

การพัฒนาเว็บเซอร์วิสบนระบบฝังตัว
Embedded Web Service Development



นายสุวิทย์ มัชคาม
นายเอกราช อินนอก

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2545

เลขหมู่.....
เลขทะเบียน.....49980
วัน,เดือน,ปี 16 ส.ย. 2547

b.....
i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาเว็บเซอร์วิสบนระบบฝังตัว
Embedded Web Service Development

โดย

นายสุวิทย์ มัชคัม

นายเอกราช อินนอก

อาจารย์ที่ปรึกษา

ศต. อภินทร อุณาอูถ

อ. ดุลิต นิยะโต

ปริญญาโทฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท ปีการศึกษา 2545

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

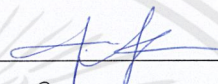
เรื่อง การพัฒนาเว็บเซอร์วิสบนระบบฝังตัว

Embedded Web Service Development

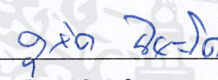
ผู้จัดทำ

1. นาย สุวิทย์ มัชฌิม รหัสนักศึกษา 42010411
2. นาย เอกธัช อินนอก รหัสนักศึกษา 42010452





(ผศ. อภินันท์ อุนากุล)



(อ. คุณิต นิชะโต)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาเว็บเซอร์วิสบนระบบฝังตัว

นาย สุวิทย์ มัชฌิม 42010411

นาย เอกกราช อินนอก 42010452

ผศ. อภิเนตร อุณาภูล อาจารย์ที่ปรึกษา

อ. คุณิต นิยะโต อาจารย์ที่ปรึกษา

ปีการศึกษา 2545

บทคัดย่อ

ในปัจจุบันการสื่อสารระหว่างแอปพลิเคชันผ่านออบเจกต์ระยะไกล (Remote Procedure Call) ได้รับการพัฒนาขึ้นด้วยเทคโนโลยีเว็บเซอร์วิส (Web Service) ที่นำเสนอแนวคิดในกำหนดมาตรฐานการเชื่อมต่อระหว่างแอปพลิเคชัน เพื่อให้ระบบที่มีความแตกต่างกันสามารถติดต่อสื่อสารถึงกันได้ โดยอาศัยอาศัยอาศัยข้อดีของโปรโตคอลเอชทีทีพี (HTTP Protocol) ที่มีความนิยมใช้กันอย่างแพร่หลาย และกลไกในการแลกเปลี่ยนข้อมูลผ่านทาง SOAP (Simple Object Access Protocol) ที่ใช้โครงสร้างของ XML Language ที่ใช้ในการนิยามความหมายของข้อมูล

การนำเอาเว็บเซอร์วิสมาพัฒนาบนระบบฝังตัวจะช่วยให้อุปกรณ์มีความสามารถในการเชื่อมต่อกับอุปกรณ์ต่าง ๆ และสามารถในการเชื่อมโยงข้อมูลที่ใช้ในควบคุมอุปกรณ์ได้อย่างมีประสิทธิภาพสูงสุด โครงการนี้จึงได้นำเสนอซอฟต์แวร์บนระบบฝังตัวที่ช่วยในการจัดการเอกสารที่มีรูปแบบตามมาตรฐานซิมเปิล ออบเจกต์ โปรโตคอล และเป็นเซิร์ฟเวอร์ที่ให้บริการเซอร์วิสกับอุปกรณ์ต่างๆได้

วิทยานิพนธ์ฉบับนี้จัดทำขึ้นเพื่อวัตถุประสงค์ในการนำเสนอแนวความคิด ทฤษฎีและหลักการในการสร้างและพัฒนาเว็บเซอร์วิสบนระบบฝังตัวที่สามารถเป็นเซิร์ฟเวอร์ที่ให้บริการเซอร์วิส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Embedded Web Service Development

Suwit	Madchakham	42010411
Akekarach Innok		42010452
Assistant Professor Apinetr	Unagul	Advisor
Dusit	Niyato	Advisor

ABSTRACT

Presently, application communicated via Remote Procedure Call is developed with Web Service Technology, the concept is to define standard communication among applications, that have different technologies are accessible. Web Service uses the benefit of ubiquitous HTTP Protocol that is simple and lightweight mechanism for exchanging structured and type information among peers in a decentralized, distributed environment using SOAP, based on XML

Embedded Web Service Development will enhance connection with equipment and link data for distributed environment. This project presents the software that manage SOAP Message and responds the request SOAP Message as Server

The objective of this project is to present the idea, the theory, and the concept of developing “Embedded Web Service Development” which is the software system that provide service to equipment of the network

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และความร่วมมือจากหลายๆ ฝ่ายด้วยกัน บุคคล 2 ท่านแรกที่ต้องกล่าวถึง เพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือ ผศ. อภินันท์ อุณาภูล และ อาจารย์ ดุสิต นิยะโต ซึ่งท่านทั้งสองเป็นอาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และความช่วยเหลือเสมอมา นอกจากนั้นยังได้ให้ความรู้และแนวคิดในการทำงานรวมถึงการใช้ชีวิต อันมีค่ายิ่งให้แก่ผู้เขียน ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

ขอขอบคุณที่ ๆ เพื่อนๆ ห้อง ESL และเพื่อนๆ ในห้องโปรเจกต์อื่นๆ ห้อง ที่ช่วยเป็นกำลังใจและให้ประสบการณ์ที่ประทับใจต่อการใช้ชีวิตในมหาวิทยาลัย

ขอขอบคุณบริษัทเน็ตเกตเจ็ตที่เอื้อเฟื้อบอร์ดคอม86 ขอขอบคุณที่เต่าที่สนับสนุนทุกสิ่งอย่างเต็มใจ ขอขอบคุณที่โอที่เอื้อเฟื้อซอฟต์แวร์ ขอขอบใจอึ้งกับหมี่ที่ช่วยพิมพ์งาน ขอใจเป็นที่ช่วยให้คำแนะนำอย่างเต็มใจเสมอมา ขอขอบใจก๊อที่คอยมาดูแลคู่โปรเจกต์ผม

สุดท้ายนี้ ต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา รวมทั้งญาติพี่น้องอันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจเอาใจใส่เสมอมา ในทุก ๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

นาย สุวิทย์ มัชฌิม

นาย เอกราช อินนอก

สารบัญ

หน้าที่

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VIII
สารบัญรูปภาพ.....	IX
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญและที่มา.....	1
1.2 วัตถุประสงค์ของงานวิจัย.....	2
1.3 ขอบเขตของงานวิจัย.....	2
1.4 เป้าหมายของโครงการ.....	3
1.5 วิธีการดำเนินงาน.....	3
บทที่ 2 ทฤษฎีและหลักการ.....	5
2.1 เว็บเซอร์วิส (Web Service).....	5
2.1.1 การพัฒนาของเว็บ.....	5
2.1.1.1 สเตติกเว็บเพจ (Static Web Page).....	5
2.1.1.2 ไดนามิกเว็บเพจ (Dynamic Web Page).....	5
2.1.1.2 บริการบนเว็บ (Web Service).....	6
2.1.2 เว็บเซอร์วิส.....	7
2.1.3 ลักษณะของเว็บเซอร์วิส.....	8
2.1.4 เทคโนโลยีพื้นฐานของเว็บเซอร์วิส.....	9
2.2 เอ็กซ์เอ็มแอล (XML).....	10
2.2.1 ข้อจำกัดของเอชทีเอ็มแอล.....	10
2.2.2 ลักษณะที่สำคัญของภาษาเอ็กซ์เอ็มแอล.....	11
2.2.3 ประโยชน์ของเอ็กซ์เอ็มแอล.....	12
2.2.4 ความแตกต่างระหว่างเอชทีเอ็มแอลและเอ็กซ์เอ็มแอล.....	13
2.2.5 โครงสร้างเอ็กซ์เอ็มแอล.....	14
2.2.6 ข้อควรระวังในการเขียนเอ็กซ์เอ็มแอล.....	15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.7	การกำหนดโครงสร้างของเอกสารเ็กซ์เอ็มแอล.....	16
2.2.8	ข้อดีและข้อเสียของดีทีดีและเ็กซ์เอ็มแอลสคริปต์.....	17
2.2.9	การแสดงผลเอกสารเ็กซ์เอ็มแอลด้วยเ็กซ์เอสแอล.....	18
2.2.10	การแปลความหมายของเ็กซ์เอ็มแอลด้วยเ็กซ์เอ็มแอลพาร์เซอร์.....	19
2.2.11	การแปลเอกสารโดยใช้ค็อกกุเมนที่อบเจ็กต์โมเดลลิง.....	20
2.2.12	การแปลเอกสารโดยใช้ซิมเปิ้ลเอพีไอฟอร์เ็กซ์เอ็มแอล.....	21
2.3	ซิมเปิ้ลอบเจ็กต์แอกเซสโพรโทคอล (Simple Object Access Protocol).....	22
2.3.1	ส่วนประกอบของโซบ.....	23
2.3.1.1	เอนวิลอป (Envelope).....	26
2.3.1.2	บอดี (Body).....	26
2.3.1.3	เฮดเดอร์ (Header).....	27
2.3.1.3.1	โซบแอกเคอร์แอททริบิวต์.....	27
2.3.1.3.2	โซบมีสแอกเคอร์สแตนด์แอททริบิวต์.....	27
2.3.2	โซบพอลเอลเลอเมนต์.....	28
2.2.3	โซบเอนโด้คั้ง.....	29
2.3.3.1	สตรัก.....	30
2.3.3.2	อะเรย์.....	32
2.3.4	โซบในแอชทีที.....	34
2.3.5	โซบสำหรับอาร์พีซี.....	35
2.3.5.1	ยูอาร์ไอของอบเจ็กต์ปลายทาง (Target Object).....	35
2.3.6	โซบทูลคิต (SOAP Toolkit).....	36
2.3.7	โครงสร้างภายในของโซบทูลคิตของไมโครซอฟต์.....	37
2.3.7.1	การสร้างโซบไคลเอนต์อบเจ็กต์ (SoapClient Object).....	37
2.3.7.2	การประมวลผลภายในโซบไคลเอนต์อบเจ็กต์.....	38
2.3.7.3	การสร้างโซบเซิร์ฟเวอร์อบเจ็กต์ (SoapServer Object).....	39
2.3.7.4	การทำงานทางฝั่งเซิร์ฟเวอร์.....	39
2.3.7.5	การประมวลผลภายในโซบเซิร์ฟเวอร์อบเจ็กต์.....	40
2.4	ดับบิวเอสดีแอล (WSDL).....	41
2.4.1	ดับบิวเอสดีแอลคืออะไร.....	41
2.4.2	ความสัมพันธ์ระหว่างดับบิวเอสดีแอลกับเว็บเซอร์วิส.....	41
2.4.3	โครงสร้างของดับบิวเอสดีแอล.....	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3.1	แอ็บสแตร็กต์เดฟิเนชันกรุป (Abstract Definitions Group).....	42
2.4.3.2	คอนกรีตเดสคริปต์ชันกรุป (Concrete Descriptions Group).....	42
2.5	ยูดีดีไอ (UDDI).....	48
2.5.1	ยูดีดีไอคืออะไร.....	48
2.5.2	โครงสร้างของยูดีดีไอ.....	48
2.5.3	ประเภทของเอกสารดับบิวเอสดีแอล.....	49
2.5.4	การประกาศและค้นหาเอกสารดับบิวเอสดีแอลในยูดีดีไอ.....	50
2.6	คอม86 (COM86).....	52
2.6.1	จุดเด่นของคอม86.....	51
2.6.1.1	ใช้ไมโครคอนโทรลเลอร์ที่มีความสามารถสูง.....	51
2.6.1.2	มีเครื่องมือในการพัฒนาโปรแกรมมาก.....	52
2.6.1.3	ง่ายในการเรียนรู้และพัฒนา.....	52
2.6.1.4	มีความสามารถในการเพิ่มขยายได้.....	53
2.6.2	แอปพลิเคชันเป้าหมายของชุดพัฒนา (Target Application).....	54
2.6.2.1	เกี่ยวกับอุตสาหกรรม.....	54
2.6.2.2	อุปกรณ์ยูเอสบี (USB Peripheral).....	54
2.6.2.3	เว็บเซิร์ฟเวอร์ (Embedded web server).....	54
2.6.2.4	โรโบติก (Robotics).....	54
2.6.2.5	การควบคุมบ้านหรือสิ่งก่อสร้าง.....	54
2.6.3	คุณสมบัติของบอร์คคอม86.....	55
บทที่ 3	การออกแบบพัฒนาเว็บเซอร์วิสบนระบบฝังตัว.....	58
3.1	ขอบเขตของโครงการ.....	58
3.2	โครงสร้างของเว็บเซอร์วิสบนระบบฝังตัว.....	58
3.3	การตอบสนองของระบบ.....	60
3.4	ยูสเคสไดอะแกรม (Use Case Diagram).....	61
3.5	คอมโพเนนต์ไดอะแกรม (Component Diagram).....	64
3.6	คลาสไดอะแกรม (Class Diagram).....	65
3.7	ซีควเอนส์ไดอะแกรม (Sequence Diagram).....	71
3.7.1	การประกาศเว็บเซอร์วิส (Publish Web Service).....	71
3.7.2	การค้นหาเว็บเซอร์วิส (Find Web Service).....	72
3.7.3	การร้องขอเอกสารดับบิวเอสดีแอล (Retrieve WSDL).....	72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

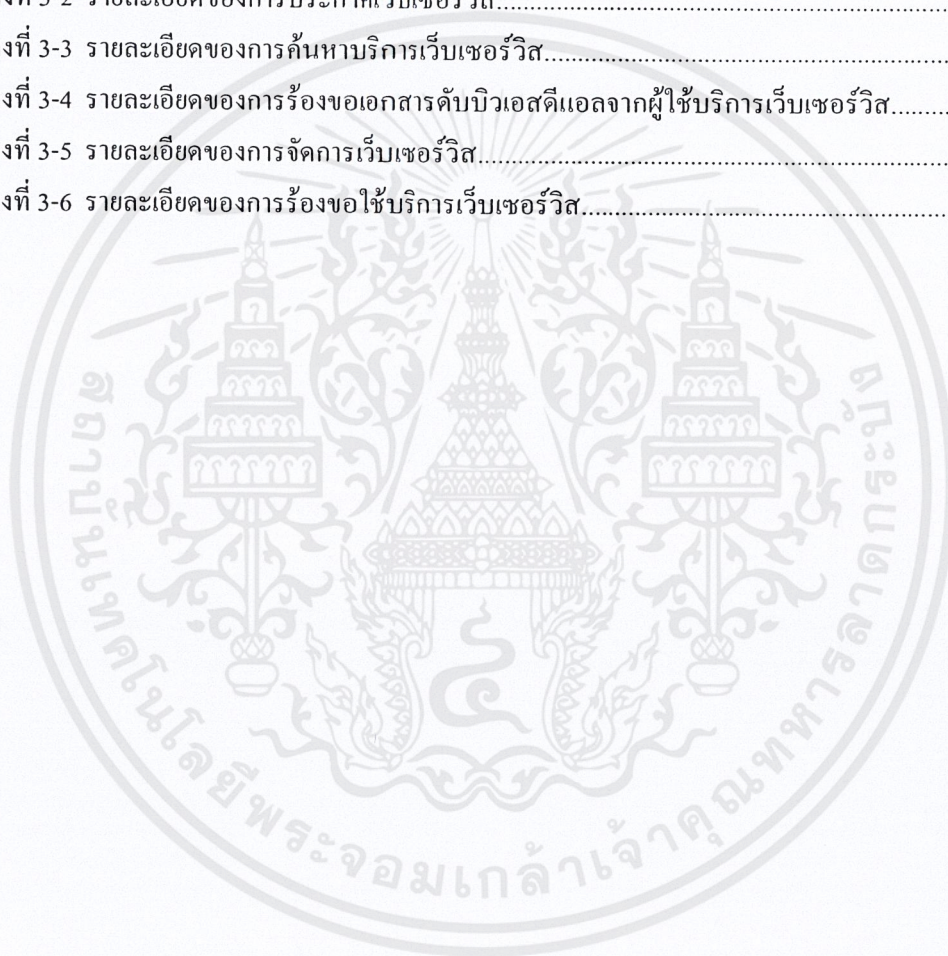
3.7.4	การจัดการเว็บเซอร์วิส (Manage Web Service).....	73
3.7.4.1	การดีพลอยเว็บเซอร์วิส (Deploy Web Service).....	73
3.7.4.2	การอันดีพลอยเว็บเซอร์วิส (Undeploy Web Service).....	74
3.7.4.3	การตรวจสอบ (Monitor Available Space).....	75
3.7.4.4	การตรวจสอบล็อกไฟล์ (Monitor logfile).....	75
3.7.4.5	การเรียกใช้เว็บเซอร์วิส (Call Web Service).....	76
3.8	สเตตชาร์ตไดอะแกรม (Statechart Diagram).....	79
3.9	ดีพลอยเมนต์ไดอะแกรม (Deployment Diagram).....	80
บทที่ 4	การพัฒนาการให้บริการเว็บเซอร์วิสบนระบบฝังตัว.....	81
4.1	นักพัฒนา.....	81
4.2	ผู้ที่สร้างและดูแลการให้บริการเว็บเซอร์วิส.....	82
4.3	ผู้ที่สร้างเซอร์วิสบนระบบฝังตัว.....	84
4.3.1	การพัฒนาแอปพลิเคชันเซอร์วิส.....	84
4.3.2	การสร้างเอกสารอธิบายรายละเอียดของเซอร์วิสสำหรับเว็บเซิร์ฟเวอร์.....	84
4.3.3	การสร้างเอกสารอธิบายรายละเอียดของเว็บเซอร์วิส (WSDL Document).....	86
4.4	ผู้ที่เรียกใช้บริการเซอร์วิสจากระบบฝังตัว.....	89
บทที่ 5	การทดสอบและวิเคราะห์ผล.....	90
5.1	วัตถุประสงค์ของการทดสอบ.....	90
5.2	วิธีการทดสอบ.....	90
5.3	ผลการทดสอบและการวิเคราะห์ผลการทดสอบ.....	91
5.3.1	การทดสอบการให้บริการทางฝั่งเซิร์ฟเวอร์.....	91
บทที่ 6	บทวิจารณ์และสรุป.....	95
6.1	บทวิจารณ์และสรุป.....	96
6.2	แนวทางในการพัฒนาต่อ.....	96
ภาคผนวก	97
บรรณานุกรม	102

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้าที่

ตารางที่ 2-1 ตัวอย่างของเอลเลอเมนต์ย่อยๆ.....	29
ตารางที่ 2-2 ตัวอย่างของค่าพอลโก๊ด.....	29
ตารางที่ 2-3 ตัวอย่างของโชนบูลคิด.....	36
ตารางที่ 3-1 รายการเหตุการณ์ที่เกิดขึ้นกับระบบ.....	60
ตารางที่ 3-2 รายละเอียดของการประกาศเว็บเซอร์วิส.....	62
ตารางที่ 3-3 รายละเอียดของการค้นหาบริการเว็บเซอร์วิส.....	62
ตารางที่ 3-4 รายละเอียดของการร้องขอเอกสารดับบิวเอสดีแอลจากผู้ให้บริการเว็บเซอร์วิส.....	62
ตารางที่ 3-5 รายละเอียดของการจัดการเว็บเซอร์วิส.....	63
ตารางที่ 3-6 รายละเอียดของการร้องขอใช้บริการเว็บเซอร์วิส.....	63



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

หน้าที่

รูปที่ 2-1 ลักษณะของเว็บเซอร์วิส.....	6
รูปที่ 2-2 แสดงความสัมพันธ์ของผู้ให้บริการ ผู้ขอใช้บริการ และตัวแทนผู้ให้บริการ.....	9
รูปที่ 2-3 ส่วนประกอบของเอกสารอิเล็กทรอนิกส์.....	14
รูปที่ 2-4 อักษรชุดพิเศษ และ เอนทิตีรีเฟอเรนส์.....	16
รูปที่ 2-5 ตัวอย่าง SI.dtd.....	17
รูปที่ 2-6 ตัวอย่างไฟล์ slstyle.xml.....	18
รูปที่ 2-7 โครงสร้างของโซบ.....	24
รูปที่ 2-8 ตัวอย่างของโซบรีเคเวส.....	24
รูปที่ 2-9 ตัวอย่างของโซบเรสสปอน.....	25
รูปที่ 2-10 เอกซ์เอ็มแอลเอนเอสเอททริบิว.....	26
รูปที่ 2-11 เอกซ์เอ็มแอลเอนเอสเอททริบิวที่ไม่ขึ้นด้วยเนมสเปซ.....	26
รูปที่ 2-12 ตัวอย่างของโซบเฮคเคอร์.....	27
รูปที่ 2-13 ตัวอย่างของโซบเมสเซจที่มีความผิดพลาด.....	28
รูปที่ 2-14 ตัวอย่างของการใช้เอกซ์เอสแอล.....	30
รูปที่ 2-15 ตัวอย่างของสตรักเจอร์.....	30
รูปที่ 2-16 ตัวอย่างของโซบเมสเซจที่ใช้เอกซ์เอสไอ.....	31
รูปที่ 2-17 ตัวอย่างของโซบเมสเซจที่ใช้ดับเบิลเอนตี้.....	32
รูปที่ 2-18 ตัวอย่างของอะเรย์ของสตรักเจอร์ที่ใช้เอกซ์เอสไอ.....	32
รูปที่ 2-19 ตัวอย่างของอะเรย์ของสตรักเจอร์ที่ใช้ดับเบิลเอนตี้.....	33
รูปที่ 2-20 ตัวอย่างของเรสสปอนที่มีสแตตัสโค้ดเป็น 500.....	34
รูปที่ 2-21 ตัวอย่างของเมธอด.....	35
รูปที่ 2-22 ตัวอย่างของโซบเมสเซจที่เรียกเมธอดโดยรีเคเวสดีคโควท.....	35
รูปที่ 2-23 ตัวอย่างของชื่อเมธอดตามด้วยเรสสปอน.....	35
รูปที่ 2-24 ตัวอย่างไดอะแกรมของการสร้างโซบไคลเอนต์ออบเจกต์.....	37
รูปที่ 2-25 ตัวอย่างของการทำงานการประมวลผลภายในโซบไคลเอนต์ออบเจกต์.....	38
รูปที่ 2-26 ตัวอย่างของโซบเซิร์ฟเวอร์ออบเจกต์.....	39
รูปที่ 2-27 ตัวอย่างเซิร์ฟเวอร์ไซด์ค่าไฟล์.....	40
รูปที่ 2-28 ตัวอย่างของการทำงานการประมวลผลภายในโซบเซิร์ฟเวอร์ออบเจกต์.....	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2-29 ตัวอย่างของโครงสร้างของเอกสารฉบับวีเอสดีแอล.....	43
รูปที่ 2-30 แสดงการส่งเอกสารฉบับวีเอสดีแอล.....	44
รูปที่ 2-31 แสดงส่วนประกอบของฉบับวีเอสดีแอล.....	45
รูปที่ 2-32 เอกสารฉบับวีเอสดีแอล.....	46
รูปที่ 2-33 โครงสร้างของยูติลิตี้ไอ.....	49
รูปที่ 2-34 ประเภทของเอกสารฉบับวีเอสดีแอล.....	49
รูปที่ 2-35 การประกาศและค้นหาเอกสารฉบับวีเอสดีแอลในยูติลิตี้ไอ.....	51
รูปที่ 2-36 แสดงถึงรูปแบบการพัฒนาโปรแกรมโดยใช้บอร์คคอม86.....	52
รูปที่ 2-37 แสดงการเชื่อมต่อของบอร์คขยายกับบอร์คคอม86.....	53
รูปที่ 2-38 ตัวอย่างของภาพชุดพัฒนาคอม86.....	56
รูปที่ 2-39 บล็อกไดอะแกรมแสดงส่วนประกอบต่างๆของคอม86.....	57
รูปที่ 3-1 คอนเท็กซ์ไดอะแกรม (Context Diagram) ของเว็บเซอร์วิสบนระบบฝังตัว.....	58
รูปที่ 3-2 แสดงโครงสร้างภายในระบบเว็บเซอร์วิสบนระบบฝังตัว.....	59
รูปที่ 3-3 โครงสร้างทางสถาปัตยกรรมของเว็บเซอร์วิสบนระบบฝังตัว.....	60
รูปที่ 3-4 ยูสเคสไดอะแกรม (Use Case Diagram) ของระบบ.....	61
รูปที่ 3-5 แสดงคอมโพเนนต์ไดอะแกรม.....	64
รูปที่ 3-6 คลาสไดอะแกรมในส่วนของโซลาร์เซอร์.....	65
รูปที่ 3-7 คลาสไดอะแกรมในส่วนของเอ็กซ์เอ็มแอลพาร์เซอร์.....	68
รูปที่ 3-8 คลาสไดอะแกรมในส่วนของเว็บเซิร์ฟเวอร์.....	69
รูปที่ 3-9 คลาสไดอะแกรมในส่วนของเซอร์วิสโค้ด.....	70
รูปที่ 3-10 ซีควেনส์ไดอะแกรมแสดงการประกาศเว็บเซอร์วิส.....	71
รูปที่ 3-11 ซีควেনส์ไดอะแกรมแสดงการค้นหาเว็บเซอร์วิส.....	72
รูปที่ 3-12 ซีควেনส์ไดอะแกรมแสดงการร้องขอเอกสารฉบับวีเอสดีแอล.....	72
รูปที่ 3-13 ซีควেনส์ไดอะแกรมแสดงการดีพลอยเว็บเซอร์วิส.....	73
รูปที่ 3-14 ซีควেনส์ไดอะแกรมแสดงการอันดีพลอยเซอร์วิส.....	74
รูปที่ 3-15 ซีควেনส์ไดอะแกรมแสดงการตรวจสอบ.....	75
รูปที่ 3-16 แสดงการตรวจสอบล็อกไฟล์.....	75
รูปที่ 3-17 ซีควেনส์ไดอะแกรมแสดงการเรียกใช้เว็บเซอร์วิส.....	76
รูปที่ 3-18 ซีควেনส์ไดอะแกรม แสดงข้อผิดพลาดที่เกิดตอนเช็คเซอร์วิสเนมและเมธอดเนม.....	77
รูปที่ 3-19 ซีควেনส์ไดอะแกรม แสดงข้อผิดพลาดที่เกิดตอนเช็ครูปแบบของพารามิเตอร์.....	78
รูปที่ 3-20 แสดงสเตตชาร์ตไดอะแกรม.....	79

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3-21 แสดงดีฟลอยเมนท์ไดอะแกรม.....	80
รูปที่ 4-1 แสดงกระบวนการทำงานในส่วนของคลาสโซบเซิร์ฟเวอร์.....	81
รูปที่ 4-2 แสดงกระบวนการในการทำงานในส่วนของโซบริเคสแฮนเดอ์เลอร์.....	82
รูปที่ 4-3 แสดงกระบวนการในการทำงานในส่วนของโซบเรสสปอน.....	83
รูปที่ 4-4 แสดงการเรียกใช้งานแอปพลิเคชันเพื่อทำการลงทะเบียนเซอร์วิสเข้าในระบบ.....	83
รูปที่ 4-5 แสดงตัวอย่างแอปพลิเคชันเซอร์วิสเปิด-ปิดดวงไฟ.....	84
รูปที่ 4-6 แสดงตัวอย่างเอกสารอธิบายรายละเอียดของเซอร์วิสสำหรับเว็บเซิร์ฟเวอร์.....	84
รูปที่ 4-7 แสดงตัวอย่างเอกสารอธิบายรายละเอียดของเว็บเซอร์วิส.....	86
รูปที่ 5- 1 แสดงตัวอย่างของโซบแมสเซจที่บอกถึงการทำงานในภาวะปกติ.....	91
รูปที่ 5- 2 แสดงตัวอย่างของโซบแมสเซจที่บอกถึงการทำงานเมื่อเกิดข้อผิดพลาดขึ้น.....	92
รูปที่ ก- 1 ภาพการเชื่อมต่อสาย serial console เข้ากับบอร์ดคอม86.....	98
รูปที่ ก- 2 ภาพการเชื่อมต่อสายอีเธอร์เน็ตเข้ากับบอร์ดคอม86.....	98
รูปที่ ก- 3 ภาพการตั้งค่าต่างๆ ของโปรแกรมเทอร์มินัล.....	99
รูปที่ ก- 4 ภาพแสดงการเชื่อมต่อเคเบิลกับบอร์ดคอม86.....	100
รูปที่ ก- 5 ภาพแสดงตัวอย่างการแสดงผลของบอร์ดคอม86เมื่อบอร์ดเริ่มทำงาน.....	100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

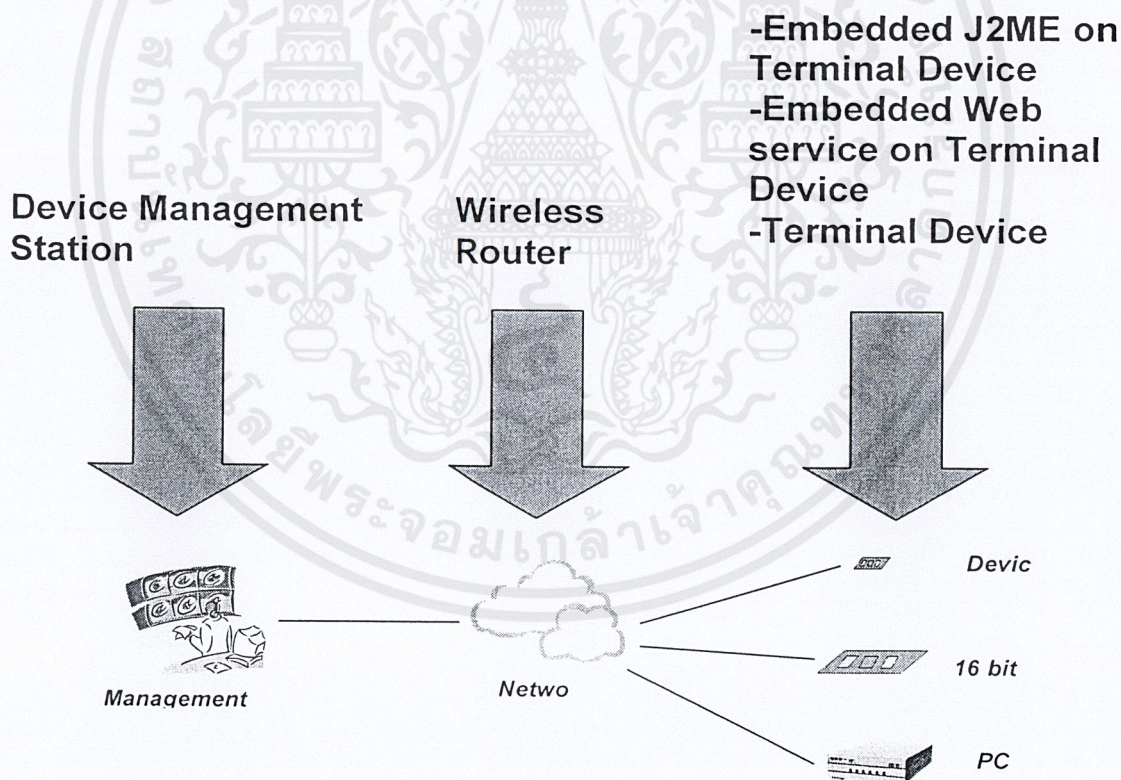
บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

ในปัจจุบันการควบคุมการทำงานของแอปพลิเคชัน (Application) บนอุปกรณ์ต่าง ๆ นั้น มักจะใช้คนเพื่อคอยควบคุมการทำงาน ซึ่งทำให้เกิดปัญหาต่างๆ ที่คนเป็นสาเหตุตามมา ยกตัวอย่าง เช่น เกิดจากความความสะเพร่า ความประมาท รวมถึงความล่าช้าในการทำงาน เป็นต้น ดังนั้นจึงได้มีแนวคิดในการที่จะปรับปรุงการทำงาน กล่าวคือ เปลี่ยนจากการใช้คนในการควบคุมมาเป็นการควบคุมด้วยแอปพลิเคชันแทน

ในการนำเครื่องคอมพิวเตอร์ที่มีราคาแพง มาใช้ในการควบคุมการทำงานระหว่างแอปพลิเคชัน อาจจะไม่คุ้มค่าต่อการลงทุน อีกทั้งยังมีปัญหาสถานะแวดล้อมของสถานที่ในการทำงานที่ไม่เหมาะสมกับการทำงานของเครื่องคอมพิวเตอร์ ซึ่งจะมีผลต่ออายุการใช้งานเครื่องคอมพิวเตอร์ดังกล่าว ทำให้มีค่าใช้จ่ายในส่วนของการบำรุงรักษามากขึ้น จากปัญหาเหล่านี้ จึงได้เกิดแนวคิดในการนำอุปกรณ์บนระบบฝังตัวที่มีความทนทานต่อสถานะแวดล้อมมากกว่าเครื่องคอมพิวเตอร์ ราคาถูกกว่า อีกทั้งยังมีความสะดวกในด้านการเชื่อมต่อกับอุปกรณ์ภายนอกที่ดีสร้างการเชื่อมต่อเข้ากับเครือข่ายเพื่อให้สามารถเรียกใช้งานได้จากทุกหนทุกแห่ง



รูปที่ 1-1 การเชื่อมต่ออุปกรณ์บนระบบฝังตัวผ่านเครือข่ายเพื่อให้สามารถเรียกใช้งานได้จากทุกแห่ง

การสื่อสารระหว่างแอปพลิเคชันจะช่วยให้การเชื่อมโยงข้อมูลผ่านทางอินเทอร์เน็ตโดยใช้เทคโนโลยีเว็บเซอร์วิส ที่มีการติดต่อสื่อสารโดยอาศัยซิมเปลอบเจกต์แอกเซสโปรโตคอล ที่ทำงานอยู่บนโปรโตคอลเอชทีทีพี ที่เป็นที่ยอมรับใช้งานกันอย่างแพร่หลาย เราจึงไม่ต้องทำการเปิดพอร์ตการสื่อสารขึ้นมาใหม่ และไม่เกิดปัญหาขึ้นในส่วนของไฟร์วอลล์ (Firewall) และพร็อกซีเซิร์ฟเวอร์ (Proxy Server) อีกทั้งซิมเปลอบเจกต์โปรโตคอลนั้น ยังเป็นมาตรฐานเปิดทำให้สามารถติดต่อกับเครื่องคอมพิวเตอร์ที่มีความแตกต่างกันทั้งระบบปฏิบัติการ เทคโนโลยี และภาษาในการพัฒนาซึ่งจะช่วยแก้ปัญหาของการเชื่อมโยงข้อมูลระหว่างเครื่องในแต่ละส่วนของขั้นตอนการทำงานที่แตกต่างกัน

ดังนั้นจึงได้มีความต้องการพัฒนาเว็บเซอร์วิส ให้สามารถนำไปประยุกต์ใช้งานร่วมกับระบบฝังตัว (Embedded System) ได้ เพื่อใช้เทคโนโลยีของเว็บเซอร์วิสซึ่งเป็นมาตรฐานกลางในการติดต่อสื่อสารผ่านเว็บมาพัฒนาการเชื่อมต่อของระบบฝังตัวให้สามารถที่จะใช้งานได้จากทุกหนทุกแห่งโดยจะทำการสร้างซอฟต์แวร์ไลบรารีแพ็คเกจที่สามารถให้บริการเว็บเซอร์วิสขึ้นมาเพื่อให้สามารถบริการเว็บเซอร์วิสบนระบบฝังตัวได้ โดยในโครงการนี้จะได้ทำการทดลองนำเอาซอฟต์แวร์ไปทดลองสร้างการให้บริการเว็บเซอร์วิสบนระบบฝังตัวซึ่งทำการเลือกบอร์ดคอม 86 เป็นแพลตฟอร์มตัวอย่างในการสร้าง

1.2 วัตถุประสงค์ของงานวิจัย

1. เพื่อเป็นการพัฒนาและทดสอบการให้บริการเว็บเซอร์วิสบนระบบฝังตัวว่าสามารถนำไปประยุกต์ใช้งานได้จริง
2. เพื่อพัฒนาการทำงานของอุปกรณ์บนระบบฝังตัวผ่านบริการเว็บเซอร์วิส ให้มีประสิทธิภาพมากขึ้น
3. เพื่อเรียนรู้และทดลองออกแบบและสร้างซอฟต์แวร์ที่สามารถจัดการเอกสารอิเล็กทรอนิกส์และเอกสารในรูปแบบมาตรฐานซิมเปลอบเจกต์โปรโตคอล
4. เพื่อเป็นการพัฒนาองค์ความรู้ (Knowledge Base) สำหรับนำไปใช้ในการวิจัยและพัฒนาทางด้านเว็บเซอร์วิสและระบบฝังตัวต่อไปในอนาคต

1.3 ขอบเขตของงานวิจัย

งานวิจัยนี้ต้องการพัฒนาซอฟต์แวร์ไลบรารีแพ็คเกจเพื่อให้สามารถนำไปประยุกต์ใช้งานในการสร้างบริการเว็บเซอร์วิสบนระบบฝังตัวต่างๆ โดยจะทำการสร้างแอปพลิเคชันที่ใช้จัดการเอกสารที่มีรูปแบบมาตรฐานซิมเปลอบเจกต์โปรโตคอล หรือ แอปพลิเคชันโซบพาสเซอร์ (SOAP Parser Application) ขึ้นมาเพื่อทำการติดต่อระหว่างแอปพลิเคชันที่ตั้งอยู่บนระบบเครือข่ายเดียวกัน โดยระบบฝังตัวจะทำหน้าที่เป็นเครื่องเซิร์ฟเวอร์ (Server) โดยนำไปติดตั้งใช้งานในระบบเครือข่าย (Network System) ที่มีโปรโตคอลพื้นฐานทีซีพีไอพี (TCP/IP Protocol) ทำให้ผู้ใช้สามารถเข้าใช้งานแอปพลิเคชันที่อยู่บนระบบฝังตัวจากระยะไกลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อีกส่วนหนึ่งจะทำการสร้างแอปพลิเคชันที่นำเสนอตัวอย่างในการให้บริการเว็บเซอร์วิสบนระบบฝั่งตัว กล่าวคือ จะเป็นโปรแกรมในฝั่งของไคลเอนต์ (Client) เพื่อตรวจสอบการทำงานของตัวไลบรารีแพ็คเกจว่าสามารถทำงานได้อย่างถูกต้องตามหลักการและมาตรฐานที่วางไว้

อนึ่งงานวิจัยนี้มีการนำชุดพัฒนาบอร์คคอม86ของบริษัทเน็ตเก็ทเจ็ต (Netgadgets) มาสำหรับใช้เป็นตัวอย่างแพลตฟอร์มในการวิจัยและพัฒนาตัวอย่างการให้บริการเว็บเซอร์วิสในงานวิจัยนี้

1.4 เป้าหมายของโครงการ

1. ซอฟต์แวร์ไลบรารีที่จัดการกับเอกสารเอ็กซ์เอ็มแอล (XML Parser Library) พร้อมทั้งองค์ประกอบต่างๆของตัวซอฟต์แวร์
2. ซอฟต์แวร์ไลบรารีที่จัดการกับเอกสารที่มีรูปแบบตามมาตรฐานซิมเปลอบเจ็คต์แอกเซส โพรโตคอล (SOAP Parser Library) พร้อมทั้งองค์ประกอบต่างๆของตัวซอฟต์แวร์
3. ระบบการให้บริการเว็บเซอร์วิสบนระบบฝั่งตัวที่นำเอาส่วนของซอฟต์แวร์ไลบรารีที่สร้างขึ้นมาประกอบกันโดยนำเอาบอร์คคอม 86 มาเป็นตัวอย่างแพลตฟอร์มของระบบฝั่งตัว

1.5 วิธีการดำเนินงาน

1. ศึกษาทฤษฎีพื้นฐานต่างๆ ที่เกี่ยวข้องกับงานวิจัย ซึ่งได้แก่ ระบบเว็บเซอร์วิส ภาษาเอ็กซ์เอ็มแอล มาตรฐานซิมเปลอบเจ็คต์แอกเซส โพรโตคอล มาตรฐานดับบิวเอสดีแอล โพรโตคอล ยูดีดีไอ โพรโตคอล ระบบฝั่งตัว และบอร์คคอม86
2. นำความรู้ที่ได้ศึกษามาทั้งหมดนั้น นำมาวิเคราะห์และออกแบบตัวซอฟต์แวร์หรือไลบรารีโดยการทำการศึกษาวិศกรรมย้อนกลับ (Reverse Engineering) ตัวโครงสร้างของแอปพลิเคชันที่ใช้จัดการเอกสารที่มีรูปแบบมาตรฐานซิมเปลอบเจ็คต์โพรโตคอล ทำการออกแบบรายละเอียดและโครงสร้างของแอปพลิเคชันนั้นใหม่ เพื่อให้เหมาะสมกับการใช้งานในระบบฝั่งตัว โดยใช้การออกแบบด้วยยูเอ็มแอล (UML) หรือ ยูนิฟาย โมเดลลิง แลนเกจ (Unified Modeling Language) ซึ่งเป็นโมเดลที่สนับสนุนการวิเคราะห์และออกแบบระบบเชิงวัตถุ (Methodology)
3. พัฒนาซอฟต์แวร์ไลบรารีที่จัดการกับภาษาเอ็กซ์เอ็มแอล (XML Parser Library) พร้อมทั้งองค์ประกอบต่างๆของตัวซอฟต์แวร์
4. พัฒนาซอฟต์แวร์ไลบรารีที่จัดการกับเอกสารที่มีรูปแบบตามมาตรฐานซิมเปลอบเจ็คต์แอกเซส โพรโตคอล (SOAP Parser Library) พร้อมทั้งองค์ประกอบต่างๆของตัวซอฟต์แวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. พัฒนาส่วนของเซิร์ฟเวอร์ที่จัดการกับการเรียกใช้บริการเว็บเซอร์วิส โดยมีการนำเอาซอฟต์แวร์ไลบรารีที่ได้จัดทำขึ้นมาประกอบกันและจัดทำส่วนของอินเตอร์เฟซในส่วน of เว็บเซิร์ฟเวอร์และส่วนของการสร้างเซอร์วิสขึ้นมาเพื่อให้บริการ
6. ขั้นตอนของการทดสอบซอฟต์แวร์ที่สร้างขึ้นมา ตรวจสอบความถูกต้องว่าสามารถทำงานได้ตรงตามมาตรฐานที่ได้กำหนดไว้ตอนต้น โดยการสร้างแอปพลิเคชันขึ้นมาเพื่อทำหน้าที่ในการทดสอบซอฟต์แวร์ รายละเอียดต่างๆของการทดสอบตัวซอฟต์แวร์หลังจากนั้นก็ให้นำผลลัพธ์ต่างๆที่ได้จากการทดสอบตัวซอฟต์แวร์
7. สรุปผลการทำงานซอฟต์แวร์ ผลที่ได้รับจากงานวิจัยชิ้นนี้ และแนวทางในการพัฒนางานวิจัยนี้เพิ่มเติมและแนวทางในการนำไปประยุกต์ใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

2.1 เว็บเซอร์วิส (Web Service)

2.1.1 การพัฒนาของเว็บ

การพัฒนาทางเทคโนโลยีใด ๆ ล้วนแล้วแต่มีพัฒนาการเป็นยุคสมัย แต่ละยุคสมัยมีการใช้เทคโนโลยีที่มีจุดสำคัญของการพัฒนาที่เด่นชัด การพัฒนาเทคโนโลยีการประยุกต์ใช้งานบนเว็บก็เช่นเดียวกัน

2.1.1.1 สแตติกเว็บเพจ (Static Web Page)

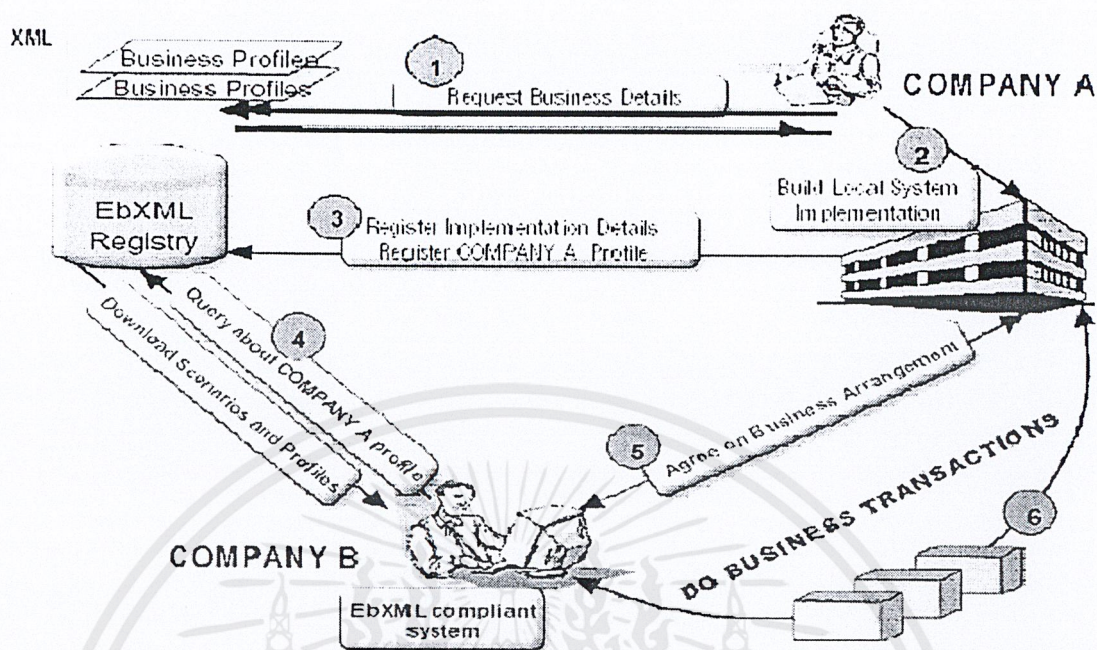
เป็นยุคที่เริ่มต้นการใช้งานเว็บมีการวางข้อมูลตามมาตรฐานเอชทีเอ็มแอล (HTML) ไว้เป็นไฟล์บนเซิร์ฟเวอร์ ผู้ใช้งานใช้เบราว์เซอร์ (browser) เรียกข้อมูลด้วยโปรโตคอลเอชทีทีพี การเรียกใช้ข้อมูลจะได้ข้อมูลที่เก็บไว้บนเซิร์ฟเวอร์ โดยลักษณะของข้อมูลยังเป็นข้อมูลแบบสแตติก คือเก็บไว้เพื่อรอการเรียกใช้ บนฝั่งของเซิร์ฟเวอร์เองยังไม่มีฝั่งโปรแกรมพิเศษใด ๆ เพียงแต่มีโปรแกรมที่รับการติดต่อด้วยเอชทีทีพีเท่านั้น ข้อมูลที่เรียกใช้ได้รับมาเป็นไฟล์ โดยทางฝั่งไคลเอนต์จะนำไปแสดงผลตามข้อมูลที่ปรากฏในเท็กซ์ (text) ตามมาตรฐานในยุคที่หนึ่งนี้ทำให้มีการเก็บข้อมูลบนเซิร์ฟเวอร์เพื่อเป็นจุดที่ทำให้ผู้ใช้งานเรียกเข้าใช้ได้ง่าย การพัฒนาระบบบริการข้อมูลแบบเว็บเพจ (web page) หรือ โฮมเพจ (home page) จึงได้รับความสนใจ ทำให้มีผู้สร้างเว็บเพจและผู้ผ่านเบราว์เซอร์เป็นจำนวนมาก

2.1.1.2 ไดนามิกเว็บเพจ (Dynamic Web Page)

มีการพัฒนาโปรแกรมทางฝั่งเซิร์ฟเวอร์ เพื่อให้เซิร์ฟเวอร์ทำงานตามคำเรียกขอมมาจากทางไคลเอนต์ผ่านทางโปรโตคอลเอชทีทีพี หรือการเรียกของบนเว็บ มีการเชื่อมโยงกับโปรแกรมเฉพาะ เช่นเมื่อผู้เรียกใช้คลิกบนเบราว์เซอร์ ทำให้เกิดการเรียกขอมที่เซิร์ฟเวอร์ เซิร์ฟเวอร์จะทำงานตอบสนอง โดยอาจจะไปรันโปรแกรม หรือเรียกโปรแกรมอื่นที่ทางฝั่งเซิร์ฟเวอร์เตรียมไว้ ในยุคสมัยนี้มีการสร้างระบบการเรียกเข้าหาโปรแกรมที่เรียกว่า ซีจีไอ หรือ คอมมอนเกตเวย์อินเทอร์เฟซ (CGI-Common Gateway Interface) การเขียนโปรแกรมมีเทคนิควิธีการพิเศษที่ทำให้เข้าถึงฐานข้อมูล มีการพัฒนาเครื่องมือการพัฒนา และภาษาสำหรับพัฒนาเพื่อทำให้เรียกข้อมูลจากฐานข้อมูลหรือเรียกเพิ่มข้อมูลอื่นเพื่อส่งต่อให้ผู้เรียกใช้ ภาษาที่นิยมใช้ในการพัฒนามีภาษาซี (C Language) ภาษาเพิร์ล (PERL Language) และเครื่องมือพิเศษที่ใช้เชื่อมโยงกับฐานข้อมูลที่ผู้ผลิตซอฟต์แวร์ทางฐานข้อมูลดำเนินการให้ ตลอดจนมีการใช้โปรแกรมทางด้านไคลเอนต์ที่เริ่มซับซ้อนขึ้น

การที่เรียกว่าไดนามิกเว็บเพจก็ด้วยเหตุผลที่ว่าข้อมูลที่เรียกดูจะแปรเปลี่ยนตามการเรียกขอ การปรับเปลี่ยนข้อมูลกระทำตามโปรแกรมที่สร้างขึ้น สามารถคำนวณหรือประมาณผลข้อมูลได้ เช่น การดูข้อมูลสรุปของแต่ละวัน ดังนั้นการเรียกดูข้อมูลจะได้ผลที่ทันสมัยได้ ระบบเซิร์ฟเวอร์จึงเชื่อมโยงกับฐานข้อมูลหรือระบบงานอื่น แต่เมื่อจะส่งข้อมูลไปยังผู้ใช้ที่เรียกขอเข้ามา จะแปลงให้อยู่ในรูปแบบของเอชทีเอ็มแอลก่อนเพื่อให้เบราว์เซอร์แสดงผลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-1 ลักษณะของเว็บเซอร์วิส

2.1.1.3 บริการบนเว็บ (Web Service)

การพัฒนาเว็บมาจนถึงปัจจุบันได้ก้าวหน้าขึ้นไปอีกมาก จุดสำคัญอยู่ที่ความต้องการรองรับการใช้งานที่มีความต้องการสูงขึ้น โดยเฉพาะการดำเนินการทางธุรกิจการค้าต่างๆ ที่เพิ่มมากขึ้นบนเครือข่าย ลักษณะงานที่ต้องการมีลักษณะการทำงานร่วมกันระหว่างองค์กร (Interoperability) โดยให้โปรแกรมประยุกต์ขององค์กรหนึ่งส่งคำขอผ่านเครือข่ายอินเทอร์เน็ตด้วยโปรโตคอลเอชทีทีพี ไปยังเว็บบริการของอีกองค์กรหนึ่ง มีการโต้ตอบเพื่อรับส่งข้อมูลระหว่างกันแบบอัตโนมัติได้ การเรียกใช้บริการต่างๆ มีลักษณะการเรียกใช้บริการ โดยมีตัวกลางช่วยและเชื่อมโยงเข้ากับเว็บเซิร์ฟเวอร์ การแลกเปลี่ยนข้อมูลตามคำเรียกขอใช้มาตรฐานเอ็กซ์เอ็มแอล หรือเอ็กซ์เทนซิเบิลมาร์กอัปแลงกเวจ (XML-eXtensible Mark Up Language) ข้อมูลที่เป็นเอ็กซ์เอ็มแอลจะถูกส่งผ่านมายังการเรียกขอเอชทีทีพี การบริการบนเว็บจึงเป็นการก้าวเข้ามาในระบบที่ทำให้เกิดการแลกเปลี่ยนข้อมูลระหว่างกัน ซึ่งแต่เดิมมีวิธีการแลกเปลี่ยนข้อมูลกันตามรูปแบบที่เรียกว่า อีดีไอ (EDI) แต่จากการที่มีการใช้เอ็กซ์เอ็มแอล ทำให้การทำงานได้ตรงตามความต้องการ

ลักษณะสำคัญของการบริการบนเว็บจึงเป็นงานประยุกต์ที่เซิร์ฟเวอร์บริการสามารถเชื่อมโยงไปกับการประยุกต์อื่น ๆ โดยสามารถเรียกหาข้อมูลจากที่อื่นเพื่อนำมาประมวลผล และส่งต่อให้ผู้ใช้ที่เรียกเข้ามาต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 เว็บเซอร์วิส

การพัฒนาทางเทคโนโลยีใดๆ ล้วนแล้วแต่มีการพัฒนาเป็นยุคสมัย แต่ละยุคสมัยก็จะมีการใช้เทคโนโลยีที่มีจุดสำคัญของการพัฒนาที่เด่นชัด การพัฒนาเทคโนโลยีประยุกต์ใช้งานบนเว็บก็เช่นเดียวกัน เริ่มจากยุคแรกที่เป็นแบบสแตติกเว็บเพจที่เป็นเพียงแค่ไฟล์เอชทีเอ็มแอลไว้ให้ผู้ดูข้อมูลเฉยๆ จนสู่ยุคที่ 2 เป็นแบบไดนามิกเว็บเพจ (Dynamic Web Page) ที่สามารถแปรเปลี่ยนตามคำร้องขอของผู้ใช้งานได้ ซึ่งเป็นการเพิ่มประสิทธิภาพของเว็บเพจ จนกระทั่งในปัจจุบันมีการพัฒนาเว็บเพจเข้ามาสู่ยุคที่ 3 หรือแบบการบริการผ่านเว็บเซอร์วิส โดยจุดประสงค์เพื่อรองรับความต้องการการใช้งานที่สูงขึ้น และเป็นมาตรฐานในการทำงานร่วมกัน

เว็บเซอร์วิส คือ ซอฟต์แวร์คอมโพเนนท์ (Software Component) ที่ผู้ให้บริการนำมาสร้างเป็นแอปพลิเคชันสำหรับให้บริการการทำงานหนึ่งๆ ให้กับผู้ใช้บริการทางอินเทอร์เน็ต หรือผู้ใช้บริการสามารถที่จะขอบริการจากหลายๆ ที่มาประกอบกันได้ โดยที่แต่ละระบบนั้นมีความเป็นอิสระจากกัน (Loosely Coupled) ซึ่งปัจจัยพื้นฐานของเว็บเซอร์วิสที่ควรมี ซึ่งมีความเกี่ยวข้องกันในทางเทคนิคและทางธุรกิจคือ

- การรวมซอฟต์แวร์ต่างระบบกัน จะต้องอนุญาตให้แต่ละระบบมีความเป็นอิสระจากกัน
- อินเทอร์เน็ตทางด้านบริการของซอฟต์แวร์ควรเผยแพร่สู่สาธารณชนและสามารถเข้าถึงได้ง่าย
- เมสเซจที่ใช้ติดต่อกันของการทำงานแบบแอปพลิเคชันกับแอปพลิเคชันต้องสอดคล้องกับมาตรฐานเปิดบนอินเทอร์เน็ต
- แอปพลิเคชันสามารถสร้างได้จากการใช้ซอฟต์แวร์คอมโพเนนท์ทั้งจากภายในและภายนอกองค์กร โดยสร้างตามการดำเนินธุรกิจหลักขององค์กร
- มีแหล่งซอฟต์แวร์คอมโพเนนท์ที่ทำได้ง่าย ซึ่งช่วยเพิ่มความยืดหยุ่นในการสร้างแอปพลิเคชันที่มีคุณลักษณะเฉพาะตามกระบวนการดำเนินการทางธุรกิจ
- การนำซอฟต์แวร์จากภายนอกมาใช้ใหม่ช่วยให้เกิดการลดต้นทุนและช่วยปรับปรุงบริการให้แก่ลูกค้า
- ซอฟต์แวร์สามารถขายเป็นบริการได้

หลักการพื้นฐานของเทคโนโลยีต่างๆ ที่พัฒนามาสู่เว็บเซอร์วิส

- การพัฒนาโปรแกรมแบบซอฟต์แวร์คอมโพเนนท์ ตามแนวคิดของการออกแบบและพัฒนาโปรแกรมเชิงวัตถุ (Object-Oriented Concept)
- การออกแบบระบบแบบกระจายจากศูนย์กลาง (Distributed Computing) ซึ่งเป็นเป้าหมายสำคัญของการพัฒนาระบบตามสถาปัตยกรรมแบบไคลเอนต์เซิร์ฟเวอร์ (Client-Server)

- การทำอีดีไอ หรือ อิเล็กทรอนิกส์ค้าค้าอินเทอร์เน็ต (Electronic Data Interchange) ซึ่งสร้างขึ้นโดยกำหนดรูปแบบและมาตรฐานของข้อมูลสำหรับการทำธุรกิจ
- การบูรณาการของซอฟต์แวร์ต่างระบบ หรือ เอนเทอร์ไพรส์ แอปพลิเคชัน อินทิเกรชัน : อีเอไอ (Enterprise Application Integration : EAI) ที่อยู่บนพื้นฐานของความต้องการใช้ข้อมูลร่วมกัน รวมทั้งการแลกเปลี่ยนข้อมูลระหว่างแอปพลิเคชันให้สามารถทำงานได้อย่างถูกต้องเหมาะสม
- รูปแบบการให้บริการซอฟต์แวร์แบบเอเอสพี (ASP) หรือ แอปพลิเคชันเซอร์วิสโพรไวเดอร์ (Application Service Provider)
- แนวคิดการทำเหมืองข้อมูล (Data Mining) ซึ่งต้องการนำข้อมูลที่ถูกจัดเก็บไว้ในรูปแบบที่แตกต่างกันตามแหล่งต่างๆมาใช้งานร่วมกัน

โดยที่เว็บเซอร์วิสได้นำหลักการต่างๆ เหล่านี้มาดำเนินการปรับปรุงแก้ไขข้อจำกัดต่างๆ โดยการคิดค้นเทคโนโลยีที่เป็นมาตรฐานเปิด (Opened Standard) และไม่มีใครเป็นเจ้าของ (Neural Vendor)

2.1.3 ลักษณะของเว็บเซอร์วิส

การพัฒนาเว็บเซอร์วิสใช้สถาปัตยกรรมบริการในลักษณะที่เรียกว่า สถาปัตยกรรมของแนวคิดทางด้านบริการ หรือ เอส โอเอ (Service-Oriented Architecture : SOA) เป็นแนวคิดเบื้องต้น แอปพลิเคชันส่วนใหญ่ในโลกของธุรกิจที่ใช้งานในปัจจุบัน เป็นแอปพลิเคชันและระบบย่อยที่ถูกสร้างขึ้น มีการทำงานที่ต้องสัมพันธ์กันอย่างไม่เป็นอิสระจากกัน การเปลี่ยนแปลงการทำงานในระบบย่อยหรือแอปพลิเคชันหนึ่ง อาจจะมีผลกระทบต่ออีกแอปพลิเคชันหนึ่ง หรือบางครั้งอาจจะกระทบทั้งระบบ ทำให้การบำรุงรักษานั้นทำได้ยากและมีต้นทุนที่สูงขึ้น รวมทั้งยังเป็นข้อจำกัดในการเชื่อมต่อกับระบบของคู่ค้าอื่นๆ

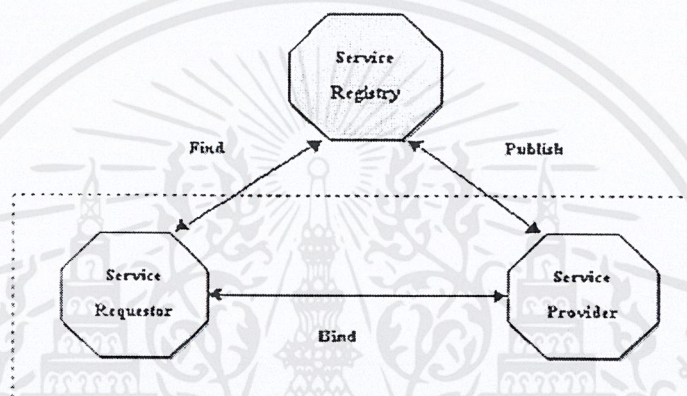
เอส โอเอไม่ใช่แนวคิดใหม่ แต่ได้เกิดขึ้นมานานแล้ว ซึ่งอยู่ในส่วนหนึ่งของแนวคิดการออกแบบระบบแบบกระจายจากศูนย์กลาง (Distributed Computing Concepts) เอสโอเอ นั้นเป็นแนวคิดที่ได้รับการยอมรับและประสบความสำเร็จในระดับหนึ่ง การพัฒนาเอสโอเอ นั้นทำกันบนมาตรฐานเปิด ซึ่งได้รับการรับรองจากผู้ผลิตซอฟต์แวร์ชั้นนำและยังได้ร่วมมือกันรับรองมาตรฐานของเทคโนโลยีต่างๆ ที่ประกอบกันเป็นเว็บเซอร์วิส เช่น ซิมเปลอบเจกต์เอกเซสโพรโตคอล (SOAP), คับบิวเอสดีแอล (WSDL) และ ยูดีดีไอ (UDDI) เป็นต้น

เอสโอเอมีส่วนประกอบหลัก 3 ส่วนคือ

1. ผู้ให้บริการ หรือที่เรียกว่า เซอร์วิสโพรไวเดอร์ (Service Provider)
2. ผู้ขอใช้บริการ หรือที่เรียกว่า เซอร์วิสรีควีสเตอร์ (Service Requester)
3. ตัวแทนของผู้ให้บริการ หรือที่เรียกว่า เซอร์วิสโบรกเกอร์ (Service Broker)

โดยส่วนประกอบหลักทั้ง 3 ส่วนนี้ สามารถติดต่อถึงกันได้โดยใช้ฟังก์ชันพื้นฐาน (Primary Function) คือการประกาศ (Publish) การค้นหา (Find) และการเรียกใช้ (Bind) ซึ่งฟังก์ชันทั้ง 3 มีการทำงานดังนี้

1. ผู้ให้บริการทำการประกาศบริการที่ตนเองให้บริการ ไปยังตัวแทนของผู้ให้บริการ ซึ่งตัวแทนของผู้ให้บริการจะทำการบันทึกเก็บไว้ในไดเรกทอรีของบริการ (Directory Service)
2. ผู้ขอใช้บริการจะทำการค้นหาบริการจากตัวแทนของผู้ให้บริการ
3. เมื่อพบบริการที่ต้องการแล้ว ผู้ให้บริการและผู้ขอใช้บริการจะทำการติดต่อกัน โดยผู้ขอใช้บริการจะทำการเรียกใช้บริการไปยังผู้ให้บริการนั้น



รูปที่ 2-2 แสดงความสัมพันธ์ของผู้ให้บริการ ผู้ขอใช้บริการ และตัวแทนผู้ให้บริการ

2.1.4 เทคโนโลยีพื้นฐานของเว็บเซอร์วิส

จากแนวคิดของเอสไอเอ ถูกนำมาใช้เป็นหลักการพื้นฐานของการพัฒนาเทคโนโลยีด้านเว็บเซอร์วิส ซึ่งเทคโนโลยีที่ถูกพัฒนาขึ้นมานี้เป็นมาตรฐานเปิดบนอินเทอร์เน็ตที่เกิดจากการทำงานร่วมกันของนักวิจัย และที่ปรึกษาจากบริษัทซอฟต์แวร์ต่างๆ

- เอ็กซ์เอ็มแอล หรือ เอ็กเทนซิเบิลมาร์กอัปแลงกเวจ เป็นภาษามาร์กอัป ที่เป็น เท็กซ์เบส (Text-Based) ที่ใช้เป็นมาตรฐานในการแลกเปลี่ยนข้อมูลบนอินเทอร์เน็ตในปัจจุบัน ผู้ที่ทำหน้าที่รับผิดชอบ และกำหนดมาตรฐานของ เอ็กซ์เอ็มแอล คือ ดับเบิลยูทีซี หรือ เวิลด์ไวด์เว็บคอนซอร์ตียม (W3C -World Wide Web Consortium)
- โซบ หรือ ซิมเปลออปเจกต์แอคเซสโพรโตคอล เป็นโพรโตคอลที่สร้างบนพื้นฐานของเอ็กซ์เอ็มแอล และยังเป็นแมสเซจจิงโพรโตคอล (Messaing Protocol) สำหรับใช้ในการแลกเปลี่ยนข้อมูลในสภาวะแวดล้อมแบบกระจายศูนย์ (Decentralized, Distributed Environment) โดยโซบ ได้กำหนดแมสเซจจิงโพรโตคอลระหว่างผู้ขอบริการกับผู้ให้บริการในการติดต่อสื่อสารกัน เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดให้ผู้ขอใช้บริการต้องส่งข้อมูล เช่น ข้อมูลที่ระบุฟังก์ชัน และค่าพารามิเตอร์ต่างๆที่ต้องใช้ในแอปพลิเคชันที่ร้องขอ ส่งไปให้กับผู้ให้บริการ

- **ดับเบิลวีเอสดีแอล** หรือ **เว็บเซอร์วิสเดสคริปต์ซันแลงแวก** (WSDL - Web Services Description Language) เป็นภาษาที่ใช้อธิบายคุณลักษณะการให้บริการของเว็บเซอร์วิส และวิธีการติดต่อขอรับบริการจากเว็บเซอร์วิส ดับเบิลวีเอสดีแอลสร้างขึ้นโดยใช้ภาษาเอ็กซ์เอ็มแอล ซึ่งเกิดจากการรวมแนวความคิดของเอนเอเอสเอสแอล หรือ **เน็ตเวิร์คแอคเซสซิเบิลเซอร์วิสสเปคซิฟิเคชันแลงแวก** (NASSL - The Network Accessible Service Specification Language) และ **ดับเบิลวีเอสดี** หรือ **เวลดีไฟายเซอร์วิส** (WDS - Well-Defined Services) ของบริษัทไอบีเอ็มกับเอสดีแอล หรือ **เซอร์วิสเอสคริปต์ซันแลงแวก** (SDL - The Service Description Language) และ **เอสซีแอล** หรือ **โซบคอนแทร็กแลงแวก** (SCL - The SOAP Contract Language) ของบริษัทไมโครซอฟต์ ปัจจุบันดับเบิลวีเอสดีแอลเป็นภาษาที่อยู่ในการดูแลขององค์กรดับเบิลวีทีซี

2.2 เอ็กซ์เอ็มแอล (XML)

มาร์กอัปแลงแวก (Markup language) คือ ภาษาที่ใช้อธิบายความหมายของเอกสารหรือข้อมูล เราจะพบอยู่ในรูปของแท็ก (tag) เป็นส่วนมาก มาร์กอัปแลงแวกมีข้อดีคือสามารถใช้ได้กับทุกแพลตฟอร์ม วิศวกรของ มาร์กอัปแลงแวกนั้น เริ่มต้นจากจีเอ็มแอล (GML) หรือ **เจนเนอไรซ์มาร์กอัปแลงแวก** (Generalized Markup Language) ซึ่งมีแนวคิดคือผู้ใช้สามารถคิดกลุ่มของแท็กขึ้นมาเองเพื่อใช้อธิบายเอกสาร ต่อมามีการพัฒนาจีเอ็มแอล ให้มีความเป็นมาตรฐานขึ้นจนกลายเป็น **เอสจีเอ็มแอล** (SGML) หรือ **สแตนดาร์ดเจนเนอไรซ์มาร์กอัปแลงแวก** (Standard Generalized Markup Language) ซึ่งมี **ดีทีดี** (DTD) หรือ **ด็อกคิวเมนต์ไทป์เดฟิเนชัน** (Document Type Definition) เป็นตัวนิยามโครงสร้างของเอกสาร หรือสามารถพูดได้อีกว่า เราจะอาศัย ดีทีดี เป็นต้นแบบในการสร้างเอกสารเอสจีเอ็มแอล

ต่อมา ในปี ค.ศ .1991 **ทิม เบอเนอลี** (Tim Berners-Lee) แห่งสถาบันวิจัยเซอน (CERN) ประเทศสวิสเซอร์แลนด์ได้นำพื้นฐานของภาษาเอสจีเอ็มแอล ผสมเข้ากับความสามารถของไฮเปอร์เท็กซ์ (Hypertext) และ **ยูอาร์แอล** (URL) เพื่อการเชื่อมโยงระหว่างคอมพิวเตอร์หลายๆเครื่อง จนกลายมาเป็น **เอชทีเอ็มแอล** หรือ **ไฮเปอร์เท็กซ์มาร์กอัปแลงแวก** (Hypertext Markup Language) โดยมีการสร้างกลุ่มของแท็ก หรือ **เอลเมนต์** (Element) ขึ้นเพื่อใช้ในการแสดงผลบนเว็บผ่านเว็บเบราว์เซอร์ (Web Browser) ซึ่งนับว่าเป็นมาตรฐานที่มีประสิทธิภาพในการใช้ออกแบบเว็บเพจ

2.2.1 ข้อจำกัดของเอชทีเอ็มแอล

ถึงแม้ว่าเราจะสามารถสร้างเว็บเพจ ได้อย่างสวยงามและหลากหลายด้วยเอชทีเอ็มแอล แต่ความสามารถที่แท้จริงของเอชทีเอ็มแอล ก็เพียงแค่การแสดงผลทางเว็บเบราว์เซอร์เท่านั้น ข้อมูลที่นิยามด้วย

แท็กเอชทีเอ็มแอล ไม่สามารถจะไปใช้งานอย่างอื่นได้เลย นอกจากนี้เอชทีเอ็มแอลก็ยังมีข้อจำกัดหลายประการซึ่งทำให้มันยังไม่สามารถตอบสนองความต้องการของผู้ใช้ได้เต็มที่ ดังนี้

- เราไม่สามารถกำหนดแท็กขึ้นมาเองได้ ทำให้การทำงานมีขีดจำกัดแท็กของเอชทีเอ็มแอล มีความหมายที่กำหนดไว้แล้วชัดเจนโดยองค์กรดับบิวทรีซี หรือ เวิลด์ไวด์เว็บคอนซอร์ตียม และเป็นมาตรฐานเดียวกันเพื่อให้เว็บเบราว์เซอร์ สามารถแปล ความหมายของแท็กที่อยู่ในเว็บเพจและแสดงผลได้อย่างถูกต้อง
- เอชทีเอ็มแอลถูกออกแบบมาเพื่อจัดรูปแบบข้อมูลในการแสดงผลเป็นหลัก เช่น บอกให้เบราว์เซอร์รู้ว่าต้องจัดข้อความใดเป็นหัวหน้าตัวเองเท่านั้นไม่ได้สื่อความหมายของข้อมูลเอกสาร ถ้าไม่มีเว็บเบราว์เซอร์ ภาษาเอชทีเอ็มแอลก็ไม่มีความหมายเลย
- เอชทีเอ็มแอลไม่จำเป็นต้องอยู่ในรูปแบบที่ฟอร์มที่ถูกต้อง (Well-Formed) ก็ยังสามารถทำงานได้ ทำให้มีปัญหาคือ เมื่อแสดงผลไฟล์เอชทีเอ็มแอลเดียวกันจากเบราว์เซอร์ต่างกัน ผลลัพธ์ออกมาไม่เหมือนกัน เพราะมีความตีความแตกต่างกันไป
- เอชทีเอ็มแอลไม่เหมาะกับการที่ต้องมีการเปลี่ยนแปลงข้อมูลบ่อย เพราะการแก้ไขเอกสารแต่ละครั้งทำได้ยาก
- เมื่อแปลงข้อมูลจากไฟล์ข้อความธรรมดาเป็น เอชทีเอ็มแอล แล้ว ในการแปลงจากไฟล์ เอชทีเอ็มแอล กลับมาเป็นไฟล์ข้อความธรรมดาทำได้ยาก

จากข้อจำกัดเหล่านี้ทำให้องค์กรดับ บิวทรีซี มีความเห็นว่าแนวโน้มต่อไปของภาษาคควรจะยืดหยุ่นได้มากกว่าภาษาเอชทีเอ็มแอล จึงได้สร้างภาษาเอ็กซ์เอ็มแอล หรือ เอ็กซ์เทนซิเบิลมาร์กอัปแลงเกจขึ้นมา โดยมีพื้นฐานมาจากภาษา 2 ภาษา คือ ภาษาเอสจีเอ็มแอลโดยเฉพาะในแง่ของสามารถนิยามของข้อมูล และ ภาษาเอชทีเอ็มแอล

2.2.2 ลักษณะที่สำคัญของเอ็กซ์เอ็มแอล

เอ็กซ์เอ็มแอล เป็นภาษามาร์กอัปที่เป็นแท็กเบสที่ใช้เป็นมาตรฐานในการแลกเปลี่ยนข้อมูลบนอินเทอร์เน็ตในปัจจุบัน โดยมีผู้ที่ทำหน้าที่รับผิดชอบและกำหนดมาตรฐานของเอ็กซ์เอ็มแอล คือ องค์กรดับบิวทรีซี

- จุดประสงค์ในการออกแบบ เพื่อกำหนดเป็นมาตรฐานในการทำเอกสารบนเครือข่ายอินเทอร์เน็ต เพื่อการแลกเปลี่ยนข้อมูลที่เป็นลำดับขั้น ดังนั้นจึงต้องมีรูปแบบที่สามารถใช้ได้ทั่วไปบนอินเทอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีคุณลักษณะไม่ขึ้นกับแพลตฟอร์ม (Platform Independence) สามารถนำไปใช้กับคอมพิวเตอร์ระบบใด แพลตฟอร์มไหนก็ได้ เนื่องจากเอกสารเอ็กซ์เอ็มแอล เป็นเท็กซ์ไฟล์ (text file) ธรรมดา
- เป็นภาษาที่ได้รับการออกแบบมาเพื่อให้สามารถนิยามความหมายของข้อมูลได้ จึงมีการจัดโครงสร้างข้อมูล แบ่งข้อมูลออกเป็นหมวดหมู่และส่วนประกอบย่อย
- ไม่มีแท็กที่ถูกนิยามไว้ก่อน อนุญาตให้ผู้ใช้สร้างแท็กขึ้นมาเองเพื่อใช้อธิบายข้อมูลได้ โดยที่ผู้ใช้กำหนดแท็ก และใช้งาน ได้ทันที เนื่องจากแท็กเป็นอะไรก็ได้ที่ผู้ใช้กำหนดจึงทำให้ เอ็กซ์เอ็มแอล ขยายขีดความสามารถต่อไปได้ (Extensible) และข้อมูลสามารถอธิบายความหมายข้อมูลของตัวเองได้ (Self Describe)
- ส่วนข้อมูลและส่วนการแสดงผลของเอกสารเอ็กซ์เอ็มแอล ถูกแยกออกจากกันอย่างชัดเจน ในเอกสารเอ็กซ์เอ็มแอล นั้น จะมีแต่ตัวเนื้อข้อมูล ส่วนการแสดงผลนั้น เราสามารถใช้สไตลชีท (style sheet) ได้หลายประเภท เช่น ซีเอสเอส (CSS), เอ็กซ์เอสแอล (XSL) เป็นต้น
- เนื่องจากเอกสารเอ็กซ์เอ็มแอล เป็นเพียงข้อความง่ายๆที่ประกอบด้วยแท็กบางอย่างเท่านั้นจึงสามารถสร้างเอกสารเอ็กซ์เอ็มแอล ด้วยเท็กซ์อีดิเตอร์ (text editor) ทั่วๆไปได้
- การอ่านและแปรความหมายของเอกสาร เอ็กซ์เอ็มแอล สามารถทำได้โดยใช้โปรแกรมที่เรียกว่า เอ็กซ์เอ็มแอลพาร์เซอร์ (XML Parser) ได้ ตัวอย่างของโปรแกรมประเภท เอ็กซ์เอ็มแอลพาร์เซอร์ เช่น เอ็มเอสเอ็กซ์เอ็มแอล (MSXML) ซึ่งอยู่ในอินเทอร์เน็ตเอ็กซ์พลอเรอร์ (Internet Explorer) ของไมโครซอฟต์ และ เจเอเอ็กซ์พี (JAXP) ของบริษัทซัน (Sun) เป็นต้น
- มีวิธีกำหนดโครงสร้างเอกสาร 2 วิธีคือ คีทีดี และ เอ็กซ์เอ็มแอลสครีม่า (XML Schema) ซึ่งไม่ได้บังคับว่าจำเป็นต้องมีไฟล์กำหนดโครงสร้างเอกสาร แต่ถ้ามีและเอกสารเอ็กซ์เอ็มแอล มีรูปแบบถูกต้องตาม คีทีดี หรือ เอ็กซ์เอ็มแอลสครีม่า จะถือว่าเอกสารเอ็กซ์เอ็มแอลนั้นมีคุณสมบัติถูกต้อง (valid)
- มีความกะทัดรัด เข้าใจง่ายและใช้ประโยชน์ได้กว้างขวาง
- สามารถใช้ได้หลายภาษาผสมกัน เนื่องจากเอ็กซ์เอ็มแอล สนับสนุน ยูนิโคด(UNICODE)
- ได้รับการสนับสนุนในโปรแกรมระบบฐานข้อมูลหลายๆค่าย สามารถดึงข้อมูลจากฐานข้อมูล (Database) มาอยู่ในรูปของเอ็กซ์เอ็มแอลได้

2.2.3 ประโยชน์ของเอ็กซ์เอ็มแอล

ประโยชน์สำคัญของ เอ็กซ์เอ็มแอล ความจริงแล้วมีเพียงอย่างเดียวคือ การเป็นแม่แบบหรือต้นแบบในการนิยามข้อมูลเพื่อใช้งานต่างๆ ซึ่งเป็นประโยชน์ที่ครอบคลุมที่สุด แต่สามารถแจกแจงได้หลายประการได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ใช้สำหรับสร้างข้อมูลที่สามารถอธิบายความหมายของตัวเองได้ (Self-Describe Data) จากการที่เราสามารถกำหนดแท็ก มาอธิบายข้อมูลที่อยู๋ภายในแท็กเองได้ ทำให้ข้อมูลมีความหมายในตัวเอง และสามารถเขียนโปรแกรมมาดึงข้อมูลไปใช้งานง่าย และคนทั่วไปก็อ่านได้เข้าใจ

2. ใช้สำหรับการแลกเปลี่ยนข้อมูล (Data Exchange) ด้วยความที่เป็นไฟล์ข้อความธรรมดา ทำให้เอกสารเอ็กซ์เอ็มแอล เป็นภาษากลางที่สามารถใช้ได้ทุกแพลตฟอร์ม จึงไม่แปลกที่เราจะสามารถแลกเปลี่ยนข้อมูลข้ามแพลตฟอร์มได้

3. เป็นรูปแบบข้อความในการสื่อสาร (Messaging Format) ระหว่างแอปพลิเคชัน หรือ โปรแกรมเอ็กซ์เอ็มแอลเป็นรูปแบบการสื่อสารระหว่างองค์ประกอบต่างๆ ตามแนวคิดของเว็บเซอร์วิส นอกจากเอ็กซ์เอ็มแอลแล้ว ยังมีมาตรฐานต่างๆที่เป็นส่วนสำคัญในการทำงานของเว็บเซอร์วิส เช่น โซบ หรือ ซิมเปิลออปเจกต์เอกเซสโปรโตคอล และ ยูดีดีไอ หรือ ยูนิเวอร์ซัลเดสคริปต์ซันคิสต์ฟเวอร์เอนด์อินทิเกรชัน ซึ่งทั้ง โซบ และ ยูดีดีไอ ล้วนมีพื้นฐานมาจาก เอ็กซ์เอ็มแอล เช่นกัน

4. ประโยชน์ในเชิงเทคโนโลยีอินเทอร์เน็ตและการพัฒนาเว็บ

5. เป็นรากฐานของภาษาใหม่ๆในการพัฒนาเว็บ ภาษาใหม่ๆนี้ได้แก่ ภาษา เอ็กซ์เอชทีเอ็มแอล (XHTML), แมทเอ็มแอล (MathML คือ กลุ่มของแท็กเพื่อใช้นิยามเครื่องหมายในทางคณิตศาสตร์ชั้นสูง), วีเอ็มแอล (VML คือภาษาที่ใช้วาดรูปกราฟิกเพื่อแสดงผ่านเว็บเบราว์เซอร์) และ ดับบีวีเอ็มแอล (WML คือภาษาที่ใช้ในการสร้าง เว็บไซต์ (Wap Site)) เป็นต้น

6. ใช้ในแวดวงธุรกิจแบบบีทูบี (B2B = Business to Business) กรณีนี้จะต้องใช้ภาษาเฉพาะ อย่างเช่น ซีเอ็มเอ็กซ์แอล (cXML = Commerce XML), เอ็กซ์ซีบีแอล (xCBL = XML Common Business Language) เป็นต้น โดยมีแท็กที่ใช้สนับสนุนการจัดการเกี่ยวกับแคตตาล็อกสินค้า และธุรกรรมเกี่ยวกับอีคอมเมิร์ซ (E-Commerce)

นอกจากนี้ ต่างประเทศยังมีโครงการนำ เอ็กซ์เอ็มแอลมา ใช้แทนระบบ อีดีไอ หรือ อิเล็กทรอนิกส์ดาต้าอินเทอร์เชนจ์ ซึ่งเป็นการแลกเปลี่ยนเอกสารอิเล็กทรอนิกส์ระหว่างองค์กร

2.2.4 ความแตกต่างระหว่างเอชทีเอ็มแอล และ เอ็กซ์เอ็มแอล

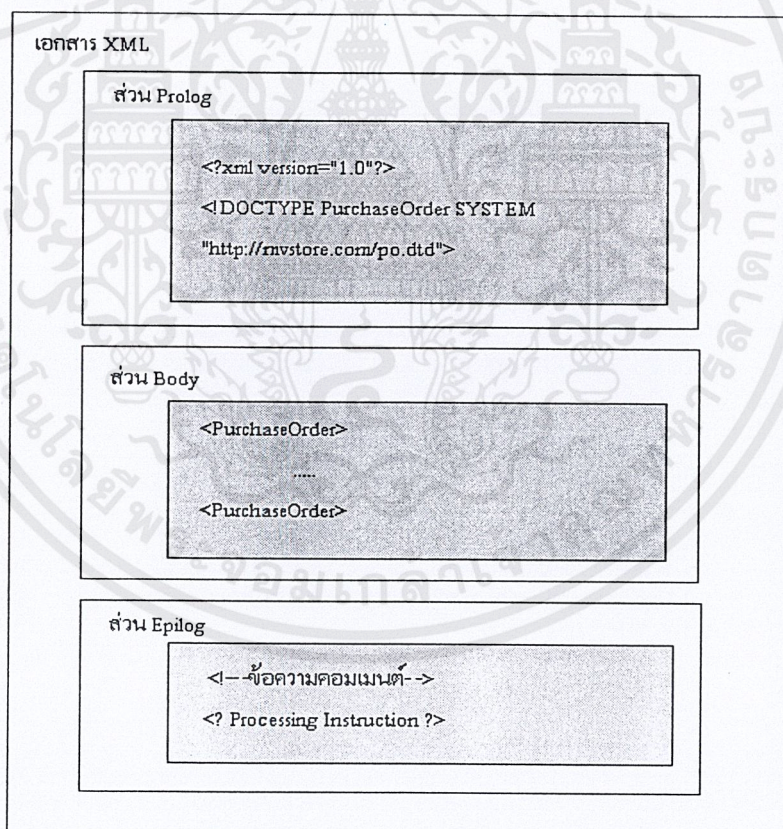
ถึงแม้ เอ็กซ์เอ็มแอล จะพัฒนามาจาก เอชทีเอ็มแอล แต่ทั้ง เอชทีเอ็มแอล และ เอ็กซ์เอ็มแอล ก็ล้วนมีวัตถุประสงค์หลักที่แตกต่างกัน เอชทีเอ็มแอล นั้นใช้เพื่อการสร้างเว็บ แต่ เอ็กซ์เอ็มแอล นั้นมีจุดประสงค์เพื่อใช้นิยามข้อมูล เอ็กซ์เอ็มแอล ไม่สามารถแสดงผลได้ด้วยตัวของมันเอง เพราะฉะนั้น เอ็กซ์เอ็มแอล ก็ไม่สามารถที่จะมาแทน เอชทีเอ็มแอลได้ และในบางกรณีทั้ง เอ็กซ์เอ็มแอล และ เอชทีเอ็มแอล ก็สามารถทำงานร่วมกันได้อย่างดี มีการพัฒนาภาษา เอ็กซ์เอชทีเอ็มแอล หรือ เอ็กซ์เทนซิเบิลไฮเปอร์เท็กซ์มาร์กอัปแลงแวกจากข้อดีของภาษา เอชทีเอ็มแอล และ เอ็กซ์เอ็มแอล รวมกันเพื่อใช้แสดงผลข้อมูลผ่านเว็บ แบบเดียวกับ

เอชทีเอ็มแอล โดยที่สามารถใช้แท็กของ เอชทีเอ็มแอล ได้ แต่ต้องทำตามกฎของ เอกซ์เอ็มแอล คือเขียนให้อยู่ในรูปแบบที่เป็นมาตรฐาน

2.2.5 โครงสร้างเอกซ์เอ็มแอล

โครงสร้างของเอกสารเอกซ์เอ็มแอล ประกอบด้วย 3 ส่วน

- โปรล็อก (Prolog) แบ่งเป็น ดีคลิเรชัน (Declaration) และ ค็อกคิวเมนต์ไทป์ดีคลิเรชัน (Document Type Declaration)
- บอดี้ (Body) เป็นส่วนของเนื้อเอกสารจริงๆ ได้แก่ข้อความหรือข้อมูลในเอกสาร และ แท็ก ที่นิยามข้อความหรือข้อมูลเหล่านั้น
- อีพิล็อก (Epilog) คือส่วนที่เป็นข้อความจำพวกคอมเมนต์ (Comment) และ พีไอ (PI) หรือ โปรเซสซิ่งอินสตรัคชัน (Processing Instruction) จริงๆแล้วส่วนนี้ไม่จำเป็นต้องอยู่บริเวณล่างสุดของเอกสารดังในภาพ แต่คอมเมนต์ และ พีไอ จะแทรกอยู่ในส่วนบอดี้ของเอกสาร แต่ในภาพนี้แยกไว้ด้านล่างเพื่อให้เห็นว่าในเอกสาร เอกซ์เอ็มแอล หนึ่งๆ มีองค์ประกอบ 3 ส่วน



รูปที่ 2-3 ส่วนประกอบของเอกสารเอกซ์เอ็มแอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.6 ข้อควรระวังในการเขียนเอ็กซ์เอ็มแอล

สิ่งที่สำคัญในการเขียน เอ็กซ์เอ็มแอล คือ คุณสมบัติเวลฟอร์มเนส (well-formedness) ซึ่งเกี่ยวข้องกับ การใช้แท็ก และเอลเลอเมนต์ต่างๆอย่างถูกต้อง ไม่ว่าจะเป็นการระบุแท็ก เปิด-ปิด การระบุค่าแอททริบิวต์ (Attribute) การตั้งชื่อแท็ก เป็นต้น การที่ เอ็กซ์เอ็มแอล มีกฎเคร่งครัดเช่นนี้ก็เพื่อให้เอกสารเป็นระเบียบแบบแผน สามารถสร้างตัวแปรภาษา หรือ เอ็กซ์เอ็มแอลพาร์เซอร์ ได้ง่าย ไม่ต้องมากังวลว่าจะจัดการกับเอกสารที่ ระบุแท็ก ไม่ถูกต้องอย่างไร เพราะฉะนั้น เอ็กซ์เอ็มแอลพาร์เซอร์ ทุกโปรแกรม จะทำงานกับเอกสาร เอ็กซ์เอ็มแอลที่เวลฟอร์มเท่านั้น ถ้าเอกสารไม่อยู่ในรูปเวลฟอร์ม จะเกิดข้อผิดพลาด (Error) ซึ่งการจะเขียนเอกสาร เอ็กซ์เอ็มแอล ให้มีคุณสมบัติเวลฟอร์มเนส มีข้อควรระวังต่อไปนี้

- เอกสารเอ็กซ์เอ็มแอล หนึ่งๆ จะมีรูทเอลเลอเมนต์ (root element) ได้เพียงอันเดียวเท่านั้น ซึ่งทำหน้าที่คลุมเอลเลอเมนต์อื่นๆทั้งหมด
- แท็กเปิด และ แท็กปิดต้องเหมือนกัน ต่างกันแค่เพียงในแท็กปิดต้องมีเครื่องหมาย (/) นำหน้าชื่อแท็กเท่านั้น
- ห้ามระบุแท็ก เหลื่อมซ้อนกัน (Overlap) คือ แท็กที่เปิดก่อน ต้องปิดทีหลัง
- ชื่อแท็กมีคุณสมบัติ เคสเซ็นซิทีฟ (case-sensitive) คือ ตัวอักษรพิมพ์เล็ก และพิมพ์ใหญ่ถือว่าแตกต่างกันดังนั้นแท็ก <Name>, <NAME>, <name> ถือว่าแตกต่างกัน
- สำหรับแท็กว่าง (Empty Tag) คือ แท็กที่ไม่มีข้อมูลข้างใน มีวิธีเขียนได้ 2 แบบ แบบหนึ่งคือ <title> และ อีกแบบคือ <title/> ส่วนมากจะนิยมแบบแรกมากกว่า
- ค่าของแอททริบิวต์ ต้องอยู่ในเครื่องหมายคำพูดแบบ (“” หรือ ‘’) อย่างใดอย่างหนึ่งเช่น <book isbn = "123-45678-9-0" /> กรณีที่ในแอททริบิวต์ มีเครื่องหมายคำพูดแบบใดแบบหนึ่งอยู่ เราต้องเลี่ยงไปใช้เครื่องหมายคำพูดอีกแบบในการครอบค่าแอททริบิวต์
- ในภาษาเอ็กซ์เอ็มแอล มีอักขระซึ่งสงวนไว้ 5 ตัว เพราะต้องใช้เป็นส่วนประกอบตามโครงสร้างของภาษา ดังนั้นหากเนื้อความเอกสารจำเป็นต้องมีอักขระเหล่านี้ ก็ต้องพิมพ์เป็นชุดอักขระพิเศษแทน ซึ่งอักขระชุดพิเศษนี้เรียกว่า เอนทิตีรีเฟอเรนซ์ (Entity Reference) ซึ่งอักขระ และ เอนทิตีรีเฟอเรนซ์ ทั้ง 5 มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อักขระพิเศษ	Entity Reference
<	<
&	&
>	>
“	"
’	'

รูปที่ 2- 4 อักขระชุดพิเศษ และ เอนทิตีรีเฟอเรนส์

การตั้งชื่อแท็ก มีหลักเกณฑ์ดังนี้

- ชื่อแท็ก ต้องขึ้นต้นด้วยตัวอักษร หรือเครื่องหมายอันเดอร์สกออร์ underscore(_) เท่านั้น
- ตัวถัดไปต้องเป็นตัวอักษร ตัวเลข เครื่องหมายจุด(.) , เครื่องหมายขีดทึบ(-), เครื่องหมายอันเดอร์สกออร์ underscore(_), หรือเครื่องหมายโคลอน(:) เท่านั้น
- อักษร 3 ตัวแรกห้ามเป็นคำว่าเอ็กซ์เอ็มแอล ไม่ว่าจะเป็นตัวเล็กหรือใหญ่ เพื่อสงวนเอาไว้ใช้ในอนาคต
- การกำหนดแอททริบิวทของเอ็กซ์เอ็มแอลดีแคล์เรชั่น จะต้องเรียงตามลำดับที่กำหนดไว้เท่านั้น คือ เวอร์ชัน (Version) ก่อน แล้วจึงตามด้วยเอนโค้ดดิ้ง (Encoding) แล้วสุดท้ายจึงเป็นแอททริบิวทสแตนดอลน (Attribute Standalone)

2.2.7 การกำหนดโครงสร้างของเอกสารเอ็กซ์เอ็มแอล

เนื่องจากการที่เอ็กซ์เอ็มแอลยอมให้ผู้ใช้สร้างแท็กขึ้นมาเอง ผู้ที่เกี่ยวข้องกับการใช้งานเอกสารเอ็กซ์เอ็มแอลทุกฝ่ายจึงต้องมีการตกลงกันก่อนว่าจะเก็บข้อมูลส่วนไหนไว้ในแท็กอะไร เมื่อได้ข้อสรุปแล้วจึงยึดถือปฏิบัติในการกำหนดโครงสร้างเอ็กซ์เอ็มแอล (หรือที่เรียกว่าสคริปมา) โดยมีวิธีกำหนดโครงสร้างเอกสาร 2 วิธี ได้แก่ ดีทีดี และ เอ็กซ์เอ็มแอลสคริปมา เอกสารที่มีรูปแบบตามโครงสร้างที่กำหนดไว้ ไม่ว่าจะวิธีใดก็ได้แล้วแต่ เรียกว่าเอกสาร เอ็กซ์เอ็มแอล นี้มีคุณสมบัติเฉลี่ย

ความจริงแล้วเอกสารเอ็กซ์เอ็มแอล ไม่ต้องมี ดีทีดี หรือ เอ็กซ์เอ็มแอลสคริปมา กำกับก็สามารถใช้งานได้เช่นกัน แต่เอกสารเอ็กซ์เอ็มแอล จะต้องมีคุณสมบัติคุณสมบัติเวลฟอร์มเนส อยู่แล้ว ส่วนคุณสมบัติเฉลี่ยนั้น เป็นเพียงส่วนเสริมซึ่งถ้ามีก็จะดีมาก ถ้าหน่วยงานที่เกี่ยวข้องกับเอกสารนั้นไว้วางใจซึ่งกันและกัน ก็อาจไม่สนใจคุณสมบัติข้อนี้ ทำให้เอกสารเอ็กซ์เอ็มแอล ไม่จำเป็นต้องมี ดีทีดี หรือ เอ็กซ์เอ็มแอลสคริปมา กำกับก็ได้ เราจะสรุปความสำคัญของ ดีทีดี และ เอ็กซ์เอ็มแอลสคริปมา ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เป็นการกำหนดรูปแบบโครงสร้างของเอกสาร เอ็กซ์เอ็มแอล หรือเรียกว่า คาด้าโมเดล (Data Model) ว่าประกอบด้วยแท็กอะไรบ้าง, มีแอททริบิวต์อะไรบ้าง, ค่าของแอททริบิวต์เป็นอะไรได้บ้าง, ลักษณะการซ้อนแท็กเป็นอย่างไรได้บ้าง เป็นต้น
- เป็นสัญญาระหว่างองค์กร ที่จะแลกเปลี่ยนข้อมูลกันในรูปของเอ็กซ์เอ็มแอล เพื่อให้เอกสารที่สร้างขึ้นมามีรูปแบบตรงตามที่ตกลงกันไว้ และจะได้สามารถนำเอาเอกสารเอ็กซ์เอ็มแอลนั้นไปใช้งานอย่างถูกต้อง

2.2.8 ข้อดีและข้อเสียของดีทีดี และ เอ็กซ์เอ็มแอลสกริม่า

ดีทีดีนั้นเข้าใจได้ง่ายกว่าเอ็กซ์เอ็มแอลสกริม่า จากรูปไฟล์ SI.dtd ซึ่งเป็นไฟล์ดีทีดี ที่เอาไว้จัดรูปแบบโครงสร้างไฟล์ test.xml บรรทัดแรกประกาศว่า รุทเอลเลอเมนต์ที่มีโนดลูก (node) เป็น สติวเด้น (student) ซึ่งเอลเลอเมนต์สติวเด้น (element student) ก็มีแอททริบิวต์สติวเด้น (attribute studentid) ซึ่งมีชนิดเป็น ไอดี (ID) และ #REQUIRED แปลว่า ทุกๆ เอลเลอเมนต์สติวเด้น จะต้องมี แอททริบิวต์สติวเด้น บรรทัดที่ 3 เป็นการบอกว่า แอททริบิวต์สติวเด้น มี เอลเลอเมนต์ย่อยลงไปอีกเป็น พรีฟิก (prefix) , เอฟเนม (fname) และ เอลเนม (lname) ส่วนบรรทัดที่เหลือเป็นการบอกชนิดข้อมูลของเอลเลอเมนต์ย่อยเหล่านี้ว่าเป็นข้อมูลชนิดพีซีดาต้า (PCDATA) ย่อมาจาก พาสคาร์เรคเทอร์ดาต้า (Parsed Character Data) หมายความว่าเนื้อหาส่วนที่เป็นข้อความนี้เอาไว้ให้ โปรแกรมเอ็กซ์เอ็มแอลพาร์เซอร์ อ่านเพื่อประมวลผล ดีทีดี มีข้อจำกัดหลายอย่าง เช่น รับรองชนิดของข้อมูล (Data Type) ได้น้อยกว่า, ไม่ได้ใช้ไวยากรณ์ของเอ็กซ์เอ็มแอล เมื่อเราต้องการแปลความหมายของดีทีดี เราจึงไม่สามารถแปลพร้อมกับเอกสารเอ็กซ์เอ็มแอล โดยใช้เอ็กซ์เอ็มแอลพาร์เซอร์ได้

```
<!ELEMENT StudentList (student)>
<!ATTLIST student studentid ID #REQUIRED>
<!ELEMENT student (prefix,fname,lname)>
<!ELEMENT prefix (#PCDATA)>
<!ELEMENT fname (#PCDATA)>
<!ELEMENT lname (#PCDATA)>
```

รูปที่ 2-5 ตัวอย่าง SI.dtd

ส่วนเอ็กซ์เอ็มแอลสกริม่าจะรองรับชนิดของข้อมูลได้มากกว่าและหลากหลายกว่า, สามารถกำหนดรูปแบบข้อมูลเองได้, สร้างชนิดข้อมูลใหม่ได้, กำหนดจำนวนและลำดับของ เอลเลอเมนต์ลูก ได้ รวมทั้งใช้ไวยากรณ์ของเอ็กซ์เอ็มแอล และที่สำคัญที่สุดคือเอ็กซ์เอ็มแอลสกริม่า สนับสนุน เนมสเปซ (namespace) เพื่อใช้สำหรับป้องกันความสับสนในการระบุชื่อองค์ประกอบใดๆ ภายในเอกสารเอ็กซ์เอ็มแอลคนละไฟล์ แต่ใช้

ชื่อเดียวกัน เพราะเป็นไปได้ว่าเอกสารเอ็กซ์เอ็มแอล ที่ใช้ในงานที่ต่างกัน อาจมีชื่อเอลเมนต์เหมือนกัน ชื่อชนิดข้อมูลเหมือนกัน แล้วอาจถูกเรียกใช้ภายในเอกสารเอ็กซ์เอ็มแอลไฟล์เดียวกัน แต่ เอ็กซ์เอ็มแอลสคริปต์ ก็มีชื่อเสียคือเรียนรู้ได้ยากกว่าที่ดี

2.2.9 การแสดงผลเอกสารเอ็กซ์เอ็มแอลด้วยเอ็กซ์เอสแอล (XSL)

เอ็กซ์เอสแอล หรือ เอ็กซ์เทนซิเบิลสไตลชีทแลงแกจ (Extensible Stylesheet Language) เป็นภาษาที่ใช้สำหรับกำหนดรูปแบบการแสดงผลเอกสารเอ็กซ์เอ็มแอล ไปเป็นเอกสารรูปแบบอื่นๆ เช่น เอกซ์เอ็มแอล ดับบีวีเอ็มแอล หรือแม้แต่การเปลี่ยนจากเอ็กซ์เอ็มแอล รูปแบบหนึ่งไปเป็นเอ็กซ์เอ็มแอล รูปแบบอื่นๆ เอ็กซ์เอสแอลมีความสำคัญเพราะโดยปกติแล้ว เอกสารเอ็กซ์เอ็มแอล ก็คือไฟล์ธรรมดาที่มีแท็กคร่อมข้อความอยู่เท่านั้น เราจึงต้องอาศัยเอ็กซ์เอสแอลมาช่วยในการนำเอกสารเอ็กซ์เอ็มแอล มาแสดงผลไม่ว่าจะผ่านทางบราวเซอร์หรืออะไรก็ตาม

ในการแสดงผลเอกสารเอ็กซ์เอ็มแอล นั้นมีทางเลือกสองทาง ทางแรกคือแสดงข้อมูลเอ็กซ์เอ็มแอล ทั้งส่วนแท็กและส่วนข้อมูลด้วย อินเทอร์เน็ตเอ็กซ์พลอเรอร์ 5 (Internet Explorer 5) ซึ่ง ไออี 5 (IE5) จะสร้างเอกสารแบบอินเทอร์แอคทีฟ (Interactive) ที่อนุญาตให้บราวเซอร์ทำงานกับข้อมูล, ยูน และ ขยายรายการภายในแท็กได้

ทางเลือกอีกทางคือ การใช้ เอ็กซ์เอสแอล เพื่อเปลี่ยนข้อมูล เอ็กซ์เอ็มแอล ให้เป็นเอกซ์เอ็มแอล เมื่อใช้ เอ็กซ์เอสแอล เราสามารถฝังข้อมูลเข้าไปในไฟล์ เอ็กซ์เอ็มแอล โดยตรง หรือใส่ข้อมูลเข้าไปในสไตลชีทเอ็กซ์เอ็มแอล (style sheet XML) แยกต่างหากโดย เอ็กซ์เอสแอล มีความสามารถเหมือนกับการใช้ ซีเอสเอส กับ เอกซ์เอ็มแอล แต่ยังช่วยให้เราสามารถแยกข้อมูลออกจากไฟล์เอ็กซ์เอ็มแอล เพื่อการแสดงผล นอกจากเอ็กซ์เอสแอล แล้ว เราก็สามารถใช้ ซีเอสเอส ได้ เช่นเดียวกัน ซีเอสเอส สามารถทำความเข้าใจได้ง่าย แต่มีข้อจำกัดในการจัดรูปแบบการแสดงผล ส่วนเอ็กซ์เอสแอล สามารถกำหนดลักษณะการแสดงผลได้หลากหลายกว่า เพราะเป็นการเปลี่ยนรูปแบบข้อมูล เช่น กรองข้อมูลได้ (Filter), เรียกลำดับข้อมูลได้ (Sort) และ จัดรูปแบบการแสดงผลได้ตามต้องการ

ทุกครั้งที่เปิดปิดดูเอกสารเอ็กซ์เอ็มแอลธรรมดา ด้วยไออี5 เราไม่จำเป็นต้องกำหนดสไตลชีทก็ได้ เพราะ ไออี5 มีสไตลชีทพิเศษฝังอยู่ข้างในอยู่แล้ว แต่ถ้าเรากำหนด สไตลชีท เข้าไปเอง ไม่ว่าจะแบบ ซีเอสเอส หรือ เอ็กซ์เอสแอล ก็ตามบราวเซอร์ ก็จะใช้ สไตลชีทของเราแทนของเราแทน ตามหลักการโอเวอร์ไรด์ (Override)

```
<?xml version=" "1.0encoding="windows-"874?>
<xsl:stylesheet xmlns:xsl="http://www.w.3org/TR/WD-xsl">
<xsl:template match="/">
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    รายชื่อนักเรียน<hr/>
    <xsl:for-each select="StudentList/student">
    <p> <xsl:value-of select="fname"/>
        <xsl:value-of select="lname"/> </p>
    </xsl:for-each>
</xsl:template>
</xsl:stylesheet>

```

รูปที่ 2-6 ตัวอย่างไฟล์ *slstyle.xml*

2.2.10 การแปลความหมายของ เอกซ์เอ็มแอลด้วยเอกซ์เอ็มแอลพาร์เซอร์

ในการพัฒนาแอปพลิเคชันโดยมีการนำเอกซ์เอ็มแอล ไปใช้งานนั้น สิ่งหนึ่งที่ผู้พัฒนาจำเป็นต้องทำคือการพาส (Parse) เอกสารเอกซ์เอ็มแอล เนื่องจากโปรแกรมมองเห็นเอกสารเอกซ์เอ็มแอล เป็นเพียงอักขระธรรมดาเท่านั้น เราจึงต้องมีการเขียนโปรแกรมเพื่อให้ แอปพลิเคชัน สามารถเข้าใจได้ว่าเอกสารเอกซ์เอ็มแอลของเรามีการจัดโครงสร้างเป็นอย่างไร โดยเราเรียกโปรแกรมนั้นว่าพาร์เซอร์

เอกซ์เอ็มแอลพาร์เซอร์ หรือ เอกซ์เอ็มแอลโพรเซสเซอร์ (XML Processor) เป็นตัวแปลเอกสาร เอกซ์เอ็มแอล ซึ่งทำหน้าที่พื้นฐานคือ อ่าน (Read) , สร้าง (Create) , แก้ไข (Update) และเข้าถึง (Manipulate) เอกสาร เอกซ์เอ็มแอล ปัจจุบันมี โปรแกรมเอกซ์เอ็มแอลพาร์เซอร์ หลาย โปรแกรม เช่น เอ็มเอสเอกซ์เอ็มแอลของไมโครซอฟต์ เป็นโปรแกรมที่พ่วงมากับ ไออี5 ขึ้นไป สร้างโดยเทคโนโลยีคอม (COM = Component Object Model) , เอกซ์เอ็มแอลโฟร์เจ (XML4J = XML for Java) ของไอบีเอ็ม สร้างด้วยภาษาจาวา (Java Language) และ เจเอเอกซ์พี (JAXP = Java API for XML Parsing) ของบริษัทซัน (Sun) ซึ่งสร้างโดยภาษาจาวา เป็นต้น จากตัวอย่างที่ผ่านมา เราสามารถใช้โปรแกรม ไออี5 ขึ้นไป กับเอกสารเอกซ์เอ็มแอลของเราได้ก็เพราะว่าโปรแกรมนี้มีการฝัง พาร์เซอร์ ชื่อ เอ็มเอสเอกซ์เอ็มแอล เอาไว้ในตัวอยู่แล้ว เราจึงไม่ต้องทำการกำหนด พาร์เซอร์ เหมือนเวลาเราเขียนโปรแกรมทั่วไป

การจำแนกชนิดของ เอกซ์เอ็มแอลพาร์เซอร์ แบ่งได้เป็นหลายวิธี แต่หลักเกณฑ์ที่นิยมกันมี 2 วิธี คือ

1. เอกซ์เอ็มแอลพาร์เซอร์ แบบ เวลิดเดตติ้ง (Validating) และ แบบนอนเวลิดเดตติ้ง (Non-Validating) ซึ่งหมายถึง พาร์เซอร์ ที่มีตรวจสอบคุณสมบัติเวลฟอร์มเนส และ คุณสมบัติเวลิดของเอกสาร ส่วนเอกซ์เอ็มแอลพาร์เซอร์ แบบนอนเวลิดเดตติ้ง หมายถึง เอกซ์เอ็มแอลพาร์เซอร์ ที่ไม่ตรวจสอบคุณสมบัติเวลิด และสนใจเฉพาะคุณสมบัติเวลฟอร์มเนส ถึงแม้ว่า เอกซ์เอ็มแอลพาร์เซอร์แบบเวลิดเดตติ้ง จะมีการตรวจสอบที่มากกว่า ก็ไม่ได้หมายความว่าดีกว่าเสมอไป ในบางกรณีก็ควรใช้ เอกซ์เอ็มแอลพาร์เซอร์ แบบนอนเวลิดเดตติ้ง เพราะว่า

- ทำงานเร็วกว่า เพราะไม่ต้องมาคอยตรวจสอบ โครงสร้างของเอกสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ในกรณีที่มีมั่นใจอยู่แล้วว่าเอกสารนั้นอยู่ในรูปแบบที่ผลิต
 - ต้องการแค่ค้นหาแท็กบาง แท็กในเอกสารนั้นเท่านั้น ไม่ได้ต้องการตรวจสอบคุณสมบัติแวดล้อม โปรแกรมเอ็กซ์เอ็มแอลพาร์เซอร์ ส่วนใหญ่เป็นแบบนอนเวลิดเคตติ้ง เพราะว่าจะสร้างได้ง่ายกว่า
2. เอ็กซ์เอ็มแอลพาร์เซอร์ แบบที่รองรับ คือคคิวเมนต์ออบเจกต์โมเดลลิ่ง (DOM = Document Object Model) และแบบที่รองรับที่เรียกว่า แซก หรือ ซิมเปิ้ลเอพีไอฟอร์เอ็กซ์เอ็มแอล (SAX – Simple API for XML) การแปลเอกสารเอ็กซ์เอ็มแอล ต้องอาศัยโปรแกรม เอ็กซ์เอ็มแอลพาร์เซอร์ ซึ่งโปรแกรมจำพวกนี้มีวิธีการดึงข้อมูลจากเอกสารเอ็กซ์เอ็มแอล หมายถึง แต่ที่นิยมกันมากมี 2 วิธี ก็คือ คิวเมนต์ออบเจกต์โมเดลลิ่ง และ ซิมเปิ้ลเอพีไอฟอร์เอ็กซ์เอ็มแอล ซึ่งมีลักษณะต่างกันให้ เลือกใช้ตามความเหมาะสมของลักษณะงาน

2.2.11 การแปลเอกสารโดยใช้คิวเมนต์ออบเจกต์โมเดลลิ่ง

คิวเมนต์ออบเจกต์โมเดลลิ่ง เป็น เอพีไอ (API = Application Programming Interface) สำหรับเอกสารเอ็กซ์เอ็มแอล ซึ่งจะกำหนดว่าเอกสารเอ็กซ์เอ็มแอล จะมีการเข้าถึง การโยกย้ายและการจัดการต่างๆ ได้อย่างไรบ้าง ด้วยวิธีการแบบคิวเมนต์ออบเจกต์โมเดลลิ่ง ผู้พัฒนาจะมาสารถเข้าถึงโครงสร้างระดับต่างๆ ของเอกสารเอ็กซ์เอ็มแอลได้ นอกจากนั้นยังสามารถเพิ่ม ลบและแก้ไขเอลเลอเมนต์ต่างๆของเอกสารได้อีกด้วย

คิวเมนต์ออบเจกต์โมเดลลิ่ง ได้รับการรับรองเป็นมาตรฐานโดย คับบิวทีซี มีหลักการคือเมื่อโปรแกรม หรือ แอปพลิเคชัน เรียกพาร์เซอร์ เพื่อทำการโหลด(Load) เอกสารเอ็กซ์เอ็มแอล เข้าสู่หน่วยความจำคอมพิวเตอร์ เมื่อเอกสารถูกโหลดเรียบร้อยแล้ว คิวเมนต์ออบเจกต์โมเดลลิ่ง ก็จะทำให้โปรแกรมสามารถดึงข้อมูลมาใช้งานได้ โดยคิวเมนต์ออบเจกต์โมเดลลิ่ง จะนำข้อมูลของเอกสารเอ็กซ์เอ็มแอลมาวางเป็นโครงสร้างทรีไดอะแกรม (Tree Diagram) ในหน่วยความจำของเครื่องคอมพิวเตอร์ ซึ่งทรีไดอะแกรม จะประกอบด้วยข้อมูลต่างๆในเอกสารเอ็กซ์เอ็มแอล เช่น เอลเลอเมนต์ และ แอททริบิวต์ต่างๆ เป็นต้น โดยแต่ละเอลเลอเมนต์ก็จะเป็นโหนดในทรีไดอะแกรม ดังนั้นการเข้าถึงเอกสารเอ็กซ์เอ็มแอลจะรู้ว่าแต่ละเอลเลอเมนต์มีค่าเป็นอะไร ก็คือการทราเวอร์ส (Traverse) ไปตามโหนดต่างๆในทรีไดอะแกรม จึงเรียกได้ว่า เอ็กซ์เอ็มแอลพาร์เซอร์แบบที่รองรับ คิวเมนต์ออบเจกต์โมเดลลิ่งเป็นทรีเบสพาร์เซอร์ (Tree-Base Parser)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีของค็อกคิวเมนที่ออบเจกต์โมเดลลิ่ง

- หลังจากโปรแกรมทำการพาข้อมูลแล้ว ก็จะเก็บทรีไดอะแกรมไว้ในหน่วยความจำตลอดจนกว่าเราจะสั่งยกเลิก เหมาะกับกรณีที่ต้องเข้าถึงข้อมูลหลายครั้ง เช่น ดึงข้อมูล แก้ไขข้อมูล หรือกรณีที่เราต้องการรู้ข้อมูลหลายอย่างเกี่ยวกับข้อมูลในเอกสารเอ็กซ์เอ็มแอล เพราะเมื่อเก็บเอกสารไว้ในหน่วยความจำแล้ว เราก็สามารถเข้าถึงข้อมูลในทรีไดอะแกรมได้อย่างรวดเร็ว
- กรณีที่ต้องมีการเคลื่อนย้ายข้อมูลบางส่วนของเอกสาร เช่น มีการเรียงข้อมูลใหม่ (Sort) เพราะการทำงานลักษณะนี้จะเกิดผลกระทบกับทั้งเอกสาร จึงควรที่จะโหลดเอกสารทั้งเอกสารเก็บไว้ในหน่วยความจำ
- ค็อกคิวเมนที่ออบเจกต์โมเดลลิ่งสามารถสร้าง, เขียน และ อัปเดตข้อมูลเอ็กซ์เอ็มแอล จัดเก็บไว้ในไฟล์ และ สตริม (Stream)
- เราสามารถเรียกใช้ข้อมูลแบบสุ่ม (Random) เนื่องจากข้อมูลทั้งหมดอยู่ในหน่วยความจำ
- ข้อมูลที่สร้างเป็นทรีทำให้ข้อมูลเกิดความสัมพันธ์ ทำให้การเรียกใช้ทำได้ง่าย

ข้อเสียของค็อกคิวเมนที่ออบเจกต์โมเดลลิ่ง

- เนื่องจากวิธีของค็อกคิวเมนที่ออบเจกต์โมเดลลิ่งจะโหลดเอกสารเอ็กซ์เอ็มแอล ทั้งหมดมาเก็บในหน่วยความจำ กรณีที่เอกสารมีขนาดใหญ่ก็จะเปลืองเนื้อที่หน่วยความจำมาก การทำงานเป็นไปได้ช้า และในบางกรณีอาจไม่มีเนื้อที่เหลือพอจะทำงานได้

2.2.12 การแปลเอกสารโดยใช้ซิมเบิลเอพีไอฟอว์เอ็กซ์เอ็มแอล

ซิมเบิลเอพีไอฟอว์เอ็กซ์เอ็มแอลไม่ได้เป็นมาตรฐานของหน่วยงานใด แต่เป็นวิธีที่นักพัฒนาคนหนึ่งคิดค้นขึ้น และนำไปใช้กันมากจนได้รับการยอมรับทั่วไป ปัจจุบันเอ็กซ์เอ็มแอลพาร์เซอร์ จำนวนมากล้วนแต่รองรับซิมเบิลเอพีไอฟอว์เอ็กซ์เอ็มแอล กันแล้ว

ซิมเบิลเอพีไอฟอว์เอ็กซ์เอ็มแอลจะไม่โหลดเอกสารเอ็กซ์เอ็มแอล เข้ามาในหน่วยความจำจึงไม่มีการสร้างภาพรวมของเอกสารไว้ก่อนว่าเอกสารนั้นมีโครงสร้างเป็นอย่างไร วิธีการของมันคือเมื่อแอปพลิเคชันมีการร้องขอมา มันจึงจะอ่านเอกสารเอ็กซ์เอ็มแอลไล่ตั้งแต่ต้นเอกสารและตัวพาร์เซอร์จะสร้างอีเวนต์ (Event) ให้กับจุดต่างๆที่สำคัญของเอกสารทุกๆจุด เช่นทุกๆ เอลเลอเมนต์ที่เปิด เอลเลอเมนต์ที่ปิด ทุกๆแอททริบิว เป็นต้น ลักษณะการทำงานเช่นนี้เรียกว่า อีเวนต์ไดรเวนพาร์เซอร์ (Event-Driven Parser)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในทางปฏิบัติแล้วซิมเบิลเอพีไอฟอร์เอ็กซ์เอ็มแอล มักไม่ได้มีการแปลเอกสารทั้งเอกสาร เมื่อเราได้ข้อมูลที่ต้องการแล้วเราก็จะหยุด วิธีการแปลแบบซิมเบิลเอพีไอฟอร์เอ็กซ์เอ็มแอล จึงทำงานได้รวดเร็วและเขียนโปรแกรมได้ง่าย

ข้อดีของซิมเบิลเอพีไอฟอร์เอ็กซ์เอ็มแอล

- ไม่เปลืองหน่วยความจำ เพราะไม่ได้โหลดทั้งเอกสารมาเก็บไว้เหมือน ค็อกคิวเมนท้ออบเจกต์โมเดลลิ่ง จึงเหมาะกับกรณีทีเอกสารเอ็กซ์เอ็มแอล มีขนาดใหญ่ และเครื่องคอมพิวเตอร์มีหน่วยความจำไม่มาก
- ซิมเบิลเอพีไอฟอร์เอ็กซ์เอ็มแอลเข้าถึงข้อมูลได้ง่ายกว่า ค็อกคิวเมนท้ออบเจกต์โมเดลลิ่ง เพราะการทำงานของ แชก จะมองเพียงจุดเริ่มต้นและจุดสิ้นสุดของเอกสารและของ เอลเลอเมนต์ และจะสนใจเฉพาะสิ่งที่ต้องการเท่านั้น จึงเหมาะกับกรณีต้องการใช้ข้อมูลจากเอลเลอเมนต์เพียงไม่กี่ตัวในเอกสาร ซึ่งถ้าใช้วิธี ค็อกคิวเมนท้ออบเจกต์โมเดลลิ่ง จะเป็นงานที่กินแรงกินความจำเป็นอย่างมาก
- สามารถเขียน โปรแกรม ได้ง่ายกว่า เพราะมองข้อมูลในลักษณะเส้นตรง

ข้อเสียของซิมเบิลเอพีไอฟอร์เอ็กซ์เอ็มแอล

- ไม่สามารถบอกความสัมพันธ์ของข้อมูลได้
- เราไม่สามารถใช้ข้อมูลแบบลุ่มได้ เพราะไม่มีข้อมูลใดเหลืออยู่ในหน่วยความจำหลังจากที่ซิมเบิลเอพีไอฟอร์เอ็กซ์เอ็มแอล ได้ประมวลผลและสร้างอีเวนต์ที่มีความสัมพันธ์กันเสร็จแล้ว
- ยากที่จะสร้างฟังก์ชัน ในการค้นหาข้อมูล
- อ่านได้เพียงอย่างเดียว ไม่สามารถเขียนหรือ อัปเดตข้อมูลได้
- เป็นวิธีที่ไม่มีประสิทธิภาพถ้าต้องการเข้าถึงข้อมูลเดิมบ่อยครั้ง

2.3 ซิมเบิลออบเจกต์แอคเซสโพรโตคอล (Simple Object Access Protocol)

เนื่องจากจุดประสงค์หลักของการใช้งานเว็บเซอร์วิส เราต้องการให้แอปพลิเคชัน มีการทำงานกับแอปพลิเคชันที่ทำงานอยู่บนเครื่องอื่น โดยผ่านทางเครือข่าย ซึ่งเทคโนโลยีที่มีอยู่ปัจจุบันที่ใช้มีการสื่อสารระหว่างออบเจกต์ในระยะไกล หรือ อาร์พีซี (Remote Procedure Calls : RPC) เช่น ดีคอม (DCOM) , อีเจบี (EJB) หรือคอบร้า (COBRA) นั้นไม่ได้ถูกออกแบบมาใช้สำหรับ โพร โตคอลเอชทีทีพี

เทคนิคอาร์พีซีของเทคโนโลยีที่กล่าวข้างต้นต่างก็มีปัญหาในด้านการนำมาใช้งานในแง่ของความเข้ากันได้ของการเรียกใช้งานข้ามเทคโนโลยี เนื่องจากเป็นเทคโนโลยีเฉพาะของแต่ละค่าย (ยกเว้น คอบร้า) ผู้พัฒนาระบบจะต้องพัฒนาโปรแกรมที่มีความซับซ้อน และยังมีปัญหาในส่วนของไฟร์วอลล์ และพร็อกซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซิร์ฟเวอร์ ด้วยเนื่องจากโดยปกติเซิร์ฟเวอร์จะปิดการสื่อสารที่ไม่ใช่โปรโตคอลเอชทีทีพีออกไป เพื่อความปลอดภัยของระบบที่มีการติดต่อสื่อสารกับภายนอก

ดังนั้นทางเลือกของการสื่อสารที่จะนำมาใช้ในการให้บริการเว็บเซอร์วิส คือ ให้ทำงานอยู่บนโปรโตคอลเอชทีทีพี ซึ่งโฉบ นอกจากจะทำงานบนโปรโตคอลเอชทีทีพีแล้ว ยังเป็นมาตรฐานเปิดที่จะทำให้สามารถติดต่อสื่อสารกับคอมพิวเตอร์ที่มีความแตกต่างทางระบบปฏิบัติการ, เทคโนโลยีรวมไปถึงภาษาที่ใช้ในการพัฒนา

โฉบ หรือ ซิมเปิลอบเจกต์แอคเซสโพรโตคอล (SOAP = Simple Object Access Protocol) เป็นโปรโตคอลที่ใช้ เอ็กซ์เอ็มแอล เป็นพื้นฐาน เพื่อให้ซอฟต์แวร์คอมพิวเตอร์ และแอปพลิเคชันสามารถติดต่อกันผ่านเอชทีทีพี ซึ่งเป็นมาตรฐานอินเทอร์เน็ตโพรโตคอลได้ ซึ่งข้อดีของโฉบก็คือ โฉบไม่ขึ้นกับคอมพิวเตอร์เทคโนโลยี และภาษาการเขียนโปรแกรมใดๆ สามารถเขียนได้ง่ายและขยายเพิ่มเติมได้

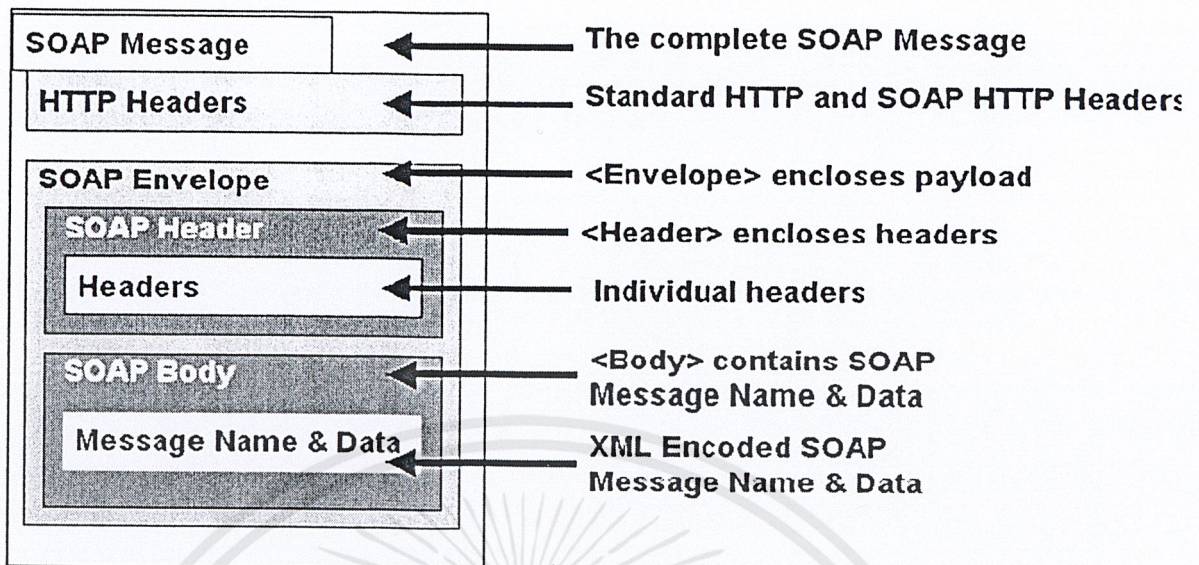
2.3.1 ส่วนประกอบของโฉบ

โฉบแมสเชจ ใช้ไวยากรณ์ของเอ็กซ์เอ็มแอล ในการสร้างประกอบด้วย 3 เอลเมนต์มาตรฐาน คือ โฉบเอนวิลลอป, โฉบเฮดเคอร์, โฉบบอดี

โฉบนั้นประกอบด้วยกัน 3 ส่วนคือ

- โฉบเอนวิลลอป (SOAP Envelop) จะเป็นส่วนสำหรับใช้ในการระบุสิ่งที่อยู่ในเอกสาร ว่าจะต้องจัดการอย่างไร และบอกถึงความจำเป็นในการใช้งาน
- โฉบเอนโค้ดดิ้งรูล (SOAP Encoding Rule) จะเป็นส่วนสำหรับกำหนดกลไกที่ใช้ในการแลกเปลี่ยนข้อมูล ว่ามีข้อตกลงอย่างไรในการแลกเปลี่ยนข้อมูล

โฉบอาร์พีซีรีพีเรซันเทชัน (SOAP RPC Representation) จะเป็นส่วนสำหรับนิยามรูปแบบ รีโมท โพรซีเยร์คอล (Remote Procedure Call) และ การตอบสนอง (Response)



รูปที่ 2-7 โครงสร้างของโซบ

โซบแมสเซจ จะต้องเป็นไปตามกฎนี้

- เอนวิลีพเป็นเอลเลอเมนต์ที่อยู่บนสุด ต้องมีเอลเลอเมนต์นี้เสมอ
- เฮดเดอร์อาจจะมีหรือไม่มีก็ได้ แต่ถ้ามีต้องเป็นชายเอลเลอเมนต์ (Child Element) แรกของเอนวิลีพ
- บอดีต้องเป็นชายเอลเลอเมนต์แรกของเอนวิลีพ หรือ เฮดเดอร์

ตัวอย่างของ โซบแมสเซจ ตัวอย่างนี้คือ GetLastTradePrice โซบรีเควส (SOAP Request) ที่ส่งไปยัง StockQuote เซอร์วิสประกอบด้วยสตริงพารามิเตอร์ ซิม โบ (Symbol) และ คีนค่า โฟลต (Float) กลับมากับ โซบเรสสพอน (SOAP Response) โซบเอนวิลีพเอลเลอเมนต์ที่อยู่จุดบนสุดของเอกสารเอ็กซ์เอ็มแอล เอ็กซ์เอ็มแอลเนมสเปซใช้เพื่อป้องกันการสับสนของ โซบไอดี้นทิฟาย (SOAP Identifier) โซบรีเควสจะเป็นดังนี้

```
POST /StockQuote HTTP/1.1
```

```
HOST:www. Stockquoteserver.com
```

```
Content-Type: text/xml; charset="utf-8"
```

```
Content-Length: nnnn
```

```
SOAPAction: "Some-URI"
```

```
<SOAP-ENV:Envelope
```

```
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
<m:GetLastTradePrice xmlns:m="Some-URI">
<symbol>DIS</symbol>
</m:GetLastTradePrice>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

รูปที่ 2-8 ตัวอย่างของโซบริเคเวส

โซบเรสปลอนจะเป็นดังนี้

```

HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn

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

```

รูปที่ 2-9 ตัวอย่างของโซบเรสปลอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนั้นในโซบแมสเซจจะมีการใช้ เอ็กซ์เอ็มแอลเนมสเปซ ทุกๆ เอลเลอเมนต์ในเอกสารจะขึ้นต้นด้วยเนมสเปซซึ่งเนมสเปซจะถูกกำหนดโดยใช้เอ็กซ์เอ็มแอลเอนเอสแอททริบิว (xmlns attribute) ดังนี้

SOAP-Envelope

```
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
```

รูปที่ 2-10 เอ็กซ์เอ็มแอลเอนเอสแอททริบิว

แต่หากไม่ต้องการให้ขึ้นต้นด้วยเนมสเปซทำได้ดังนี้

<Envelope

```
xmlns="http://schemas.xmlsoap.org/soap/envelope/">
```

รูปที่ 2-11 เอ็กซ์เอ็มแอลเอนเอสแอททริบิวที่ไม่ขึ้นด้วยเนมสเปซ

2.3.1.1 เอนวิลีอพ (Envelope)

เอนวิลีอพเอลเลอเมทเป็นเอลเลอเมนต์บนสุดของเอกสารเอ็กซ์เอ็มแอลที่ใช้แสดงเมสเซจอาจประกอบด้วยแอททริบิว คือ เอนวิลีอพเนมสเปซ และ เอน โค้ดดิ้งสไตล์ (EncodingStyle)

- เอนวิลีอพเนมสเปซ (Envelop Namespace)– โซบแมสเซจจะแสดงเวอร์ชัน โดยใช้เนมสเปซของเอนวิลีอพเอลเลอเมนต์เนมสเปซ (Element Namespace) จะถูกนำไปใช้ขึ้นต้นเอนวิลีอพ, เฮดเคอร์ และ บอดีเอลเลอเมนต์ (Body Element)
- เอน โค้ดดิ้งสไตล์แอททริบิว (EncodingStyle Attribute) – ใช้แสดงวิธีซีรี่ไรท์เซชัน (Serialization) ของ โซบแมสเซจ แอททริบิวนี้สามารถมีได้ในทุกเอลเลอเมนต์และจะมีผลกับคอนเทนต์ (Content) ของเอลเลอเมนต์นั้นและ ซายเอลเลอเมนต์ทั้งหมดของมันที่ได้ประกาศแอททริบิว

2.3.1.2 บอดี (Body)

เป็นส่วนของข้อมูลที่ใช้ในการแลกเปลี่ยน โดยทั่วไปข้อมูลจะเป็นการมาร์แชลลอร์ฟี่ซีคอล (Marshall RPC Call) และเออเรอร์รีพอร์ตซายเอลเลอเมนต์ (Error Report Child Element) ของบอดีเอลเลอเมนต์จะถูกเรียกว่าบอดีเอนทรี (Body Entry)

บอดีเอนทรีจะต้องเป็นไปตามกฎนี้

- บอดีเอนทรีจะต้องแสดงด้วยชื่อเต็มที่ประกอบด้วยเนมสเปซยูอาร์ไอ (Namespace URI) และ ชื่อของมัน (Local Name)
- โซบเอน โค้ดดิ้งสไตล์แอททริบิว (SOAP EncodingStyle Attribute) อาจจะมีได้ เพื่อแสดงถึงวิธี เอน โค้ด (Encode) ของบอดีเอนทรีนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1.3 เฮดเดอร์ (Header)

เป็นส่วนเพิ่มเติมของแมสเซจสามารถประกอบด้วยอินฟอร์เมชันที่ระบุไปยังแอปพลิเคชัน ส่วนเพิ่มเติมนี้อาจนำไปใช้อิมพลิเมนต์เป็นออเทENTIเคชัน (Authentication), ทรานแซคชันแมนเนทเมนต์ (Transaction Management) เป็นต้น ทุก ๆ ซายเอลเลอเมนต์ของเฮดเดอร์เอลเลอเมนต์ จะถูกเรียกว่า เฮดเดอร์เอนทรี (Header Entry)

เฮดเดอร์เอนทรีจะต้องเป็นไปตามกฎดังนี้

- เฮดเดอร์เอนทรีจะต้องแสดงด้วยชื่อเต็มประกอบด้วยเนมสเปซยูอาร์ไอ และ ชื่อของมัน
- อาจมีโซบเอน โคลด์คิงส์ไคล์แอททริบิวต์ที่ใช้สำหรับเฮดเดอร์เอนทรี
- โซบมัสอันเดอร์สแตนแอททริบิวต์ (SOAP mustUnderStand Attribute) และ โซบแอกเทอร์
- แอททริบิวต์ (SOAP actor attribute) จะมีหรือไม่ก็ได้ เพื่อใช้บอกว่าจะจัดการกับเอนทรี นั้น อย่างไรและโดยใคร

```
<soap:Header>
```

```
<m:local xmlns:m="http://www.w3schools.com/local"
  soap:actor="http://www.w3schools.com/appml/">
```

```
<m:language>en</m:language>
```

```
<m:currency>USD</m:currency>
```

```
</m:local>
```

```
</soap:Header>
```

รูปที่ 2-12 ตัวอย่างของโซบเฮดเดอร์

2.3.1.3.1 โซบแอกเทอร์แอททริบิวต์

โซบแมสเซจสามารถถูกส่งไปยังปลายทางผ่านตัวกลางของโซบ (SOAP Intermediary) ตัวกลางของโซบ คือ แอปพลิเคชันที่มีความสามารถทั้งรับและส่งต่อ โซบแมสเซจ ในการส่งต่อตัวกลางจะต้องไม่ส่งต่อเฮดเดอร์เอลเลอเมนต์ไปให้กับแอปพลิเคชัน โดยตัวกลางนี้จะถูกกำหนดเป็นยูอาร์ไอ ถ้าไม่มีแอททริบิวต์นี้หมายถึงผู้รับโซบแมสเซจเป็นปลายทางสุดท้าย

2.3.1.3.2 โซบมัสอันเดอร์สแตนแอททริบิวต์

ใช้แสดงว่า เฮดเดอร์เอนทรีนี้จำเป็นหรือเป็นเพียงออปชันสำหรับผู้รับที่จะนำไปประมวลผล ค่าของมันคือ “1” หรือ “0” ถ้าไม่มีการระบุจะมีค่าเท่ากับ “0” โดย “0” หมายถึงเป็นออปชัน และ “1” หมายถึงจำเป็น โดยจะต้องประมวลผลให้ถูกต้องตามซีเมนติก (Semantic) หรือ ต้องไม่เฟว (Fail)

เช่น แมสเซจเป็นส่วนหนึ่งของทรานแซกชัน ถ้าปลายทางไม่รองรับทรานแซกชัน จะต้องไม่ประมวลผลและส่งผลผิดพลาดกลับไปในกรณีที่มันส์อันเดอร์สแตนแอททริบิว เป็น “1” แต่ถ้าเป็น “0” จะยังคงประมวลผลต่อไป

2.3.2 โซบฟอลเอลเลอเมนต์ (SOAP Fault Element)

ข้อความแสดงความผิดพลาดจากแอปพลิเคชันของโซบจะเก็บอยู่ในฟอลเอลเลอเมนต์ ซึ่งถ้ามีจะต้องปรากฏในบอดีเอลเลอเมนต์เพียงครั้งเดียวในโซบแมสเซจ ตัวอย่างอาจเป็นดังนี้

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:MustUnderstand</faultcode>
      <faultstring>Mandatory Header error.</faultstring>
      <faultactor>http://www.wrox.com/heroes/endpoint.asp</faultactor>
      <detail>
        <w:source xmlns:w=" http://www.wrox.com/">
          <module>endpoint.asp</module>
          <line>203</line>
        </w:source>
      </detail>
    </soap:Fault>
  </soap:Body>
```

รูปที่ 2-13 ตัวอย่างของโซบแมสเซจที่มีความผิดพลาด

โชนฟอลเอลเลอเม้นที่มีเอลเลอเม้นท์ย่อย ๆ ดังตารางนี้

Sub Element	Description
<faultcode>	โค้ดที่ระบุถึงการ error
<faultstring>	ข้อความการ error
<faultactor>	ใครเป็นสาเหตุของการ error
<detail>	ระบุอินฟอร์เมชันของการ error

ตารางที่ 2-1 ตัวอย่างของเอลเลอเม้นท์ย่อยๆ

ค่าของฟอลโค้ด (Fault Code) สามารถมีค่าได้ดังตารางนี้

Error	Description
VersionMismatch	namespace ภายใน SOAP Envelope element ไม่ถูกต้อง
MustUnderstand	child element ของ Header element กับ mustUnderstand attribute ที่มีค่า "1" ผู้รับ ไม่รับรอง
Client	แมสเซจที่รูปแบบไม่ถูกต้อง หรือมีอินฟอร์เมชันไม่ถูกต้อง
Server	เกิดปัญหากับเซิร์ฟเวอร์ ไม่สามารถประมวลผลได้

ตารางที่ 2-2 ตัวอย่างของค่าฟอลโค้ด

2.3.3 โชนเอนโค้ดดิ้ง (SOAP Encoding)

โชนเอนโค้ดดิ้ง มีวิธีการแมพ (Map) จากไทป์ของโปรแกรมมิ่งไปเป็นเอ็กซ์เอ็มแอล 2 วิธี คือจากภายนอกโดยใช้ดับบิวเอสดีแอล ที่บอกถึงไทป์ของข้อมูลที่ได้รับ หรือ ส่ง หรือใช้เอ็กซ์เอสไอไทป์แอททริบิว (xsi:type Attribute) ในกรณีที่ภาษาที่ใช้ไม่รองรับ ดับบิวเอสดีแอล โดยทั้ง 2 วิธีจะใช้เอ็กซ์เอ็มแอลสคริปต์ในการระบุไทป์ ตัวอย่าง ของการใช้เอ็กซ์เอสไอจะเป็นดังนี้

```

<soap:Envelope xmlns:soap=" http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyl="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <soap:Body>
    <m:MixedMessage xmlns:m="http://www.wrox.com/mix/">
      <param1 xsi:type="xsd:string">OU812</param1>
      <param2 xsi:type="xsd:integer">2001</param2>
      <param3 xsi:type="xsd:double">3.14159</param3>
    </m:MixedMessage>
  </soap:Body>
</soap:Envelope>

```

รูปที่ 2-14 ตัวอย่างของการใช้เอ็กซ์เอสแอล

โชนั้นรองรับไทป์ของข้อมูลได้ทั้งซิมเปิ้ลไทป์ (Simple Type) และ คอมพาวไทป์ (Compound Type) เช่น สตริก (Struct) และ อะเรย์ (Array)

2.3.3.1 สตริก

สตริกบุ๊ก (Struct Book) ที่เขียนด้วยภาษาซีพลัสพลัส (C++) จะเป็นดังนี้

```

Struct Book
{
  string author;
  string name;
  int page;
};

```

รูปที่ 2-15 ตัวอย่างของสตริกบุ๊ก

โซบแมสเซจที่ไม่ใช้คัมบิวเอสดีแอล แต่ใช้เอ็กซ์เอสไอ จะเป็นดังนี้

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
  <ns1:searchBook xmlns:ns1="http://soapinterop.org" SOAP-
  ENV:encodingStyle="http://schemas.xmlsoap.org/soap/envelope/">
  <bookResult xmlns:ns2="http://soapinterop.org/xsd" xsi:type="ns:Book">
    <author xsi:type="xsd:string">Ed Roman</author>
    <name xsi:type="xsd:string">Masterring EJB</name>
    <page xsi:type="xsd:int">500</page>
  </bookResult>
  </ns1:searchBook>
  </SOAP-ENV:Body>
  </SOAP_ENV:Envelope>
```

รูปที่ 2-16 ตัวอย่างของโซบแมสเซจที่ใช้เอ็กซ์เอสไอ

แต่ถ้าใช้คัมบิวเอสดีแอล ในส่วนของการประกาศสคริปต์ จะต้องประกอบด้วยสคริปต์ ดังนี้

```
<types>
  <schema targetNamespace="http://soapinterop.org/xsd"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">
  xmlns:wSDL="http://schemas.xmlsoap.org/wSDL"
  <elementFormDefault="qualified">
    <complexType name="Book">
      <sequence>
        <element name="author" type="string"/>
        <element name="name" type="string"/>
        <element name="page" type="int"/>
      </sequence>
    </complexType>
  </element>
</schema>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    </sequence>
  </complexType>
</schema>
</type>

```

รูปที่ 2-17 ตัวอย่างของโชนแมสเซจที่ใช้ดับบิวเอสดีแอล

2.3.3.2 อะเรย์

ตัวอย่างนี้เป็นอะเรย์ของสตริงในตัวอย่างก่อน โชนแมสเซจที่ไม่ใช้ดับบิวเอสดีแอล แต่ใช้ เอ็กซ์เอสไอจะเป็นดังนี้

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:BookArrayResult xmlns:ns1="http://soapinterop.org/" SOAP-ENV:encodingStyle
      ="http://schemas.xmlsoap.org/soap/encoding/">
      <BookArray xmlns:ns2="http://schemas.xmlsoap.org/soap/encoding/" xsi:type="ns2:Array"
        xmlns:ns3="http://soapinterop.org/xsd" ns2:arrayType=ns3:Book[3]>
        <item xsi:type="ns3:Book">
          <author xsi:type="xsd:string">Ed Roman</author>
          <name xsi:type="xsd:string">Mastering EJB</name>
          <page xsi:type="xsd:int">500</page>
        </item>
        <item xsi:type="ns3:Book">
          <author xsi:type="xsd:string">Roger Session</author>
          <name xsi:type="xsd:string">The battle of the middleware</name>
          <page xsi:type="xsd:int">350</page>
        </item>
        <item xsi:type="ns3:Book">
          <author xsi:type="xsd:string">Chris Dix</author>
          <name xsi:type="xsd:string">Professional XML Web Service</name>
          <page xsi:type="xsd:int">550</page>
        </item>
      </BookArray>
    </ns1:BookArrayResult>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

```

</item>
</BookArray>
</ns1:BookArrayResult>
<SOAP-ENV:Body>
<SOAP-ENV:Envelope>

```

รูปที่ 2-18 ตัวอย่างของอะเรย์ของสตริงที่ใช้อีเอชเอสบีไอ

แต่ถ้าใช้ ดับเบิลแอสกีแอลในส่วนของการประกาศสคริปต์จะต้องประกอบด้วยสคริปต์ดังนี้

```

<types>
  <schema targetNamespace="http://soapinterop.org/xsd"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    <elementFormDefault="qualified">
      <complexType name="Book">
        <sequence>
          <element name="author" type="string"/>
          <element name="name" type="string"/>
          <element name="page" type="int"/>
        </sequence>
      </complexType>
      <complexType name="ArrayOfBook">
        <complexContent>
          <restriction base="SOAP-ENC:Array">
            <attribute ref="SOAP-ENC:Array">
          </restriction>
        </complexContent>
      </complexType>
    </schema>
  </type>

```

รูปที่ 2-19 ตัวอย่างของอะเรย์ของสตริงที่ใช้อีเอชเอสบีไอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 โขบในเอชทีทีพี

ในการส่งโอบผ่านทางเอชทีทีพีจะต้องใช้คอนเทนไทป์ (Content-Type) เป็นเท็กซ์เอ็กซ์เอ็มแอล (Text/xml) แต่ในโอบริเวณจะต้องมี เฮดเดอร์โอบแอคชั่น (Header SOAPAction) ภายในเอชทีทีพีเฮดเดอร์ (HTTP Header) ตัวโอบแอคชั่น (SOAPAction) จะเป็นตัวบอกให้เซิร์ฟเวอร์รู้ว่าเอชทีทีพีโอบ (HTTP Post) นั้นเป็นโอบเมสเสจ และค่าของเฮดเดอร์ คือ ยูอาร์ไอที่แสดงถึงจุดหมายของโอบเมสเสจ ส่วนโอบเรสสปอนจะต้องมีสเตตัสโค้ด (Status Code) ตามมาตรฐานของเอชทีทีพี โดย 200 - 299 แสดงว่าสำเร็จ แต่ถ้าเรสสปอนเมสเสจเป็นการฟอล (Fault) แล้วสเตตัสโค้ดจะต้องเป็น 500 ซึ่งแสดงถึง อินเทอร์เน็ตเซิร์ฟเวอร์เออเรอ (Internal Server Error) ตัวอย่าง ของเรสสปอนที่มีสเตตัสโค้ดเป็น 500 อาจเป็นดังนี้

```

HTTP/1.1 500 Internal Server Error
Content-Type: text/xml
Content-Length: ###
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:VersionMismatch</faultcode>
      <faultstring>The SOAP namespace is incorrect.</faultstring>
      <faultactor>http://www.wrox.com/endpoint.asp</faultactor>
      <detail>
        <w:errorinfo xmlns:w="http://www.wrox.com/">
          <desc>The SOAP namespace was blank.</desc>
        </w:errorinfo>
      </detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>

```

รูปที่ 2-20 ตัวอย่างของเรสสปอนที่มีสเตตัสโค้ดเป็น 500

2.3.5 โขบสำหรับอาร์พีซี

จุดประสงค์ของการออกแบบโขบ คือ ทำอาร์พีซีโดยใช้เอ็กซ์เอ็มแอลการเรียกอาร์พีซีนั้นจะแมพเข้ากับเอชทีทีพีรีควีส และ อาร์พีซีเรสพอน (RPC Response) จะแมพกับเอชทีทีพีเรสพอน

ในการทำเมธอดคอล (Mechod Call) จำเป็นที่จะต้องมียีนฟอร์มเช่นดังนี้

2.3.5.1 ยูอาร์ไอ ของออบเจกต์ปลายทาง (Target Object)

- ชื่อของเมธอด
 - คำอธิบายของเมธอด (Method Signature) ส่วนนี้เป็นออปชัน
 - พารามิเตอร์สำหรับเมธอด
 - ข้อมูลส่วนหัว (Header Data) เป็นออปชัน
- พิจารณาเมธอดดังนี้

```
double GetStockQuote ( [in] string sSymbol );
```

รูปที่ 2-21 ตัวอย่างของเมธอด

ถ้าเนมสเปซของเมธอดคือ “http://www.worxstox.com/” แล้วการเรียกเมธอดโดยรีควีสสต็อกโควท (Request Stock Quote) ใช้ซิมโบบีโอยูเปคหนึ่งสอง (Symbol OU812) จะเป็นดังนี้

```
<q:GetStockQuote xmlns:q="http://www.wroxstox.com/">
  <q:sSymbol xsi:type="xsd:string">OU812</q:sSymbol>
</q:GetStockQuote>
```

รูปที่ 2-22 ตัวอย่างของโขบเมสเซจที่เรียกเมธอดโดยรีควีสสต็อกโควท

ชื่อเมธอดและชื่อของเอลเลอเมนต์จะต้องตรงกัน (Match) เช่นเดียวกับพารามิเตอร์สำหรับเรสพอน จะต้องชื่อเอลเลอเมนต์ จะต้องเป็นชื่อเมธอดตามด้วยเรสพอน ดังนี้

```
<q:GetStockQuoteResponse xmlns:q="http://www.wroxstox.com/">
  <q:ret xsi:type="xsd:double">100.0</q:ret>
</q:GetStockQuoteResponse>
```

รูปที่ 2-23 ตัวอย่างของชื่อเมธอดตามด้วยเรสพอน

ชื่อเอลเลอเมนต์ของค่าที่คืนมาสามารถเปลี่ยนได้ โดยจะกำหนดได้จากโปรแกรมที่เขียน หรือจาก ดับบิวเอสดีแอล

2.3.6 โซบทุลคิต (SOAP Toolkit)

โซบทุลคิต คือเครื่องมือที่จะทำหน้าที่ในการประมวลผลโซบรีควีส และส่งเรสสปอนกลับไปที่ไคลเอนต์โดยโซบทุลคิต แต่ละตัวใช้ได้บนแพลตฟอร์มที่ต่างกัน รองรับเทคโนโลยีเว็บเซอร์วิสที่ต่างกัน บางตัวอาจจะรองรับดับบิวเอสดีแอล หรือ ยูดีดีไอ ในขณะที่บางตัวรองรับอย่างใดอย่างหนึ่งหรือไม่รองรับทั้งสองอย่าง และถึงแม้ว่าโซบจะเป็นอิสระต่อแพลตฟอร์ม แต่ในโซบทุลคิตบางตัวยังไม่สามารถทำงานข้ามผลิตภัณฑ์ (Interoperability) ได้ เนื่องจากรองรับเทคโนโลยีของเว็บเซอร์วิสไม่เหมือนกัน เช่น เอ็กซ์เอสไอในบางผลิตภัณฑ์จะบังคับให้โซบแมสเชจจะต้องระบุไทม์ด้วยเอ็กซ์เอสไอ หากไม่มีก็จะเป็นฟอล (Fault) ซึ่งในปัจจุบันแต่ละผลิตภัณฑ์ก็พยายามพัฒนาให้สามารถทำงานด้วยกันได้

ชื่อ	แพลตฟอร์ม	แหล่งที่มา
4s4c	COM	http://www.4s4c.com
A SOAP for RPC NT Service	COM	http://www.whitemesa.com
Apache Axis	Java	http://xml.apache.org/axis
Apache SOAP	Java	http://xml.apache.org/soap
CapeConnect	Java	http://www.capeclear.com
Glue	Java	http://www.themindelectric.com
IBM Web Service Toolkit	Java	http://www.alphaworks.ibm.com
IdooXoap for Java and C++	Java, C++	http://www.zvon.org
Microsoft SOAP Toolkit	COM	http://www.microsoft.com
PacketSOAP	COM	http://www.packetsoap.com
SOAP::Lite	Perl	http://www.soaplite.com
Visual Studio .NET	COM	http://www.microsoft.com

ตารางที่ 2-3 ตัวอย่างของโซบทุลคิต

2.3.7 โครงสร้างภายในของโซปทูลลิตของไมโครซอฟต์

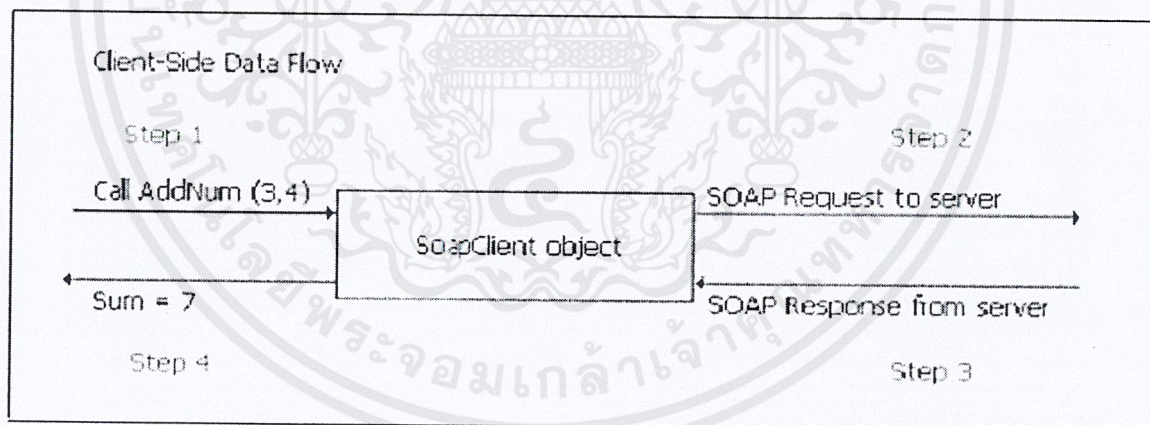
เว็บเซอร์วิสเคสสคริปต์ชั้นแลงแวก หรือ ดับบิวเอสดีแอลเป็นรูปแบบหนึ่งของเอ็กซ์เอ็มแอลเพื่อใช้อธิบายว่าเซิร์ฟเวอร์นั้นมีเซอร์วิส และ ออปเปอร์เรชัน (Operation) ภายในเซอร์วิสอะไรให้ใช้บ้าง ซึ่งออปเปอร์เรชันภายในเซอร์วิสนั้นจะอธิบายรูปแบบที่ไคลเอนต์ต้องใช้ในการเรียกออปเปอร์เรชัน

เว็บเซอร์วิสเมต้าแลงแวก หรือ ดับบิวเอสเอ็มแอลไฟล์ (WSML File) เป็นส่วนที่เพิ่มขึ้นมาจากไฟล์ดับบิวเอสดีแอลเพื่อใช้ในการเชื่อมโยงแต่ละออปเปอร์เรชันในเซอร์วิสกับคอมออบเจกต์ (COM Object) โดยดับบิวเอสเอ็มแอลไฟล์จะพิจารณาว่าคอมออบเจกต์ตัวใดที่จะต้องถูกโหลดขึ้นมาเมื่อมีการเรียกใช้ออปเปอร์เรชันในเซอร์วิส

ทั้งไฟล์ดับบิวเอสดีแอล และ ดับบิวเอสเอ็มแอลจะต้องอยู่ที่เซิร์ฟเวอร์โดยไคลเอนต์ที่ต้องการที่จะส่งโซปรีควีสไปที่เซิร์ฟเวอร์จะต้องได้รับดับบิวเอสดีแอลจากเซิร์ฟเวอร์ก่อน เพื่อที่จะรู้รูปแบบของโซปรีควีสในการติดต่อ

2.3.7.1 การสร้างโซปไคลเอนต์ออบเจกต์ (SoapClient Object)

ทางฝั่งไคลเอนต์เมื่อแอปพลิเคชันส่งแมสเซจไปที่โซปไคลเอนต์ออบเจกต์ เพื่อเรียกใช้ออปเปอร์เรชันหนึ่งๆนั้น (ตามรูปนี้ คือการบวกเลข 2 ตัว) โซปไคลเอนต์ออบเจกต์ จะทำการสร้างโซปรีควีสส่งไปที่เซิร์ฟเวอร์ เมื่อเซิร์ฟเวอร์ทำการประมวลผลเสร็จเรียบร้อยแล้ว ก็จะส่งผลลัพธ์ในรูปแบบของโซปเรซสปอนกลับมาให้ไคลเอนต์ หลังจากนั้นโซปไคลเอนต์ออบเจกต์ จึงทำการแปลและส่งแมสเซจที่เก็บผลลัพธ์ของออปเปอร์เรชัน นั้นกลับไปให้แอปพลิเคชันตามไดอะแกรมด้านล่าง

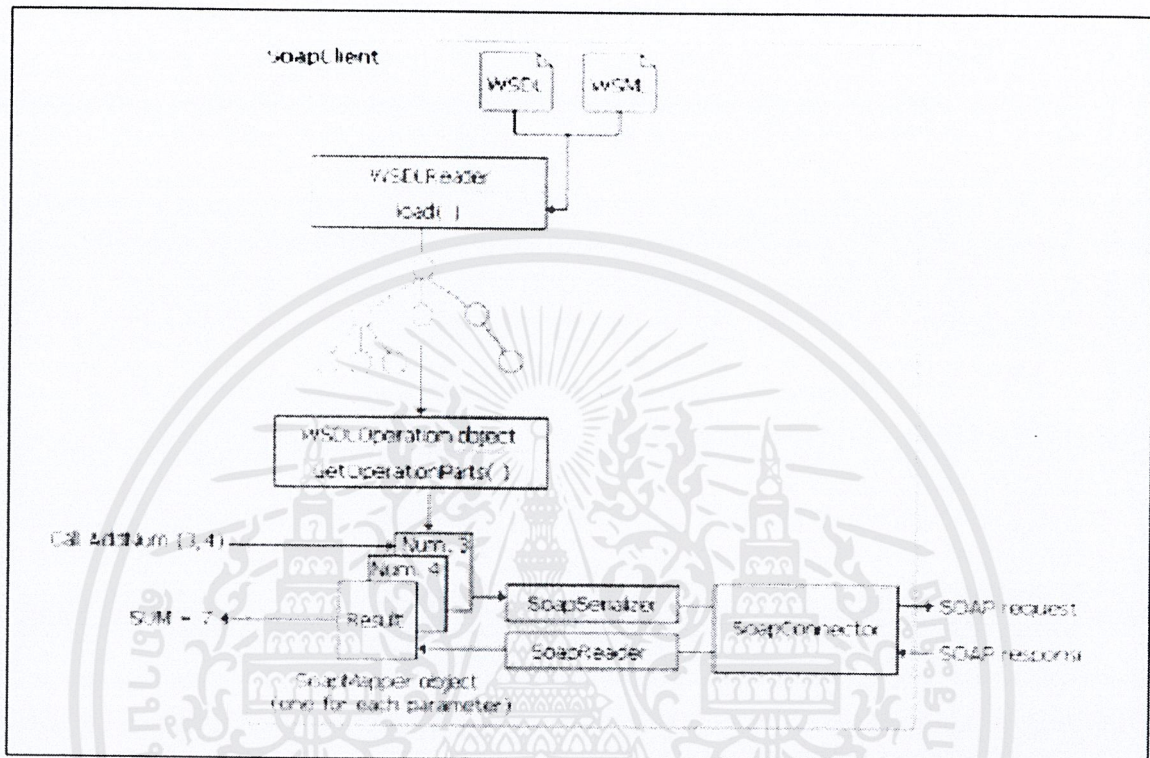


รูปที่ 2- 24 ตัวอย่างไดอะแกรมของการสร้างโซปไคลเอนต์ออบเจกต์

แอปพลิเคชันต้องอินสแตนทิเอตโซปออบเจกต์ (Instantiate Soap Object) ก่อนที่จะทำการเรียกใช้ออปเปอร์เรชันใดๆ (โซปไคลเอนต์ออบเจกต์ นี้ถูกสร้างเก็บไว้ในไฟล์ MSSOAPI.dll) โดยใช้เมธอด SoapClient.mssoapinit (ชื่อไฟล์ ดับบิวเอสดีแอล, ชื่อเซอร์วิส, ข้อมูลพอร์ต) โดยข้อมูลพอร์ตจะใช้ในการ

แบ่งแยกช่องทางที่ใช้ในการส่ง โขบแมสเซจ

2.3.7.2 การประมวลผลภายในโฉบไคลเอนต์ออบเจกต์ มีการทำงานดังรูป



รูปที่ 2-25 ตัวอย่างของการทำงานการประมวลผลภายในโฉบไคลเอนต์ออบเจกต์

ดับเอสดีแอลรีดเดอร์ออบเจกต์ (WSDL Reader Object) จะทำการโหลดไฟล์ดับบิวเอสดีแอล และไฟล์ดับบิวเอสเอ็มแอลเข้าไปไว้ในดอม (DOM) แล้วทำการวิเคราะห์โดยดับบิวเอสดีแอลรีดเดอร์ออบเจกต์ (WSDLReader Object) จะสร้างดับบิวเอสแอลโอเปอเรชันออบเจกต์ (WSDLOperation object) เพื่อทำการเรียกใช้งานออปเปอเรชัน (ตามรูปนี้คือ AddNum(3, 4)) หลังจากนั้นดับบิวเอสแอลโอเปอเรชัน จะเรียกเมธอด GetOperationParts เพื่อดึงรูปแบบแมสเซจของอินพุตและเอาท์พุทของออปเปอเรชันออกมาแล้วโฉบไคลเอนต์ (SoapClient) จะสร้างโฉบแมพเปอร์ออบเจกต์ (Soap Mapper Objects) ให้กับแต่ละส่วนที่ได้จากเมธอด GetOperationParts และโหลดค่าที่ได้มาจากการเรียกใช้ออปเปอเรชัน เข้าไว้ในออบเจกต์เหล่านี้ พอถึงขั้นตอนนี้ โฉบซีริไรส์ออบเจกต์ (SoapSerializer Objects) จะสร้างโฉบรีควีสแมสเซจ (SOAP Request Message) จาก โฉบแมพเปอร์ออบเจกต์ (SoapMapper Objects) ที่เหมาะสมและส่งไปที่เซิร์ฟเวอร์

เมื่อเซิร์ฟเวอร์ประมวลผลเสร็จแล้วก็จะส่งโฉบเรสพอนกลับไปที่ไคลเอนต์ โฉบไคลเอนต์ออบเจกต์จะทำการโหลดผลลัพธ์ของออปเปอเรชันไปเก็บในโฉบแมพเปอร์ออบเจกต์ที่เหมาะสมแล้ว

ส่งผลลัพธ์กลับไปให้แอปพลิเคชันของผู้ใช้

2.3.7.3 การสร้างโอบเซิร์ฟเวอร์ออบเจกต์ (SoapServer Object)

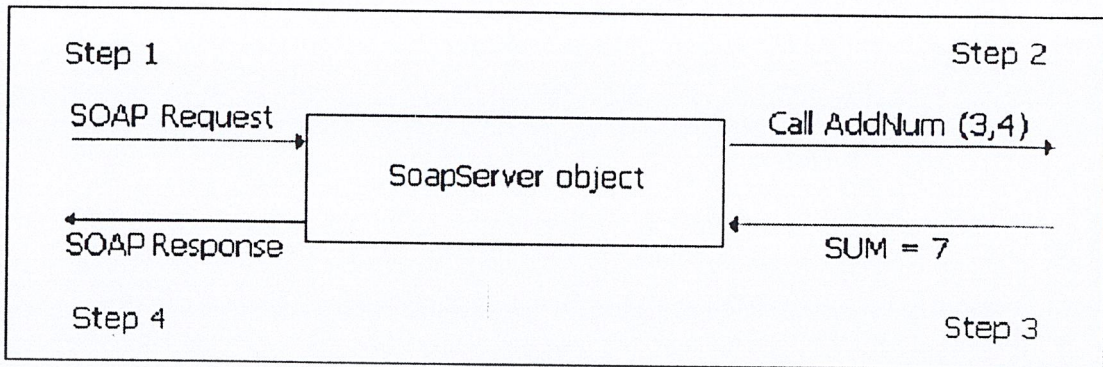
ส่วนทางฝั่งเซิร์ฟเวอร์เราสามารถกำหนดรูปแบบการรับโซบรีควีส (SOAP Listener) ได้ 2 รูปแบบ คือ อินเทอร์เน็ตเซิร์ฟเวอร์เอพีไอเซิร์ฟเวอร์ (Internet Server API Server) หรือ ไอเอสเอพีไอเซิร์ฟเวอร์ (ISAPI Server) และ แอกทีฟเซิร์ฟเวอร์เพจเซิร์ฟเวอร์ (Active Server Pages Server) หรือเอเอสพีเซิร์ฟเวอร์ (ASP Server)

```
<definitions>
...
<service name="DocSample1">
  <port name="DocSample1PortType" binding="tns:DocSample1Binding">
    <soap:address location="http://localhost/ DocSample1Test/ DocSample1.wsdl"/>
  </port>
</service>
...
</definitions>
```

รูปที่ 2-26 ตัวอย่างของโอบเซิร์ฟเวอร์ออบเจกต์

2.3.7.4 การทำงานทางฝั่งเซิร์ฟเวอร์

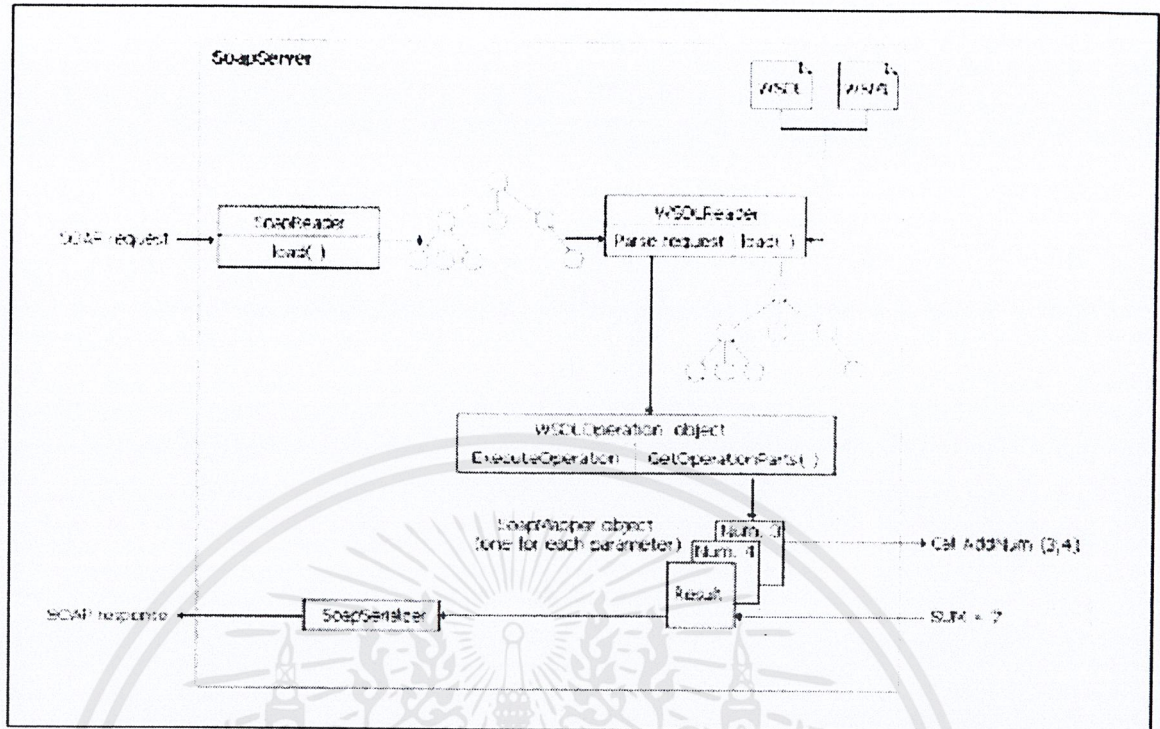
ไม่ว่าโซบรีควีสจะเรียกไปยังไอเอสเอพีไอเซิร์ฟเวอร์ หรือ เอเอสพีเซิร์ฟเวอร์ลิสเทนเนอร์ก็ตาม ข้อมูลที่รับส่งก็ยิ่งเหมือนกัน เมื่อโอบเซิร์ฟเวอร์ออบเจกต์ได้รับโซบรีควีสจากไคลเอนต์ ก็จะทำการแปลและเรียกไปยังเมธอด COM ที่ตรงกับออปเปอร์เรชั่นที่ต้องการ (ตามไคอะแกรมนี้คือเมธอด AddNum) หลังจากนั้นเมธอดที่ถูกเรียกใช้งานก็จะส่งผลลัพธ์กลับมาให้โอบเซิร์ฟเวอร์ออบเจกต์ซึ่งมันก็จะแปลงผลลัพธ์ให้อยู่ในรูปแบบของ โอบเรสสปอนแล้วจึงส่งกลับไปให้ไคลเอนต์ ดังรูป (Server – Side Data Flow)



รูปที่ 2-27 ตัวอย่างเซิร์ฟเวอร์ไซต์คาต้าโพล

2.3.7.5 การประมวลผลภายใน โซบเซิร์ฟเวอร์ออบเจกต์

เมื่อได้รับโซบรีควีสจากไคลเอนต์แล้วโซบรีควีสออบเจกต์ (SoapReader Object) ก็จะโหลดรีควีสแมสเชงไปไว้ คอมสตรักเตอร์ (DOM Structure) หนึ่ง ในขณะที่ดับบิวเอสดีแอลรีดเดอร์ (WSDLReader) จะโหลดไฟล์ ดับบิวเอสดีแอล และไฟล์ดับบิวเอสเอ็มแอลไปไว้ในอีก คอมสตรักเตอร์หนึ่ง ตัว ดับบิวเอสดีแอลรีดเดอร์จะแปลรีควีสและสร้างดับบิวเอสดีแอลออปเปอร์เรชั่นออบเจกต์ ขึ้นเพื่อใช้ในรีควีสออปเปอร์เรชั่นหลังจากนั้น ดับบิวเอสดีแอลออปเปอร์เรชั่นออบเจกต์ จะเรียกใช้เมธอด GetOperationParts เพื่อดึงรูปแบบแมสเชงของอินพุทและเอาท์พุทของออปเปอร์เรชั่น ออกมา แล้วโซบเซิร์ฟเวอร์ จะสร้างโซบเมพอเรียออบเจกต์ ให้กับแต่ละส่วนที่ได้จากเมธอด GetOperationParts และโหลดค่าที่ได้มาจากการเรียกใช้ออปเปอร์เรชั่น เข้าไว้ในออบเจกต์ เหล่านี้ พอถึงขั้นตอนนี้ โซบเซิร์ฟเวอร์ออบเจกต์ จะทำการเรียกไปที่ คอมเมธอด (COM method) ที่ตรงกับที่ร้องขอเข้ามา ดังรูป



รูปที่ 2-28 ตัวอย่างของการทำงานการประมวลผลภายในโซบเซอร์ฟเวอร์ออบเจกต์

2.4 ดับบิวเอสดีแอล (WSDL)

2.4.1 ดับบิวเอสดีแอลคืออะไร

ดับบิวเอสดีแอล หรือ เว็บเซอร์วิสเดสคริปต์ชั้นแลงแกจ (Web Service Description Language) นั้นถูกคิดค้นขึ้นโดยบริษัทไอบีเอ็ม (IBM) และบริษัทไมโครซอฟต์ (Microsoft) ซึ่งเป็นภาษาที่ใช้อธิบายคุณลักษณะการให้บริการของเว็บเซอร์วิส และวิธีการติดต่อขอรับบริการจากเว็บเซอร์วิส เช่น ชื่อเว็บเซอร์วิส, ชื่อเมธอดของคอมโพเนนท์ที่เปิดให้บริการ, พารามิเตอร์ (Parameter) ที่ส่งไปยังเมธอด, ชนิดข้อมูลของพารามิเตอร์ เป็นต้น โดยรายละเอียดเหล่านี้จะเป็นไปตามไวยากรณ์ของภาษาเอ็กซ์เอ็มแอล

2.4.2 ความสัมพันธ์ระหว่างดับบิวเอสดีแอลกับเว็บเซอร์วิส

โซบ หรือ ซิมเปลออบเจกต์เอกเซสโปรโตคอล, ยูดีดีไอ หรือ ยูนิเวอร์ซัลเดสคริปต์ชั้นคิสต์ฟเวอร์เอนดointer ชั้น และ ดับบิวเอสดีแอล หรือ เว็บเซอร์วิสเดสคริปต์ชั้นแลงแกจ ต่างก็เป็นแกนหลักสำคัญของเว็บเซอร์วิสทั้งสิ้น สิ่งสำคัญในการเรียกใช้งานเว็บเซอร์วิส คือ ต้องรู้ที่อยู่ (จากยูดีดีไอ) รู้คุณลักษณะของเว็บเซอร์วิสนั้นๆ จากเมธอดของเว็บเซอร์วิส และ พารามิเตอร์ต่างๆที่จำเป็น (จากเอกสารดับบิวเอสดีแอล) และร้องขอบริการ รอดผลลัพธ์จากผู้ให้บริการ (ด้วยโซบโปรโตคอล)

ความสัมพันธ์ระหว่าง 3 ส่วน แสดงได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โขบทจัดเป็นโปรโตคอลสื่อสารที่อาศัยไวยากรณ์ของภาษาเอ็กซ์เอ็มแอล และทำงานกับโปรโตคอลอื่นๆ ได้หลายชนิด เนื่องจากใช้ไวยากรณ์เอ็กซ์เอ็มแอล จึงทำให้ทำงานได้ทุกแพลตฟอร์ม ดังนั้นเราก็เลยสามารถเรียกใช้คอมโพเนนท์ข้ามแพลตฟอร์มได้
- คำบิวดิสคิแอสทำหน้าที่ที่เรียกว่า เซอร์วิสเดสคริปชัน (Service Description) ถือเป็นเอกสารที่ใช้อธิบายคุณลักษณะของเว็บเซอร์วิสที่ได้กล่าวมาแล้ว โดยใช้ไวยากรณ์ของเอ็กซ์เอ็มแอล เช่นกัน
- ยูติลิตี้โอเปรียบเสมือนฐานข้อมูลที่เกี่ยวข้องของเว็บเซอร์วิสไว้ และรอให้ผู้มาใช้บริการมาค้นหาบริการ บทบาทนี้เรียกว่า เซอร์วิสดีคัฟเวอรี (Service Discovery) ส่วนในกรณีของผู้ให้บริการ ก็ต้องนำข้อมูลเกี่ยวกับเว็บเซอร์วิสของตนไปเก็บไว้ในยูติลิตี้โอเปรียบของยูติลิตี้โอ คือ เซอร์วิสพับลิเคชัน (Service Publication)

2.4.3 โครงสร้างของดับบิวเอสดีแอล

โครงสร้างของเอกสารดับบิวเอสดีแอลนั้น แบ่งออกเป็น 2 ส่วนใหญ่ๆด้วยกัน คือ

2.4.3.1 แอ็บสแตร็กเดฟนิชันกรุป (Abstract Definitions Group)

ทำหน้าที่ คือ จะกำหนด เมสเสจ โขบทที่ไม่ขึ้นกับแพลตฟอร์ม หรือรูปแบบภาษา กล่าวคือไม่ได้บรรจุส่วนประกอบที่มีเฉพาะในเครื่องบางเครื่อง หรือภาษาบางภาษา โดยที่ส่วนประกอบของแอ็บสแตร็กนั้น จะประกอบไปด้วย ไทป์ (type), เมสเสจ (Message) และพอร์ตไทป์ (portType)

- ไทป์ ประกอบด้วยส่วนประกอบสคริปต์ที่กำหนดรูปแบบข้อมูลต่างๆ โดยที่ข้อมูลเหล่านั้นไม่มียึดติดแบบรูปแบบ แพลตฟอร์ม และภาษา
- เมสเสจ ทำหน้าที่อธิบายข้อมูลที่ใช้ในการแลกเปลี่ยนระหว่างผู้ให้บริการเว็บเซอร์วิส กับผู้ขอใช้บริการ โดยที่เว็บเมธอด (Web Method) แต่ละตัวจะมี 2 เมสเสจ คือ อินพุท กับ เอาท์พุท โดยที่อินพุทนั้น ทำหน้าที่อธิบายพารามิเตอร์สำหรับเว็บเมธอด ส่วนเอาท์พุทจะอธิบายข้อมูลที่เว็บเมธอดนั้นส่งกลับมา ซึ่งเมสเสจแต่ละตัวนั้นอาจจะไม่มีหรือมีการระบุพารามิเตอร์ของเว็บเมธอด โดยผ่านทางพารามิเตอร์ <พอร์ต> (parameter <port>) ซึ่งพารามิเตอร์แต่ละตัว จะเกี่ยวข้องกับรูปแบบของข้อมูลคอนกรีตที่กำหนดไว้ในส่วนประกอบคอนเทนเนอร์ <ไทป์> (Container <types>)
- พอร์ตไทป์ ประกอบไปด้วยเซต (Set) ของ แอ็บสแตร็กโอเปอเรชัน (Abstract Operations) ซึ่งตัวโอเปอเรชันนี้ ก็จะอ้างอิงถึงอินพุทเมสเสจ (Input Message) และ เอาท์พุทเมสเสจ (Output Message)

2.4.3.2 คอนกรีตเดสคริปชันกรุป (Concrete Descriptions Group)

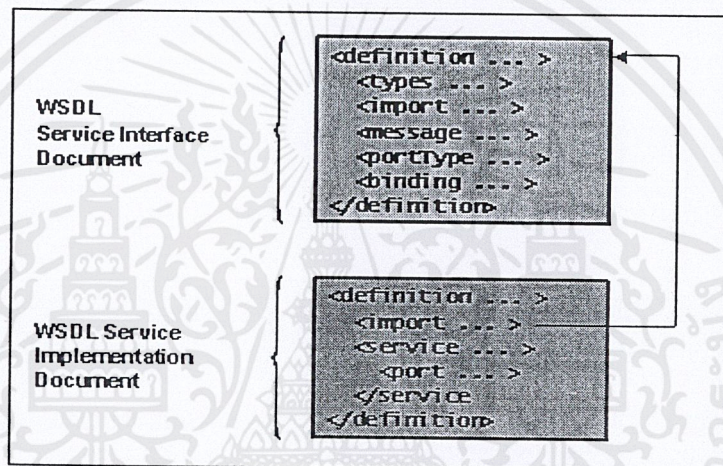
จะทำหน้าที่ตรงกันข้ามกับส่วนของแอ็บสแตร็กเซกชัน (Abstract Section) ก็จะเป็นข้อมูลที่ขึ้นอยู่กับขนาด (Size), เครื่องและภาษา โดยประกอบด้วย บายดิง (Binding) และ เซอร์วิส (Service)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

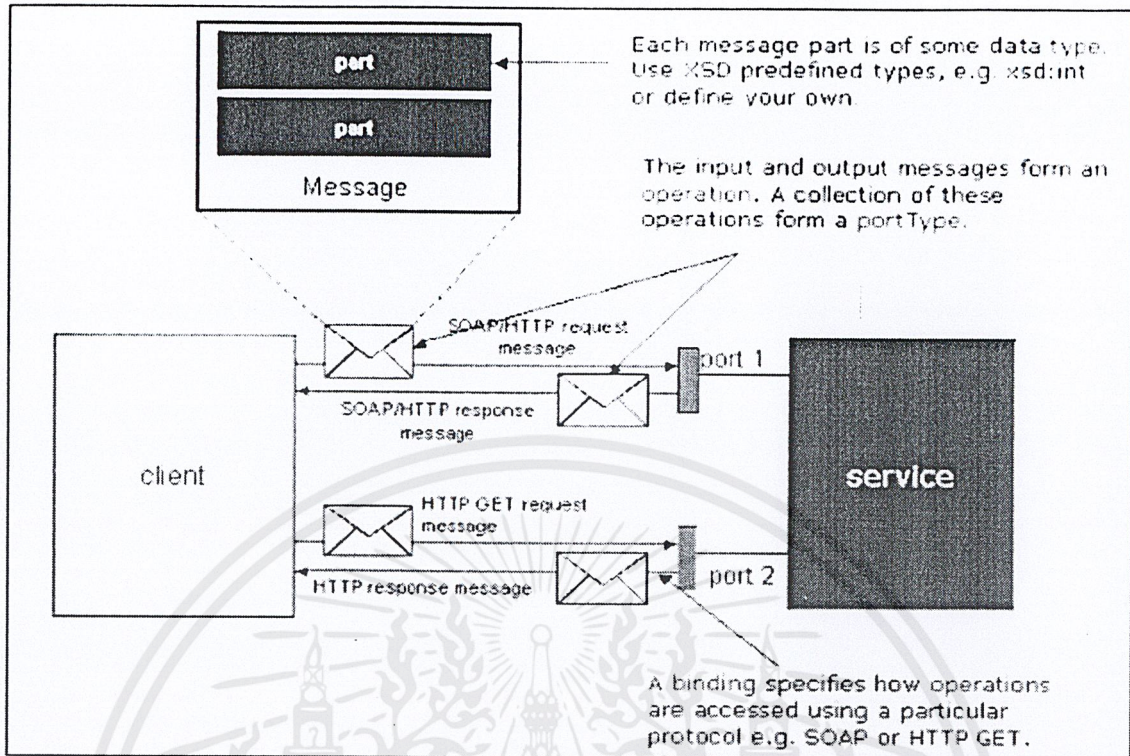
- **บายนด์** ทำหน้าที่อธิบายโปรโตคอลโดยที่บายนด์นั้นประกอบไปด้วยออปเปอเรชัน, อินพุทแมสเสจ โดยที่ส่วนประกอบของบายนด์นั้นจะเหมือนกันกับพอร์ตไทป์
- **เซอร์วิส** ประกอบไปด้วยคอลเล็กชัน (Collection) ของพอร์ตต่างๆ โดยที่แต่ละพอร์ตนั้นจะจับพอร์ตโปรดัก (Support Product) ที่ต่างกัน

นอกจากนั้นแล้วยังมีส่วนอื่นๆอีกด้วย

- **พอร์ต** จะระบุแอดเดรส (Address) ของบายนด์ แล้วทำการดีไฟน์ซิงเกิลคอมมูนิเคชันเอนด์พอย (Define Single Communication Endpoint)
- **ออปเปอเรชัน** (Operation) จะทำการรวมพอร์ตต่างๆที่เกี่ยวข้องนั้นเข้าด้วยกัน

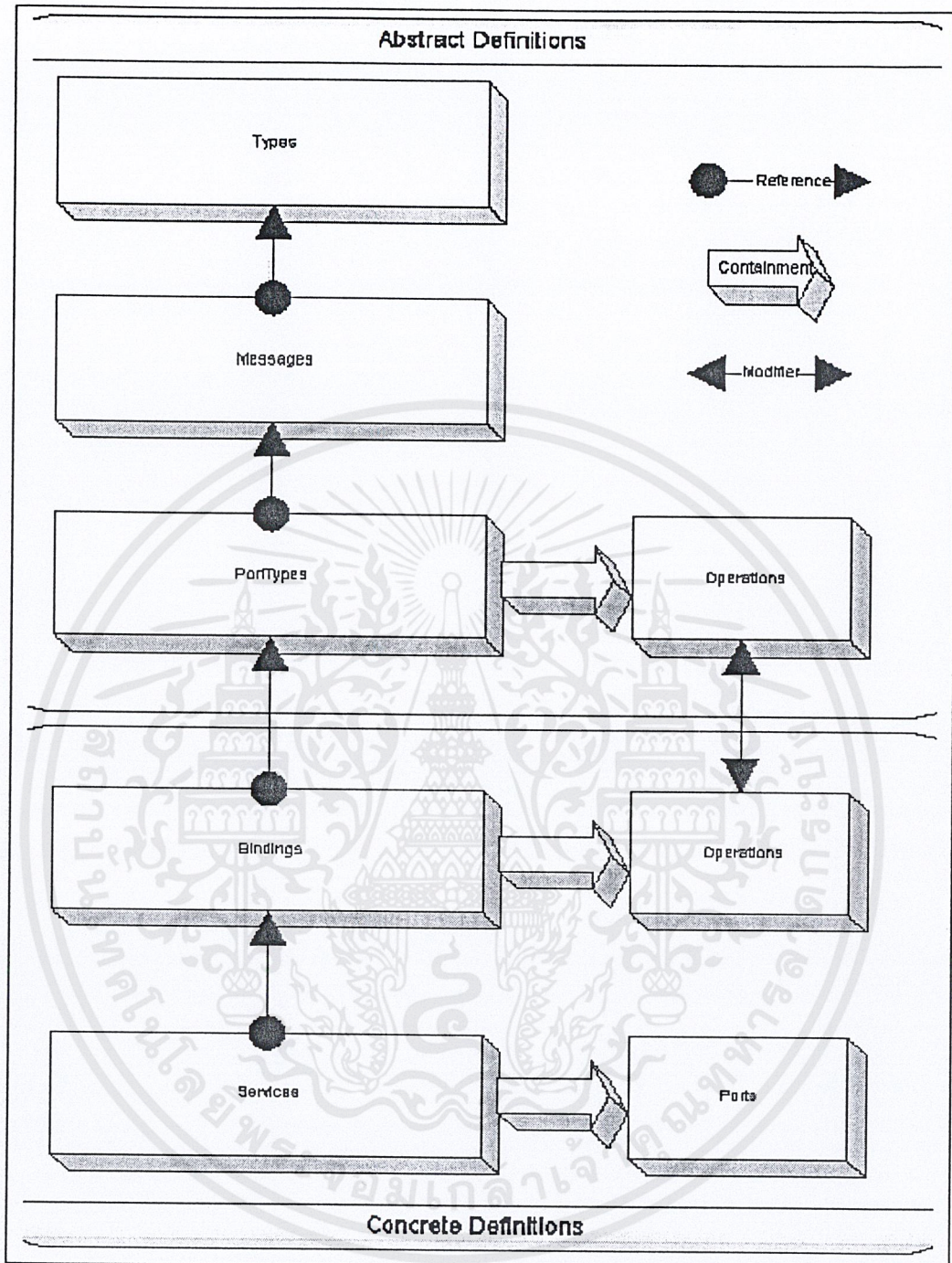


รูปที่ 2-29 ตัวอย่างของโครงสร้างของเอกสารดับเบิลยูเอสดีแอล



รูปที่ 2-30 แสดงการส่งเอกสารด้วยเว็บเซตซีแอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-31 แสดงส่วนประกอบของคัมบิวเอสดีแอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<?xml version = "1.0"?>
<definitions name = "StockQuoteService" targetNamespace =
  "http://www.xmethods.net/sd/StockQuoteService.wsdl"
  xmlns.tns = "http://www.xmethods.net/sd/StockQuoteService.wsdl"
  xmlns.xsd = "http://www.w3.org/1999/XMLSchema"
  xmlns.soap = "http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns = "http://schemas.xmlsoap.org/wsdl/">
  <message name = "getQuoteRequest"                // จุดสังเกตที่ 3
    <part name = "xsd:symbol" type = ":string"/>
  </message>
  <message name = "getQuoteResponse"              // จุดสังเกตที่ 4
    <part name = "Result" type = "xsd:float"/>
  </message>
  <portType name = "StockQuotePortType">
    <operation name = "getQuote">
      <input message = "tns:getQuoteResponse" name = "getQuoteResponse"/>
    </operation>
  </portType>
  <binding name = "getQuoteBinding" type = "tns:getQuotePortType">
  <soap:binding style = "rpc" transport = "http://schemas.xmlsoap.org/soap/http">
    <operation name = "getQuote"                // จุดสังเกตที่ 2
      <soap:operation soapAction="">
        <input>
          <soap:body use = "encode" namespace = "urn:xmethods-delayed-quotes"
            encodingStyle = "http://schemas.xmlsoap.org/soap/encoding/">
        </input>
        <output>
          <soap:body use = "encode" namespace = "urn:xmethods-delayed-quotes"
            encodingStyle = "http://schemas.xmlsoap.org/soap/encoding/">
        </output>
      </operation>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

</binding>
<service name = "StockQuoteService"> // จุดสังเกตที่ 1
  <documentation>Obtains 20 minute delayed quotes from Yahoo</documentation>
  <port name = "StockQuotePort" binding = "tns:StockQuoteBinding">
    <soap:address location = "http://service.xmethods.net:80/soap/"> // จุดสังเกตที่ 5
  </port>
</service>
</definitions>

```

รูปที่ 2-32 เอกสารฉบับวีวเอสดีแอล

ในเอกสารฉบับวีวเอสดีแอล แต่ละไฟล์ มี 5 จุด ที่ควรรู้ ดังที่แสดงข้างต้น โดยเรียงลำดับตามความสำคัญจากมากไปน้อยดังนี้

จุดสังเกตที่ 1 บอกรหัสเว็บเซอร์วิส

สำหรับเว็บเซอร์วิสหนึ่งๆ จะมีเว็บเซอร์วิส จำนวนกี่บริการก็ได้ และชื่อเว็บเซอร์วิส ก็เป็นตัวจำแนกและบอกแต่ละบริการซึ่งห้ามซ้ำกัน โดยตำแหน่งในเอกสารฉบับวีวเอสดีแอลที่บอกรหัสเว็บเซอร์วิสคือในแอททริบิวต์ (Attribute Name) ของ เอลเมนต์เซอร์วิส (Element <service>) สำหรับชื่อเว็บเซอร์วิส StockQuoteService

จุดสังเกตที่ 2 บอกรหัสของเว็บเซอร์วิส

สำหรับเว็บเซอร์วิสหนึ่งๆ จะมีเมธอดจำนวนกี่เมธอดก็ได้ ยกตัวอย่าง กรณีเว็บเซอร์วิสที่ให้บริการเครื่องคิดเลข อาจจะมีเมธอดแอด (Method Add (บวก)), ซับเทรซ (Subtract (ลบ)), มัลติพลาย (Multiply (คูณ)), ดีไวด์ (Divide (หาร)) และอื่นๆ โดยตำแหน่งในเอกสารฉบับวีวเอสดีแอลที่บอกรหัสเมธอดอยู่ที่แอททริบิวต์ของเอลเมนต์อ็อปเรชัน (Element <operation>) ซึ่งชื่อเมธอดของเว็บเซอร์วิสในตัวอย่างนี้คือ getQuote

จุดสังเกตที่ 3 และ 4 บอกรหัสพารามิเตอร์ของเมธอด ของเว็บเซอร์วิส และค่าที่ส่งกลับมา

ปกติแล้วในเอกสารฉบับวีวเอสดีแอลจะมีเอลเมนต์เมสเซจ (Element <message>) เป็นคู่ๆ ซึ่งก็คือข้อมูลพารามิเตอร์ของ เมธอดที่ส่ง ไปให้เว็บเซอร์วิสประมวลผล และข้อมูลผลลัพธ์ที่ส่งกลับมา ส่วนใหญ่แล้วเราจะดูที่จุดสังเกตที่ 3 และ 4 เพื่อให้ทราบชนิดข้อมูล ว่าพารามิเตอร์ที่ส่งไปและข้อมูลที่ส่งกลับเป็นชนิดอะไร สำหรับตัวอย่างนี้ข้อมูลพารามิเตอร์เป็นชนิด สตริง (String เพราะเมธอด getQuote ให้ได้พารามิเตอร์ เป็นชื่อย่อของบริษัทที่ต้องการทราบราคาหุ้น) ส่วนข้อมูลที่ส่งกลับเป็นชนิด โฟลต (Float ตัวเลขดัชนีราคาหุ้นซึ่งเป็นเลขทศนิยม)

จุดสังเกตที่ 5 บอกตำแหน่งของไฟล์ซึ่งทำหน้าที่เป็นโซบลิสเทนเนอร์ (SOAP Listener)

ดังที่เลขกล่าวไว้แล้วว่าไฟล์โซบลิสเทนเนอร์ (File SOAP Listener) จะทำหน้าที่คอยฟังว่ามีการเรียกเว็บเซอร์วิสหรือไม่ ถ้ามีก็จะไปปลุกให้เว็บเซอร์วิสนั้นๆทำงาน และ ตำแหน่งของไฟล์นี้ก็บอกด้วยแอททริบิวต์โลเคชัน (Attribute Location) ใน เอลเมนต์แอดเดรส (Element <address>)

ตามทฤษฎีแล้วไฟล์เอกสารดับเบิลเอชดีแอล แต่ละไฟล์สามารถอธิบายคุณลักษณะของบริการเว็บเซอร์วิสได้มากกว่า 1 บริการ โดยแต่ละเว็บเซอร์วิสจะมีพอร์ตสื่อสารเฉพาะตัว ซึ่งบ่งบอกไว้ในเอกสารดับเบิลเอชดีแอล อยู่แล้ว

2.5 ยูดีดีไอ (UDDI)

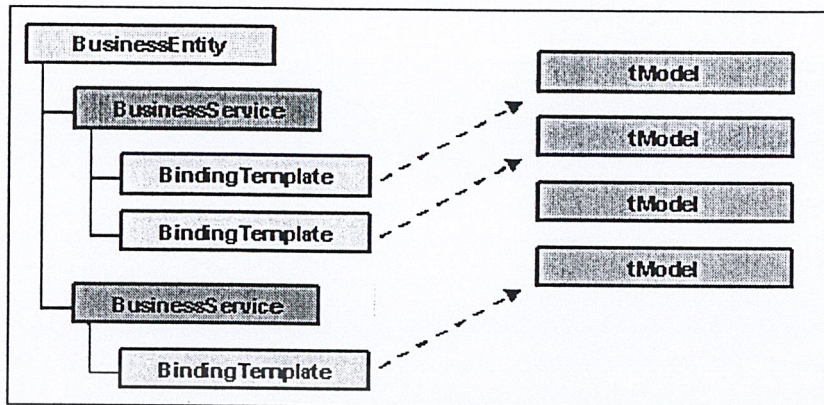
2.5.1 ยูดีดีไอคืออะไร

ยูดีดีไอ หรือ ยูนิเวอร์ซัลเดสคริปชันดิคคัพเวอรีแอนด์อินทิเกรชัน (Universal Description Discovery and Integration) คือเป็นตัวที่ให้เมธอด สำหรับการสร้างและการหารายละเอียดของเซอร์วิส โดยที่ ยูดีดีไอดาต้าเอนทิตี (UDDI Data Entities) จะเป็นตัวกำหนดทั้งข้อมูลบิสซิเนส (Business Data) และข้อมูลเซอร์วิส (Service Data) ที่ถูกกำหนดในดับเบิลเอชดีแอลจะถูกคอมพลีท (Complete) ไปไว้ในยูดีดีไอเรจิสทรี (UDDI Registry) ซึ่ง ยูดีดีไอนั้นจะสนับสนุนเซอร์วิสได้หลายๆประเภท ดังนั้น ยูดีดีไอนั้นจะไม่ขึ้นกับดับเบิลเอชดีแอล

2.5.2 โครงสร้างของยูดีดีไอ

จะมีประเภทของข้อมูลอยู่ 4 ตัวด้วยกัน เรียกว่าไพรมารีดาต้าไทป์ (Primary Data Types) ในดาต้าเรจิสทรี

1. บิสซิเนสเอนทิตี (Business Entity)
2. บิสซิเนสเซอร์วิส (Business Service)
3. บายคิง์เทมเพลต (Binding Template)
4. ทีโมเดล (tModel)



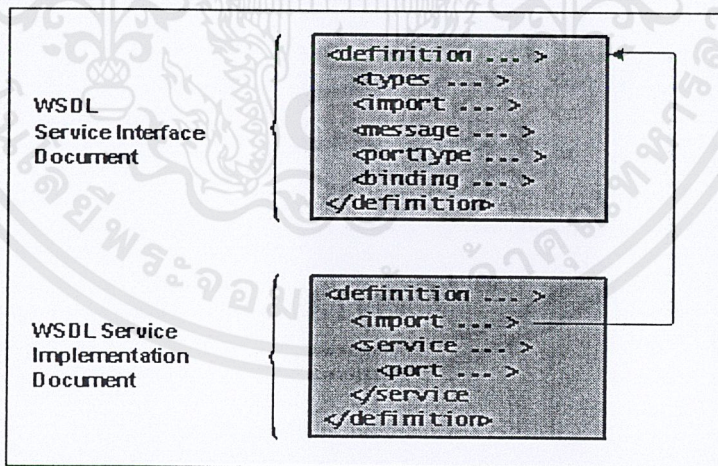
รูปที่ 2-33 โครงสร้างของยูดีดีไอ

บิสซิเนสเอนทิตีนั้นจะให้ข้อมูลเกี่ยวกับบิสซิเนส และ บรรจুবิสซิเนสเซอร์วิสหนึ่งอัน หรือมากกว่านั้น ถ้าเป็น บิสซิเนสเซอร์วิสนั้น เทคนิคและวิธีการอธิบายลักษณะของบิสซิเนส สำหรับเว็บเซอร์วิสจะถูกกำหนด ลงใน บิสซิเนสเซอร์วิส และ บายคิงเทมเพลต ซึ่งแต่ละ บายคิงเทมเพลตจะบรรจุที่โมเดลหนึ่งอัน หรือมากกว่านั้น โดยที่แต่ละที่โมเดล จะใช้ในการกำหนดรายละเอียดทางเทคนิค (Technical Specification) สำหรับแต่ละเซอร์วิส

2.5.3 ประเภทของเอกสารดับิวเอสดีแอล

เอกสารดับิวเอสดีแอลที่จะทำการประกาศในยูดีดีไอ มี 2 ประเภทคือ

- เซอร์วิสอินเทอร์เฟซ (Service Interfaces)
- เซอร์วิสอิมพลีเม้นท์เทชัน (Service Implementations)



รูปที่ 2-34 ประเภทของเอกสารดับิวเอสดีแอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซอร์วิสอินเทอร์เฟซนั้นถูกอธิบายโดยเอกสารฉบับวิเอสดีแอลที่ประกอบไปด้วยไทล์, อิมพอร์ต, แมสเซจ และ บายคิง ส่วน เซอร์วิสอินเทอร์เฟซ จะประกอบด้วย ฉบับวิเอสดีแอลเซอร์วิสเคฟินิชัน (WSDL service definition) ซึ่งจะใช้ในการอิมพลิเมนต์ 1 เซอร์วิส หรือ มากกว่านั้น

เซอร์วิสอินเทอร์เฟซคือคิวเมนต์ (Service Interface Document) นั้นสามารถอ้างอิง เซอร์วิสอินเทอร์เฟซคือคิวเมนต์ตัวอื่น ได้ในการอิมพอร์ตเอลเลอเมนต์ (Import Element) โดยเซอร์วิสอินเทอร์เฟซคือคิวเมนต์ที่ถูกอิมพลิเมนต์แล้ว จะมีอิมพอร์ต และ เซอร์วิสเอลเลอเมนต์ซึ่งเซอร์วิสคือคิวเมนต์ ที่อิมพลิเมนต์เท่านั้นแล้วจะบรรจุเดสคริปต์ชันของเซอร์วิสที่อิมพลิเมนต์ตัวเซอร์วิสอินเทอร์เฟซแล้ว

อิมพอร์ตเอลเลอเมนต์อย่างน้อย 1 ตัว หรือมากกว่านั้น จะบรรจุเรเฟอเรนซ์ (Reference) ของ ฉบับวิเอสดีแอลเซอร์วิสอินเทอร์เฟซคือคิวเมนต์ (WSDL Service Interface Document) ได้มากกว่า 1 อัน

อิมพอร์ตเอลเลอเมนต์ (Import Element) ใน วิเอสดีแอลเซอร์วิสอินเทอร์เฟซคือคิวเมนต์ นั้น จะบรรจุ 2 แอททริบิวต์ โดยที่ แอททริบิวต์เนมสเปซ (Attribute Namespace) จะเก็บยูอาร์แอล ที่ตรงกับ targetNamespace ใน เซอร์วิสอินเทอร์เฟซคือคิวเมนต์

ส่วนแอททริบิวต์โลเคชัน (Attribute Location) จะเก็บยูอาร์แอลที่ใช้เรเฟอเรนซ์ฉบับวิเอสดีแอลคือคิวเมนต์ที่บรรจุคอมพลีทเซอร์วิสอินเทอร์เฟซเคฟินิชัน (Complete Service Interface Definition)

ส่วนแอททริบิวต์บายคิง (Attribute Binding) บน พอร์ตจะเรเฟอเรนซ์ถึงสเปกซิฟิค (Specific Binding) ในเซอร์วิสอินเทอร์เฟซคือคิวเมนต์

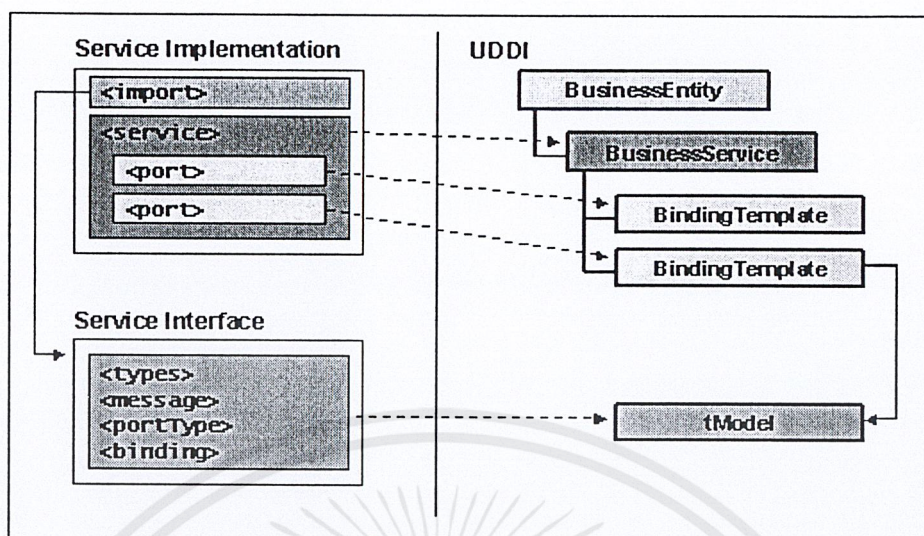
- เซอร์วิสอินเทอร์เฟซคือคิวเมนต์ ถูกพัฒนาและประกาศโดย เซอร์วิสอินเทอร์เฟซโพรไวเดอร์ (Service Interface Provider)
- เซอร์วิสอินเทอร์เฟซคือคิวเมนต์ ถูกพัฒนาและประกาศโดย เซอร์วิสโพรไวเดอร์ (Service Provider)

2.5.4 การประกาศและค้นหาเอกสารฉบับวิเอสดีแอลในยูติลิตี้ไอ

เซอร์วิสอินเทอร์เฟซ (Service Interface) เมื่อรวมกับ เซอร์วิสอิมพลิเมนต์ที่แท่นคือคิวเมนต์ (Service Implementation Document) ก็จะกลายเป็น เอกสารฉบับวิเอสดีแอลที่สมบูรณ์แล้ว

เมื่อเซอร์วิสอินเทอร์เฟซนั้นทำการกำหนดเซอร์วิส (Define Service) มันจะต้องทำการประกาศ เซอร์วิส (Published Service) ลงในยูติลิตี้โอริจิสทรีเหมือนกับที่โมเดล หลังจากนั้นเซอร์วิสอิมพลิเมนต์ที่แท่นจะอธิบายอินสแตนซ์ (Describe Instances) ของเซอร์วิส ซึ่งแต่ละอินสแตนซ์ (Instance) ก็จะทำการกำหนดการใช้เอกสารฉบับวิเอสดีแอลเซอร์วิสเอลเลอเมนต์ (WSDL Service Element) ซึ่งเซอร์วิสเอลเลอเมนต์ (Service Element) จะถูกเคยประกาศในยูติลิตี้โอปิซซิเนสเซอร์วิส

เมื่อเอกสารฉบับวิเอสดีแอลเซอร์วิสเดสคริปต์ชันนั้นถูกประกาศ (Publishing) แล้ว ในส่วนของ เซอร์วิสอินเทอร์เฟซก็จะต้องทำการประกาศ เหมือนกับส่วนของที่โมเดล ก่อนที่ในส่วนของเซอร์วิสอิมพลิเมนต์ที่แท่นจะทำการประกาศเหมือนกับส่วนของบิสซิเนสเซอร์เซอร์วิส หรือที่เรียกว่าการแมปปิง (Mapping)



รูปที่ 2-35 การประกาศและค้นหาเอกสารฉบับวิเอสดีแอลในยูดีดีไอ

2.6 คอม86 (COM86)

ชุดพัฒนาคอม86 เป็นชุดเครื่องมือสำหรับใช้ในการพัฒนาอุปกรณ์ทางด้านระบบฝังตัวที่อยู่บนพื้นฐานการพัฒนาโดยใช้แพลตฟอร์มที่มีไมโครโปรเซสเซอร์ตระกูล x86 ซึ่งเป็นที่คุ้นเคยกับผู้ใช้งานในปัจจุบันเป็นส่วนประกอบหลักในการทำงาน

2.6.1 จุดเด่นของคอม86

2.6.1.1 ใช้ไมโครคอนโทรลเลอร์ที่มีความสามารถสูง

ภายในชุดพัฒนาคอม86 มีบอร์ดคอม86 ซึ่งได้เลือกใช้ไมโครคอนโทรลเลอร์ Am186CC ของบริษัทเอเอ็มดี (AMD) ซึ่งเป็นไมโครคอนโทรลเลอร์ขนาด 16 บิตทำงานอยู่ที่ความเร็ว 24 เมกะเฮิร์ต (MHz) ทำให้มีความสามารถในการประมวลผลต่างๆเพิ่มมากขึ้นเมื่อเทียบกับการพัฒนาโดยใช้ไมโครคอนโทรลเลอร์ขนาด 8 บิตที่นิยมใช้กันอยู่ในปัจจุบัน นอกจากนี้ภายในตัวไมโครคอนโทรลเลอร์เองยังได้รวมเอาอุปกรณ์พื้นฐานต่างๆที่จำเป็นในการใช้งาน เช่น ดีเรียมคอมโทรลเลอร์ (DRAM controller), อินเทอร์รัพคอนโทรลเลอร์ (Interrupt controller), ยูอาร์ที (UART) เข้าไว้ภายในตัวไมโครคอนโทรลเลอร์ซึ่งทำให้ง่ายต่อการออกแบบอุปกรณ์เพื่อทำเป็นสินค้าภายหลังการพัฒนาตัวอย่างเช่น การนำบอร์ดคอม86 ไปพัฒนาเป็นอุปกรณ์อย่างหนึ่งที่ต้องการควบคุมการทำงานผ่านทางยูเอสบีพอร์ต (USB port) จะสามารถทำได้โดยง่ายเนื่องจากไม่ต้องการออกแบบวงจรควบคุมการทำงานในส่วนยูเอสบี นี้เพิ่มอีก เพียงแต่ออกแบบวงจรการทำงานส่วนอื่นที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

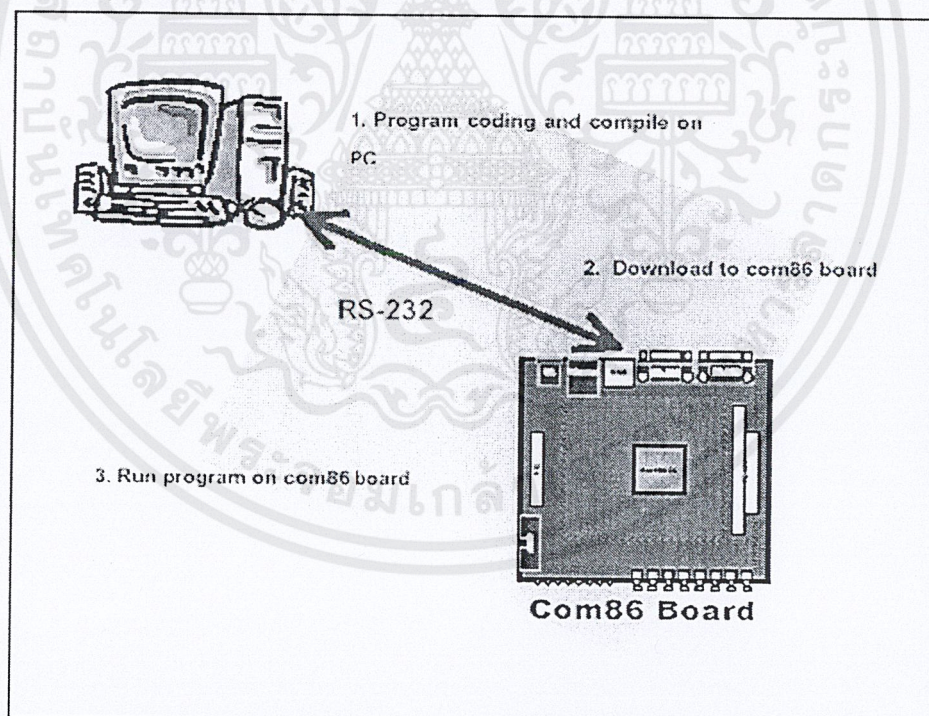
เพิ่มเติมทำให้ทุ่นเวลาในการพัฒนา เป็นต้น นอกจากนี้ภายในไมโครคอนโทรลเลอร์ยังเพิ่มส่วนโมดูล (Module) ทางด้านการสื่อสารต่างๆ เข้าไปภายในตัว ทำให้เป็นไมโครคอนโทรลเลอร์ที่มีความสามารถเด่นทางด้าน การสื่อสารเหมาะกับการพัฒนาอุปกรณ์ที่ต้องใช้คุณสมบัติการสื่อสารต่างๆ ซึ่งจะสามารถทำได้ง่ายเมื่อเทียบกับการใช้ไมโครคอนโทรลเลอร์ขนาดเล็กแบบเดิมซึ่งมีข้อจำกัดต่างๆ อยู่มากมาย

2.6.1.2 มีเครื่องมือในการพัฒนาโปรแกรมมาก

เนื่องจากการพัฒนาโปรแกรมโดยใช้ชุดพัฒนาคอม86 นั้นเป็นเสมือนกับการพัฒนาโปรแกรมบนของคอมพิวเตอร์ส่วนบุคคลทั่วไป ดังนั้นจึงสามารถนำโปรแกรมเครื่องมือต่างๆที่ใช้พัฒนาโปรแกรมสำหรับคอมพิวเตอร์ส่วนบุคคลมาใช้งานได้มากมาย และโปรแกรมเครื่องมือเหล่านี้ก็เป็นที่ยุ่เคยของนักพัฒนาจึงทำให้สามารถพัฒนาอุปกรณ์ได้อย่างรวดเร็ว ลดต้นทุนในการพัฒนาต่างๆได้เป็นจำนวนมากนอกจากนี้ภายในชุดพัฒนายังมีชุดซอฟต์แวร์เครื่องมือต่างๆที่จำเป็นสำหรับช่วยในการพัฒนาโปรแกรมด้วยอีกทางหนึ่ง

2.6.1.3 ง่ายในการเรียนรู้และพัฒนา

บอร์ดคอม86 ภายในชุดพัฒนาคอม86 นี้มีสถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบ x86 ซึ่งเป็นสถาปัตยกรรมเดียวกับคอมพิวเตอร์ส่วนบุคคลทั่วไป โดยมีชุดคำสั่งที่เข้ากันได้กับชุดคำสั่งของ 80286 (Real mode) ทำให้ง่ายต่อการเรียนรู้ในการพัฒนาโปรแกรมต่างๆ และยังสามารถพัฒนาและทดสอบซอฟต์แวร์ต่างๆได้บนคอมพิวเตอร์ ก่อนที่จะทดสอบกับบอร์ดคอม86 ทำให้สามารถพัฒนาอุปกรณ์ทางด้านระบบฝังตัวต่างๆได้อย่างรวดเร็ว ซึ่งรูปแบบการพัฒนาที่มีลักษณะเป็นไปดังภาพ

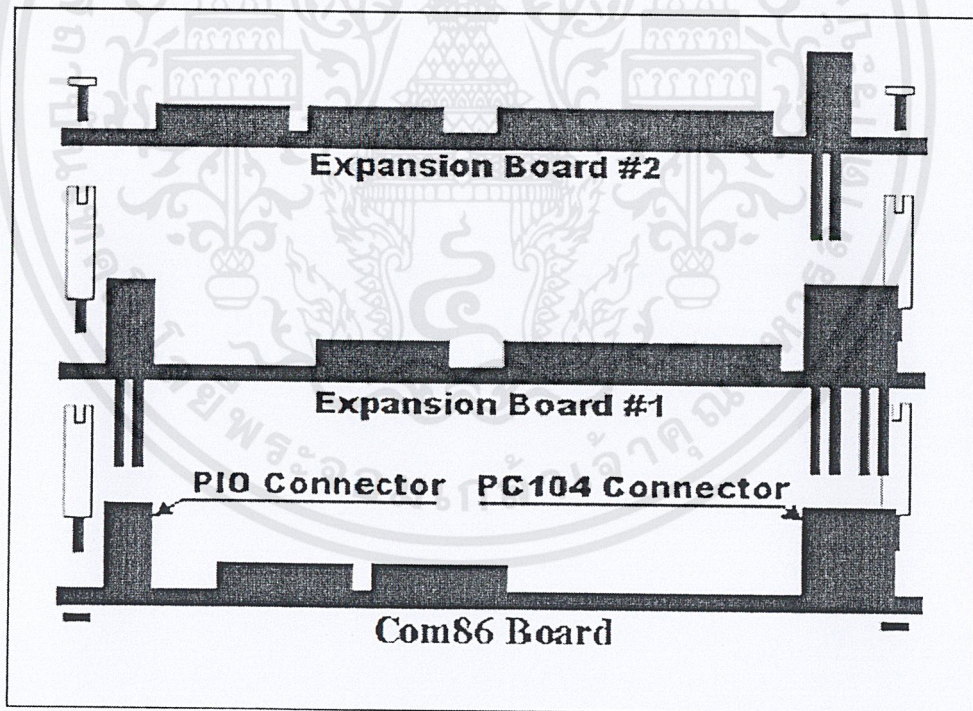


รูปที่ 2-36 แสดงถึงรูปแบบการพัฒนาโปรแกรมโดยใช้บอร์ดคอม86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1.4 มีความสามารถในการเพิ่มขยายได้

ในส่วนของการเพิ่มขยายของฮาร์ดแวร์ (Hardware) สามารถทำได้โดยผ่านทางคอนเนคเตอร์ (Connector) ของบอร์ดในรูปแบบของมาตรฐาน (ซีพีหนึ่งศูนย์สี่) PC/104 ซึ่งจะมีการเชื่อมต่อแบบบัส อินเทอร์เฟซ (BUS Interface) แบบไอเอสเอบัส (ISA BUS) ขนาด 8 บิต คล้ายกับไอเอสเอบัส ที่มีอยู่ในคอมพิวเตอร์ส่วนบุคคลทั่วไป ซึ่งผู้พัฒนาสามารถออกแบบฮาร์ดแวร์ให้เป็นลักษณะของบอร์ดขยายและนำมาเชื่อมต่อกับบอร์ดคอม86 และเพียงเขียนซอฟต์แวร์ไดรเวอร์ (Software Driver) สำหรับควบคุมการทำงานของส่วนเพิ่มเข้ามาใหม่ซึ่งจะทำงานร่วมกับระบบปฏิบัติการในการรองรับการทำงานของแอปพลิเคชันต่างๆที่เพิ่มเข้ามา นอกจากนี้ตัวบอร์ดคอม86 ยังสามารถเชื่อมต่อกับอุปกรณ์ภายนอกในลักษณะของพีไอโอ (PIO) หรือโปรแกรมเมเบิลอินพุท/เอาต์พุท (Programmable Input/Output) เหมือนกับการใช้งานไมโครคอนโทรลเลอร์รุ่นอื่นๆได้ด้วยเช่นกัน



รูปที่ 2-37 แสดงการเชื่อมต่อของบอร์ดขยายกับบอร์ดคอม86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.2 แอปพลิเคชันเป้าหมายของชุดพัฒนา (Target Application)

จากความสามารถของไมโครคอนโทรลเลอร์ของบอร์ดคอม86 ทำให้การประยุกต์ใช้งานกับแอปพลิเคชันต่าง ๆ นั้น สามารถทำได้มากมาย โดยตัวอย่างแอปพลิเคชันต่างๆที่เหมาะสมแก่การพัฒนาโดยบอร์ดคอม86 มีดังนี้

2.6.2.1 เกี่ยวกับอุตสาหกรรม

ตัวอย่างของอุปกรณ์ควบคุมภายในโรงงาน เช่น การควบคุมเครื่องจักรจากระยะไกล การตรวจวัดสถานะต่างๆ ภายในโรงงานและส่งกลับไปยังศูนย์ควบคุม โดยการใช้บอร์ดคอม86 เป็นอุปกรณ์ที่ช่วยทำหน้าที่ในการสื่อสาร ควบคุมสถานะแวดล้อม หรือเครื่องจักรต่างๆภายในโรงงานและรายงานกลับไปยังศูนย์กลางตามกำหนดเวลา เป็นต้น

2.6.2.2 อุปกรณ์ยูเอสบี (USB Peripheral)

อุปกรณ์ยูเอสบีต่างๆ สามารถพัฒนาได้โดยการใช้บอร์ดคอม86 ซึ่งภายในบอร์ดคอม86 นี้มียูเอสบีคอนโทรลเลอร์ในตัวไมโครคอนโทรลเลอร์ Am186CC สามารถเชื่อมต่อเข้ากับคอมพิวเตอร์ได้ด้วยความเร็ว 12 เมกกะบิตเปอร์เซ็ค (Mbps) ซึ่งการพัฒนาเป็นอุปกรณ์ ยูเอสบี ต่างๆ โดยบอร์ดคอม86 นี้สามารถทำได้โดยไม่ต้องมีการต่อวงจรทางด้านยูเอสบีใดๆ เพิ่มเติมอีก

2.6.2.3 เว็บเซิร์ฟเวอร์ (Embedded web server)

เอ็มเบดเดดเว็บเซิร์ฟเวอร์ (Embedded web server) สามารถสร้างได้บนบอร์ดคอม86 โดยไม่จำเป็นต้องเชื่อมต่อวงจรอื่นๆ เพิ่มเติม ซึ่งบอร์ดคอม86 นี้สามารถสื่อสารกับภายนอกผ่านทางอีเทอร์เน็ตที่อยู่บนบอร์ดด้วยความเร็ว 10 เมกกะบิตเปอร์เซ็ค ทำให้สามารถนำไปประยุกต์ใช้งานในกาออกแบบอุปกรณ์ควบคุมต่างๆ ผ่านเอ็มเบดเดดเว็บเซิร์ฟเวอร์ ที่ทำงานอยู่บนบอร์ดคอม86 ได้โดยง่าย

2.6.2.4 โรโบติก (Robotics)

บอร์ดคอม86 มีไมโครคอนโทรลเลอร์ที่มีประสิทธิภาพสูงในการทำงาน การพัฒนาซอฟต์แวร์สามารถพัฒนาได้ซับซ้อนมากยิ่งขึ้น การเชื่อมต่อขยายทางด้านฮาร์ดแวร์ต่างๆสามารถทำได้โดยง่ายเหมาะสำหรับการพัฒนาหุ่นยนต์ที่มีการทำงานที่ซับซ้อน

2.6.2.5 การควบคุมบ้านหรือสิ่งก่อสร้าง

อุปกรณ์สำหรับควบคุมระบบต่างๆภายในบ้าน หรืออาคารสามารถนำบอร์ดคอม86 ไปประยุกต์ใช้งานได้เช่นกัน โดยใช้เป็นส่วนในการควบคุมการทำงานของอุปกรณ์ต่างๆ หรือคอยตรวจสอบการทำงานของเซนเซอร์ (Sensor) ต่างๆ ภายในบ้าน และแจ้งไปยังเจ้าของบ้านเมื่อเกิดเหตุร้ายขึ้น

นอกจากนี้ตัวไมโครคอนโทรลเลอร์ยังรองรับการนำไปสร้างเป็นอุปกรณ์ต่างๆ ที่เกี่ยวข้องกับงาน

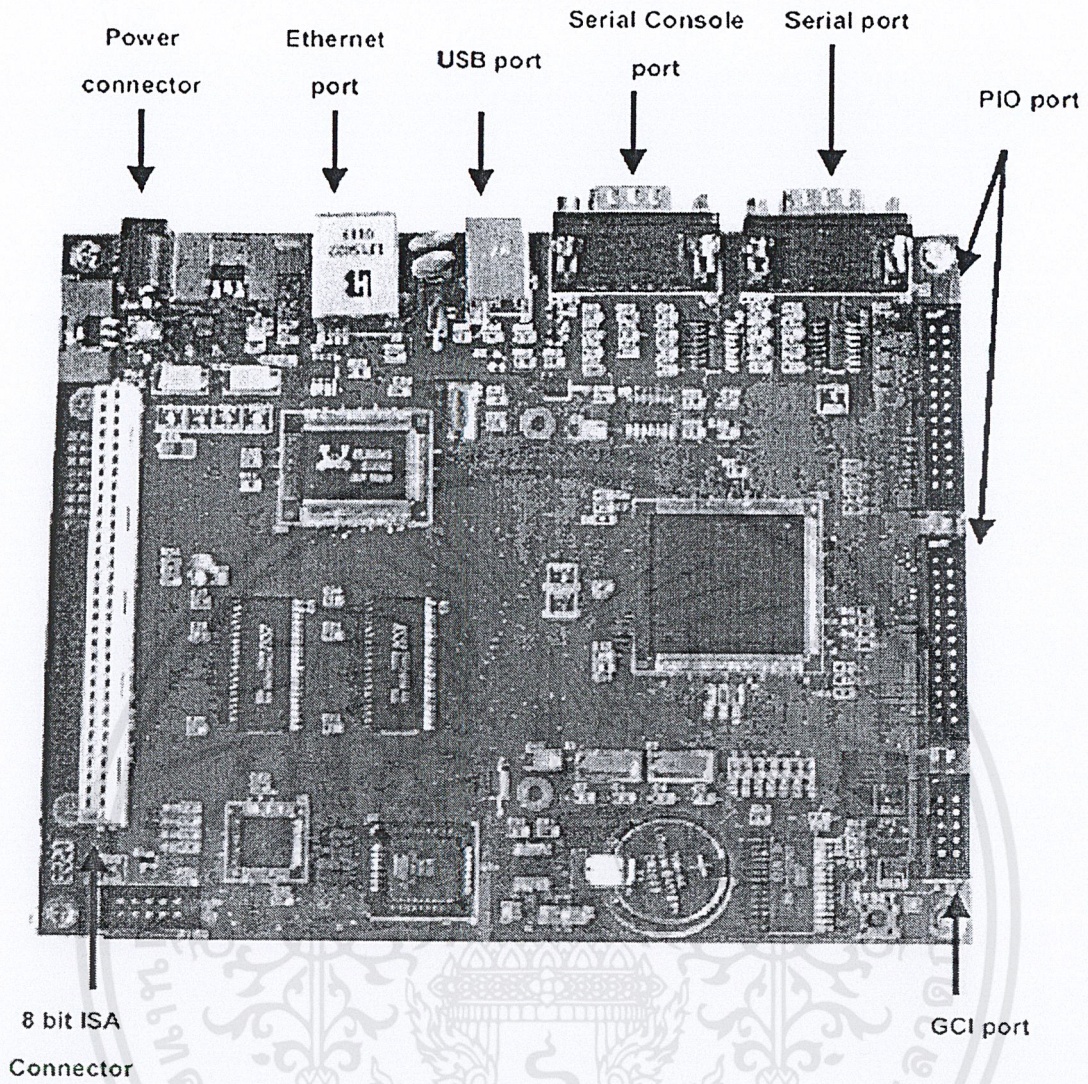
ทางด้าน การสื่อสารต่างๆเช่น ไอเอสดีเอ็นโมเด็มแอนด์เทอร์มินัลแอดแอดเตอร์ (ISDN Modem and Terminal Adaptor), โลเอนด์เร้าเตอร์ (Low-end Router), ไลน์การ์ดแอปพลิเคชัน (Line card application), เอ็กซ์ดีเอสแอลแอปพลิเคชัน (xDSL application), ดิจิตอลคอร์ดโฟน (Digital corded phone) เป็นต้น ซึ่งสามารถนำบอร์ดคอม86 ไปประยุกต์ใช้งานในการออกแบบพัฒนาได้เช่นกัน

2.6.3 คุณสมบัติของบอร์ดคอม86

บอร์ดคอม86มีคุณสมบัติดังนี้

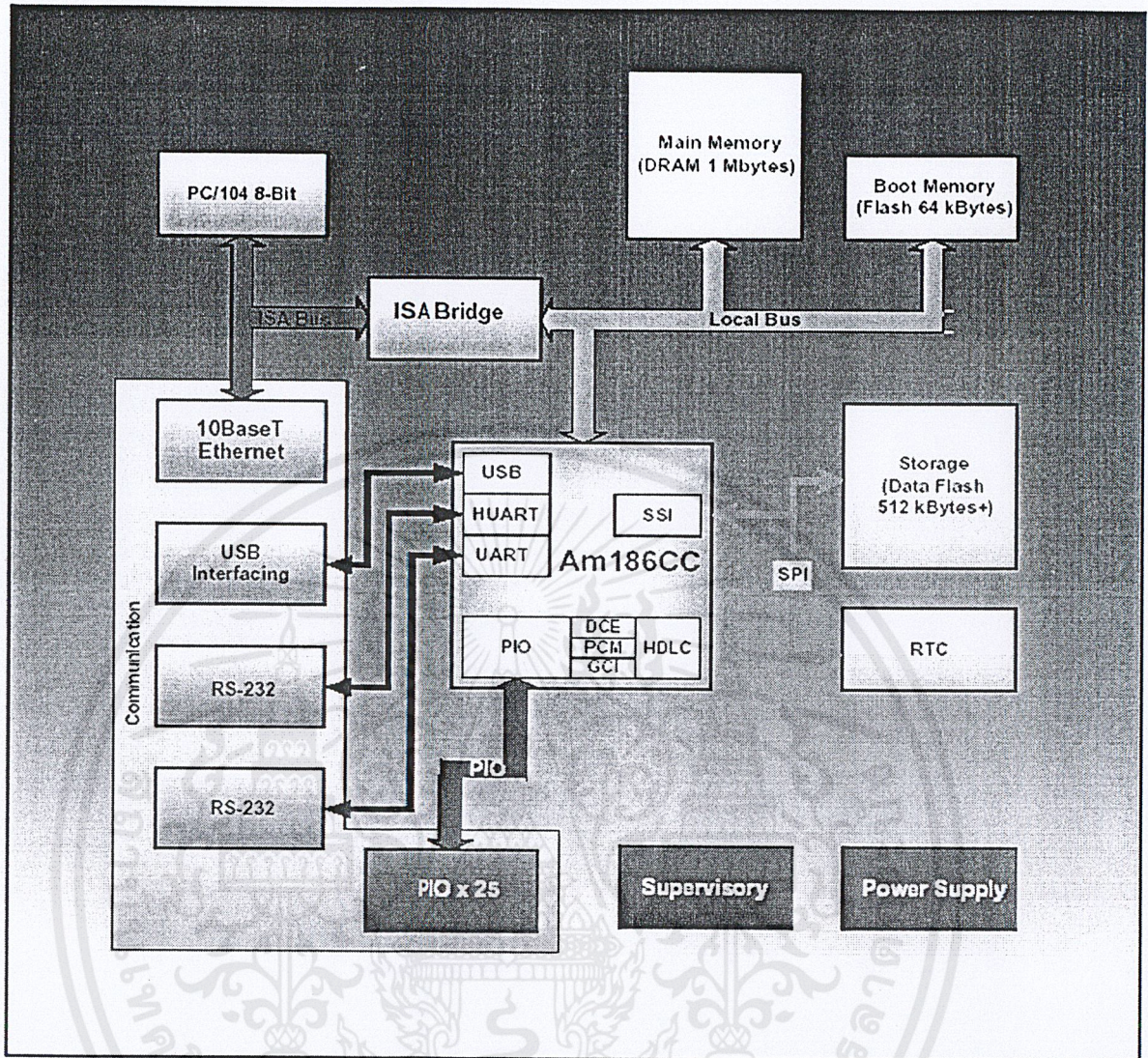
- ซีพียู เอซีหนึ่งแปดหกซีซีซี (CPU Am186CC)
- ดีแรม 1 เมกกะไบต์ (1 MBytes DRAM)
- แฟลตไบออส 64 กิโลไบต์ (64 kBytes Flash BIOS)
- 512 kBytes Serial DataFlash® storage (upgradeable)
- Full duplex IEEE 802.3 compliant on board Ethernet controller with RJ45 connector
- USB peripheral controller version 1.1 with on-board type B USB connector
- Dual RS232-serial port with on-board DB-9 connector
- Four High level Data Link Control (HDLC) channels (multiplex with PIO)
- Four independent Time slot Assigner (TSAs)
- GCI Controller with on-board GCI port for external interface
- Three programmable 16-bit timers
- Real-Time clock
- Watchdog timer and reset controller
- Interrupt Controller (10 external sources)
- 25 programmable I/O
- 8 bit ISA bus in PC/104 form factor
- 3.3 V and 5 V on-board power supply
- LED Indicators for power supply and Ethernet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-38 ตัวอย่างของภาพชุดพัฒนาคอม86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



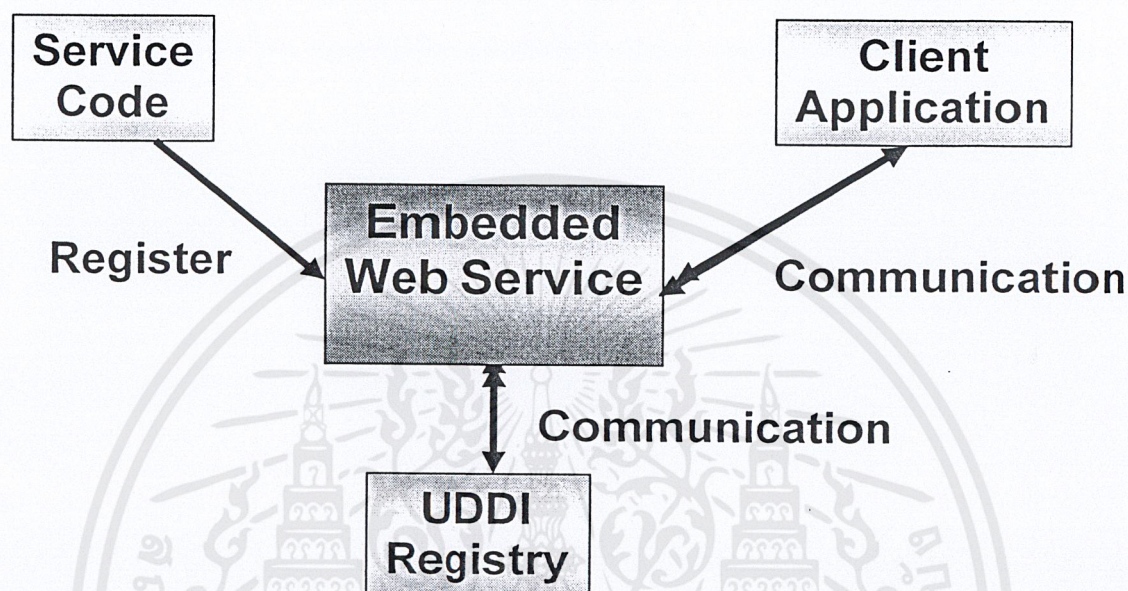
รูปที่ 2-39 บล็อกไดอะแกรมแสดงส่วนประกอบต่างๆของคอม86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบพัฒนาเว็บเซอร์วิสบนระบบฝังตัว

3.1 ขอบเขตของโครงการ



รูปที่ 3-1 คอนเท็กซ์ไดอะแกรม (Context Diagram) ของเว็บเซอร์วิสบนระบบฝังตัว

สำหรับการทำงานของเว็บเซอร์วิสบนระบบฝังตัวจะทำการติดต่อกับ 3 ส่วน คือ

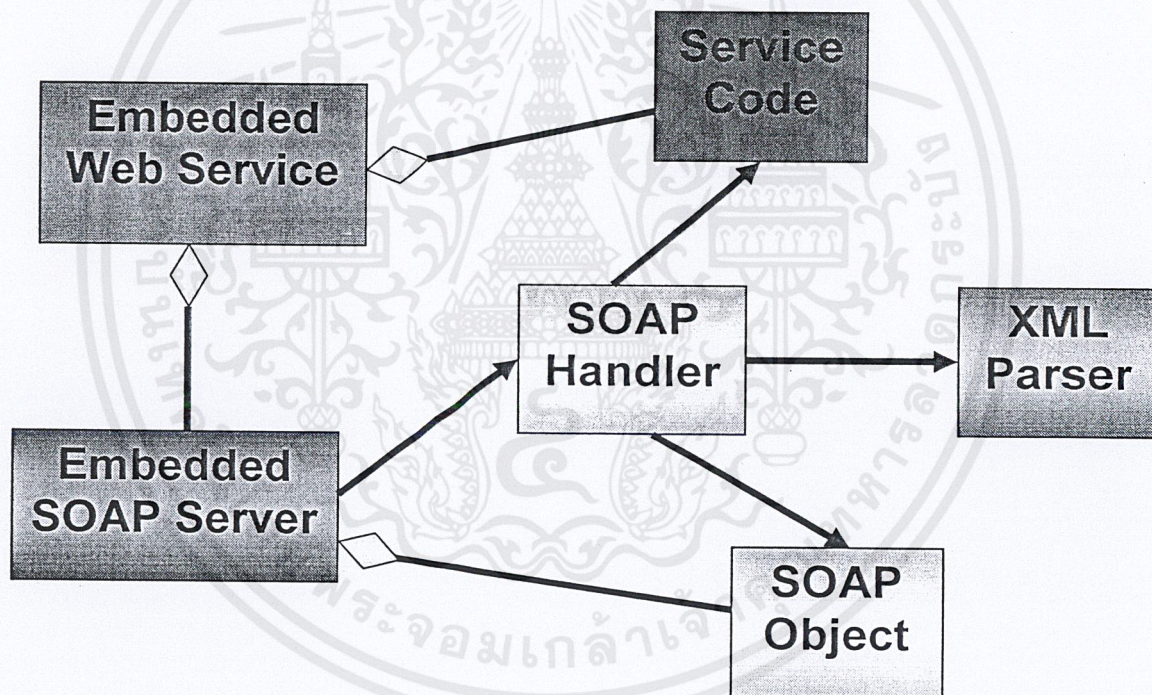
- ส่วนที่ผู้ให้บริการเว็บเซอร์วิสทำการลงทะเบียนเซอร์วิสโค้ด (Service Code) กับระบบเว็บเซอร์วิสบนระบบฝังตัวเพื่อให้ระบบสามารถให้บริการเซอร์วิสนั้นได้
- ส่วนที่ทำการติดต่อกับผู้ใช้ที่ทำการเรียกใช้บริการเว็บเซอร์วิส โดยที่ผู้ใช้บริการเว็บเซอร์วิสจะทำการเรียกใช้งานผ่านตัวแอปพลิเคชันทางฝั่งไคลเอนต์
- ในส่วนของเว็บเซอร์วิสนั้นจะต้องทำการประกาศบริการเว็บเซอร์วิส ไปยังยูดีดีไอรีจิสทรี โดยจะต้องทำการรีจิสเตอร์รายละเอียดต่างๆของตัวบริการ เพื่อให้ผู้ใช้บริการนั้น สามารถเข้ามาทำการค้นหาเพื่อที่จะรู้วิธีการติดต่อกับตัวบริการนั้น แล้วจึงค่อยทำการติดต่อกับบริการ

3.2 โครงสร้างของเว็บเซอร์วิสบนระบบฝังตัว

สำหรับโครงสร้างภายในเว็บเซอร์วิสนี้มีส่วนที่สำคัญ คือ

- โซบเซิร์ฟเวอร์บนระบบฝังตัว (Embedded Server) เป็นส่วนที่จัดการเซคเตอร์เอชทีทีพีและการติดต่อผ่านทางโพรโตคอลทีซีพีไอพี

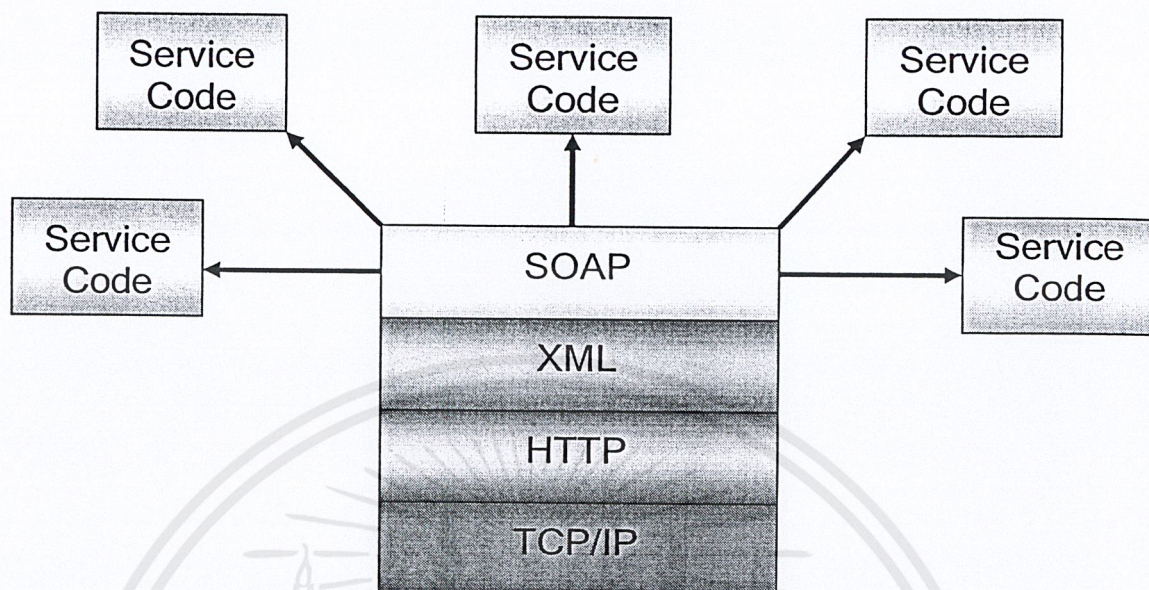
- โขบแฮนเดิลเลอร์ (SOAP Handler) เป็นส่วนที่ทำหน้าที่จัดการโชมแมสเสจ ซึ่งจะแบ่งเป็น 2 ส่วน คือ ส่วนที่ทำหน้าที่จัดการโชมแมสเสจที่ได้รับมาเมื่อมีการเรียกใช้เซอร์วิสโดยถอดโชมแมสเสจ ให้เป็นรายละเอียดและข้อมูลอินพุตในการเรียกใช้เซอร์วิส และส่วนที่ทำหน้าที่จัดการสร้างโชมแมสเสจเพื่อตอบรับการเรียกใช้เซอร์วิสโดยโยนเอาผลลัพธ์ที่ได้จากไฟล์เอาท์พุทที่ได้จากการเรียก เซอร์วิสมาเป็นโชมแมสเสจ
- เซอร์วิสโค้ด (Service Code) เป็นตัวแอปพลิเคชันที่อยู่บนตัวอุปกรณ์คอม86 ซึ่งจะถูกรหัสเรียกใช้งานโดย ส่วนของโชมแฮนเดิลเลอร์
- โชมออบเจกต์ (SOAP Object) เป็นส่วนที่ทำหน้าที่เสมือนเป็นส่วนประกอบของโชมแมสเสจซึ่งจะถูกเรียกใช้โดยโชมแฮนเดิลเลอร์และเป็นส่วนหนึ่งของโชมเซิร์ฟเวอร์
- เอ็กซ์เอ็มแอลพาร์เซอร์ (XML Parser) จะทำหน้าที่ในการแปลเอกสารเอ็กซ์เอ็มแอล โดยที่จะถูกโชมแฮนเดิลเลอร์ นั้นทำการเรียกใช้โชมแฮนเดิลเลอร์ จะทำหน้าที่ในการจัดการโชม โดยจะถูกเรียกใช้งานโดยโชมเซิร์ฟเวอร์ในระบบฝังตัว



รูปที่ 3-2 แสดงโครงสร้างภายในระบบเว็บเซอร์วิสบนระบบฝังตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างทางสถาปัตยกรรมของเว็บเซอร์วิสบนระบบฝังตัว



รูปที่ 3-3 โครงสร้างทางสถาปัตยกรรมของเว็บเซอร์วิสบนระบบฝังตัว

3.3 การตอบสนองของระบบ

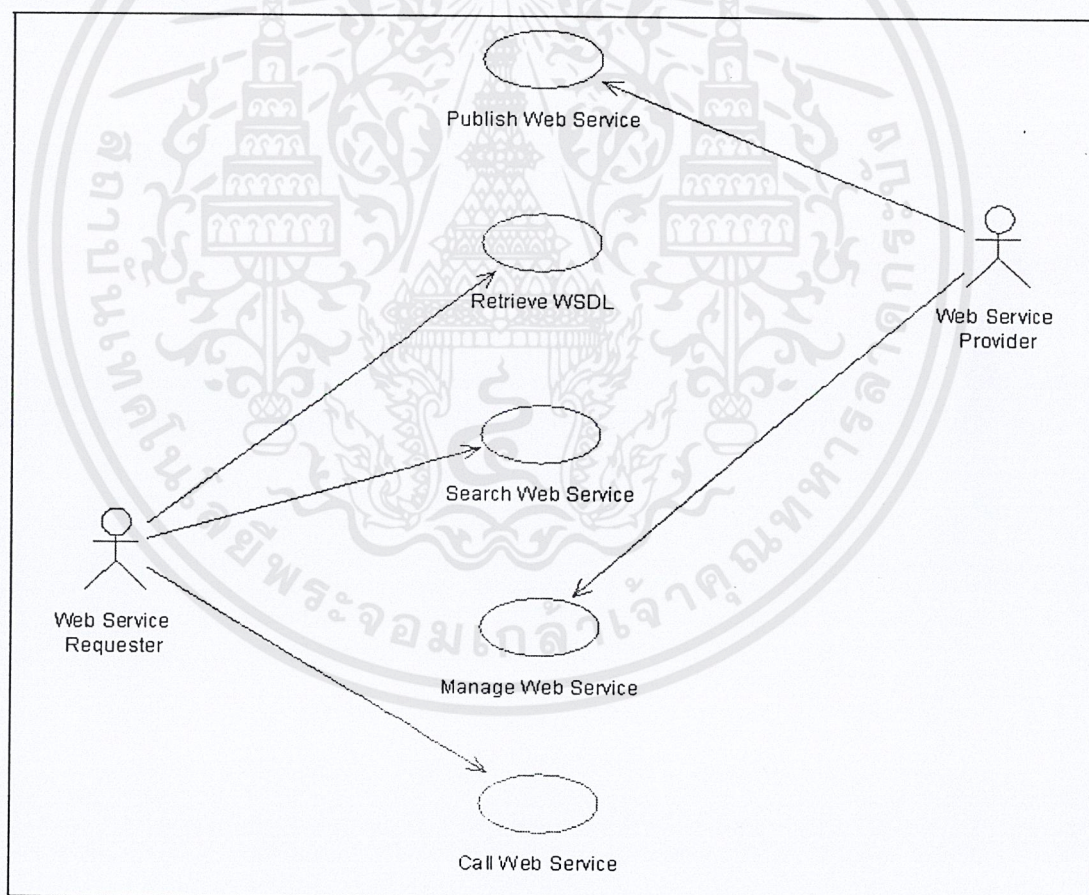
เหตุการณ์	อินพุต/เอาต์พุต
1. การติดต่อเรียกใช้งานเว็บเซอร์วิสบนระบบฝังตัว	<p><u>อินพุต</u></p> <ul style="list-style-type: none"> - ทำการรับเอชทีทีพีรีเควส (HTTP Request) จากตัวแอปพลิเคชันของผู้ใช้บริการ <p><u>เอาต์พุต</u></p> <ul style="list-style-type: none"> - ทำการส่งเอชทีทีพีเรสพอน (HTTP Response) ไปยังตัวแอปพลิเคชันของผู้ใช้บริการ
2. การลงทะเบียนของแอปพลิเคชันไปยังเว็บเซอร์วิสบนระบบฝังตัว	<p><u>อินพุต</u></p> <ul style="list-style-type: none"> - ทำการส่งข้อมูลในส่วนของตัวบริการอิมพลีเม้นท์ (Service Implementation) ไปยังเว็บเซอร์วิสบนระบบฝังตัว <p><u>เอาต์พุต</u></p> <ul style="list-style-type: none"> - รายละเอียดของข้อมูลต่างๆของบริการที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	ลงทะเบียนไป เพื่อเป็นประโยชน์ในการเพิ่ม แก้ไข หรือ เปลี่ยนแปลง ข้อมูลของบริการในครั้งต่อไป
3. การติดต่อเพื่อประกาศเว็บเซอร์วิสบนระบบฝังตัวไปยังตัวยูทิลิตี้โอริจิสทรี	<p>อินพุท</p> <ul style="list-style-type: none"> - ข้อความร้องขอ (Request Message) รายละเอียดที่เกี่ยวกับบริการที่เราต้องการจะติดต่อด้วย <p>เอาต์พุท</p> <ul style="list-style-type: none"> - รายละเอียดที่เกี่ยวกับบริการที่เราต้องการจะติดต่อด้วย โดยจะส่งเซอร์วิสอินเทอร์เฟส ซึ่งถูกกำหนดโดยตัวเอกสารฉบับวีเอสดีแอล (WSDL)

ตารางที่ 3-1 รายการเหตุการณ์ที่เกิดขึ้นกับระบบ

3.4 ยูสเคสไดอะแกรม (Use Case Diagram) ของระบบ



รูปที่ 3-4 ยูสเคสไดอะแกรม (Use Case Diagram) ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประกาศเว็บเซอร์วิส

Use Case	Publish Web Service
ผู้ใช้	ผู้ให้บริการเว็บเซอร์วิส
เงื่อนไข	ผู้ให้บริการเว็บเซอร์วิสจะต้องทำการประกาศบริการเว็บเซอร์วิสไปยังส่วน ยูดีดีไอรีจิสทรี (UDDI Registry) เพื่อให้ผู้ใช้บริการเว็บเซอร์วิสที่ทราบว่ามีบริการใดบ้างและรายละเอียดต่างๆในการใช้บริการ โดยที่ต้องใส่รายละเอียดต่างๆที่เกี่ยวข้องตัวบริการนั้น เช่น ชื่อบริการ, ชื่อเมธอด วิธีการติดต่อกับบริการ เพื่อให้ข้อมูลสำหรับผู้ใช้บริการในการที่จะติดต่อขอใช้บริการนั้นๆต่อไป
ข้อมูลที่เกี่ยวข้อง	ข้อมูลที่ผู้ให้บริการเว็บเซอร์วิสส่งไปประกาศเป็นส่วนหนึ่งของเซอร์วิสอินเทอร์เฟซที่กำหนดโดยเอกสารฉบับวเอสดีแอล ซึ่งส่วนของเซอร์วิสอินเทอร์เฟซจะใช้ในการอิมพลิเมนต์ 1 บริการ หรือมากกว่านั้น

ตารางที่ 3-2 รายละเอียดของการประกาศเว็บเซอร์วิส

การค้นหบริการเว็บเซอร์วิส

Use Case	Search Web Service
ผู้ใช้	ผู้ใช้บริการเว็บเซอร์วิส
เงื่อนไข	ผู้ใช้บริการต้องทำการหาข้อมูลเกี่ยวกับบริการที่ให้บริการผ่านยูดีดีไอรีจิสทรี เพื่อค้นหาบริการที่ต้องการเรียกใช้ว่ามีอยู่หรือไม่ ถ้ามีอยู่จะทำการติดต่อกันได้อย่างไร โดยดูในรายละเอียดเอกสารฉบับวเอสดีแอล
ข้อมูลที่เกี่ยวข้อง	ข้อมูลที่ผู้ใช้บริการเว็บเซอร์วิสได้รับจะเป็นข้อมูลในทางธุรกิจและข้อมูลส่วนเซอร์วิสอินเทอร์เฟซที่กำหนดโดยเอกสารฉบับวเอสดีแอล ซึ่งเมื่อนำส่วนเซอร์วิสอินเทอร์เฟซไปรวมกับส่วนเซอร์วิสอิมพลิเมนต์ ก็จะกลายเป็นตัวฉบับวเอสดีแอลที่สมบูรณ์ ซึ่งจะแสดงรายละเอียดต่างๆที่ใช้ในการติดต่อกับบริการนั้นๆ

ตารางที่ 3-3 รายละเอียดของการค้นหบริการเว็บเซอร์วิส

การร้องขอเอกสารฉบับวเอสดีแอล (WSDL) จากผู้ให้บริการเว็บเซอร์วิส

Use Case	Retrieve WSDL
ผู้ใช้	ผู้ใช้บริการเว็บเซอร์วิส
เงื่อนไข	ผู้ใช้ต้องทำการร้องขอเอกสารฉบับวเอสดีแอล จากผู้ให้บริการก่อนจึงจะสามารถได้บริการได้ โดยเอกสารฉบับวเอสดีแอลนั้นจะได้จากตัวยูดีดีไอ เพื่อดูรายละเอียดต่างๆในการทำการติดต่อกับตัวบริการนั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลที่เกี่ยวข้อง	ข้อมูลที่ใช้บริการเว็บเซอร์วิสได้รับจะเป็นข้อมูลส่วนเซอร์วิสอิมพลีเมนต์ที่กำหนดให้ โดยเอกสารฉบับเอสดีแอล ซึ่งเมื่อนำไปรวมกับส่วนของเซอร์วิสอินเทอร์เฟซก็จะเป็นเอกสารฉบับเอสดีแอลที่สมบูรณ์
---------------------	--

ตารางที่ 3-4 รายละเอียดของการร้องขอเอกสารฉบับเอสดีแอลจากผู้ใช้บริการเว็บเซอร์วิส

การจัดการเว็บเซอร์วิส

Use Case	Manage WSDL
ผู้ใช้	ผู้ให้บริการเว็บเซอร์วิส
เงื่อนไข	ผู้ให้ต้องทำการติดต่อกับยูติลิตี้โอ เพื่อทำการล็อกอินเข้าไปจัดการกับรายละเอียดของเว็บเซอร์วิสของตน โดยที่ในการติดต่อกับตัวยูติลิตี้โอ ผู้ให้บริการจะต้องมีไอดี และ คีย์ ที่ใช้ในการล็อกอิน ซึ่งจะเป็นที่ตกลงกันระหว่างผู้ให้บริการเว็บเซอร์วิสกับยูติลิตี้โอ และสามารถที่จะตรวจสอบเกี่ยวกับเว็บเซอร์วิสได้
ข้อมูลที่เกี่ยวข้อง	ก็เป็นคำร้องขอในการตรวจสอบการจัดการเว็บเซอร์วิส รวมถึงไอดีและคีย์ ที่ใช้ในการเข้าไปติดต่อกับตัวยูติลิตี้โอ หรือแอปพลิเคชันที่ต้องการจะติดต่อกับ

ตารางที่ 3-5 รายละเอียดของการจัดการเว็บเซอร์วิส

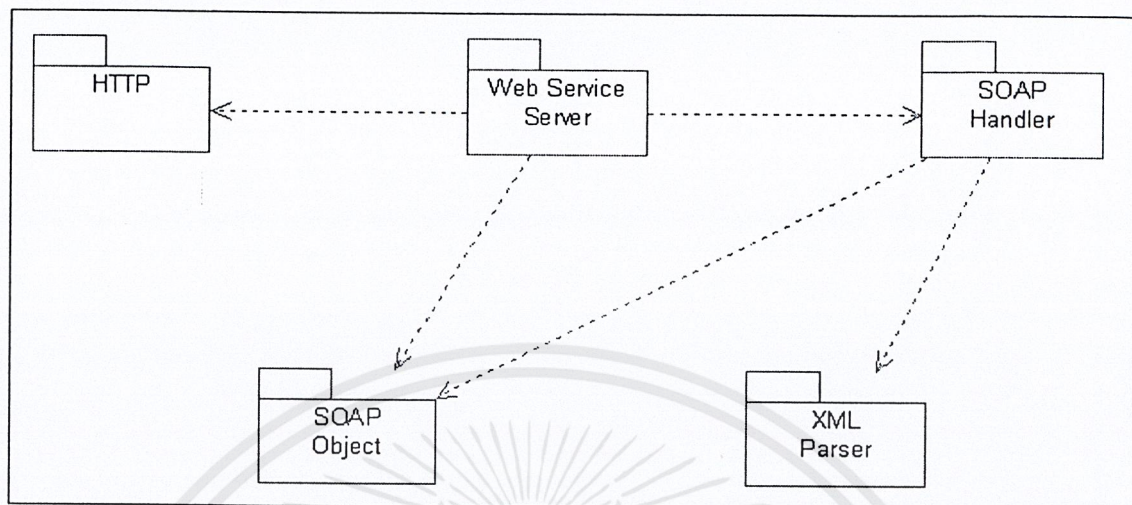
การร้องขอใช้บริการเว็บเซอร์วิส

Use Case	Call Web Service
ผู้ใช้	ผู้ให้บริการเว็บเซอร์วิส
เงื่อนไข	ผู้ให้บริการเว็บเซอร์วิสจะต้องร้องขอตามข้อกำหนดในเอกสารฉบับเอสดีแอลที่ได้มาจากตัวยูติลิตี้โอ และแอปพลิเคชันของผู้ใช้จะต้องสนับสนุนการทำงานของ โปรโตคอล โฉบ และ เอชทีทีพี
ข้อมูลที่เกี่ยวข้อง	เมื่อได้รับเอชทีทีพีรีควีส มา จากนั้นแอปพลิเคชัน โซฟต์แวร์เซอร์จะทำการแปลงให้อยู่ในรูปแบบเมสเสจแล้วค่อยทำการดึงรายละเอียดในการเรียกใช้บริการ เพื่อนำมาใช้ในการติดต่อกับตัวบริการนั้นๆ

ตารางที่ 3-6 รายละเอียดของการร้องขอใช้บริการเว็บเซอร์วิส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 คอมโพเนนต์ไคอะแกรม (Component Diagram)



รูปที่ 3-5 แสดงคอมโพเนนต์ไคอะแกรม

- เว็บเซิร์ฟเวอร์บนระบบฝังตัว (Web Service Server) เป็นส่วนที่จัดการเฮดเดอร์เอชทีทีพีและการติดต่อผ่านทางโพรโตคอลที่ซีพีไอพีโดยเรียกใช้ส่วนของเอชทีทีพี และส่วนจัดการโซบแมสเสจโดยการเรียกใช้ส่วนโซบแฮนเดิลเลอร์และส่วนของโซบออปเจกต์
- ส่วนเอชทีทีพี เป็นส่วนที่จัดการเฮดเดอร์เอชทีทีพีและการติดต่อผ่านทางโพรโตคอลที่ซีพีไอพี
- โซบแฮนเดิลเลอร์ (SOAP Handler) เป็นส่วนที่ทำหน้าที่จัดการโซบแมสเสจ ซึ่งจะแบ่งเป็น 2 ส่วน คือ ส่วนที่ทำหน้าที่จัดการโซบแมสเสจที่ได้รับมาเมื่อมีการเรียกใช้เซอร์วิสโดยถอดโซบแมสเสจให้เป็นรายละเอียดและข้อมูลอินพุตในการเรียกใช้เซอร์วิส และส่วนที่ทำหน้าที่จัดการสร้างโซบแมสเสจเพื่อตอบรับการเรียกใช้เซอร์วิสโดยนำเอาผลลัพธ์ที่ได้จากไฟล์เอาท์พุทที่ได้จากการเรียกเซอร์วิสมาเป็นโซบแมสเสจ
- โซบออปเจกต์ (SOAP Object) เป็นส่วนที่ทำหน้าที่เสมือนเป็นส่วนประกอบของโซบแมสเสจซึ่งจะถูกเรียกใช้โดยโซบแฮนเดิลเลอร์และเป็นส่วนหนึ่งของโซบเซิร์ฟเวอร์
- เอ็กซ์เอ็มแอลพาร์เซอร์ (XML Parser) จะทำหน้าที่ในการแปลเอกสารเอ็กซ์เอ็มแอล โดยที่จะถูกโซบแฮนเดิลเลอร์นั้นทำการเรียกใช้

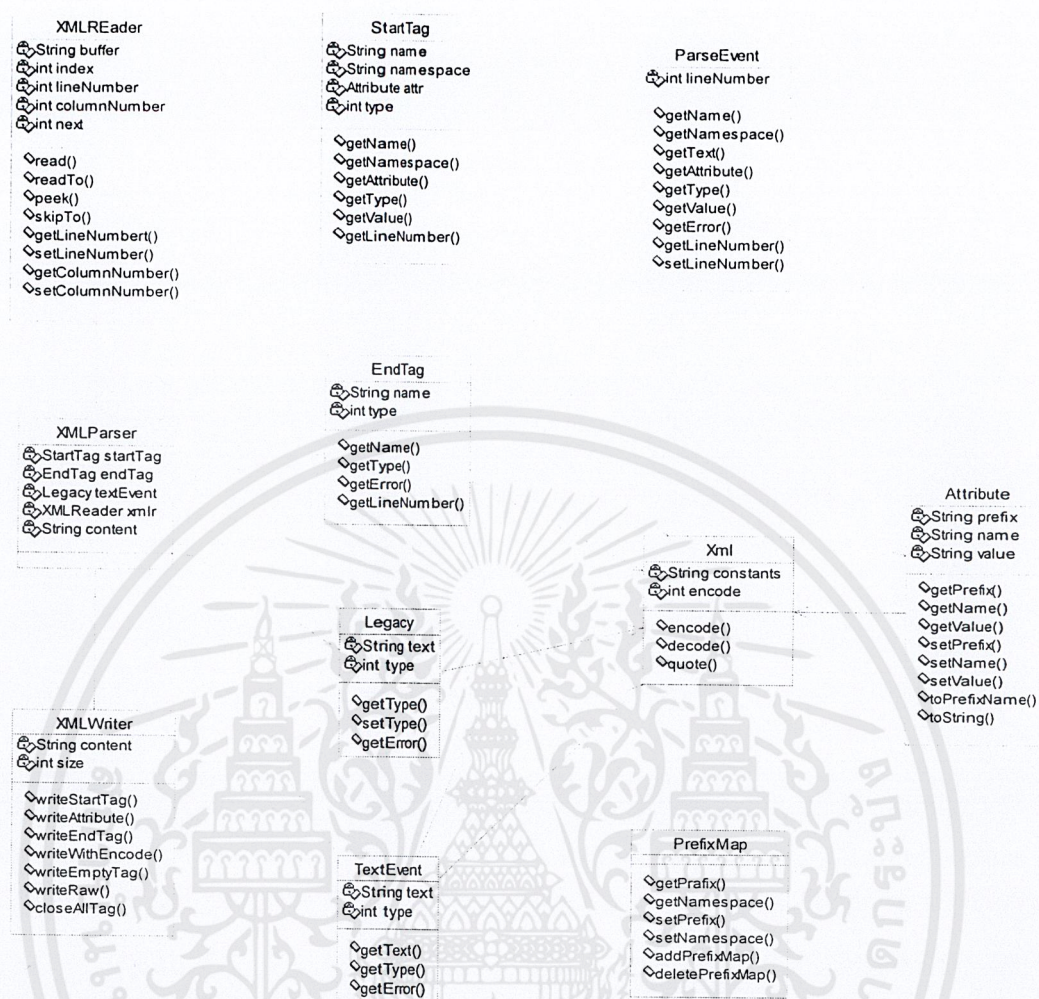
- คลาสบอดี้ (Body Class) เป็นคลาสที่ใช้แทนส่วนเนื้อหาของโซบ (SOAP Body) ภายในโซบแมสเซจ ซึ่งจะมีแอททริบิวต์ คือ คลาส Method และ คลาสแอททริบิวต์แฮนเดิลเลอร์ ซึ่งจะมีฟังก์ชันในการกำหนดค่า และเรียกค่าคืนกลับจากคลาสที่เป็นสมาชิกส่วนตัวในคลาส ฟังก์ชันในการสร้างโซบแมสเซจ และฟังก์ชันในการนำโซบแมสเซจ มาสร้างเป็นคลาส
- คลาสเมทอด (Method Class) เป็นคลาสที่ใช้แทนส่วนเมทอดของส่วนเนื้อหาของโซบ ภายในโซบแมสเซจ ซึ่งจะมีแอททริบิวต์ คือ คลาสเมทอด และ คลาสแอททริบิวต์แฮนเดิลเลอร์ ซึ่งจะมีฟังก์ชันในการกำหนดค่า และเรียกค่าคืนกลับจากคลาสที่เป็นสมาชิกส่วนตัวในคลาส ฟังก์ชันในการสร้างโซบแมสเซจ และฟังก์ชันในการนำโซบแมสเซจ มาสร้างเป็นคลาส
- คลาสพารามิเตอร์ (Parameter Class) เป็นคลาสที่ใช้แทนส่วนพารามิเตอร์ภายในคลาสเมทอด ซึ่งจะมีแอททริบิวต์ คือ คลาสพารามิเตอร์เอนทิตี (ParameterEntries Class) และ คลาสแอททริบิวต์แฮนเดิลเลอร์ ซึ่งจะมีฟังก์ชันในการกำหนดค่า และเรียกค่าคืนกลับจากคลาสที่เป็นสมาชิกส่วนตัวในคลาส ฟังก์ชันในการสร้างโซบแมสเซจ และฟังก์ชันในการนำโซบแมสเซจ มาสร้างเป็นคลาส
- คลาสฟอลท์ (Fault Class) เป็นคลาสที่ใช้เก็บค่าความผิดพลาดต่าง ๆ ได้แก่ ใ้ค้บอกความผิดพลาด, ข้อความบอกความผิดพลาด, ข้อความระบุสิ่งที่ทำให้เกิดความผิดพลาด, เอลเลอเมนต์ภายในที่ใช้ระบุความผิดพลาด ซึ่งจะมีฟังก์ชันที่ใช้ในการกำหนดค่าให้กับสมาชิกของคลาส และเรียกค่าคืนกลับจากสมาชิกของคลาส และฟังก์ชันที่ใช้ในการสร้างโซบแมสเซจ
- คลาสแอททริบิวต์แฮนเดิลเลอร์ (AttributeHandler Class) เป็นคลาสที่ใช้ในการจัดการกับส่วนของแอททริบิวต์ ซึ่งจะมีฟังก์ชันที่ใช้ในการเรียกขอค่าของแอททริบิวต์ และฟังก์ชันเรียกขอค่าเนมสเปซจากค่าของส่วนนำหน้า (Prefix)
- คลาสคอนเทน (Constants Class) เป็นคลาสที่ใช้เก็บค่าคงที่ต่าง ๆ ที่จะนำมาใช้ภายใน SOAP Message
- คลาสซีเรียลไรเซอร์ (Serializer Class) เป็นคลาสที่ใช้ในการแปลงชนิดของข้อมูลให้เป็นค่าที่ถูกกำหนดรูปแบบภายในโซบแมสเซจ ตามข้อกำหนดในส่วนของการเข้ารหัสของโซบ ซึ่งจะมีฟังก์ชันที่ใช้ในการแปลงชนิดจากชนิดของข้อมูลพื้นฐาน, ชนิดข้อมูลแบบอะเรย์ และชนิดข้อมูลแบบคลาสหรือโครงสร้าง ให้เป็นรูปแบบของการเข้ารหัสของโซบ
- คลาสดีซีเรียลไรเซอร์ (Deserializer Class) เป็นคลาสที่ใช้ในการแปลงค่าที่ถูกกำหนดรูปแบบภายในโซบแมสเซจตามข้อกำหนดในส่วนของการเข้ารหัสของโซบให้เป็นชนิดของข้อมูล ซึ่งจะมีฟังก์ชันที่ใช้ในการแปลงรูปแบบของการเข้ารหัสของโซบ ให้เป็นชนิดจากชนิดของข้อมูลพื้นฐาน, ชนิดข้อมูลแบบอะเรย์ และชนิดข้อมูลแบบคลาสหรือโครงสร้าง
- คลาสโซบพาร์เซอร์ (SOAPParser Class) เป็นคลาสที่ใช้ในการตรวจสอบโครงสร้างของเอกสารโซบว่าถูกต้องตามมาตรฐานของโซบ หรือไม่โดยจะมีการเรียกใช้งานจากคลาสเอ็กซ์เอ็มแอลพาร์เซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(XMLParser Class) ซึ่งจะมีฟังก์ชันที่ใช้ในการตรวจสอบโครงสร้างตามมาตรฐานโซป และฟังก์ชันที่ใช้ในการสร้างโซปออบเจกต์ (SOAP Object) จากโซปแมสเซจ

- คลาสโซปรีเควสแฮนเดิลเลอร์ (SOAPRequestHandler) เป็นคลาสที่ใช้ในการจัดการโซปแมสเซจ ที่เข้ามาเพื่อเรียกใช้บริการเว็บเซอร์วิส โดยจะมีฟังก์ชันที่ใช้ในการตรวจสอบรายชื่อ และเมธอดของบริการนั้นว่ามีภายในระบบหรือไม่ ฟังก์ชันที่ใช้ในการกำหนดส่วนกำหนดข้อผิดพลาดของโซป (SOAP Fault) ฟังก์ชันที่เรียกใช้คลาสซีรี่ไรซ์เซอร์ เพื่อทำการแปลงค่าที่ถูกกำหนดรูปแบบภายในโซปแมสเซจตามข้อกำหนดในส่วนของการเข้ารหัสของโซปให้เป็นชนิดของข้อมูล และฟังก์ชันที่ใช้ในการเขียนข้อมูลนำเข้าไปยังไฟล์ และฟังก์ชันที่ใช้ในการตรวจสอบไวยากรณ์ของเอกสารว่าตรงกับมาตรฐานโซป ฟังก์ชันที่ใช้ในการตรวจสอบรายการของพารามิเตอร์นำเข้าว่าตรงกับรายละเอียดของบริการหรือไม่
- คลาสโซปเรสพอนแฮนเดิลเลอร์ (SOAPResponseHandler Class) เป็นคลาสที่ใช้ในการสร้างโซปแมสเซจ เพื่อตอบรับการเรียกใช้บริการเว็บเซอร์วิส โดยจะมีฟังก์ชันที่ใช้ในการสร้างโซปแมสเซจของบริการนั้นโดยดูจากรายละเอียดของบริการนั้น ฟังก์ชันที่ใช้ในการกำหนดส่วนกำหนดข้อผิดพลาดของโซป (SOAP Fault) ฟังก์ชันที่เรียกใช้คลาสซีรี่ไรซ์เซอร์ เพื่อทำการแปลงชนิดของข้อมูลให้เป็นค่าที่ถูกกำหนดรูปแบบภายในโซปแมสเซจ ตามข้อกำหนดในส่วนของการเข้ารหัสของโซป และฟังก์ชันที่ใช้ในการเขียนข้อมูลนำเข้าไปยังไฟล์ และฟังก์ชันที่ใช้ในการตรวจสอบไวยากรณ์ของเอกสารว่าตรงกับมาตรฐานโซป ฟังก์ชันที่ใช้ในการตรวจสอบรายการของผลลัพธ์พารามิเตอร์ว่าตรงกับรายละเอียดของบริการหรือไม่
- คลาสโซปเซิร์ฟเวอร์ (SOAPServer Class) เป็นคลาสที่ใช้ในการให้บริการเว็บเซอร์วิส โดยจะทำหน้าที่ในการรับ SOAP Message เข้ามาแล้วเรียกใช้คลาสโซปรีเควสแฮนเดิลเลอร์ และคลาสโซปเรสพอนแฮนเดิลเลอร์ เพื่อจัดการสร้างโซปออบเจกต์ และตรวจสอบความถูกต้องตามที่ได้กล่าวแล้ว และยังมีหน้าที่ในการเชื่อมต่อกับส่วนของเว็บเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



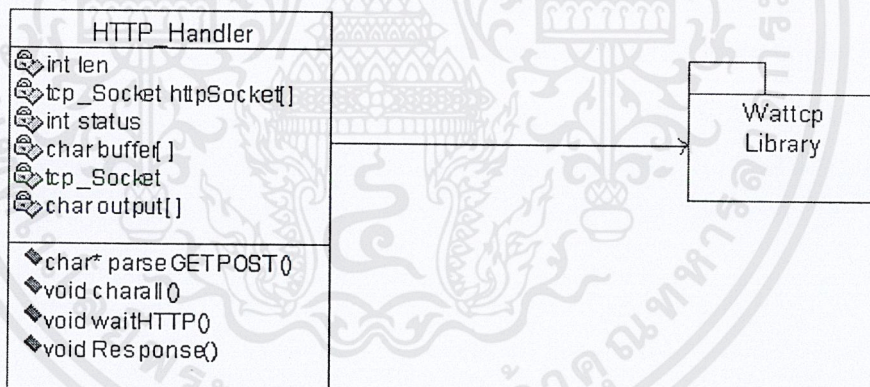
รูปที่ 3-7 คลาสไคอะแกรมในส่วนของเอ็กซ์เอ็มแอลพาร์เซอร์

เอ็กซ์เอ็มแอลพาร์เซอร์

- คลาสเอ็กซ์เอ็มแอลรีดเดอร์ (XMLReader Class) เป็นคลาสที่ใช้ในการจัดการกับข้อมูลเอ็กซ์เอ็มแอล โดยจะทำหน้าที่ในส่วนของการอ่านข้อมูลภาษาเอ็กซ์เอ็มแอล
- คลาสเอ็กซ์เอ็มแอลพาร์เซอร์ (XMLParser Class) เป็นคลาสที่ใช้ในการจัดการกับข้อมูลเอ็กซ์เอ็มแอล โดยจะทำหน้าที่ในส่วนของการแปลข้อมูลภาษาเอ็กซ์เอ็มแอล ซึ่งจะเรียกใช้คลาสเอ็กซ์เอ็มแอลไรท์เทอร์ และ เอ็กซ์เอ็มแอลรีดเดอร์
- คลาสเอ็กซ์เอ็มแอลไรท์เทอร์ (XMLWriter Class) เป็นคลาสที่ใช้ในการจัดการกับข้อมูลเอ็กซ์เอ็มแอล โดยจะทำหน้าที่ในส่วนของการเขียนข้อมูลภาษาเอ็กซ์เอ็มแอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คลาสสตาร์ทแท็ก (StartTag Class) เป็นคลาสที่ใช้ในการจัดการกับแท็กข้อมูลของเอ็กซ์เอ็มแอล โดยทำหน้าที่ในส่วนของตัวสตาร์ทแท็กเอ็กซ์เอ็มแอล เช่น ตรวจสอบความถูกต้องของส่วนสตาร์ทแท็กตามรูปแบบของภาษาเอ็กซ์เอ็มแอล
- คลาสเอนแท็ก (EndTag Class) เป็นคลาสที่ใช้ในการจัดการกับแท็กข้อมูลของเอ็กซ์เอ็มแอล โดยทำหน้าที่ในส่วนของตัวเอนแท็กเอ็กซ์เอ็มแอล เช่น เช็คความถูกต้องของตัวเอนแท็กเอ็กซ์เอ็มแอล
- คลาสเล็กลาจี (Legacy Class) เป็นคลาสที่ใช้ในการจัดการกับข้อมูลของเอ็กซ์เอ็มแอล โดยทำหน้าที่ในส่วนของตัวข้อมูลของเอ็กซ์เอ็มแอลที่มีอยู่แล้ว
- คลาสเท็กซ์อีเวนต์ (TextEvent Class) เป็นคลาสที่ใช้ในการจัดการกับข้อมูลของเอ็กซ์เอ็มแอล โดยทำหน้าที่ในส่วนอีเวนต์ของเท็กซ์เอ็กซ์เอ็มแอลที่ส่งเข้ามา
- คลาสพาสอีเวนต์ (ParseEvent Class) เป็นคลาสที่ใช้ในการจัดการกับข้อมูลของเอ็กซ์เอ็มแอล โดยทำหน้าที่ในแปลงข้อมูลส่วนอีเวนต์ของเท็กซ์เอ็กซ์เอ็มแอลที่ส่งเข้ามา
- คลาสแอททริบิวต์ (Attribute Class) เป็นคลาสที่ใช้ในการจัดการกับข้อมูลของเอ็กซ์เอ็มแอล โดยทำหน้าที่ในแอททริบิวต์ต่างๆของข้อมูลเอ็กซ์เอ็มแอล
- คลาสเอ็กซ์เอ็มแอล (Xml Class) เป็นคลาสที่ใช้ในการจัดการกับข้อมูลของเอ็กซ์เอ็มแอล โดยทำหน้าที่ในส่วนของการเอนโค้ด และ การดีโค้ดข้อมูลภาษาเอ็กซ์เอ็มแอล
- คลาสพรีฟิกแมพ (PrefixMap Class) เป็นคลาสที่ใช้ในการจัดการกับข้อมูลของเอ็กซ์เอ็มแอล โดยทำหน้าที่ในส่วนของพรีฟิก และ นามสเปซข้อมูลภาษาเอ็กซ์เอ็มแอล

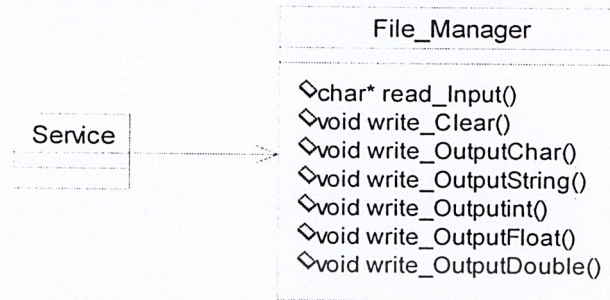


รูปที่ 3-8 คลาสโคอะแกรมในส่วนของเว็บเซิร์ฟเวอร์

ส่วนเว็บเซิร์ฟเวอร์

- คลาสเอชทีทีพีแฮนด์เลอร์ (HTTPHandler Class) เป็นคลาสที่ใช้ในการจัดการข้อมูลในส่วนของโปรโตคอลเอชทีทีพีที่ได้รับเข้ามา แล้วทำการถอดเอาแฮดเดอร์และข้อมูลที่ต้องการออกมาและสร้างผลลัพธ์แล้วส่งกลับไปโดยจะทำการนำส่วนของวัตถุซีพีไลบรารีเข้ามาใช้เนื่องจากเป็นไลบรารีที่ใช้ในการติดต่อกับโปรโตคอลทีซีพีของบอร์ดคอม 86

- วัตชีพีไลบรารีเป็นไลบรารีที่จัดการเกี่ยวกับการติดต่อสื่อสารผ่านทางโพรโทคอลของบอร์ดคอม86 ซึ่งเป็นตัวอย่างแพลตฟอร์มที่เลือกขึ้นมา



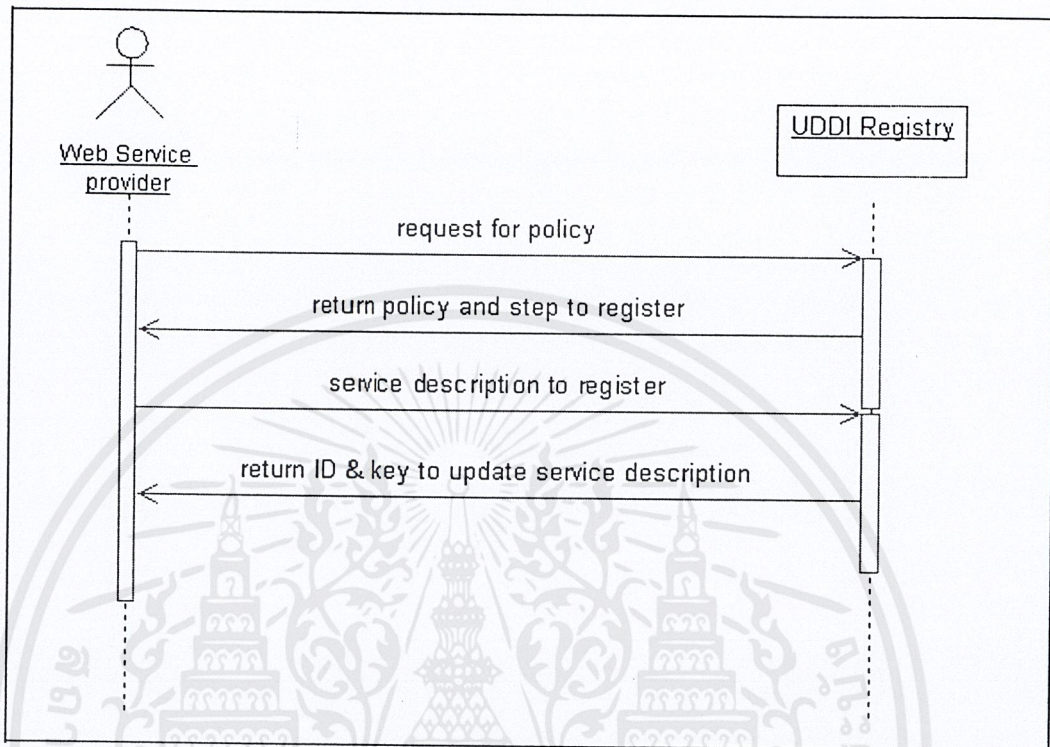
รูปที่ 3-9 คลาสไดอะแกรมในส่วนของเซอร์วิสโค้ด

ส่วนของเซอร์วิสโค้ด

- คลาสไฟล์แมนเนจเจอร์ (FileManage Class) เป็นคลาสที่ใช้ในการติดต่อกับไฟล์ในการเขียนข้อมูลผลลัพธ์ที่ต้องการจะตอบกลับไปอยู่ในรูปของเท็กซ์ไฟล์เพื่อให้ซอฟต์แวร์เซอร์เปลี่ยนให้อยู่ในรูปแบบมาตรฐานซิมเบิลออบเจกต์แอกเซสโพรโทคอลต่อไป
- คลาสเซอร์วิสโค้ด (ServiceCode Class) เป็นคลาสที่ใช้ทำการสร้างขึ้นเพื่อสร้างเป็นเซอร์วิสโดยจะต้องทำการเรียกใช้คลาสไฟล์แมนเนจเจอร์เพื่อเขียนข้อมูลผลลัพธ์ตามชนิดของข้อมูล โดยที่ส่วนประกอบส่วนอื่น ๆ ของเซอร์วิสนั้นผู้ใช้สามารถสร้างขึ้นได้เองตามที่ต้องการ

3.7 ซีเควนส์ไดอะแกรม (Sequence Diagram) ของระบบ

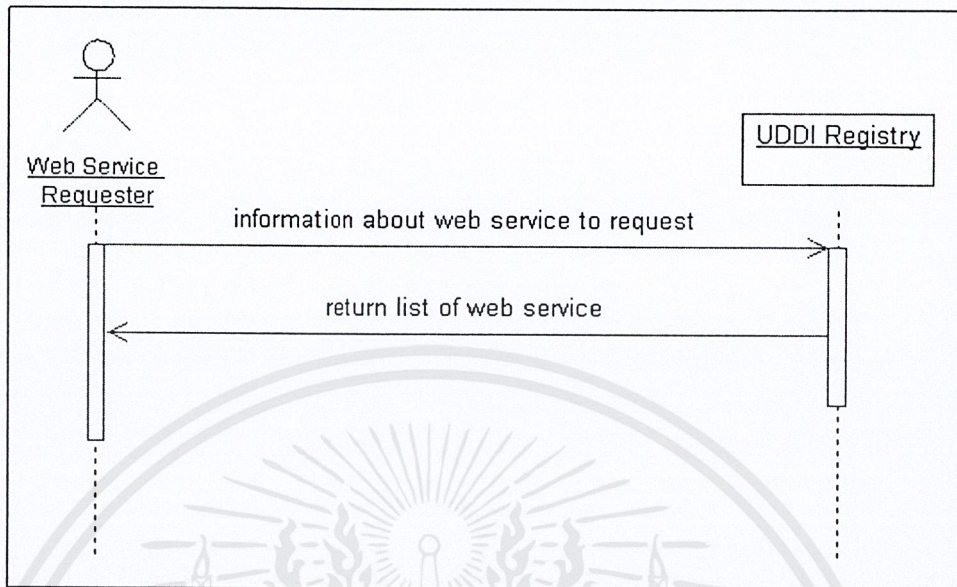
3.7.1 การประกาศเว็บเซอร์วิส (Publish Web Service)



รูปที่ 3-10 ซีเควนส์ไดอะแกรมแสดงการประกาศเว็บเซอร์วิส

อธิบายการทำงานของ การประกาศเว็บเซอร์วิส คือ ทางฝั่งผู้ให้บริการเซอร์วิส (Service Provider) นั้น จะทำการติดต่อกับยูดีดีไอรีจิสทรี (UDDI Registry) โดยการส่งคำร้องสำหรับ โพลีซี่ (Policy Request) หลังจาก ที่ยูดีดีไอได้รับ โพลีซี่รีเคสแล้ว ก็จะทำการส่งโพลีซี่พร้อมขั้นตอนการลงทะเบียนกลับไปให้ผู้ให้บริการ หลังจากนั้นผู้ให้บริการจะทำการลงทะเบียนรายละเอียดของบริการที่ตนเองนั้นให้ผู้ให้บริการ หลังจากลงทะเบียน เรียบร้อยแล้ว ยูดีดีไอก็จะส่งไอดีกับคีย์ (ID & key) ให้กับผู้ให้บริการ เพื่อใช้ในการจัดการบริการในครั้งต่อไป

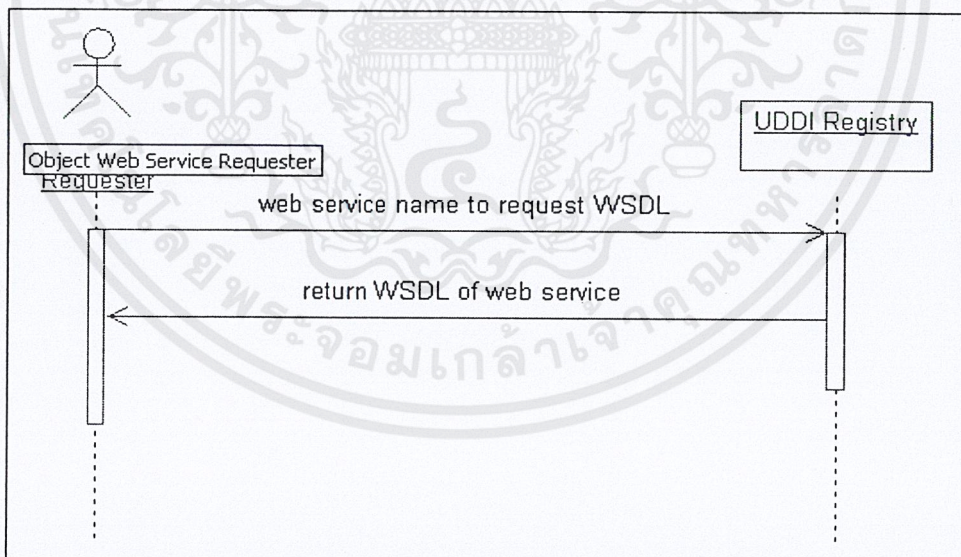
3.7.2 การค้นหาเว็บเซอร์วิส (Find Web Service)



รูปที่ 3-11 ซีควนส์ไดอะแกรมแสดงการค้นหาเว็บเซอร์วิส

อธิบายการทำงานของการทำงานการค้นหาเว็บเซอร์วิส คือ ทางฝั่งของผู้ขอใช้บริการ(Web Service Request) จะทำการติดต่อยูดีดีไอริจิสทรี ส่งคำร้องขอข้อมูลรายละเอียดที่เกี่ยวกับเว็บเซอร์วิสที่ต้องการใช้ หลังจากที่ยูดีดีไอได้รับคำร้องขอจากผู้ใช้แล้ว ก็จะทำการส่งรายละเอียดเกี่ยวกับเว็บเซอร์วิสนั้นกลับมาให้กับผู้ขอใช้บริการ

3.7.3 การร้องขอเอกสารดับบีวเอสดีแอล (Retrieve WSDL)



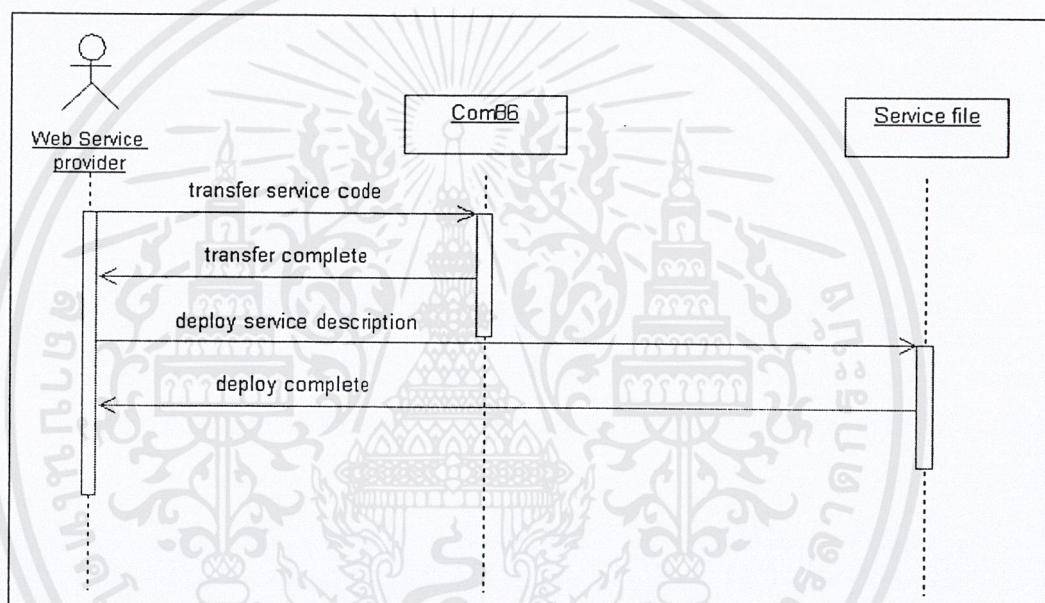
รูปที่ 3-12 ซีควนส์ไดอะแกรม แสดงการร้องขอเอกสารดับบีวเอสดีแอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อธิบายการทำงานของกรร็องขอเอกสารฉบับวีเอสดีแอล คือ ทางฝั่งของผู้ขอใช้บริการ ต้องการเรียกใช้งานบริการ แต่ในการที่จะเรียกใช้บริการนั้น จำเป็นต้องรู้รายละเอียดต่างๆของบริการนั้น เช่น วิธีในการที่จะติดต่อกับบริการนั้น ซึ่งรายละเอียดเหล่านี้ จะอยู่ในรูปของเอกสารฉบับวีเอสดีแอล ซึ่งอยู่ที่ยูติลิตี้โอ ในการขอรายละเอียดเอกสารฉบับวีเอสดีแอลนั้น ก่อนอื่นผู้ขอใช้บริการต้องทำการติดต่อกับยูติลิตี้โอ โดยการส่งชื่อเว็บเซอร์วิสเพื่อขอเอกสารฉบับวีเอสดีแอลของเว็บเซอร์วิสนั้น หลังจากยูติลิตี้โอได้รับรีควิสต์แล้วก็จะส่งเอกสารฉบับวีเอสดีแอลของตัวเว็บเซอร์วิสนั้นกลับไปให้ผู้ขอใช้บริการ

3.7.4 การจัดการเว็บเซอร์วิส (Manage Web Service)

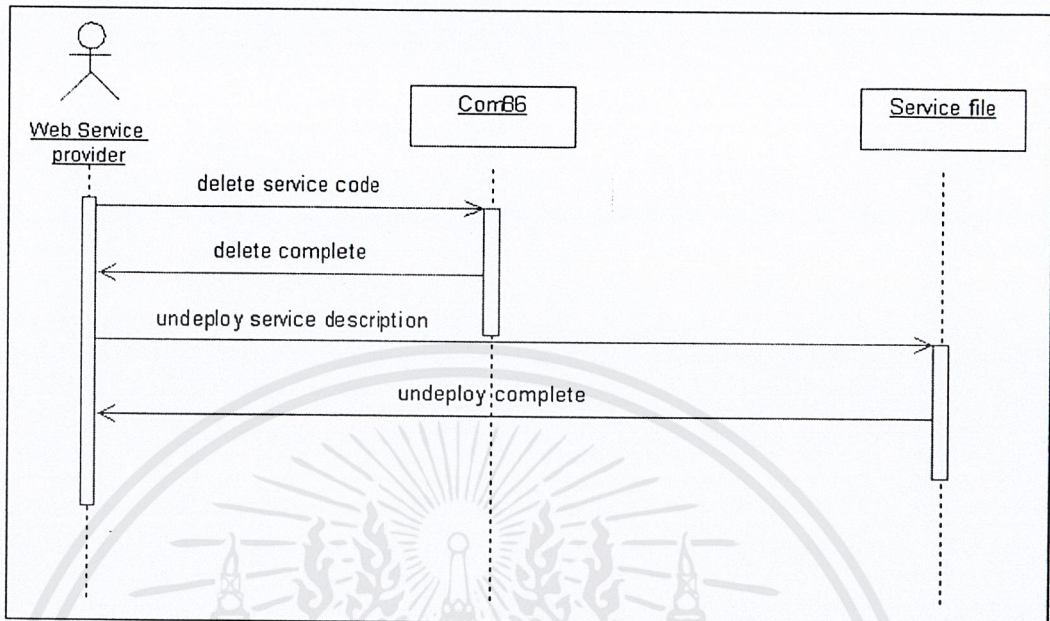
3.7.4.1 การดีพลอยเว็บเซอร์วิส (Deploy Web Service)



รูปที่ 3-13 ซีควนส์ไดอะแกรมแสดงการดีพลอยเว็บเซอร์วิส

อธิบายการทำงานของกรร็องขอเอกสารฉบับวีเอสดีแอล ซึ่ง จะมีส่วนประกอบอยู่ 3 ส่วนด้วยกัน คือ ผู้ให้บริการ เว็บเซอร์วิส (Web Service Provide), คอม (Com86) และ เซอร์วิสไฟล์ (Service file) โดยที่การเริ่มต้นของการดีพลอยเว็บเซอร์วิส คือ เริ่มแรก ผู้ให้บริการจะทำการขอทรานเฟอร์เซอร์วิสโค้ด (Transfer Service Code) หลังจากขอทรานเฟอร์เสร็จแล้ว ฝั่ง คอม86 ก็จะส่งรายละเอียดมาบอกผู้ให้บริการ ซึ่งผู้ใช้จะต้องการทำการดีพลอยรายละเอียดต่างๆของบริการนั้นลงในเซอร์วิสไฟล์ หลังจากดีพลอยเสร็จแล้ว ทางฝั่ง เซอร์วิสไฟล์ก็จะส่งกลับไปบอกผู้ขอใช้บริการว่าการดีพลอยเสร็จสิ้นแล้ว

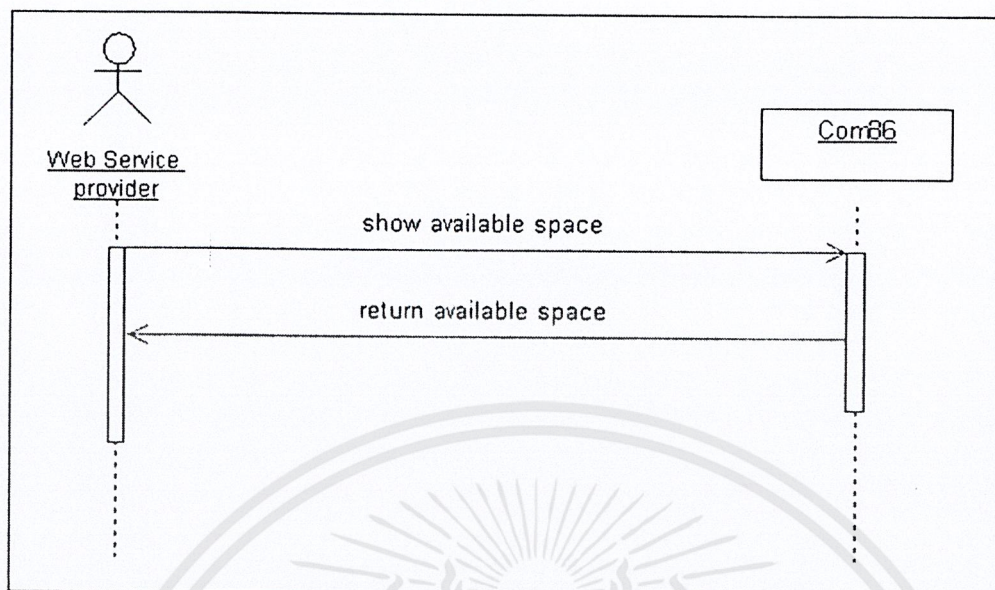
3.7.4.2 การอันดีพลอยเว็บเซอร์วิส (Undeploy Web Service)



รูปที่ 3-14 ซีควนส์ไดอะแกรมแสดงการอันดีพลอยเซอร์วิส

อธิบายการทำงานของ การอันดีพลอยเว็บเซอร์วิส ซึ่งจะมีส่วนประกอบอยู่ 3 ส่วนด้วยกัน คือ ผู้ให้บริการเว็บเซอร์วิส (Web Service Provide), คอม86 (Com86) และ เซอร์วิสไฟล์ (Service file) โดยที่การเริ่มต้นของการอันดีพลอยเว็บเซอร์วิส คือ เริ่มแรก ผู้ให้บริการเว็บเซอร์วิสจะทำการขอคืนเซอร์วิสโค้ด (Delete Service Code) หลังจากขอคืนเสร็จแล้ว ฟังก์คอม86ก็จะส่งรายละเอียดมาบอกผู้ขอใช้บริการ ซึ่งผู้ใช้งานต้องการทำการอันดีพลอยรายละเอียดต่างๆของบริการนั้นลงในเซอร์วิสไฟล์ หลังจากอันดีพลอยเสร็จแล้ว ทางฟังก์เซอร์วิสไฟล์ก็จะส่งกลับไปบอกผู้ขอใช้บริการว่าการอันดีพลอยเสร็จสิ้นแล้ว

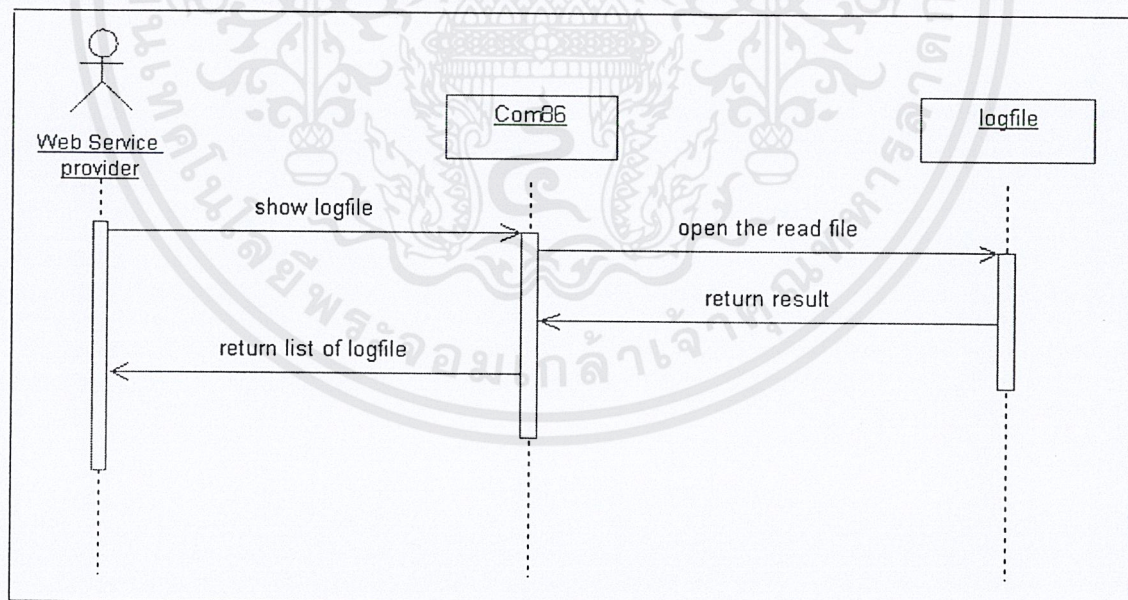
3.7.4.3 การตรวจสอบ (Monitor Available Space)



รูปที่ 3-15 ซีควেনส์ไดอะแกรมแสดงการตรวจสอบ

อธิบายการทำงานของ การตรวจสอบเว็บเซอร์วิส (Monitor Available Space) คือ ผู้ให้บริการเว็บเซอร์วิสจะร้องขอให้ฝั่งคอม86 ส่งข้อมูลสถานะของเว็บเซอร์วิสนั้นมา เพื่อทำการตรวจสอบ หลังจากนั้นคอม86 ก็จะทำการส่งรายละเอียดต่างๆกลับไปให้ผู้ให้บริการเว็บเซอร์วิส

3.7.4.4 การตรวจสอบล็อกไฟล์ (Monitor logfile)

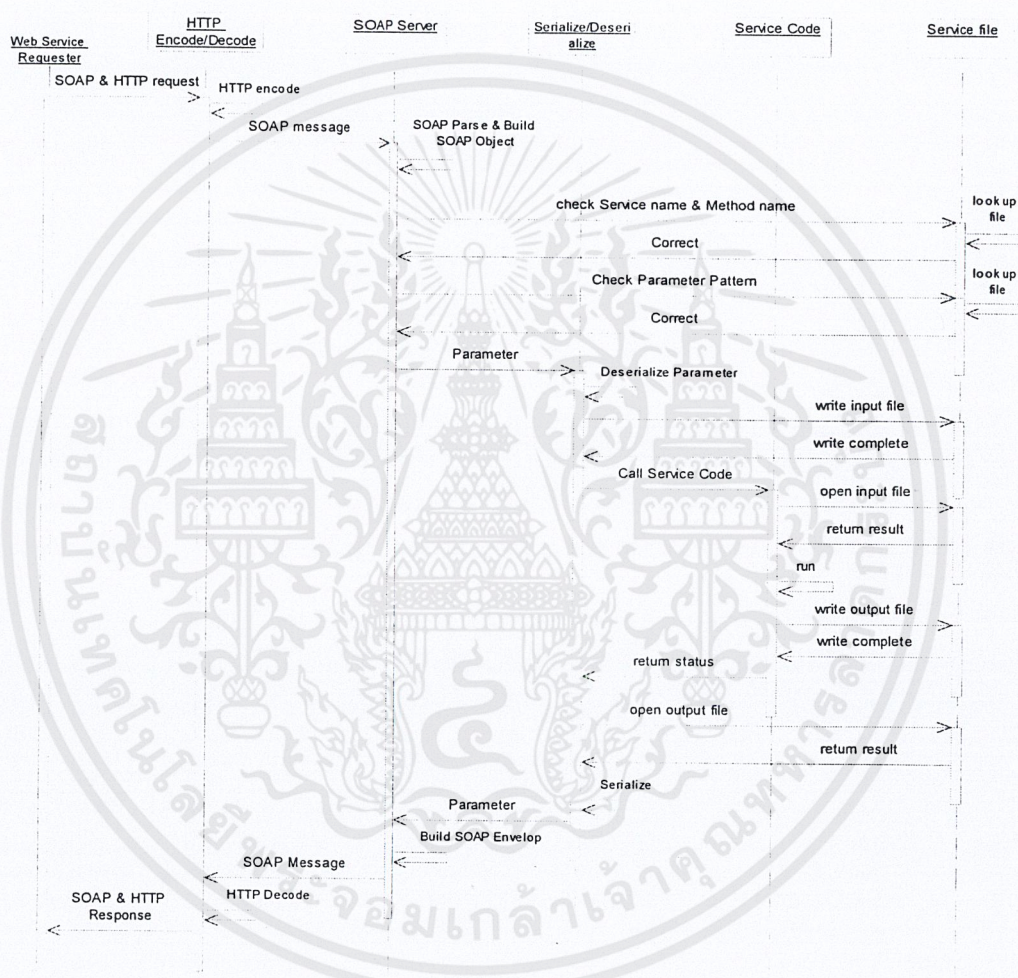


รูปที่ 3-16 แสดงการตรวจสอบล็อกไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อธิบายการทำงานของตรวจสอบล๊อคไฟล์ คือ เริ่มต้น ผู้ให้บริการเว็บเซอร์วิสจะทำการส่งไปฝั่งคอม86 ให้ทำการโซวล๊อคไฟล์ หลังจากนั้น คอม86 จะทำการติดต่อกับล๊อคไฟล์ เพื่อขอให้ล๊อคไฟล์ทำการเปิดไฟล์เพื่อให้อ่านไฟล์ได้ หลังจากนั้นล๊อคไฟล์ก็จะทำการส่งผลลัพธ์กลับไปให้คอม86 ซึ่ง คอม86 ก็จะส่งรายชื่อของล๊อคไฟล์กลับไปให้ผู้ให้บริการอีกที

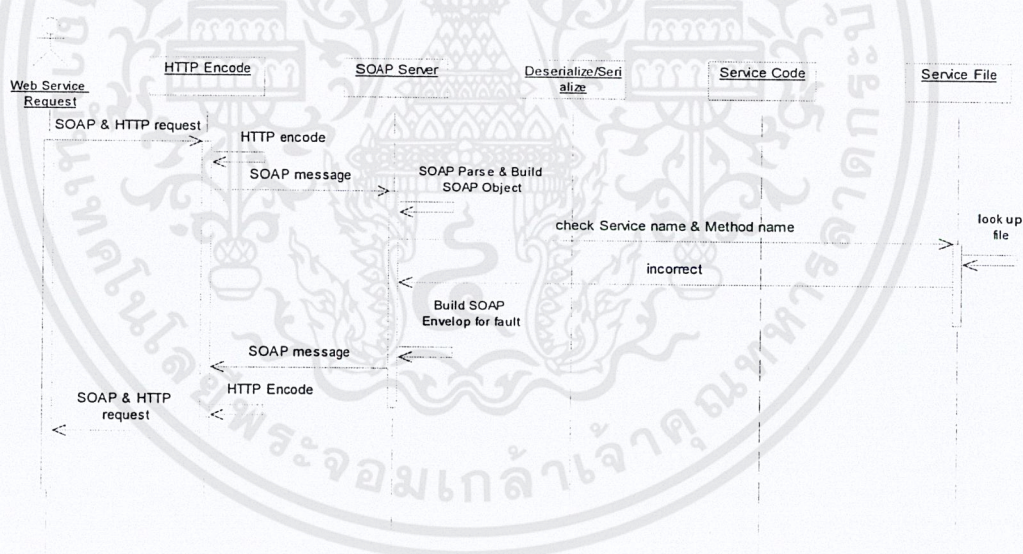
3.7.4.5 การเรียกใช้เว็บเซอร์วิส (Call Web Service)



รูปที่ 3-17 ซีควเอนส์ไดอะแกรมแสดงการเรียกใช้เว็บเซอร์วิส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

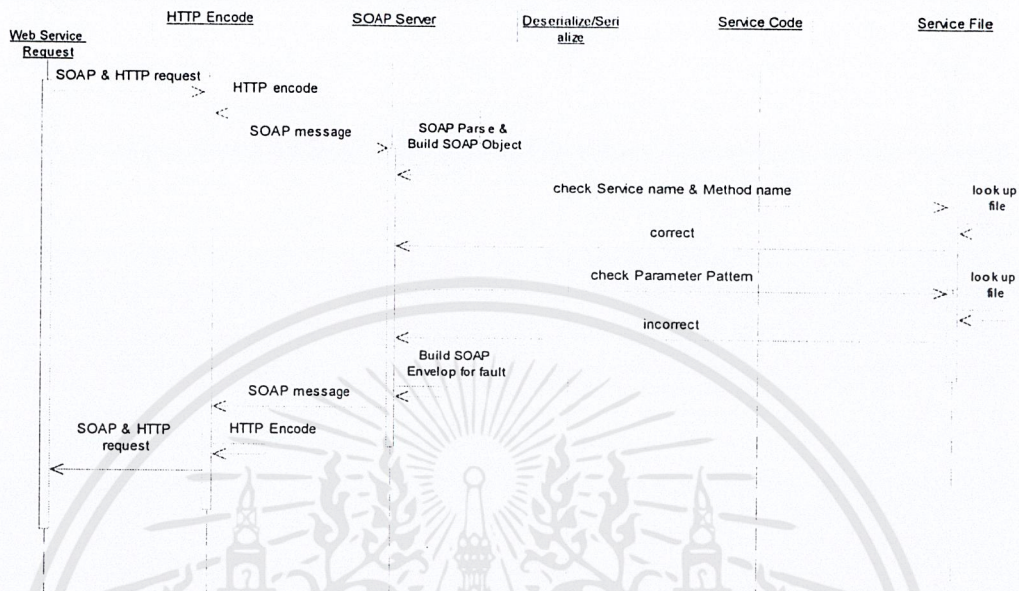
อธิบายการทำงานของกรเรียกใช้เว็บเซอร์วิส ซึ่งก็จะมีส่วนประกอบ คือ ผู้ขอใช้บริการ, เอชทีทีพีเอนโค้ด/ดีโค้ด (HTTP Encode/Decode), โซบเซิร์ฟเวอร์ (SOAP Server), ซีรี่ไรซ์/ดีซีรี่ไรซ์ (Serialize/Deserialize), เซอร์วิสโค้ด (Service Code), เซอร์วิสไฟล์ (Service File) โดยที่เริ่มแรกผู้ขอใช้บริการ จะทำการส่งโซบและเอชทีทีพีรีควีส (SOAP & HTTP request) ไปที่เอชทีทีพีเอนโค้ด/ดีโค้ด หลังจากนั้นจะทำการเอชทีทีพีเอนโค้ด (HTTP encode) แล้วทำการส่งโซบแมสเซจ (SOAP Message) ไปยังโซบเซิร์ฟเวอร์ เพื่อทำการโซบพาสและสร้างโซบออบเจกต์ (SOAP Parse & Build SOAP Object) จากนั้นจึงทำการเช็คชื่อ บริการและชื่อเมธอด (Check Service Name & Method Name) ที่เซอร์วิสไฟล์ ทำการเรียกดูไฟล์ ถ้าถูกต้องจะส่งข้อความไปบอกโซบเซิร์ฟเวอร์ว่าถูกต้อง หลังจากนั้นทำการเช็ครูปแบบของพารามิเตอร์ว่าถูกต้องหรือไม่ที่ เซอร์วิสไฟล์ ทำการเรียกดูไฟล์ ถ้าถูกต้องจะทำการส่งข้อความว่าถูกต้องไปบอกที่โซบเซิร์ฟเวอร์ จากนั้นโซบเซิร์ฟเวอร์จะส่งพารามิเตอร์ไปยังซีรี่ไรซ์ / ดีซีรี่ไรซ์ ทำการดีซีรี่ไรซ์พารามิเตอร์ หลังจากนั้นนำไปเขียนอินพุท (Write Input) ที่เซอร์วิสไฟล์ เมื่อเสร็จแล้วก็จะส่งมาที่ซีรี่ไรซ์ / ดีซีรี่ไรซ์ แล้วทำการเรียกเซอร์วิสโค้ดที่ เซอร์วิสโค้ด แล้วทำการเรียกคำสั่งเปิดไฟล์อินพุทที่เซอร์วิสไฟล์ จากนั้นทำการประมวลผลบริการ แล้วเขียนลงเอาต์พุทไฟล์ที่เซอร์วิสไฟล์ แล้วทำการส่งผลลัพธ์กลับมาที่ซีรี่ไรซ์ หลังจากนั้นส่งพารามิเตอร์กลับมาที่ เซิร์ฟเวอร์เพื่อทำการสร้าง SOAP Envelop แล้วส่งโซบแมสเซจไปยังเอชทีทีพีเอนโค้ด / ดีโค้ด ทำการเอชทีทีพีดีโค้ด แล้วส่งโซบแอนด์เอชทีทีพีเรสพอนกลับไปยังผู้ขอใช้บริการเว็บเซอร์วิส



รูปที่ 3-18 ซีควนส์ไดอะแกรม แสดงข้อผิดพลาดที่เกิดตอนเช็คเซอร์วิสเนมและเมธอดเนม

อธิบายการทำงานของกรเรียกใช้เว็บเซอร์วิส กรณีนี้เป็นตัวอย่างที่เกิดความผิดพลาดขึ้นในขณะที่มีการเช็คชื่อบริการและชื่อเมธอด ซึ่งจะทำให้ไม่เกิดการซีรี่ไรซ์ หรือ ดีซีรี่ไรซ์ขึ้น แต่จะทำการสร้างโซบ

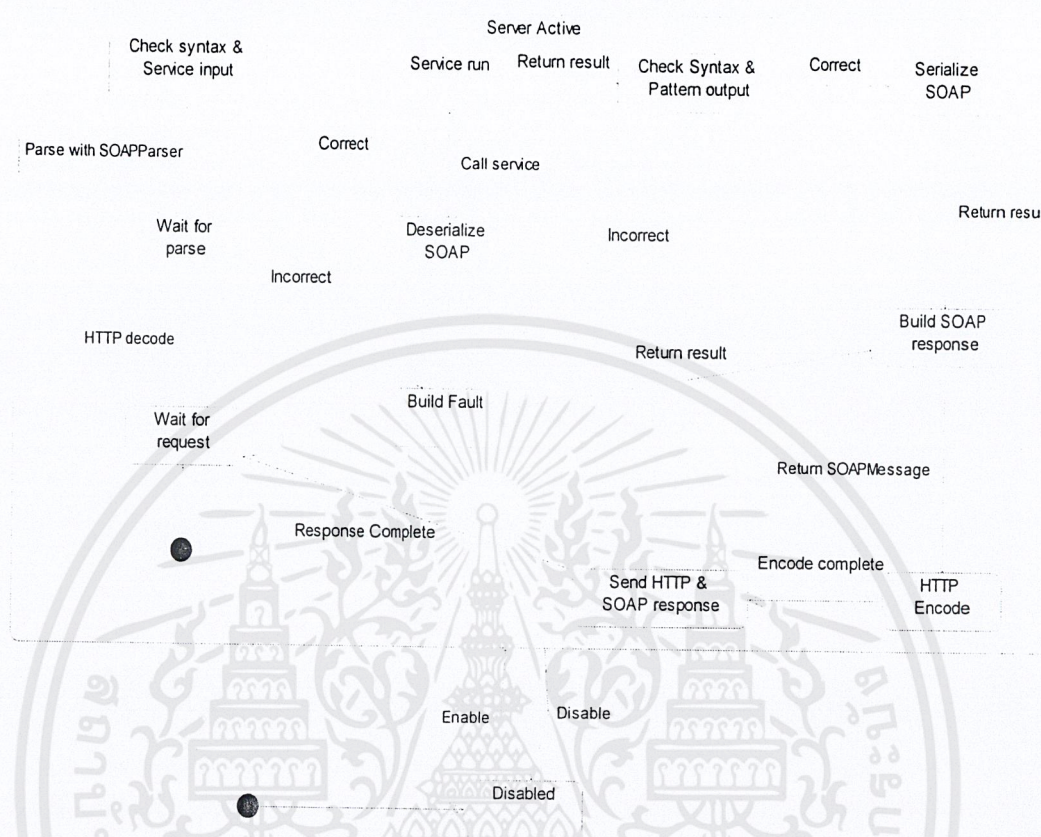
เอลิวลิอพอร์ฟอล (Build SOAP Envelop for fault) แล้วส่งกลับไปเป็น โขบแมสเซจ, โขบแอนด์เอชทีทีพีเรชสปอนกลับไปที่ผู้ขอใช้บริการเว็บเซอร์วิส



รูปที่ 3-19 ซีควนส์ไดอะแกรม แสดงข้อผิดพลาดที่เกิดตอนเซ็กรูปแบบของพารามิเตอร์

อธิบายการทำงานของกรเรียกใช้เว็บเซอร์วิส กรณีนี้เป็นตัวอย่างที่เกิดความผิดพลาดขึ้นในขณะที่มีการรูปแบบของพารามิเตอร์ ซึ่งจะทำให้ไม่เกิดการซีรี่ไรซ์ หรือ ดีซีรี่ไรซ์ขึ้น แต่จะทำการสร้าง โขบเอลิวลิอพอร์ฟอล (Build SOAP Envelop for fault) แล้วส่งกลับไปเป็น โขบแมสเซจ, โขบแอนด์เอชทีทีพีเรชสปอนกลับไปที่ผู้ขอใช้บริการเว็บเซอร์วิส

3.8 สเตตชาร์ตไดอะแกรม (Statechart Diagram)



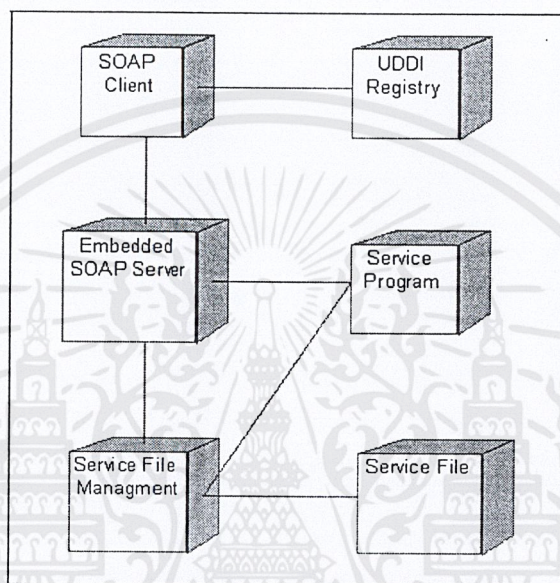
รูปที่ 3-20 แสดงสเตตชาร์ตไดอะแกรม

ลำดับการทำงาน คือ เริ่มแรกจะไปอยู่สเตตดิสเอเบิล (Disabled State) ซึ่งสามารถเอนาเบิ้ล (Enable) เพื่อไปสู่สเตตเซิร์ฟเวอร์แอกทีฟ (Server Active State) และจากสเตตเซิร์ฟเวอร์แอกทีฟสามารถกลับมาที่สเตตดิสเอเบิลได้โดยการดิสเอเบิล (Disable) ในสเตตเซิร์ฟเวอร์แอกทีฟ จะมีซับสเตตหลายซับสเตต โดยเริ่มที่สเตตเวดฟอร์รีควีส (Wait for request State) จากสเตตเวดฟอร์รีควีสพอทำการเอชทีทีพีดีโค้ด (HTTP Decode) จะเข้าสู่สเตตเวดฟอร์พาส (Wait for parse State) จากนั้นเมื่อทำการพาสวูดโซบพาร์เซอร์ (Parse with SOAP Parser) ก็จะเข้าสู่สเตตเช็กซินแท็กแอนด์เซอร์วิสอินพุท (Check syntax & Service input State) ซึ่งถ้าซินแท็กและเซอร์วิสอินพุทนั้นมีความถูกต้องก็จะเข้าสู่สเตตดีซีรี่ไรซ์โซบ (Deserialize SOAP State) ซึ่งก็จะทำการเรียกเซอร์วิส (call server) เพื่อให้เข้าสู่สเตตเซอร์วิสรัน (Service Run State) เมื่อส่งผลลัพธ์กลับมาจะเข้าสู่สเตตเช็กซินแท็กแอนด์แพตเทิร์นเอาท์พุท (Check Syntax & Pattern output State) เมื่อซินแท็กและแพตเทิร์นมีความถูกต้องแล้ว ก็จะเข้าสู่สเตตซีรี่ไรซ์โซบ (Serialize SOAP State) หลังจากส่งผลลัพธ์กลับมา ก็จะเข้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้เสดทของการสร้างโฆบเรชสปอน (Build SOAP response State) หลังจากนั้นก็จะส่งโฆบแมสเซจกลับมาเข้าผู้เสดทเอชทีทีพีเอนโค้ด (HTTP Encode State) ซึ่งจะหมายถึงการเอนโค้ดสำเร็จเข้าสู่เสดทเช่นเอชทีทีพีแอนด์โฆบเรชสปอน (Send HTTP & SOAP response State) หลังจากการส่งเรชสปอนกลับไปแล้วก็จะเข้าสู่เสดทเวดฟอร์คอมพลีทอีกครั้้ง

3.9 ดีพลอยเมนที่ไคอะแกรม (Deployment Diagram)



รูปที่ 3-21 แสดงดีพลอยเมนที่ไคอะแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ลักษณะการใช้งานเว็บเซอร์วิสบนระบบฝังตัว

ลักษณะการใช้งานเว็บเซอร์วิสบนระบบฝังตัวนั้นสามารถแบ่งได้ออกเป็น 4 ลักษณะ คือ

1. นักพัฒนา
2. ผู้ที่สร้างและดูแลการให้บริการเว็บเซอร์วิสบนระบบฝังตัว
3. ผู้ที่สร้างเซอร์วิสบนระบบฝังตัว
4. ผู้ที่เรียกใช้บริการเซอร์วิสจากระบบฝังตัว

4.1 นักพัฒนา

ในลักษณะการใช้งานนี้นักพัฒนาสามารถที่จะนำสโโค้ดไปพัฒนาต่อโดยจะสามารถที่จะแก้ไขเพิ่มเติมในส่วนต่างๆ ได้อย่างอิสระเนื่องจากใช้แนวคิดแบบออบเจกต์โอเรียลเท็ด นอกจากนี้การแยกส่วนของซอฟต์แวร์ออกเป็น 2 ส่วนคือส่วนที่ไม่ขึ้นกับแพลตฟอร์มและส่วนที่ขึ้นกับแพลตฟอร์มออกจากกัน ทำให้นักพัฒนาสามารถนำส่วนที่ไม่ขึ้นกับแพลตฟอร์ม

4.2 ผู้ที่สร้างและดูแลการให้บริการเว็บเซอร์วิส

ในลักษณะการใช้งานนี้ ผู้ที่ต้องการสร้างและดูแลการให้บริการเว็บเซอร์วิสบนระบบฝังตัวนั้น ต้องทำการสร้าง โปรแกรมเซิร์ฟเวอร์ขึ้นมาเพื่อจัดการเกี่ยวกับเอกสารเอกสารที่มีรูปแบบมาตรฐานตามซิมเปลแอคเซสโพรโตคอลที่มีการเรียกใช้เข้ามาและการสร้าง เอกสารที่มีรูปแบบมาตรฐานตามซิมเปลแอคเซสโพรโตคอลเพื่อตอบรับการเรียกใช้บริการกลับไป ซึ่งจะต้องมีการเรียกใช้ในส่วนต่าง ๆ ในส่วนของไลบรารีแพ็คเกจที่ได้จัดทำขึ้นแล้ว โดยสามารถที่จะทำการเลือกเฉพาะลักษณะการทำงานให้เหมาะสมกับสภาพแวดล้อมการใช้งาน เพื่อทำการสร้างเป็นแอปพลิเคชันเฉพาะเนื่องจากข้อจำกัดของแต่ละแพลตฟอร์ม ซึ่งการทำงานหลัก ๆ ของแอปพลิเคชันที่ให้บริการเว็บเซอร์วิสนี้สามารถเขียนเป็นโค้ดแสดงการทำงานอย่างง่ายได้ดังนี้

- โซบเซิร์ฟเวอร์โพรเซส (SOAP Server Process)

```

SOAP Server process
{
    Wait for Service Request;
    HTTP Encode ();
    SOAP Request Handler (SOAP Message);
    Check Error ();
    If Has Error
        Build SOAP Envelope With Fault();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Else
    SOAP Response Handler (SOAP Object);
HTTP Decode (SOAP Envelope);
Send HTTP Response();
}

```

รูปที่ 4-1 แสดงกระบวนการทำงานในส่วนของคลาสโซบเซิร์ฟเวอร์

- โซบรีเคสแฮนเดอ์เลอร์โพรเซส (SOAP Request Handler Process)

```

SOAP Request Handler Process
{
    SOAP Parse (SOAP Message);
    Build SOAP Object ();
    Check Syntax and Service Name Error ();
    If Has Error
        Build SOAP Envelope With Fault ();
    Else {
        Deserialize (Parameter);
        Check Parameter ();
        If Has Error
            Build SOAP Envelope With Fault ();
        Else {
            Write Input File ();
            Call Service Code();
        }
    }
}

```

รูปที่ 4-2 แสดงกระบวนการในการทำงานในส่วนของโซบรีเคสแฮนเดอ์เลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

□ โขบเรสสปอนแฮนเดอเรโพรเซส (SOAP Response Handler Process)

```

SOAP Response Handler Process
{
    Read Output File();
    Check Pattern Output File();
    If Has Error
        Build SOAP Envelope With Fault ();
    Else {
        Serialize (Oput File);
        Build SOAP Envelope With Parameter Serialized ();
    }
}

```

รูปที่ 4-3 แสดงกระบวนการในการทำงานในส่วนของโขบเรสสปอน

นอกจากนี้ยังมีส่วนของการลงทะเบียนเซอร์วิสเข้าในระบบ โดยต้องทำการสร้างแอปพลิเคชันที่ใช้การลงทะเบียนเว็บเซอร์วิสเข้าในระบบ เป็นแอปพลิเคชันที่ทำการใส่ชื่อของบริการเข้าไปในไฟล์ที่เก็บรายชื่อของไฟล์ที่เป็นบริการของระบบทั้งหมด ซึ่งก่อนที่จะทำการใส่เข้าไปจะต้องทำการตรวจสอบชื่อไม่ให้ซ้ำกับชื่อที่มีอยู่แล้ว

```

com86 - HyperTerminal
File Edit View Ctrl Transfer Help
[Icons]
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance" xmlns:xsd="http://www.w3.org/1999/XMLSchema">
<SOAP-ENV:Body>
<SOAP-ENV:Fault>
<faultcode>Server</faultcode>
<faultstring>service name unknow</faultstring>
<faultactor>SOAP Request Handler</faultactor>
</SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

A:\DEMO>register
Calculator was Registered
List of Services
1 Address
2 StockQuote
3 Calculator

A:\DEMO>_

```

รูปที่ 4-4 แสดงการเรียกใช้งานแอปพลิเคชันเพื่อทำการลงทะเบียนเซอร์วิสเข้าในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผู้ที่สร้างเซอร์วิสบนระบบฝังตัว

ในลักษณะการใช้งานนี้ ผู้ที่สร้างเซอร์วิสบนระบบฝังตัวจะต้องทำการพัฒนาแอปพลิเคชัน เซอร์วิสขึ้นมาและสร้างเอกสารอธิบายรายละเอียดของเว็บเซอร์วิส

4.3.1 การพัฒนาแอปพลิเคชันเซอร์วิส

ผู้ที่สร้างเซอร์วิสบนระบบฝังตัวสามารถสร้างแอปพลิเคชันที่จะให้บริการได้ตามต้องการ แต่ในส่วนของการรับข้อมูลเข้าและส่งข้อมูลออกไปนั้น จะทำผ่านฟังก์ชันของคลาสไฟล์มาเนจเจอร์ในการอ่านและเขียนข้อมูล ส่วนการทำงานอื่น ๆ สามารถใช้ไลบรารีทั่วไปหรือไลบรารีที่สร้างขึ้นเองได้ตามปกติ ยกตัวอย่างเช่น

การสร้างแอปพลิเคชันเครื่องคิดเลขที่มีเมธอดในการบวก ลบ คูณ และหาร โดยจะมีพารามิเตอร์นำเข้าสู่สองตัวมีชนิดเป็นจำนวนจริงชนิดดับเบิล (double) และมีผลลัพธ์พารามิเตอร์ตัวเดียวเป็นจำนวนจริงชนิดดับเบิล สามารถเขียนเป็นคลาสได้ดังนี้

```

Class Calculator {
    void main()
    {
        char* input;
        input=FileManager::readInput();
        //process string to data type value
        //find command and call function command (open or close light )
        FileManager::writeOutput();
    }
}

```

รูปที่ 4-5 แสดงตัวอย่างแอปพลิเคชันเซอร์วิสเปิด-ปิดดวงไฟ

4.3.2 การสร้างเอกสารอธิบายรายละเอียดของเซอร์วิสสำหรับเว็บเซิร์ฟเวอร์

เอกสารในส่วนนี้นั้นเป็นเอกสารที่มีรูปแบบเฉพาะที่สร้างขึ้นเพื่อลดภาระในกระบวนการจัดการเอกสารที่อธิบายลักษณะและรูปแบบของเซอร์วิสของเว็บเซิร์ฟเวอร์ลดลงเพื่อให้สามารถทำงานได้ดียิ่งขึ้น โดยทำการสร้างรูปแบบของเอกสารตามไวยากรณ์เอ็กซ์เอ็มแอลและให้มีขนาดเล็กที่สุด แต่ต้องมีรายละเอียดที่จำเป็นต่อเว็บเซิร์ฟเวอร์อย่างครบถ้วนในการที่จะติดต่อและเรียกใช้งานเซอร์วิสนั้น ๆ ได้อย่างถูกต้องแม่นยำ

```

<service namespaceid="urn:xml-soap-demo-calculator">
    <provider>
        <plus>
            <input>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        <arg1 xsi:type="xsd:double"/>
        <arg2 xsi:type="xsd:double"/>
    </input>
    <output>
        <return xsi:type="xsd:double"/>
    </output>
</plus>
<minus>
    <input>
        <arg1 xsi:type="xsd:double"/>
        <arg2 xsi:type="xsd:double"/>
    </input>
    <output>
        <return xsi:type="xsd:double"/>
    </output>
</minus>
<times>
    <input>
        <arg1 xsi:type="xsd:double"/>
        <arg2 xsi:type="xsd:double"/>
    </input>
    <output>
        <return xsi:type="xsd:double"/>
    </output>
</times>
<divide>
    <input>
        <arg1 xsi:type="xsd:double"/>
        <arg2 xsi:type="xsd:double"/>
    </input>
    <output>
        <return xsi:type="xsd:int"/>
    </output>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        </divide>
    </provider>
    < mappings>
        < map encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            xmlns:x="urn:xml-soap-demo-calculator" qname="x:argument"
            headerFile="Argument.h">
            < arg1 xsi:type="xsd:double"/>
            < arg2 xsi:type="xsd:double"/>
        </map>
        < map encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            xmlns:x="urn:xml-soap-demo-calculator" qname="x:result"
            headerFile="Result.h">
            < return xsi:type="xsd:int"/>
        </map>
    </mapping>
</service>

```

รูปที่ 4- 6 แสดงตัวอย่างเอกสารอธิบายรายละเอียดของเซอร์วิสสำหรับเว็บเซิร์ฟเวอร์

4.3.3 การสร้างเอกสารอธิบายรายละเอียดของเว็บเซอร์วิส (WSDL Document)

แอปพลิเคชันในการสร้างเอกสารที่ใช้อธิบายรายละเอียดของเว็บเซอร์วิส นั้นจะทำการรับชื่อไฟล์เข้ามา แล้วทำการอ่านไฟล์ขึ้นแล้วทำการสร้างเป็นเอกสารอธิบายรายละเอียดเว็บเซอร์วิส ซึ่งมีรูปแบบตามที่กำหนด โดยจะมีการเรียกใช้คลาสเอ็กซ์เอ็มแอลไรท์เทอร์ (XMLWriter) ในการสร้างเอกสารเนื่องจากเอกสารอธิบายรายละเอียดเว็บเซอร์วิส มีการใช้ไวยากรณ์ของภาษาเอ็กซ์เอ็มแอล นอกจากนี้ยังสามารถใช้ซอฟต์แวร์อื่น ๆ เข้ามาช่วยในการสร้างได้

```

<definitions name="Calculator" targetNamespace="http://example.com/calculator.wsdl"
xmlns:tns="http://example.com/calculator.wsdl"
xmlns:xsd1="http://example.com/calculator.xsd"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns="http://schemas.xmlsoap.org/wsdl/">
    <types>
        <schematargetNamespace="http://example.com/calculator.xsd"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

xmlns="http://www.w3.org/2000/10/XMLSchema">
  <element name="argument">
    <complexType>
      <all><elementname="arg1" type="double"/></all>
      <all><elementname="arg2" type="string"/></all>
    </complexType>
  </element>
  <element name="request">
    <complexType>
      <all><elementname="return" type="int"/></all>
    </complexType>
  </element>
</schema>
</types>
<message name="getAddressFromNameInput">
  <part name="body" element="xsd:argument"/>
</message>
<message name="getAddressFromNameOutput">
  <part name="body" element="xsd:request"/>
</message>
<message name="addEntryInput">
  <part name="body" element="xsd:argument"/>
</message>
<message name="addEntryOutput">
  <part name="body" element="xsd:request"/>
</message>
<message name="getAllListingsInput">
  <part name="body" element="xsd:argument"/>
</message>
<message name="getAllListingsOutput">
  <part name="body" element="xsd:request"/>
</message>
<message name="putListingsInput">

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    <part name="body" element="xsd:argument"/>
  </message>
  <message name="putListingsOutput">
    <part name="body" element="xsd:request"/>
  </message>
  <portType name="CalculatorPortType">
    <operation name="plus">
      <input message="tns:plusInput"/>
      <output message="tns:plusOutput"/>
    </operation>
    <operation name="minus">
      <input message="tns:minusInput"/>
      <output message="tns:minusOutput"/>
    </operation>
    <operation name="times">
      <input message="tns:timesInput"/>
      <output message="tns:timesOutput"/>
    </operation>
    <operation name="divide">
      <input message="tns:divideInput"/>
      <output message="tns:divideOutput"/>
    </operation>
  </portType>
  <binding name="CalculatorSoapBinding" type="tns:CalculatorPortType">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="plus">
      <soap:operation soapAction="http://example.com/plus"/>
      <input> <soap:body use="literal"/> </input>
      <output> <soap:body use="literal"/> </output>
    </operation>
    <operation name="minus">
      <soap:operation soapAction="http://example.com/minus"/>
      <input> <soap:body use="literal"/> </input>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        <output> <soap:body use="literal"/> </output>
    </operation>
    <operation name="times">
        <soap:operation soapAction="http://example.com/times"/>
            <input> <soap:body use="literal"/> </input>
            <output> <soap:body use="literal"/> </output>
        </operation>
    <operation name="divide">
        <soap:operation soapAction="http://example.com/divide"/>
            <input> <soap:body use="literal"/> </input>
            <output> <soap:body use="literal"/> </output>
        </operation>
</binding>
<service name="CalculatorService">
    <documentation>My first service</documentation>
    <port name="CalculatorPort" binding="CalculatorSoapBinding">
        <soap:address location="http://example.com/calculator"/>
    </port>
</service>
</definitions>

```

รูปที่ 4-7 แสดงตัวอย่างเอกสารอธิบายรายละเอียดของเว็บเซอร์วิส

4.4 ผู้ที่เรียกใช้บริการเซอร์วิสจากระบบฝังตัว

ในลักษณะการใช้งานนี้ ผู้ที่เรียกใช้บริการเซอร์วิสจากระบบฝังตัวจะต้องทำการร้องขอรายละเอียดของเซอร์วิสก่อนเพื่อให้ทราบถึงข้อมูลเกี่ยวกับอินเตอร์เฟซในการติดต่อกับเซอร์วิสนั้น จากนั้นสามารถที่จะทำการสร้างแอปพลิเคชันจากไลบรารีต่างๆ ขึ้นมาได้ตามสภาพแวดล้อมการใช้งานของผู้ใช้ได้ เช่น อาปาเช่โซบ คอทเน็ตเฟรมเวิร์ก เป็นต้น โดยผู้ใช้งานสามารถนำข้อมูลที่ได้รับกลับมาสร้างเป็นรูปแบบต่างๆ ได้ตามต้องการ อีกทั้งยังสามารถสร้างอินเตอร์เฟซได้เอง

บทที่ 5

การทดสอบและวิเคราะห์ผล

5.1 วัตถุประสงค์ของการทดสอบ

1. ทดสอบการให้บริการทางฝั่งเซิร์ฟเวอร์ เพื่อตรวจสอบการทำงานของเซอร์วิสในฝั่งเซิร์ฟเวอร์ที่อาจเกิดความผิดพลาดจากส่วนต่าง ๆ เนื่องจากการให้บริการเว็บเซอร์วิสนั้นหากมีความผิดพลาดเกิดขึ้นจะต้องทำการแจ้งสาเหตุของความผิดพลาดที่เกิดขึ้นนั้นกลับไปที่ผู้เรียกใช้ ถ้าหากความผิดพลาดที่แจ้งกลับไปที่ผู้เรียกใช้ไม่ตรงกับสาเหตุที่แท้จริงอาจทำให้การให้บริการล้มเหลวได้ โดยสาเหตุที่อาจทำให้เกิดความผิดพลาดมีดังนี้
 - ไม่มีรายชื่อบริการตรงกับบริการที่เรียกใช้
 - ไม่มีรายชื่อเมธอดตรงกับเมธอดที่เรียกใช้
 - ไวยากรณ์ของเอกสาร โซบ หรือ โซบแมสเชจ ไม่ตรงตามหลักไวยากรณ์
 - ลำดับรายการของพารามิเตอร์นำเข้า (Input Parameter) ไม่ตรงกับลำดับรายการของพารามิเตอร์ในเอกสารอธิบายรายละเอียดของบริการ
 - ลำดับรายการของผลลัพธ์พารามิเตอร์ (Output Parameter) ไม่ตรงกับลำดับรายการของผลลัพธ์พารามิเตอร์ในเอกสารอธิบายรายละเอียดของบริการ
 - ไม่มีชนิดของข้อมูลในพารามิเตอร์ที่ตรงกับชนิดของข้อมูลที่กำหนดในเอกสารอธิบายรายละเอียดของบริการ
 - บริการ ไม่สามารถทำงานได้สำเร็จ

5.2 วิธีการทดสอบ

ขั้นตอนในการทดสอบสามารถแสดงได้ดังนี้

1. ทดลองสร้างแอปพลิเคชันที่ให้บริการทางฝั่งเซิร์ฟเวอร์
2. ทดลองป้อนพารามิเตอร์นำเข้าทุกค่าที่เป็นไปได้ให้กับแอปพลิเคชันที่สร้างขึ้นแล้วเก็บผลลัพธ์ที่เกิดขึ้นทุกกรณี
3. สร้างเอกสารอธิบายรายละเอียดของบริการ
4. ทดลองลงทะเบียนบริการเข้าในระบบเว็บเซอร์วิส
5. ทดลองประกาศบริการไปยังระบบฐานข้อมูลเว็บเซอร์วิส (UDDI Registry)
6. ทดลองค้นหาบริการจากระบบฐานข้อมูลเว็บเซอร์วิส (UDDI Registry)
7. ทดลองสร้างแอปพลิเคชันในฝั่งไคลเอนต์แล้วทดลองส่งค่าพารามิเตอร์นำเข้าทุกค่าที่เป็นไปได้แล้วบันทึกผลลัพธ์ที่เกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. ทดลองสร้างสภาพแวดล้อมที่ทำให้เกิดความผิดพลาดที่เป็นไปได้ทุกกรณีแล้วบันทึกผลลัพธ์ที่เกิดขึ้น แล้วเปรียบเทียบผลลัพธ์ที่เกิดขึ้นกับรายการบันทึกความผิดพลาดที่เกิดขึ้นกับแต่ละบริการ

5.3 ผลการทดสอบและการวิเคราะห์ผลการทดสอบ

ในการทดสอบจะทำการทดสอบเป็นกรณีของการทำงานในแต่ละส่วนที่สามารถเกิดความผิดพลาดขึ้น โดยจะแสดงสาเหตุของความผิดพลาดที่เป็นไปได้และรูปแบบของเอกสารที่แสดงถึงความผิดพลาด

5.3.1 การทดสอบการให้บริการทางฝั่งเซิร์ฟเวอร์

- ผลการทดสอบการให้บริการทางฝั่งเซิร์ฟเวอร์ ซึ่งผลลัพธ์ที่เกิดขึ้นจากการเรียกใช้มี 2 รูปแบบคือ
 - 1) โขบแมสเซจที่บอกถึงการทำงานในภาวะปกติจะมีรายละเอียดของผลลัพธ์อยู่ภายในส่วนเนื้อหาของโขบแมสเซจ

```

com96 - HyperTerminal
File Edit View Call Transfer Help
[Icons]
</SOAP-ENV:Envelope>
Response...
HTTP/1.0 200 OK
Content-Type: text/xml; charset="utf-8"
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance" xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:minus xmlns:ns1=urn:xml-soap-demo-calculator SOAP-ENV:encodingStyle=http://schemas.xmlsoap.org/soap/encoding/>
      <arg1 xsi:type=xsd:double>5.0</arg1>
      <arg2 xsi:type=xsd:double>4.0</arg2>
    </ns1:minus>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
A:\DEMO>
Connected 1:31:05  ANSI  57600 8N-1  SCROLL  LAPS  NUM  Caps  Print echo
  
```

รูปที่ 5- 1 แสดงตัวอย่างของโขบแมสเซจที่บอกถึงการทำงานในภาวะปกติ

- 2) โขบแมสเซจที่บอกถึงความผิดพลาดจะมีส่วนความผิดพลาดของผลลัพธ์อยู่ภายในส่วนเนื้อหาของโขบแมสเซจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

com86 - HyperTerminal
File Edit View Call Transfer Help
[Icons]
</SOAP-ENV:Envelope>
Response...
HTTP/1.0 200 OK
Content-Type: text/xml; charset="utf-8"
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance" xmlns:xsd="http://www.w3.org/1999/XMLSchema">
<SOAP-ENV:Body>
<SOAP-ENV:Fault>
<faultcode>Server</faultcode>
<faultstring>service name unknow</faultstring>
<faultactor>SOAP Request Handler</faultactor>
</SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
A:\DEMO>
Connected 2.14.42 ANSI 57600 8-N-1 SERIAL COPS NUM Captures Print who

```

รูปที่ 5- 2 แสดงตัวอย่างของโชมแมสเซจที่บอกถึงการทำงานเมื่อเกิดข้อผิดพลาดขึ้น

กรณีที่ โชมแมสเซจบอกถึงความผิดพลาดได้อย่างถูกต้องตามวัตถุประสงค์

- กรณีที่ไม่มีรายชื่อบริการตรงกับบริการที่เรียกใช้จะมีข้อมูลในส่วนของข้อผิดพลาดคือ


```

<faultcode>Server</faultcode>
<faultstring>Service name unknow</faultstring>
<faultactor>SOAP Request Handler</faultactor>

```
- กรณีที่ไม่มีรายชื่อเมธอดตรงกับเมธอดที่เรียกใช้จะมีข้อมูลในส่วนของข้อผิดพลาดคือ


```

<faultcode>Server</faultcode>
<faultstring>Method name unknow</faultstring>
<faultactor>SOAP Request Handler</faultactor>

```
- กรณีที่ไวยากรณ์ของเอกสาร โชมไม่ตรงตามหลักไวยากรณ์จะมีข้อมูลในส่วนของข้อผิดพลาดคือ


```

<faultcode>Client</faultcode>
<faultstring>SOAP Envelope syntax error</faultstring>
<faultactor>SOAP Request Handler</faultactor>

```
- กรณีที่ลำดับรายการของพารามิเตอร์นำเข้าไปไม่ตรงกับลำดับรายการของพารามิเตอร์ในเอกสารอธิบายรายละเอียดของบริการจะมีข้อมูลในส่วนของข้อผิดพลาดคือ


```

<faultcode>Client</faultcode>
<faultstring> Input parameter mismatch with Service </faultstring>
<faultactor>SOAP Request Handler</faultactor>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- กรณีที่ลำดับรายการของผลลัพธ์พารามิเตอร์ไม่ตรงกับลำดับรายการของผลลัพธ์พารามิเตอร์ในเอกสารอธิบายรายละเอียดของบริการจะมีข้อมูลในส่วนของข้อผิดพลาดคือ


```
<faultcode>Client</faultcode>
<faultstring>Output parameter mismatch with Service</faultstring>
<faultactor>SOAP Response Handler</faultactor>
```
- กรณีที่ไม่มีชนิดของข้อมูลในพารามิเตอร์ที่ตรงกับชนิดของข้อมูลที่กำหนดในเอกสารอธิบายรายละเอียดของบริการจะมีข้อมูลในส่วนของข้อผิดพลาดคือ


```
<faultcode>Client</faultcode>
<faultstring>Encoding type mismatch</faultstring>
<faultactor>SOAP Request Handler</faultactor>
```
- กรณีที่บริการไม่สามารถทำงานได้สำเร็จจะมีข้อมูลในส่วนของข้อผิดพลาดคือ


```
<faultcode>Server</faultcode>
<faultstring>Service Program can't successful complete</faultstring>
<faultactor>SOAP Response Handler</faultactor>
```

กรณีที่โพรเซสเซอร์บอกถึงความผิดพลาดไม่ถูกต้องตามวัตถุประสงค์

- กรณีที่ทำการมีบริการแอปพลิเคชันทางฝั่งเซิร์ฟเวอร์แล้วแต่ไม่ได้ทำการลงทะเบียนจะมีข้อมูลในส่วนของข้อผิดพลาดคือ


```
<faultcode>Server</faultcode>
<faultstring>Service name unknow</faultstring>
<faultactor>SOAP Request Handler</faultactor>
```
- กรณีที่มีการสร้างเอกสารอธิบายรายละเอียดของบริการแล้วไม่ได้กำหนดถึงฟังก์ชันที่ให้บริการจะมีข้อมูลในส่วนของข้อผิดพลาดคือ


```
<faultcode>Server</faultcode>
<faultstring>Method name unknow</faultstring>
<faultactor>SOAP Request Handler</faultactor>
```
- กรณีที่มีการสร้างเอกสารอธิบายรายละเอียดของบริการแล้วทำการกำหนดลำดับรายการของพารามิเตอร์นำเข้าผิดจะมีข้อมูลในส่วนของข้อผิดพลาดคือ


```
<faultcode>Client</faultcode>
<faultstring> Input parameter mismatch with Service </faultstring>
<faultactor>SOAP Request Handler</faultactor>
```
- กรณีที่มีการสร้างเอกสารอธิบายรายละเอียดของบริการแล้วทำการกำหนดลำดับรายการของผลลัพธ์พารามิเตอร์ผิดจะมีข้อมูลในส่วนของข้อผิดพลาดคือ


```
<faultcode>Client</faultcode>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<faultstring>Output parameter mismatch with Service</faultstring>
```

```
<faultactor>SOAP Response Handler</faultactor>
```

- กรณีที่มีการสร้างเอกสารอธิบายรายละเอียดของบริการแล้วทำการกำหนดชื่อชนิดของข้อมูลผิดพลาดจะมีข้อมูลในส่วนของข้อผิดพลาดคือ

```
<faultcode>Client</faultcode>
```

```
<faultstring>Encoding type mismatch</faultstring>
```

```
<faultactor>SOAP Request Handler</faultactor>
```

□ การวิเคราะห์ผลของการทดสอบการให้บริการทางฝั่งเซิร์ฟเวอร์

จากผลการทดสอบจะเห็นได้ว่า ระบบอาจจะสร้างข้อมูลในส่วนของข้อผิดพลาดได้ไม่ถูกต้องตามวัตถุประสงค์ที่แท้จริงได้ โดยสาเหตุเกิดจากความผิดพลาดในการลงทะเบียนบริการเข้าในระบบ และการสร้างเอกสารอธิบายรายละเอียดของบริการที่บอกข้อมูลเกี่ยวกับบริการได้ไม่ตรงกับการทำงานของบริการนั้น ๆ

□ วิธีการแก้ปัญหา

1. ก่อนทำการลงทะเบียนบริการเข้าสู่ระบบควรทำการตรวจสอบว่าเอกสารอธิบายรายละเอียดของบริการนั้นบอกข้อมูลเกี่ยวกับบริการได้อย่างถูกต้อง
2. เมื่อทำการลงทะเบียนควรตรวจสอบว่ามีบริการนั้น ได้ถูกนำเข้าไปในรายการแล้ว
3. จัดบริการพื้นฐานที่ให้บริการในการบอกถึงบริการที่ให้บริการอยู่บนระบบนั้น
4. จัดให้มีการเก็บข้อมูลเกี่ยวกับการสร้างส่วนของข้อผิดพลาดในแต่ละครั้งลงไป ไฟล์บันทึกรายการที่เกิดขึ้น (Log file)
5. ทดลองทำงานส่วนแอปพลิเคชันในฝั่งเซิร์ฟเวอร์และตรวจสอบผลลัพธ์ก่อนนำไปใช้ในระบบจริง

บทที่ 6

บทวิจารณ์และสรุป

6.1 บทวิจารณ์และสรุป

เนื่องจากปัจจุบันแนวคิดในการเชื่อมต่อระหว่างแอปพลิเคชันเพื่อให้มีการทำงานอย่างอัตโนมัติ และสามารถที่จะเรียกใช้แอปพลิเคชันเพื่อเรียกข้อมูลได้ตลอดเวลาเป็นที่ต้องการมากขึ้นในทุก ๆ ด้าน รวมไปถึงด้านอุตสาหกรรมที่ต้องการให้เกิดการควบคุมเครื่องมือและอุปกรณ์ต่างภายในโรงงานและสถานประกอบการเพื่อลดความเสียหายที่กระทบต่อระบบอุตสาหกรรมน้อยที่สุด และยังช่วยให้ทราบถึงกระบวนการการทำงานของระบบในขณะนั้นมาได้อย่างรวดเร็วมากขึ้น

เทคโนโลยีเว็บเซอร์วิสเป็นอีกหนึ่งแนวคิดในการเชื่อมต่อแอปพลิเคชันเข้าด้วยกันด้วยการกำหนดมาตรฐานร่วมกันเพื่อระบบต่างๆ ภายในเครือข่ายที่มีความแตกต่างทางด้านเทคโนโลยีสามารถติดต่อสื่อสารถึงกันได้ และยังช่วยแก้ปัญหาในการเชื่อมโยงข้อมูลที่มาจากระบบเข้าหากันได้ โดยเว็บเซอร์วิสได้อาศัยข้อดีในส่วนของโพรโทคอลเอชทีทีพี (HTTP Protocol) ที่นิยมใช้กันอย่างแพร่หลาย และการนิยามข้อมูลที่เป็นอิสระของภาษาเอ็กซ์เอ็มแอล

ระบบฝังตัว (Embedded System) คือ ระบบที่มีคอมพิวเตอร์ขนาดเล็กฝังอยู่ร่วมกับอุปกรณ์ต่าง ๆ ทำหน้าที่ควบคุมการทำงานของระบบ โดยที่ไม่มีไมโครคอนโทรลเลอร์ หรือ ไมโครโปรเซสเซอร์เป็นหัวใจสำหรับควบคุมการทำงานของระบบ ในระบบอุตสาหกรรมได้มีการนำเอาระบบฝังตัวเข้ามาใช้ในการควบคุมเครื่องมือและอุปกรณ์ต่าง ๆ กันอย่างแพร่หลายมากขึ้น เพราะมีขนาดเล็ก มีความทนทานสูง มีราคาไม่แพงมากนัก และสามารถเชื่อมต่อเข้ากับระบบที่มีอยู่เดิมได้ง่าย

แนวความคิดในการพัฒนาเว็บเซอร์วิสบนระบบฝังตัวเป็นการพัฒนาเพื่อให้อุปกรณ์บนระบบฝังตัวสามารถที่จะทำการเชื่อมต่อเข้ากับอุปกรณ์ที่มีการใช้เทคโนโลยีแตกต่างกันให้สามารถทำการสื่อสารข้อมูลกันได้ โดยการพัฒนาให้อุปกรณ์บนระบบฝังตัวทำหน้าที่เป็นเซิร์ฟเวอร์ที่คอยให้บริการกับอุปกรณ์ต่าง ๆ โดยอาศัยผลผลิตของตัวอุปกรณ์ที่มีความสามารถในการเชื่อมต่อที่กับอุปกรณ์

การพัฒนาการให้บริการเว็บเซอร์วิสบนระบบฝังตัวนั้นจะต้องลดกระบวนการจัดการในส่วนของโซบแมสเซจเพื่อให้เวลาในการตอบสนองของบริการ และต้องมีการจัดสรรหน่วยความจำอย่างเหมาะสม เพื่อให้สามารถใช้หน่วยความจำได้อย่างมีประสิทธิภาพ ซึ่งเป็นอุปสรรคที่สำคัญต่อการพัฒนาเว็บเซอร์วิสบนระบบฝังตัว การนำเอาแนวคิดการพัฒนาโปรแกรมแบบซอฟต์แวร์คอมพิวเตอร์ ตามแนวคิดของการออกแบบและพัฒนาโปรแกรมเชิงวัตถุ ทำให้สามารถที่จะแก้ไขโครงสร้างการทำงานของระบบเว็บเซอร์วิสให้เหมาะสมกับสภาพแวดล้อมการทำงานที่มีอยู่อย่างจำกัด

อย่างไรก็ตามเนื่องจากความสามารถอย่างจำกัดของอุปกรณ์บนระบบฝังตัวในการทำหน้าที่เป็นเซิร์ฟเวอร์นั้นจำกัดอย่างมาก ในกรณีที่ต้องการขยายขอบเขตการให้บริการอาจทำได้โดยการสร้างเว็บเซอร์วิสเซิร์ฟเวอร์บนคอมพิวเตอร์ที่มีประสิทธิภาพมากขึ้นเพื่อทำหน้าที่เป็นตัวแทนในการเรียกใช้บริการก่อนที่จะส่งไปยังเว็บเซอร์วิสบนระบบฝังตัว อีกทั้งยังอาจจัดให้มีการตรวจสอบเอกสารที่มาถึง

ผ่านไฟร์วอลล์ (Firewall) หรือวีพีเอ็น (VPN: Virtual Private Network) ก่อนที่จะทำการเรียกใช้บริการเพื่อเพิ่มประสิทธิภาพในการรักษาความปลอดภัยของการให้บริการเว็บเซอร์วิสบนระบบฝังตัว

6.2 แนวทางในการพัฒนาต่อ

เนื่องจากเว็บเซอร์วิสเป็นเทคโนโลยีที่อาศัยโปรโตคอลโชน ที่มีความสามารถในการฝังตัวและเข้ากันได้กับโปรโตคอลอื่นๆ ทำให้เราสามารถที่จะทำการสร้างระบบที่ให้บริการร่วมกันได้ อีกทั้งการเพิ่มความสามารถในการเป็นตัวกลางในการส่งเอกสารโชนแมสเซจ เพื่อสร้างกระบวนการบางอย่างกับเอกสารให้สามารถใช้ประโยชน์ในด้านการรักษาความปลอดภัย และการเป็นตัวแทนในการเรียกใช้บริการ และการสร้างระบบการรักษาความปลอดภัยภายในโชนแมสเซจ ในส่วนของการทำลายเซ็นดิจิทัล (Digital Sign) และการสร้างไลบรารีแพ็คเกจที่สนับสนุนการทำงานด้านการรักษาความปลอดภัยในระดับซ็อกเก็ต (Secure Socket Layer) นอกจากนี้ยังอาจนำซอฟต์แวร์ไลบรารีที่สร้างขึ้นไปพัฒนาเพื่อให้บริการเว็บเซอร์วิสบนระบบฝังตัวในแพลตฟอร์มอื่นๆ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

การปรับแต่งคอม86 (Com86 Configuration)

1. สิ่งที่มาพร้อมกับชุดพัฒนา

ในชุดพัฒนาคอม86 จะประกอบด้วยอุปกรณ์ต่างๆดังนี้

- | | | |
|------------------------------------|---|-------|
| ▪ บอร์ดคอม86 | 1 | บอร์ด |
| ▪ อแดปเตอร์สำหรับบอร์ดคอม86 | 1 | อัน |
| ▪ สาย Serial ชนิด null modem | 1 | เส้น |
| ▪ สาย USB ชนิด A-B | 1 | เส้น |
| ▪ ซีดีรอมโปรแกรมและคู่มือการใช้งาน | 1 | แผ่น |

2. ความต้องการของระบบเบื้องต้น

ในการใช้งานชุดพัฒนาคอม86 นั้นจะมีความต้องการเบื้องต้นทั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์ดังนี้

- เครื่องคอมพิวเตอร์สำหรับใช้คอมไพล์และตรวจสอบการทำงานของโปรแกรม 1 เครื่อง โดยจะต้องมีพอร์ตอนุกรมสำหรับใช้งานได้อย่างน้อย 1 พอร์ต* และมีการ์ดอีเธอร์เน็ต อย่างน้อย 1 การ์ด** หากต้องการพัฒนาโปรแกรมที่ใช้คุณสมบัติทางด้าน Network
- โปรแกรมเทอร์มินัล เช่น Hyperterminal เป็นต้น
- สายแลนชนิดแบบสายตรง (straight-through twisted pair cable) และ 10Base-T hub หรือสายแลนชนิดแบบครอส (twisted pair cross cable)**
- สาย serial***
- คอมไพล์เลอร์ที่สามารถคอมไพล์โปรแกรมสำหรับไมโครโพรเซสเซอร์ตระกูล 186 แบบ Real mode เช่น turbo C, Turbo assembly หรืออื่นๆ ตามที่ผู้พัฒนาต้องการใช้งาน

* สำหรับหรับใช้ เป็น Serial console ในการพัฒนาโปรแกรมต่างๆ

** เพิ่มเติมสำหรับใช้ในการพัฒนาที่ต้องการใช้ความสามารถทางด้าน Network

*** เพิ่มเติมสำหรับผู้ที่ต้องการใช้งานพอร์ตอนุกรมที่เหลืออีกพอร์ตหนึ่งของบอร์ดคอม86

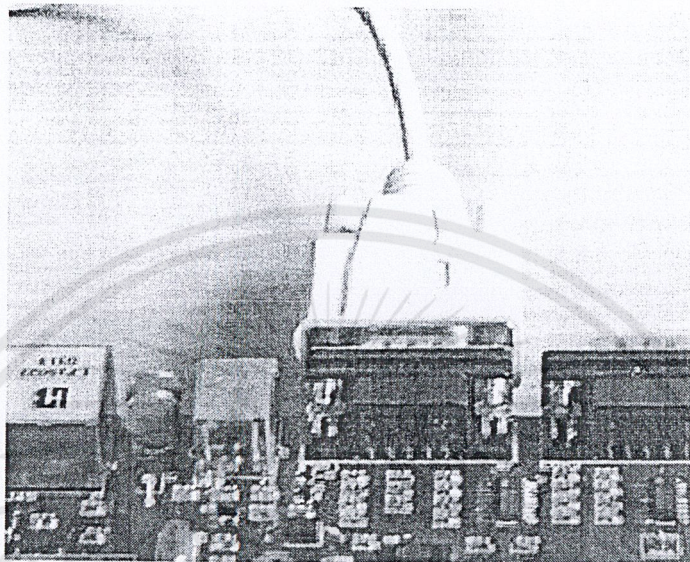
3. การติดตั้งใช้งาน

ในการติดตั้งชุดพัฒนาคอม86 ให้ปฏิบัติตามขั้นตอนต่างๆ ดังนี้

1. นำบอร์ดคอม86 ออกมาจากช่องที่ใช้บรรจุเพื่อการขนส่งและตรวจสอบสภาพของบอร์ดใน

เบื้องต้นว่าเกิดการชำรุดเสียหายจากการขนส่งหรือไม่

2. เชื่อมต่อสายของ Serial cable ที่ให้มากับพอร์ตอนุกรมของคอมพิวเตอร์ที่ต้องการใช้งาน และเชื่อมต่อสายอีกด้านเข้ากับบอร์ดคอม86



รูปที่ ก-1 ภาพการเชื่อมต่อสาย serial console เข้ากับบอร์ดคอม86

3. เชื่อมต่อสาย Ethernet เข้าบอร์ดคอม86 และเครื่องคอมพิวเตอร์หรือเน็ตเวิร์กสับ*

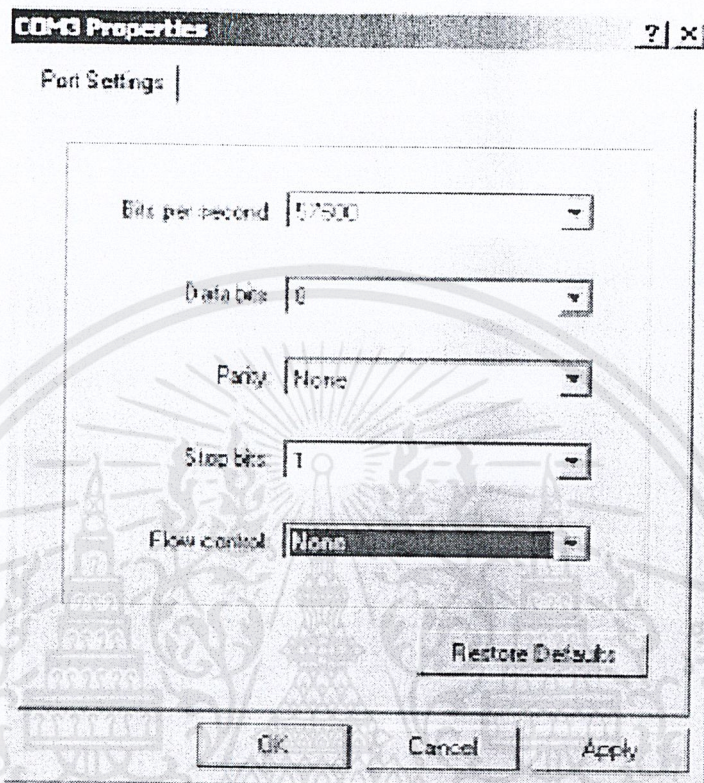


รูปที่ ก-2 ภาพการเชื่อมต่อสายอีเทอร์เน็ตเข้ากับบอร์ดคอม86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

* สำหรับผู้ที่ต้องการใช้ความสามารถทางด้านเน็ตเวิร์ก

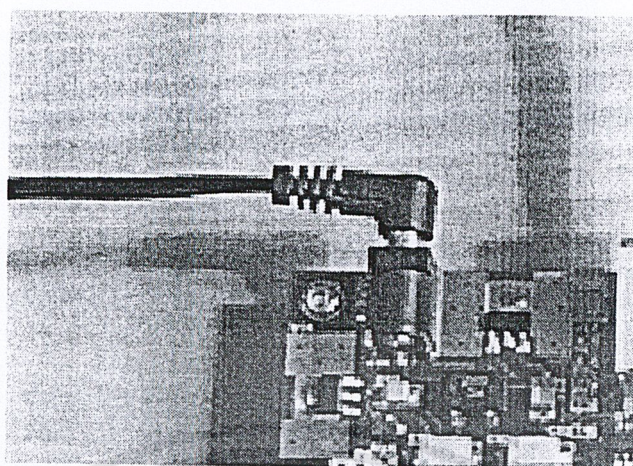
4. เปิดโปรแกรม terminal ที่ใช้งานขึ้นมาและทำการกำหนดค่าต่างๆ ดังนี้



รูปที่ ก-3 ภาพการตั้งค่าต่างๆ ของโปรแกรมเทอร์มินัล

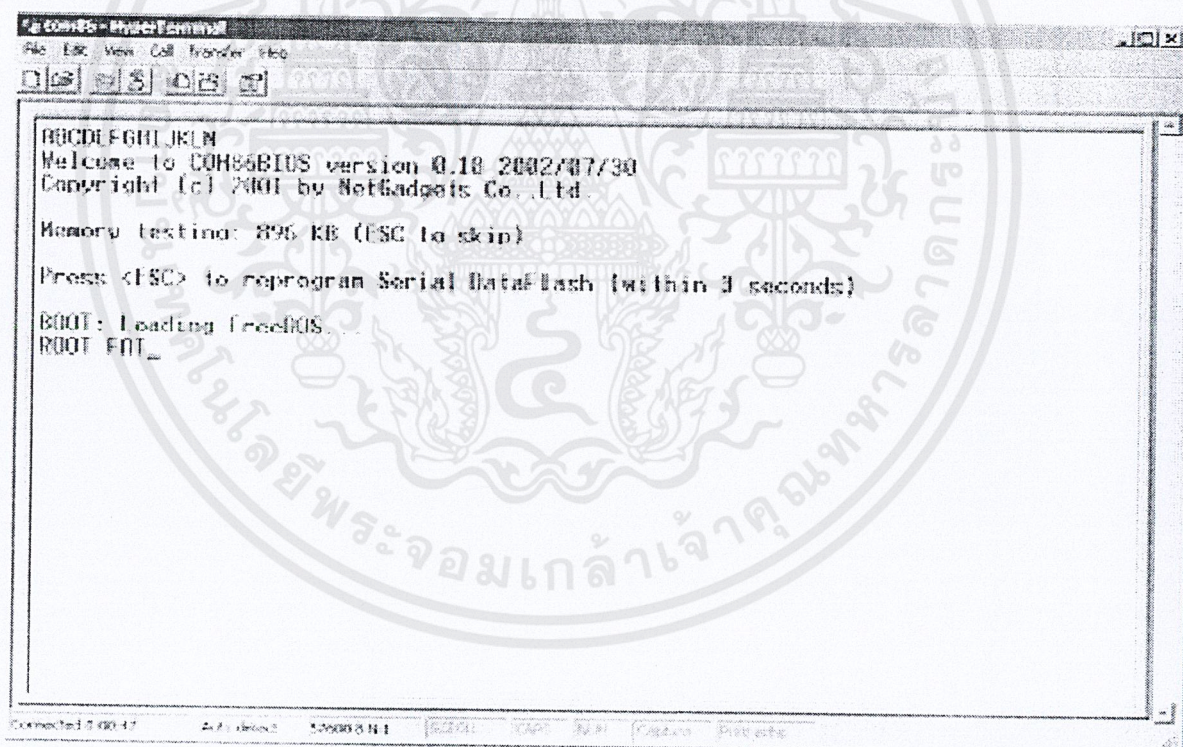
- เลือกพอร์ตการเชื่อมต่อของคอมพิวเตอร์ ตามที่ได้ทำการเชื่อมต่อในขั้นตอนที่ 2
- กำหนดอัตราการรับส่งข้อมูลอยู่ที่ 57600 บิตต่อวินาที
- ข้อมูลเป็นแบบ 8 บิต
- ไม่มีพาริตี
- Stop บิตเท่ากับ 1
- ยกเลิกการทำโฟลว์คอนโทรลทั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์ และตั้งเริ่มการเชื่อมต่อ

5. ต่อปลั๊กของอแดปเตอร์เข้ากับบอร์ดคอม86 และต่ออแดปเตอร์เข้ากับแหล่งจ่ายไฟของอาคาร



รูปที่ ก-4 ภาพแสดงการเชื่อมต่ออแดปเตอร์กับบอร์ดคอม86

6. กดรีเซตสวิทช์จะเห็นโปรแกรมบนบอร์ดเริ่มทำงาน โดยการแสดงผลผ่านโปรแกรมเทอร์มินัลที่ใช้งาน



รูปที่ ก-5 ภาพแสดงตัวอย่างการแสดงผลของบอร์ดคอม86เมื่อบอร์ดเริ่มทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ข้อควรระวังในการใช้งาน

ในการใช้งานชุดพัฒนาคอม86 มีข้อควรระวังในการใช้งานต่างๆ ดังนี้

- ควรถอดปลั๊กแอดAPTERทุกครั้งในการเชื่อมต่อวงจรต่างๆเข้ากับบอร์ดคอม86
- ไม่ควรติดตั้งใช้งานในบริเวณที่เปียกชื้นหรือมีความร้อนสูง
- ไม่ควรยกเลิกการจ่ายกระแสไฟฟ้าในขณะที่กำลังทำการ โปรแกรมเฟลชไบออสหรือโปรแกรม อิมเมจของดิสก์เนื่องจากอาจก่อให้เกิดความเสียหายต่อบอร์ดคอม86 ได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] Richard Johnsonbaugh. : Applications Programming in C++ : Prentice-Hall International, Inc. , 2002
- [2] Deitel & Deitel : C++ How to Program : McGRAW-HILL International , 2000
- [3] เปิดโลกของ TCP/IP และโปรโตคอลของอินเทอร์เน็ต : Provision, 2542
- [4] กิตติชัย พินิจคำ : การเขียนโปรแกรม XML และ SOAP สำหรับ BizTalk Server : สำนักพิมพ์สามย่าน.COM, 2544
- [5] ราบินเดอร์ ศรีกิจจากรณ์ : การเขียนโปรแกรมแบบโอโอพีด้วยเทอร์โบและบอร์แลนด์ C++ : ซีเอ็ดดูเคชั่น, 2542
- [6] ดร. วีระศักดิ์ ชิงดาวาร : Fundamental of JAVA Programming : ซีเอ็ดดูเคชั่น, 2543
- [7] สุรศักดิ์ สงวนพงษ์ : สถาปัตยกรรมและโปรโตคอลที่ซีพี/ไอพี : โครงการตำราวิชา ภาควิชาวิศวกรรมศาสตร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์, 2545
- [8] ผศ. อภินทร อุณาภูล : กระบวนการและวิธีการพัฒนาเว็บแอปพลิเคชันด้วยภาษา UML : ภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2543.
- [9] นฤต กระจาย : การเขียนโปรแกรมในคอสมและวินโดวส์ด้วยบอร์แลนด์C++5.0 : ซีเอ็ดดูเคชั่น, 2540
- [10] อภินทร อุณาภูล : Object-Oriented Analysis and Design : โครงการผลิตตำราของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2543
- [11] ชันวา ศรีประโม่ง : การเขียนโปรแกรมภาษาซี สำหรับวิศวกรรม : โครงการตำราวิชาการ มหาวิทยาลัยมหานคร, 2538

เว็บไซต์อ้างอิง

Knowledge about web service

- <http://www.w3.org>
- <http://www.xmethods.com>
- <http://www-106.ibm.com>
- <http://www.webservicesarchitect.com>
- <http://www.abewbservice.com>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Knowledge about SOAP

- <http://ws.apache.org/soap/>
- <http://www.soap.org>
- <http://www.soapware.org>
- <http://www.soapware.com>
- <http://schemas.xmlsoap.org/soap/envelope>
- <http://schemas.xmlsoap.org/soap/encoding>

Knowledge about XML

- <http://www.rubyxml.com>
- <http://www.xmlspy.com>
- <http://www.xmlrpc.com>
- <http://www.xmlrpc.com>

Knowledge about protocols

- <http://www.protocols.com>
- <http://www.cs.umbc.edu/agents/technology/acl.shtml>

Knowledge about UDDI

- <http://www.uddi.org>
- <http://www-3.ibm.com/services/uddi>
- <http://soapclient.com/uddisearch.html>

Knowledge about WSDL

- <http://www.pocketsoap.com/wsdl>
- <http://www.w3.org/TR/wsdl>
- <http://www.pocketsoap.com/wsdl>

Other

- <http://www.thaiesf.org> (Com 86 information)
- <http://doesrv.caldera.com:8457> (How to C++ programming)
- <http://www.embedded.com> (Embedded systems programming)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้