สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาซอฟต์แวร์เชิงคอมโพเน็นต์โดยใช้เทคโนโลยีดอทเน็ต

Web Services & Component Software Development Using .Net

T 055075

นายจตุพร ตุ่นกลาง

นายจตุพล นิวันติ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เลขหมู่.....
เลขทะเบียน 55075
วัน,เดือน,ปี - 8 มิ.ย. 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาซอฟต์แวร์เชิงคอมโพเน็นต์โดยใช้เทคโนโลยีดอทเน็ต

Web Services & Component Software Development Using .Net

โดย

นายจตุพร ตุ่นกลาง

นายจตุพล นิวันติ

อาจารย์ที่ปรึกษา

ดร. วรวัฒน์ ลิ้มโภคา

ดร. ชุติเมษฏ์ ศรีนิลทา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2546

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาซอฟต์แวร์เชิงคอมโพเน็นต์โดยใช้เทคโนโลยีดอทเน็ต

Web Services & Component Software Development Using .Net

ผู้จัดทำ นายจตุพร ตุ่นกลาง รหัส 43010047

นายจตุพล นิวันติ รหัส 43010050

_____ อาจารย์ที่ปรึกษา

(ดร.วรวัฒน์ ลิ้มโภคา)

_____ อาจารย์ที่ปรึกษา

(ดร.ชุติเมษฎ์ ศรีนิลทา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาซอฟต์แวร์เชิงคอมโพเน็นต์โดยใช้เทคโนโลยีดอทเน็ต

นาย จตุพร ตุ่นกลาง 43010047
นาย จตุพล นิวันติ 43010050
ดร. วรวัฒน์ ลิ้มโภคา อาจารย์ที่ปรึกษา
ดร. ชุติเมษฎ์ ศรีนิลทา อาจารย์ที่ปรึกษา
ปีการศึกษา 2546

บทคัดย่อ

ปัจจุบันทิศทางของเทคโนโลยีถูกกำหนดค่อนข้างแน่ชัด ให้มุ่งไปสู่ ดอทเน็ต (.Net) และ เว็บเซอร์วิส (Web Services) ซึ่งถือเป็นอีกหนึ่งเทคโนโลยีบนโลกอินเตอร์เน็ตที่กำลังได้รับความสนใจเป็นอย่างยิ่ง เพราะไม่ว่าจะเป็นบริษัทเอกชนหรือหน่วยงานของรัฐบาลทั่วโลก ต่างก็ให้การสนับสนุนเทคโนโลยีนี้เป็นอย่างมาก จึงทำให้เว็บเซอร์วิส ก้าวเข้ามามีบทบาทอย่างมากในการพัฒนาโปรแกรมสมัยใหม่ กล่าวคือ เทคโนโลยีนี้จะช่วยขยายขีดความสามารถของอินเทอร์เน็ต

เว็บเซอร์วิส เป็นพื้นฐานสำคัญของการบริการใหม่ๆ ในลักษณะที่ต้องการเชื่อมโยงระบบสารสนเทศที่มีความแตกต่างเข้าด้วยกันอย่างอัตโนมัติ เช่น การพาณิชย์อิเล็กทรอนิกส์ระหว่างธุรกิจกับธุรกิจ

โดยปริญญานิพนธ์นี้จะอธิบายถึงรายละเอียดของเว็บเซอร์วิส พร้อมทั้งได้จัดทำระบบขึ้นมาระบบหนึ่ง ซึ่งเป็นเว็บเซอร์วิสที่ให้บริการ (Services) เกี่ยวกับการซื้อขายบ้าน และแบบบ้าน และยิ่งไปกว่านั้น ก็ยังมีเนื้อหาที่เกี่ยวข้องกับภาษาคอมพิวเตอร์ที่ใช้สร้างรูปเสมือนจริงเป็นรูปภาพกราฟิก 3 มิติ หรือที่เรียกว่า VRML (Virtual Reality Modeling Language) ซึ่งจะช่วยให้ลูกค้าที่เข้าไปใช้บริการดูบ้านได้เห็นราวกับว่าได้ไปเดินดูด้วยตัวเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WEB SERVICES & COMPONENT SOFTWARE DEVELOPMENT USING .NET

Mr. Jatuporn Tunklang
Mr. Jatupon Niwanti
Dr. Voravat Limpoka Advisor
Dr. Chutimet Srinilta Advisor
Academic Year 2003

ABSTRACT

Today, the changing way of technology has been determined clearly to .NET and web services. That's so another interesting very much now. Not only general business but also the government in each country support this technology that affect the web services come more important in developing new programs. And this technology will increase ability of Internet.

Web Services is a important basic of new services in characteristics of connection between different IT system automatically, electronics business between each others.

This thesis explains details of web services and builds a system that is web services provide services about the house. And the more is relate to computer language , creates virtual house in 3D graphics calls VRML , helps the customers see more real like see that's exist.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดี ด้วยคำแนะนำ คำปรึกษา และคอยดูแลจากหลายๆ ฝ่ายด้วยกัน โดยเฉพาะอาจารย์ที่ปรึกษาทั้งสองท่านที่ให้โอกาสให้ข้าพเจ้าได้ทำปริญญานิพนธ์ฉบับนี้ คอยให้ความเอาใจใส่ แนะนำ และความช่วยเหลือเสมอมา คือ ดร.วรวัฒน์ ลิ้มโภคา และ ดร.ชุติเมษฎ์ ศรีนิลทา ซึ่งต้องขอขอบพระคุณเป็นอย่างยิ่ง

ขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ที่ได้จัดเตรียมสิ่งอำนวยความสะดวก เพื่อให้การวิจัยและการพัฒนาโปรแกรมเป็นไปได้ด้วยความสะดวก และรวดเร็ว รวมทั้งยังมีอินเทอร์เน็ตความเร็วสูงให้บริการ สำหรับการค้นคว้าหาความรู้ต่างๆ ซึ่งท้ายที่สุดแล้วก็ประกอบกันเป็นส่วนหนึ่งของโครงงานนี้

ขอขอบคุณพี่ๆ เพื่อนๆ ในห้องปฏิบัติการคอมพิวเตอร์ (Olala) ที่คอยสร้างความครึกครื้นยามอยู่ในห้อง เป็นกำลังใจเสมอมา

ขอขอบคุณ คุณอภิลักษณ์ ปนาทกูล และอีกหลายๆ ท่าน (ต้องขออภัยด้วยที่ไม่ได้กล่าวชื่อ) ที่ให้ความกรุณาตอบคำถามต่างเป็นอย่างดีตลอดเรื่อยมา

และสุดท้ายต้องขอขอบคุณบุคคลที่สำคัญที่สุดในชีวิตที่ทำให้ข้าพเจ้ามีวันนี้ นั่นคือ บิดา มารดา และ บุคคลในครอบครัวอันเป็นที่เคารพรัก ซึ่งได้เลี้ยงดู คอยสั่งสอนข้าพเจ้ามาเป็นอย่างดี พร้อมให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ ความรักเสมอมา ข้าพเจ้าขอกราบขอบพระคุณมา ณ ที่นี้ด้วย

นายจตุพร ตุ่นกลาง

นายจตุพล นิวันติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VII
สารบัญตาราง	IX
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของงานวิจัย	2
1.3 ขอบเขตของงานวิจัย	2
1.4 วิธีการดำเนินการ	3
1.5 ประโยชน์ที่ได้รับ	3
บทที่ 2 คอมโพเน็นต์ (Component)	4
2.1 คอมโพเน็นต์	4
2.2 งานที่เหมาะสมกับการพัฒนาเชิงคอมโพเน็นต์	8
2.3 ประโยชน์ของการพัฒนาเชิงคอมโพเน็นต์	8
บทที่ 3 เว็บเซอร์วิส (Web Services)	9
3.1 XML	10
3.2 SOAP	35
3.3 UDDI	53
3.4 WSDL	53
บทที่ 4 ASP .NET	56
4.1 ASP คืออะไร	56
4.2 การเปลี่ยนแปลงจาก ASP ไปสู่ ASP.NET	56
4.3 เซิร์ฟเวอร์คอนโทรล (Server Control)	58
4.4 วาริเดชั่นคอนโทรล (Validation Controls)	59
4.5 ภาษาที่ใช้เขียนโค้ด ASP.NET	59
4.6 รู้จักกับ System.Web.UI Namespace	60
4.7 การพัฒนาเว็บแอพพลิเคชั่นด้วย ASP.NET	62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 ASP .NET เว็บเซอร์วิส 67
5.1 องค์ประกอบในการใช้งาน .NET เว็บเซอร์วิส 67
5.2 การสร้าง ASP.NET เว็บเซอร์วิส 68
5.3 การเรียกใช้งานเว็บเซอร์วิส 70
5.4 การจัดการเสตทในเว็บเซอร์วิส 71
5.5 ทรานเซกชัน 74
5.6 การเรียกใช้เว็บเซอร์วิสแบบอซิงโครนัส 85
บทที่ 6 ADO .NET 90
6.1 สถาปัตยกรรมของ ADO.NET 90
บทที่ 7 Microsoft .NET Technology 94
7.1 รู้จักกับ Mircrosoft .NET 94
7.2 .NET Platform 94
7.3 .NET Framework 95
7.4 .NET Compact Framework 99
บทที่ 8 VRML (Virtual Reality Modeling Language) 100
8.1 ที่มา และวัตถุประสงค์ของภาษา VRML 100
8.2 ข้อแตกต่างของภาษา VRML เวอร์ชัน 1.0 และ 2.0 101
8.3 ภาษา VRML และภาษา HTML 102
8.4 องค์ประกอบของภาษา VRML 103
8.5 การสร้างไฟล์ VRML 105
8.6 การจัดเตรียมองค์ประกอบเพื่อสนับสนุนการทำงานของภาษา VRML 105
8.7 สรุปลักษณะเด่นของภาษา VRML 106
8.8 รูปแบบไวยากรณ์โหนดต่างๆ (Syntax) 106
บทที่ 9 การสร้าง และการออกแบบระบบซื้อขายแบบบ้าน และสร้างบ้าน (HomeForYou) 122
9.1 Use case diagram ของระบบ Home For You 122
9.2 Sequence diagram Home For You 123
9.3 การออกแบบระบบฐานข้อมูล 127
9.4 การออกแบบคลาสไดอะแกรม 128
บทที 10 ลักษณะของแอพพลิเคชัน 129
10.1 เว็บไซต์ของระบบออกแบบ และสร้างบ้าน 129
10.2 ส่วนที่เปิดเป็นเว็บเซอร์วิสให้บริการกับเว็บหรือบริษัทอื่น 136
10.3 โมเดลบ้าน VRML 3 มิติ 137

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 11 บทวิจารณ์ และสรุป 141
11.1 สรุปผลการดำเนินงาน 141
11.2 แนวทางการพัฒนาต่อ 141
ภาคผนวก ก. 142
ภาคผนวก ข. 145
บรรณานุกรม 147

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

	หน้า
รูปที่ 2-1 คอมโพเน็นต์ และส่วนประกอบต่างๆ ของมัน	5
รูปที่ 2-2 ตัวอย่างของอินเทอร์เฟส	6
รูปที่ 2-3 คอมโพเน็นต์ และการแทนที่	7
รูปที่ 2-4 การเพิ่มเติมคอมโพเน็นต์	7
รูปที่ 3-1 โครงสร้างของ DOM	32
รูปที่ 3-2 ความสัมพันธ์ของการ parse ด้วย DOM	33
รูปที่ 3-3 โครงสร้างของ SOAP	36
รูปที่ 3-4 การ Client-Side Data Flow	48
รูปที่ 3-5 โครงสร้างภายในของ SOAP	49
รูปที่ 3-6 Server-Side Data Flow	50
รูปที่ 3-7 โครงสร้างภายใน SOAP Server	51
รูปที่ 3-8 การใช้ ASP.NET ทำ SOAP เว็บเซอร์วิส	52
รูปที่ 3-9 Runtime Callable Wrapper	52
รูปที่ 3-10 การใช้ .NET ทำ SOAP client	52
รูปที่ 3-11 COM callable wrapper ที่ทำให้ COM client สามารถเห็น .NET object	53
รูปที่ 4-1 ส่วนประกอบของเว็บฟอร์ม	62
รูปที่ 5-1 การติดต่อจากไคลเอนต์ไปยังเว็บเซิร์ฟเวอร์ผ่านพร็อกซี่อ็อบเจ็กต์	70
รูปที่ 5-2 แอพพลิเคชันที่ทำงานในระบบแบบกระจาย	76
รูปที่ 5-3 การใช้งานฐานข้อมูลพร้อมๆ กัน	76
รูปที่ 5-4 Flat Transaction	79
รูปที่ 5-5 Nested Transaction	79
รูปที่ 5-6 การติดต่อแบบซิงโครนัส	85
รูปที่ 5-7 การติดต่อแบบอซิงโครนัส	86
รูปที่ 6-1 สถาปัตยกรรม ADO.NET	91
รูปที่ 6-2 การเปรียบเทียบระหว่าง SQL Server .NET Data Provider และ OLEDB.NET Data Provider	92
รูปที่ 7-1 องค์ประกอบ .NET Platform	94
รูปที่ 7-2 องค์ประกอบ .NET Framework	95
รูปที่ 7-3 รูปแบบการคอมไพล์โค้ดไปเป็น IL Code	96
รูปที่ 8-1 ระบบแกน 3 มิติ	103
รูปที่ 8-2 การใช้กฎมือขวา	103

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 8-3 ตัวอย่างโครงสร้างของโหนด 104
รูปที่ 8-4 ตัวอย่างวัตถุที่เกิดจากการนำโหนดหลายๆ โหนดมาประกอบกัน 104
รูปที่ 9-1 Use case diagram ของระบบออกแบบ และสร้างบ้าน 122
รูปที่ 9-2 Sequence ของการลงทะเบียนกรณีข้อมูลถูกต้อง 123
รูปที่ 9-3 Sequence ของการลงทะเบียนกรณีข้อมูลไม่ถูกต้อง 123
รูปที่ 9-4 Sequence ของการ Sign in กรณีข้อมูลถูกต้อง 124
รูปที่ 9-5 Sequence ของการ Sign in กรณีข้อมูลไม่ถูกต้อง 124
รูปที่ 9-6 Sequence ของการเข้าดูแบบบ้าน 124
รูปที่ 9-7 Sequence ของการค้นหาแบบบ้าน 125
รูปที่ 9-8 Sequence ของการสั่งซื้อแบบบ้าน 125
รูปที่ 9-9 Sequence ของการว่าจ้างสร้างบ้าน 126
รูปที่ 9-10 Sequence ของการประเมินราคา 126
รูปที่ 9-11 ER-diagram แสดงความสัมพันธ์ของตารางทั้งหมดในฐานข้อมูล 127
รูปที่ 9-12 Class diagram ของระบบ 128
รูปที่ 10-1 หน้าแรกของเว็บไซต์ 129
รูปที่ 10-2 แบบฟอร์มกรลงทะเบียน 130
รูปที่ 10-3 การล็อกอินเข้าสู่ระบบ 131
รูปที่ 10-4 การค้นหาแบบบ้านที่ต้องการ 131
รูปที่ 10-5 ผลลัพท์จากการค้นหา 132
รูปที่ 10-6 แบบบ้านทั้งหมด 132
รูปที่ 10-7 หน้าจอแสดงเมื่อต้องการดูรายละเอียดของบ้าน 133
รูปที่ 10-8 รายละเอียดของแบบบ้าน 133
รูปที่ 10-9 รายละเอียดการสั่งซื้อแบบบ้าน 134
รูปที่ 10-10 รายละเอียดการสร้างบ้าน 136
รูปที่ 10-11 โมเดลบ้าน 3 มิติ 137
รูปที่ 10-12 การหมุนโมเดลบ้านเพื่อดูบ้านในมุมมองต่างๆ 138
รูปที่ 10-13 การเข้าไปชมภายในบ้าน 138
รูปที่ 10-14 การโต้ตอบระหว่างผู้ใช้กับวัตถุภายในบ้าน 139
รูปที่ 10-15 การเข้าชมห้องต่างๆ ภายในบ้าน 139
รูปที่ 10-16 การเข้าชมห้องต่างๆ ภายในบ้าน 140

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 3-1 ชนิดของ Attribute ใน XML	23
ตารางที่ 3-2 ค่าดีฟอลต์ของ Attribute ใน XML	24
ตารางที่ 3-3 Entity Reference มาตรฐาน	31
ตารางที่ 3-4 XML Parser	34
ตารางที่ 3-5 SOAP Fault Element	40
ตารางที่ 3-6 ค่า Default code	40
ตารางที่ 3-7 SOAP Toolkit	47
ตารางที่ 4-1 เปรียบเทียบความแตกต่างระหว่าง ASP กับ ASP.NET	58
ตารางที่ 5-1 ระดับของ Isolation	81
ตารางที่ 8-1 การเปรียบเทียบชื่อโหนดของเวอร์ชัน 1.0 กับเวอร์ชัน 2.0	102
ตารางที่ 8-2 การเปรียบเทียบของภาษา VRML กับภาษา HTML	102

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

เมื่อกล่าวถึงคำว่า เว็บ (web) เรามักจะนึกถึงเอกสารบนเว็บ (web document) ที่ให้บริการบนเว็บเซอร์เวอร์ (web server) ผ่านโปรโตคอล Hypertext Transfer Protocol (**HTTP**) โดยข้อมูลที่ส่งมาเก็บอยู่ในรูปของ Hypertext Markup Language (**HTML**) เพื่อนำมาแสดงผลบนเว็บเบราเซอร์ (browser) ในตอนแรกเอกสารบนเว็บถูกสร้างขึ้นเพื่อให้นักวิทยาศาสตร์ที่ต้องการแลกเปลี่ยน เอกสารงานวิจัยกัน โดยมีภาคธุรกิจหนุนอยู่เนื่องจากมันสามารถใช้โฆษณาผลิตภัณฑ์ และใช้ในงานบริการได้ ซึ่งเราเรียกว่า เว็บแอพพลิเคชั่น (Web Application)

ในภายหลังภาคธุรกิจมีความต้องการที่จะเชื่อมต่อระบบเข้าหากัน นั่นเป็นเหตุที่ทำให้เกิดบริการบนเว็บรูปแบบใหม่ โดยเก็บอยู่ในรูปของ Simple Object Access Protocol (**SOAP**) โดยเก็บข้อมูลทั้งหมดในรูปของ Extensible Markup Language(**XML**) ซึ่งเราเรียกว่า เว็บเซอร์วิส (Web Service)

ผู้อ่านคงจะเคยได้ยินคำว่า ดอทเน็ต (.Net) กันมาบ้าง และหลายคนน่าจะเคยได้ยินคำว่า เว็บเซอร์วิส (Web Services) บ้างเช่นกัน แล้วดอทเน็ต คืออะไร เว็บเซอร์วิส คืออะไร วิทยานิพนธ์ เล่มนี้จะช่วยอธิบายให้ผู้อ่านได้เข้าใจ พร้อมทั้งได้สร้างระบบขึ้นมาเพื่อเป็นตัวอย่าง ซึ่งจะช่วยให้เข้าใจได้ง่าย และรู้ถึงการนำไปประยุกต์ใช้ต่อไป

เว็บเซอร์วิส เป็นพื้นฐานสำคัญของการบริการใหม่ๆ ในลักษณะที่ต้องการเชื่อมโยง ระบบสารสนเทศที่มีความแตกต่างเข้าด้วยกันอย่างอัตโนมัติ เช่น การพาณิชย์อิเล็กทรอนิกส์ระหว่างธุรกิจกับธุรกิจ

ระบบการบริการรัฐบาลอิเล็กทรอนิกส์ ในลักษณะที่มีการเชื่อมบริการของภาครัฐกับภาครัฐเข้าด้วยกัน เพื่อทำให้เกิดการบริการต่อประชาชนจากจุดเดียวได้ (One Stop Services) เป็นต้น

ตัวอย่างพื้นฐานหนึ่ง คือ การเชื่อมโยงระหว่างระบบการสั่งซื้อ ของบริษัทผู้ขายบริษัทหนึ่ง ไปยังระบบสารสนเทศการ จัดส่งของบริษัทจัดส่งอีกบริษัทหนึ่ง จะเป็นไปได้ง่ายมากขึ้นด้วยเทคโนโลยีของเว็บเซอร์วิส พื้นฐานดังกล่าวนี้ จะทำให้ "การจัดการห่วงโซ่อุปทานด้วยสื่ออิเล็กทรอนิกส์" (e-SCM : Supply Chain Management) และธุรกิจผ่านสื่ออิเล็กทรอนิกส์ที่มีประสิทธิภาพ และมีความคุ้มค่ามากขึ้น โดยมีการแลกเปลี่ยนข้อมูลธุรกิจ ระหว่างองค์กรในห่วงโซ่อุปทาน

ทั้งนี้แต่ละบริษัทจะมีระบบงาน (แอพพลิเคชั่น) และใช้รูปแบบของข้อมูล (ฟอร์แมท ดาต้า) ที่แตกต่างกันได้ ซึ่งเมื่อมีการส่งข้อมูลจากองค์กรหนึ่งไปอีกองค์กรหนึ่ง ระบบคอมพิวเตอร์ของบริษัทผู้รับสามารถนำข้อมูล ที่ได้ไปประมวลผลต่อได้ทันที การเชื่อมโยงระบบอีคอมเมิร์ซของหน้าร้าน กับระบบหลังร้าน แล้วยังเชื่อมโยง กับระบบสารสนเทศของพันธมิตรทางการค้านี้ จะทำให้การทำธุรกิจผ่านสื่ออิเล็กทรอนิกส์ มีความคุ้มค่า มีประสิทธิภาพ และสร้างผลกำไรให้กับบริษัทได้ นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของงานวิจัย

1.2.1. ศึกษาทฤษฎีพื้นฐานของคอมโพเนนต์

1.2.2. ศึกษา ออกแบบ และพัฒนาซอฟต์แวร์เชิงคอมโพเน็นต์

1.2.3. ศึกษาโครงสร้าง และการทำงานของเว็บเซอร์วิส (Web Service) และเทคโนโลยีที่เกี่ยวข้อง ได้แก่ XML, WSDL, SOAP

1.2.4. ศึกษาทฤษฎีและหลักการของ Virtual Reality Modeling Language (VRML) เพื่อนำไปประยุกต์ใช้กับเว็บเซอร์วิส

1.2.5. ศึกษาการเขียนโปรแกรมด้วย Visual Studio .Net

1.3 ขอบเขตของของงานวิจัย

ระบบที่พัฒนาเป็นระบบอี-คอมเมิร์ซ ที่เกี่ยวกับการซื้อขายบ้าน แบบบ้าน ที่อยู่อาศัย ที่ดิน และเครื่องใช้ภายในบ้าน โดยใช้เทคโนโลยีดอทเน็ต (.Net Technology) โดยจะรับผิดชอบในส่วนเว็บเซอร์วิสที่เป็นระบบการซื้อขายแบบบ้าน และการสร้างบ้าน โดยจะมีการติดต่อกับเว็บเซอร์วิสส่วนกลาง เพื่อขอใช้บริการเซอร์วิสอื่นๆ

รายละเอียดของระบบที่จะพัฒนา ประกอบด้วย 2 ส่วน คือ

1.3.1 แอพพลิเคชั่นที่พัฒนาด้วยภาษา Visual Basic .Net บนเทคโนโลยีเว็บเซอร์วิส โดยภาพรวมของระบบคือ ประกอบด้วยการทำงานร่วมกันของบริษัทจำลองที่ให้บริการผ่านเว็บเซอร์วิส 3 บริษัท ด้วยกัน คือ บริษัทที่ให้บริการเกี่ยวกับบ้าน และที่ดิน (Land and Home) บริษัทที่ให้บริการเกี่ยวกับอุปกรณ์ที่ใช้ภายในบ้าน (Home and Décor) และบริษัทที่ให้บริการเกี่ยวกับการซื้อขายแบบบ้าน และการสร้างบ้าน (Home For You) ซึ่งบริการอย่างหลังนี้ จะกล่าวถึงอย่างครอบคลุมในวิทยานิพนธ์ฉบับนี้

1.3.2 แอพพลิเคชั่นที่ทำการติดต่อกับผู้ใช้โดยใช้ภาษา VRML (Virtual Reality Modeling Language) โดยการแสดงผลในรูปแบบ 3 มิติ เพื่อช่วยให้การเข้าไปชมบ้านตัวอย่างกระทำผ่านเว็บได้เสมือนได้ไปเดินดูด้วยตัวเอง

โครงงานนี้เป็นโครงงาน เรื่องการพัฒนาเชิงคอมโพเนนต์ โดยทำศึกษาร่วมกันทั้ง 3 กลุ่ม ประกอบด้วย กลุ่มที่พัฒนาบนแพลตฟอร์ม .NET 2 กลุ่ม และ EJB 1 กลุ่ม มีพื้นฐานอยู่บนการพัฒนาเชิงคอมโพเนนต์ ประกอบกับ เทคโนโลยีเว็บเซอร์วิส

เนื่องจากเป็นการพัฒนาโครงงานร่วมกัน โดยต้องอาศัยพื้นฐานความรู้เดียวกัน ดังนั้นเนื้อหาและทฤษฎีบางส่วนจะมีความเหมือนกัน การแบ่งหัวข้อเพื่อแยกกันไปศึกษา จะทำให้เสียเวลา เพราะทุกกลุ่มก็ต้องศึกษาเนื้อหาเรื่องเดียวกัน ทำให้ผู้พัฒนาโครงงานได้แบ่งหัวข้อในการศึกษาให้แต่ละกลุ่มรับผิดชอบเพื่อนำความรู้ที่ได้มาใช้ร่วมกัน และ ถ่ายทอดให้กลุ่มที่เหลือได้เข้าใจตรงกัน เพื่อเป็นการประหยัดเวลา โดยหัวข้อที่แบ่ง เป็นดังนี้

กลุ่มการพัฒนาเชิงคอมโพเนนต์ ด้วย .NET กลุ่มที่ 1 (Land and House) ศึกษาเรื่องดังต่อไปนี้

1. เว็บเซอร์วิส (Web Services)
2. ADO.NET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. .NET Framework

กลุ่มการพัฒนาเชิงคอมโพเนนต์ ด้วย .NET กลุ่มที่ 2 (Home for you) ศึกษาเรื่องดังต่อไปนี้

1. ASP.NET
2. ASP.NET and Web services
3. VRML

กลุ่มการพัฒนาเชิงคอมโพเนนต์ ด้วย EJB กลุ่มที่ 1 (Home and Decor) ศึกษาเรื่องดังต่อไปนี้

1. ทฤษฎีคอมโพเน็นต์ (Component Theory)
2. สถาปัตยกรรม J2EE

1.4 วิธีการดำเนินงาน

แบ่งเป็นขั้นตอนได้ดังนี้คือ

1.4.1 ศึกษาค้นคว้า

- ศึกษาทฤษฎีและความรู้ที่เกี่ยวข้องกับงานวิจัย และแอพพลิเคชั่นที่จะพัฒนา ได้แก่ COM+, Component, XML, SOAP, Web Services, EJB, .Net
- ศึกษา และค้นคว้าถึงความเป็นไปได้ในการนำเอาเทคโนโลยีต่างๆ มาใช้กับแอพพลิเคชั่นที่จะพัฒนา
- ศึกษาการเขียนโปรแกรม และการสร้างเว็บเซอร์วิส ด้วย Microsoft Visual Studio .Net
- ศึกษาถึงความเป็นไปได้ในการนำภาษา VRML มาใช้ร่วมกับการพัฒนาเว็บเซอร์วิส

1.4.2 ออกแบบ วางแผนโครงสร้าง และกำหนดรายละเอียดของระบบ รวมทั้งการออกแบบอินเทอร์เฟส (Interface) ของระบบ

1.4.3 สร้างแอพพลิเคชั่น และพัฒนา

- ทดลองสร้างเว็บเซอร์วิสอย่างง่ายเพื่อเป็นต้นแบบในการพัฒนาระบบ
- สร้างโมเดลบ้านเพื่อใช้ในแอพพลิเคชั่น โดยใช้ภาษา VRML ในการจัดการ
- สร้างแอพพลิเคชั่นที่จะให้บริการ รวมทั้งทดสอบการเรียกใช้งานแอพพลิเคชั่น
- ศึกษาถึงปัญหาที่เกิดขึ้น พร้อมทั้งหาวิธีแก้ไข

1.5 ประโยชน์ที่คาดว่าจะได้รับ

เพื่อเป็นการศึกษาเทคโนโลยีใหม่ที่จะเป็นแนวโน้มของการพัฒนาซอร์ฟแวร์ในอนาคต เพื่อที่จะเข้าใจถึงข้อดี และข้อเสียของเทคโนโลยีนี้ และนำมาประยุกต์ใช้งานกับระบบอีคอมเมิร์ช (E-Commerce) โดยอาศัยข้อดีของเว็บเซอร์วิสที่สามารถสร้างซอร์ฟแวร์ที่เป็นอิสระต่อกัน รวมทั้งการได้ศึกษาการใช้ภาษา VRML เพื่อนำไปประยุกต์ใช้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีคอมโพเน็นต์

2.1 คอมโพเน็นต์

คอมโพเน็นต์ คือ ส่วนย่อยของระบบที่ไม่ขึ้นกับส่วนอื่นๆ และซ่อนรายละเอียดการทำงานไว้ภายใน โดยมีส่วนติดต่อกับภายนอกเพื่อให้สามารถเรียกใช้งานได้ ความสามารถนี้ทำให้คอมโพเน็นต์สามารถนำ กลับ มาใช้ใหม่ได้ (reuse)

การพัฒนาเชิงคอมโพเน็นต์นั้น มีความหมายกับคนแต่ละกลุ่มแตกต่างกันไปตามความต้องการ ซึ่งมีลักษณะเด่นบางอย่าง ที่นิยามความเป็นระบบคอมโพเน็นต์ดังต่อไปนี้

1. คอมโพเน็นต์มีลักษณะของออบเจ็กต์ คือ การซ่อนรายละเอียด (Encapsulated), โพลีมอร์ฟิค (polymorphic), การกำหนดหน้าที่ และการกำหนดอินเทอร์เฟซ
2. คอมโพเน็นต์ออกแบบภายใต้เฟรมเวิร์ก (Framework) ซึ่งได้สร้างข้อจำกัดบางอย่างไว้ เช่น ต้องไม่มีหลายเธรด, ไม่มีการติดต่อกับภายนอกโดยไม่ผ่านบริการของเฟรมเวิร์ก
3. คอมโพเน็นต์สามารถอยู่ได้ด้วยตัวเอง โดยไม่พึ่งพาคอมโพเน็นต์อื่นๆ ยกเว้น คอมโพเน็นต์ของเฟรมเวิร์ก ที่คอมโพเน็นต์ดังกล่าวใช้อยู่
4. ทุกคอมโพเน็นต์มีอินเทอร์เฟซสามัญที่แน่นอน (Fix and common) และอินเทอร์เฟซนี้เปลี่ยนแปลงไม่ได้
5. คอมโพเน็นต์สามารถอธิบายตนเองได้โดยอินเทอร์เฟซของคอมโพเน็นต์จะต้องมีข้อมูลมากพอ ที่สามารถทำให้ไคลเอ็นต์สามารถเข้าใจวิธีใช้คอมโพเน็นต์นั้นได้

2.1.1 เฟรมเวิร์ก คือ สภาพแวดล้อมที่มีลักษณะดังต่อไปนี้

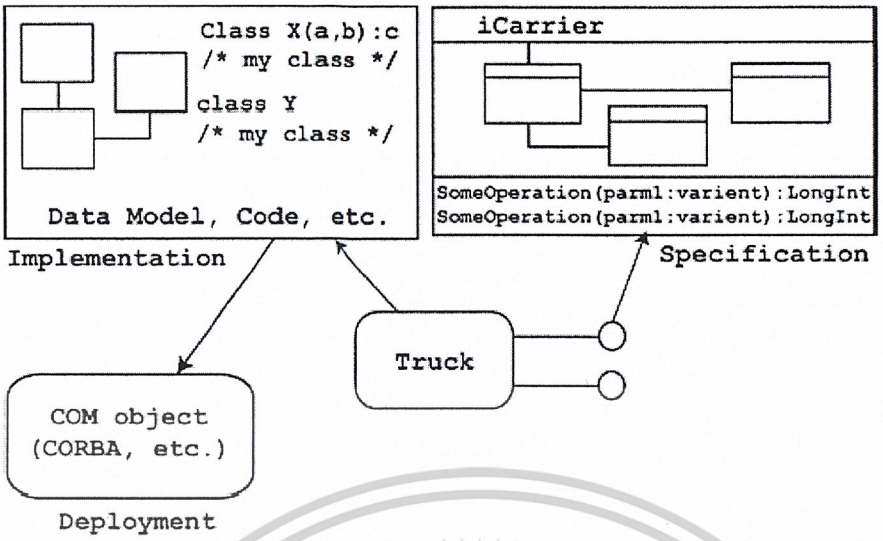
1. สร้างอินสแทนซ์ของคอมโพเน็นต์ตอนรันไทม์
2. ทำให้คอมโพเน็นต์ค้นพบคอมโพเน็นต์อื่นๆ ได้
3. ทำให้คอมโพเน็นต์ต่างๆ สามารถติดต่อกันได้
4. จัดหาบริการสามัญ เช่น เพอร์ซิสเทนซ์ (persistence), ทรานส์แอ็กชัน (transaction), ความไม่ขึ้นกับสถานที่, การรักษาความปลอดภัย, มอนิเตอริง (monitoring)

2.1.2 ส่วนประกอบของคอมโพเน็นต์

คอมโพเน็นต์จะประกอบด้วยส่วนสำคัญ 3 ส่วนดังนี้

- อินเทอร์เฟซ (interface) - คอมโพเน็นต์จะถูกเรียกใช้งานผ่านทางอินเทอร์เฟซ
- อิมพลีเมนต์เตชัน (implementation) - เป็นโค้ดที่กำหนดการทำงานของคอมโพเน็นต์
- ดีพลอยเมนต์ (deployment) – เป็นเอ็กซิคิวต์ไฟล์จะใช้ในการทำให้คอมโพเน็นต์ทำงานได้ ทำหน้าที่จัดหารันไทม์เอ็นไวรอนเมนต์ในการควบคุมการทำงานของคอมโพเน็นต์และจัดหาเซอร์วิสที่จำเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

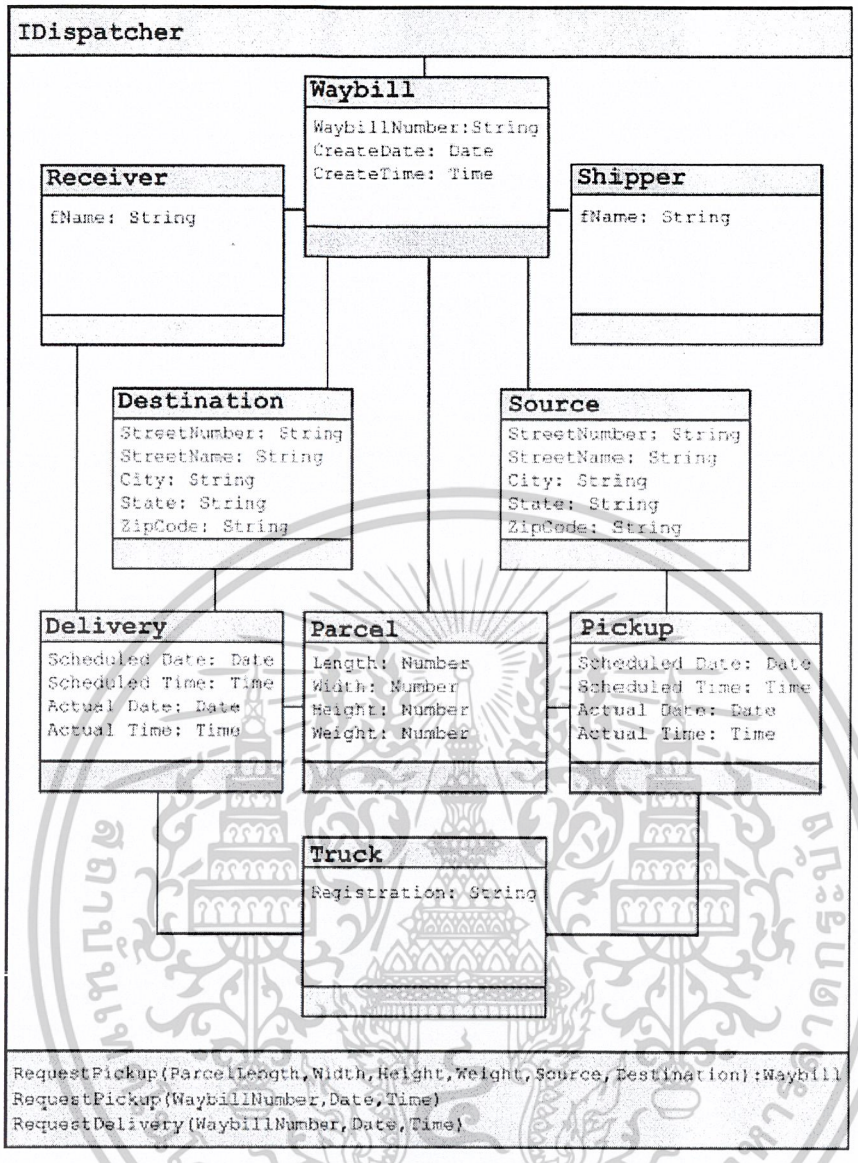


รูปที่ 2-1 คอมโพเน็นต์และส่วนประกอบต่าง ๆ ของมัน

2.1.3 คุณสมบัติของคอมโพเน็นต์

- **Encapsulated** - เป็นกระบวนการในการซ่อนโค้ดหรืออิมพลีเมนต์เตชันของคอมโพเน็นต์ ผู้ใช้จะรู้เพียง อินเทอร์เฟซและจะใช้งานผ่านมัน โดยไม่จำเป็นต้องรู้ถึงอิมพลีเมนต์เตชันหรือการทำงาน ทำให้ การเปลี่ยนแปลงอิมพลีเมนต์เตชันไม่มีผลกระทบต่อผู้ใช้
- **Descriptive** - เนื่องจากคอมโพเน็นต์จะต้องติดต่อผ่านอินเทอร์เฟซเท่านั้น มันจะต้องมีอินฟอร์เมชันของ มันเอง ที่ผู้ใช้สามารถจะเข้าใจได้ โดยอินฟอร์เมชันนั้นจะต้องอธิบายถึง อินเทอร์เฟซ, อิมพลีเมนต์เตชัน และดีพลอยเมนต์

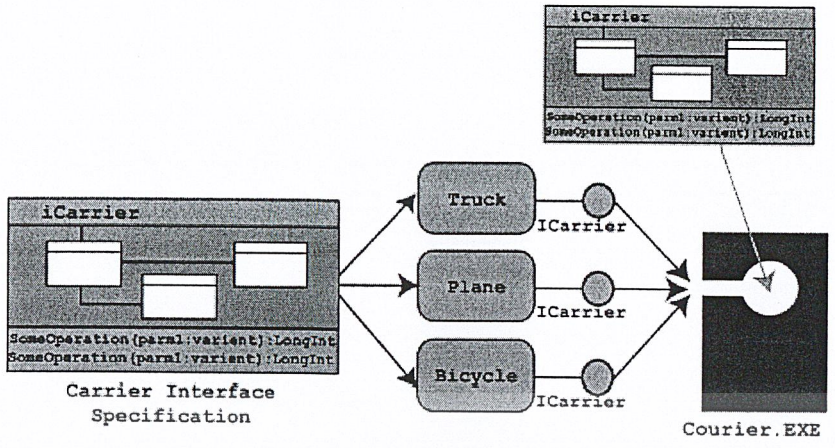
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-2 ตัวอย่างของอินเทอร์เฟซ

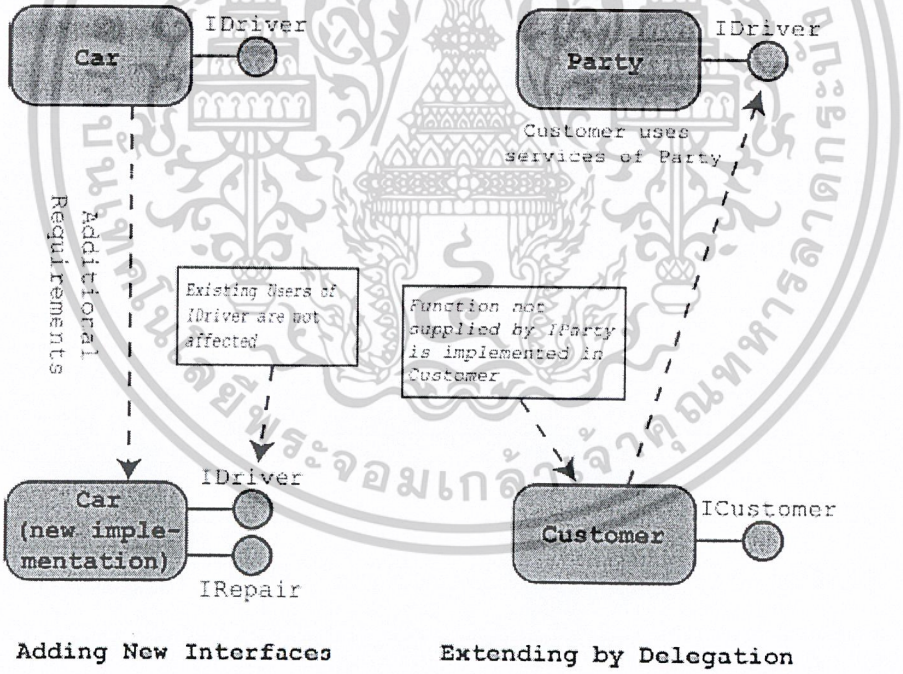
- **Replaceable** –
  คอมโพเน็นต์มีความสามารถในการเปลี่ยนคอมโพเน็นต์หนึ่งกับคอมโพเน็นต์ ใดๆ ที่มีอินเทอร์เฟซเหมือนกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-3 คอมโพเน็นต์และการแทนที่

- **Extensible** - สามารถที่จะเพิ่มความสามารถได้ มี 2 วิธีคือ การเพิ่มอินเทอร์เฟซและมอบหมายหน้าที่ (Delegating Responsibility) การสร้างคอมโพเน็นต์ใหม่สามารถมอบหมายหน้าที่ให้กับบริการที่มีอยู่ในคอมโพเน็นต์ที่มีอยู่แล้ว



รูปที่ 2-4 การเพิ่มเติมคอมโพเน็นต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 งานที่เหมาะสมกับการพัฒนาเชิง คอมโพเน็นต์

การพัฒนาเชิงคอมโพเน็นต์ เป็นวิธีการพัฒนาคอมโพเน็นต์โดยใช้คอมโพเน็นต์เฟรมเวิร์ก ซึ่งการคงหน้าที่ของคอมโพเน็นต์ไว้อย่างถูกต้อง (Well defined responsibility) กระทำได้โดย ให้คอมโพเน็นต์เชิงธุรกิจ หลีกเลี่ยงโค้ดที่ไม่สามารถทำงานกับอินเทอร์เฟซที่แน่นอนได้

เฟรมเวิร์กคอมโพเน็นต์จะซ่อนบริการต่างๆ ของเอ็นไวรอนเม็นต์ ซึ่งบริการต่างๆ เหล่านี้เป็นมาตรฐานแล้ว เช่น message oriented middle-ware, Transaction Monitors, การรักษาความปลอดภัย และพิสูจน์สิทธิ์

2.3 ประโยชน์ของการพัฒนาเชิงคอมโพเน็นต์

การพัฒนาเชิงคอมโพเน็นต์มีประโยชน์อย่างมากด้านเกี่ยวกับเทคนิค และประโยชน์เหล่านี้ ยังนำไปสู่ประโยชน์ด้านธุรกิจด้วย

ประโยชน์ด้านเทคนิค

1. จัดการความซับซ้อนได้ง่ายขึ้น ทำให้คุณภาพของการแก้ปัญหาซอฟต์แวร์ง่ายขึ้น
2. งานซับซ้อนที่ไม่เกี่ยวกับฟังก์ชันธุรกิจ ถูกรวมไว้ในเฟรมเวิร์ก
3. การออกแบบ, อิมพลีเมนต์, ทดสอบ ส่วนต่างๆอย่างอิสระ ทำให้สามารถพัฒนาส่วนต่างๆ ได้ พร้อมกัน
4. การไม่ขึ้นต่อกันของคอมโพเน็นต์ลดผลกระทบของการเปลี่ยนแปลง requirement ไม่ให้กระจาย ไปทั้งระบบ

ประโยชน์ด้านธุรกิจ

1. ผลิตภัณฑ์มีคุณภาพมากขึ้น
2. เร่งเวลาออกสู่ตลาด
3. ใช้ทรัพยากรบุคคลได้คุ้มค่าขึ้น
4. ลดค่าใช้จ่าย
5. เพิ่มปริมาณการนำกลับมาใช้ ในโครงการต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
เว็บเซอร์วิส

เว็บเซอร์วิส (Web Services) คือ แอพพลิเคชั่น หรือ โปรแกรมที่ทำงานอย่างใดอย่างหนึ่ง ในลักษณะให้บริการ โดยจะถูกเรียกใช้งานจาก แอพพลิเคชั่น อื่นๆ ในรูปแบบ RPC (Remote Procedure Call) ซึ่งการเปลี่ยนคือ XML ทำให้เราสามารถเรียกใช้คอมโพเน็นต์ ใดๆ ก็ได้ ในเพลตฟอร์ม (platform) ใดๆ ก็ได้ บนโพรโตคอล HTTP ซึ่งเป็นโพรโตคอลสำหรับเวิร์ด ไวด์ เว็บ (World Wide Web) อันเป็นช่องทางที่ได้รับการยอมรับทั่วโลกในการติดต่อสื่อสารกันระหว่างแอพพลิเคชั่นกับแอพพลิเคชั่นในปัจจุบัน

เทคโนโลยีในการกระจายข่าวสารข้อมูลทางอินเทอร์นีต ในปัจจุบันก็คือ เว็บเพจ แต่จากการที่มันมีความสามารถที่จะทำงานได้ด้วยการรวมภาษาทั้ง ไคล์เอนต์ (Client) และเซิร์ฟเวอร์ สไลด์ สคริปต์ (Server Side Script) ไว้ในตัวเองเช่นภาษา VBScript, Java Script หรือ ASP, PHP, JSP นั้นทำให้เว็บเพจมีลักษณะคล้ายแอพพลิเคชั่น จึงถูกเรียกรวมกันว่า เว็บแอพพลิเคชั่น

เว็บแอพพลิเคชั่น สามารถตอบสนองความคิดดิททิบิวต์เตต โปรเซสซิ่ง (Distributed Processing) ได้ในระดับหนึ่งซึ่งก็คือ การแบ่งการประมวลผลไว้ที่ฝั่งไคล์เอนต์ และฝั่งเซิร์ฟเวอร์ และมักจะมีการใช้ฐานข้อมูล ควบคู่กับการทำ เว็บแอพพลิเคชั่น ไปด้วยตามความต้องการในการทำอี-บิชสิเนส (E-Business) และอี-คอมเมิร์ช (E-Commerce) ที่กำลังเป็นที่นิยมในปัจจุบัน และเกิดปัญหาที่ตามาคือ เรื่องของการจ่ายเงินหรือที่เรียกว่า อี-เพเมนต์ (E-payment) หรือ เพเมนต์-เกตเวย์ (Payment-Gateway) ซึ่ง เว็บแอพพลิเคชั่น ที่ทำอี-คอมเมิร์ช ต้องใช้บริการจากธนาคารออนไลน์ ในการจัดเก็บเงินกับลูกค้า เพราะด้วยเทคโนโลยีนี้การใช้บริการเก็บเงินจากธนาคารออนไลน์ จำเป็นที่ผู้ค้าต้องไปทำการตกลงกับธนาคารและเขียนโปรแกรมให้ตรงตามมาตรฐานที่ธนาคารออนไลน์ กำหนดไว้

ด้วยปัญหายุ่งยากในการค้นหา ติดต่อและตกลงในการขอใช้บริการเก็บเงินจากธนาคาร ออนไลน์ แนวคิดเว็บเซอร์วิสจึงดูเหมือนเป็นทางออกของปัญหานี้ ความเด่นของเทคโนโลยีเว็บเซอร์วิส นี้ก็คือ การทำให้เว็บกับเว็บ สามารถติต่อสื่อสารกันได้ด้วยเอกสาร XML ที่ทั้งคนและคอมพิวเตอร์เข้าใจ และคอมพิวเตอร์ยังสามารถนำข้อมูลนั้นไปประมวลผลต่อได้ ด้วยเอกสาร XML นี้เองทำให้เว็บสามารถส่งข้อมูลที่จำเป็นไปให้อีก เว็บ หนึ่งทำงานบางอย่างให้หรือใช้บริการนั่นเอง ทำให้เป็นการง่ายที่จะเขียนโปรแกรมที่จะติดต่อสื่อสารหรือขอใช้บริการเก็บเงินจากธนาคาร ออนไลน์ แต่สำหรับ เว็บแอพพลิเคชั่น ที่ใช้การส่งข้อมูลเป็น html ทำให้ข้อมูลนั้นไม่สามารถนำไปใช้ต่อได้ การเขียนโปรแกรมจึงยุ่งยากตามที่กล่าวด้านบน

แนวคิดของ เว็บเซอร์วิสก็คือ เว็บ ที่สามารถทำงานอะไรบางอย่างหรือก็คือให้บริการบางอย่างจากการร้องขอจากต่างเซิร์ฟเวอร์ ด้วยเหตุนี้ทำให้เทคโนโลยี เว็บเซอร์วิสเอื้อต่อแนวคิดดิททิบิวต์เตต โปรเซสซิ่ง มากกว่า เว็บแอพพลิเคชั่น และเมื่อประกอบกับการที่ เว็บเซอร์วิสมี UDDI ทำให้ เว็บเซอร์วิสสามารถค้นหาบริการต่างๆ ที่ต้องการได้จากทั่วทุกมุมโลก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในอนาคตอาจเป็นไปได้ว่า แอพพลิเคชั่น ของก็อาจเป็นเพียงแค่การรวมบริการ (Service) ที่
เว็บเซอร์วิสมีบริการมาให้ใช้เท่านั้นเอง

องค์ประกอบในการใช้งาน .NET เว็บเซอร์วิส

3.1 XML

3.1.1 แนะนำ XML

XML ย่อมาจาก eXtensible Markup Language XML ถูกออกแบบมาเพื่อใช้อธิบายข้อมูล ใน XML จะไม่มีแท็ก (tag) ที่กำหนดไว้ล่วงหน้า เราจะต้องสร้างแท็กของเราขึ้นมาเอง และจะใช้ Document Type Definition (DTD) ในการอธิบายรูปแบบของข้อมูล

ข้อแตกต่างที่สำคัญระหว่าง XML และ HTML คือ XML ถูกออกแบบมาเพื่อใช้แลกเปลี่ยนข้อมูล ไม่ใช่มาแทนที่ HTML XML กับ HTML ถูกออกแบบมาเพื่อจุดประสงค์ที่ต่างกัน HTML จะเกี่ยวข้องกับการแสดงข้อมูล XML จะเกี่ยวข้องกับการอธิบายข้อมูลแท็กของ HTML เช่น <H1></H1> เป็นแท็กสำหรับบอกรูปแบบการแสดงผล ในขณะที่แท็กของ XML ต้องสร้างขึ้นมาเองเช่น

```
<BOOK>
   <NAME>Beginning XML</NAME>
   <ISDN>1-861003-41-2</ISDN>
   <PAGE>800</PAGE>
<BOOK>
```

จะใช้แสดงข้อมูลของหนังสือที่ประกอบด้วย element BOOK , NAME , ISDN และ PAGE ข้อมูลที่อยู่ในรูปเท็กซ์นี้เป็นอิสระต่อซอฟต์แวร์และฮาร์ดแวร์ จึงเหมาะที่จะนำมาใช้ในการแลกเปลี่ยนข้อมูลระหว่างระบบบนอินเทอร์เน็ต การเปลี่ยนข้อมูลให้อยู่ในรูป XML เป็นการลดความซับซ้อนและสามารถสร้างข้อมูลที่ถูกอ่านโดยแอพพลิเคชันใด ๆ ก็ได้ นอกจากนั้นการเพิ่มเติมอีรีเมนต์ (element) จะไม่ทำให้แอพพลิเคชันเสียหาย เนื่องจากการอ่านข้อมูลโดย XML parser จะใช้วิธีแยกข้อมูลจาก element ที่ต้องการ จาก XML ข้างต้นนี้จะแยกข้อมูล element NAME , ISDN และ PAGE หากเพิ่มอีรีเมนต์ AUTHOR เข้าไป แอพพลิเคชันก็ยังคงดึงข้อมูลในอีรีเมนต์เดิม โดยจะมองข้ามอีรีเมนต์ที่ไม่ได้กำหนดให้แยกข้อมูลออกมาในตอนพัฒนาแอพพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 ไวยากรณ์ของ XML

กฎไวยากรณ์ของ XML นั้นง่าย และเคร่งครัดมาก กฎของมันง่ายต่อการเรียนรู้ และง่ายต่อการใช้ ด้วยเหตุนี้การสร้างซอฟต์แวร์ในการอ่าน และจัดการกับข้อมูลจึงเป็นเรื่องง่าย ตัวอย่างของเอกสาร XML ดังนี้

```
<?XML version="1.0"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

ในบรรทัดแรกคือการประกาศเอกสาร XML ที่บอกถึง XML เวอร์ชัน

บรรทัดต่อมาคือ root element (note) อีก 4 บรรทัดถัดมาเป็นการอธิบายถึง 4 child element ของ root (to , from , heading , body) และบรรทัดสุดท้ายเป็นการกำหนดจุดสิ้นสุดของ root

กฎของเอกสาร XML มีดังนี้

- **XML Element ทุกตัวต้องมีการปิดแท็ก**

ใน HTML บาง element ไม่จำเป็นต้องมีการปิดแท็ก แต่ใน XML ทุก element จำเป็นที่จะต้องมีการปิดแท็ก

This is a paragraph	X
This is a paragraph	√

- **XML แท็กมีความแตกต่างระหว่างตัวพิมพ์เล็กและตัวพิมพ์ใหญ่ (case sensitive)**

ใน XML แท็ก <Letter> นั้นต่างจาก <letter> เนื่องจาก XML นั้นมีคุณสมบัติที่แยกความแตกต่างระหว่างตัวพิมพ์เล็กและตัวพิมพ์ใหญ่

<Message>This is incorrect</message>	X
<message>This is correct</message>	√

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **XML Element ทุกตัวต้องมีการซ้อนกันอย่างเหมาะสม**

  ใน HTML บาง element สามารถเหลื่อมกันได้ แต่ใน XML นั้นต้องซ้อนกันเป็นขั้นโดยสมบูรณ์

```
<b><i>This text is bold and italic</b></i>    X
<b><i>This text is bold and italic</i><b>     √
```

- **ทุกเอกสาร XML ต้องมี root แท็ก**

  แท็กแรกในเอกสาร XML นั้นจะเป็น root แท็กและจะมีได้เพียงหนึ่งเดียวเท่านั้น element อื่น ๆ ทั้งหมดจะต้องถูกซ้อนอยู่ภายใน root element เท่านั้น และ element เหล่านั้นก็สามารถมี element ย่อย (child element) ได้อีก ดังนี้

```
<root>
   <child>
      <subchild>.....</subchild>
   </child>
</root>
```

- **ค่าของ Attribute ต้องมีอยู่ภายในเครื่องหมายคำพูด (“”)**

  XML element สามารถมี attributeได้ โดยที่ค่าของ attribute จะต้องอยู่ภายในเครื่องหมายคำพูด (“) เอกสาร XML ข้างล่างนี้ อันแรกผิด อันที่สองถูก

```
<note date=12/11/99>          X
<note date=”12/11/99”>        √
```

3.1.3 XML Element

XML element มีคุณสมบัติดังนี้

- **XML Element สามารถขยายได้**

  ตามตัวอย่างนี้

```
<note>
   <to>Tove</to>
   <from>Jani</from>
   <body>Don't forget me this weekend!</body>
</note>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าเราจะสร้างแอพพลิเคชันที่สามารถแยก element to , from และ body จาก เอกสาร XML เพื่อแสดงผลลัพธ์ดังนี้

MESSAGE

To: Tove

From: Jani

Don't forget me this weekend!

และ ถ้าเพิ่มข้อมูลเพิ่มเติมลงไปดังนี้

```
<note>
   <date>1999-08-01</date>
   <to>Tove</to>
   <from>Jani</from>
   <heading>Reminder</heading>
   <body>Don't forget me this weekend!</body>
</note>
```

เมื่อเอกสาร XML เป็นดังนี้แล้ว แอพพลิเคชันที่สร้างไว้จะ ไม่เสียหาย เนื่องจาก แอพพลิเคชันจะยังคงค้นหา element to, from และ heading ในเอกสาร XML และยังคงให้ผลลัพธ์เหมือนเดิม

- **XML Element มีความสัมพันธ์กัน**

ในการที่จะเข้าใจถึงภาพรวมของ XML จะต้องเข้าใจถึงความสัมพันธ์ระหว่างชื่อของ XML element ที่ตั้งและ content ของ element

คำอธิบายของหนังสือดังนี้

Book Title: My First XML

Chapter 1: Introduction to XML

What is HTML

What is XML

Chapter 2: XML Syntax

Elements must have a closing tag

Elements must be correctly nested

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถมีเอกสาร XML ที่อธิบายหนังสือนั้น ได้ดังนี้

```
<book>
   <title>My First XML</title>
   <prod id="33-657" media="paper"></prod>
   <chapter>Introduction to XML
      <para>What is HTML</para>
     <para>What is XML</para>
   </chapter>
   <chapter>XML Syntax
      <para>Elements must have a closing tag</para>
      <para>Elements must be properly nested</para>
   </chapter>
</book>
```

book เป็น root element title และ chapter เป็น child element ของ book book เป็น parent element ของทั้ง title และ chapter title และ chapter จึงเป็น sibling กันเนื่องจากมี parent เดียวกัน

- **Element มี Content**

XML element คือทั้งหมด ตั้งแต่เปิดแท็กจนถึงปิดแท็ก element หนึ่งสามารถมี element content, mixed content, simple content หรือ empty content และยังมี attribute ได้ด้วย

จากตัวอย่างข้างบน book มี element content เพราะมันประกอบด้วย element อื่น chapter เป็น mixed content เพราะมันประกอบด้วยเท็กซ์และ element อื่น ๆ para เป็น simple content (text content) เพราะมันประกอบด้วยเท็กซ์เท่านั้น prod เป็น empty element เนื่องจากมันไม่มีข้อมูลใดๆ และเฉพาะ prod element เท่านั้นที่มี attribute attribute ที่ชื่อ id มีค่าเท่ากับ "33-657" attribute ที่ชื่อ medis มีค่าเท่ากับ "paper"

- **การตั้งชื่อ Element**

XML element มีกฎการตั้งชื่อดังนี้

- ชื่อประกอบด้วย ตัวอักษร , ตัวเลข และอักขระอื่นๆ
- ชื่อต้องไม่ขึ้นต้นด้วยตัวเลขหรือ "_ "(underscore)
- ชื่อต้องไม่ขึ้นด้วยคำว่า XML (หรือ Xml หรือ xml)
- ชื่อไม่สามารถมีช่องว่างได้ (space)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ชื่อต้องไม่ควรประกอบด้วย ":" เพราะมันถูกสงวนไว้กับการใช้ namespace

ทุกๆชื่อสามารถใช้ได้ ไม่มีคำใดเป็นคำสงวน แต่จะเป็นการดีในการที่จะหลีกเลี่ยงการใช้ "-" หรือ "." แล้วใช้เฉพาะ "_" เท่านั้น เนื่องจากอาจเกิดการสับสนของซอฟต์แวร์ในกรณีพยายามที่จะตัดคำ (first-name) หรือคิดว่า "name" เป็นคุณสมบัติของ object "first" (first.name)

3.1.4 XML Attribute

XML attribute มีคุณสมบัติดังนี้

- รูปแบบของ **Quote , "female" or 'female'**

ค่าของ attribute ต้องอยู่ในเครื่องหมายคำพูดเสมอ แต่อาจเป็น ' (single quote) หรือ " (double quote) ก็ได้

```
<person sex="female">   √
หรือ
<person sex='female'>   √
```

Double quote จะเป็นที่ใช้โดยทั่วไปมากกว่า แต่ในบางครั้ง (ถ้าค่าของ attribute จำเป็นที่จะต้องมี quote) จึงจำเป็นที่จะต้องใช้ single quote ดังนี้

```
<gangster name='George "Shotgun" Ziegler'>
```

- การใช้ **Element และ Attribute**

ข้อมูลสามารถที่จะถูกเก็บใน child element หรือ attribute ดังนี้

```
<person sex="female">
    <firstname>Anna</firstname>
    <lastname>Smith</lastname>
</person>
```

หรือ

```
<person>
    <sex>female</sex>
    <firstname>Anna</firstname>
    <lastname>Smith</lastname>
</person>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างแรก sex เป็น attribute ตัวอย่างที่ 2 sex เป็น child element ซึ่งทั้งสองตัวอย่างให้ความหมายเดียวกัน มันไม่มีกฎว่าจะใช้ attribute เมื่อไร หรือ child element เมื่อไร แต่แนะนำว่าควรจะหลีกเลี่ยงการใช้attribute ควรจะใช้ child element ถ้าอินฟอร์เมชันนั้นดูเหมือนจะเป็นข้อมูล (data)

- **หลีกเลี่ยงการใช้ Attribute**

  บางปัญหาของการใช้ attribute

  - attribute ไม่สามารถมีหลายค่า (child element สามารถมีได้)
  - attribute ไม่ง่ายที่จะขยาย (ในอนาคต)
  - attribute ไม่สามารถอธิบายโครงสร้าง (child element สามารถทำได้)
  - attribute ยากที่จะจัดการด้วยโค้ดของโปรแกรม
  - ค่าของ attribute ไม่ง่ายในการจะตรวจสอบกับ DTD ที่มีอยู่

แต่อย่างไรก็ตาม จะสามารถใช้ attribute ก็ต่อเมื่อ อินฟอร์เมชันนั้นไม่สัมพันธ์กับข้อมูล ตัวอย่างเช่น ในบางครั้ง ID reference สามารถถูกใช้ในการเข้าถึง XML element

```
<messages>
  <note ID="501">
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
  </note>
  <note ID="502">
    <to>Jani</to>
    <from>Tove</from>
      <heading>Re: Reminder</heading>
      <body>I will not!</body>
  </note>
</messages>
```

ID ในตัวอย่างนี้เป็น counter หรือ unique identifier ในการอ้างถึง note ที่ต่างกันของไฟล์ XML ไม่ใช่ส่วนหนึ่งของข้อมูล note

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

3.1.5 XML Validation

เอกสาร XML แบ่งการตรวจสอบความถูกต้องของเอกสารเป็น 2 ชนิดคือ

1) **"Well Formed" XML Document**

"Well Formed" XML document เป็นเอกสารที่เป็นไปตามกฎไวยากรณ์ XML ที่ได้กล่าวมาข้างต้น

2) **"Valid" XML Document**

"Valid" XML document เป็นเอกสาร XML ที่มีรูปแบบที่เป็นไปตามกฎของ DTD โดยจะต้องตรงตามโครงสร้างข้อมูลของเอกสารที่ได้ประกาศโดย DTD

- **XML DTD**

วัตถุประสงค์ของ DTD คือกำหนดโครงสร้างของเอกสาร XML เพื่อใช้ในการตรวจสอบว่าเอกสาร XML นั้นถูกต้องตามรูปแบบหรือไม่ เช่น element BOOK จะต้องมี element NAME เสมอ อาจจะมี element PAGE หรือไม่ก็ได้ เป็นต้น

- **XML Schema**

XML schema มีวัตถุประสงค์เหมือนกับ DTD จะถูกนำมาใช้แทน DTD เนื่องจาก

- ง่ายในการเรียนรู้มากกว่า DTD
- ใช้ได้ดีกว่า DTD
- schema เขียนด้วย XML
- รองรับ datatype
- รองรับ namespace

3.1.6 แนะนำ DTD

1) การประกาศ **DOCTYPE** ภายใน

ถ้า DTD ถูกรวมอยู่ในไฟล์ XML นั้น มันจะต้องอยู่ภายในการให้คำจำกัดความ DOCTYPE ตามไวยากรณ์ นี้

```
<!DOCTYPE root-element [element-declarations]>
```

ตัวอย่างดังนี้

```
<?XML version="1.0"?>
<!DOCTYPE note [
  <!ELEMENT note (to,from,heading,body)>
  <!ELEMENT to     (#PCDATA)>
```

55075

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        <!ELEMENT from   (#PCDATA)>
        <!ELEMENT heading (#PCDATA)>
        <!ELEMENT body   (#PCDATA)>
    ]>
    <note>
        <to>Tove</to>
        <from>Jani</from>
        <heading>Reminder</heading>
        <body>Don't forget me this weekend</body>
</note>
```

DTD ข้างบนนี้สามารถแปลความหมายได้ดังนี้

!DOCTYPE note (บรรทัดที่ 2) กำหนดว่าเป็นเอกสารของ type note

!ELEMENT note (บรรทัดที่ 3) กำหนดว่า note element มี 4 element คือ to , from , heading และ body

!ELEMENT to (บรรทัดที่ 4) กำหนดว่า to element เป็นข้อมูลชนิด #PCDATA

!ELEMENT from (บรรทัดที่ 5) กำหนดว่า from element เป็นข้อมูลชนิด #PCDATA

2) การประกาศ **DOCTYPE** ภายนอก

ถ้า DTD อยู่ภายนอกไฟล์ XML มันจะต้องอยู่ภายในการให้คำจำกัดความ DOCTYPE ตามไวยากรณ์นี้

```
<!DOCTYPE root-element SYSTEM "filename">
```

ตัวอย่างไฟล์ XML เป็นดังนี้

```
    <?XML version="1.0"?>
    <!DOCTYPE note SYSTEM "note.dtd">
    <note>
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
</note>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์ของ DTD ชื่อ note.dtd เป็นดังนี้

note.dtd

```
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
```

3.1.6.1 โครงสร้างของ XML

เอกสาร XML จะถูกสร้างจากส่วนต่างๆ ดังนี้

- Element – เป็นโครงสร้างหลักของ XML
- Tag - ใช้ในการกำหนด markup element
- Attribute - ให้ข้อมูลเพิ่มเติมกับ element
- Entity – เป็นตัวแปรที่ใช้ในการกำหนดเท็กซ์ทั่ว ๆ ไป entity reference เป็นการอ้างถึง entity เช่น entity ชื่อ domain มีค่าเท่ากับ www.w3.org entity reference คือ &domain หากภายในเอกสาร XML มี &domain ค่าของมันจะมีค่าเท่ากับ www.w3.org
- PCDATA -หมายถึง parse character data PCDATA เป็นเท็กซ์ที่จะถูก parse โดย parser
- CDATA -หมายถึง character data เป็นเท็กซ์ที่จะไม่ถูก parse โดย parser

3.1.6.2 การกำหนด Element ใน DTD

วิธีการในการประกาศ element จะเป็นดังนี้

- **การประกาศ Element**

ประกาศตามไวยากรณ์นี้

```
<!ELEMENT element-name category>
```

หรือ

```
<!ELEMENT element-name (element-content)>
```

- **Element ว่างเปล่า (Empty)**

ถูกประกาศโดยคีย์เวิร์ด EMPTY

```
<!ELEMENT element-name EMPTY>
```

DTD example:

```
<!ELEMENT br EMPTY>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XML example:

```
<br />
```

- **Element ที่เป็นข้อมูลชนิดตัวอักษร**

  element ที่เป็นตัวอักษรเท่านั้นจะถูกประกาศด้วยคีย์เวิร์ด PCDATA ที่เปิดและปิดด้วย "(" ")"

```
<!ELEMENT element-name (#PCDATA)>
```

DTD example:

```
<!ELEMENT note (#PCDATA)>
```

- **Element ที่เป็นข้อมูลใด ๆ**

  ประกาศกับคีย์เวิร์ด ANY content จะสามารถเป็นข้อมูลชนิดใดก็ได้

```
<!ELEMENT element-name ANY>
```

DTD example:

```
<!ELEMENT note ANY>
```

- **Element และ Children (Sequences)**

  ประกาศโดยใช้ "," ในการแยก child element แล้วเปิดและปิดด้วย "(" ")"

```
<!ELEMENT element-name (child-element-name)>
```

หรือ

```
<!ELEMENT element-name (child-element-name,child-element-name,.....)>
```

DTD example:

```
<!ELEMENT note (to,from,heading,body)>
```

ถ้ามีการประกาศ child element จะต้องประกาศ child element ให้สมบูรณ์ด้วยดังตัวอย่างข้างล่างนี้

```
<!ELEMENT note (to,from,heading,body)>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<!ELEMENT to     (#PCDATA)>
<!ELEMENT from   (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body   (#PCDATA)>
```

- **การประกาศเพียงครั้งเดียวของ Element เดียวกัน**

```
<!ELEMENT element-name (child-name)>
```

DTD example:

```
<!ELEMENT note (message)>
```

จากตัวอย่างหมายความว่า note element จะประกอบด้วย child element message เท่านั้น

- **การประกาศอย่างน้อย 1 ครั้งของ Element เดียวกัน**

```
<!ELEMENT element-name (child-name+)>
```

DTD example:

```
<!ELEMENT note (message+)>
```

เครื่องหมาย + ถูกใช้ในความหมายหนึ่งหรือมากกว่า จึงหมายความว่า note element จะประกอบด้วย child element message ตั้งแต่ 1 ค่าขึ้นไป

- **การไม่ประกาศหรือประกาศเท่าไรก็ได้ของ Element เดียวกัน**

```
<!ELEMENT element-name (child-name*)>
```

DTD example:

```
<!ELEMENT note (message*)>
```

เครื่องหมาย * ถูกใช้ในความหมายว่าไม่มีหรือมีเท่าไรก็ได้ จึงหมายความว่า note element จะประกอบด้วย child element message กี่ค่าก็ได้ หรือไม่มีเลยก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **การไม่ประกาศหรือประกาศเพียงครั้งเดียวของ Element เดียวกัน**

```
<!ELEMENT element-name (child-name?)>
```

DTD example:

```
<!ELEMENT note (message?)>
```

เครื่องหมาย ? ถูกใช้ในความหมาย ไม่มีหรือมีเพียงหนึ่ง จึงหมายความว่า note element จะประกอบด้วย child element message หนึ่งค่า หรือไม่มีก็ได้

- **การประกาศ Content แบบให้เลือก**

  จะใช้เครื่องหมาย | แทนความหมายหรือ

DTD example:

```
<!ELEMENT note (to,from,header,message|body)>
```

ดังนั้นตัวอย่างนี้จึงหมายถึง note element จะต้องประกอบด้วย to element , from element , header element แล้วก็ message element หรือ body element อย่างใดอย่างหนึ่งเท่านั้น

- **การประกาศ Content แบบผสม**

  เป็นการผสมการประกาศหลาย ๆ อย่างเข้าด้วยกัน

DTD example:

```
<!ELEMENT note (#PCDATA|to|from|header|message)*>
```

ตัวอย่างนี้หมายถึง note element จะประกอบด้วย character data หรือ element to, from, header และ message จำนวนเท่าไรก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.6.3 การกำหนด Attribute ใน DTD

วิธีการประกาศ attribute จะเป็นดังนี้

- **การประกาศ Attribute**

attribute ต้องถูกประกาศตามไวยากรณ์ นี้

```
<!ATTLIST element-name attribute-name attribute-type default-value>
```

DTD example:

```
<!ATTLIST payment type CDATA "check">
```

XML example:

```
<payment type="check">
```

ชนิดของ attribute สามารถมีค่าได้ดังตารางที่ 3-1

ค่า	คำอธิบาย
CDATA	ค่าต้องเป็น character ที่จะไม่ถูก parse โดย parser
(en1|en2|..)	ค่าต้องเป็นอันใดอันหนึ่งในรายการนั้น
ID	ค่าต้องเป็น unique id
IDREF	ค่าต้องเป็น id ของ element อื่น
IDREFS	ค่าต้องเป็นรายการของ id อื่น โดยจะเว้นแต่ละรายการด้วยช่องว่าง
NMTOKEN	ค่าต้องเป็นชื่อที่ประกอบด้วยตัวอักษร , ตัวเลข , เครื่องหมายจุด (.) , เครื่องหมายขีดกลาง (-) , เครื่องหมายขีดล่าง (_) และสามารถมีเครื่องหมายโคลอน (:) ยกเว้นในตำแหน่งแรกได้
NMTOKENS	ค่าต้องเป็นรายการของ NMTOKEN โดยจะเว้นแต่ละรายการด้วยช่องว่าง
ENTITY	ค่าต้องเป็น entity
ENTITIES	ค่าต้องเป็นรายการของ entity
NOTATION	ค่าต้องเป็นชื่อของ notation (notation คือชนิดของข้อมูลภายนอกที่กำหนดขึ้นเอง เช่น GIF เพื่อบอกว่าเป็นไฟล์รูปภาพ)
XML:	ค่าต้องขึ้นต้นด้วย XML

ตารางที่ 3-1 ชนิดของ Attribute ใน XML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

default-value สามารถมีค่าได้ดังตารางที่ 3-2

ค่า	คำอธิบาย
#DEFAULT value	ค่าดีฟอลต์ของ attribute
#REQUIRED	element ต้องมี attribute
#IMPLIED	element จะมี attribute หรือไม่ก็ได้
#FIXED value	attribute ต้องเหมือนกับที่กำหนดไว้

ตารางที่ 3-2 ค่าดีฟอลต์ของ Attribute ใน XML

- ตัวอย่างของการประกาศ Attribute

DTD example:

```
<!ELEMENT square EMPTY>
<!ATTLIST square width CDATA "0">
```

XML example:

```
<square width="100"></square>
```

ตัวอย่างนี้หมายความว่า square element เป็น empty element กับ attribute width ซึ่งเป็นชนิด CDATA ถ้าไม่ระบุจะยึดค่า default เป็น 0

- ค่าของ Attribute แบบดีฟอลต์

```
<!ATTLIST element-name attribute-name attribute-type "default-value">
```

DTD example:

```
<!ATTLIST payment type CDATA "check">
```

XML example:

```
<payment type="check">
```

ถ้าในเอกสาร XML ไม่มีการระบุจะใช้ค่าดีฟอลต์ แต่ถ้าระบุก็จะใช้ตามที่ระบุไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Attribute โดยนัย**

```
<!ATTLIST element-name attribute-name attribute-type #IMPLIED>
```

DTD example:

<!ATTLIST contact fax CDATA #IMPLIED>

XML example:

```
<contact fax="555-667788">
```

จะใช้ #IMPLIED ในกรณีจะระบุ attribute หรือไม่ก็ได้ ถ้าไม่ระบุก็จะไม่มีค่า default ให้

- **Attribute ที่ต้องระบุเสมอ**

```
<!ATTLIST element-name attribute_name attribute-type #REQUIRED>
```

DTD example:

<!ATTLIST person number CDATA #REQUIRED>

XML example:

```
<person number="5677">
```

ใช้ในการบังคับให้ต้องระบุ attribute ถ้าในเอกสาร XML ไม่ระบุ attribute เอกสารนั้นจะไม่ถูกต้อง

- **ค่าของ Attribute คงที่**

```
<!ATTLIST element-name attribute-name attribute-type #FIXED "value">
```

DTD example:

<!ATTLIST sender company CDATA #FIXED "Microsoft">

XML example:

```
<sender company="Microsoft">
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในแบบนี้ถ้าไม่ได้ระบุค่า attribute จะใช้ค่าดีฟอลต์ แต่ถ้าจะระบุต้องระบุให้ตรงกับค่าดีฟอลต์มิฉะนั้นจะถือว่าผิด

- **ค่าของ Attribute กำหนดเอง**

```
<!ATTLIST element-name attribute-name (en1|en2|..) default-value>
```

DTD example:

```
<!ATTLIST payment type (check|cash) "cash">
```

XML example:

```
<payment type="check">
```

หรือ

```
<payment type="cash">
```

จะใช้เมื่อเราต้องการกำหนดค่าของ attribute เอง

3.1.6.4 การกำหนด Entity ใน DTD

- **การประกาศ Entity ภายใน**

```
<!ENTITY entity-name "entity-value">
```

DTD Example:

```
<!ENTITY writer "Donald Duck.">
<!ENTITY copyright "Copyright W3Schools.">
```

XML example:

```
<author>&writer;&copyright;</author>
```

- **การประกาศ Entity ภายนอก**

```
<!ENTITY entity-name SYSTEM "URI/URL">
```

DTD Example:

```
<!ENTITY writer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SYSTEM "http://www.w3schools.com/entities/entities.XML">
<!ENTITY copyright
SYSTEM "http://www.w3schools.com/entities/entities.dtd">
XML example:
<author>&writer;&copyright;</author>
```

3.1.7 XML Namespace

XML namespace จะหาวิธีหลีกเลี่ยงความสับสนที่เกิดจากชื่อ element (Name Conflicts)

- ความสับสนที่เกิดจากชื่อ

ความสับสนที่เกิดจากชื่อ element จะเกิดขึ้นบ่อยเมื่อเอกสารที่แตกต่างกัน ใช้ชื่อเดียวกันในการอธิบายชนิดของ element ที่แตกต่างกัน เช่น table ใน 2 ตัวอย่างต่อไปนี้

```
<table>
  <tr>
  <td>Apples</td>
  <td>Bananas</td>
  </tr>
</table>
```

และ

```
<table>
  <name>African Coffee Table</name>
  <width>80</width>
  <length>120</length>
</table>
```

ถ้านำเอกสาร XML ทั้ง 2 มารวมกัน จะเกิดความสับสนจากชื่อขึ้นเพราะทั้ง 2 มี element (table) ซึ่งมีใช้ในความหมายที่ต่างกัน

- **การแก้ปัญหาความสับสนที่เกิดจากชื่อโดยใช้ Prefix**

เอกสาร XML ทั้ง 2 คือ

```
<h:table>
  <h:tr>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  <h:td>Apples</h:td>
  <h:td>Bananas</h:td>
  </h:tr>
</h:table>
```

และ

```
<f:table>
  <f:name>African Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>
```

ความสับสนที่เกิดจากชื่อจะถูกกำจัดไปโดยการเพิ่ม prefix ทำให้ได้ element ที่ต่างกัน 2 ชนิด (<h:table> และ <f:table>)

- **การใช้ Namespaces**

  จากเอกสาร XML ทั้ง2 คือ

```
<h:table xmlns:h="http://www.w3.org/TR/html4/">
  <h:tr>
  <h:td>Apples</h:td>
  <h:td>Bananas</h:td>
  </h:tr>
</h:table>
```

และ

```
<f:table xmlns:f="http://www.w3schools.com/furniture">
  <f:name>African Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>
```

เราสามารถเพิ่ม attribute xmlns ลงในแท็ก เพื่อบอกชื่อที่ระบุไว้ ที่เกี่ยวข้องกับ namespace

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Attribute Namespace**

attribute namespace จะวางในแท็กเริ่มต้น (start tag) ตามไวยากรณ์ ดังนี้

```
Xmlns:namespace-prefix="namespace"
```

จากตัวอย่างข้างบน namespace จะสามารถกำหนดด้วย Internet address

```
Xmlns:f="http://www.w3schools.com/furniture">
```

จาก namespace specification ของ W3C กล่าวไว้ว่า namespace สามารถเป็น Uniform Resource Identifier (URI)

ถ้า namespace ถูกกำหนดในแท็กเริ่มต้นแล้ว child element ทุกตัวที่มี prefix เหมือนกันจะต้อง มีความสัมพันธ์กับ namespace เดียวกันด้วย

และ address ที่ใช้บอก namespace ใช้เพื่อให้ชื่อมีความเป็นเอกลักษณ์ไม่ได้ใช้เป็น address เพื่ออ้างอิงหาข้อมูล แต่หลายบริษัทใช้ namespace เพื่อเป็นตัวชี้ไปยัง web page ที่มีอยู่จริงซึ่งเก็บข้อมูลเกี่ยวกับ namespaceไว้

- **Default Namespaces**

การกำหนด default namespace จะป้องกันการใช้ prefix ใน child element ทุกตัว มีไวยากรณ์ดังนี้

```
<element xmlns="namespace">
```

ซึ่งจากตัวอย่างที่ผ่านมาจะได้ดังนี้

```
<table xmlns="http://www.w3.org/TR/html4/">
  <tr>
  <td>Apples<td>
  <td>Bananas<td>
  <tr>
</table>
```

และ

```
<table xmlns="http://www.w3schools.com/furniture">
  <name>African Coffee Table</name>
  <width>80</width>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  <length>120</length>
</table>
```

3.1.8 XML PCDATA and CDATA

Parsable Character Data (PCDATA) คือข้อมูลแบบเท็กซ์ที่จะถูก parse โดย parser

Character Data (CDATA) คือข้อมูลแบบเท็กซ์ที่ไม่ถูก parse โดย parser

3.1.8.1 PCDATA

- **XML Parser จะมองทุกเท็กซ์เป็น Parsable Characters (PCDATA)**

  เมื่อ XML ถูก parse เท็กซ์ที่อยู่ระหว่าง XML แท็กจะถูก parse ไปด้วย

```
<message>This text is also parsed</message>
```

parser ทำเช่นนี้เนื่องจาก XML element สามารถมีหลาย element อยู่ภายใน ดังในตัวอย่าง

```
<name><first>Bill</first><last>Gates</last></name>
```

และ parser จะแบ่งเป็น sub-element ดังนี้

```
<name>
  <first>Bill</first>
  <last>Gates</last>
</name>
```

- **Escape Character**

  XML character ที่ไม่สามารถใช้ได้จะถูกแทนที่ด้วย entity reference

  ถ้าใส่ตัวอักษร "<" ใน XML element จะทำให้เกิดความผิดพลาด เพราะ parser จะแปลเป็นจุดเริ่มต้นของ element ใหม่ ดังนั้นจึงไม่สามารถเขียน อย่างนี้ได้

```
<message>if salary < 1000 then</message>
```

การหลีกเลี่ยง สามารถทำได้โดย แทนที่ "<" ด้วย entity reference ดังนี้

```
<message>if salary &lt; 1000 then</message>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

entity reference มาตรฐานที่ได้กำหนดมาให้ก่อนแล้วใน XML มีทั้งหมด 5 ตัว ดังตารางที่ 3-3

Entity Reference	ค่า	ความหมาย
<	<	less than
>	>	greater than
&	&	ampersand
'	'	apostrophe
"	"	quotation mark

ตารางที่ 3-3 Entity Reference มาตรฐาน

entity reference จะขึ้นต้นด้วย & และลงท้ายด้วย ;

เฉพาะ "<" และ "&" เท่านั้นที่ไม่สามารถใช้ได้ใน XML ส่วนตัวที่เหลือควรทำให้เป็นนิสัย

3.1.8.2 CDATA

parser จะไม่ parse ทุกอย่างที่อยู่ในส่วนของ CDATA

ถ้าเท็กซ์มี "<" หรือ "&" อยู่ภายใน XML element นั้นจะถูกกำหนดเป็นส่วนของ CDATA ส่วนของ CDATA จะเริ่มต้นด้วย "<![CDATA[" และลงท้ายด้วย "]]>"

```
<script>
<![CDATA[
function matchwo(a,b)
{
if (a < b && a < 0) then
  {
  return 1
  }
else
  {
  return 0
  }
}
]]>
</script>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่างข้างต้น ทุกอย่างที่อยู่ในส่วนของ CDATA จะถูกเพิกเฉยโดย parser

3.1.9 XML Parser

XML parser คือโปรแกรมที่ใช้ในการอ่าน สร้าง และจัดการกับเอกสาร XML รวมถึงตรวจความถูกต้องของเอกสาร XML parser สามารถเขียนได้ด้วยภาษาใด ๆ ก็ได้ ในบราวเซอร์ IE 5.0 ขึ้นไปก็มี Microsoft XML parser ทำให้ IE สามารถแสดงผลเอกสาร XML ได้ XML parser ของภาษาจาวา อย่างเช่น XML parser ของซัน (http://java.sun.com/xml) หากเราต้องการสร้างแอพพลิเคชันที่สามารถจัดการกับ XML เราก็ต้องมี parser ของภาษานั้นและเขียนโค้ดติดต่อกับ API ของมัน

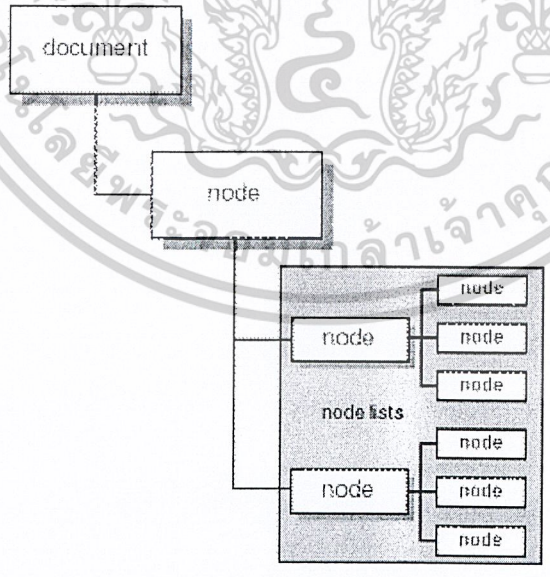
XML parser มี 2 ชนิดคือ

- DOM (Document Object Model)
- SAX (Simple API for XML)

3.1.9.1 DOM

DOM จะใช้วิธี parse เอกสาร XML เป็นลำดับชั้น (hierachy) ของออบเจ็กต์ดังนี้ (ดังรูปที่ 3-1)

1) Document Object – แหล่งข้อมูล XML
2) Node Object – Parent node ของ child node
3) Node List Object – list ของ sibling node
4) Parse Error – ใช้ในการรับค่าผิดพลาดที่เกิดจากการ parse



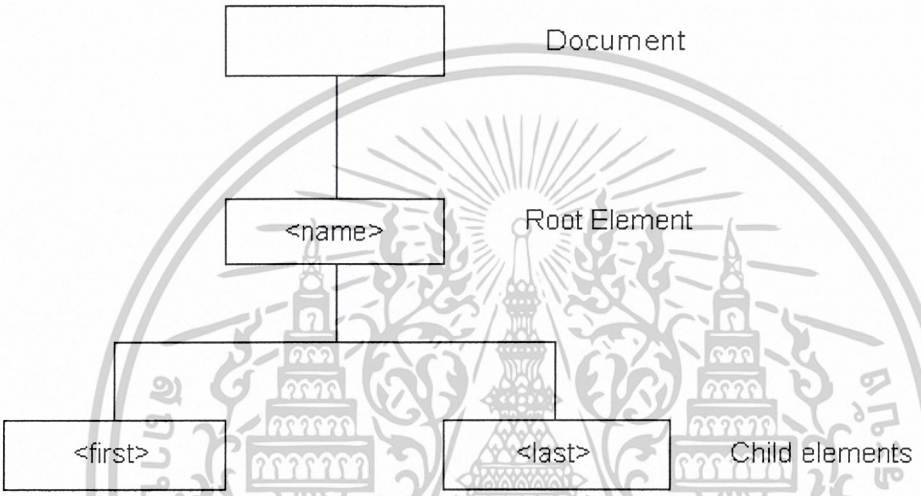
รูปที่ 3-1 โครงสร้างของ DOM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากเอกสาร XML เป็นดังนี้

```
<name>
   <first>Jim(/first>
   <last>Jones</last>
</name>
```

จะได้ความสัมพันธ์ดังรูปที่ 3-2



รูปที่ 3-2 ความสัมพันธ์ของการ parse ด้วย DOM

3.1.9.2 SAX

SAX (Simple API for XML) วิธีนี้จะเป็นแบบ event-driven โดย event จะประกอบด้วย

1) Start Element – จะตอบสนองกับ event นี้เมื่อพบจุดเริ่มต้นของแท็ก XML
2) End element – จะตอบสนองกับ event นี้เมื่อพบจุดสิ้นสุดของแท็ก
3) Character – จะตอบสนองกับ event นี้เมื่อพบกับตัวอักษร
4) Start Document – จะตอบสนองกับ event นี้ในตอนเริ่ม parse
5) End Document - จะตอบสนองกับ event นี้ในเมื่อจบการ parse

ดังนั้นจากโค้ดข้างบนจะเกิด event ดังนี้

```
                                  →  Start Document
<name>                            →  Start Element name
      <first>                     →  Start Element first
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Jim	→	Character Jim
</first>	→	End Element first
<last>	□	Start Element last
Jones	□	Character Jones
</last>	□	End Element last
</name>	□	Start Element name
	□	End Document

ทั้ง 2 วิธีมีข้อดีข้อเสียต่างกันคือ DOM จะใช้เมโมรีและมีการประมวลผลมาก แต่สามารถจัดการได้ง่าย สามารถที่จะแก้ไขค่าได้ ไม่เหมาะกับเอกสาร XML ที่มีขนาดใหญ่ ถูกนำไปใช้ในบราวเซอร์ ในขณะที่ SAX ไม่ต้อง parse เอกสาร XML ทั้งหมดจึงใช้โมเมรีและมีการประมวลผลน้อยกว่า เหมาะกับการสร้างและรับค่าจากเอกสาร จะถูกใช้ในการประมวลผลทางฝั่งเซิร์ฟเวอร์ ตัวอย่าง XML parser ดังตารางที่ 3-4

ชื่อ	แพล็ตฟอร์ม	คำบรรยาย
ActiveDOM	Microsoft Window	http://www.vivid-creations.com
ActiveSAX	Microsoft Window	http://www.vivid-creations.com
JavaTM API for XML	Java	http://java.sun.com/xml
MS XML	Microsoft Window	http://www.microsoft.com
Oracle XML Developer's Kit	Java , C++ , PL/SQL	http://technet.oracle.com/tech/xml
Xerces	Java , C++	http://xml.apache.org
XP	Java	http://www.jclark.com/xml

ตารางที่ 3-4 XML Parser

3.1.10 ประโยชน์จาก XML

สำหรับประโยชน์ของ XML นั้น เป็นด้านความยืดหยุ่นในการใช้งานสำหรับแอพพลิเคชันที่อิงกับ Web Base ที่ใช้ง่ายในการค้นหาข้อมูล มีความยืดหยุ่นในการพัฒนาเว็บ สามารถผสมผสานข้อมูลจากหลายแหล่ง จากแอพพลิเคชันที่ต่างกัน สามารถแสดงข้อมูลแบบต่างๆ และสามารถ update ข้อมูลให้ทันสมัยเสมอ และคาดว่าจะเป็นมาตรฐานใหม่ของระบบเปิด ซึ่งนับเป็น format ใหม่สำหรับการส่งข้อมูลบนเว็บที่มากด้วยข้อมูลหลายแบบ แต่ส่งผ่านด้วยเทคโนโลยีที่บีบอัดข้อมูล ที่ให้ความเร็วได้รับการสนับสนุนจากผลิตภัณฑ์ค่ายไมโครซอฟท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถาปัตยกรรม XML

ภาษา XML ได้รับการสนับสนุนจาก W3C ให้นักพัฒนาเว็บแอพพลิเคชันได้หันมาใช้เป็นส่วนประกอบของการพัฒนาเว็บไซต์ เพราะ XML มีประสิทธิภาพและมีความน่าเชื่อสูงในการแปลงข้อมูลและโครงสร้างข้อมูลให้สามารถนำไปใช้งาน สถาปัตยกรรม 3 Tier ที่ XML สามารถสร้างขึ้นจากระบบข้อมูลที่ใช้โมเดลของ 3-Tier โครงสร้าง ของข้อมูลต่างๆ สามารถนำมาแสดงตามข้อกำหนด หรือรูปแบบที่ต้องการตามการใช้งานได้

XML เป็นการทำงานในระดับกลาง middle tier ที่สามารถเรียกใช้ฐานข้อมูลได้หลากหลายระบบ ฐานข้อมูลและโอนข้อมูลให้อยู่ในรูปแบบของ XML และมีการให้รายละเอียดเกี่ยวกับตัวข้อมูล โครงสร้างต่างๆ ของระบบฐานข้อมูลได้ XML เป็นระบบเปิดที่นำเสนอข้อมูลในรูปแบบ text ผ่านทาง HTTP เหมือนกับ HTML แต่จะมีคุณสมบัติในการให้ข้อมูลแบบ real time อัพเดทหรือเปลี่ยนแปลงได้ตามข้อกำหนด การแสดงข้อมูลจาก XML ใน HTML จะเป็นการเพิ่มในส่วนของรายละเอียดข้อมูล ที่มีการเรียกใช้จากแหล่งหรือฐานข้อมูลที่เชื่อมโยงกันในหลายแหล่งเพื่อให้HTMLมีความสมบูรณ์มากขึ้น ในอนาคตการพัฒนาเว็บหรือการเขียน และสร้าง HTML ไม่จำเป็นต้องมีการเขียนชุดคำสั่งที่ยุ่งยากซับซ้อนมากก็สามารถทำงานร่วมกับระบบข้อมูลได้อย่างมีประสิทธิภาพ XML จะทำการกำหนดค่าสำหรับโครงสร้างข้อมูลที่จะนำไปแสดงใน HTML นอกจากนั้นยังสามารถนำไปสนับสนุนระบบการแลกเปลี่ยนข้อมูลข่าวสารทาง Electronic ได้อย่างดีอีกด้วย

3.1.11 บทสรุปของ XML

XML มีความยืดหยุ่นพอในการนำเสนอข้อมูลแบบต่างๆ ที่สามารถให้รายละเอียดโครงสร้างข้อมูลตามระดับและความต้องการในการนำไปใช้งาน

3.2 SOAP

3.2.1 แนะนำ SOAP

SOAP (Simple Object Access Protocol) เป็นโพรโตคอลที่ใช้ XML เป็นพื้นฐาน เพื่อให้ซอฟต์แวร์คอมโพเนนต์ และแอพพลิเคชันสามารถติดต่อกันผ่าน HTTP ซึ่งเป็นมาตรฐานอินเตอร์เน็ตโพรโตคอลได้

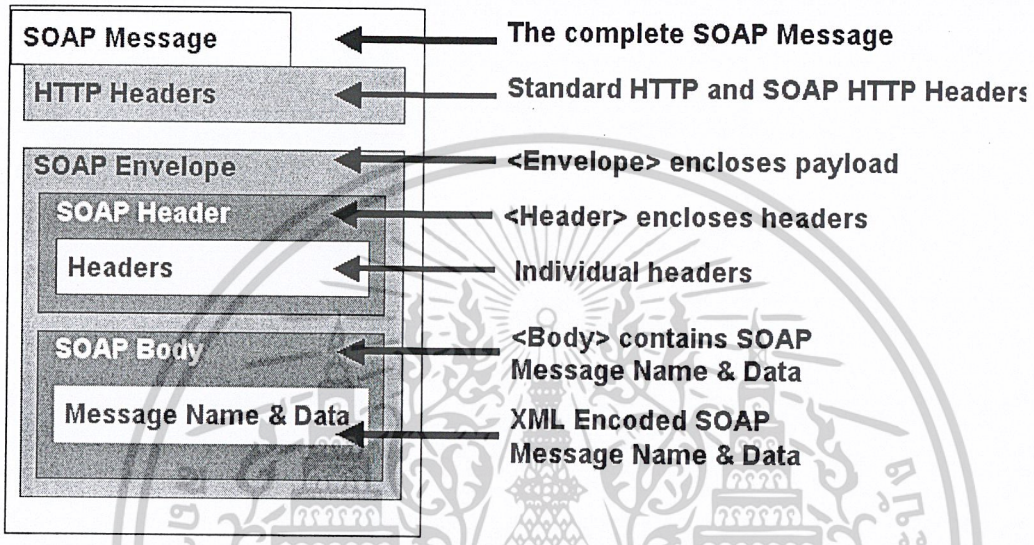
เนื่องจากการสื่อสารระหว่างแอพพลิเคชันจำเป็นต่อการพัฒนาแอพพลิเคชันบนอินเตอร์เน็ต เป็นอย่างมาก แต่แอพพลิเคชันแบบกระจายกันทำงาน (distributed application) ในปัจจุบันใช้ Remote Procedure Call (RPC) สื่อสารกันระหว่างออบเจ็กต์ เช่น DCOM และ CORBA ซึ่งไม่ได้ใช้พอร์ตเดียวกับ HTTP ที่เป็นพอร์ตมาตรฐานสำหรับให้บริการเว็บ ดังนั้น RPC จึงนำมาปรับใช้กับอินเตอร์เน็ตได้ยาก และมีปัญหาทางด้านความปลอดภัย ไฟร์วอลล์และพร็อกซีเซิร์ฟเวอร์จะไม่ยอมให้ส่งข้อมูลชนิดนี้ได้ตามปกติ วิธีที่ดีกว่าคือใช้ HTTP เพราะเป็นที่ยอมรับโดยอินเตอร์เน็ตบราวเซอร์ และเซิร์ฟเวอร์ทุกชนิด ซึ่ง SOAP ถูกสร้างมาเพื่อใช้ในกรณีนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีของ SOAP ก็คือ SOAP ไม่ขึ้นกับคอมโพเนนต์เทคโนโลยี และภาษาการเขียนโปรแกรมใดๆ สามารถเขียนได้ง่ายและขยายเพิ่มเติมได้

3.2.2 ส่วนประกอบของ SOAP

SOAP เมสเซจใช้ไวยากรณ์ของ XML ในการสร้าง ประกอบด้วย 3 อีลีเมนต์มาตรฐาน คือ SOAP Envelope, SOAP Header และ SOAP Body



รูปที่ 3-3 โครงสร้างของ SOAP

SOAP เมสเซจ จะต้องเป็นไปตามกฎนี้

- Envelope เป็นอีลีเมนต์ที่อยู่บนสุด ต้องมีอีลีเมนต์นี้เสมอ
- Header อาจจะมีหรือไม่มีก็ได้ แต่ถ้ามีต้องเป็น child element แรกของ Envelope
- Body ต้องเป็น child element แรกของ envelope หรือ header

ตัวอย่างของ SOAP เมสเซจ ตัวอย่างนี้คือ GetLastTradePrice SOAP Request ที่ส่งไปยัง StockQuote service ประกอบด้วยสตริงพารามิเตอร์ symbol และคืนค่า float กลับมากับ SOAP response SOAP envelope element อยู่ที่จุดบนสุดของเอกสาร XML XML เนมสเปซ ใช้เพื่อป้องกันการสับสนของ SOAP identifier

SOAP request จะเป็นดังนี้

```
POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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/"
 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>
```

SOAP response จะเป็นดังนี้

```
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-ENV: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>
```

นอกจากนั้นใน SOAP เมสเซจ จะมีการใช้ XML เนมสเปซ ทุกๆ อีลีเมนต์ในเอกสารจะขึ้นต้นด้วย เนมสเปซ เนมสเปซ จะถูกกำหนดโดยใช้ xmlns attribute ดังนี้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

แต่หากไม่ต้องการให้ขึ้นต้นด้วย เนมสเปซ ก็สามารถทำได้ ดังนี้

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

3.2.2.1 Envelope

envelope element เป็นอีลีเมนต์บนสุดของเอกสาร XML ที่ใช้แสดงเมสเซจ อาจประกอบด้วยแอตทริบิวต์ คือ envelope เนมสเปซ และ encodingStyle

1. Envelope เนมสเปซ – SOAP เมสเซจจะแสดงเวอร์ชันโดยใช้ เนมสเปซ ของ envelope element เนมสเปซ จะถูกนำไปใช้ขึ้นต้น envelope, header และ body element
2. EncodingStyle Attribute – ใช้แสดงวิธี serialization ของ SOAP เมสเซจ แอตทริบิวต์นี้สามารถมีได้ในทุกอีลีเมนต์และจะมีผลกับ content ของอีลีเมนต์นั้นและ child element ทั้งหมดของมันที่ไม่ได้ประกาศแอตทริบิวต์

3.2.2.2 Body

เป็นส่วนของข้อมูลที่ใช้ในการแลกเปลี่ยน โดยทั่ว ๆ ไปข้อมูลจะเป็นการ marshall RPC call และ error report child element ของ body element จะถูกเรียกว่า Body entry

body entry จะต้องเป็นไปตามกฎนี้

- body entry จะต้องแสดงด้วยชื่อเต็มที่ประกอบด้วย เนมสเปซ URI และ ชื่อของมัน (local name)
- SOAP encodingStyle attribute อาจจะมีได้ เพื่อแสดงถึงวิธี encode ของ body entry นั้น

3.2.2.3 Header

เป็นส่วนเพิ่มเติมของเมสเซจสามารถประกอบด้วยอินฟอร์เมชันที่ระบุไปยังแอพพลิเคชัน ส่วนเพิ่มเติมนี้อาจนำไปใช้อิมพลีเม็นต์เป็น authentication, transaction management เป็นต้น ทุก ๆ child element ของ header element จะถูกเรียกว่า Header entry

header entry จะต้องเป็นไปตามกฎดังนี้

- header entry จะต้องแสดงด้วยชื่อเต็มที่ประกอบด้วย เนมสเปซ URI และ ชื่อของมัน (local name)
- อาจมี SOAP encodingStyle attribute ที่ใช้สำหรับ header entry
- SOAP mustUnderStand attribute และ SOAP actor attribute จะมีหรือไม่ก็ได้ เพื่อใช้บอกว่าจะจัดการกับ entry นั้นอย่างไรและโดยใคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<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>
```

1) SOAP Actor Attribute – SOAP เมสเซจสามารถถูกส่งไปยังปลายทางผ่านตัวกลางของ SOAP (SOAP intermediary) ตัวกลางของ SOAP คือแอพพลิเคชันที่มีความสามารถทั้งรับและส่งต่อ SOAP เมสเซจ ในการส่งต่อตัวกลางจะต้องไม่ส่งต่อ header element ไปให้กับแอพพลิเคชัน โดยตัวกลางนี้จะถูกกำหนดเป็น URI ถ้าไม่มี attribute นี้หมายถึงผู้รับ SOAP เมสเซจเป็นปลายทางสุดท้าย
2) SOAP MustUnderStand Attribute – ใช้แสดงว่า header entry นี้จำเป็นหรือเป็นเพียงอ็อปชัน สำหรับผู้รับที่จะนำไปประมวลผล ค่าของมันคือ “1” หรือ “0” ถ้าไม่มีการระบุจะมีค่าเท่ากับ “0” โดย “0” หมายถึงเป็นอ็อปชัน และ “1” หมายถึงจำเป็น โดยจะต้องประมวลผลให้ถูกต้องตาม semantic หรือต้อง fail เช่น เมสเซจเป็นส่วนหนึ่งของ transaction ถ้าปลายทางไม่รองรับ transaction จะต้องไม่ประมวลผลและส่งผลผิดพลาดกลับไปในกรณีที่ mustUnderStand attribute เป็น “1” แต่ถ้าเป็น “0” จะยังคงประมวลผลต่อไป

3.2.3 SOAP Fault Element

ข้อความแสดงความผิดพลาดจากแอพพลิเคชันของ SOAP จะเก็บอยู่ใน fault element ซึ่งถ้ามีจะต้องปรากฏใน body element เพียงครั้งเดียวใน SOAP เมสเซจ 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/">
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            <module>endpoint.asp</module>
            <line>203</line>
          </w:source>
        </detail>
      </soap:Fault>
    </soap:Body>
</soap:Envelope>
```

SOAP fault element มี element ย่อย ๆ ดังตารางที่ 3-5

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

ตารางที่ 3-5 SOAP Fault Element

ค่าของ fault code สามารถมีค่าได้ดังตารางที่ 3-6

Error	Description
VersionMismatch	เนมสเปซ ภายใน SOAP Envelope element ไม่ถูกต้อง
MustUnderstand	child element ของ Header element กับ mustUnderstand attribute ที่มีค่า "1" ผู้รับไม่รองรับ
Client	เมสเซจมีรูปแบบไม่ถูกต้อง หรือมีอินฟอร์เมชันที่ไม่ถูกต้อง
Server	เกิดปัญหากับเซิร์ฟเวอร์ ไม่สามารถประมวลผลได้

ตารางที่ 3-6 ค่า fault code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 SOAP Encoding

SOAP encoding มีวิธีการ map จาก type ของโปรแกรมมิ่งไปเป็น XML 2 วิธี คือจากภายนอก โดยใช้ WSDL (Web Services Description Language) ที่บอกถึง type ของข้อมูลที่รับหรือส่ง หรือใช้ xsi:type attribute ในกรณีที่ภาษาที่ใช้ไม่รองรับ WSDL โดยทั้ง 2 วิธีจะใช้ XML schema ในการระบุ type ตัวอย่างของการใช้ xsi จะเป็นดังนี้

```
<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>
    <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>
```

SOAP รองรับ type ของข้อมูลได้ทั้ง simple type และ compound type เช่น struct และ array

3.2.4.1 ตัวอย่าง Struct

struct book ที่เขียนด้วยภาษา c++ จะเป็นดังนี้

```
struct Book
{
   string author;
   string name;
   int page;
};
```

Soap เมสเซจที่ไม่ใช้ WSDL แต่ใช้ xsi จะเป็นดังนี้

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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/encoding/">
<bookResult xmlns:ns2="http://soapinterop.org/xsd" xsi:type="ns2:Book">
 <author xsi:type="xsd:string">Ed Roman</author>
 <name xsi:type="xsd:string">Mastering EJB</intro>
 <page xsi:type="xsd:int">500</page>
 </bookResult>
 </ns1:searchBook>
</SOAP-ENV:Body>
 </SOAP-ENV:Envelope>
```

แต่ถ้าใช้ WSDL ในส่วนของการประกาศ schema จะต้องประกอบด้วย schema ดังนี้

```
<types>
  <schema targetเนมสเปซ='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>
  </schema>
</types>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4.2 ตัวอย่าง Array

ตัวอย่างนี้เป็น array ของ struct book ในตัวอย่างก่อน Soap เมสเซจที่ไม่ใช้ WSDL แต่ใช้ xsi 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: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>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ถ้าใช้ WSDL ในส่วนของการประกาศ schema จะต้องประกอบด้วย schema ดังนี้

```
<types>
  <schema targetเนมสเปซ='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:arrayType' wsdl:arrayType='typens:Book[]'/>
      </restriction>
    </complexContent>
  </complexType>
  </schema>
</types>
```

3.2.5 SOAP ใน HTTP

ในการส่ง SOAP ผ่านทาง HTTP จะต้องใช้ content-type เป็น text/xml แต่ใน SOAP request จะต้องมี header SOAPAction ภายใน HTTP header SOAPAction จะเป็นตัวบอกให้เซิร์ฟเวอร์รู้ว่า HTTP Post นั้นเป็น SOAP เมสเซจ และค่าของ header คือ URI ที่แสดงถึงจุดหมายของ SOAP เมสเซจ ส่วน SOAP response จะต้องมี status code ตามมาตรฐานของ HTTP โดย 200-299 แสดงว่าสำเร็จ แต่ถ้า response เมสเซจ เป็นการ fault แล้ว status code จะต้องเป็น 500 ซึ่งแสดงถึง internal server error ตัวอย่างของ response ที่มี status code เป็น 500 อาจเป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
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 เนมสเปซ is incorrect.</faultstring>
      <faultactor>http://www.wrox.com/endpoint.asp</faultactor>
      <detail>
        <w:errorinfo xmlns:w="http://www.wrox.com/">
          <desc>The SOAP เนมสเปซ was blank.</desc>
        </w:errorinfo>
      </detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

3.2.6 SOAP สำหรับ RPC

จุดประสงค์ของการออกแบบ SOAP คือทำ RPC โดยใช้ XML การเรียก RPC นั้นจะ map เข้ากับ HTTP request และ RPC response จะ map กับ HTTP response

ในการทำ method call จำเป็นที่จะต้องมีอินฟอร์เมชันดังนี้

1. URI ของออบเจ็กต์ปลายทาง (target object)
2. ชื่อของเมธอด
3. คำอธิบายของเมธอด (method signature) ส่วนนี้เป็นอ็อปชัน
4. พารามิเตอร์สำหรับเมธอด
5. ข้อมูลส่วนหัว (header data) เป็นอ็อปชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณาเมธอดดังนี้

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

ถ้า เนมสเปซ ของเมธอดคือ “http://www.worxstox.com/” แล้วการเรียกเมธอดโดย request stock quote ใช้ symbol OU812 จะเป็นดังนี้

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

ชื่อเมธอดและชื่อของอีลีเมนต์จะต้อง match กันเช่นเดียวกับพารามิเตอร์

สำหรับ response จะต้องชื่ออีลีเมนต์จะต้องเป็นชื่อเมธอดตามด้วย response ดังนี้

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

ชื่ออีลีเมนต์ของค่าที่คืนมา สามารถเปลี่ยนได้โดยจะกำหนดได้จาก โปรแกรมที่เขียนหรือจาก WSDL

3.2.7 SOAP Toolkit

SOAP toolkit คือเครื่องมือที่จะทำหน้าที่ในการประมวลผล SOAP request และส่ง response กลับไปให้ไคลเอนต์ โดย SOAP toolkit แต่ละตัวใช้ได้บนแพลตฟอร์มที่ต่างกัน รองรับเทคโนโลยี เว็บเซอร์วิส ต่างกัน บางตัวอาจจะรองรับ WSDL หรือ UDDI ในขณะที่บางตัวรองรับอย่างใดอย่างหนึ่งหรือไม่รองรับทั้งสองอย่าง และถึงแม้ว่า SOAP จะเป็นอิสระต่อแพลตฟอร์ม แต่ใน SOAP toolkit บางตัวยังไม่สามารถทำงานข้ามผลิตภัณฑ์ (interoperability) ได้ เนื่องจากรองรับเทคโนโลยีของ เว็บเซอร์วิส ไม่เหมือนกัน เช่น xsi ในบางผลิตภัณฑ์จะบังคับให้ SOAP เมสเซจจะต้องระบุ type ด้วย xsi หากไม่มีก็จะ falut ซึ่งในปัจจุบันแต่ละผลิตภัณฑ์ก็พยายามพัฒนาให้สามารถทำงานด้วยกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 Services 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
PocketSOAP	COM	http://www.pocketsoap.com
SOAP::Lite	Perl	http://www.soaplite.com
Visual Studio .NET	COM	http://www.microsoft.com

ตารางที่ 3-7 SOAP Toolkit

3.2.8 โครงสร้างภายในของ SOAP Toolkit ของ Microsoft

The Web Services Description Language (WSDL) เป็นรูปแบบหนึ่งของ XML เพื่อใช้อธิบายว่าเซิร์ฟเวอร์นั้นมีเซอร์วิส (Services) และ operation ภายในเซอร์วิสอะไรให้ใช้บ้าง ซึ่ง operation ภายในเซอร์วิสนั้นจะอธิบายรูปแบบที่ไคลเอนต์ (client) ต้องใช้ในการเรียก operation

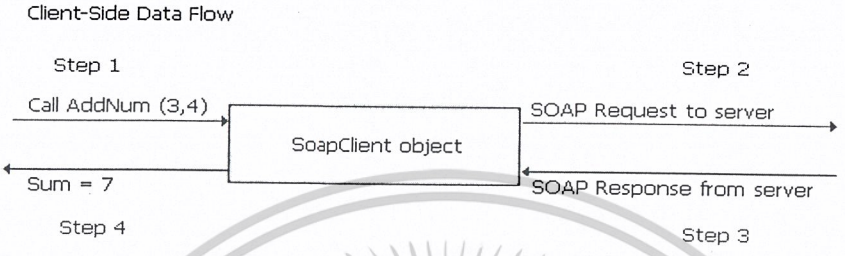
The Web Services Meta Language (WSML) file เป็นส่วนที่เพิ่มขึ้นมาจากไฟล์ WSDL เพื่อใช้ในการเชื่อมโยงแต่ละ operation ในเซอร์วิสกับ COM object โดย WSML file จะพิจารณาว่า COM object ตัวใดที่จะต้องถูกโหลด (load) ขึ้นมาเมื่อมีการเรียกใช้ operation ในเซอร์วิส

ทั้งไฟล์ WSDL และ WSML จะต้องอยู่ที่เซิร์ฟเวอร์ โดยไคลเอนต์ที่ต้องการที่จะส่ง SOAP request ไปที่เซิร์ฟเวอร์ จะต้องได้รับ WSDL จากเซิร์ฟเวอร์ก่อน เพื่อหารู้รูปแบบของ SOAP request ในการติดต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.8.1 การสร้าง SoapClient Object

ทางฝั่งไคลเอนต์ เมื่อแอพพลิเคชั่นส่ง message ไปที่ Soap Client object เพื่อเรียกใช้ operation หนึ่งๆ นั้น(ตามรูปนี้คือการบวกเลข 2 ตัว) SoapClient object จะทำการสร้าง SOAP request ส่งไปที่เซิร์ฟเวอร์ เมื่อเซิร์ฟเวอร์ทำการประมวลผลเสร็จเรียบร้อยแล้ว ก็จะส่งผลลัพธ์ในรูปแบบของ SOAP response กลับมาให้ไคลเอนต์ หลังจากนั้น SoapClient object จึงจะทำการแปลและส่ง message ที่เก็บผลลัพธ์ของ operation นั้นกลับไปให้แอพพลิเคชั่น ตามไดอะแกรมด้านล่าง



รูปที่ 3-4 การ Client –Side Data Flow

แอพพลิเคชั่นต้อง instantiate Soap object ก่อนที่จะทำการเรียกใช้ operation ใดๆ (SoapClient object นี้ถูกสร้างเก็บไว้ในไฟล์ MSSOAP1.dll) โดยใช้เมธอด SoapClient.mssoapinit (ชื่อไฟล์ WSDL, ชื่อเซอร์วิส, ข้อมูลพอร์ต) โดยข้อมูลพอร์ตจะใช้ในการแบ่งแยกช่องทางที่ใช้ในการส่ง SOAP message

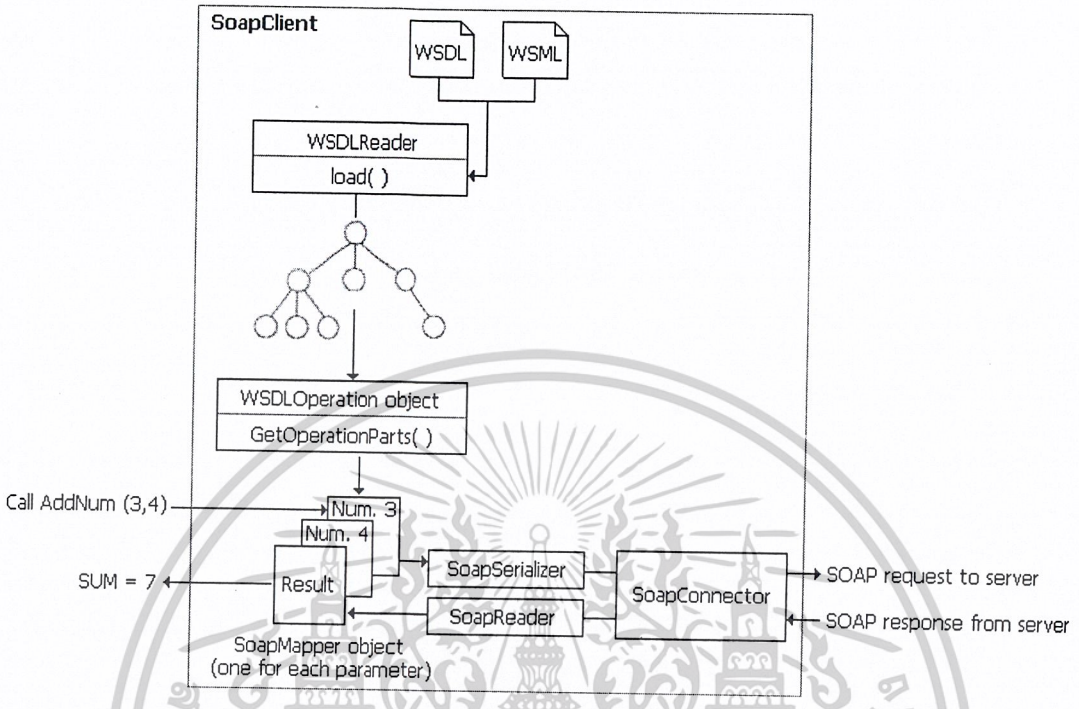
ในระหว่าง initialize SoapClient object จะตรวจสอบเมธอดทั้งหมดที่อยู่ในไฟล์ WSDL ด้วยการทำงานแบบไดนามิคนี้ทำให้เราสามารถที่จะเรียกใช้เมธอดใดก็ได้ที่ได้อธิบายไว้ในไฟล์ WSDL ดังแสดงตาม Microsoft Visual Basic Scripting Edition (VBScript) code ด้านล่างนี้

```
SET soapclient = CreateObject("MSSOAP.SoapClient")
Call soapclient.mssoapinit("http://localhost/MyWSDL.wsdl", "TestService", "TestPort")
' Now the client can call an operation listed in the portType element
' specified when calling mssoapinit().
wscript.echo soapclient.AddNumbers(2, 3)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.8.2 การประมวลผลภายใน SoapClient Object

การทำงานภายใน SoapClient object แสดงได้ดังรูปที่ 3-5



รูปที่3-5 โครงสร้างภายในของ SOAP Client

WSDL Reader object จะทำการ โหลดไฟล์ WSDL และไฟล์ WSML เข้าไปไว้ใน DOM แล้วทำการวิเคราะห์ โดย WSDLReader object จะสร้าง WSDLOperation object เพื่อทำการเรียกใช้งาน operation (ตามรูปนี้คือ AddNum(3, 4)) หลังจากนั้น WSDLOperation จะเรียกเมธอด GetOperationParts เพื่อดึงรูปแบบ message ของอินพุทและเอาท์พุทของ operation ออกมา แล้ว SoapClient จะสร้าง Soap Mapper objects ให้กับแต่ละส่วนที่ได้จากเมธอด GetOperationParts และโหลดค่าที่ได้มาจากการเรียกใช้ operation เข้าไว้ใน objects เหล่านี้ พอถึงขั้นตอนนี้ SoapSerializer objects จะสร้าง SOAP request message จาก SoapMapper objects ที่เหมาะสมและส่งไปที่เซิร์ฟเวอร์

เมื่อเซิร์ฟเวอร์ประมวลผลเสร็จแล้วก็จะส่ง SOAP response กลับไปที่ไคลเอนต์ SoapClient object จะทำการโหลดผลลัพธ์ของ operation ไปเก็บใน SoapMapper object ที่เหมาะสมแล้วส่งผลลัพธ์กลับไปให้แอพพลิเคชั่นของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.8.3 การสร้าง SoapServer Object

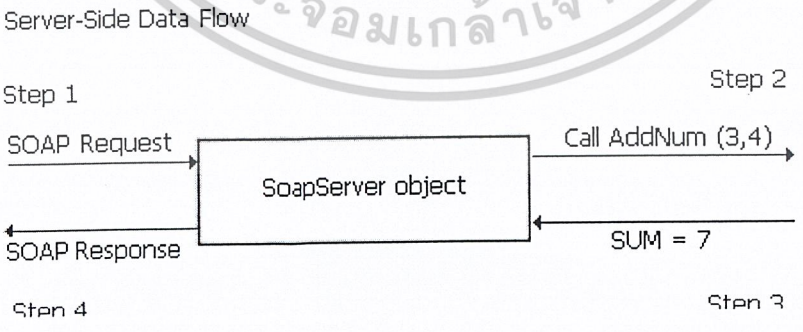
ส่วนทางฝั่งเซิร์ฟเวอร์ เราสามารถกำหนดรูปแบบการรับ SOAP request (SOAP listener) ได้ 2 รูปแบบ คือ Internet Server API (ISAPI) server และActive Server Pages (ASP) server

ในไฟล์ WSDL จะมี URL ที่ระบุว่าจะใช้ไฟล์ใดบนเซิร์ฟเวอร์ในการรับ SOAP request เพื่อใช้ในการพิจารณาว่าจะรับ SOAP request อย่างไรบนเซิร์ฟเวอร์ ตามโค้ดด้านล่างนี้ ตัวพิมพ์หนาจะเป็นส่วนที่ใช้ในการเรียกไปที่ ISAPI listener

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

การทำงานทางฝั่งเซิร์ฟเวอร์

ไม่ว่า SOAP request จะเรียกไปยัง ISAPI หรือ ASP listener ก็ตาม ข้อมูลที่รับส่งก็ยังเหมือนกัน เมื่อ SoapServer object ได้รับ SOAP request จากไคลเอนต์ ก็จะทำการแปลและเรียกไปยังเมธอด COM ที่ตรงกับ operation ที่ต้องการ (ตามไดอะแกรมนี้คือเมธอด AddNum) หลังจากนั้นเมธอดที่ถูกเรียกใช้งานก็จะส่งผลลัพธ์กลับมาให้ SoapServer object ซึ่งมันก็จะแปลงผลลัพธ์ให้อยู่ในรูปของ SOAP response แล้วจึงส่งกลับไปที่ไคลเอนต์

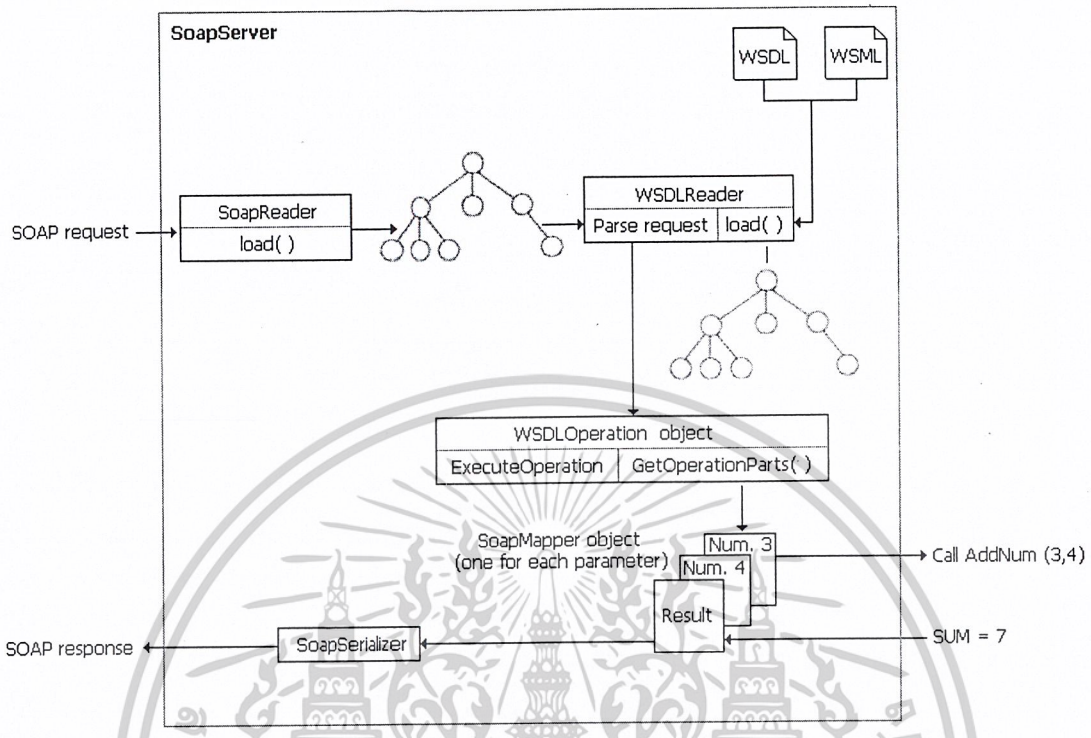


รูปที่ 3-6 Server-Side Data Flow

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.8.4 การประมวลผลภายใน SoapServer Object

การทำงานภายใน SoapServer object แสดงได้ดังรูปด้านล่าง



รูปที่ 3-7 โครงสร้างภายใน SOAP Server

เมื่อได้รับ SOAP request จากไคลเอนต์แล้ว SoapReader object ก็จะโหลด request message ไปไว้ใน DOM structure หนึ่ง ในขณะที่ WSDLReader จะโหลดไฟล์ WSDL และไฟล์ WSML ไปไว้ในอีก DOM structure หนึ่ง ตัวWSDLReader จะแปล request และสร้าง WSDLOperation object ขึ้นเพื่อใช้ใน requested operation หลังจากนั้น WSDLOperation object จะเรียกใช้เมธอด GetOperationParts เพื่อดึงรูปแบบ message ของอินพุทและเอาท์พุทของ operation ออกมา แล้ว SoapServer จะสร้าง Soap Mapper objects ให้กับแต่ละส่วนที่ได้จากเมธอด GetOperationParts และโหลดค่าที่ได้มาจากการเรียกใช้ operation เข้าไว้ใน objects เหล่านี้ พอถึงขั้นตอนนี้ SoapServer object จะทำการเรียกไปที่ COM method ที่ตรงกับที่ร้องขอเข้ามา

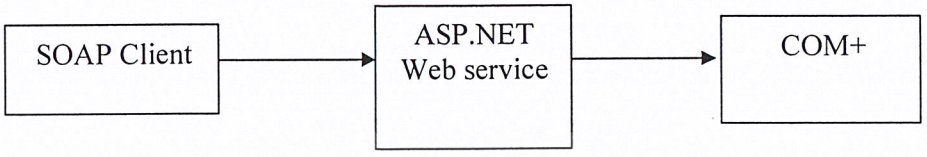
เมื่อ COM method ประมวลผลเสร็จแล้วก็จะส่งผลลัพธ์กลับไปที่ SoapServer object ซึ่ง SoapSErver object ก็จะทำการโหลดผลลัพธ์ของ operation ไปเก็บใน SoapMapper object ที่เหมาะสม แล้วจึงใช้ SoapSerializer object ส่ง SOAP response message กลับไปให้ไคลเอนต์แอพพลิเคชั่นของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.9 การสร้าง SOAP Client และ SOAP Server ด้วย .NET Framework

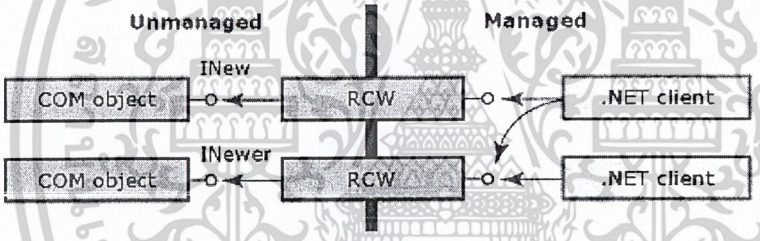
3.2.9.1 การใช้ ASP.NET ทำ SOAP เว็บเซอร์วิส

ใช้ความสามารถในการทำ COM+ Interop โดยสร้างคลาสของ ASP.NET มาเรียกใช้งาน business logic ที่สร้างด้วย Visual Basic 6



รูปที่ 3-8 การใช้ ASP.NET ทำ SOAP เว็บเซอร์วิส

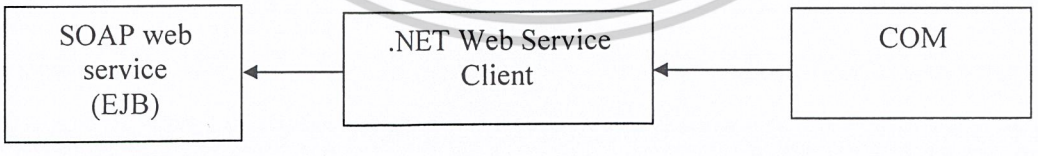
อาศัยความสามารถในการทำ marshall / unmarshall object ให้อยู่ในรูปแบบ SOAP ของ ASP.NET โดย ASP.NET ต้องทำ Interop โดยอาศัย Runtime Callable Wrapper ช่วยจำลอง COM object ให้ .NET มองเห็นเป็น .NET object ธรรมดา



รูปที่ 3-9 Runtime Callable Wrapper

3.2.9.2 การใช้ .NET ทำ SOAP Client

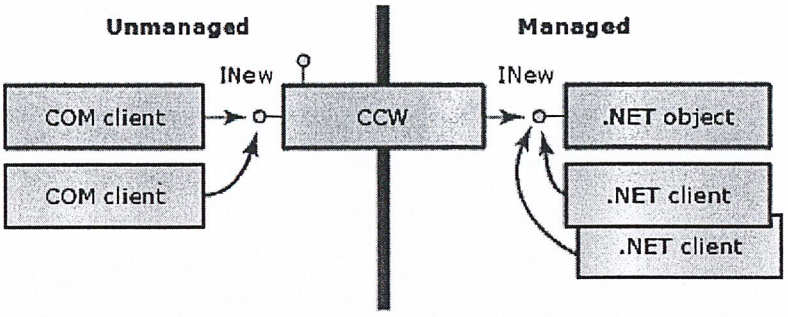
ในทางกลับกันที่ COM ต้องการใช้บริการจาก SOAP ก็กระทำผ่าน .NET เว็บเซอร์วิส client



รูปที่3-10 การใช้ .NET ทำ SOAP client

แต่การที่จะทำให้ COM ติดต่อกับ .NET class ได้นั้นต้องผ่านกระบวนการ COM Interop โดยการสร้าง COM callable wrapper ซึ่งจะทำให้ COM สามารถเห็น .NET class เป็น COM object ธรรมดา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-11 COM callable wrapper ที่ทำให้ COM client สามารถเห็น .NET object

3.3 UDDI

UDDI (Universal Description, Discovery and Integration) เป็นมาตรฐานที่จัดตั้งขึ้น โดยบริษัท IBM, Microsoft และบริษัทยักษ์ใหญ่ทางธุรกิจ B2B (Business-to-Business) อื่น ๆ UDDI ถูกสร้างขึ้นมาเป็นมาตรฐาน ในการค้นหาบริการ เว็บเซอร์วิส สำหรับคู่ค้าทางธุรกิจ ซึ่งเปรียบได้กับฐานข้อมูล ขนาดใหญ่ ซึ่งมีข้อมูลของ เว็บเซอร์วิส ที่เปิดให้บริการ โดยที่ web site สำหรับค้นหา เว็บเซอร์วิส มีอยู่หลายที่ อย่างเช่น

- http://uddi.microsoft.com/search.aspx
- http://www-3.ibm.com/services/uddi/testregistry/find
- http://uddi.org

3.4 WSDL

WSDL (Web Services Description Language) เกิดจากความร่วมมือระหว่าง IBM และ Microsoft WSDL เป็นภาษาที่ใช้อธิบายคุณลักษณะของ เว็บเซอร์วิส และวิธีการติดต่อกับ เว็บเซอร์วิส นั้น ๆ โดยใช้ไวยากรณ์ของภาษา XML ซึ่ง WSDL อยู่ในความดูแลของ W3C version ล่าสุด คือ WSDL 1.1 รายละเอียด สามารถหาอ่านเพิ่มเติมได้จากเว็บไซต์ของ W3C ที่ http://www.w3.org/TR/wsdl ส่วนในทางปฏิบัติ หากเราต้องการ สร้าง เว็บเซอร์วิส ขึ้นมาเป็นของตนเอง ก็สามารถสร้างเอกสาร WSDL ได้โดยอัตโนมัติ เราจึงไม่ต้องไปกังวลในรายละเอียด ในข้อกำหนดใน WSDL มากนักโดยตัวอย่างเอกสาร WSDL เช่น

```
<?xml version="1.0"?>
<definitions name="StockQuote"target namespace="http://example.com/stockquote.wsdl"
    xmlns:tns="http://example.com/stockquote.wsdl"
    xmlns:xsd1="http://example.com/stockquote.xsd"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    xmlns="http://schemas.xmlsoap.org/wsdl/">
  <types>
    <schema target namespace="http://example.com/stockquote.xsd"
        xmlns="http://www.w3.org/2000/10/XMLSchema">
      <element name="TradePriceRequest">
        <complexType>
          <all>
            <element name="tickerSymbol" type="string"/>
          </all>
        </complexType>
      </element>
      <element name="TradePrice">
        <complexType>
          <all>
            <element name="price" type="float"/>
          </all>
        </complexType>
      </element>
    </schema>
  </types>
  <message name="GetLastTradePriceInput">
    <part name="body" element="xsd1:TradePriceRequest"/>
  </message>
  <message name="GetLastTradePriceOutput">
    <part name="body" element="xsd1:TradePrice"/>
  </message>
  <portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
      <input message="tns:GetLastTradePriceInput"/>
      <output message="tns:GetLastTradePriceOutput"/>
    </operation>
  </portType>
  <binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    <operation name="GetLastTradePrice">
      <soap:operation soapAction="http://example.com/GetLastTradePrice"/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
  <service name="StockQuoteService">
    <documentation>My first service</documentation>
    <port name="StockQuotePort" binding="tns:StockQuoteBinding">
      <soap:address location="http://example.com/stockquote"/>
    </port>
  </service>
</definitions>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ASP .NET

4.1 ASP คืออะไร

ASP (Active Service Pages) เป็นการเขียนโปรแกรมเพื่อประมวลผลคำสั่งบนเว็บเซิร์ฟเวอร์ ก่อนจะส่งผลลัพธ์กลับมายังเบราเซอร์ที่ร้องขอข้อมูลไป

เหตุผลที่ต้องมีการประมวลผลข้อมูลบนเว็บเซิร์ฟเวอร์ ก็คือ แม้ว่าเราสามารถเขียนโปรแกรมเพื่อทำตามเงื่อนไขบางอย่างได้ทางฝั่งไคลเอนต์ หรือฝั่งของผู้ใช้ โดยใช้ภาษาสคริปต์ เช่น JavaScript, VBScript แต่ก็สามารถทำได้ภายในขอบเขตที่จำกัดเท่านั้น มูลเหตุสำคัญประการหนึ่งของการพัฒนาเว็บไซต์ในปัจจุบัน คือ จำเป็นต้องใช้ข้อมูลนการพัฒนา โดยที่ข้อมูลเหล่านี้จะมีปริมาณค่อนข้างมากตามขนาดของเว็บไซต์ ซึ่งการที่จะเคลื่อนย้ายข้อมูลเหล่านี้มายังไคลเอนต์แล้วประมวลที่นี่ย่อมเป็นไปได้ยาก นอกจากนี้แล้วความต้องการข้อมูลของผู้ใช้แต่ละคนจะมีเงื่อนไขที่ต่างกันออกไป ดังนั้นจึงจำเป็นต้องมีกระบวนการ ในการคัดเลือก และประมวลผลข้อมูลเหล่านั้นก่อนที่จะส่งไปให้กับผู้ใช้ ซึ่งการสร้างเว็บเพจแบบไดนามิก (Dynamic) นี้จะช่วยให้เราสามารถพัฒนาเว็บไซต์ในรูปแบบที่หลากหลาย และสามารถตอบสนองต่อผู้ใช้ได้อย่างสมบูรณ์มากยิ่งขึ้น

4.2 การเปลี่ยนแปลงจาก ASP ไปสู่ ASP .NET

เนื่องจากอินเทอร์เน็ตนั้นเติบโตขึ้นทุกวัน และไม่ได้จำกัดอยู่เฉพาะบนเครื่องพีซีเท่านั้น แต่ยังรวมไปถึงอุปกรณ์พกพาอย่าง มือถือ, PDA, PocketPC เป็นต้น ด้วยเหตุนี้ไมโครซอรฟต์จึงต้องการให้ ASP สามารถรองรับการทำงานร่วมกับอุปกรณ์เหล่านี้ได้ด้วย

ก่อนหน้าที่จะมาเป็น ASP .NET ไมโครซอฟต์ได้พัฒนา ASP มาแล้ว 3 เวอร์ชั่น ซึ่งแต่ละเวอร์ชั่นก็ได้รับความนิยมอย่างดีจากผู้ใช้ทั่วโลก แต่อย่างไรก็ตาม ASP เวอร์ชั่นก่อนๆ นั้นยังมีข้อจำกัดอยู่ เช่น ต้องอาศัยภาษา HTML เป็นหลักในการเขียนโปรแกรม แล้วแทรกด้วยสคริปต์ของ VBScript ทำให้มีข้อจำกัดในการพัฒนาโปรแกรมเพราะเราไม่สามารถดึงความสามารถของภาษามาใช้ได้อย่างเต็มที่ ในเวอร์ชั่น .NET จึงได้เปลี่ยนจากการใช้ VBScript มาเป็นภาษาที่สามารถขยายขีดความสามารถได้อย่างเต็มที่อย่าง VB .NET หรือ ซี-ชาร์ป (C#) ให้ผู้ใช้เลือกตามความถนัด

นอกจากนี้แล้ว ASP .NET ยังสร้างสิ่งอำนวยความสะดวกมากมายมาช่วยให้เราสามารถเขียนโปรแกรมได้ง่าย สะดวกรวดเร็ว และช่วยลดข้อผิดพลาดลงได้มาก ซึ่งต่อไปแม้ว่าผู้เขียนโปรแกรมจะไม่มีพื้นความรู้ทางด้าน Programming มากนักก็สามารถสร้างเว็บไซต์ของตนเองได้แล้ว

การเปลี่ยนแปลงที่สำคัญอีกอย่างหนึ่ง คือเปลี่ยนจากการแปลภาษาสคริปต์ทีละบรรทัด (interpret) ใน ASP แบบเดิม มาเป็นการคอมไพล์โค้ด ASP .NET ให้เป็นรูปแบบ IL (Intermediate Language) ตรงนี้เองที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สนับสนุนความจริงที่ว่า ASP .NET จะเขียนด้วยภาษาอะไรก็ได้ที่สามารถคอมไพล์เป็นรูปแบบ IL (นั่นคือ CLR ยอมรับ) ด้วยเหตุนี้ เราสามารถกล่าวได้ว่า ASP .NET ก็คือ ASP เวอร์ชั่นที่ถูกคอมไพล์นั่นเอง

นักพัฒนาที่ใช้ Visual Basic มักจะคุ้นเคยกับความสะดวกสบายในการเขียนโปรแกรมโดยใช้ฟอร์มและคอนโทรลซึ่งมีรูปแบบที่สะดวกโดยลากสิ่งที่ต้องการลงมาไว้บนฟอร์มและเขียนโค้ดโดยใช้การเขียนโปรแกรมแบบ event-handling เนื่องจากวิธีนี้เป็นวิธีที่ไม่ยุ่งยาก จึงได้ถูกนำมาใช้ใน ASP.NET

ASP.NET ทำให้การพัฒนาหน้าเว็บสะดวกขึ้นด้วยการเขียนโปรแกรมโดยใช้ฟอร์ม ใน ASP.NET เราเรียกฟอร์มเหล่านี้ว่า เว็บฟอร์ม (Web Forms) ซึ่งนำมาแทนที่หน้า ASP การเขียนโปรแกรมแบบเว็บฟอร์มนั้นเป็นแบบ event base เช่นเดียวกับใน visual basic นอกจากนี้ ASP.NET ยังแยกส่วนระหว่างส่วนแอพพลิเคชันและส่วนในการแสดงผลอีกด้วย

ASP.NET เกี่ยวข้องกับรูปแบบการเขียนโปรแกรมASP โดยมีคุณสมบัติพิเศษเพิ่มเติมขึ้นมา ดังนี้

- มีการแยกส่วนเด็ดขาดระหว่างส่วนโค้ดการประมวลผลของแอพพลิเคชันและส่วนแสดงผล (HTML markup)จึงไม่มีความสับสนในโค้ดทั้งสองส่วนอีก
- มีกลุ่มของเซิร์ฟเวอร์คอนโทรลจำนวนมากให้เลือกใช้ ซึ่งจะถูกแปลงเป็น HTML โดยอัตโนมัติ
- มีการจัดการสถานะโดยใช้อีอบเจ็กต์เซสชั่น
- เป็นรูปแบบการเขียนโปรแกรมแบบ event-base ทางฝั่งเซิร์ฟเวอร์ ทำให้ง่ายขึ้น
- สามารถเขียนโค้ดส่วนแอพพลิเคชันโดยใช้ภาษาใดใน Microsoft.NETก็ได้ เช่นภาษา VB, C# เป็นต้น โค้ดของแอพพลิเคชันบนฝั่งเซิร์ฟเวอร์จะถูกคอมไพล์เพื่อประสิทธิภาพการทำงานที่ดีกว่า
- มีVisual Studio.NET เป็นเครื่องมืออำนวยความสะดวกในการพัฒนาเว็บฟอร์ม

สรุปข้อดีของ ASP .NET

- เขียนได้ง่าย และดูแลรักษาได้ง่าย เนื่องจาก ASP .NET มีโครงสร้างที่จะช่วยทำให้การนำโค้ดกลับมาใช้ใหม่ และการใช้งานโค้ดร่วมกันทำได้ง่ายขึ้น
- การส่งมอบ (Deployment) แอพพลิเคชั่นทำได้สะดวกมากขึ้น ซึ่งจากในเวอร์ชั่นเดิมของ ASP จะมีความยุ่งยากมาก สำหรับระบบที่มีคอมโพเนนต์อยู่มากมาย แต่สำหรับ ASP .NET นี้การรีจิสเตอร์คอมโพเนนต์จะไม่มีอีกต่อไป และจะไม่มีปัญหาการล็อกไฟล์ DLL เกิดขึ้น ซึ่งในอดีตนั้น ถ้าเราจำเป็นต้องส่งมอบแอพพลิเคชั่นสำหรับเว็บไซต์ขนาดใหญ่ เราจำเป็นจะต้อง Shut down เซิร์ฟเวอร์ก่อนที่จะติดตั้งเวอร์ชั่นใหม่ลงไป ซึ่งสำหรับเว็บไซต์ขนาดใหญ่นั้น เรื่องนี้จะมีผลกระทบเป็นอย่างมาก แต่สำหรับ ASP .NET นั้นจะใช้งานไฟล์ XML เพื่อเก็บข้อมูลต่างๆ ซึ่งจะทำให้เราส่งมอบแอพพลิเคชั่นได้ง่ายขึ้น โดยก็อปปี้ทั้งไดเรคทอรี่เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ ASP .NET ยังสนับสนุนบราวเซอร์หลายๆ ตัวด้วยกัน ทำให้ปัญหาเรื่องเข้ากันได้ (Compatible) นั้นหมดลงไป และ ASP .NET ยังสนับสนุนการคอมไพล์โค้ดด้วย ทำให้เราไม่จำเป็นต้องห่วงในเรื่องของประสิทธิภาพการทำงานซึ่งเกิดขึ้นในเวอร์ชั่นก่อน ที่ใช้ภาษาสคริปต์วิ่งจะต้องมีการแปลความตอนรันแอพพลิเคชั่น นอกจากนี้เรายังสามารถคอมไพล์โค้ดคำสั่งได้ด้วย ซึ่งการคอมไพล์โค้ดจะช่วยซ่อนซอร์สโค้ดที่เราได้เขียนขึ้นมา ทำให้เมื่อเราต้องส่งมอบโปรแกรม เราก็ไม่ต้องกังวลในเรื่องของการเปิดเผยซอร์สโค้ดให้คนอื่นเห็นอีกต่อไป

สรุปตารางเปรียบเทียบระหว่าง ASP กับ ASP .NET

คุณลักษณะ	ASP	ASP .NET
นามสกุลของไฟล์	.asp	.aspx, .ascx, .asmx
การทำงานของโปรแกรม	คอมไพล์โค้ดด้วยตัว Interpreter	คอมไพล์โค้ดด้วยตัว JIT Complier
การจัดการเรื่องสถานะของโปรแกรม	- มีการใช้คุกกี้ (Cookies) ที่ไคลเอ็นต์ - ไม่สามารถทำงานร่วมกันผ่านแพลตฟอร์มได้	- สามารถฝังลงใน URL ได้เลยโดยไม่ต้องใช้คุกกี้ (Cookies) - สามารถบันทึกข้อมูลที่ใช้ร่วมกันลงในฐานข้อมูล SQL ภายนอกได้
อัพเดตไฟล์ DLL Library	ต้องปิดเครื่องเว็บเซิร์ฟเวอร์ก่อน	สามารถคัดลอกไฟล์ DLL ลงในไดเร็กทอรี่ bin ได้ทันที (ไม่ต้องมีการลงทะเบียน)

ตารางที่ 4 -1 เปรียบเทียบความแตกต่างระหว่าง ASP กับ ASP .NET

4.3 เซิร์ฟเวอร์คอนโทรล (Server Control)

การสร้างกราฟฟิกอินเทอร์เฟส (Graphic Interface) ใน ASP เวอร์ชั่นก่อนๆ ต้องอาศัยแท็ก <input ...> ของ HTML เป็นหลักแต่เนื่องจากข้อจำกัดหลายๆ ประการของแท็กดังกล่าวทำให้เราไม่สามารถสร้างกราฟฟิกอินเทอร์เฟสในรูปแบบที่แปลกๆ ใหม่ๆ และขั้นตอนการเขียนโปรแกรมก็ค่อนข้างยุ่งยาก ดังนั้นในเวอร์ชั่น .NET นี้ไมโครซอฟต์จึงได้นำเอาคอนโทรลที่เราคุ้นเคยกันดีในการสร้าง วินโดว์แอพพลิเคชั่น เช่น TextBox, Lable, Button เป็นต้น มาทำเป็นกราฟฟิกอินเทอร์เฟสบนเว็บฟอร์ม (แต่รูปแบบเดิมนั้นก็ยังสามารถใช้ได้อยู่) และสามารถเขียนโปรแกรมเพื่อจัดการบางอีเวนต์ที่สำคัญๆ ของคอนโทรลตัวนั้นๆ ได้ (แต่ทำได้เพียงบางอีเวนต์เท่านั้น ไม่ครบทุกอีเวนต์ที่มีในวินโดว์แอพพลิเคชั่น)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างโค้ดของคอนโทรลพื้นฐานที่ทำงานบนเซิร์ฟเวอร์

```
<asp : textbox id = text1 runat=server/>
text1.text = "Hello World"

<asp : checkbox id = check1 runat=server/>
check1.checked = True

<asp : button id = button1 runat=server/>
button1_onClick()

<asp : DropDownList  id = DropDownList1 runat=server/>
DropDownList1.SelectedItem.Text = "Hello"
```

4.4 วาริเดชั่นคอนโทรล (Validation Controls)

โดยทั่วไปการกรอกข้อมูลของผู้ใช้มักมีความผิดพลาดอยู่เสมอ ทั้งแบบที่ตั้งใจ และไม่ได้ตั้งใจ ในบางกรณีหากผู้ใช้กรอกข้อมูลที่ผิดพลาด อาจส่งผลให้โปรแกรมไม่สามารถทำงานต่อได้เลย วิธีการเดิมที่ใช้กันคือ การนำข้อมูลที่ผู้ใช้กรอกเข้าไปตรวจสอบด้วยการเขียนโปรแกรมเองทั้งหมด แต่ใน ASP .NET นี้มีคอนโทรลที่มาทำหน้าที่ในการตรวจสอบข้อมูลที่ผู้ใช้กรอกเข้ามาโดยเฉพาะ ว่าเป็นไปตามรูปแบบที่ต้องการหรือไม่ โดยที่เราไม่ต้องเขียนโปรแกรมให้ยุ่งยาก เพียงแต่กำหนดรูปแบบการตรวจสอบให้กับคอนโทรลเท่านั้น ก็สามารถใช้งานได้แล้ว

ตัวอย่างโค้ดของคอนโทรลวาริเดชั่น

```
<asp : RequiredFieldValidator ControlToValidate = "txtName"
ErrorMessage = "Please Enter Your Name" runat = "server"/>
```

4.5 ภาษาที่ใช้เขียน ASP .NET

ภาษาที่ใช้ในการพัฒนา ASP .NET นั้นนับได้ว่าเป็นจุดเปลี่ยนแปลงที่สำคัญที่สุดจุดหนึ่ง ของ ASP .NET จากเดิมที่เราเคยใช้กันแต่ VBScript มาคราวนี้ VBScript ได้ถูกแทนที่ด้วย VB .NET, C# และ Jscript ทั้งนี้เพื่อให้เราสามารถดึงความสามารถจากภาษาเหล่านี้มาใช้ได้อย่างเต็มที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 รู้จักกับ System.Web.UI Namespace

ในส่วนนี้จะอธิบายถึงคลาส ที่สำคัญที่อยู่ในเนมสเปซชื่อ System.Web.UI ในเฟรมเวิร์คของ ASP.NET เนมสเปซ System.Web.UI นั้นระบุคลาสและอินเทอร์เฟสที่ใช้ในการสร้างสิ่งต่างๆในคลาสเว็บฟอร์ม ที่สำคัญที่สุดในSystem.Web.UI คือคลาส Control ซึ่งบอกคุณสมบัติ เมธอด และ event ของแต่ละเซิร์ฟเวอร์คอนโทรลในเฟรมเวิร์คของเว็บฟอร์ม อีกคลาสที่สำคัญคือ เพจ ซึ่งจะกล่าวถึงในลำดับต่อไป

4.6.1 คลาส Control

คลาส Control เป็นรากฐานของคอนโทรลทั้งหมด ตัวอย่างเช่น textbox หรือ button ก็คือคอนโทรล โดยทั่วไปคลาส Control จะมีฟังก์ชั่นและคุณสมบัติของส่วนติดต่อกับผู้ใช้ทุกตัว ทุกอย่างที่เห็นใน ASP.NET ก็เป็นคลาสที่สืบทอดมาจากแหล่งใดแหล่งหนึ่งทั้งสิ้น

4.6.1.1 คุณสมบัติของ Control

คลาส Control มีคุณสมบัติที่สำคัญดังต่อไปนี้ Control, ID, Parent, EnableViewState, Visible, Context และ ViewState คุณสมบัติของคอนโทรลเหล่านี้แทนลูกของ instace ของคอนโทรลคุณสมบัติของคลาสแม่จะบอกคลาส แม่ของคอนโทรล คุณสมบัติทั้งสองนี้ ทำให้มีลำดับชั้น(hierarchy)ของคอนโทรลในหน้าเว็บคุณสมบัติ ID ทำให้คอนโทรลถูกเข้าถึงจากการเขียนโปรแกรมได้โดยใช้ ID ตามด้วยจุดและชื่อคุณสมบัติหรือเมธอดอื่นๆที่เราต้องการจะเข้าถึงของคอนโทรลนั้น เช่น MyObjectId.propertyname และยังทำให้เราสามารถเขียนตัวจัดการกับเหตุการณ์ที่เกิดขึ้นกับคอนโทรลได้ด้วย

EnableViewState เป็น flag ที่บอกว่าคอนโทรลจะยังคงเก็บview ของสถานะมันไว้หรือไม่ เช่นเดียวกับ view สถานะทั้งหมดของคอนโทรลลูกของมัน คุณสมบัติของ Context ทำให้เราสามารถดึงข้อมูลของ HTTP request ได้

ViewState เป็นอินสแตนท์ของคลาส StateBag ซึ่งใช้ในการเก็บชื่อและค่าของข้อมูลที่สามารถสร้างให้เข้าถึงได้โดยหลาย request สำหรับหน้าเพจเดียว ชื่อและค่าเหล่านี้คืออินสแตนท์ของคลาส StateItem

4.6.1.2 เมธอดในคลาส Control

เมธอดของคลาส Control นั้นมีเป็นจำนวนมาก อย่างไรก็ดีเราจะกล่าวถึงเพียงเมธอดที่มีความสำคัญ

เมธอด DataBind

เป็นการโยงคอนโทรลเข้ากับแหล่งข้อมูล เมื่อเมธอด นี้ถูกเรียก แท็กของการผูกข้อมูล <%#%> จะถูกนำมาพิจารณาใหม่เพื่อทำให้ข้อมูลใหม่ถูกโยงเข้ากับตำแหน่งtagที่เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมธอด CreateChildControls

จะถูกเรียกก่อนที่คอนโทรล compositional custom จะถูกจัดแสดงคอนโทรล compositional custom นั้นคล้ายกับคอนโทรล ActiveX คือมันจะจัดการสร้างคอนโทรลอื่นๆ ในการพัฒนาคอนโทรล custom เมธอดนี้สามารถถูก overide เพื่อให้ผู้พัฒนาคอนโทรล custom สามารถสร้าง คอนโทรลลูก ก่อนทำการแสดงคอนโทรลไม่ว่าในการทำครั้งแรกหรือครั้งต่อๆไปก็ตาม

เมธอด Render

คล้ายกับ CreateChildControls เริ่มต้นใช้สำหรับพัฒนาคอนโทรล custom ผู้พัฒนาคอนโทรลจะ overrideเมธอดนี้ เพื่อแสดงตัวเนื้อหาในคอนโทรลผ่านทางพารามิเตอร์ HtmlTextWriter ที่เตรียมไว้

เมธอด SaveViewState และ LoadViewState

เก็บบันทึกและเรียกซ้ำสถานะของคอนโทรล คอนโทรลของเซิร์ฟเวอร์ คงสถานะของมันระหว่างการrequest ผ่านทางเมธอด นี้

4.6.2 คลาสเพจ

คลาสเพจนี้สืบทอดมาจากคลาส Control ซึ่งหมายความว่ามันสืบทอดมาทั้งคุณสมบัติเมธอด และ event ที่แสดงโดยคลาส Control นอกจากการสืบทอดแล้วคลาส เพจ ระบุคุณสมบัติเมธอด และ event ที่เจาะจงมากขึ้น สำหรับหน้าเว็บในเฟรมเวิร์คของ ASP.NET

ใน ASP.NET ทั้ง Application, Request , Respond , Server และ Session เป็นคุณสมบัติของคลาสเพจ นอกเหนือจากอ๊อบเจ็กต์เหล่านี้คลาส เพจ ยังมีคุณสมบัติอื่นๆอีก เช่น Cache, ErrorPage, Postback, IsValid, Trace และ Validators

4.6.2.1 ตัวอย่างคุณสมบัติและเมธอดของคลาสเพจได้แก่

คุณสมบัติ Cache

ชี้ไปยังอ๊อบเจ็กต์ cache ของขอบเขตของหน้าปัจจุบัน ในที่นี้ ทรัพยากรเช่นการติดต่อฐานข้อมูล (database connection)จะถูกเก็บเพื่อนำมาใช้ใหม่โดยไม่ต้องสร้างการติดต่อใหม่ หากว่า cache item นั้นยังไม่หมดอายุ

คุณสมบัติ ErrorPage

บอกหน้าที่จะถูกเรียกให้แสดงเมื่อมี error เกิดขึ้น อาจใช้ @Pageแทนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติ IsPostBack

บอกว่าการเรียกหน้าเว็บเป็นการโหลดครั้งแรกหรือไม่ หาก IsPostBack มีค่าเป็น true แสดงว่าไม่ใช่การโหลดครั้งแรก จึงไม่ควรทำการ initialize หน้าเพจซ้ำ เพื่อทำให้ประสิทธิภาพดีขึ้น

คุณสมบัติ Validator

รวมกลุ่มของเซิร์ฟเวอร์คอนโทรลซึ่งสามารถกำหนดระยะการทำงานของตัวเองลงในคุณสมบัติ Validator ของหน้าเว็บ เมื่อหน้าเว็บใดต้องมีการกำหนดอายุของตัวเอง ก็จะกำหนดอายุของคอนโทรลทั้งหมดในหน้านั้นแล้วกำหนดค่าที่เหมาะสมให้แก่คุณสมบัติ IsValid

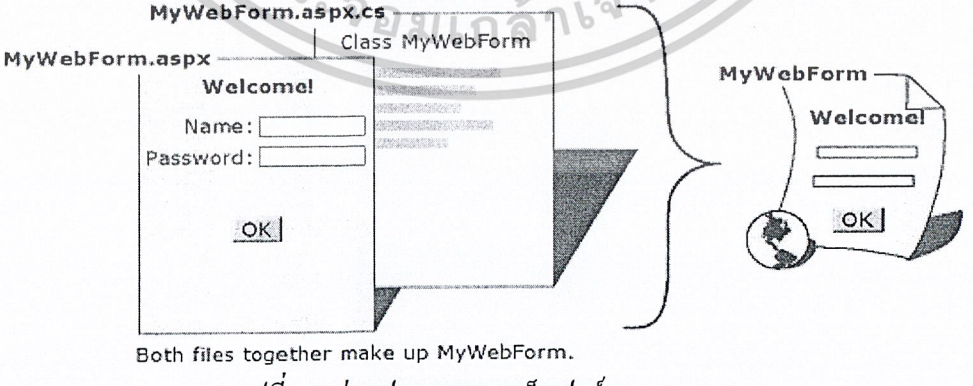
คุณสมบัติ Trace

อ้างอิงจากอ๊อบเจ็ค TraceContext ซึ่งใช้บอกคำเตือนหรือข้อความบอกความผิดพลาดที่เกิดขึ้น สามารถกำหนดให้ทำการตรวจสอบหรือไม่ทำก็ได้โดยกำหนดใน web.config ซึ่งเป็นเท็กซ์ไฟล์ที่เก็บค่าขณะรันไทม์ของแอพพลิเคชันของASP.NET

4.7 การพัฒนาเว็บแอพพลิเคชันด้วย ASP.NET

4.7.1 ส่วนประกอบคอมโพเนนต์ของ Web Form

ใน Web Forms pagesนั้น การเขียน program ของ user interface แบ่งเป็นสองส่วนคือ visual Component และ logic ซึ่งก็คล้ายกับการที่ Visual Basicมีการแบ่งระหว่างส่วนของ form และส่วน code เบื้องหลังที่ form นั้นทำงานด้วยนั่นเอง ส่วนแสดงผลที่มองเห็นได้นั้นก็คอ Web Form page ซึ่งประกอบด้วยไฟล์ ที่มี HTML หรอ ASP.NET server control. Web Form page ทำงานเหมือนเป็น container สำหรับ static text และส่วนควบคุมที่เราต้องการแสดงผล ในส่วน logic ของ Web Form page ประกอบด้วย โค้ด ที่เรา สร้างขึ้นเพื่อสื่อสารกับ form logic การเขียนโปรแกรม จะอยู่แยกกับส่วน file ของ user interface ซึ่งไฟล์นี้จะเปรียบเสมือน โค้ดที่อยู่เบื้องหลังการทำงาน สามารถเขียนด้วยภาษา Visual Basic หรือ Visual C#



รูปที่ 4-1 ส่วนประกอบของเว็บฟอร์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์ที่ประกอบด้วยคลาสต่างๆและ เบื้องหลังการทำงาน สำหรับ Web Form ทั้งหมดจะถูก คอมไพล์เป็น ไฟล์ dynamic-link library (.dll) page ที่เป็น .aspx ก็จะถูก compile เช่นกัน แต่มีกระบวนการที่ต่างกันในครั้งแรกที่ ผู้ใช้ เรียก file หน้า .aspx ASP.NET จะสร้างไฟล์ .NET class มาเป็นตัวแทนของ page โดยอัตโนมัติ แล้ว คอมไพล์ เป็น ไฟล์ .dll ตัวที่สอง class ที่ถูกสร้างสำหรับ .aspx page สืบทอดคุณสมบัติมาจาก class ที่เป็น โค้ดการทำงานเบื้องหลังที่ถูก compile ไปแล้วก่อนหน้านี้ เมื่อผู้ใช้ร้องขอ URLของ web page ไฟล์ .dll จะ run บน server และสร้างผลเป็น HTMLแบบไดนามิคสำหรับ page ที่เรียก

4.7.2 วงจรชีวิตของ Web Form

ใน ASP เว็บเพจจะเริ่มต้นวงจรชีวิตเมื่อไคลเอนท์เริ่มทำการร้องขอเพจที่เจาะจงมา IIS ทำการกระจายคำและทำการใช้งานสคริปต์บน ASP เพจ เพื่อทำให้เป็นเนื้อความ HTML ในที่สุดการทำการแปลงก็จะสิ้นสุดอายุของเพจนั้นก็จะสิ้นสุด ถ้าคุณมีฟอร์มที่ส่งข้องมูลกลับไปที่ ASP เพจก็จะเริ่มกระบวนการใหม่เหมือนกับ ASP กระทำคำร้องใหม่ โดยไม่ต้องรู้เกี่ยวกับขั้นตอนก่อนหน้าเลย การส่งผ่านข้อมูลกลับไปที่เพจเริ่มต้นสำหรับการประมวลผลก็เป็นการถูกอ้างแบบโพสต์แบ็ค (postback) ด้วย

ใน ASP.NET มีหลายสิ่งที่แตกต่างกันเล็กน้อย เพจยังคงเริ่มที่การร้องขอของไคลเอนท์อย่างไรก็ตามมันจะยังคงอยู่เรื่อยๆเท่าที่ไคลเอนท์ยังคงติดต่อกับเพจอยู่ เพื่อทำให้งายขึ้นอาจจะกล่าวได้ว่าเพจยังคงอยู่ไปเรื่อยแต่ในความเป็นจริงมีเพียงขั้นตอนการดู(view)ของเพจระหว่างการร้องขอเท่านั้น ขั้นตอนการดูเท่านั้นที่อนุญาตให้ควบคุมบนเซิร์ฟเวอร์ที่จะปรากฏว่า ถ้ามันยังคงครองอีเวนต์ของเซิร์ฟเวอร์ เราสามารถค้นหาขั้นตอนการโพสต์แบ็คของเพจผ่านคุณสมบัติ IsPostBack ของเพจอ็อบเจ็กต์และการกำหนดค่าเริ่มต้นใหม่การครองอีเวนต์ระหว่างการโพสต์แบ็คมันก็คือการทำ ASP.NET แตกต่างกว่าการพัฒนา ASP ตามแบบธรรมดาอย่างมาก

จากตัวอย่างเราทำการขยายส่วนต้น เช่นการครองโพสต์แบ็ค เมื่อทำการโหลดอีเวนต์ที่ถูกถือครองครั้งแรก เราจะทำการย้ายข้อมูลลงใน drop-down list box หลังจากนั้นเราจะระบุเวลาของอีเวนต์ที่เพิ่มขึ้นโดยไม่มีการโหลดข้อมูลเข้ามาใหม่เท่านั้น ตัวอย่างนี้แสดงให้เห็นว่าเซิร์ฟเวอร์อีเวนต์แฮนด์เลอร์(handler)ที่ชื่อ handlerButtonClick เป็นขอบเขตของอีเวนต์ที่ชื่อ ServerClick ของปุ่ม

```
<html>
    <head><title>Testing Page Event</title></head>
    <body>
        <script language="C#" runat=server>
            void Page_Init(Object oSender,EventArgs oEvent){
                labelInit.Text = DataTime.Now.ToString( );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
        void Page_Load (Object oSender, EventArgs oEvent) {
            labelInit.Text = DataTime.Now.ToString( );
            if (!IsPostBack){
                selectCtrl.Items.Add("Acrua");
                selectCtrl.Items.Add("BMW");
                selectCtrl.Items.Add("Cadillac");
                selectCtrl.Items.Add("Mercedes");
                selectCtrl.Items.Add("Porche");
            }else{
                labelLoad.text += "(Postback)";
            }
        }
        void handleButtonClick(Object oSender, EventArgs oEvent){
            labelOutput.Text = "You've selected: "+ selectCtrl.Value;
            labelEvent.Text = DateTime.Now.ToString ( );
        }
    </script>
    <form runat = server>
        Init Time: <asp:Label id=labelInit runat=server/><br/>
        Load Time: <asp:Label id=labelLoad runat=server/><br/>
        Event Time: <asp.Label id=labelEvent runat=server/><br/>
        Choice: <select id=selectCtrl runat=server/><br/>
        <asp:Label id=labelOutput runat=server></select><br/>
        <input type=button value=update
            onServerClick="handleButtonClick" runat=server />
    </form>
</body>
</html>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรชีวิตของเว็บฟอร์ม ประกอบด้วยสามขั้นตอนหลัก การกำหนดรูปแบบ การถือครองอีเวนต์ และการสิ้นสุด โดยขั้นตอนเหล่านี้จะข้ามผ่านคำร้องขอที่เหมือนกันซึ่งแย้งกับนโยบาย serving-one-page-at-a-time ที่พบใน ASP

4.7.2.1 การกำหนดรูปแบบ

ขั้นตอนการกำหนดรูปแบบนี้ โหลดอีเวนต์ของเพจจะถูกยกขึ้นมันเป็นงานของคุณในการถือครองอีเวนต์นี้เพื่อติดตั้งเพจคุณ เพราะว่าโหลดอีเวนต์จะถูยกขึ้นเมื่อการควบคุมพร้อมแล้วงานคุณตอนนี้คือการอ่านและทำการอัพเดตคุณสมบัติการควบคุมเหมือนเป็นส่วนของการติดตั้งเพจ ในตัวอย่างโค้ดข้างต้น เราจะถือครอง โหลดอีเวนต์เพื่อย้ายข้อมูลบางอย่างลง drop-down list เราอัพเดตข้อความของการควบคุม labelLoad เพื่อแสดง เวลาโหลดอีเวนต์ที่เกิดขึ้น ในแอพลิเคชันของคุณคุณจะต้องข้อมูลที่เป็นไปได้จาก ฐานข้อมูลและทำการหนดค่าเริ่มต้นฟอร์มฟิลด์ด้วยค่าดีฟอลต์

คุณสมบัติ IsPostBack ของเพจแสดงว่าเพจแรกได้ถูกโหลดหรือมันเป็นการ โพสต์แบคเช่นถ้าคุณควบคุมเนื้อหารายการของข้อมูลคุณจะต้อง โหลดการควบคุมเพจแรกที่ถูกโหลด โดยการตรวจสอบคุณสมบัติ IsPostBackของเพจนี้เท่านั้นเมื่อ IsPostBack มีค่าเป็นจริงคุณจะรู้ว่ารายการควบคุมอ็อบเจ็กต์ได้ถูกโหลดด้วยข้อมูลเรียบร้อยแล้ว ไม่จำเป็นที่ต้องย้ายข้อมูลในรายการใหม่ตัวอย่างโค้ดข้างต้นเราจะข้ามการการย้ายข้อมูลของ drop-down และแสดงเพียง สตริงว่า “(postback)”

คุณอาจจะต้องกระทำการการผูก (bind) ข้อมูลและกำหนดค่าใหม่ data-binding expressions ในตอนแรก และลำดับต่อมาของเพจนี้

4.7.2.2 การถือครองอีเวนต์

ในขั้นตอนกลางนี้ ฟังก์ชัน server event-handlingของเพจจะถูกเรียกเป็นผลของบางอีเวนต์เป็นการถูกเรียกให้ทำงานจากฝั่งไคลเอนต์ อีเวนต์เหล่านี้มาจากการควบคุมที่คุณแทนที่บนเว็บฟอร์ม

4.7.2.3 การจบการทำงาน

ที่ขั้นตอนนี้เพจจะสิ้นสุดการแปลงและถูกกำจัดทิ้งแล้วสิ่งที่คุณต้องรับผิดชอบคือการทำความสะอาดไฟล์แฮนด์เลอร์ ยกเลิกการติดต่อกับฐานข้อมูลและปล่อยอ็อบเจ็กต์ถึงแม้คุณจะเชื่อมั่นการทำงานของตัวกำจัดขยะ(garbage collection)ของ CLR ก็ตามแต่เรายังแนะนำอย่างยิ่งยวดว่าควรจำทำความสะอาดด้วยตัวเองเพราะตัวกำจัดขยะจำทำงานเป็นระยะๆเท่านั้น บนการทำงานหนักๆถ้าตัวกำจัดขยะไม่เหมาะสมจะมีทรัพยากรที่ไม่ได้คืนให้ระบบทำให้หน่วยความจำหมดไปได้และทำให้ระบบคุณค้างไป

เราสามารถทำความสะอาดตัวอย่างข้างต้นได้ด้วยการอันโหลดอีเวนต์แฮนด์เลอร์ดังที่จะแสดงต่อไปนี้เพราะว่ าไม่มีแสดงในตัวอย่างเราจะแสดงเฉพาะฟังก์ชันเป็น เทมเพลต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Void Page_Unload(Object oSender,Eventargs oEvent){
        //cleaning up code here
}
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ASP .NET เว็บเซอร์วิส

" เว็บเซอร์วิส คือ แอพพลิเคชั่น หรือโปรแกรมซึ่งทำงานอย่างใดอย่างหนึ่งในลักษณะให้บริการ โดยจะถูกเรียกใช้งานจากแอพพลิเคชั่น หรือโปรแกรมอื่นๆ ผ่านเว็บ ซึ่งเว็บเซอร์วิสได้อาศัยโปรโตคอลที่ชื่อว่า SOAP (Simple Object Access Protocol) ด้วยโปรโตคอลนี้เองที่ทำให้เราสามารถเรียกใช้แอพพลิเคชั่นใดๆ ในแพลตฟอร์มใดๆ ก็ได้ การให้บริการของเว็บเซอร์วิส จะมีเอกสารที่อธิบายคุณสมบัติของบริการกำกับไว้ (WSDL : Web Service Description Language) และมีการนำเสนอให้สาธารณชนรับทราบ ผู้ใช้บริการจึงสามารถค้นหาเว็บเซอร์วิส ได้จาก UDDI (Universal Description Discovery and Integration) โดยที่ผู้ใช้บริการไม่จำเป็นต้องรู้ที่อยู่จริงของแอพพลิเคชั่นหรือโปรแกรมนั้น "

5.1 องค์ประกอบในการใช้งาน .NET เว็บเซอร์วิส

5.1.1รูปแบบการเชื่อมต่อ

Microsoft .NET Web Services ตอนนี้รองรับ 3 โปรโตคอล ได้แก่ HTTP GET, HTTP POST และ SOAP (Simple Object Access Protocol) เนื่องจากโปรโตคอลเหล่านี้เป็นมาตรฐานของเว็บอยู่แล้ว จึงง่ายที่ไคลเอนต์จะเรียกใช้

HTTP GET

query string จะรวมอยู่ใน URL ของเซิร์ฟเวอร์นั้น

HTTP POST

จะทำการรวม ชื่อ ค่าตัวแปรต่างๆ ไว้ในส่วนของบอดี้ ของ HTTP request message

SOAP

SOAP ถูกสร้างบน XML เป็นการทำให้แอพพลิเคชั่นสามารถติดต่อกันได้ตามมาตรฐาน และง่ายในการเชื่อมต่อ ทำให้สามารถสร้างแอพพลิเคชั่นส่งไปตามอินเทอร์เน็ตได้ ดังรูปที่ 1

ถ้าเราคิดเกี่ยวกับการโต้ตอบกันระหว่าง XML Web Service เช่น การโทรศัพท์ XML จะอธิบายสิ่งที่แอพพลิเคชั่นพูดกัน โดยการใช้ภาษาของแอพพลิเคชั่น ส่วน SOAP จะอธิบายวิธีที่ติดต่อกันโดยใช้โทรศัพท์

SOAP เหมือน XML คือเป็นส่วนหนึ่งของมาตรฐาน W3C และไมโครซอฟท์เป็นผู้ร่วมออกมาตรฐานด้วย พร้อมกับบริษัทชั้นนำอื่น เช่น IBM

SOAP จะแตกต่างจากทั้งสองวิธีข้างต้น มันจะใช้ XML ในการจัดวางรูปแบบเมสเสจที่ส่งไปจะมีโครงสร้างที่ดีกว่า และสามารถส่งผ่านข้อมูลที่ซับซ้อนได้ ข้อแตกต่างอีกอย่างคือ SOAP สามารถใช้บนทรานสปอร์ต โปรโตคอล (transport protocols) อื่นๆได้ เช่น SMTP นอกเหนือจาก HTTP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.2 WSDL

WSDL (Web Services Description Language) มีวิธีที่เป็นมาตรฐานในการอธิบายฟังก์ชันที่ใช้ได้จาก XML Web Services เฉพาะ และอาร์กิวเมนต์ที่จะต้องส่งผ่านเพื่อติดต่อกัน WSDL เป็นเช่นเดียวกับ SOAP คือได้รับการดูแลโดย W3C เพื่อกำหนดเป็นมาตรฐาน

5.1.3 Web Services Discovery

เป็นกระบวนการในการค้นหาเซอร์วิสและทำการตรวจสอบ WSDL ของเว็บเซอร์วิสว่ามีอยู่จริงไหม แบ่งเป็น 2 แบบ

Static discovery

เราต้องทำการบอกชัดเจนในไฟล์ .disco ว่า wsdl อยู่ที่ไหน

```
<?xml version="1.0" ?>
<disco:discovery xmlns:disco="http://schemas.xmlsoap.org/disco">
</disco:discovery>
```

Dynamic discovery

เราสามารถตั้งค่าให้เป็นแบบ dynamic discovery ได้ ซึ่งจะทำให้เว็บเซอร์วิสทั้งหมดที่อยู่ภายใต้ URL ที่เรากำหนดจะถูกลิสต์ขึ้นมาโดยอัตโนมัติ

5.1.4 UDDI (Universal Description, Discovery and Integration)

ถ้า XML เป็นเหมือนการสนทนา และ SOAP เหมือนเป็นหลักสำหรับวิธีการติดต่อผู้อื่น UDDI ก็คือสมุดโทรศัพท์ UDDI จะจัดเตรียมไดเร็กทอรีของ XML Web Services ช่วยให้สามารถค้นหาธุรกิจที่ใช้ XML Web Services ได้

5.2 การสร้าง ASP.NET เว็บเซอร์วิส

5.2.1 ขั้นตอนในการสร้างเว็บเซอร์วิส

1. สร้างไฟล์ .asmx สำหรับเว็บเซอร์วิสที่สร้างด้วย ASP.NET ต้องประกอบไปด้วย directive <% webservice . . . %> การเขียนก็คล้ายกับการเขียนคลาส
2. เว็บเซอร์วิสคลาส (Web Services Class) นั้นต้องทำการ inherite มาจาก System.Web.Services namespace รวมทั้งทำการกำหนด namespace ของเว็บเซอร์วิสของคุณด้วยเนื่องจาก defalut ของมันคือ http://tempuri.org/ ซึ่งจะไม่ unique
3. เมธอดที่ต้องการให้เป็น เมธอดสาธารณะที่คนอื่นๆเข้ามาเรียกใช้ได้ ต้องทำการกำหนดให้เป็น <Web Method> ด้วย
4. นำไฟล์ .asmx ที่สร้างไปไว้ในไดเร็กทอรีที่ต้องการใน IIS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2 ตัวอย่างเว็บเซอร์วิส

```
<%@ WebService Language="VB" Class="Math"%>
Imports System.Web.Services
Imports System
    <WebService(Namespace:="http://www.olala.com/")> _
    Public Class Math
     <WebMethod()> _
      Public Function Add(num1 As Integer, num2 As Integer) _
            As Integer
        Return num1 + num2
      End Function
     <WebMethod()> _
      Public Function Hello() _
            As String
        Return "Hello World"
      End Function
    End Class
```

5.2.3 การกำหนดค่าแอททริบิวต์ต่างๆให้แก่เว็บเซอร์วิส

BufferResponse (boolean)

ควบคุมให้มีบัฟเฟอร์ (buffer) ของผลตอบรับของเมธอดหรือไม่

CacheDuration

กำหนดค่าช่วงเวลา (วินาที) ที่ทำการเก็บ method response ใน cache ค่าปกติคือ 0 วินาที

Description

กำหนดคำอธิบายของเมธอดนี้

EnableSession (boolean)

ให้มีการใช้ เซสชั่นสำหรับเมธอดนี้หรือไม่ โดยค่าปกติเป็น true ถ้าเราไม่ได้ ใช้งานควรกำหนดเป็น false เพื่อเพิ่มประสิทธิภาพ

MessageName

เพื่อใช้ในการแย่งแยกเว็บเมธอดในกรณีที่มี ชื่อของเมธอดเหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TransactionOption

เป็นการกำหนด transaction ของเมธอดนั้น มีทั้งหมด 5 แบบ ดังที่ได้กล่าวไปแล้วในเรื่อง Transaction ได้แก่

1. Disabled - object ที่สร้างขึ้นจะไม่มี transaction ใน COM+ transaction แต่สามารถติดต่อโดยตรงกับ DTC ในกรณีที่ต้องการการสนับสนุน

2. NotSupported - ไม่มีการทำ transaction เลย

3. Supported - ถ้า object นั้นจะถูกรันใน context transaction ของตัวที่สร้างมันขึ้นซึ่งมามีการทำ transaction, แต่ถ้าตัวมันเอง เป็นตัวเริ่ม(root - ไม่ได้ถูกใครสร้างมาอีกที) หรือ context ที่สร้างมัน ไม่ทำ transaction มันก็จะไม่ทำ transaction

4. Required - ถ้า object นั้นจะถูกรันใน context transaction ของตัวที่สร้างมันขึ้นซึ่งมามีการทำ transaction, แต่ถ้าตัวมันเอง เป็นตัวเริ่ม(root - ไม่ได้ถูกใครสร้างมาอีกที) หรือ context ที่สร้างมัน ไม่ทำ transaction มันก็จะทำ transactionใหม่

5. RequiresNew - บอกว่า object ต้องทำ transaction ใหม่

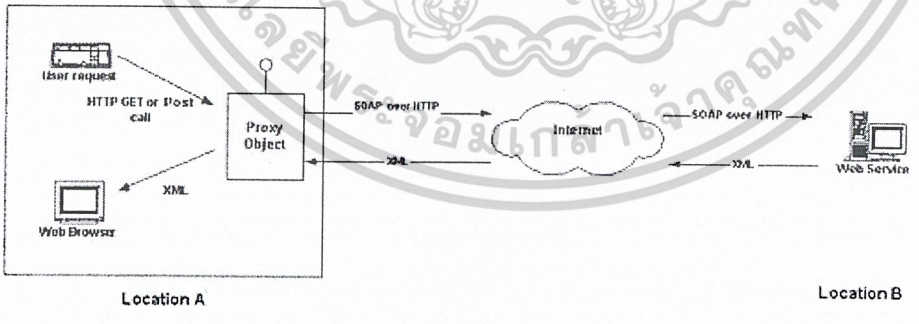
5.3 การเรียกใช้งานเว็บเซอร์วิส

ในการเรียกใช้งานเว็บเซอร์วิสนั้นเราสามารถที่จะเรียกใช้ได้ทั้ง 3 แบบ คือ HTTP GET, HTTP POST, SOAP

HTTP GET นั้นเราสามารถเรียกใช้ Web Service ได้โดยตรงผ่านทาง URL โดยใส่ parameter รวมไปด้วยเลย เช่น

http://localhost/Xtest.asmx/Add?num1=3&num2=6

แต่ในส่วนของ HTTP POST และ SOAP นั้นเราต้องทำการสร้าง Proxy Class ขึ้นมาก่อนเพื่อทำการติดต่อกับ Web Service



รูปที่ 5-1 การติดต่อจากไคลเอนต์ไปยังเว็บเซิร์ฟเวอร์ผ่านพร็อกซี่อ็อบเจ็กต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการสร้าง Proxy Class

1. ใช้ wsdl.exe ในการคอมไพล์โดยใส่พารามิเตอร์เป็น ชนิดโปรโตคอล, ตำแหน่งของเอกสาร wsdl, ภาษาที่ใช้ และอื่นๆ
2. จะได้ไฟล์ออกมา .cs ซึ่งเป็นไฟล์ของ C# เราใช้ตัวคอมไพล์ของ C# ก็จะได้ไฟล์ .dll ออกมา
3. นำไฟล์ .dll นี้ไป add reference ในโปรเจ็กที่จะเรียกใช้

5.4 การจัดการเสตทในเว็บเซอร์วิส

ในส่วนของ ASP.NET มีฟีเจอร์ในการจัดการเสตท ทั้งในส่วนของระดับแอพพลิเคชัน และระดับของเซสชั่นของผู้ใช้ ในส่วนนี้เราจะกล่าวถึงการใช้งานเสตทของ ASP.NET เพื่อนำมาพัฒนาเสตทฟูลเว็บเซอร์วิส

คำว่า เสตท นั้น เราอาจจะนิยามได้ว่า มันคือความสามารถของอ็อบเจ็กต์, ตัวแปร หรือคอนเทนเนอร์อื่นๆ ที่สามารถจดจำค่าของมันได้ ซึ่งเรามีอยู่สองระดับคือ แอพพลิเคชัน และเซสชั่น ในระดับแอพพลิเคชันนั้น เปรียบเหมือนร้านอาหาร บริกรต้องจำได้ว่า วันนี้ร้านมีเมนูพิเศษอะไรบ้าง ซึ่งจะนำไปบอกลูกค้าทุกโต๊ะเหมือนกันหมด แต่ระดับเซสชั่น บริกรจะจำได้ว่า ลูกค้าคนนี้สั่งอะไร แต่ละคนสั่งคนละอย่างไม่เกี่ยวกัน

ทั้งสองระดับที่กล่าวมานั้นใน ASP.NET เรียกว่า แอพพลิเคชันอ็อบเจ็กต์ ซึ่งให้บริการกระบวนการในการจัดเก็บข้อมูล ซึ่งสามารถเข้าถึงได้จากทุกโค้ด ที่อยู่ในเว็บแอพพลิเคชันนั้น และเซสชั่นอ็อบเจ็กต์ ซึ่งจัดเก็บข้อมูลตามเซสชั่นของไคลเอนท์แต่ละคนที่เข้ามา

5.4.1 แอพพลิเคชันอ็อบเจ็กต์

แอพพลิเคชัน อ็อบเจ็กต์จะถูกสร้างขึ้น หลังจากเกิดการเรียกใช้เว็บหรือเว็บเซอร์วิสนั้นขึ้นเป็นครั้งแรกจนกระทั่ง ปิดระบบเว็บเซอร์วิสลงไป ซึ่งแอพพลิเคชันอ็อบเจ็กต์นั้นให้การข้อดีต่างๆดังเช่น

- ง่ายต่อการเรียกใช้เสตท และสามารถใช้จากภาษาใดๆก็ได้ที่ .NET รองรับ
- มีฟังก์ชั่นในการจัดการ ซิงโครไนเซชั่น ทำให้นักพัฒนาง่ายต่อการจัดการการเข้าถึง ตัวแปรแบบโกบอลใน แอพพลิเคชันเสตท
- แอพพลิเคชันเสตท เข้าถึงได้เพียงโค้ดในส่วนของแอพพลิเคชันนั้นๆเท่านั้น ไม่สามารถที่แอพพลิเคชันอื่นจะเข้าถึงได้

ตัวอย่างการเรียกใช้งาน แอพพลิเคชัน อ็อบเจ็กต์

```
Application["DatabaseName"] = "DataStore" ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.2 เซสชั่นอ็อบเจ็กต์

เซสชั่นหมายถึง ช่วงเวลาที่ผู้ใช้คนฟนึ่งๆทำการติดต่อกับเว็บแอพพลิเคชันหนึ่ง ซึ่งใน ASP.NET มีการทำงานที่ดีขึ้นกว่า ASP เวอร์ชั่นก่อนหน้าดังนี้

- เซสชั่น อ็อบเจ็กต์โปรเซสจะถูกแยกออกมา ไม่ได้อยู่ในโปรเซสของ ASP เหมือนของเดิม ดังนั้นถ้าโปรเซสของ ASP ล้มเหลว ตัวเซสชั่นก็ยังอยู่
- เซสชั่น อ็อบเจ็กต์ใน ASP.NET ให้การสนับสนุนการใช้ เซิร์ฟเวอร์หลายๆตัว(server farm) จากแนวคิดของ out-of-process ข้างต้น เราจึงสามารถทำให้ เซิร์ฟเวอร์หลายๆใช้ เซสชั่นเสตทร่วมกันได้
- เซสชั่นอ็อบเจ็กต์ สามารถเรียกใช้โดยไม่ผ่าน คุกกี้ ได้ง่าย

การจัดการของเซสชั่นนั้น จะบอกได้จาก เลขเซสชั่นไอดี ซึ่งเป็นเลข 128 บิต หรือเรารู้จักกันดีว่า GUID's(Globally Unique Identifier) ซึ่งถูกสร้างขึ้นจากอัลกอริทึมที่มั่นใจว่าจะไม่เกิดการซ้ำกัน เลขเซสชั่นนี้จะถูกจัดเก็บในสองรูปแบบ คือ

1. ผ่านคุกกี้ ซึ่งคุกกี้ นั้นสามารถบ่งบอกถึงเซสชั่นของไคลเอนต์
2. จากการรวมค่าของไอดี เข้าไปยัง URL โดยตรง

เว็บเซอร์วิสนั้นก็ต้องการตัวในการจัดการเสตทกับไคลเอนต์ เครื่องไคลเอนต์ที่จะเรียกใช้เว็บเซอร์วิสต้องสนับสนุนการใช้งานคุกกี้ด้วย เพื่อที่จะเก็บ เซสชั่นไอดี

ตัวอย่างการใช้งาน เซสชั่นอ็อบเจ็กต์ใน ASP.NET

```
Session["UserName"] = "John Smith" ;
```

5.4.3 โหมดการจัดการเซสชั่นและการคอนฟิกค่า

การคอนฟิกค่าต่างๆใน ASP.NET แบ่งออกได้เป็นสองส่วนด้วยกัน คือ machine.config และ web.config ไฟล์แรกนั้นจะส่งผลกับทุกเว็บแอพพลิเคชันในเครื่อง และ ไฟล์หลังจะเป็นการคอนฟิกเฉพาะแอพพลิเคชันซึ่งจะทับแมชชีนคอนฟิกอีกที

โหมดในการใช้งานเซสชั่นนั้นมีด้วยกัน 3 โหมด ได้แก่

1. In Proc เป็นค่าเริ่มต้นที่ติดตั้ง คือจะจัดเก็บเซสชั่นในโปรเซสเดียวกับเว็บเซิร์ฟเวอร์
2. State Server ทำการจัดเก็บลงในแบบ out-of-process ลงในเมมโมรี่ โดยเครื่องที่จะทำเซิร์ฟเวอร์นี้สามารถเป็นเครื่องเดียวกับ เว็บเซิร์ฟเวอร์ได้

   ```
   stateConnectionString="tcpip=127.0.0.1:42424"
   ```

3. SqlServer ทำการจัดเก็บลงในแบบ out-of-process เช่นกัน แต่จัดเก็บลงในดาต้าเบส

   ```
   sqlConnectionString="data source=127.0.0.1;user id=sa;password="
   ```

ค่าอื่นๆที่สามารถปรับแต่งได้ก็คือ โหมดของการใช้ คุกกี้ ว่าจะเป็นแบบ ใช้คุกกี้ หรือว่า ผ่าน url และการกำหนดค่าช่วงเวลาของคุกกี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
cookieless="true"
timeout="40"
```

ในเรื่องของประสิทธิภาพ โหมด In Proc มีประสิทธิภาพมากกว่า State Server และ SqlServer ตามลำดับ แต่ สองโหมดหลังจะสามารถใช้กับแบบ เซิร์ฟเวอร์ฟาร์มได้ โหมด SqlServer ก็จะช้าที่สุดแต่ก็มีความน่าเชื่อถือที่สุดเช่นกัน

5.4.4 การเรียกใช้งาน แอพพลิเคชัน และ เซสชั่นเสตทใน เว็บเซอร์วิส

เว็บเซอร์วิสเป็นส่วนหนึ่งของ ASP.NET จึงสามารถใช้งานในส่วนเสตทนี้ทำให้เป็น เสตทฟูล เว็บเซอร์วิสได้ ซึ่งการใช้งานนั้นต้องทำการเซ็ตค่าของเว็บเมธอดเพื่อให้จำค่าเซสชั่นได้

```
[WebMethod(EnableSession:=True)]
```

ในกรณีที่เมธอดนั้นมีการใช้งาน แอพพลิเคชันอ็อบเจ็กต์เท่านั้น ไม่จำเป็นที่ต้องกำหนดค่าดังกล่าว

การเขียนแอพพลิเคชัน เรียกใช้งานเว็บเซอร์วิส

การเรียกใช้งานนั้นอย่างที่บอกไปแล้ว ว่าต้องใช้คุกกี้เพื่อให้จำเลขเซสชั่น ไอดีได้

1. ทำการสร้างเว็บเซอร์วิสอ็อบเจ็กต์ขึ้นมา

```
localhost.StatefulService objService = new localhost.StatefulService();
```

2. สร้าง CookieContainer ซึ่งต้อง using System.Net ด้วย

```
CookieContainer objCookie = new CookieContainer();
```

3. ทำการกำหนดค่าให้แก่ เว็บเซอร์วิสอ็อบเจ็กต์

```
objService.CookieContainer = objCookie;
```

ถ้าเป็นในส่วนของเว็บฟอร์มนั้น ต้องมีการกำหนดค่า

```
if(Session["Container"] == null)
{
        objContainer = new CookieContainer(); // first call
        Session["Container"]=objContainer;
}
else
        objContainer = (CookieContainer)Session["Container"]; // all subsequent
callwb.CookieContainer = objContainer;
```

เพื่อให้ทุกครั้งที่มีการโหลดเพจใหม่ขึ้นมา ยังจำค่าคุกกี้เดิมได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 ทรานแซกชัน

5.5.1 เหตุผลที่ต้องใช้ทรานแซกชัน

ตัวอย่างปัญหาที่ต้องใช้ทรานแซกชันในการแก้

- **Atomic Operation**

กรณีที่ต้องการทำคำสั่งหลาย ๆ คำสั่ง โดยมองทั้งหมดเป็นเหมือนคำสั่งเดียว (atomic operation) เช่น กรณีบัญชีธนาคาร เมื่อต้องการโอนเงินจากบัญชีหนึ่งไปยังอีกบัญชีหนึ่ง ต้องถอนเงินออกจากบัญชีหนึ่ง จากนั้นจึงฝากเงินเข้าอีกบัญชีหนึ่ง ถ้ามีคำสั่งใดคำสั่งหนึ่งล้มเหลว ต้องให้คำสั่งทั้งคู่ล้มเหลว ไม่เช่นนั้นยอดเงินคงเหลือในบัญชีจะไม่ถูกต้องจึงต้องไม่เกิดสถานการณ์ที่คำสั่งหนึ่งทำงานสำเร็จในขณะที่อีกคำสั่งทำงานล้มเหลว เนื่องจากทั้งสองคำสั่งนั้นเป็นทรานแซกชันเดียวกัน

การแก้ปัญหานี้วิธีหนึ่งคือการใช้ exception ดังตัวอย่างต่อไปนี้

```
try {
// Withdraw funds from account 1
}
catch (Exception e) {
// If an error occurred, do not proceed.
return;
}
try {
// Otherwise, deposit funds into account 2
}
catch (Exception e) {
// If an error occurred, do not proceed,
// and redeposit the funds back into account 1.
return;
}
```

ตัวอย่างโค้ดข้างต้น ถอนเงินจากบัญชีที่ 1 ก่อน หากเกิดข้อผิดพลาดขึ้น แอพพลิเคชันจะจบการทำงาน และไม่มีผลของการทำงานใด ๆ เกิดขึ้น หากไม่เกิดข้อผิดพลาดก็จะฝากเงินเข้าไปยังบัญชีที่ 2 ในขณะฝากเงินนั้น หากเกิดข้อผิดพลาด จะต้องทำการฝากเงินกลับเข้าไปในบัญชีที่ 1 ใหม่ จากนั้นจึงจบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาที่เกิดขึ้นจากการแก้ปัญหาด้วยวิธีนี้คือ

- โปรแกรมมีความซับซ้อน
- ต้องตรวจสอบทุกความเป็นไปได้ที่จะทำให้เกิดการทำงานผิดพลาดในทุก ๆ ขั้นตอน และต้องเขียนโค้ดเพื่อยกเลิกการเปลี่ยนแปลงที่ได้ทำไปก่อนที่จะเกิดความผิดพลาดนั้นเอง
- ในการตรวจสอบความผิดพลาด แทบจะไม่สามารถควบคุมได้หากกระบวนการทั้งหมดมีความซับซ้อน เช่นการปรับปรุงข้อมูลทางการเงิน ที่ต้องอาศัยการทำงานทั้งหมด 10 ขั้นตอนด้วยกัน จะต้องหาวิธีการในการรองรับการทำงานที่ผิดพลาดในทุกขั้นตอนที่เป็นไปได้
- การตรวจสอบการทำงานได้ยาก เพราะต้องทำการจำลองการทำงานที่ผิดพลาดที่เกิดขึ้นได้ในทุกขั้นตอน

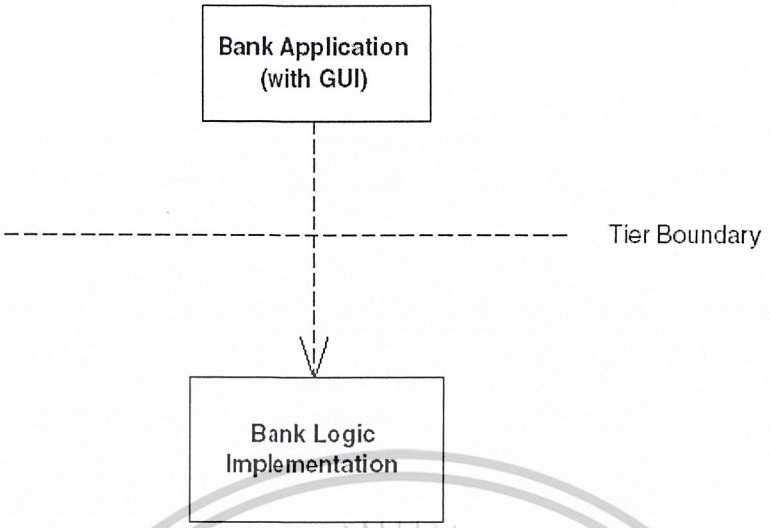
ในทางอุดมคตินั้นต้องการวิธีการที่สามารถทำคำสั่งทั้งสองได้ โดยมองเป็นคำสั่งใหญ่เพียงคำสั่งเดียว และรับประกันว่าคำสั่งทั้งสองจะต้องทำงานสำเร็จทั้งคู่ หรือทำงานผิดพลาดทั้งคู่เท่านั้น

- **การทำงานผิดพลาดของเครื่องคอมพิวเตอร์ หรือระบบเครือข่าย**

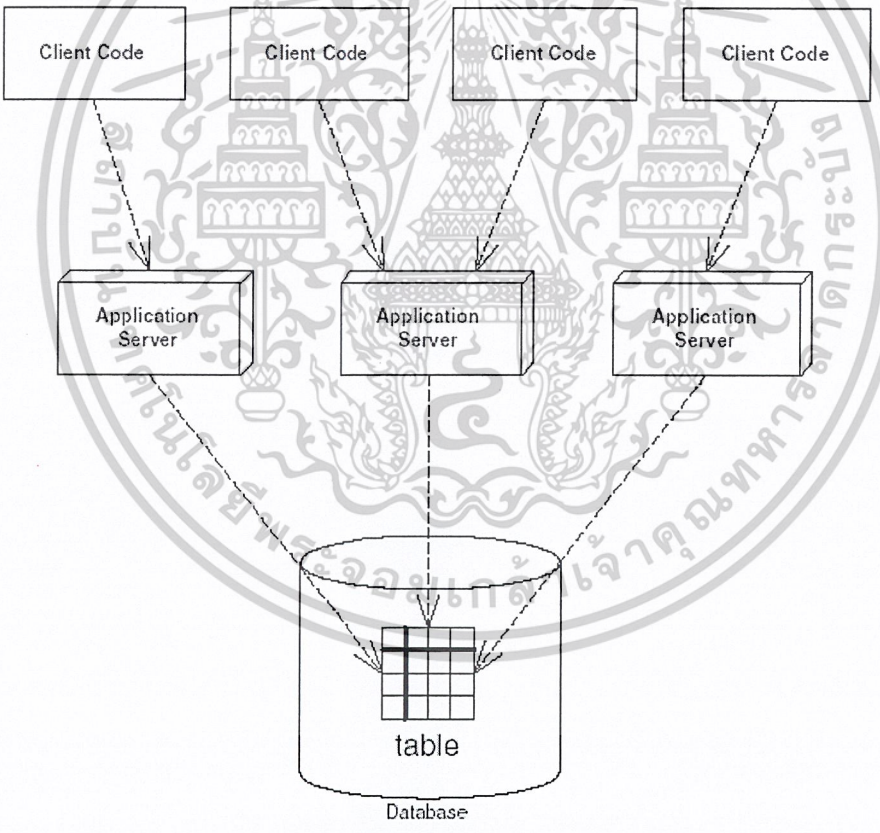
จากตัวอย่างข้างต้น ถ้าลอจิกต่าง ๆ ที่ทำงานกับบัญชีธนาคารถูกกระจายบนเครือข่ายโดยดีพลอยแบบมัลติเทียร์ เพื่อเหตุผลทางด้านความปลอดภัย , การเพิ่มขยาย หรือประโยชน์จากการแบ่งโปรแกรมออกเป็นโมดูล ในการดีพลอยแบบมัลติเทียร์ ไคลเอนต์ที่ต้องการใช้งานแอพพลิเคชันบัญชีธนาคารนี้จะต้องกระทำผ่านทาง remote method invocation แสดงในรูปที่ 5-2 ทำให้ต้องคำนึงถึงเรื่องของการทำงานผิดพลาดและความน่าเชื่อถือ เช่น ระบบเครือข่ายแครชระหว่างที่โปรแกรมบัญชีธนาคารกำลังทำคำสั่งใด ๆ อยู่ แม้ว่าเมื่อระบบเครือข่ายแครชจะมีการสร้าง exception ขึ้นมา แต่การแครชของระบบเครือข่ายนั้นอาจจะเกิดขึ้นก่อนหน้าการถอนเงิน หรืออาจเกิดขึ้นภายหลังการถอนเงินก็ได้ ซึ่งไม่อาจแยกความแตกต่างระหว่าง exception ที่เกิดขึ้นจากกรณีทั้งสองนี้ได้เลย

ในความเป็นจริง ระบบเครือข่ายไม่ได้เป็นจุดเดียวที่ทำให้เกิดปัญหานี้ขึ้น เนื่องจากข้อมูลบัญชีธนาคารก็คือข้อมูลแบบ persistent ที่อยู่ในฐานข้อมูล มีความเป็นไปได้เสมอที่ระบบฐานข้อมูลจะแครช หรือเครื่องคอมพิวเตอร์ที่ดีพลอยระบบฐานข้อมูลไว้จะเกิดแครช ถ้าหากการแครชนั้นเกิดขึ้นในระหว่างที่กำลังเขียนข้อมูลลงไปในฐานข้อมูล อาจทำให้ฐานข้อมูลอยู่ในสถานะที่ไม่ consistent ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-2 แอพพลิเคชันที่ทำงานในระบบแบบกระจาย



รูปที่ 5-3 การใช้งานฐานข้อมูลพร้อม ๆ กัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **การที่ผู้ใช้หลายคนสามารถใช้งานข้อมูลเดียวกันพร้อมกันได้**

ในระบบแบบ enterprise-level distributed system เป็นรูปแบบที่ไคลเอนต์หลาย ๆ ตัวติดต่อกับแอพพลิเคชันเซิร์ฟเวอร์หลาย ๆ ตัว ถ้าแอพพลิเคชันเซิร์ฟเวอร์ทั้งหมดใช้ฐานข้อมูลเดียวกัน (ดังรูปที่ 5-3) เซิร์ฟเวอร์สามารถที่จะแก้ไขข้อมูลตัวเดียวกันที่อยู่ในฐานข้อมูลได้

ถ้ามีไคลเอนต์สองตัวพยายามแก้ไขข้อมูลเดียวกันที่อยู่ในฐานข้อมูลในเวลาเดียวกันแล้ว คำสั่งของไคลเอนต์ทั้งสองอาจทำงานซ้อนกัน และได้ผลลัพธ์สุดท้ายผิดพลาด ซึ่งการมีข้อมูลที่ไม่ถูกต้องในฐานข้อมูล ในระบบทั่วไปถือว่าเป็นสิ่งที่ยอมรับไม่ได้ (unacceptable) และอาจสร้างความเสียหายอย่างมากให้กับองค์กรดังนั้นจึงจำเป็นต้องหาวิธีการที่จะจัดการกับการแก้ไขข้อมูลพร้อมกันของไคลเอนต์หลาย ๆ ตัว โดยจะต้องไม่ทำให้ผลลัพธ์สุดท้ายที่ได้นั้นเป็นผลลัพธ์ที่ไม่ถูกต้อง

5.5.2 ประโยชน์ของทรานแซกชัน

ทรานแซกชัน คือเซ็ตของคำสั่งที่ถูกเอ็กซิคิวต์โดยมองว่าเป็นคำสั่งใหญ่เพียงอันเดียว ทรานแซกชันรับประกันว่าคำสั่งทั้งหมดทำงานสำเร็จ มิฉะนั้นจะต้องไม่มีผลการทำงานจากคำสั่งใดเกิดขึ้นเลย ทรานแซกชันสามารถแก้ปัญหาที่เกิดจากการทำงานผิดพลาดของระบบเครือข่ายได้ อนุญาตให้ไคลเอนต์หลาย ๆ ตัวใช้งานข้อมูลเดียวกันพร้อมกันได้ และรับประกันว่าข้อมูลทั้งหมดที่คำสั่งเหล่านี้ทำการแก้ไขจะถูกต้องสมบูรณ์ โดยไม่ถูกรบกวนจากการแก้ไขข้อมูลโดยไคลเอนต์อื่น ๆ

การใช้ทรานแซกชันอย่างถูกต้องทำให้สามารถมีหลายไคลเอนต์ทำงานกับฐานข้อมูลได้โดยไม่ขึ้นต่อกัน เช่น ถ้ามีไคลเอนต์สองตัวอ่านและเขียนข้อมูลลงในฐานข้อมูล ไคลเอนต์ทั้งสองจะ mutual exclusive กันระบบจัดการฐานข้อมูลจะจัดการเกี่ยวกับการควบคุมการทำงานพร้อมกันที่จำเป็นบนฐานข้อมูล เช่น การล็อคข้อมูล เพื่อไม่ให้การทำงานของแต่ละเธรดมีผลกระทบซึ่งกันและกัน

5.5.2.1 ACID Properties

เมื่อใช้งานทรานแซกชันอย่างถูกต้อง คำสั่งจะถูกเอ็กซิคิวต์ โดยมีการรับประกันผลการทำงานทั้งสิ้น 4 อย่างด้วยกัน เรียกว่า ACID properties โดยคำว่า ACID ย่อมาจาก Atomicity, Consistency, Isolation และ Durability

- **Atomicity** - รับประกันว่าคำสั่งทั้งหมดจะถูกรวมเข้าไว้ด้วยกัน และมองเป็นหน่วยของการทำงานเพียงอันเดียว การทำงานของทรานแซกชันจะเป็นไปในลักษณะของ All-or-Nothing คือ การอัพเดตข้อมูลในฐานข้อมูลทั้งหมดจะต้องเสร็จสิ้น หรือถ้าเกิดข้อผิดพลาดใด ๆ ขึ้นระหว่างการทำงาน การอัพเดตฐานข้อมูลทั้งหมดนั้นจะต้องไม่เกิดขึ้นเลย โดยการอัพเดตฐานข้อมูลที่เกิดขึ้นก่อนแล้ว จะต้องถูกยกเลิก
- **Consistency** - รับประกันว่าทรานแซกชันจะต้องทำให้ระบบอยู่ในสถานะที่ consistent เมื่อทรานแซกชันทำงานเสร็จเรียบร้อย เช่น การโอนเงินจะต้องไม่มีเงินหายไปจากระบบ แต่ในระหว่างการทำงานของทรานแซกชันนี้ ขณะที่เงินถูกถอนออกมาจากบัญชีแรก ข้อมูลอยู่ในสถานะที่ไม่ consistent แต่เนื่องจากทรานแซกชันจะมองเห็นคำสั่งทั้งหมดเปรียบเสมือนเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งเดียว ดังนั้นถ้าทรานแซกชันสำเร็จก็คือคำสั่งทั้งหมดจะต้องสำเร็จ แต่ถ้าทรานแซกชันทำงานไม่สำเร็จคำสั่งทั้งหมดก็จะต้องไม่สำเร็จ และการเปลี่ยนแปลงใด ๆ ที่ได้กระทำไประหว่างนั้นจะต้องถูกยกเลิก ตามคุณสมบัติข้อแรกของทรานแซกชัน ดังนั้นข้อมูลจะอยู่ในสถานะที่ consistent เสมอ

- **Isolation -** เป็นการป้องกันทรานแซกชันที่ทำงานพร้อมกัน ไม่ให้เห็นผลลัพธ์ที่ยังไม่สมบูรณ์ที่เกิดจากทรานแซกชันอื่น isolation อนุญาตให้หลายทรานแซกชันสามารถอ่านหรือเขียนฐานข้อมูลโดยไม่รู้ถึงการมีอยู่ของทรานแซกชันอื่น ถ้าไม่มีคุณสมบัติ isolation สถานะของข้อมูลอาจจะอยู่ในสถานะที่ไม่ consistent ได้ ซึ่งมีประโยชน์มากในกรณีไคลเอนต์หลายตัวแก้ไขฐานข้อมูลในเวลาเดียวกัน โดยแต่ละไคลเอนต์จะทำงานโดยเปรียบเสมือนว่ามีเพียงตัวมันเองเท่านั้นที่กำลังใช้งานฐานข้อมูลอยู่ ทรานแซกชันสร้างคุณสมบัติ isolation โดยการใช้ synchronization โพรโตคอลต่าง ๆ กับข้อมูลในฐานข้อมูล เช่น ระหว่างการทำงานของทรานแซกชัน ข้อมูลจะถูกล็อคและปลดล็อคโดยอัตโนมัติตามความจำเป็น และการอัพเดตข้อมูลของทรานแซกชันอื่น จะไม่มีผลกับข้อมูลที่ทรานแซกชันนี้กำลังทำงานอยู่
- **Durability** - รับประกันว่าการเปลี่ยนแปลงข้อมูลใด ๆ ในฐานข้อมูลที่เสร็จสิ้น (commit) แล้วจะต้องมีความถาวร คือต้องไม่สูญหายเมื่อเกิดความผิดพลาดใด ๆ ขึ้น เช่น เครื่องคอมพิวเตอร์แครช ,ระบบเครือข่ายแครช , ฮาร์ดดิสก์แครช หรือไฟฟ้าดับ โดยต้องมีการเก็บ log file ของการอัพเดตข้อมูลใด ๆ ในทรานแซกชัน เมื่อระบบเกิดแครชขึ้น ข้อมูลสามารถสร้างขึ้นมาใหม่โดยการทำงานตามลำดับใน log file อีกครั้งหนึ่ง

5.5.3 ทรานแซกชันโมเดล

ทรานแซกชันโมเดลเป็นวิธีการที่ต่างกันในการทำทรานแซกชัน มีหลายโมเดลที่ใช้จัดการทรานแซกชัน แต่ละโมเดลจะมีความซับซ้อนและฟีเจอร์ที่ต่างกัน 2 โมเดลที่ได้รับความนิยมมากที่สุดคือ flat transaction และ nested transaction

- **Flat Transactions**

flat transaction เป็นทรานแซกชันโมเดลที่เข้าใจได้ง่าย flat transaction เป็นกลุ่มของ operation ที่ทำงานเป็นหน่วยเดียว ถ้า operation ไม่สำเร็จทั้งหมดจะเกิด rollback ขึ้น แต่ถ้าสำเร็จจะเกิด commit กระบวนการของ flat transaction ดังรูปที่ 5-4

- **Nested Transactions**

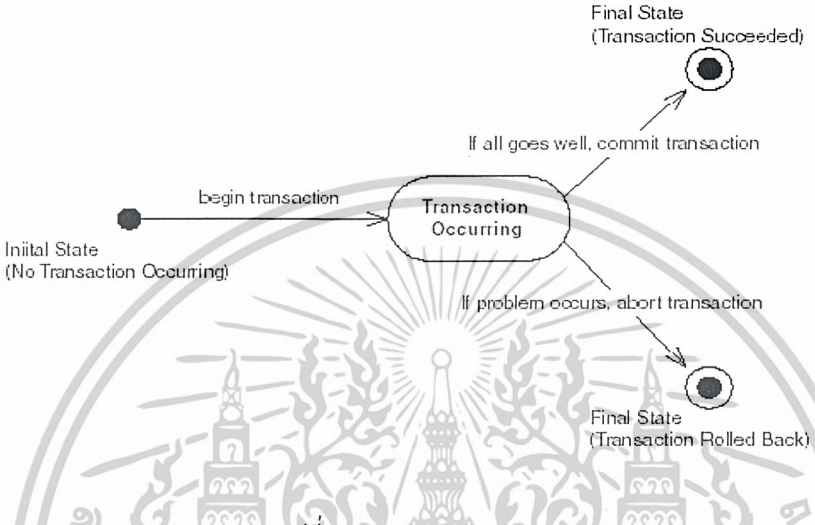
ตัวอย่างของงานที่ต้องใช้ nested transaction คือปัญหา trip-planning โดยแอพพลิเคชันจะมี operation ดังนี้

1. ซื้อตั๋วรถไฟจากบอสตันไปยังนิวยอร์ค
2. ซื้อตั๋วเครื่องบินจากนิวยอร์คไปยังลอนดอน
3. ซื้อตั๋วบอลลูนจากลอนดอนไปยังปารีส
4. แอพพลิเคชันพบว่าไม่มีไฟลท์ไปยังปารีส

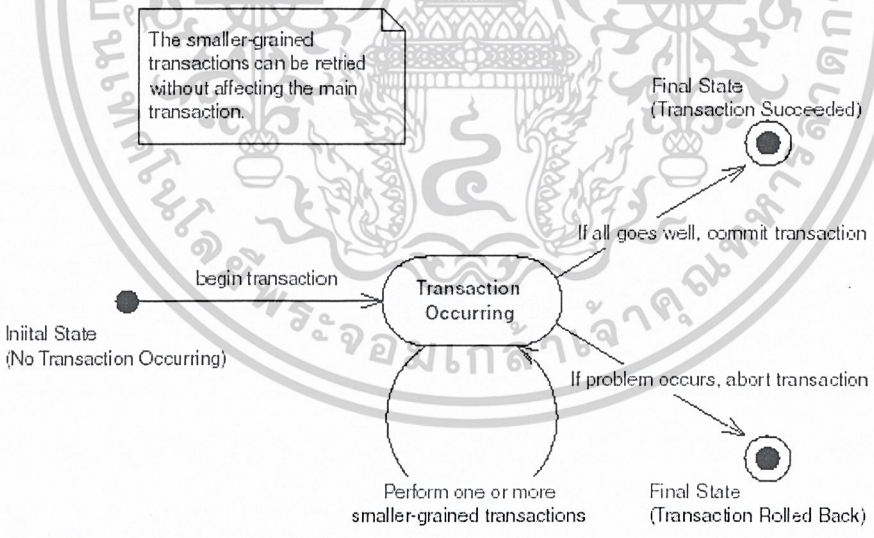
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าลำดับของโอเปอร์เรชั่น (operation) นี้ทำกับ flat transaction จะเกิดโรแบ็ก (rollback) เท่านั้น เนื่องจากไม่มีไฟลท์ไปยังปารีส แอพพลิเคชันจะต้องไม่เก็บการจองใด ๆ ไว้เลย แต่ตรงจุดนี้อาจมีทางในการเปลี่ยนการเดินทางด้วยบอลลูนไปเป็นทางอื่น ทำให้เก็บข้อมูลการจองไว้ได้

nested transaction ช่วยในการแก้ปัญหานี้ มันจะยอมให้เกิดทรานแซกชันซ้อนกันได้ รูปที่ 5-5 แสดงแนวคิดของ nested transaction



รูปที่ 5-4 Flat Transaction



รูปที่ 5-5 Nested Transaction

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.4 Transactional Isolation

คุณสมบัติ isolation รับประกันว่าผู้ใช้ที่ทำงานอยู่พร้อม ๆ กันจะไม่รู้สึกและได้รับผลกระทบจากผู้ใช้คนอื่น ๆ แม้ว่าผู้ใช้ทั้งหมดนั้นกำลังทำงานกับข้อมูลเดียวกันในฐานข้อมูลก็ตาม โดยสามารถระบุระดับของ isolation ที่ต้องการได้

การระบุระดับของ isolation สามารถควบคุมได้ว่าจะให้แต่ละทรานแซกชัน isolate จากกันมากน้อยเพียงใด ถ้าตั้งระดับของ isolation ที่สูง ก็มั่นใจได้ว่าแต่ละทรานแซกชันนั้น จะถูก isolate ออกจากทรานแซกชันอื่น ๆ ทั้งหมด แต่การตั้งระดับของ isolation สูงก็ทำให้ประสิทธิภาพในการทำงานที่ได้ลดลง เนื่องจากในการทำ isolation นั้นใช้วิธีการล็อคค่าในฐานข้อมูลเป็นหลัก แต่ถ้าตั้งระดับของ isolation ไว้ต่ำ ทรานแซกชันก็จะไม่ isolate แต่ประสิทธิภาพในการทำงานร่วมกันของหลาย ๆ ทรานแซกชันจะดีขึ้น

ดังนั้น ต้องพิจารณาให้รอบคอบว่าต้องการความเป็น isolation สูงต่ำเพียงใดในการทำงาน การกำหนด transaction isolation มีอยู่ทั้งหมด 4 ระดับด้วยกัน ดังนี้

- **READ UNCOMMITTED**

เป็นระดับการทำงานที่ต่ำที่สุด ไม่มีการรับประกันคุณสมบัติ isolation เลย แต่จะให้ประสิทธิภาพในการทำงานสูงสุด การทำงานในระดับนี้อาจเกิดปัญหา dirty read ขึ้นได้ โดยหากทรานแซกชันหนึ่งทำงานพร้อมกันไปกับทรานแซกชันอื่น แล้วอีกทรานแซกชันนั้นเขียนข้อมูลโดยยังไม่ได้ commit ทรานแซกชันเดิมจะอ่านค่านั้นเข้ามาได้ทำให้ได้ค่าที่ผิดพลาดไป โดยไม่ขึ้นกับระดับของ isolation ที่อีกทรานแซกชันหนึ่งใช้งานอยู่ นอกจากนี้ยังเกิดปัญหา unrepeatable read และ phantom phenomenon ได้อีกด้วย

- **READ COMMITTED**

การทำงานในระดับนี้คล้ายกับระดับ READ UNCOMMITTED แตกต่างกันที่ในระดับนี้โค้ดจะอ่านค่าข้อมูลที่ commit แล้วเท่านั้น จะไม่อ่านค่าที่ถูกเขียนลงไปยังฐานข้อมูลแล้วแต่ยังไม่ commit ดังนั้นระดับของ isolation ระดับนี้จึงสามารถแก้ปัญหา dirty read ได้ แต่ยังไม่สามารถแก้ปัญหาที่ซับซ้อนเช่นปัญหา unrepeatable read และ phantom phenomenon

- **REPEATABLE READ**

เป็นการรับประกันที่สูงขึ้นไปจาก READ COMMITTED เมื่อใดก็ตามที่อ่านค่าข้อมูลที่คอมมิท (commit) แล้วจากฐานข้อมูล สามารถจะอ่านค่าข้อมูลนั้นซ้ำได้ และค่าที่ได้จะเป็นค่าที่เหมือนกับค่าที่เราอ่านขึ้นมาได้ครั้งแรก ซึ่งระดับของ isolation ที่ READ COMMITTED หรือต่ำกว่า ทรานแซกชันอื่นที่ทำงานอยู่พร้อมกันอาจจะคอมมิทข้อมูลในระหว่างการอ่านข้อมูลแต่ละครั้งได้ ทำให้ค่าของข้อมูลเดียวกันที่อ่านได้ในแต่ละครั้งไม่ตรงกัน การทำงานในโหมดนี้แก้ปัญหาที่เรียกว่า unrepeatable read

- **SERIALIZABLE**

สามารถแก้ปัญหา phantom phenomenon คือการที่เซตของข้อมูลใหม่ถูกเพิ่มเข้ามาในฐานข้อมูลระหว่างการอ่านข้อมูลของทรานแซกชัน รวมถึงแก้ปัญหาอื่น ๆ ที่กล่าวมาก่อนหน้านี้ทั้งหมด รับประกันว่าทุก ๆ ทรานแซกชันจะทำงานเป็นลำดับกับทรานแซกชันอื่น ๆ และมีคุณสมบัติ ACID properties ของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฐานข้อมูลครบถ้วน หมายความว่าแต่ละทรานแซกชันจะทำงานแยกกับทรานแซกชันอื่น ๆ อย่างแท้จริง ตารางที่ 5-1 จะแสดงถึงปัญหาที่อาจเกิดในแต่ละระดับของ isolation

ISOLATION LEVEL	DIRTY READ?	UNREPEATABLE READS?	PHANTOM READ?
READ UNCOMMITTED	Yes	Yes	Yes
READ COMMITTED	No	Yes	Yes
REPEATABLE READ	No	No	Yes
SERIALIZABLE	No	No	No

ตารางที่ 5-1ระดับของ Isolation

การกำหนด isolation ใน EJB ในกรณีของ bean-managed persistent จะกำหนด isolation โดยใช้ resource manager API เช่น JDBC โดยการเรียก *java.sql.Connection.SetTransactionIsolation(...)* ส่วนในกรณีของ container-managed persistent จะไม่สามารถกำหนดระดับของ isolation ใน deployment descriptor ได้ การกำหนดระดับของ isolation จะขึ้นอยู่กับผลิตภัณฑ์ที่ใช้เป็นคอนเทนเนอร์หรือฐานข้อมูล

5.5.5 ทรานแซกชันแบบกระจาย

นอกจากทรานแซกชันพื้นฐาน คือแอพพลิเคชันเซิร์ฟเวอร์หนึ่งตัวเชื่อมต่อกับฐานข้อมูลหนึ่งตัวแล้ว ยังสามารถที่จะทำ distributed transaction ได้เช่นเดียวกัน โดยขึ้นอยู่กับการรองรับของแอพพลิเคชันเซิร์ฟเวอร์ ซึ่ง distributed transaction ยึดหลักการเดียวกับทรานแซกชันธรรมดา โดยที่หากมีคอมโพเน็นต์หนึ่งบนเครื่องคอมพิวเตอร์เครื่องหนึ่ง abort ทรานแซกชัน ทรานแซกชันทั้งหมดจะต้องถูก abort ด้วยเช่นกัน distributed transaction สามารถทำงานได้กับหลายแอพพลิเคชันเซิร์ฟเวอร์และหลายฐานข้อมูล หมายความว่าคอมโพเน็นต์ที่ดีพลอยในแอพพลิเคชันเซิร์ฟเวอร์หนึ่ง ทำงานในทรานแซกชันเดียวกันร่วมกับอีกคอมโพเน็นต์ซึ่งถูกดีพลอยในอีกแอพพลิเคชันเซิร์ฟเวอร์ ในการทำงานบิซิเนสเมธอดหนึ่ง ๆ เครื่องหลายเครื่องอาจจำเป็นต้องทำงานร่วมกัน distributed transaction จึงอนุญาตให้หลายแอพพลิเคชันเซิร์ฟเวอร์ซึ่งอาจเป็นซอฟต์แวร์จากผู้พัฒนาคนละรายกันทำงานร่วมกันในทรานแซกชันได้

5.5.6 ทรานแซกชันใน .NET

ทรานแซกชัน มีทั้งแบบโลคอล ซึ่งเป็นทรานแซกชัน ซึ่งมีขอบเขตการจัดการภายใต้แหล่งข้อมูลตัวเดียวกัน (single- transaction-aware data resource) หรือเรียกว่า internal transaction และอีกแบบ เป็นทรานแซกชันแบบกระจาย (Distributed transaction) เป็นการจัดการทรานแซกชันจากหลากหลายแหล่งข้อมูลร่วมกัน เช่น SQL Server , MSMQ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.6.1 การจัดการทรานแซกชันแบบจัดการด้วยตนเอง (Manual Transaction)

Manual transaction ให้เราสามารถควบคุมขอบเขตการทำ transaction ได้อย่างชัดแจ้ง (explicit) แต่ผลเสียก็คือ เราต้องจัดการขอบเขตของ transaction ให้ดีและ ที่สำคัญคือ เราต้องจัดการควบคุม distributed transaction เอง จึงมีงานมากมายให้เราทำ เราต้องงควบคุมทุกๆ connection และ resource เพื่อให้เป็นไปตาม ACID ของ transaction ตัวอย่างของ transaction หนึ่งคือ ADO.NET สามารถกำหนด จุดเริ่มต้น, commit, rollback ของ transaction ได้เอง

5.5.6.2 การจัดการทรานแซกชันแบบการจัดการอัตโนมัติ (Automatic Transaction)

.Net มี MTS/COM+ ในการจัดการทรานแซกชันแบบการจัดการอัตโนมัติ ซึ่ง COM+ ใช้ Microsoft Distributed Transaction Coordinator (DTC) เป็นตัวจัดการทรานแซกชันทำให้ .NET แอพพลิเคชัน สามารถจัดการการใช้ทรัพยากรจาก ที่ต่างๆร่วมกันได้ เช่นจาก SQL SERVER , MSMQ , ORACLE ประสิทธิภาพของมันก็ขึ้นอยู่กับ DTC

ข้อดีคือ ง่ายในการเขียนโปรแกรม ไม่ต้องมากำหนดเองตรงๆ แต่ข้อเสียคือ ยากในการดูแลจัดการและเรามักจะละเลยไม่สังเกตถึงว่ามีการทำทรานแซกชันทั้งหมดฟังก์ชัน และในแบบแรกนั้น เราสามารถที่จะจัดการเกี่ยวกับ exception ที่ได้รับได้โดยตรง

ในแต่ละวิธีการจัดการทำทรานแซกชัน นั้นเป็นการแลกเปลี่ยนระหว่างประสิทธิภาพของแอพพลิเคชัน และ ความสามารถในการดูแลแอพพลิเคชัน

โดยสรุปรูปแบบการใช้งานทรานแซกชัน ใช้ Database Transaction ดูแลในเรื่องการจัดการทรานแซกชันใน สโตร์โพรซีเยอร์ (Stored procedure) Manual Transaction โดยการใช้ ADO.NET จะง่ายในการเขียน โค้ด และมีความยืดหยุ่นในการใช้ควบคุมทรานแซกชัน โดยทำได้โดยใช้คำสั่งภายนอกกำหนด begin transaction และ end transaction ส่วน Automatic Transaction จะมีประโยชน์มากกับงานประเภทที่มีการทำงานเกี่ยวกับทรานแซกชันจากหลายๆ แหล่งข้อมูลร่วมกัน

5.5.7 การจัดการทรานแซกชันใน เว็บเซอร์วิส

มีการจัดการทรานแซกชันในเว็บเซอร์วิสอยู่สองส่วนที่เราต้องจัดการ อันแรก คือ เราต้องการที่จะควบคุมทรานแซกชันเป็นส่วนหนึ่งของเว็บเซอร์วิสที่เราให้บริการ เนื่องจากเรามีการควบคุมระดับเน็ตเวิร์กเลเยอร์ภายใน ASP.NET แอพพลิเคชัน เราสามารถใช้ COM+ เข้ามาช่วยได้ ส่วนที่สอง เรามองไปยังการรวมไปถึงทรานแซกชันของแอพพลิเคชันที่เรียกใช้ เว็บเซอร์วิสนั้น ซึ่งไม่มีการควบคุมในระดับของเน็ตเวิร์กเลเยอร์ ทำให้เป็นปัญหาในเรื่องของ การควบคุมทรานแซกชันแบบกระจายขึ้น (Distributed Transaction Coordinator)

การเรียกใช้งานในเว็บเซอร์วิสนั้น ต้องทำการกำหนดค่า เว็บเมธอดแอททริบิวต์

```
[WebMethod(TransactionOption=TransactionOption.RequiresNew)]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาสนั้นต้องทำการ inherit มาจาก System.EnterpriseServices.ServicedComponent เป็นการสั่งให้ทำงานใน COM+ แล้ว COM+ จะทำการติดต่อกับ DTC ในการสร้าง distributed transaction และจัดการ resource ทั้งหมดที่ติดต่อ โดยมีการกำหนดตัวเลือก ไว้ 5 แบบคือ

1. Disabled - อ็อบเจ็กต์ที่สร้างขึ้นจะไม่มีทรานแซกชัน ใน COM+ ทรานแซกชัน แต่สามารถติดต่อ โดยตรงกับ DTC ในกรณีที่ต้องการการสนับสนุน

2. NotSupported - ไม่มีการทำทรานแซกชันเลย

3. Supported - ถ้าอ็อบเจ็กต์นั้นจะถูกรันในทรานแซกชัน context ของตัวที่สร้างมันขึ้นมาซึ่งมีการทำทรานแซกชัน, แต่ถ้าตัวมันเอง เป็นตัวเริ่ม (root - ไม่ได้ถูกใครสร้างมาอีกที) หรือ context ที่สร้างมัน ไม่ทำทรานแซกชัน มันก็จะไม่ทำทรานแซกชัน

4. Required - ถ้าอ็อบเจ็กต์นั้นจะถูกรันในทรานแซกชัน context ของตัวที่สร้างมันขึ้นมามีการทำทราแซกชัน, แต่ถ้าตัวมันเอง เป็นตัวเริ่ม (root - ไม่ได้ถูกใครสร้างมาอีกที) หรือ context ที่สร้างมันไม่ทำทรานแซกชัน มันก็จะทำทรานแซกชันใหม่

5. RequiresNew - บอกว่าอ็อบเจ็กต์ ต้องทำทรานแซกชัน ใหม่

ในการบ่งบอกถึง การ commit และ rollback นั้น เราทำได้โดยการ try /catch เพื่อตรวจจับ exception แต่เราก็สามารถให้มันทำการจัดการ commit, abort ให้เองได้ด้วย การเรียกใช้ System.EnterpriseServices.AutoComplete

การกำหนดค่านี้ จะเป็นการกำหนดให้ไปเรียกใช้เซอร์วิสของคอมพลัส โดยอัตโนมัติ โดยมีโหมดให้เรียกใช้ TransactionOption เหมือนกับ การที่เขียนคอมโพเนนต์ที่ สืบทอดมาจาก EnterpriseServices ถ้ามี exception ขึ้นทรานแซกชันจะทำการ roll back โดยอัตโนมัติ แต่ถ้าการทำงานนั้นอยู่ภายใต้ try...catch ทรานแซกชันจะคอมมิท ยกเว้นแต่ เราจะทำการ SetAbort ด้วยตัวเอง โดยเรียกใช้ผ่านอ็อบเจ็กต์ ContextUtil

5.5.7.1 ตัวอย่างการใช้งาน เว็บเซอร์วิสที่มีการใช้ทรานแซกชัน

มี 2 ฟังก์ชันคือ UpdateDoctor ทำการอัพเดทสถานะของหมอ และ InsertLog เพื่อทำการเก็บบันทึกการเปลี่ยนแปลง และมีฟังก์ชันที่เป็นเว็บเมธอด ให้ใช้งานซึ่งจะไปเรียก 2 ฟังก์ชันดังกล่าว

โดยหลังจากทำการอัพเดทสถานะแล้วก็จะบันทึกการเปลี่ยนแปลงเก็บไว้ด้วย แต่จะมีการโยน exception ออกมาทำให้เกิดการ roll back ผลลัพธ์ที่ได้นั้นจะส่งผลให้ ฟังก์ชัน UpdateDoctor เกิดการ roll back ด้วยเช่นกัน

```
[WebMethod(TransactionOption=TransactionOption.RequiresNew)]
public void UpdateStatus(int DoctorID, int StatusID)
{
        UpdateDoctor(DoctorID, StatusID);
        InsertLog(DoctorID, StatusID);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.7.2 การเรียกใช้เว็บเซอร์วิสที่มีทรานแซกชัน

จากส่วนแรก การเรียกใช้งานก็เป็นไปตามปกติเหมือนการเรียกใช้เว็บเซอร์วิสโดยทั่วไป ในหัวข้อนี้จะพูดถึงการเรียกใช้เว็บเซอร์วิสให้เป็นส่วนหนึ่งของทรานแซกชันภายใต้คอนเท็กซ์ของไคลเอนต์แอพพลิเคชัน จากเอกสารของไมโครซอฟท์กล่าวว่า

"เว็บเซอร์วิสเมธอดแต่ละอันจะเป็นส่วนหนึ่งของทรานแซกชันของตัวเองเท่านั้น เนื่องจาก เว็บเซอร์วิสเมธอด สามารถเป็นได้เพียง root ของทรานแซกชัน เท่านั้น "

หมายความว่า เราไม่สามารถที่จะสร้างแอพพลิเคชันที่เรียกใช้เว็บเซอร์วิสในทรานแซกชันคอนเท็กซ์ในโลคอลเมธอดได้

ปัญหาที่เกิดขึ้นก็คือ สมมุติว่า เรามี ASP.NET แอพพลิเคชัน มีการเรียกใช้เว็บเซอร์วิสเมธอดสองตัวที่เป็นทรานแซกชันเมธอด โดยที่ ASP.NET เรามีการกำหนดให้เป็นทรานแซกชันคอนเท็กซ์ด้วย ถ้าเว็บเมธอดอันที่สองเกิดการ roll back เมธอดแรกจะไม่ roll back ด้วย

ถึงแม้เราจะการกำหนดให้เป็นทรานแซกชันทั้งระดับของโลคอลใน ASP.NET และระดับเว็บเซอร์วิสก็ตามแต่ก็ไม่สามารถแก้ปัญหานี้ได้เนื่องจาก เว็บเซอร์วิสไม่ได้เป็นส่วนหนึ่งของทรานแซกชันที่โลคอลแอพพลิเคชัน

5.5.7.3 ปัญหาในการเรียกใช้ทรานแซกชันเว็บเซอร์วิส

โดยตัวนิยามแล้ว เว็บเซอร์วิสแล้วไม่ตรงตามมาตรฐานของ ACID มันเป็น loosely coupled มันจึงไม่สามารถควบคุมได้ จากแต่เดิมใน DTC (distributed transaction control) ใช้ข้อมูลของเน็ตเวิร์กเลเยอร์ในการควบคุมความถูกต้องของทรานแซกชัน เนื่องจากทุกๆเน็ตเวิร์กทราฟฟิกของไมโครซอฟท์ ทรานแซกชันแบบกระจาย จะรายงานความถูกต้องผ่านเน็ตเวิร์กไปยังผู้เรียก โดยตัว DTC จะใช้ข้อมูลนี้ในการพิจารณาตัดสินใจ roll back ในกรณีที่ต้องทำ ถ้าเกิดมีการล้มเหลวระดับเน็ตเวิร์ก ตัว DTC จะรับรู้แล้วจัดการตามสิ่งที่เกิดขึ้น

เรามี WSDL ในการอธิบาย อินเทอร์เฟสของรีโมทเซอร์วิส ซึ่งไม่ได้อธิบายเกี่ยวกับการเรียกใช้งานกับ DTC ซึ่งก็เป็นความจริงที่ว่า คอมพลัสไม่ได้จัดการกับทั้งหมดของโลก .NET และไม่ได้เกี่ยวข้องกับ WSDL

จากเหตุผลที่กล่าวมาข้างต้น เราจึงไม่สามารถที่จะรวมเว็บเซอร์วิสเข้ากับ ทรานแซกชันใน ASP.NET ได้ นี่เป็นอีกจุดอ่อนหนึ่งของเว็บเซอร์วิสโดยทั่วไป ไม่ได้เฉพาะการอิมพลีเมนต์เว็บเซอร์วิสของไมโครซอฟท์เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5.7.4 แนวทางการแก้ไขปัญหา

ในการพัฒนาแอพพลิเคชันที่มีการทำงานของทรานแซกชันด้วยเว็บเซอร์วิส ระบบจำเป็นต้องมี Transaction Monitor ซึ่งทำหน้าที่ในการดูแล และจัดการทรานแซกชันทั้งหมด (เช่น roll back ในกรณีที่มีปัญหาเกิดขึ้นกับทรานแซกชัน) แน่นอนว่า TM จะต้องทำงานร่วมกันกับแอพพลิเคชันอย่างใกล้ชิด เพื่อให้มีคุณสมบัติพิเศษ เช่น การล็อกข้อมูลที่ใช้งานร่วมกัน หรือการย้อนกลับเมธอดที่ทำงานไปแล้ว ในกรณีของการ roll back เป็นต้น ดังนั้น TM จึงจำเป็นต้องมีการทำงานกับระบบเว็บเซอร์วิสอย่างใกล้ชิดซึ่งส่งผลให้ความอิสระต่อกันในเชิงโครงสร้างของซอฟท์แวร์สูญหายไป

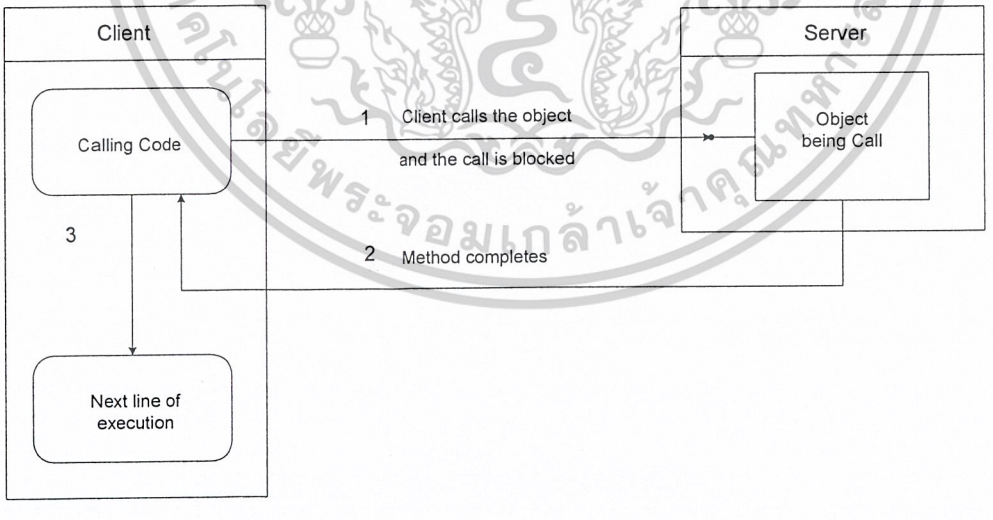
มีโซลูชั่นในการแก้ปัญหาที่ถูกนำเสนอออกมาใช้กับระบบงานแบบทรานแซกชันซึ่งผู้ที่เกี่ยวข้องกำลังพัฒนามาตรฐานสำหรับเว็บเซอร์วิส เพื่อให้รองรับระบบการจัดการทรานแซกชัน ตัวอย่างของมาตรฐานเหล่านี้ได้แก่ Transaction Authority Markup Language (XAML) และ Business Transaction Protocol (BTP) ซึ่งคาดว่าจะได้รับการแก้ไขโดยสมบูรณ์ราวปี 2547

อีกวิธีการแก้ปัญหาหนึ่งคือ สำหรับระบบที่มีการใช้งานภายในองค์กร ที่มีการใช้งาน .NET เพียงระบบเดียว เราก็สามารถเลือกใช้โปรโตคอลการสื่อสาร .NET Remoting ได้ ซึ่งถูกพัฒนามาจาก DCOM ซึ่งสามารถจัดการปัญหาดังกล่าวได้

5.6 การเรียกใช้เว็บเซอร์วิสแบบอซิงโครนัส(Asynchronous)

5.6.1 การประมวลผลแบบซิงโครนัส(Synchronous Processing)

การทำงานแบบซิงโครนัสนั้น ประกอบไปด้วย ฟังก์ชันและคอมโพเน็นต์ ที่มีการเรียกใช้งาน การเรียกใช้งานจะถูกบล็อกจนกว่าการทำงานของฟังก์ชันนั้นจะครบถ้วนเรียบร้อย รูปแบบการทำงานเป็นไปตามรูปที่ 5-6



รูปที่ 5-6 การติดต่อแบบซิงโครนัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.ไคลเอนต์ทำการเรียกใช้ ฟังก์ชันการทำงานไปยังอ็อบเจ็กต์คอมโพเน็นต์ การทำงานฝั่งไคลเอนต์ถูกบล็อก

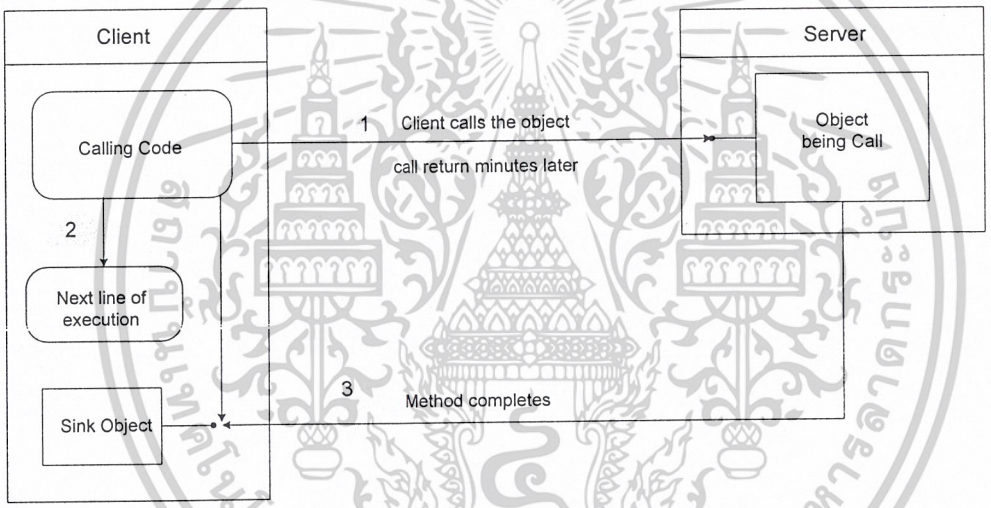
2.เมื่อฝั่งเซิร์ฟเวอร์ทำงานเสร็จส่งผลลัพธ์กลับมา

3.เมื่อได้รับผลลัพธ์ ไคลเอนต์ ทำการประมวลผลต่อได้

การเรียกใช้งานเว็บเซอร์วิส แบบซิงโครนัสเป็นการเรียกใช้งานตามแบบปกติทั่วไป ซึ่งง่ายในการจัดการ แต่มีข้อเสียคือเรื่องของประสิทธิภาพโดยฝั่งไคลเอนต์ต้องคอยรอผลลัพท์จากเว็บเซอร์วิส

5.6.2 การประมวลผลแบบอซิงโครนัส(Asynchronous Processing)

การเรียกใช้งานแบบอซิงโครนัส จะไม่มีการบล็อกจากคอมโพเน็นต์ฝั่งเซิร์ฟเวอร์ ไคลเอนต์จะรู้ว่าการประมวลผลเสร็จเรียบร้อยจาก polling หรือจาก การอินเทอร์รัพจากโปรแกรม โมเดลการทำงานแบบนี้ดังรูปที่ 5-7



รูปที่ 5-7 การติดต่อแบบอซิงโครนัส

1.ไคลเอนต์ทำการเรียกไปยังเซิร์ฟเวอร์

2.ไคลเอนต์สามารถที่จะประมวลผลโค้ดในบรรทัดต่อไปได้เลย โดยไม่ต้องรอผลลัพธ์

3.เมื่อฝั่งเซิร์ฟเวอร์ทำงานเสร็จจะส่งผลลัพธ์กับมาให้ ยังจุดที่สร้างขึ้นเพื่อรอรับผล(Sink point)

การเลือกโมเดลในการพัฒนาระบบ ไม่ได้เป็นเพียงเรื่องเล็กน้อย เราต้องทำการตัดสินใจจากความต้องการของระบบที่เราพัฒนาอย่างรอบคอบเพื่อตัดสินใจใช้งานการเรียกใช้แบบอซิงโครนัสหรือไม่ โดยอาจจะนำคำแนะนำต่อไปนี้เพื่อมาช่วยในการตัดสินใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การเรียกใช้งานแบบอซิงโครนัส จะช่วยขยายการรองรับการใช้งานของระบบ(scalability) ถ้าไคลเอนต์เป็น ASP.NET แอพพลิเคชัน หรือเป็น เว็บเซอร์วิสก็ตาม เนื่องจากการเรียกใช้แบบซิงโครนัสจะมีการบล็อกทำให้เทรดของ ASP.NET ที่ทำงานเกิดการรอ(stall) ส่งผลให้การร้องขออื่นๆมายังแอพพลิเคชัน ต้องเข้าคิว ส่งผลกระทบต่อประสิทธิภาพของระบบ การเรียกใช้แบบอซิงโครนัสแทนอย่างน้อยก็สามารถที่จะปลดปล่อยเทรด ASP.NET ที่ไม่ได้เรียกใช้เว็บเซอร์วิสในขณะนั้น
- การเรียกใช้งานแบบรีโมทโพรซีเยอร์คอลไปยังเว็บเซอร์วิสนั้นอาจจะใช้เวลานานก่อนการส่งผลลัพท์กลับ ในกรณีนี้การเรียกใช้แบบอซิงโครนัสจะมีประโยชน์ โดยฝั่งไคลเอนต์สามารถที่จะทำงานอย่างอื่นได้ ถ้าไม่จำเป็นต้องใช้ผลลัพธ์ในทันที
- ในกรณีที่ไคลเอนต์ต้องการเรียกใช้งานไปยังหลายเว็บเซอร์วิสพร้อมๆกันหลายที่ หรือที่เดียวกันแต่หลายฟังก์ชันการใช้งานแบบอซิงโครนัสจะเป็นประโยชน์มากเนื่องจากไคลเอนต์สามารถที่จะเรียกใช้งานไปยังทุกๆที่พร้อมๆกันได้เลย ไม่ต้องคอยรอผลลัพธ์ของแต่ละที่แล้วค่อยเรียกต่อแบบในกรณีของซิงโครนัส ทำให้เพิ่มประสิทธิภาพได้

5.6.3 การเรียกใช้แบบอซิงโครนัสใน .NET

.NET มีโมเดลการเรียกใช้งานแบบอซิงโครนัส มาให้พร้อมอยู่แล้วดังนี้

- .NET เฟรมเวิร์ก จะให้บริการเซอร์วิสที่จำเป็นในการรองรับโมเดลการใช้งานแบบอซิงโครนัส
- การเรียกใช้อยู่ที่โค้ดไคลเอนต์เรียกใช้งานในฟังก์ชันที่จำเป็นต้องใช้
- ไม่จำเป็นที่อ็อบเจ็กต์ที่ถูกเรียกต้องทำการเขียนโค้ดเพิ่มเติมเพื่อรองรับการใช้งาน ตัว CLR จะทำการจัดการให้
- รันไทม์ CLR ให้บริการตรวจสอบเกี่ยวกับความปลอดภัยในเรื่องของชนิดตัวแปร โดยการเรียกใช้ ดีลิเกต(delegate) ซึ่งก็เหมือนกับ พอยน์เตอร์ไปยังฟังก์ชันที่มีส่วนของการอธิบายตัวเองด้วยซึ่งต่างจากแต่ก่อนใน C++ พอยน์เตอร์ไปยังฟังก์ชันจะไม่บอกจำนวนพารามิเตอร์ที่ส่งผ่าน ชนิด แของตัวแปรพารามิเตอร์ และพอยน์เตอร์ฟังก์ชัน ก็เหมือนพอยน์เตอร์ทั่วๆไป (เป็น void)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.6.4 การเรียกใช้งานเว็บเซอร์วิสแบบอซิงโครนัส

การเรียกใช้เว็บเซอร์วิสแบบอซิงโครนัส ทำให้เราสามารถไปทำงานอย่างอื่นได้โดยไม่ถูก block เพื่อที่จะรอผลของ services นั้นๆ มีประโยชน์ในกรณีที่ services นั้นมีการเวลาประมวลผลนาน หรือในกรณีที่เราเรียกใช้ services จากหลายที่ เราส่งผลไปพร้อมกันทั้ง 3 ที่ โดยไม่ต้องมานั่งรอผลทีละที่

การทำงาน จะเรียกใช้ฟังก์ชันโดยเติม Begin นำหน้า เช่นฟังก์ชัน Hello() จะเป็น BeginHelllo() โดยที่ฝั่งเว็บเซอร์วิสเราไม่ต้องเขียนโค้ดเพื่อรองรับเลยการเรียกใช้กำหนดตามความต้องการทางไคลเอนต์ เราสามารถจบการทำงาน asynchronous ได้หลายวิธี

1. ทำการจบโดยเรียกใช้ EndHello() ซึ่งจะจบก็ต่อเมื่อ services นั้นทำงานเสร็จ ไม่งั้นก็จะถูกบล็อกรอเช่นกัน
2. ตรวจสอบจาก flag IAsncResult.IsComplete ซึ่งจะมีค่าเป็น true เมื่อทำงานเสร็จสิ้น

ตัวอย่างการใช้งาน

```
IAsyncResult AsyncResult;  //ประกาศตัวแปรแบบเพื่อรอรับผลของการเรียกใช้
AsyncResult = ObjectWS.BeginHello();  //เรียกใช้งานแบบอซิงโครนัส
while(AsyncResult.IsCompleted == false)
{
        //other job here
        // วนลูปทำการเช็คจากแฟล็กว่าเสร็จรึยัง
}
```

3. ในขณะเรียก BeginHello() ทำการใส่พารามิเตอร์ ดิลีเกต(delegate) ที่ต้องการให้ทำงาน เมื่อ เว็บเซอร์วิสทำงานเสร็จจะเรียกกลับมายังฟังก์ชันหรือดิลีเกตนั้นๆ

ขั้นตอนการใช้งาน

- ทำการประกาศ call back delegate โดยส่งค่าฟังก์ชันที่ต้องการให้เรียกกลับไปด้วย

```
AsyncCallback AsyncCallback = new AsyncCallback(MyCallBack);
```

- ฟังก์ชันที่จะทำการเรียกใช้ ต้องรับค่าพารามิเตอร์ IasyncResult ด้วย และภายในฟังก์ชันต้องมีการทำการเรียก END เพื่อจบการใช้งานฟังก์ชันที่เรียกไปยังเว็บเซอร์วิสด้วย private void MyCallBack(System.IAsyncResult AsyncResult
- การเรียกใช้งาน Begin Hello() ต้องมีการส่งผ่านค่าพารามิเตอร์ดังนี้

```
AsyncResult = objWS.BeginHello(parameter1,AsyncCallback,asycState);
```

Parameter1 เป็นพารามิเตอร์ที่ฟังชัน Hello ต้องการอาจมีหลายพารามิเตอร์ก็ได้

AsyncCallback ค่าของ delegate ที่ต้องการให้มีการเรียกกลับเมื่อทำงานเสร็จ

AsycState ใช้ในการส่งข้อมูลเกี่ยวกับข้อมูลของการเรียกแบบอซิงโครนัสไปยัง delegate สำหรับตัวแปรสองตัวท้ายนั้น ถ้าไม่ต้องการใช้งานก็ทำการส่งค่า null ไปแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.6.5 ปัญหาที่เกิดขึ้นกับการนำไปใช้งานจริง

จากการทำการทดสอบการใช้งานใน 3 รูปแบบข้างต้นสามารถทำได้ตามทฤษฎีในไคลเอนต์วินโดวส์ฟอร์ม แต่สำหรับไคลเอนต์ที่เป็น ASP.NET แอพพลิเคชัน ผลลัพธ์ที่ได้จากการทดลองผิดพลาดดังนี้

ผลการทดลองในรูปแบบของASP.NET เว็บฟอร์ม

1. แบบ END ฟังก์ชัน : ได้ผลลัพธ์ถูกต้อง
2. ทำการเช็คผลจากแฟล็ก AsyncResult.IsCompleted : รอผลที่ได้นานมาก จนสุดท้ายไม่มีผลขึ้นแสดง
3. ทำการเรียกกลับมายัง delegate : ไม่สามารถ เรียกไปยังฟังก์ชัน delegate ที่ต้องการได้ ผลที่ได้คือหน้าจอเดิมที่ทำการเรียก

จากการเข้าไปค้นหาข้อมูลจากอินเทอร์เน็ตได้ข้อมูลดังนี้ สาเหตุที่ไม่สามารถทำงานในเว็บฟอร์มได้เนื่องจาก ตัวไคลเอนต์ซึ่งในที่นี้คือ เว็บเซิร์ฟเวอร์จะไม่ส่งผลลัพธ์กลับมาจนกว่า การเรียกใช้งานทั้งหมดจะเสร็จเรียบร้อย หมายถึงขั้นตอนตั้งแต่เว็บแอพพลิเคชันถูกเรียกใช้งาน, เรียกไปยังเว็บเซอร์วิส จนกระทั่งเว็บส่งผลลัพธ์กลับมา วิธีที่เดียวจะเรียกใช้แบบอซิงโครนัสได้คือ เรียกใช้โดยตรงจากไคลเอนต์ที่ไม่ผ่านเว็บเซิร์ฟเวอร์ ซึ่งสามารถทำได้โดยไคลเอนต์สคริปต์ เช่น java script ซึ่งต้องมีการกำหนด time out ให้นานพอสมควร

การเรียกใช้งานแบบอซิงโครนัสโดยไคลเอนต์เป็นจาวานั้นไม่สามารถทำได้เนื่องจากตัวจัดการการเรียกใช้แบบนี้ฝังอยู่ในอินฟราสตรักเจอร์ของ CLR ใน .NET เราจึงไม่สามารถเรียกใช้ความสามารถนี้ในจาวาได้

เนื่องจากมันไม่สามรถเรียกใช้จากค่ายจาวาได้ รวมทั้งค่อนข้างยุ่งยากในการเรียกใช้งานกรณีที่ไคลเอนต์เป็นเว็บฟอร์ม ดังนั้นจึงไม่เหมาะในการนำมาพัฒนา เราจะทำการศึกษาการทำงานแบบอซิงโครนัส โดยใช้เทคโนโลยีของคิว (MSMQ) ต่อไป

การแก้ปัญหาอ้างอิงจาก

www.dotnet247.com/247reference/msgs/21/106614.aspx

www.dotnet247.com/247reference/msgs/24/123480.aspx

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

ADO.NET

6.1 สถาปัตยกรรมของ ADO.NET

เนื่องจากในการพัฒนาโปรแกรมด้านฐานข้อมูลแต่เดิมนั้นได้ใช้เทคโนโลยี ADO (ActiveX Data Object) เป็นเครื่องมือในการติดต่อและจัดการข้อมูลในแบบ connected database คือจะเปิดการเชื่อมต่อกับฐานข้อมูลไว้ตลอดช่วงเวลาใช้งานซึ่งจะทำให้การเพิ่ม ลบ แก้ไขหรือการกระทำใดๆก็ตามที่เกี่ยวข้องกับข้อมูลจะเป็นการกระทำกับข้อมูลจริงในไฟล์ฐานข้อมูลเป็นจำนวนมากและพร้อมๆกันหรือการประมวลผลข้อมูลที่มีปริมาณมากจนทำให้ระยะเวลาที่ต้องเชื่อมต่อกับ server นานเกินไป ซึ่งจะทำให้อัตราการสิ้นเปลืองทรัพยากรบนเครื่อง server สูง ส่งผลให้ประสิทธิภาพโดยรวมของระบบลดลงได้ ดังนั้นใน Visual Studio .NET จึงได้นำเทคโนโลยีตัวใหม่ล่าสุดคือ ADO.NET ที่พัฒนาจาก ADO มาใช้จัดการข้อมูลแบบ disconnected database คือสามารถจัดการข้อมูลในฐานข้อมูลโดยไม่ต้องเปิดการเชื่อมต่อไว้ตลอดเวลาเหมือนใน ADO เนื่องจากข้อมูลจะถูกอ่านจากฐานข้อมูลมาเก็บไว้ในหน่วยความจำก่อน จากนั้นจะตัดการเชื่อมต่อกับฐานข้อมูลทำให้การกระทำใดๆหลังจากนี้จะเป็นการกระทำกับข้อมูลในหน่วยความจำเท่านั้น เมื่อเสร็จสิ้นกระบวนการแล้วจึงจะบันทึกกลับลงฐานข้อมูล ซึ่งจะช่วยประหยัดทรัพยากรของระบบและเพิ่มประสิทธิภาพการทำงานของเซิร์ฟเวอร์ (ที่จริงใน ADO เองก็มีความสามารถในการทำ disconnected database ได้ แต่โปรแกรมเมอร์จะต้องเป็นผู้ควบคุมด้วยตนเองว่าจะให้มีการเชื่อมต่อหรือยกเลิกการเชื่อมต่อเมื่อใด ซึ่งต่างกับใน ADO.NET ที่จะตัดการเชื่อมต่อให้โดยอัตโนมัติ เมื่อไม่มีความจำเป็นต้องใช้ฐานข้อมูลในแบบออนไลน์ และจะเปิดการเชื่อมต่อให้ใหม่เมื่อจำเป็นต้องทำงานร่วมกับฐานข้อมูลจริง)

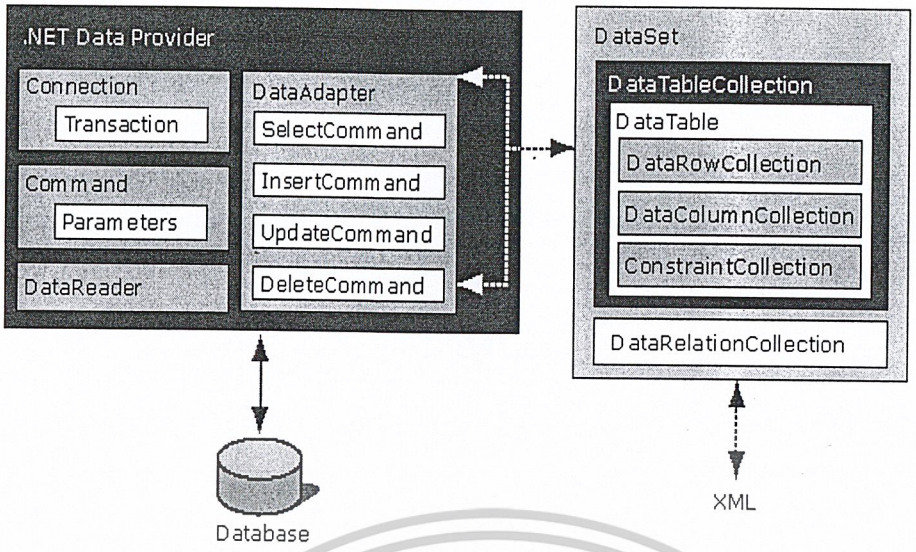
ในอีกประเด็นหนึ่งคือ ปัจจุบันการส่งข้อมูลเป็นลักษณะผ่านเว็บมากขึ้น ภาษา XML มีบทบาทมากขึ้น โดยส่วนของ ADO.NET ก็จะมีความสามารถในการรับส่งข้อมูลเป็น XML อีกด้วย

องค์ประกอบของคอมโพเนนต์ใน ADO.NET

ADO.NET ประกอบด้วย 2 คอมโพเนนต์หลักคือ ดาต้าเซ็ต (DataSet) และ ดอทเน็ต ดาต้า โพไวด์เดอร์ (.NET data provider) ซึ่งประกอบด้วย กลุ่มของ คอมโพเนนต์คือ คอนเน็กชั่น (Connection), คอมมาน (Command), ดาต้ารีดเดอร์ (DataReader) และ ดาต้าอเด็บเตอร์ (DataAdapter) ดังรูปที่ 6-1

ADO.NET DataSet เป็นคอมโพเนนต์หลักในการเชื่อมต่อข้อมูลแบบดีสคอนเน็กเตต (Disconnected), ดาต้าเซ็ต (DataSet) ถูกออกแบบมาให้มีความเป็นอิสระจากแหล่งข้อมูลด้วยเหตุนี้ มันจึงสามารถใช้ได้ในดาต้าซอร์ส (Data Source) หลายประเภท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-1 สถาปัตยกรรม ADO.NET

ดาต้าเซ็ต ประกอบด้วยกลุ่มของ ดาต้าเทเบิล (DataTable) ซึ่ง เทเบิล (Table) ที่ว่านี้ก็เกิดมาจากการส่ง Query ไปดึงมาจาก ฐานข้อมูล โดยดาต้าเทเบิลก็จะประกอบไปด้วย DataRow, DataColumn รวมถึง primary key, foreign key, ข้อจำกัดต่างๆ พวกข้อมูล (Data) Integrity และ ความสัมพันธ์ของข้อมูล (DataRelation) ใน ดาต้าเทเบิลเอง ในส่วนของการส่งข้อมูล ดาต้าเซ็ต ได้ใช้ XML เป็นตัวกลางในการส่งข้อมูลระหว่าง Tier ทำให้สามารถนำไปใช้กับการทำเว็บเซอร์วิส ได้

ส่วนหลักอีกส่วนหนึ่งคือ .NET Data Provider ซึ่งสร้างมาเพื่อการประมวลผลข้อมูลแบบ รวดเร็ว และแบบส่งไปข้างหน้าอย่างเดียว (forward-only) และเป็นการเข้าถึงข้อมูลแบบอ่านอย่างเดียว (read-only) โดยประกอบด้วยส่วนย่อยๆ ดังต่อไปนี้คือ

1. Connection ทำหน้าที่ติดต่อกับฐานข้อมูล SQL Server หรือ OLE DB เช่นเดียวกับออบเจ็ค Connection ใน ADO

2. Command จะใช้ร่วมกับออบเจ็คพารามิเตอร์ (Parameter) ในการประมวลผลคำสั่ง SQL, Query หรือ Stored Procedure ในแบบที่สามารถส่งผลลัพธ์คืนกลับมาหรือจะไม่ส่งคืนก็ได้เช่นเดียวกับการใช้ออบเจ็ค Command ใน ADO

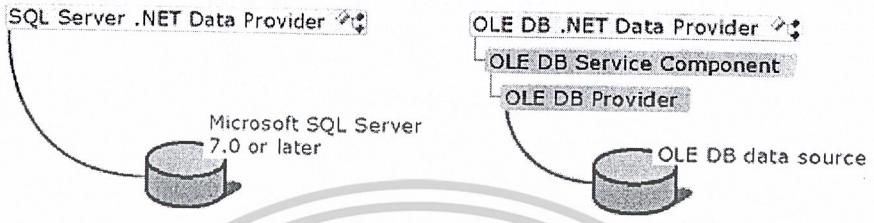
3. DataReader จะใช้สำหรับการดึงข้อมูลแบบทีละเรคอร์ดจากฐานข้อมูลบนเซิร์ฟเวอร์ โดย server cursor จะเป็นแบบ forward-only คือเดินหน้าอย่างเดียวถอยหลังไม่ได้และไม่สามารถแก้ไขข้อมูลได้ด้วย (read-only) ทำให้ความเร็วในการเข้าถึงข้อมูลสูง

4. DataAdapter จะเป็นเหมือนสะพานเชื่อมต่อระหว่างดาต้าเซ็ต และ ดาต้าซอร์ส โดยการทำงานคือเมื่อมีการเปิด Connection และกำหนด Query String SQL แล้ว DataAdapter ใช้ Command Object เพื่อประมวลผลคำสั่งภาษา SQL และทำการดึงข้อมูลลงมาที่ดาต้าเซ็ต เพื่อใช้งานต่อไป โดย .NET Framework มี .NET data provider ให้ใช้ 2 แบบคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. SQL Server .NET data provider ใช้ติดต่อและทำงานร่วมกับไฟล์ฐานข้อมูล Microsoft SQL Server ประกอบด้วยคลาส SqlConnection, SqlCommand, SqlDataReader และ SqlDataAdapter

2. OLE DB .NET data provider ใช้ติดต่อและทำงานร่วมกับไฟล์ฐานข้อมูลเช่น Microsoft Access, Oracle เป็นต้น ประกอบด้วยคลาส OleDbConnection, OleDbCommand, OleDbDataReader และ OleDataAdapter



รูปที่ 6-2 เปรียบเทียบระหว่าง SQL Server .NET Data Provider และ OLEDB .NET Data Provider

ก็คือ ถ้าใช้ SQL Server .NET Data Provider ต้องใช้กับ Microsoft SQL Server 7.0 หรือใหม่กว่าเท่านั้น โดย คลาสของ SQL Server .NET Data Provider กำหนดโดย namespace System.Data.SqlClient สำหรับเวอร์ชันที่เก่ากว่าก็ใช้ OLE DB .NET Data Provider แทน

ในการเลือกใช้ DataReader หรือ ดาต้าเซ็ตใน แอปพลิเคชั่นชัน นั้น ควรพิจารณาถึงชนิดของหน้าที่การทำงานของแอปพลิเคชัน โดยจะเลือกใช้ดาต้าเซ็ต ก็ต่อเมื่อ

- มีการส่งข้อมูลระหว่างเทียร์ หรือมีต้องมีการใช้ XML Web service
- มีการติดต่อข้อมูลแบบไดนามิค โดยผูกติดกับวินโดว์ฟอร์ม (Window From)หรือ การรวมข้อมูลที่เกี่ยวข้องกันมาใช้งานจากแหล่งข้อมูลหลายๆที่
- ให้มีการเก็บข้อมูลลงในแคชของเครื่อง
- มีการประมวลผลข้อมูลโดยไม่ต้องเปิดการเชื่อมต่ออยู่ตลอดเวลา ซึ่งทำให้ไคลเอนต์อื่นๆ สามารถเข้ามาใช้งานฐานข้อมูลได้มากขึ้น

โดยถ้าไม่มีความจำเป็นเหล่านี้ ก็สามารถใช้ DataReader แทนได้ โดยการทำงานจะเป็นแบบ Forward-Only and read-only คือเป็นการส่งข้อมูลให้อย่างเดียว และส่งไปข้างหน้าเท่านั้น และเป็นแบบอ่านได้อย่างเดียว คือ Update ข้อมูลกลับไปไม่ได้ โดยการใช้ DataReader จะเป็นการประหยัดหน่วยความจำลงไปได้มาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปข้อเปรียบเทียบระหว่าง ADO และ ADO.NET

1. ในการใช้งานกับ Web ซึ่งเป็นลักษณะ ดึงข้อมูลเว็บเพจนั้นเสร็จแล้วก็ทำการตัดสายอัตโนมัติ ซึ่งส่งผลทำให้การทำงานเร็วขึ้น ใช่ว่า ADO จะไม่มี แต่ ADO.NET ทำให้มันสะดวกขึ้นโดยทำโดยอัตโนมัติแทนที่โปรแกรมเมอร์ต้องไปควบคุมทุกขั้นตอน

   หมายเหตุ (ADO.NET จะทำ snapshot ของฐานข้อมูลที่เรากำลังติดต่ออยู่เก็บไว้ในหน่วยความจำของเครื่อง โดยการทำ Snapshot นั้นไม่ได้ทำทั้งฐานข้อมูล แต่ทำเฉพาะส่วนที่เรา query เท่านั้น โดยเมื่อดึงข้อมูลเสร็จก็จะทำการ Disconnect ทันที)
2. ส่วนการติดต่อกับฐานข้อมูลและส่วน การจัดการ client cursor (ภาษา ADO เรียกว่า Recordset) นั้นจะเป็นอิสระต่อกันโดยสิ้นเชิง ทำให้ทั้งสองส่วนพัฒนาได้โดยอิสระ
3. ใน ADO เมื่อก่อน Recordset ไม่สามารถพัฒนาได้มากนักเพราะว่าต้องผูกติดกับ Database Server ที่ตัวเองติดต่อด้วย ดังนั้นเพื่อให้สามารถรองรับได้กับ Database ทุกประเภท ทาง Recordset จึงจำเป็นต้องถูกออกแบบมาให้ทำงานเป็นแบบกลางๆ ทำให้ความสามารถด้อยลงไป ดังนั้น ใน ADO.NET จึงแยกในส่วนการติดต่อฐานข้อมูลออกมา โดยจะมีหนึ่งคลาสต่อหนึ่งต่อหนึ่งประเภทฐานข้อมูล เช่น ถ้าเราต่อกับ Oracle ต้องใช้คลาส หนึ่ง และถ้าต่อกับ Access ต้องใช้อีกคลาสไม่เหมือนกับ ADO ที่ใช้ OLEDB , คลาส ADODB.Connection เพื่อติดต่อกับทุก Database ข้อดีข้อการแยกก็จะทำให้การเชื่อมต่อกับ DBMS แต่ละยี่ห้อเป็นไปได้โดยอิสระและสามารถดึงเอาประสิทธิภาพมาใช้ได้อย่างเต็มที่
4. ส่วนของ Client Cursor นั้นปรับปรุงใหม่ แทนที่จะเป็น Recordset ซึ่งรองรับเพียงแค่ตารางเดียวเหมือนเมื่อก่อน แต่ใน ADO.NET สามารถรองรับทั้งฐานข้อมูล และแต่ละเทเบิล สามารถมี reference integrity ด้วย
5. การส่งผ่านข้อมูลระหว่าง ADO.NET และ DBMS นั้น แทนที่จะส่งโดยใช้ DCOM ซึ่งจะมีปัญหากับ Firewall ทำให้การสื่อสารยุ่งยาก แต่ ADO.NET นั้นจะแปลงข้อมูลอยู่ในรูปของ XML ก่อนแล้วค่อยส่งออกไป ทำให้ใช้พอร์ต (port) ของเว็บมาตรฐาน ทะลุกำแพง firewall ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

Microsoft .NET Technology

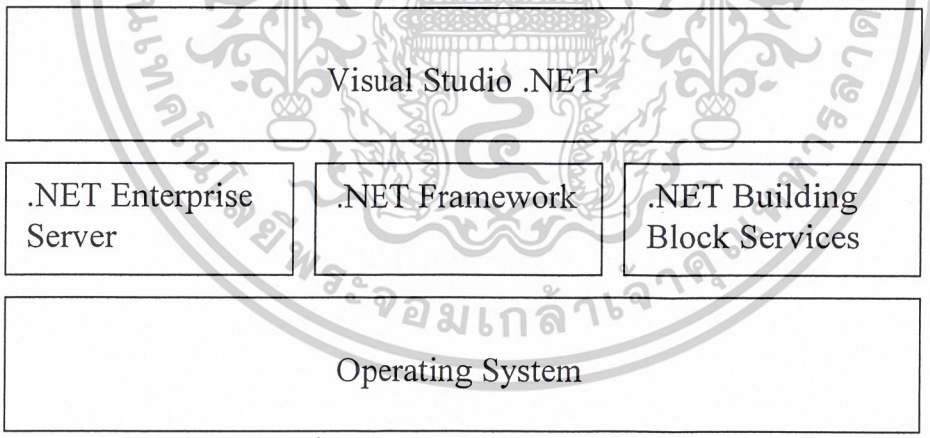
7.1 รู้จักกับ Microsoft .NET

Microsoft .NET หรือเรียกสั้นๆว่า ดอทเน็ต (.NET) เป็นเทคโนโลยีใหม่ล่าสุดที่พัฒนาขึ้นโดยบริษัทไมโครซอฟท์ .NET คือ แพลตฟอร์มในการพัฒนาซอฟต์แวร์สำหรับระบบปฏิบัติการ Windows โดยถือเป็นก้าวสำคัญในการพัฒนาซอฟต์แวร์ยุคใหม่ซึ่งนำเสนอหลักการพัฒนาซอฟต์แวร์ด้วยภาษาอะไรก็ได้ที่เราถนัด และสามารถเรียกใช้งานโปรแกรมที่เขียนด้วยภาษาอื่นๆได้อย่างกลมกลืน ใน .NET มีภาษาโปรแกรมใหม่ๆเกิดขึ้นเช่น Visual Basic .NET, C# .NET, C++ .NET, J# .NET หรือแม้กระทั่ง COBOL .NET เป็นต้อน

ทุกภาษาที่สนับสนุน .NET จะอยู่ภายใต้กฎเกณฑ์มาตรฐานเดียวกันที่เรียกว่า Common Language Specifications (CLS) และโครงสร้างพื้นฐานตั้งแต่ชนิดข้อมูล, ชุดคำสั่งพื้นฐานเช่นการจัดการ I/O, ฐานข้อมูล ที่อยู่ภายใต้มาตรฐานเดียวกัน ทำให้เราสามารถพัฒนาซอฟต์แวร์โดยใช้ภาษาโปรแกรมได้หลายภาษา

7.2 .NET Platform

องค์ประกอบของแพลตฟอร์ม .NET แบ่งออกได้เป็น 3 ชั้น



รูปที่ 7-1 องค์ประกอบ .NET Platform

ชั้นระบบปฏิบัติการ (Operating System)

ชั้นถัดมาแบ่งออกเป็น 3 ส่วน ได้แก่

- .NET Enterprise Server : SQL Server 2000, BizTalk Server, Exchange 2000
- .NET Framework : เป็น Framework ที่สำคัญในการพัฒนา .NET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

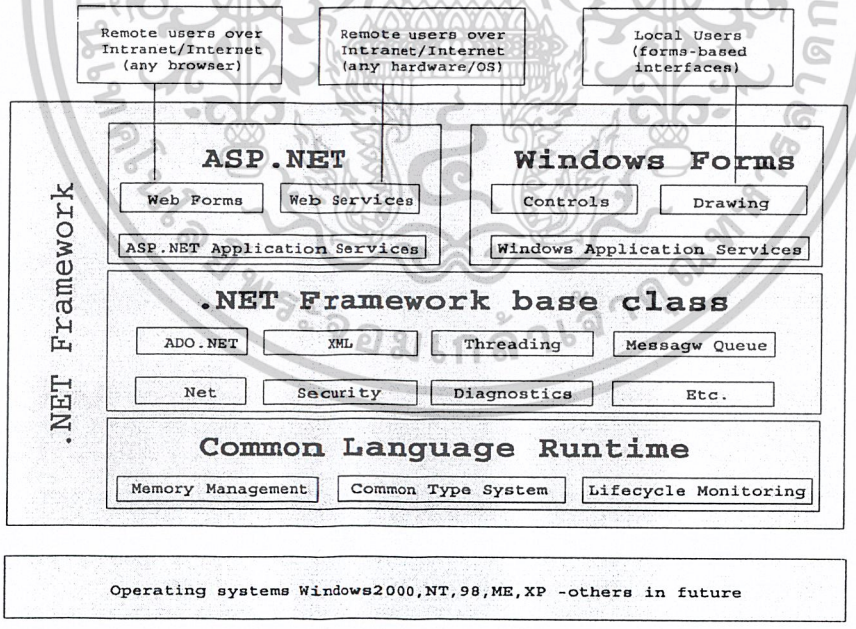
- .NET Building Block Services : Microsoft passport, Notification and Messaging, Personalization , XML Store, Calendar, Directory and Search

ชั้น Visual Studio .NET เครื่องมือในการพัฒนา .NET

7.3 .NET Framework

.NET Framework มีองค์ประกอบ ดังนี้

1. Common Language Runtime (CLR) มีหน้าที่ในการจัดเตรียมบริการต่างๆที่เกี่ยวข้องเพื่อให้โปรแกรมที่ขอใช้บริการสามารถทำงานได้
2. .NET Framework Class Library (Unified programming classes) คือไฟล์ไลบารีที่ทำหน้าที่จัดเก็บ และรวมรวมข้อมูลที่จำเป็นต้องเรียกใช้งาน ในการพัฒนาโปรแกรม
3. ASP.NET เป็น Web application model ซึ่งประกอบไปด้วย กลุ่มของ control และ infrastructure ที่ใช้ในการสร้าง ASP Web applications , ASP.NET ประกอบไปด้วยส่วนของ control ซึ่งรวมถึงส่วนของ HTML ด้วย ส่วนของ control รันบนเซิร์ฟเวอร์ ส่วนHTML รันบน browser บนเซิร์ฟเวอร์นั้นส่วน control จะถูกเขียนในรูปแบบของ object-oriented programming , ASP.NET ยังให้บริการส่วน infrastructure ด้วยเช่น session state management และ process recycling ยิ่งไปกว่านั้น ASP.NET ก็ใช้คอนเซปนี้ในการพัฒนาจาก ซอร์ฟแวร์ เป็น เซอร์วิส โดยผ่าน กลไกสำคัญคือ XML เราสามารถเขียน business logic และใช้ ASP.NET infrastructure เพื่อให้บริการผ่าน SOAP



รูปที่ 7-2 องค์ประกอบ .NET Framework

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3.1 Common Language Runtime (CLR)

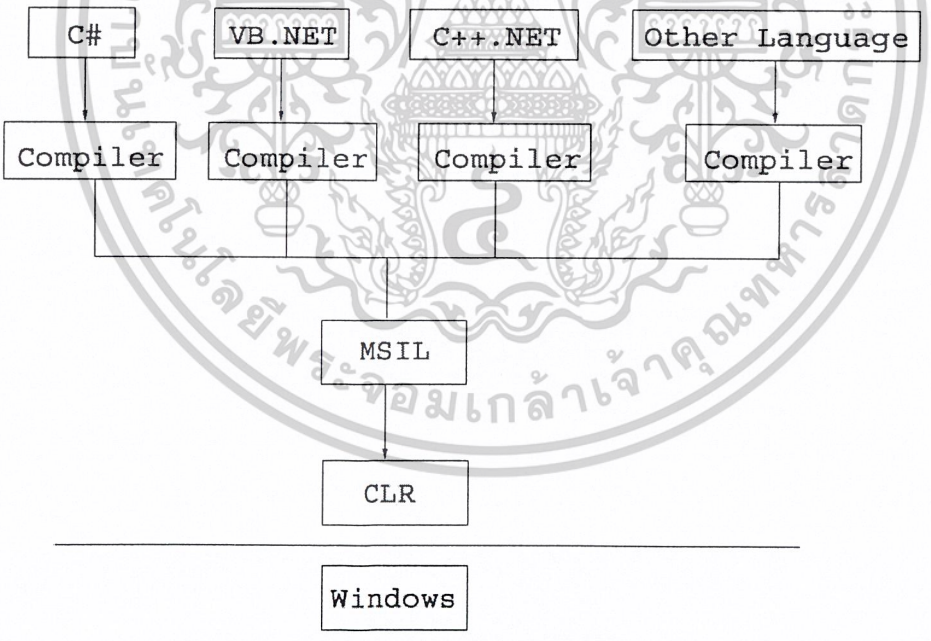
CLR จะทำหน้าที่เป็นตัวกลางในการจัดเตรียมบริการและทรัพยากรสำหรับรองรับการประมวลผล และการทำงานของโปรแกรมประยุกต์ที่ทำงานบนเทคโนโลยีของ .NET เช่น การจัดการหน่วยความจำ, ความปลอดภัยในการเข้ารหัสโปรแกรม ด้วยความสามารถของ CLR ทำให้เราสามารถพัฒนาโปรแกรมโดยไม่ต้องขึ้นกับระบบปฏิบัติการ

การทำงานของโปรแกรมนั้น เริ่มจากคอมไพเลอร์ของแต่ละภาษาจะคอมไพล์โค้ดให้เป็นแบบ Microsoft's Intermediate Language (MSIL) หรือเรียกสั้นๆว่า IL Code ซึ่งจะมีลักษณะคล้ายภา Assembly จาก IL Code ก็แปลงเป็นโปรแกรมที่รันโดย CLR อีกทีหนึ่ง

นักพัฒนาสามารถที่จะเลือกได้ว่า เราจะ build แอพพลิเคชั่นไปอยู่ในรูปของ .exe หรือ MISL ซึ่งจะกลาย เป็น Just-In-Time (JIT) คือเมื่อเราได้แอพพลิเคชั่นในรูปแบบของ MISL แล้ว เมื่อรันโปรแกรมใช้งานจริง มันจะถูก คอมไพเลอร์ JIT ทำการคอมไพล์โค้ด MSIL ในส่วนที่ต้องการใช้ไปเป็น Native Code อีกทีซึ่งนำไปให้เครื่องรันต่อ หากมีการใช้โค้ดในส่วนเดิมอีกก็ไม่ต้องมีการคอมไพล์ซ้ำ การแปลงโค้ด MSIL ไปเป็น Native Code ประโยชน์ที่เราจะได้คือ

1. เราสามารถแปลงไปเป็น Native Code ที่เหมาะสมกับระบบปฏิบัติการนั้นรันอยู่

2. เราสามารถได้ Native Code ที่ใช้ความสามารถของ CPU ได้อย่างเต็มที่ เช่นใน CPU P4 เราสามารถใช้คำสั่งในส่วนของ SSE2 ได้



รูปที่ 7-3 รูปแบบการคอมไพล์โค้ดไปเป็น IL Code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Managed Code เป็นโค้ดที่คอมไพล์ให้ทำงานใน .NET Framework

Native Code เป็นโค้ดที่คอมไพล์เป็นภาษาเครื่องโดยตรง ๆไม่ต้องผ่านตัว CLR ในการรันโปรแกรม

Unmanaged Code เป็นโค้ดที่ไม่ได้คอมไพล์เพื่อให้ทำงานใน .NET Framework เช่น COM Object ใน version ก่อนหน้านี้ ซึ่ง .NET ก็ยังสามารถรันโค้ดพวกนี้ได้ แต่ไม่สามารถใช้ความสามารถของ .NET ได้อย่างเต็มที่

7.3.1.1 Memory Management

สำหรับ .NET แล้วจะมี Garbage Collector (GC) ซึ่งทำหน้าที่คอยรวบรวมหาหน่วยความจำที่ถูกทิ้งไว้เป็นขยะ แต่ไม่มีใครอ้างอิงเรียกใช้แล้ว โดยการทำงานจะแตกต่างจากใน VB Runtime รุ่นเก่าที่ให้ Object รับผิดชอบในการจัดการ Garbage Collector เป็นผู้ดูแลจัดการทั้งหมด โดย GC จะเรียก method ชื่อ Object.Finalize โดยอัตโนมัติเมื่อ object เลิกใช้งาน

การจัดการหน่วยความจำที่ดีกว่าของ .NET เกิดขึ้นเมื่อ object ถูกสร้างขึ้นมา CLR จะจัดการกันพื้นที่ส่วนหนึ่งไว้สำหรับโปรแกรมที่เรียกว่า Heap โดยเริ่มต้นจากพื้นที่หน่วยความจำที่ว่าง จากนั้นก็อ้างอิงชี้ตำแหน่งไปยังส่วนบนสุดของหน่วยความจำ เมื่อหน่วยความจำถูกใช้แล้วยกเลิกไปเรื่อยๆ จะเหลือพื้นที่ว่างเป็นช่วงๆ ซึ่ง GC จะเข้ามาทำการ Compact หน่วยความจำเสียใหม่ ลองนึกถึงหลักการทำงานแบบเดียวกันกับตอนที่ทำ Defragment บนฮาร์ดดิสก์

7.3.1.2 Common Type System

ใน .NET จะมี common Type System ที่จะให้ทุกภาษามี Types ที่เหมือนกัน เป็นมาตฐาน โดยที่ทุก Types ที่สนับสนุนโดย Common Types System นั้น จะสืบทอดคุณสมบัติมาจาก System.object ดังนั้นคุณจะพบว่า Object ทั้งหลายส่วนมากแล้วจะสนับสนุน Method เหล่านี้คือ

Equal(Object) = Boolean

GetHashCode() = Int32

GetType () = Type

Tostring () =String

ข้อดีของ Common Language Runtime

- ไม่มีปัญหาเรื่องการรัน บน MS Windows Platform ต่างๆ เนื่องจาก CLR จะทำการตรวจสอบระบบปฏิบัติการให้โดยอัตโนมัติ และทำการจำลองสภาพให้เหมาะสมกับการทำงานของโปรแกรมโดยอัตโนมัติเช่นเดียวกัน

- ไม่ต้องสนใจเรื่อง Registry การเข้าถึง คอมโพเนนต์ ต่างๆที่เกี่ยวข้องจะอยู่ในความดูแลของ CLR ทั้งหมด เราจำเป็นต้องรู้แค่ว่าใช้ Namespace ตัวใดในการอ้างอิงถึงเท่านั้น

- รองรับการพัฒนาจากหลายภาษา ในปัจจุบันเราสามารถใช้หลายภาษามาทำงานร่วมกันได้ด้วยเทคโนโลยีของ COM แต่อย่างไรก็ตาม เนื่องจากโปรแกรมยังผูกติดกับภาษาอยู่ ความเร็วของโปรแกรมที่ได้ก็จะแตกต่างกัน ขึ้นอยู่กับความสามารถของ Compiler ภาษานั้นๆ Microsoft Common Language Specification จะทำให้ compiler ทุกตัวที่ใช้ .NET ทำการแปลภาษาไปยังเป้าหมายเดียวกัน คือ การทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับ CLR ได้อย่างไร้ปัญหา ส่งผลให้ โปรแกรมที่ได้จากและภาษามีผลลัพธ์ที่เท่าเทียมกันเมื่อทำงานอยู่ภายใต้การดูแลของ CLR

- นำ Source code มาใช้งานใหม่ได้ COM ยังถูกจำกัดการ reuseในวงแคบเนื่องจากมีการผูกติดกับระบบปฏิบัติการ .NET แยกออกได้ 3 รูปแบบได้แก่

1. เราสามารถเขียน คลาส ด้วยภาษาหนึ่ง สามารถเรียกใช้ผ่านอีกภาษาหนึ่งได้

2. คลาสสามารถ Inherit ข้ามภาษาได้

3.เว็บเซอร์วิส เป็นอีกรูปแบบหนึ่งโดยที่เสนอบริการให้แก่ทุกภาษาที่อ้างอิง CLR ได้

4. ชนิดของข้อมูลมีประสิทธิภาพ จากข้อผิดพลาดเกี่ยวกับตัวแปรเช่น ประกาศตัวแปรขนาด 10 ไบต์ แต่ทำงานจริงกลับมีขนาด 20 ไบต์ ถ้าไม่มีการเขียนโปรแกรมดักจับ โปรแกรมจะทำงานผิดพลาด ใน .NET สิ่งเหล่านี้จะไม่เกิดขึ้น CLR จำตรวจสอบการทำงานของ code อย่างใกล้ชิดก่อนที่จะประมวลผลขั้นสุดท้าย CLR จะไม่อนุญาตให้โปรแกรมใช้ตัวแปรที่มีการผิดพลาดเหล่านั้น

5. การ Debug ที่มีประสิทธิภาพ CLR จะทำการแยกส่วนของโปรแกรมเป็นส่วนๆทำให้การตรวจสอบข้อผิดพลาดทำได้ง่ายสะดวกและรวดเร็ว การแก้ไขสามารถแก้ไขเฉพาะส่วนที่ผิดพลาดได้โดยไม่รบกวนส่วนอื่นๆของโปรแกรม และเนื่องจาก CLR จัดการกับทุกภาษาด้วยกัน ข้อผิดพลาดต่างๆที่แสดงออกมาในแต่ละภาษาจะอยู่ในรูปแบบเดียวกัน

6. ระบบจัดสรรทรัพยากรที่ดีขึ้น การดำเนินการที่เกี่ยวข้องกับทรัพยากรต่างๆ ระหว่างโปรแกรมไม่ว่าจะเป็นระบบการจัดการไฟล์ระบบการเชื่อมต่อ หรือพื้นที่ว่างของหน้าจอ เป็นต้น จะถูกดูแลจัดการอย่างใกล้ชิด โดย CLR ไม่สามารถมีโปรแกรมตัวใดตัวหนึ่งครอบครองทรัพยากรไว้แต่เพียงผู้เดียว

7. การแจกจ่ายโปรแกรมง่ายขึ้น การติดตั้งโปรแกรมแบบเดิม โปรแกรมหนึ่งๆมีไฟล์เป็นจำนวนมาก ,ต้องมีการลงทะเบียนใน registry ,การสร้าง short cut , การจัดเตรียมระบบที่เกี่ยวข้อง ใน .NET การติดตั้งโปรแกรมเป็นไปโดยง่าย เนื่องจาก CLR จะทำการแยกโปรแกรมออกเป็นส่วนๆ การติดตั้งโปรแกรมจึงเป็นการ คัดลอก ไฟล์ที่เกี่ยวข้องมาไว้บน hard disk เท่านั้น ในทางกลับกัน การยกเลิกการติดตั้ง ก็เพียงลบไฟล์เหล่านั้นออก

8. ระบบความปลอดภัยที่ดีขึ้น ระบบรักษาความปลอดภัยใหม่นี้จะทำการตรวจสอบพฤติกรรมการทำงานของโปรแกรมอย่างใกล้ชิด อย่างในระบบเดิมนั้น การทำงานของ ภาษาสคริปต์ต่างๆหรือโปรแกรมที่มาจาก Internet อาจเป็นโปรแกรมที่สร้างช่องโหว่ให้กับระบบได้

7.3.2 .NET Framework Class Library

ไฟล์ไลบรารีที่ทำหน้าที่จัดเก็บและรวมรวมข้อมูลที่จำเป็นต้องเรียกใช้งาน ในการพัฒนาโปรแกรม ซึ่งไฟล์เหล่านี้ เป็นไลบรารีร่วมกันในทุกภาษา ส่งผลให้เราสามารถ inheritance, error handling, debugging ข้ามภาษาได้ และการที่ .NET Framework เสนอบริการสำเร็จรูปผ่าน Class Library เหล่านี้ทำให้ รูปแบบการเขียนโปรแกรมเปลี่ยนไป โดยมีความเป็นระบบ และเป็นลักษณะของ OOP เพิ่มขึ้น เช่นใน VB.NET ต้องมีการอ้างอิงถึง class library ที่เกี่ยวข้องและต้องการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3.3 Presentation Layer

การแสดงผลสามารถทำได้ 2 รูปแบบหลัก คือในรูป Windows form และ Web form ซึ่งก่อนนี้ทั้งคำสั่ง และกระบวนการเขียนโปรแกรม ของทั้ง 2 แบบ แต่ใน .Net แล้วได้รวมทั้งสองอย่างมาทำให้เหมือนกันที่สุด คือเราเขียนโปรแกรมบน Windows อย่างไร คุณก็เขียน Web Application ด้วยวิธีเดียวกัน โดยเรียกใช้ คลาส ด้วยคำสั่งที่เหมือนๆกัน บน Windows จะอยู่ใน คลาส ที่ชื่อ system.windows แต่ Web จะอยู่ใน คลาส ชื่อ system.web.ui ซึ่งวิธีใช้เหมือนกันมาก ที่พิเศษคือ มี Object สำหรับ Web ให้คุณเขียนโปรแกรม แบบ visual basic ธรรมดา แต่เมื่อรันบน browser แล้วจะสร้าง tag html และ javascript ให้อัตโนมัติ ทำให้ browser ทั่วๆไปก็สามารถรันได้ ไม่จำเป็นต้องสนับสนุน .net โดยเฉพาะ ตัวอย่างเช่น เราสร้างเว็บเพจที่มี ปฏิทิน ด้วย(นึกถึง calendar ใน VB) ในส่วนของ รูปแบบ web จะแสดงในรูปของ html code ส่วนปฏิทินจะอยู่ในรูปของ java script ซึ่งทางโปรแกรมจะทำการแปลงให้เอง

7.4 .NET Compact Framework

.NET Compact Framework คือ platformของ Microsoft .NET ที่ใช้ในการพัฒนาอุปกรณ์เล็กๆ ความจริงแล้ว .NET Compact Framework ก็คือ .NET Framework ที่ตัดเอาส่วนที่ไม่สำคัญหรือไม่ได้ใช้ออกไป เพื่อทำให้ขนาด เล็กลงจนขนาดที่สามารถ install ลงบนอุปกรณ์เล็กๆ ที่ไม่ใช่คอมพิวเตอร์ได้ เช่น โทรศัพท์มือถือเป็นต้น

คุณสมบัติหลักของ .NET Compact Framework

- .NET Compact Framework ถูกออกแบบให้เป็น platformในการพัฒนา สำหรับการเขียนและเรียกใช้ XML เว็บเซอร์วิส
- เนื่องจาก .NET Compact Framework ใช้ programming model เดียวกันกับ desktop .NET Framework จึงง่ายต่อ developer ในการเขียนapplication ใหม่บน .NET Compact Framework
- ใช้ Visual Studio .NET ตัวเดียวกับที่ใช้ใน desktop .NET Framework ในการเขียน program device application
- ใช้ระบบ evidence-base security คืออนุญาตให้เฉพาะ applicationที่ผ่านการรับรอง จากผู้ผลิต device นั้น จึงจะสามารถ access secure resource ได้ ถ้าเป็น application ที่ยังไม่ผ่านการรับรองความปลอดภัย จะ access ได้เพียง basic system resourceเท่านั้น
- ออกแบบสำหรับอุปกรณ์ที่มี resource จำกัด เช่น โทรศัพท์มือถือ โดยที่ตัว frameworkเองใช้ memory อย่างมีประสิทธิภาพ และมีระบบดึง resource ที่ไม่ได้ใช้แล้วกลับมาจาก application อย่างมีประสิทธิภาพ
- ใช้ Just–in-time complier ซึ่งมีประสิทธิภาพดีกว่า ระบบ device programming อื่นๆ ที่ใช้ code interpreter
- เพิ่มประสิทธิภาพและลดต้นทุนในการพัฒนา เนื่องจากใช้ programming model และ toolset ของ Visual Studio .NET อันเดียวกันกับของ desktop .NET framework ทำให้นักพัฒนาสามารถใช้ความรู้ความชำนาญที่มีอยู่แล้วมาพัฒนา application บน อุปกรณ์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8

VRML

8.1 ที่มาและวัตถุประสงค์ของภาษา VRML

VRML ย่อมาจาก Virtual Reality Modeling Language(Virtual Reality World) หรือเรียกว่า "เวอร์มอล" ซึ่งเป็นภาษาคอมพิวเตอร์ที่ใช้ในการสร้างรูปเสมือนจริงเป็นรูปภาพกราฟิก 3 มิติ ประกอบความสามารถในการโต้ตอบกับผู้ใช้ทันที(Real-time interactive)ที่ผ่านทางระบบเบราเซอร์ของระบบเวิลด์ไวด์เว็บ (World Wide Web) ของเครือข่ายอินเตอร์เน็ตเสมือนกับว่าผู้ใช้เข้าไปอยู่ในระบบ 3 มิตินั้นจริงๆ นอกจากความสามารถทางด้านการสร้างกราฟิก 3 มิติ แล้วยังสามารถนำเสนอด้วยสื่อมัลติมีเดียเพื่อเพิ่มความสมจริงมากยิ่งขึ้น เช่น ระบบเสียงที่เป็นลักษณะ 3 มิติ ภาพเคลื่อนไหว ซึ่งสามารถโต้ตอบการเปลี่ยนแปลงมุมมองของผู้ใช้ได้ในเวลาจริง (Real-time Interaction) โดยผ่านการรับรู้และเปลี่ยนแปลงมุมต่างๆ

ภาษา VRML อาศัยหลักการแสดงผลกราฟิกทั้ง 2 มิติ และ 3 มิติ โดยอาศัยวิธีการแบบ "โอเพ่นจีแอล(OpenGL)" ซึ่งเป็นวิธีการแสดงผลของบริษัท Silicon Graphic ที่ถูกนำเสนอในปี ค.ศ. 1992 ใช้สำหรับการแสดงค่าของวัตถุต่างๆ ไม่ว่าจะเป็นตำแหน่ง (Coordinate) รายละเอียดพื้นผิว (Texture) จุดเด่นของ โอเพ่นจีแอล คือไม่ขึ้นอยู่กับระบบปฏิบัติการ จึงทำให้ผู้พัฒนาโปรแกรมไม่จำเป็นต้องคำนึงถึงว่าจะใช้เครื่องชนิดใดและระบบปฏิบัติการใด ในการสร้างภาพนั้น โอเพ่นจีแอล จะทำการสร้างภาพวัตถุโดยสร้างรูปร่างพื้นฐานของวัตถุก่อน และเก็บใน เฟรมบัฟเฟอร์ (Frame Buffer) ภาพวัตถุพื้นฐานในระบบคอมพิวเตอร์ คือ จุด เส้นตรง ภาพหลายหลายเหลี่ยมและบิตแมป (Bitmap) ภาพของวัตถุที่สร้างขึ้นนั้นส่วนประกอบแต่ละส่วนจะเป็นอิสระต่อกัน ดังนั้นในการเปลี่ยนแปลงลักษณะของวัตถุหรือภาพจะไม่ต้องกระทำทั้งวัตถุ เพียงแต่กระทำเฉพาะส่วนที่ต้องการเท่านั้น ไม่จำเป็นต้องเปลี่ยนแปลงทั้งวัตถุ เมื่อภาพของวัตถุที่แยกเป็นรูปพื้นฐานถูกเก็บในเฟรมบัฟเฟอร์ แล้วการเปลี่ยนแปลงใดๆ ในเรื่องของคุณสมบัติของวัตถุจะถูกควบคุมโดยโอเพ่นจีแอล โดยตรงไม่ว่าจะเป็นการเปลี่ยนแปลงตำแหน่ง ขนาด สี หรือการกำหนดค่าความเข้มของแสงต่างๆ ส่วนการของการนำผลของค่าที่อยู่ในเฟรมบัฟเฟอร์ มาแสดงเป็นหน้าที่ของวินโดวส์ว่าจะแสดงค่าใดในจอภาพ

ภาษา VRML มี ลักษณะเด่นในการแสดงวัตถุทั้งคงที่และเคลื่อนไหวและยังสามารถทำงานร่วมกับมัลติมีเดียอื่นๆ เช่น เสียง (Voice), ภาพ (Image), ภาพยนตร์(Movies)โดยผ่านตัวประมวลผลคือ เบราเซอร์ (Browser) นอกจากนั้นยังสนับสนุนลักษณะ 3 มิติ แบบ แอพพริเคชันโปรแกรมมิ่งอินเตอร์เฟซ (Application Programming Interface, API) อีกด้วย ภาษา VRML ได้รับการรองรับจาก ISO (International Organization for Standardization) และ IEC (International Electronic Commission) ซึ่งเป็นหน่วยงานที่ทำหน้าที่ควบคุมมาตรฐานต่างๆ ของอินเตอร์เน็ต โดยอนุญาตมาตรฐาน ISO/IEC 14772

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายใต้หัวข้อของ Information Technology-Computer graphics and Image Processing-vertebral Reality Modeling Language (VRML)

ภาษา VRML ทำงานภายใต้แบบอย่างพื้นฐานของ เว็บเบราเซอร์-เซิร์ฟเวอร์ (Web Browser-server)ทั่วไป โดยอาศัยรูปแบบยูอาแอล (URL) ซึ่งเป็นรูปแบบมาตรฐานสำหรับระบบเวิลด์ไวด์เว็บ โดยกำหนดให้ขึ้นต้นด้วย HTTP (Hypertext Transfer Protocol) โดยเซิร์ฟเวอร์ ภาษา VRML จึงถือว่าเป็นเครื่องมือในการสร้างโลกเสมือนจริงแบบใหม่ที่เรียกว่า "สังคมไร้พรมแดน"(Cyberspace) และ " สังคมเสมือนจริง"(On-line Virtual Communications) ขึ้นมาจำลองสังคมมนุษย์ในโลกแห่งความจริงมาไว้ในโลกของคอมพิวเตอร์ 3 มิติ ทำหน้าที่แปลคำร้องของเบราเซอร์ ขณะดาวโหลด (download) เซิร์ฟเวอร์ทำการส่งเอกสารที่เป็น แทก (Tag) ของเอกสารหรือเรียกว่า มัลติเพอร์เพิส อินเตอร์เน็ต เมล เอกซ์เทนชัน (Multipurpose Internet Mail Extensions,MIME) ซึ่งภาษา VRML มีลักษณะเป็น cross-world/cross-vrml (x-world/x-vrml)โดยผู้ใช้สามารถดูด้วยเบราเซอร์ที่เรียกว่า วีอาเอ็มแอล เบราเซอร์ (VRML Browser) ได้โดยอาศัยไฟล์ข้อมูลในรูปแบบ VRML (*.wrl) ซึ่งเป็นรูปแบบกลาง สำหรับแลกเปลี่ยนข้อมูล 3 มิติ อาศัยการนำเสนอวัตถุ (Object) เป็นแบบเนสต์ (Nesting) โดยส่งข้อมูลรูปแบบทั้งหมดมาก่อนและตามด้วยระดับความละเอียดภายหลังอาศัยหลักการ แอลโอดี (LOD,Level Of Detail) เปลี่ยนแปลงไปมาโดยอัตโนมัติและทำการเรนเดอร์ (Render) เพื่อสร้างแบบจำลองกราฟิก 3 มิติ ที่วีอาเอ็มแอล เบราเซอร์ ส่วนเอกสารที่เป็นเสียงหรือวีดีโอจะถูกส่งมาตามลำดับ

ลักษณะเด่นของภาษา VRML อาจเรียกได้ว่าภาษา VRML มีลักษณะเป็นโฮมเพจ 3 มิติ (3D-Homepage) มีลักษณะภาษาคอมพิวเตอร์ชนิดง่ายๆ ไม่ขึ้นอยู่กับระบบคอมพิวเตอร์ใดๆ พร้อมทั้งตีพิมพ์เอกสารโฮมเพจที่เป็น 3 มิติ บนบริการเวิลด์ ไวด์ เว็บ (World Wide Web) ของเครือข่ายอินเตอร์เน็ต รวมทั้งการนำเสนอข้อมูลจะเกิดประโยชน์สูงสุดเมื่อสามารถเคลื่อนไหวเปลี่ยนมุมมองได้ ผู้เข้าชมสามารถมีส่วนร่วมในการโต้ตอบกับวัตถุที่อยู่ในฉากได้ ถือได้ว่าภาษา VRML เป็นเทคโนโลยีการนำเสนอข้อมูลหลายรูปแบบทั้งกราฟิก 2 มิติ และ 3 มิติ รวมทั้งข้อมูลมัลติมีเดีย เช่น ภาพนิ่ง ภาพเคลื่อนไหว และระบบเสียง 3 มิติ ทำให้เกิดระบบการติดต่อกับผู้ใช้ (User Interface) ในรูปแบบใหม่ขึ้นมาโดยระบบ 2 มิติ ที่ใช้อยู่ปัจจุบันกลายเป็นส่วนประกอบย่อยใน 3 มิติ

8.2 ข้อแตกต่างของภาษา VRML เวอร์ชัน 1.0 และ 2.0

ภาษา VRML ได้มีการพัฒนาจาก เวอร์ชัน 1.0 และมาเป็น 2.0 ในปัจจุบัน ซึ่งมีข้อแตกต่างอย่างชัดเจน 2 ส่วนคือ ในส่วนของการโต้ตอบกับผู้ใช้ (Interactive) กับส่วนของความสมจริง (Realistic) ซึ่งภาษา VRML เวอร์ชัน 1.0 ไม่มีคุณสมบัติดังกล่าวคือ ไม่มีการโต้ตอบและรูปทรงจะคงที่ไม่มีการเคลื่อนไหว นอกจากนั้นภาษา VRML เวอร์ชัน 2.0 สามารถทำงานร่วมกันกับภาษา Java และ Javascript รวมทั้งมีส่วนของเสียง 3 มิติ และภาพเคลื่อนไหวเข้ามาเพิ่มความสมจริงยิ่งขึ้น รวมถึงการเปลี่ยนแปลงชื่อโหนดดังตารางต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโหนด VRML เวอร์ชัน 1.0	ชื่อโหนด VRML เวอร์ชัน 2.0
AsciiText	Text
Coordinate3	Coordinate
Cube	Box
Info	WorldInfo
MatrixTransform	Transform
OrthogrphicCamera	Viewpoint
PerspectiveCamera	Viewpoint
Rotation	Transform
Scale	Transform
Separator	Transform
Texture2	Texture
Texture2Transform	TextureTransform
TextureCoordinate2	TextureCoordinate
Translation	Transform
WWWAnchor	Anchor
WWWInline	Inline

ตารางที่ 8.1 การเปรียบเทียบชื่อโหนดของเวอร์ชัน 1.0 กับเวอร์ชัน 2.0

8.3 ภาษา VRML และ ภาษา HTML

ภาษา VRML และ ภาษา HTML มีส่วนการสร้างขึ้นมาคล้ายกันคือการเขียนโปรแกรมขึ้นมาในลักษณะของเท็กซ์โหมด โดยใช้เท็กซ์อิดิเตอร์ เป็นเครื่องมือพื้นฐานแล้วจึงใช้เบราเซอร์เป็นตัวอินเตอร์พรีเตอร์ (Interpreter) ในทางกลับกันลักษณะที่แตกต่างกันชัดเจนคือ การประมวลผลออกมา ภาษา VRML เป็นเว็บ 3 มิติ ส่วนภาษา HTML เป็นเว็บ 2 มิติ จากคุณสมบัติข้างต้น ทำให้ภาษา VRML มีขนาดของไฟล์ใหญ่กว่าภาษา HTML

คุณสมบัติ	ภาษา HTML	ภาษา VRML
ลักษณะพื้นฐานโครงสร้างภาษา	Text mode	Text mode
ลักษณะการทำงาน	Web Browser-server	Web Browser-server
แกน	X,Y	X,Y,Z
การประมวลกราฟิกหรือเรนเดอร์	2 มิติ	3 มิติ
ขนาดไฟล์	เล็ก	ใหญ่
ชนิดของนามสกุล	.htm	.wrl

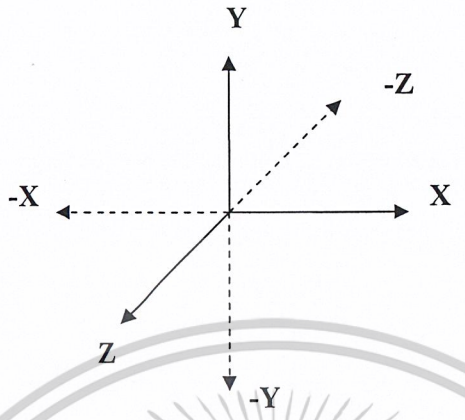
ตารางที่ 8.2 การเปรียบเทียบของภาษา VRML กับภาษา HTML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.4 องค์ประกอบของภาษา VRML

8.4.1 ระบบแกน 3 มิติ

ภาษา VRML มีการทำงานภายใต้ระบบแกน 3 มิติ ซึ่งจะมีการใช้กฎมือขวาดังนี้



รูปที่ 8.1 ระบบแกน 3 มิติ

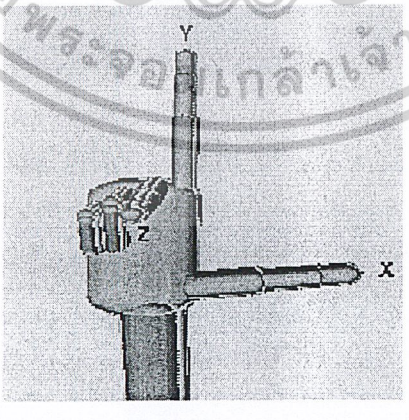
แกนประกอบไปด้วย 3 แกนคือแกน X,Y,Z โดยที่

แกน X มีทิศทางไปด้านขวาจากจุดกำเนิด (origin)

แกน Y มีทิศทางไปด้านบนจากจุดกำเนิด (origin)

แกน Z มีทิศทางตั้งฉากกับแกน X และแกน Y (มีทิศพุ่งออกมาทางนอกฉาก) ซึ่งภายในแต่ละแกนจะสามารถหมุนรอบแกนมันเอง

ในแต่ละแกนยังประกอบไปด้วยค่าบวกและค่าลบ ถ้ามีค่าเป็นบวกจะอยู่ด้านขวา (แกน X) และด้านบน (แกน Y) และด้านหน้า (แกน Z) เช่น พิกัด 1 0 1 แสดงว่าไปทางขวา 1 หน่วย และไปด้านหน้า 1 หน่วยในแกน Z ส่วนค่าลบจะมีทิศทางตรงข้ามกับค่าบวก รวมทั้งยังสามารถหมุนพร้อมกันได้โดยอาศัยการหมุนรอบจุดกำเนิด



รูปที่ 8.2 การใช้กฎมือขวา

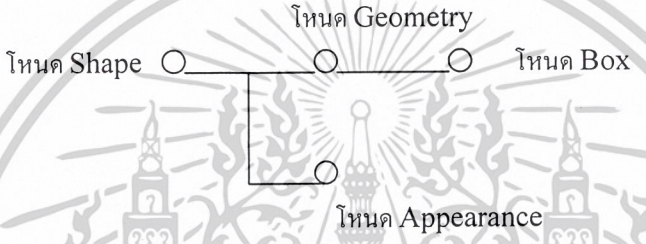
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.4.2 องค์ประกอบพื้นฐานในไฟล์ VRML

การสร้างวัตถุในโลกเสมือนจริงของภาษา VRML เกิดจากองค์ประกอบพื้นฐานคือ

- โหนด (node)
- ฟิลด์ (field)

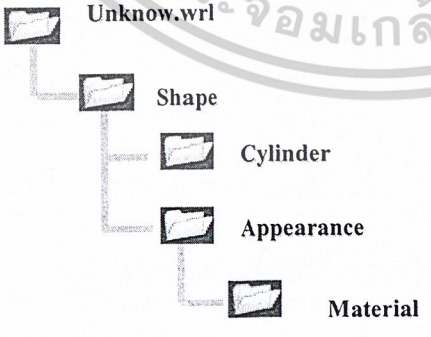
8.4.2.1 โหนด (node) คือหน่วยพื้นฐานในไฟล์ของภาษา VRML ทำหน้าที่เก็บค่าคุณสมบัติพื้นฐานต่างๆ ของฟิลด์ เช่น โหนด Shape, โหนด Timesensor เป็นต้น ซึ่งโหนดมีโครงสร้างอิสระ สามารถกำหนดโหนดเดียวภายในไฟล์ได้ แต่ฟิลด์ไม่สามารถกำหนดเองได้ ต้องอาศัยโหนดภายในไฟล์ สามารถกำหนดโหนดอื่นได้ (.ในแต่ละโหนดจะมีฟิลด์ที่ใช้เก็บคุณสมบัติต่างๆ) เช่น โหนด Shape จะประกอบไปด้วย โหนด Geometry และโหนด Appearance ดังรูปที่ 8.2



รูปที่ 8.3 ตัวอย่างโครงสร้างของโหนด

ภายในโหนด Geometry ยังสามารถแตกโหนดย่อยลงไปอีกได้ หรือเรียกว่า "โหนดลูก (children node)" เช่น โหนด Box และโหนด Appearance เป็นโหนดลูกของโหนด Shape เป็นต้น

8.4.2.2 ฟิลด์ (field) คือหน่วยพื้นฐานที่เล็กที่สุดของไฟล์ภาษา VRML โดยทำหน้าที่เก็บค่าคุณสมบัติของโหนดนั้นๆ เช่น โหนด Box ก็จะมีฟิลด์รอรับค่า ความกว้าง ความยาว ความสูง ภายในนั้นเอง จึงอาจสรุปได้ว่าหลายๆฟิลด์ประกอบกันเป็นโหนดและในทำนองเดียวกันหลายๆโหนดก็จะรวมกันเป็นวัตถุขึ้นมาชิ้นหนึ่ง



รูปที่ 8.4 ตัวอย่างวัตถุที่เกิดจากการนำโหนดหลายๆโหนดมาประกอบกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟิลด์ในภาษา VRML จะมีส่วนคล้ายกับตัวแปลของภาษาจาวา คือมี 2 ชนิดดังนี้

1. ชนิดซิงเกิล (single) เป็นตัวแปรชนิดซิงเกิล
2. ชนิดมัลติ (multi) เป็นตัวแปรอาร์เรย์ของตัวแปรชนิดซิงเกิล

จะเห็นได้ว่าโหนดเป็นเสมือนคลาสของภาษาจาวา ส่วนฟิลด์คล้ายตัวแปรซึ่งเก็บค่าคุณสมบัติต่างๆไว้

8.5 การสร้างไฟล์ VRML

การสร้างไฟล์ VRML นั้นหลักๆ จะมีอยู่ 2 แนวทางด้วยกันคือ

1. การเขียนโค้ดโดยใช้เท็กซ์อิดิเตอร์ ทั่วไป หรืออาจเป็น เท็กซ์อิดิเตอร์ สำหรับเขียนโค้ดภาษา VRML โดยเฉพาะ ซึ่งการสร้างด้วยวิธีนี้จะต้องเข้าใจโครงสร้างของภาษา และคุณสมบัติของโหนดและฟิลด์ต่างๆ ที่ใช้ในภาษาเป็นอย่างดี

2. การใช้ โปรแกรมสร้างโมเดล 3 มิติ เช่น 3D-Max หรือโปรแกรมอื่นๆที่สามารถสร้างโมเดล3 มิติได้ จากนั้นจึงทำการแปลงรูปแบบไฟล์ให้อยู่ในรูปแบบของไฟล์ VRML (*.wrl) โดยอาจเป็นเป็นการแปลงโดยใช้โปรแกรมที่สร้างโมเดลนั้น หรือใช้โปรแกรมที่ใช้สำหรับการแปลงรูปแบบของไฟล์ ข้อดีของการสร้างไฟล์VRML ด้วยวิธีนี้ก็คือทำได้ง่ายกว่าการเขียนโค้ดด้วยตัวเองโดยใช้เท็กซ์อิดิเตอร์และเหมาะสำหรับผู้ที่มีความรู้ทางด้านการใช้โปรแกรมสร้างโมเดล 3 มิติ แต่จะมีข้อเสียคือ ขนาดของไฟล์ที่ได้จะมีขนาดใหญ่มากเนื่องจากโครงสร้างของไฟล์นั้นโปรแกรมจะเป็นตัวกำหนด และโครงสร้างภาษาที่ออกมาจะทำความเข้าใจได้ยากเพราะมีความซับซ้อน

8.6 การจัดเตรียมองค์ประกอบเพื่อสนับสนุนการทำงานของภาษา VRML

การจัดเตรียมองค์ประกอบเพื่อสนับสนุนการทำงานของภาษา VRML นั้นจะต้องเตรียมทั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์ เพื่อให้ VRML สามารถทำงานได้ ซึ่งประกอบด้วย

8.6.1 การเตรียมเบราเซอร์ของภาษา VRML

องค์ประกอบหนึ่งที่สำคัญของเบราเซอร์ซึ่งจะทำหน้าที่ในการเรนเดอร์หรือประมวลผลกราฟิก (rendering) แบบจำลองกราฟิก 3 มิติ ดังนั้นจึงจำเป็นต้องมีเบราเซอร์ที่สามารถสนับสนุนภาษา VRML หรือเรียกว่า VRML เบราเซอร์ ขึ้นเพื่อรองรับการทำงานของภาษา VRML ซึ่งซอฟต์แวร์เหล่านั้นได้แก่

- ระบบปฏิบัติการ Windows 95 ขึ้นไป
- เบราเซอร์ คือ โปรแกรม Netscape Navigator version 3.01 ขึ้นไป หรือโปรแกรม Internet Explorer version 4.0 ขึ้นไป
- โปรแกรมเสริมปลักอิน(plug-in) ที่สามารถสนับสนุนภาษา VRML

8.6.2 การเตรียมทางด้านฮาร์ดแวร์

อุปกรณ์ฮาร์ดแวร์ที่สนับสนุนการประมวลผลของภาษา VRML ซึ่งควรคุณสมบัติอย่างต่ำดังต่อไปนี้

- ไมโครโปรเซสเซอร์ เพนเทียม 133 เมกะเฮิรตซ์ขึ้นไป
- หน่วยความจำหลัก (RAM) อย่างน้อย 32 เมกะไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การ์ดแสดงผลแบบ VGA หรือ SuperVGA
- แรมแสดงผล 2 เมกะไบต์ขึ้นไป จำนวนสี 256 สีขึ้นไป
  หรือชนิดที่สนับสนุนการทำงานลักษณะ 3 มิติ

8.7 สรุปลักษณะเด่นของภาษา VRML

ลักษณะเด่นๆ ของภาษา VRML อาจสรุปได้ดังนี้

- สร้างแบบจำลองกราฟิก 3 มิติ (3D Graphic model)
- สร้างการโต้ตอบกับผู้ใช้ทันที (Real-time Interactive)
- สร้าง แสง เสียงในระบบ 3 มิติ(Light and Sound 3D)
- สร้างภาพเคลื่อนไหว (Animation)
- มุมมองในการชมแบบจำลอง 3 มิติ ได้แก่ การเดิน (Walk) การหมุน (Rotate) การบิน (Fly)

8.8 รูปแบบไวยากรณ์โหนดต่างๆ (Syntex)

โหมดที่ใช้ในการรวมกลุ่มวัตถุ (Grouping node)

โหนด Transform

Transform {

eventIn	MFNode	addChildren	
eventIn	MFNode	removeChildren	
exposedField	SFVec3f	center	0 0 0
exposedField	MFNode	children	[ ]
exposedField	SFRotation	rotation	0 0 1 0
exposedField	SFVec3f	scale	1 1 1
exposedField	SFRotation	scaleOrientation	0 0 1 0
exposedField	SFVec3f	translation	0 0 0
field	SFVec3f	bboxCenter	0 0 0
field	SFVec3f	bboxSize	-1 -1 -1

}

โหนดInline

Inline {

exposedField	MFString	url	[ ]
field	SFVec3f	bboxCenter	0 0 0
field	SFVec3f	bboxSize	-1 -1 -1

}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหนด LOD

LOD {

exposedField	MFNode	leve	[ ]
field	SFVec3f	center	0 0 0
field	SFVec3f	range	[ ]

}

โหนด Switch

Switch {

exposedField	MFNode	level	[ ]
exposedField	SFInt32	whichChoice	-1

}

โหมดเกี่ยวกับการสร้างแสงชนิดต่างๆ ให้แก่วัตถุ

โหนด Directionalight

Directionalight {

exposedField	SFFloat	ambientIntensity	0
exposedField	SFColor	color	1 1 1
exposedField	SFVec3f	direction	0 0 -1
exposedField	SFFloat	intensity	1
exposedField	SFBool	on	TRUE

}

โหนด Positionlight

Positionlight {

exposedField	SFFloat	ambientIntensity	0
exposedField	SFVec3f	attenuation	1 0 0
exposedField	SFColor	color	1 1 1
exposedField	SFFloat	intensity	1
exposedField	SFBool	location	0 0 0
exposedField	SFBool	on	TRUE
exposedField	SFFloat	radius	1 0 0

}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหนด SpotLight

SpotLight {

exposedField	SFFloat	ambientIntensity	0
exposedField	SFVec3f	attenuation	1 0 0
exposedField	SFFloat	beamWidth	1.570796
exposedField	SFColor	color	1 1 1
exposedField	SFFloat	cutoffangle	0.785398
exposedField	SFVec3f	direction	0 0 -1
exposedField	SFFloat	intensity	1
exposedField	SFVec3f	location	0 0 0
exposedField	SFBool	on	TRUE
exposedField	SFFloat	radius	1 0 0

}

โหมดเกี่ยวกับการสร้างรูปทรง

โหนด Shape

Shape {

exposedField	SFNode	appearance	NULL
exposedField	SFNode	geometry	NULL

}

โหนด Box

Box {

field	SFVec3fsize	2 2 2	

}

โหนด Cone

Cone {

field	SFFloat	bottomRadius	1
field	SFFloat	height	2
field	SFBool	side	TRUE
field	SFBool	bottom	TRUE

}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหนด ElevationGrid

ElevationGrid {

exposedField	SFNode	color	NULL
exposedField	SFNode	normal	NULL
exposedField	SFNode	textCoord	NULL
field	MFFloat	height	[ ]
field	SFBool	ccw	TRUE
field	SFBool	colorPerVertex	TRUE
field	SFFloat	creaseAngle	0
field	SFBool	normalPerVertex	TRUE
field	SFBool	solid	TRUE
field	SFint32	xDimension	0
field	SFFloat	xSpacing	0.0
field	SFint32	zDimension	0
field	SFFloat	zPacing	0.0

}

เหตุการณ์ (Event)

event	MFFloat	set_height

โหนด Extrusion

Extrusion {

field	SFBool	beginCap	TRUE
field	SFBool	CCW	TRUE
field	SFBool	convex	TRUE
field	SFFloat	creaseAngle	0
field	SFVec2f	crossSection	[11,1 -1,-1 -1,-1 1, 11]
field	SFBool	endCap endCap	TRUE
field	SFRotation	orientation	0010
field	SFVec2f	creaseAngle	0
field	SFBool	scale	1 1
field	SFBool	solid	TRUE
field	SFVec3f	spine	[0 0 0,0 1 0]

}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหตุการณ์ (Event)

eventIn	MFVec2f	set_ceossSection
eventIn	MFRotation	set_orientation
eventIn	MFVec2f	set_scale
eventIn	MFVec3f	set_spine

โหนด IndexedFaceSet

IndexedFaceSet {

exposedField	SFNode	color	NULL
exposedField	SFNode	coord	NULL
exposedField	SFNode	convex	NULL
exposedField	SFNode	creaseAngle	NULL
field	SFBool	CCW	TRUE
field	SFInt32	colorIndex	[ ]
field	SFBool	colorPerVertex	TRUE
field	SFBool	convex	TRUE
field	SFInt32	coordIndex	[ ]
field	SFFloat	creaseAngle	0
field	SFInt32	normalIndex	[ ]
field	SFBool	normalPerVertex	TRUE
field	SFBool	solid	TRUE
field	SFInt32	texCoordIndex	[ ]

}

เหตุการณ์ (Event)

eventIn	MFInt32	set_colorIndex
eventIn	MFInt32	set_coordIndex
eventIn	MFInt32	set_normalIndex
eventIn	MFInt32	set_textCoordIndex.

โหนด IndexedLineSet

IndexedLineSet {

eventIn	MFInt32	set_colorIndex	
eventIn	MFInt32	set_coordIndex	
exposedField	SFNode	color	NULL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

exposedField	SFNode	coord	NULL
field	MFInt32	colorIndex	[ ]
field	SFBool	colorPerVertex	TRUE
field	MFInt32	colorIndex	[ ]

}

โหนด Pointset

Pointset {

exposedField	SFNode	color	NULL
exposedField	SFNode	coord	NULL

}

โหนด Sphere

Sphere {

Field	SFFloat	radius	1

}

โหนด Text

Text {

exposedField	MFString	string	[ ]
exposedField	SFNode	coord	NULL
exposedField	MFFloat	length	[ ]
exposedField	SFFloat	maxExtent	0.0

}

โหนด Color

Color {

exposedField	MFColor	color	[ ]

}

โหนด Coordinate

Coordinate {

exposedField	MFVec3t	point	[ ]

}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหนด Normal

Normal {

exposedField	MFVec3f	vector	[ ]

}

โหมดเกี่ยวกับการสร้างเสียง

โหนด Sound

Sound {

exposedField	SFVec3f	direction	0 0 1
exposedField	SFFloat	Intensity	1
exposedField	SFVec3f	location	0 0 0
exposedField	SFFloat	maxBack	1 0
exposedField	SFFloat	maxFront	1 0
exposedField	SFFloat	maxBack	1
exposedField	SFFloat	minFront	1
exposedField	SFFloat	priority	0
exposedField	SFNode	source	NULL
field	SFBool	spatialize	TRUE

}

โหนด Audioclip

Audioclip {

exposedField	SFString	description	" "
exposedField	SFBool	loop	FALSE
exposedField	SFFloat	pitch	1.0
exposedField	SFTime	startTime	0
exposedField	SFTime	stopTime	0
exposedField	MFString	url	[ ]
exposedField	SFTime	duration_changed	
exposedField	SFBool	isActive	

}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมดที่ทำงานร่วมกับสคริปต์

โหนด Script

Script {

exposedField	MFString	url	[ ]
field	SFBool	directOutput	FALSE
field	SFBool	mustEvaluate	FALSE
field	fieldTypename	fieldname initialValue	

}

เหตุการณ์ (Event)

eventIn	eventTypeName	eventname
eventIn	eventTypeName	eventname

โหมดเกี่ยวกับการสร้างการโต้ตอบโดยอาศัยตัวตรวจจับ (Sensors)

โหนด CylinderSensor

CylinderSensor {

exposedField	SFFloat	diskAngle	0.262
exposedField	SFBool	enabled	TRUE
exposedField	SFFloat	maxAngle	-1
exposedField	SFFloat	minAngle	0
exposedField	SFRotation	offset	0 1 0 0
exposedField	SFBool	autoOffset	TRUE

}

เหตุการณ์ (Event)

eventOut	SFBool	isActive
eventOut	SFRotation	rotation_changed
eventOut	SFVec3f	trackPoint_changed

โหนด PlaneSensor

PlaneSensor {

exposedField	SFBool	enabled	TRUE
exposedField	SFVec2f	axPosition	-1 -1
exposedField	SFFloat	inPosition	0 0
exposedField	SFFloat	offset	0 0 0
exposedField	SFRotation	autoOffset	0 1 0 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

}

เหตุการณ์ (Event)

eventOut	SFBool	isActive	
eventOut	SFVec3f	rackPoint_changed	
eventOut	SFVec3f	ranslation_changed	

โหนด ProximitySensor

ProximitySensor {

exposedField	SFVec3f	Center	0 0 0
exposedField	SFVec3f	size	0 0 0
exposedField	SFBool	enabled	TRUE

}

เหตุการณ์ (Event)

eventOut	SFBool	isActive	
eventOut	SFVec3f	Position_changed	
eventOut	SFTime	enterTime	
eventOut	SFTime	exitTime	

โหนด SphereSensor

SphereSensor {

exposedField	SFBool	enabled	TRUE
exposedField	SFRotation	offset	0 1 0 0
exposedField	SFBool	autoOffset	TRUE

}

เหตุการณ์ (Event)

eventOut	SFBool	isActive	
eventOut	SFRotation	rotation_changed	
eventOut	SFTime	trackPoint_ changed	

โหนด TimeSensor

TimeSensor {

exposedField	SFTime	cycleInterval	1
exposedField	SFBool	offset	TRUE
exposedField	SFBool	loop	FALSE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

exposedField	SFTime	startTime	0
exposedField	SFTime	stopTime	0

}

เหตุการณ์ (Event)

eventOut	SFTime	isActive
eventOut	SFFloat	fraction_changed
eventOut	SFTime	time

โหนด TouchSensor

TouchSensor {

exposedField	SFBool	enabled	TRUE
eventOut	SFVec3f	hitNormal_changed	
eventOut	SFVec3f	hitPoint_changed	

}

เหตุการณ์ (Event)

eventOut	SFVec2f	hitTextCoord_changed
eventOut	SFBool	isActive
eventOut	SFBool	isOver
eventOut	SFTime	touchTime

โหนด VisibilitySensor

VisibilitySensor {

exposedField	SFVec3f	center	0 0 0
eventOut	SFBool	enabled	TRUE
eventOut	SFVec3f	size	0 0 0

}

เหตุการณ์ (Event)

eventOut	SFTime	enterTime
eventOut	SFTime	exitTime
eventOut	SFBool	isActive

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหนดเกี่ยวกับการกำหนดคุณสมบัติพื้นผิวของวัตถุ (Apperance)

โหนด Appearance

Appearance {

exposedField	SFNode	material	NULL
exposedField	SFNode	texture	NULL
exposedField	SFNode	textureTransform	NULL

}

โหนด FontStyle

FontStyle {

field	SFString	family	“SERIF”
field	SFBool	horizontal	TRUE
field	SFString	justify	“BEGIN”
field	SFString	language	“ ”
field	SFBool	leftToRight	TRUE
field	SFFloat	size	1.0
field	SFFloat	spacing	1.0
field	SFStringstyle		“ ”
field	SFBool	toptoBottom	TRUE

}

โหนด ImageTexture

ImageTexture {

exposedField	MFString	url	[ ]
field	SFBool	repeatS	TRUE
field	SFBool	repeatT	TRUE

}

โหนด Material

Material {

exposedField	SFFloat	ambientIntensity	0.2
exposedField	SFColor	diffuseColor	0.8 0.8 0.8
exposedField	SFColor	emissiveColor	0 0 0
exposedField	SFFloat	shininess	0.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

exposedField	SFColor	specularColor	0 0 0
exposedField	SFFloat	transparency	0

}

โหนด Movietexture

Movietexture {

exposedField	SFBool	loop	FALSE
exposedField	SFFloat	speed	1
exposedField	SFTime	startTime	0
exposedField	SFTime	stopTime	0
exposedField	MFString	url	[ ]
field	SFBool	repeatS	TRUE
field	SFBool	repeatT	TRUE

}

เหตุการณ์ (Event)

eventOut	SFFloat	duration_changed
eventOut	SFBool	isActive

โหนด PixelTexture

PixelTexture {

exposedField	SFImage	image	0 0 0
field	SFBool	repeatS	TRUE
field	SFTime	repeatT	TRUE

}

โหนด TextureTransform

TextureTransform {

exposedField	SFVec2f	center	0 0 0
exposedField	SFFloat	rotation	0
exposedField	SFVec2f	scale	1 1
exposedField	SFVec2f	translation	0 0

}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหนด TextureCoordinate

TextureCoordinate {

exposedField	MFVec2f	point	[ ]

}

โหนดที่ใช้ในการสร้างภาพเคลื่อนไหว (Interpolators)

โหนด ColorInterpolator

ColorInterpolator {

eventIn	SFFloat	set_fraction	
exposedField	MFFloat	key	[ ]
exposedField	MFColor	keyValue	[ ]

}

เหตุการณ์ (Event)

eventOut	SFColor	value_changed	

โหนด CoordinateInterpolator

CoordinateInterpolator {

eventIn	SFFloat	set_fraction	
exposedField	MFFloat	key	[ ]
exposedField	MFVec3f	keyValue	[ ]

}

เหตุการณ์ (Event)

eventOut	SFColor	value_changed	[ ]

โหนด NormalInterpolator

NormalInterpolator {

exposedField	MFFloat	key	[ ]
exposedField	MFVec3f	keyValue	[ ]

}

เหตุการณ์ (Event)

eventIn	SFFloat	set_fraction	[ ]
eventOut	MFRotation	value_changed	[ ]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหนด OrientationInterpolator

OrientationInterpolator {

exposedField	MFFloat	key	[ ]
exposedField	MFRotation	keyValue	[ ]

}

เหตุการณ์ (Event)

eventIn	SFFloat	set_fraction	[ ]
eventOut	SFRotation	value_changed	[ ]

โหนด PositionInterpolator

PositionInterpolator {

exposedField	MFFloat	key	[ ]
exposedField	MFVec3f	keyValue	[ ]

}

เหตุการณ์ (Event)

eventIn	SFFloat	set_fraction
eventOut	SFVec3f	value_changed

โหนด ScalarInterpolator

ScalarInterpolator {

exposedField	MFFloat	key	[ ]
exposedField	MFVec3f	keyValue	[ ]

}

เหตุการณ์ (Event)

eventIn	SFFloat	set_fraction
eventOut	SFFloat	value_changed

โหนดเกี่ยวกับการสร้างสิ่งแวดล้อม

โหนด Background

Background {

exposedField	MFFloat	groundAngle	[ ]
exposedField	MFColor	groundColor	[ ]
exposedField	MFString	backUrl	[ ]
exposedField	MFString	bottomUrl	[ ]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

exposedField	MFString	frontUrl	[ ]
exposedField	MFString	leftUrl	[ ]
exposedField	MFString	rightUrl	[ ]
exposedField	MFString	topUrl	[ ]
exposedField	MFFloat	skyAngle	[ ]
exposedField	MFColor	skyColor	[ ]

}

เหตุการณ์ (Event)

eventIn	SFBool	set_bind
eventOut	SFBool	isBound

โหนด Fog

Fog {

exposedField	SFColor	color	1 1 1
exposedField	SFString	fogType	"LINEAR"
exposedField	SFFloat	visibilityRange	1000

}

เหตุการณ์ (Event)

eventIn	SFBool	set_bind
eventOut	SFBool	isBound

โหนด NavigationInfo

NavigationInfo {

exposedField	MFFloat	avatarSize	[0.25, 1.6,0.75]
exposedField	SFBool	headligh	TRUE
exposedField	SFFloat	speed	1.0
exposedField	MFString	Type	"WALK"
exposedField	SFFloat	visibilityLimit	0.0

}

เหตุการณ์ (Event)

eventIn	SFBool	set_bind
eventOut	SFBool	isBound

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหนด Viewpoint

Viewpoint {

exposedField	SFFloat	fieldOfView	0.785398
exposedField	SFBool	jump	TRUE
exposedField	SFRotation	orientation	0 0 1 0
exposedField	SFVec3f	position	0 0 0
field	SFString	description	“”

}

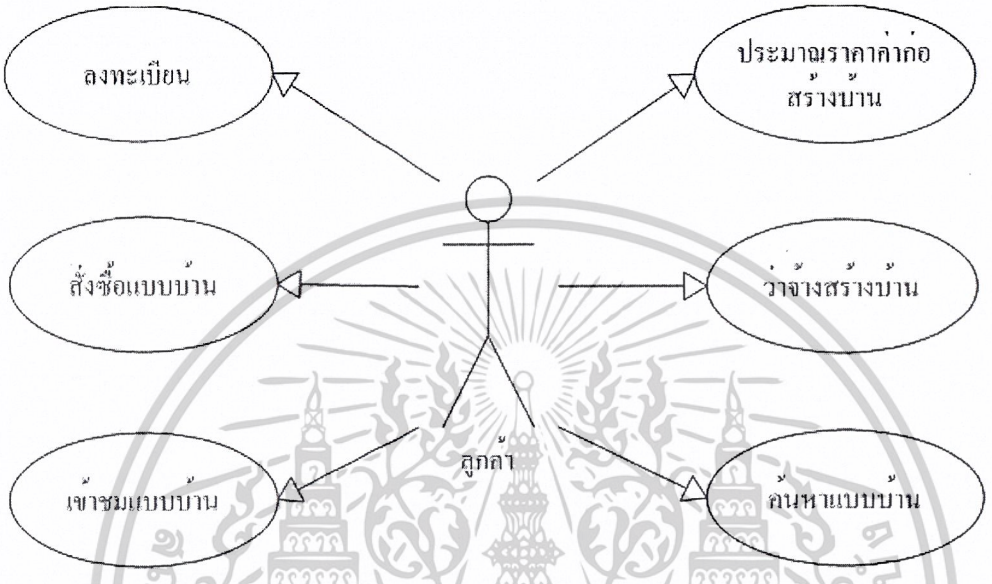
เหตุการณ์ (Event)

eventIn	SFBool	set_bind
eventOut	SFTime	bindTime_changed
eventOut	SFBool	isBound

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 9
การสร้าง และออกแบบระบบ Home For You

9.1 ยูสเคสไดอะแกรม (Use case diagram) ของระบบ Home For You



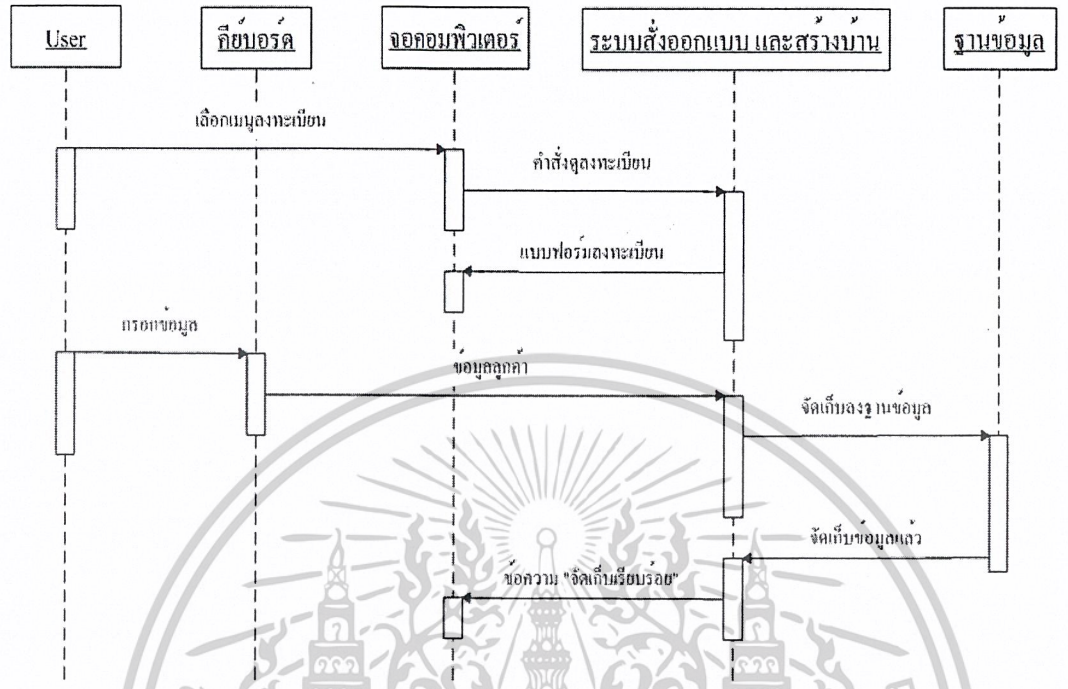
รูปที่ 9.1 ยูสเคสไดอะแกรม ของระบบออกแบบ และสร้างบ้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.2 Sequence diagram Home For You

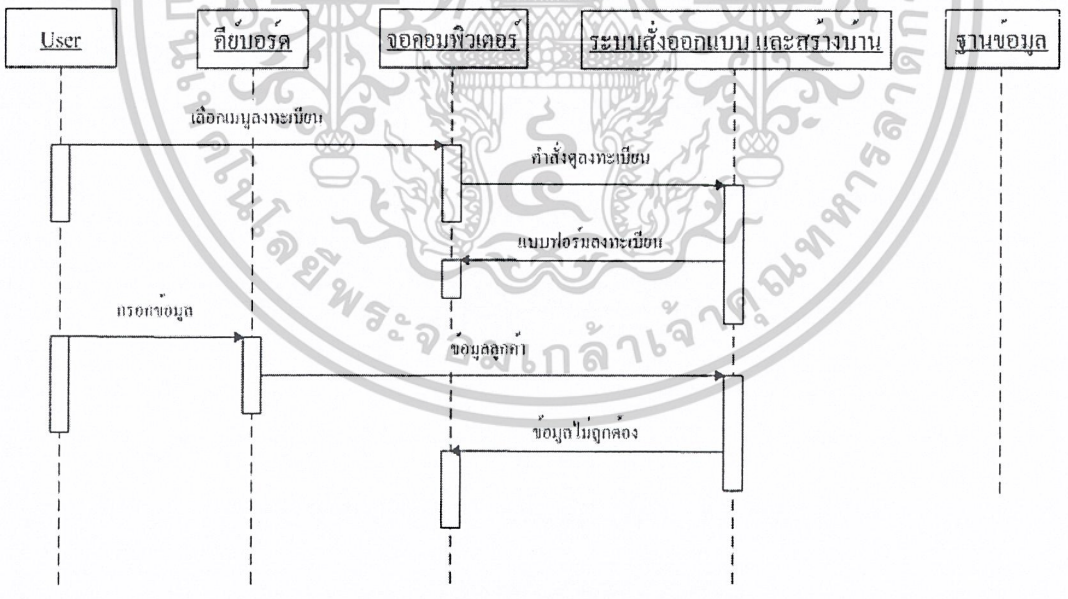
ขั้นตอน (Sequence) ของการลงทะเบียน (Sing Up)

- กรณีข้อมูลถูกต้อง



รูปที่ 9-2 ขั้นตอน ของการลงทะเบียนกรณีข้อมูลถูกต้อง

- กรณีข้อมูลไม่ถูกต้อง

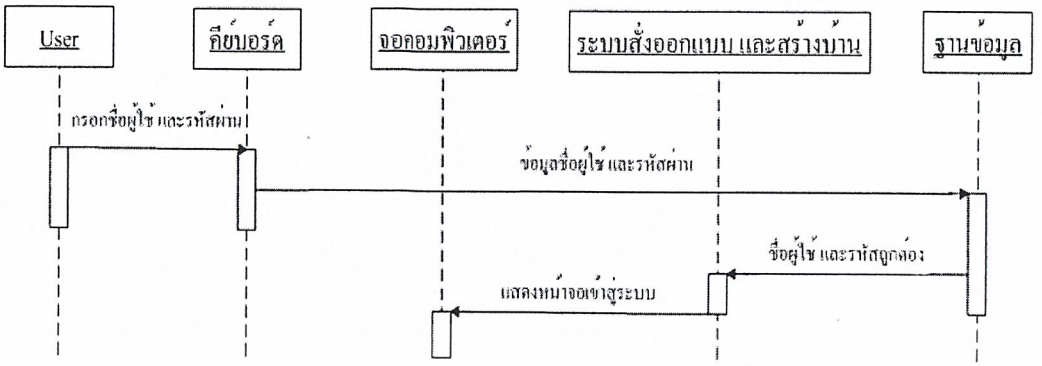


รูปที่ 9-3 ขั้นตอน ของการลงทะเบียนกรณีข้อมูลไม่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

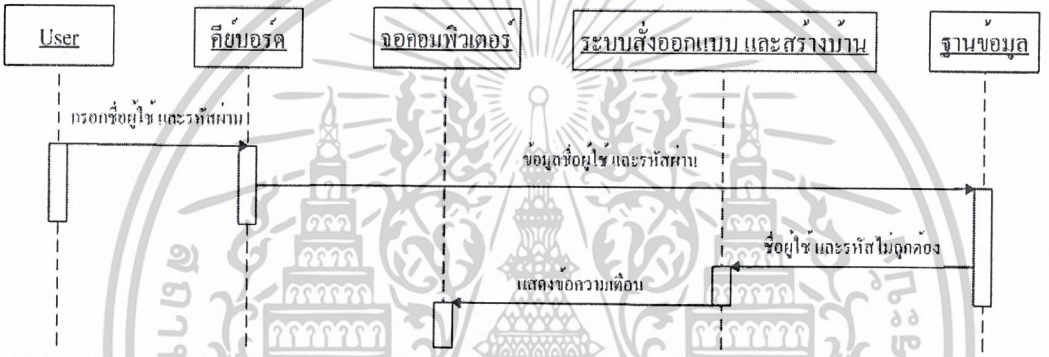
ขั้นตอนการ Sign in

- กรณีชื่อผู้ใช้ และรหัสผ่านถูกต้อง



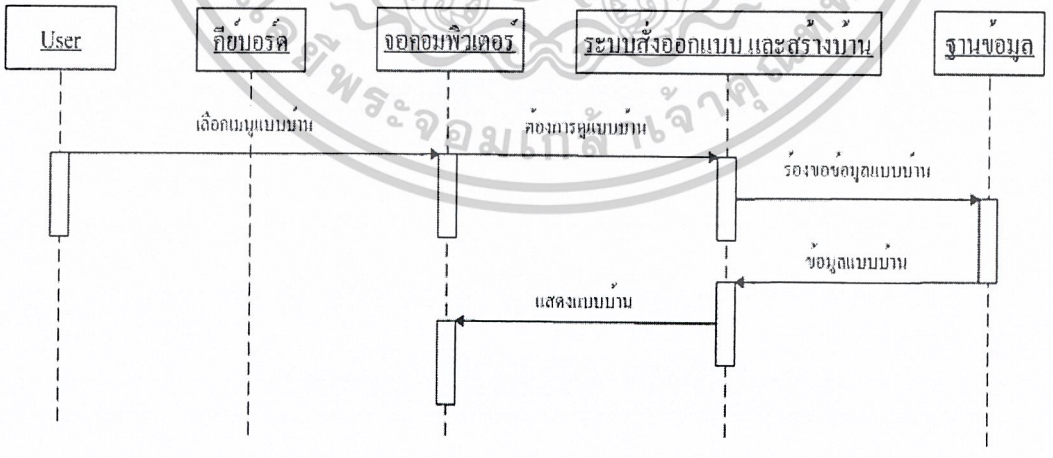
รูปที่ 9-4 ขั้นตอน ของการ Sign in กรณีข้อมูลถูกต้อง

- กรณีชื่อผู้ใช้ และรหัสผ่านไม่ถูกต้อง



รูปที่ 9-5 ขั้นตอน ของการ Sign in กรณีข้อมูลไม่ถูกต้อง

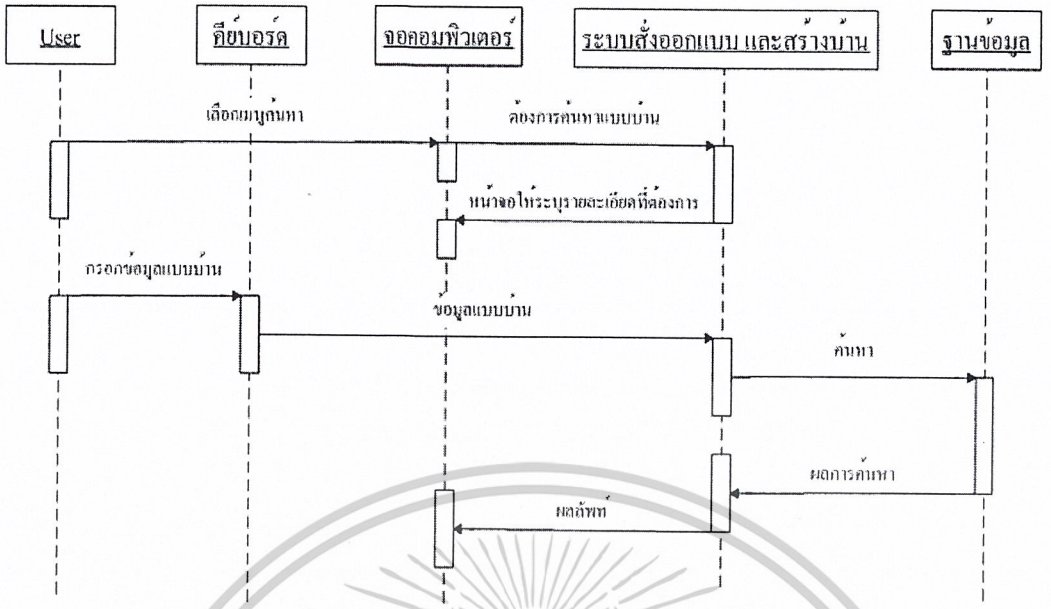
ขั้นตอนของการดูแบบบ้าน และบ้าน



รูปที่ 9-6 ขั้นตอน ของการเข้าดูแบบบ้าน

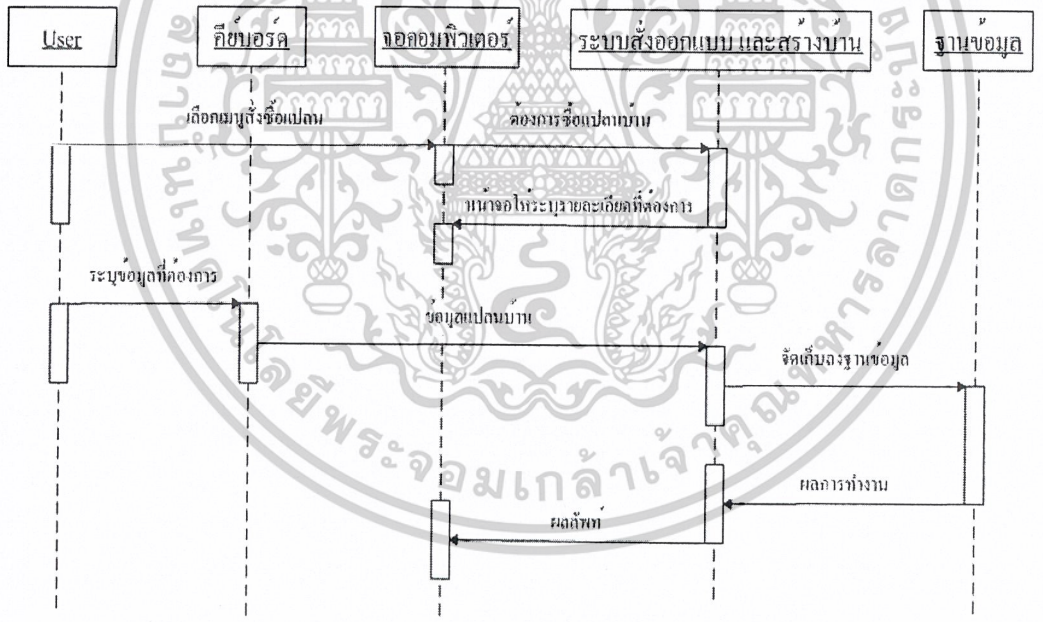
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการค้นหาแบบบ้าน



รูปที่ 9-7 ขั้นตอน ของการค้นหาแบบบ้าน

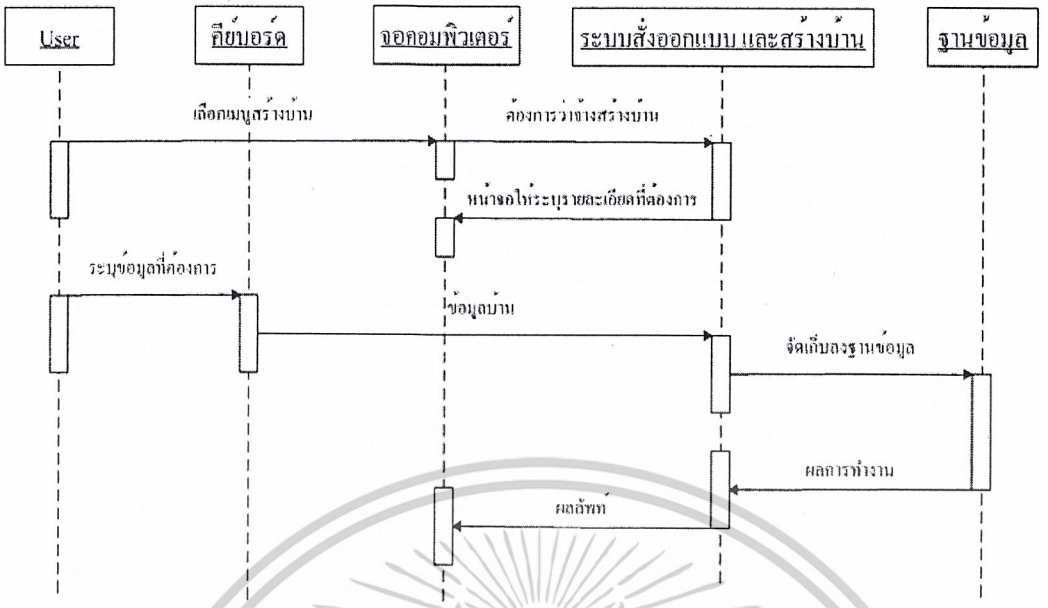
ขั้นตอนการสั่งซื้อแบบบ้าน



รูปที่ 9-8 ขั้นตอน ของการสั่งซื้อแบบบ้าน

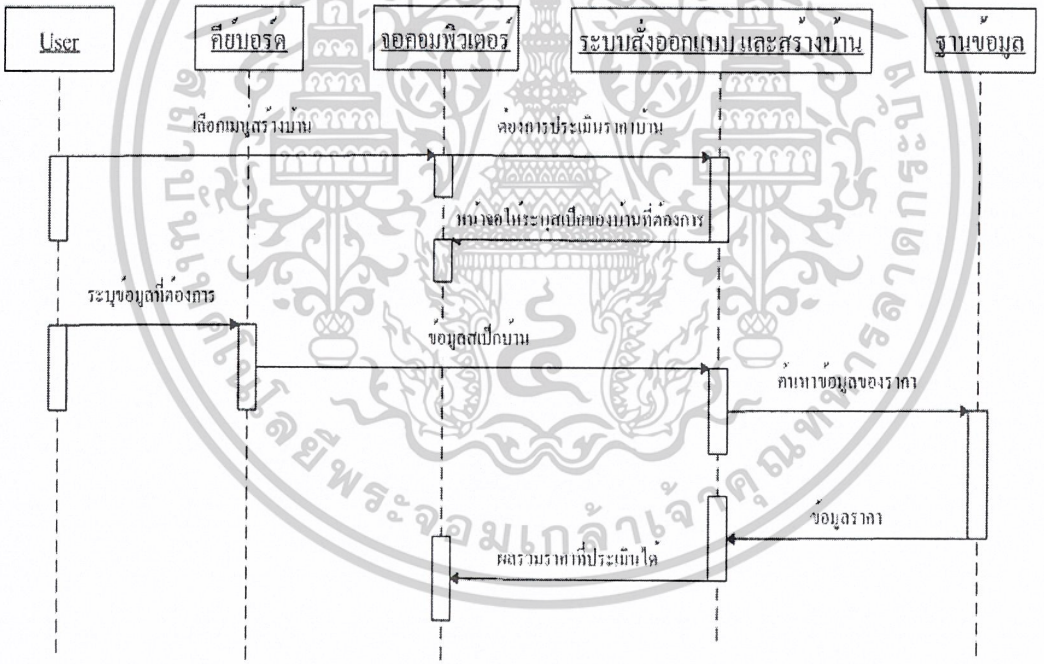
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนว่าจ้างสร้างบ้าน



รูปที่ 9-9 ขั้นตอน ของการว่าจ้างสร้างบ้าน

ขั้นตอนการประเมินราคาบ้าน

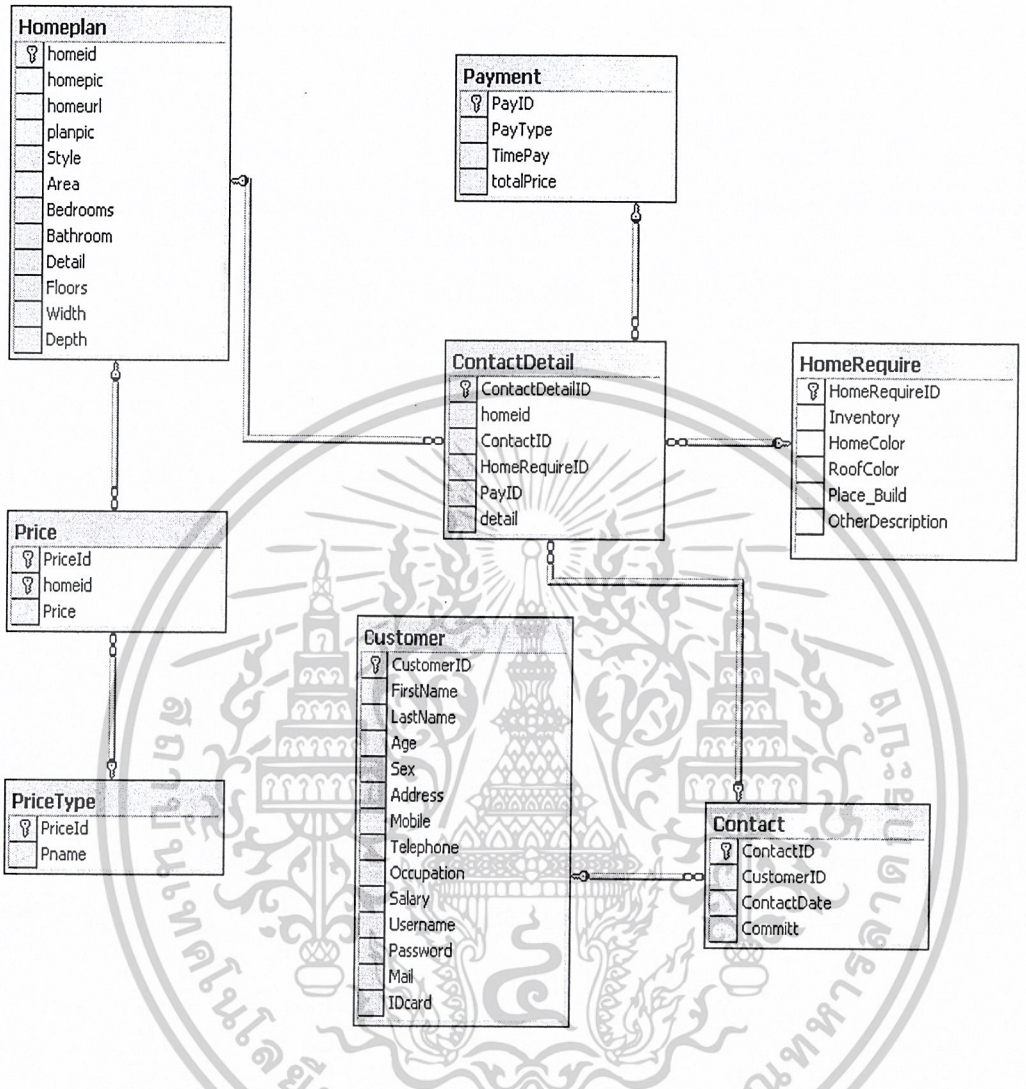


รูปที่ 9-10 ขั้นตอน ของการประเมินราคาบ้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.3 การออกแบบระบบฐานข้อมูล

อีอา-ไดอะแกรม (ER-Diagram) แสดงความสัมพันธ์ของตารางทั้งหมดในฐานข้อมูล

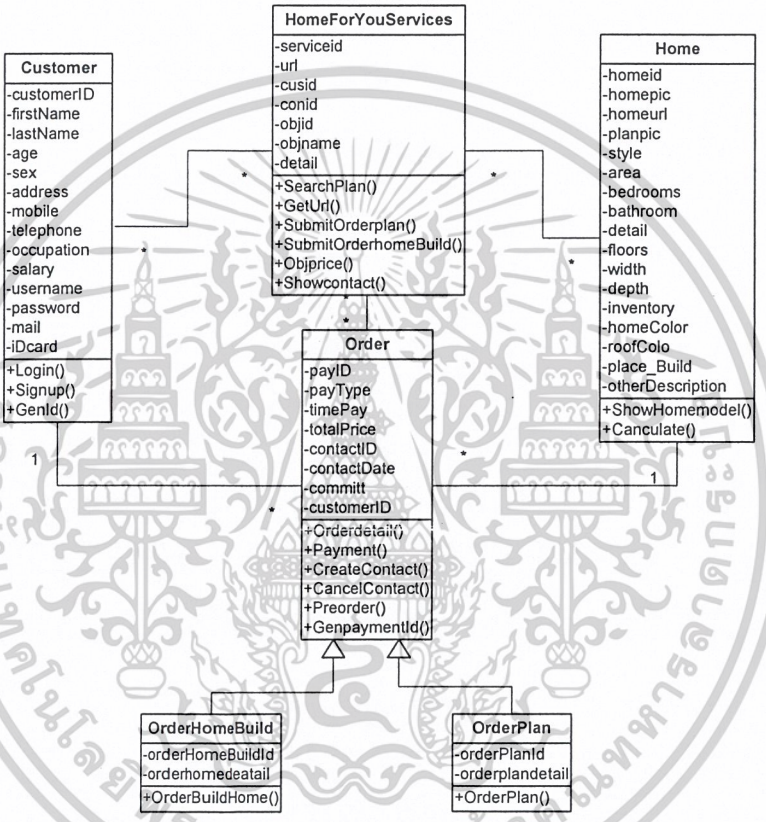


รูปที่ 9-11อีอา-ไดอะแกรมความสัมพันธ์ของตารางทั้งหมดในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.4 การออกแบบคลาสไดอะแกรม

คลาสไดอะแกรม (Class-diagram) ของระบบ Home For You



รูปที่ 9-12 คลาสไดอะแกรม ของระบบ Home For You

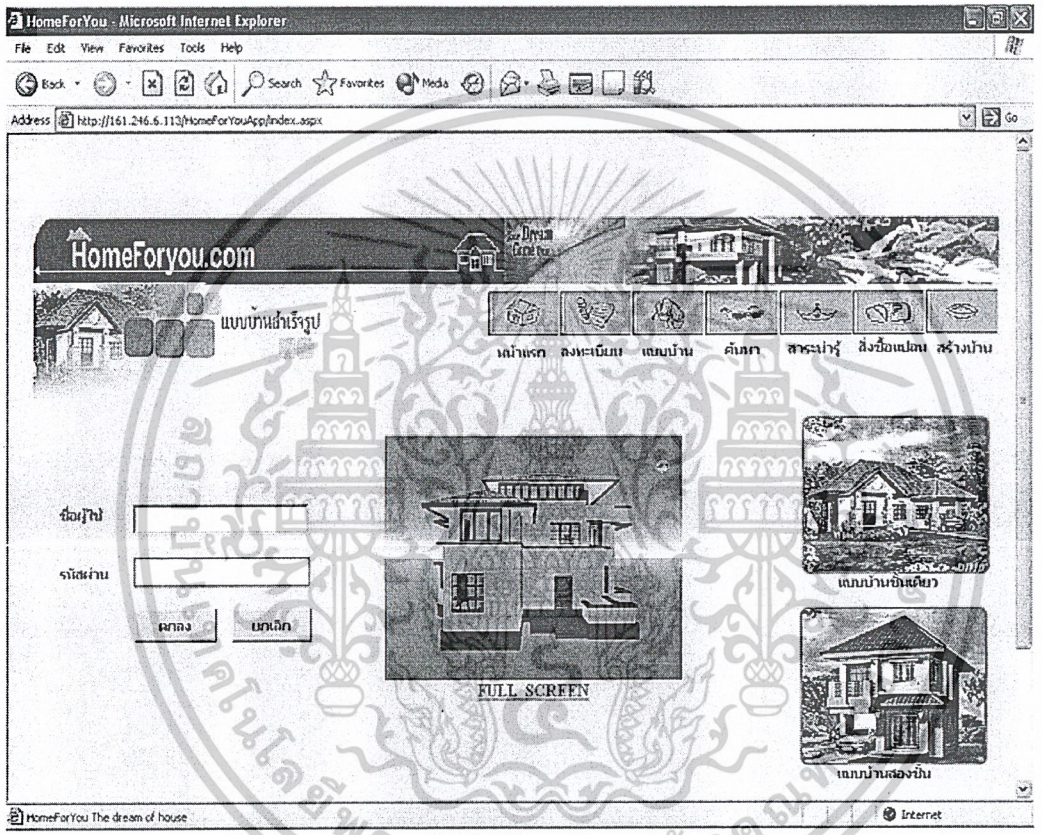
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 10
ลักษณะของแอพลิเคชัน

10.1 เว็บไซต์ของระบบออกแบบ และสร้างบ้าน (Home For You)

◆ หน้าแรก

ในหน้าแรกจะมีช่องสำหรับล็อกอินสำหรับลูกค้าที่เป็นสมาชิก และเมนูต่างๆ ให้เลือกรวมถึงแสดงแบบบ้านที่แนะนำในรูปแบบของโมเดล 3 มิติ

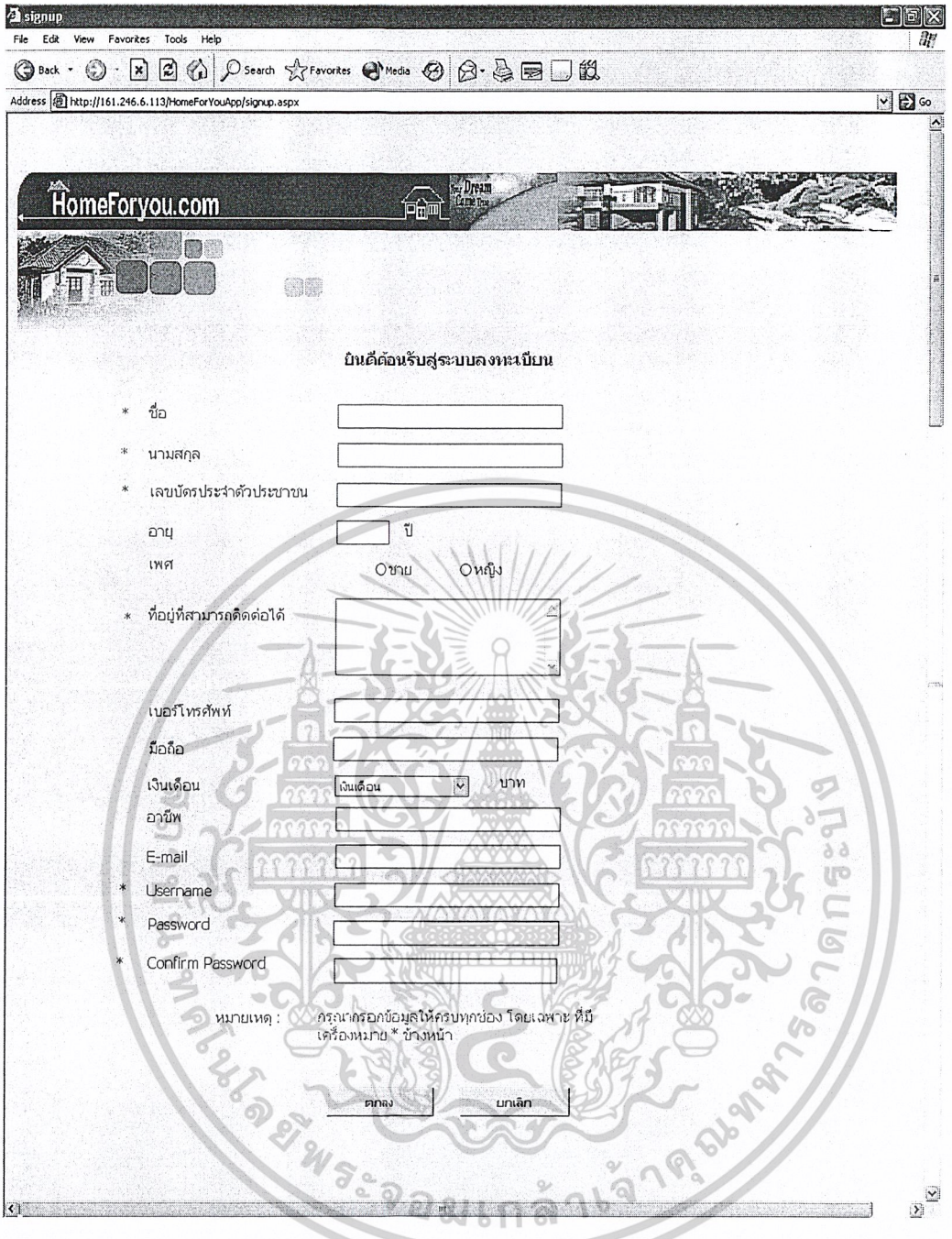


รูปที่ 10.1 หน้าแรก

◆ การลงทะเบียนกับทางระบบ

กรณีที่ผู้ใช้งานระบบยังไม่เป็นสมาชิกกับทางระบบ จะต้องลงทะเบียนเป็นสมาชิกก่อนจึงสามารถทำธุรกรรมต่างๆ กับทางระบบได้ โดยเลือกที่เมนูลงทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

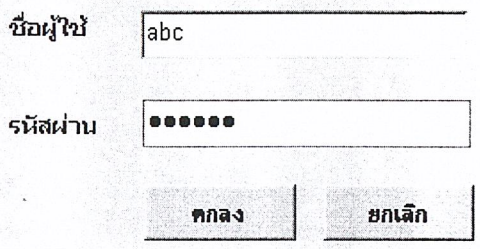


รูปที่ 10.2 แบบฟอร์มการลงทะเบียน

◆ การล็อกอินเข้าสู่ระบบ

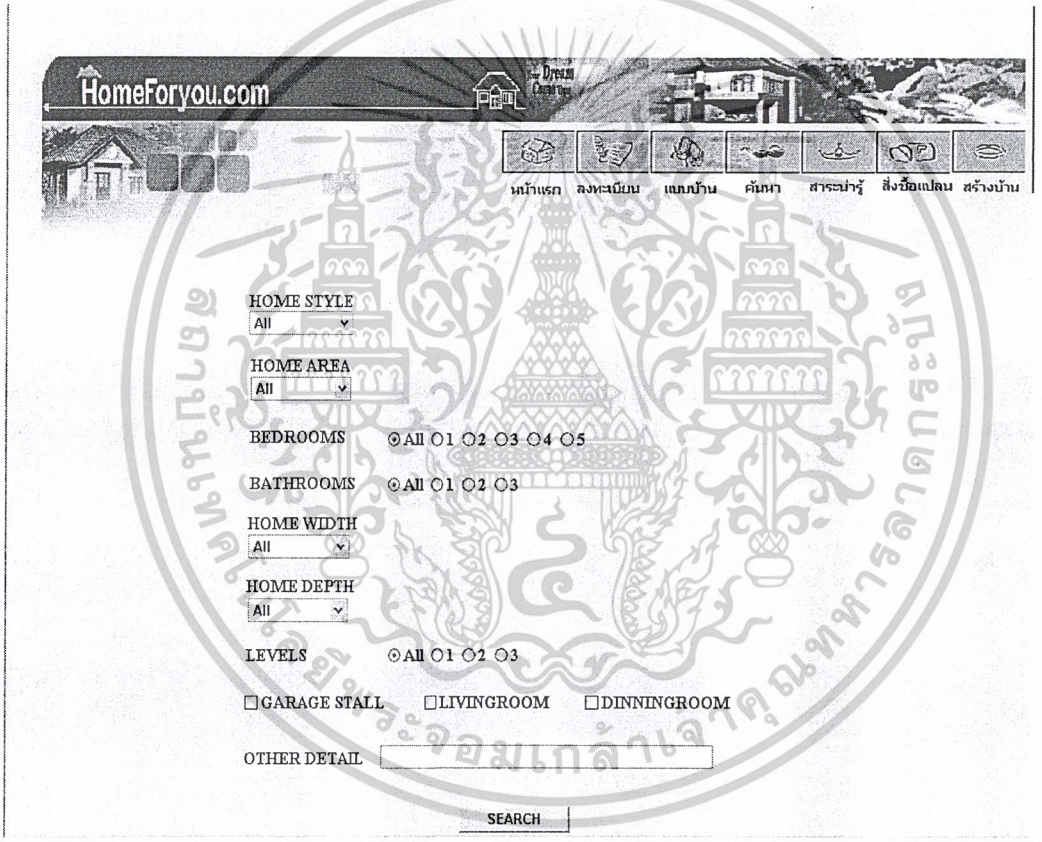
หากเป็นสมาชิกกับทางระบบแล้ว ผู้ใช้งานระบบจะต้อง ล็อกอินเข้าสู่ระบบก่อน จึงสามารถเข้าทำธุระกรรมกับทางระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 10.3 การล็อกอินเข้าสู่ระบบ

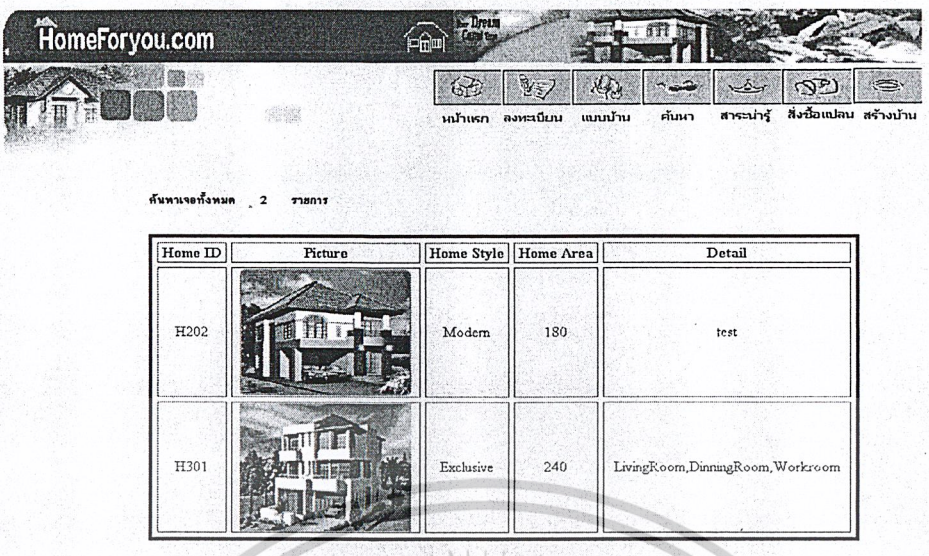
- การค้นหาแบบบ้านที่ต้องการ เลือกที่เมนูค้นหา



รูปที่ 10.4 การค้นหาแบบบ้านที่ต้องการ

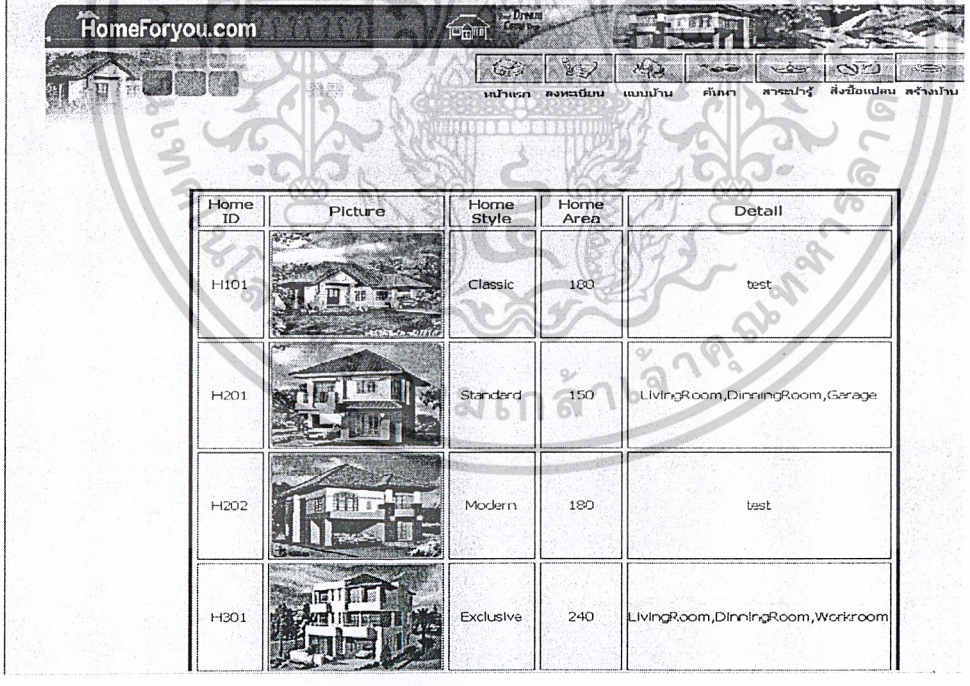
โดยกรอกข้อมูลเกี่ยวกับบ้านที่ต้องการค้นหา ได้แก่ สไตล์ ขนาดพื้นที่ใช้สอย จำนวนห้องนอน จำนวนห้องน้ำ ขนาดความกว้าง ขนาดความยาว จำนวนชั้น รายละเอียดอื่นๆ จากนั้นเมื่อคลิกค้นหาก็จะแสดงผลที่ได้ออกมา ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 10.5 ผลลัพธ์จากการค้นหา

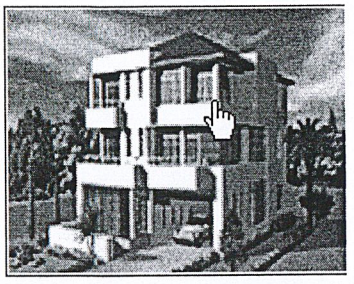
- การดูแบบบ้านทั้งหมด เลือกที่เมนูแบบบ้าน จะแสดงแบบบ้านทั้งหมด



รูปที่ 10.6 แสดงแบบบ้านทั้งหมด

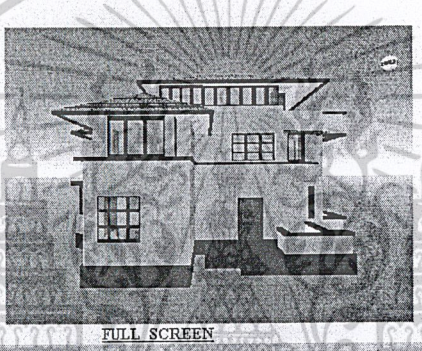
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

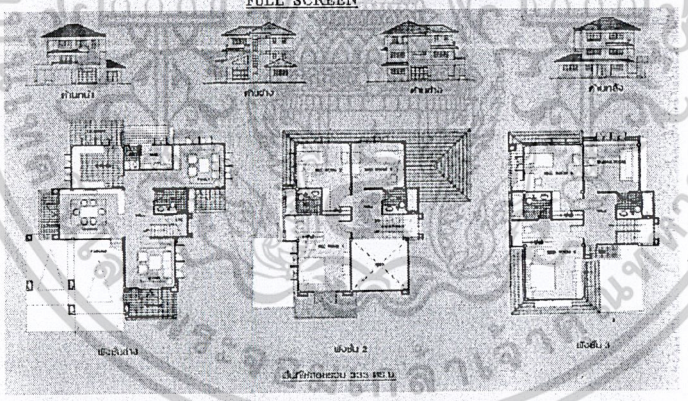
◆ เมื่อต้องการดูรายละเอียดของบ้านแต่ละหลัง คลิกที่รูปบ้าน



รูปที่ 10.7 เมื่อต้องการดูรายละเอียดของบ้าน

ซึ่งจะแสดงรายละเอียดของบ้าน ได้แก่โมเดลบ้านที่เป็นรูปแบบ 3 มิติโดยใช้ภาษา VRML รูปแปลนบ้าน รวมถึงราคาของแบบบ้านรูปแบบต่างๆ



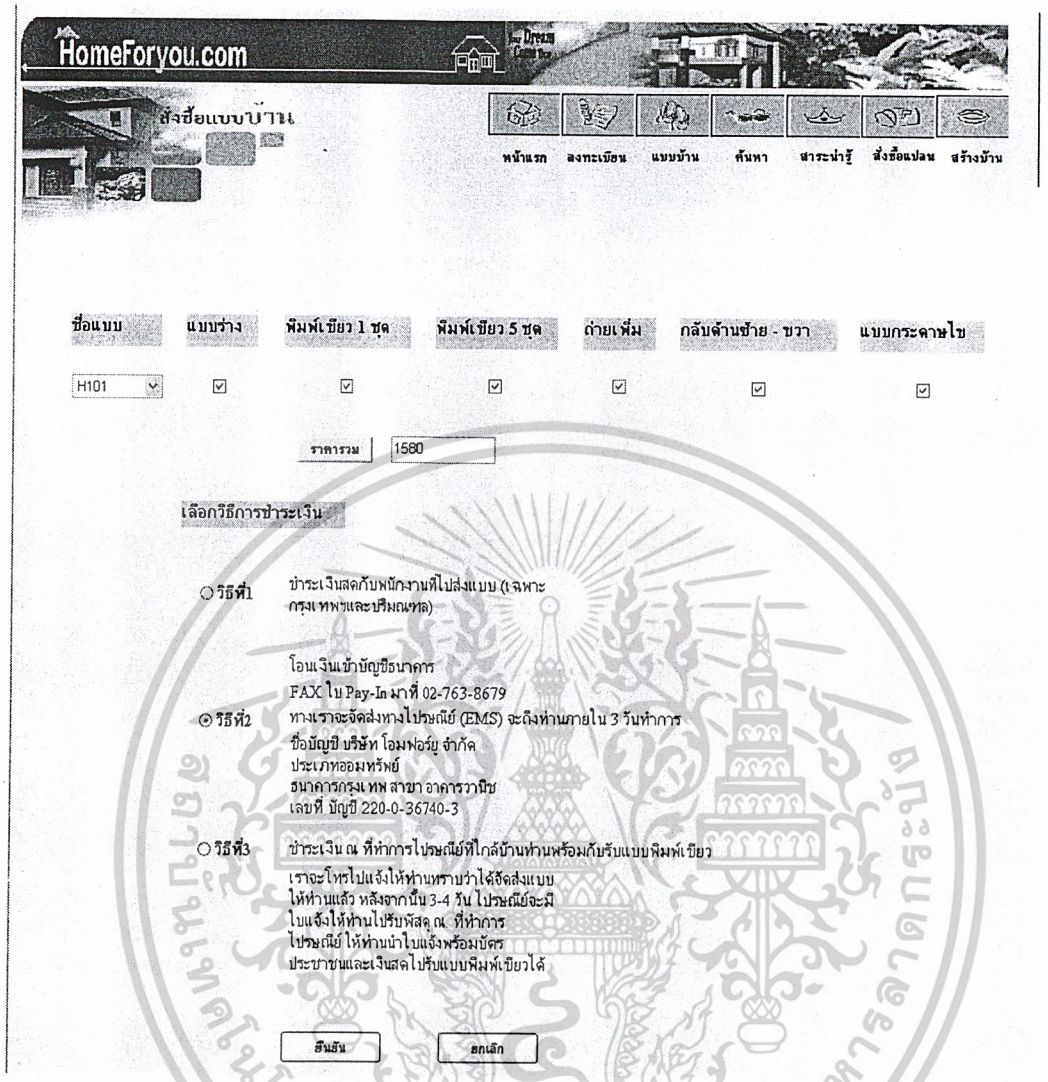


HomeID	ราคาค่าก่อสร้างClassA (ล้านบาท)	ราคาค่าก่อสร้างClassB (ล้านบาท)	ราคาค่าก่อสร้างClassC (ล้านบาท)	แบบกระดาษไข (บาท)	พิมพ์เขียว 1 ชุด (บาท)	พิมพ์เขียว 5 ชุด (บาท)	ถ่ายเพิ่ม (บาท)	แบบร่าง (บาท)	แบบกลับด้านซ้าย-ขวา (บาท)
H301	15	13	11	300	1000	1000	5000	500	600

รูปที่ 10.8 แสดงรายละเอียดของแบบบ้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

◆ การสั่งซื้อแปลนบ้าน



รูปที่ 10.9 แสดงรายละเอียดการสั่งซื้อแบบบ้าน

จะมีรายละเอียดให้ลูกค้าเลือกชนิดของแบบบ้านที่ต้องการ และมีการคำนวณราคารวมของแบบบ้านที่ลูกค้าเลือก รวมถึงรูปแบบของวิธีการชำระเงิน

- ชนิดของแบบบ้าน
  1. *แบบพิมพ์เขียว 5 ชุด* สำหรับผู้ที่เห็นรูปบ้านจำลองและแบบแปลนรวมถึงฟังก์ชั่นการใช้งานต่างๆ แล้วมีความพร้อมที่จะสร้างบ้าน ท่านก็สามารถนำแบบชุดนี้ไปยื่นขออนุญาติกับทางราชการ และใช้ก่อสร้างได้เลย เมื่อได้ใบอนุญาตแล้วทางราชการจะคืนแบบมาให้ 2 ชุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. *แบบร่าง*

1.) สำหรับผู้ที่เห็นรูปบ้านจำลอง และแบบแปลนแล้วมีความถูกใจ แต่อยากจะเปลี่ยนแปลงแก้ไข หรือตำแหน่งของห้อง โดยท่านสามารถนำแบบ ไปพิจารณาได้อย่างละเอียด หลังจากท่านมีข้อสรุปที่ลงตัวแล้ว ก็จะกลับมาสู่ขั้นตอน การแก้ไขแบบ กับทางเราต่อไป

2.) สำหรับผู้ที่เห็นรูปบ้านจำลอง และแบบแปลนแล้วมีความถูกใจ แต่มีงบประมาณไม่เพียงพอ ท่านสามารถนำแบบนี้ ไปขอทราบวงเงินกู้กับทางธนาคาร อย่างไม่เป็นทางการได้

3. *แบบกลับด้านซ้าย-ขวา* สำหรับผู้ที่เห็นรูปบ้านจำลองและแบบแปลนแล้วมีความถูกใจ แต่อยากจะสลับที่จอดรถ หรือเฉลียงหน้าบ้านให้ไปอยู่ ด้านตรงข้ามกับที่เห็นในรูปบ้าน

4. *ถ่ายแบบเพิ่ม* สำหรับผู้ที่เห็นแบบจำนวน 1 ชุด หรือ 5 ชุด ยังไม่พอกับความต้องการ เช่น ต้องการเอาไปให้ผู้รับเหมาคิดราคามาเปรียบเทียบกัน

5. *แบบกระดาษไข* สำหรับผู้ที่เห็นรูปบ้านจำลอง และแบบแปลนแล้วมีความถูกใจ และต้องการแบบจำนวนหลายชุด หรือท่านพอมีความรู้ที่จะเปลี่ยนแปลงแก้ไขเอง ท่านก็สามารถนำแบบกระดาษไขไปถ่ายแบบพิมพ์เขียวเองได้

- รูปแบบการจ่ายเงิน มี 3 วิธี

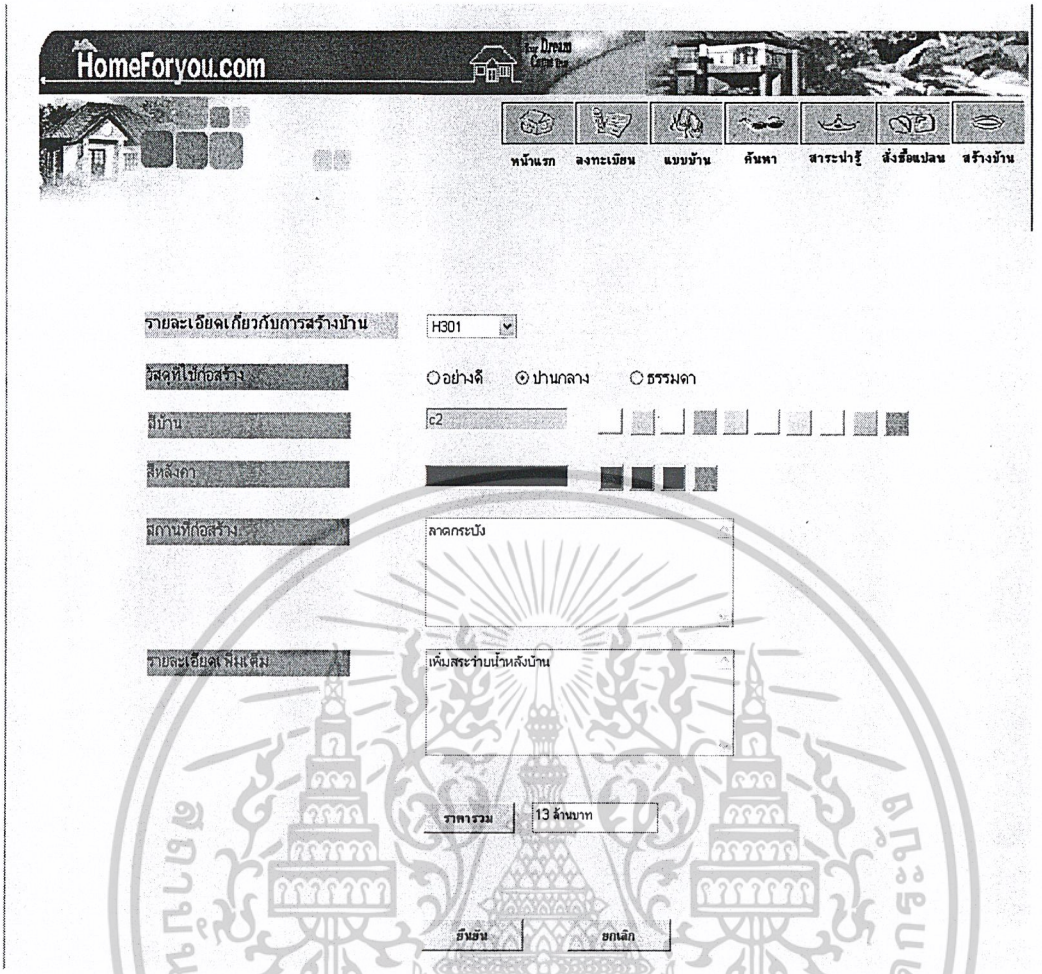
วิธีที่ 1 ชำระเงินสดกับพนักงานที่ไปส่งแบบ (เฉพาะกรุงเทพฯและปริมณฑล)

วิธีที่ 2 โอนเงินเข้าบัญชีธนาคาร FAX ใบ Pay-In มาที่ 02-763-8679 ทางเราจะจัดส่งทางไปรษณีย์ (EMS) จะถึงท่านภายใน 3 วันทำการชื่อบัญชี บริษัท โอมฟอร์ยู จำกัด ประเภท ออมทรัพย์ ธนาคารกรุงเทพ สาขา อาคารวานิช เลขที่ บัญชี 220-0-36740-3

วิธีที่ 3 ชำระเงิน ณ ที่ทำการไปรษณีย์ที่ใกล้บ้านท่านพร้อมกับรับแบบพิมพ์เขียว เราจะโทรไปแจ้งให้ท่านทราบว่าได้จัดส่งแบบให้ท่านแล้ว หลังจากนั้น 3-4 วัน ไปรษณีย์จะมีใบแจ้งให้ท่านไปรับพัสดุ ณ. ที่ทำการไปรษณีย์ ให้ท่านนำใบแจ้งพร้อมบัตรประชาชนและเงินสดไปรับแบบพิมพ์เขียวได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อลูกค้ามีความต้องการที่จะสร้างบ้านกับทางบริษัท



รูปที่ 10.10 แสดงรายละเอียดการสร้างบ้าน

เมื่อลูกค้าได้แบบบ้านที่ต้องการแล้วมีความต้องการที่จะสร้างบ้านกับทางบริษัท จะมีข้อมูลที่ลูกค้าจะต้องกรอกเพิ่มเติมจากรายละเอียดที่มีอยู่แล้วในแบบบ้านที่เลือก เช่น คุณสมบัติและระดับของวัสดุที่จะใช้ในการก่อสร้าง สีบ้าน สีหลังคา สถานที่ก่อสร้าง หรือรายละเอียดต่างๆเพิ่มเติมที่ลูกค้าต้องการ จากนั้นจะมีการคำนวณราคาค่าก่อสร้างทั้งหมดอย่างคร่าวๆ โดยอิงตามแบบบ้านและคุณสมบัติของวัสดุ

10.2 ส่วนที่เปิดเป็นเว็บเซอร์วิสให้บริการกับเว็บหรือบริษัทอื่น

โดยจะมีเซอร์วิสที่เปิดไว้เพื่อให้เว็บอื่นรวมถึงเว็บแอพลิเคชันได้เรียกใช้ ได้แก่

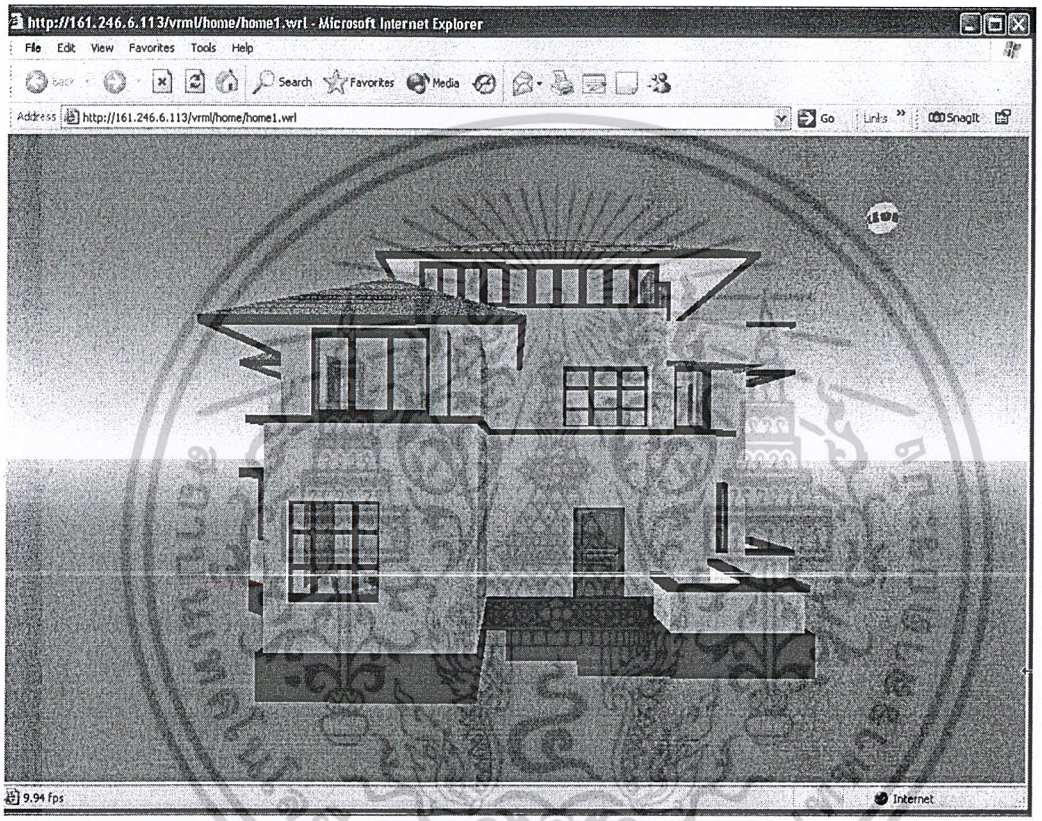
- เซอร์วิสที่เกี่ยวกับการสั่งซื้อแบบบ้าน
- เซอร์วิสที่เกี่ยวกับการสร้างบ้าน
- เซอร์วิสที่เกี่ยวกับการค้นหาแบบบ้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เซอร์วิสที่เกี่ยวกับระบบสมาชิก
- เซอร์วิสที่เกี่ยวกับการนำเอาโมเดลบ้านไปแสดงบนหน้าเว็บ
- เซอร์วิสที่เกี่ยวกับการดูราคาของแบบบ้าน

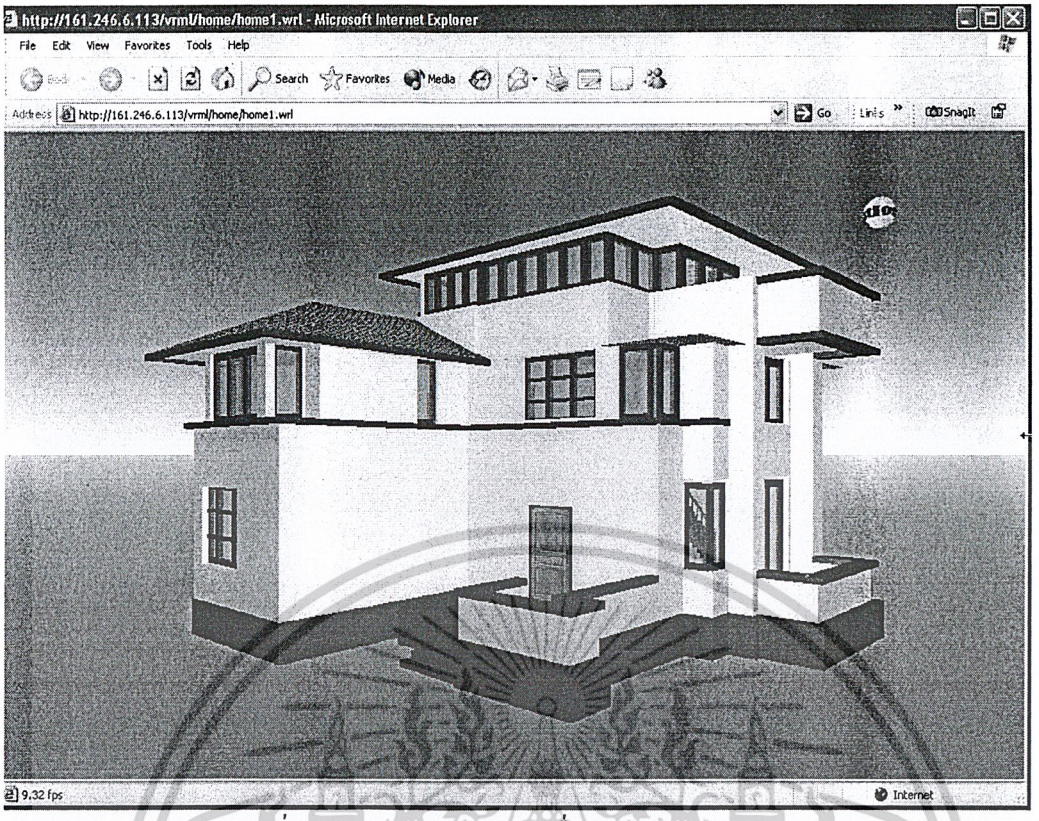
10.3 โมเดลบ้าน VRML 3 มิติ

ระบบ Home For You มีส่วนการนำเสนอโมเดลบ้านในรูปแบบของ โมเดล 3 มิติ โดยใช้ภาษา VRML ซึ่งสามารถโต้ตอบกับลูกค้าได้(Interactive)



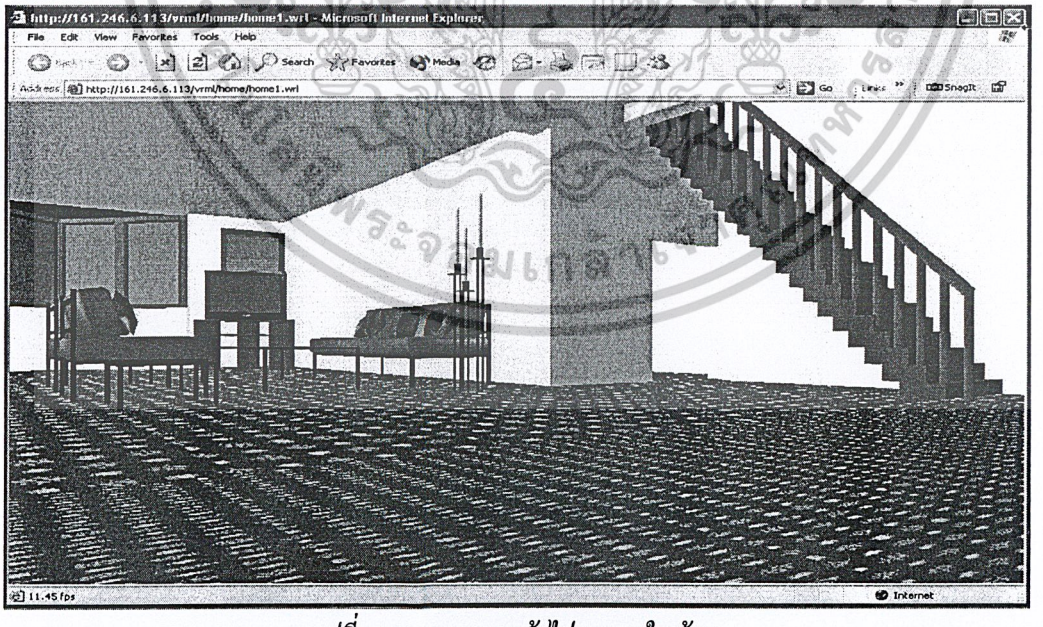
รูปที่ 10.11 โมเดลบ้าน 3มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



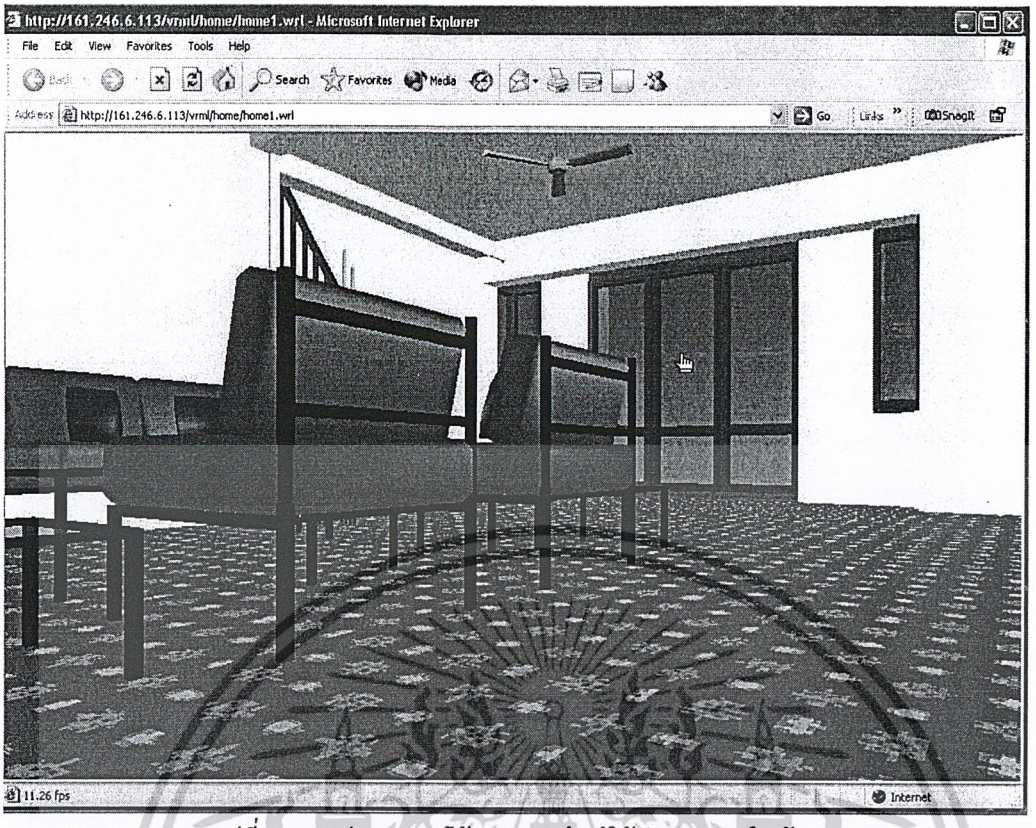
รูปที่ 10.12 การหมุนโมเดลบ้านเพื่อดูบ้านในมุมมองต่างๆ

จากรูปที่ 10.11 เป็นการหมุนโมเดลบ้านซึ่งทำให้ลูกค้าสามารถดูโมเดลบ้านได้จากมุมมองต่างๆ นอกจากนี้ลูกค้ายังสามารถเข้าไปชมภายในบ้านตามห้องต่างๆได้อีกด้วย เสมือนกับว่าลูกค้าได้เข้าไปชมในบ้านจริงๆ



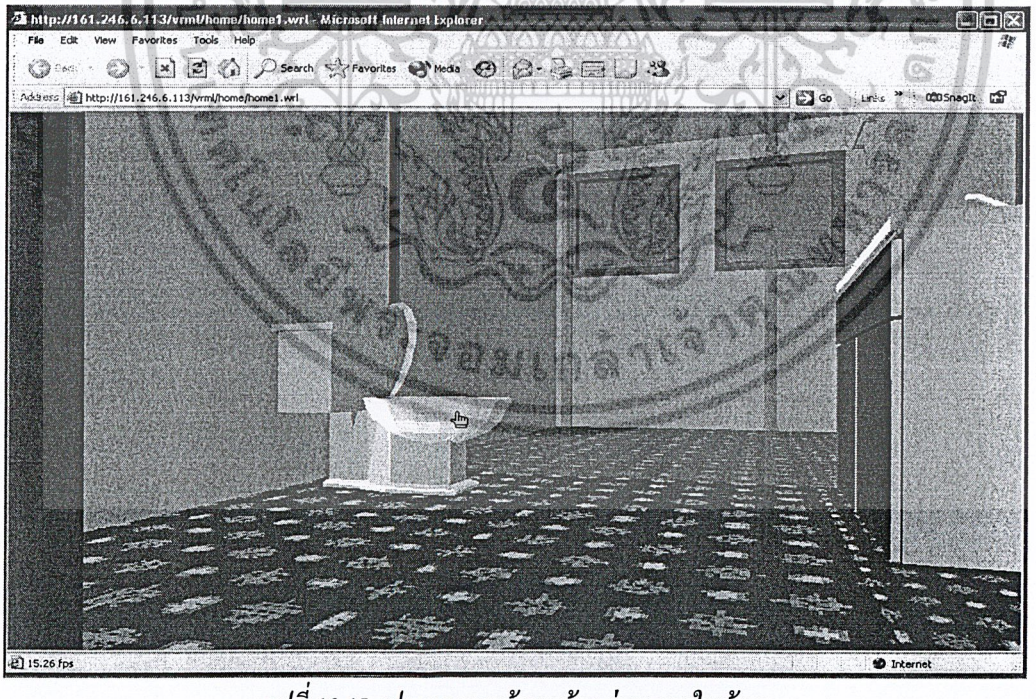
รูปที่ 10.13 แสดงการเข้าไปชมภายในบ้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



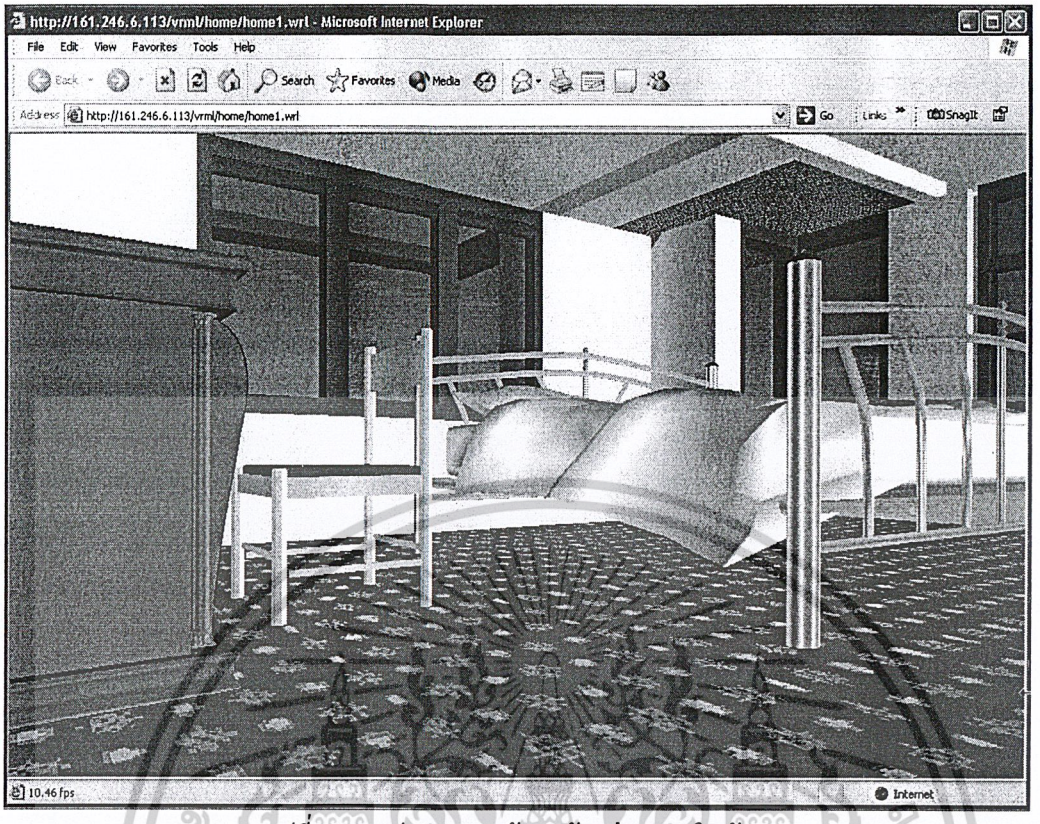
รูปที่ 10.14 รูปแสดงการโต้ตอบระหว่างผู้ใช้กับวัตถุภายในบ้าน

นอกจากนี้แล้ววัตถุภายในบ้านยังสามารถโต้ตอบกับลูกค้าได้ ทำให้มีความสมจริงมากยิ่งขึ้นด้วย



รูปที่ 10.15 รูปแสดงการเข้าชมห้องต่างๆภายในบ้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 10.16 รูปแสดงการเข้าชมห้องต่างๆภายในบ้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 11
บทวิจารณ์ และสรุป

11.1 สรุปผลการดำเนินงาน

ในโครงงานนี้ได้แบ่งการดำเนินงานออกเป็น 4 ส่วนหลัก คือ

1. ศึกษาทฤษฎีพื้นฐานของ คอมโพเน็นต์, .NET เฟรมเวิร์ก, ASP.NETเว็บเซอร์วิส, VRML และเทคโนโลยีอื่นๆ ที่เกี่ยวข้อง

2. ออกแบบระบบงาน เพื่อทำการทดสอบทฤษฎีต่างๆ ที่ได้ศึกษามา ซึ่งระบบงานที่สร้างขึ้นมาคือ ระบบโฮมฟอร์ยู(Home For You),ระบบโฮมแอนด์เดคอร์ (Home&Decor) และระบบแลนแอนด์โฮม (Land and Home) โดยส่วนที่รับผิดชอบคือ ระบบโฮมฟอร์ยู (Home For You) ซึ่งเป็นระบบซื้อขายแบบบ้าน และการสร้างบ้าน

3. ทำการอิมพลีเมนต์ระบบงานที่ได้ออกแบบไว้ โดยใช้ภาษา VB.NET, VB Scripts, HTML และภาษาVRML ซึ่งพัฒนาตามมาตรฐานของ เว็บเซอร์วิส และอิงตามแพลตฟอร์ม .NET

4. ทำการสร้างรูปแบบการนำเสนอที่ใช้ระบบ 3 มิติ ของภาษา VRML โดยสร้างเป็นโมเดลบ้าน 3 มิติ และนำไปใช้งานบนเว็บ

11.2 แนวทางการพัฒนาต่อ

การออกแบบและพัฒนาระบบเว็บเซอร์วิสนั้น มีแนวโน้วที่จะมีการใช้งานอย่างกว้างขวางในอนาคตเนื่องจากมีความสามารถในด้านต่างๆ รวมถึงกระแสของระบบอีคอมเมอร์ซ ที่มีเพิ่มมากขึ้น ซึ่งสามารถนำเอาเว็บเซอร์วิสไปประยุกต์ใช้ได้ ระบบเว็บเวอร์วิสจึงเป็นทางเลือกที่น่าศึกษาและพัฒนาต่อไป

ภาษา VRML ก็เป็นอีกทางเลือกหนึ่งที่น่าสนใจในการนำเสนอรูปแบบของ 3 มิติบนเว็บเนื่องจากให้ความสมจริงและมีลักษณะเด่น ทั้งทางด้านแสง เสียง ในระบบ 3 มิติ ซึ่งน่าจะเป็นกระแสใหม่ในอนาคตได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.
การติดตั้ง Visual Studio .net

1. ความต้องการของระบบ

สำหรับการติดตั้งโปรแกรม Visual Studio .net นั้น เราควรตรวจสอบเครื่องคอมพิวเตอร์ของเราให้พร้อมด้วย โดยแบ่งความต้องการของระบบออกเป็น 2 ด้าน คือ ฮาร์ดแวร์ และซอฟท์แวร์

ความต้องการของระบบด้านฮาร์ดแวร์

ส่วนประกอบทางด้านฮาร์ดแวร์สำหรับเครื่องที่ต้องการติดตั้ง Visual Studio .net มีความต้องการขั้นต่ำดังต่อไปนี้

- หน่วยประมวลผลกลาง (CPU) Pentium II 450 MHZ เป็นอย่างน้อย (แนะนำให้ใช้เครื่อง Pentium III 600 MHZ ขึ้นไป)
- หน่วยความจำ (RAM) ขั้นต่ำ 96 MB สำหรับ Windows 2000 Professional 192 MB สำหรับ Windows 2000 Server, 64 MB สำหรับ Window NT 4.0 Workstation, 160 MB สำหรับ Windows NT 4.0 Server และ 160 MB สำหรับ Windows XP Professional
- เนื้อที่ในฮาร์ดดิสก์ 3 GB
- การ์ดแสดงผล 800 x 600 , 256 สี
- ไดรฟ์ CD-ROM หรือ DVD-ROM

ส่วนบ่ระกอบของ Visual Studio .net ต้องการระบบฮาร์ดแวร์ที่ค่อนข้างสูง ถ้าเรามีระบบฮาร์แวร์ที่ไม่เหมาะสม เราอาจจะพบความล่าช้าในการพัฒนาโปรแกรม เช่น ความเร็วของซีพียู หรือหน่วยความจำที่น้อยเกินไป เป็นต้น

ความต้องการทางด้านซอฟท์แวร์

สำหรับระบบปฏิบัติการที่รองรับการติดตั้ง Visual Studio .net นั้นมีดังต่อไปนี้

- Windows 2000
- Windows XP
- Windows NT 4.0 (ไม่สนับสนุนการสร้างแอพพลิเคชั้น ASP)

สำหรับการสร้างเว็บแอพพลิเคชั่นที่ใช้เทคโนโลยี ASP หรือ XML Web Services เราจำเป็นต้องติดตั้ง Internet Information Service (IIS) ก่อนที่จะติดตั้ง Visual Studio .net โดยโปรแกรมนี้เป็นโปรแกรมเสริมที่มาพร้อมกับแผ่นติดตั้ง Windows 2000 หรือ Windows XP อยู่แล้ว เราสามารถลงมือติดตั้ง IIS ได้ดังขั้นตอนต่อไปนี้ (ในที่นี้จะยกตัวอย่างการติดตั้ง IIS บน Windows XP รุ่น Professional ซึ่งบน Windows 2000 ก็มีขั้นตอนการติดตั้งที่คล้ายกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ใส่แผ่นติดตั้ง Windows XP ลงในไดร์ฟซีดีรอม
2. Clice mouse ที่ปุ่ม Start เลือกเมนู Settions>Control Panel
3. จากหน้าต่าง Control Panel ให้เรา Double click ที่ไอคอน Add or Remove Programs
4. จะปรากฏหน้าต่าง Add or Remove Programs ให้เรา Click mouse ที่ปุ่ม Add/Remove Windows Components
5. จากหน้าจอ Windows Components Wizard ที่ปรากฏขึ้น ให้เรา Click mouse เลือก Internet Information Service (IIS) และ Click mouse ที่ปุ่ม Details
6. จากหน้าต่าง Internet Information Service (IIS) หาก FrontPage 2000 Server Extensions ยังไม่ได้ถูกเลือก ก็ให้เลือกด้วย แล้ว Click mouse ที่ปุ่ม OK
7. หลังจากนั้นเครื่องจะทำการติดตั้ง IIS จนเสร็จ ให้เรา Click mouse ที่ปุ่ม Finish เป็นอันจบขั้นตอน

การติดตั้ง Visual Studio .net

ในการติดตั้ง Visual Studio .net นั้น จะมีแผ่นซีดีติดตั้งทั้งหมด 5 แผ่นด้วยกัน คือ CD1, CD2, CD3, CD4 และแผ่นสุดท้ายเป็นแผ่นซีดี Windows Component Update ในขั้นตอนการติดตั้งนั้นโปรแกรมจะให้เราติดตั้งส่วนประกอบย่อยที่จำเป็นต่อการใช้งาน (Windows Component Update) ก่อน

ให้เราเริ่มต้นโดยใส่แผ่นซีดี Windows Component Update ลงในไดร์ฟซีดีรอม จากนั้นเครื่องจะรันโปรแกรมติดตั้งขึ้นมาโดยอัตโนมัติ (หากโปรแกรมติดตั้งไม่แสดงขึ้นมา ให้เราเข้าไปดับเบิ้ลคลิ๊กที่ไฟล์ setup.exe

จากหน้าต่าง Visual Studio .net Setup โปรแกรมจะบอกถึงขั้นตอนในการติดตั้ง Visual Studio .net ดังต่อไปนี้

ขั้นตอนที่ 1 Windows Component Update : เป็นการติดตั้งส่วนประกอบเพิ่มเติมที่ใช้ทำงานร่วมกับ Visual Studio .net

ขั้นตอนที่ 2 ติดตั้ง Visual Studio .net หลังจากติดตั้งส่วนประกอบเพิ่มเติมบน Windows แล้ว

ขั้นตอนที่ 3 สำหรับการติดตั้งเกี่ยวกับส่วนประกอบเพิ่มเติมที่อาจมีมาภายหลัง

ติดตั้ง Windows Component Update

ขั้นตอนแรกก่อนการติดตั้ง Visual Studio .net เราต้องทำการเลือกติดตั้งส่วนประกอบย่อยเพิ่มเติมบน Windows ที่ใช้ทำงานร่วมกับ Visual Studio .net ก่อน โดยคลิ๊กเมาส์ ที่เมนู Windows Component Update จากหน้าต่างหลักของการติดตั้ง

1. จากหน้าต่างติดตั้งหลักของ Visual Studio ให้เลือกเมนู Windows Component Update เพื่อเลือกติดตั้งส่วนประกอบ Windows ที่จำเป็นเพิ่มเติม

2. เลือก I accept the agreement เพื่อยอมรับข้อตกลงในการใช้งาน หลังจากนั้นให้ คลิ๊กเมาส์ที่ Continue

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. โปรแกรมติดตั้งจะแสดงโปรแกรมเพิ่มเติมทั้งหมดที่จะติดตั้ง โดยมีรายละเอียดดังนี้

- Windows 2000 Service Pack 2
- Microsoft Windows Installer 2.0
- Microsoft FrontPage 2000 Web Extensions Client
- Setup Runtime Files
- Microsoft Internet Explorer 6.0 and Internet Tools
- Microsoft Data Access Components 2.7
- Microsoft .NET Framework

ให้เราคลิ๊กเมาส์ที่ Continue

4. ในการติดตั้งนั้นอาจมีการรีสตาร์ทเครื่องขึ้นใหม่ โปรแกรมจะให้เรากำหนดให้เครื่องสามารถล็อกออนเข้าสู่ Windows โดยอัตโนมัติหลังจากรีสตาร์ทเครื่อง โดยเลือกที่ Automatically log on และกำหนดรหัสผ่านที่ใช้สำหรับการล็อกออน จากนั้นให้คลิ๊กเมาส์ที่ Install Now! เพื่อเริ่มต้นการติดตั้ง

5. โปรแกรมจะเริ่มต้นติดตั้งส่วนประกอบเพิ่มเติมสำหรับ Windows ตามต้องการ หลังจากติดตั้งเสร็จเรียบร้อยแล้ว ให้เราคลิ๊กเมาส์ที่ Done

จากนั้นจะกลับเข้าสู่หน้าต่างหลักของการติดตั้งโปรแกรม Visual Studio .NET

เริ่มต้นติดตั้ง Visual Studio .net

หลังจากการติดตั้ง Windows Component Update เรียบร้อยแล้ว ต่อไปเราสามารถเริ่มต้นติดตั้ง Visual Studio .net ได้ โดยคลิ๊กเมาส์เลือกเมนู Visual Studio .NET

1. จากหน้าต่างหลักของการติดตั้ง Visual Studio .net ให้เราคลิ๊กเมาส์เลือกเมนู Visual Studio .net เพื่อเริ่มการติดตั้งโปรแกรม

2. โปรแกรมติดตั้งจะเรียกหาแผ่นซีดีติดตั้ง Visual Studio .net แผ่นที่ 1 ให้เราเอาซีดีติดตั้ง Windows Component Update ออกแล้วใส่ซีดีติดตั้งแผ่นแรกเข้าไปแทน จากนั้นคลิ๊กเมาส์ที่ปุ่ม OK

3. จากหน้าต่างติดตั้ง ให้เราเลือก I accept the agreement เพื่อยอมรับข้อตกลงในการใช้งานโปรแกรม และกรอก Product Key จากนั้นให้คลิ๊กเมาส์ที่ Continue

4. ที่หัวข้อ Feature properties โปรแกรมติดตั้งจะแสดงรายละเอียดของพื้นที่ที่ใช้ในการติดตั้งและไดเร็กทอรี่สำหรับการติดตั้ง ให้เราคลิ๊กเมาส์ที่ Install Now! เพื่อเริ่มติดตั้งโปรแกรม

5. จากนั้นโปรแกรมติดตั้งจะเริ่มต้นติดตั้ง Visual Studio .net ลงบนเครื่อง โดยเมื่อติดตั้งไปซักพักจะปรากฏไดอะล็อกบ็อกซ์ถามหาแผ่นติดตั้งแผ่นต่อไป ให้เราเปลี่ยนใส่แผ่นติดตั้งลงไปตามลำดับและคลิ๊กเมาส์ที่ปุ่ม OK

6.หลังจากติดตั้งจนเสร็จสมบูรณ์แล้ว ให้เราคลิ๊กเมาส์ที่ปุ่ม Done

7. จะกลับเข้าสู่หน้าจอหลักของการติดตั้ง Visual Studio .net ให้เราคลิ๊กเมาส์ที่ปุ่ม Exit เพื่อออกจากโปรแกรม เป็นอันจบขั้นตอนการติดตั้ง Visual Studio .net

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.
แนะนำเว็บไซต์เพื่อศึกษาเพิ่มเติม

เว็บไซต์เกี่ยวกับเว็บเซอร์วิส

- **http://www-106.ibm.com/developerworks/webservices/** เป็นเว็บไซต์ของไอบีเอ็มที่จัดทำไว้สำหรับนักพัฒนาโดยเฉพาะ ซึ่งรวบรวมบทความ, tutorial, ความรู้ต่างๆ เกี่ยวกับเว็บเซอร์วิส, SOAP, WSDL และอื่นๆ ไว้อย่างครบครัน

- **http://msdn.microsoft.com** เป็นเว็บไซต์ของไมโครซอฟต์ที่มีไว้สำหรับนักพัฒนาเช่นกัน นอกจากเนื้อหาเกี่ยวกับเทคโนโลยีของไมโครซอฟต์แล้ว ยังรวมไปถึงเทคโนโลยีเว็บเซอร์วิส, .NET, XML เป็นต้น พร้อมทั้งมีบริการให้ดาวน์โหลดซอฟต์แวร์อย่าง SOAP Toolkit, .NET Framework, MSXML (XML Parser ของไมโครซอฟต์) ด้วย

- **http://www.sun.com/software/sunone/** เป็นเว็บไซต์ของบริษัทซันไมโครซิสเต็มส์ซึ่งเสนอข่าวคราวความเคลื่อนไหว เกี่ยวกับแพลตฟอร์ม Sun ONE

- **http://www.soaplite.com/** เป็นเว็บไซต์ที่ให้ดาว์นโหลดโมดูล SOAP Lite สำหรับนักพัฒนาภาษา Perl ส่วนใหญ่จะเน้นที่วิธีการใช้โมดูล SOAP Lite

- **http://xml.apache.org/soap/index.html** เป็น เว็บไซต์ของหน่วยงาน OpenSource อย่าง Apache ที่ให้ดาว์นโหลดโมดูล Apache SOAP สำหรับนักพัฒนาเว็บเซอร์วิสด้วยภาษา Java นอกจากนี้ที่ **http://xml.apache.org/soap/docs/index.html** ยังมีบทความให้อ่านอีกเพียบ

- **http://www.theserverside.com/** เป็นแหล่งรวมเทคโนโลยีทางฝั่งเซิร์ฟเวอร์ไว้มากมาย โดยเฉพาะที่น่าสนใจคือการเปรียบเทียบ .NET กับ J2EE หรือ Sun ONE นับเป็นอีกเว็บไซต์หนึ่งที่ดีมากๆ

เว็บไซต์เกี่ยวกับ .NET

- **http://www.gotdotnet.com/** เป็นเว็บไซต์ของไมโครซอฟต์ที่เสนอความรู้เกี่ยวกับเทคโนโลยี .NET มีบทความมากมาย แต่มักจะเน้นที่ผลิตภัณฑ์ของไมโครซอฟต์ เช่น .NET Framework, Visual Studio .NET เป็นต้น

- **http://www.devx.com/dotnet/resources/vsresources-17.asp** เป็นแหล่งรวมเว็บไซต์ที่เกี่ยวข้องกับ .NET, ASP .NET และอื่นๆ อีกเยอะ

- **http://www.topxml.com/dotnet/** รวบรวมบทความที่เกี่ยวกับ .NET และเทคโนโลยีที่เกี่ยวข้องโดยเฉพาะ ASP .NET

- **http://dotnet.oreilly.com/** รวบรวมบทความเกี่ยวกับ .NET แบบลึกซึ้งในสไตล์ของสำนักพิมพ์ OReilly โดยครอบคลุมเกี่ยวกับภาษา C# และ VB.NET เป็นส่วนใหญ่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **http://www.c-sharpcorner.com/** เป็นแหล่งรวบรวมความรู้เกี่ยวกับ .NET ที่ดีมาก นอกจากจะมีบทความเกี่ยวกับภาษา C# ในเชิงลึกแล้วยังมีบทความเกี่ยวกับ Asp.NET, VB.NET, SOAP, XML ด้วย รวมทั้งส่วนที่พิเศษก็คือภาษาอื่นๆ อย่าง COBOL.NET, FoxPro.NET เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] สราวุธ อ้อยศรีสกุล, ถอดรหัส .NET + Web Services, H.N. Group, 2544

[2] จักรพันธุ์ โพธิวรรณา, อัมรินทร์ เพ็ชรกุล, Microsoft Visual Studion .net, บริษัท ซัคเซส มีเดีย จำกัด

[3] บัญชา ปะสีละเตสัง, การเขียนโปรแกรม ASP .NET ด้วย VB.NET และ C#, บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน), 2546

[4] ทวีชัย หงษ์สุมาลย์, สงวนชัย สุวรรณชีวะศิริ, อินไซท์ ASP.NET (ฉบับสมบูรณ์), บริษัท โปรวิชั่น จำกัด, 2546

[5] สุรสิทธิ์ คิวประสพศักดิ์, นันทนี แขวงโสภา, อินไซท์ Visual Basic .NET (ฉบับสมบูรณ์), บริษัท โปรวิชั่น จำกัด, 2546

[6] Rikk Carey, Gavin Bell, Chris Marrin, ISO/IEC 14772-1 Virtual Reality Modeling Language (VRML97), 1997

[7] Daniel K. Schneider, Sylevere Martin-Michiellot, VRML Primer and Tutorial, March 18, 1998

[8] Harvey M. Deitel, Paul J. Deitel, Tem R. Nieto, Visual Basic. NET How to Program Second Edition, Deitel, 2002

[9] Harvey M. Deitel, Visual Basic .NET For Experienced Programmers, Deitel, 2002

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้