในอินเทอร์เน็ต ไว้ในแหล่งเดียวกันเพื่อให้ผู้ใช้บริการสามารถค้นหาได้ง่าย ๆ หากเปรียบเทียบง่าย ๆ ให้มองเสมือนสมุดหน้าเหลืองที่เราใช้ในการเปิดดูเบอร์โทรศัพท์

- ผู้เริ่มก่อตั้ง UDDI ในช่วงแรกคือ IBM, Microsoft และ Ariba ซึ่งเป็นบริษัทที่ทำธุรกิจ B2B

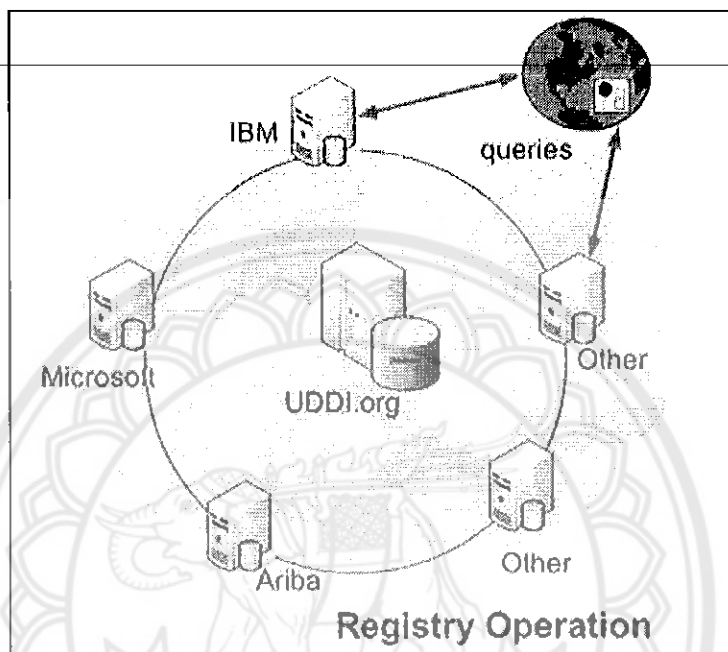
ปัจจุบันมีบริษัทที่มีส่วนร่วมในการกำหนดมาตรฐานของ UDDI มีมากกว่า 70 บริษัท

- UDDI ถูกสร้างขึ้นมาเพื่อเป็นมาตรฐานในการค้นหาบริการของเว็บเซอร์วิส สำหรับคู่ค้าทาง

ธุรกิจ (Business Partner)

- UDDI Business Registry เป็นฐานข้อมูลเว็บเซอร์วิสของบริษัทคู่ค้าทางธุรกิจ
- ในปัจจุบันบางบริษัทก็ตั้งตัวเองเป็นตัวแทนผู้ให้บริการ (Service brokers)
- สามารถศึกษาข้อมูลเกี่ยวกับ UDDI ได้ที่ <http://uddi.org/> หรือดาวน์โหลด UDDI อ่าน

เพิ่มเติมได้ที่ http://www.uddi.org/pubs/UDDI_Overview_Presentation.ppt



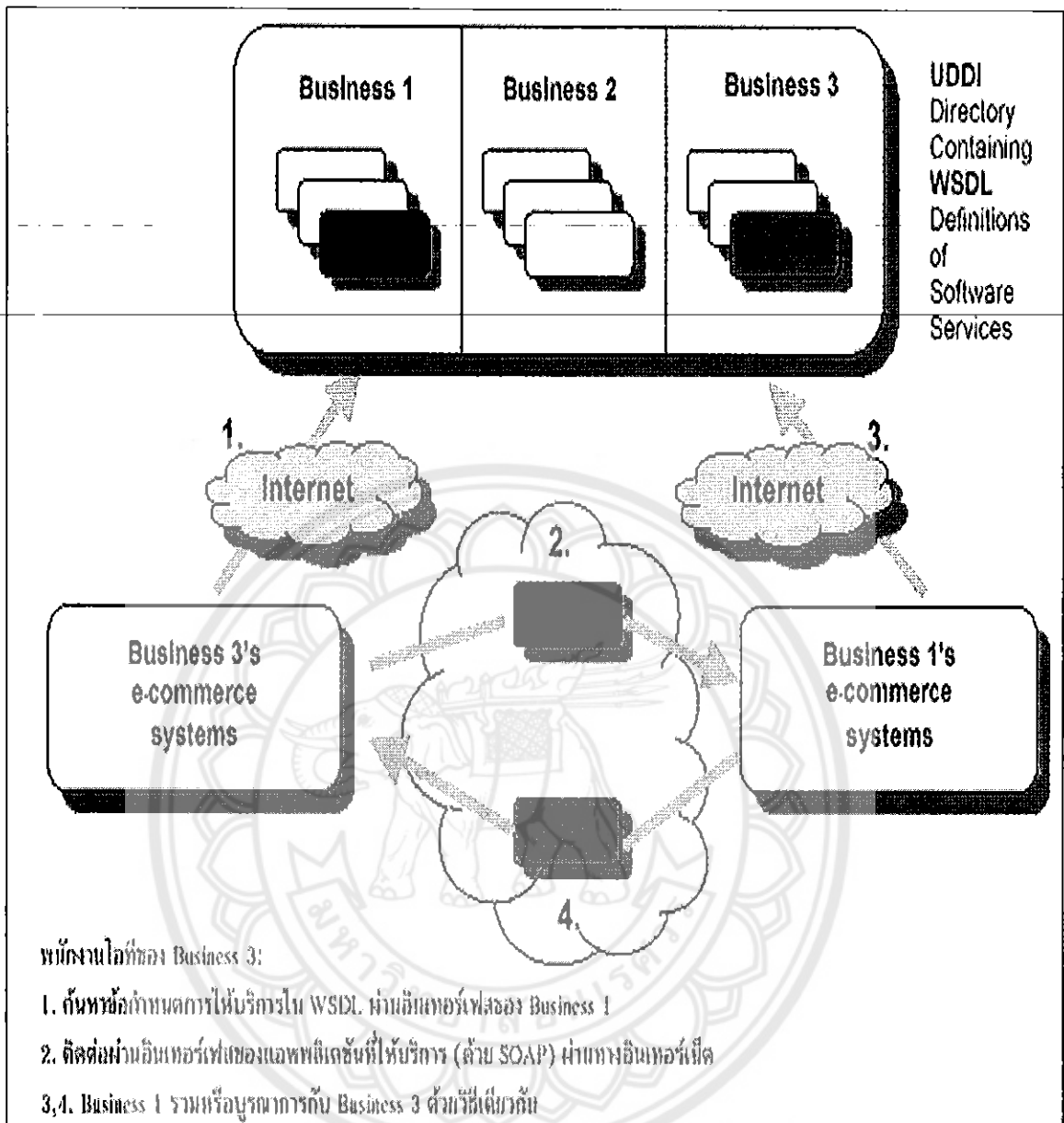
รูปที่ 2.9 แสดงฐานข้อมูลเว็บเซอร์วิสของบริษัทคู่ค้าทางธุรกิจ (Registry Operation)

ความสัมพันธ์ระหว่าง SOAP WSDL และ UDDI

- SOAP เป็นโปรโตคอลสื่อสารที่อาศัยไวยากรณ์ของภาษา XML และทำงานกับโปรโตคอลอื่นๆ ได้หลายชนิด เนื่องจากใช้ไวยากรณ์ XML จึงทำงานได้ทุกแพลตฟอร์มจึงสามารถเรียกใช้คอมโพเนนต์ต่างๆ ได้ในทุกแพลตฟอร์ม

- WSDL เป็นเอกสารที่ใช้อธิบายคุณลักษณะของ เว็บเซอร์วิส ดังที่ได้กล่าวมาแล้ว โดยใช้ไวยากรณ์ของ XML เช่นกัน

- UDDI เปรียบเสมือนฐานข้อมูลที่เก็บรายละเอียดของเว็บเซอร์วิสไว้ เพื่อให้ผู้ใช้บริการค้นหาบริการ ในกรณีของผู้ให้บริการต้องนำข้อมูลเกี่ยวกับเว็บเซอร์วิสของตนไปเก็บไว้ใน UDDI



รูปที่ 2.10 แสดง Web Services with SOAP, UDDI และ WSDL

2.3 Apache Axis Tools

Axis เป็น Open Source Implementation ที่สำคัญของ SOAP เวอร์ชัน 3.0 พัฒนาขึ้นโดยกลุ่ม W3C ใช้สำหรับสร้างเว็บเซอร์วิส เปิดตัวเมื่อเดือนตุลาคม พ.ศ. 2545 จุดเด่นของ Axis มีดังนี้

- ความเร็ว : Axis ใช้ตัวแปร SAX (event-based) เป็นหลัก จึงมีความเร็วกว่า Apache SOAP เวอร์ชันก่อน ๆ

- ความยืดหยุ่น : โครงสร้างของ Axis ทำให้นักพัฒนาเป็นอิสระในการขยายลงใน engine สำหรับกระบวนการ custom header, การจัดการระบบ, และอื่น ๆ ในจินตนาการคุณ
- เสถียรภาพ : Axis กำหนดเขตของการติดต่อภายนอกโดยการเปลี่ยนความสัมพันธ์อย่างซ้ำ ๆ เปรียบเทียบกับ the rest of Axis.
- การพัฒนา Component-oriented : คุณสามารถกำหนดให้ network กลับมาใช้ใหม่ได้อย่างง่าย ๆ โดยที่ผู้พัฒนาสามารถสร้าง common patterns ของโปรแกรม หรือ distribute to partners ได้
- การสนับสนุน WSDL : Axis สนับสนุน WSDL version 1.1, ซึ่งสามารถสร้างช่องทางการขอใช้บริการได้อย่างง่าย ๆ และเป็นไปโดยอัตโนมัติ



บทที่ 3

การใช้ Apache Axis พัฒนาเว็บเซอร์วิส

3.1 Building the Web Services

3.1.1 ThaiPalaceMenuservice

มีด้วยกันไฟล์ 2 ไฟล์ดังนี้

- `menu.xsd` เป็นเอกสารเมนูในรูปแบบ XML-based ที่ SiamDelivery กำหนดให้ทุกภัตตาคารมีรูปแบบของการนำเสนอรายการอาหารที่มีแบบฟอร์มเดียวกัน ดังนี้

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      A sample XML Schema for a Restaurant Menu
    </xs:documentation>
  </xs:annotation>
  <xs:element name="menu" type="menuType"/>
  <xs:complexType name="menuType">
    <xs:sequence>
      <xs:element name="name" type="nameType"/>
      <xs:element name="fax" type="faxType"/>
      <xs:element name="fax-order" type="fax-orderType"/>
      <xs:element name="category" type="categoryType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="categoryType">
```

```

    <xs:sequence>
      <xs:element name="name" type="nameType"/>
      <xs:element name="item" type="itemType" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="itemType">

```

```

    <xs:sequence>
      <xs:element name="name" type="nameType"/>
      <xs:element name="description" type="descriptionType"/>
      <xs:element name="price" type="priceType"/>
    </xs:sequence>
    <xs:attribute name="veggie" type="xs:boolean"/>
    <xs:attribute name="spicy" type="xs:boolean"/>
    <xs:attribute name="number" type="numberType"/>
  </xs:complexType>
  <xs:simpleType name="nameType">
    <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
      <xs:maxLength value="40"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="faxType">
    <xs:restriction base="xs:string">
      <xs:minLength value="8"/>
      <xs:maxLength value="14"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="fax-orderType">
    <xs:restriction base="xs:string">

```

```
<xs:minLength value="0"/>
  <xs:maxLength value="240"/>
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="numberType">
  <xs:restriction base="xs:decimal">
    <xs:totalDigits value="3"/>
    <xs:fractionDigits value="0"/>
    <xs:minExclusive value="0"/>
    <xs:maxInclusive value="125"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="descriptionType">
  <xs:restriction base="xs:string">
    <xs:minLength value="4"/>
    <xs:maxLength value="120"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="priceType">
  <xs:restriction base="xs:decimal">
    <xs:fractionDigits value="2"/>
    <xs:minInclusive value="0"/>
    <xs:maxExclusive value="10"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>
```

- menu.xml เป็นรายการอาหารที่เขียนอยู่ในรูปแบบภาษา XML ตามฟอร์มของ menu.xsd ดังนี้

```

<?xml version="1.0" encoding="UTF-8"?>
<menu>
  <name>Thai Palace</name>
  <fax>(800) 867-5309</fax>
  <fax-order>Fax this menu back with your selections circled. If you want more than one item,
  indicate the number of items in the number column. Your order will be ready 30 minutes from when we
  receive the fax.</fax-order>
  <category>
    <name>appetizers</name>
    <item number="1" veggie="true" spicy="false">
      <name>Spring Roll</name>
      <description>Thai style egg roll with glass noodles and vegetables served in a plum
      sauce
      </description>
      <price>1.75</price>
    </item>
    <item number="2" veggie="true" spicy="false">
      <name>Fried Tofu</name>
      <description>Deep-fried tofu served with a plum sauce</description>
      <price>2.25</price>
    </item>
  </category>
  <category>
    <name>soups</name>
    <item number="3" veggie="true" spicy="true">
      <name>Tom Yum</name>

```

```

    <description>Mushroom in hot and sour soup with a touch of lemon grass
  </description>
  <price>4.75</price>
</item>

```

```

<item number="4" veggie="false" spicy="false">
  <name>Wonton soup</name>

```

```

    <description>Thai-style chicken dumpling soup, topped with chicken
barbecued pork, and green onions</description>

```

```

  <price>5.25</price>

```

```

</item>

```

```

</category>

```

```

<category>

```

```

  <name>salads</name>

```

```

  <item number="5" veggie="false" spicy="true">

```

```

    <name>Larb</name>

```

```

    <description>

```

```

      Ground eat mied with lime juice, onion, chili, and mint leaves served
with cabbage

```

```

    </description>

```

```

    <price>4.75</price>

```

```

  </item>

```

```

<item number="6" veggie="false" spicy="true">

```

```

  <name>Yum Yai</name>

```

```

  <description>shrimp, chicke, cucumber, and onion served on abed of lettuce
with sweet and sour dressing</description>

```

```

  <price>3.25</price>

```

```

</item>

```

	<pre> </category> <category> <name>curry</name> <item number="7" veggie="false" spicy="true"> <name>Prig King</name> <description>Green bean stir fry with chili paste</description> </pre>
	<pre> <price>5.75</price> </item> <item number="8" veggie="true" spicy="true"> <name>Pa-Nang</name> <description>Ground peanut in curry with Thai spices and coconut milk</description> <price>5.25</price> </item> </category> </pre>
	<pre> <category> <name>rice</name> <item number="9" veggie="true" spicy="false"> <name>House Fried Rice</name> <description>Fried rice with onion, tomat, peas, carrot, and egg</description> <price>4.75</price> </item> <item number="10" veggie="false" spicy="true"> </pre>
	<pre> <name>spicy Fried Rice</name> <description>Fried rice with tomato, onion, bell pepper, and sweet basil</description> <price>5.25</price> </item> </category> </pre>

```

    <category>
      <name>noodles</name>
      <item number="11" veggie="true" spicy="false">
        <description>Pan-fried rice nodles with egg, onion, and bean sprouts topped
with crushed peanuts</description>
        <price>5.75</price>
      </item>
      <item number="12" veggie="true" spicy="true">
        <name>Phad-See-Yew</name>
        <description>Pan-fried flat noodles with egg and broccoli in brown
sauce</description>
      </item>
    </category>
  </menu>

```

ทั้ง menu.xsd และ menu.xml เก็บไว้ที่ c:\project\menu

3.1.2 Nutty Palace

มีด้วยกันไฟล์ 2 ไฟล์ดังนี้

- menu.xsd เป็นเอกสารเมนูในรูปแบบ XML-based ที่ SiamDelivery กำหนดให้ทุกภัตตาคารมีรูปแบบของการนำเสนอรายการอาหารที่มีแบบฟอร์มเดียวกัน ดังนี้

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      A sample XML Schema for a Restaurant Menu
    </xs:documentation>
  </xs:annotation>
  <xs:element name="menu" type="menuType"/>

```

```

<xs:complexType name="menuType">
  <xs:sequence>
    <xs:element name="name" type="nameType"/>
    <xs:element name="fax" type="faxType"/>
    <xs:element name="fax-order" type="fax-orderType"/>
    <xs:element name="category" type="categoryType"

```

```
maxOccurs="unbounded"/>
```

```
  </xs:sequence>
```

```
</xs:complexType>
```

```
<xs:complexType name="categoryType">
```

```
  <xs:sequence>
```

```
    <xs:element name="name" type="nameType"/>
```

```
    <xs:element name="item" type="itemType" maxOccurs="unbounded"/>
```

```
  </xs:sequence>
```

```
</xs:complexType>
```

```
<xs:complexType name="itemType">
```

```
  <xs:sequence>
```

```
    <xs:element name="name" type="nameType"/>
```

```
    <xs:element name="description" type="descriptionType"/>
```

```
    <xs:element name="price" type="priceType"/>
```

```
  </xs:sequence>
```

```
  <xs:attribute name="veggie" type="xs:boolean"/>
```

```
  <xs:attribute name="spicy" type="xs:boolean"/>
```

```
  <xs:attribute name="number" type="numberType"/>
```

```
</xs:complexType>
```

```
<xs:simpleType name="nameType">
```

```
  <xs:restriction base="xs:string">
```

```
    <xs:minLength value="1"/>
```

```
    <xs:maxLength value="40"/>
```



```
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="faxType">
        <xs:restriction base="xs:string">
            <xs:minLength value="8"/>
            <xs:maxLength value="14"/>
        </xs:restriction>
    </xs:simpleType>
```

```
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="fax-orderType">
        <xs:restriction base="xs:string">
            <xs:minLength value="0"/>
            <xs:maxLength value="240"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="numberType">
        <xs:restriction base="xs:decimal">
            <xs:totalDigits value="3"/>
            <xs:fractionDigits value="0"/>
            <xs:minExclusive value="0"/>
            <xs:maxInclusive value="125"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="descriptionType">
        <xs:restriction base="xs:string">
```

```
            <xs:minLength value="4"/>
            <xs:maxLength value="120"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="priceType">
```

```

<xs:restriction base="xs:decimal">
    <xs:fractionDigits value="2"/>
    <xs:minInclusive value="0"/>
    <xs:maxExclusive value="10"/>
</xs:restriction>
</xs:simpleType>
</xs:schema>

```

- menu.xml เป็นรายการอาหารที่เขียนอยู่ในรูปแบบภาษา XML ตามฟอร์มของ menu.xsd ดังนี้

```

<?xml version="1.0" encoding="UTF-8"?>
<menu>
    <name>Nutty Palace</name>
    <fax>(900) 715458</fax>
    <fax-order>Fax this menu back with your selections circled. If you want more than one item,
indicate the number of items in the number colum.Your order will be ready 30 minutes from when we
receive the fax.</fax-order>
    <category>
        <name>appetizers</name>
        <item number="1" veggie="true" spicy="false">
            <name>Asian appetizer</name>
            <description>This selection of authentic Asian foods includes 20 each of vegetable
gyoza, seafood shumai, sesame chicken, vegetable spring rolls and egg rolls.
            </description>
            <price>1.75</price>
        </item>
        <item number="2" veggie="true" spicy="false">
            <name>Kosher egg</name>
            <description> Serve egg rolls with an Oriental style dipping sauce of your
choice.</description>

```

	<pre> <price>2.25</price> </item> </category> <category> </pre>
	<pre> <name>soups</name> <item number="3" veggie="true" spicy="true"> </pre>
soup	<pre> <name>Great Northern Beans</name> <description>small white bean with a slightly nutty taste. It makes a delicious </description> <price>4.75</price> </item> <item number="4" veggie="false" spicy="false"> <name>Red Lentil </name> <description>The red lentils are great with rice, vegetables, or served alone as a nutritious meals</description> <price>5.25</price> </item> </category> <category> <name>salads</name> <item number="5" veggie="false" spicy="true"> <name>Larb</name> <description> </pre>
with cabbage	<pre> Ground eat mied with lime juice, onion, chili, ans mint leaves served </description> <price>3.75</price> </item> </pre>

```

<item number="6" veggie="false" spicy="true">
  <name>Yum Yai</name>
  <description>shrimp, chicke, cucumber, and onion served on abed of lettucc
with sweet and sour dressing</description>
  <price>4.15</price>
</item>

```

```

</category>
<category>
  <name>curry</name>
  <item number="7" veggie="false" spicy="true">
    <name>Prig King</name>
    <description>Green bean stir fry with chili paste</description>
    <price>5.65</price>
  </item>
  <item number="8" veggie="true" spicy="true">
    <name>Pa-Nang</name>
    <description>Ground peanut in curry with Thai spices and coconut
milk</description>
    <price>5.35</price>
  </item>
</category>

```

```

<category>

```

```

  <name>rice</name>

```

```

<item number="9" veggie="true" spicy="false">
  <name>House Fried Rice</name>
  <description>Fried rice with onion, tomat, peas, carrot, and egg</description>
  <price>4.35</price>
</item>

```

```

<item number="10" veggie="false" spicy="true">
  <name>spicy Fried Rice</name>
  <description>Fried rice with tomato, onion, bell pepper, and sweet basil</description>
  <price>4.35</price>
</item>
</category>

<category>
  <name>noodles</name>
  <item number="11" veggie="true" spicy="false">
    <name>Pad Thai</name>
    <description>Pan-fried rice nodles with egg, onion, and bean sprouts topped
with crushed peanuts</description>
    <price>5.65</price>
  </item>
  <item number="12" veggie="true" spicy="true">
    <name>Phad-See-Yew</name>
    <description>Pan-fried flat noodles with egg and broccoli in brown
sauce</description>
    <price>5.75</price>
  </item>
</category>
</menu>

```

ทั้ง menu.xsd และ menu.xml เก็บไว้ที่ c:\project\menu2

ไฟล์ menu.xml เป็นรายการอาหารที่เขียนในแบบภาษา XML ที่อธิบายลักษณะเมนูดังนี้

- Number (ลำดับรายการ)
- Veggie (สำหรับผู้ที่เป็นมังสาวิรัต)
- Spicy (รสเผ็ด)
- Name (ชื่อของอาหาร)

- Description (รายละเอียดของอาหาร)
 - Price (ราคามีหน่วยเป็นดอลลาร์)
- จากนั้นสร้างไฟล์ ThaiPalaceMenuService.java

```
//
// The class that implements the Web service
// This service is the Thai Palace Menu Web service
// A client can receive the Menu xml document, the Menu
// Schema document, and the Menu Stylesheet for generating
// an HTML menu representation. Other views can be supported
// by adding the appropriate methods, such as an XSL-FO
// method that returns an FO stylesheet that generates a PDF menu.
//
import java.io.*;
public class ThaiPalaceMenuService {
// The private class variables.
// The buffer size is used for reading the relevant files.
private final static int cBufferSize = 4096;
// Menu path
private final static String sMenuPath="c:\\project\\lmenu\\";
// The location of the Menu file
private final static String sMenu = "menu.xml" ;
// The location of the Menu Schema File
private final static String sMenuSchema = "menu.xsd";

// Here are the methods that implement the Web services.
public String getMenu() throws IOException {
    return(fileToString(sMenuPath + sMenu));
}

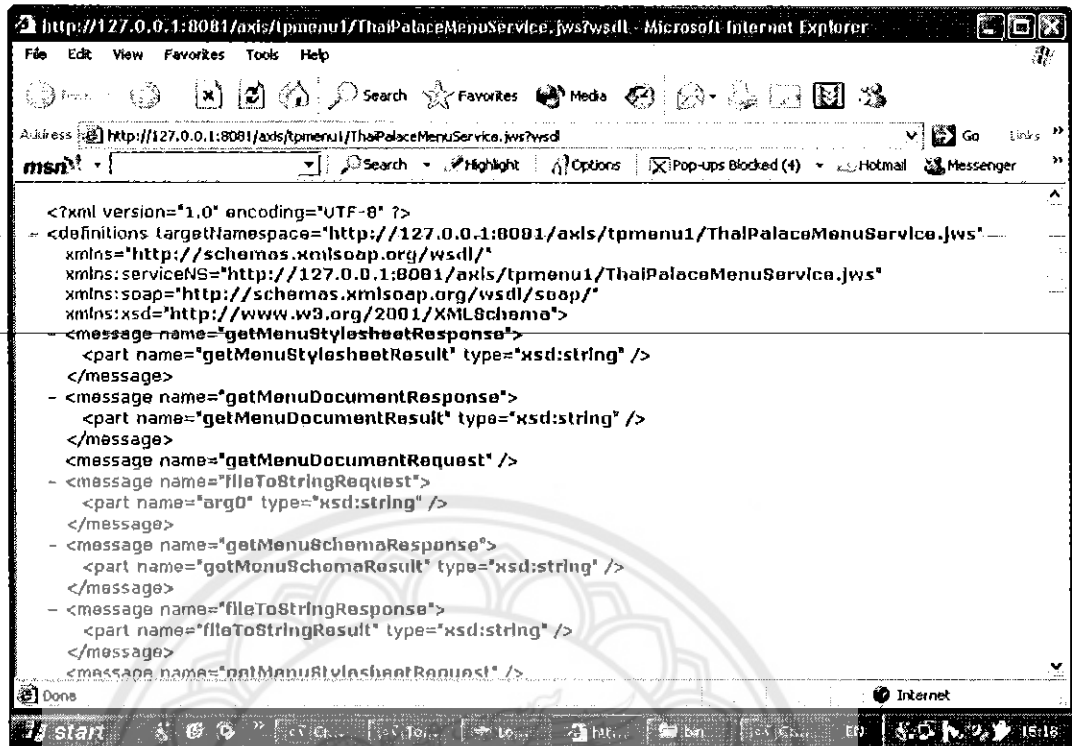
public String getMenuSchema() throws IOException {
```

```

        return(fileToString(sMenuPath + sMenuSchema));
    }
    //
    // This method reads afile into a String. It is based on recipe 905
    //
    private String fileToString(String filename) throws IOException {
        File fMenu = new File(filename);
        FileReader fr = new FileReader(fMenu);
        StringBuffer sb = new StringBuffer();
        char[] buffer = new char[cBufferSize];
        int n;
        while((n = fr.read(buffer)) > 0){
            sb.append(buffer, 0, n);
        }
        return(sb.toString());
    }
}

```

จากนั้นเปลี่ยน ThaiPalaceMenuService.java เป็น ThaiPalaceMenuService.jws แล้วเก็บไว้ที่ C:\Tomcat 5.0\webapps\axis\tpmenu1 แล้วสตาร์ท Server Tomcat เปิดโปรแกรม IE ขึ้นมาพิมพ์ <http://127.0.0.1:8081/axis/tpmenu1/ThaiPalaceMenuService.jws?wsdl> จะได้ผลลัพธ์ดังภาพ



รูปที่ 3.1 แสดงไฟล์ ThaiPalaceMenuService.wsdl ในรูปแบบ XML

จากนั้นสร้างไฟล์ NuttyMenuService.java

```
//
// The class that implements the Web service
// This service is the Thai Palace Menu Web service
// A client can receive the Menu xml document, the Menu
// Schema document, and the Menu Stylesheet for generating
// an HTML menu representation. Other views can be supported
// by adding the appropriate methods, such as an XSL-FO
// method that returns an FO stylesheet that generates a PDF menu.
//
import java.io.* ;
public class NuttyMenuService {
// The private class variables.
// The buffer size is used for reading the relevant files.
```



```
private final static int cBufferSize = 4096 ;
// Menu path
private final static String sMenuPath="c:\\project\\menu2\\" ;
// The location of the Menu file
private final static String sMenu = "menu.xml" ;
// The location of the Menu Schema File

private final static String sMenuSchema = "menu.xsd" ;
// Here are the methods that implement the Web services.
public String getMenu() throws IOException {
    return(fileToString(sMenuPath + sMenu)) ;
}
public String getMenuSchema() throws IOException {
    return(fileToString(sMenuPath + sMenuSchema)) ;
}
//
// This method reads afile into a String. It is based on recipe 905
//
private String fileToString(String filename) throws IOException {
    File fMenu = new File(filename) ;
    FileReader fr = new FileReader(fMenu) ;
    StringBuffer sb = new StringBuffer() ;
    char[] buffer = new char[cBufferSize] ;

    int n ;
    while((n = fr.read(buffer)) > 0){
        sb.append(buffer, 0, n) ;
    }
    return(sb.toString()) ;
}
}
```

จากนั้นเปลี่ยน NuttyMenuService.java เป็น NuttyMenuService.jws แล้วเก็บไว้ที่ C:\Tomcat 5.0\webapps\axis\tpmenu2 แล้วสตาร์ท Server Tomcat เปิดโปรแกรม IE ขึ้นมาพิมพ์ <http://127.0.0.1:8081/axis/tpmenu2/NuttyMenuService.jws?wsdl> จะได้ผลลัพธ์ดังภาพ

```

<?xml version="1.0" encoding="UTF-8" ?>
- <definitions targetNamespace="http://127.0.0.1:8081/axis/tpmenu2/NuttyMenuService.jws"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:serviceNS="http://127.0.0.1:8081/axis/tpmenu2/NuttyMenuService.jws"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
- <message name="getMenuSchemaResponse">
  <part name="getMenuSchemaResult" type="xsd:string" />
</message>
- <message name="getMenuResponse">
  <part name="getMenuResult" type="xsd:string" />
</message>
  <message name="getMenuSchemaRequest" />
  <message name="getMenuStyleSheetRequest" />
- <message name="fileToStringRequest">
  <part name="arg0" type="xsd:string" />
</message>
- <message name="fileToStringResponse">
  <part name="fileToStringResult" type="xsd:string" />
</message>
- <message name="getMenuDocumentResponse">
  <part name="getMenuDocumentResult" type="xsd:string" />
</message>
  </definitions>

```

รูปที่ 3.2 แสดงไฟล์ NuttyMenuService.wsdl ในรูปแบบ XML

3.2 The Menu WSDL

ไฟล์ ThaiPalaceMenuService.wsdl ที่ได้ก็คือ WSDL (Web Service Description Language) มีรายละเอียดดังนี้

```
<?xml version="1.0" encoding="UTF-8" ?>
- <definitions targetNamespace="http://192.168.67.56:8081/axis/tpmenu1/ThaiPalaceMenuService.jws"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:serviceNS="http://192.168.67.56:8081/axis/tpmenu1/ThaiPalaceMenuService.jws"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
- <message name="fileToStringResponse">
  <part name="fileToStringResult" type="xsd:string" />
</message>
  <message name="getMenuSchemaRequest" />
  <message name="getMenuStylesheetRequest" />
- <message name="getMenuResponse">
  <part name="getMenuResult" type="xsd:string" />
</message>
- <message name="fileToStringRequest">
  <part name="arg0" type="xsd:string" />
</message>
- <message name="getMenuDocumentResponse">
  <part name="getMenuDocumentResult" type="xsd:string" />
</message>
  <message name="getMenuDocumentRequest" />
- <message name="getMenuStylesheetResponse">
  <part name="getMenuStylesheetResult" type="xsd:string" />
</message>
- <message name="getMenuSchemaResponse">
  <part name="getMenuSchemaResult" type="xsd:string" />
```

```

</message>
  <message name="getMenuRequest" />
- <portType name="ThaiPalaceMenuServicePortType">
- <operation name="getMenu">
  <input message="serviceNS:getMenuRequest" />
  <output message="serviceNS:getMenuResponse" />
</operation>
- <operation name="getMenuSchema">
  <input message="serviceNS:getMenuSchemaRequest" />
  <output message="serviceNS:getMenuSchemaResponse" />
</operation>
- <operation name="getMenuDocument">
  <input message="serviceNS:getMenuDocumentRequest" />
  <output message="serviceNS:getMenuDocumentResponse" />
</operation>
- <operation name="getMenuStylesheet">
  <input message="serviceNS:getMenuStylesheetRequest" />
  <output message="serviceNS:getMenuStylesheetResponse" />
</operation>
- <operation name="fileToString">
  <input message="serviceNS:fileToStringRequest" />
  <output message="serviceNS:fileToStringResponse" />
</operation>
</portType>
- <binding name="ThaiPalaceMenuServiceSoapBinding"
  type="serviceNS:ThaiPalaceMenuServicePortType">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
- <operation name="getMenu">
  <soap:operation soapAction="" style="rpc" />

```

```

- <input>
  <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace=""
    use="encoded" />
</input>
- <output>
  <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace=""
    use="encoded" />
</output>
</operation>
- <operation name="getMenuSchema">
  <soap:operation soapAction="" style="rpc" />
- <input>
  <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace=""
    use="encoded" />
</input>
- <output>
  <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace=""
    use="encoded" />
</output>
</operation>
- <operation name="getMenuDocument">
  <soap:operation soapAction="" style="rpc" />
- <input>
  <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace=""
    use="encoded" />
</input>
- <output>
  <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace=""
    use="encoded" />

```

```

        </output>
    </operation>
    - <operation name="getMenuStylesheet">
        <soap:operation soapAction="" style="rpc" />
    - <input>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace=""
        use="encoded" />
    </input>
    - <output>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace=""
        use="encoded" />
    </output>
    </operation>
    - <operation name="fileToString">
        <soap:operation soapAction="" style="rpc" />
    - <input>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace=""
        use="encoded" />
    </input>
    - <output>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace=""
        use="encoded" />
    </output>
    </operation>
</binding>
- <service name="ThaiPalaceMenuService">
- <port binding="serviceNS:ThaiPalaceMenuServiceSoapBinding"
    name="ThaiPalaceMenuServicePort">
    <soap:address location="http://192.168.67.56:8081/axis/tpmenu1/ThaiPalaceMenuService.jws" />

```

```

</port>

</service>

</definitions>

```

และไฟล์ NuttyMenuService.wsdl ที่ได้ก็คือ WSDL (Web Service Description Language) มีรายละเอียดดังนี้

```


<?xml version="1.0" encoding="UTF-8" ?>
- <definitions targetNamespace="http://192.168.67.56:8081/axis/tpmenu2/NuttyMenuService.jws"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:serviceNS="http://192.168.67.56:8081/axis/tpmenu2/NuttyMenuService.jws"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
- <message name="getMenuResponse">
  <part name="getMenuResult" type="xsd:string" />
</message>
  <message name="getMenuDocumentRequest" />
- <message name="fileToStringRequest">
  <part name="arg0" type="xsd:string" />
</message>
- <message name="getMenuStylesheetResponse">
  <part name="getMenuStylesheetResult" type="xsd:string" />
</message>
- <message name="getMenuDocumentResponse">
  <part name="getMenuDocumentResult" type="xsd:string" />
</message>
- <message name="getMenuSchemaResponse">
  <part name="getMenuSchemaResult" type="xsd:string" />
</message>
  <message name="getMenuRequest" />
- <message name="fileToStringResponse">

```

```

    <part name="fileToStringResult" type="xsd:string" />
  </message>
  <message name="getMenuStylesheetRequest" />
  <message name="getMenuSchemaRequest" />
- <portType name="NuttyMenuServicePortType">
- <operation name="getMenu">
  <input message="serviceNS:getMenuRequest" />
  <output message="serviceNS:getMenuResponse" />
  </operation>
- <operation name="getMenuSchema">
  <input message="serviceNS:getMenuSchemaRequest" />
  <output message="serviceNS:getMenuSchemaResponse" />
  </operation>
- <operation name="getMenuDocument">
  <input message="serviceNS:getMenuDocumentRequest" />
  <output message="serviceNS:getMenuDocumentResponse" />
  </operation>
- <operation name="getMenuStylesheet">
  <input message="serviceNS:getMenuStylesheetRequest" />
  <output message="serviceNS:getMenuStylesheetResponse" />
  </operation>
- <operation name="fileToString">
  <input message="serviceNS:fileToStringRequest" />
  <output message="serviceNS:fileToStringResponse" />
  </operation>
</portType>
- <binding name="NuttyMenuServiceSoapBinding" type="serviceNS:NuttyMenuServicePortType">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
- <operation name="getMenu">

```


<pre> <soap:operation soapAction="" style="rpc" /> - <input> <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="" use="encoded" /> </input> - <output> </pre>	
<pre> <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="" use="encoded" /> </output> </operation> - <operation name="getMenuSchema"> <soap:operation soapAction="" style="rpc" /> - <input> <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="" use="encoded" /> </input> - <output> <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="" use="encoded" /> </output> </operation> - <operation name="getMenuDocument"> <soap:operation soapAction="" style="rpc" /> - <input> </pre>	
<pre> <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="" use="encoded" /> </input> - <output> <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="" </pre>	

```

    use="encoded" />
    </output>
  </operation>
- <operation name="getMenuStylesheet">
    <soap:operation soapAction="" style="rpc" />
- <input>

    <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace=""
    use="encoded" />
    </input>
- <output>
    <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace=""
    use="encoded" />
    </output>
  </operation>
- <operation name="fileToString">
    <soap:operation soapAction="" style="rpc" />
- <input>
    <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace=""
    use="encoded" />
    </input>
- <output>
    <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace=""
    use="encoded" />
    </output>
  </operation>
</binding>
- <service name="NuttyMenuService">
- <port binding="serviceNS:NuttyMenuServiceSoapBinding" name="NuttyMenuServicePort">
    <soap:address location="http://192.168.67.56:8081/axis/tpmenu2/NuttyMenuService.jws" />

```

```

</port>
</service>
</definitions>

```

3.3 The Menu Web Service Client

SiamDelivery ก็จะนำไฟล์ WSDL ของ ThaiPalaceMenuService.wsdl และ NuttyMenuService.wsdl มาสร้างเป็น Web Service Client เพื่อขอรับเซอร์วิสรายการอาหารมาแสดง ดังนี้

- สร้าง Folder sd3 ไว้ใน ดังนี้ C:\Tomcat 5.0\webapps\ ภายใน Folder sd3 จะมี Folder WEB-INF และ ภายใน Folder WEB-INF จะประกอบด้วย 2 Folder คือ

1. lib ภายในมีไฟล์ที่ copy มาจาก C:\Tomcat 5.0\webapps\axis\WEB-INF\lib

2. classes

- ทำการ copy ไฟล์ axis.jar, chutil.jar, log4j-core.jar, wsdl4j.jar, xerces.jar จาก axis-1_0 ไปไว้ที่ C:\Program Files\Java\jdk1.5.0_01\jre\lib\ext เพื่อให้สามารถเรียกใช้งานโดยไม่ต้องเซตค่า path

- ทำการ copy ไฟล์ servlet-api.jar จาก C:\Tomcat 5.0\common\lib\ ไปไว้ที่ C:\Program Files\Java\jdk1.5.0_01\jre\lib\ext เพื่อให้สามารถใช้งาน Servlet ได้

- ใช้คำสั่งดังนี้เพื่อให้ Axis สร้างไฟล์ติดต่อขอเรียกใช้ ThaiPalaceMenuService

```

C:\Tomcat 5.0\webapps\sd3\WEB-INF\classes>java org.apache.axis.wsdl.Wsdl2java --verbose
http://192.168.67.56:8081/axis/tpmenu/ThaiPalaceMenuService.jws?wsdl

```

และ NuttyMenuService

```

C:\Tomcat 5.0\webapps\sd3\WEB-INF\classes>java org.apache.axis.wsdl.Wsdl2java --verbose
http://192.168.67.56:8081/axis/tpmenu2/NuttyMenuService.jws?wsdl

```

Note: 192.168.67.56 เป็น IP Address ของฝั่ง ThaiPalace และ NuttyMenuService ส่วน 8081 คือ Port ที่ใช้

```

E:\system32\cmd.exe
04/01/2005  10:10 PM          567 ThaiPalaceMenuServicePortType.java
04/01/2005  10:10 PM         4,000 ThaiPalaceMenuServiceSoapBindingStub.java

04/01/2005  10:10 PM          1,098 ThaiPalaceMenuService.java
          5 File(s)          0,480 bytes
          5 Dir(s)         20,110,040 bytes free

E:\Tomcat 5.0\webapps\sd1\WEB-INF\classes>java org.apache.axis.wsdl.Wsd12.java --
verbose http://192.168.67.29:8001/axis/tpmenu2/ThaiPalaceMenuService.jws?wsdl
Parsing XML File: http://192.168.67.29:8001/axis/tpmenu2/ThaiPalaceMenuService.j
ws?wsdl
Generating portType interface: ThaiPalaceMenuServicePortType.java
Generating client-side stub: ThaiPalaceMenuServiceSoapBindingStub.java
Generating service class: ThaiPalaceMenuService.java

E:\Tomcat 5.0\webapps\sd1\WEB-INF\classes>_

```

รูปที่ 3.3 แสดงการ Generating โดยใช้ Axis

จะได้ไฟล์ 3 ไฟล์ของ ThaiPalaceMenuService ที่ถูก Generating ดังนี้

1. ThaiPalaceMenuServicePortType.java

```

/**
 * ThaiPalaceMenuServicePortType.java
 *
 * This file was auto-generated from WSDL
 * by the Apache Axis Wsd12java emitter.
 */
public interface ThaiPalaceMenuServicePortType extends java.rmi.Remote {
    public String fileToString(String arg0) throws java.rmi.RemoteException;
    public String getMenu() throws java.rmi.RemoteException;
    public String getMenuDocument() throws java.rmi.RemoteException;
    public String getMenuSchema() throws java.rmi.RemoteException;
    public String getMenuStylesheet() throws java.rmi.RemoteException;
}

```

2. ThaiPalaceMenuService.java

```
/**
 * ThaiPalaceMenuService.java
 * This file was auto-generated from WSDL
 * by the Apache Axis Wsd12java emitter.
 */

public class ThaiPalaceMenuService {

    // Use to get a proxy class for ThaiPalaceMenuServicePort
    private final java.lang.String ThaiPalaceMenuServicePort_address =
"http://192.168.67.29:8081/axis/tpmenu2/ThaiPalaceMenuService.jws";

    public ThaiPalaceMenuServicePortType getThaiPalaceMenuServicePort() {
        java.net.URL endpoint;
        try {
            endpoint = new java.net.URL(ThaiPalaceMenuServicePort_address);
        }
        catch (java.net.MalformedURLException e) {
            return null; // unlikely as URL was validated in wsdl2java
        }
        return getThaiPalaceMenuServicePort(endpoint);
    }

    public ThaiPalaceMenuServicePortType getThaiPalaceMenuServicePort(java.net.URL portAddress) {
        try {
            return new ThaiPalaceMenuServiceSoapBindingStub(portAddress);
        }

        catch (org.apache.axis.SerializationException e) {
            return null; // ???
        }
    }
}
```

3. ThaiPalaceMenuServiceSoapBindingStub.java

```

/**
 * ThaiPalaceMenuServiceSoapBindingStub.java
 *
 * This file was auto-generated from WSDL
 * by the Apache Axis Wsd12java emitter.
 */
public class ThaiPalaceMenuServiceSoapBindingStub extends org.apache.axis.wsdl.Stub implements
ThaiPalaceMenuServicePortType {
    private org.apache.axis.client.ServiceClient call = new org.apache.axis.client.ServiceClient(new
org.apache.axis.transport.http.HTTPTransport());
    private java.util.Hashtable properties = new java.util.Hashtable();
    public ThaiPalaceMenuServiceSoapBindingStub(java.net.URL endpointURL) throws
org.apache.axis.SerializationException {
        this();
        call.set(org.apache.axis.transport.http.HTTPTransport.URL, endpointURL.toString());
    }
    public ThaiPalaceMenuServiceSoapBindingStub() throws org.apache.axis.SerializationException {
    }
    public void _setProperty(String name, Object value) {
        properties.put(name, value);
    }
    //From org.apache.axis.wsdl.Stub
    public Object _getProperty(String name) {
        return properties.get(name);
    }
    // From org.apache.axis.wsdl.Stub
    public void _setTargetEndpoint(java.net.URL address) {

```

```
call.set(org.apache.axis.transport.http.HTTPTransport.URL, address.toString());
}

// From org.apache.axis.wsdl.Stub
public java.net.URL _getTargetEndpoint() {
    try {
        return new java.net.URL((String) call.get(org.apache.axis.transport.http.HTTPTransport.URL));
    }
    catch (java.net.MalformedURLException mue) {
        return null; // ???
    }
}

// From org.apache.axis.wsdl.Stub
public synchronized void setMaintainSession(boolean session) {
    call.setMaintainSession(session);
}

// From javax.naming.Referenceable
public javax.naming.Reference getReference() {
    return null; // ???
}

public String fileToString(String arg0) throws java.rmi.RemoteException {
    if (call.get(org.apache.axis.transport.http.HTTPTransport.URL) == null) {
        throw new org.apache.axis.NoEndPointException();
    }
}

call.set(org.apache.axis.transport.http.HTTPTransport.ACTION, "");

Object resp = call.invoke("", "fileToString", new Object[] {new
org.apache.axis.message.RPCParam("arg0", arg0)});

if (resp instanceof java.rmi.RemoteException) {
    throw (java.rmi.RemoteException)resp;
}
```

```
    }  
    else {  
        return (String) resp;  
    }  
}  
  
public String getMenu() throws java.rmi.RemoteException {  
    if (call.get(org.apache.axis.transport.http.HTTPTransport.URL) == null) {  
        throw new org.apache.axis.NoEndPointException();  
    }  
    call.set(org.apache.axis.transport.http.HTTPTransport.ACTION, "");  
    Object resp = call.invoke("", "getMenu", new Object[] {});  
    if (resp instanceof java.rmi.RemoteException) {  
        throw (java.rmi.RemoteException)resp;  
    }  
    else {  
        return (String) resp;  
    }  
}  
  
public String getMenuDocument() throws java.rmi.RemoteException {  
    if (call.get(org.apache.axis.transport.http.HTTPTransport.URL) == null) {  
        throw new org.apache.axis.NoEndPointException();  
    }  
    call.set(org.apache.axis.transport.http.HTTPTransport.ACTION, "");  
    Object resp = call.invoke("", "getMenuDocument", new Object[] {});  
    if (resp instanceof java.rmi.RemoteException) {  
        throw (java.rmi.RemoteException)resp;  
    }  
    else {  
        return (String) resp;  
    }  
}
```



```
    }  
  }  
  public String getMenuSchema() throws java.rmi.RemoteException {  
    if (call.get(org.apache.axis.transport.http.HTTPTransport.URL) == null) {  
      throw new org.apache.axis.NoEndPointException();  
    }  
  }
```

```
    call.set(org.apache.axis.transport.http.HTTPTransport.ACTION, "");  
    Object resp = call.invoke("", "getMenuSchema", new Object[] {});  
    if (resp instanceof java.rmi.RemoteException) {  
      throw (java.rmi.RemoteException)resp;  
    }  
    else {  
      return (String) resp;  
    }  
  }  
  public String getMenuStylesheet() throws java.rmi.RemoteException {  
    if (call.get(org.apache.axis.transport.http.HTTPTransport.URL) == null) {  
      throw new org.apache.axis.NoEndPointException();  
    }  
    call.set(org.apache.axis.transport.http.HTTPTransport.ACTION, "");  
    Object resp = call.invoke("", "getMenuStylesheet", new Object[] {});  
    if (resp instanceof java.rmi.RemoteException) {  
      throw (java.rmi.RemoteException)resp;  
    }  
    else {  
      return (String) resp;  
    }  
  }  
}
```

และจะได้ไฟล์ 3 ไฟล์ของ NuttyMenuService ที่ถูก Generating ดังนี้

1. NuttyMenuServicePortType.java

```
/**
 * NuttyMenuServicePortType.java
 *
 * This file was auto-generated from WSDL
 * by the Apache Axis Wsd12java emitter.
 */
public interface NuttyMenuServicePortType extends java.rmi.Remote {
    public String fileToString(String arg0) throws java.rmi.RemoteException;
    public String getMenu() throws java.rmi.RemoteException;
    public String getMenuDocument() throws java.rmi.RemoteException;
    public String getMenuSchema() throws java.rmi.RemoteException;
    public String getMenuStylesheet() throws java.rmi.RemoteException;
}
```

2. NuttyMenuService.java

```
/**
 * NuttyMenuService.java
 * This file was auto-generated from WSDL
 * by the Apache Axis Wsd12java emitter.
 */
public class NuttyMenuService {
    // Use to get a proxy class for NuttyMenuServicePort
    private final java.lang.String NuttyMenuServicePort_address =
"http://192.168.67.56:8081/axis/tpmenu2/NuttyMenuService.jws";
    public NuttyMenuServicePortType getNuttyMenuServicePort() {
        java.net.URL endpoint;
        try {
            endpoint = new java.net.URL(NuttyMenuServicePort_address);
```

```

    }
    catch (java.net.MalformedURLException e) {
        return null; // unlikely as URL was validated in wsdl2java
    }
    return getNuttingMenuServicePort(endpoint);
}

```

```

public NuttingMenuServicePortType getNuttingMenuServicePort(java.net.URL portAddress) {
    try {
        return new NuttingMenuServiceSoapBindingStub(portAddress);
    }
    catch (org.apache.axis.SerializationException e) {
        return null; // ???
    }
}
}
}

```

3. NuttingMenuServiceSoapBindingStub.java

```

/**
 * NuttingMenuServiceSoapBindingStub.java
 * This file was auto-generated from WSDL
 * by the Apache Axis Wsdl2java emitter.
 */
public class NuttingMenuServiceSoapBindingStub extends org.apache.axis.wsdl.Stub implements
NuttingMenuServicePortType {
    private org.apache.axis.client.ServiceClient call = new org.apache.axis.client.ServiceClient(new
org.apache.axis.transport.http.HTTPTransport());
    private java.util.Hashtable properties = new java.util.Hashtable();

    public NuttingMenuServiceSoapBindingStub(java.net.URL endpointURL) throws
org.apache.axis.SerializationException {

```

```
this();
    call.set(org.apache.axis.transport.http.HTTPTransport.URL, endpointURL.toString());
}
public NuttyMenuServiceSoapBindingStub() throws org.apache.axis.SerializationException {
}
public void _setProperty(String name, Object value) {
    properties.put(name, value);
}
// From org.apache.axis.wsdl.Stub
public Object _getProperty(String name) {
    return properties.get(name);
}
// From org.apache.axis.wsdl.Stub
public void _setTargetEndpoint(java.net.URL address) {
    call.set(org.apache.axis.transport.http.HTTPTransport.URL, address.toString());
}
// From org.apache.axis.wsdl.Stub
public java.net.URL _getTargetEndpoint() {
    try {
        return new java.net.URL((String) call.get(org.apache.axis.transport.http.HTTPTransport.URL));
    }
    catch (java.net.MalformedURLException mue) {
        return null; // ???
    }
}
}
// From org.apache.axis.wsdl.Stub
public synchronized void setMaintainSession(boolean session) {
    call.setMaintainSession(session);
}
}
```

```
// From javax.naming.Referenceable
public javax.naming.Reference getReference() {
    return null; // ???
}

public String fileToString(String arg0) throws java.rmi.RemoteException {
    if (call.get(org.apache.axis.transport.http.HTTPTransport.URL) == null) {

        throw new org.apache.axis.NoEndPointException();
    }

    call.set(org.apache.axis.transport.http.HTTPTransport.ACTION, "");
    Object resp = call.invoke("", "fileToString", new Object[] {new
org.apache.axis.message.RPCParam("arg0", arg0)});
    if (resp instanceof java.rmi.RemoteException) {
        throw (java.rmi.RemoteException)resp;
    }
    else {
        return (String) resp;
    }
}

public String getMenu() throws java.rmi.RemoteException {
    if (call.get(org.apache.axis.transport.http.HTTPTransport.URL) == null) {
        throw new org.apache.axis.NoEndPointException();
    }

    call.set(org.apache.axis.transport.http.HTTPTransport.ACTION, "");

    Object resp = call.invoke("", "getMenu", new Object[] {});

    if (resp instanceof java.rmi.RemoteException) {
        throw (java.rmi.RemoteException)resp;
    }
    else {
        return (String) resp;
    }
}
```

```

    }
}
public String getMenuDocument() throws java.rmi.RemoteException {
    if (call.get(org.apache.axis.transport.http.HTTPTransport.URL) == null) {
        throw new org.apache.axis.NoEndPointException();
    }
}

```

```

    call.set(org.apache.axis.transport.http.HTTPTransport.ACTION, "");
    Object resp = call.invoke("", "getMenuDocument", new Object[] {});
    if (resp instanceof java.rmi.RemoteException) {
        throw (java.rmi.RemoteException)resp;
    }
    else {
        return (String) resp;
    }
}

public String getMenuSchema() throws java.rmi.RemoteException {
    if (call.get(org.apache.axis.transport.http.HTTPTransport.URL) == null) {
        throw new org.apache.axis.NoEndPointException();
    }
    call.set(org.apache.axis.transport.http.HTTPTransport.ACTION, "");
    Object resp = call.invoke("", "getMenuSchema", new Object[] {});
    if (resp instanceof java.rmi.RemoteException) {
        throw (java.rmi.RemoteException)resp;
    }
}

```

```

    else {
        return (String) resp;
    }
}

public String getMenuStylesheet() throws java.rmi.RemoteException {

```

```

if (call.get(org.apache.axis.transport.http.HTTPTransport.URL) == null) {
    throw new org.apache.axis.NoEndPointException();
}
call.set(org.apache.axis.transport.http.HTTPTransport.ACTION, "");
Object resp = call.invoke("", "getMenuStylesheet", new Object[] {});
if (resp instanceof java.rmi.RemoteException) {
    throw (java.rmi.RemoteException)resp;
}
else {
    return (String) resp;
}
}
}

```

ไฟล์ที่ได้ทั้งหมดจะอยู่ที่ C:\Tomcat 5.0\webapps\sd3\WEB-INF\classes

3.4 The SiamDelivery Server

- Deploying the Web Application

สร้าง Context deployment descriptor เก็บไว้ที่ webapps\sd3\WEB_INF ของ Server Tomcat web.xml แสดงดังนี้

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
    "http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
    <!-- Define servlets that are included in the SiamDelivery application -->
    <servlet>
        <servlet-name>SiamDeliveryServer</servlet-name>
        <servlet-class>SiamDeliveryServer</servlet-class>
    </servlet>

```

```

    <servlet-mapping>
        <servlet-name>SiamDeliveryServer</servlet-name>
        <url-pattern>/SiamDeliveryServer</url-pattern>
    </servlet-mapping>
</web-app>

```

3.5 The Servlet

สร้าง Servlet ไฟล์ชื่อ SiamDeliveryServer.java เป็นตัวควบคุมการทำงานของ SiamDelivery ดังนี้

```

import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.Serializable;

public class SiamDeliveryServer extends HttpServlet {
    public void doGet(HttpServletRequest request,HttpServletResponse response) throws
ServletException, IOException {
        processRequest(request, response);
    }
    public void doPost(HttpServletRequest request,HttpServletResponse response) throws
ServletException, IOException {
        processRequest(request, response);
    }
    public void processRequest(HttpServletRequest request,HttpServletResponse response) throws
ServletException,IOException {
        Enumeration e = request.getParameterNames();
        PrintWriter out = response.getWriter();

        String actionName = null;

        try

```



```

        {
            actionName = request.getParameter("actionName");
        }
    catch (Exception ex)
    {
    } finally {

        if (null == actionName ) {
getServletConfig().getServletContext().getRequestDispatcher("/welcome.jsp").forward(request, response)
;

        }else if(actionName.equals("menu")) {
            bean.SiamDeliveryMenuBean bgBean = new bean.SiamDeliveryMenuBean() ;
while (e.hasMoreElements() ) {
            String name = (String)e.nextElement() ;
            String value = request.getParameter (name) ;
            System.out.println(name + "\t" + value) ;
            if(name.equals("restaurant")) {
                if(value.equals("thaiPalace")) {
                    ThaiPalaceMenuService service = new ThaiPalaceMenuService() ;
                    ThaiPalaceMenuServicePortType port = service.getThaiPalaceMenuServicePort() ;
                    bgBean.setMenu(port.getMenu()) ;
                    request.setAttribute("menuBean", bgBean) ;

                    getServletConfig().getServletContext().getRequestDispatcher("/menu.jsp").forward(request,
response) ;
                }

                if(value.equals("nuttyPalace")) {
                    NuttyMenuService service = new NuttyMenuService() ;
                    NuttyMenuServicePortType port = service.getNuttyMenuServicePort() ;

                    bgBean.setMenu(port.getMenu()) ;
                }
            }
        }
    }
}

```


3.6 The JavaServer Page

สร้าง Welcome.jsp เก็บไว้ที่ C:\Tomcat 5.0\webapps\sd3 ซึ่งเป็นหน้าแรกที่แสดงทุกครั้งที่มีการใช้บริการของ SiamDelivery และเป็นหน้าแสดงรายการให้เลือกร้านอาหาร แสดงดังนี้

```
<html>
<head>
<title> Welcome to SiamDeliveryServer </title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"></head>
<body bgcolor="#006699" background="pig/flowerBG6.gif">
<table width="97%" height="80" border="0" align="center" bordercolor="#000000"
bgcolor="#FFFFFF">
<tr>
<td width="9%" height="76"></td>
<td width="80%"><div align="center"><font color="#666666" size="+7"><strong>Welcome
to SiamDelivery </strong></font></div></td>
<td width="11%"></td>
</tr>
</table>
<table width="13%" align="center">
<tr>
<td></td>
</tr>
</table>
<hr>
<h4 align="center"><font color="#FFFFFF" ><font color="#FF0000">Select</font>
you favorite <font color="#FFFF00">restaurant</font>
from the list and you will be given its menu. </font></h4>
<table width="30%" align="center">
<tr>
```

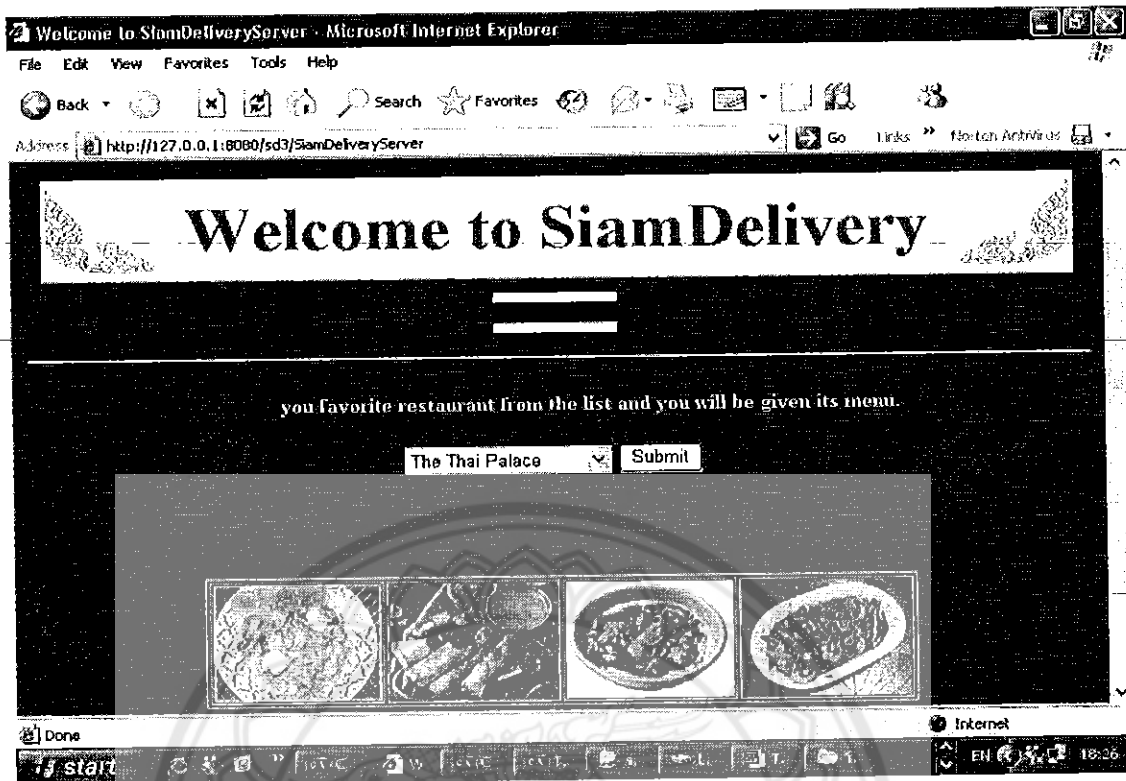
```

<td><form method="POST" action="SiamDeliveryServer?actionName=menu">
<select name="restaurant">
<option value="thaiPalace" selected>The Thai Palace
<option value="thaiPalace">VeggieBurger Palace
<option value="thaiPalace" >Pizza Palace
<option value="nuttyPalace">Nutty Palace

</select>
<input type="submit" value="Submit">

</form></td>
</tr>
</table>
<p>&nbsp;</p>
<table width="66%" height="96" border="1" align="center">
<tr>
<td width="28%"></td>
<td width="28%"></td>
<td width="31%"></td>
<td width="13%"></td>
</tr>
</table>
<p>&nbsp;</p>
<p align="center">&nbsp;</p>
</body>
</html>

```



รูปที่ 3.4 แสดงหน้า welcome

เนื่องจากเราใช้ JavaBean จึงต้องมีตัวเรียกใช้ซึ่งก็คือ menu.jsp สำหรับเรียก thaiPalace และ menu2.jsp สำหรับเรียก nuttyPalace เก็บไว้ที่ C:\Tomcat 5.0\webapps\sd3 แสดงดังนี้

- menu.jsp

```
<%@ taglib uri="http://jakarta.apache.org/taglibs/xtags-1.0" prefix="xtags" %>
<%@ page import = "bean.SiamDeliveryMenuBean" %>
<%
    SiamDeliveryMenuBean menu = (SiamDeliveryMenuBean)request.getAttribute("menuBean");
%>

<xtags:parse id="mydoc">
<%=menu.getMenu()%>
</xtags:parse>

<xtags:style xsl="test_stylesheet.xsl" document="<%= mydoc %>"/>
```

- menu2.jsp

```
<%@ taglib uri="http://jakarta.apache.org/taglibs/xtags-1.0" prefix="xtags" %>
<%@ page import = "bean.SiamDeliveryMenuBean" %>
<%
    SiamDeliveryMenuBean menu =
(SiamDeliveryMenuBean)request.getAttribute("menuBean");
%>
<xtags:parse id="mydoc">
<%=menu.getMenu()%>
</xtags:parse>
<xtags:style xsl="test_stylesheet2.xsl" document="<%= mydoc %>"/>
```

ข้อสังเกตส่วนที่เป็น xtags จะอธิบายในหัวข้อถัดไป

3.7 XTags library

เนื่องจากข้อมูลที่ส่งมาจาก Web Service เป็นภาษา XML ฉะนั้นเราจะต้องทำการแปลงให้อยู่ในรูปแบบที่เราต้องการ ในที่นี้จะให้แสดงออกมาอยู่ในรูป HTML สิ่งจำเป็นที่จะต้องมียกคือ

-ไฟล์ taglibs-xtags.tld เก็บไว้ที่ C:\Tomcat 5.0\webapps\sd3\WEB-INF

-ไฟล์ dom4j-1.5.2.jar เก็บไว้ที่ C:\Tomcat 5.0\webapps\sd3\WEB-INF\lib

จากนั้นสร้างไฟล์ test_stylesheet.xsl สำหรับ menu.jsp และ test_stylesheet2.xsl สำหรับ menu2.jsp เพื่อกำหนดรูปแบบข้อมูลภาษา XML ให้แสดงผลออกมาเป็น HTML ตามที่เราต้องการ

มีรูปแบบดังนี้

1.test_stylesheet.xsl

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
<xsl:output method="html" />
<xsl:template match="/" >
```

```

<html>
  <xsl:apply-templates />
</html>
</xsl:template>
<xsl:template match="menu">
  <head>
    <title>
      <xsl:value-of select=" name" />
    </title>
  </head>
  <body bgcolor="#FFFF99">
    <h1>
      <font color="#0000FF">
        <xsl:value-of select="name" /> :
      </font>
      <font color="#FF0000">
        <xsl:value-of select="fax" /> (FAX )
      </font>
    </h1>
    <h4>
      <xsl:value-of select="fax-order" />
    </h4>
    <hr />
  <xsl:apply-templates />
</body>
</xsl:template>
<xsl:template match="category">
  <h2 align="center">

```

```

        <xsl:value-of select="name" />
    </h2>
<table border="1" width="100%">
    <tr>
        <th width="10%" bgcolor="#CC9900">Number</th>
        <th width="10%" bgcolor="#CC9900">Veggie</th>
        <th width="10%" bgcolor="#CC9900">Spicy</th>
        <th width="10%" bgcolor="#CC9900">Name</th>
        <th width="50%" bgcolor="#CC9900">Description</th>
        <th width="10%" bgcolor="#CC9900">Price</th>
    </tr>
    <xsl:apply-templates />
</table>
</xsl:template>
<xsl:template match="item">
    <tr>
        <td align="center">
            <xsl:value-of select="@number" />
        </td>
        <td align="center">
            <xsl:choose>
                <xsl:when test="@veggie='true' ">Yes</xsl:when>
                <xsl:when test="@veggie='false' ">No</xsl:when>
            </xsl:choose>
        </td>
        <td align="center">
            <xsl:choose>
                <xsl:when test="@spicy='true' ">Yes</xsl:when>
                <xsl:when test="@spicy='false' ">No</xsl:when>
            </xsl:choose>
        </td>
    </tr>
</xsl:template>

```



```

</xsl:choose>
    </td>
    <td align="center" bgcolor="#FFFF00">
        <xsl:value-of select="name" />
    </td>
    <td align="left" bgcolor="#99CC00">
        <xsl:value-of select="description" />
    </td>
    <td align="right">$
        <xsl:value-of select="price" />
    </td>
</tr>
</xsl:template>
<xsl:template match="*" />
</xsl:stylesheet>

```

2.test_stylesheer2.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
<xsl:output method="html" />

<xsl:template match="/" >
    <html>
        <xsl:apply-templates />
    </html>
</xsl:template>

<xsl:template match="menu">
    <head>

```

```

<title>
    <xsl:value-of select=" name" />
</title>
</head>
<body bgcolor="#CCFF99">
<h1>
<font color="#FF0000">
<xsl:value-of select="name" /> :
</font>
<font color="#009900">
<xsl:value-of select="fax" /> (FAX )
</font>
</h1>
<h4>
<xsl:value-of select="fax-order" />
</h4>
<hr />
<xsl:apply-templates />
</body>
</xsl:template>
<xsl:template match="category">
<h2 align="center">
    <xsl:value-of select="name" />
</h2>
<table border="1" width="100%">
<tr>
<th width="10%" bgcolor="#C0C0C0">Number</th>
<th width="10%" bgcolor="#C0C0C0">Veggie</th>
<th width="10%" bgcolor="#C0C0C0">Spicy</th>

```

<pre> <th width="10%" bgcolor="#C0C0C0">Name</th> <th width="50%" bgcolor="#C0C0C0">Description</th> <th width="10%" bgcolor="#C0C0C0">Price</th> </tr> </pre>	
<pre> <xsl:apply-templates /> </table> </pre>	<pre> </xsl:template> <xsl:template match="item"> <tr> <td align="center"> <xsl:value-of select="@number" /> </td> <td align="center"> <xsl:choose> <xsl:when test="@veggie='true' ">Yes</xsl:when> <xsl:when test="@veggie='false' ">No</xsl:when> </xsl:choose> </td> <td align="center"> <xsl:choose> <xsl:when test="@spicy='true' ">Yes</xsl:when> <xsl:when test="@spicy='false' ">No</xsl:when> </xsl:choose> </td> </tr> </pre>
<pre> <xsl:choose> <xsl:when test="@spicy='true' ">Yes</xsl:when> <xsl:when test="@spicy='false' ">No</xsl:when> </xsl:choose> </td> <td align="center" bgcolor="#99CC00"> <xsl:value-of select="name" /> </td> </pre>	

```

<td align="left" bgcolor="#FFCC00">
    <xsl:value-of select="description" />
</td>
<td align="right">$
    <xsl:value-of select="price" />
</td>
</tr>
</xsl:template>
<xsl:template match="*" />
</xsl:stylesheet>

```

3.8 การเรียกใช้งาน SiamDeliveryServer

<http://127.0.0.1:8080/sd3/SiamDeliveryServer> จะปรากฏหน้าแสดงการต้อนรับของ SiamDeliveryServer พร้อมกับรายการร้านอาหารให้เลือก ดังนี้



รูปที่แสดง 3.5 หน้า Welcome

เมื่อคลิกเลือก The Thai Palaece จะปรากฏ หน้าเมนูดังนี้

Thai Palace : (800) 999999 (FAX)

Fax this menu back with your selections circled. If you want more than one item, indicate the number of items in the number column. Your order will be ready 30 minutes from when we receive the fax.

appetizers

Number	Veggie	Spicy	Name	Description	Price
1	Yes	No	Spring Roll	Thai style egg roll with glass noodles and vegetables served in a plum sauce.	\$ 1.75
2	Yes	No	Fried Tofu	Deep-fried tofu served with a plum sauce.	\$ 2.25

soups

Number	Veggie	Spicy	Name	Description	Price
3	Yes	Yes	Tom Yum	Mushroom in hot and sour soup with a touch of lemon grass.	\$ 4.75

รูปที่ 3.6 แสดง หน้าเมนูของ Thai Palace

เมื่อคลิกเลือก Nutty Palaece จะปรากฏ หน้าเมนูดังนี้

Nutty Palace - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites Go Links Norton Antivirus

Address <http://127.0.0.1:8080/sd3/SiamDeliveryServer?actionName=menu>

Nutty Palace : (900) 715458 (FAX)

Fax this menu back with your selections circled. If you want more than one item, indicate the number of items in the number column. Your order will be ready 30 minutes from when we receive the fax.

appetizers

Number	Veggie	Spicy	Name	Description	Price
1	Yes	No	Asian appetizer	This selection of authentic Asian foods includes 20 each of vegetable gyoza, seafood shumai, sesame chicken, vegetable spring rolls and egg rolls.	\$ 1.75
2	Yes	No	Kosher egg	Serve egg rolls with an Oriental style dipping sauce of your choice.	\$ 2.25

soups

Number	Veggie	Spicy	Name	Description	Price
--------	--------	-------	------	-------------	-------

Done Internet

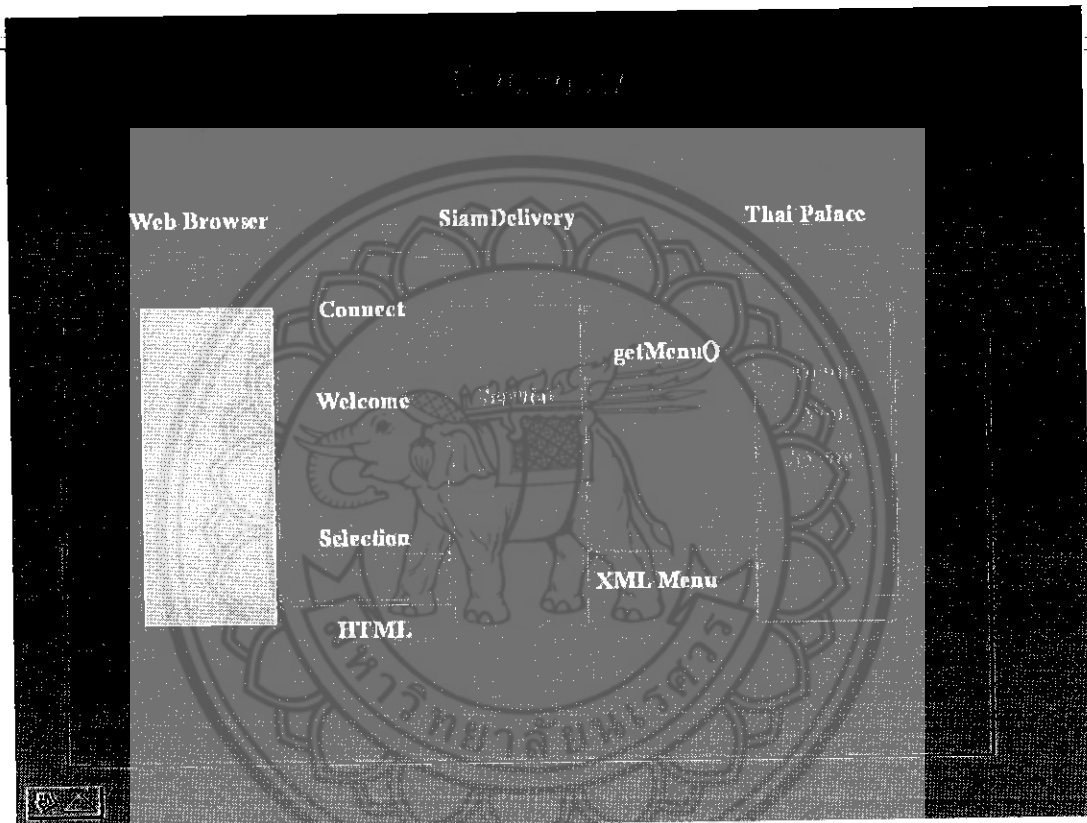
start [Taskbar icons] 0:10

รูปที่ 3.7 แสดง หน้าเมนูของ Nutty Palace

บทที่ 4

ผลการทดลอง

4.1 สรุปขั้นตอนการทำงานของเซอร์วิสเมนูอาหาร



รูปที่ 4.1 สรุปขั้นตอนการทำงานของเว็บเซอร์วิส

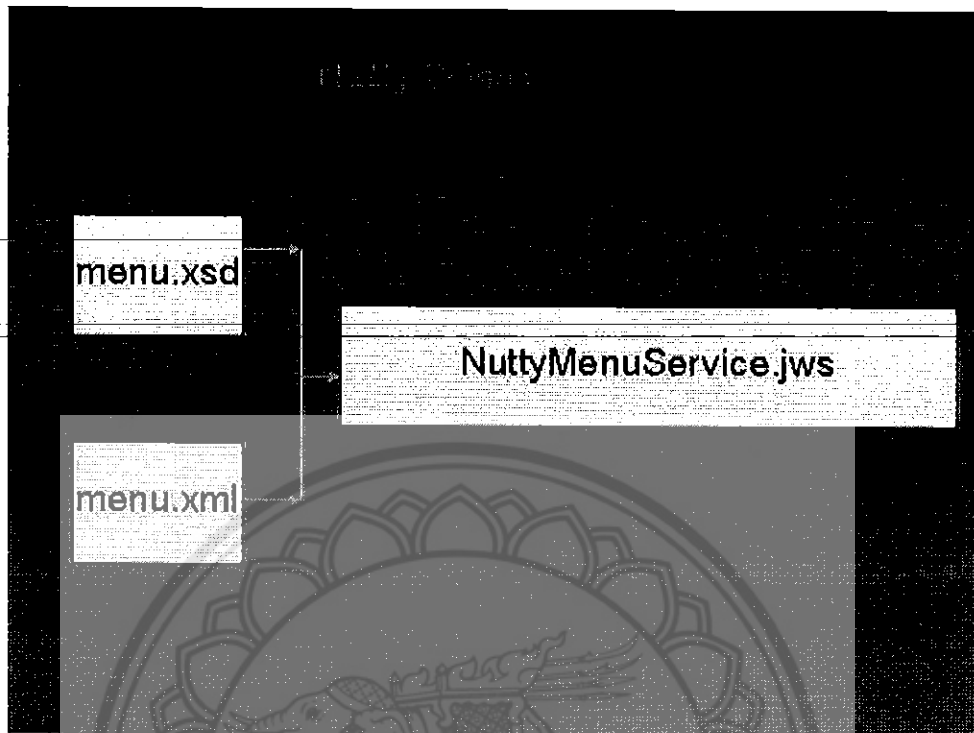
1. เมื่อลูกค้าเข้ามาที่ web browser แล้ว connect ไปยัง SiamDelivery ก็จะแสดงหน้า HTML ของต้อนรับสู่ SiamDelivery พร้อมกับมีเมนูรายชื่อร้านอาหารให้เลือก
2. เมื่อเราเลือกชื่อร้านแล้วกด Submit Server ของ SiamDelivery ก็จะร้องขอรายการอาหารไปยัง Menu Web Service ซึ่งก็คือ Thai Palace หรือ Nutty Palace
3. เมื่อ Menu Web Service ซึ่งก็คือ Thai Palace หรือ Nutty Palace ได้รับคำร้องขอมาก็จะส่งเมนูอาหารที่อยู่ในรูปแบบภาษา XML ออกไปให้ทาง SiamDelivery
4. เมื่อ SiamDelivery รับเมนูอาหารที่อยู่ในรูปแบบภาษา XML มา ก็จะแปลงโดย กระบวนการของ XSL ให้ออกมาอยู่ในรูป HTML Menu แล้วส่งไปแสดงทาง web browser ต่อไป

4.2 อธิบายแต่ละส่วนย่อย ๆ พอสังเขป

- Web Service Menu

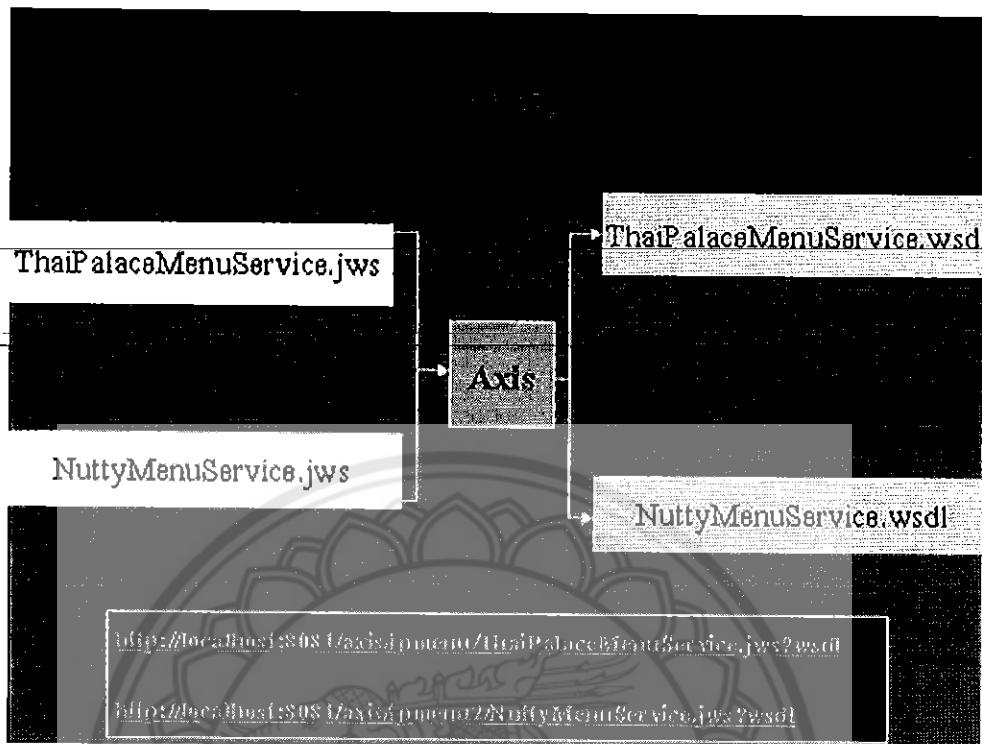


รูปที่ 4.2 Thai Palace



รูปที่ 4.3 Nutty Palace

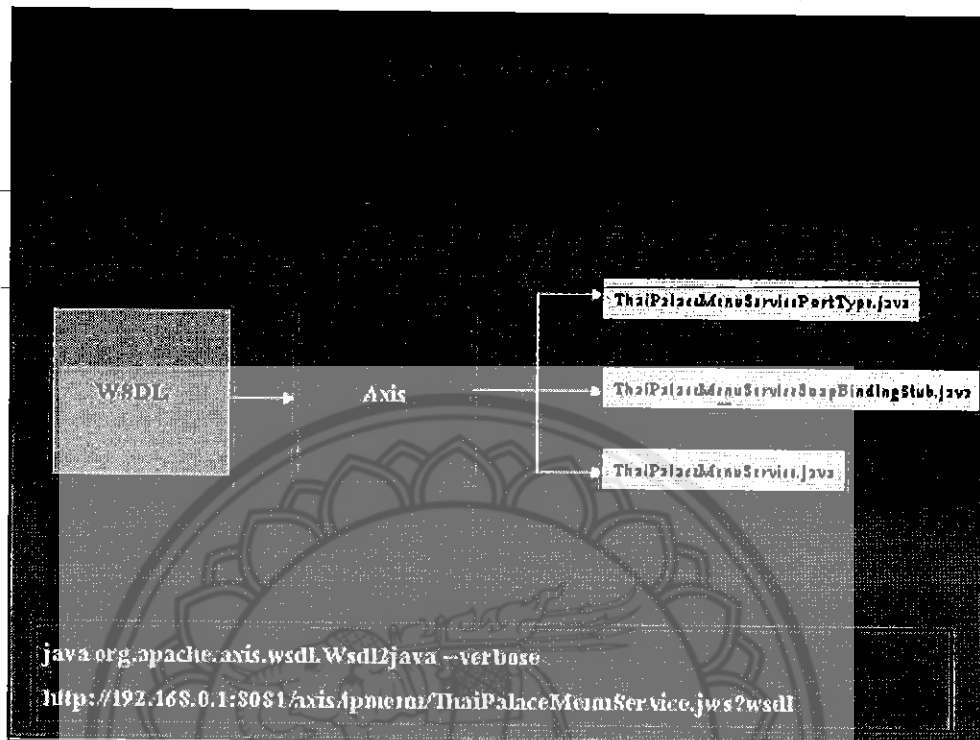
จากรูปข้างบนทั้ง Thai Palace และ Nutty Palace ซึ่งเป็น Web Service Menu จะสร้างไฟล์จาวาเว็บเซอร์วิส (JWS) คือ ThaiPalaceMenuService.jws กับ NuttyMenuService.jws โดยภายในจะอ้างถึงไฟล์ menu.xsd และ menu.xml พร้อมกับอ้าง path ในการเข้าถึงข้อมูล source code แสดงไว้แล้วอย่างละเอียดในบทที่ 3



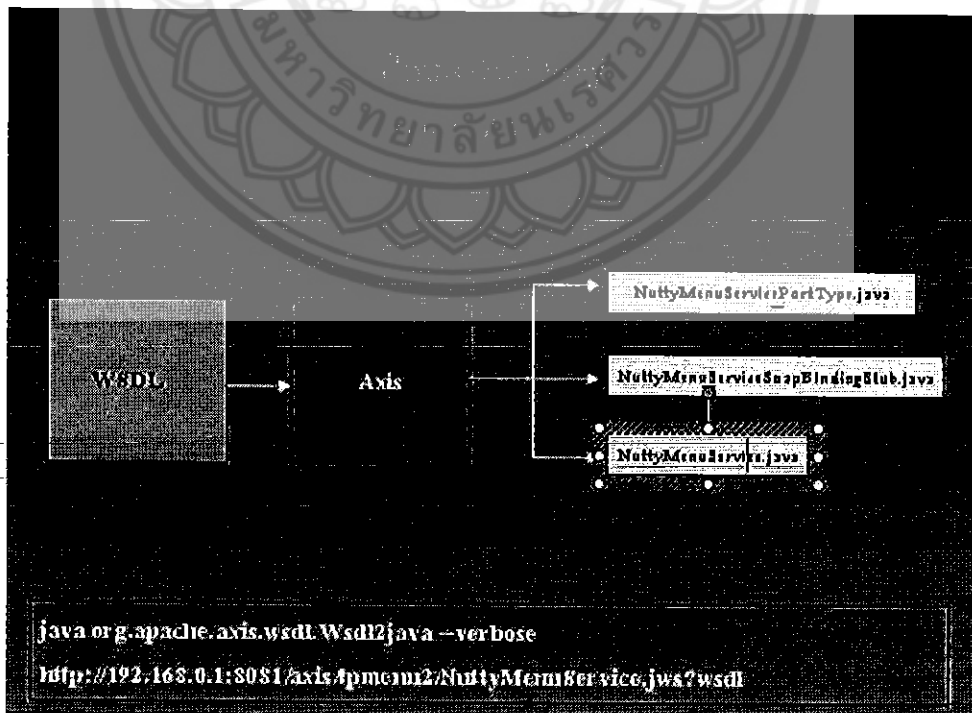
รูปที่ 4.4 การทำงานของ Axis Tool

จากรูปเป็นการแปลงไฟล์ `ThaiPalaceMenuService.jws` และ `NuttyMenuService.jws` โดย Axis Tool เพื่อให้ได้ไฟล์ `ThaiPalaceMenuService.wsdl` และ `NuttyMenuService.wsdl` ตามลำดับ คำสั่งที่ใช้คือ `http://localhost:8081/axis/tpmenu/ThaiPalaceMenuService.jws?wsdl` และ `http://localhost:8081/axis/tpmenu2/NuttyMenuService.jws?wsdl` ตามลำดับ

- **SiamDelivery**



รูปที่ 4.5 การเชื่อมต่อไฟล์ ThaiPalaceMenuService.wsdl ไปยัง Web Service Menu



รูปที่รูปที่ 4.6 การเชื่อมต่อไฟล์ NuttyMenuService.wsdl ไปยัง Web Service Menu

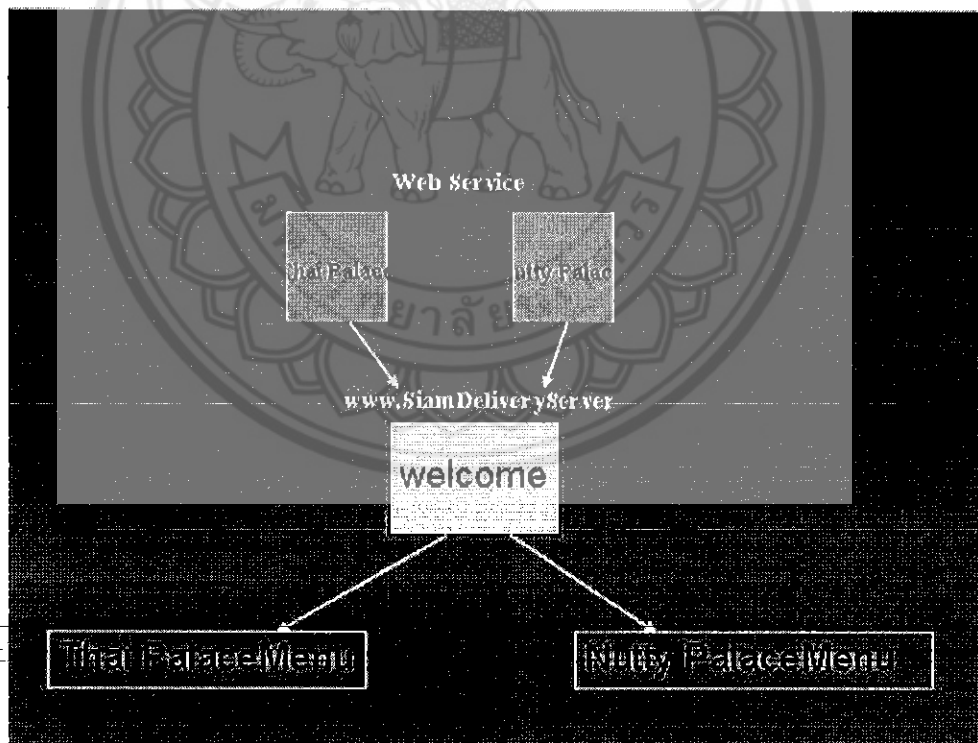
ทั้ง 2 รูปข้างบนเป็นการนำ ไฟล์ WSDL คือ ThaiPalaceMenuService.wsdl และ NuttyMenuService.wsdl มาเข้า Axis Tool เพื่อ สร้างตัวเชื่อมต่อ ไปยัง Web Service Menu โดยจะ Generate ออกมาให้ 3 ไฟล์ดังนี้สำหรับ Thai Palace

1. ThaiPalaceMenuServicePortType.java
2. ThaiPalaceMenuServiceSoapBindingStub.java
3. ThaiPalaceMenuService.java

และอีก 3 ไฟล์สำหรับ Nutty Palace

1. NuttyMenuServicePortType.java
2. NuttyMenuServiceSoapBindingStub.java
3. NuttyMenuService.java

รายละเอียดศึกษาได้จากบทที่ 3



รูปที่ 4.7 การแสดงไฟล์ html ทางบราวเซอร์

SiamDelivery จะสร้างไฟล์ SiamDeliveryServe.java เป็นตัวควบคุมการทำงานทั้งหมด เมื่อเรา connect เข้ามาที่ www.SiamDeliveryServer (<http://127.0.0.1:8080/sd3/SiamDeliveryServer>) หน้าแรกที่เราจะเห็นคือ หน้า welcome เมื่อเราคลิกเลือกชื่อร้านที่เราต้องการมันก็จะแสดงหน้าเมนูอาหารของร้านนั้น ๆ มาแสดง



บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

องค์ประกอบของเว็บเซอร์วิสมีอยู่ด้วยกันหลายส่วนแต่ที่สำคัญ ได้แก่

- XML ซึ่งเป็นภาษามาตรฐานในการพัฒนาเว็บเซอร์วิสเพราะสามารถทำงานได้ในทุก plat form ที่แตกต่างกัน
- XSD เป็นภาษาที่กำหนดโครงสร้างไวยากรณ์ของเอกสาร XML
- XSL เป็นภาษาที่ใช้แปลงเอกสาร XML ให้เป็นเอกสาร HTML
- SOAP คือ โพรโตคอลที่ใช้สื่อสารข้อมูลเว็บเซอร์วิส
- WSDL คือ เอกสารที่ใช้อธิบายการทำงานของเว็บเซอร์วิส
- UDDI คือ แหล่งรวบรวมบริการเว็บเซอร์วิสซึ่งเปรียบเสมือนกับ Search engine

โดยเว็บเซอร์วิสที่พัฒนาขึ้นคือ บริการแลกเปลี่ยนไฟล์เมนูอาหารระหว่างภัตตาคาร 2 แห่ง เพื่อเป็นการอำนวยความสะดวกให้กับผู้จัดการร้านอาหารในการกำหนดเมนูที่จะให้บริการแก่ลูกค้า เครื่องมือที่ใช้พัฒนาคือ Apache Axis Tools ซึ่งเว็บเซอร์วิสนี้อยู่ในรูปของการพาณิชย์อิเล็กทรอนิกส์ประเภทองค์กรธุรกิจด้วยกัน (B2B-Business to Business)

5.2 ปัญหาในการทดลอง

- เทคโนโลยีนี้เป็นเรื่องใหม่ในประเทศไทย เอกสารอ้างอิงหาได้ยาก จึงเสียเวลากับการอ่าน textbook เพื่อทำความเข้าใจในหลักการอยู่นาน

- เว็บเซอร์วิสที่พัฒนาขึ้นยังไม่สามารถนำไปใช้ในเชิงพาณิชย์ได้ เพราะเป็นเพียงโมเดลจำลองเพื่อให้เห็นใจหลักการพัฒนาเท่านั้น

5.3 ข้อเสนอแนะ

- ควรมีการพัฒนาต่อให้สามารถใช้งานได้จริงในทางธุรกิจ แต่เนื่องจากเป็นองค์ความรู้ใหม่ที่เพิ่งเกิดขึ้นได้ไม่นาน ดังนั้นผู้พัฒนาควรให้เวลาในการศึกษาพัฒนาแต่นั้น ๆ

- ถ้าจัดให้มีการเรียนการสอนเรื่องเว็บเซอร์วิสในหลักสูตรได้ ก็จะเป็นการดียิ่ง เพราะแนวโน้มเทคโนโลยีนี้จะเข้ามาเป็นส่วนสำคัญในการดำเนินธุรกิจในประเทศไทยอย่างแน่นอน



เอกสารอ้างอิง

- [1] ฉัตรชัย สุขสะอาด, “Web Service เครื่องมือธุรกิจยุคใหม่” [Online]. Available :
<http://www.wsiam.com/document/abcwebservices/webservicesabc.jsp#benifit> 2005
- [2] อานัฐ “ปี 48 เว็บเซอร์วิสมาแรง รัฐและเอกชนหนุนเต็มที่.” Telecom journal-หนังสือพิมพ์ประจำ
 วงการสื่อสารโทรคมนาคม. ฉบับที่ 534, วันที่ 11 ตุลาคม, 2547
- [3] The Apache Software Foundation. “ Web Services – Axis.” [Online]. Available :
<http://ws.apache.org/axis/> Copyright © 2000-2005.
- [4] EXBIZ (THAILAND) Co., Ltd. [Online]. Available : <http://www.wsiam.com/>. Copyright 2003
- [5] Pongsakorn Poosankam, Chaiwat Bootchai. “ web service group : ชุมชนเว็บเซอร์วิส.” [Online].
 Available : <http://campus.en.kku.ac.th/webservices/> 2005
- [6] Web Services Solutions Ltd. [Online]. Available : <http://www.webservices.org/> Copyright 2001 -
 2004
- [7] กมลวัฒน์ สุพรรณพัฒน์, กิตติไกร ลีละพันธ์ “การพัฒนาโปรแกรมสำหรับ B2B E-Marketplace ด้วย
 XML” วิทยานิพนธ์ตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, ปีการศึกษา 2544

ภาคผนวก

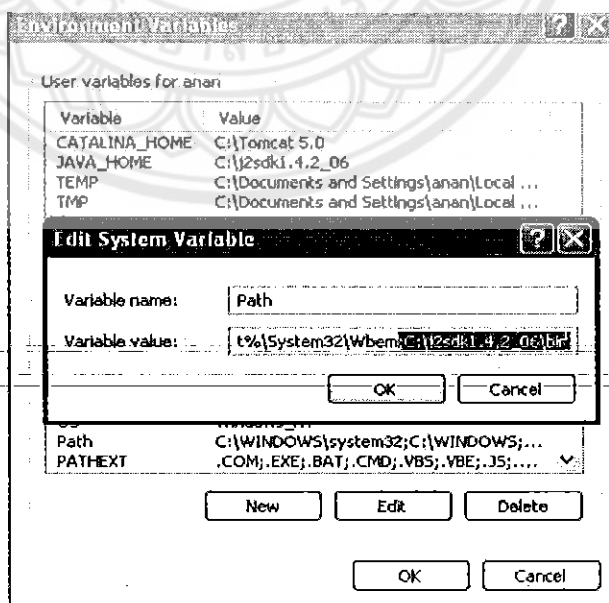
การติดตั้งซอฟต์แวร์

ซอฟต์แวร์ที่ใช้ในการพัฒนาโครงการ ได้แก่

- j2sdk1.4.2_06
- Tomcat Server (Version 5.0)
- Program Editor
- Apache Axis Tools (Axis-1_0)

1. Java 2 SDK Standard edition (เรียกย่อว่า J2SE) ซึ่งเป็นคอมไพเลอร์ของภาษาจาวา (Java) ที่จะต้องใช้ในการคอมไพล์โค้ดของ JSP และ JAVA สามารถดาวน์โหลดได้ที่ <http://java.sun.com/j2ec/> สำหรับในโครงการนี้ใช้เวอร์ชัน 1.4 ได้แก่ไฟล์ j2sdk1.4.2_06-windows-i586.exe หลังจาก Install เรียบร้อยแล้วให้ทำการเซตค่าดังนี้ ไปที่หน้าต่าง System Properties ให้คลิกที่แถบ Advanced แล้วคลิก Enviroment Variables ที่ช่อง User Variables ให้สร้าง Variables name = JAVA_HOME สร้าง Variables value = C:\j2sdk1.4.2_06 (ที่ j2sdk ติดตั้งอยู่)

ที่ช่อง System Variables ให้เพิ่มค่า path ดังนี้ ;C:\j2sdk1.4.2_06\bin ดังรูป



2. ติดตั้งเว็บเซิร์ฟเวอร์ในโครงการนี้ใช้ Tomcat 5.0 หลังจากติดตั้งเรียบร้อยแล้วให้ทำการ Set Path ที่ Enviroment Variables ที่ช่อง User Variables ให้สร้าง Variables name = CATALINA_HOME สร้าง Variables value = C:\tomcat 5.0 (ที่ Tomcat ติดตั้งอยู่)

3. โปรแกรมเมอร์ทั่วไป เช่น EditPlus,eclipse เพื่อใช้ในการเขียนโค้ด JSP, JAVA

4. ติดตั้ง Axis ใน Tomcat เมื่อได้ไฟล์ Axis-1_0.zip แล้วให้ unzip Axis-1_0 ไว้ที่ directory c:\axis-1_0 จากนั้น copy folder “%AXIS%\webapps\axis” ไปใส่ใน “%TOMCAT%\webapps\” หลังจากนั้นให้ download ไฟล์ xerces.jar ซึ่งเป็นตัวคอมไพล์ XML ไว้ที่ lib ของ AXIS ที่ copy ไว้ใน “%TOMCAT%\webapps\” ตรวจสอบว่าใน lib ของ AXIS ว่ามี ไฟล์ครบ 5 ไฟล์ดังนี้

- Axis.jar
- Clutil.jar
- Log4j-core.jar
- Wsdl4j.jar
- Xerces.jar



ประวัติผู้เขียนโครงการ

นาย นราศักดิ์ ภูผายาง

ภูมิลำเนา บ้านเลขที่ 63 หมู่ 3 ต.ดงสมบูรณ์ อ.ท่าคันโท จ.กาฬสินธุ์ 46190

ประวัติการศึกษา

ระดับมัธยมศึกษา การศึกษานอกโรงเรียนท่าคันโท (เรียนควบคู่กับการศึกษาในระบบ)

ระดับมัธยมศึกษา โรงเรียนท่าคันโทวิทยาคาร จังหวัดกาฬสินธุ์

ขณะนี้กำลังศึกษาอยู่ที่ คณะวิศวกรรมศาสตร์ สาขาวิศวกรรมคอมพิวเตอร์ ชั้นปีที่ 4 มหาวิทยาลัยนเรศวร

นาย อนันต์ ภูถนงรัตน์แดง

ภูมิลำเนา บ้านเลขที่ 81 หมู่ 9 ต.หนองทอง อ.โพนทราย จ.กำแพงเพชร 62150

ประวัติการศึกษา

ระดับมัธยมศึกษา โรงเรียนกำแพงเพชรพิทยาคม

ระดับมัธยมศึกษา โรงเรียนกำแพงเพชรพิทยาคม

ขณะนี้กำลังศึกษาอยู่ที่ คณะวิศวกรรมศาสตร์ สาขาวิศวกรรมคอมพิวเตอร์ ชั้นปีที่ 4 มหาวิทยาลัยนเรศวร