

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาซอฟต์แวร์เชิงคอมโพเนนต์
Component Software Development



นายนิลาต วนาพรรณ
นายนิวัฒน์ ลิ้มวานิชรัตน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2542

เลขหมู่.....
เลขทะเบียน 37029
วัน, เดือน, ปี ๖.๑.๒๕๔๓

สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาซอฟต์แวร์เชิงคอมโพเนนต์
Component Software Development



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท ปีการศึกษา 2542

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาซอฟต์แวร์เชิงคอมโพเนนต์

(Component Software Development)

ผู้จัดทำ

1. นายนิมาถ วนาพรรณี รหัสประจำตัว 39014263

2. นายนิวัฒน์ สีมวานิชรัตน์ รหัสประจำตัว 39014268



(ดร.วรวัฒน์ สีมโกกา)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาซอฟต์แวร์เชิงคอมพิวเตอร์

นายนิภาต วนาพรรณ รหัส 39014263

นายนิวัฒน์ ลีวานิชรัตน์ รหัส 39014268

ดร.วรวัฒน์ ลีมโกคา อาจารย์ที่ปรึกษา

ปีการศึกษา 2542

บทคัดย่อ

การพัฒนาซอฟต์แวร์เพื่อใช้งานทั้งในรูปแบบซอฟต์แวร์สำเร็จรูปหรือซอฟต์แวร์ที่จัดทำขึ้นเพื่อการใช้งานเฉพาะองค์กรมักเกิดปัญหาในระหว่างการพัฒนา การพัฒนาซอฟต์แวร์เชิงคอมพิวเตอร์เป็นหลักการในการพัฒนาซอฟต์แวร์ที่ทำให้การพัฒนามีความชัดเจนมากขึ้น ปัญหาต่าง ๆ ที่เกิดขึ้นในการพัฒนาซอฟต์แวร์ลดน้อยลงทำให้ซอฟต์แวร์ที่ได้มีคุณภาพสูงขึ้นและมีข้อผิดพลาดน้อยลง

งานวิจัยเล่มนี้แสดงถึงหลักการพัฒนาซอฟต์แวร์แบบคอมพิวเตอร์โดยอ้างอิงเทคโนโลยี COM (Component Object Model) เป็นพื้นฐาน ซึ่งเทคโนโลยีนี้ถูกนำไปพัฒนาต่อเพื่อการใช้งานมากมาย เช่น ActiveX ซึ่งเป็นเทคโนโลยีที่ใช้ในซอฟต์แวร์อินเทอร์เน็ต โดยการนำเทคโนโลยีที่มีพื้นฐานมาจาก COM ไปใช้ในการพัฒนาซอฟต์แวร์ในแขนงต่าง ๆ นั้นจะทำให้ซอฟต์แวร์ที่ได้รับประโยชน์จากเทคโนโลยีด้วย เช่น การที่ซอฟต์แวร์ที่ได้สามารถถูกนำกลับมาใช้ใหม่โดยไม่ขึ้นกับเครื่องมือที่พัฒนา เป็นต้น นอกจากนี้ยังแสดงวิธีการสร้างซอฟต์แวร์โดยใช้เทคโนโลยีที่อ้างอิงเทคโนโลยี COM เป็นพื้นฐาน เช่น ActiveX Control, ActiveX Document, ActiveX Server และ ActiveX Scripting เมื่อนำเทคโนโลยีเหล่านี้มาใช้ร่วมกันสามารถพัฒนาซอฟต์แวร์ที่มีความซับซ้อนได้

ข้อมูลสำคัญจำเป็นต้องมีการเข้ารหัสข้อมูลก่อนการส่ง ในงานวิจัยเล่มนี้ได้สรุปรายละเอียดของการเข้ารหัสในแบบต่าง ๆ เพื่อใช้ในการเลือกซอฟต์แวร์ในการใช้งานด้านความปลอดภัยได้อย่างเหมาะสม

ข้อมูลมัลติมีเดียเป็นข้อมูลที่มีความนิยมในการใช้งานเป็นอย่างมากเนื่องจากเป็นข้อมูลที่แสดงผลในรูปแบบที่ข้อมูลในรูปแบบเดิม เช่น เท็กซ์และรูปภาพ ไม่สามารถทำได้ แต่ข้อมูลแบบนี้มีปัญหาในเรื่องขนาดของข้อมูล ดังนั้นการส่งข้อมูลผ่านระบบเครือข่ายจำเป็นต้องใช้เทคโนโลยีสตรีมมิ่งซึ่งมีการแบ่งส่วนข้อมูลก่อนการส่งและให้ผู้รับสามารถดูข้อมูลได้ด้วยในขณะที่ทำการดึงข้อมูลอยู่

เมื่อนำเทคโนโลยีต่าง ๆ มารวมกันจะทำให้สามารถทำระบบที่มีความซับซ้อนได้ ในงานวิจัยนี้ได้ทำระบบมัลติมีเดียพาณิชยอิเล็กทรอนิกส์ที่มีความยืดหยุ่นในการใช้งานจริง

Component Software Development

Mr. Ninat Wanapan

Mr. Niwat Limvanitcharat

Dr. Worawat Limpoka Advisor

Abstract

Software developments that make software packages and bespoke software always see the problems in Development process. Component Software Development is a method of software development that makes the development clear. The problems in software development decrease so the software products have a good quality and a little mistake.

This research mainly presents the Component Software Development that based on COM (Component Object Model) technology. This technology is used to develop other technologies such as ActiveX that is technology used in Internet software. To use Com-based technology to develop software will give products good benefits from technology such as reused software that is not dependent on development tools etc. Furthermore in this research also presents a method to build Com-based software such as ActiveX Control, ActiveX Document, ActiveX Server and ActiveX Scripting. We can use all of these technologies to develop complex software.

Important information must to encrypt before sending. This research summarizes the details of variety encryption that is used to select software to be appropriate for security work.

Multimedia information is favor to use very much because this information can output in the style that old style information, such as text and picture, can't output. But this type of information has a size problem so to send information through network that you need to use streaming technology.

You can use variety of technology to build complex system. In this research, we have build multimedia electronic commerce system that has flexibility for the real usage.

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลาย ๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือ อาจารย์ วรวัฒน์ ลิ้มโกคา อาจารย์ที่ปรึกษาปริญญาานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำและช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

ขอบคุณเพื่อนทุกคน ที่ห้อง ปฏิบัติการฐานข้อมูลและออบเจ็ก โอเรียนเต็ลที่ร่วมให้ความช่วยเหลือในการทำงาน เพื่อนๆ ห้อง 4 ที่คอยให้กำลังใจ

และต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจเอาใจใส่เสมอมา ในทุก ๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VIII
สารบัญภาพ	IX
บทที่ 1 บทนำ	1
1.1 ความสำคัญและความเป็นมา	1
1.2 วัตถุประสงค์	3
1.3 ขอบเขตของงานวิจัย	3
1.4 วิธีการดำเนินงาน	3
บทที่ 2 การวิเคราะห์ ออกแบบและการพัฒนาระบบเชิงวัตถุ	5
2.1 ความหมายของ Object-Oriented Programming	5
2.2 คุณสมบัติของ Object-Oriented Programming	5
2.2.1 แอบสแตร็คชัน (Abstraction)	5
2.2.2 เอ็นแคปซูลชัน (Encapsulation)	5
2.2.3 การสืบทอด (Inheritance)	5
2.2.4 โพลีมอร์ฟิซึม (Polymorphism)	6
2.3 ออบเจกต์	6
2.4 ข่าวสาร	7
2.5 คลาสและอินสแตนซ์	7
2.6 ตัวอย่างการออกแบบออบเจกต์	8
2.7 UML (Unification Modeling Language)	10
2.7.1 ความรู้เบื้องต้น	10
2.7.2 การวิเคราะห์ระบบ	11
2.7.3 การออกแบบ	21
บทที่ 3 เทคโนโลยี COM และ DCOM และการพัฒนา	22
3.1 COM(Component Object Model)	22
3.1.1 คำอธิบายวัตถุ COM	22
3.1.1.1 อินเตอร์เฟส	22
3.1.1.2 คลาส COM	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1.3 เซิร์ฟเวอร์สำหรับวัตถุ COM	26
3.1.2 การสร้างวัตถุ COM	27
3.1.2.1 ไบบริารี COM	27
3.1.2.2 การสร้างวัตถุ COM หนึ่งวัตถุ	27
3.1.2.3 การสร้างวัตถุ COM หลายวัตถุ	28
3.1.3 การนำกลับมาใช้ใหม่ในวัตถุ COM	29
3.1.3.1 คอนเทนเมนต์	29
3.1.3.2 แอ็กกรีเกรชัน	30
3.1.4 ประโยชน์ของวัตถุ COM	30
3.1.4.1 การแก้ปัญหาในด้านเวอร์ชันของซอฟต์แวร์	30
3.2 กรณีศึกษาการพัฒนาซอฟต์แวร์ COM และ DCOM ใน Visual Basic และ Delphi	31
3.2.1 การพัฒนาซอฟต์แวร์ COM	31
3.2.2 การพัฒนาซอฟต์แวร์ DCOM	35
3.2.2.1 การพัฒนา โดย MTS (Microsoft Transaction Server)	35
บทที่ 4 เทคโนโลยี ActiveX	36
4.1 การทำงานแบบ Drag and Drop	36
4.1.1 OLE Drag and Drop	37
4.2 ActiveX Control	39
4.2.1 รูปแบบการสร้าง ActiveX Control	39
4.2.2 การทดสอบ ActiveX Control	45
4.2.3 การนำ ActiveX Control ไปใช้	47
4.2.4 การประยุกต์ใช้ ActiveX Control	48
4.3 ActiveX Document	49
4.3.1 ชื่อดีของ ActiveX Document	50
4.3.2 ประเภทของ ActiveX Document	50
4.3.3 ขั้นตอนการสร้าง ActiveX Document	50
4.3.4 ชื่อเปรียบเทียบการนำ ActiveX Control และ Active Document ไปใช้งานบน เว็บเพจ	52
4.3.5 ชื่อดีและชื่อเสียของการนำ ActiveX Control และ Active Document ไปใช้งาน บนเว็บเพจ	52
4.4 ActiveX Server	53
4.4.1 ตัวอย่างการสร้าง ActiveX Server	54
4.4.2 ผลการทดสอบ	56
4.5 ActiveX Scripting	57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.1 ส่วนประกอบของเว็บเพจที่ใช้งานกับ VBScript	57
4.5.2 ความแตกต่างระหว่าง Visual Basic กับ VBScript	57
บทที่ 5 IIS Application	61
5.1 ความหมายของ Active Server Pages (ASP)	61
5.2 กระบวนการทำงานของ Active Server Pages	61
5.3 Built-in Object ของ Active Server Pages	62
5.4 IIS Application	64
5.5 ตัวอย่างการสร้างแอปพลิเคชัน IIS	67
5.6 การทดสอบแอปพลิเคชัน IIS	70
บทที่ 6 ระบบความปลอดภัยและ Certificate Server	71
6.1 เทคโนโลยีการเข้ารหัสข้อมูลในอินเทอร์เน็ต	71
6.1.1 เทคนิคการเข้ารหัสแบบ Secret Key	71
6.1.2 เทคนิคการเข้ารหัสแบบ Public Key	71
6.1.3 เทคนิคการเข้ารหัสแบบผสม	72
6.2 คุณสมบัติพื้นฐานของระบบรักษาความปลอดภัยข้อมูล	72
6.2.1 Authentication	73
6.2.2 Integrity	73
6.2.3 Non-Repudiation	74
6.2.4 Confidentiality	74
6.3 RSA มาตรฐานในการเข้ารหัสข้อมูลในอินเทอร์เน็ต	74
6.4 มาตรฐานความปลอดภัยในอินเทอร์เน็ต SSL	75
6.4.1 SSL(Secure Socket Layer)	75
6.4.2 SSL กับการซื้อขายสินค้าผ่านระบบอินเทอร์เน็ต	76
6.4.3 SSL กับการใช้ใบรับรองดิจิทัล	77
6.5 มาตรฐานความปลอดภัยในอินเทอร์เน็ต SET	77
6.5.1 องค์ประกอบ SET	78
6.5.2 รูปแบบการซื้อขายด้วย SET	78
6.6 Certificate Server	79
บทที่ 7 ระบบมัลติมีเดียแบบสตรีมและ Windows Media Server	80
7.1 คำอธิบายเบื้องต้นเกี่ยวกับ Advance Streaming Format(ASF)	80
7.2 คำอธิบายเบื้องต้นเกี่ยวกับ Windows Media Server	80
7.3 การส่งข้อมูลสตรีมแบบ Unicast และ Multicast	80
7.3.1 Unicast	81
7.3.2 Multicast	82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.4 Windows Media Services Protocols	84
บทที่ 8 การวิเคราะห์และออกแบบระบบพาณิชย์อิเล็กทรอนิกส์เชิงคอมพิวเตอร์	85
8.1 การวิเคราะห์และออกแบบระบบในส่วนคอมพิวเตอร์การทำงานส่วนหน้า	85
8.1.1 คำอธิบายปัญหา	85
8.1.2 สร้าง Use Case	86
8.1.3 สร้างชินาิโอ	86
8.1.4 กำหนดโครงสร้างออบเจกต์	90
8.1.5 กำหนดโครงสร้างคลาส	90
8.1.6 รายละเอียดของคลาสในระบบ	92
8.2 ฟังก์ชันการทำงานของแต่ละเว็บคลาสในระบบ	96
บทที่ 9 ผลการทดสอบระบบ	98
9.1 การทดสอบในส่วนต่าง ๆ ของระบบ	98
9.2 ผลการทดสอบในส่วนต่าง ๆ ของระบบ	118
9.3 ปัญหาข้อจำกัดในการทำงานของระบบ	119
9.4 ข้อจำกัดในการทำงานของระบบ	120
บทที่ 10 บทวิจารณ์และสรุป	121
10.1 สรุปผลการดำเนินงาน	121
10.2 แนวทางการพัฒนาต่อ	122
ภาคผนวก ก การเขียนโปรแกรมเชิงออบเจกต์โอเรียนเท็ดใน Visual Basic 6	123
ภาคผนวก ข เอกสารอ้างอิง SAPI	140
ภาคผนวก ค การติดตั้ง Microsoft Certificate server ใช้ร่วมกับ IIS	157
ภาคผนวก ง ปัญหาที่พบและวิธีแก้	165
ภาคผนวก จ การติดตั้ง คอมพิวเตอร์ฝั่งเซิร์ฟเวอร์ และการเชื่อมต่อ MTS (Microsoft Transaction server) เพื่อทำระบบ Distributed Object	168

ตารางที่	สารบัญตาราง	หน้าที่
2-1	แสดงความหมายของพารามิเตอร์ต่างๆ ของทรานซิสต์	20
4-1	แสดงอีเวนต์ที่เกิดขึ้นของ Source และ Target Control	37
4-2	แสดงข้อแตกต่างระหว่าง VB กับ Visual Basic	57
9-1	แสดงผลการทดสอบในส่วนต่างๆ ของระบบ	119



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่	หน้าที่
2-1 การสืบทอดจากคลาสต้นแบบไปเป็นคลาสสืบทอด	6
2-2 การจำลองส่วนประกอบของออบเจกต์	7
2-3 การส่งข่าวสารระหว่างออบเจกต์	7
2-4 รายละเอียดของคลาส	8
2-5 ไดอะแกรมของตัวอย่างการออกแบบ	8
2-6 การส่งเมสเสจที่มีพารามิเตอร์	9
2-7 คลาสไดอะแกรมของรถยนต์	9
2-8 ลำดับชั้นของคลาส	10
2-9 การเปรียบเทียบระบบแบบ Onions Skin	12
2-10 Use Case Diagram	13
2-11 Sequence Diagram	14
2-12 Collaboration Diagram	15
2-13 ขั้นตอนการวิเคราะห์แบบออบเจกต์โอเรียนเท็ด	15
2-14 ความสัมพันธ์ระหว่างคลาสในแบบต่าง ๆ	18
2-15 สเตทไดอะแกรมของตัวจับเวลา	19
2-16 สเตทไดอะแกรมสำหรับซูเปอร์สเตทและสับสเตท	20
3-1 วัตถุ COM, ไคลเอนต์ และอินเทอร์เน็ตเฟส	22
3-2 การนำ Type Library ไปใช้ในเครื่องมือพัฒนาต่าง ๆ	23
3-3 โครงสร้างของอินเทอร์เน็ตเฟสในออบเจกต์	24
3-4 การใช้งาน QueryInterface	25
3-5 การทำงานของ In-process เซิร์ฟเวอร์	26
3-6 การทำงานของ Local เซิร์ฟเวอร์	26
3-7 การทำงานของ Remote เซิร์ฟเวอร์	27
3-8 การสร้างออบเจกต์หนึ่งออบเจกต์	28
3-9 การสร้างออบเจกต์หลาย ๆ ตัวจากคลาสเดียวกัน	29
3-10 การนำกลับมาใช้ใหม่ในลักษณะ Containment	29
3-11 การนำกลับมาใช้ใหม่แบบ Aggregation	30
3-12 การสร้าง COM Object	31
3-13 Type Library ที่เรากำหนดค่าที่ต้องการเรียบริ้อย	32
3-14 การเรียกใช้ Type Library	32
3-15 โปรแกรมไคลเอนต์ที่ใช้ในการทดสอบที่เขียนด้วย Delphi	33
3-16 การอ้างใช้คอมโพเนนต์ใน Visual Basic	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3-17 การนำเอา Type Library มาใช้ช่วยในการเขียนโปรแกรมฝั่งไคลเอนต์	34
3-18 โปรแกรมไคลเอนต์ที่ใช้ในการทดสอบที่เขียนด้วย Visual Basic	34
4-1 การทำงานแบบ Drag and Drop	36
4-2 Default Mouse Icon สำหรับ OLE Drag and Drop	38
4-3 ตัวอย่างการประยุกต์ใช้ Automatic OLE Drag and Drop	38
4-4 หน้าจอเริ่มต้นการสร้าง ActiveX Control	40
4-5 User Control ใน Project ActiveX Control	40
4-6 พร็อพเพอร์ตี้ BackStyle และ ToolboxBitmap	41
4-7 หน้าตาของ ActiveX Control ที่จะสร้าง	41
4-8 การเพิ่มพร็อพเพอร์ตี้, เมททอดและอีเวนต์ให้กับ ActiveX Control	42
4-9 การโหลด ActiveX Control Interface Wizard	44
4-10 การโหลด Property Page Wizard	44
4-11 การใช้ ActiveX Control ที่สร้างขึ้นใหม่	46
4-12 การตรวจสอบพร็อพเพอร์ตี้ต่างๆ และทดลองเรียกใช้ Property Page	46
4-13 การทดลองเขียนโค้ดใช้งาน ActiveX Control	47
4-14 การ Compile ActiveX Control เป็นไฟล์ .OCX	47
4-15 การประยุกต์ใช้งาน ActiveX Control ในบราวเซอร์	48
4-16 การโหลด Package and Deployment Wizard	49
4-17 การเลือก project เป็น ActiveX Document EXE	51
4-18 การแกะคอนโทรลต่างๆ ลงบน UserDocument	51
4-19 การรัน ActiveX Document EXE ทั้งใน VB และ Microsoft Office Binder	52
4-20 การทำงานของ ActiveX Server โดยการอ้าง Dispatch Interface	53
4-21 การกำหนดค่าให้กับ ActiveX Server	54
4-22 Type Library Editor	54
4-23 แสดงการทดสอบประสิทธิภาพวิธีการเรียกเมททอดในแบบต่างๆ	55
4-24 ข้อมูลของ ActiveX Server ในรีจิสทรีของระบบ	55
4-25 การอ้างถึง ActiveX Server ในโปรแกรม	55
4-26 การทำงานของโปรแกรมเพื่อควบคุม ActiveX Server	56
4-27 การทำงานของ ActiveX Scripting	57
4-28 การประยุกต์ใช้งาน web page ที่มี VBScript	58
4-29 การประยุกต์ใช้งาน web page ที่มี ActiveX Scripting	59
5-1 แอปพลิเคชัน IIS กับ Default Webclass	64
5-2 สถาปัตยกรรมของแอปพลิเคชัน IIS	64
5-3 ความสัมพันธ์ระหว่าง ASP และเว็บคลาส	66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5-4 หน้าจอเริ่มต้นการสร้าง IIS Application	68
5-5 Webclass Designer	68
5-6 ทำการเพิ่มธีมเพลตเข้าไปในโปรเจ็ค	68
5-7 อีเว้นต์ของฟอร์มในธีมเพลต	69
5-8 โค้ดควบคุมการทำงานส่วนวีจิสเตอร์ของลูกค้า	69
5-9 การสร้าง DLL ของ IIS Application	70
5-10 การทดสอบแอปพลิเคชัน IIS ที่ทำการสร้าง	70
6-1 เลขอร์ของ SSL เปรียบเทียบกับเลขอร์ของโมเดล OSI	76
6-2 การทำงานในแต่ละเลขอร์ของ SSL	76
7-1 On-demand Unicast	81
7-2 Broadcast Unicast	82
7-3 Broadcast Multicast	83
7-4 การใช้โพรโตคอลต่าง ๆ ในการสื่อสารระหว่างคอมพิวเตอร์	83
8-1 Use Case ของระบบ	86
8-2 ซินาโรไอการซื้อของ (1)	87
8-3 ซินาโรไอการซื้อของ (2)	88
8-4 ซินาโรไอการตั้งร้าน	89
8-5 ออบเจ็คต์ไดอะแกรมในระดับการวิเคราะห์เบื้องต้น	90
8-6 คลาสไดอะแกรมของระบบ	91
9-1 แสดงหน้าจอของร้านค้าที่พัฒนาขึ้น	98
9-2 แสดงหน้าจอเริ่มการลงทะเบียนโดยมีการเข้ารหัสข้อมูลในการสื่อสาร	99
9-3 แสดงหน้าจอเพื่อให้กรอกข้อมูลการลงทะเบียนสำหรับลูกค้า	99
9-4 แสดงการทดสอบการลงทะเบียนในฐานะเจ้าของร้าน	100
9-5 แสดงการล็อกอินเข้าใช้งานระบบในรูปแบบต่าง ๆ	101
9-6 แสดงการล็อกอินเข้าใช้ระบบเพื่อปรับเปลี่ยนข้อมูลและดูรายละเอียด Shopping Bag	101
9-7 แสดงส่วนของระบบในการติดตามว่าใบสั่งซื้อมีสถานะเป็นอย่างไร	102
9-8 แสดงการทดสอบการเข้าไปร้านแต่ละร้านที่เข้ามาเปิดกิจการได้	102
9-9 แสดงการเข้าไปดูรายละเอียดของสินค้าแต่ละชนิด	103
9-10 แสดงการเรียกดูข้อมูลของสินค้าในส่วนของมัลติมีเดียสตรีมมิ่ง	103
9-11 แสดงการเลือกสินค้าไปไว้ใน Shopping Bag โดยที่ไม่ได้ล็อกอินต้องล็อกอินก่อน	104
9-12 แสดงผลลัพธ์การใช้ฟังก์ชันการค้นหาสินค้าที่ต้องการในร้าน	104
9-13 แสดงการเข้าใช้งานระบบเพื่อจัดการร้านค้าของเจ้าของร้าน	105
9-14 แสดงการจัดการในส่วนของประเภทสินค้าในร้านโดยสามารถอัปเดตข้อมูลได้	105
9-15 แสดงการทดสอบการเพิ่มประเภทของสินค้าในร้าน	106

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9-16 แสดงการลบประเภทของสินค้าที่ไม่ต้องการในร้าน	106
9-17 แสดงการจัดการจัดการในส่วนของสินค้าโดยเพิ่มสินค้าในประเภทที่ต้องการ	107
9-18 แสดงการลบสินค้าในประเภทที่ต้องการ	107
9-19 แสดงการทดสอบโดยอัปเดตข้อมูลสินค้าเดิมในประเภทที่ต้องการ	108
9-20 แสดงการทดสอบในส่วนของการใช้ Shopping Bag	108
9-21 แสดงการทดสอบในส่วนการทำใบสั่งซื้อ	109
9-22 แสดงการทดสอบขั้นตอนการประมวลผลการสั่งซื้อ	109
9-23 แสดงหน้าจอเมื่อการประมวลผลการสั่งซื้อเสร็จเรียบร้อยแล้ว	110
9-24 แสดงการใช้ ActiveX Document ในส่วนของการเพิ่ม ผู้ใช้ระบบ	110
9-25 แสดงการใช้ ActiveX Document ในส่วนของการเพิ่ม ประเภทของสินค้า	111
9-26 แสดงการใช้ ActiveX Document ในส่วนของการเพิ่ม ผู้ใช้ระบบ	112
9-27 แสดงการใช้ ActiveX Document ในส่วนของการลบ ผู้ใช้ระบบ	112
9-28 แสดงการใช้ ActiveX Document ในส่วนของการลบร้านออกจากระบบ	113
9-29 แสดงการใช้ ActiveX Document ในส่วนของการลบประเภทสินค้า	113
9-30 แสดงการใช้ ActiveX Document ในส่วนของการลบสินค้า	114
9-31 แสดงการใช้ ActiveX Document ในส่วนของการดึงข้อมูลลูกค้าขึ้นมาดู	114
9-32 แสดงการใช้ ActiveX Document ในส่วนของการดึงข้อมูลร้าน	115
9-33 แสดงการใช้ ActiveX Document ในส่วนของการดึงข้อมูลประเภทของสินค้า	115
9-34 แสดงการใช้ ActiveX Document ในส่วนของการดึงข้อมูลสินค้า	116
9-35 แสดงการใช้ ActiveX Document ในส่วนของการอัปเดตข้อมูลลูกค้า	116
9-36 แสดงการใช้ ActiveX Document ในส่วนของการอัปเดตข้อมูลร้าน	117
9-37 แสดงการใช้ ActiveX Document ในส่วนของการอัปเดตข้อมูลประเภทสินค้าในร้าน	117
9-38 แสดงการใช้ ActiveX Document ในส่วนของการอัปเดตข้อมูลสินค้าในร้าน	118
9-39 แสดงการตั้งค่า Time Out ให้มีค่ามากขึ้น	120

บทที่ 1

บทนำ

1.1 ความสำคัญและความเป็นมา

ในปัจจุบัน ความเป็นอยู่ของมนุษยชาติได้พึ่งพิงคอมพิวเตอร์ในการดำรงชีวิตเป็นอย่างมากซึ่ง การที่คอมพิวเตอร์จะทำงาน เพื่อสนองตอบต่อความต้องการของเราได้นั้นจะต้องมีองค์ประกอบที่สำคัญ คือ ฮาร์ดแวร์และซอฟต์แวร์ ซึ่งฮาร์ดแวร์นั้นมีการพัฒนาอย่างต่อเนื่องและเป็นระบบ ปัญหาที่จะเกิดจากความผิดพลาดในตัวเองที่ไม่ได้เกิดจากการเสื่อมสภาพนั้นน้อยมาก จึงสามารถออกแบบระบบที่มีขนาดใหญ่และความซับซ้อนมากได้ไม่ยากนัก แต่ในซอฟต์แวร์ การพัฒนายังเป็นไปอย่างไม่ค่อยมีระบบ ไม่มีการออกแบบหรือพัฒนาที่เป็นขั้นตอน จึงเกิดปัญหาในการพัฒนามากมาย เช่น การใช้เวลาและเงินทุนมากกว่าที่กำหนดรวมถึงซอฟต์แวร์ที่ได้ไม่มีคุณภาพหรือไม่ตรงกับความต้องการของผู้ใช้งาน

จากสาเหตุดังกล่าวได้มีผู้เสนอแนวทางตลอดจนวิธีการต่าง ๆ เพื่อให้การพัฒนาซอฟต์แวร์มีปัญหาน้อยลง เช่น การพัฒนาระบบแบบโครงสร้าง การพัฒนาระบบเชิงออบเจกต์โอเรียนเท็ดแต่ก็ยังมีปัญหาต่าง ๆ ตามมามากมาย ในกรณีของการพัฒนาระบบแบบโครงสร้างจะประสบปัญหาด้านโมเดลของระบบที่ไม่เชื่อมต่อกันตลอดวัฏจักรการพัฒนา ในส่วนกรณีการพัฒนาระบบแบบออบเจกต์โอเรียนเท็ดจะมีปัญหาที่เห็นได้ชัด คือ การที่ออบเจกต์มีการผูกติดกันมากเกินไป (Tightly Coupling) เมื่อมีการแก้ไขอาจจะทำให้กระทบกระเทือนออบเจกต์ลูกหลานได้และการพัฒนาเป็นไลบรารีแบบไบนารีก็เกิดปัญหาด้านเวอร์ชันของซอฟต์แวร์ (Versioning) กล่าวคือ การเปลี่ยนแปลงไลบรารีเพื่อเพิ่มเติมฟังก์ชันใหม่จะทำให้ซอฟต์แวร์เดิมอาจมีปัญหาในการทำงาน เป็นต้น

COM (Component Object Model) เป็นมาตรฐานไบนารีในการทำงานร่วมกันระหว่างออบเจกต์ที่ไม่ขึ้นกับภาษาที่ใช้พัฒนา โดยจะมีการติดต่อกันผ่านอินเตอร์เฟซ (Interface) เท่านั้น อีกทั้ง COM ยังเพิ่มความสามารถในด้านการเรียกใช้แบบไม่ขึ้นกับรูปแบบ (Location Transparency) ทำให้ผู้พัฒนาสามารถเรียกใช้ออบเจกต์ได้ตั้งแต่ในรูปแบบพื้นฐาน คือ อยู่เครื่องเดียวกัน จนถึงในรูปแบบการเรียกใช้แบบข้ามเครื่อง จึงสามารถทำระบบแบบกระจาย (Distributed System) ซึ่งเป็นประโยชน์หลายประการ เช่น การทำ Load Balancing เพื่อเป็นการลดภาระของเครื่องเซิร์ฟเวอร์ ประเด็นที่สำคัญในออบเจกต์ COM คือ การนำกลับมาใช้ใหม่ (Reusability) ซึ่ง COM มีรูปแบบการนำกลับมาใช้ใหม่อยู่ 2 รูปแบบ คือ Aggregation และ Delegation ซึ่งทั้ง 2 รูปแบบจะช่วยแก้ปัญหาในการพัฒนาซอฟต์แวร์ต่าง ๆ ที่กล่าวมาแล้วได้

ActiveX เป็นโครงสร้างทางเทคโนโลยีที่รองรับแอปพลิเคชันแบบใหม่ทั้งในรูปแบบแอปพลิเคชันทางอินเทอร์เน็ตที่มีการยึดหยุ่นในการใช้งานเป็นอย่างมาก

ActiveX Control เป็นคอมโพเนนต์ที่ถูกสร้างและคอมไพล์แล้วพร้อมใช้งานได้ทันทีโดยไม่ต้องคอมไพล์ใหม่ นอกจากนี้ยังมีคุณสมบัติ การใช้ร่วมกับอินเทอร์เน็ต เป็นต้น โดยสามารถใช้ร่วมกันระหว่างเครื่องมือพัฒนาที่สนับสนุน ActiveX

ActiveX Document เป็นแอปพลิเคชันที่ถูกสร้างขึ้นมาเพื่อให้สามารถใช้งานภายใต้คอนเทนเนอร์ได้หลายแบบ ยกตัวอย่างเช่น Internet Explorer หรือแม้กระทั่ง Microsoft Office Binder ที่มากับ Microsoft Office โดยจุดประสงค์เพื่อเป็นการยืดหยุ่นในการใช้งานโดยสามารถพัฒนาโปรแกรมให้รันได้หลายแพลตฟอร์ม โดยแพลตฟอร์มที่สำคัญ คือ แพลตฟอร์มบนอินเทอร์เน็ตที่ทำให้สามารถทำงานจากที่ใดก็ได้ในโลก (Globalization)

ActiveX Server เป็นเทคโนโลยีในการพัฒนาซอฟต์แวร์ที่ทำให้ฟังก์ชันการทำงานสามารถถูกนำไปเรียกใช้ในการพัฒนาซอฟต์แวร์อื่นได้

ActiveX Scripting เป็นภาษาที่ใช้เพื่อเพิ่มความสามารถให้กับเว็บเพจ

IIS Application เป็นเทคโนโลยีในการพัฒนาเว็บเพจแบบอินเทอร์แอคทีฟ โดยมีการควบคุมและสั่งการจากเซิร์ฟเวอร์ (Server Side) นอกจากนี้ยังมีการพัฒนาเป็นเว็บโมดูลทำให้การออกแบบ การพัฒนา และการบำรุงรักษาสามารถทำได้ง่าย

จากที่เรากล่าวมา เทคโนโลยีต้องมีการสื่อสาร โดยส่งข้อมูลจากต้นทางไปยังปลายทางเสมอ การส่งข้อมูลนี้อาจจะเป็นข้อมูลที่สำคัญและไม่สำคัญ ในส่วนของข้อมูลสำคัญเราก็คงอยากให้ความปลอดภัยสูงสุด เช่น หมายเลขบัตรเครดิต เป็นต้น วิธีการสร้างความปลอดภัยเบื้องต้น ได้แก่ การเข้ารหัสข้อมูล (Encryption) ที่จะส่ง ซึ่งมีมาตรฐานในการส่งข้อมูลมากมาย เช่น SSL เป็นต้น การที่จะให้ระบบมีการเข้ารหัสข้อมูลสามารถทำได้หลายวิธี แต่ในงานวิจัยนี้ได้เลือกชุดเซิร์ฟเวอร์ของไมโครซอฟต์ คือ Certificate Server ซึ่งเป็นเซิร์ฟเวอร์ที่ทำหน้าที่เป็น Certificate Authority และยังสามารถเข้ารหัสข้อมูลที่ส่งในการสื่อสารได้ด้วย

ข้อมูลที่เราใช้ในการสื่อสารกันมีหลายรูปแบบตั้งแต่รูปแบบพื้นฐาน คือ เท็กซ์ สตริมมิ่งธรรมดาจนถึงข้อมูลที่เป็นมัลติมีเดีย ได้แก่ เสียงหรือวิดีโอ เป็นต้น ในการส่งข้อมูลเหล่านี้มีวิธีการส่งได้หลายรูปแบบตั้งแต่การส่งเป็นไฟล์ทั้งไฟล์ไป ซึ่งมีข้อเสีย คือ ต้องเสียเวลาในการเคลื่อนย้ายข้อมูลทั้งหมดก่อนจึงจะเรียกข้อมูลเหล่านั้นมาดูได้ อีกรูปแบบหนึ่ง คือ การทำสตรีมมิ่ง (Streaming) โดยข้อมูลจะมีการแบ่งส่วนข้อมูลเพื่อส่งไปเป็นส่วน ๆ ทางผู้รับสามารถเรียกดูข้อมูลได้โดยไม่ต้องรอให้เคลื่อนย้ายข้อมูลทั้งหมดเสร็จ โดยในงานวิจัยนี้ได้เลือก Windows Media Server services ซึ่งเป็นแอคตอนของ Microsoft ที่เพิ่มความสามารถในการทำมัลติมีเดียสตรีมมิ่งให้กับ Windows NT Server 4

จากเทคโนโลยีและความรู้ที่กล่าวมา งานวิจัยนี้ได้มีการประยุกต์ในการทำระบบมัลติมีเดียพาณิชย์อิเล็กทรอนิกส์ โดยอ้างอิงการพัฒนาซอฟต์แวร์เชิงคอมโพเนนต์ โครงสร้างเป็นสถาปัตยกรรมแบบเอ็น-tier ระบบมีการส่งผ่านข้อมูลที่สำคัญผ่านโพรโตคอล SSL 40 บิต โดยจะจัดทำในส่วนของระบบส่วนหน้า (Front-end System) ในส่วนของระบบส่วนหลัง (Back-end System) จะจัดทำโดยงานวิจัยเรื่องการพัฒนาไคลเอนต์เซิร์ฟเวอร์ชนิด 3-tier

ระบบที่พัฒนาขึ้น สามารถให้มีการเปิดร้านจากลูกค้าได้ กล่าวคือ อยู่ในลักษณะห้างสรรพสินค้า (Shopping Mall) โดยข้อมูลของสินค้าสามารถถูกเก็บในรูปแบบต่าง ๆ กันได้ เช่น เป็นเท็กซ์หรือมัลติมีเดีย โดยข้อมูลมัลติมีเดียจะมีการส่งผ่านในลักษณะสตรีม จากระบบที่พัฒนาขึ้นเพื่อเพิ่มความแปลกใหม่ในโลกธุรกิจบนอินเทอร์เน็ต คือ ไม่จำเป็นที่ผู้ต้องการขายของต้องทำการเปิดร้านอิเล็กทรอนิกส์ของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตนเองหรือการใช้เพียงเว็บบอร์ดในการซื้อขายเท่านั้น ซึ่งเป็นวิธีการที่ไม่มีประสิทธิภาพและประสิทธิผล แต่สามารถเลือกใช้บริการจากระบบที่เราพัฒนาขึ้นมาได้

1.2 วัตถุประสงค์ของงานวิจัย

1.2.1 เพื่อศึกษาการวิเคราะห์และออกแบบระบบแบบออบเจกต์โอเรียนเท็ด โดยอ้างอิงโมเดล UML

1.2.2 เพื่อศึกษาการพัฒนาซอฟต์แวร์เชิงคอมโพเนนต์โดยอ้างอิงเทคโนโลยี COM (Component Object Model) เพื่อแก้ไขข้อบกพร่องบางประการของการวิเคราะห์และออกแบบระบบแบบออบเจกต์แบบเดิม

1.2.3 เพื่อศึกษาการพัฒนาระบบแบบกระจายเชิงออบเจกต์ (Distributed Object System)

1.2.4 เพื่อศึกษาการพัฒนาแอปพลิเคชันโดยอ้างอิงจากเทคโนโลยี ActiveX

1.2.5 เพื่อศึกษาระบบการรักษาความปลอดภัยข้อมูลในรูปแบบต่าง ๆ

1.2.6 เพื่อศึกษาเทคโนโลยีการแปลงข้อความเสียง (Text-to-Speech)

1.2.7 เพื่อศึกษาเทคโนโลยีมัลติมีเดียสตรีมมิ่ง

1.3 ขอบเขตงานวิจัย

งานวิจัยนี้จะเป็นงานวิจัยที่ครอบคลุมหลักการวิเคราะห์และออกแบบระบบแบบออบเจกต์โอเรียนเท็ด การพัฒนาซอฟต์แวร์เชิงคอมโพเนนต์โดยอ้างอิงเทคโนโลยี COM ตลอดจนเทคโนโลยีอื่นที่เกี่ยวข้องเพื่อการพัฒนาระบบมัลติมีเดียพาณิชย์อิเล็กทรอนิกส์ที่มีความซับซ้อนและ มีความยืดหยุ่นในการใช้งานและบำรุงรักษา

เครื่องมือที่ใช้พัฒนาคือ Visual Basic 6.0 Enterprise Edition พัฒนาทั้งในส่วนคอมโพเนนต์ COM และ IIS Application บนแพลตฟอร์ม Windows NT Server 4.0 Service Pack 5 โดยมี Microsoft Transaction Server จัดการด้านทรานแซกชันและการกระจายออบเจกต์ Microsoft Certificate Server จัดการเรื่องระบบความปลอดภัย Window Media Server จัดการระบบมัลติมีเดียสตรีมมิ่ง SAPI (Sound Application Programming Interface) ใช้ในการพัฒนาระบบการแปลงข้อมูลเป็นเสียงและ ASPEmail ช่วยในการพัฒนาระบบตอบรับเมลล์อิเล็กทรอนิกส์

เมื่อเสร็จสิ้นโครงการนี้จะทำให้ได้ระบบมัลติมีเดียพาณิชย์อิเล็กทรอนิกส์ที่สามารถใช้งานได้จริง เพื่อให้บริการธุรกิจการค้าบนอินเทอร์เน็ต

1.4 วิธีการดำเนินงาน

โครงการวิจัยนี้จะเริ่มด้วยการศึกษาหลักการที่ว่าด้วยเรื่อง การวิเคราะห์และออกแบบระบบแบบออบเจกต์โอเรียนเท็ด การพัฒนาซอฟต์แวร์เชิงคอมโพเนนต์ เทคโนโลยี ActiveX ต่าง ๆ แอปพลิเคชัน IIS ระบบความปลอดภัยต่าง ๆ ระบบมัลติมีเดียสตรีมมิ่ง เทคโนโลยี SAPI จากเครื่องมือที่ได้กล่าวไป โดยเนื้อหาจะถูกสรุปอยู่ในบทที่ 2-7 หลังจากนั้นจะเป็นการนำหลักการที่ได้ศึกษามาใช้ในการพัฒนาระบบ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มัลติมีเดียพาณิชย์อิเล็กทรอนิกส์ โดยเริ่มตั้งแต่การออกแบบไปจนถึงการทดสอบระบบ โดยจะถูกสรุปเป็นเนื้อหาในบทที่ 8-9

ในขั้นตอนสุดท้ายจะเริ่มการพัฒนาระบบที่ได้ออกแบบไว้ ทั้งในส่วนของคอมพิวเตอร์และส่วนของแอปพลิเคชัน IIS เพื่อให้ได้มาซึ่งระบบที่ต้องการเพื่อเป็นตัวอย่างในการพัฒนาต่อไปหรือนำไปใช้จริง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

การวิเคราะห์ ออกแบบและการพัฒนาระบบเชิงวัตถุ (Object-Oriented System Analysis, Design and Development)

2.1 ความหมายของ Object-Oriented Programming (OOP)

โอโอพี หรือ OOP เป็นคำย่อของ Object-Oriented Programming ซึ่งหมายถึง วิธีการเขียนโปรแกรมแบบออบเจกต์ โดยหลักการสำคัญของโอโอพี คือ การสร้างออบเจกต์ขึ้นมาแทนระบบซึ่งสามารถประมวลผลออบเจกต์นั้นเพื่อให้ได้ผลลัพธ์ออกมา

การเขียนโปรแกรมแบบออบเจกต์ คือ วิธีการเขียนโปรแกรมคอมพิวเตอร์ที่ใช้แนวความคิดในการจำลองวัตถุในโลกแห่งความเป็นจริงด้วยออบเจกต์ซึ่งทำให้โครงสร้างของโปรแกรมเป็นระเบียบมากขึ้นและเอื้ออำนวยต่อการพัฒนาโปรแกรมในรุ่นต่อ ๆ ไป

2.2 คุณสมบัติของ Object-Oriented Programming

2.2.1 แอบสแตรักชัน (Abstraction)

แอบสแตรักชันเป็นความสามารถในการสร้างเอกลักษณ์ของออบเจกต์ ตัวอย่างเช่น ออบเจกต์รถจะมีคุณสมบัติต่าง ๆ เช่น ความเร็วสูงสุด ความจุและสี ซึ่งจะแตกต่างจากออบเจกต์คนที่มีคุณสมบัติต่าง ๆ เช่น ความสูงและน้ำหนัก เป็นต้น

2.2.2 เอ็นแคปซูลชัน (Encapsulation)

เอ็นแคปซูลชันเป็นการรวมข้อมูลและโพรซีเจอร์หรือฟังก์ชันเข้าด้วยกันเป็นออบเจกต์ โดยรายละเอียดเกี่ยวกับข้อมูล ส่วนประกอบ โครงสร้างและการทำงานภายในของออบเจกต์จะถูกซ่อนจากผู้ใช้ (Client) โดยผู้ใช้งานจะเห็นเพียงอินเตอร์เฟซของออบเจกต์ซึ่งการทำงานภายในของออบเจกต์จะซ่อนอยู่ภายใต้อินเตอร์เฟซ ตัวอย่างเช่น ในการพิมพ์งานจะมีคีย์บอร์ดเป็นอินเตอร์เฟซซึ่งใช้ในการสร้างตัวอักษรในโปรแกรมเวิร์ดโปรเซสเซอร์ ส่วนการทำงานภายในที่ทำการแปลงจากการกดคีย์ไปเป็นตัวอักษรบนหน้าจอจะถูกซ่อนอยู่ภายใต้อินเตอร์เฟซ

2.2.3 การสืบทอด (Inheritance)

การสืบทอดเป็นวิธีการในการสืบทอดจากคลาสต้นแบบ (Base class) ให้กับคลาสสืบทอด (Derived class) ที่จะถูกสร้างขึ้น โดยคลาสสืบทอดจะทำการสืบทอดทั้งคุณสมบัติ (Property) และคุณลักษณะ (Characteristic) ทุกอย่างจากคลาสต้นแบบ ในขณะที่เดียวกันคลาสสืบทอดก็สามารถเพิ่มคุณสมบัติและคุณลักษณะให้กับตัวเองได้ ตัวอย่างเช่น คลาสต้นแบบ คือ คลาสเครื่องเขียนและทำการสืบทอดคลาสไปเป็นคลาสสืบทอด คือ คลาสดินสอ คลาสปากกาถูกสิ้นและคลาปากกาหมึกซึม เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-1 การสืบทอดจากคลาสต้นแบบไปเป็นคลาสสืบทอด

2.2.4 โพลิมอร์ฟิซึม (Polymorphism)

โพลิมอร์ฟิซึมเป็นความสามารถของออบเจกต์ในการทำงานเมื่อได้รับคำสั่งเดียวกันจากโปรแกรม โดยออบเจกต์จะทำงานตามแบบของตัวเอง ตัวอย่างเช่น ออบเจกต์คนกับออบเจกต์สุนัขสามารถที่จะเดินได้แต่วิธีการเดินของทั้ง 2 ออบเจกต์จะแตกต่างกัน เป็นต้น การที่ออบเจกต์จะสามารถทำงานได้ในลักษณะนี้ คลาสต้นแบบจะต้องประกาศเมทอด (Method) ที่จะใช้เป็นเวอร์ชวลเมทอด

2.2.4.1 เมทอดแบบสแตติก (Static method)

เมทอดแบบสแตติกมีลักษณะการทำงานคล้ายกับการทำงานของโพรซีเจอร์และฟังก์ชันทั่วไปในภาษาปาสคาล กล่าวคือ เมื่อเมทอดเหล่านี้เรียกใช้เมทอดอื่นก็จะมี การเรียกใช้เมทอดเดิมนั้นเสมอ ที่เป็นเช่นนี้ก็เพราะได้มีการผูกติดกันตั้งแต่ตอนคอมไพล์โปรแกรม ซึ่งเรียกว่า เออร์ลีไบน์ดิง (Early binding) ผลของเออร์ลีไบน์ดิง ก็คือ เมทอดที่ทำการเรียกเมทอดอื่นจะรู้ว่าจะเรียกเมทอดนั้นจากที่ใด และตำแหน่งการเรียกเมทอดนั้นจะคงที่ไม่สามารถเปลี่ยนแปลงได้

2.2.4.2 เมทอดแบบเวอร์ชวล (Virtual method)

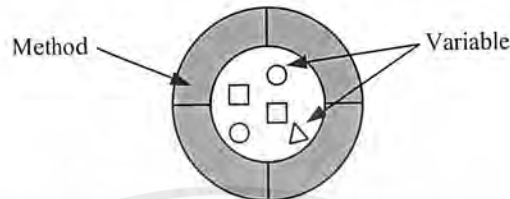
เมทอดแบบนี้จะไม่มีการผูกติดกับเมทอดอื่นตั้งแต่ตอนคอมไพล์โปรแกรม แต่จะมีการผูกติดกันตอนรันโปรแกรม ซึ่งเรียกว่า เลตไบน์ดิง (Late binding) ผลของเลตไบน์ดิง ก็คือ เมทอดที่ทำการเรียกเมทอดอื่นจะไม่รู้ว่าจะเรียกเมทอดนั้นจากที่ไหนแต่จะรู้ในตอนรันโปรแกรม

2.3 ออบเจกต์ (Object)

ออบเจกต์ คือ สิ่งที่เกิดจากการรวมกันของข้อมูลและวิธีการจัดการกับข้อมูลนั้น ออบเจกต์เป็นส่วนสำคัญในการเขียนโปรแกรมแบบออบเจกต์ โดยตัวอย่างของออบเจกต์ในโลกแห่งความจริง เช่น คน เป็นสิ่งมีชีวิตที่มีคุณสมบัติ เช่น ความสูงและน้ำหนัก เป็นต้นและมีพฤติกรรม เช่น สามารถเดินได้และสามารถวิ่งได้ เป็นต้น สำหรับการเขียนโปรแกรมแบบออบเจกต์นั้นเราสามารถที่จะแทนคุณสมบัติด้วยตัวแปร (Variable) และแทนพฤติกรรมด้วยเมทอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

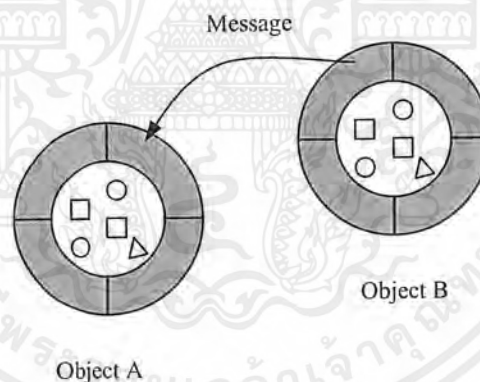
จากรูปที่ 2-2 จะเห็นได้ว่าตัวแปของออบเจ็กต์ถูกจัดให้อยู่ตรงกลางของวงกลมออบเจ็กต์และมีเมธอดล้อมรอบเพื่อทำหน้าที่ซ่อนส่วนกลางของออบเจ็กต์ไว้ ประโยชน์ของการเ็นแคปซูลเลชัน ก็คือ ถ้าหากต้องการเปลี่ยนแปลงข้อมูลภายในออบเจ็กต์ ก็สามารถทำได้โดยไม่ส่งผลกระทบต่อออบเจ็กต์อื่น ๆ และประโยชน์อีกข้อหนึ่ง ก็คือ สามารถจัดการกับข้อมูลภายในออบเจ็กต์ได้ว่าข้อมูลส่วนไหนต้องการที่จะเปิดเผยหรือไม่ ซึ่งสามารถกำหนดได้โดยประกาศคุณสมบัติให้กับออบเจ็กต์



รูปที่ 2-2 การจำลองส่วนประกอบของออบเจ็กต์

2.4 ข่าวสาร (Message)

การติดต่อสื่อสารกันระหว่างออบเจ็กต์สามารถทำได้โดยการรับส่งข่าวสาร ตัวอย่างเช่น เมื่อออบเจ็กต์ A ต้องการทำงานกับออบเจ็กต์ B ออบเจ็กต์ A ก็จะส่งข่าวสารไปบอกแก่ออบเจ็กต์ B ดังรูป 2-3

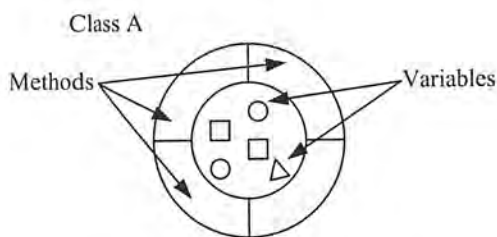


รูปที่ 2-3 การส่งข่าวสารระหว่างออบเจ็กต์

2.5 คลาสและอินสแตนซ์ (Class and Instance)

คลาส คือ ชนิดของออบเจ็กต์ซึ่งจะแสดงถึงคุณสมบัติและคุณลักษณะของออบเจ็กต์ คลาสจะถูกสร้างขึ้นโดยการเขียนรายละเอียดของคลาส ซึ่งคลาสหนึ่งสามารถที่จะเป็นต้นแบบให้แก่ออบเจ็กต์ได้หลายออบเจ็กต์ การที่เราสร้างออบเจ็กต์ขึ้นมา นั้น เราเรียกว่า อินสแตนซ์ (Instance) โดยเราต้องทำการสร้างอินสแตนซ์ของคลาสขึ้นมาก่อนที่จะมีการใช้ตัวแปรหรือเมธอดของคลาสนั้น ซึ่งจะเรียกอินสแตนซ์ว่าเป็นออบเจ็กต์เมื่อมีการเปลี่ยนแปลงข้อมูลหรือเรียกใช้เมธอดของอินสแตนซ์นั้นเกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-4 รายละเอียดของคลาส

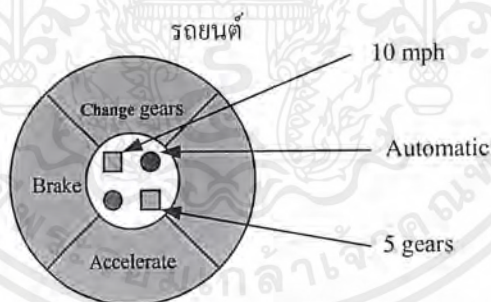
2.6 ตัวอย่างการออกแบบออบเจกต์

ในตัวอย่างนี้จะทำการออกแบบออบเจกต์รถยนต์ที่มีการกำหนดตัวแปรดังต่อไปนี้
สถานะของรถยนต์

1. รถยนต์วิ่งด้วยความเร็ว 10 เมตรต่อวินาที
2. พวงมาลัยรถยนต์เป็นแบบออโตเมติก
3. มีเกียร์ทั้งหมด 5 เกียร์

เมทอดของรถยนต์

1. การเบรก
2. การเร่งความเร็ว
3. การเปลี่ยนเกียร์



รูปที่ 2-5 ไลอะแกรมของตัวอย่างการออกแบบ

จากรูปที่ 2-5 ตัวแปรของออบเจกต์จะอยู่ตรงส่วนกลางของออบเจกต์ โดยเมทอดจะอยู่รอบ ๆ ตัวแปรของออบเจกต์และซ่อนตัวแปรของออบเจกต์จากออบเจกต์อื่น ๆ เช่น เมื่อเราต้องการที่จะเปลี่ยนเกียร์ของรถยนต์ เราก็ไม่จำเป็นต้องรู้ว่าระบบของเกียร์ทำงานอย่างไรบ้าง เพียงแต่เรารู้วิธีการที่จะเลื่อนคันโยกสำหรับเปลี่ยนเกียร์ก็พอแล้ว

เมสเสจของรถยนต์

ในการทำงานของออบเจกต์ ออบเจกต์จะต้องได้รับข้อมูลที่เพียงพอเพื่อที่ออบเจกต์จะรู้ว่าจะต้องทำอะไร จากตัวอย่างของการออกแบบ เมื่อต้องการเปลี่ยนเกียร์จะต้องทราบว่าเปลี่ยนเกียร์เป็นเกียร์อะไร โดยข้อมูลเหล่านี้จะถูกส่งไปกับเมสเสจเหมือนเป็นพารามิเตอร์

ส่วนประกอบของเมสเสจประกอบด้วย

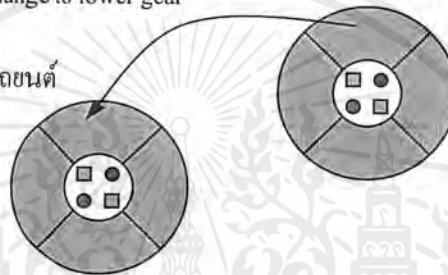
1. ออบเจกต์ที่จะส่งเมสเสจไปให้ (รถยนต์)
2. ชื่อของเมทอดที่จะทำ (เปลี่ยนเกียร์)
3. พารามิเตอร์ต่าง ๆ ที่เมทอดต้องใช้

A Message with Parameters

Change to lower gear

รถยนต์

คนขับ

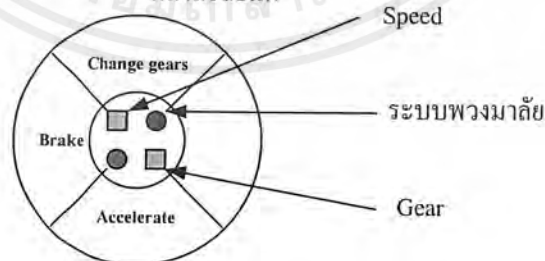


รูปที่ 2-6 การส่งเมสเสจที่มีพารามิเตอร์

คลาสของรถยนต์

เราต้องสร้างคลาสของรถยนต์โดยประกาศตัวแปรเพื่อใช้เก็บสถานะต่าง ๆ เช่น ความเร็วของรถยนต์ สถานะของเกียร์ปัจจุบันและระบบของพวงมาลัย เป็นต้น และสร้างเมทอดต่าง ๆ ที่ใช้ในการทำงานของออบเจกต์ เช่น การเบรก การเร่งความเร็วและการเปลี่ยนเกียร์ เป็นต้น

คลาสรถยนต์



รูปที่ 2-7 คลาสโคดแแกรมของรถยนต์

ค่าของตัวแปรจะถูกกำหนดโดยอินสแตนซ์ของคลาส ดังนั้นหลังจากที่ทำการสร้างคลาสของรถยนต์แล้วจะต้องทำการสร้างอินสแตนซ์ของคลาสด้วย ถ้าทำการสร้างอินสแตนซ์ของคลาสแล้วตัวแปรที่

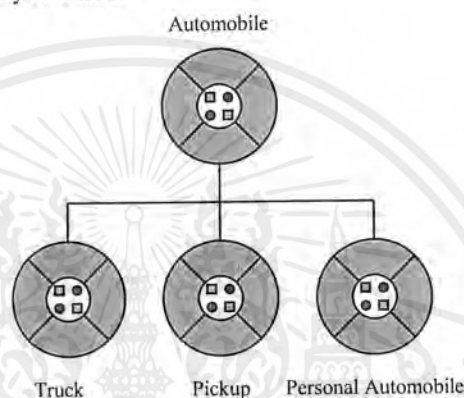
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประกาศไว้โดยคลาสจะต้องเก็บไว้ในหน่วยความจำ จากนั้นจะสามารถใช้เมทอดเพื่อกำหนดค่าให้ตัวแปรได้

Inheritance

ในการเขียนโปรแกรมแบบออบเจกต์จะมีการสืบทอดคลาสอย่างเป็นลำดับชั้นและกำหนดให้คลาสหนึ่งเป็นคลาสต้นแบบของคลาสอื่นได้ เช่น รถกระบะ รถบรรทุกและรถยนต์ส่วนบุคคลเป็นคลาสที่ทำการสืบทอดมาจากคลาสรถยนต์ โดยจะเรียกคลาสทั้ง 3 แบบเป็นสับคลาส (Sub class) และเรียกคลาสรถยนต์ว่าเป็นซูเปอร์คลาส (Super class)

Hierarchy of Classes



รูปที่ 2-8 ลำดับชั้นของคลาส

2.7 UML (Unification Modeling Language)

2.7.1 ความรู้เบื้องต้น

UML หรือ (Unification Modeling Language) เป็นภาษาที่ถูกสร้างเพื่อใช้แสดงโมเดลทางออบเจกต์โอเรียนเท็ด เกิดจากการพัฒนาของผู้นำทางเทคโนโลยีทางออบเจกต์ 3 ท่านคือ Grady Booch, Ivar Jacobsen และ Jim Rumbaugh

UML นั้นได้รวมแนวความคิดของวิธีการต่าง ๆ ทางออบเจกต์เข้าด้วยกัน จุดประสงค์ของ UML คือการสร้างโมเดลในการพัฒนาที่เข้าใจง่ายแต่สามารถนำไปประยุกต์ใช้ได้กับทุก ๆ ระบบตัว UML เองนั้นได้รับการสนับสนุนจากผู้นำในอุตสาหกรรมทางด้านคอมพิวเตอร์หลาย ๆ ราย

ใน UML มีโมเดลที่แสดงมุมมองต่อระบบที่ไม่เหมือนกันซึ่งประกอบไปด้วย

- Use-case Diagram ที่ใช้แสดงการติดต่อระหว่างระบบกับผู้ใช้
- คลาสไดอะแกรม (Class Diagram) ใช้แสดงโครงสร้างทางลอจิกของระบบ
- ออบเจกต์ไดอะแกรม (Object Diagram) ใช้แสดงออบเจกต์และความสัมพันธ์ระหว่างออบเจกต์
- สเตทไดอะแกรม (State Diagram) ใช้แสดงพฤติกรรมของออบเจกต์แต่ละตัวในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คอมโพเนนต์ไดอะแกรม (Component Diagram) ใช้แสดงโครงสร้างทางกายภาพของซอฟต์แวร์
- Deployment Diagram ใช้แสดงการติดต่อระหว่างฮาร์ดแวร์และซอฟต์แวร์
- Interactive Diagram ใช้แสดงพฤติกรรมของระบบ
- Activity Diagram ใช้แสดงการไหลของอีเวนต์ใน Use-case

2.7.2 การวิเคราะห์ระบบ

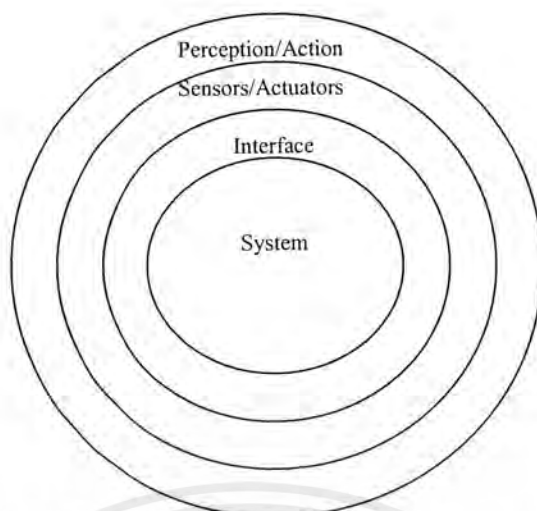
โดยปกติระบบจะต้องติดต่อกับสถานะแวดล้อมภายนอกซึ่งการที่เราจะรู้ว่าระบบติดต่อกับสถานะแวดล้อมภายนอกอย่างไรนั้นก็จำเป็นต้องมีการศึกษาความต้องการของระบบ หรือ Requirement Analysis จากเดิมในการวิเคราะห์ระบบแบบโครงสร้างเมื่อเราวิเคราะห์ความต้องการของระบบเป็นที่เรียบร้อยแล้ว จะได้ไดอะแกรม 2 ตัว คือ คอนเท็กซ์ไดอะแกรม และดาต้าโฟลว์ไดอะแกรม (Data flow Diagram) แต่สำหรับการวิเคราะห์ระบบด้วยออบเจกต์โอเรียนเท็ด หลังจากวิเคราะห์ความต้องการของระบบแล้วจะได้เป็น Use Case Diagram และ External Event Context โดย External Event Context จะจำลองระบบเป็นเหมือนกับออบเจกต์ใหญ่ ๆ ตัวหนึ่ง ซึ่งประกอบด้วยออบเจกต์ย่อยภายใน และออบเจกต์ของระบบนี้จะแสดงการติดต่อสื่อสารทั้งการรับและส่งแมสเซจ กับออบเจกต์ ที่เป็นแอกเตอร์ (Actor) ภายนอก ส่วน Use Case จะแสดงฟังก์ชันการทำงานหลักของระบบและโปรโตคอล หรือข้อตกลงในการแลกเปลี่ยนแมสเซจที่จำเป็นเพื่อให้ระบบสามารถติดต่อกับแอกเตอร์ภายนอกได้อย่างถูกต้องตามความต้องการของระบบ

อีเวนต์ภายนอก (External Event)

System Context เป็นสิ่งที่แสดงขอบเขตของระบบที่ผู้พัฒนาหรือผู้ใช้สนใจต่อระบบ ซึ่งเราจะใช้ Context Diagram ซึ่งจะแสดงสถานะแวดล้อมที่มีต่อระบบ ซึ่งประกอบไปด้วยผู้กระทำหรือแอกเตอร์ที่กระทำต่อระบบ นอกจากนี้ยังแสดงการแลกเปลี่ยนแมสเซจและอีเวนต์ของระบบกับสถานะแวดล้อมภายนอกอีกด้วย

Context-Level Objects

Context Diagram จะทำหน้าที่แสดงการติดต่อระหว่างออบเจกต์ของระบบกับออบเจกต์ภายนอก Ellis ได้แบ่งออบเจกต์ของระบบและออบเจกต์ภายนอกโดยการจำลองระบบออกมาเป็นหัวหอม (Onions) ซึ่งจะมีผิวเป็นชั้น ๆ ซึ่งแต่ละชั้นก็มีความสำคัญต่อระบบที่แตกต่างกันไปดังรูป 2-9



รูปที่ 2-9 การเปรียบเทียบระบบแบบ Onions Skin

สำหรับผิวนอกสุดของภาพเปรียบเทียบเป็น Perception/Action ซึ่งเป็นส่วนแสดงแอกชันของระบบซึ่งชั้นนี้ทำให้ผู้ใช้สามารถเห็นการกระทำของระบบที่มีต่อสภาพแวดล้อมได้

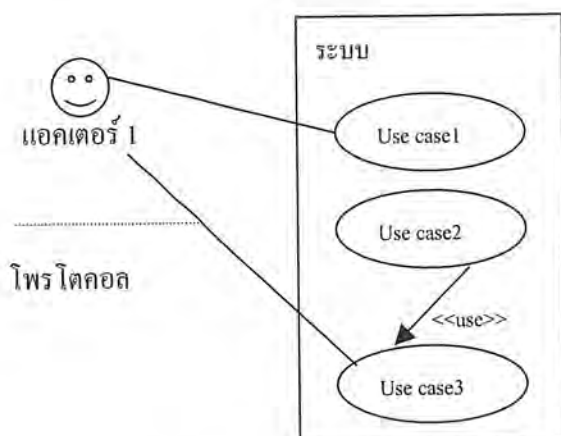
สำหรับชั้นต่อไป Sensors/Actuators เป็นชั้นของอุปกรณ์ที่ทำหน้าที่ติดต่อกับสิ่งแวดล้อม

สำหรับชั้นต่อไปอินเตอร์เฟซจะเป็นสำหรับอุปกรณ์ที่ใช้ในการสื่อสาร และชั้นสุดท้าย คือชั้นของระบบศูนย์กลางที่ทำหน้าที่ตัดสินใจ และประมวลผลข้อมูล

Use Case

Context Diagram ที่ได้จากการวิเคราะห์นั้นจะบอกถึงการติดต่อกันระหว่างสิ่งแวดล้อมภายนอกกับระบบ สำหรับการวิเคราะห์ระบบแบบ ออบเจกต์โอเรียนเท็ดนั้นเราสามารถสร้าง Use case Diagram ที่เพิ่มกรณีการติดต่อกันระหว่างระบบกับแอกเตอร์ภายนอก นอกจากนี้ยังใช้ในการตรวจสอบความถูกต้องของ Context Diagram

Use case Diagram จะแสดงมุมมองต่อฟังก์ชันการทำงานหลัก ๆ ของระบบ ซึ่งผู้พัฒนาระบบสามารถใช้ในการสื่อสารกับผู้ใช้ระบบได้ด้วย Use case Diagram ในระดับบนสุดนั้นจะแสดงขอบเขตของระบบทั้งหมดซึ่งเป็นมุมมองจากฟังก์ชันการทำงาน ไม่ใช่มุมมองต่อคลาสและออบเจกต์ที่มีในระบบ โดยฟังก์ชันการทำงานของระบบนั้นจะแทนที่ด้วย วงรีและอยู่ในระบบใหญ่ซึ่งแสดงโดยกรอบสี่เหลี่ยม โดยมีออบเจกต์กระจายอยู่นอกระบบ ออบเจกต์ภายนอกเหล่านี้จะสัมพันธ์กับ Use Case ผ่านโปรโตคอลของระบบนอกจากนั้น Use Case หนึ่ง ๆ อย่างจะมีความสัมพันธ์กับ Use Case อื่น ๆ ซึ่งเราสามารถแทนได้โดยการลากเส้นแสดงความสัมพันธ์ระหว่าง Use Case ดังรูป 2-10



รูปที่ 2-10 Use Case Diagram

การกำหนด Use Case

ในการวิเคราะห์ระบบส่วนใหญ่จะใช้วิธีการกำหนด Scenario ขึ้นมาก่อนซึ่ง Scenario ที่กำหนดขึ้นมานี้อาจจะมีมากกว่า Use case จากนั้นนักพัฒนาจะเลือก Scenario ที่สำคัญต่อระบบขึ้นมาเพื่อทำเป็น Use case

Scenarios

Scenario คือตัวตนของ Use case ในโมเดลของ Scenario จะกำหนดลำดับของแมสเชจของระบบ รวมถึงโครงสร้างของออบเจกต์ที่ทำงานร่วมกันเพื่อให้ระบบสามารถทำหน้าที่ที่รับผิดชอบได้ ข้อดีของ Scenario คือ ทำให้ผู้พัฒนาระบบและผู้ใช้สามารถเข้าใจการทำงานของระบบได้ง่ายขึ้น นอกจากนี้จึงใช้เป็นเครื่องมือในการตรวจสอบความต้องการของระบบ

ในขั้นตอนของการวิเคราะห์ระบบนั้นเมื่อเราได้ Context Diagram และ Use case Diagram แล้วเราก็วิเคราะห์ต่อโดยแยกระบบออกเป็นออบเจกต์ย่อย ๆ จากนั้นก็กำหนด Scenario ขึ้นสำหรับแต่ละ Use case โดย Scenario นี้จะแสดงการรับส่งแมสเชจระหว่างออบเจกต์ที่เรากำหนดขึ้นมาเพื่อทำงานที่รับผิดชอบตาม Use Case ได้สำเร็จ จะเห็นได้ว่าในขั้นตอนนี้จะเป็นการตรวจสอบ Use Case และออบเจกต์ที่เรากำหนดขึ้นมาว่าครบหรือไม่

การวิเคราะห์ Scenario ไม่มีกฎเกณฑ์ที่แน่นอนแต่อาศัยความชำนาญของผู้วิเคราะห์และความเข้าใจในระบบ

UML มีโมเดลที่รองรับการวิเคราะห์ Scenario อยู่ 2 โมเดล คือ

- Sequence Diagram ที่แสดงถึงลำดับของแมสเชจที่ออบเจกต์ในระบบส่งและรับซึ่งกันและกัน
- Collaboration Diagram จะแสดงโครงสร้างของออบเจกต์ที่ทำงานร่วมกันเป็นหลัก

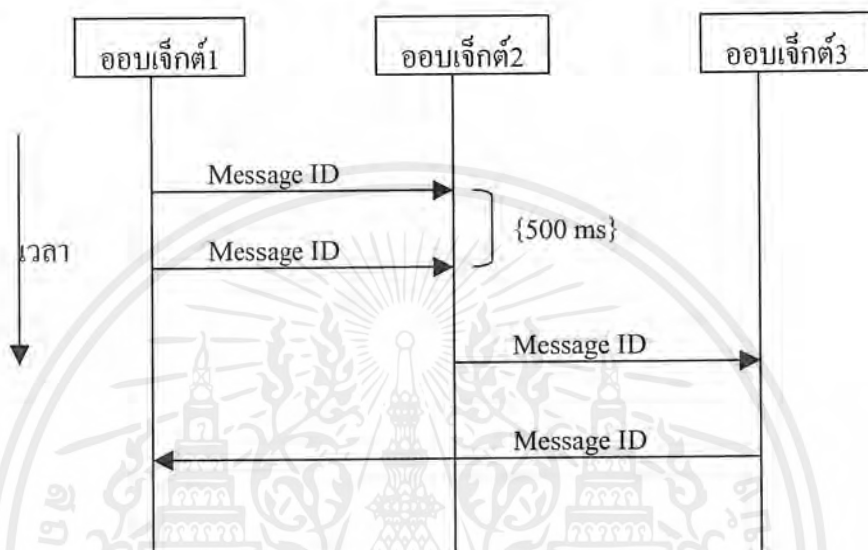
Sequence Diagram

ส่วนประกอบของ Sequence Diagram จะมีเส้นในแนวตั้งหมายถึงออบเจกต์โดยมีชื่อของออบเจกต์เขียนอยู่ด้านบนหรือด้านล่างของเส้น ส่วนเส้นในแนวนอนนั้นแสดงถึงแมสเชจ เส้นตรงที่แทนแมสเชจทุกเส้นจะต้องมีออบเจกต์ต้นกำเนิดของแมสเชจ ส่วนจุดสิ้นสุดของแมสเชจก็จะเป็นออบเจกต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปลายทาง โดยแต่ละเส้นของแมสเสจจะมีชื่อกำกับอยู่ โดยชื่อนี้อาจมีการเพิ่มรายการของพารามิเตอร์และค่าคงที่คืนกลับมา (Return Value) ด้วย สำหรับเวลานั้นจะนับจากบนลงล่าง โดยสเกลของแกนตั้งจะบอกเพียงลำดับของแมสเสจเท่านั้น ไม่สามารถใช้เป็นอัตราส่วนในการคำนวณเวลาระหว่างแมสเสจได้

Sequence Diagram ส่วนใหญ่จะประกอบไปด้วยองค์ประกอบต่าง ๆ คือ ออบเจกต์ แมสเสจ คาบเวลาและสัญลักษณ์ของ Scenario นอกจากนี้ยังสามารถเพิ่มข้อมูลที่จำเป็นเข้าไปใน Sequence Diagram ได้สำหรับตัวอย่าง Sequence Diagram ที่เขียนขึ้นดังรูป 2-11

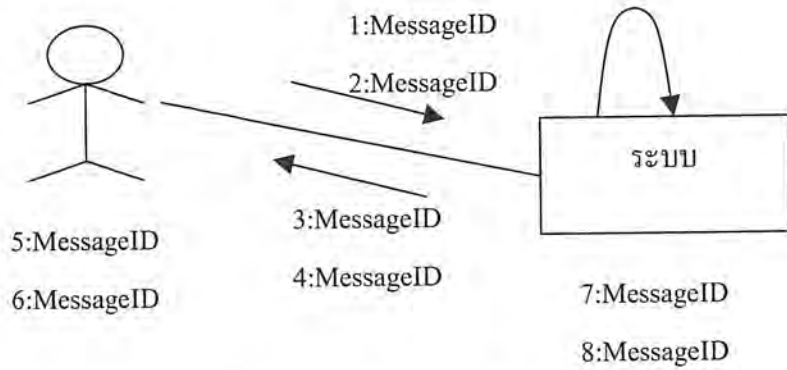


รูปที่ 2-11 Sequence Diagram

สำหรับ Sequence Diagram ที่ได้กล่าวไปจะมีแมสเสจในรูปแบบ Peer-to-Peer คือ มีออบเจกต์ตัวหนึ่งส่งแมสเสจให้ออบเจกต์อีกตัวหนึ่งแต่ก็สามารถมีแมสเสจแบบกระจายได้ ซึ่งเป็นแมสเสจที่ส่งออกจากออบเจกต์หนึ่ง ไปให้กับออบเจกต์อีกหลาย ๆ ตัว

Collaboration Diagram

Collaboration Diagram เป็นเครื่องมือสำหรับใช้แสดง Scenario อีกตัวหนึ่งที่มีประโยชน์ Collaboration Diagram นั้นจะแสดงข้อมูลพื้นฐาน เช่นเดียวกับ Sequence Diagram แต่ความแตกต่างคือ Sequence Diagram จะแสดงลำดับของแมสเสจเป็นหลัก ส่วน Collaboration Diagram นั้นจะแสดงโครงสร้างของออบเจกต์ที่ทำงานร่วมกันเป็นหลัก ตัวอย่างของ Collaboration Diagram ดังรูป 2-12



รูปที่ 2-12 Collaboration Diagram

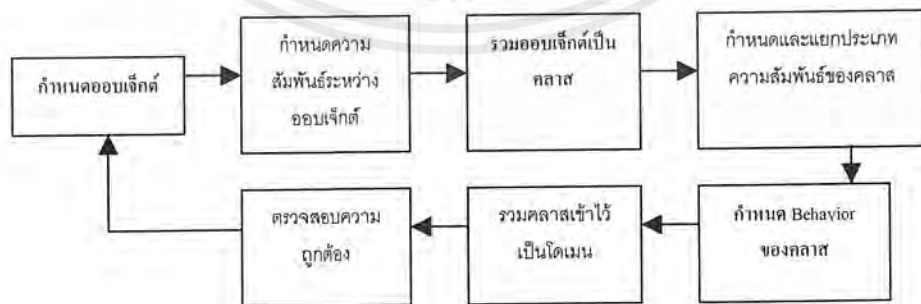
สำหรับข้อเสีย Collaboration Diagram คือ เมื่อมีการเปลี่ยนแปลงโครงสร้างของออบเจกต์แล้วจำเป็นต้องมีการแก้ไขไคอะแกรม มากกว่า Sequence Diagram อย่างไรก็ตาม Collaboration Diagram นี้ก็สามารถใช้แสดงโครงสร้างของออบเจกต์เพื่อเป็นแนวทางในการสร้างคลาสไคอะแกรมต่อไป

สำหรับ Collaboration Diagram ประกอบด้วยส่วนต่าง ๆ ดังนี้ คือ

- ชื่อออบเจกต์
- ความสัมพันธ์ระหว่างออบเจกต์ที่มีการแลกเปลี่ยนเมสเสจกัน
- เมสเสจที่ประกอบไปด้วยหมายเลขลำดับ , ตัวระบุ(Identifier) และทิศทางของเมสเสจ

กระบวนการทางออบเจกต์

Use Case และ Context Diagram ที่เราสร้างขึ้นมานั้นเป็นจุดเริ่มต้นในการวิเคราะห์ระบบแบบออบเจกต์โอเรียนเท็ดซึ่งเมื่อวิเคราะห์ระบบเสร็จเรียบร้อยแล้วจะได้เป็นโมเดลซึ่งประกอบไปด้วยออบเจกต์ คลาส ความสัมพันธ์ระหว่างออบเจกต์และความสัมพันธ์ระหว่างคลาส ลำดับชั้นของการสืบทอดคุณสมบัติซึ่งถือว่าเป็นคุณสมบัติที่สำคัญที่สุดของออบเจกต์โอเรียนเท็ด กระบวนการทางออบเจกต์แสดงได้ดังรูป 2-13



รูปที่ 2-13 ขั้นตอนการวิเคราะห์แบบออบเจกต์โอเรียนเท็ด

การกำหนดออบเจกต์

การขีดเส้นใต้คำถาม

วิธีนี้ใช้กับคำอธิบายปัญหาที่ได้จากผู้ใช้ระบบโดยตรง คือการขีดเส้นใต้คำถาม หรือวลีที่เป็นคำถามอยู่ในคำอธิบายปัญหา หลังจากที่เราได้ขีดเส้นใต้คำถามแล้วเราจะได้ออบเจกต์ต่าง ๆ ที่แบ่งออกเป็น 3 ประเภท คือ

- ออบเจกต์ที่พัฒนาระบบสนใจ
- ออบเจกต์ที่พัฒนาระบบไม่สนใจ
- คุณสมบัติของออบเจกต์

โดยจุดประสงค์หลักในการกำหนดออบเจกต์ คือ ต้องการหาออบเจกต์ที่ผู้พัฒนาสนใจ ส่วนออบเจกต์ที่ไม่ได้สนใจนั้นก็อาจจะมิประโยชน์ต่อการวิเคราะห์ระบบในภายหลัง

กลยุทธ์ในการขีดเส้นใต้คำถามเพื่อกำหนดออบเจกต์นั้นเป็นวิธีที่ง่ายและรวดเร็วในการค้นหาออบเจกต์รวมทั้งคุณสมบัติของออบเจกต์จากคำอธิบายปัญหา แต่ว่าการใช้กลยุทธ์นี้ทำให้เราได้ออบเจกต์ที่ไม่สนใจมากอีกทั้งยังมีความกำกวมซึ่งผู้วิเคราะห์ระบบจำเป็นต้องแก้ปัญหาเหล่านี้ด้วย

การกำหนดความสัมพันธ์ระหว่างออบเจกต์

จะดูจากการส่งแมสเซจเป็นหลักเพราะว่าแมสเซจทุกแมสเซจจะหมายถึงความสัมพันธ์ระหว่างออบเจกต์ซึ่งเป็นผู้ส่งแมสเซจกับออบเจกต์ซึ่งเป็นผู้รับแมสเซจ

การกำหนดคุณสมบัติของออบเจกต์

คุณสมบัติของออบเจกต์เป็นส่วนข้อมูลของออบเจกต์ที่ไม่สามารถแบ่งแยกย่อยลงไปได้อีก ถ้าในระหว่างการวิเคราะห์พบว่าคุณสมบัติที่กำหนดยังสามารถแยกย่อยลงไปได้แสดงว่าควรกำหนดคุณสมบัตินั้นเป็นออบเจกต์ที่อยู่ภายใน

ในระบบงานทั่ว ๆ ไป บางครั้งพบว่าบางออบเจกต์ไม่จำเป็นต้องมีคุณสมบัติแต่มีเพียงเมธอดก็เพียงพอแล้ว

การกำหนดคลาส

ในการกำหนดออบเจกต์โดยใช้คำอธิบายของปัญหานั้นเราจะได้ออบเจกต์ออกมามากมายโดยมีหลายออบเจกต์ที่มีโครงสร้างเหมือนกัน ออบเจกต์ที่มีโครงสร้างข้อมูลและเมธอดเหมือนกันเราสามารถโมเดลให้เป็นคลาสได้ในขั้นตอนของการวิเคราะห์คือการกำหนดคลาสสำหรับแต่ละออบเจกต์ในระบบ

คลาสเป็นการสร้างเอกลักษณ์ของออบเจกต์ เราสามารถกำหนดชนิดของออบเจกต์ให้เป็นคลาสได้ คลาสที่กำหนดขึ้นจะต้องมีตัวระบุให้กับแต่ละออบเจกต์ด้วย เช่น ออบเจกต์บัญชี เราอาจจะให้ตัวระบุเป็นหมายเลขบัญชีเป็นต้น

คลาสไดอะแกรม

คลาสไดอะแกรมเป็นไดอะแกรมที่มีความสำคัญมากที่สุดในการวิเคราะห์และออกแบบโดยใช้วิธีทางออบเจกต์โอเรียนเท็ด เนื่องจากจะแสดงโครงสร้างของออบเจกต์และคลาสที่มีในระบบรวมทั้งแสดงความสัมพันธ์ด้วย คลาสไดอะแกรมจะเป็นโครงสร้างหลักของระบบและใช้ในการแยกย่อยรายละเอียดของระบบด้วยวิธีทางออบเจกต์โอเรียนเท็ด อย่างไรก็ตามก็ดีที่คลาสไดอะแกรมนั้นอาจจะมีลักษณะคล้ายเอกสารนี้เป็นเอกสารที่ส่งวนไวกว้างสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออบเจกต์ไคอะแกรมแต่ว่าคลาสไคอะแกรมจะแสดงเฉพาะคลาสโดยไม่แสดง Instance หรือออบเจกต์ทั้งหมด

ในบางกรณีอาจจำเป็นต้องเพิ่มคลาสที่แสดงความสัมพันธ์ (Association Class) เนื่องจากความสัมพันธ์นี้สามารถมีคุณสมบัติและเมธอดเป็นของตัวเอง

ความสัมพันธ์ระหว่างออบเจกต์

ในระบบออบเจกต์โอเรียนท์เตดแบ่งความสัมพันธ์ระหว่างออบเจกต์ออกเป็น 4 ประเภทดังนี้ คือ

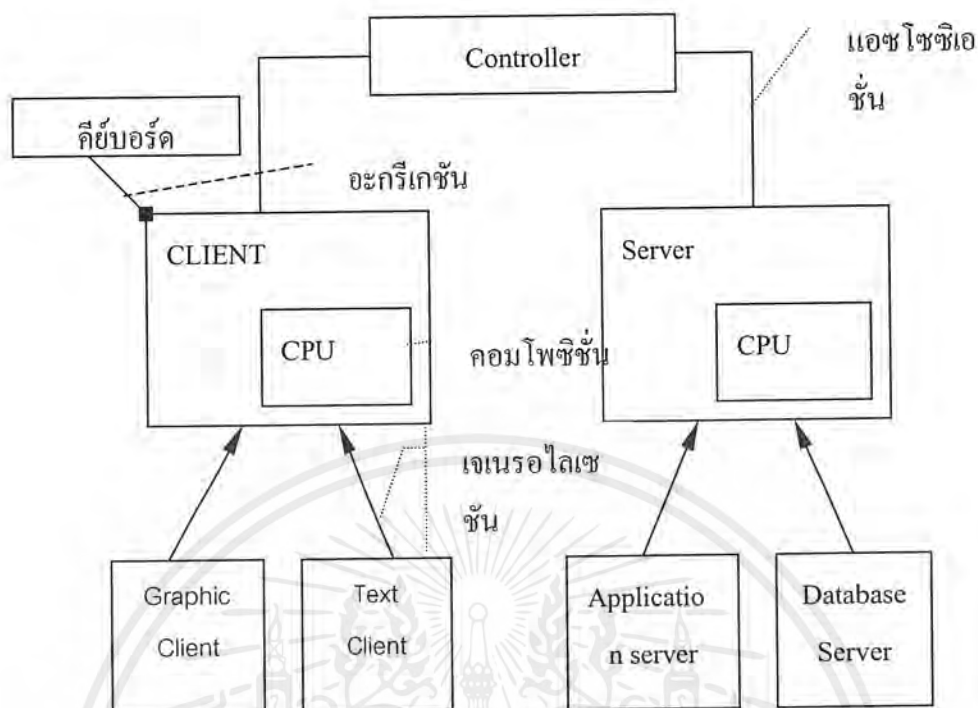
- แอชโซซิเอชัน (Association)
- อะกรีเกชัน (Aggregation)
- คอมโพสิชัน (Composition)
- เจเนอไรเซชัน (Generalization)

ความสัมพันธ์แอสโซซิเอชันนั้น เป็นความสัมพันธ์ระหว่างคลาส เมื่อออบเจกต์ของ คลาสมีการส่งเมสเสจถึงกัน ในการสร้างความสัมพันธ์แบบแอสโซซิเอชันสามารถทำได้โดยประกาศ Instance ของออบเจกต์ที่เป็นผู้รับเมสเสจไว้ภายในออบเจกต์ของผู้ส่งเมสเสจ จากนั้นก็ให้ Instance ที่ประกาศไว้เป็นตัวอ้างอิงไปยังออบเจกต์ที่เป็นผู้รับเมสเสจ และเมื่อออบเจกต์ต้องการส่งเมสเสจก็เรียกฟังก์ชันจาก Instance ออบเจกต์ผู้รับเมสเสจ

ความสัมพันธ์แบบอะกรีเกชัน และคอมโพสิชันเป็นความสัมพันธ์ระหว่างคลาสที่มีลักษณะคล้ายกัน กล่าวคือการที่มีออบเจกต์ของคลาสหนึ่งเป็นส่วนหนึ่งของอีกออบเจกต์ของคลาสหนึ่งในการสร้างความสัมพันธ์แบบอะกรีเกชัน และคอมโพสิชันสามารถทำได้โดยการประกาศ Instance ของออบเจกต์ไว้ภายในคลาสที่โดนสร้างเป็นออบเจกต์เจ้าของโดยออบเจกต์เจ้าของจะเป็นผู้สร้างและทำลายออบเจกต์ที่อยู่ภายใน

ความสัมพันธ์แบบเจเนอไรเซชัน เป็นความสัมพันธ์เมื่อมีออบเจกต์หนึ่งสืบทอดคุณสมบัติมาจากอีกออบเจกต์หนึ่ง โดยใน ความสัมพันธ์นี้ เจเนอไรเซชันนั้นคลาสที่ได้รับการสืบทอดคุณสมบัติมาหรือสับคลาสนั้นได้รับการสร้างเป็นออบเจกต์ที่มีคุณสมบัติและเมธอดที่เหมือนกับออบเจกต์ที่สร้างจากซูเปอร์คลาสและยังมีคุณสมบัติและเมธอดที่เพิ่มเข้ามาด้วย

จากความสัมพันธ์ทั้งหมดสามารถแสดงสรุปได้ดังรูป 2-14



รูปที่ 2-14 ความสัมพันธ์ระหว่างคลาสในแบบต่าง ๆ

การกำหนดเม็ททอดของออบเจกต์

เม็ททอดเป็นส่วนที่บ่งบอกถึงคุณสมบัติของและหยาที่รับผิดชอบของออบเจกต์ ในการโมเดลโอเปอเรชันและเม็ททอดจะใช้สเตทโคอะแกรม ด้วยการใส่สเตทโคอะแกรมจะทำให้เราสามารถตรวจสอบว่าออบเจกต์สามารถทำงานร่วมกันได้อย่างถูกต้องหรือไม่

ออบเจกต์เม็ททอด

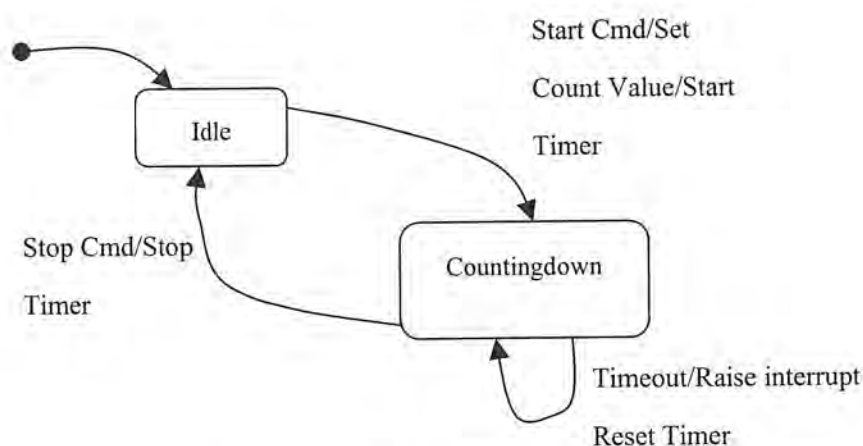
จากที่นำเสนอไปก่อนหน้านี้เป็นการกำหนดออบเจกต์ คุณสมบัติของออบเจกต์รวมทั้งโครงสร้างของออบเจกต์และความสัมพันธ์ ความสัมพันธ์ระหว่างออบเจกต์จะเกิดขึ้นได้ก็ต่อเมื่อมีการเรียกเม็ททอดของออบเจกต์

การกำหนดสเตทของออบเจกต์


สำหรับการวิเคราะห์และออกแบบระบบแบบออบเจกต์โอเรียนเต็ลเราจะกำหนดให้ 1 คลาสมี 1 สเตทโคอะแกรม ตัวสเตทนั้นแสดงด้วยสี่เหลี่ยมที่มีขอบมน สำหรับการเปลี่ยนสเตทจะแสดงด้วยเส้นตรงที่มีหัวลูกศร โดยเริ่มต้นจากสเตทที่เป็นต้นทางไปยังสเตทที่เป็นปลายทาง สำหรับรายละเอียดการเปลี่ยนสเตทนั้นจะประกอบไปด้วย อีเวนต์ที่ทำให้เกิดการเปลี่ยนสเตทตามด้วยแอคชัน(ฟังก์ชันหรือโอเปอเรชัน)ที่จะต้องทำเมื่อมีการเปลี่ยนสเตท ส่วนเครื่องหมายวงกลมระบายสีดำเป็นจุดเริ่มต้นของสเตทโคอะแกรม

ดังรูป 2-15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-15 สเตตไคอะแกรมของตัวจับเวลา

สัญลักษณ์  นั้นเรียกว่า คอนดิชันนอลคอนเน็คเตอร์ (Condition Connector) ซึ่งยอมให้มีการเปลี่ยนสเตตได้หลายสเตตขึ้นอยู่กับเงื่อนไขหรือ การ์ด ตัวการ์ดจะเขียนอยู่ในวงเล็บ [] ถ้ามีอีเวนต์เกิดขึ้นแล้วการ์ดเป็นจริงหรือไม่เป็นจริงจะทำให้เกิดการเปลี่ยนแปลงสเตตตามที่กำหนดไว้

สเตตไคอะแกรมพื้นฐานใน UML

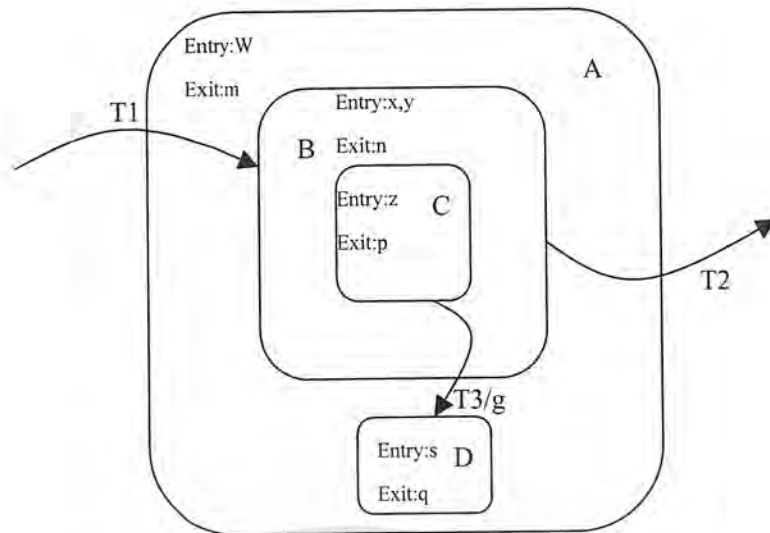
ไวยากรณ์ของสเตตไคอะแกรมใน UML ได้อธิบายไปแล้วบางส่วนในรายงานส่วนนี้จะนำเสนอส่วนที่เพิ่มความสามารถต่าง ๆ ที่สำคัญสำหรับ สเตตไคอะแกรม

- สามารถทำสเตตที่ซ้อนสเตตได้โดยสเตตภายในเรียกว่า สับสเตต ส่วนสเตตภายนอกเรียกชุปเปอร์สเตต
- มีการกำหนดสเตตเริ่มต้นให้กับชุปเปอร์สเตตได้เมื่อมีการเปลี่ยนสเตตมาที่ชุปเปอร์สเตตนี้ก็ให้เริ่มที่สเตตเริ่มต้นนี้
- สามารถใช้ฮิสทอรีคอนเน็คเตอร์ (History Connector) คือ เมื่อกลับเข้ามาชุปเปอร์สเตตนี้อีกทำให้เริ่มสเตตที่ค้างอยู่ก่อนที่จะมีการเปลี่ยนสเตตออกไป ถ้าไม่มีก็ให้เริ่มที่ฮิสทอรีสเตต

สเตตไคอะแกรมสามารถมีแอคชัน ใน ตัวสเตตเองหรือในช่วงการเปลี่ยนแปลงสเตตก็ได้โดยสามารถแบ่งเป็น

- แอคชันที่ทำเมื่อมีการเข้าสู่สเตต หรือ Entry Action (ใช้สัญลักษณ์ “Entry:”)
- แอคชันที่ทำเมื่อมีการออกจากสเตต หรือ Exit Action (ใช้สัญลักษณ์ “Exit:”)
- แอคชันที่ทำเมื่อสเตตนั้น ๆ แอคทีฟหรือ Activities (ใช้สัญลักษณ์ “Do:”)

จากที่กล่าวมาสามารถสรุปได้ดังรูป 2-16



รูปที่ 2-16 สเตทไดอะแกรมสำหรับซูเปอร์สเตทและสับสเตท

สำหรับทรานซิชันอาจจะมีพารามิเตอร์และการ์ดรวมทั้งแอคชันโดยไวยากรณ์สรุปได้ดังนี้
 ชื่ออีเวนต์(รายการพารามิเตอร์)[การ์ด]/รายการแอคชันสำหรับการเปลี่ยนสเตท ^ รายการอีเวนต์
 ซึ่งสรุปได้ดังตาราง 2-1

คำ	คำอธิบาย
ชื่ออีเวนต์	เป็นชื่ออีเวนต์ที่ทำให้เกิดการเปลี่ยนสเตท
การ์ด	เป็นเงื่อนไขที่อยู่ในรูปประโยคบูลีน ซึ่งจะต้องมีค่าเป็นจริงจึงจะทำให้เกิดการเปลี่ยนสเตทได้ โดยปกติจะใช้ร่วมกับคอนดิชันนอลคอนเนคเตอร์เพื่อตัดสติใจว่าจะให้เปลี่ยนไปยังสเตทใด
รายการแอคชัน	เป็นรายการที่ใช้เครื่องหมายลูกน้ำแยกแต่ละส่วน โดยเป็นแอคชันที่จะต้องเกิดขึ้นเมื่อมีการเปลี่ยนแปลงสเตท
รายการอีเวนต์	เป็นรายการอีเวนต์ที่ใช้ลูกน้ำขึ้นลำดับ โดยจะเป็นอีเวนต์ที่จะต้องเกิดขึ้นเมื่อมีการเปลี่ยนแปลงสเตท

ตารางที่ 2-1 แสดงความหมายของพารามิเตอร์ต่าง ๆ ของทรานซิชัน

การกำหนดเม็ททอดหรือโอเปอเรชัน

การกำหนดเม็ททอดหรือ โอเปอเรชันมีหลักการดังนี้คือ

- กำหนดเม็ททอดหรือ โอเปอเรชันสำหรับออบเจกต์โดยเป็นอิสระต่อกัน
- พยายามซ่อนรายละเอียด โครงสร้างภายในออบเจกต์โดยให้ผู้ต้องการติดต่อต้องผ่านอินเตอร์

เฟสที่กำหนดไว้เท่านั้น

- ซูเปอร์คลาสควรมีเม็ททอดหรือ โอเปอเรชันที่สับคลาสส่วนใหญ่ต้องการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมทรูดหรือ โอเปอเรชันที่ทำหน้าที่จัดการเกี่ยวกับแมสเซจหรืออีเวนต์ประกอบไปด้วย
 - โอเปอเรชันที่จัดการกับอีเวนต์ที่เกิดขึ้น
 - โอเปอเรชันที่จัดการการรับส่งแมสเซจ
 - โอเปอเรชันที่ทำหน้าที่อ่านหรือเขียนค่าข้อมูลของคุณสมบัติต่าง ๆ ของออบเจกต์
 - เมทรูดหรือ โอเปอเรชันสำหรับแอคชันที่ปรากฏอยู่บนสเตทโคอะแกรม

จากที่กล่าวมาทั้งหมดเป็นวิธีการหาเมทรูดหรือ โอเปอเรชันเพื่อทำให้การวิเคราะห์ระบบมีความถูกต้องมากที่สุดโดยจะได้ผลลัพธ์นิคคลาสโคอะแกรมที่ค่อนข้างจะสมบูรณ์โดยแต่ละคลาสประกอบไปด้วย คุณสมบัติ และ เมทรูดหรือ โอเปอเรชัน

2.7.3 การออกแบบ

ในกระบวนการการออกแบบนั้นจะเป็นการปรับเปลี่ยนโครงสร้างและการทำงานของโมเดลที่ได้จากการวิเคราะห์ระบบ โดยสามารถแบ่งออกเป็น 3 ระดับ คือ

- การออกแบบในระดับสถาปัตยกรรม (Architectural Design) เป็นการออกแบบในระดับโครงสร้างของระบบในระดับใหญ่คือ ระบบย่อย (Subsystem) แพคเกจ (Package) และทาสก์ (Task)
- การออกแบบในระดับโครงสร้างการทำงาน (Mechanistic Design) เป็นการออกแบบที่เน้นเกี่ยวกับโครงสร้างการทำงานระหว่างคลาสและออบเจกต์ที่มีอยู่ในระบบเพื่อให้บรรลุหน้าที่ที่รับผิดชอบ
- การออกแบบในระดับรายละเอียด (Detail Design) จะกล่าวถึงโครงสร้างข้อมูลและอัลกอริทึมในการทำงานของคลาสดังกล่าว

จากที่กล่าวมาทั้งหมดแนวทางการออกแบบระบบเพื่อปรับปรุงสิ่งที่ได้จากการวิเคราะห์ให้ถูกต้องมากที่สุดและสามารถนำกลับมาใช้ใหม่ได้มากที่สุด ซึ่งเป็นจุดประสงค์ที่สำคัญที่สุดในการวิเคราะห์และออกแบบระบบแบบออบเจกต์โอเรียนเต็ล ดังนั้นในปัจจุบันได้มีการเก็บรูปแบบการออกแบบในระบบเก่าเพื่อนำกลับมาประยุกต์ใช้กับระบบใหม่โดยอาจจะมีการปรับเปลี่ยนในบางส่วนบ้างก็ตาม รายละเอียดที่เก็บจะเกี่ยวกับ ข้อกำหนดของคลาสและออบเจกต์รวมทั้งความสัมพันธ์ระหว่างกันในระบบ ดังนั้นผู้ที่ออกแบบระบบมือใหม่ก็สามารถออกแบบระบบได้อย่างมีประสิทธิภาพด้วยการศึกษารูปแบบการออกแบบที่มีผู้นำเสนอไว้แล้วไปประยุกต์ใช้ทำให้ไม่ต้องเสียเวลาในการศึกษาและสร้างวิธีการในการออกแบบระบบของตนเองขึ้นมา รูปแบบการออกแบบนี้เราเรียกว่า ดีไซน์แพทเทิร์น (Design Pattern) ในภาคผนวก ข จะแสดงรายละเอียดของดีไซน์แพทเทิร์นที่ใช้ในการออกแบบระบบในระดับต่าง ๆ

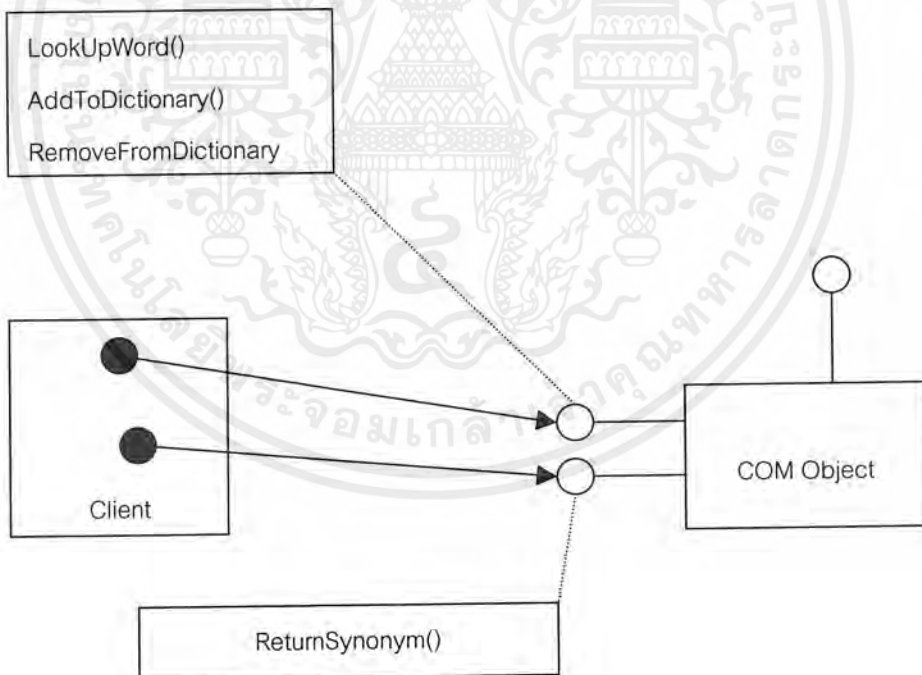
บทที่ 3

เทคโนโลยี COM และ DCOM และการพัฒนา (COM/DCOM Technology and Development)

3.1 COM (Component Object Model) และ DCOM (Distribute COM)

3.1.1 คำอธิบายวัตถุ COM

COM เป็นมาตรฐานแบบไบนารีในการทำงานร่วมกันระหว่างคอมพิวเตอร์ ซึ่งมาตรฐานนี้ไม่ขึ้นกับตัวแอปพลิเคชันและภาษาที่ใช้พัฒนา โดยแอปพลิเคชันจะติดต่อกับแอปพลิเคชันอื่น ๆ และกับระบบ โดยผ่านกลุ่มของฟังก์ชันหรืออินเทอร์เฟซ ดังรูปที่ 3-1 ซึ่งแสดงการเรียกใช้บริการของไคลเอนต์ด้านดิคชันนารีจากวัตถุ COM



รูปที่ 3-1 วัตถุ COM, ไคลเอนต์ และอินเทอร์เฟซ

3.1.1.1 อินเทอร์เฟซ (Interface)

ใน COM อินเทอร์เฟซ คือ ข้อตกลงระหว่างซอฟต์แวร์คอมพิวเตอร์กับผู้ใช้งาน อินเทอร์เฟซทุกตัวจะมีรูปแบบการอ้างถึงสองแบบ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การอ้างแบบเรียกชื่อ: ชื่อของอินเตอร์เฟซขึ้นต้นด้วย "I" เป็นการอ้างกันในมนุษย์
- การอ้างแบบใช้หมายเลข: เป็นการอ้างในระดับซอฟต์แวร์โดยจะมีการใช้หมายเลขที่เรียกว่า *globally unique identifier (GUID)* เป็นเลขขนาด 128 บิตโดยแต่ละอินเตอร์เฟซจะต้องมีหมายเลขนี้ที่ไม่ซ้ำกันเราเรียกหมายเลขที่กำหนดให้กับอินเตอร์เฟซว่า "IID" (Interface Identifier)

อินเตอร์เฟซสามารถระบุโดยภาษาที่เรียกว่า Interface Definition Language (IDL) ซึ่งเป็นภาษาที่ระบุอินเตอร์เฟซว่ามี IID อะไรและมีฟังก์ชันการทำงานรวมถึงพารามิเตอร์ต่างๆ เป็นอย่างไร ดังตัวอย่างเป็นการแสดงภาษา IDL ที่ระบุการทำงานของอินเตอร์เฟซของวัตถุ COM ด้านฟังก์ชัน ดิกชันนารี

[Object,

uuid(E7CD0D00-1827-11CF-9946-444553540000)]

interface ISpellChecker :IUnknown {

import "unknown.idl";

HRESULT LookUpWord([in] Olechar woed[31],[out] boolean *found);}

เราจะนำโค้ดที่เขียนขึ้นเพื่อระบุอินเตอร์เฟซนี้ไปใช้ในการสร้าง Type Library (โดยผ่านการคอมไพล์) เพื่ออำนวยความสะดวกในการพัฒนาซอฟต์แวร์แบบ COM Type Library เป็น ไบนารีโค้ดที่อธิบายอินเตอร์เฟซที่คอมโพเนนต์นั้น ๆ มีซึ่งเครื่องมือพัฒนาสมัยใหม่สามารถนำเอา Type Library ไปใช้ในการเขียนโปรแกรมโดยไม่ต้องรู้ถึงรายละเอียดของอินเตอร์เฟซ (เช่น IID) เพราะเครื่องมือพัฒนาจะทำให้ Type Library อยู่ในรูปแบบที่สามารถเรียกใช้ได้ง่ายในสภาวะแวดล้อมนั้น ๆ ดังรูปที่ 3-2



รูปที่ 3-2 การนำ Type Library ไปใช้ในเครื่องมือพัฒนาต่าง ๆ

อินเตอร์เฟซทุกตัวต้องสืบทอดมาจากอินเตอร์เฟซ IUnknown เนื่องจากอินเตอร์เฟซ IUnknown ได้มีฟังก์ชันในการทำงานด้านการสร้างและจัดการ COM ออบเจ็กต์ประกอบด้วย

- เมธอด QueryInterface ทำหน้าที่ในการถามออบเจ็กต์ว่ามีอินเตอร์เฟซอื่นนอกจาก IUnknown ที่ออบเจ็กต์นั้นสนับสนุนหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

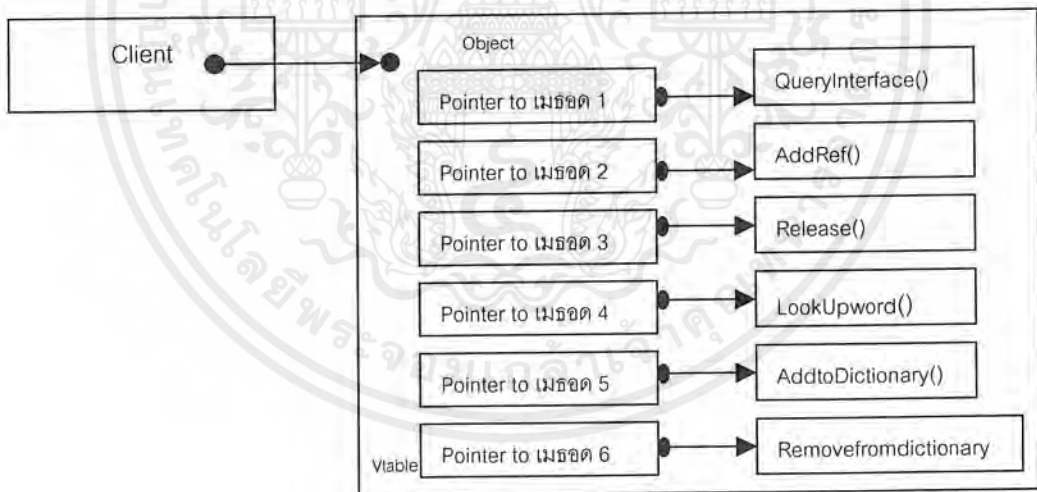
• เมธอด AddRef และ Release เป็นตัวจัดการนับการใช้ออบเจกต์และตรวจสอบว่าอบเจกต์ยังถูกเรียกใช้งานอีกหรือไม่ ถ้าไม่จะได้รีปล่อยตัวเองออกจากหน่วยความจำ เพื่อประหยัดทรัพยากรของระบบ

อินเตอร์เฟสที่เรามีการประกาศเพื่อใช้งานในคอมโพเนนต์แล้วเราไม่สามารถที่จะเปลี่ยนแปลงได้ เนื่องจากจะส่งผลกระทบต่อไคลเอนต์ที่มีการเรียกใช้อินเตอร์เฟสนั้น

COM สามารถสร้างคอมโพเนนต์ที่เปลี่ยนแปลงได้ง่ายเพื่อรองรับการทำงานในอนาคต แต่เดิมปัญหาในการพัฒนาซอฟต์แวร์เชิงคอมโพเนนต์มีปัญหาที่สำคัญ คือ ปัญหาด้านเวอร์ชัน (Versioning) กล่าวคือเมื่อคอมโพเนนต์มีการเปลี่ยนแปลงต่าง ๆ จะทำให้ซอฟต์แวร์เดิมที่เรียกใช้คอมโพเนนต์นั้นในการทำงานมีปัญหาบางอย่าง เช่น ไฟล์ DLL เป็นต้น ในการพัฒนาโปรแกรมบนวินโดวส์แบบเดิมเมื่อมีการเปลี่ยนแปลงเป็นเวอร์ชันที่ซอฟต์แวร์ที่เรียกใช้ไม่รู้จักก็จะเกิดปัญหาได้ ใน COM จะแก้ปัญหานี้โดยจะไม่อนุญาตให้มีการเปลี่ยนแปลงอินเตอร์เฟสของคอมโพเนนต์การปรับเปลี่ยนหรือเพิ่มความสามารถให้กับคอมโพเนนต์จะเป็นในลักษณะแก้ไขการทำงานของอินเตอร์เฟส หรือ การเพิ่มอินเตอร์เฟสใหม่ COM อนุญาตให้คอมโพเนนต์สามารถมีหลายอินเตอร์เฟสได้ (Multiple Interface)

โครงสร้างของอินเตอร์เฟส

COM ได้กำหนดอินเตอร์เฟสพื้นฐานขึ้นมาเพื่อให้ไคลเอนต์ที่สนับสนุนอินเตอร์เฟสพื้นฐานนี้สามารถใช้งานอินเตอร์เฟสในออบเจกต์ได้ ดังรูปที่ 3-3



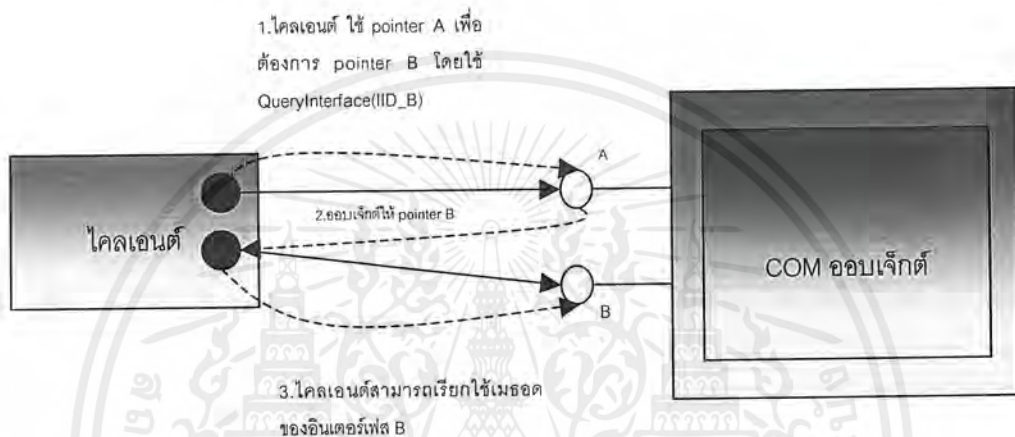
รูปที่ 3-3 โครงสร้างของอินเตอร์เฟสในออบเจกต์

VTable จะเริ่มต้นด้วย พอยน์เตอร์ ที่ชี้ไปยังเมธอดของ IUnknown ดังรูปที่ 3-3 เพื่อการสร้างและจัดการออบเจกต์ที่จะสร้างขึ้นมามีการใช้งาน ทุกออบเจกต์ต้องมีโครงสร้างดังรูปที่ 3-3 เสมอ

อินเตอร์เฟส IUnknown

COM ออบเจกต์ทุกตัวต้องสนับสนุนการทำงานของ IUnknown และอินเตอร์เฟสต้องสืบทอดมาจาก Iunknown เพราะฉะนั้นอินเตอร์เฟสทุกตัวที่ไปยังออบเจกต์สามารถเรียกใช้เมธอดที่เป็นของ IUnknown ได้ เมธอดของ IUnknown มี 3 เมธอดได้แก่

- **QueryInterface:** เป็นเมธอดที่ใช้เพื่อให้ไคลเอนต์ถาม COM ออบเจกต์ว่ามีอินเตอร์เฟสที่ต้องการหรือไม่ เพราะในตอนเริ่มต้นสร้างออบเจกต์ขึ้นมา ตัวไคลเอนต์จะมีเพียง พอยน์เตอร์ ของอินเตอร์เฟสมาตรฐาน คือ IUnknown เท่านั้นการใช้งานจริงไคลเอนต์ต้องการ พอยน์เตอร์ของอินเตอร์เฟสอื่นที่มีเมธอดที่ต้องการ ไคลเอนต์จะถามไปยัง COM ออบเจกต์โดยเมธอด QueryInterface ดังรูปที่ 3-4



รูปที่ 3-4 การใช้งาน QueryInterface

เรานำ QueryInterface ไปแก้ปัญหาเวอร์ชันของซอฟต์แวร์ที่ได้อีกว่าไปในตอนต้นได้โดยสามารถเพิ่มอินเตอร์เฟสใหม่ ๆ เข้าไปในคอมโพเนนต์โดยที่ไคลเอนต์ที่เรียกใช้อินเตอร์เฟสเดิมไม่ได้รับผลกระทบเพราะการ QueryInterface นั้นยังต้องการเพียง พอยน์เตอร์ ตัวเดิมเท่านั้น ส่วนไคลเอนต์ที่ต้องการใช้ฟังก์ชันใหม่ก็ QueryInterface อินเตอร์เฟสที่ต้องการได้

- **AddRef and Release:** เป็นเม็ททอดที่ควบคุมการสร้างและทำลายออบเจกต์ที่ถูกสร้างขึ้นมาจาก COM คลาส) ออบเจกต์ที่สร้างขึ้นมามีค่าต้องรู้ว่าเมื่อใดสมควรจะทำลายเพื่อประหยัดทรัพยากรของระบบ ใน COM ออบเจกต์ที่ถูกสร้างขึ้นมาจะเก็บสถานะการถูกใช้งานของตนเองไว้ใน Reference Counting ซึ่งบอกว่ามีไคลเอนต์เป็นจำนวนเท่าใดที่ยังใช้งานมันอยู่ เมื่อไม่มีไคลเอนต์ใด ๆ ใช้งานค่า Reference Counting จะเท่ากับศูนย์ระบบจะรู้ทันทีว่าออบเจกต์นี้สามารถทำลายหรือรีไซเคิลได้ เมธอด AddRef เป็นตัวเพิ่มค่า Reference Counting เมื่อมีไคลเอนต์เรียกใช้งานออบเจกต์ และ Release จะทำหน้าที่ลดค่า Reference Counting เมื่อไคลเอนต์เลิกใช้งานออบเจกต์

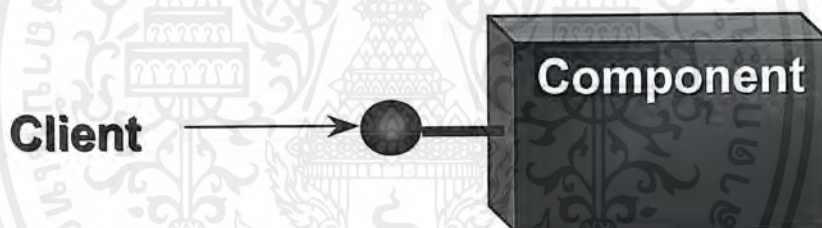
3.1.1.2 คลาส (Classes)

COM ออบเจ็กต์ทุกตัวถูกสร้างมาจาก COM คลาสซึ่งเป็นที่ระบุถึงโครงสร้างการทำงานภายในของกลุ่มอินเตอร์เฟซ แต่ละคลาสจะมีหมายเลขระบุถึงคลาสเรียกว่า *CLSID (Class Identifier)* CLSID เป็นตัวระบุว่าส่วนไหนของโค้ดของ ไลบรารี COM ที่จะถูกโหลดขึ้นมาเพื่อทำการสร้างออบเจ็กต์และ CLSID ต้องมีค่าไม่ซ้ำกัน

3.1.1.3 ประเภทของเซิร์ฟเวอร์สำหรับ COM ออบเจ็กต์

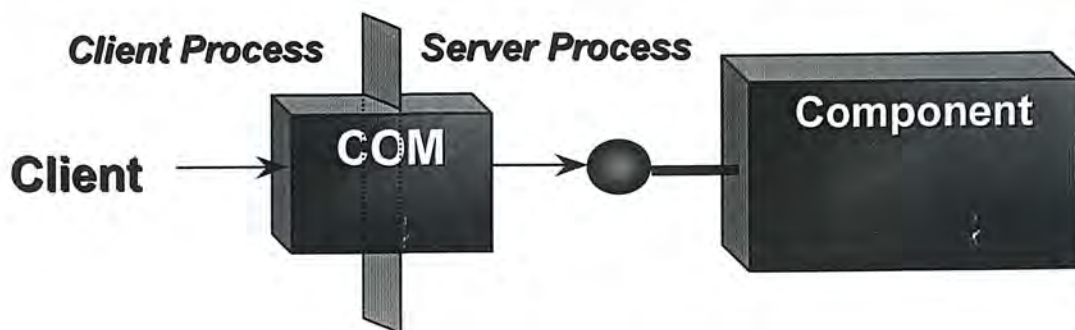
COM ออบเจ็กต์ทุกตัวต้องถูกสร้าง (Implement) ในเซิร์ฟเวอร์ซึ่งเป็นที่เก็บโค้ดสำหรับแต่ละเมธอดของอินเตอร์เฟซของออบเจ็กต์และเก็บข้อมูลหรือสถานะของออบเจ็กต์ในขณะที่ออบเจ็กต์กำลังทำงาน เซิร์ฟเวอร์สามารถรองรับได้มากกว่าหนึ่งออบเจ็กต์ที่สร้างมาจากคลาสเดียวกันและรองรับได้มากกว่าหนึ่งคลาส โดยเซิร์ฟเวอร์มี 3 ชนิด คือ

- **In-process server** เป็นรูปแบบที่ออบเจ็กต์ถูกสร้างจาก DLL (Dynamic Link Library) และทำงานในโปรเซสเดียวกับไคลเอนต์ ข้อดีของรูปแบบนี้ คือ การทำงานจะเร็วมากเพราะว่าออบเจ็กต์จะถูกสร้างและทำงานในโปรเซสของไคลเอนต์โดยไม่ต้องมีการทำ context switch เพื่อเรียกใช้เซอร์วิสของออบเจ็กต์นั้น ตัวอย่างการนำไปใช้ คือ ActiveX Controls ดังรูปที่ 3-5



รูปที่ 3-5 การทำงานของ In-process เซิร์ฟเวอร์

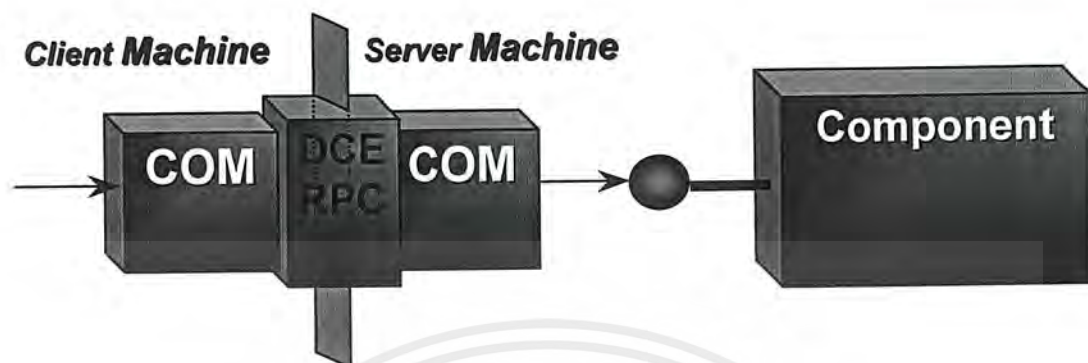
- **Local servers** เป็นรูปแบบที่ออบเจ็กต์ถูกสร้างในโปรเซสที่ทำงานคนละโปรเซสกับไคลเอนต์แต่อยู่บนเครื่องเดียวกับไคลเอนต์ ดังรูปที่ 3-6



รูปที่ 3-6 การทำงานของ Local เซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

• **Remote servers** เป็นรูปแบบที่ออบเจ็กต์ถูกสร้างใน DLL หรือในโปรเซสที่แยกต่างหากแต่ทำงานบนเครื่องที่ไม่ใช่เครื่องที่โคลเอนต์ทำงาน การติดต่อและจัดการด้านการสื่อสารระหว่างเครื่องนั้นจะเป็นหน้าที่ของ DCOM (Distributed COM) ดังรูปที่ 3-7



รูปที่ 3-7 การทำงานของ Remote เซิร์ฟเวอร์

3.1.2 การสร้างวัตถุ COM

จากที่กล่าวไปแล้วว่าโคลเอนต์จะได้พอยน์เตอร์ของอินเตอร์เฟสมาตรฐาน (IUnknown) เมื่อสร้างออบเจ็กต์ขึ้นใช้งาน

3.1.2.1 ไลบรารี COM

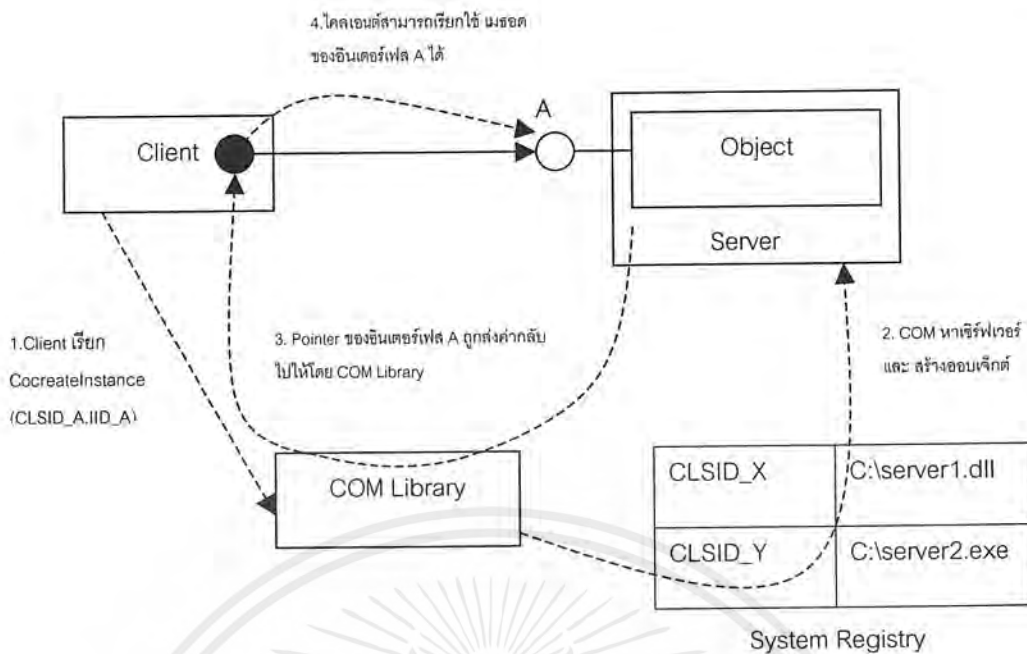
เป็นไลบรารีที่เก็บเซอร์วิสพื้นฐานที่ออบเจ็กต์และโคลเอนต์จำเป็นต้องใช้งาน การค้นหาเซิร์ฟเวอร์

เมื่อโคลเอนต์ต้องการสร้างออบเจ็กต์จะผ่าน CLSID ไปให้ ไลบรารี COM เพื่อหาเซิร์ฟเวอร์สำหรับคลาสของออบเจ็กต์นี้ โดยอาศัยข้อมูลจากฐานข้อมูลรีจิสทรี โดยข้อมูลจะบ่งบอกสิ่งดังต่อไปนี้

- CLSID เปรียบเสมือนเป็นคีย์เพื่อใช้ในการหาข้อมูล
- ชนิดของเซิร์ฟเวอร์สำหรับแต่ละ CLSID
- ถ้าเป็น in-process หรือ โลคัลเซิร์ฟเวอร์จะมีข้อมูลที่บอกถึงที่อยู่ของไฟล์ (ทั้ง DLL และ EXE) ถ้าเป็นรีโมทเซิร์ฟเวอร์จะบอกถึงตำแหน่งของเครื่องที่เก็บเซิร์ฟเวอร์

3.1.2.2 การสร้างออบเจ็กต์หนึ่งออบเจ็กต์ (Single Object Creation)

โคลเอนต์จะเรียกใช้ฟังก์ชัน CoCreateInstance เพื่อสร้างออบเจ็กต์ที่ต้องการขึ้นมาโดยมีข้อมูลที่ต้องผ่านไปให้ฟังก์ชัน คือ CLSID และ IID จากนั้น ไลบรารี COM จะนำข้อมูลที่มีไปหารายละเอียดอื่น ๆ ในรีจิสทรีของระบบ พร้อมทั้งสร้างออบเจ็กต์ที่โคลเอนต์ต้องการถ้าทุกอย่างถูกต้อง หลังจากนั้นจะให้ค่าพอยน์เตอร์ ของอินเตอร์เฟสที่โคลเอนต์ต้องการกลับไป ดังรูปที่ 3-8



รูปที่ 3-8 การสร้างออบเจกต์หนึ่งออบเจกต์

3.1.2.3 การสร้างออบเจกต์หลายออบเจกต์ที่มาจากคลาสเดียวกัน

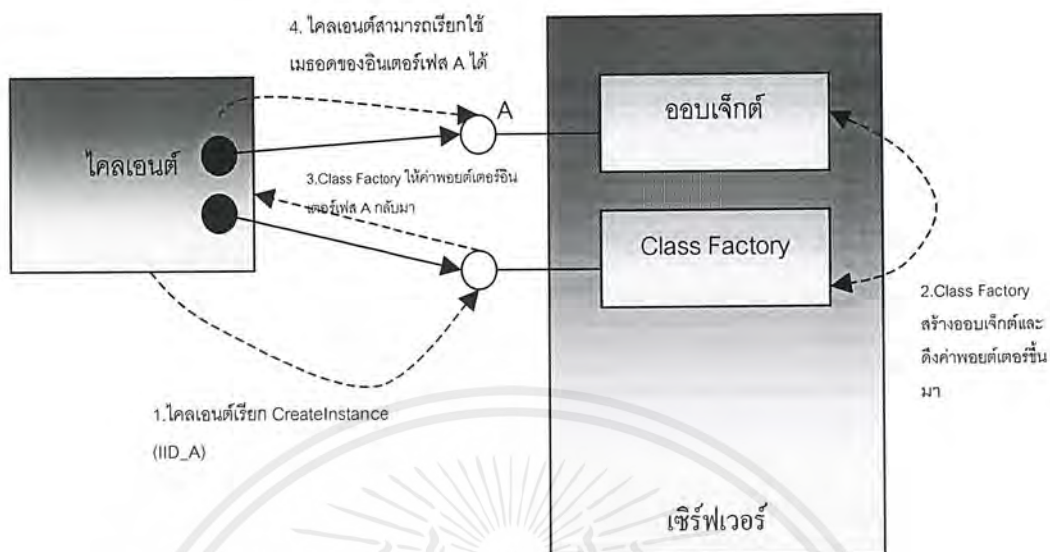
Class Factory คือ ออบเจกต์ที่สามารถสร้างออบเจกต์อื่นขึ้นมาได้ เป็น COM ออบเจกต์ ในความเป็นจริงการสร้างออบเจกต์ เดียวก็มีการเรียกใช้ Class Factory อยู่ภายในอีกทีหนึ่งโดยมีการซ่อนรายละเอียดไว้ เพราะฉะนั้นอาจจะพูดได้ว่าการสร้าง COM ออบเจกต์ ทุกตัวขึ้นมาใช้งานต้องผ่านการเรียกใช้ Class Factory

Class Factory จะสนับสนุนการทำงานโดย IClassFactory ซึ่งมี 2 เมธอดที่สำคัญ คือ

- **CreateInstance** เป็นเมธอดที่สร้างออบเจกต์จากคลาสที่ถูกเก็บรายละเอียดไว้ใน Class Factory โดยไม่ต้องมีการระบุ CLSID เพียงแต่ระบุ IID สำหรับอินเตอร์เฟสที่ต้องการ การที่เราจะนำข้อมูลมาเก็บไว้ใน Factory ได้นั้นต้องเรียกใช้ CoGetClassObject โดยผ่าน CLSID ที่ต้องการก็จะให้ค่าพอยน์เตอร์ที่ต้องการมาเก็บไว้ใน Factory

- **LockServer** เป็นเมธอดที่ทำให้สามารถเก็บเซิร์ฟเวอร์ไว้ในหน่วยความจำได้ Class Factory ก็เหมือนกับ COM ออบเจกต์อื่น ๆ คือ ต้องทำลายตัวเองเมื่อ reference count เท่ากับศูนย์ เพราะฉะนั้นถ้าต้องการเก็บเซิร์ฟเวอร์ไว้เพื่อให้มีการสร้างออบเจกต์ได้เร็ว ก็สามารถใช้ Lock Server ที่ต้องการได้เพื่อไม่ให้ถูกรื้อออกจากหน่วยความจำ

จากหลักการดังกล่าวเราสามารถสร้างออบเจกต์ที่ต้องการหลาย ๆ ออบเจกต์ได้ ดังรูปที่ 3-9

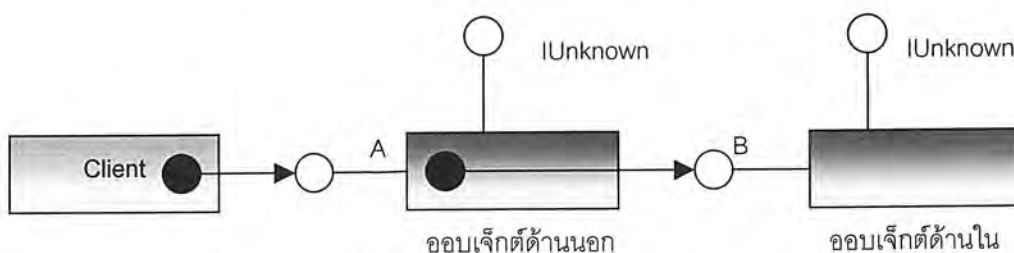


รูปที่ 3-9 การสร้างออบเจกต์หลายๆ ตัวจากคลาสเดียวกัน

3.1.3 การนำกลับมาใช้ใหม่ใน COM ออบเจกต์ (Reusing COM Objects)

COM ออบเจกต์ไม่มีความสามารถการนำกลับมาใช้ใหม่ในระดับโค้ด เพราะว่าในระบบงานจริงๆ นั้นการที่นำโค้ดมาใช้ใหม่นั้นไม่สามารถทำได้ในสภาวะแวดล้อมที่ออบเจกต์มีความแตกต่างกันโดยสิ้นเชิง นอกจากนี้การแก้ไขที่ตัวบรรพบุรุษก็อาจจะทำให้ลูกหลานที่สืบทอดโค้ดมานั้นเกิดปัญหาในการทำงาน ปัญหาในลักษณะนี้เกิดจากแต่ละออบเจกต์มีความสัมพันธ์ในลักษณะ Tightly Coupling และเมื่อมีการสืบทอดโค้ดมากยิ่งขึ้นระดับของความผูกพันก็จะยิ่งมากขึ้นไปอีก เพราะฉะนั้นในรูปแบบของ COM จะเกิดการนำกลับมาใช้ใหม่ในลักษณะของ Containment และ Aggregation ซึ่งเทคนิคทั้งสองนั้นมีกลไกเหมือนกัน คือ ออบเจกต์ด้านนอกจะนำออบเจกต์ภายในกลับมาใช้ใหม่

● **Containment หรือ Delegation** เป็นรูปแบบที่ออบเจกต์ด้านนอกจะเปรียบเสมือนเป็นไคลเอนต์ของออบเจกต์ภายใน ดังรูปที่ 3-10

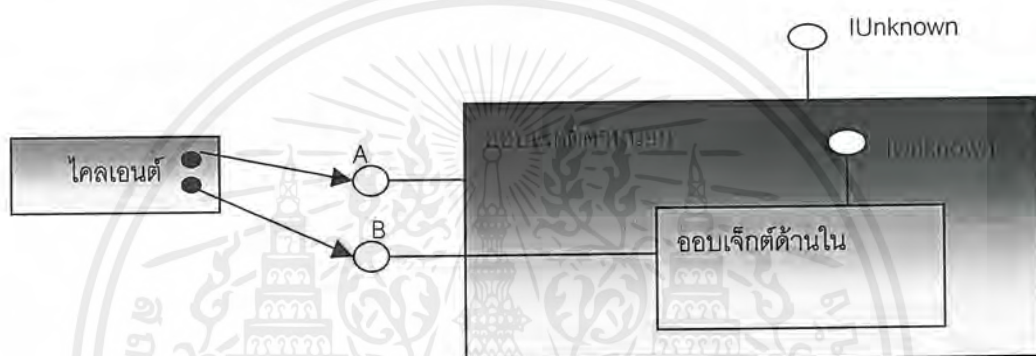


รูปที่ 3-10 การนำกลับมาใช้ใหม่ในลักษณะ Containment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Containment เป็นรูปแบบที่ไม่ซับซ้อนเหมือนเป็นการเรียกใช้เซอร์วิสธรรมดา แต่ทำให้เราไม่ต้องเขียนโค้ดในเซอร์วิสที่มีการใช้ในหลาย ๆ ออบเจ็กต์โดยนำออกมาสร้างเป็น COM ออบเจ็กต์แยกต่างหาก แต่รูปแบบนี้เนื่องจากมีความง่ายก็ย่อมต้องมีข้อจำกัด คือ ประสิทธิภาพในการทำงานของซอฟต์แวร์จะลดลงเนื่องจากแทนที่จะสามารถเรียกใช้เซอร์วิสได้โดยตรงก็ต้องไปผ่านออบเจ็กต์อื่น ๆ อีก ทำให้เกิดโอเวอร์เฮดในการทำงาน

• **Aggregation** เป็นรูปแบบที่ออบเจ็กต์ด้านนอกแสดงอินเตอร์เฟซของออบเจ็กต์ด้านในเหมือนกับว่าเป็นอินเตอร์เฟซของตัวเอง เวลาที่ไคลเอนต์ต้องการอินเตอร์เฟซของออบเจ็กต์ภายในนั้น ออบเจ็กต์ภายนอกจะให้ค่าพอยต์เตอร์ของอินเตอร์เฟซของออบเจ็กต์ด้านในแก่ไคลเอนต์ หลังจากนั้นไคลเอนต์ก็สามารถเรียกใช้เซอร์วิสต่าง ๆ ของออบเจ็กต์ด้านในได้ ดังรูปที่ 3-11



รูปที่ 3-11 การนำกลับมาใช้ใหม่แบบ Aggregation

จากที่กล่าวมาในเรื่อง COM (Component Object Model) เราสามารถนำหลักการไปสร้างซอฟต์แวร์ที่มีคุณภาพได้ นอกจากจะตรงความต้องการแล้วซอฟต์แวร์ที่ได้ยังมีประสิทธิภาพสูงด้วย

3.1.4 ประโยชน์ของวัตถุ COM

• วัตถุ COM สามารถให้ความสามารถทางออบเจ็กต์โอเรียนเท็ดได้ โดยสามารถออกแบบซอฟต์แวร์เป็นส่วน ๆ ได้ตามหน้าที่ของต่าง ๆ ของซอฟต์แวร์และมีความสามารถทางออบเจ็กต์โอเรียนเท็ดทุกประการ

• วัตถุ COM จะมีความแน่นอนในการเรียกใช้ไม่ว่าตัวไลบรารีโคดจะอยู่ในรูปแบบใดก็ตามเช่น อยู่ต่างเครื่องกันผู้พัฒนาซอฟต์แวร์ก็เรียกใช้ตามปกติเสมือนหนึ่งว่าอยู่บนเครื่องเดียวกัน

• วัตถุ COM จะไม่ขึ้นกับภาษาที่พัฒนาโดยอาจจะพัฒนาและถูกเรียกใช้โดยภาษาแตกต่างกันก็ได้

3.1.4.1 ปัญหาด้านเวอร์ชันของซอฟต์แวร์ (Versioning)

COM ที่ถูกพัฒนาโดยมีความสามารถเพิ่มเติมจากเวอร์ชันเดิมสามารถนำมาแทนที่ซอฟต์แวร์เดิมได้โดยไม่กระทบต่อซอฟต์แวร์ตัวอื่นที่เรียกใช้บริการจากซอฟต์แวร์ตัวเดิม โดยอาศัยหลักการของการมีเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้หลายอินเตอร์เฟซ (Multiple Interface) ซอฟต์แวร์ตัวเดิมก็เรียกใช้อินเตอร์เฟซตัวเดิม ซอฟต์แวร์ตัวใหม่ก็เรียกใช้ความสามารถใหม่ผ่านอินเตอร์เฟซตัวใหม่ โดยไม่มีผลกระทบซึ่งกันและกัน

3.2 กรณีศึกษาการพัฒนาซอฟต์แวร์ COM และ DCOM ใน Visual Basic และ Delphi เพื่อพิสูจน์คุณสมบัติต่าง ๆ ของเทคโนโลยี COM และ DCOM

3.2.1 การพัฒนาซอฟต์แวร์ COM

จุดประสงค์

เพื่อทดสอบคุณสมบัติต่าง ๆ ของเทคโนโลยี COM คือ การไม่ขึ้นกับเครื่องมือที่พัฒนา การแก้ปัญหาด้านเวอร์ชันของซอฟต์แวร์

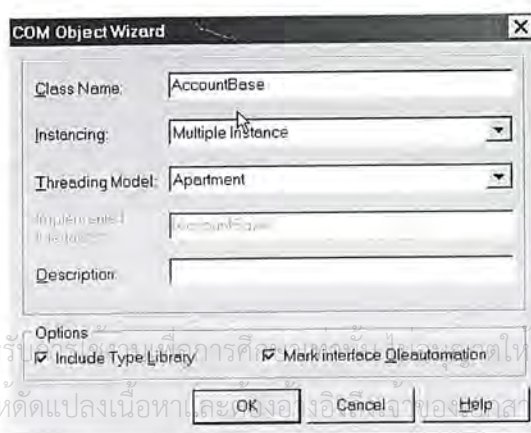
เครื่องมือที่ใช้พัฒนา

1. Visual Basic 6.0 Enterprise Edition
2. Borland Delphi 5.0(Built5.62) Enterprise

คำอธิบายโปรแกรม

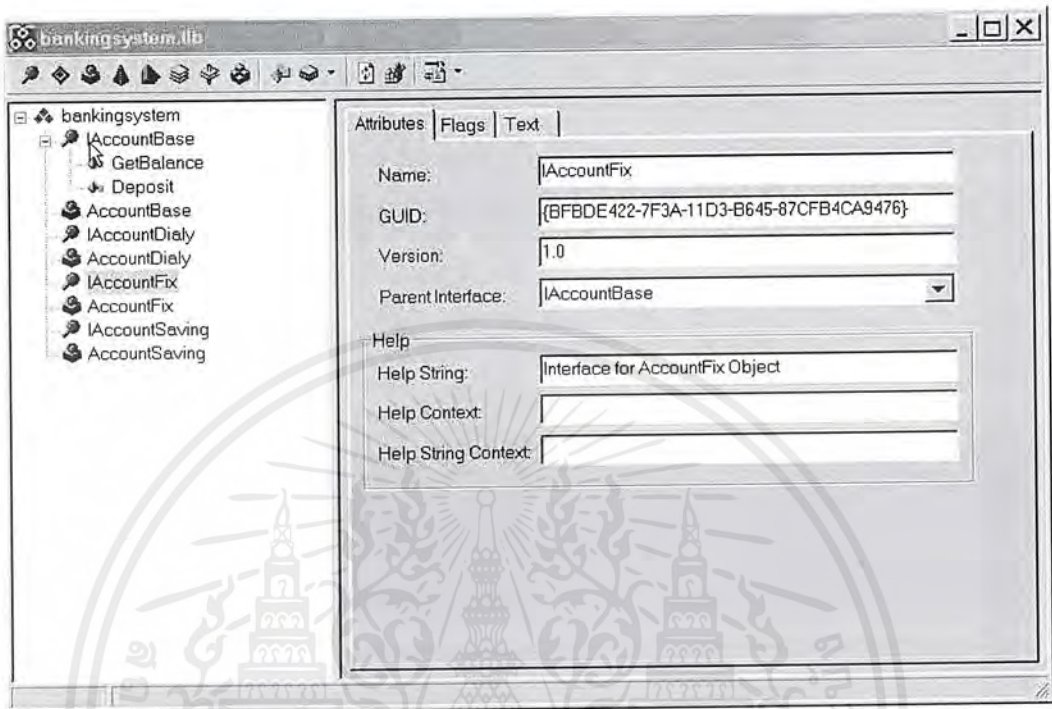
1. เป็นโปรแกรมระบบบัญชีพื้นฐานของธนาคารที่สามารถสอบถามยอดคงเหลือได้ ผากได้และถอนได้ โดยโปรแกรมจะเน้นในด้านการวางโครงสร้างโปรแกรมเพราะฉะนั้นฟังก์ชันการทำงานจึงไม่สามารถทำงานได้จริงเพียงแค่แสดงการตอบรับการเรียกใช้เท่านั้น
 2. ภายในเมธอดของแต่ละอินเตอร์เฟซจะมีการแสดงการโต้ตอบว่าได้ใช้เมธอดของอินเตอร์เฟซใดเพื่อทดสอบคุณสมบัติต่าง ๆ ของ COM
- ขั้นตอนการพัฒนา

1. ออกแบบอินเตอร์เฟซที่จำเป็นในที่นี้จะประกอบไปด้วย
 - IAccountBase เป็นอินเตอร์เฟซเกี่ยวกับบัญชีพื้นฐาน โดยจะมีเมธอดที่เกี่ยวข้องกับการทำงานของบัญชีพื้นฐาน เช่น การสอบถามยอดเงิน,การฝากเงิน และการถอนเงิน เป็นต้น
 - IAccountDialy เป็นอินเตอร์เฟซของบัญชีเงินฝากกระแสรายวัน
 - IAccountFix เป็นอินเตอร์เฟซของบัญชีเงินฝากประจำ
 - IAccountSaving เป็นอินเตอร์เฟซของบัญชีฝากออมทรัพย์
2. สร้างโปรเจกต์ ActiveX Library ใหม่ใน Delphi
3. สร้างวัตถุ COM ใหม่แล้วตั้งค่าที่เหมาะสมดังรูปที่ 3-12 สำหรับอินเตอร์เฟซทุกตัวที่จำเป็นตามข้อหนึ่ง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่ควรนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และแจ้งข้อผิดพลาดของเอกสารทุกครั้งที่มีการนำไปใช้

4. Delphi จะ generate Type Library เราต้องการทำการเพิ่มเมธอดที่จำเป็นแก่อินเตอร์เฟส เช่น GetBalance และ Deposit เป็นต้น ตลอดจนการกำหนดพารามิเตอร์ให้แก่เมธอดที่เราสร้างด้วย เช่น หมายเลขบัญชี จะได้โครงสร้างของคอมโพเนนต์ ดังรูปที่ 3-13



รูปที่ 3-13 Type Library ที่เรากำหนดค่าที่ต้องการเรียบร้อย

จากรูป 3-13 จะพบว่าถ้ามีการสืบทอดระหว่างอินเตอร์เฟส เราสามารถตั้งค่าได้ที่ Parent Interface ในที่นี้บัญชีทุกประเภทจะสืบทอดอินเตอร์เฟสมาจากอินเตอร์เฟส IAccountBase

5. เมื่อ Refresh โปรแกรม Delphi จะ generate โครงร่างของคอมโพเนนต์มาให้เราสามารถเขียนส่วนการทำงานของแต่ละเมธอดของอินเตอร์เฟส ตามที่ต้องการได้

6. ท้ายสุดเราต้องรีจิสเตอร์คอมโพเนนต์ที่เราเขียนขึ้นมาโดยเลือกที่เมนู Run → Register ActiveX Server

7. พัฒนาโปรแกรมฝั่งไคลเอนต์เพื่อทดสอบคอมโพเนนต์ที่เราทำขึ้นมา โดยต้องมีการอ้างอิงไฟล์ Type Library ที่ได้จาก Type Library Editor ดังรูปที่ 3-14

```
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Spin, Bankingsystem_TLB; //For Access By Type Library ←
type
```

รูปที่ 3-14 การเรียกใช้ Type Library

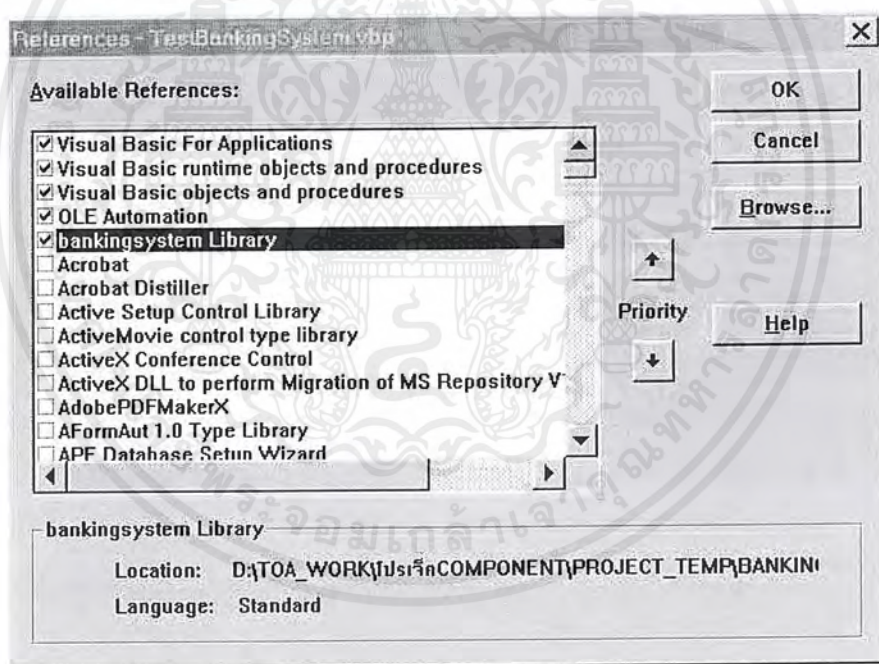
8. โปรแกรมฝั่งไคลเอนต์ทำการทดสอบจะให้ผล ดังรูปที่ 15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-15 โปรแกรมไคลเอนต์ที่ใช้ในการทดสอบที่เขียนด้วย Delphi

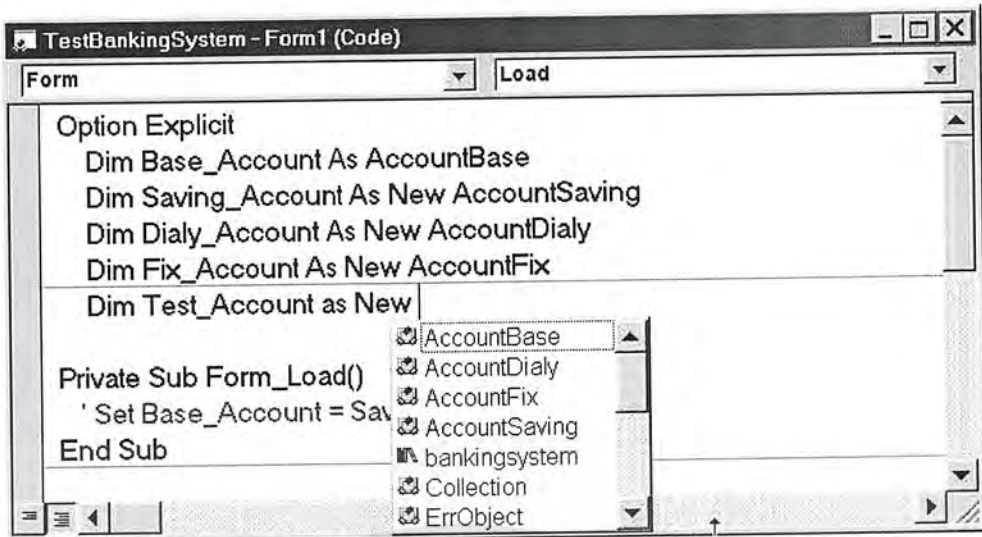
9. ทดลองพัฒนาโปรแกรมลักษณะเดียวกันในทูล Visual Basic โดยต้องมีการตั้ง Reference ก่อน ดังรูปที่ 3-16



รูปที่ 3-16 การอ้างอิงคอมโพเนนต์ใน Visual Basic

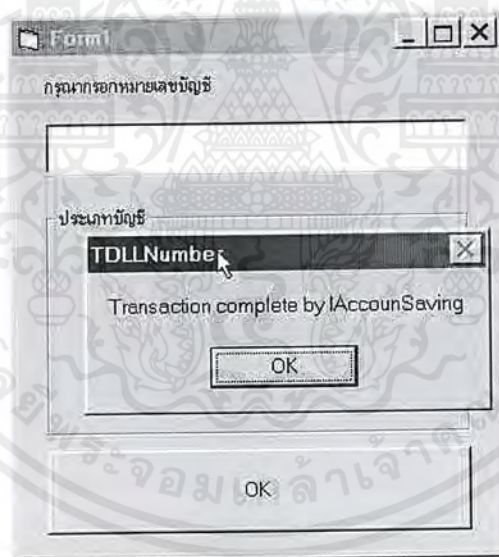
10. ทำการเขียนโปรแกรมฝั่งไคลเอนต์โดยในระหว่างเขียนนั้นทูล Visual Basic ได้นำ Type Library มาช่วยให้เราสามารถเขียนโปรแกรมกับคอมโพเนนต์ได้สะดวก ดังรูปที่ 3-17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-17 การนำเอา *Type Library* มาใช้ช่วยในการเขียนโปรแกรมฝั่งไคลเอนต์

11. เมื่อทำการทดสอบ โปรแกรมไคลเอนต์ที่เขียนด้วย Visual Basic สามารถทำงานได้ ดังรูปที่ 3-18



รูปที่ 3-18 โปรแกรมไคลเอนต์ที่ใช้ในการทดสอบที่เขียนด้วย Visual Basic

ผลการทดสอบ

1. การพัฒนาซอฟต์แวร์ในรูปแบบ COM มีคุณสมบัติ “ไม่ขึ้นกับเครื่องมือที่พัฒนา”
2. สามารถแก้ปัญหาด้านเวอร์ชันของซอฟต์แวร์ได้โดยเพิ่มอินเตอร์เฟสเข้าไปใน Type Library แล้วเพิ่มโค้ดโปรแกรมในส่วนของอินเตอร์เฟสที่เพิ่มเข้าไปใหม่
3. คุณสมบัติต่าง ๆ ของ COM ในด้านออบเจกต์โอเรียนเท็ดนั้น COM สามารถทำ
 - อินเทอร์เฟซด้านอินเตอร์เฟสได้แต่อินเทอร์เฟซด้านโค้ดไม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โพลิมอร์ฟิซึมสามารถทำได้โดย COM ออบเจกต์ที่มีอินเตอร์เฟซที่มีลูกหลานสืบทอดสามารถเก็บหรืออ้างถึงออบเจกต์ที่มีอินเตอร์เฟซลูกหลานได้

- เอนแคปซูเลชันใน COM ออบเจกต์มีลักษณะซ่อนรายละเอียดของออบเจกต์โดยการติดต่อระหว่างออบเจกต์นั้นสามารถทำได้โดยผ่านอินเตอร์เฟซเท่านั้น

การพัฒนาวัตถุ COM จาก Microsoft Visual Basic 6.0

- ทำการออกแบบด้วยวิธีการทางออบเจกต์โอเรียนเท็ด (อาจอ้างอิงจากภาษา UML)
- เริ่มโปรเจกต์ใน Visual Basic เป็น ActiveX DLL หรือ ActiveX EXE ขึ้นอยู่กับประเภทของ

เซิร์ฟเวอร์ที่เก็บวัตถุ COM

- นำคลาสโคดอะแกรมที่ได้มาสร้างคลาสใน Visual Basic โดยใช้เครื่องมือที่ชื่อว่า Class

Builder

- หลังจากนั้นก็ทำการเขียนโปรแกรมในส่วนที่เหลือคือในส่วนเมธอดหรือฟังก์ชันที่จำเป็นต่าง ๆ จะได้เป็นวัตถุ COM ที่ต้องการ

3.2.2.1 การพัฒนา โดย MTS (Microsoft Transaction Server)

- ทำการออกแบบด้วยวิธีการทางออบเจกต์โอเรียนเท็ด (อาจอ้างอิงจากภาษา UML)
- เริ่มโปรเจกต์ใน Visual Basic เป็น ActiveX DLL เท่านั้น
- นำคลาสโคดอะแกรมที่ได้มาสร้างคลาสใน Visual Basic โดยใช้เครื่องมือที่ชื่อว่า Class

Builder

หลังจากนั้นก็ทำการเขียนโปรแกรมในส่วนที่เหลือคือในส่วนเมธอดหรือฟังก์ชันที่จำเป็นต่าง ๆ จะได้เป็นวัตถุ COM ที่ต้องการ

- นำคอมโพเนนต์ที่คอมไพล์แล้ว (ได้เป็นไฟล์ DLL) มาริจิสเตอร์ใน MTS ของเครื่องที่เป็นเซิร์ฟเวอร์ที่ให้บริการในการเรียกใช้คอมโพเนนต์ซึ่งขั้นตอนโดยละเอียดจะกล่าวในภาคผนวก จ

บทที่ 4

เทคโนโลยี ActiveX

(ActiveX Technology)

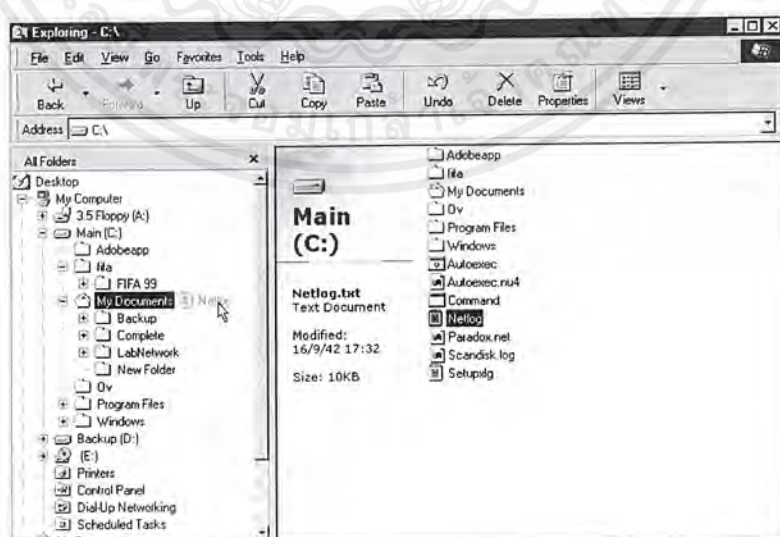
เมื่อแอปพลิเคชันถูกสร้างขึ้นมาจากต่างผู้ผลิต การใช้งานร่วมกันนั้นเป็นไปได้ค่อนข้างลำบาก เพราะต่างคนต่างก็กำหนดรูปแบบหรือมาตรฐานการทำงานที่ไม่เหมือนกันและทำให้ผู้ใช้พบความยุ่งยากในการใช้งานแอปพลิเคชันร่วมกัน

เพื่อแก้ไขปัญหาดังกล่าวทางไมโครซอฟต์ได้พยายามสร้างวิธีการที่จะทำให้แอปพลิเคชันต่าง ๆ ใช้งานร่วมกันได้อย่างราบรื่น โดยเริ่มจากการพัฒนาเทคโนโลยี DDE (Dynamic Data Exchange) และ OLE 1.0 (Object Linked and Embedded 1.0) ต่อมาก็พัฒนาเป็น OLE 2.0 และ COM (Component Object Model) โดยรายละเอียดของเทคโนโลยีเหล่านี้ได้กล่าวไว้ในส่วนของ OLE and COM ในบทที่ 3

ต่อมาเมื่ออินเทอร์เน็ตได้เข้ามามีบทบาทในการทำงานมีผลให้รูปแบบของข้อมูลที่มีการใช้งานกับคอมพิวเตอร์มีเพิ่มขึ้นมากมายทำให้ต้องมีวิธีการมาตรฐานที่แอปพลิเคชันที่ใช้งานได้ในปัจจุบันให้สามารถใช้งานกับอินเทอร์เน็ตได้ด้วยและนั่นเป็นต้นกำเนิดของ ActiveX

4.1 การทำงานแบบ Drag and Drop

สำหรับการทำงานแบบ Drag and Drop นั้นมีให้ใช้งานกันอย่างแพร่หลาย โดยการคลิกเมาส์เลือกสิ่งที่ต้องการ กดเมาส์แล้วลากไปยังปลายทาง จากนั้นปล่อยเมาส์ เท่านั้นก็เป็นการทำงานโดยใช้ Drag and Drop ดังรูป 4-1



รูปที่ 4-1 การทำงานแบบ Drag and Drop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อมาได้มีการพัฒนาเป็น OLE Drag and Drop ซึ่งมีคุณสมบัติเหนือกว่า Drag and Drop ธรรมดา โดยแทนที่จะ drag ออบเจกต์หนึ่งไปยังอีกออบเจกต์หนึ่ง ก็เป็นการย้ายข้อมูลจากคอนโทรลหรือแอปพลิเคชันไปยังคอนโทรลหรือแอปพลิเคชันอื่น ๆ เช่น ทำการ drag text file จาก Windows Explorer ไปยัง Notepad ที่เปิดอยู่ เป็นต้น โดยในที่นี่จะแสดงตัวอย่างการทำงานโดยใช้ Visual Basic 6.0

4.1.1 OLE Drag and Drop ทำงานได้อย่างไร

อีเวนต์ทั้งหมดที่เกิดขึ้นระหว่างแอปพลิเคชันแบ่งได้เป็นอีเวนต์ที่เกิดขึ้นกับ Source Control ซึ่งเกิดขึ้นกับทั้ง Automatic และ Manual Processing ส่วนอีเวนต์ที่เกิดขึ้นกับ Target Control จะเกิดขึ้นกับ Manual Processing เท่านั้น ดังตารางที่ 4-1

Source	Target
OleStartDrag	OleDragDrop
OleSetData	OleDragOver
OleGiveFeedback	OleCompleteDrag

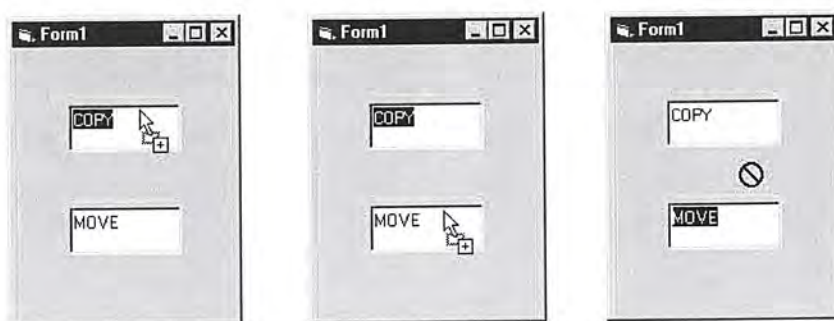
ตารางที่ 4-1 แสดงอีเวนต์ที่เกิดขึ้นของ Source และ Target Control

โดยการทำงานของ OLE Drag and Drop จะมีอยู่ 2 รูปแบบ คือ

- **Automatic OLE Drag and Drop** ทุกโอเปอเรชันจะถูกควบคุมโดย Windows และ Visual Basic Process ภายใน เราสามารถทำการ drag text จาก TextBox ไปยัง TextBox อีกอันหนึ่งได้โดยทำการกำหนดค่าของคุณสมบัติ OLEDragMode และ OLEDropMode เป็น Automatic โดยไม่ต้องทำการเขียนโค้ดเพิ่มเติมอย่างไร ซึ่งวิธีการนี้เป็นวิธีที่ง่ายและสะดวกมากที่สุด

- **Manual OLE Drag and Drop** เราจะต้องทำการจัดการเกี่ยวกับอีเวนต์ที่เกิดขึ้นเองโดยจะเหมาะสำหรับผู้ใช้ที่ต้องการควบคุมการทำงานของแต่ละขั้นตอนใน process หรือต้องการจัดการการตอบสนองต่อผู้ใช้ เช่น ถ้าต้องการเปลี่ยนรูปร่างของ mouse cursor หรือทำฟังก์ชันที่แตกต่างกันขึ้นอยู่กับปุ่มที่กดหรือตรวจสอบว่าปุ่ม shift ถูกกดหรือไม่ เป็นต้น

โดยการทำงานของ OLE Drag and Drop จะเริ่มเมื่อผู้ใช้ทำการ drag ข้อมูลโดยกดเมาส์ปุ่มซ้ายค้างไว้จะเกิดอีเวนต์ OleStartDrag เมื่อ drag ข้อมูลไปอยู่เหนือ Target Control จะเกิดอีเวนต์ OLEDragOver เราสามารถระบุว่าจะให้ Target Control ทำอย่างไรถ้าข้อมูลถูก drop โดยมีทางเลือก 3 ทาง คือ copy, move หรือ refuse ข้อมูล หลังจากการ drop ถูกกำหนดแล้วจะเกิดอีเวนต์ OLEGiveFeedback ซึ่งจะถูกใช้ในการกำหนด feedback action ที่จะทำเมื่อผู้ใช้ทำการ drop ข้อมูล ดังรูป 4-2

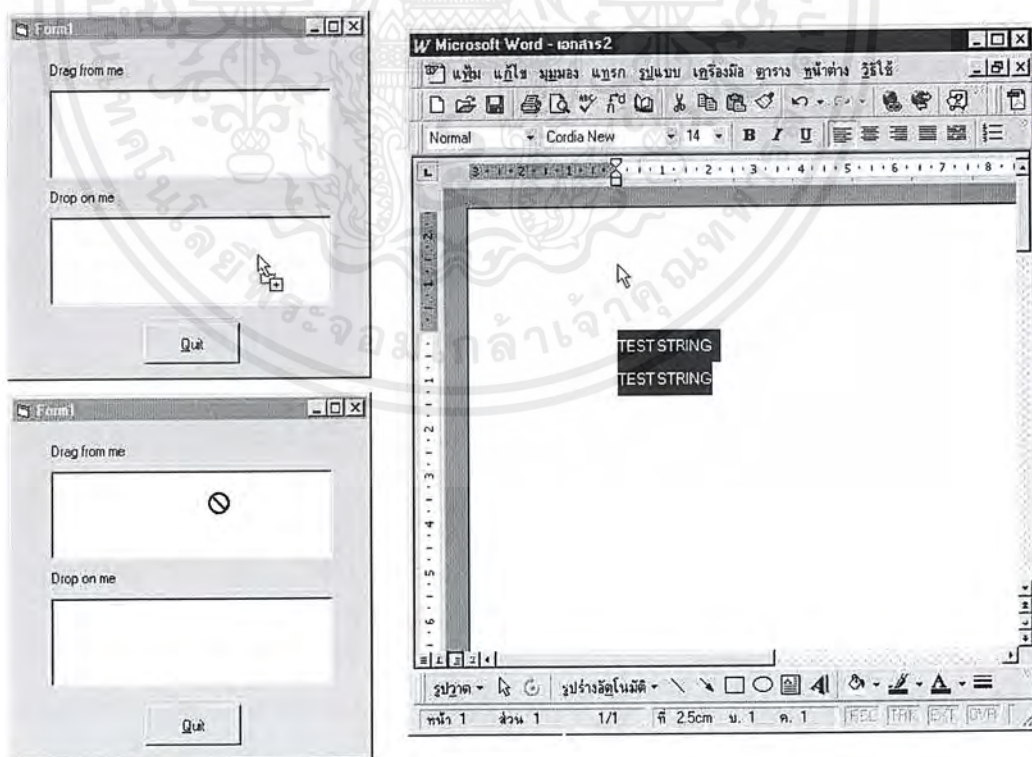


รูปที่ 4-2 Default Mouse Icon สำหรับ OLE Drag and Drop

หลังจากทำการ drop ข้อมูลลง Target Control แล้วจะเกิดอีเวนต์ OLEDragDrop โดย Target Control จะตรวจสอบว่าชนิดของข้อมูลว่าเหมาะกับชนิดของคอนโทรลหรือไม่

ถ้าข้อมูลถูกรรจลงในคอนโทรลเมื่อการ drag เริ่มต้น เมทอด GetData จะดึงข้อมูลเอาไว้ แต่ถ้าข้อมูลไม่ถูกรรจลงคอนโทรลเมื่อการ drag เริ่มต้นจะเกิดอีเวนต์ OLESetData เมทอด SetData จะดึงข้อมูลเอาไว้ เมื่อข้อมูลถูกรับหรือปฏิเสธจะเกิดอีเวนต์ OLECompleteDrag และจะมีการลบข้อมูลของ source ที่จำเป็น ถ้าข้อมูลถูกรับและข้อมูลถูกย้ายแล้ว source จะต้องลบข้อมูลทิ้ง

ในที่นี้จะแสดงเฉพาะตัวอย่างการประยุกต์ใช้งานของ Automatic OLE Drag and Drop โดยเพียงแต่กำหนดค่าของคุณสมบัติ OLEDragMode และ OLEDropMode เป็น Automatic ดังรูป 4-3



รูปที่ 4-3 ตัวอย่างการประยุกต์ใช้ Automatic OLE Drag and Drop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการใช้งานจะมี ActiveX รูปแบบต่าง ๆ มากมายให้ใช้งานดังนี้

- **ActiveX Control** เป็นออบเจ็กต์ที่เรานำมาประกอบกันเป็นแอปพลิเคชันใน Development Tool ต่าง ๆ และนำมาใช้งานกับแอปพลิเคชันอินเทอร์เน็ต
- **ActiveX Document** เป็นรูปแบบการนำเสนอเอกสารผ่านบราวเซอร์โดยเน้นให้ผู้ที่คุ้นเคยกับ Development Tool ต่าง ๆ สามารถสร้างเอกสารที่ใช้งานผ่านบราวเซอร์ได้อย่างไม่ยาก
- **ActiveX Server** เป็นแอปพลิเคชันที่คอยให้บริการในการทำงาน แอปพลิเคชันอื่น ๆ

4.2 ActiveX Control

แม้ว่า Development Tool ต่าง ๆ ได้เตรียม ActiveX Control ไว้ให้ใช้งานมากมาย แต่บางครั้งความต้องการในการทำงานสร้างแอปพลิเคชันมีความละเอียดอ่อนเกินกว่าบริษัทไมโครซอฟต์จะตอบสนองเราได้ทั้งหมด ทำให้มีบริษัทซอฟต์แวร์ที่สร้าง ActiveX Control เกิดขึ้นมากมาย แต่ละแห่งก็จะสร้าง ActiveX Control ที่มีความสามารถเฉพาะด้านที่โดดเด่น เช่น งานด้านฐานข้อมูล การติดต่อฮาร์ดแวร์หรืออินเทอร์เน็ต เป็นต้น

4.2.1 รูปแบบการสร้าง ActiveX Control

ActiveX Control นั้นส่วนใหญ่จะเก็บในไฟล์ .DLL หรือ .OCX ซึ่งสามารถสร้างได้จากภาษาโปรแกรมหลาย ๆ ภาษา เช่น C/C++, Object Pascal (Delphi), Visual Basic และอื่น ๆ โดยมีรูปแบบการสร้าง ActiveX Control 3 รูปแบบ คือ

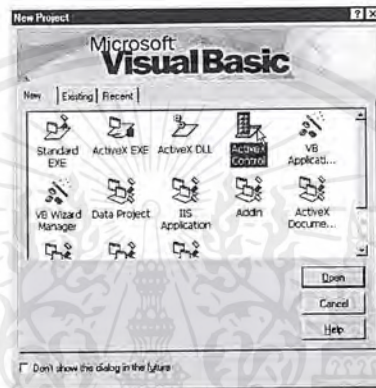
- **พัฒนาต่อจาก ActiveX Control** ตัวเดิม เป็นการนำเอา ActiveX Control ที่มีอยู่อย่างมากมายมาขยายความสามารถให้เพิ่มขึ้น เช่น สร้างให้ ListBox มีความสามารถในการเรียงลำดับทั้งจากมากไปหาน้อยและจากน้อยไปหามาก เป็นต้น
- **พัฒนา ActiveX Control** จากการประกอบกันของ ActiveX Control ตัวเดิม เป็นการสร้าง ActiveX Control จากการนำเอา ActiveX Control ที่มีอยู่แล้วมาประกอบกันเป็น ActiveX Control ตัวใหม่
- **พัฒนา ActiveX Control** ชนิดใหม่ เป็นการสร้าง ActiveX Control ชนิดใหม่ที่ไม่มีเคยมีมาก่อน ซึ่งเรียก ActiveX Control ใหม่นี้ว่า User-drawn Control

สำหรับรูปแบบที่ง่ายที่สุดในการพัฒนา ActiveX Control นั้น คือ แบบที่ 2 เนื่องจากเป็นการรวมกันของ ActiveX Control ที่มีอยู่แล้ว (Constituent control) ซึ่งคอนโทรลเหล่านี้ไม่สามารถสร้างขึ้นได้ด้วยตัวของมันเองและจะต้องถูกเก็บอยู่ในตัวคอนเทนเนอร์ โดยพร็อพเพอร์ตี้ เมททอดและอีเวนต์ของคอนโทรลเหล่านี้จะถูกกำหนดโดยคอนเทนเนอร์ของคอนโทรลเหล่านั้น เช่น ถ้าใช้ ActiveX Control ในแอปพลิเคชัน คอนเทนเนอร์ก็จะเป็น Form แต่ถ้าใช้ ActiveX Control ใน HTML Page คอนเทนเนอร์ก็จะเป็น Page นั่นเอง

ในบางครั้งคอนโทรลอาจต้องการข้อมูลเกี่ยวกับคอนเทนเนอร์ของตัวเอง หรือพเพอร์ตีทั้งหมด เช่น สี background และ foreground จะถูกจัดการโดย Ambient Object ซึ่งบางครั้งก็เรียกว่า Ambient Properties เช่น ในบางครั้งเราต้องการใช้ font ให้เหมือนกันกับของคอนเทนเนอร์ คอนโทรลสามารถทำได้โดยการตรวจสอบว่า font ของ Ambient Object เป็นอะไร จากนั้นก็ทำการกำหนด font ของคอนโทรลให้เหมือนกัน

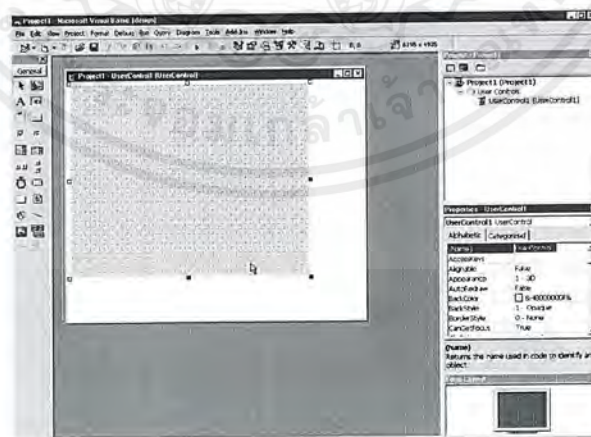
ต่อไปจะเป็นตัวอย่างการสร้าง ActiveX Control ที่ใช้ในการตรวจสอบ password ซึ่งมีขั้นตอนดังนี้

1. เปิดโปรแกรม Visual Basic 6.0 ขึ้นแล้วเลือกเมนู *File > New Project* ดังรูป 4-4



รูปที่ 4-4 หน้าจอเริ่มต้นการสร้าง ActiveX Control

2. Visual Basic จะกำหนด User Control มาให้ซึ่ง User Control นี้จะทำหน้าที่เหมือนกับฟอร์มในการสร้างแอปพลิเคชันธรรมดา ดังรูป 4-5



รูปที่ 4-5 User Control ใน Project ActiveX Control

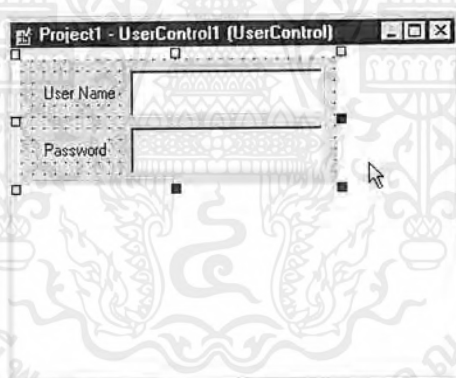
3. หรือพเพอร์ตีส่วนใหญ่ของ UserControl นั้นคล้ายกับของ Form แต่มีหรือพเพอร์ตีอยู่ 2 หรือพเพอร์ตีที่ต่างจาก Form คือ BackStyle ซึ่งจะบอกว่าจะให้มองเห็นพื้นส่วนของ UserControl หรือไม่ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ ToolboxBitmap ไว้กำหนดรูปภาพของ ActiveX Control เมื่อเราจะนำมันไปเพิ่มให้กับ Toolbox โดยเป็นภาพขนาด 16*15 pixel (โดยห้ามเป็นไฟล์ .ICO) ดังรูป 4-6



รูปที่ 4-6 พร็อพเพอร์ตี้ BackStyle และ ToolboxBitmap

4. ทำการแปะคอนโทรลต่าง ๆ ลงบน UserControl ซึ่งประกอบไปด้วย Label และ TextBox และกำหนดค่าพร็อพเพอร์ตี้ต่าง ๆ ของ ActiveX Control ดังรูป 4-7

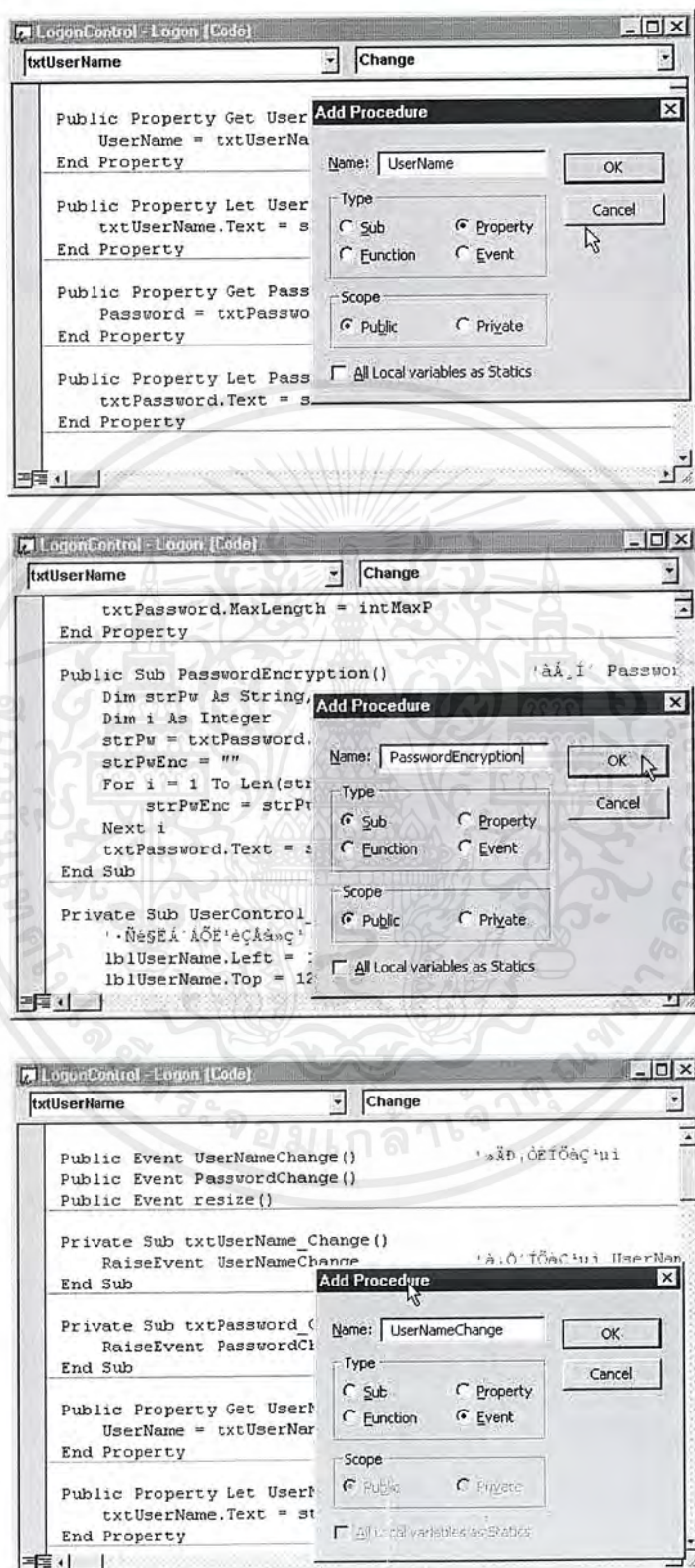


รูปที่ 4-7 หน้าตาของ ActiveX Control ที่จะสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. เขียนโค้ดเพื่อเพิ่มพรีอเพอร์ตี เมททอดและอีเวนต์โดยใช้เมนู *Tools > Add Procedure ...* ดัง

รูป 4-8



รูปที่ 4-8 การเพิ่มพรีอเพอร์ตี, เมททอดและอีเวนต์ให้กับ ActiveX Control

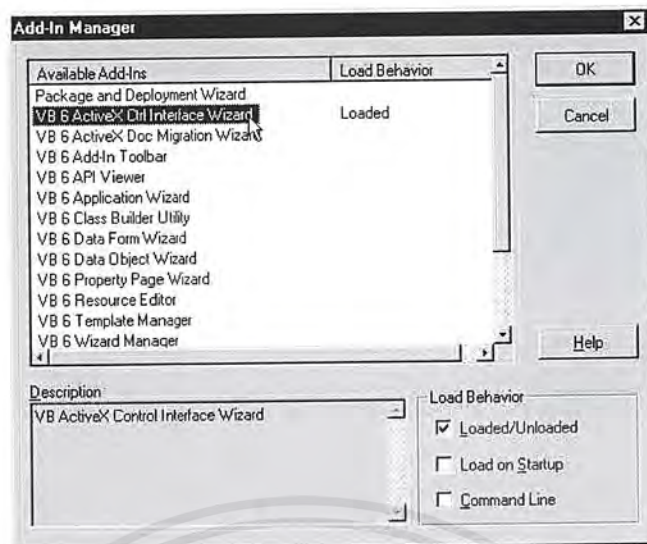
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. เนื่องจากต้องนำ ActiveX Control นี้ไปใช้ใน Visual Basic ดังนั้นต้องทำการเขียนโค้ดเพื่อรองรับการแสดงผลของการเปลี่ยนขนาดของ ActiveX Control โดยสามารถเขียนโค้ดได้ดังนี้

```
Private Sub UserControl_Initialize() 'เกิดอีเวนต์นี้เมื่อเราวาง ActiveX บนฟอร์ม
    lblUserName.Left = 120 'ทั้งหมดมีหน่วยเป็น Twip
    lblUserName.Left = 120
    lblUserName.Top = 120
    lblPassword.Left = lblUserName.Left
    lblPassword.Top = 600
    txtUserName.Left = 1080
    txtUserName.Top = lblUserName.Top
    txtPassword.Left = txtUserName.Left
    txtPassword.Top = lblPassword.Top
End Sub

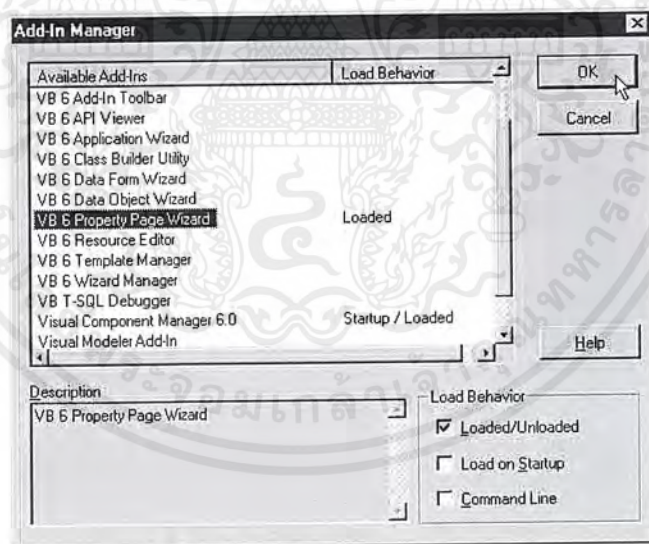
Private Sub UserControl_Resize()
    RaiseEvent resize
    'ไม่อนุญาตให้หดเล็กกว่าขนาดเริ่มต้น แต่ถ้าใหญ่กว่าเดิมให้ขยายได้
    If UserControl.Width < txtUserName.Left + txtUserName.Width + 20 Then
        UserControl.Width = txtUserName.Left + txtUserName.Width + 20
    Else 'เมื่อขยายขนาดใหญ่กว่าเดิม
        txtUserName.Width = UserControl.Width - txtPassword.Left - 20
        txtPassword.Width = UserControl.Width - txtPassword.Left - 20
    End If
    'ความสูงกำหนดตายตัว
    If UserControl.Height <> 1095 Then UserControl.Height = 1095
End Sub
```

7. เราสามารถนำพร็อพเพอร์ตี้ เมทธอดและอีเวนต์ของคอนโทรลที่นำมาประกอบกันเป็น ActiveX Control มาเพิ่มให้กับ ActiveX Control โดยใช้ wizard ของ Visual Basic ซึ่งสามารถเรียกใช้ได้โดยเมนู *Add-Ins > Add-In Manager* แล้วโหลด wizard ตัวนี้ ดังรูป 4-9



รูปที่ 4-9 การโหลด ActiveX Control Interface Wizard

8. ActiveX Control สามารถกำหนดพรีอเพอร์ตีต่างๆ จากทั้ง Property Window และ Property Page ซึ่งเราสามารถเพิ่ม Property Page ได้โดยใช้ wizard ของ Visual Basic โดยใช้เมนู *Add-Ins > Add-In Manager* แล้วโหลด wizard ตัวนี้ ดังรูป 4-10



รูปที่ 4-10 การโหลด Property Page Wizard

9. เนื่องจากคอนโทรลสามารถถูกทำลายและสร้างขึ้นใหม่ได้ ดังนั้น Visual Basic จึงสร้าง *Property Bag* เพื่อใช้ในการเก็บค่าและดึงข้อมูลของพรีอเพอร์ตีของคอนโทรล โดยจะทำในอ็อบเจกต์ *WriteProperties* และ *ReadProperties* ของ UserControl โดยเราสามารถสรุปขั้นตอนของการเปลี่ยนแปลงค่าพรีอเพอร์ตีได้ 2 ขั้นตอน คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- บอกให้ Visual Basic ทราบว่ามี การเปลี่ยนแปลงพรีอเพอร์ตี้เกิดขึ้น โดยทำการเพิ่มฟังก์ชัน *PropertyChanged* ไว้ใน procedure *Property Let* หรือ *Property Set*

- ทำการบันทึกพรีอเพอร์ตี้ที่เปลี่ยนแปลงลงในออบเจกต์ *Property Bag* โดยเมื่อฟังก์ชัน *PropertyChanged* ถูกเรียก เราต้องทำการเขียนโค้ดเพิ่มในอีเวนต์ *WriteProperties* เพื่อทำการบันทึกค่าลงไปในออบเจกต์ *Property Bag* ซึ่งเป็นที่เก็บค่าพรีอเพอร์ตี้ทั้งหมดของ ActiveX Control โดยโค้ดตัวอย่าง การทำ *Property Bag* สามารถเขียนได้ดังนี้

เขียนค่าพรีอเพอร์ตี้ลงในออบเจกต์ *Property Bag*

```
Private Sub UserControl_WriteProperties(PropBag As PropertyBag)
    Call PropBag.WriteProperty("BackColor", UserControl.BackColor, &H8000000F)
    Call PropBag.WriteProperty("BackStyle", UserControl.BackStyle, 0)
    Call PropBag.WriteProperty("BorderStyle", UserControl.BorderStyle, 0)
    Call PropBag.WriteProperty("ForeColor", UserControl.ForeColor, &H80000012)
    Call PropBag.WriteProperty("ToolTipText", txtUserName.ToolTipText, "")
End Sub
```

อ่านค่าพรีอเพอร์ตี้ที่เก็บในออบเจกต์ *Property Bag*

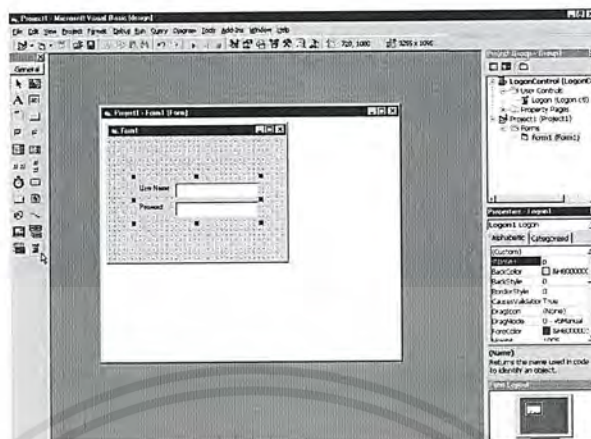
```
Private Sub UserControl_ReadProperties(PropBag As PropertyBag)
    UserControl.BackColor = PropBag.ReadProperty("BackColor", &H8000000F)
    UserControl.BackStyle = PropBag.ReadProperty("BackStyle", 0)
    UserControl.BorderStyle = PropBag.ReadProperty("BorderStyle", 0)
    UserControl.ForeColor = PropBag.ReadProperty("ForeColor", &H80000012)
    txtUserName.ToolTipText = PropBag.ReadProperty("ToolTipText", "")
End Sub
```

4.2.2 การทดสอบ ActiveX Control

หลังจากที่เราสร้าง ActiveX Control แล้วเราสามารถทำการทดสอบโดยใช้เครื่องมือในการดีบั๊กของ Visual Basic สำหรับขั้นตอนของการทดสอบมีดังนี้ คือ

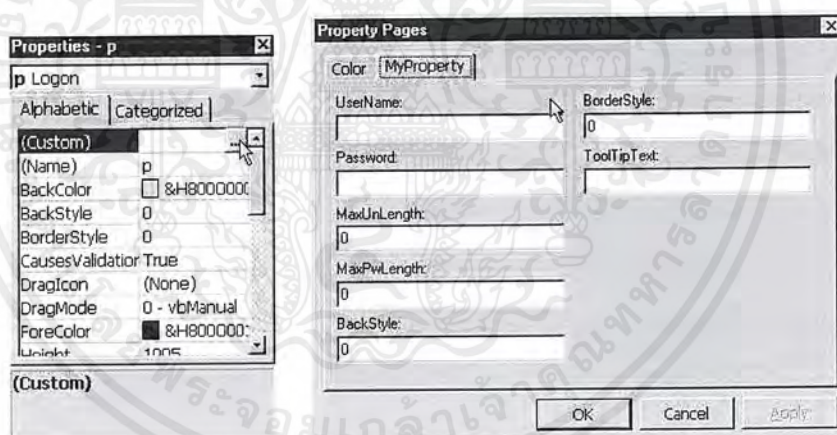
1. เพิ่ม project Standard EXE เข้าไปใน Project Explorer โดยเลือกเมนู *File > Add Project ...* แล้วเลือกไอคอน Standard EXE
2. ปิด window ของ UserControl ถ้าเปิดอยู่ Visual Basic จะมองว่าอยู่ในระหว่างการสร้าง ActiveX Control และจะไม่ยอมให้ทดสอบ

3. กลับไปที่ project Standard EXE จะเห็นว่าที่ Toolbox จะปรากฏไอคอนของ ActiveX Control ตัวใหม่ที่เราสร้าง ให้ทำการ double click เลือกมาวางบน form ดังรูป 4-11



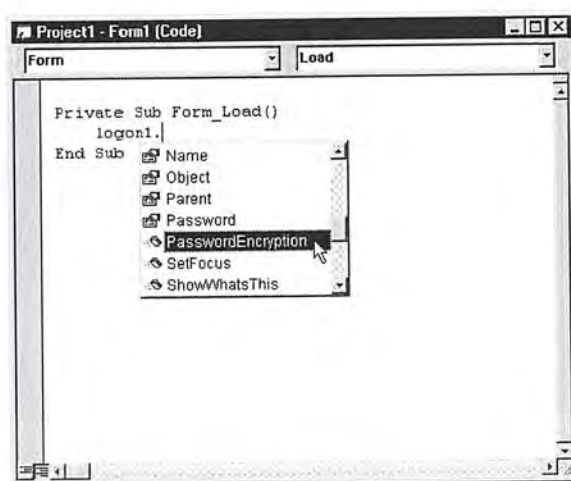
รูปที่ 4-11 การใช้ ActiveX Control ที่สร้างขึ้นใหม่

4. ตรวจสอบ Property Window ว่ามีพร็อพเพอร์ตี้ต่าง ๆ ครบหรือไม่ แล้วคลิกที่พร็อพเพอร์ตี้ custom เพื่อเรียก Property Page ดังรูป 4-12



รูปที่ 4-12 การตรวจสอบพร็อพเพอร์ตี้ต่าง ๆ และทดลองเรียกใช้ Property Page

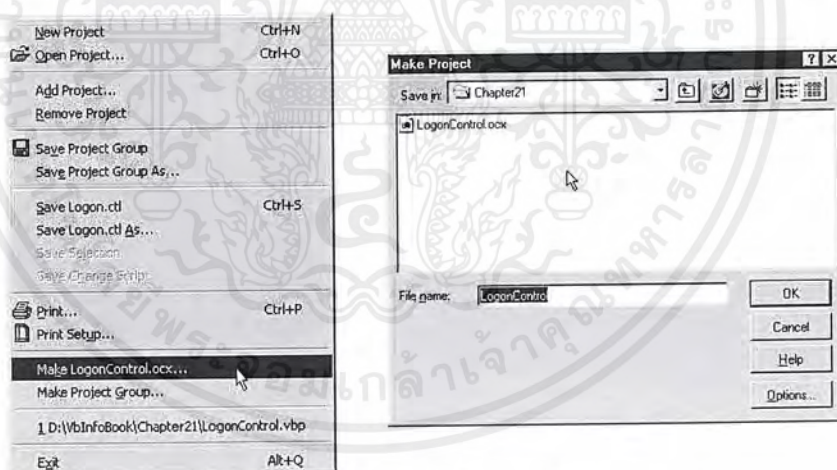
5. Double click ที่ ActiveX Control เพื่อเปิด Code Editor จากนั้นเขียนโค้ดเพื่ออ้างอิงถึงพร็อพเพอร์ตี้และเมทอดต่าง ๆ ของ ActiveX Control เพื่อดูว่าใช้งานได้จริงหรือไม่ ดังรูป 4-13



รูปที่ 4-13 การทดลองเขียนโค้ดใช้งาน ActiveX Control

4.2.3 การนำ ActiveX Control ไปใช้งานจริง

เมื่อทดสอบจนแน่ใจว่า ActiveX Control ทำงานถูกต้องแล้ว ให้ทำการ compile เพื่อให้ได้ไฟล์เป็น .OCX โดยเลือกเมนู *File > Make ...* ซึ่งจะได้ ActiveX Control ที่ register ในระบบของเราแล้วและพร้อมใช้งานใน Visual Basic ดังรูป 4-14

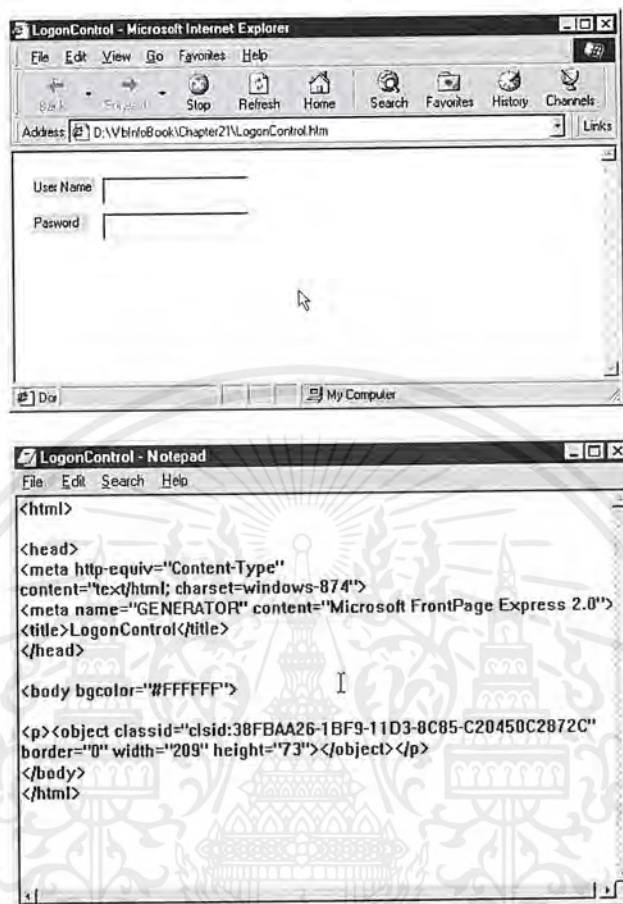


รูปที่ 4-14 การ Compile ActiveX Control เป็นไฟล์ .OCX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 การประยุกต์ใช้งาน ActiveX Control

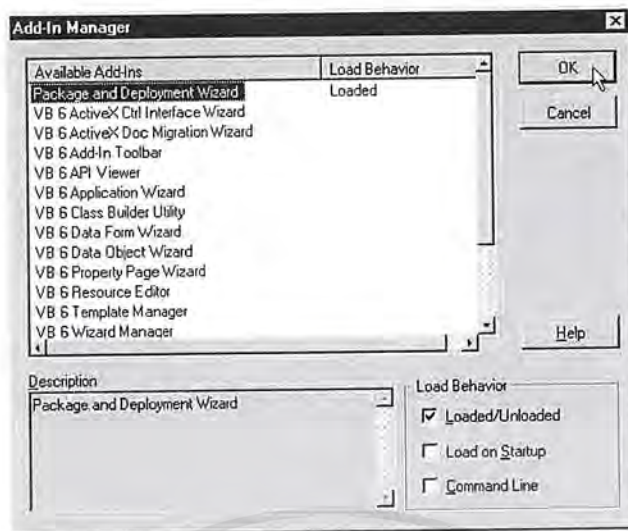
เราสามารถนำ ActiveX Control ที่สร้างขึ้น ไปใช้งานในบราวเซอร์ได้ ดังรูป 4-15



รูปที่ 4-15 การประยุกต์ใช้งาน ActiveX Control ในบราวเซอร์

แต่ถ้าเราต้องการให้ ActiveX Control ตัวนี้เป็นส่วนหนึ่งของอินเทอร์เน็ตหรืออินทราเน็ตโดยสามารถรันบนบราวเซอร์เครื่องอื่นได้นั้น เราต้องทำการติดตั้ง ActiveX Control ยังเครื่องคอมพิวเตอร์ที่เราต้องการก่อน เช่น ใน Web Server ของเรามีบาง web page ต้องใช้ ActiveX Control ที่เราสร้างขึ้น บราวเซอร์ก็จะทำการโหลดไฟล์ setup ของ ActiveX Control ที่เก็บไว้ที่ Web Server มาติดตั้งที่เครื่องของตนเองก่อน จากนั้นจึงจะให้ความสามารถของ ActiveX Control ใน web page นั้นได้

ในการจะสร้างไฟล์ setup สำหรับ ActiveX Control ที่ใช้งานร่วมกับบราวเซอร์นั้นจะใช้ *Package and Deployment Wizard* โดยทำการเลือกเมนู *Add-Ins > Add-In Manager ...* จากนั้นก็ทำการโหลด wizard ตัวนี้ ดังรูป 4-16



รูปที่ 4-16 การโหลด *Package and Deployment Wizard*

กล่าวโดยสรุป คือ ในการสร้าง ActiveX Control สามารถสร้างได้จาก Development Tool ต่าง ๆ เช่น Visual Basic 6.0 และ Delphi 4.0 เป็นต้น แต่สำหรับ Delphi นั้นในการสร้างค่อนข้างจะยุ่งยากมากกว่า Visual Basic เนื่องจากไม่มี wizard สำหรับช่วยในการสร้างมากนักทำให้ต้องทำการเขียนโค้ดเองเป็นส่วนใหญ่ ส่วนในการนำ ActiveX Control ไปประยุกต์ใช้นั้นสามารถนำไปประยุกต์ใช้ได้ทั้งในแอปพลิเคชันอินเทอร์เน็ตหรืออินทราเน็ตและในบราวเซอร์ โดยในการที่จะให้เครื่องอื่นสามารถใช้งาน ActiveX Control ได้ นั้น เราต้องทำการติดตั้ง ActiveX Control ลงในเครื่องนั้นก่อนสำหรับแอปพลิเคชันส่วนบราวเซอร์นั้นเราต้องทำการสร้างไฟล์ setup ไว้ที่ Web Server เมื่อมีคนมาโหลด web page แล้วจะได้ นำไฟล์ setup ไปติดตั้งยังเครื่องของตนเอง

4.3 ActiveX Document

ActiveX Document เป็นแอปพลิเคชันซึ่งรันภายในคอนเทนเนอร์ แทนที่จะเป็นการรัน โปรแกรมแบบ stand-alone โดยไฟล์ของ ActiveX Document นั้นเป็นไฟล์ .VBD และมีลักษณะเหมือนกับไฟล์ของ Microsoft Word และ Microsoft Excel คือ ต้องการคอนเทนเนอร์สำหรับเปิดดูและไม่สามารถเปิดขึ้นมาดูได้ โดยปราศจากคอนเทนเนอร์เหล่านั้น เช่น คุณสามารถดู spreadsheet ได้ก็ต่อเมื่อเครื่องของคุณได้ทำการติดตั้ง Microsoft Excel

แต่สำหรับ ActiveX Document นั้นมีคอนเทนเนอร์ที่รองรับอยู่หลายตัว ซึ่งรวมถึง Internet Explorer, Microsoft Office Binders และ Visual Basic Integrated Development Environment (IDE) โดยสามารถเลือกที่จะรัน ActiveX Document บนคอนเทนเนอร์ตัวไหนนั้น

คุณสมบัติของ ActiveX Document ก็คือ เป็นแอปพลิเคชันตัวหนึ่งซึ่งแตกต่างจาก ActiveX Control ที่ถูกแปะลงบน form เพื่อสร้างแอปพลิเคชัน โดย ActiveX Document เป็นออบเจกต์ form ที่มีคอนโทรลแปะอยู่เพื่อใช้ในการสร้าง user interface ซึ่งวิธีนี้เป็นวิธีการในการสร้างแอปพลิเคชันที่เรียกว่า

Component Based

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1 ข้อดีของ ActiveX Document

ข้อดีหลักของ ActiveX Document ใน Visual Basic ก็คือ ในการสร้างแอปพลิเคชันแบบ component-based นั้นสามารถใช้ได้ทั้งบนเดสก์ท็อปและอินเทอร์เน็ตโดยไม่ต้องใช้อินเทอร์เน็ตที่แตกต่างกัน โดย Active desktop ของ Internet Explorer 4.0 เป็นตัวอย่างที่ยอดเยี่ยมของ ActiveX Document ซึ่งสามารถใช้ในการสร้างอินเทอร์เน็ตที่เป็นแบบ component-based นอกจากนี้ข้อดีของ ActiveX Document สามารถสรุปได้ดังนี้

- ในการสร้าง ActiveX Document ไม่จำเป็นต้องรู้ภาษาในการเขียนโปรแกรมมากมาย เพียงแต่รู้ Application Programming Interface (API) ที่สามารถถูกเรียกโดย Visual Basic
- Visual Basic IDE ใช้เวลาในการสร้างแอปพลิเคชันทางอินเทอร์เน็ตน้อยกว่า เมื่อเทียบกับภาษาที่ไม่มี tool สำหรับช่วยในการสร้าง โดย ActiveX Document สามารถนำ ActiveX Control ที่มีอยู่แล้วมาใช้ในการสร้างได้ (ยกเว้นบางตัว เช่น OLE Container)
- ในการดื่บัก ActiveX Document นั้นไม่แตกต่างจากการดื่บักแอปพลิเคชันทั่ว ๆ ไปของ Visual Basic
- เราสามารถที่จะแปลงแอปพลิเคชันที่สร้างไว้แล้วไปเป็น ActiveX Document โดยใช้ wizard ของ Visual Basic
- ออบเจกต์ Hyperlink สามารถเชื่อมโยงเอกสารไปยัง page อื่น ๆ โดย page เหล่านั้นสามารถเป็นได้ทั้ง ActiveX Document หรือ Web page

4.3.2 ประเภทของ ActiveX Document

เราสามารถแบ่ง ActiveX Document เป็น 2 ประเภท คือ

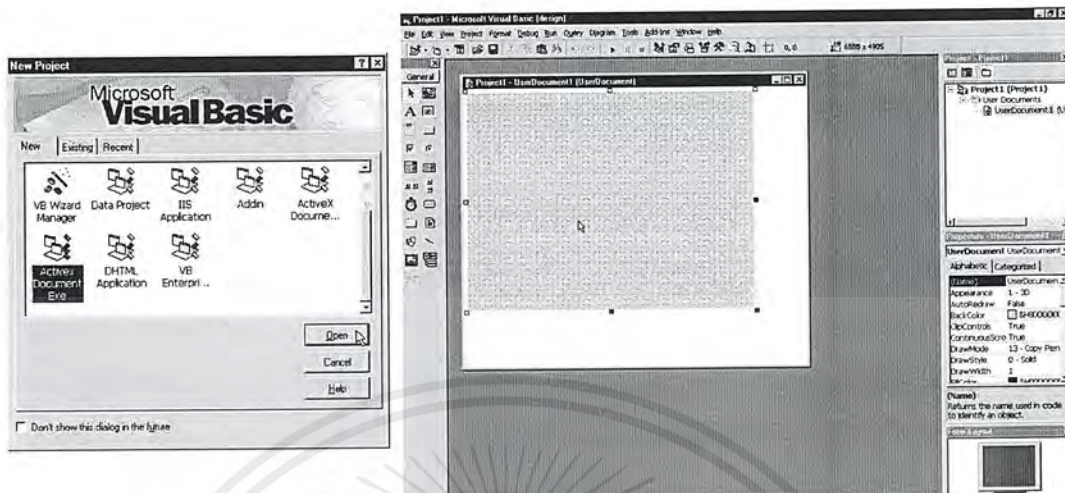
- ActiveX Document EXE เป็น Out-process ActiveX Document คือ เมื่อทำการรันแอปพลิเคชันจะใช้หน่วยความจำคนละส่วนกับบราวเซอร์
- ActiveX Document DLL เป็น In-process ActiveX Document คือ เมื่อทำการรันแอปพลิเคชันจะใช้หน่วยความจำเดียวกับบราวเซอร์

เมื่อเปรียบเทียบการทำงานของแอปพลิเคชันทั้งสองแบบก็จะเห็นว่า ActiveX Document DLL จะทำงานได้เร็วกว่า ActiveX Document EXE แต่มีความปลอดภัยน้อยกว่า เพราะถ้าใน ActiveX Document DLL มีความผิดพลาดในการทำงานเกิดขึ้นจะทำให้การทำงานของบราวเซอร์ไม่เสถียรภาพ ต่างจาก ActiveX Document EXE ซึ่งความผิดพลาดจากการทำงานของแอปพลิเคชันจะไม่มีผลต่อการทำงานของบราวเซอร์

4.3.3 ขั้นตอนการสร้าง ActiveX Document

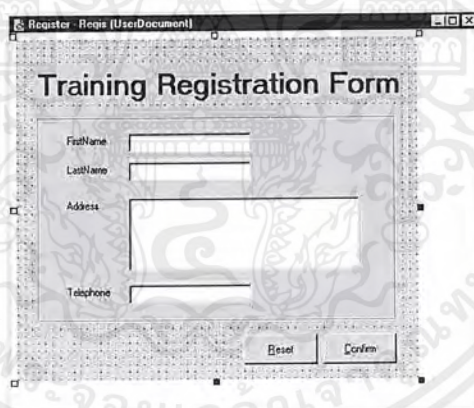
ในที่นี้จะนำเสนอเฉพาะการสร้าง ActiveX Document EXE ซึ่งมีขั้นตอนดังนี้

1. เลือก project เป็น ActiveX Document EXE แล้ว Visual Basic จะสร้าง UserDocument Designer ซึ่งจะคล้ายกับ form แต่จะมีพร็อพเพอร์ตี้และอีเวนต์บางอย่างต่างจาก form ดังรูป 4-17



รูปที่ 4-17 การเลือก project เป็น ActiveX Document EXE

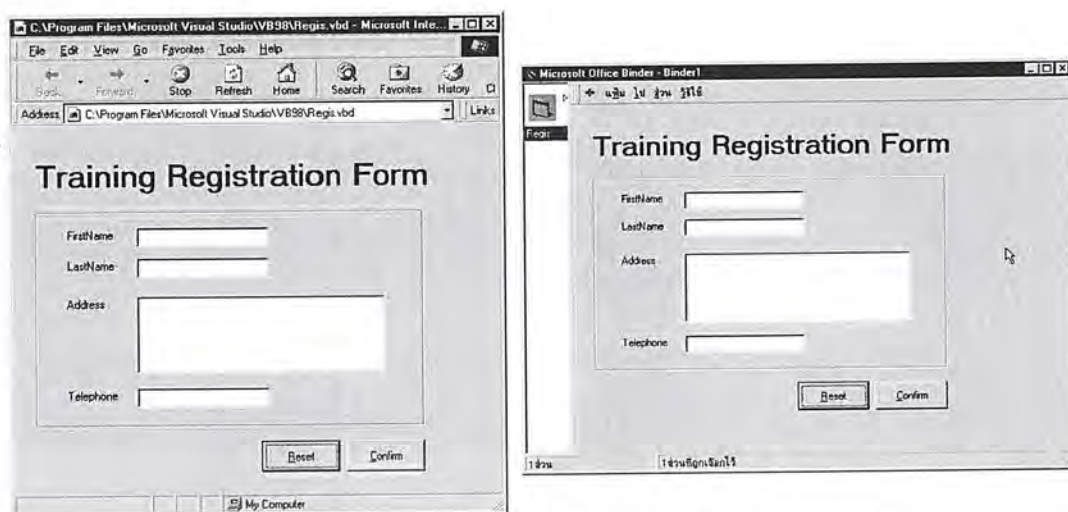
2. ทำการแปะคอนโทรลต่าง ๆ ลงบน UserDocument ดังรูป 4-18



รูปที่ 4-18 การแปะคอนโทรลต่าง ๆ ลงบน UserDocument

3. ทำการ compile เป็นไฟล์ .EXE จากนั้นก็ทำการทดสอบโดยการรันไฟล์ .VBD ทั้งใน Visual Basic และ Microsoft Office Binder เพื่อตรวจสอบการทำงานของ ActiveX Document EXE ดังรูป 4-19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-19 การรัน ActiveX Document EXE ทั้งใน VB และ Microsoft Office Binder

เมื่อเราสร้าง ActiveX Document เสร็จเรียบร้อยแล้ว เราสามารถเผยแพร่ผลงานที่ทำเพื่อให้นักอื่นที่สนใจนำไปติดตั้งและเรียกใช้ ActiveX Document นั้น โดยเราต้องสร้าง package หรือไฟล์ setup โดยใช้ Package and Deployment Wizard ของ Visual Basic ซึ่งเหมือนกับในการสร้างไฟล์ setup ของ ActiveX Control ที่ได้กล่าวมาแล้วข้างต้น

4.3.4 ข้อเปรียบเทียบการนำ ActiveX Control และ Active Document ไปใช้งานบนเว็บเพจ

- ActiveX Control ถูกนำไปใช้บนเว็บเพจในรูปแบบ DHTML (Dynamic Hypertext Markup Language) เมื่อโคลนเน็ตเรียกดูเว็บเพจจะต้องดึง ActiveX Control ที่อยู่ในเว็บเพจไปติดตั้งยังเครื่องฟังก์โคลนเน็ต
- Active Document เป็นการพัฒนาแอปพลิเคชันเพื่อใช้งานข้ามแพลตฟอร์มได้โดยสามารถใช้งานบนเว็บเบราว์เซอร์ที่สนับสนุน ActiveX Technology ในการเรียกใช้งานจะต้องโหลดโปรแกรมทั้งหมดซึ่งอยู่ในรูปไฟล์บีบอัด (.CAB) ไปติดตั้งยังเครื่องฟังก์โคลนเน็ต

4.3.5 ข้อดีและข้อเสียของการนำ ActiveX Control และ Active Document ไปใช้งานบนเว็บเพจ

- Active Document ถูกนำไปใช้ในระบบอินเทอร์เน็ตโดยสามารถพัฒนาแอปพลิเคชันที่มีความซับซ้อนมากกว่า DHTML
- ส่วน ActiveX Control จะถูกใช้ในเว็บเพจทั่วไปเพื่อทำให้เว็บเพจสามารถโต้ตอบกับผู้ใช้ได้อย่างมีประสิทธิภาพ
- ขนาดของ ActiveX Control ที่ใช้ในเว็บเพจมีขนาดเล็กกว่าแอปพลิเคชัน Active Document ทำให้ลดภาระของเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ActiveX Server (OLE Automation)

ActiveX Server เป็นเทคโนโลยีที่พัฒนาซอฟต์แวร์ขึ้นเพื่อใช้งานและให้สามารถนำฟังก์ชันการทำงานในซอฟต์แวร์นั้นมาใช้พัฒนาในซอฟต์แวร์ตัวอื่นได้ ยกตัวอย่างเช่น ไมโครซอฟท์เวิร์ดมีอินเตอร์เฟสให้เรียกใช้มากมาย เช่น การตัดคำ ทำให้ผู้พัฒนาซอฟต์แวร์อื่นสามารถเรียกใช้ความสามารถนี้โดยไม่ต้องพัฒนาเองได้

ActiveX Server จะมีการทำงานแบ่งออกเป็น 2 ส่วน คือ

- ส่วน ActiveX Server คือ โปรแกรมที่ให้เซอร์วิสหรือ Class(Object) กับ Automation Controller เรียกใช้

- ส่วน Automation Controller คือ โปรแกรมที่เรียกใช้ Automation Server

ActiveX Server ก็มีการแบ่งออกเป็น 2 ชนิด คือ

- In-Process ActiveX Server คือ ตัว ActiveX Server อยู่ใน DLL

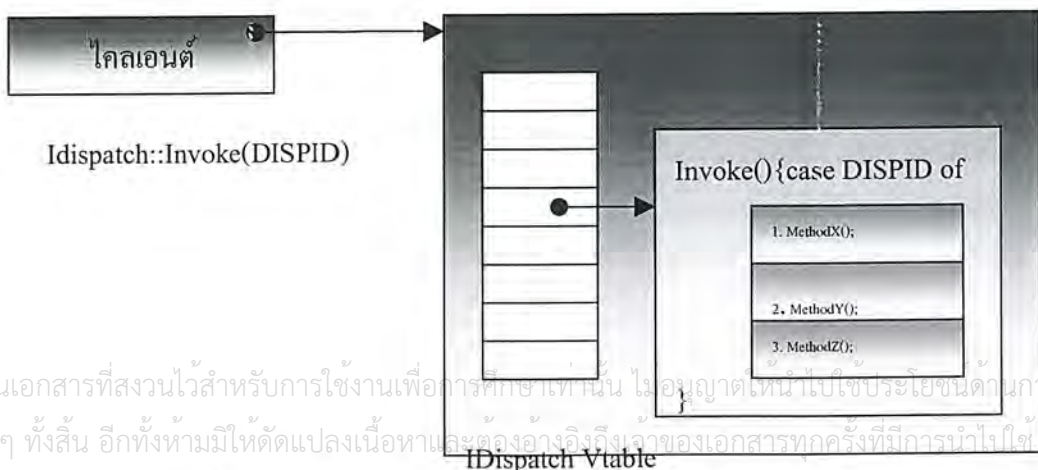
- Out-Process ActiveX Server คือ ตัว ActiveX Server อยู่ในโปรแกรมคนละตัวกับ Automation Controller

Automation Controller มีวิธีการเรียกใช้ Method ที่ Automation Server มีให้ 3 วิธี คือ

- ใช้ Variant (เป็นรูปแบบข้อมูลที่สามารถเก็บข้อมูลได้ทุกรูปแบบ) เรียก ActiveX Server เป็นวิธีที่ง่ายแต่จะช้าและเกิดโอเวอร์เฮดที่คอมไพเลอร์ต้องทำ Type Checking

- ใช้ Definition อินเตอร์เฟสของออบเจกต์ (Type Library) บน ActiveX Server โดย Import เข้ามาเรียกใช้ใน Automation Controller ซึ่งเป็นการเรียกโดยตรงมากกว่าวิธีอื่น คอมไพเลอร์จะสามารถผลิตโค้ดที่สามารถรันได้เร็วกว่าแบบอื่น วิธีนี้ยังสามารถดัดแปลงไปใช้ Dispatch Interface

- ใช้เรียกผ่าน Dispatch Interface โดยใช้อินเตอร์เฟส IDispatch ซึ่งเหมือนกับอินเตอร์เฟสทั่วไป คือ มี VTable ที่เก็บพอยน์เตอร์ไปยังเมธอดอื่น ๆ ของอินเตอร์เฟสแต่ที่พิเศษ คือ มีเมธอด Invoke() ที่ทำหน้าที่ในการเรียกเมธอดอื่นได้อีกชั้นหนึ่งโดยเมธอดที่สามารถเรียกผ่าน Dispatch Interface ซึ่งกำหนดโดยผู้พัฒนาซอฟต์แวร์และเราสามารถเรียกอินเตอร์เฟสด้วย IDispatch::Invoke โดยแต่ละเมธอดจะถูกกำหนดหมายเลขกำกับเป็น Dispatch Identifier (DISPID) โดยเมื่อไคลเอนต์จะเรียกใช้ Dispatch Interface ก็จะใช้เรียกผ่านเมธอด Invoke(DISPID) ดังรูป 4-20 โดยออบเจกต์ที่มีการอินเตอร์เฟส IDispatch จะเปรียบเสมือนมี Case Statement เพื่อเลือกเมธอดที่ต้องการ



รูปที่ 4-20 การทำงานของ ActiveX Server โดยการอ้าง Dispatch Interface

4.4.1 ตัวอย่างการสร้าง ActiveX Server

เป้าหมาย

1. เพื่อพิสูจน์คุณสมบัติเรื่อง “ไม่ขึ้นกับเครื่องมือที่พัฒนา”
2. เพื่อทดสอบการเรียกใช้เมธอดของ ActiveX Server แบบต่าง ๆ
3. เพื่อแสดงเทคโนโลยีการพัฒนาซอฟต์แวร์ที่อิงตามรูปแบบ COM

เครื่องมือที่ใช้พัฒนา

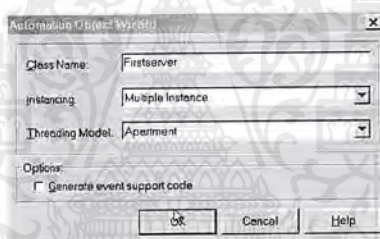
1. Visual Basic 6.0 Enterprise Edition
2. Borland Delphi 5.0(Built5.62) Enterprise

คำอธิบายโปรแกรม

ตัวอย่างการเปลี่ยนสีของฟอร์มเซิร์ฟเวอร์โดยสั่งจากไคลเอนต์และการจับเวลาการทำงานของเมธอดโดยมีการเรียกใช้ที่แตกต่างกัน

ขั้นตอนการพัฒนา

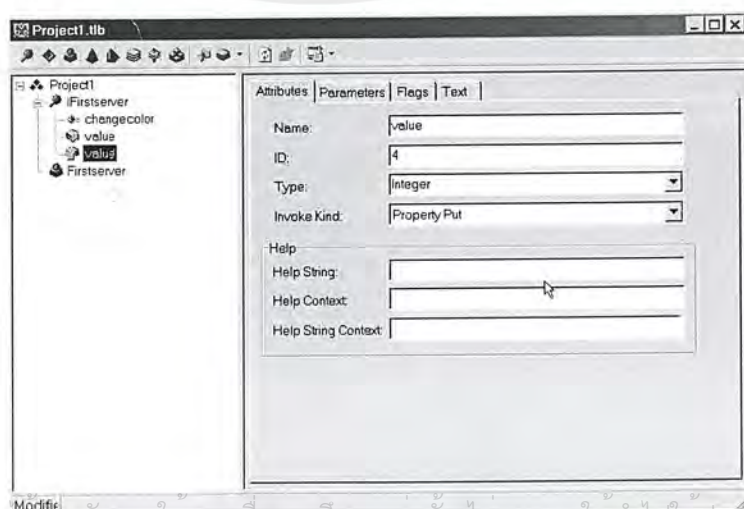
1. เปิด project ใหม่เป็น Application ธรรมดา
2. เรียกใช้ Delphi's Automation Object Wizard ใส่ชื่อ Class ที่จะสร้าง ดังรูป 4-21



รูปที่ 4-21 การกำหนดค่าให้กับ ActiveX Server

หลังจากนั้น Delphi จะทำการ Generate Code รวมถึง Type Library มาให้ ดังรูป 4-22

2. ทำการเพิ่มเมธอด ChangeColor และพรีอเพอร์ตี้ Value เข้าไป ดังรูป 4-22 เมื่อเรากดปุ่ม refresh จะมีการ Generate Code ตาม Type Library ออกมา



รูปที่ 4-22 Type Library Editor

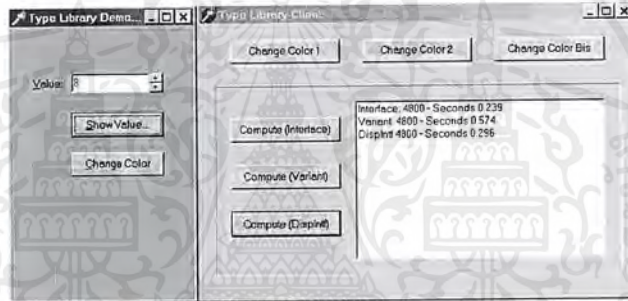
3. จากนั้นก็เขียนโค้ดเพิ่มในส่วนของเมธอดต่าง ๆ
4. Register Automation Server ที่เขียนขึ้นมาโดยใช้คำสั่งที่ command line ดังตัวอย่าง

```
C:\MastDelphi4\md4code\Part4\18\TLIBDEMO>tlibdemo/regserver
C:\MastDelphi4\md4code\Part4\18\TLIBDEMO>
```

ถ้าต้องการ Unregistered ก็ทำดังตัวอย่าง คือ

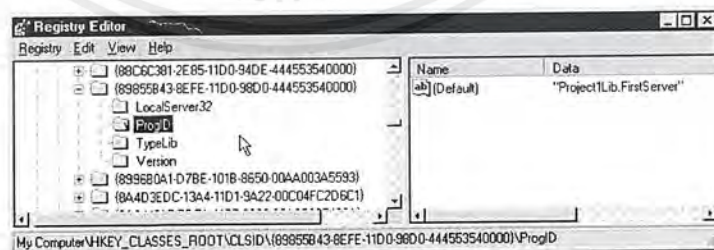
```
C:\MastDelphi4\md4code\Part4\18\TLIBDEMO>tlibdemo/unregserver
C:\MastDelphi4\md4code\Part4\18\TLIBDEMO>
```

5. ทำการทดสอบเซิร์ฟเวอร์ที่เราเขียนขึ้น โดยจะใช้โปรแกรม Libcli.exe ที่เขียนขึ้นเป็นตัวทดสอบ ซึ่งเป็นโปรแกรมที่แสดงการควบคุม ActiveX Server และแสดงถึงประสิทธิภาพการเรียกใช้เมธอดด้วยวิธีต่างๆ โดยวัดจากเวลาที่ใช้ในการคำนวณ ดังรูป 4-23



รูปที่ 4-23 แสดงการทดสอบประสิทธิภาพวิธีการเรียกเมธอดในรูปแบบต่างๆ

6. ทำการสร้างโปรแกรมฝั่งไคลเอนต์โดยเครื่องมือ Visual Basic เพื่อทดสอบคุณสมบัติ “ไม่ขึ้นกับเครื่องมือที่พัฒนา” โดยต้องหาในรีจิสทรีของระบบว่า ActiveX Server ที่เราสร้างขึ้นมาจากเดลไฟก่อน ดังรูป 4-24

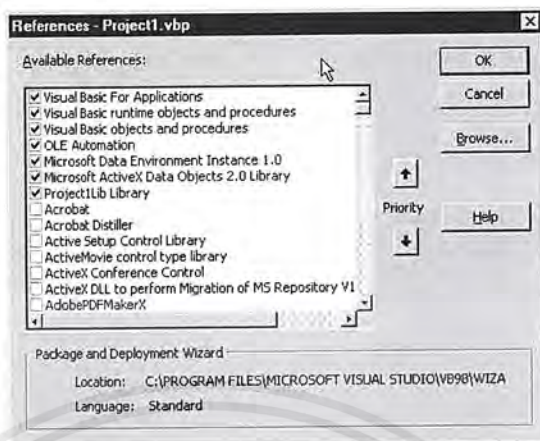


รูปที่ 4-24 ข้อมูลของ ActiveX Server ในรีจิสทรีของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

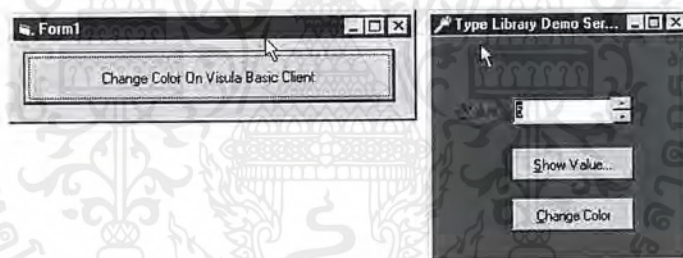
7. ทำการสร้างโปรแกรม EXE ธรรมดาโดย Visual Basic โดยต้องมีการตั้งค่า Reference ดังรูป

4-25



รูปที่ 4-25 การอ้างอิง ActiveX Server ในโปรแกรม

8. เขียนโค้ดในส่วนของฟังก์ชันการทำงานที่จำเป็นเพิ่ม
9. โปรแกรมจะมีลักษณะการทำงาน ดังรูป 4-26



รูปที่ 4-26 การทำงานของโปรแกรมเพื่อควบคุม ActiveX Server

4.4.2 ผลการทดสอบ

1. การอ้างเมธอดในรูปแบบต่าง ๆ นั้นมีผลต่อการทำงานของโปรแกรมโดยการผ่าน VTable เป็นวิธีที่เร็วที่สุด วิธี Dispatch Interface มีความเร็วรองลงมา และ การอ้างโดย Variant เป็นวิธีที่ช้าที่สุด
2. ActiveX Server สามารถถูกเรียกใช้ผ่านเครื่องมือพัฒนาใด ๆ ที่สนับสนุนเทคโนโลยี COM

เนื่องจาก OLE Automation เป็นมาตรฐานเพราะฉะนั้นเราสามารถรวมโปรแกรมที่เราเขียนขึ้นมา กับโปรแกรมที่มีอยู่ในปัจจุบันซึ่งสนับสนุน OLE Automation ได้ ซึ่งมีประโยชน์ คือ

- ในการเขียนโปรแกรมที่มีความซับซ้อนมากๆ เราอาจจะไม่ต้องเขียนทั้งหมดเราอาจจะใช้คอมโพเนนต์ที่ซับซ้อนซึ่งอยู่ในซอฟต์แวร์บางตัวในการพัฒนาโปรแกรมได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อเสีย

- ผู้ใช้โปรแกรมต้องมีซอฟต์แวร์ที่เราไปใช้คอมพิวเตอร์นั้น
- เป็นการผูกมัดโปรแกรมที่เราเขียนขึ้นกับเวอร์ชันของเซิร์ฟเวอร์โดยเฉพาะถ้าเราเรียกใช้แบบอินเทอร์เน็ตเฟสแทนที่จะใช้ Variant

4.5 ActiveX Scripting

เนื่องจาก HTML มีความสามารถในการโปรแกรมและติดต่อกับผู้ใช้ค่อนข้างต่ำ ทำให้ต้องมีสิ่งที่เพิ่มความสามารถให้กับ web page นั่นคือ *script* สำหรับ *script* ที่ได้รับความนิยมมากที่สุดก็คือ VBScript เพราะมีลักษณะการโปรแกรมใกล้เคียงกับ Visual Basic จึงเหมาะกับผู้คุ้นเคยกับ Visual Basic มาก่อนแล้ว

โดย VBScript เป็นการเขียนโปรแกรมในแบบของ Visual Basic มาเขียนเพิ่มเติมเข้าไปใน web page โดยเพิ่มเข้าไประหว่าง Tag <SCRIPT>

4.5.1 ส่วนประกอบของ web page ที่ใช้งานกับ VBScript

- คำสั่ง HTML เป็นส่วนที่บรรจุข้อความภาษา HTML ซึ่งบราวเซอร์ทุกตัวสามารถอ่านได้เข้าใจ
- VBScript Delimiter เป็นส่วนที่แยกระหว่าง VBScript กับ HTML ออกจากกัน ซึ่งคำสั่งของ VBScript จะอยู่ระหว่าง Tag <SCRIPT> และอาจถูกครอบงำได้ด้วย <!-- กับ --> เพื่อแยกความแตกต่างของทั้ง 2 ส่วนออกจากกัน ซึ่งมีข้อดี คือ ถึงแม้บราวเซอร์ไม่สนับสนุน VBScript ก็ยังคงเปิดแสดงผลข้อมูลในส่วนที่ต้องการได้อย่างถูกต้อง
- Subroutine หรือ Function ผู้ใช้สามารถสร้าง โปรแกรมย่อย ซึ่งเหมือนกับ subroutine และ function ของ Visual Basic
- VBScript Built-in Object เป็นออบเจกต์ที่มีอยู่แล้วของ VBScript

4.5.2 ความแตกต่างระหว่าง VB Basic และ VBScript

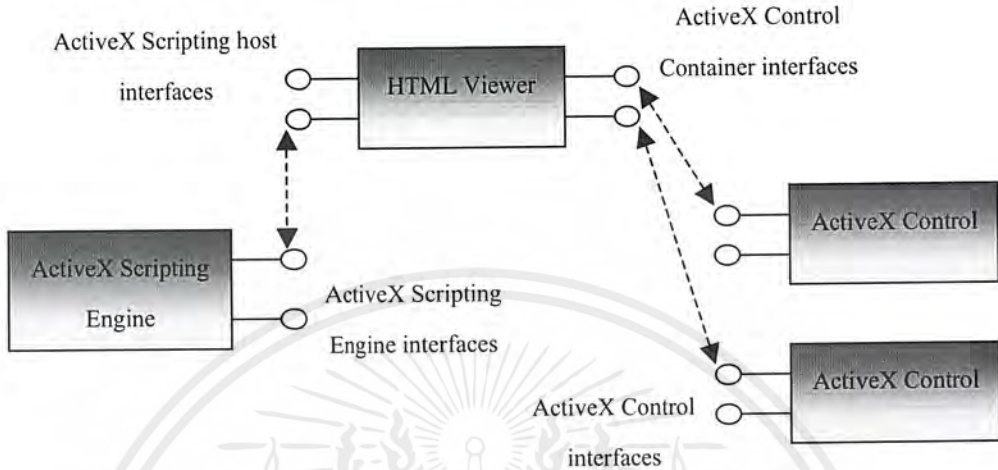
สำหรับความแตกต่างของ Visual Basic กับ VBScript นั้นสามารถสรุปได้ ดังตาราง 4-2

ข้อแตกต่าง	Visual Basic	VBScript
ชนิดข้อมูลของตัวแปร	กำหนดชนิดข้อมูลของตัวแปร ได้ทุกชนิด	ข้อมูลทุกชนิดเป็น variant
การใช้งาน Goto	สนับสนุน	ไม่สนับสนุน
การประกาศค่าคงที่	ต้องระบุชนิดของค่าคงที่	ไม่ต้องระบุชนิดของค่าคงที่
การเรียกใช้ ActiveX Control	เลือกวาง ActiveX Control จาก ToolBox ได้ตั้งแต่ตอนออกแบบ	ต้องใช้ในการเขียนโค้ดเพื่อเรียกใช้ ActiveX Control เท่านั้น

ตารางที่ 4-2 แสดงข้อแตกต่างระหว่าง VB กับ Visual Basic

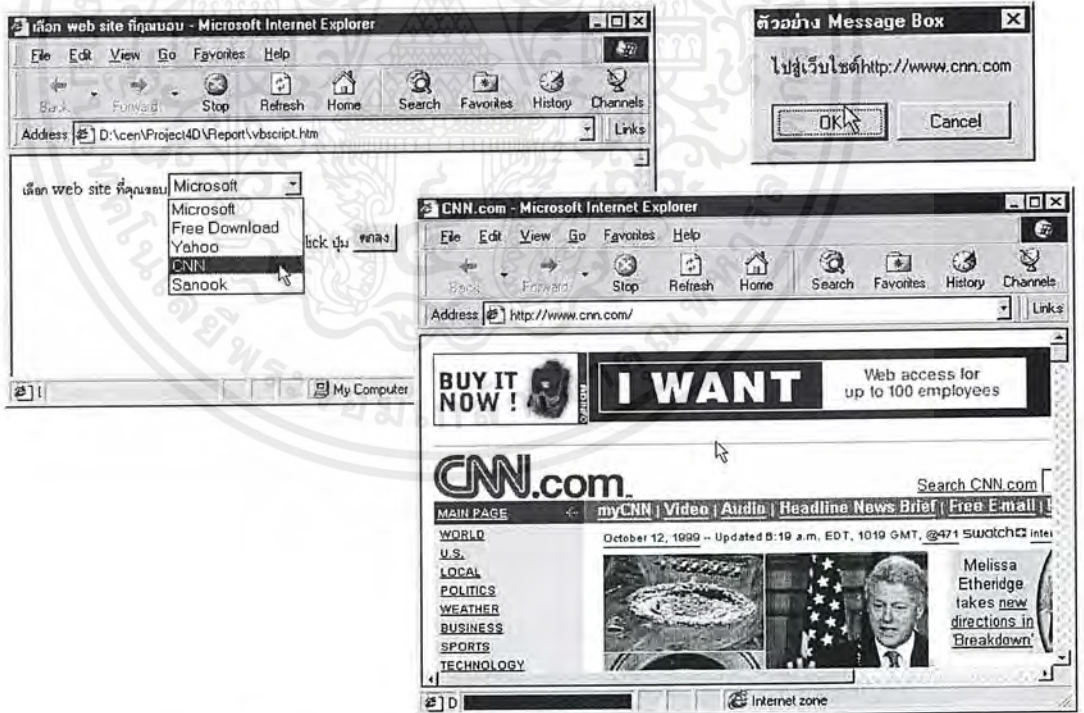
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย ActiveX Scripting เป็นเทคโนโลยีที่นำมา implementation โดยภาษา script ต่าง ๆ เช่น VBScript โดย script จะถูก execute โดย script engine ภายใต้การควบคุมของโฮสต์ เมื่อขณะ script กำลัง execute มันสามารถเรียกใช้ม็อดของออบเจกต์ของ ActiveX Control โดยผ่าน HTML Viewer และสามารถรับอีเวนต์จากออบเจกต์เหล่านั้นได้ ดังรูป 4-27



รูปที่ 4-27 การทำงานของ ActiveX Scripting

ตัวอย่างการประยุกต์ใช้งาน web page ที่มี VBScript ดังรูป 4-28



รูปที่ 4-28 การประยุกต์ใช้งาน web page ที่มี VBScript

ส่วนโค้ดนั้นสามารถเขียนได้ดังนี้

```
<body bgcolor="#FFFFFF">
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

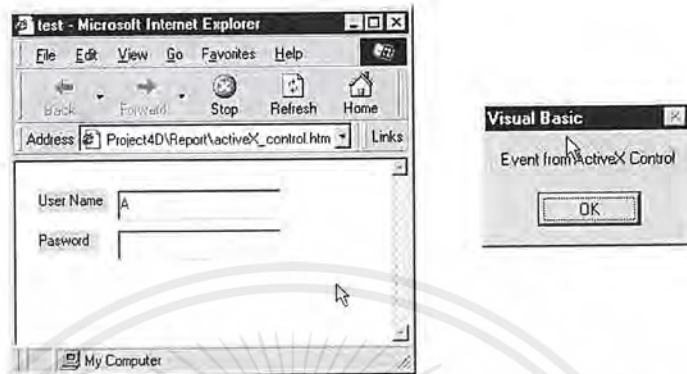
<form name="frmsample">
  <p><font face="Times New Roman">เลือก web site </font>ที่คุณชอบ<select
name="selweb" size="1">
  <option selected value="http://www.microsoft.com">Microsoft</option>
  <option value="http://www.download.com">Free Download</option>
  <option value="http://www.yahoo.com">Yahoo</option>
  <option value="http://www.cnn.com">CNN</option>
  <option value="http://www.sanook.com">Sanook</option>
</select></p>
  <p align="center"><font face="Times New Roman">เลือกแล้ว
click </font>ปุ่ม <input type="button" name="cmdweb"
value="ตกลง"></p>
</form>
<script language="VBScript">
<!--
  Dim vbokcancel,vbok
  vbokcancel = 1
  vbok = 1
  sub cmdweb_onclick
    dim idx,site,resp
    idx = frmsample.selweb.selectedindex
    site = frmsample.selweb.options(idx).value
    resp = msgbox("ไปสู่เว็บไซต์" &site,vbokcancel,"ตัวอย่าง Message Box")
    if resp = vbok then
      call gotoweb site(site)
    else
      msgbox("วันนี้ตรงกับ"&Date)
    end if
  end sub
  sub gotoweb site(site)
    window.open site
  end sub
-->
</script>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

</body>

ตัวอย่างการประยุกต์ใช้งาน web page ที่มี ActiveX Scripting โดยใช้ ActiveX Control ที่สร้างไว้ในตอนต้น ดังรูป 4-29



รูปที่ 4-29 การประยุกต์ใช้งาน web page ที่มี ActiveX Scripting

ส่วนโค้ดนั้นสามารถเขียนได้ดังนี้

```
<body bgcolor="#FFFFFF">
<p><object id="test"
classid="clsid:38FBAA26-1BF9-11D3-8C85-C20450C2872C"
align="baseline" border="0" width="209" height="73"></object></p>
<script language="vbscript">
<!--
sub test_usernamechange()
msgbox("Event from ActiveX Control")
end sub
-->
</script>
</body>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

IIS Application

(Internet Information Services Application)

5.1 ความหมายของ Active Server Pages (ASP)

ASP เป็นคำที่ย่อมาจาก Active Server Pages ซึ่งเป็นโปรแกรมที่พัฒนาขึ้นโดยบริษัท ไมโครซอฟต์เพื่อใช้งานทางด้านอินเทอร์เน็ต โดย ASP จะทำหน้าที่ตีความเอกสารที่เขียนด้วยภาษา สคริปต์ เช่น VBScript โดยที่มีแท็ก ASP (คือ คำสั่งที่มีเครื่องหมาย <% %>) กำกับอยู่ ซึ่งเบราว์เซอร์ทั่ว ๆ ไป เช่น Netscape Navigator หรือ Internet Explorer ไม่สามารถนำไปแสดงผล จากนั้นจึงสร้างเอกสารผลลัพธ์เป็นเอกสาร HTML อันเป็นเอกสารที่ประกอบด้วยแท็ก HTML ต่าง ๆ (คือ คำสั่งที่มีเครื่องหมาย <>) กำกับอยู่ ซึ่งเบราว์เซอร์ทั่วไปสามารถนำไปสร้างเป็นเว็บเพจขึ้นเพื่อใช้แสดงผล

การทำงานของโปรแกรม ASP จะเกิดขึ้นเฉพาะทางฝั่งเซิร์ฟเวอร์เท่านั้น ซึ่งเรียกว่า การทำงานแบบ server side จากนั้นผลลัพธ์ที่ได้จะถูกส่งไปให้เว็บเซิร์ฟเวอร์ แล้วเว็บเซิร์ฟเวอร์ก็จะส่งเอกสารดังกล่าวไปยังเว็บเบราว์เซอร์อีกทีหนึ่ง เบราว์เซอร์ก็จะสามารถแสดงผลได้อย่างถูกต้อง

5.2 กระบวนการทำงานของ Active Server Pages

การทำงานทั้งหมดจะเริ่มจากเบราว์เซอร์ร้องขอเอกสาร HTML ไปยังเว็บเซิร์ฟเวอร์ผ่านทาง HTTP โดยที่เอกสารที่ขอไปจะเป็นแฟ้มข้อมูลที่มีนามสกุลเป็น .asp เมื่อเว็บเซิร์ฟเวอร์ได้รับการร้องขอดังกล่าว ก็จะส่งเอกสารนั้นไปให้ ASP ตีความ จากนั้นเว็บเซิร์ฟเวอร์ก็จะสร้างเอกสาร HTML ส่งกลับไปให้เว็บเซิร์ฟเวอร์เพื่อส่งต่อไปยังเบราว์เซอร์และใช้แสดงผลทางฝั่งผู้ใช้ต่อไป ซึ่งการทำงานของ ASP นี้แทบจะไม่ต่างจากการทำงานของโปรแกรม CGI (Common Gateway Interface) จนอาจกล่าวได้ว่า ASP ก็เป็นโปรแกรม CGI ประเภทหนึ่งเช่นกัน

ไฟล์ ASP นั้น แตกต่างจากไฟล์ HTML ทั่วไปตรงที่มีส่วนของคำสั่ง ASP อยู่ในไฟล์ด้วย โดยทั่วไปเราสามารถนำไฟล์ HTML มาเปลี่ยนเป็นไฟล์ ASP ได้เลย นั่นคือ วิธีการสร้างเอกสาร ASP อย่างง่าย ๆ เช่น เราสามารถเปลี่ยนเอกสาร HTML ที่ชื่อ default.htm ไปเป็น default.asp ได้เลย โดยที่เมื่อโปรแกรม ASP ตีความส่วนใดของเอกสารที่มีแท็ก HTML กำกับอยู่ก็จะไม่เกิดความเปลี่ยนแปลงใด ๆ ในเอกสารนั้นเลย แต่หากว่าส่วนใดมีแท็ก ASP กำกับอยู่ ASP ก็จะเปลี่ยนเอกสารส่วนดังกล่าวไปอยู่ในรูปของแท็ก HTML แทน เช่น หากในเอกสารมีคำสั่งดังนี้

```
<br><% response.write("Hello " & now) %>
```

ก็จะถูกเปลี่ยนเป็น

```
<br>Hello แล้วตามด้วยวันเวลาปัจจุบัน
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 Built-in Objects ของ Active Server Pages

ASP ได้สร้างออบเจกต์มาตรฐานสำหรับใช้งานทั่วไปเอาไว้ 5 ออบเจกต์ดังนี้

- **Request** ใช้ในการรับข้อมูลจากผู้ใช้
- **Response** ใช้ในการส่งข้อมูลให้ผู้ใช้
- **Server** ใช้ในการควบคุม Internet Information Server
- **Session** ใช้ในการเก็บข้อมูลและการกำหนดค่าของเซสชันปัจจุบันของผู้ใช้
- **Application** ใช้ในการใช้ข้อมูลระดับแอปพลิเคชันร่วมกันและควบคุมการกำหนดค่าของ

แอปพลิเคชันตลอดการใช้งาน

ออบเจกต์ Request และ Response จะมีลักษณะเป็นคอลเลกชัน (บิทของข้อมูลที่ถูกเข้าใช้โดยวิธีการเดียวกัน) ออบเจกต์ใช้เม็ทโอดในการทำโพธิเซอร์ (ถ้าคุณรู้จักภาษาที่ใช้เขียน โปรแกรมแบบออบเจกต์ คุณก็จะรู้ว่าเม็ทโอด คือ อะไร) และใช้พร็อพเพอร์ตี้ในการเก็บคุณสมบัติของออบเจกต์ (เช่น สีสันหรือขนาด)

ออบเจกต์ Request

ออบเจกต์ Request ถูกใช้ในการรับข้อมูลจากผู้ใช้ซึ่งถูกส่งมาด้วยการร้องขอ HTTP โดยภายในออบเจกต์ Request จะมีลักษณะเป็นคอลเลกชันซึ่งประกอบด้วย

- **ClientCertificate** ใช้รับ certification fields จากการร้องขอที่ส่งโดยเบราว์เซอร์ ไฟล์ที่สามารถร้องขอได้ได้ระบุไว้ในมาตรฐาน X.509
- **QueryString** ใช้รับข้อความ เช่น ชื่อ นามสกุลและที่อยู่ เป็นต้น
- **Form** ใช้รับข้อมูลจากฟอร์มในไฟล์ HTML
- **Cookies** ใช้กำหนดค่า cookie ของแอปพลิเคชัน
- **ServerVariables** ใช้รับข้อมูล HTTP เช่น ชื่อของเครื่องเซิร์ฟเวอร์ เป็นต้น

ออบเจกต์ Response

ออบเจกต์ Response ใช้ในการส่งข้อมูลไปยังผู้ใช้ ออบเจกต์ Response สนับสนุน cookie ที่เป็นคอลเลกชัน (ใช้ในการกำหนดค่าของ cookie) นอกจากนี้ออบเจกต์ Response ยังสนับสนุนพร็อพเพอร์ตี้และเม็ทโอดจำนวนมาก พร็อพเพอร์ตี้ของออบเจกต์ Response มีดังนี้

- **Buffer** ใช้กำหนดบัฟเฟอร์ของการแสดงผลของเว็บเพจที่เซิร์ฟเวอร์ เมื่อค่าถูกกำหนดเป็น true เซิร์ฟเวอร์จะไม่แสดงผลจนกว่าสคริปต์ทั้งหมดจะถูกประมวลผลเสร็จหรือเม็ทโอด Flush หรือ End ถูกเรียกใช้

- **ContentType** ใช้กำหนดชนิดของข้อมูล (ยกตัวอย่างเช่น text/HTML และ Excel เป็นต้น)
- **Expires** ใช้กำหนดเวลาหมดอายุเป็นนาที
- **ExpiresAbsolute** ใช้กำหนดวันและเวลาการหมดอายุ
- **Status** แสดงสถานะการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เม็ททอดของออบเจกต์ Response มีดังนี้

- **AddHeader** ทำการใส่ HTML header ด้วยค่าที่กำหนด
- **AppendToLog** ทำการต่อข้อความที่ส่วนท้ายของล็อกไฟล์ (Log file) ของเซิร์ฟเวอร์
- **BinaryWrite** ใช้เขียนข้อมูลประเภทไบนารี เช่น ข้อมูล Excel spreadsheet
- **Clear** ใช้ลบบัฟเฟอร์ของการแสดงผล HTML ทั้งหมด
- **End** ใช้หยุดการประมวลผลของสคริปต์
- **Flush** ใช้ส่งข้อมูลทั้งหมดที่อยู่ในบัฟเฟอร์
- **Redirect** ให้นำผู้ใช้ไปยัง URL อื่น
- **Write** ใช้เขียนข้อความลงในเว็บเพจโดยใช้โครงสร้างดังนี้

Response.write("hello") หรือ <%="hello"%>

ออบเจกต์ Server

ออบเจกต์ Server มีพร็อพเพอร์ตี้เพียงพร็อพเพอร์ตี้เดียว คือ ScriptTimeout ซึ่งใช้กำหนดเวลาในการประมวลผลสคริปต์ (Time out) และมี เม็ททอด ดังนี้

- **CreateObject** ใช้สร้างอินสแตนซ์ของคอมโพเนนต์ใด ๆ ที่ทำการติดตั้งไว้บนเครื่องเซิร์ฟเวอร์ เช่น ActiveX
- **HTMLEncode** ใช้ในการเข้ารหัสข้อความใน HTML
- **MapPath** ใช้แม็ป Virtual path ปัจจุบันไปยัง Physical directory เพื่อใช้ path นั้นโดยคอมโพเนนต์ในการสร้างไดเรกทอรีหรือไฟล์บนเครื่องเซิร์ฟเวอร์
- **URLEncode** ทำการเข้ารหัสชื่อ URL

ออบเจกต์ Session

ออบเจกต์ Session ใช้ในการเก็บข้อมูลและการกำหนดค่าของเซสชันปัจจุบันของผู้ใช้ ตัวแปรที่เก็บไว้กับออบเจกต์นี้จะยังคงอยู่นานเท่าที่เซสชันของผู้ใช้ยังคงทำงานอยู่ถึงแม้ว่าจะมีหลายแอปพลิเคชันใช้งานอยู่ ออบเจกต์นี้มีเม็ททอดเพียงเม็ททอดเดียว คือ เม็ททอด Abandon ซึ่งใช้ในการทำลายเซสชันของเว็บเซิร์ฟเวอร์ ทำลายออบเจกต์ใด ๆ และมีพร็อพเพอร์ตี้ 2 พร็อพเพอร์ตี้ คือ SessionID ใช้เก็บ ID ของเซสชันปัจจุบันและ Timeout ใช้กำหนดเวลาหมดอายุของเซสชัน สิ่งหนึ่งที่สำคัญก็คือ ID ไม่ใช่ GUID ถ้าปิดบริการเว็บเซิร์ฟเวอร์ค่า ID จะเริ่มนับใหม่ทั้งหมด ดังนั้นไม่ควรใช้ SessionID เป็นชื่อในการ logon เพราะจะเกิดปัญหาเนื่องตัวเลขซ้ำ

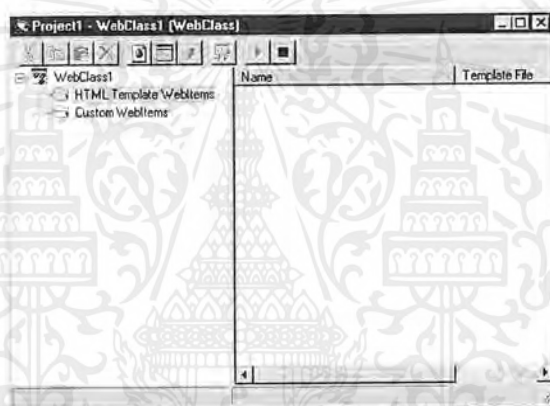
ออบเจกต์ Application

ออบเจกต์ Application สามารถใช้ในการเก็บข้อมูลที่ไม่สูญหายถ้าแอปพลิเคชันยังทำงานอยู่ โดยปกติก็คือ ถ้า IIS server ยังทำงานอยู่ นอกจากนี้ยังใช้ในการเก็บข้อมูลที่ผู้ใช้หลาย ๆ คน เช่น เคาท์เตอร์ เนื่องจากออบเจกต์จะไม่ถูกสร้างขึ้นใหม่สำหรับผู้ใช้แต่ละคน ดังนั้นความคิดพลาดที่จะไม่เกิดเมื่อโค้ดถูกเรียกใช้เพียงครั้งเดียว จะเกิดขึ้นเมื่อถูกเรียกใช้มากกว่า 10,000 ครั้งขึ้นไป นอกจากนี้เนื่องจากเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

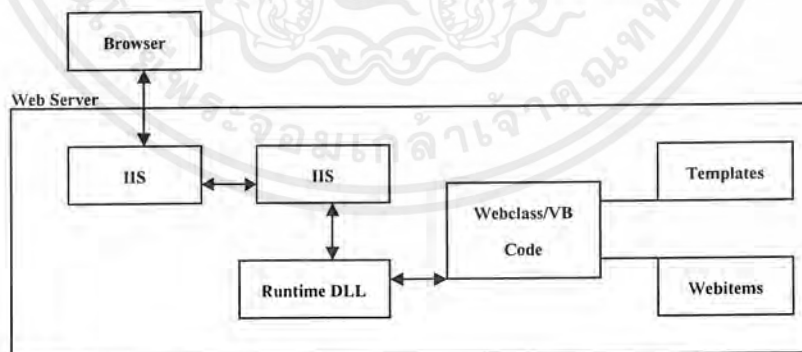
ออบเจ็กต์ Application จะถูกใช้งานร่วมกันโดยผู้ใช้ทุกคน ดังนั้นการเขียนโปรแกรมควบคุมเทรดดิ้ง (Threading) จะทำได้ยาก

5.4 IIS Applications

แอปพลิเคชัน IIS จะประมวลผลที่เครื่องเว็บเซิร์ฟเวอร์ ซึ่งจะตอบสนองการร้องขอจากเบราว์เซอร์ อินเทอร์เน็ตของแอปพลิเคชัน IIS ก็คือ HTML ซึ่งอาจจะรวมถึง DHTML (Dynamic Hypertext Markup Language), โพรเจ็กต์ DHTML, ActiveX Control และเทคโนโลยีอื่น ๆ ที่เบราว์เซอร์สนับสนุน ดังรูป 5-1 ผู้ใช้ไม่ได้ทำงาน โดยตรงกับแอปพลิเคชัน IIS เพียงแต่คลิกที่ลิงค์แทน ผู้ใช้จะทำงานโต้ตอบกับเว็บเพจราย ได้เบราว์เซอร์ ซึ่งส่งการร้องขอไปยังแอปพลิเคชัน IIS บนเครื่องเซิร์ฟเวอร์ เมื่อเซิร์ฟเวอร์ได้รับการร้องขอ ก็จะส่งการตอบสนองกลับมา จากนั้นเบราว์เซอร์ก็จะรับการตอบสนองนั้นแล้วแสดงผลเป็นเว็บเพจใหม่ขึ้นมา แอปพลิเคชัน IIS เป็นแอปพลิเคชันที่ไม่ขึ้นอยู่กับเบราว์เซอร์ โดยสถาปัตยกรรมของแอปพลิเคชัน สามารถแสดงได้ดังรูป 5-2



รูปที่ 5-1 แอปพลิเคชัน IIS กับ Default Webclass



รูปที่ 5-2 สถาปัตยกรรมของแอปพลิเคชัน IIS

เว็บคลาส (WebClass)

เว็บคลาสเป็นคอมโพเนนต์ของ VB ที่ถูกแสดงโดย DLL ของแอปพลิเคชัน IIS ที่คุณสร้างขึ้น โดยเว็บคลาสจะอยู่บนเครื่องเซิร์ฟเวอร์และคอยตอบสนองต่อการร้องขอของเบราว์เซอร์ แอปพลิเคชัน IIS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถมีเว็บคลาสได้มากกว่า 1 เว็บคลาส โดยเว็บคลาสหนึ่งจะทำงานกับผู้ใช้เพียงคนเดียวตลอดช่วงการทำงานของมัน โดยปกติเว็บคลาสจะบรรจุเว็บไอเท็ม (Webitems) และโค้ดที่จะนำเว็บไอเท็มไปให้กับผู้ใช้เว็บไอเท็มนั้นจริง ๆ แล้วก็คือ เพจ HTML แต่ก็สามารถเป็นไฟล์ประเภท MIME ได้ เช่น ไฟล์ภาพหรือไฟล์ .wav

VB จะทำการสร้างอินสแตนซ์ของเว็บคลาสสำหรับไคลเอนท์แต่ละคนที่ทำการเข้าใช้มัน อย่างไรก็ตาม เว็บคลาสก็สามารถที่จะรักษาสถานะของการร้องขอเอาไว้ได้ เราสามารถกำหนดให้เว็บคลาสทำงานอยู่นอกว่าเซสชันของผู้ใช้จะถูกทำลายหรือให้ทำลายทันทีเมื่อเว็บคลาสส่งการตอบสนองมายังไคลเอนท์แล้วโดยการกำหนดค่าของพร็อพเพอร์ตี้ StateManagement ของเว็บคลาสเป็น wcNoState หรือ wcRetainInstance (ถ้าเว็บคลาสถูกกำหนดให้ทำงานอยู่ตลอดระหว่างการร้องขอ ผู้ใช้จะต้องมีเบรเซอร์ที่สนับสนุนการใช้ cookie และเปิดให้ใช้งาน cookie ได้

การเรียกใช้เว็บคลาส

สำหรับแต่ละเว็บคลาสที่ถูกสร้างขึ้น VB จะทำการสร้างเพจ ASP ขึ้นมา ผู้ใช้เริ่มต้นการใช้งานเว็บคลาสโดยการร้องขอเพจ ASP โดยกำหนด URL เช่น <http://161.246.6.152/ecom/main.asp> จากนั้นเพจ ASP ก็จะสร้างอินสแตนซ์ของเว็บคลาสและร้องขอไปยังเว็บคลาสเพื่อให้เว็บคลาสทำการประมวลผลการร้องขอของผู้ใช้ต่อไปเป็นตัวอย่างของเพจ ASP ที่สร้างโดย VB

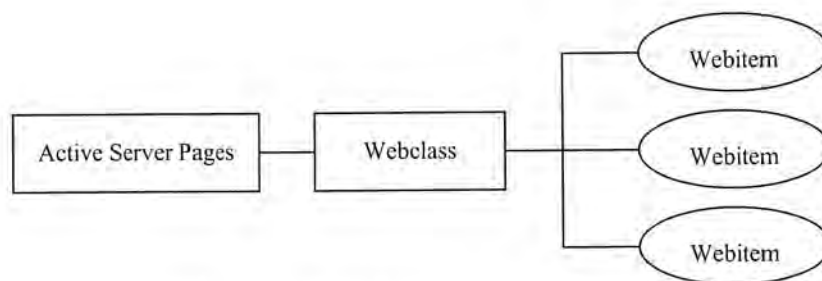
```
<%
Response.Buffer=True
Response.Expires=0

If (VarType(Application("~/WC~WebClassManager")) = 0) Then
    Application.Lock
    If (VarType(Application("~/WC~WebClassManager")) = 0) Then
        Set Application("~/WC~WebClassManager") = Server.CreateObject
            ("WebClassRuntime.WebClassManager")
    End If
    Application.Unlock
End If

Application("~/WC~WebClassManager").ProcessNoStateWebClass "Shop2000.Main", _
    Server, _
    Application, _
    Session, _
    Request, _
    Response
%>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสัมพันธ์ของเพจ ASP เว็บคลาสและเว็บไอเท็มสามารถแสดงได้ดังรูป 5-3 โดยแต่ละเว็บคลาสจะเกี่ยวข้องกับเพจ ASP



รูปที่ 5-3 ความสัมพันธ์ระหว่าง ASP และเว็บคลาส

การทำงานกับเท็มเพลต

สมมติว่าผู้ใช้ทำการร้องขอมายังเว็บคลาสให้ทำการส่งเพจ HTML กลับมาโดยที่มีข้อมูลที่ผู้ใช้ระบุ เช่น

- ชื่อผู้ใช้
- ลิสต์ของบริษัทที่ผู้ใช้สนใจ
- ลิสต์ของเหตุการณ์ที่เกิดขึ้นกับแต่ละบริษัท
- ลิสต์ของงานสำหรับผู้ใช้

วิธีหนึ่งที่ใช้ในการทำงานนี้ คือ ทำการรวบรวมข้อมูลแล้ว สร้างไฟล์ HTML ขึ้นเพื่อส่งไปให้ผู้ใช้ วิธีนี้เป็นวิธีการง่าย ๆ แต่จะทำให้ไฟล์ HTML ของคุณเต็มไปด้วยข้อความมากมาย ทำให้เวลาที่ต้องแก้ไขเว็บเพจนั้นชุลมุนยากมาก งานง่าย ๆ เช่น การเพิ่ม ไฟล์ภาพ .GIF จะต้องทำการแก้ไขโค้ด VB และทำการคอมไพล์ใหม่ ทางที่ดีกว่า คือ ทำการแยกส่วน HTML ไว้ในเท็มเพลตต่างหาก เรียกว่า เท็มเพลต แล้วทำการแทรกจุดเปลี่ยนข้อมูล (Placeholder) ไว้สำหรับข้อมูลที่ต้องการ จุดเปลี่ยนข้อมูลสามารถเป็นแท็กใด ๆ ก็ได้ที่คุณกำหนด ยกตัวอย่างเช่น \$\$USERNAME\$\$ หรือ PUTUSERNAMEHERE เป็นต้น จากนั้นก็ทำการเขียนรูทีน (Routine) ที่ทำการเปิดไฟล์ขึ้นอ่านทีละบรรทัดแล้วตรวจหาจุดเปลี่ยนข้อมูล ถ้าหาจุดเปลี่ยนข้อมูลเจอก็ให้ทำการเปลี่ยนด้วยข้อมูลที่แท้จริง เช่น ชื่อผู้ใช้ จากนั้นก็ส่งทุกบรรทัดไปให้กับเบราว์เซอร์ จะเห็นได้ว่าวิธีการนี้เป็นวิธีการที่ดีกว่าเพราะว่าสามารถเปลี่ยนแปลงเท็มเพลตได้โดยไม่กระทบต่อโค้ด VB

เท็มเพลต DCOM ก็เหมือนกับเท็มเพลตอื่น ๆ แต่มีข้อดีตรงที่ไม่ต้องเขียนโค้ดในการอ่านไฟล์ เท็มเพลตและตรวจหาจุดเปลี่ยนข้อมูลเอง คุณเพียงใช้ชื่อของจุดเปลี่ยนข้อมูล เช่น Shopname จากนั้นทำการเรียกเม็ทธอด WriteTemplate ของออบเจกต์เท็มเพลต จากนั้น Webclass manger จะทำการตรวจหาจุดเปลี่ยนข้อมูลในเท็มเพลตแล้วสร้างอีเวนต์

```

Templatenam_ ProcessTag(Byval TagName As String, TagContents As String _
, SendTags As Boolean)
  
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อีเวนต์นี้ทำให้เราทราบว่าจุดเปลี่ยนข้อมูลอันไหนที่จะต้องประมวลผล (TagContents) และคุณสามารถเปลี่ยนข้อมูลที่เกี่ยวข้องได้และข้อมูลนี้จะถูกส่งไปยังเบราว์เซอร์ ซึ่งลักษณะนี้จะทำให้การพัฒนาให้สำเร็จและบำรุงรักษาเพจ HTML แบบไดนามิกนั้นทำได้ง่าย

ต่อไปเป็นตัวอย่างของอีเวนต์ ProcessTag

```
Select Case TagContents
```

```
Case "ShopName"
```

```
TagContents = rs!shopName
```

```
Case "Name"
```

```
TagContents = rs!ownername
```

```
Case "Surname"
```

```
TagContents = rs!ownersurname
```

```
End Select
```

เพิ่มเพลตเป็นจุดสำคัญที่ทำให้เราสามารถแยกข้อมูลกับรูปแบบออกจากกัน เมื่อคุณทำการเพิ่มเพิ่มเพลตขึ้น เช่น Mytemplate.htm VB จะทำการ copy เอาไว้แล้วตั้งชื่อว่า Mytemplate.htm แล้วจะถูกรวมเอาไว้ในโปรเจกต์ หากต้องการแก้ไขเพิ่มเพลตนอก VB ต้องมั่นใจว่าได้แก้ไขไฟล์ก็อปปีไม่ใช่ไฟล์ต้นแบบ

Custom WebItems

Custom Webitem คือคอนเทนเนอร์ที่เก็บโค้ดของโปรซีเจอร์ที่ต้องการให้เว็บคลาสสามารถใช้จากจุดใดก็ได้ของแอปพลิเคชัน Custom Webitem เป็นวิธีการที่ดีในการเอ็นแคปซูลชิ้นโค้ดของแอปพลิเคชัน IIS ที่ต้องทำการเรียกใช้บ่อย เช่น ส่วนหัวของเพจหรือตาราง เป็นต้น นอกจากนี้ Custom Webitem ยังสามารถส่งการตอบสนองไปยังเบราว์เซอร์ถึงแม้ว่าเพิ่มเพลตจะไม่สามารถทำงานได้และช่วยให้การเขียนโปรแกรมมีลักษณะเป็นโมดูลขึ้น

ในการเรียกใช้สามารถทำได้โดยใช้ฟังก์ชัน URLFor ในการสร้าง URL ไปยังอีเวนต์นั้น ยกตัวอย่างเช่น

```
Response.Write "<A HREF="" &
```

```
URLFor(CustomWebItem1,"CustomEventName") & ""></A>"
```

แอปพลิเคชัน IIS และฟอร์มของ VB

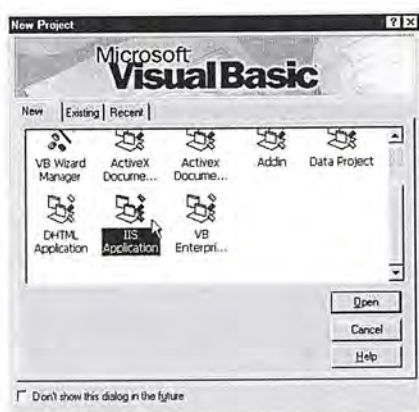
แม้ว่าแอปพลิเคชัน IIS จะเป็นแอปพลิเคชัน VB ชนิดหนึ่ง แต่ก็ไม่สามารถรวมฟอร์มของ VB เข้าไปด้วย เพราะไม่สามารถแสดงผลมันบนเครื่องเซิร์ฟเวอร์ได้ เราสามารถทำได้เพียงส่งการตอบสนองเป็น HTML ต่อการร้องขอของเบราว์เซอร์เท่านั้น อย่างไรก็ตามเราสามารถรวมเพจ DHTML เข้าไปในแอปพลิเคชัน IIS ได้

5.5 ตัวอย่างการสร้างแอปพลิเคชัน IIS

ในการสร้างแอปพลิเคชัน IIS นั้นมีขั้นตอนในการสร้างดังนี้

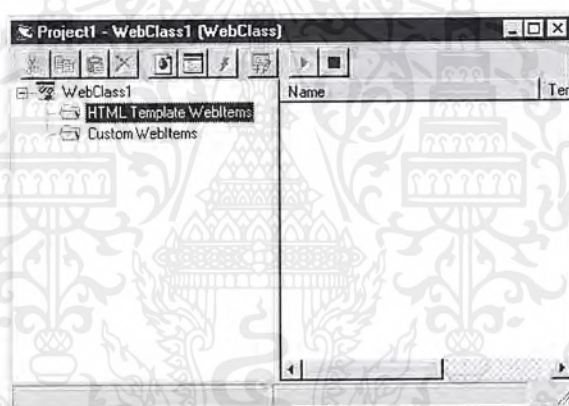
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ทำการเปิดโปรแกรม Visual Basic ขึ้นมาแล้วเลือกโปรเจกต์เป็น IIS Application ดังรูป 5-4



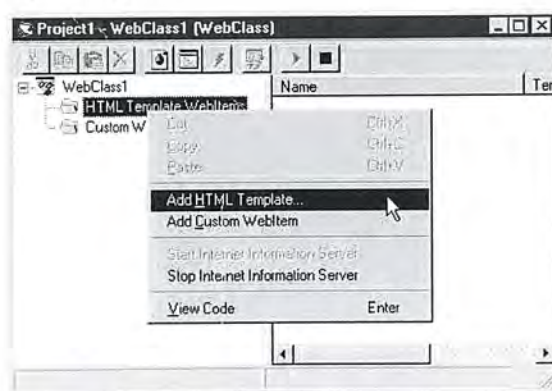
รูปที่ 5-4 หน้าจอเริ่มต้นการสร้าง IIS Application

2. Visual Basic จะกำหนด Webclass Designer มาให้เพื่อใช้ในการจัดการเกี่ยวกับการทำงานของเว็บคลาส ดังรูป 5-5



รูปที่ 5-5 Webclass Designer

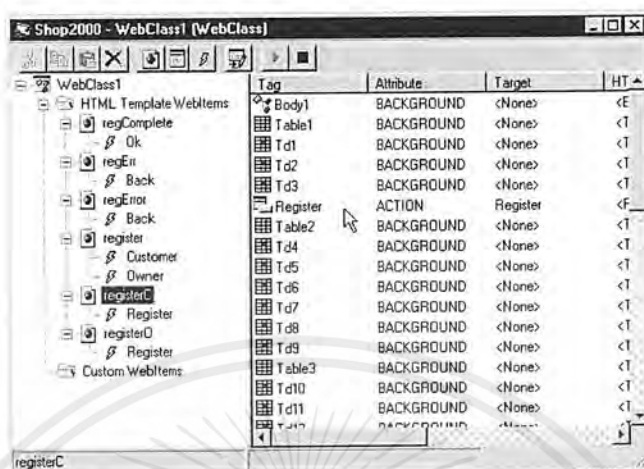
3. จากนั้นทำการเพิ่มเพิ่มเพลตเข้าไปในโปรเจกต์โดยคลิกขวาที่ HTML Template WebItems แล้วเลือกเมนู Add HTML Template ดังรูป 5-6



รูปที่ 5-6 ทำการเพิ่มเพิ่มเพลตเข้าไปในโปรเจกต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. หลังจากที่เพิ่มเพิ่มเพลตเข้าไปในโปรเจกต์แล้ว หน้าต่างทางขวาของ Webclass Designer จะแสดงรายละเอียดของแต่ละเพิ่มเพลต เช่น อีเว้นต์ของฟอร์มในเพิ่มเพลต จากนั้นทำการคลิกที่แอกชั่นทางหน้าต่างฝั่งขวาเพื่อเขียนโค้ดควบคุมการทำงาน ดังรูป 5-7



รูปที่ 5-7 อีเว้นต์ของฟอร์มในเพิ่มเพลต

5. จากนั้นทำการเขียนโค้ดเพื่อทำการควบคุมการทำงานของแอปพลิเคชัน IIS ดังรูป 5-8

```

Shop2000 - WebClass1 [Code]
registerC
register

Private Sub registerC_register()
Dim inf As Object
On Error GoTo errorRegister
If Request.Form("password") <> Request.Form("cpassword") Then
regErr.WriteTemplate 'Password is not duplicated
Exit Sub
End If
With Request
If .Form("Name") = "" Or .Form("Surname") = "" Or .Form("Age"
Or .Form("Addr") = "" Or .Form("SubProvince") = "" Or .For
Or .Form("Postcode") = "" Or .Form("Phone") = "" Or .Form(
Or .Form("Email") = "" Or .Form("Password") = "" Or .Form(
regErr.WriteTemplate ' Error
Exit Sub
End If
If .Form("Country") = "Please choose a country" Or .Form("Occ
regErr.WriteTemplate ' Error
Exit Sub
End If
End With

Set inf = Server.CreateObject("Ecomm.account")
With Request
inf.cusname = .Form("Name")
inf.cussurname = .Form("Surname")
inf.email = .Form("Email")

```

รูปที่ 5-8 โค้ดควบคุมการทำงานส่วนริจิสเตอร์ของลูกค้า

6. หลังจากเขียน โปรแกรมเสร็จแล้วก็ทำการคอมไพล์โปรแกรมเพื่อสร้าง DLL ขึ้นมา ดังรูป 5-9

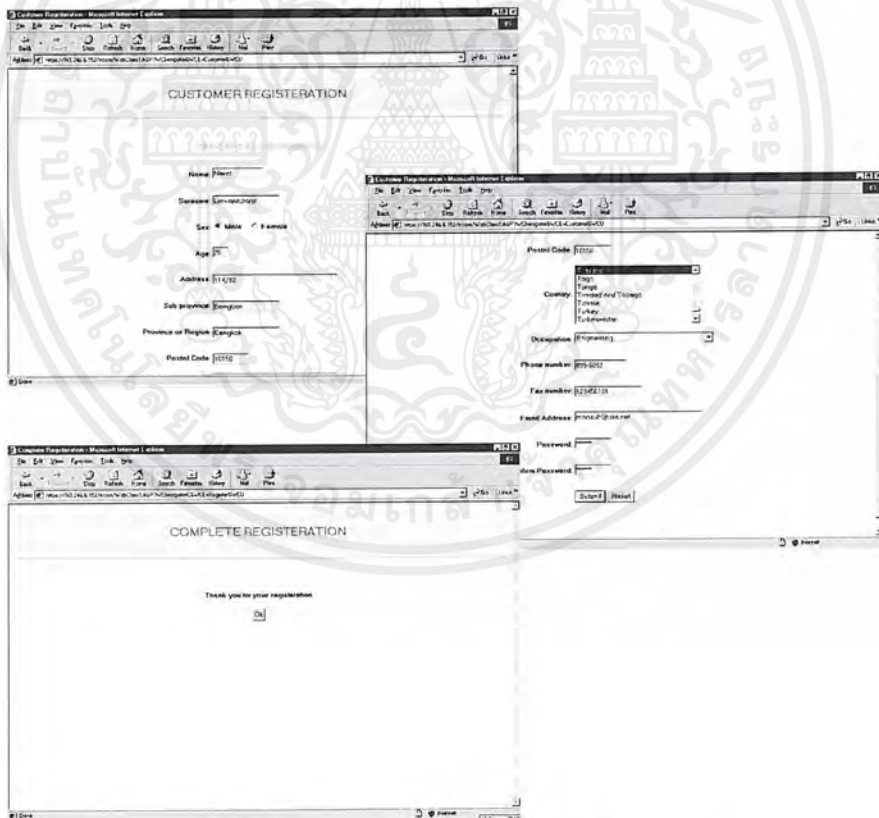
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-9 การสร้าง DLL ของ IIS Application

5.6 การทดสอบแอปพลิเคชัน IIS

ในการทดสอบแอปพลิเคชัน IIS สามารถทำได้โดยกดปุ่ม F5 เพื่อทำการรันด้วยเบราว์เซอร์ หรือทดสอบโดยการเข้าใช้เว็บผ่านเว็บเซิร์ฟเวอร์ ดังรูป 5-10 และ 5-11



รูปที่ 5-10 การทดสอบแอปพลิเคชัน IIS ที่ทำการสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

ระบบความปลอดภัยและ Certificate Server (Security System and Certificate Server)

6.1 เทคโนโลยีการเข้ารหัสข้อมูลในอินเทอร์เน็ต

เทคนิคการเข้ารหัสข้อมูลในปัจจุบันนี้แบ่งออกได้ตามประเภทของกุญแจรหัสที่ใช้เป็น 2 พวกใหญ่ ๆ พวกแรกใช้เทคนิคของกุญแจรหัสที่เรียกว่า Secret Key หรือบางทีเรียกกันว่า Symmetric Key ส่วนพวกที่ 2 จะใช้เทคนิค Public Key

6.1.1 เทคนิคการเข้ารหัสแบบ Secret Key

เทคนิคการเข้ารหัสแบบ Secret Key เกิดขึ้นก่อนเทคนิคการเข้ารหัสแบบ Public Key โดยใช้หลักการที่ว่าผู้รับและผู้ส่งจะถือกุญแจรหัสดอกเดียวกัน โดยที่ผู้ส่งจะใช้กุญแจรหัสเพื่อเข้ารหัสข้อมูลก่อนส่ง และผู้รับจะต้องใช้กุญแจรหัสที่มีรูปแบบเดียวกันสำหรับถอดรหัสข้อมูล

ข้อเสียของการเข้ารหัสแบบ Secret Key อยู่ตรงที่ตัวกุญแจรหัส เนื่องจากผู้รับและผู้ส่งใช้กุญแจรหัสดอกเดียวกันนั้นหมายความว่าถ้ามีคนอื่น ซึ่งถือกุญแจรหัสแบบเดียวกันนี้อยู่ ก็สามารถที่จะถอดรหัสข้อมูลได้อย่างง่ายดาย และยิ่งถ้ากุญแจจะต้องถูกส่งไปยังผู้รับข้อมูลปลายทางเพื่อใช้ในการถอดรหัสข้อมูล โดยส่งผ่านเครือข่ายอินเทอร์เน็ตด้วยแล้ว โอกาสที่กุญแจรหัสจะถูกลักลอบนำไปใช้ถอดรหัสข้อมูลโดยบุคคลอื่นก็ยิ่งมีความเสี่ยงสูง จากข้อบกพร่องนี้เองจึงได้มีการพัฒนาเทคนิคการเข้ารหัสมาเป็น การเข้ารหัสแบบ Public Key ที่ผู้ส่งและผู้รับข้อมูลจะถือกุญแจรหัสคนละดอกที่ไม่เหมือนกันในการเข้ารหัสและถอดรหัสข้อมูล

6.1.2 เทคนิคการเข้ารหัสแบบ Public Key

เทคนิคการเข้ารหัสแบบ Public Key มีหลักการคือ การใช้กุญแจรหัสคนละดอกระหว่างผู้รับและผู้ส่งข้อมูล กุญแจรหัสที่ได้จากโปรแกรมการสร้างกุญแจรหัสด้วยวิธีนี้จะมีด้วยกัน 2 ชุด ชุดแรกเรียกว่า ไพรเวทคีย์ (Private Key) ส่วนอีกชุดหนึ่งเรียกว่า พับบลิคคีย์ (Public Key) ชุดที่เป็น ไพรเวทคีย์จะไม่ถูกส่งผ่านไปทางเครือข่ายหรือช่องสัญญาณอื่น ๆ ในขณะที่ชุดที่ถูกส่งไปทางเครือข่าย หรือช่องสัญญาณติดต่ออื่น ๆ นั้นจะเป็นกุญแจพับบลิคคีย์

ข้อดีของการเข้ารหัสแบบนี้คือเมื่อมีการลักลอบขโมยกุญแจรหัสแบบพับบลิคคีย์ จะไม่สามารถนำไปถอดรหัสข้อมูลที่ถูกรหัสด้วยกุญแจพับบลิคคีย์ได้ต้องใช้กุญแจไพรเวทคีย์เท่านั้น การเข้ารหัสด้วยเทคนิคพับบลิคคีย์นี้จึงมีความปลอดภัยสูงตรงไปตรงมาที่กุญแจอีกดอกหนึ่งที่เป็นไพรเวทคีย์ไม่ถูกส่งไปทางเครือข่าย ข้อมูลที่ถูกรหัสด้วยกุญแจดอกหนึ่งจะต้องใช้กุญแจอีกดอกหนึ่งในการถอดรหัสเสมอ

ข้อเสียในการเข้ารหัสแบบพับบลิคคีย์คือ จะช้ากว่าการเข้ารหัสแบบซีเคร็ทคีย์ คือประมาณตั้งแต่ 100 – 1,000 เท่า เนื่องจากความซับซ้อนของเทคนิคการเข้ารหัสที่ต้องสร้างกุญแจสองชุดให้สัมพันธ์กันนั่นเอง

6.1.3 เทคนิคการเข้ารหัสแบบผสมระหว่างซีเคร็ทคีย์กับพับบลิคคีย์

จากจุดบกพร่องที่มีอยู่ในแต่ละประเภทของการเข้ารหัสจึงมีแนวความคิดในการรวมเทคนิคการเข้ารหัสทั้งสองแบบเข้าด้วยกัน

เทคนิคการเข้ารหัส คือ ใช้ซีเคร็ทคีย์ในการเข้ารหัสตัวเนื้อข้อมูล ข้อมูลที่ถูกเข้ารหัสจะถูกส่งไปยังผู้รับพร้อมกุญแจรหัสซีเคร็ทคีย์โดยที่ทั้งตัวข้อมูลที่ถูกเข้ารหัสด้วยซีเคร็ทคีย์และตัวซีเคร็ทคีย์เองจะถูกเข้ารหัสด้วยอีกชั้นหนึ่งด้วยกุญแจรหัสแบบพับบลิคคีย์ก่อนที่จะส่งไปยังผู้รับและเมื่อผู้รับได้รับข้อมูลพร้อมกุญแจรหัสซีเคร็ทคีย์ที่ถูกเข้ารหัสซ่อนด้วยกุญแจรหัสแบบพับบลิคคีย์ ผู้รับจะใช้กุญแจรหัสแบบไพรเวทคีย์ที่ตัวเองเก็บไว้ในการถอดรหัส

จะเห็นได้ว่าการส่งข้อมูลโดยการเข้ารหัสผสมแบบนี้ เราสามารถส่งกุญแจซีเคร็ทคีย์ไปทางเครือข่ายได้ โดยการใส่กุญแจรหัสพับบลิคคีย์เข้ารหัสซึ่งจะลดจุดด้อยจากการที่จะต้องส่งกุญแจรหัสซีเคร็ทคีย์อย่างเปิดเผยไปทางเครือข่ายทำให้โอกาสในการลักลอบขโมยกุญแจรหัสซีเคร็ทคีย์แทบจะเป็นไปไม่ได้เลย

6.2 คุณสมบัติพื้นฐานของระบบรักษาความปลอดภัยข้อมูล

คุณสมบัติของระบบรักษาความปลอดภัยข้อมูลจะประกอบด้วย 4 ข้อด้วยกัน คือ

1. Authentication
2. Integrity
3. Non – repudiation
4. Confidentiality

6.2.1 Authentication

Authentication เป็นกระบวนการระบุตัวผู้ส่ง หรือใช้ยืนยันว่าข้อมูลที่ถูกส่งไปยังผู้รับนั้นจะต้องไม่ถูกเปลี่ยนแปลงแก้ไข กระบวนการ Authentication มีไว้เพื่อป้องกันการปลอมแปลงตัวผู้ส่งหรือข้อมูลที่จะส่งเพื่อให้ผู้รับแน่ใจว่าข้อมูลที่ได้รับนั้นมาจากตัวผู้ส่งตัวจริง หรือไม่ถูกดัดแปลงแก้ไขไปจากข้อมูลเดิม

กระบวนการ Authentication จะใช้เทคนิคการเข้ารหัสข้อมูลโดยใช้กุญแจรหัสแบบพับบลิคคีย์เพื่อสร้างลายเซ็นดิจิทัลหรือ Digital Signature ในการระบุตัวผู้ส่งและยืนยันข้อมูลต้นฉบับโดยกุญแจดอกที่เป็นไพรเวทคีย์จะใช้สร้างลายเซ็นและกุญแจดอกที่เป็นพับบลิคคีย์จะใช้ยืนยันลายเซ็น

ลายเซ็นดิจิทัลเป็นชิ้นเอกสารอิเล็กทรอนิกส์ที่ถูกเข้ารหัสและแนบไปพร้อมกับเอกสารที่เป็นข้อมูล ตัวข้อมูลเองจึงไม่จำเป็นต้องถูกเข้ารหัสเพื่อลงลายเซ็นดิจิทัลนี้ แต่จะถูกเข้ารหัสเพื่อจุดประสงค์เรื่องความปลอดภัยของตัวข้อมูลเอง เพราะฉะนั้นลายเซ็นดิจิทัลจะไม่มีผลทำให้ข้อมูลมีการเปลี่ยนแปลง

6.2.2 Integrity

ข้อมูลที่ถูกส่งออกมาจากผู้ส่งและถึงผู้รับ โดยไม่มีการเปลี่ยนแปลงคือคุณสมบัติที่เรียกว่า Integrity เนื่องจากข้อมูลต้องมีการส่งผ่านเครือข่ายอินเทอร์เน็ตที่เป็นเครือข่ายสาธารณะจึงต้องมีกระบวนการที่ช่วยให้เกิดความมั่นใจว่าในระหว่างทางข้อมูลจะต้องไม่ถูกเปลี่ยนแปลงแก้ไข หรือในกรณีที่ข้อมูลถูกเปลี่ยนแปลงไปแล้วจะต้องมีวิธีที่ตรวจสอบได้เพื่อจะได้ไม่ต้องเชื่อถือข้อมูลนั้นอีกต่อไป วิธีที่ใช้ตรวจสอบความถูกต้องของข้อมูลก็คือ การใช้ลายเซ็นดิจิทัล

เนื่องจากการสร้างลายเซ็นดิจิทัลนั้นเราใช้แฮชโคด (Hash Code) ที่สร้างมาจากตัวข้อมูลจริงจากผู้ส่งเมื่อผู้รับได้รับข้อมูลจะทำการสร้าง แฮชโคด ใหม่ขึ้นมาจากข้อมูลที่รับเพื่อทำการเปรียบเทียบ ซึ่งหากข้อมูลไม่ได้ถูกแก้ไขค่าแฮชโคดจะต้องเหมือนกัน ในทางกลับกันถ้าข้อมูลถูกแก้ไข ค่าที่ได้จะไม่เหมือนกัน

แฮชโคดจะถูกสร้างจากฟังก์ชันแฮช ที่เป็นการคำนวณทางคณิตศาสตร์ที่ซับซ้อน จากข้อมูลในลักษณะทิศทางเดียวคือ เราไม่สามารถใช้ฟังก์ชันแฮชเดียวกันนี้กระทำกับแฮชโคด เพื่อให้ได้มาซึ่งข้อมูลเดิมในลักษณะย้อนกลับ ฟังก์ชันแฮชไม่สามารถสร้างแฮชโคดที่ซ้ำกันจากข้อมูลที่ต่างกัน ได้จึงทำให้ลายเซ็นดิจิทัลจากแฮชโคดมีความปลอดภัยและเชื่อถือได้ว่าข้อมูลจะไม่ถูกเปลี่ยนแปลงแก้ไขโดยที่เราไม่ทราบ

ตัวอย่างการใช้ลายเซ็นดิจิทัล

- นาย ก เขียนใบเบิกเงินอิเล็กทรอนิกส์ที่มีมูลค่า 100 บาท
- นาย ก สร้าง แฮชโคดจากใบเบิกดังกล่าว ด้วย แฮชฟังก์ชัน
- นาย ก ทำการเข้ารหัส แฮชโคด ด้วยกุญแจรหัส ใพรเวทีย์ ของนาย ก แล้วจึงส่งใบเบิกพร้อมกับ แฮชโคดที่ถูกเข้ารหัสซึ่งเรียกว่า ลายเซ็นดิจิทัลไปให้แก่ธนาคาร

กระบวนการเมื่อธนาคาร ได้รับใบเบิกและแฮชโคดที่ถูกเข้ารหัสมาจากนาย ก

- ธนาคารจะใช้กุญแจรหัส พิบบลิตีย์ของนาย ก ถอดรหัสแฮชโคดที่ได้ถูกเข้ารหัสไว้
- ธนาคารจะสร้างแฮชโคดใหม่จากใบเบิกที่ถูกส่งมาด้วยแฮชฟังก์ชันเดียวกันกับที่นาย ก ใช้
- ธนาคารจะทำการเปรียบเทียบแฮชโคดที่ทางธนาคารสร้างขึ้นใหม่กับแฮชโคดของนาย ก หากแฮชโคดที่สร้างขึ้นมีค่าตรงกันกับแฮชโคดที่นาย ก ส่งให้ก็แสดงว่าใบเบิกนั้นถูกส่งมาจากนาย ก จริง และไม่ได้ถูกเปลี่ยนแปลงแก้ไขเพราะหากข้อมูลในใบเบิกถูกเปลี่ยนแปลงแก้ไขระหว่างทาง เช่นถูกเปลี่ยนมูลค่าจาก 100 บาทเป็นล้านบาท แฮชโคดที่สร้างจากใบเบิกนั้นก็จะไม่ตรงกันกับแฮชโคดที่นาย ก สร้าง ทางธนาคารก็จะทราบว่าใบเบิกนั้นถูกเปลี่ยนแปลงแก้ไขและไม่อนุมัติตามใบเบิก ทั้งนี้เพื่อป้องกันความเสียหายที่อาจจะเกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้ลายเซ็นดิจิทัลในกระบวนการ Authenticate นั้น จะต้องใช้เทคนิคการเข้ารหัสแบบ พับบลิคคีย์ ซึ่งกระบวนการที่นาย ก จะทำการสร้างกุญแจรหัส ไพรวุทคีย์ให้กับตัวเองแล้วสร้างกุญแจ รหัสพับบลิคคีย์ให้กับทางธนาคารนั้นยังมีช่องโหว่ กล่าวคือ หากมีคนแอบอ้างชื่อนาย ก แล้วทำการสร้าง กุญแจรหัสทั้งไพรวุทคีย์และพับบลิคคีย์ขึ้นมาลายเซ็นดิจิทัลนั้นก็ยิ่งระบุว่า ผู้ส่งข้อมูลนั้นคือนาย ก ซึ่ง จริง ๆ แล้วเป็นคนอื่น

จากจุดที่เป็นช่องโหว่นี้เอง จึงต้องมีเอกสารอิเล็กทรอนิกส์ ที่เรียกว่า ใบรับรองดิจิทัล (Digital Certificate) ซึ่งออกโดยสถาบันหรือบุคคลที่สามที่เรียกว่า CA หรือ Certification Authority เพื่อนำมาใช้ รับรองและยืนยันตัวผู้ส่งข้อแตกต่างระหว่างลายเซ็นดิจิทัล ที่ทำหน้าที่ระบุว่าใครคือผู้ส่ง กับใบรับรอง ดิจิทัลที่ทำหน้าที่ยืนยันว่าตัวผู้ส่งนั้นคือตัวจริง

6.2.3 Non-Repudiation

Non-Repudiation หรือการไม่สามารถที่จะปฏิเสธได้กล่าวคือ เมื่อมีการส่งข้อมูลออกจากผู้ส่ง กระบวนการ Non-Repudiation จะทำให้ผู้ส่งไม่สามารถปฏิเสธได้ว่า ตัวเองเป็นผู้ส่งข้อมูล และข้อมูลไม่ ได้ถูกแก้ไขระหว่างทาง เพราะในกระบวนการดังกล่าวจะสามารถใช้ลายเซ็นดิจิทัลเป็นเครื่องพิสูจน์

กระบวนการ Non-Repudiation นอกจากจะมีระบบที่ป้องกันไม่ให้ผู้ส่งเอกสารอิเล็กทรอนิกส์ ปฏิเสธการส่งเอกสารแล้ว ในฝั่งผู้รับเองก็จะต้องมีกระบวนการที่ทำให้ผู้รับไม่สามารถปฏิเสธได้ว่าไม่ได้อรับเอกสารทั้ง ๆ ที่ได้รับเอกสารนั้นไปแล้ว

6.2.4 Confidentiality

เป็นคุณสมบัติพื้นฐานของระบบรักษาความปลอดภัย คือ การรักษาความลับของข้อมูลโดยไม่ อนุญาตให้ผู้อื่นสามารถอ่านข้อมูลหรือนำข้อมูลไปใช้ได้ หรือสืบหาที่มาของข้อมูลได้ ซึ่งจะใช้หลักการ เข้ารหัสข้อมูลซึ่งเป็นไปในลักษณะที่ตรงกันข้ามกับการใช้ลายเซ็นดิจิทัล

ในการใช้ลายเซ็นดิจิทัลนั้นฝั่งผู้ส่งจะทำการสร้างลายเซ็นดิจิทัลจากกุญแจรหัสแบบไพรวุทคีย์ ส่วนฝั่งผู้รับจะใช้กุญแจรหัสแบบพับบลิคคีย์ในการยืนยันลายเซ็นดิจิทัล

ในการเข้ารหัสข้อมูลผู้ส่งจะทำการเข้ารหัสข้อมูลโดยใช้กุญแจรหัสแบบพับบลิคคีย์ของผู้รับ ฝั่งผู้ รับจะใช้กุญแจรหัสแบบไพรวุทคีย์ในการถอดรหัสข้อมูล

จากที่กล่าวมาเป็นคุณสมบัติที่สำคัญสำหรับระบบรักษาความปลอดภัยของข้อมูลในอินเทอร์เน็ต

6.3 RSA มาตรฐานในการเข้ารหัสข้อมูลในอินเทอร์เน็ต

มาตรฐาน RSA เป็นชื่อย่อที่ตั้งมาจากศาสตราจารย์ที่สถาบัน MIT 3 ท่านคือ Dr. Rivest, Dr. Shamir และ Dr. Adleman ซึ่งเป็นผู้คิดค้นการเข้ารหัสแบบพับบลิคคีย์

ในปัจจุบันมีโปรแกรมสำเร็จรูปที่ใช้ในการสร้างกุญแจรหัสแบบ RSA ที่สามารถนำไปใช้งานได้ iley เช่น PGP หรือ (Pretty Good Privacy)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่กล่าวไปก่อนหน้านี้แล้ว RSA มีเทคโนโลยีที่ใช้ในระบบรักษาความปลอดภัยข้อมูลที่สำคัญคือ ลายเซ็นดิจิทัลและใบรับรองดิจิทัล ซึ่งสามารถทำให้ระบบรักษาความปลอดภัยมีคุณสมบัติทั้ง 4 ข้อตามที่กำหนดไว้ได้

ตัวอย่างมาตรฐานรักษาความปลอดภัยคือ SSL หรือ Secure Socket Layer และอีกมาตรฐานหนึ่งคือ SHTTP หรือ Secure Hypertext Transfer Protocol ซึ่งทั้งสองโพรโตคอลเป็นการเข้ารหัสแบบ RSA จนในปัจจุบัน RSA ได้ยอมรับว่าเป็นมาตรฐานที่ใช้กันทั่วโลกจากสถาบันมาตรฐานต่าง ๆ เช่น ISO 9796 หรือ X.509 เป็นต้น

6.4 มาตรฐานความปลอดภัยในอินเทอร์เน็ต SSL

ในการใช้เว็บเบราว์เซอร์อย่าง Internet Explorer หรือ Netscape ในการค้นหาข้อมูลบนอินเทอร์เน็ตเมื่อมีการเข้าไปในเว็บที่มีการเข้ารหัสหรือใช้ระบบรักษาความปลอดภัยข้อมูลแบบ SSL จะมีรูปไอคอนที่เป็นกุญแจปรากฏขึ้นและโพรโตคอลก็จะเปลี่ยนจาก HTTP ไปเป็น HTTPS

6.4.1 SSL (Secure Socket Layer)

รูปแบบการรักษาความปลอดภัยข้อมูลแบบ SSL หรือ Secure Socket Layer เป็นมาตรฐานที่คิดค้นโดยบริษัท Netscape โดยมีจุดประสงค์ในการรักษาความลับของข้อมูลในระหว่างการรับส่ง และเพื่อให้แน่ใจว่าข้อมูลนั้นถูกรับส่งตามที่ได้ระบุไว้จริง โดยที่ข้อมูลจะต้องไม่มีการแก้ไข

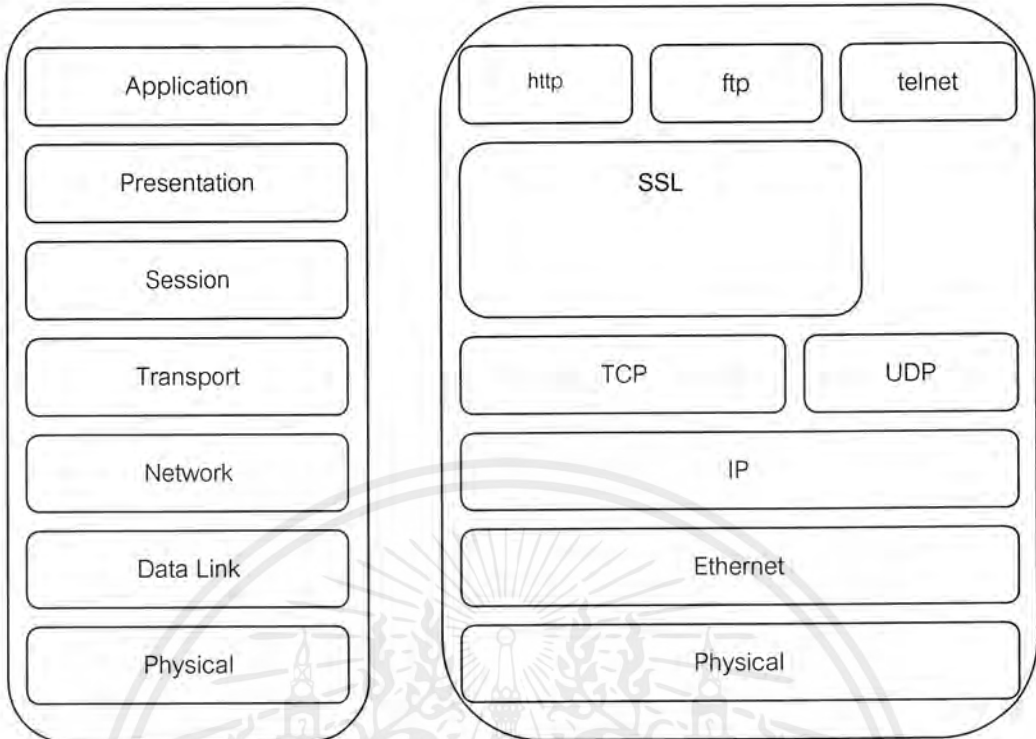
โพรโตคอล SSL ประกอบด้วยระดับโพรโตคอลย่อย 2 ระดับ คือ

- **SSL Handshake Protocol**

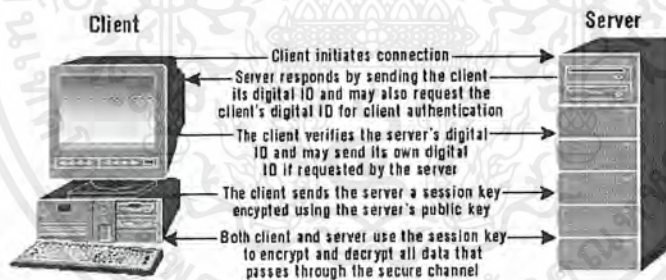
ทำหน้าที่สร้างกระบวนการรับส่งข้อมูลระหว่างผู้รับ และ ผู้ส่ง แลกเปลี่ยนรูปแบบ หรือ อัลกอริทึม ที่ใช้ในการเข้ารหัส รวมไปถึงกุญแจรหัสที่ใช้ในการรับ-ส่งข้อมูล

- **SSL Record Protocol**

จะอยู่เหนือ Transport Layer ในโมเดล OSI ทำหน้าที่จัดแบ่งบล็อกข้อมูลที่ได้จากโพรโตคอลที่อยู่ระดับเหนือขึ้นไปให้เป็นเรคคอร์ด ทำการบีบอัดข้อมูล กำหนดรหัสข้อมูล (Message Authentication Code หรือ MAC) และดำเนินการเข้ารหัสข้อมูลก่อนที่จะส่งข้อมูลไปยังโพรโตคอลในระดับถัดไป ดังแสดงดังรูป 6-1 และ 6-2



รูปที่ 6-1 เลเยอร์ของ SSL เปรียบเทียบกับเลเยอร์ของโมเดล OSI



รูปที่ 6-2 การทำงานในแต่ละเลเยอร์ของ SSL

คุณสมบัติที่สำคัญของ SSL คือ

- การรักษาความลับของข้อมูลขณะที่รับ-ส่ง จากการใช้การเข้ารหัสข้อมูล
- การระบุตัวตนฝั่งเซิร์ฟเวอร์ หรือ Server Authentication
- การระบุตัวตนฝั่งไคลเอนต์ (เป็นอ็อปชัน)
- ความถูกต้องของข้อมูลที่ตรวจสอบได้ว่าจะต้องไม่ถูกเปลี่ยนแปลงแก้ไขในระหว่างการรับส่ง

6.4.2 SSL กับการซื้อขายสินค้าผ่านระบบอินเทอร์เน็ต

เนื่องจากการซื้อขายสินค้าผ่านระบบอินเทอร์เน็ตนั้นเมื่อเปรียบเทียบความสำคัญระหว่างการระบุตัวตนของผู้ซื้อและผู้ขายว่าเป็นคน ๆ นั้นจริงตามที่กล่าวอ้างหรือไม่ เรามักให้ความสำคัญในการระบุตัวเอกสารนี้เป็นเอกสารที่ส่งมอบไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตนของผู้ขายหรือร้านค้ามากกว่า เพราะหากผู้ซื้อไม่มีความมั่นใจในเว็บไซค์ร้านค้าก็เท่ากับว่าโอกาสที่ผู้ขายหรือร้านค้าจะได้รับคำสั่งซื้อ ย่อมเป็นไปได้

ในทางกลับกัน หากผู้ขายเป็นบริษัทที่มีชื่อเสียงเป็นที่รู้จักกันดี ผู้ซื้อก็มีแนวโน้มที่จะยินดีชำระค่าสินค้าให้ ในขณะที่ความสำคัญของการระบุตัวตนในฝั่งผู้ซื้อนั้นจะไม่ค่อยมีความจำเป็นมากนัก เนื่องจากผู้ขายต้องการจะรับชำระเงินอยู่แล้วโดยไม่คำนึงถึงว่าผู้ซื้อจะเป็นใคร

SSL จึงดูเหมือนเป็นระบบรักษาความปลอดภัยที่ใช้ในการดำเนินธุรกรรมที่มีความนิยมมากที่สุด ในปัจจุบัน อีกทั้งรูปแบบการทำ Authentication ในระบบ SSL จะกระทำที่ฝั่งเซิร์ฟเวอร์ซึ่งเป็นร้านค้าหรือผู้ขาย ในขณะที่ทางฝั่งไคลเอนต์หรือผู้ซื้อจะเลือกกระทำหรือไม่ก็ได้ ซึ่งสอดคล้องกับพฤติกรรมการจับจ่ายซื้อสินค้าในอินเทอร์เน็ตตามที่กล่าวมาข้างต้นนั่นเอง

6.4.3 SSL กับการใช้ใบรับรองดิจิทัล

แม้ว่า SSL จะสามารถใช้เพื่อระบุตัวตนผู้ขายหรือร้านค้าได้จริงจากการ Authentication โดยใช้เทคโนโลยีพีบบลิคคีย์ ในลักษณะใบรับรองดิจิทัล แต่ SSL ก็ไม่ยืนยันว่าร้านค้านั้น ๆ คือร้านค้าที่กล่าวอ้างจริง ถ้าหากใบรับรองดิจิทัลนั้นไม่ได้ถูกรับรองจากหน่วยงาน CA (Certification Authority) ที่เชื่อถือได้

ในปัจจุบัน SSL ได้ถูกปรับปรุงและพัฒนาจนถึงเวอร์ชันที่ 3.0 และ IETF (Internet Engineering Task Force) ได้มีการเปลี่ยนชื่อใหม่ว่า Transport Layer Security (TLS)

6.5 มาตรฐานความปลอดภัยในอินเทอร์เน็ต SET

SET (Secure Electronic Transaction) เป็นมาตรฐานข้อตกลงในการทำธุรกรรมทางอินเทอร์เน็ต โดยใช้รูปแบบการรับส่งข้อมูลทรานแซกชันที่มีความปลอดภัย ระหว่างผู้ซื้อ ร้านค้าและธนาคารหรือสถาบันการเงิน ในลักษณะที่อิงอยู่กับบัตรเครดิต

SET ใช้เทคโนโลยีการเข้ารหัสข้อมูลเพื่อจุดประสงค์ในการ

- รักษาความลับของข้อมูล (Confidentiality) เช่น ข้อมูลหมายเลขบัตรเครดิต , เงินโอนการสั่งซื้อ หรือที่อยู่ลูกค้า เป็นต้น

- คงความถูกต้องของข้อมูล (Integrity) เช่น ข้อมูลการชำระเงิน เป็นต้น

- ตรวจสอบและยืนยันสถานะภาพของทั้งร้านค้าและลูกค้าผู้ถือบัตร (Authentication)

วิธีการที่ SET ใช้เพื่อบรรลุวัตถุประสงค์ข้างต้นได้แก่

- การใช้การเข้ารหัสข้อมูลสำหรับการรักษาความลับของข้อมูล

- การใช้ลายเซ็นดิจิทัลในการรับประกันความถูกต้องของข้อมูล

- การใช้ลายเซ็นดิจิทัลและใบรับรองดิจิทัลของผู้ถือบัตรในการตรวจสอบและยืนยันสถานะภาพของผู้ถือบัตร

- การใช้ลายเซ็นดิจิทัลและใบรับรองดิจิทัลของร้านค้าในการตรวจสอบและยืนยันสถานะภาพของร้านค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.5.1 องค์ประกอบของ SET

ในระบบ SET ประกอบด้วย 4 องค์ประกอบหลักคือ

- **ลูกค้าผู้ถือบัตรเครดิต (Cardholder)**

คือผู้ถือบัตรเครดิตที่ออกให้โดยธนาคารหรือบริษัทผู้ให้บริการบัตรเครดิต โดยมีซอฟต์แวร์ประเภทกระเป๋าเงินอิเล็กทรอนิกส์ (Electronic Wallet) ติดตั้งอยู่ในเครื่องคอมพิวเตอร์ เพื่อสร้างกระบวนการในการรับ-ส่งข้อมูล เช่น ข้อมูลการสั่งซื้อ ข้อมูลการชำระเงินที่อยู่ภายใต้โพรโตคอลของ SET ระหว่างร้านค้าและ CA เป็นต้น

- **ร้านค้า (Merchant)**

จะมีเซิร์ฟเวอร์คอยรับคำสั่งซื้อจากลูกค้า ตรวจสอบสถานภาพของผู้ซื้อจาก CA รวมทั้งทำหน้าที่ประมวลผลค่าของอนุมัติการชำระเงินไปยัง Payment gateway

- **Payment gateway**

ทำหน้าที่ตรวจสอบและอนุมัติการชำระเงิน รวมทั้งการเรียกเก็บเงินจากลูกค้าผ่านเครือข่ายที่เป็น Private Network ผู้ที่ทำหน้าที่ Payment gateway ส่วนใหญ่จะเป็นธนาคารหรือสถาบันการเงินที่ทางร้านค้าเปิดบัญชีไว้ เพื่อรับสลิปที่ใช้จ่ายจากลูกค้าบัตรเครดิต

- **Certification Authority**

จะทำหน้าที่ออกใบรับรองดิจิทัลรวมถึงรับรองสถานภาพของร้านค้า ลูกค้าผู้ถือบัตรและผู้ให้บริการ Payment gateway เมื่อมีการร้องขอ นอกจากนี้ CA ยังมีหน้าที่ในการดูแลรับผิดชอบฐานข้อมูลใบรับรองรวมทั้งเผยแพร่ข้อมูลที่สำคัญ ๆ ในกรณีที่สถานะของใบรับรองมีการเปลี่ยนแปลง

6.5.2 รูปแบบการซื้อขายด้วย SET

รูปแบบการซื้อขายสินค้าผ่านอินเทอร์เน็ตในระบบ SET เป็นดังนี้

1. ลูกค้าเลือกชมสินค้าต่าง ๆ ในเว็บไซต์ของร้านค้าและเมื่อตัดสินใจซื้อสินค้า ลูกค้าจะทำการส่งคำขอเบื้องต้น (Initial Request) ไปยังร้านค้าจากซอฟต์แวร์กระเป๋าเงินอิเล็กทรอนิกส์ (Wallet)
2. ร้านค้ารับคำขอเบื้องต้น และดำเนินการส่งคำตอบรับเบื้องต้น (Initial Response) ที่ลงลายเซ็นดิจิทัลด้วยไพรเวทคีย์ของร้านค้าไปพร้อมกับใบรับรองดิจิทัลของร้านค้าและ Payment Gateway ไปให้กับลูกค้า
3. ลูกค้าจะตรวจสอบและยืนยันสถานภาพของร้านค้าและ Payment Gateway จาก CA จากนั้นก็จะส่งคำสั่งซื้อที่ถูกลงลายเซ็นดิจิทัลด้วยไพรเวทคีย์ของลูกค้าและส่งคำสั่งการชำระเงิน (Payment Instruction) ที่ถูกเข้ารหัสด้วยกุญแจรหัสแบบ Symmetric Key เพื่อรักษาความลับของข้อมูลในคำสั่งการชำระเงินตลอดจนส่งข้อมูลบัตรเครดิต อาทิ หมายเลขบัตรเครดิต และวันหมดอายุพร้อมกับกุญแจรหัส Symmetric Key ที่ถูกเข้ารหัสด้วยฟังก์ชันของ Payment Gateway (ทั้งนี้เพื่อไม่ให้ร้านค้าทราบข้อมูลบัตรเครดิต) และส่งใบรับรองดิจิทัลของลูกค้าให้แก่ร้านค้า
4. ร้านค้าตรวจสอบและยืนยันสถานภาพของลูกค้าและตรวจสอบความถูกต้องของคำสั่งซื้อด้วยฟังก์ชันของร้านค้า จากนั้นจึงส่งคำสั่งการชำระเงินไปยัง Payment Gateway เพื่อขออนุมัติ พร้อมกันนั้นเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะส่งคำตอบรับคำสั่งซื้อที่ถูกลายเซ็นดิจิทัลไปให้แก่ลูกค้าพร้อมกับใบรับรองดิจิทัลของร้านค้า หากคำขออนุมัติเป็นผลร้านค้าก็จะดำเนินการส่งสินค้าไปให้ลูกค้า

5. ลูกค้าตรวจสอบและยืนยันสถานภาพของร้านค้า แล้วตรวจสอบความถูกต้องของคำตอบรับคำสั่งซื้อด้วยพบบลิตีย์ของร้านค้า จากนั้นจึงเก็บคำตอบรับคำสั่งซื้อไว้เป็นหลักฐาน

6.6 Certificate Server

เป็นหนึ่งในชุด Back Office ที่ทำหน้าที่ในการดูแลด้านความปลอดภัยให้กับระบบเช่น การเข้ารหัสข้อมูลที่ส่งระหว่างกัน เป็นต้น รายละเอียดความสามารถตลอดจนการติดตั้งให้ดูที่ภาคผนวก ก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

ระบบมัลติมีเดียแบบสตรีมและ Windows Media Server (Streaming Multimedia Systems and Windows Media Server)

7.1 คำอธิบายเบื้องต้นเกี่ยวกับ Advance Streaming Format (ASF)

Windows Media Services ใช้ ASF ซึ่งเป็นมาตรฐานเปิดที่สนับสนุนการเคลื่อนย้ายข้อมูลผ่านเครือข่ายรูปแบบต่าง ๆ และโพรโทคอล ASF ถูกใช้ในการจัดการและรวบรวมข้อมูลมัลติมีเดียสำหรับทำสตรีมมิ่ง (Streaming) ผ่านเครือข่าย ASF เป็นรูปแบบของไฟล์ซึ่งสามารถใช้ในการถ่ายทอดสดได้และยังเป็นวิธีการที่ดีที่สุดในการส่งสตรีมมัลติมีเดีย (Multimedia streams) ผ่านเครือข่าย นอกจากนี้ยังสามารถเปิดไฟล์เสียงขึ้นมาฟังซ้ำได้

อัลกอริทึมในการบีบอัดและขยาย (Compression-decompression: codec) สามารถใช้ในการเข้ารหัสของสตรีม ASF โดยข้อมูลที่เก็บอยู่ในสตรีม ASF จะบอกผู้ใช้ว่าอัลกอริทึมไหนจะใช้ในการนำมาขยายสตรีม

Windows Media Audio มีส่วนขยายที่ต่อท้ายชื่อไฟล์เป็น .wma เป็นไฟล์ ASF ที่เกี่ยวกับเสียงอย่างเดียวซึ่งบีบอัดอยู่กับ Windows Media Audio codec โดยข้อมูลชนิดนี้จะต่างจากไฟล์ .asf ตรงส่วนขยายที่ต่อท้ายชื่อไฟล์ โดย Windows Media Server สามารถทำการสตรีมไฟล์ .wma และกระจายเสียงไฟล์ .wma (ไฟล์ .wax) โดยใช้ Program Manager ทางบริษัทไมโครซอฟท์สร้างไฟล์ .wma สำหรับใช้ทำงานเกี่ยวกับเสียงเพียงอย่างเดียว

7.2 คำอธิบายเบื้องต้นเกี่ยวกับ Windows Media

ไฟล์ Windows Media Audio เป็นไฟล์ .asf ที่ทำงานกับเสียงเพียงอย่างเดียว โดยถูกบีบอัดไว้กับ Windows Media Audio codec แล้วถูกเปลี่ยนชื่อไปเป็นไฟล์ที่มีส่วนขยายเป็น .wma ในการเปลี่ยนไฟล์ ASF ไปเป็น Windows Media Audio นั้นทำโดยเปลี่ยนชื่อส่วนขยายจาก .asf ไปเป็น .wma ซึ่งไฟล์ .wma ทุกไฟล์จะมีไฟล์ Stream redirector ของตัวเอง เรียกว่า ไฟล์ .wax โดยไฟล์ .wax นั้นจะมีหน้าที่การทำงานคล้ายกับไฟล์ AFS Stream redirector (.asx) แตกต่างกันตรงไฟล์ .wax ถูกออกแบบเพื่อใช้กระจายเสียง Windows Media Audio เราสามารถสร้างไฟล์ .wax โดยทำการเปลี่ยนชื่อส่วนขยายจาก .asx ไปเป็น .wax

7.3 การส่งข้อมูลสตรีมแบบ Unicast และ Multicast

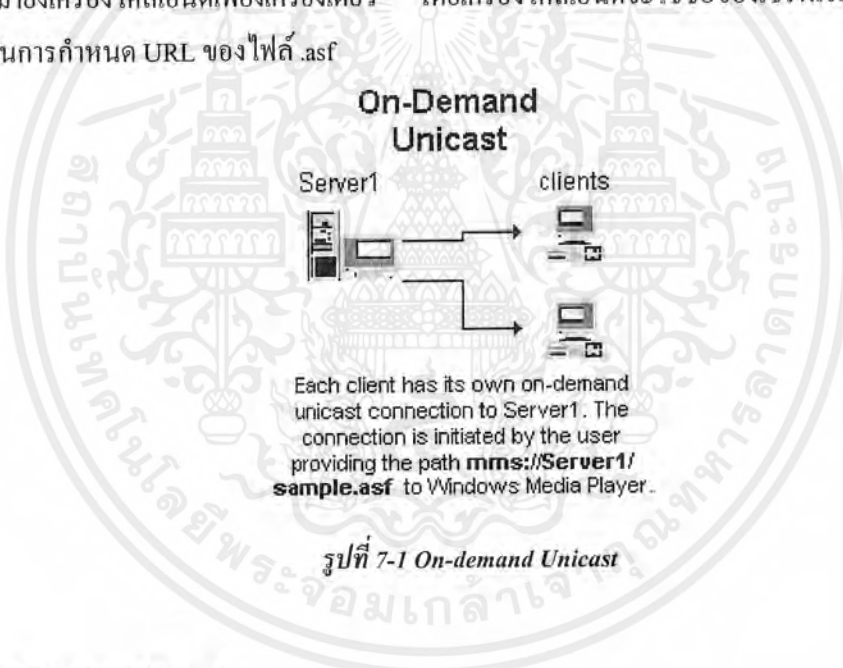
วิธีการที่ผู้ใช้รับข้อมูลจาก Windows Media Server นั้นสามารถแบ่งออกได้เป็น 2 วิธี คือ

7.3.1 Unicast

Unicast เป็นการติดต่อสื่อสารแบบ point-to-point ระหว่างผู้ใช้กับเซิร์ฟเวอร์ คำว่า point-to-point หมายความว่า ผู้ใช้แต่ละคนจะรับสตรีมคนละอันกันจากเซิร์ฟเวอร์ Unicast stream จะถูกส่งก็ต่อเมื่อผู้ใช้ทำการร้องขอมายังเซิร์ฟเวอร์ โดยสามารถเผยแพร่ Unicast stream ไปยังผู้ใช้ได้ใน 2 ลักษณะ คือ

- **On – demand Unicast**

On-demand เป็นวิธีการหนึ่งที่ได้รับข้อมูลที่ป็นสตรีมจาก Windows Media Server การติดต่อแบบ On-demand ผู้ใช้สร้างการเชื่อมต่อจากเครื่องไคลเอนต์ไปยังเซิร์ฟเวอร์ โดยทำการเลือกข้อมูลที่ต้องการ ข้อมูลจะถูกส่งจากเซิร์ฟเวอร์ไปยังไคลเอนต์ในรูปแบบสตรีม ASF ถ้าข้อมูลถูกจัดเรียงเรียบร้อยแล้วผู้ใช้สามารถเล่น หยุด ถอยหลัง เดินหน้าหรือหยุดชั่วขณะกับสตรีมนั้นได้ การเชื่อมต่อแบบ On-demand ช่วยให้สามารถควบคุมการทำงานกับสตรีมได้ แต่จะใช้แบนด์วิดท์ของระบบเครือข่ายจนหมด เพราะแต่ละไคลเอนต์จะมีการเชื่อมต่อของตนเองไปยังเซิร์ฟเวอร์ ตัวอย่างของ On-demand Unicast คือ เมื่อผู้ใช้ร้องขอไฟล์ .asf ไคลเอนต์จะทำการเชื่อมต่อไปยังเซิร์ฟเวอร์เพื่อรับข้อมูล จากนั้นข้อมูลจะถูกส่งเป็นสตรีมมายังเครื่องไคลเอนต์เพียงเครื่องเดียว โดยเครื่องไคลเอนต์จะใช้ชื่อของเซิร์ฟเวอร์และชื่อของไฟล์ .asf ในการกำหนด URL ของไฟล์ .asf



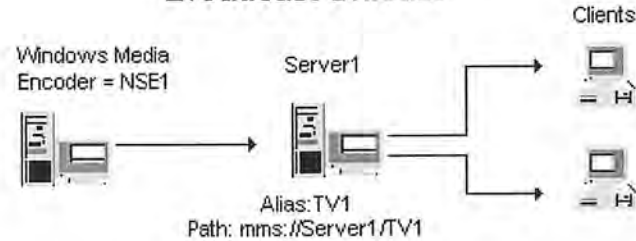
รูปที่ 7-1 On-demand Unicast

- **Broadcast Unicast**

สำหรับ Broadcast Unicast นั้น สตรีมจะถูกเข้าใช้จาก alias ที่จุดกระจายสัญญาณ (Publishing point) ผู้ใช้สามารถคลิกที่ลิงค์ในเว็บเพจหรือกำหนด URL ที่อ้างอิงไปถึง alias แล้วทำการเชื่อมต่อไปยังสตรีม ผู้ใช้แต่ละคนที่เชื่อมต่อไปยังสตรีมจะมีการเชื่อมต่อและสตรีมของข้อมูลจากเซิร์ฟเวอร์เป็นของตนเอง ยกตัวอย่างเช่น Windows Media Encoder ส่งข้อมูลสตรีมไปยังจุดกระจายสัญญาณบน Windows Media Server ทุกจุดกระจายสัญญาณมี alias โดย alias ในตัวอย่างนี้ของจุดกระจายสัญญาณ คือ TV1 ไคลเอนต์จะใช้ TV 1 ในการระบุ path ที่ชี้ไปยังที่เก็บสตรีม สำหรับ URL ที่ชี้ไปยังสตรีม ASF จะคล้ายกับของไฟล์ .asf อย่างไรก็ตามแทนที่จะระบุชื่อของไฟล์ .asf ก็ระบุ alias ของจุดกระจายสัญญาณแทน TV1 ดังนั้นในตัวอย่างนี้ จะใช้ path `mms://Servername/TV1` ในการเข้าใช้สตรีม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Broadcast Unicast



รูปที่ 7-2 Broadcast Unicast

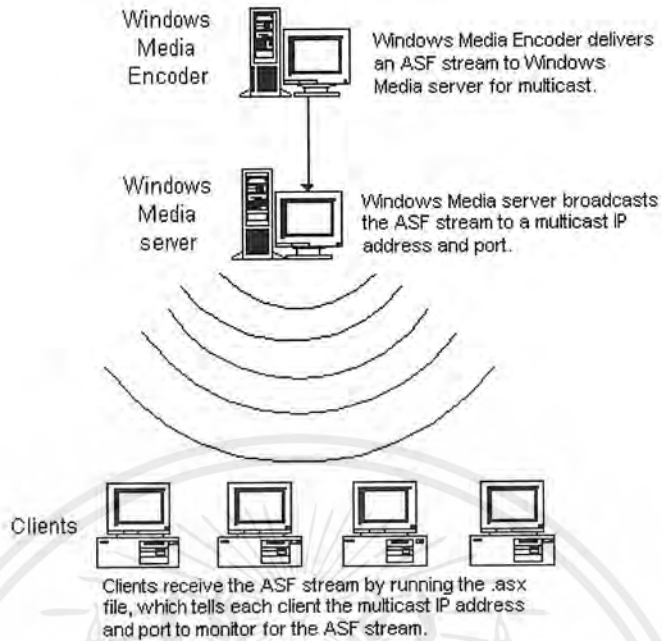
7.3.2 Multicast

Multicast เป็นการเคลื่อนย้ายข้อมูลแบบสตรีมผ่านเครือข่ายที่สนับสนุนการทำ Multicast โดยไคลเอนต์ทุกคนบนเครือข่ายจะใช้สตรีมอันเดียวกันร่วมกัน ข้อดีของการทำสตรีมข้อมูล ASF คือ จะช่วยลดการใช้งานแบนด์วิดท์ของเครือข่าย เราสามารถขยายการใช้งาน Multicast ไปยังเครือข่ายที่สนับสนุนการทำงาน Multicast โดยทำการติดตั้ง Windows Media Server ในแต่ละ Segment ของเครือข่าย เรียกว่า Server Distribution เนื่องจากการทำ Multicast จากเซิร์ฟเวอร์เราสามารถส่งสตรีมไปยัง Windows Media Servers ที่ Segment อื่น ๆ บนเครือข่ายได้โดยอาจจะใช้วิธี Unicast หรือ Multicast ซึ่งเรียกวธีการนี้ว่า re-distribution โดยการผูกเซิร์ฟเวอร์หนึ่งกับเซิร์ฟเวอร์เครื่องอื่น นอกจากนั้นคุณยังแก้ปัญหาที่ Router ไม่สนับสนุนการทำ Multicast ได้ โมเดลนี้สามารถทำงานผ่านไฟลว์อัลลีได้โดยผู้ดูแล Windows Media Services ต้องสร้าง 3 ไซต์เพื่อสนับสนุนการทำ Multicast ซึ่งได้แก่ Station, Program และ Stream โดย Station จะทำหน้าที่เป็นจุดอ้างอิงสำหรับไคลเอนต์ที่ต้องการเชื่อมต่อไปยังสตรีม โปรแกรมทำหน้าที่จัดโครงสร้างของข้อมูลที่จะทำการกระจายสัญญาณผ่านสถานี เมื่อทุกไซต์ถูกสร้างขึ้น ผู้ดูแล Windows Media Services จะสร้างไฟล์ .asx เพื่อทำการลิงก์ไคลเอนต์ไปยังไอพีแอดเดรสของสถานี

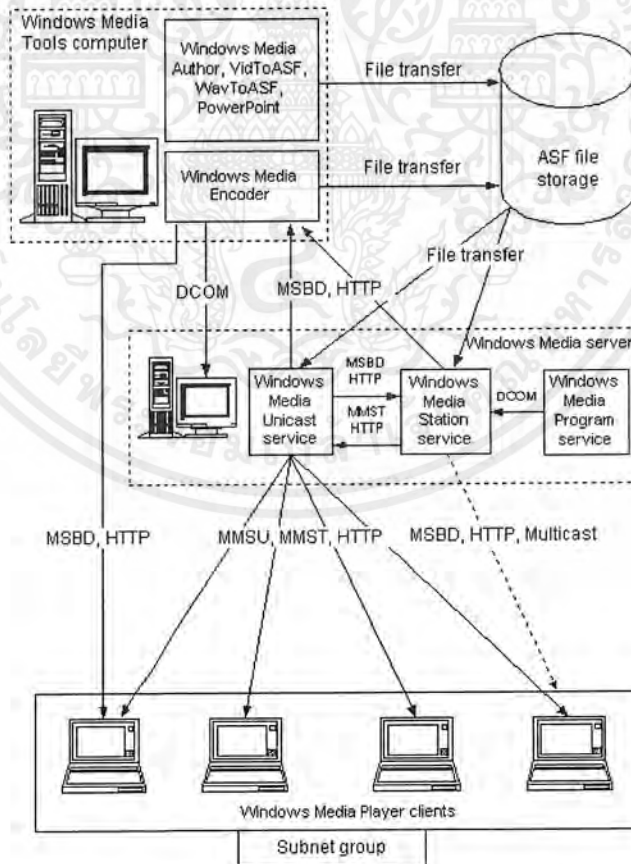
● Broadcast Multicast

Broadcast Multicast ไคลเอนต์จะรับสตรีม asf โดยการเชื่อมต่อไปยังไอพีแอดเดรสที่กำหนดไว้ (คล้ายกับการรับสัญญาณจากสถานีวิทยุที่ความถี่ต่าง ๆ กัน) ประโยชน์ของการทำ Multicast คือ สตรีมเพียงสตรีมเดียวสามารถส่งข้อมูลไปยังไคลเอนต์หลาย ๆ เครื่องผ่านเครือข่ายและยังช่วยลดการใช้แบนด์วิดท์ของเครือข่ายซึ่งเป็นประโยชน์ต่อเล่นที่ความเร็วต่ำ

Broadcast Multicast



รูปที่ 7-3 Broadcast Multicast



รูปที่ 7-4 การใช้โปรโตคอลต่างๆ ในการสื่อสารระหว่างคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.4 Windows Media Services Protocols

ไดอะแกรมดังรูป 7-4 แสดงให้เห็นว่าโพรโทคอลต่าง ๆ ถูกใช้ในการสื่อสารระหว่างคอมพิวเตอร์ของระบบ Windows Media Services ได้อย่างไร

คุณสามารถเชื่อมต่อคอมพิวเตอร์ทุกตัวผ่าน HTTP และสามารถทำให้สื่อสารกันได้แม้ว่าจะถูกขวางกั้นโดย firewall โคลเอนต์ที่ทำการเชื่อมต่อไปยัง multicast จะไม่ต้องใช้โพรโทคอลเพียงแต่รับข้อมูลเป็นแบบสตรีมผ่าน multicast IP address และไม่จำเป็นต้องสร้างการเชื่อมต่อ

โพรโทคอล MMS

โพรโทคอล MMS ถูกใช้ในการเข้าใช้ข้อมูล Unicast จากจุดกระจายสัญญาณ MMS เป็นวิธีการพื้นฐานในการเชื่อมต่อเข้าไปยัง Windows Media Unicast services ถ้าผู้ใช้ต้องการพิมพ์ URL ใน Windows Media Player เพื่อเชื่อมต่อโดยไม่ผ่านไฮเปอร์ลิงก์ในเว็บเพจ ผู้ใช้จะต้องใช้โพรโทคอล MMS และอ้างอิงถึงชื่อของสตรีมนั้นด้วย

เมื่อทำการเชื่อมต่อไปยังจุดกระจายสัญญาณโดยใช้โพรโทคอล MMS โพรโทคอล Rollover จะถูกใช้ในการหาทางเชื่อมต่อที่ดีที่สุด โดยเริ่มจากการเชื่อมต่อโคลเอนต์ไปยังเซิร์ฟเวอร์ผ่าน MMSU ซึ่ง MMSU คือ โพรโทคอล MMS ที่ใช้กับ UDP Data Transport ถ้าการเชื่อมต่อ MMSU ไม่สำเร็จเซิร์ฟเวอร์จะพยายามใช้ MMST ซึ่ง MMST คือ โพรโทคอล MMS ที่ใช้กับ TCP Data Transport

เมื่อทำการเชื่อมต่อไปยังจุดกระจายสัญญาณจาก Stand-alone Windows Media Player คุณจะต้องการระบุ URL ไปยังข้อมูล Unicast ถ้าข้อมูลถูกเผยแพร่แบบ On-demand ผ่านจุดกระจายสัญญาณ URL จะประกอบไปด้วยชื่อของเซิร์ฟเวอร์และชื่อไฟล์ .asf เช่น `mms://Windows_media_server/sample.asf` ถ้าคุณมีข้อมูลที่ต้องการถ่ายทอดสดผ่าน Broadcast Unicast URL จะประกอบไปด้วยชื่อของเซิร์ฟเวอร์และ alias ของจุดกระจายสัญญาณ เช่น `mms://Windows_media_server/LiveEvents`

โพรโทคอล MSBD

โพรโทคอล MSBD ถูกใช้ในการเผยแพร่สตรีมระหว่าง Windows Media Encoder และคอมพิวเตอร์ของ Windows Media Server และใช้แลกเปลี่ยนสตรีมระหว่างเซิร์ฟเวอร์ MSBD เป็นโพรโทคอลแบบ Connection-oriented ซึ่งเป็นวิธีการที่ดีที่สุดสำหรับการทำสตรีมของข้อมูล MSBD ถูกใช้ในการทดสอบความสามารถในการเชื่อมต่อระหว่างโคลเอนต์กับเซิร์ฟเวอร์และคุณภาพของข้อมูล ASF แต่ไม่ควรใช้เป็นวิธีการหลักในการรับข้อมูล ASF โดย Windows Media Encoder สามารถรองรับโคลเอนต์ MSBD ได้มากที่สุด 15 โคลเอนต์ส่วน Windows Media Server สามารถรองรับโคลเอนต์ MSBD ได้มากที่สุด 5 โคลเอนต์

บทที่ 8

การวิเคราะห์และการออกแบบระบบ

8.1 การวิเคราะห์และออกแบบระบบในส่วนคอมพิวเตอร์การทำงานส่วนหน้า

8.1.1 คำอธิบายปัญหา (Problem Statement)

เมื่อเราพิจารณาระบบพาณิชย์อิเล็กทรอนิกส์ (Electronic Commerce) เราจะได้คำอธิบายของระบบออกมาดังนี้

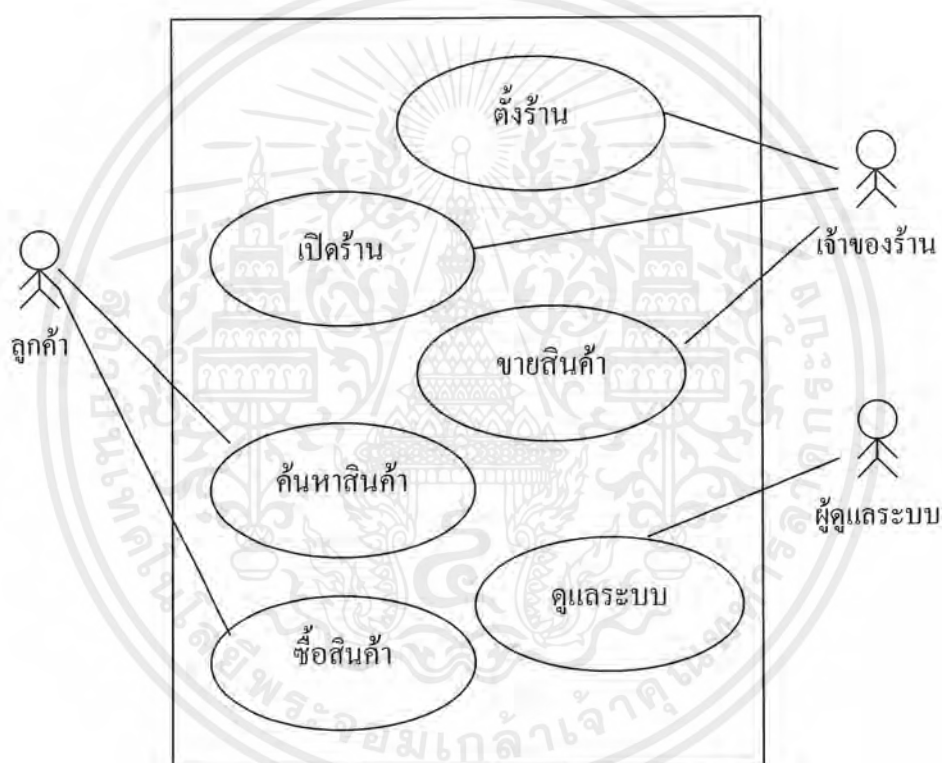
- ลูกค้าสามารถมาเปิดร้าน (เจ้าของร้าน) โดยมีรายละเอียดตามที่ลูกค้าต้องการได้ เช่น รูปแบบเว็บเพจของร้านที่เขาต้องการขายของ
- เจ้าของร้านสามารถจัดการรายละเอียดเกี่ยวกับสินค้าในร้านได้ เช่น การเพิ่มสินค้าเข้าไปในร้าน การเปลี่ยนแปลงข้อมูลของสินค้า หรือข้อมูลต่างๆในร้านของเขา และการลบข้อมูล สินค้าในร้านของเขาได้
- ข้อมูลของสินค้าจะอยู่ในรูปมัลติมีเดียสตรีม (Multimedia Stream) คือ ภาพ เสียง และวิดีโอ
- ร้านค้าที่เปิดขึ้นสามารถขายสินค้าใดๆ ก็ได้โดยมีการเก็บข้อมูลทุกอย่างที่จำเป็นตามแต่ชนิดของสินค้า
- ลูกค้าที่เข้ามาซื้อสินค้าสามารถเรียกดูข้อมูลต่างๆ เกี่ยวกับสินค้าที่ต้องการได้ และถ้ามีความสนใจในสินค้าสามารถทำการเลือกสินค้านั้นเก็บลงในกระเป๋าสินค้าได้ (Shopping Bag) ซึ่งข้อมูลนี้สามารถทำการเก็บไว้เพื่อใช้ในการซื้อครั้งต่อไปได้
- การเข้าใช้ระบบจะมีการควบคุมในระดับระบบสารสนเทศ (Information System Control) โดยมีการกรอกชื่อผู้ใช้และรหัสผ่าน ทั้งในส่วนของเจ้าของร้านและลูกค้าเพื่อเป็นการ ปกป้องทรัพยากรระบบ
 - ในการสื่อสารข้อมูลมีการเข้ารหัสในรูปแบบของ SSL (Secure Socket Layer) 40 บิต
 - ระบบการเก็บข้อมูลเป็นระบบจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์
 - มีระบบจัดการทรานแซกชันทั้งในระดับโลคอล (Local) และ โกลบอล (Global) หรือทูเฟสคอมมิต (Two-Phase Commit)
- ลูกค้าสามารถหาสินค้าตามคุณสมบัติของสินค้าที่ต้องการได้
- ระบบสามารถยืนยันการสั่งซื้อผ่านระบบเพจเจอร์และระบบส่งข้อความสั้น (Short Message) ของผู้ให้บริการต่างๆ ได้ เช่น โฟนลิงค์, GSM เป็นต้น และรวมถึงระบบอีเมลล์และระบบ ไอซีคิวได้ด้วย
- ผู้ซื้อสินค้าสามารถติดตามผลการส่งสินค้าได้ว่าขณะนี้อยู่ในขั้นตอนใดแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.1.2 สร้าง Use Case

เรานำคำอธิบายปัญหาที่แยกออกเป็นฟังก์ชันการทำงานโดยรวมของระบบ ในรูปแบบของ Use Case ซึ่งจะมี Use Case ดังนี้

- ตั้งร้าน
- เปิดร้าน
- ขายสินค้า
- ค้นหาสินค้า
- ซื้อสินค้า
- การดูแลระบบ

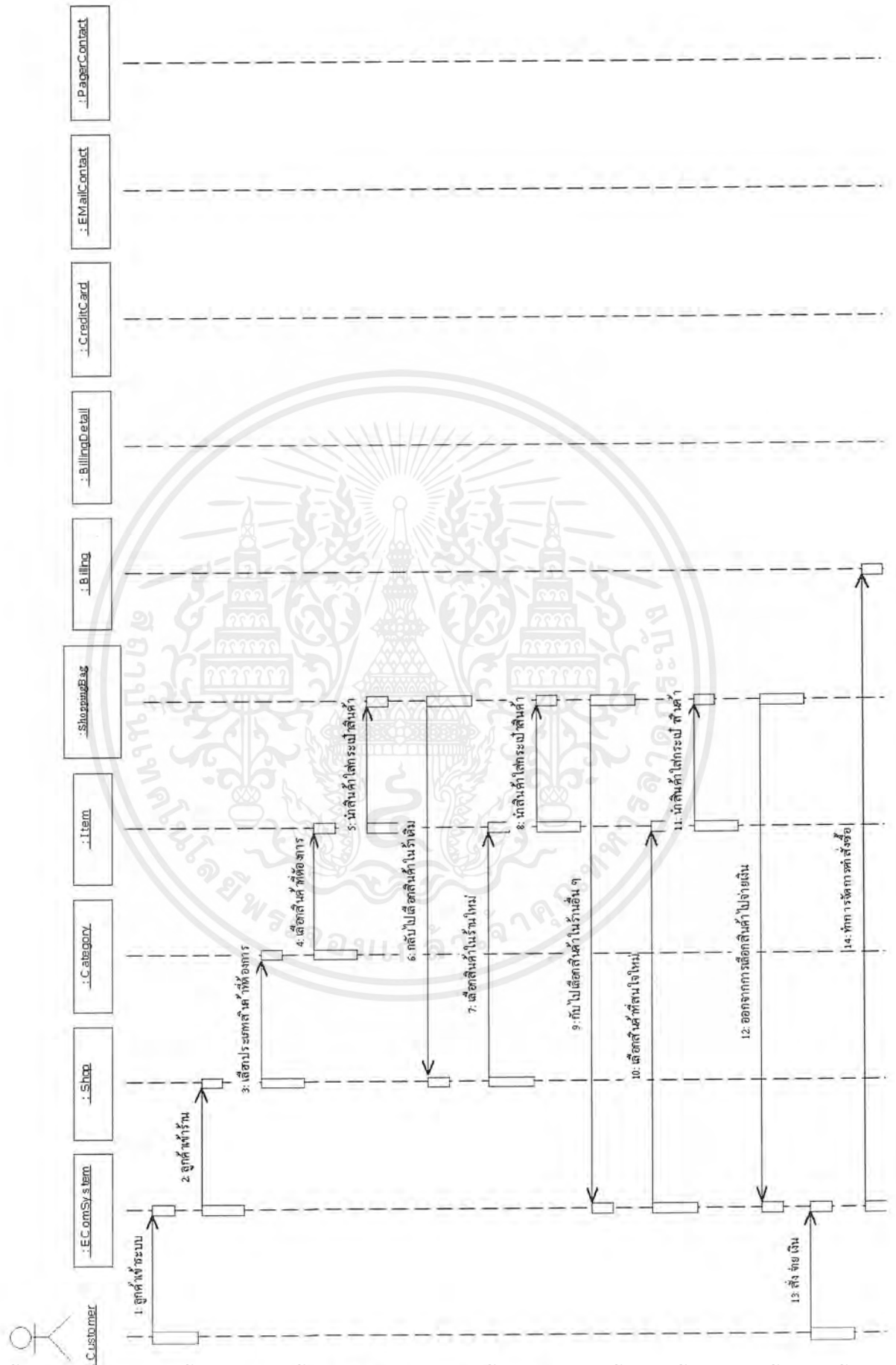


รูป 8-1 Use Case ของระบบ

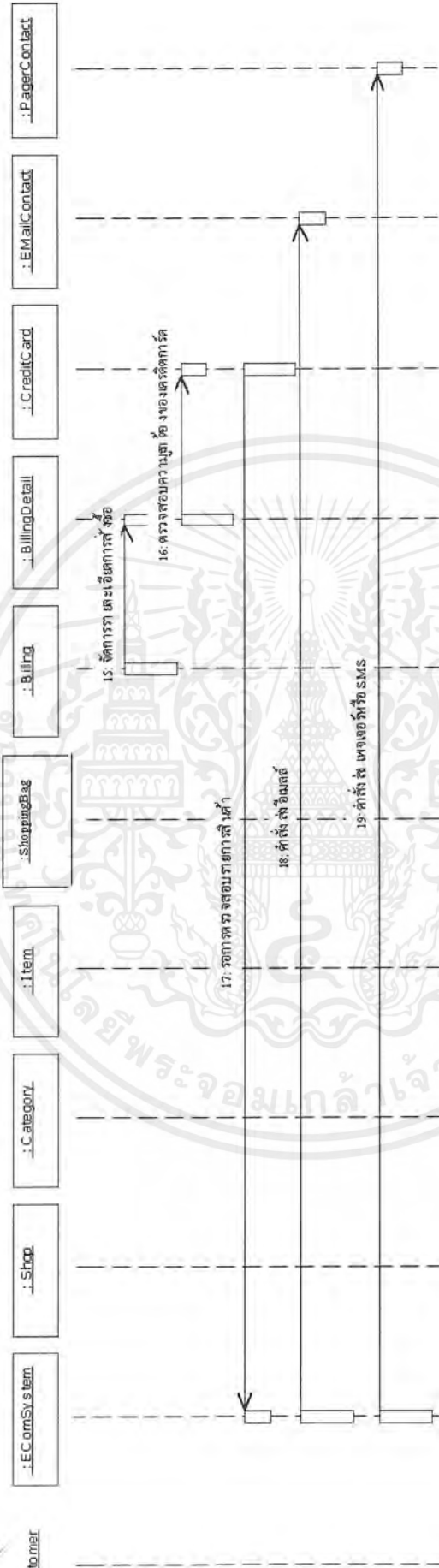
8.1.3 สร้างชินริโอ

ในที่นี้จะนำเพียง 2 Use Case คือ การซื้อสินค้า และการตั้งร้าน มาสร้างชินริโอเพื่อใช้ในการตรวจสอบความถูกต้องของ Use Case และช่วยในการกำหนดขอบเขตเบื้องต้น ซึ่งแสดงได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

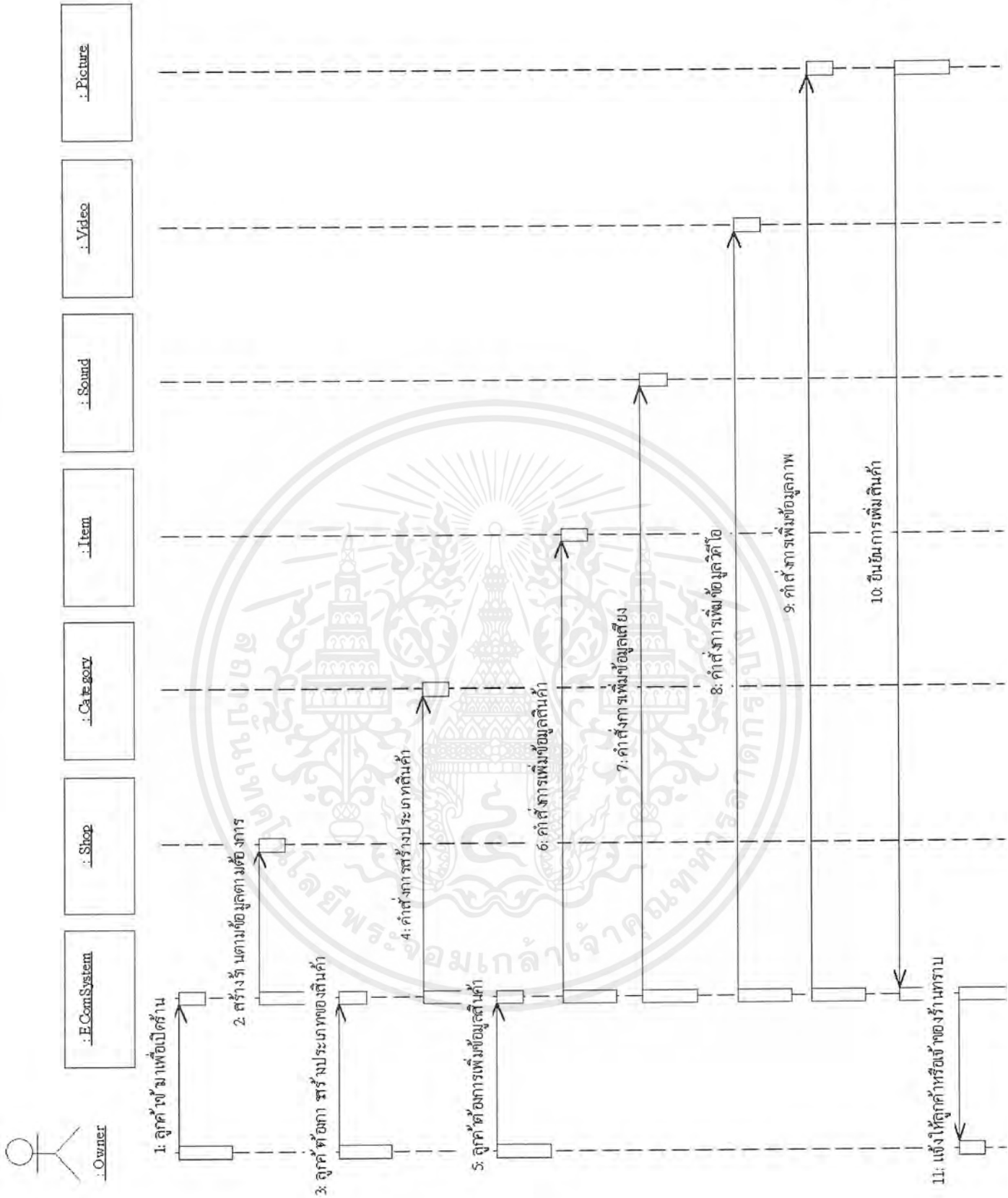


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเฉพาะเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 8-2 ขบวนการซื้อของ (1)
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต่ออ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 8-3 ขันาริโการซื้อของ (2)

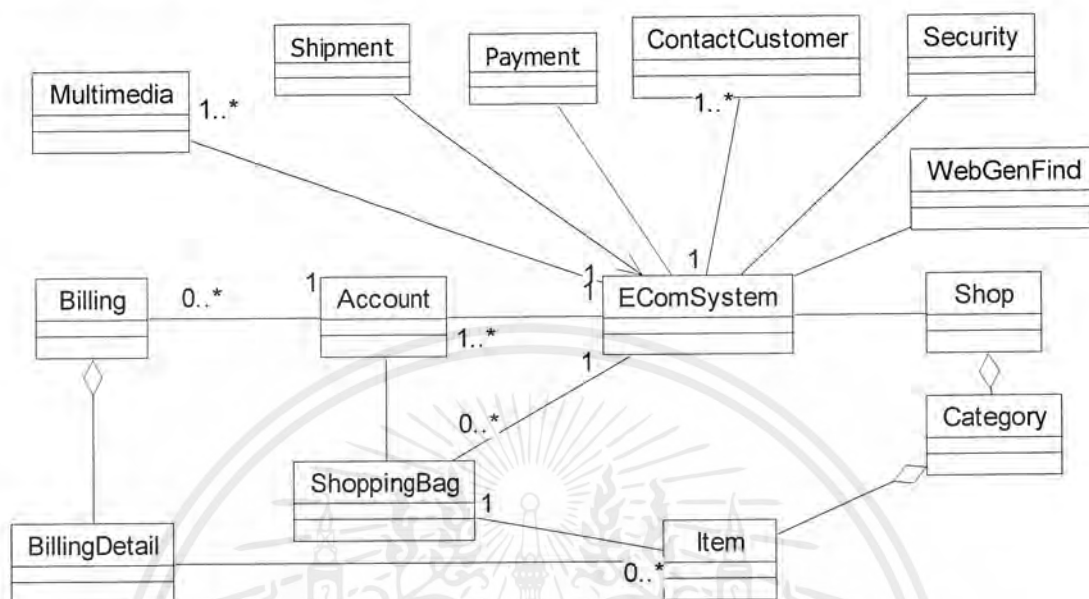


รูปที่ 8-4 ซินาวิโอการตั้งร้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.1.4 กำหนดโครงสร้างออบเจกต์

จากการวิเคราะห์ระบบทำให้ได้ออบเจกต์ไดอะแกรมในระดับการวิเคราะห์เบื้องต้น (First Cut Object Diagram) ออกมาดังนี้

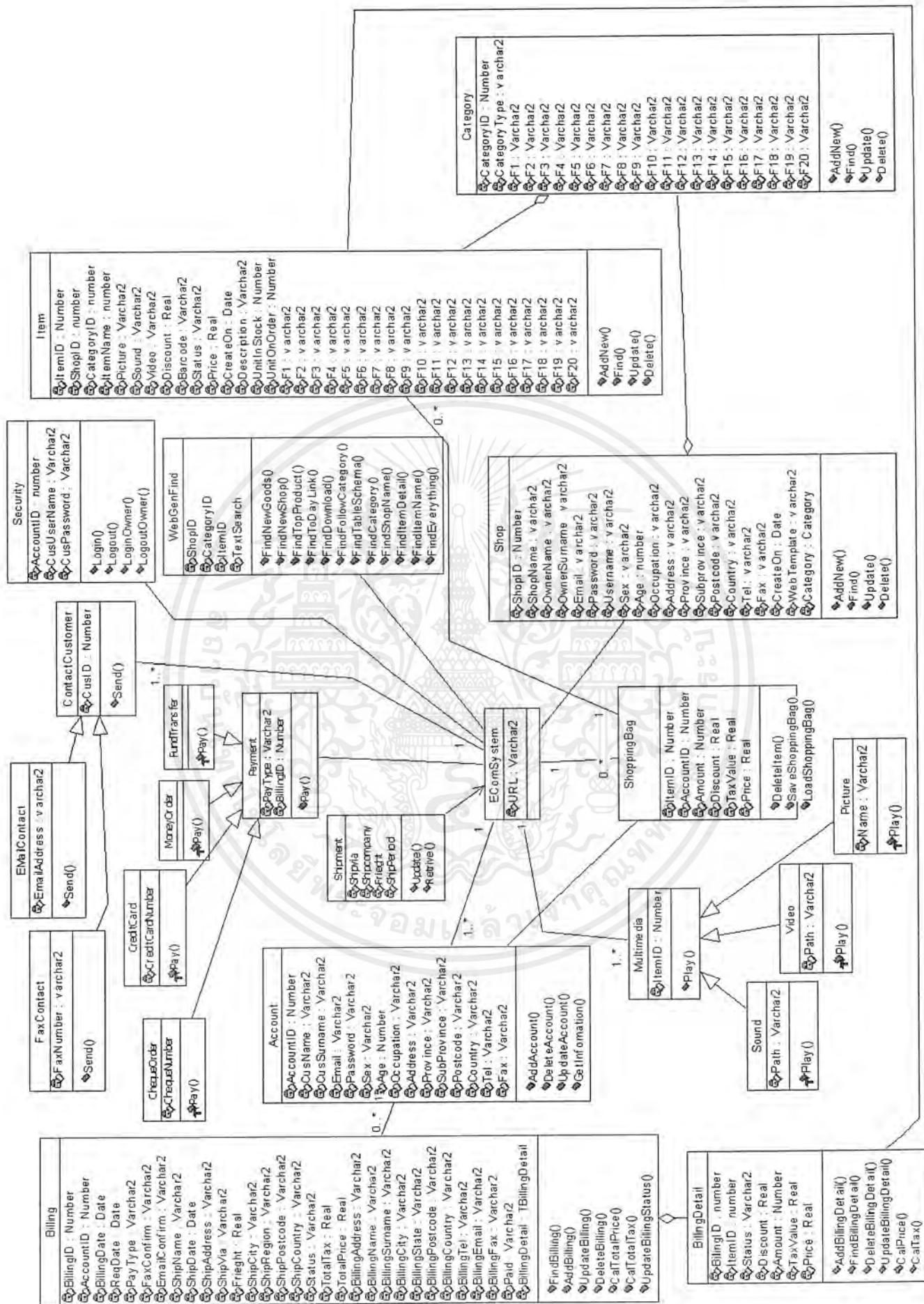


รูปที่ 8-5 ออบเจกต์ไดอะแกรมในระดับการวิเคราะห์เบื้องต้น

โดยรายละเอียดของออบเจกต์แต่ละตัวจะนำมาอธิบายอีกครั้งในส่วนของการสร้างโครงสร้างคลาสซึ่งอยู่ในส่วนต่อไป

8.1.5 กำหนดโครงสร้างคลาส

จากออบเจกต์ไดอะแกรมเราสามารถสร้างเอกลักษณ์ของระบบเพื่อแสดงโครงสร้างและความสัมพันธ์ต่างๆ ของออบเจกต์ในระบบ รวมถึงการสร้างสเตทไดอะแกรมเพื่อดูพฤติกรรมต่างๆ ของคลาสในระบบเพื่อใช้ระบุโอเปอเรชันและเม็ทรูดของคลาสจากที่กล่าวมาเราสามารถสร้างคลาสไดอะแกรมได้ดังรูปที่ 8-6



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบุคลากรใช้งานเพื่อการศึกษารวบรวม ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 8-6 กลาสโคเคแกรมของระบบ
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.1.6 รายละเอียดของคลาสในระบบ

ในส่วนของโครงการนี้จะเป็นการพัฒนาในส่วนของคอมโพเนนต์ส่วนหน้า(Front-end Component) ซึ่งเป็นคอมโพเนนต์ที่ทำหน้าที่ให้บริการต่าง ๆ แก่ ผู้ใช้ปลายทางโดยมีการเรียกใช้บริการพื้นฐานจากคอมโพเนนต์ส่วนหลัง(Back-end Component) อีกระดับหนึ่ง คอมโพเนนต์ส่วนหน้าประกอบไปด้วย

- PagerContact
- EmailContact
- ContactCustomer
- ChequeOrder
- CreditCard
- MoneyOrder
- FundTransfer
- Payment
- Multimedia
- Sound
- Video
- Picture
- ShoppingBag
- Shipment
- Security
- WebGenFind

ซึ่งรายละเอียดของแต่ละคลาสคอมโพเนนต์มีดังต่อไปนี้

ชื่อคลาส: PagerContact

คุณสมบัติ: PagerNumber ระบุเบอร์อุปกรณ์สื่อสาร

PagerType ระบุชนิดหรือผู้บริการ

PagerUsername ระบุชื่อผู้ใช้งาน

PagerPassword ระบุรหัสผ่านผู้ใช้งาน

เม็ทธอด: Send() ใช้ในการสั่งให้ส่งข้อความเมื่อพร้อม

หน้าที่รับผิดชอบ: จัดการการส่งข้อความการยืนยันการส่งชื่อผ่านระบบเพจเจอร์หรือบริการข้อความสั้นของระบบโทรศัพท์เคลื่อนที่

ชื่อคลาส: EmailContact

คุณสมบัติ: EmailAddress เป็นส่วนที่ระบุถึง Email Address ของผู้รับหรือลูกค้า

เม็ทธอด: Send() ใช้ในการสั่งให้ส่งเมลล์เมื่อพร้อม

หน้าที่รับผิดชอบ: จัดการการส่งข้อความการยืนยันการสั่งซื้อผ่านระบบเมลล์อิเล็กทรอนิกส์

ชื่อคลาส: ContactCustomer

คุณสมบัติ: CustID เป็นส่วนที่ระบุถึง หมายเลขลูกค้า

เม็ทธอด: Send() ใช้ในการสั่งให้ส่งข่าวสารการยืนยันการสั่งซื้อเมื่อพร้อม

หน้าที่รับผิดชอบ: จัดการการส่งข้อความการยืนยันการสั่งซื้อผ่านระบบการสื่อสารในแบบ

ต่าง ๆ

ชื่อคลาส: ChequeOrder

คุณสมบัติ: ChequeNumber เป็นหมายเลขของเช็ค

เม็ทธอด: Pay() เป็นเม็ทธอดในการจัดการคำสั่งการจ่ายเงินด้วยเช็ค เช่น การอัปเดตสถานะการจ่ายเงินของลูกค้าเป็นต้น

หน้าที่รับผิดชอบ: จัดการการจ่ายเงินด้วยเช็ค

ชื่อคลาส: CreditCard

คุณสมบัติ: CreditCardNumber เป็นหมายเลขที่ใช้ในการตรวจสอบความถูกต้องของเครดิตการ์ด

CreditCardExpire เป็นในหมดอายุของเครดิตการ์ด

เม็ทธอด: Pay() จัดการการจ่ายเงินด้วยเครดิตการ์ด เช่น การตรวจสอบความถูกต้องของเครดิตการ์ด

หน้าที่รับผิดชอบ: จัดการการจ่ายเงินด้วยเครดิตการ์ด เช่น การตรวจสอบความถูกต้องของเครดิตการ์ด

ชื่อคลาส: MoneyOrder

คุณสมบัติ:

เม็ทธอด: Pay() จัดการการจ่ายเงินด้วยธนาคัต

หน้าที่รับผิดชอบ: จัดการการจ่ายเงินด้วยธนาคัต

ชื่อคลาส: FundTransfer

คุณสมบัติ:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เม็ทธอด: Pay() จัดการการจ่ายเงินด้วยการ โอนเงินผ่านธนาคาร

หน้าที่รับผิดชอบ: จัดการการจ่ายเงินด้วยการ โอนเงินผ่านธนาคาร

ชื่อคลาส: Payment

คุณสมบัติ: Paytype เป็นตัวระบุประเภทของการจ่ายเงิน

BillingID เป็นตัวระบุหมายเลขของใบสั่งซื้อที่ต้องการจ่ายเงินด้วยวิธีที่กำหนด

เม็ทธอด: Pay() จัดการการจ่ายเงินด้วยวิธีต่าง ๆ

หน้าที่รับผิดชอบ: จัดการการจ่ายเงินด้วยวิธีต่าง ๆ พร้อมกับอัปเดตสถานะที่จำเป็นอื่น ๆ

ชื่อคลาส: Shipment

คุณสมบัติ: ShipVia เป็นการระบุวิธีการการส่งของ

ShipCompany เป็นสิ่งที่ระบุถึงชื่อบริษัทที่ทำการส่งสินค้า

Freight จะระบุถึงค่าขนส่งสินค้า

ShipPeriod จะระบุถึงช่วงเวลาโดยประมาณในการส่งสินค้าไปถึงปลายทาง

เม็ทธอด: Update() เป็นเม็ทธอดที่ทำหน้าที่อัปเดตรายละเอียดในการจัดส่งสินค้าแบบต่าง ๆ

Retrieve() เป็นเม็ทธอดที่ทำหน้าที่ดึงข้อมูลที่จำเป็นเกี่ยวกับการจัดส่งสินค้า

หน้าที่รับผิดชอบ: จัดการเก็บข้อมูลการจัดส่งสินค้า

ชื่อคลาส: Multimedia

คุณสมบัติ: ItemID เป็นตัวระบุหมายเลขของสินค้า

เม็ทธอด: Play() จัดการหน้าที่ทางด้านมัลติมีเดียในแบบต่าง ๆ

หน้าที่รับผิดชอบ: จัดการหน้าที่ทางด้านมัลติมีเดียในแบบต่าง ๆ

ชื่อคลาส: Sound

คุณสมบัติ: Path เป็นตัวระบุ Path ที่เก็บข้อมูลเสียงของสินค้าที่กำหนด

Text เป็นตัวที่ระบุข้อมูลข้อความที่ต้องการแปลงเป็นไฟล์เสียงเก็บไว้

เม็ทธอด: Play() จัดการหน้าที่ทางด้านข้อมูลเสียงด้านการดึงข้อมูลเสียงขึ้นมาเล่น

Convert() จัดการการแปลงข้อมูลข้อความเป็นเสียงแล้วเก็บไว้ใน Path ที่กำหนด

หน้าที่รับผิดชอบ: จัดการหน้าที่ทางด้านข้อมูลเสียง

ชื่อคลาส: Video

คุณสมบัติ: Path เป็นตัวระบุ Path ที่เก็บข้อมูลวิดีโอของสินค้าที่กำหนด

เม็ทธอด: Play() จัดการหน้าที่ทางการดึงข้อมูลวิดีโอมาแสดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่รับผิดชอบ: จัดการหน้าที่ทางการดึงข้อมูลวีดีโอมาแสดง

ชื่อคลาส: Picture

คุณสมบัติ: Name เป็นตัวเก็บชื่อของรูปภาพ

เม็ทธอด: Play() จัดการหน้าที่ทางด้านดึงข้อมูลจากรูปภาพจากฐานข้อมูลมาแสดง

หน้าที่รับผิดชอบ: จัดการการดึงข้อมูลรูปภาพจากฐานข้อมูลมาแสดง

ชื่อคลาส: ShoppingBag

คุณสมบัติ: ItemID เป็นตัวระบุหมายเลขสินค้า

AccountID เป็นตัวระบุว่าเป็นของลูกค้าคนใด

Amount เป็นตัวระบุปริมาณสินค้า

Discount เป็นตัวระบุส่วนลด

TaxValue เป็นตัวระบุภาษี

Price เป็นตัวระบุราคาของสินค้า

เม็ทธอด: DeleteItem() เป็นตัวลบสินค้าออกจาก Shopping Bag

SaveShoppingBag() เป็นส่วนจัดการการจ้ดเก็บ Shopping Bag

LoadShoppingBag() เป็นส่วนที่จัดการดึงข้อมูล Shopping Bag ขึ้นมาใช้งาน

หน้าที่รับผิดชอบ: จัดการเรื่อง Shopping Bag

ชื่อคลาส: Security

คุณสมบัติ: AccountID เป็นคุณสมบัติที่ระบุถึงหมายเลขสมาชิกที่เข้าใช้ระบบ

CusUsername เป็นคุณสมบัติที่ระบุถึงชื่อผู้ใ้

CusPassword เป็นคุณสมบัติที่ระบุถึงรหัสผ่าน

เม็ทธอด: Login() ส่วนที่จัดการเรื่องการเข้าใช้ระบบโดยตรวจสอบข้อมูลก่อนว่าถูกต้องหรือไม่

Logout() ส่วนที่จัดการเรื่องการออกจากระบบ

LoginOwner() เหมือน Login() แต่เป็นของเจ้าของร้าน

LogoutOwner() เหมือนกับ Logout() แต่เป็นของเจ้าของร้าน

หน้าที่รับผิดชอบ: จัดการเรื่องความปลอดภัยและการจำกัดสิทธิ์ในการเข้าใช้ระบบ

ชื่อคลาส: WebGenFind

หน้าที่รับผิดชอบ: จัดการการดึงข้อมูลที่จำเป็นในการสร้างเว็บแบบไดนามิกขึ้นมาใช้งาน

8.2 ฟังก์ชันการทำงานของแต่ละเว็บคลาสในระบบ

ชื่อเว็บคลาส: Main

หน้าที่รับผิดชอบ: เป็นเว็บคลาสที่จัดการหน้าจอหลักของ Shop2000 โดยดึงข้อมูลที่จำเป็นจากฐานข้อมูลเพื่อทำการสร้างโค้ด HTML ให้กับเว็บเพจ ยกตัวอย่าง เช่น สินค้าที่เป็นที่นิยม และ สินค้าใหม่ เป็นต้น

ชื่อเว็บคลาส: ClsAdminCategory

หน้าที่รับผิดชอบ: เป็นเว็บคลาสที่จัดการเกี่ยวกับประเภทของสินค้าในร้านของแต่ละร้านค้า ไม่ว่าจะเป็นการเพิ่ม ลบ หรือ อัปเดต ประเภทสินค้า

ชื่อเว็บคลาส: ClsShowMainSearchResult

หน้าที่รับผิดชอบ: เป็นเว็บคลาสที่จัดการด้านการนำเสนอผลลัพธ์การหาสินค้าและการหาร้านค้าจากหน้าจอหลัก

ชื่อเว็บคลาส: ItemDetail

หน้าที่รับผิดชอบ: เป็นเว็บคลาสที่จัดการด้านแสดงรายละเอียดของสินค้า ในที่นี้รวมถึงข้อมูลมีเดีย เช่น เสียง วิดีโอ และภาพ

ชื่อเว็บคลาส: SB (Shopping Bag)

หน้าที่รับผิดชอบ: เป็นเว็บคลาสที่จัดการเกี่ยวกับ Shopping Bag ในที่นี้รวมถึง การเก็บข้อมูล การดึงข้อมูลที่เก็บไว้ และการนำไปประมวลผลการสั่งซื้อ

ชื่อเว็บคลาส: WebClass1

หน้าที่รับผิดชอบ: เป็นเว็บคลาสที่จัดการการลงทะเบียนเป็นสมาชิกทั้งในส่วนของลูกค้าปกติ และ ลูกค้าที่ต้องการเปิดร้าน

ชื่อเว็บคลาส: WebClass2

หน้าที่รับผิดชอบ: เป็นเว็บคลาสที่จัดการการเข้าใช้ระบบของสมาชิกทั้งในส่วนของลูกค้าปกติ และ ลูกค้าที่ต้องการเปิดร้าน

ชื่อเว็บคลาส: WebClass3

หน้าที่รับผิดชอบ: เป็นเว็บคลาสที่จัดการแก้ไขเกี่ยวกับรายละเอียดของร้านค้า เช่น เจ้าของร้าน หรือ รูปแบบหน้าร้าน เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อเว็บคลาส: WebBilling

หน้าที่รับผิดชอบ: เป็นเว็บคลาสที่จัดการการประมวลผลใบสั่งซื้อ(Billing) โดยรับข้อมูลมาจากเว็บคลาส Shopping Bag อีกทีหนึ่ง

ชื่อเว็บคลาส: CIsItemList

หน้าที่รับผิดชอบ: เป็นเว็บคลาสที่จัดการแสดงรายการสินค้าทั้งหมดหลังจากเลือกประเภทสินค้า

ชื่อเว็บคลาส: CIsSearchResult

หน้าที่รับผิดชอบ: เป็นเว็บคลาสที่แสดงผลลัพธ์รายการที่ได้จากค้นหาจากคีย์เวิร์ดที่กำหนด

ชื่อเว็บคลาส: CIsOneShop

หน้าที่รับผิดชอบ: เป็นเว็บคลาสที่จัดการหน้าจอหลักของร้านแต่ละร้าน โดยดึงข้อมูลที่จำเป็นจากฐานข้อมูลเพื่อทำการสร้างโค้ด HTML ให้กับเว็บเพจ ยกตัวอย่าง เช่น รูปแบบของหน้าจอร้านค้า สินค้าที่เป็นที่นิยม และ สินค้าใหม่ เป็นต้น

ชื่อเว็บคลาส: Item

หน้าที่รับผิดชอบ: เป็นเว็บคลาสที่จัดการรายละเอียดของสินค้า ทั้งการ เพิ่ม ลบ และ อัปเดตรายละเอียดของสินค้า

ชื่อเว็บคลาส: Payment

หน้าที่รับผิดชอบ: เป็นเว็บคลาสที่จัดการจำลองสถานการณ์การจ่ายเงิน เช่น การจ่ายเงินโดยบัตรเครดิต การจ่ายเงินโดยธนาคาร เป็นต้น

ชื่อเว็บคลาส: Tracking

หน้าที่รับผิดชอบ: เป็นเว็บคลาสการติดตามผลการประมวลผลการสั่งซื้อว่าปัจจุบัน อยู่ในขั้นตอนใด

บทที่ 9

ผลการทดสอบระบบ

9.1 การทดสอบในส่วนต่าง ๆ ของระบบ

การทดสอบระบบที่ทำการพัฒนาขึ้นนี้จะทำการทดสอบในทุกส่วนฟังก์ชันการทำงานซึ่งจะแสดงดังต่อไปนี้พร้อมรูปภาพประกอบ

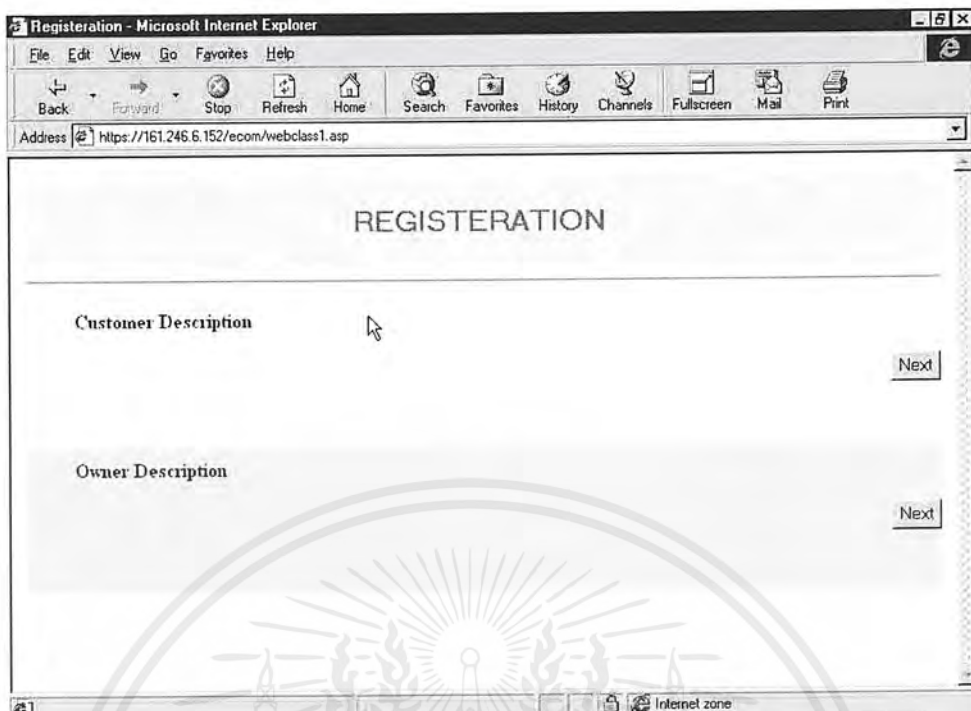
1. เราจะทำการทดสอบระบบโดยในขั้นตอนแรกต้องทำการเข้าใช้ระบบก่อน โดยเข้าไปที่ <http://161.246.6.152/ecom> ซึ่งหลังจากการเข้าใช้ระบบจะมีหน้าจอปรากฏดังรูปที่ 9-1



รูปที่ 9-1 แสดงหน้าจอของร้านค้าที่พัฒนาขึ้น

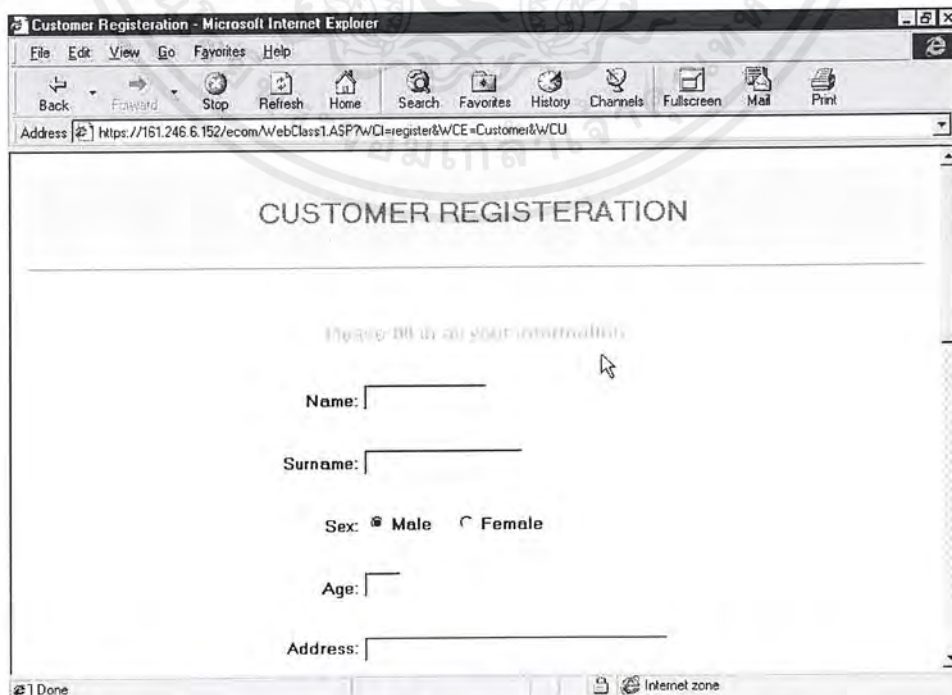
2. จะเป็นส่วนที่ทดสอบเกี่ยวกับการลงทะเบียนเป็นสมาชิกของระบบ ทั้งในส่วนของการเป็นลูกค้าที่เข้าในเว็บเพื่อเลือกซื้อสินค้าต่าง ๆ จากร้านค้าที่มีอยู่มากมาย หรือจะเป็นในส่วนผู้ที่ต้องการเปิดร้านค้าเป็นของตนเองซึ่งให้เราเลือกที่ลิงค์ "Register" หลังจากนั้นจะมีหน้าจอปรากฏให้เราเลือกประเภทของการลงทะเบียนดังรูปที่ 9-2 ซึ่งส่วนบนเป็นการลงทะเบียนแบบลูกค้า ในส่วนล่างเป็นการลงทะเบียนแบบเจ้าของร้าน ซึ่งในการเริ่มขั้นตอนการลงทะเบียนนี้จะมีการเข้ารหัสข้อมูลด้วยมาตรฐาน SSL 40 บิต ดังจะมีรูปกุญแจทองที่ทาสถ์บาร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 9-2 แสดงหน้าจอเริ่มการลงทะเบียนโดยมีการเข้ารหัสข้อมูลในการสื่อสาร

3. ในส่วนต่อไปเป็นการทดสอบการกรอกข้อมูลจริงโดยให้เลือกรลงทะเบียนในโหมดของลูกค้าที่เลือกซื้อสินค้าจากทางระบบเท่านั้นซึ่งจะมีหน้าจอที่ปรากฏขึ้นมาเพื่อให้กรอกข้อมูลดังรูปที่ 9-3 ซึ่งเราต้องกรอกข้อมูลให้ครบและถูกต้องในกฎพื้นฐาน เช่น รหัสผ่านที่กรอกครั้งแรกกับยืนยันต้องเหมือนกันมิฉะนั้นจะไม่ยอมให้ผ่าน นอกจากนี้ส่วนของอีเมลซึ่งเป็นส่วนที่ใช้แทนรหัสผู้ใช้ต้องจำไว้เนื่องจากใช้ในการเข้าใช้ระบบต่อไป



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์รูปที่ 9-3 แสดงหน้าจอเพื่อให้กรอกข้อมูลการลงทะเบียนสำหรับลูกค้าใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การทดสอบต่อไปเป็นการทดสอบในส่วนของการลงทะเบียนในฐานะการเป็นเจ้าของร้านโดยเลือกซึ่งจะมีหน้าจอในการลงทะเบียนที่คล้ายกับการลงทะเบียนเป็นลูกค้ามาก ซึ่งเราต้องกรอกข้อมูลที่จำเป็นให้ครบ ซึ่งรวมถึงการเลือกหน้าจอของร้านค้าได้ตามต้องการตามที่ระบบกำหนดรูปแบบพื้นฐานไว้ให้เลือก ซึ่งแสดงดังรูปที่ 9-4

The screenshot shows a web browser window titled "Owner Registration - Microsoft Internet Explorer". The address bar contains the URL: "https://161.246.6.152/ecom/WebClass1.ASP?WC=register&WCE=Owner&WCU". The main content area displays a form titled "OWNER REGISTRATION" with a large watermark of a Thai university seal in the background. The form fields are as follows:

- Shop_name: _____
- Name: _____
- Surname: _____
- Sex: Male Female
- Age: _____

At the bottom of the browser window, there is a status bar showing "Internet zone" with a lock icon.

รูปที่ 9-4 แสดงการทดสอบการลงทะเบียนในฐานะเจ้าของร้าน

5. ขั้นตอนต่อไปที่เราจะทดสอบเป็นขั้นตอนในการเข้าใช้ระบบในรูปแบบต่าง ๆ ซึ่งต้องมีการกรอกข้อมูลคือรหัสผู้ใช้และรหัสผ่านให้ถูกต้องกับที่ลงทะเบียนไว้ ถ้าไม่ถูกต้องระบบจะแจ้งเตือนกลับมาว่าท่านไม่สามารถเข้าใช้ระบบได้ โดยการเข้าใช้ระบบก็จะถูกแบ่งออกเป็น 2 รูปแบบเช่นเดียวกับการลงทะเบียนคือ แบบลูกค้าทั่วไป กับ แบบเจ้าของร้านหลังจากทำการกรอกข้อมูลเสร็จให้กดปุ่ม Login โดยการส่งข้อมูลนี้ต้องมีการเข้ารหัสข้อมูลเช่นเดียวกันเพื่อความปลอดภัยในการใช้งาน เนื่องจากถ้าไม่มีการเข้ารหัสข้อมูลจะทำให้ข้อมูลสามารถดักฟัง หรือ แอบดูได้จากโปรแกรม Sniffer ทั่วไป ดังแสดงในรูปที่ 9-5

Customer Login

Email:

Password:

Login Reset

Shop Owner Login

Username:

Password:

Login Reset

รูปที่ 9-5 แสดงการล็อกอินเข้าใช้งานระบบในรูปแบบต่างๆ

6. ต่อไปเป็นการทดสอบในส่วนล็อกอินเพื่อการแก้ไขข้อมูลส่วนตัวและการดูรายละเอียด Shopping Bag ซึ่งแสดงดังรูปที่ 9-6

LOGIN FOR UPDATE YOUR INFORMATION

Login for update your account

Email:

Password:

Login Reset

Login for see your shopping bag

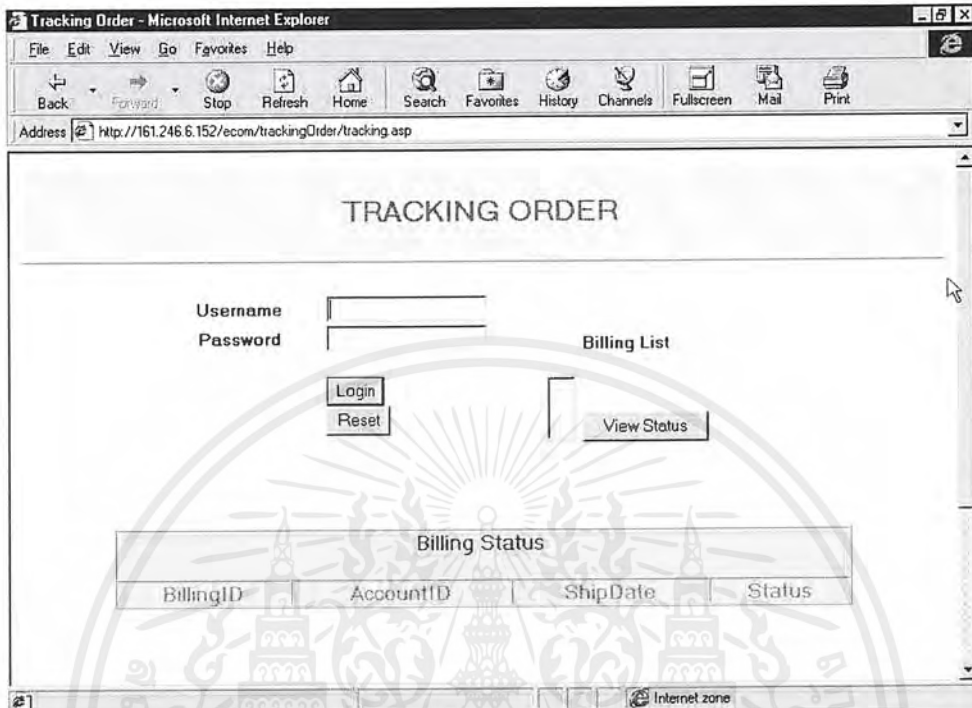
Email:

Password:

รูปที่ 9-6 แสดงการล็อกอินเข้าใช้ระบบเพื่อปรับเปลี่ยนข้อมูลและดูรายละเอียด Shopping Bag

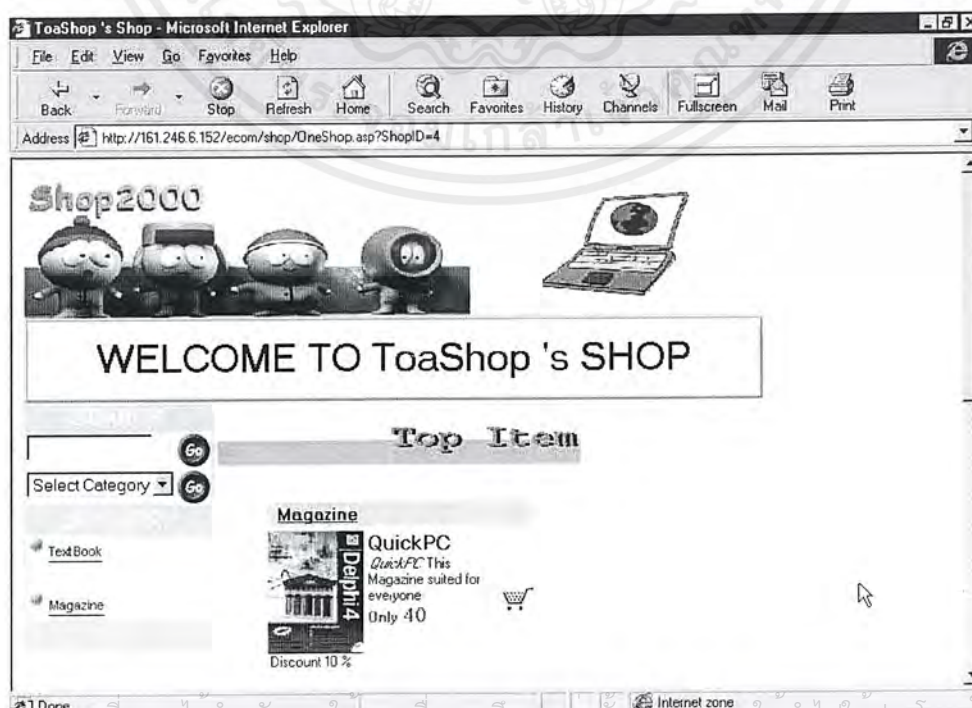
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. การทดสอบในส่วนต่อไปเป็นการทดสอบการตรวจสอบใบสั่งซื้อของผู้ใช้โดยต้องมีการล็อกอินเข้าสู่ระบบก่อนซึ่งแสดงดังรูป 9-7



รูปที่ 9-7 แสดงส่วนของระบบในการติดตามว่าใบสั่งซื้อมีสถานะเป็นอย่างไร

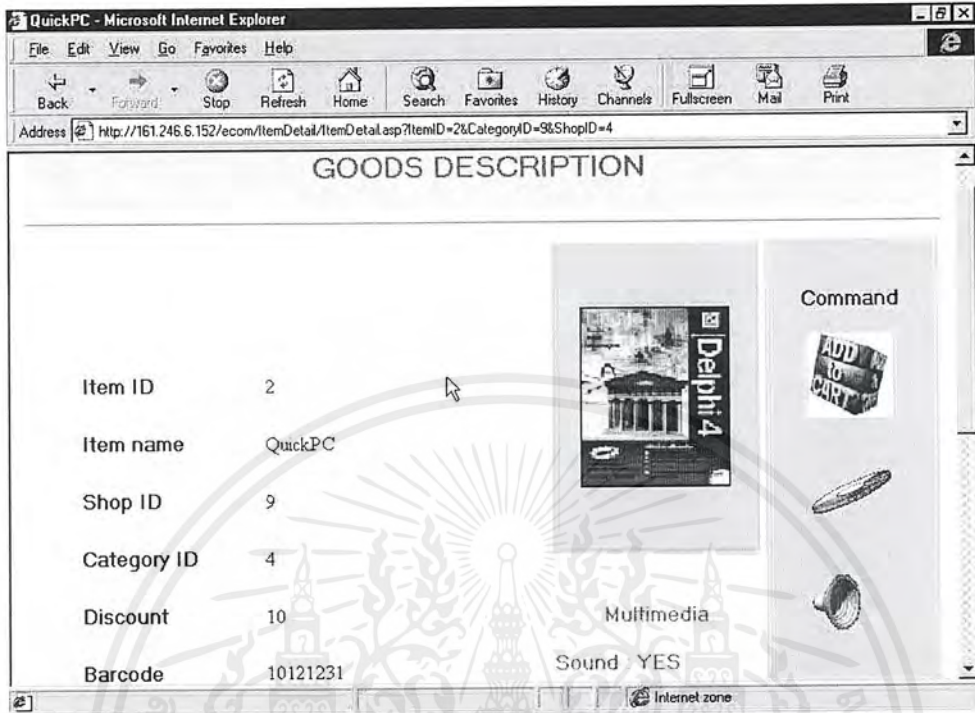
8. การทดสอบในส่วนต่อไปเป็นการทดสอบในการเข้าไปในร้านต่าง ๆ ซึ่งมาลงทะเบียนเป็นสมาชิก ซึ่งเป็นดังรูปที่ 9-8



เอกสารนี้เป็นลิขสิทธิ์ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีก

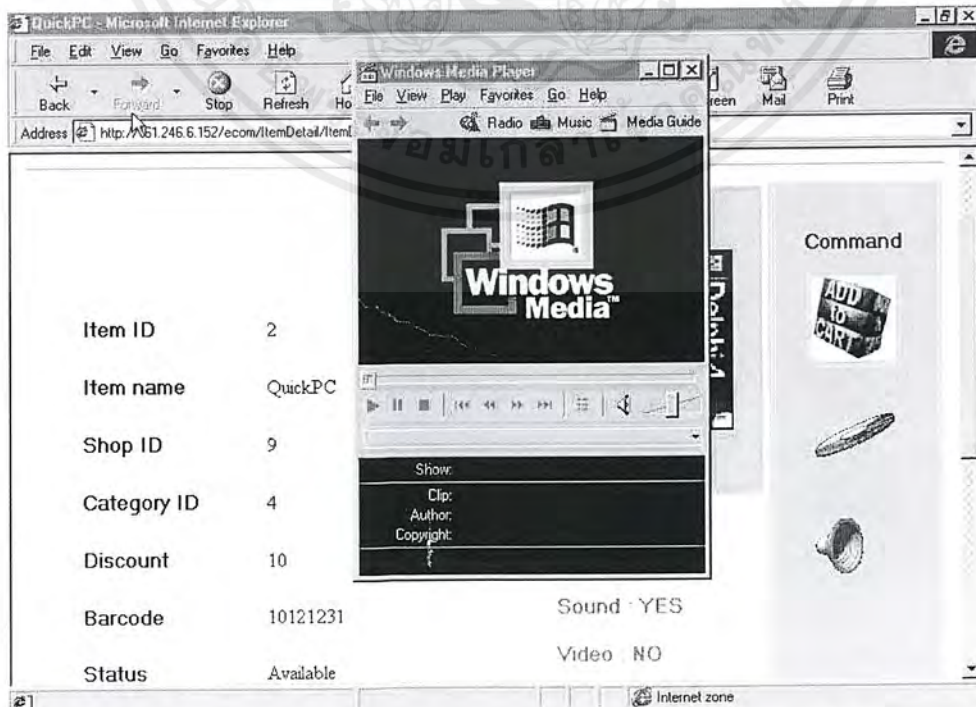
รูปที่ 9-8 แสดงการทดสอบการเข้าไปร้านแต่ละร้านที่เข้ามาเปิดกิจการได้ซึ่งที่มีการนำไปใช้

9. เป็นการทดสอบในส่วนรายละเอียดของสินค้าที่ต้องการซึ่งแสดงดังรูปที่ 9-9



รูปที่ 9-9 แสดงการเข้าไปดูรายละเอียดของสินค้าแต่ละชนิด

10. การทดสอบในส่วนต่อไปเป็นการทดสอบการเรียกดูข้อมูลมัลติมีเดียของสินค้าทั้งในรูปเสียงและวิดีโอ ซึ่งจะมีการเรียก Window Media Player ขึ้นมาใช้งานอีกทีหนึ่ง ดังแสดงในรูปที่ 9-10



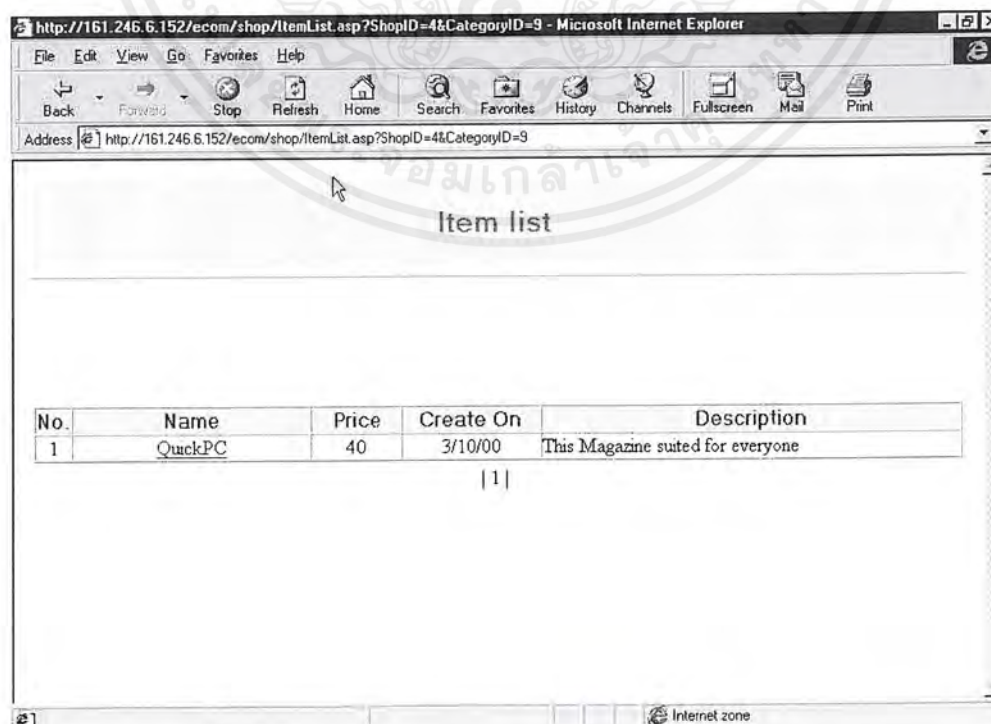
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้แก้ไขประโยชน์ด้านการค้า
รูปที่ 9-10 แสดงการเรียกดูข้อมูลของสินค้าในส่วนของมัลติมีเดียสตรีมมิ่ง
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. การทดสอบในส่วนการจะนำสินค้าไปใส่ Shopping Bag ซึ่งถ้าไม่ได้ล็อกอินเข้าสู่ระบบต้องทำการล็อกอินก่อน เพื่อดึงข้อมูลเดิมขึ้นมาด้วยถ้ามี ดังรูปที่ 9-11



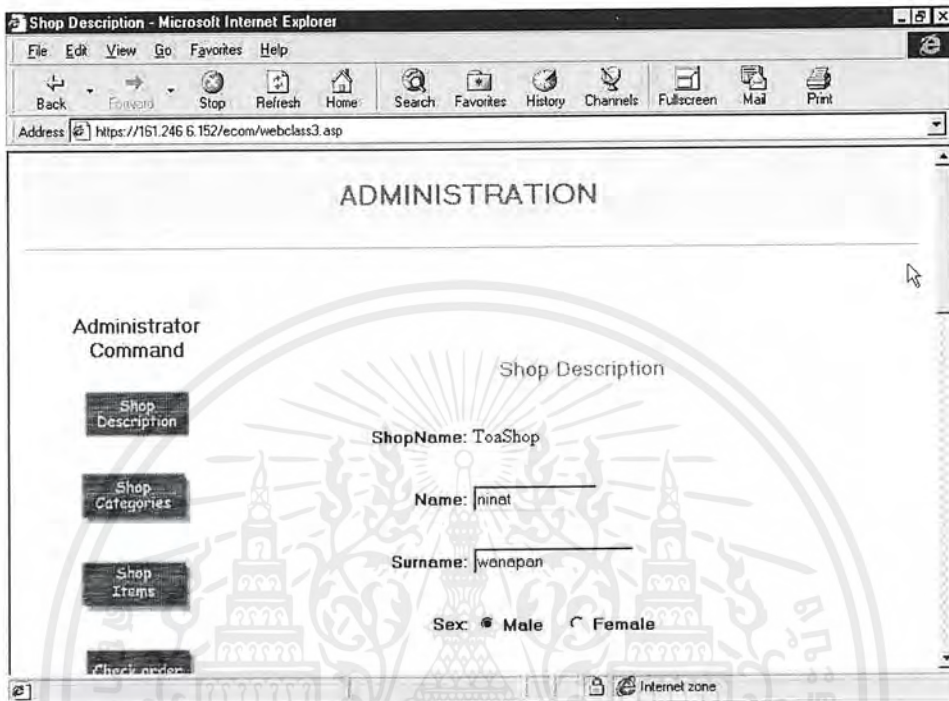
รูปที่ 9-11 แสดงการเลือกสินค้าไปไว้ใน Shopping Bag โดยที่ไม่ได้ล็อกอินต้องล็อกอินก่อน

12. เป็นการทดสอบในเรื่องการค้นหาคำในในระบบ โดยเมื่อมีการค้นหาจากคีย์เวิร์ดที่ต้องการระบบจะให้ผลลัพธ์กลับมาดังรูปพร้อมกับมีลิงค์เชื่อมโยงไปดูรายละเอียดของสินค้าเพิ่มเติม ดังรูปที่ 9-12



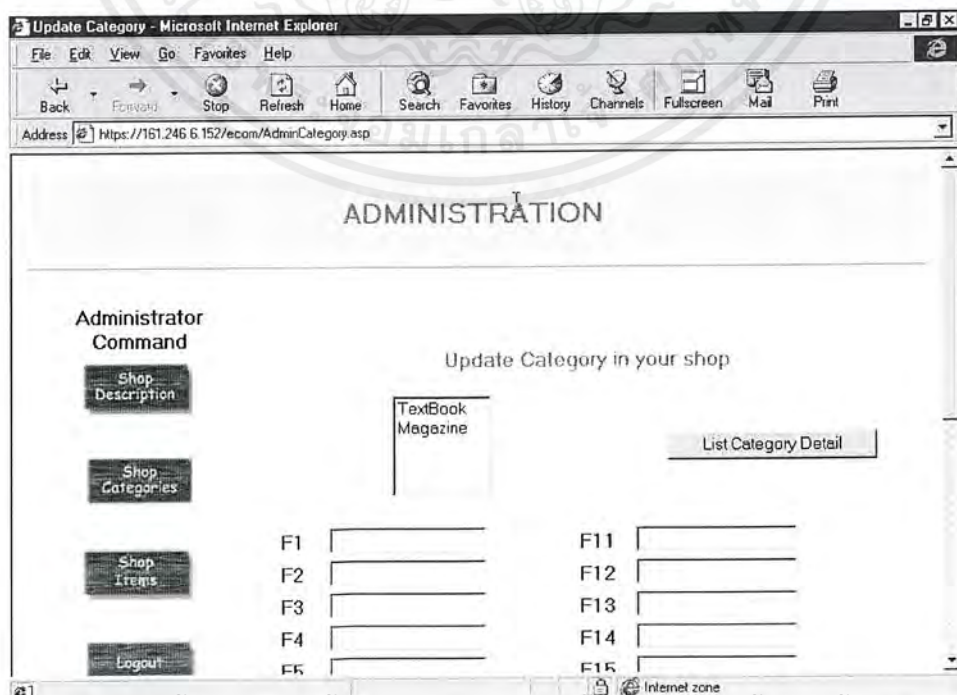
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่ขอเอาผิดในแง่ไปใช้ประโยชน์ด้านการค้า
รูปที่ 9-12 แสดงผลลัพธ์การใช้ฟังก์ชันการค้นหาสินค้าที่ต้องการในร้าน
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13. การทดสอบส่วนการจัดการต่าง ๆ ในร้านค้าของเจ้าของร้าน โดยเมื่อมีการเข้าใช้ระบบในรูปแบบเจ้าของร้านจะมีหน้าจอ ขึ้นมาแสดงรายละเอียดเบื้องต้นที่เกี่ยวกับร้านเพื่อให้สามารถแก้ไขข้อมูลของร้านได้ ซึ่งแสดงดังรูปที่ 9-13



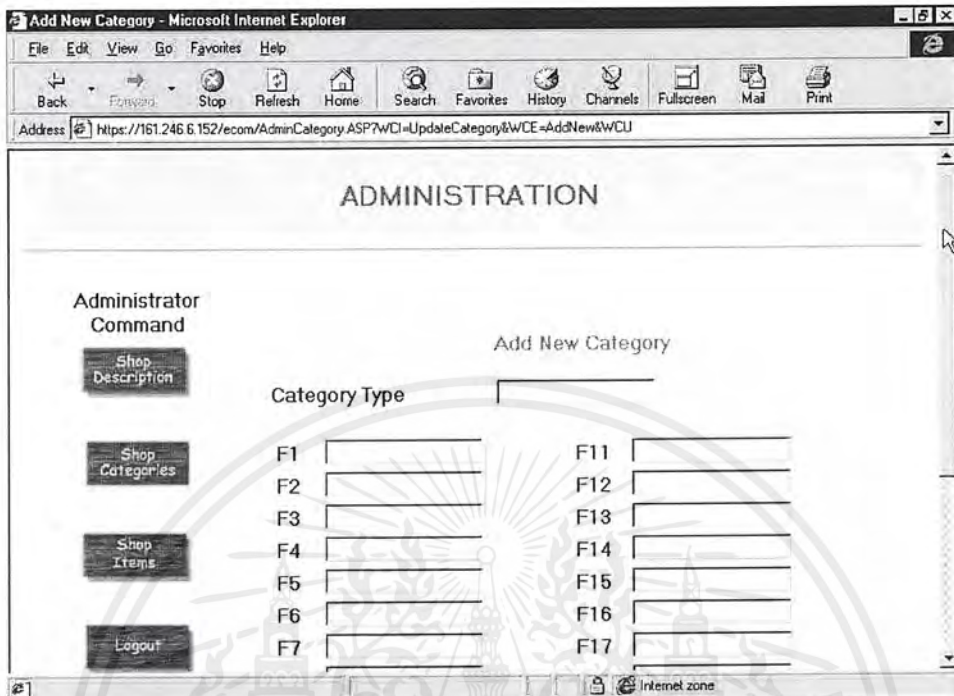
รูปที่ 9-13 แสดงการเข้าใช้งานระบบเพื่อจัดการร้านค้าของเจ้าของร้าน

14. การทดสอบส่วนการเข้าจัดการประเภทของสินค้าในร้าน โดยเมื่อเข้าไปจะเป็นการอัปเดตข้อมูลประเภทสินค้าที่มีอยู่แล้ว ซึ่งสามารถเปลี่ยนแปลงข้อมูลที่เก็บของแต่ละประเภทสินค้าได้ดังรูปที่ 9-14



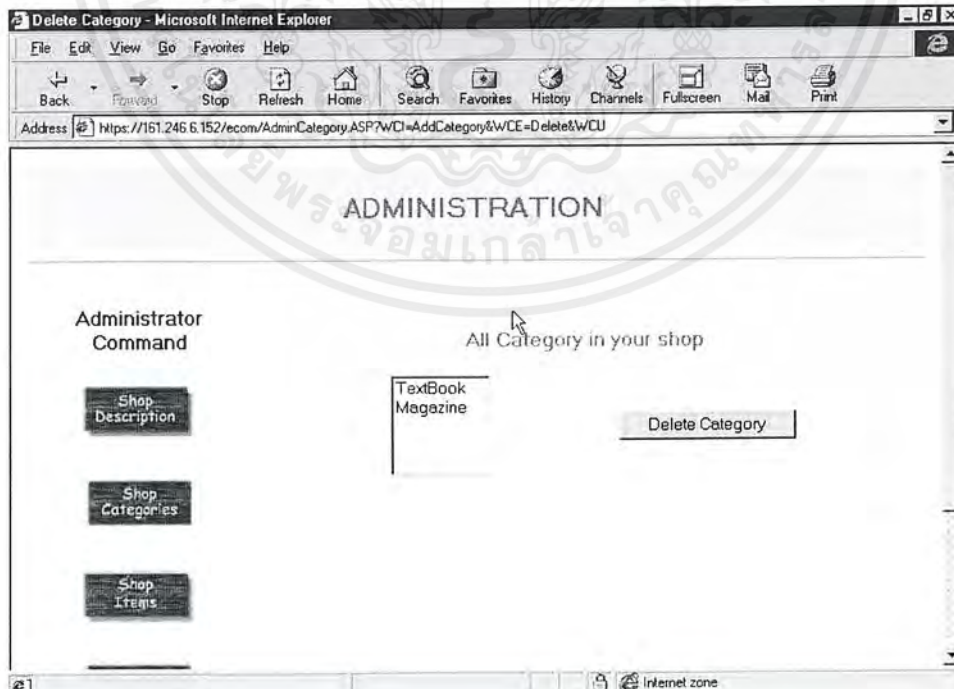
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 9-14 แสดงการจัดการในส่วนของประเภทสินค้าในร้านโดยสามารถอัปเดตข้อมูลได้
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

15. การทดสอบส่วนการเพิ่มประเภทของสินค้าในร้านใหม่โดยกรอกข้อมูลที่จะเก็บของประเภทสินค้านั้น ๆ ดังแสดงดังรูป 9-15



รูปที่ 9-15 แสดงการทดสอบการเพิ่มประเภทของสินค้าในร้าน

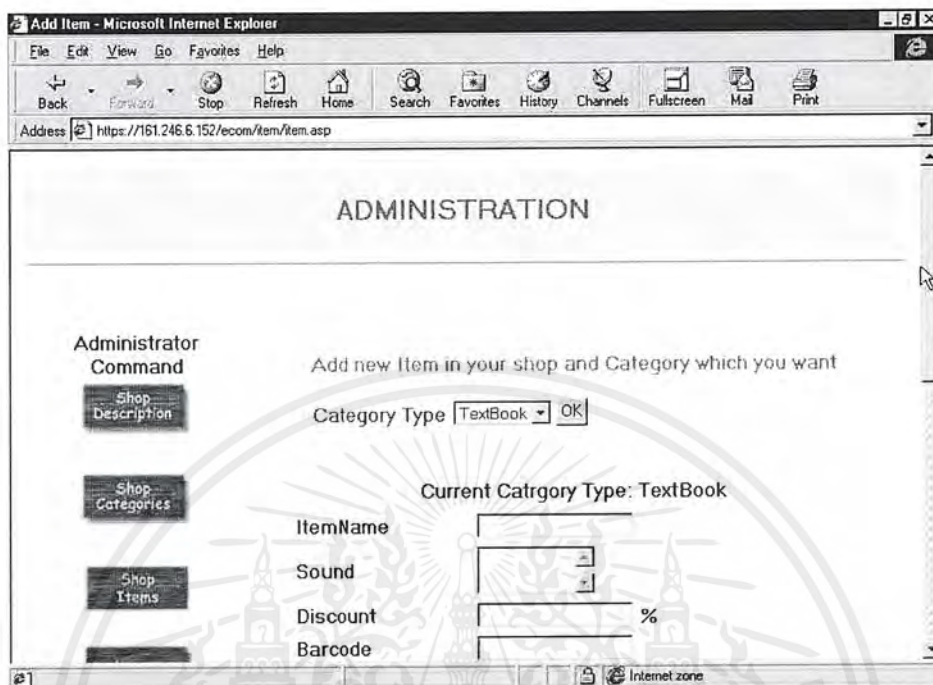
16. การทดสอบส่วนการลบประเภทของสินค้าจากร้านค้าซึ่งแสดงดังรูปที่ 9-16



รูปที่ 9-16 แสดงการลบประเภทของสินค้าที่ไม่ต้องการในร้าน

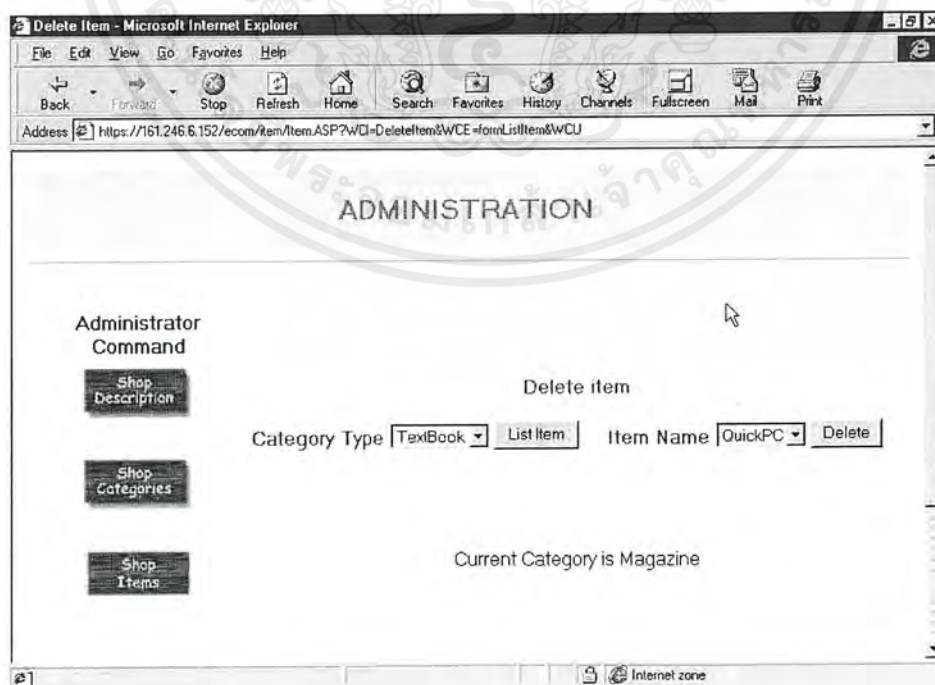
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

17. การทดสอบส่วนการจัดการด้านสินค้าโดยเป็นการเพิ่มสินค้าเข้าไปในประเภทของสินค้าที่ต้องการ ซึ่งแสดงดังรูปที่ 9-17



รูปที่ 9-17 แสดงการจัดการจัดการในส่วนของสินค้าโดยเพิ่มสินค้าในประเภทที่ต้องการ

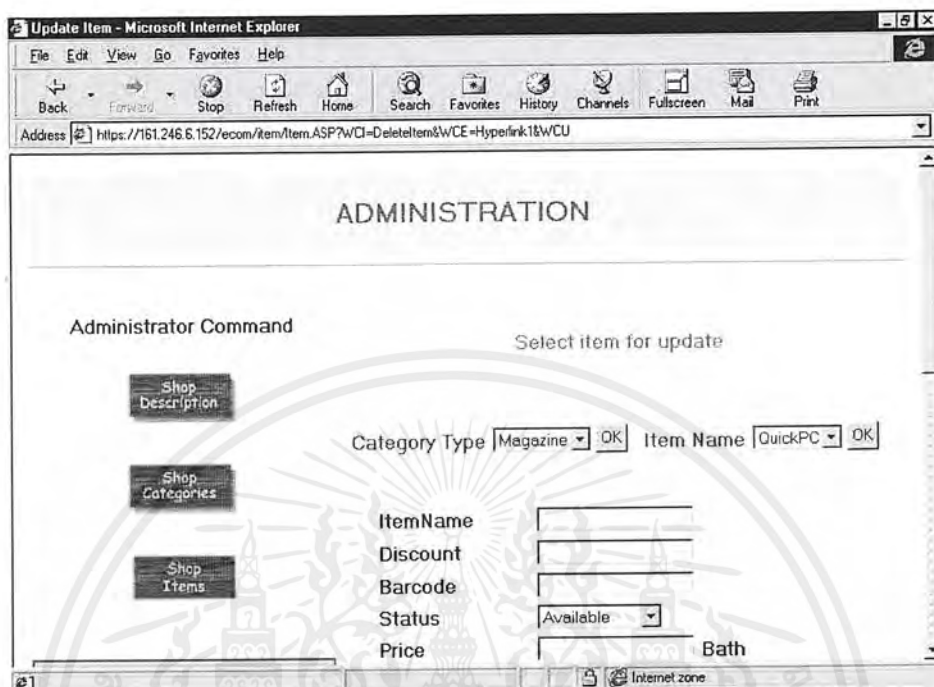
18. การทดสอบส่วนการจัดการด้านสินค้าโดยเป็นการลบสินค้าออกในประเภทของสินค้าที่ต้องการ ซึ่งแสดงดังรูปที่ 9-18



รูปที่ 9-18 แสดงการลบสินค้าสินค้าในประเภทที่ต้องการ

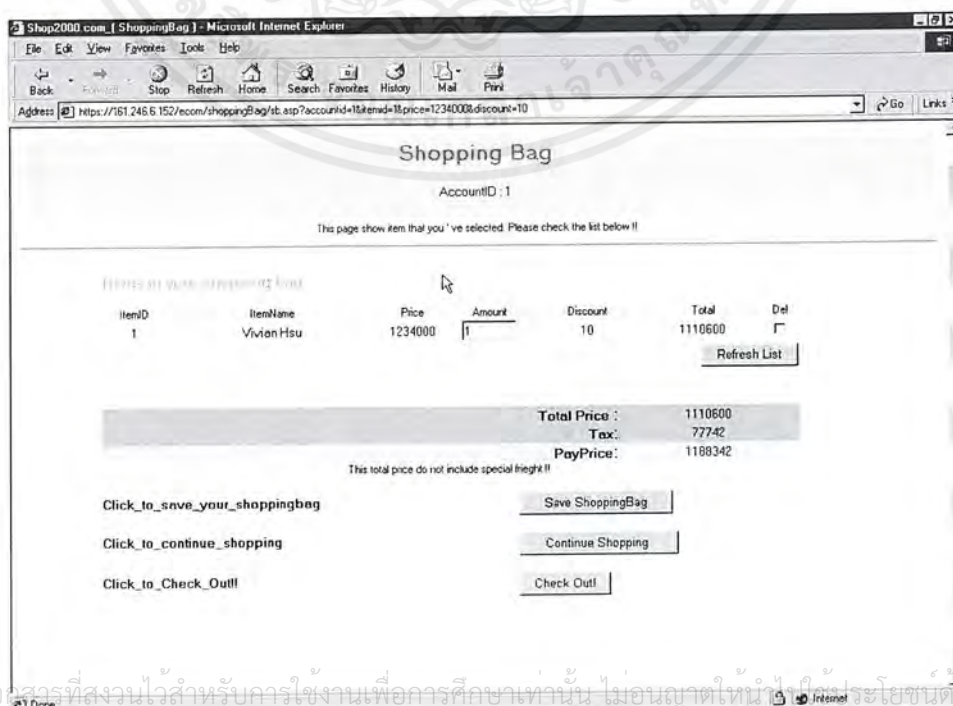
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

19. การทดสอบส่วนการจัดการด้านสินค้าโดยเป็นการอัปเดตข้อมูลสินค้าที่ต้องการ ซึ่งแสดงดังรูปที่ 9-19



รูปที่ 9-19 แสดงการทดสอบโดยอัปเดตข้อมูลสินค้าเดิมในประเภทที่ต้องการ

20. การทดสอบส่วนใช้ Shopping Bag โดยการเลือกสินค้าที่ต้องการเข้าไป นอกจากนี้ยังสามารถเก็บข้อมูลไว้ในการเพื่อใช้ในการเข้าใช้ระบบครั้งต่อไปได้ หรือ Check Out เพื่อไปประมวลผลใบสั่งซื้อ ซึ่งแสดงดังรูปที่ 9-20



รูปที่ 9-20 แสดงการทดสอบในส่วนของการใช้ Shopping Bag

21. การทดสอบในส่วนการประมวลผลใบสั่งซื้อโดยผู้ซื้อต้องกรอกรายละเอียดต่าง ๆ ที่กำหนดเพื่อใช้ในการประมวลผลต่อไปโดยแสดงดังรูปที่ 9-21

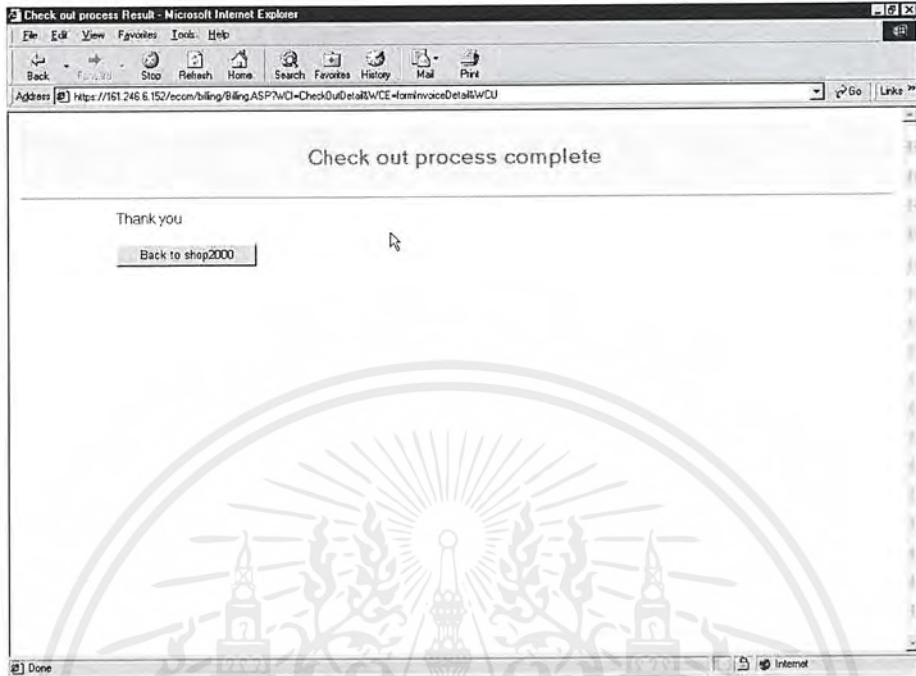
รูปที่ 9-21 แสดงการทดสอบในส่วนการทำใบสั่งซื้อ

22. หลังจากนั้นจะมีการแสดงรายละเอียดสรุปของการสั่งซื้อพร้อมทั้งให้ลูกค้าเลือกวิธีการจ่ายเงิน ดังแสดงในรูปที่ 9-22

No	Item name	Qty	Price	Status
1	Mylan Hsu	1	1320369	Available
	Total		1320369	
	Tax		66379.3	
	Freight		20.25	
	Grand Total		1320390	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ผู้ใช้ไปใช้ประโยชน์ด้านการค้า
รูปที่ 9-22 แสดงการทดสอบขั้นตอนการประมวลผลการสั่งซื้อ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

23. หลังจากนั้นจะมีการส่งเพจและอีเมลล์ตอบรับการสั่งซื้อซึ่งต้องทำการกดปุ่ม “Back to Shop2000” ก่อน โดยจะมีหน้าจอแสดงผลการส่งเพจด้วย ซึ่งแสดงดังรูปที่ 9-23



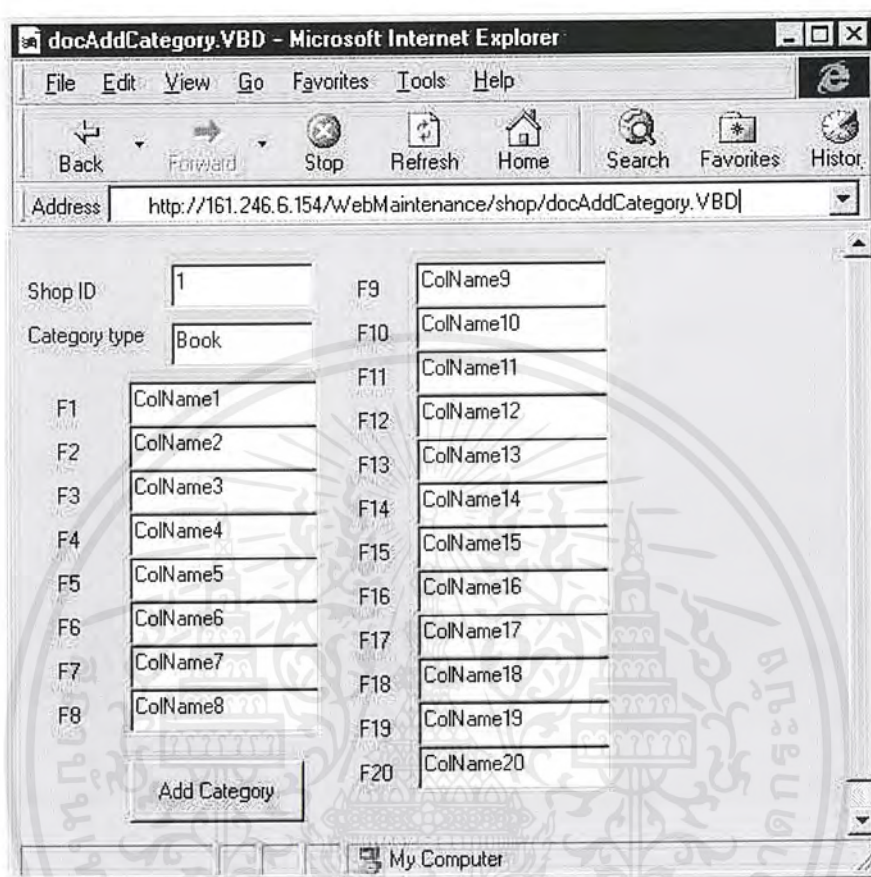
รูปที่ 9-23 แสดงหน้าจอเมื่อการประมวลผลการสั่งซื้อเสร็จเรียบร้อยแล้ว

24. การทดสอบในส่วน ActiveX Document ที่ทำหน้าที่ในการเพิ่มผู้ใช้ให้กับระบบซึ่งแสดงดังรูปที่ 9-24

Name	Niran	Province	Bangkok
Surname	Luckcanakul	Postcode	10310
Email	boy@yahoo.com	Country	Thailand
Password	boy	Tel	933-7606
Sex	male	Age	20
Occupation	Engineer	Fax	933-7606
Address	42 Ladphrao		
Subprovince	Bangkapi		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้านการค้า
รูปที่ 9-24 แสดงการใช้ ActiveX Document ในส่วนของการเพิ่ม ผู้ใช้ระบบ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

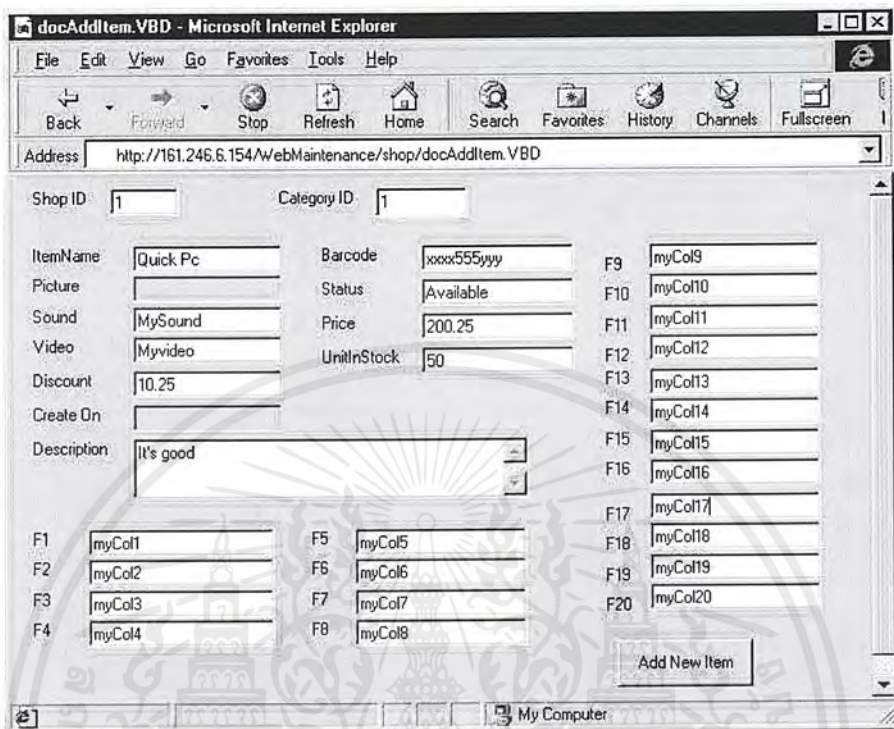
25. การทดสอบในส่วน ActiveX Document ที่ทำหน้าที่ในการเพิ่มประเภทสินค้าในร้านต่าง ๆ ซึ่งแสดงดังรูปที่ 9-25



รูปที่ 9-25 แสดงการใช้ ActiveX Document ในส่วนของการเพิ่ม ประเภทของสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

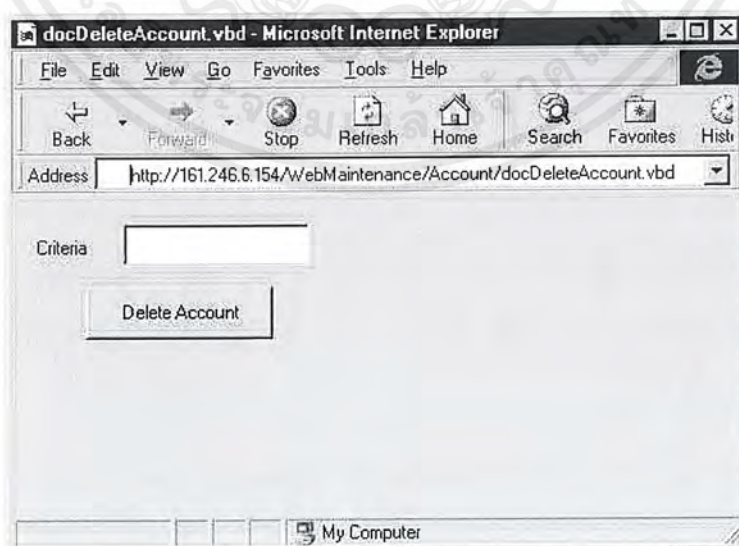
26. การทดสอบในส่วน ActiveX Document ที่ทำหน้าที่ในการเพิ่มเพิ่มสินค้าเข้าไปในร้านต่าง ๆ ซึ่งแสดงดังรูปที่ 9-26



รูปที่ 9-26 แสดงการใช้ ActiveX Document ในส่วนของการเพิ่ม ผู้ใช้ระบบ

27. การทดสอบในส่วน ActiveX Document ที่ทำหน้าที่ในการลบผู้ใช้ระบบซึ่งแสดงดังรูปที่

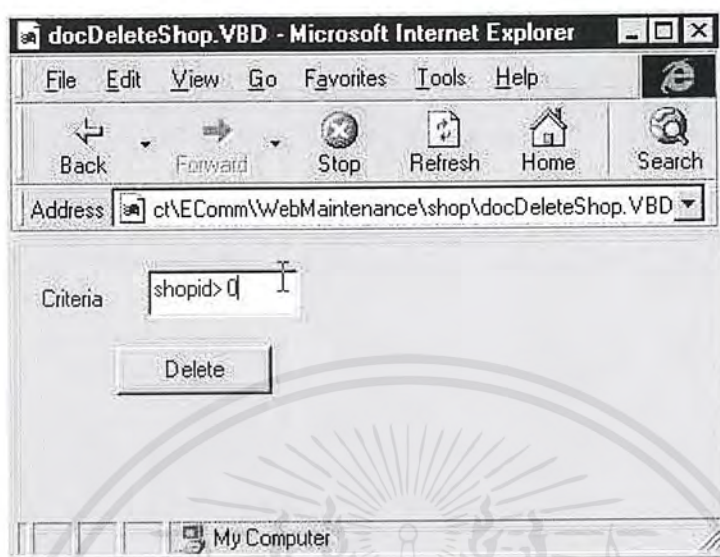
9-27



รูปที่ 9-27 แสดงการใช้ ActiveX Document ในส่วนของการลบ ผู้ใช้ระบบ

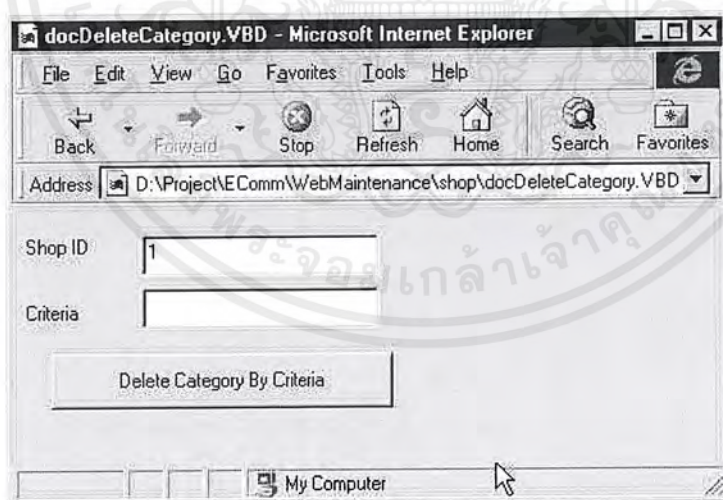
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

28. การทดสอบในส่วน ActiveX Document ที่ทำหน้าที่ในการลบร้านออกจากระบบ ซึ่งแสดงดังรูปที่ 9-28



รูปที่ 9-28 แสดงการใช้ ActiveX Document ในส่วนของการลบร้านออกจากระบบ

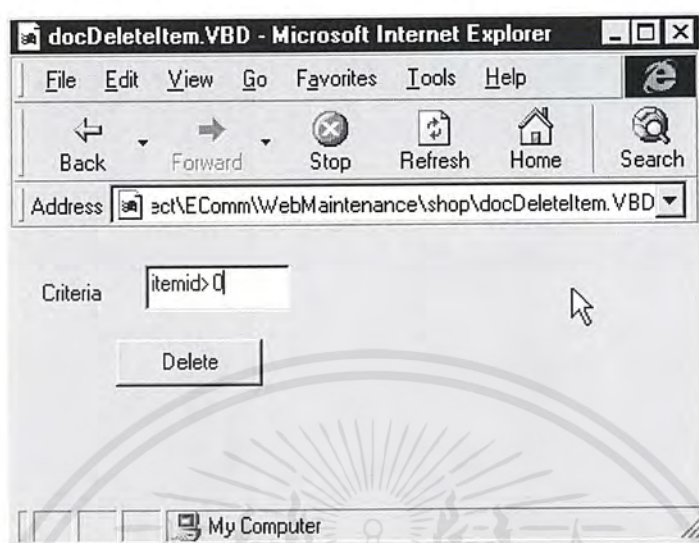
29. การทดสอบในส่วน ActiveX Document ที่ทำหน้าที่ในการประเภทของสินค้าออกจากระบบ ซึ่งแสดงดังรูปที่ 9-29



รูปที่ 9-29 แสดงการใช้ ActiveX Document ในส่วนของการลบประเภทสินค้า

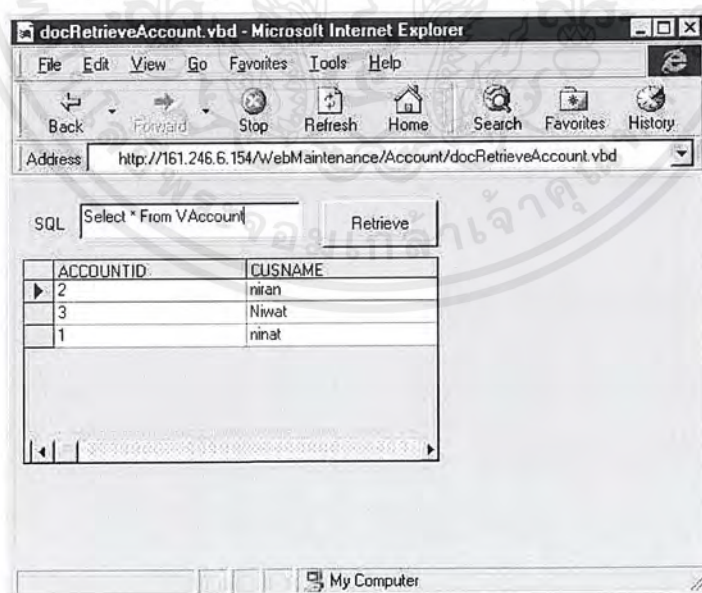
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

30. การทดสอบในส่วน ActiveX Document ทำหน้าที่ในการลบสินค้าออกจากระบบซึ่งแสดงดังรูปที่ 9-30



รูปที่ 9-30 แสดงการใช้ ActiveX Document ในส่วนของการลบสินค้า

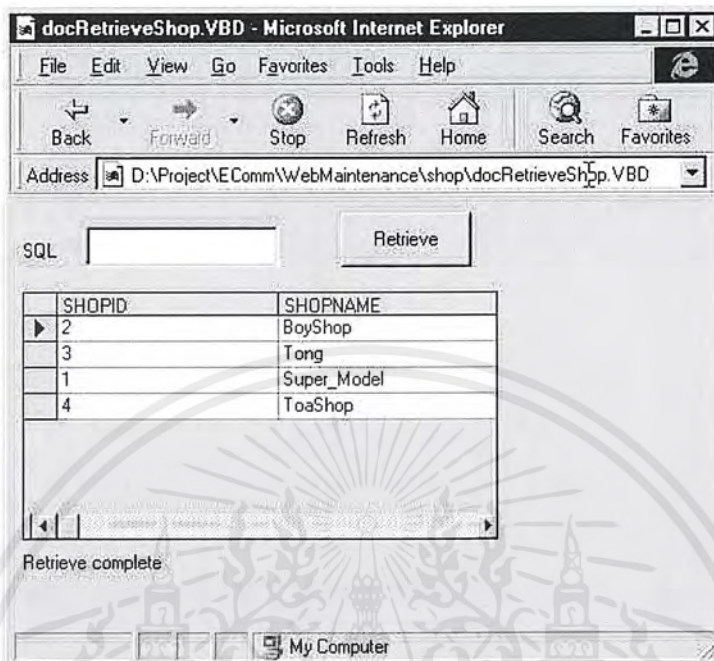
31. การทดสอบในส่วน ActiveX Document ที่ทำหน้าที่ในการดึงข้อมูลผู้ใช้งานระบบขึ้นมา ซึ่งแสดงดังรูปที่ 9-31



รูปที่ 9-31 แสดงการใช้ ActiveX Document ในส่วนของการดึงข้อมูลลูกค้าขึ้นมาดู

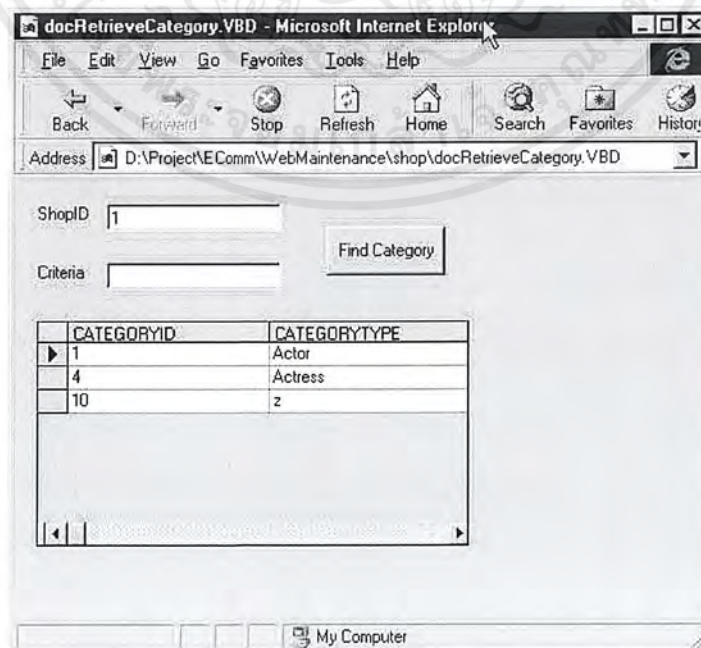
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

32. การทดสอบในส่วน ActiveX Document ที่ทำหน้าที่ในการดึงข้อมูลร้านในระบบขึ้นมาใช้งาน ซึ่งแสดงดังรูปที่ 9-32



รูปที่ 9-32 แสดงการใช้ ActiveX Document ในส่วนของการดึงข้อมูลร้าน

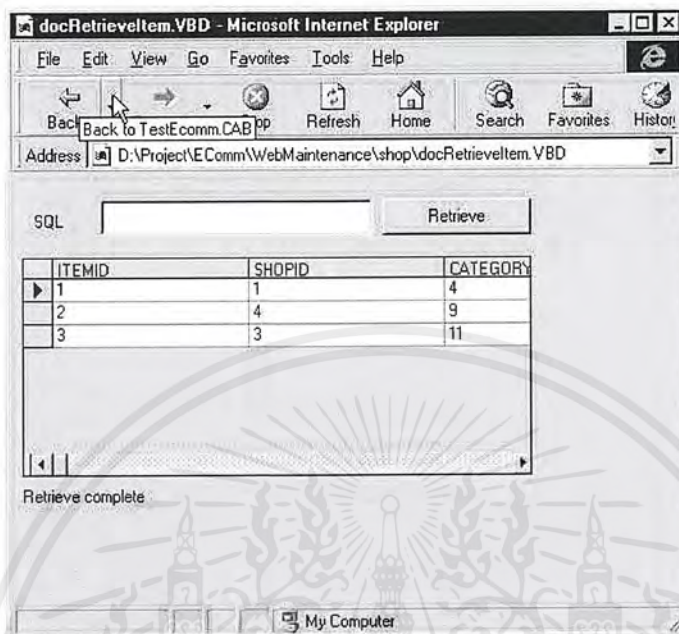
33. การทดสอบในส่วน ActiveX Document ที่ทำหน้าที่ในการดึงข้อมูลประเภทของสินค้าในร้านขึ้นมาใช้งาน ซึ่งแสดงดังรูปที่ 9-33



รูปที่ 9-33 แสดงการใช้ ActiveX Document ในส่วนของการดึงข้อมูลประเภทของสินค้า

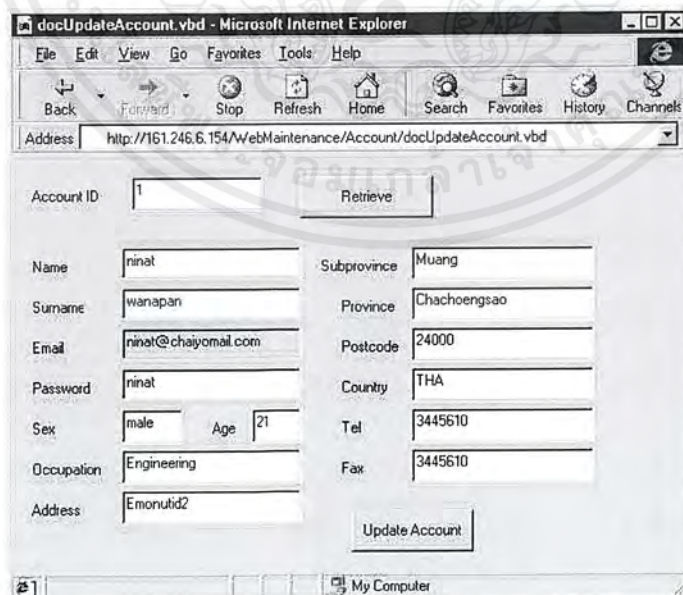
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปเผยแพร่ขึ้นด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

34 . การทดสอบในส่วน ActiveX Document ที่ทำหน้าที่ในการดึงข้อมูลสินค้าร้านในระบบขึ้นมาใช้งานโดยมีการคิวรีข้อมูลตรง ๆ โดยใช้ภาษา SQL ซึ่งแสดงดังรูปที่ 9-34



รูปที่ 9-34 แสดงการใช้ ActiveX Document ในส่วนของการดึงข้อมูลสินค้า

35 . การทดสอบในส่วนการอัปเดตข้อมูลผู้ใช้ระบบโดยต้องมีการดึงข้อมูลของลูกค้าที่ต้องการอัปเดตขึ้นมาก่อน ซึ่งแสดงดังรูปที่ 35



รูปที่ 9-35 แสดงการใช้ ActiveX Document ในส่วนของการอัปเดตข้อมูลลูกค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

36. การทดสอบในส่วน ActiveX Document ที่ทำหน้าที่อัปเดตข้อมูลของร้านค้า โดยต้องมีการดึงข้อมูลของร้านที่จะอัปเดตขึ้นมาก่อน ซึ่งแสดงดังรูปที่ 9-36

docUpdateShop.VBD - Microsoft Internet Explorer

File Edit View Go Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen

Address: D:\Project\IE Comm\WebMaintenance\shop\docUpdateShop.VBD

ShopID: 1 Retrieve

SHOPNAME: OCCUPATION: Engineering

OWNERNAME: Nwt ADDRESS: 114/80

OWNERSURNAME: Lvncr PROVINCE: Bangkok

EMAIL: moosub@chaiyo.com SUBPROVINCE: Bangbon

PASSWORD: z POSTCODE: 10150

USERNAME: COUNTRY: THA

SEX: male TEL: 1111222

AGE: 20 FAX: 123456789

WEBTEMPLATE: Template2

Update

My Computer

รูปที่ 9-36 แสดงการใช้ ActiveX Document ในส่วนของการอัปเดตข้อมูลร้าน

37. การทดสอบในส่วน ActiveX Document ที่ทำหน้าที่อัปเดตข้อมูลประเภทสินค้าในร้านค้า โดยต้องมีการดึงข้อมูลของประเภทสินค้าที่จะอัปเดตขึ้นมาก่อน ซึ่งแสดงดังรูปที่ 9-37

docUpdateCategory.VBD - Microsoft Internet Explorer

File Edit View Go Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen

Address: D:\Project\IE Comm\WebMaintenance\shop\docUpdateCategory.VBD

Shop ID: 1 CategoryID: Retrieve

Category type: Actor

F1	Name	F11	
F2	Age	F12	
F3	Height	F13	
F4	Weight	F14	
F5	Hobby	F15	
F6		F16	
F7		F17	
F8		F18	
F9		F19	
F10		F20	

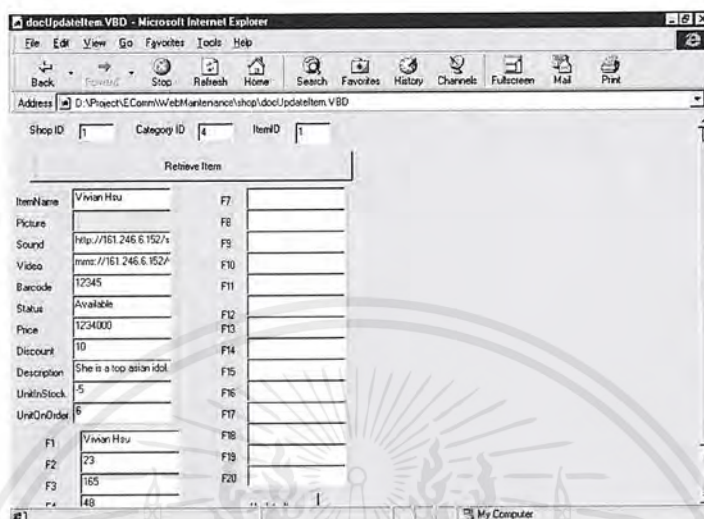
Update Category

My Computer

รูปที่ 9-37 แสดงการใช้ ActiveX Document ในส่วนของการอัปเดตข้อมูลประเภทสินค้าในร้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

38. การทดสอบในส่วน ActiveX Document ที่ทำหน้าที่อัปเดตข้อมูลสินค้าในร้านค้าโดยต้องมีการดึงข้อมูลสินค้าของร้านที่จะอัปเดตขึ้นมาก่อน ซึ่งแสดงดังรูปที่ 9-38



รูปที่ 9-38 แสดงการใช้ ActiveX Document ในส่วนของการอัปเดตข้อมูลสินค้าในร้าน

9.2 ผลการทดสอบในส่วนต่าง ๆ ของระบบ

จากที่ได้ทำการทดสอบระบบในส่วนต่าง ๆ สามารถสรุปผลได้ดังตาราง 9-1

กรณีการทดสอบ	ผลการทดสอบ
กรณีการทดสอบที่ 1	ถูกต้อง
กรณีการทดสอบที่ 2	ถูกต้อง
กรณีการทดสอบที่ 3	ถูกต้อง
กรณีการทดสอบที่ 4	ถูกต้อง
กรณีการทดสอบที่ 5	ถูกต้อง
กรณีการทดสอบที่ 6	ถูกต้อง
กรณีการทดสอบที่ 7	ถูกต้อง
กรณีการทดสอบที่ 8	ถูกต้อง
กรณีการทดสอบที่ 9	ถูกต้อง
กรณีการทดสอบที่ 10	ถูกต้อง
กรณีการทดสอบที่ 11	ถูกต้อง
กรณีการทดสอบที่ 12	ถูกต้อง
กรณีการทดสอบที่ 13	ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

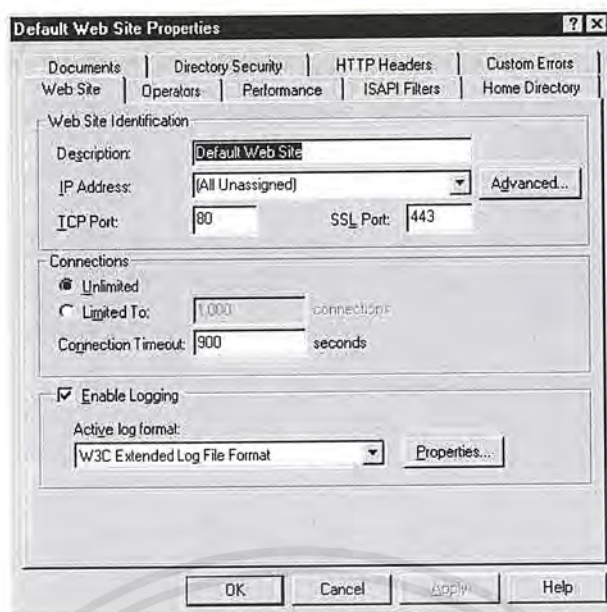
กรณีการทดสอบที่ 14	ถูกต้อง
กรณีการทดสอบที่ 15	ถูกต้อง
กรณีการทดสอบที่ 16	ถูกต้อง
กรณีการทดสอบที่ 17	ถูกต้อง
กรณีการทดสอบที่ 18	ถูกต้อง
กรณีการทดสอบที่ 19	ถูกต้อง
กรณีการทดสอบที่ 20	ถูกต้อง
กรณีการทดสอบที่ 21	ถูกต้อง
กรณีการทดสอบที่ 22	ถูกต้อง
กรณีการทดสอบที่ 23	ถูกต้อง
กรณีการทดสอบที่ 24	ถูกต้อง
กรณีการทดสอบที่ 25	ถูกต้อง
กรณีการทดสอบที่ 26	ถูกต้อง
กรณีการทดสอบที่ 27	ถูกต้อง
กรณีการทดสอบที่ 28	ถูกต้อง
กรณีการทดสอบที่ 29	ถูกต้อง
กรณีการทดสอบที่ 30	ถูกต้อง
กรณีการทดสอบที่ 31	ถูกต้อง
กรณีการทดสอบที่ 32	ถูกต้อง
กรณีการทดสอบที่ 33	ถูกต้อง
กรณีการทดสอบที่ 34	ถูกต้อง
กรณีการทดสอบที่ 35	ถูกต้อง
กรณีการทดสอบที่ 36	ถูกต้อง
กรณีการทดสอบที่ 37	ถูกต้อง
กรณีการทดสอบที่ 38	ถูกต้อง

ตารางที่ 9-1 แสดงผลการทดสอบในส่วนต่าง ๆ ของระบบ

9.3 ปัญหาข้อจำกัดในการทำงานของระบบ

1. การเข้าเว็บเพจจากอินเทอร์เน็ตภายนอกค่อนข้างช้า จนบางครั้งเกิด Time Out จากเว็บเซิร์ฟเวอร์ ซึ่งอาจจะแก้ไขโดยตั้งค่า Time Out ให้มีค่ามากขึ้นดังรูปที่ 9-39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 9-39 แสดงการตั้งค่า Time Out ให้มีค่ามากขึ้น

2. การแปลงจากคำพูดเป็นเสียงบางครั้งไม่สำเร็จโดยตัวไฟล์มีการสร้างแต่ไม่มีข้อมูล ซึ่งขึ้นอยู่กับโครงสร้างภายในของ SAPI ไม่สามารถแก้ไขได้
3. เว็บเพจในบางครั้งการแสดงรูปไม่สำเร็จเนื่องจากการแสดงรูปนั้นต้องการดึงข้อมูลแบบไบนารีมาจากฐานข้อมูล ซึ่งต้องผลการแปลงเป็นไฟล์ *.Tmp ซึ่งทำให้เกิดการผิดพลาดในการแปลงข้อมูลได้ เนื่องจากเป็นปัญหาที่ไม่สามารถคาดการณ์ได้ ในขณะนี้จึงยังไม่สามารถแก้ไขได้

9.4 ข้อกำหนดในการทำงานของระบบ

1. เนื่องจากเซิร์ฟเวอร์ที่ใช้ในการทำงานเป็นเครื่องที่มีสมรรถนะไม่สูงมาก คือ Pentium II 233 MHz หน่วยความจำ 128 เมกะไบต์ ฮาร์ดดิสก์ 2.1 จิกะไบต์ และ 1.7 จิกะไบต์ และถูกใช้เป็นที่ เว็บเซิร์ฟเวอร์ Certificate Server และมี Microsoft Transaction Server เพื่อจุดประสงค์การทำระบบออบเจกต์แบบกระจาย ทำให้การทำงานของระบบโดยรวมค่อนข้างช้าพอสมควร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 10

บทวิจารณ์และสรุป

10.1 สรุปผลการดำเนินงาน

จากหลักการที่ได้ศึกษามาในเรื่องการวิเคราะห์และออกแบบแบบออบเจกต์โอเรียนเท็ดทำให้เราสามารถวิเคราะห์ ออกแบบ และพัฒนา ระบบอย่างมีประสิทธิภาพ นอกจากนี้ยังมีการพัฒนาโดยใช้อ้างอิงเทคโนโลยี COM ยิ่งทำให้เพิ่มความสามารถด้านต่าง ๆ เพิ่มขึ้น เช่น การเพิ่มความสามารถให้กับคอมพิวเตอร์โดยไม่เกิดผลกระทบต่อระบบเดิม ในส่วนของเทคโนโลยี ActiveX จะทำให้สามารถพัฒนาซอฟต์แวร์หรือระบบที่มีความสามารถเพิ่มขึ้นได้ เช่น การใช้งาน ActiveX Document ทำให้การพัฒนาซอฟต์แวร์สามารถย้ายไปใช้งานในแพลตฟอร์มต่าง ๆ ได้

ในส่วนของระบบความปลอดภัยเมื่อศึกษาในรายละเอียดทำให้เราเข้าใจในระบบความปลอดภัยของข้อมูล เช่น ในเรื่องการเข้ารหัสต่าง ๆ เมื่อนำหลักการนี้ไปใช้ในการเลือกติดตั้งระบบแอดออนอื่นที่จำเป็นได้อย่างไม่ผิดพลาด นอกจากนี้ยังทำให้ระบบมีความปลอดภัยในการใช้งานจริงด้วย ส่วนหลักการเรื่องมัลติมีเดียสตรีมมิ่งเมื่อศึกษาแล้วทำให้เราเข้าใจหลักการส่งข้อมูลมัลติมีเดียโดยมีประสิทธิภาพสูงสุดเพื่อประยุกต์เข้ามาใช้ในระบบที่พัฒนาได้และเมื่อมีการพัฒนาระบบขึ้นมาแล้วทำให้ระบบที่ได้ สามารถจัดการเรื่องการส่งข้อมูลมัลติมีเดียได้อย่างรวดเร็ว ในส่วนของ SAPI เมื่อมีการพัฒนาตามที่ได้ศึกษามาแล้วทำให้สามารถนำระบบการเปลี่ยนข้อความเสียงไปประยุกต์ใช้เพื่อการใช้งานในด้านต่าง ๆ ได้ เช่น การพัฒนาระบบอธิบายสินค้าด้วยเสียง

โดยสรุปแล้วระบบที่พัฒนา สามารถนำหลักการและเทคโนโลยีที่ศึกษา มาใช้ทำให้สามารถพัฒนาระบบทั้งหมดได้อย่างมีประสิทธิภาพ และ ประสิทธิภาพ เพื่อสามารถนำไปประยุกต์ใช้ในการพัฒนาต่อไปได้

10.2 แนวทางการพัฒนาต่อ

เนื่องจากระบบที่พัฒนาอ้างอิงการพัฒนาซอฟต์แวร์เชิงคอมพิวเตอร์และการวิเคราะห์และออกแบบออบเจกต์โอเรียนเท็ด ทำให้การพัฒนาต่อสามารถทำได้ง่าย ซึ่งสิ่งที่ทางกลุ่มวิจัยได้มีความปรารถนาจะให้รุ่นน้องพัฒนาต่อคือ

- การพัฒนาระบบแบบกระจายโดยใช้อ้างอิงผ่านแพลตฟอร์มอื่น เช่น ระบบยูนิกซ์ โดยใช้อ้างอิงมาตรฐาน COM เหมือนกัน
- การพัฒนาระบบที่มีการใช้เทคโนโลยี COM+ โดยไม่ต้องใช้ MTS (Microsoft Transaction Server) ซึ่งอาจจำเป็นต้องพัฒนาบน Windows 2000 Server

- การพัฒนาระบบมัลติมีเดียพาณิชย์อิเล็กทรอนิกส์ โดยอาจจะพัฒนาโดยการนำ DirectX เข้ามาใช้ เนื่องจากหลักการที่ DirectX มีการพัฒนาอ้างอิงเทคโนโลยี COM อยู่แล้ว จึงสามารถนำมาใช้งานร่วมกันได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

การเขียนโปรแกรมเชิงออบเจกต์โอเรียนเท็ดใน Visual Basic 6

1. How Visual Basic Provides Polymorphism

Most object-oriented programming systems provide polymorphism through *inheritance*. That is, the hypothetical Flea and Tyrannosaur classes might both inherit from an Animal class. Each class would override the Animal class's Bite method, in order to provide its own bite characteristics.

The polymorphism comes from the fact that you could call the Bite method of an object belonging to any class that derived from Animal, without knowing which class the object belonged to.

Providing Polymorphism with Interfaces

Visual Basic doesn't use inheritance to provide polymorphism. Visual Basic provides polymorphism through multiple ActiveX *interfaces*. In the Component Object Model (COM) that forms the infrastructure of the ActiveX specification, multiple interfaces allow systems of software components to evolve without breaking existing code.

An *interface* is a set of related properties and methods. Much of the ActiveX specification is concerned with implementing standard interfaces to obtain system services or to provide functionality to other programs.

In Visual Basic, you would create an Animal interface and implement it in your Flea and Tyrannosaur classes. You could then invoke the Bite method of either kind of object, without knowing which kind it was.

Polymorphism and Performance

Polymorphism is important for performance reasons. To see this, consider the following function:

```
Public Sub GetFood(ByVal Critter As Object, _
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ByVal Food As Object)
    Dim dblDistance As Double
    ' Code to calculate distance to food (omitted).
    Critter.Move dblDistance ' Late bound
    Critter.Bite Food      ' Late bound
End Sub

```

The Move and Bite methods are *late bound* to Critter. Late binding happens when Visual Basic can't determine at compile time what kind of object a variable will contain. In this example, the Critter argument is declared As Object, so at run time it could contain a reference to any kind of object — like a Car or a Rock.

Because it can't tell what the object will be, Visual Basic compiles some extra code to ask the object if it supports the method you've called. If the object supports the method, this extra code invokes it; if not, the extra code raises an error. Every method or property call incurs this additional overhead.

By contrast, interfaces allow *early binding*. When Visual Basic knows at compile time what interface is being called, it can check the type library to see if that interface supports the method. Visual Basic can then compile in a direct jump to the method, using a virtual function table (vtable). This is many times faster than late binding.

Now suppose the Move and Bite methods belong to an Animal interface, and that all animal classes implement this interface. The Critter argument can now be declared As Animal, and the Move and Bite methods will be early bound:

```

Public Sub GetFood(ByVal Critter As Animal, _
ByVal Food As Object)
    Dim dblDistance As Double
    ' Code to calculate distance to food (omitted).
    Critter.Move dblDistance ' Early bound (vtable).
    Critter.Bite Food      ' Early bound (vtable).
End Sub

```

For More Information "Creating and Implementing an Interface" creates an Animal interface and implements it in Flea and Tyrannosaur classes.

2. Creating and Implementing an Interface

As explained in "How Visual Basic Provides Polymorphism," an interface is a set of properties and methods. In the following code example, you'll create an Animal interface and implement it in two classes, Flea and Tyrannosaur.

You can create the Animal interface by adding a class module to your project, naming it Animal, and inserting the following code:

```
Public Sub Move(ByVal Distance As Double)
```

```
End Sub
```

```
Public Sub Bite(ByVal What As Object)
```

```
End Sub
```

Notice that there's no code in these methods. Animal is an *abstract class*, containing no implementation code. An abstract class isn't meant for creating objects — its purpose is to provide the template for an interface you add to other classes. (Although, as it turns out, sometimes it's useful to implement the interface of a class that isn't abstract; this is discussed later in this topic.)

Note Properly speaking, an abstract class is one from which you can't create objects. You can always create objects from Visual Basic classes, even if they contain no code; thus they are not truly abstract.

Now you can add two more class modules, naming one of them Flea and the other Tyrannosaur. To implement the Animal interface in the Flea class, you use the Implements statement:

Option Explicit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Implements Animal

As soon as you've added this line of code, you can click the left-hand (Object) drop down in the code window. One of the entries will be Animal. When you select it, the right-hand (Procedure) drop down will show the methods of the Animal interface.

Select each method in turn, to create empty procedure templates for all the methods. The templates will have the correct arguments and data types, as defined in the Animal class. Each procedure name will have the prefix Animal_ to identify the interface.

Important An interface is like a contract. By implementing the interface, a class agrees to respond when any property or method of the interface is invoked. Therefore, you must implement *all* the properties and methods of an interface.

You can now add the following code to the Flea class:

```
Private Sub Animal_Move(ByVal Distance As Double)
    ' (Code to jump some number of inches omitted.)
    Debug.Print "Flea moved"
End Sub
```

```
Private Sub Animal_Bite(ByVal What As Object)
    ' (Code to suck blood omitted.)
    Debug.Print "Flea bit a " & TypeName(What)
End Sub
```

You may be wondering why the procedures are declared Private. If they were Public, the procedures Animal_Jump and Animal_Bite would be part of the Flea interface, and we'd be stuck in the same bind we were in originally, declaring the Critter argument As Object so it could contain either a Flea or a Tyrannosaur.

Multiple Interfaces

The Flea class now has two interfaces: The Animal interface you've just implemented, which has two members, and the default Flea interface, which has no members. Later in this example you'll add a member to one of the default interfaces.

You can implement the Animal interface similarly for the Tyrannosaur class:

Option Explicit

Implements Animal

```
Private Sub Animal_Move(ByVal Distance As Double)
```

```
' (Code to pounce some number of yards omitted.)
```

```
Debug.Print "Tyrannosaur moved"
```

```
End Sub
```

```
Private Sub Animal_Bite(ByVal What As Object)
```

```
' (Code to take a pound of flesh omitted.)
```

```
Debug.Print "Tyrannosaur bit a " & TypeName(What)
```

```
End Sub
```

Exercising the Tyrannosaur and the Flea

Add the following code to the Load event of Form1:

```
Private Sub Form_Load()
```

```
Dim fl As Flea
```

```
Dim ty As Tyrannosaur
```

```
Dim anim As Animal
```

```
Set fl = New Flea
```

```
Set ty = New Tyrannosaur
```

```
' First give the Flea a shot.
```

```
Set anim = fl
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Call anim.Bite(ty) 'Flea bites dinosaur.
' Now the Tyrannosaur gets a turn.
Set anim = ty
Call anim.Bite(fl) 'Dinosaur bites flea.
End Sub

```

Press F8 to step through the code. Notice the messages in the Immediate window. When the variable `anim` contains a reference to the Flea, the Flea's implementation of `Bite` is invoked, and likewise for the Tyrannosaur.

The variable `anim` can contain a reference to any object that implements the `Animal` interface. In fact, it can *only* contain references to such objects. If you attempt to assign a `Form` or `PictureBox` object to `anim`, an error will occur.

The `Bite` method is early bound when you call it through `anim`, because Visual Basic knows at compile time that whatever object is assigned to `anim` will have a `Bite` method.

Passing Tyrannosaurs and Fleas to Procedures

Remember the `GetFood` procedure from "How Visual Basic Provides Polymorphism?" You can add the *second* version of the `GetFood` procedure — the one that illustrates polymorphism — to `Form1`, and replace the code in the `Load` event with the following:

```

Private Sub Form_Load()
    Dim fl As Flea
    Dim ty As Tyrannosaur

    Set fl = New Flea
    Set ty = New Tyrannosaur
    'Flea dines on dinosaur,
    Call GetFood(fl, ty)
    ' And vice versa.
    Call GetFood(ty, fl)
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Stepping through this code shows how an object reference that you pass to an argument of another interface type is converted into a reference to the second interface (in this case, `Animal`). What happens is that Visual Basic queries the object to find out whether it supports the second interface. If the object does, it returns a reference to the interface, and Visual Basic places that reference in the argument variable. If the object does not support the second interface, an error occurs.

Implementing Methods That Return Values

Suppose the `Move` method returned a value. After all, you know how far you want an `Animal` to move, but an individual specimen might not be able to move that far. It might be old and decrepit, or there might be a wall in the way. The return value of the `Move` method could be used to tell you how far the `Animal` actually moved.

```
Public Function Move(ByVal Distance As Double) _
    As Double

End Function
```

When you implement this method in the `Tyrannosaur` class, you assign the return value to the procedure name, just as you would for any other Function procedure:

```
Private Function Animal_Move(ByVal Distance _
    As Double) As Double
    Dim dblDistanceMoved As Double
    ' Code to calculate how far to pounce (based on
    ' age, state of health, and obstacles) is omitted.
    ' This example assumes that the result has been
    ' placed in the variable dblDistanceMoved.
    Debug.Print "Tyrannosaur moved"; dblDistanceMoved
    Animal_Move = dblDistanceMoved
End Function
```

To assign the return value, use the full procedure name, including the interface prefix.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

For More Information The interfaces you implement can have properties as well as methods. "Implementing Properties" discusses some differences in the way properties are implemented.

3. Implementing Properties

This topic continues the code example begun in "Creating and Implementing an Interface," adding properties to the Animal interface that was implemented in the Flea and Tyrannosaur classes. You may find it helpful to read that topic before beginning this one.

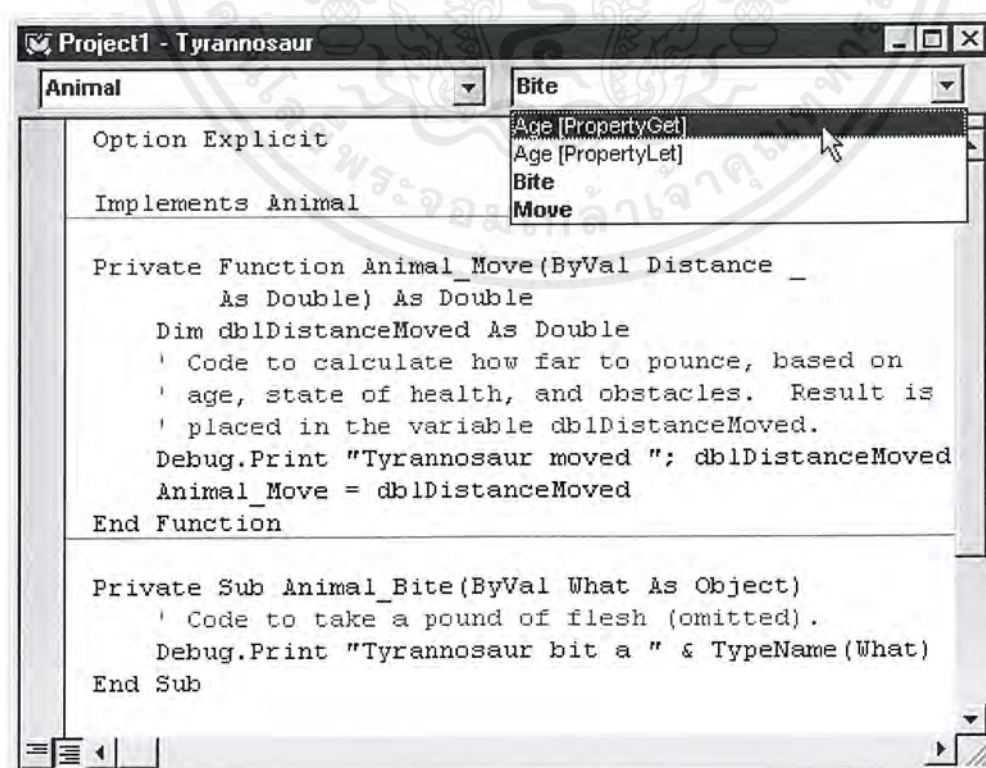
Suppose we give the Animal class an Age property, by adding a Public variable to the Declarations section:

Option Explicit

Public Age As Double

The Procedure drop downs in the code modules for the Tyrannosaur and Flea classes now contain property procedures for implementing the Age property, as shown in Figure 9.10.

Figure 9.10 Implementing property procedures



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This illustrates a point made in "Adding Properties to a Class" earlier in this chapter. Using a public variable to implement a property is strictly a convenience for the programmer. Behind the scenes, Visual Basic implements the property as a pair of property procedures.

You must implement both procedures. The property procedures are easily implemented by storing the value in a private data member, as shown here:

```
Private mdblAge As Double
```

```
Private Property Get Animal_Age() As Double
```

```
    Animal_Age = mdblAge
```

```
End Property
```

```
Private Property Let Animal_Age(ByVal RHS As Double)
```

```
    mdblAge = RHS
```

```
End Property
```

The private data member is an implementation detail, so you have to add it yourself.

Note When Implements provides the template for a Property Set or Property Let, it has no way of determining the name of the last argument, so it substitutes the name RHS, as shown in the code example above.

There's no data validation on a property implemented as a public data member, but that doesn't mean you can't add validation code to the Property Let for Animal_Age. For example, you might want to restrict the values to ages appropriate for a Tyrannosaur or a Flea, respectively.

In fact, this shows the independence of interface and implementation. As long as the interface matches the description in the type library, the implementation can be anything.

Before you go on to the next step, remove the implementation of the read-write Age property from both class modules.

Implementing a Read-Only Property

Of course, allowing the age of an animal to be set arbitrarily is bad object design. The object should know its own age, and provide it to the user as a read-only property. Remove the public variable `Age` from the `Animal` class, and add the template for a read-only age property, like this:

```
Public Property Get Age() As Double
```

```
End Property
```

Now the Procedure drop downs in the code windows for the `Tyrannosaur` and `Flea` classes contain only a single entry, `Age [PropertyGet]`. You might implement this for the `Tyrannosaur` as follows:

```
Private mdblBirth As Double
```

```
Private Property Get Animal_Age() As Double
```

```
    Animal_Age = Now - mdblBirth
```

```
End Property
```

The code above returns the age of the `Tyrannosaur` in days. You could set `mdblBirth` in the `Initialize` event of the `Tyrannosaur` class, as here:

```
Private Sub Class_Initialize()
```

```
    mdblBirth = Now
```

```
End Sub
```

And of course you could return the property value in more commonly used units, such as dog years.

For More Information We've been tossing interfaces and objects around like they were the same thing, seemingly putting references to objects into one object variable, and references to interfaces into another. "Time Out for a Brief Discussion of Objects and Interfaces" clears matters up.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Time Out for a Brief Discussion of Objects and Interfaces

This topic completes the code example begun in "Creating and Implementing an Interface," and continued in "Implementing Properties." You may find it helpful to read those topics before beginning this one.

The Tyrannosaur and Flea code example seems to play fast and loose with interfaces and objects. References to objects are assigned to one object variable, and references to interfaces to another.

In fact, *all of the references are object references*. A reference to an interface is also a reference to the object that implements the interface. Furthermore, an object may have multiple interfaces, but it's still the same object underneath.

In Visual Basic, each class has a default interface that has the same name as the class. Well, almost the same. By convention, an underscore is prefixed to the class name. The underscore indicates that this interface is hidden in the type library.

Thus the Tyrannosaur class has a default interface called `_Tyrannosaur`. Because Tyrannosaur also implements `Animal`, the class has a second interface named `Animal`.

However, underneath it all, the object is still a `Tyrannosaur`. Place a command button on `Form1`, and add the following code:

```
Private Sub Command1_Click()
    Dim ty As Tyrannosaur
    Dim anim As Animal

    Set ty = New Tyrannosaur
    Set anim = ty
    MsgBox TypeName(anim)
End Sub
```

You might expect the message box to display "Animal," but in fact it displays "Tyrannosaur."

Querying for Interfaces

When you assign a Tyrannosaur object to variable of type Animal, Visual Basic asks the Tyrannosaur object if it supports the Animal interface. (The method used for this is called QueryInterface, or QI for short; you may sometimes hear QI used as a verb.) If the answer is no, an error occurs.

If the answer is yes, the object is assigned to the variable. Only the methods and properties of the Animal interface can be accessed through this variable.

Generic Object Variables and Interfaces

What happens if you assign the object reference to a generic object variable, as in the following code?

```
Private Sub Command1_Click()
```

```
    Dim ty As Tyrannosaur
```

```
    Dim anim As Animal
```

```
    Dim obj As Object
```

```
    Set ty = New Tyrannosaur
```

```
    Set anim = ty
```

```
    Set obj = anim
```

```
    MsgBox TypeName(obj)
```

```
End Sub
```

The result is again Tyrannosaur. Now, what interface do you get when you call properties and methods through the variable obj? Add the following method to the Tyrannosaur class:

```
Public Sub Growl()
```

```
    Debug.Print "Rrrrrr"
```

```
End Sub
```

The Growl method belongs to the Tyrannosaur object's default interface. In the code for the command button's Click event, replace the MsgBox statement with the following two lines of code:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
obj.Move 42
```

```
obj.Growl
```

When you run the project and click the button, execution stops on the Growl method, with the error "Object does not support this property or method." Clearly, the interface is still Animal.

This is something to bear in mind when using variables of type Object with objects that have multiple interfaces. The interface the variable will access is the *last interface assigned*. For example:

```
Private Sub Command1_Click()
```

```
    Dim ty As Tyrannosaur
```

```
    Dim anim As Animal
```

```
    Dim obj As Object
```

```
    Set ty = New Tyrannosaur
```

```
    Set anim = ty
```

```
    Set obj = anim
```

```
    obj.Move 42 ' Succeeds
```

```
    obj.Growl ' Fails
```

```
    Set obj = ty
```

```
    obj.Move 42 ' Fails
```

```
    obj.Growl ' Succeeds
```

```
End Sub
```

Fortunately, there's very little reason to use the slower, late-bound Object data type with objects that have multiple interfaces. One of the main reasons for using multiple interfaces is to gain the advantage of early binding through polymorphism.

Other Sources of Interfaces

Visual Basic class modules are not your only source of interfaces to implement. You can implement any interface contained in a type library, as long as that interface supports Automation.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

If you have the Professional or Enterprise Edition of Visual Basic, you can create your own type libraries of abstract classes. These type libraries can be used in many projects, as described in "General Principles of Component Design" in *Creating ActiveX Components* in the *Component Tools Guide*.

The Professional and Enterprise editions also include the MkTypLib (Make Type Library) utility in the Tools directory. If you've used this utility with Microsoft Visual C++, you may find it a more congenial way to create interfaces.

Using Interfaces in Your Project

To use an interface in your project, click References on the Project menu to open the References dialog box. If the type library is registered, it will appear in the list of references, and you can check it. If the type library is not in the list, you can use the Browse button to locate it.

Once you have a reference to a type library, you can use Implements to implement any Automation interfaces the type library contains.

For More Information You're not limited to implementing abstract interfaces. "The Many (Inter)Faces of Code Reuse" describes how you can implement an interface and selectively reuse the properties and methods of the class that provides the interface.

5. The Many (Inter)Faces of Code Reuse

There are two main forms of code reuse — binary and source. Binary code reuse is accomplished by creating and using an object, while source code reuse is achieved by inheritance, which isn't supported by Visual Basic. (Source code reuse can also be achieved by copying and modifying the source code, but this technique is nothing new, and has many well-known problems.)

Visual Basic has been a pioneer of binary code reuse — controls being the classic example. You reuse the code in a control by placing an instance of the control on your form. This is known as a *containment* relationship or a *has-a* relationship; that is, the form *contains* or *has a* CommandButton.

For More Information Containment relationships are discussed in "Object Models" later in this chapter.

Delegating to an Implemented Object

Implements provides a powerful new means of code reuse. You can implement an abstract class (as discussed in "Creating and Implementing an Interface"), or you can implement the interface of a fully functional class. You can create the *inner object* (that is, the implemented object) in the Initialize event of the *outer object* (that is, the one that implements the inner object's interface).

As noted in "Creating and Implementing an Interface," an interface is like a contract — you must implement all the members of the inner object's interface in the outer object's class module.

However, you can be very selective in the way you delegate to the properties and methods of the inner object. In one method you might delegate directly to the inner object, passing the arguments unchanged, while in another method you might execute some code of your own before calling the inner object — and in a third method you might execute only your own code, ignoring the inner object altogether!

For example, suppose you have a OneManBand class and a Cacophony class, both of which generate sounds. You'd like to add the functionality of the Cacophony class to the OneManBand class, and reuse some of the implementation of the Cacophony class's methods.

' OneManBand implements the Cacophony interface.

Implements Cacophony

' Object variable to keep the reference in.

Private mcac As Cacophony

Private Sub Class_Initialize()

 ' Create the object.

 Set mcac = New Cacophony

End Sub

You can now go to the Object drop down and select Cacophony, and then get procedure templates for the methods of the Cacophony interface. To implement these methods, you can delegate to the Cacophony object. For example, the Beep method might look like this:

```
Private Sub Cacophony_Beep(ByVal Frequency As Double, _
ByVal Duration As Double)
    ' Delegate to the inner Cacophony object.
    Call mCAC.Beep(Frequency, Duration)
End Sub
```

The implementation above is very simple. The outer object (OneManBand) delegates directly to the inner (Cacophony), reusing the Cacophony object's Beep method without any changes. This is a good thing, but it's only the beginning.

The Implements statement is a very powerful tool for code reuse, because it gives you enormous flexibility. You might decide to alter the effects of the OneManBand class's Beep method, by inserting your own code before (or after) the call to the inner Cacophony object:

```
Private Sub Cacophony_Beep(ByVal Frequency As Double, _
ByVal Duration As Double)
    ' Bump everything up an octave.
    Frequency = Frequency * 2
    ' Based on another property of the OneManBand
    ' class, Staccato, cut the duration of each beep.
    If Staccato Then Duration = Duration * 7 / 8
    Call mCAC.Beep(Frequency, Duration)
    ' You can even call other methods of OneManBand.
    If Staccato Then Pause(Duration * 1 / 8)
End Sub
```

For some of the methods, your implementation may delegate directly to the inner Cacophony object, while for others you may interpose your own code before and after delegating — or even omit delegation altogether, using entirely your own code to implement a method.

Because the `OneManBand` class implements the `Cacophony` interface, you can use it with any musical application that calls that interface. Your implementation details are hidden from the calling application, but the resulting sounds are all your own.

Note COM provides another mechanism for binary code reuse, called *aggregation*. In aggregation, an entire interface is reused, without any changes, and the implementation is provided by an instance of the class being aggregated. Visual Basic does not support this form of code reuse.

Doesn't This Get Tedious?

Writing delegation code can indeed become tedious, especially if most of the outer object's properties and methods simply delegate directly to the corresponding properties and methods of the inner object.

If you have the Professional or Enterprise Edition of Visual Basic, you can use the Visual Basic Extensibility model to create your own delegation wizard to automate the task, similar to the Class Wizard that's included in the Professional and Enterprise editions.

For More Information The use of polymorphism and multiple interfaces in component software is discussed in "General Principles of Component Design" in *Creating ActiveX Components* in the *Component Tools Guide*.

Using the Extensibility Model is documented in *Extending the Visual Basic Environment with Add-Ins* in the *Component Tools Guide*.

ภาคผนวก ข

เอกสารอ้างอิง SAPI

(Sound Application Programming)

Direct Speech Synthesis Control

Properties

The Direct Speech Synthesis control provides the following properties:

- Age
- CallbacksEnabled
- CountEngines
- CurrentMode
- Dialect
- EngineFeatures
- EngineID
- Features
- FileName
- Find
- FindEngine
- Gender
- hWnd
- Initialized
- Interfaces
- JawOpen
- LanguageID
- LastError
- LastWordPosition
- LipTension
- LipType
- MaxPitch

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- MaxRealTime
- MaxSpeed
- MaxVolumeLeft
- MaxVolumeRight
- MfgName
- MinPitch
- MinRealTime
- MinSpeed
- MinVolumeLeft
- MinVolumeRight
- ModeID
- ModeName
- MouthEnabled
- MouthHeight
- MouthUpturn
- MouthWidth
- Phonemes
- Pitch
- ProductName
- RealTime
- Sayit
- Speaker
- Speaking
- Speed
- Style
- SuppressExceptions
- Tagged
- TeethLowerVisible
- TeethUpperVisible
- TonguePosn
- VolumeLeft
- VolumeRight

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Age

Age(index As Long) As Long

Age of the voice. Read only. This property can be one of the values shown below.

Parameter	Description
<i>index</i>	Index for the selected engine.

Age Value	Meaning
<i>TTSAGE_BABY</i>	About 1 year old
<i>TTSAGE_TODDLER</i>	About 3 years old
<i>TTSAGE_CHILD</i>	About 6 years old
<i>TTSAGE_ADOLESCENT</i>	About 14 years old
<i>TTSAGE_ADULT</i>	Between 20 and 60 years old
<i>TTSAGE_ELDERLY</i>	Over 60 years old

CallBacksEnabled

CallBacksEnabled As Short

When CallBacksEnabled is set to 0, events are not fired.

CountEngines

CountEngines As Long

Number of speech synthesis voices installed on this computer. Read only. This is the highest number that can be used as an index to indexed properties and methods.

CurrentMode

CurrentMode As Long

This is the index for the currently selected voice, as passed into the various engine information calls and properties.

Dialect

Dialect(index As Long) As String

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dialect specific to the language -- for example, "Texas" or "New York City." Read only. This property can be "Standard" (or its localized equivalent) for the standard dialect used by the television and radio news media.

Parameter	Description
<i>index</i>	Index for the selected engine.

EngineFeatures

EngineFeatures(*index* As Long) As Long

Features specific to a particular engine. Read only. These values are defined by the engine author.

Parameter	Description
<i>index</i>	Index for the selected engine.

EngineID

EngineID(*index* As Long) As String

Globally unique identifier (GUID) that uniquely identifies the engine. Read only.

The GUID can be generated randomly using GUIDGEN.EXE, which is provided with the OLE Software Development Kit (SDK). It is possible for more than one mode to have the same engine identifier. In this case, the engine is capable of speaking the different languages or voices of the modes. Read only.

Parameter	Description
<i>index</i>	Index for the selected engine.

Features

Features(*index* As Long) As Long

Text-to-speech features that are available. Read only. The **Features** property can be a combination of the values shown below.

Parameter	Description
<i>index</i>	Index for the selected engine.

Features

Meaning

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Value

TTSFEATUR The engine can attempt to read any word.

E_ANYWOR

D

TTSFEATUR The audio destination for the device is fixed, so an audio destination object is not

E_FIXEDAU needed for select. If an audio destination is passed in, it will be ignored. Hardware

DIO devices, such as an external synthesizer box, will support this.

TTSFEATUR The engine's **Phonemes** method supports the CHARSET_IPAPHONETIC voice

E_IPAUNICO character set and will return phonemic representations using the International Phonetic

DE Alphabet (IPA) in Unicode.

TTSFEATUR The voice is optimized to work over the computer, which provides better sound quality.

E_PCOPTIMI

ZED

TTSFEATUR The voice is designed to work over the telephone and uses an 8-kilohertz (kHz)

E_PHONEOP sampling rate.

TIMIZED

TTSFEATUR The engine can adjust the pitch of the voice while it is speaking.

E_PITCH

TTSFEATUR The engine is single instance, and so only one mode object can exist at one time. TTS

E_SINGLEIN engines that are .EXE servers use this.

STANCE

TTSFEATUR The engine can adjust the speed of the voice while it is speaking.

E_SPEED

TTSFEATUR The engine can interpret tagged text to control the voice.

E_TAGGED

TTSFEATUR The engine is completely thread safe, and can be called from any thread.

E_THREADS

AFE

TTSFEATUR Supports transplanted prosody.

E_TRANSPL

ANTED

TTSFEATUR The engine can provide mouth position information for the voice.

E_VISUAL

TTSFEATUR The engine can adjust the volume of the voice.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

E_VOLUME

TTSFEATUR The engine sends notification of word position.

E_WORDPO

SITION

FileName

FileName As String

When this variable is assigned to a filename, subsequent text-to-speech is recorded in a file of this name instead of played to the wave device. To re-enable the speakers and disable recording to a file, set FileName to "".

Find

Find(RankList As String) As Long

Returns the index of an engine which most closely matches the input parameter list, ranked in order. Read only.

The syntax for RankList is:

<field>=<value>;<field>=<value>...

Where field can be:

- EngineId
- MfgName
- ProductName
- ModeID
- ModeName
- LanguageID
- Dialect
- Speaker
- Style

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Gender
- Age
- Features
- Interfaces
- EngineFeatures

For example, Find("MfgName=Microsoft;Gender=1") finds a female Microsoft voice.

FindEngine

FindEngine(EngineID As String , MfgName As String, ProductName As String, ModeID As String, ModeName As String, LanguageID As Long, dialect As String, Speaker As String, Style As String, Gender As Long, Age As Long, Features As Long, Interfaces As Long, EngineFeatures As Long, RankEngineID As Long, RankMfgName As Long, RankProductName As Long, RankModeID As Long, RankModeName As Long, RankLanguage As Long , RankDialect As Long, RankSpeaker As Long, RankStyle As Long, RankGender As Long, RankAge As Long, RankFeatures As Long, RankInterfaces As Long , RankEngineFeatures As Long) As Long

NOTE: use **Find** instead

Returns the index of an engine which most closely matches the input parameters and rankings.
Read only.

To ignore a field, set it to zero. Rank numbers are integers, 1 means lowest priority, 2 means next and so on.

Gender

Gender(index As Long) As Long

Gender of the voice. Read only. The **Gender** property can be one of the values shown below. 1 means female, 2 means male.

Parameter	Description
<i>index</i>	Index for the selected engine.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Gender Value	Meaning
<i>GENDER_FEMALE</i>	Female
<i>GENDER_MALE</i>	Male
<i>GENDER_NEUTRAL</i>	Not perceptibly male or female

hWnd

hWnd As Long

Window handle. Read only.

Initialized

Initialized As Short

Equals 1 if the control has been initialized, 0 if not. The user can force the control to initialize by setting this to 1 (the speech engines can be slow to load, so some users may want fine control over when the control initializes).

Use of most methods and properties will automatically cause the control to initialize, if it hasn't been initialized already.

Interfaces

Interfaces(index As Long) As Long

Interfaces supported by the engine. Read only. The **Interfaces** property can be a combination of the values shown below.

Parameter	Description
<i>index</i>	Index for the selected engine.

Interfaces Value	Meaning
<i>TTSI_ILEXPRONOUNCE</i>	ILexPronounce interface
<i>TTSI_ITTSATTRIBUTES</i>	ITTSAttributes interface
<i>TTSI_ITTSCENTRAL</i>	ITTSCentral interface
<i>TTSI_ITTSDIALOGS</i>	ITTSDialogs interface

JawOpen

JawOpen As Short

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Angle to which the jaw is open. This is a linear range from 0xFF for completely open to 0x00 for completely closed.

LanguageID

LanguageID(index As Long) As Long

Language identifier as specified for the Win32 API. Read only.

Bits 0 through 9 identify the primary language, such as English, French, Spanish, and so on.

Bits 10 through 15 indicate the sublanguage, which is essentially a locale setting. For example,

Portuguese has two sublanguages, Brazilian and Standard. For a list of primary and

sublanguage identifiers, see the documentation for the MAKELANGID macro in the Win32

SDK. Language ID codes are limited to the Windows 32 definition of LANGID. Dialect strings

can be anything.

Parameter	Description
<i>index</i>	Index for the selected engine.
LastError	
LastError As Long	
	Result code from the last method or property invocation. Read only.
LastWordPosition	
LastWordPosition As Long	
	Offset, in bytes, from the beginning of the text-to-speech buffer to the word that is currently being played.

LipTension

LipTension As Short

Lip tension. This is a linear range from 0xFF if the lips are very tense to 0x00 if they are completely relaxed.

LipType

LipType As Short

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

When set to 0, the red female lips are drawn. When set to 1, the male pink lips are drawn.

MaxPitch

MaxPitch As Long

Maximum legal value for **Pitch**. Read only.

MaxRealTime

MaxRealTime As Long

Maximum legal value for **RealTime**. Read only.

MaxSpeed

MaxSpeed As Long

Maximum legal value for **Speed**. Read only.

MaxVolumeLeft

MaxVolumeLeft As Long

Maximum legal value for **VolumeLeft**. Read only.

MaxVolumeRight

MaxVolumeRight As Long

Maximum legal value for **VolumeRight**. Read only.

MfgName

MfgName(index As Long) As String

Name of the manufacturer -- for example, "Acme Corporation." Read only.

Parameter	Description
<i>index</i>	Index for the selected engine.

MinPitch

MinPitch As Long

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Minimum legal value for **Pitch**. Read only.

MinRealTime

MinRealTime As Long

Minimum legal value for **RealTime**. Read only.

MinSpeed

MinSpeed As Long

Minimum legal value for **Speed**. Read only.

MinVolumeLeft

MinVolumeLeft As Long

Minimum legal value for **VolumeLeft**. Read only.

MinVolumeRight

MinVolumeRight As Long

Minimum legal value for **VolumeRight**. Read only.

ModeID

ModeID(index As Long) As String

GUID that uniquely identifies the mode. Read only.

These numbers are guaranteed to be unique, so it is not necessary to monitor and record them.

In addition, the GUID is not stored or used directly by OLE, so it is not necessary to load it into the registry.

Parameter	Description
<i>index</i>	Index for the selected engine.

ModeName

ModeName(index As Long) As String

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Name of the text-to-speech mode -- for example, "English Talk-O-Matic" or "Cartoon Voice."

Read only.

Parameter	Description
<i>index</i>	Index for the selected engine.

MouthEnabled

MouthEnabled As Short

When MouthEnabled is set to 0, the mouth does not animate. To make the mouth not get drawn at all, set the (container's) visible property to False. Also, setting the control (container) size to zero has a similar effect.

MouthHeight

MouthHeight As Short

Height of the mouth or lips. This is a linear range from 0xFF for the maximum possible height for the mouth to 0x00 for the minimum height (that is, the mouth or lips are closed).

MouthUpturn

MouthUpturn As Short

Extent to which the mouth turns up at the corners (that is, how much it smiles). This is a linear range from 0xFF for the maximum upturn (that is, the mouth is fully smiling) to 0x00 if the corners of the mouth turn down. If this member is 0x80, the mouth is neutral.

MouthWidth

MouthWidth As Short

Width of the mouth or lips. This is a linear range from 0xFF for the maximum possible width for the face to 0x00 for the minimum width (that is, the mouth or lips are puckered).

Phonemes

Phonemes(charset As Long, flags As Long, input As String) As String

The phonemic representation of the text using the specified character set. Read only. Converts text to a phonemic representation, which is an intermediate stage between Unicode text and digital-audio data.

Parameter

Description

chars [in] Character set to use to convert the text. The *charset* parameter must be one of the values shown below, although an engine may support both.

flags [in] TTSDATAFLAG_TAGGED if the text contains text-to-speech control tags, or NULL if the text does not contain tags. Note that an engine-specific phonetic character set may use a different tagging protocol than that used by the IPA character set. This is left for future expansion.

input [in] the text string to be converted. If the engine supports the IPA character set, tags can be used to intersperse words written in IPA phonemes with words in the language's character set.

<i>charset</i> Value	Meaning
<i>CHARSET_ENGINEPHONETIC</i>	An engine-specific phonetic character set
<i>CHARSET_IPAPHONETIC</i>	The International Phonetic Alphabet (IPA) character set
<i>CHARSET_TEXT</i>	cannot be used as the character set for converted text.

This function can be used to adjust the phonetic representation of text before it is spoken by the engine, such as correcting mispronounced words.

Phonemes is synchronous; that is, the property does not return until the conversion is complete. Depending on the amount of text passed, this process can take some time to complete, so an application should use **Phoneme** with a background thread.

Pitch

Pitch As Long

The current baseline pitch, in hertz, for a text-to-speech mode.

If the text being spoken contains text-to-speech control tags, the actual pitch of the speaking voice typically fluctuates above this baseline. The **Pit** tag changes the baseline pitch setting for

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

the mode, the same as when the **Pitch** property is set. The default baseline pitch is the optimal pitch for the mode.

If the value is out of range for the engine, an error is returned and the pitch is not changed. When assigned to this property, the new value must be between **MaxPitch** and **MinPitch**, otherwise an exception occurs.

The actual pitch of the voice typically fluctuates above this baseline. It usually does not go below it.

ProductName

ProductName(index As Long) As String

Name of the product -- for example, "Acme's Talk-O-Matic." Read only.

Parameter	Description
<i>index</i>	Index for the selected engine.

RealTime

RealTime As Long

The real-time setting for the engine, which is the percentage of processor time that the engine expects to use during constant speech. This value can be over 100 for non-real-time applications.

For most engines, the amount of processor time required diminishes markedly during periods of silence. For example, if **RealTime** is 100, the engine takes 1 full minute of processor time to process one minute of speech. If **RealTime** is 50, the engine takes 30 seconds of processor time to process the same minute of speech. This value is difficult to compute precisely, so it should be regarded as an estimate.

An application can specify **MinRealTime** or **MaxRealTime** for minimum or maximum allowable values. If the value is out of range for the engine, the member function returns an error, and the real-time setting is not changed.

The engine tries to meet the real-time expectation, but the engine will not be able to consume exactly **RealTime** percentage of the CPU.

Sayit

Sayit As String

Alternative method for starting speaking. The text is spoken when Sayit is assigned to a string. This is present to support certain limited OLE containers that do not support methods, but do support properties. Generally, do not use this unless the OLE container does not support methods.

Speaker

Speaker(index As Long) As String

Name of the voice, or NULL if the name is not important. Read only.

Parameter	Description
<i>index</i>	Index for the selected engine.

Speaking

Speaking As Short

When set to 1, the synthesizer voice is speaking. When set to 0, the synthesizer voice is not speaking.

Speed

Speed As Long

Retrieves the current baseline average speed for a text-to-speech mode, in words per minute. When assigned to this property the new value must be between **MaxSpeed** and **MinSpeed**, otherwise an exception occurs. If the value is out of range for the engine, an error is returned and the speed is not changed.

If the text being played contains text-to-speech control tags, the actual speed of the speaking voice fluctuates above and below this baseline. The **Spd** tag changes the baseline speed setting for the mode, the same as when the **Speed** property is set. The default baseline speed is the optimal speed for the mode.

Style

Style(index As Long) As String

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Personality of the voice -- for example:

- "Business"
- "Casual"
- "Computer"
- "Excited"
- "Singsong"

Read only.

Parameter	Description
<i>index</i>	Index for the selected engine.
SuppressExceptions	
SuppressExceptions As Short	
	When set to 1, exceptions will never occur. You must check LastError to get the error code of the last method or property invocation.
Tagged	
Tagged As Boolean	
	Determines whether or not tags are interpreted when calls to Speak are made.
TeethLowerVisible	
TeethLowerVisible As Short	
	Extent to which the lower teeth are visible. This is a linear range from 0xFF for the maximum extent (that is, the lower teeth and gums are completely exposed) to 0x00 for the minimum (the lower teeth are completely hidden.) If this member is 0x80, only the teeth are visible.
TeethUpperVisible	
TeethUpperVisible As Short	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Extent to which the upper teeth are visible. This is a linear range from 0xFF for the maximum extent (that is, the upper teeth and gums are completely exposed) to 0x00 for the minimum (the upper teeth are completely hidden). If this member is 0x80, only the teeth are visible.

TonguePosn

TonguePosn As Short

Tongue position. This a linear range from 0xFF if the tongue is against the upper teeth, to 0x00 if it is relaxed. If this member is 0x80, the tongue is visible.

VolumeLeft

VolumeLeft As Long

The current baseline speaking volume for the left channel of text-to-speech mode. When assigned to this property, the new value must be between **MaxVolumeLeft** and **MinVolumeLeft**, otherwise an exception occurs and the speaking volume is not changed..

If the text being played contains text-to-speech control tags, the actual volume of the speaking voice fluctuates above and below this baseline. The **Vol** tag changes the baseline volume setting for the mode, the same as when the **VolumeLeft** property is set.

VolumeRight

VolumeRight As Long

The current baseline speaking volume for the right channel of text-to-speech mode. When assigned to this property, the new value must be between **MaxVolumeRight** and **MinVolumeRight**, otherwise an exception occurs and the speaking volume is not changed..

If the text being played contains text-to-speech control tags, the actual volume of the speaking voice fluctuates above and below this baseline. The **Vol** tag changes the baseline volume setting for the mode, the same as when the **VolumeRight** property is set.

© 1995-1998 Microsoft Corporation. All rights reserved.

ภาคผนวก ก

การติดตั้ง Microsoft Certificate server ใช้งานร่วมกับ IIS

How to Configure Certificate Server for Use with SSL on IIS

The information in this article applies to:

- Microsoft Internet Information Server version 4.0

SUMMARY

You can use Certificate Server to issue certificates for use with the Secure Sockets Layer (SSL). This is typically done on a local intranet, where you have the ability to directly inform your clients that they can trust your certificates.

MORE INFORMATION

IIS supports the SSL 3.0 protocol, which uses certificates to identify both the client and server during communication, and to establish a one-time session key to encrypt and decrypt data transmitted during that particular communication session.

You can use Certificate Server 1.0, which is a component of the Windows NT Option Pack, to issue certificates for your clients to use.

Before SSL can be used, the following tasks must be performed on the server:

1. Create a Root CA Certificate on the server.
2. Install the Root CA Certificate on the server.
3. Create a Key Certificate Request for the server.
4. Process the Key Certificate Request for the server.
5. Install the Key Certificate on the server.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Secure the directory on the server.

Next, perform the following tasks on the client:

1. Install the Root CA Certificate on the client.
2. Install a Certificate on the client.
3. Connect to the SSL-Secured directory from the client.

NOTE: Each of the tasks listed above correspond to a section below. Go to that section for details on how to perform that particular task.

Creating a Root CA Certificate on the Server

To create a root CA certificate on the server, simply perform the default installation of the Certificate Server component of the Windows NT Option Pack. The default installation automatically creates a root CA certificate.

NOTE: If you choose to use Advanced Configuration, do **NOT** select the **Non-root CA** option.

Installing the Root CA Certificate on the Server

1. Browse to <http://localhost/certsrv/>, click the **Certificate Enrollment Tools** link, and then click the **Install Certificate Authority Certificates** link.
2. Click the **Refresh** button to verify that the information displayed is current, and then click the **Certificate for ComputerName\CA-Name** link.
3. In the **File Download** dialog box, select the **Open this file from its current location** radio button, and then click **OK**.

Perform the following steps if Windows NT 4.0, SP4 or SP5 is installed

- a. In the Certificate dialog box, click the **Install Certificate** button.
- b. When the **Certificate Manager Import Wizard** starts, click **Next**.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- c. When prompted to select a certificate store, select the **Place all certificates into the following store** radio button, and then click **Browse**.
- d. Select the **Show Physical Stores** option, open **Trusted Root Certificate Authorities**, and then click **Local Computer**. Click **OK**.
- e. Click **Next**, and then click **Finish**. Click **OK** to close the dialog box.
- f. Restart the server to cause the root CA certificate to take effect.

For additional information, please see the following article(s) in the Microsoft Knowledge Base:

[Q194788](#) Windows NT Service Pack 4 and Client Certificates

Perform the following steps if Windows NT 4.0, SP3 is installed

- a. In the **New Site Certificate** dialog box, click **OK** (you will typically want to leave all of the check boxes selected).
- b. When prompted by **Do you want to ADD the following certificate to the Root Store?**, click **Yes**.
- c. At a command prompt, use the **CD** command to change directories to the `%SystemRoot%\System32\InetSrv` directory (for example, type `cd \winnt\system32\inetsrv` if your system root is `\winnt`).
- d. Type `iisca`, to synchronize the root CA certificate stores used by IIS and Internet Explorer.
- e. Force the registry to be re-read, so that the new root CA certificate is recognized. This is done by either restarting the server, or stopping the IISADMIN service and its dependent services (for example WWW, FTP, NNTP, SMTP, and so on) and then restarting the dependent services that you use. These services can be stopped and restarted by doing either of the following:
 - Open Control Panel, open Services, and then stop and restart the services.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-OR-

- Run NET STOP and NET START commands at a command prompt. To do this, perform the following:

1. At a command prompt, type **net stop iisadmin /y** to stop the IISADMIN service and its dependent services.
2. Restart the dependent services you use. For example, to restart the WWW service, type **net start w3svc**. To restart FTP, type **net start msftpsvc**.

Creating a Key Certificate Request for the Server

1. Start the Internet Service Manager (ISM), which loads the Internet Information Server snap-in for the Microsoft Management Console (MMC).
2. Right-click the Web site, directory, or file to be secured, and then click **Properties**. Click the **Directory Security** (or **File Security**) tab.
3. Under Secure Communications, click the **Key Manager** button.

NOTE: This button will be labeled "Edit" instead of "Key Manager" if a certificate has already been installed.

4. In Key Manager, right-click WWW, and then click **Create New Key**.
5. Click the **Put the request in a file that you will send to an authority** radio button, and then save the file to your hard disk. Be sure to remember the name and location of the file.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NOTE: C:\NewKeyRq.txt is the default path and name for this file.

6. Step through the rest of the Create New Key dialog boxes.

NOTE: When prompted for your state, be sure to spell it out completely (do not use the abbreviation), with proper capitalization, so that the certificate request will be PKCS #10 compatible.

7. Close the Key Manager, being sure to click **Yes** when prompted to **Commit all changes now?**
8. In the MMC, click **OK**.

Processing the Key Certificate Request for the Server

1. Open the text file created for the server request (C:\NewKeyRq.txt by default).
2. Select and copy the text for the key, beginning with the line:

—BEGIN NEW CERTIFICATE REQUEST—

and ending with:

—END NEW CERTIFICATE REQUEST—

(in other words, include both of these lines).

3. Browse to <http://localhost/certsrv/>, click the **Certificate Enrollment Tools** link, and then click the **Process a Certificate Request** link.
4. On the Web Server Enrollment page, paste the text from the key into the text box, and then click **Submit Request**.
5. When the certificate has been successfully processed, click the **Download** button.
6. Click the **Save this file to disk** radio button, and then save the file. Be sure to remember the name and location of the file.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NOTE: Newcert.cer is the default name for this file.

Installing the Key Certificate on the Server

1. In the MMC, right-click the Web site, directory, or file to be secured, and then click **Properties**. Click the **Directory Security** (or **File Security**) tab.
2. Under Secure Communications, click the **Edit** button (note that this changed from previously being labeled Key Manager). Now click the **Key Manager** button.
3. In Key Manager, right-click the new key request (the icon with a red slash through it), and then click **Install Key Certificate**.
4. Select the certificate file, and then when prompted, provide the password. Click **OK**.
5. In the Server Bindings dialog box, **Any Unassigned** should be displayed under both the **IP Address** and **Port Number** columns. Click **OK** (unless you want to assign the key to particular IP address and port number).
6. Close Key Manager and make sure to click **Yes** when prompted to **Commit all changes now?**
7. Click **OK** twice to return to the MMC.

Securing the Directory on the Server

1. In the MMC, right-click the the Web site, directory, or file to be secured, and then click **Properties**.
2. Click the **Directory Security** (or **File Security**) tab. Under Secure Communications, click the **Edit** button.
3. Select the **Require Secure Channel when accessing this resource** check box.
4. Select the **Require Client Certificates** radio button.
5. Click **OK** twice to return to the MMC.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Installing the Root CA Certificate on the Client

1. Browse to `http://ServerDomainName/certsrv/`, click the **Certificate Enrollment Tools** link, and then click the **Install Certificate Authority Certificates** link.
2. Click the **Refresh** button to verify that the information displayed is current, and then click the **Certificate for *ServerDomainName*\CA-Name** link.
3. In the **File Download** dialog box, select the **Open this file from its current location** radio button, and then click **OK**.
4. The dialog box displayed next will depend on which Service Pack has been applied to Windows NT 4.0.

If SP4 or SP5 Is Installed

1. In the **Certificate** dialog box, click the **Install Certificate** button.
2. When the **Certificate Manager Import Wizard** starts, click **Next**.
3. When prompted to select a certificate store, select the **Place all certificates into the following store** radio button, and then click **Browse**.
4. Select the **Show Physical Stores** checkbox, open **Trusted Root Certificate Authorities**, and then select **Local Computer**. Click **OK**.
5. Click **Next**, and then click **Finish**. Click **OK** to close the dialog box.
6. Restart the computer.

If SP3 Is Installed

1. In the **New Site Certificate** dialog box, click **OK** (you will typically want to leave all of the check boxes selected).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. When prompted by **Do you want to ADD the following certificate to the Root Store?**, click **Yes**.
3. Restart the client computer, so that the new root CA certificate will take effect.

Installing a Certificate on the Client

1. Browse to <http://ServerDomainName/certsrv/>, click the **Certificate Enrollment Tools** link, and then click the **Request a Client Authentication Certificate** link.

NOTE: In Internet Explorer, security must be set to Medium in order to download the ActiveX control on this Web page. (Netscape does not use the ActiveX control, so the security setting is not an issue for it).

2. Fill in the information requested in **Certificate Enrollment Form** the page, and then click the **Submit Request** button.
3. When the certificate has been successfully processed, click the **Download** button.
4. Click **OK** when you see the message **Your new certificate has been successfully installed!**

Connecting to the SSL-Secured Directory from the Client

1. Browse to <https://ServerDomainName/SecuredResource>

NOTE: Be sure to use the **https** protocol, not just **http**, so that the server will create a secure connection.

2. When the **Client Authentication** dialog box appears, select the certificate you just installed (in the section above), and then click **OK**.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

ปัญหาที่พบและวิธีแก้

ปัญหาที่พบใน Microsoft Certificate Server

Certificate Server Fails to Start with Event ID 7000 or 7009

The information in this article applies to:

- Microsoft Certificate Server version 1.0

SYMPTOMS

After you install the Windows NT Option pack (NTOP), the Certificate Authority service fails to start and the following Event IDs occur in the event log:

Event ID: 7000 Source: Service Control Manager Description: The Certificate Authority service failed to start due to the following error: The service did not respond to the start or control request in a timely fashion.

Event ID: 7009 Source: Service Control Manager Description: Timeout (120000 milliseconds) waiting for service to connect.

CAUSE

This is caused by a User DSN being created in the 32-Bit ODBC applet in Control Panel for the Certificate Server's Access Database. This DSN needs to be created as a System DSN.

RESOLUTION

To work around this problem, run the following registry script to create the "CertSrv" ODBC data source.

Please note that the path specified in the "Driver" and "DBQ" entries must be adjusted according to the actual configuration.

Copy and paste the following code into a new Text document using Notepad, and then save the document as "certsrv_odbcupdate.reg" (without the quotation marks).

REGEDIT4

[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\odbc.ini\CertSrv]

"Driver"="C:\WINNT\System32\odbcjt32.dll"

"DBQ"="C:\WINNT\System32\CertLog\certsrv.mdb"

"DriverId"=dword:00000019

"SafeTransactions"=dword:00000000

"UID"="Admin"

[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\odbc.ini\CertSrv\Engines\Jet]

"ImplicitCommitSync"="Yes"

"Threads"=dword:00000003

"UserCommitSync"="Yes"

[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\odbc.ini\ODBC Data Sources]

"CertSrv"="Microsoft Access Driver (*.mdb)"

Double-click the new **Certsrv_odbcupdate.reg** file to merge these settings into the Registry. After you double-click the file and it opens in Notepad, make sure that the file is saved with the .reg extension and not a .txt extension.

If simply creating the System DSN does not resolve the issue, then try the following steps:

1. In the ODBC Data Source Administrator, make sure that there is a System DSN for Certificate Server, and that there is not a User DSN. If a User DSN exists, delete it.
2. In the **Certificate Server Service Configuration** found in **Services** under Control Panel, check to see if the service is running under the context of the System Account. If it is, try selecting **Allow Service to Interact with Desktop**.
3. Also in the **Certificate Server Service Configuration**, change the account to an account that has local administrative rights.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. In the **ODBC Data Source Administrator** on the **System DSN** tab, highlight the **Certificate Server DSN** and choose **Configure**. Select **Repair the Database**. The database name is *Certsrv.mdb*, and by default it is located in `%windnt%\system32\certlog`.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก จ

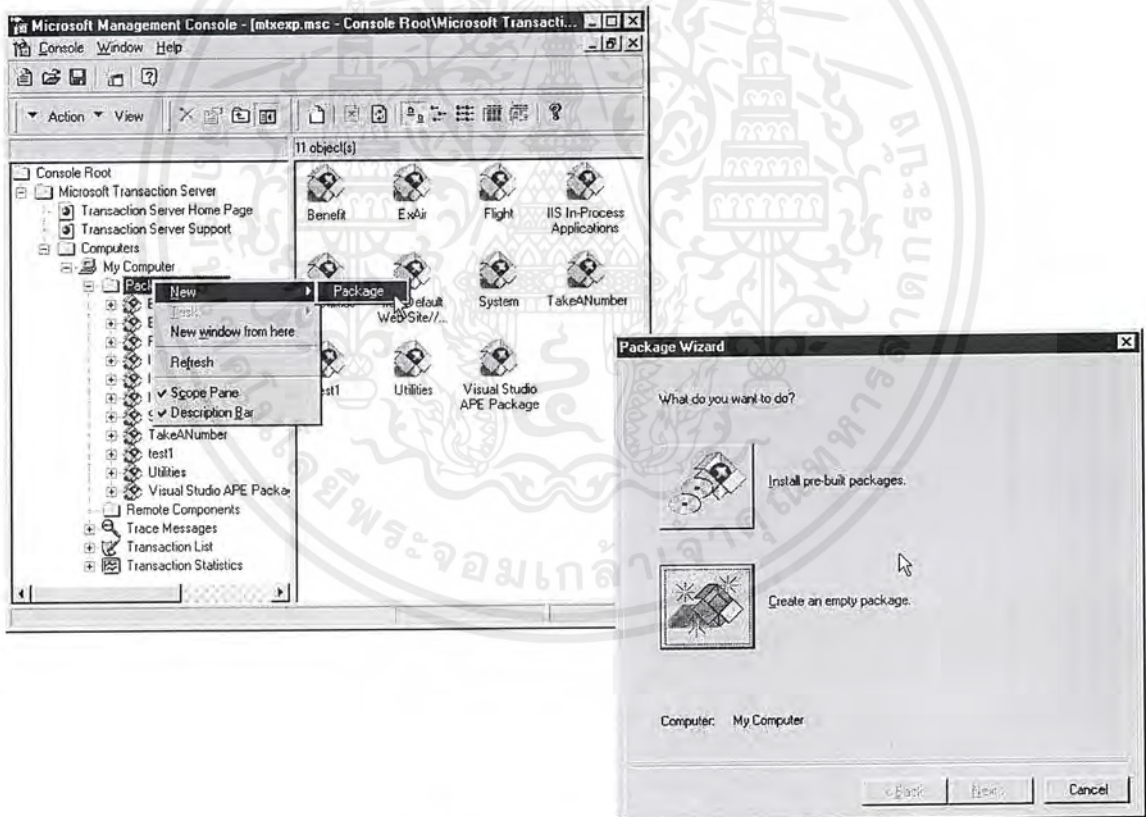
การติดตั้งคอมโพเนนต์บน Microsoft Transaction Server

ขั้นตอนการติดการติดตั้งคอมโพเนนต์บน Microsoft Transaction Server

ขั้นตอนที่ 1 เขียน COM component ที่ทำงานฝั่ง server โดยเลือกสร้างโปรเจกต์ชนิด ActiveX dll

ขั้นตอนที่ 2 นำเอา COM component ไป install ลงใน Microsoft Transaction Server (MTS)

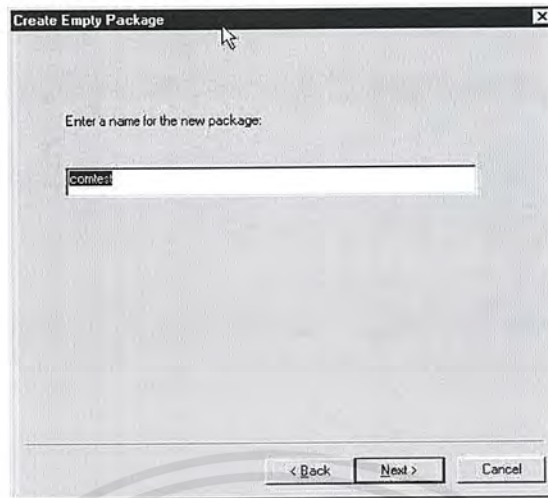
2.1 สร้างแพคเกจว่าง โดยเลือก create an empty package ดังรูป ข-1



รูปที่ จ-1 แสดงการเลือกสร้างแพคเกจใหม่

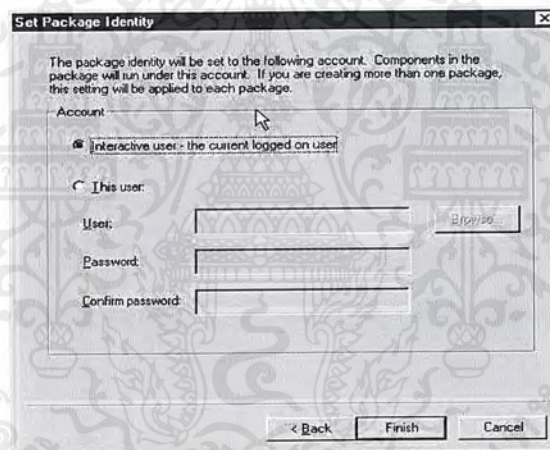
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 กำหนดชื่อแพคเกจที่ต้องการสร้าง



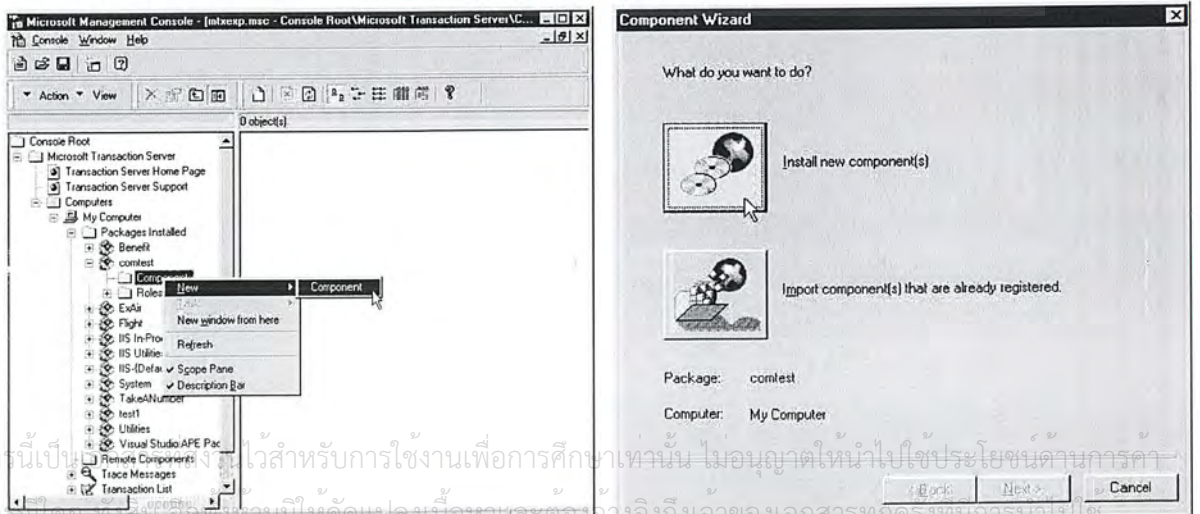
รูปที่ จ-2 แสดงการกำหนดชื่อแพคเกจ

2.3 กำหนด account ของการเรียกใช้คอมโพเนนต์ ในตัวอย่างนี้เลือก แบบ interactive user



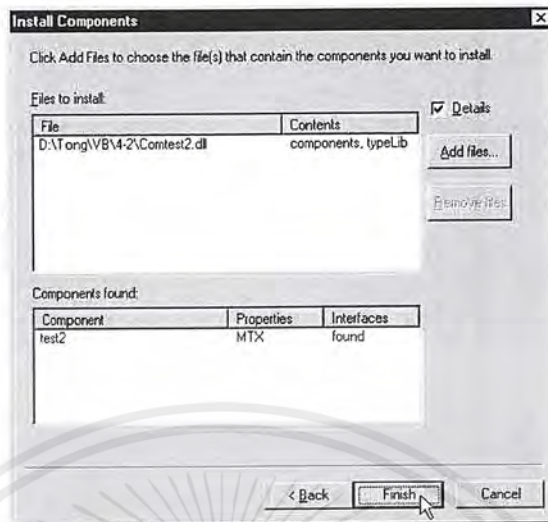
รูปที่ จ-3 แสดงการกำหนด account

2.4 หลังจากสร้าง package แล้วจะต้องเพิ่มคอมโพเนนต์ลงไปในแพคเกจ โดยเลือก install new component ดังรูป ข-4



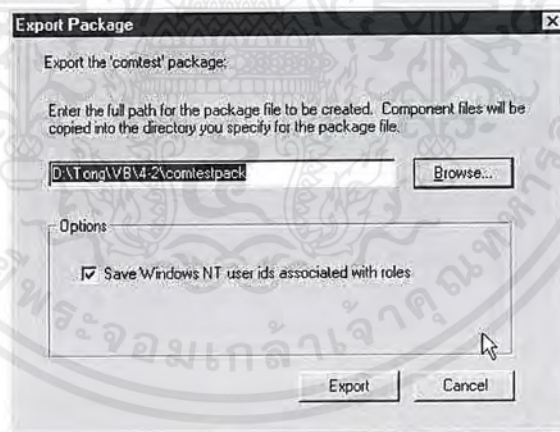
รูปที่ ข-4 แสดงการเพิ่มคอมโพเนนต์ลงไปในแพคเกจ

2.5 กำหนด path ของไฟล์ .dll



รูป จ-5 แสดงการกำหนด Path ของคอมโพเนนต์ที่เป็นไฟล์ .dll

2.6 จากนั้นจะต้องสร้างไฟล์ติดตั้งเพื่อนำไปติดตั้งที่ฝั่งไคลเอนต์โดยกดคลิกขวาที่ชื่อของแพคเกจ แล้วกด export จะปรากฏดังรูป แล้วเลือก path ที่ต้องการสร้างไฟล์ติดตั้ง



รูปที่ จ-6 แสดงการเลือก Path ที่ต้องการสร้างไฟล์ติดตั้ง

ขั้นตอนที่ 3 การเขียนฟอร์มฝั่งไคลเอนต์โดยจะต้องอ้างอิงคอมโพเนนต์ที่สร้างไว้ฝั่งเซิร์ฟเวอร์

ขั้นตอนที่ 4 นำไฟล์ติดตั้งที่ได้จากการ export และ ไฟล์แอปพลิเคชัน .exe ที่คอมไพล์จากฟอร์มฝั่งไคลเอนต์ ไปรัน โดยจะต้องรันไฟล์ติดตั้งก่อนเพื่อบอกให้ไคลเอนต์รู้ว่าเซิร์ฟเวอร์อยู่ที่ใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดลักษณะของ MTS

Microsoft Transaction Server—Complete Features at a Glance	
Feature	Description and Benefits
Ease-of-Use	
Simplified Three-Tiered Application Development	MTS completely separates three-tiered infrastructure from presentation and application logic. This lets MTS developers build their applications as a collection of single-user COM components and then deploy components in the appropriate tier. Components that run on middle-tiered servers automatically support multi-user operation.
COM Component Support	MTS supports any tool that can produce COM DLLs, including Microsoft Visual Basic, Microsoft Visual C++ development system, and Microsoft Visual J++.
Simple Application Programming Interfaces	MTS developers have to learn only a simple set of APIs to deploy their components in the MTS run-time environment. Extensive knowledge of COM APIs and Win32 APIs is not required.
Support for Component Packages	MTS lets developers combine related components into packages. Packages provide an easy way to secure, manage, and deploy components as a group.
MTS Explorer	The MTS Explorer is a GUI-based systems management console that makes it easy for administrators to create and manage packages, configure component properties such as security and transaction behavior, and monitor/manage operating servers. The MTS Explorer supports advanced ease-of-use features such as drag-and-drop editing and multiple item selection.
Microsoft Management Console Support	The MTS Explorer management interface can run as a snap-in to the Microsoft Management Console (MMC). MMC integration provides users with a consolidated management environment with a consistent look and feel.
Automatic Client Installation Utility	MTS provides a utility that simplifies and automates the process of configuring client machines that need to access MTS-based components running on a server.
Comprehensive Functionality	
Automatic Transaction Management	MTS provides automatic transaction management services to components. Developers or administrators use simple <i>property page</i> interfaces to specify what level of transaction protection is required by a given component. At run time, MTS performs all required transaction management automatically. This dramatically simplifies development and facilitates component reuse.
Database Connection Pooling	MTS allocates connections to database resources from a pool. Components only require connections when they are active. This lowers the total number of database connections required to support clients, increasing database performance.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Microsoft Transaction Server—Complete Features at a Glance

Feature	Description and Benefits
Comprehensive Functionality (Cont.)	
Transactional ODBC Driver for Oracle and DB2 Databases	MTS includes an ODBC 3.0-compliant driver for Oracle version 7.3 and later and DB2. ODBC 3.0 compliance lets developers build MTS applications that access these databases (on any platform), along with access to other resources such as SQL Server and Microsoft Message Queue Server, in transactional units of work for full data integrity protection. ODBC 3.0 compliance also lets MTS perform connection pooling of Oracle and DB2 database sessions for improved performance.
Support for Multiple Databases and Resource Managers	Components running under MTS can access multiple databases and other resources such as message queuing systems and mainframe applications, simultaneously and with full data integrity protections. MTS automatically manages transactions so that all databases and other resources included in a request commit or abort as a unit.
No MTS Client Footprint	MTS applications require no libraries or run-time environment on client machines beyond that provided by COM and Distributed COM (DCOM).
Windows NT Workstation and Windows 95/98 Support	MTS supports operation on Windows NT Workstation and Windows 95/98 so that companies can deploy stand-alone versions of their MTS applications.
Process Isolation	MTS Administrators can configure packages to run in their own process to ensure that failures do not spread to other packages.
Automatic Thread Pooling	As requests come from clients, MTS automatically assigns threads to components from a pre-allocated pool. When a component finishes executing, MTS reclaims the thread. This reduces the overhead of thread creation/deletion for better performance.
Automatic Object Instance Management	MTS creates instances of components on the server only when requested and reclaims memory resources when the component finishes executing. This form of <i>just-in-time (JIT) activation</i> and <i>as-soon-as-possible deactivation</i> minimizes memory requirements on server machines.
Shared Property Manager	MTS provides the Shared Property Manager to simplify programming of components that need to save state information or share state with other components.
Integration with Windows NT Features and Microsoft Products	
Integration with Microsoft SQL Server 6.5	Microsoft SQL Server 6.5 provides an ODBC 3.0 driver that provides extremely fast and efficient transactional integration, based on Microsoft Distributed Transaction Coordinator (DTC) to MTS applications. Because MTS also uses DTC for transaction management, administrators can use the same utilities to manage transactions between MTS and SQL Server, simplifying systems administration.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Microsoft Transaction Server—Complete Features at a Glance

Feature	Description and Benefits
Integration with Windows NT Features and Microsoft Products (Cont.)	
IIS Integration	Microsoft Internet Information Server version.4.0 is integrated with MTS and uses MTS for many run-time services such as transaction management. MTS integration makes it easy for IIS Active Server Pages to call MTS components that access databases, mainframe applications, and message queues with full data integrity protection. MTS integration also provides IIS with process isolation to prevent individual failures from affecting other parts of a Web site, enhanced run-time services such as thread and connection pooling for better performance, and easier component management.
MSMQ Integration	MTS provides integration with Microsoft Message Queue Server (MSMQ). MSMQ integration lets MTS-based applications communicate in a reliable, loosely coupled fashion. MSMQ operations (such as send and receive) enlist in MTS transactions automatically for data integrity protection
SNA Server 4.0 Integration	MTS can use Microsoft SNA Server 4.0 and the COM Transaction Integrator to include mainframe applications (e.g., running under CICS and IMS) and their data in transactions managed by MTS. MTS-based applications can update one or more databases, send or receive messages with MSMQ, and invoke a mainframe application that updates data (e.g., in VSAM), and all updates, including those on the mainframe, will commit or abort as a whole. Microsoft SNA Server 4.0 also makes it easy to generate COM-based interfaces to mainframe applications for simplified MTS programming.
Support for Windows NT Clustering Services	MTS supports operation on clustered Windows NT Servers. Clustering enables administrators to configure MTS packages for automatic failover and fault-tolerant, high-availability operation.
Windows NT Security Integration	MTS provides a <i>role-based</i> security mechanism that is integrated with the Windows NT security environment. Roles let developers design security protections into their components without knowing how the components will be ultimately deployed. Administrators map roles to users when they install components into MTS.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

หนังสืออ้างอิง

- [1] David Chappell, “*Understanding ActiveX AND OLE*”, Microsoft Press, 1996.
- [2] Rob Thayer, “*Visual Basic 6 Unleashed*”, First Edition, Sams Publishing, 1998.
- [3] Guy Eddon and Henry Eddon, “*Programming Components with Microsoft Visual Basic 6.0*”, Microsoft Press, 1998.
- [4] Marco Cantu, “*Mastering Delphi 4*”, Sybex, 1998.
- [5] สัจจะ จรัสรุ่งรวีวร, “คู่มือการสร้างแอปพลิเคชันด้วย *Visual Basic 6 Basic & Advance*”, Infopress, 2542
- [6] กนก กุศลมาลย์กุล, ไกรวุฒิ มั่นเสถียรสิน, “คู่มือการเขียนโปรแกรม *Delphi 4*”, สำนักพิมพ์ ชัคเชส มีเดีย,
- [7] ธาริน สิทธิธรรมชารี, “คู่มือการเขียนโปรแกรม *Microsoft Visual Basic version 6.0*”, สำนักพิมพ์ ชัคเชส มีเดีย
- [8] Bruce Powel Douglass, “*Real-Time UML: Developing Efficient Objects for Embedded Systems*”, Addison-Wesley, 1998
- [9] Hans-Erik Eriksson, Magnus Penker, “*UML Toolkit*”, WILEY Computer Publishing, 1998
- [10] John Clark Craig, Jeff Webb, เรียบเรียงโดย ชัชวาล ศุภเกษม, “*การเขียนโปรแกรมบนวินโดวส์ ด้วย Microsoft Visual Basic 6.0 ภาคปฏิบัติ*”, บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน), 2542
- [11] กิตติภูมิ วรรณิตร, “*เพิ่มพลังอินเทอร์เน็ตที่ไฟให้เว็บเพจด้วย ASP*”, บริษัท วิตตี้ กรุ๊ป จำกัด”, 2542
- [12] พิพัฒน์ ยอดพฤติกการ, “*เริ่มต้นธุรกิจกับอินเทอร์เน็ต*”, บริษัท วิตตี้ กรุ๊ป จำกัด”, 2542
- [13] Ash Rofail, Tony Martin, “*Building N-Tier Applications with COM and Visual Basic 6.0*”, (Wiley Computer Publishing), 1999
- [14] นางสาวจันทร์กานต์ จุหอม, นางสาวทิพย์เนตร แก้วปัดตะ, “*การพัฒนาโปรแกรมประยุกต์ในเชิง วัตถุ*”, 2541

เว็บไซต์อ้างอิง

- [15] <http://www.activex.org>
- [16] <http://www.microsoft.com/activex>
- [17] <http://www.microsoft.com/com>
- [18] <http://www.msdn.microsoft.com>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้